



Norwegian University
of Life Sciences

Master's Thesis 2021 30 ECTS
Faculty of Science and Technology

Development and Structural Analysis of a Multirotor Support Structure

Vetle Birkeland Aass
Mechanical Engineering

Development and Structural Analysis of a Multirotor Support Structure

Submitted in partial fulfillment of the requirements for the degree of

Master in Mechanical Engineering

by

Vetle Birkeland Aass

May, 2021

Acknowledgements

I would like to express my very great appreciation to my supervisor, **Prof. Tor Anders Nygaard**, for the several fruitful discussions regarding my work. Thank you for helping me understand the various aspects of wind turbines.

Thanks to **Associate Prof. Geir Terjesen** for insightful feedback on fatigue analysis.

I like to acknowledge the support and assistance rendered by **my fellow students** in several ways throughout my research work.

Finally, I would like to extend my gratitude to **my family** for their continuous encouragement and loving support during the writing of this thesis.

Place: Ås

Date: 31/05/2021

Vetle Birkeland Aass

ABSTRACT

Continuous advancement in wind turbine technology is a key factor in driving down the cost of energy from wind power. One idea that has been evaluated is the concept of having multiple rotors on a single support structure. This thesis investigates the structural integrity of a proposed design for a single rotor support in a multi rotor wind turbine. The support structure will be analyzed featuring both a two-bladed rotor as well as a four-bladed rotor. The structural analyses include fatigue, buckling and Von Mises stress assessments. The aim is to adjust the dimensions of the structural members such that they are in accordance with the design requirements.

The structure is modelled and analyzed using the 3DFloat simulation tool. Developed by the Wind Energy Department at the Institute for Energy Technology, 3DFloat is an aero-servo-hydro-elastic Finite-Element-Method code tailored for simulations of offshore structures in general, and offshore wind turbines in particular. Post-processing is executed using the Python Programming Language.

Fatigue assessments are carried out in accordance with the DNVGL-RP-C203. Buckling calculations are according to EN 1993-1-1 (2005):Eurocode 3: Design of steelstructures - Part 1-1: General rules and rules for buildings practice. The turbine's structural response is investigated for normal and extreme wind conditions.

The results showed that the support structure featuring two blades experienced a higher vulnerability to fatigue. In extreme conditions the two structures exhibit similar structural responses to eachother.

The analyses culminated in the design of two support structures which have been verified against buckling and the Von Mises stress criterion, as well as being designed to have a fatigue life of more than 30 years.

Keywords: *Multi Rotor Wind Turbine, Fatigue, Buckling, Von Mises, DNVGL-RP-C203, EN 1993-1-1 (2005):Eurocode 3.*

SAMMENDRAG

Kontinuerlig utvikling av vindturbinteknologi er en nøkkelfaktor i å drive ned prisen på energi fra vindkraft. Én idé som har blitt evaluert er konseptet som innebærer å ha flere rotorere på én enkelt bærende struktur. Denne avhandlingen undersøker den strukturelle integriteten i en foreslått støttestruktur for én slik rotor i en mult rotor vindturbin. Støttestrukturen vil bli analysert for både to og fire blader. Den strukturelle analysen inkluderer utmattning, knekking og Von Mises spenningsevaluering. Målet er å justere dimensjonene på konstruksjonsdelene slik at de er i tråd med designkravene.

Strukturen er modellert og analysert ved å bruke simuleringsverktøyet 3DFloat. 3DFloat er en aero-servo-hydro-elastic Finite-Element-Method kode skreddersydd for simuleringer av offshore strukturer generelt, og offshore vindturbiner spesielt, utviklet på Instituttet for Energiteknikk. Postprosessering er gjennomført i programmeringsspråket Python.

Utmattingsberegninger er gjennomført i tråd med DNVGL-RP-C203. Knekkeregninger er i samsvar med EN 1993-1-1 (2005):Eurocode 3: Design of steelstructures -Part 1-1: General rules and rules for buildings practice. Turbinens strukturelle respons er undersøkt for både vanlige og ekstreme vindhastigheter.

Resultatene viste at strukturen hadde kortere utmattingslevetid når den opererte med to blader. I ekstreme vindforhold utviser strukturene lik strukturell respons med hverandre.

Analysen kulminerte i dimensjoneringen av to støttestrukturer som har blitt verifisert mot knekking og Von Mises spenningsberegninger, i tillegg til å være dimensjonert for å ha mer enn 30 års utmattingslevetid.

Table of Contents

Acknowledgements	i
Abstract	ii
Table of Contents	iv
List of Figures	vii
List of Tables	ix
List of Abbreviations	x
Nomenclature	xiii
1 Introduction	1
1.1 Basics of Wind Turbines	2
1.1.1 Momentum Theory, the Betz Limit and Parametric Sensitivity	3
1.1.2 Aerodynamic Forces	4
1.1.3 Tip-speed ratio and Number of Blades	5
1.1.4 Power Curve	5
1.1.5 Offshore Windpower	6
1.2 Scope and Objectives	7
2 Theory	8
2.1 Fatigue Theory	8
2.1.1 Axial Stresses in Combined Loading Situations	8
2.1.2 Fatigue	9
2.1.3 Cyclic Loading	9
2.1.4 S-N Curve (Stress Versus Life)	9
2.1.5 Variable Loading and Miner-Palmgren	10
2.1.6 Irregular Loading and Rainflow Cycle Counting	11
2.2 Buckling Theory	12
2.2.1 Euler's Critical Load	13
2.2.2 Buckling Lengths and Slenderness Ratio	14

2.3	The Von Mises Yield Criterion	15
3	Methodology	16
3.1	The Structure	16
3.2	The Simulation Tools	18
3.2.1	3DFloat	18
3.2.2	Visualization and Post-Processing	18
3.2.3	Input - 3DFloat simulation	18
3.3	Fatigue Calculations	20
3.3.1	Points of interest	20
3.3.2	DNV GL Recommended Practice	20
3.4	Buckling Calculations	31
3.4.1	Eurocode	31
3.4.2	Buckling Algorithm	34
3.5	Von Mises Equivalent Stress Assessment	34
3.6	Adjusting Cross-Section Parameters	35
4	Results	36
4.1	Fatigue	37
4.2	Preliminary adjustment of Cross-sectional Properties	38
4.2.1	2-bladed	38
4.2.2	4-bladed	38
4.3	Extreme wind Von Mises Analysis	39
4.3.1	2-bladed	39
4.3.2	4-bladed	39
4.4	Final Adjustments and verifying against buckling	39
4.5	Comparison and Summary	42
5	Discussion	43
5.1	Dynamic Behavior, Structural Response, and Fatigue Life Assessment . . .	43
5.2	Von Mises stress Assessment	44
5.3	Buckling Assessment	44
5.4	Validity of the Results	44

6	Conclusion	46
6.1	Further Work	46
	References	46
 Appendices		
Vedlegg A	Python Routine For Partial Fatigue Damage Calculation	50
Vedlegg B	Python Routine For Calculating stresses around the cir- cumference of a CHS	55
Vedlegg C	3DFloat input	58
	C.0.1 2-bladed rotor	58
	C.0.2 4-bladed rotor	72
Vedlegg D	Python Routine For Assessing Occurences Of Buckling	86
Vedlegg E	Fatigue Life vs. Wall Thickness	92
Vedlegg F	Stress histories from Fatigue Life Assessment	102
Vedlegg G	Python Routine for Assessing Fatigue Life in the Structure and element stress history generation	110
Vedlegg H	Python Routine for Assessing Von Mises Stresses at Extreme Wind Conditions	187

List of Figures

1.1	Major components of a conventional wind turbine	2
1.2	The different orientations of a HAWT	2
1.3	Parametric definitions related to a turbine blade in motion	4
1.4	$C_P - TSR$ curve for NREL offshore 5 MW baseline wind turbine at pitch angle (β) of 0	5
1.5	Key points of a wind turbine's power curve	6
2.1	Cylinder subject to combined loading	8
2.2	Constant amplitude cycling and the associated nomenclature.	9
2.3	Best-fit curves for the S-N data points [10]	10
2.4	Illustration of an S-N curve with the associated nomenclature [11]	10
2.5	Variable amplitude loading	11
2.6	Relating different stress ranges to number of cycles to failure on the S-N curve	11
2.7	Example of an irregular loading situation with relevant definitions for the rainflow counting method.	11
2.8	Definition of a cycle in Rainflow Cycle Counting with related definitions	12
2.9	Procedure of Rainflow Cycle Counting	12
2.10	Side by side comparison between a column before and after buckling	13
2.11	Column supports and associated buckling lengths	14
2.12	Plane stress for a cylinder [12]	15
3.1	A single MRWT module featuring two blades	16
3.2	A single MRWT module featuring four blades	16
3.3	Elements in the table labelled according to table 3.1	17

3.4	Elements in the table labelled according to table 3.1	17
3.5	Wind speed as a function of time normal to the rotor at normal wind conditions	19
3.6	Wind speed as a function of time normal to the rotor at extreme wind conditions	19
3.7	Geometrical definitions for tubular joints	22
3.8	Superposition of stresses around the circumference due to combined loading	23
3.9	Geometrical definitions of a hollow cylinder's cross-section	24
3.10	Definitions of geometrical parameters for K-joints	25
3.11	Definitions of geometrical parameters for T-joint	25
3.12	Example of a Type 1 joint	27
3.13	Example of a Type 2 joint	28
3.14	Example of a Type 3 joint	29
3.15	Example of a Type 4 joint	30
4.1	Stress distribution for extreme wind conditions from ParaView	36
E.1	Fatigue life vs wall thickness images	92
F.1	Stress history images	102

List of Tables

3.1	Main elements of the support structure and their geometric data	17
3.2	Properties of structural steel s355	17
3.3	Class T S-N curve data from DNVGL-RP-C203	21
3.4	Class B1 S-N curve data from DNVGL-RP-C203	22
3.5	Classification of CHS according to EC3	31
3.6	Assessment of buckling curve for CHS members	32
3.7	Relationship between buckling curve and imperfection factor α	32
3.8	Buckling related properties of the structure's front struts	32
3.9	More buckling related properties of the structure's front struts	34
4.1	Assessed fatigue life for the structure's elements	37
4.2	Element properties and estimated fatigue life after 1st stage adjustments	38
4.3	Element properties and estimated fatigue life after 1st stage adjustments	38
4.4	Element properties and Von Mises stress after 1st stage adjustments . . .	39
4.5	Element properties and Von Mises stress after 1st stage adjustments . . .	39
4.6	Buckling assessment for 2-bladed structure in normal conditions	40
4.7	Buckling assessment for 2-bladed structure in extreme conditions	40
4.8	Complete overview for the 2nd stage adjusted 2-bladed structure	40
4.9	Buckling assessment for 4-bladed structure in normal conditions	41
4.10	Buckling assessment for 4-bladed structure in extreme conditions	41
4.11	Complete overview for the 2nd stage adjusted 4-bladed structure	41
4.12	Summary and comparison between 2-bladed and 4-bladed rotor support structure	42

List of Abbreviations

- HAWT - Horizontal Axis Wind Turbine
- RNA - Rotor-Nacelle Assembly
- TSR - Tip Speed Ratio
- CHS - Circular Hollow Section
- IFE - Institute for Energy Technology
- CJP - Complete Joint Penetration
- NREL - National Renewable Energy Laboratory

Nomenclature

α	Parameter describing the relationship between the chord length and the chord diameter
α_i	Imperfection factor
$\bar{\lambda}$	Relative slenderness ratio
β	Parameter describing the relationship between the brace diameter and the chord diameter
χ	Reduction factor for the associated buckling mode
$\Delta\sigma$	Stress range in a cyclic stress curve with constant amplitude
$\Delta M_{y,Ed}$	Moments about the y-axis caused by a shift in the neutral axis
$\Delta M_{z,Ed}$	Moments about the z-axis caused by a shift in the neutral axis
η	Combined efficiency of the drive train and the generator
γ	Parameter describing the relationship between the chord diameter and the chord wall thickness
λ	Slenderness ratio
λ_{tsr}	Tip-speed ratio
ω	Angular velocity of the blade
ρ	Density of air
σ_i	Stress range in stress block i
σ_m	Mean stress in a cyclic stress curve with constant amplitude
σ_x	axial stress
σ_{eq}	Equivalent Von Mises stress
$\sigma_{hot\ spot}$	Hot spot stress
σ_{max}	Maximum stress in a cyclic stress curve with constant amplitude
σ_{min}	Minimum stress in a cyclic stress curve with constant amplitude
σ_{my}	Stress induced due to bending about the y-axis
σ_{mz}	Stress induced due to bending about the z-axis
$\sigma_{nominal}$	Stress before multiplying with the stress concentration factor

σ_{tot}	Resultant axial stress at a point due to axial/bending forces
σ_{XY}	Stress range between the X and Y point in a X-Y-Z sequence in a stress history plot
σ_{YZ}	Stress range between the Y and Z point in a X-Y-Z sequence in a stress history plot
τ	Parameter describing the relationship between the brace wall thickness and the chord wall thickness
τ_{xy}	Shear stress
θ	Angle between the brace and the chord
ζ	Parameter describing the relationship between the chord diameter and the distance between two braces joined together with the same chord
A	Area swept by the rotor
A_{cs}	Cross-section area for a cylinder
C	Describes the interception point between the log N-axis and the S-N curve
C_p	Power coefficient
C_T	Thrust coefficient
C_m	Equivalent uniform moment factor
D	Accumulated fatigue damage
D_i	Inner diameter of a cylinder
E	Modulus of elasticity
f_u	The material's ultimate strength
f_y	The material's yield strength
G	Shear modulus
i	Radius of gyration
I_y	Moment of inertia about the z-axis
I_z	Moment of inertia about the z-axis
I_{CHS}	Moment of inertia for a circular hollow section
I_{min}	Least moment of inertia for a cross-section
K	Buckle length factor
k	Thickness exponent
k_b	Number of stress blocks
$k_{yy}, k_{yz}, k_{zy}, k_{zz}$	Interaction factors
L	Length of structural member

L_e	Effective length of a column
$\log \bar{a}$	Intercept between the S-N curve and the log N-Axis for an S-N curve with a downshift of two standard deviations
m	Inverse of the S-N curve slope
$M_{by,Rd}$	Design buckling resistance moment about the y-axis
$M_{bz,Rd}$	Design buckling resistance moment about the z-axis
$M_{y,Ed}$	Design value of the moments about the y-axis
$M_{z,Ed}$	Design value of the moments about the z-axis
N	Number of stress cycles in a cyclic stress curve with constant amplitude
N_{Ed}	Design value for the compression force
N_i	Number of stress cycles in block i
$N_{b,Rd}$	Design buckling force of the compression member
N_{fi}	Number of cycles to failure at σ_i
P	Axial Force
P_t	Extracted power from the wind by the rotor
P_{cr}	Euler's critical load
P_{el}	Electrical power generated by the turbine
R	Radius of the rotor
R_i	Inner radius of a cylinder
$s_{\log N}$	One standard deviation of log N
SCF_{AC}	Stress concentration factor at the crown
SCF_{AS}	Stress concentration factor at the saddle
SCF_{MIP}	Stress concentration factor for in-plane-bending of the brace
SCF_{MOP}	Stress concentration factor for out-of-plane bending of the brace
T	Thrust force on the rotor
t	Thickness of the material
t_{ref}	Reference thickness of the material according to DNVGL-RP-C203
u_d	Distortion energy
$u_{y,d}$	Distortion energy at yielding
v	Free stream wind speed
ν_p	Poisson's ratio
y	Distance in y-direction
z	Distance in z-direction

1. Introduction

Recent years have seen a rising global awareness surrounding the need for environmentally sustainable energy production. As a result, wind power conversion systems are emerging as an alternative to conventional energy sources. Wind offers an abundant and renewable source of energy. Harnessing the power provided by the wind can be a solution to concerns regarding climate change and fossil fuel depletion.

As of 2019, China leads the world's wind power production having an installed capacity of 237 GW [1], followed by The United States and Germany hosting 105 GW and 61 GW respectively. The 60.4 GW of new installations brought the global cumulative wind power capacity to 651 GW [2]. With its total installed capacity of 205 GW, wind power accounted for 15% of the EU's electricity demand [3].

Continuous advancements in wind turbine technology is a key factor in driving down the cost of energy from wind power. One idea that has contributed to more cost-effective wind farms is the development of bigger wind turbines with a higher rated power. The 13 MW Haliade-X is currently the most powerful turbine in operation, while the EU funded UpWind project have investigated the feasibility of a 20 MW turbine at a conceptual level. However, due to various challenges related to further upscaling, the turbine growth has slowed down dramatically in recent years.

Considering the limitations of the current design, it may be necessary to develop new architectures to further decrease the cost of producing electricity. One such design that has been evaluated is the concept of having multiple rotors on a single support structure.

This thesis investigates the structural integrity of a proposed design for a single rotor support in a multi rotor wind turbine (MRWT).

In the following sections, the fundamentals of wind turbine technology will be presented. Some important relationships will be introduced, and we will get familiar with terminology that will be used throughout the thesis. Unless otherwise stated, [4] is used as a source for this chapter.

1.1 Basics of Wind Turbines

In an attempt to optimize the wind turbine design, many different concepts have been explored. The Horizontal Axis Wind Turbine (HAWT) is by far the most widely used. Such a turbine is characterized by having an axis of rotation parallel to the ground.

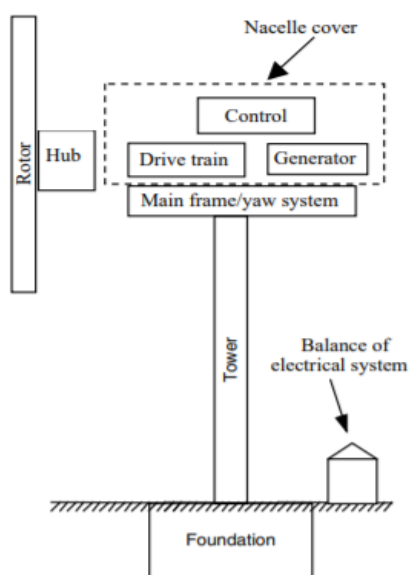


Figure 1.1 Major components of a conventional wind turbine

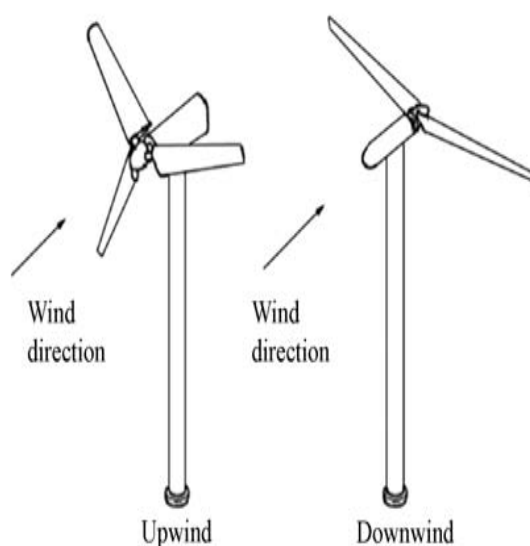


Figure 1.2 The different orientations of a HAWT

The rotor blades are made of airfoil sections which produces aerodynamic lift when interacting with the oncoming airflow. The torque generated by the blades is transmitted to the drive train inside the nacelle. Rotational motion in the drive train is then finally converted to electrical energy in the generator, which is fed into the electrical grid, or stored in batteries. The rotor, hub, and the nacelle, with all its internal components, is referred to as the rotor-nacelle assembly (RNA). Another major subsystem of the HAWT is the support structure which includes the foundation, tower, and yaw system. The yaw system allows for the RNA to be oriented in the direction of the wind, as a mean to maximize the produced power.

HAWTs are further categorized based on a great number of different design variations. The rotor can be placed on the upwind or downwind side of the tower, as illustrated in figure 2, and can feature a different number of blades. Most modern turbines have upwind rotors with three blades.

In extreme wind conditions, turbines are subject to enormous forces that can lead to structural failure. They are designed to withstand extreme winds, but only when the blades are not spinning. At a certain threshold wind speed (cut-out wind speed) brakes will therefore bring the turbine to a halt. Pitch-regulated turbines have an active control system that can turn the blades around their own axis. If necessary, blades can be pitched in such a direction that the aerodynamic lift is reduced, to prevent excessive loading. Consequently, such turbines can operate even at relatively high wind speeds, that would otherwise not be possible.

1.1.1 Momentum Theory, the Betz Limit and Parametric Sensitivity

By applying momentum theory, a simple model quantifying the extracted power, P_t , from the wind can be derived.

$$P_t = \frac{1}{2} \rho \pi R^2 v^3 C_p \quad (1.1)$$

Where ρ and v are the wind's density and speed respectively, and R is the radius of the turbine. C_p is referred to as the power coefficient.

The power coefficient is simply the ratio between the power available in the wind passing through the turbine's swept area and the power extracted by the turbine. The maximum theoretical value for C_p is $16/27$ or 59.26% and is known as the Betz limit. In practice, however, there is a decrease in the maximum achievable power coefficient due to:

- rotation of the wake behind the rotor
- finite number of blades
- non-negligable aerodynamic drag on the rotor

As seen from equation 1.1, rotor power is proportional to the freestream wind cubed, the rotor diameter squared, and the power coefficient. The cubic dependency on the freestream wind makes it the primary driver of power in the rotor. A doubling of the freestream wind speed results in a rotor power eight times higher compared to that of the baseline. This emphasizes the importance of rotor location assessment. It also explains the motivation behind taller turbines, as wind speeds generally increases with height above ground level.

1.1.2 Aerodynamic Forces

The turbine blades are of the most fundamental components in a wind turbine. Due to their shape, they are able to convert the power in the wind to rotational mechanical power in the turbine shaft. As a turbine blade spins, it experiences a relative wind speed which is the vector sum of the free stream wind and the blade's rotational speed.

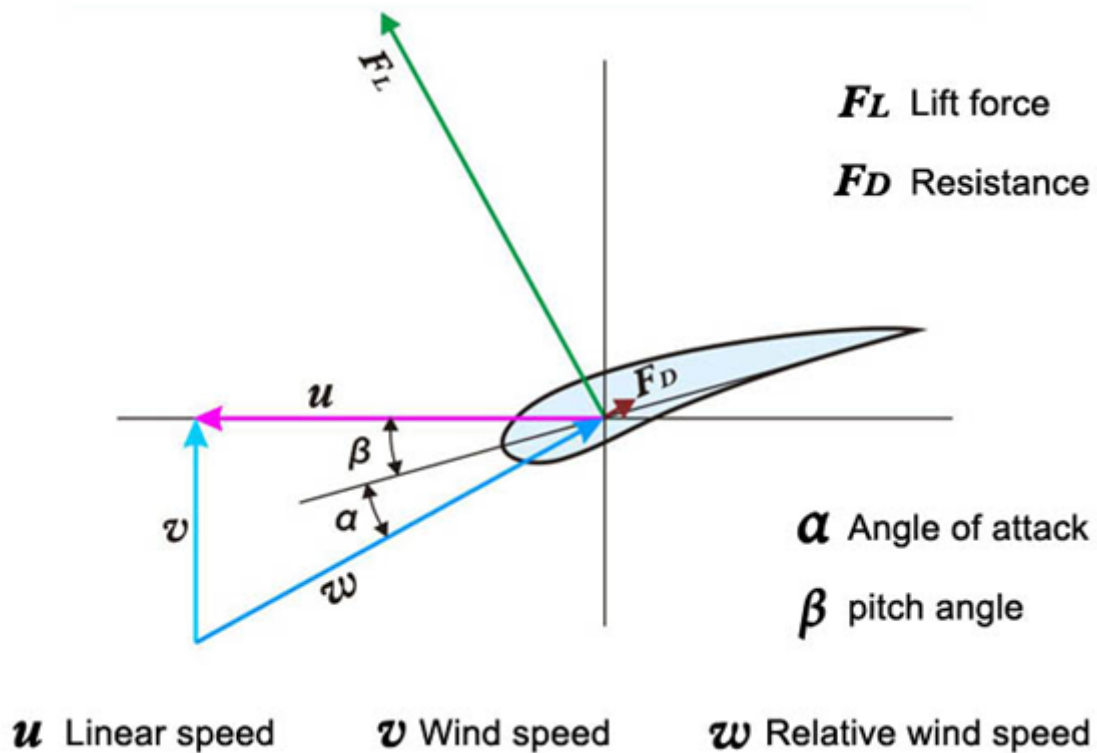


Figure 1.3 Parametric definitions related to a turbine blade in motion

As seen from figure 1.3, the **lift force** acting on the turbine blade is perpendicular to the relative air flow, while the **drag force** is parallel to it. In other words, both the direction and the magnitude of these forces are dictated by not only the wind speed, but also the rotational speed of the rotor. While the interactions between the turbine blade and the wind generates a force in the direction of rotation for the rotor, there's also a force acting on the turbine in the windward direction, called **thrust**. Similarly to the power, the thrust on a wind turbine can be characterized by a non-dimensional thrust coefficient:

$$C_T = \frac{\text{Thrust force}}{\text{Dynamic force}} = \frac{T}{\frac{1}{2}\rho v^2 A} \quad (1.2)$$

where T is the thrust force acting on the turbine, and A is the rotor's swept area.

Given all other constraints, the aim is generally to keep the rotor thrust at a minimum as it does not contribute to any power generation. It also creates a massive bending moment in the structure, which must be balanced by the tower and foundation.

1.1.3 Tip-speed ratio and Number of Blades

A rotor that rotates slowly will allow wind to pass unperturbed through the gaps between the blades. On the other hand, a rapidly rotating rotor will appear as a solid wall to the oncoming wind. Tip speed ratio (TSR) is an important design parameter that relates the tangential speed of the tip of a blade and the free stream wind speed:

$$\lambda_{tsr} = \frac{\omega R}{v} = \frac{\text{tip speed of blade}}{\text{free stream wind speed}} \quad (1.3)$$

Where ω is the angular velocity of the tip speed, R is the rotor radius, and v is the free stream wind speed.

For every rotor design there is an optimal TSR at which the turbine should operate in order to extract as much energy from the wind as possible. The optimal TSR is dependent on both blade design and the number of blades. Figure 4 shows the NREL offshore 5 MW baseline wind turbine's power coefficient as a function of TSR. The optimal TSR is where the value of C_p is at its highest.

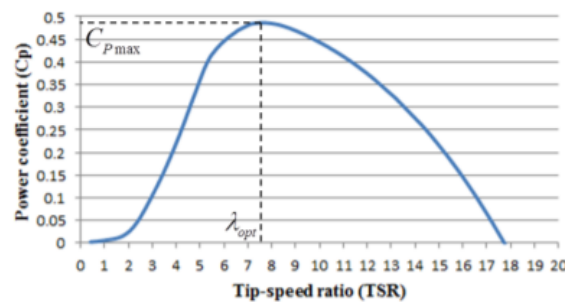


Figure 1.4 $C_p - TSR$ curve for NREL offshore 5 MW baseline wind turbine at pitch angle (β) of 0

[5]

1.1.4 Power Curve

The power coefficient, C_p , is a measure of how efficiently the given turbine extracts energy from the passing wind. By introducing an efficiency factor, η , describing the combined efficiency of the drive train and the generator, equation 1.1 can be altered to express the overall performance of the turbine:

$$P_{el} = \eta \frac{1}{2} \rho \pi R^2 v^3 C_p \quad (1.4)$$

Where P_{el} is the electrical power delivered from the turbine.

Every wind turbine has a characteristic power performance curve that gives the electrical power output as a function of hub height wind speed. With such a curve it is possible to predict the output of the turbine without considering the technical details of its many

components. A power curve for a hypothetical turbine is presented in figure 1.5. The performance of a given wind turbine relates to three key points on the wind speed scale:

- Cut in speed: the minimum wind speed at which the machine will deliver useful power
- Rated speed: the wind speed at which the rated power (the maximum output power) of the electrical generator is reached
- Cut out speed: the maximum wind speed at which the turbine is allowed to deliver power, limited by engineering design and safety constraints

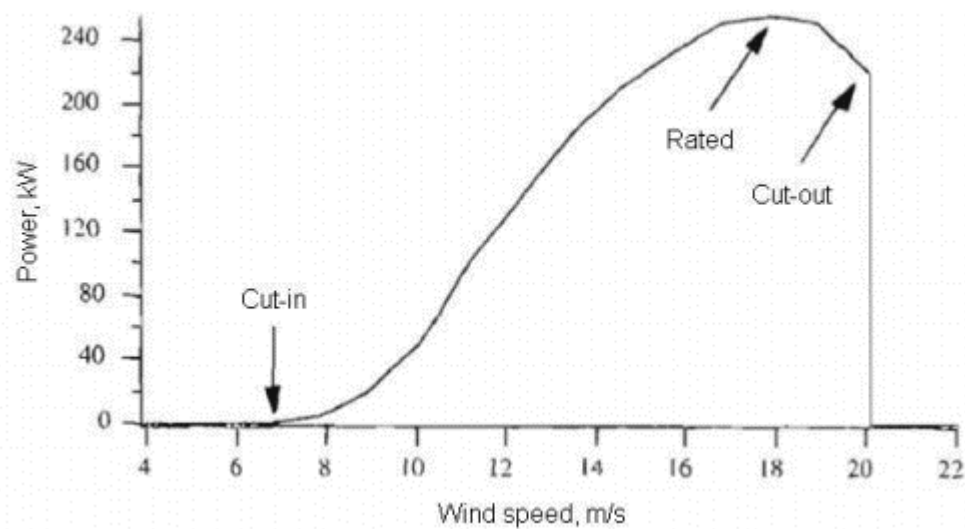


Figure 1.5 Key points of a wind turbine's power curve

1.1.5 Offshore Windpower

For a long time, wind turbines were exclusively land based structures. However, the last two decades have seen exponential growth in the installed offshore wind power capacity [6]. The most significant differences between land-based and offshore wind turbines are their support structures. While most current offshore wind farms employ fixed-foundation wind turbines in relatively shallow water, floating wind turbines are gaining increased attention in the industry. Floating wind turbine technology will unlock wind potential on deep waters and is currently in the early phase of development and deployment. The first commercial wind farm, Hywind Scotland, was commissioned in 2017 and hosts six 5 MW turbines.

Offshore wind energy has several promising aspects. These include:

- Availability of greater area for siting large projects

- Generally better wind conditions compared to onshore (i.e. higher wind speeds, less turbulence)
- Fewer considerations to be made in terms of environmental impact (i.e noise and visual pollution, urban encroachment of habitat)

Challenges related to offshore wind power include:

- Higher costs due to the necessity for specialized installation vessels and tools
- More complex and expensive support structures
- Necessity for corrosion protection
- Limited accessibility for maintenance

1.2 Scope and Objectives

Due to limited time and resources, the scope and objectives of this thesis is limited to:

- Develop routines for, and conduct, fatigue analysis on a support structure for a single rotor in a proposed MRWT design
- Verify the support structure against the Von Mises yield criterion
- Verify the support structure against buckling
- Modify the structural members such that its predicted fatigue life exceeds 30 years, and are designed to safely carry the expected loads of the wind turbine
- Make a comparison between the structural response of a two-bladed and a four-bladed rotor in both normal and extreme wind conditions

2. Theory

In this chapter, relevant theory related to the mechanics of materials will be covered. Firstly, fundamental concepts of the relationship between axial force, bending moments and stresses induced are presented. Secondly, appropriate fatigue concepts and relevant formulae will be discussed. Finally, the concept of buckling is addressed. Unless otherwise stated, [7] is used as the reference source for fatigue theory throughout this chapter, while [8] is used as source for the theory covered on buckling and stresses.

2.1 Fatigue Theory

2.1.1 Axial Stresses in Combined Loading Situations

When the cross-section of a member is subjected to combined loadings, meaning they experience different types of loading simultaneously, the stresses that develop can be determined using the method of superposition. Both axial forces and bending moments result in axial stresses in the member. Consider a member subjected to a bending moment about the y-axis, M_y , a bending moment about the z-axis, M_z , in addition to an axial force, P . The resultant stress, σ_{tot} , at a point (y, z) of the cross section can be calculated followingly:

$$\sigma_{tot} = \frac{P}{A} + \frac{M_z y}{I_z} + \frac{M_y z}{I_y} = \sigma_s + \sigma_{bz} + \sigma_{by} \quad (2.1)$$

where A is the area of the member's cross-section, I_y and I_z is the moment of inertia about the y and the z axis respectively, and σ_s , σ_{bz} , σ_{by} are the stresses induced due to P , M_z , and M_y respectively.

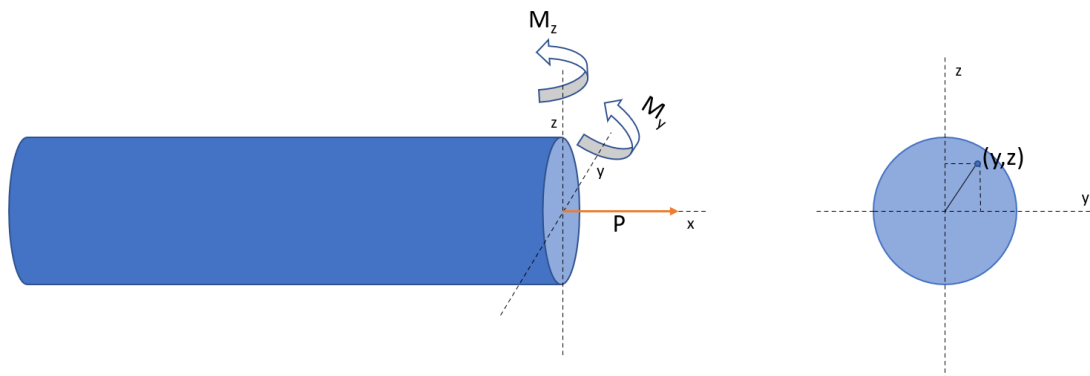


Figure 2.1 Cylinder subject to combined loading

2.1.2 Fatigue

Components which are subjected to loading which varies with time can fail at stress levels well below a material's ultimate strength. This phenomenon is known as fatigue. Rotor rotation combined with effects due to wind shear and wind turbulence causes varying loads on wind turbines. Fatigue accounts for the vast majority of mechanical engineering failures; estimations report as much as 90% [9, p. 243]. This emphasizes the importance of fatigue considerations in structural design. There are different approaches to calculating fatigue. In this thesis the stress-based approach is used.

2.1.3 Cyclic Loading

The simplest representation of a material experiencing loading which varies with time is a cyclic stress curve in which amplitude is held constant. The stress range, $\Delta\sigma$, is given as the difference between the maximum, σ_{max} , and minimum stress σ_{min} . The mean stress, σ_m , is given as the average between the maximum and minimum stress. Mathematically they can be expressed as such:

$$\sigma_m = \frac{\sigma_{max} + \sigma_{min}}{2} \quad (2.2)$$

$$\sigma_a = \frac{\Delta\sigma}{2} = \frac{\sigma_{max} - \sigma_{min}}{2} \quad (2.3)$$

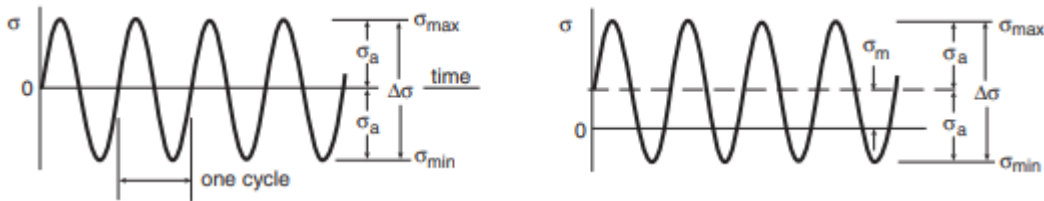


Figure 2.2 Constant amplitude cycling and the associated nomenclature.

Figure 2.2 demonstrates how the mean stress can differ even for two cases of cyclic loading with a similar stress range. In case (a) $\sigma_m = 0$. Case (b) represents a scenario in which σ_m is nonzero. A higher mean stress value is likely to affect the fatigue characteristics of a material and will typically result in a shorter fatigue life.

2.1.4 S-N Curve (Stress Versus Life)

S-N curves are used in order to predict the expected lifetime of an engineering component. Such curves are obtained through empirical data. Test specimen or engineering components are subjected to a large number of constant amplitude stress cycles, and the number of cycles until fracture is counted. By repeating this test for a number of different stress ranges, the results can be plotted and a best-fit S-N curve can be generated, as illustrated in figure 2.3. In most cases, both the stress and number of cycles are displayed on logarithmic scales.

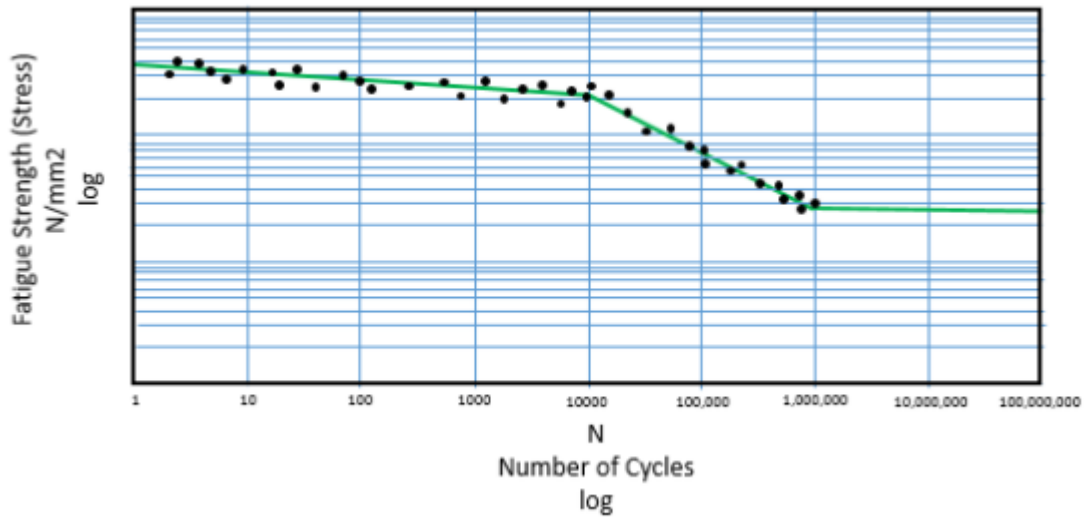


Figure 2.3 Best-fit curves for the S-N data points [10]

A linear curve on the double logarithmic scale can be expressed as:

$$N = C\Delta\sigma^{-m} \quad (2.4)$$

Where N is the number of cycles, $\Delta\sigma$ is the stress stress range, m is the inverse of the slope, and C describes the interception point between the log N -axis and the S-N curve.

Taking the logarithm of equation 2.4 yields the following:

$$\log N = \log C - m \log \Delta\sigma \quad (2.5)$$

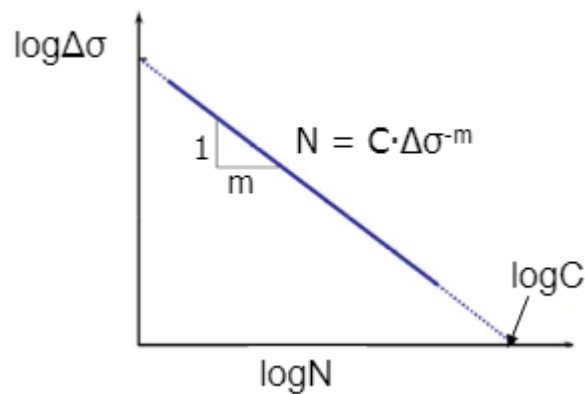


Figure 2.4 Illustration of an S-N curve with the associated nomenclature [11]

2.1.5 Variable Loading and Miner-Palmgren

Consider the variable amplitude loading situation illustrated in figure 2.5. A given stress amplitude, σ_{a1} , is applied for an N_1 number of cycles. From the associated S-N curve the number of cycles to failure, N_{f1} , for σ_{a1} can be obtained. The ratio N_1/N_{f1} is a measure of the fractional fatigue damage caused by this specific series of cyclic loading.

Now consider a different stress amplitude, σ_{a2} , corresponding to N_{f2} on the S-N curve, applied for N_2 number of cycles. A similar type of fractional fatigue damage, N_2/N_{f2} can be found. The Miner-Palmgren rule states that failure will occur in the material when these fractions sum to unity:

$$D = \sum_{i=1}^{k_b} \frac{N_i}{N_{fi}} = \frac{1}{C} \sum_{i=1}^{k_b} N_i \Delta\sigma_i^m = 1 \quad (2.6)$$

where D is the accumulated fatigue damage, k_b is the number of stress blocks, N_i is the number of stress cycles in stress block i , and N_{fi} is the number of cycles to failure at constant stress range $\Delta\sigma_i$.

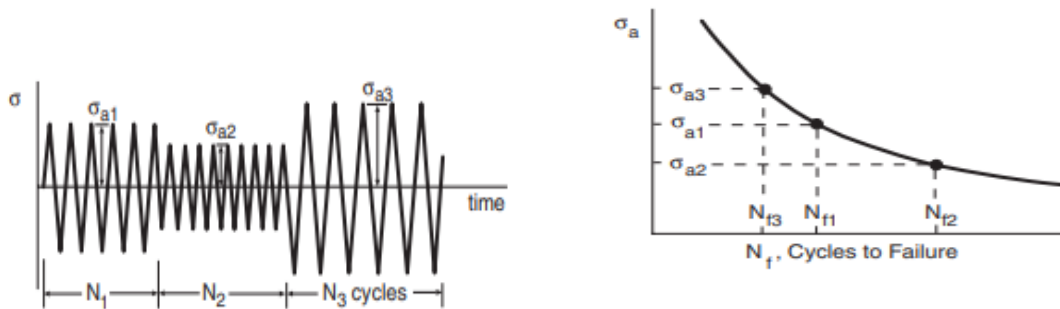


Figure 2.5 Variable amplitude loading

Figure 2.6 Relating different stress ranges to number of cycles to failure on the S-N curve

2.1.6 Irregular Loading and Rainflow Cycle Counting

In most practical applications fatigue loadings involve stress amplitudes that change in an irregular manner, such as those in figure 2.7. It is not obvious how to identify, measure, and count cycles so that the Miner-Palmgren rule can be employed. However, there is considerable consensus that a procedure known as Rainflow cycle Counting is the best approach. Before moving on to the algorithm itself, it is necessary to present some relevant terminology. As illustrated in the irregular stress history presented in figure 2.7, peaks and valleys are points in which the direction of loading changes. A simple range is the stress difference between a peak and its neighboring valley, or vice versa. Overall ranges denote stress differences measured between a peak and valleys that do not follow immediately, or between a valley and a later peak.

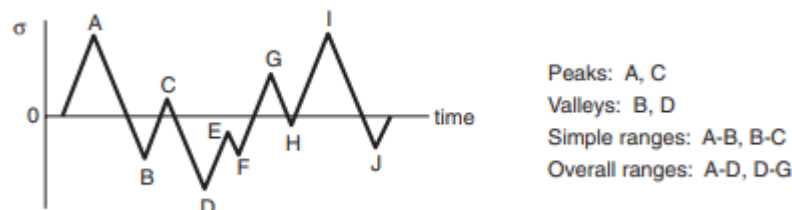


Figure 2.7 Example of an irregular loading situation with relevant definitions for the rainflow counting method.

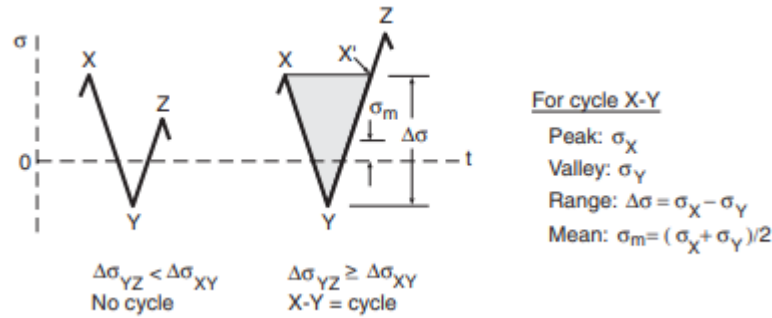


Figure 2.8 Definition of a cycle in Rainflow Cycle Counting with related definitions

Consider the X-Y-Z sequence in figure 2.8. In performing rainflow cycle counting, a peak-valley-peak or valley-peak-valley sequence is identified as a cycle if the second range, σ_{YZ} , is greater than, or equal to, the first range, σ_{XY} :

$$\sigma_{XY} \leq \sigma_{YZ} \quad (2.7)$$

The value assigned to an identified cycle is equal to the first range in the comparison sequence. If a cycle is counted, its information is recorded, and the counted cycle is assumed to not exist for further counting. This process, and how it may affect further analysis and counting, is illustrated in figure 2.9. After A-B is identified as a cycle, it is removed, and a new line is drawn between point H and point C. As H-C proves to be another valid cycle, the process is repeated until we're left with nothing but the D-G-D sequence.

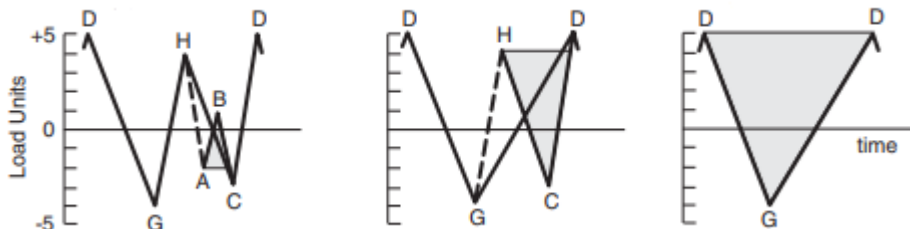


Figure 2.9 Procedure of Rainflow Cycle Counting

The complete procedure is over when the stress history is exhausted. The counted cycles are usually organized in tables, sorted by stress ranges and mean stresses. It is then possible to deploy the Miner-Palmgren rule to calculate a fatigue life for the element.

2.2 Buckling Theory

Long and slender members, known as columns, subjected to compressive loadings are susceptible to deflect laterally or sideways. Lateral deflection leads to a bending moment in the column, thus increasing the stresses experienced by the member in accordance with equation 2.1. This phenomenon is known as buckling and can lead to sudden and

dramatic failure in a structure. Buckling may occur even though the stresses that develop in the structure are well below those needed to cause failure in the material used. To make sure that they can safely support the intended loadings, special attention should be given to the design of columns.

2.2.1 Euler's Critical Load

Swiss Mathematician Leonhard Euler derived a formula to calculate the maximum axial load that a long, slender, ideal column can carry without buckling. This is known as the Euler's critical load, P_{cr} , and can be expressed as follows:

$$P_{cr} = \frac{\pi^2 EI_{min}}{L_e^2} \quad (2.8)$$

where E is the material's modulus of elasticity, I_{min} is the least moment of inertia for the column's cross-sectional area, and L_e is the effective unsupported length of the column.

Ideal Column

The Euler's critical load formula was derived assuming an ideal column. An ideal column is one that is perfectly straight prior to being subjected to loads, is made of homogeneous material, and free from initial stresses.

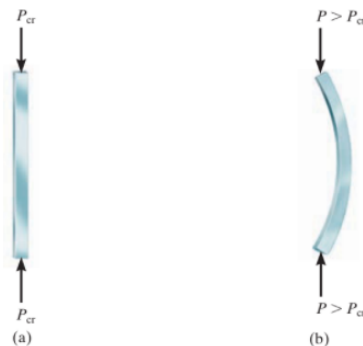


Figure 2.10 Side by side comparison between a column before and after buckling

2.2.2 Buckling Lengths and Slenderness Ratio

Depending on how a certain column is supported at its ends, the effective length, L_e , will vary. The effective length is essentially the distance between the points of zero moment in the column. Consequently, the effective length can be quantified by a dimensionless coefficient, K , describing what fraction of the column's length is deflected when it experiences buckling:

$$L_e = KL \quad (2.9)$$

where L is the column's actual length.

Some of the more common types of column supports, and their associated value for K , are illustrated in figure 2.11.

One of the more important characteristics deciding the propensity of a column to buckle is its slenderness ratio:

$$\lambda = \frac{L_e}{i} \quad (2.10)$$

where i is the radius of gyration. The radius of gyration is defined as:

$$i = \sqrt{\frac{I}{A}} \quad (2.11)$$

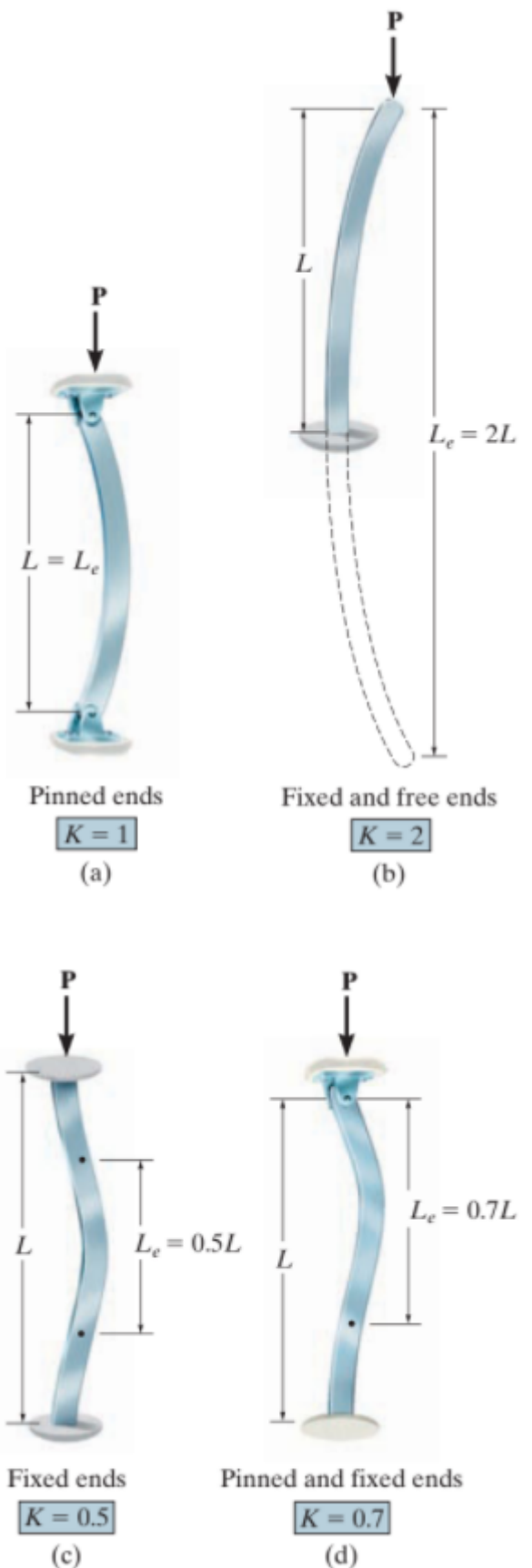


Figure 2.11 Column supports and associated buckling lengths

2.3 The Von Mises Yield Criterion

Predicting failure in ductile materials can be done using the Von Mises yield criterion. It states that a material will fail when the maximum distortion energy, u_d is equal to, or higher, than the distortion energy, $u_{y,d}$, at yielding in a uniaxial tensile test:

$$u_d \geq u_{y,d} \quad (2.12)$$

Essentially, the Von Mises equivalent stress combines normal and shear stresses to a single scalar that can be compared to that of the material's yield strength, f_y . For a two dimensional analysis, the Von Mises equivalent stress, σ_{eq} , can be expressed as such:

$$\sigma_{eq} = \sqrt{\sigma_x^2 + 3\tau_{xy}^2} \quad (2.13)$$

where σ_x is stresses due to axial forces and τ_{xy} is stresses due to shear forces.

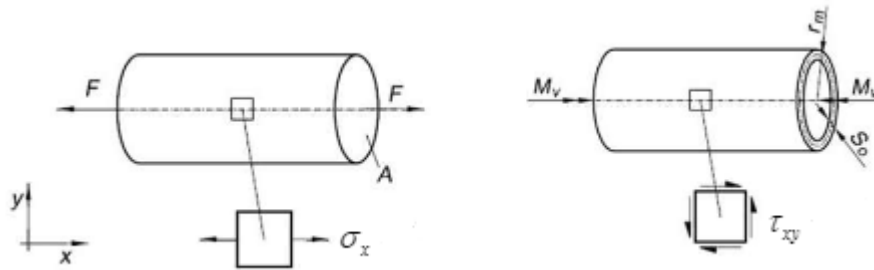


Figure 2.12 Plane stress for a cylinder [12]

Ultimately, the Von Mises yield criterion says that the Von Mises equivalent stress cannot exceed that of the material's yield strength without failing. Ductile structural members subjected to a combined loading of shear and axial stresses should be verified against the following:

$$\sigma_{eq} \leq f_y \quad (2.14)$$

It is important to note that the Von Mises yield criterion is not perfect, and materials might fail at Von Mises stress values below the yield stress. However, it is the preferred failure theory in most practical applications, owing to the fact that it agrees well with experimental data.

3. Methodology

3.1 The Structure

The structure analyzed in this thesis is inspired by one that is developed in conjunction with an ongoing project on MRWTs at the Institute for Energy Technology (IFE). In its entirety, the full structure consists of an array of rotors mounted on a tower. The scope of this project is, however, limited to investigating a single module in the turbine’s lattice structure. There will be made separate analyses on a support structure featuring four blades, and another featuring two blades. The support structure remain identical for both instances. The structural members’ parameters presented in the following section is prior to any modifications; the baseline structure.

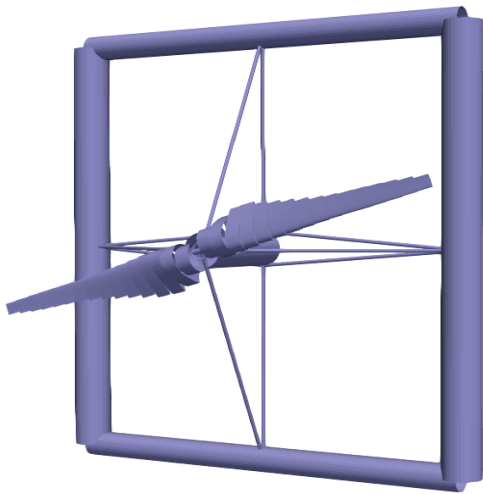


Figure 3.1 A single MRWT module featuring two blades

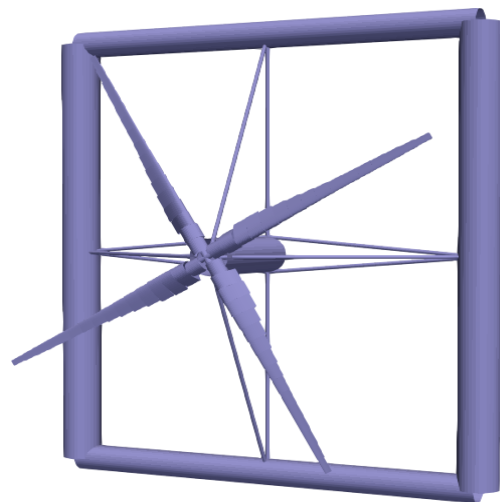


Figure 3.2 A single MRWT module featuring four blades

The structure is largely made up of circular hollow section (CHS) elements, that can further be divided into any of the categories listed under Elements in table 3.1. Four CHS elements connected at the ends make up a rectangular framework for the rotor, while struts¹ connected between the framework and the nacelle is keeping the RNA in place. The CHS elements are all joined together using Complete Joint Penetration (CJP) groove welds. All frame elements share the same diameter and wall thickness, but differ in length. The same goes for the strut elements.

¹A rod or bar forming part of a framework and designed to resist compression and/or tension

Element(s)	Diameter [m] d	Wall thickness [m] t	Length [m] L	Label
Frame (Horizontal)	1.5	0.016	22	1
Frame (Vertical)	1.5	0.016	25	2
Struts (Front Vertical)	0.2	0.008	13.87	3
Struts (Front Horizontal)	0.2	0.008	12.53	4
Struts (Rear Horizontal)	0.2	0.008	11.05	5
Struts (Rear Vertical)	0.2	0.008	12.5	6
Nacelle	2	0.08	6	7
Rotor	0.1	0.03	0.1	N/A

Table 3.1 Main elements of the support structure and their geometric data

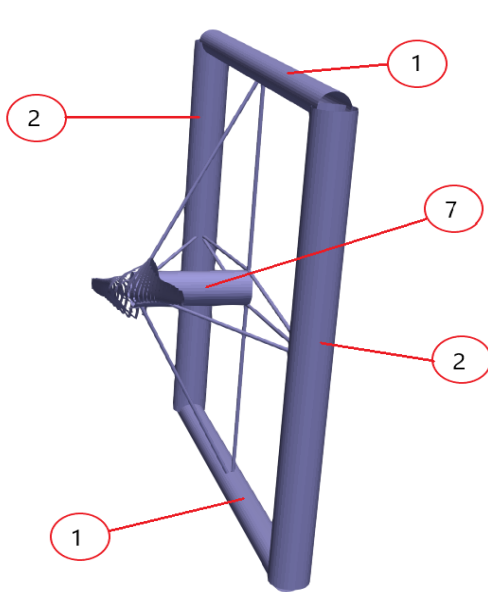


Figure 3.3 Elements in the table labelled according to table 3.1

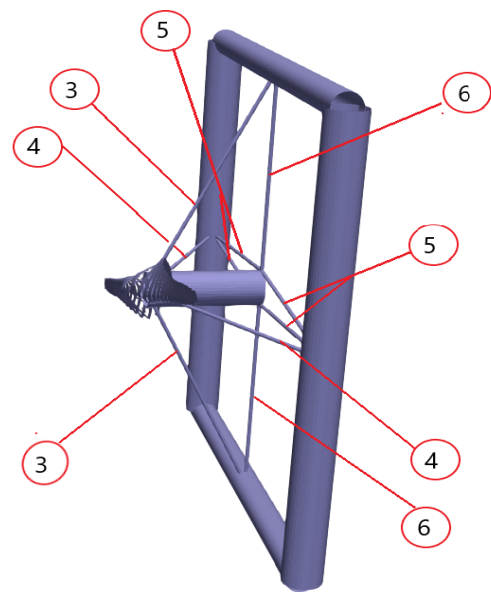


Figure 3.4 Elements in the table labelled according to table 3.1

The elements in the structure are all made of structural steel s355, which have the following properties:

Property	Value
Modulus of Elasticity, E	210000 [MPa]
Shear Modulus, G	81000 [MPa]
Poisson's Ratio, ν_p	0.30
Yield Strength, f_y	355 [MPa]
Ultimate Strength, f_u	470 [MPa]

Table 3.2 Properties of structural steel s355

3.2 The Simulation Tools

3.2.1 3DFloat

The structure is modelled and analyzed using the 3DFloat simulation tool. Developed by the Wind Energy Department at IFE, 3DFloat is an aero-servo-hydro-elastic Finite-Element-Method code tailored for simulations of offshore structures in general, and offshore wind turbines in particular. Reference is made to [13] for a more detailed and thorough model description.

Through participation in the code-to-code verification activities of the Offshore Code Comparison Collaboration projects (OC3, OC4, and OC5), the quality of 3DFloat has both been verified and improved.

3.2.2 Visualization and Post-Processing

The software does not have visualization and animation features, and so this must be done with a third party program if necessary or desirable. In this thesis, the open-source application ParaView has been used for this purpose.

The results yielded from 3DFloat simulations are tabulated in files that may be post-processed using any preferred data management tool. In the case of this thesis, the programming language Python was employed.

3.2.3 Input - 3DFloat simulation

One of the main objectives of this thesis is to compare results between a two-bladed and a four-bladed rotor. Effort is made to ensure that this is an "apples to apples" comparison. That is: they are subjected to the same winds, and output the same power. The blades used for the two-bladed rotor is similar to those of the four-bladed structure; only with a doubled chord length².

The support structure is to be analyzed for both normal and extreme wind conditions. In the case of extreme wind conditions, the rotor blades are not rotating. When analyzing the structure subjected to normal wind conditions, however, the turbine is operating as normal.

²The width of the blade

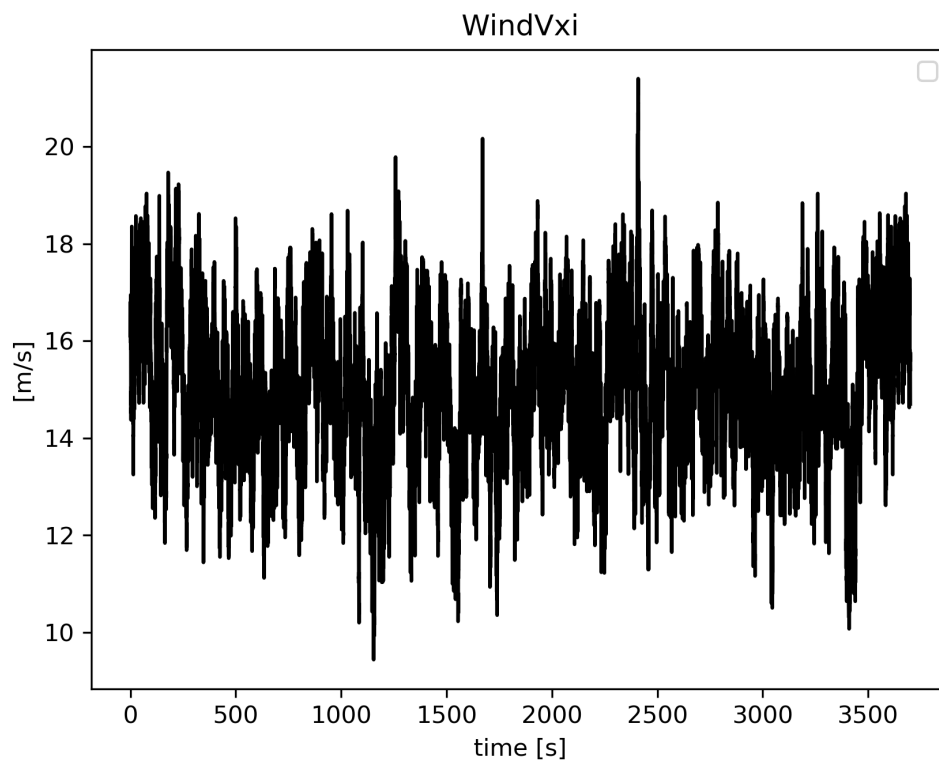


Figure 3.5 Wind speed as a function of time normal to the rotor at normal wind conditions

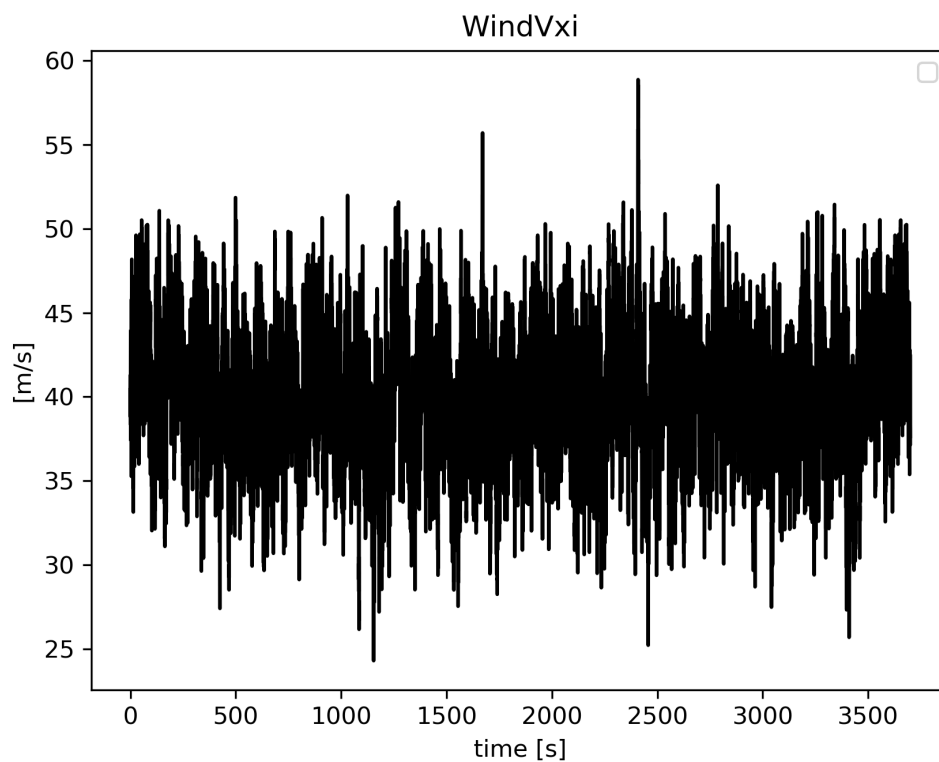


Figure 3.6 Wind speed as a function of time normal to the rotor at extreme wind conditions

Every strut is divided into eight equally sized elements along its length. 3DFloat enables monitoring elements of interest, see subsection 3.3.1 for more details.

See appendix C for the input code in its entirety.

3.3 Fatigue Calculations

The fatigue calculations are carried out according to the DNV GL Recommended Practice (DNVGL-RP-C203) [14].

To assess the fatigue damage at points of interest throughout the structure, simulation result data was imported to Python for post-processing. The imported data, in the shape of time-series, are fed into routines that convert forces and moments into stresses in the cross-section. This will be discussed in more detail in section 3.3.2. The stress-series are then processed by rainflow counting, and finally the cumulative fatigue damage can be assessed by employing the Miner-Palmgren rule and an appropriate S-N curve. The rainflow counting routine was developed by Marit Kvittem at SINTEF, see appendix D.

3.3.1 Points of interest

The analysis is limited to certain points in the structure suspected to be more vulnerable to fatigue failure than others. These includes the welds at each end of every strut, the root of the blades, and the rotating shaft. In summary, this means that the investigated points are either:

- Tubular joints (Strut ends)
- CHS made of structural steel s355 (shaft, and blade roots)

These require different S-N curves and differ in stress calculations due to hot spot stress considerations for tubular joints.

3.3.2 DNV GL Recommended Practice

The DNVGL-RP-C203 provides relevant S-N curve data and the necessary information to decide which S-N curve is appropriate for a given case. Provided are also guidelines for stress calculations and parametric equations to derive hot spot stresses.

S-N Curves

The S-N curves presented in the Recommended Practice are based on the mean-minus-two-standard-deviation curves for experimental data, and are thus associated with a

97.7% probability of survival.

Recalling that equation 2.5 represents the best-fit curve for the experimental data, subtracting two standard deviations from the intercept between the S-N curve and the log N -axis yields:

$$\log \bar{a} = \log C - 2s_{\log N} \quad (3.1)$$

where $s_{\log N}$ is one standard deviation of $\log N$

implementing the new term, $\log \bar{a}$, we get the following expression for the S-N curve which is downward shifted by two standard deviations:

$$\log N = \log \bar{a} - m \log \Delta\sigma \quad (3.2)$$

Due to the local geometry of the weld toe in relation to thickness of adjoining plates, the fatigue strength of welded joints is to some extent dependent on plate thickness. Introducing reference thickness, t_{ref} , and thickness exponent, k , yields the final and complete expression for the design S-N curve presented in the Recommended Practice:

$$\log N = \log \bar{a} - m \log \left(\Delta\sigma \left(\frac{t}{t_{ref}} \right)^k \right) \quad (3.3)$$

where t is the actual plate thickness of the geometry.

Tubular joints

All tubular joints are assumed to be class T for joints that are welded from both outside and inside, and applies to the outside hot spots at the joints. For simple tubular joints the reference thickness is 16 *mm*.

Environment	m_1	$\log \bar{a}_1$	m_2	$\log \bar{a}_2$	Fatigue limit at 10 ⁷ cycles [MPa]	Thickness exponent k
Air	N ≤ 10 ⁷ cycles		N > 10 ⁷ cycles			
	3.0	12.48	5.0	16.13	67.09	0.25

Table 3.3 Class T S-N curve data from DNVGL-RP-C203

CHS

As per table A-9 in the Recommended Practice, CHS members are to be designed according to S-N class B1. For these types of elements, the reference thickness is 25 *mm*.

Environment	m_1	$\log \bar{a}_1$	m_2	$\log \bar{a}_2$	Fatigue limit at 10^7 cycles [MPa]	Thickness exponent k
Air	$N \leq 10^7$ cycles		$N > 10^7$ cycles			
	4.0	15.117	5.0	17.146	106.97	0.25

Table 3.4 Class B1 S-N curve data from DNVGL-RP-C203

Stress Calculations

For a tubular joint, i.e. brace to chord connection, the stress to be used for design purpose is the range of hot spot stress, calculated as:

$$\sigma_{hot\ spot} = SCF \sigma_{nominal} \quad (3.4)$$

The analyzed members in the structure are subject to axial forces as well as bending moments about different axes. Consequently, the stresses induced are not uniformly distributed over the cross-section.

The stresses in tubular joints due to brace loads are calculated at the crown and the saddle points, see figure 3.7. The hot spot stresses at these points are calculated by superposition of the single stress components from axial, in-plane and out-of-plane action. However, the stresses may be higher at intermediate points between saddle and crown points. Thus, the hot spot stress is evaluated at eight points of equal spacing around the circumference of the intersection, see figure 3.8.

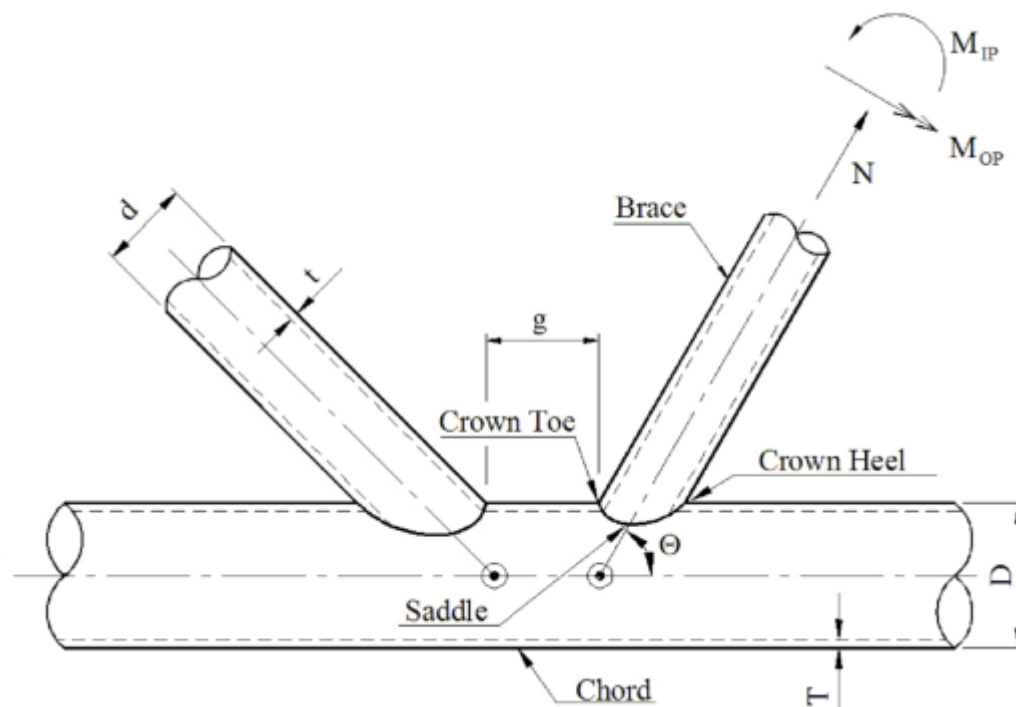


Figure 3.7 Geometrical definitions for tubular joints

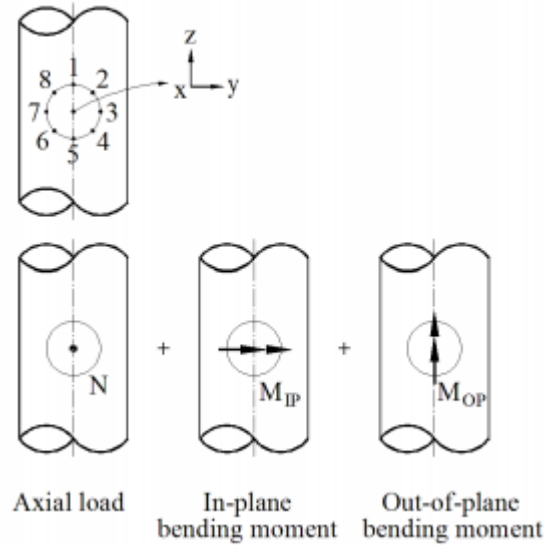


Figure 3.8 Superposition of stresses around the circumference due to combined loading

The stresses at the intermediate points are achieved by a linear interpolation of the stress due to the axial action at the crown and saddle, and a sinusoidal variation of the bending stress resulting from in-plane and out of plane bending. This results in the following set of equations, one for each point in the circumference:

$$\sigma_1 = (SCF_{AC})\sigma_x + (SCF_{MIP})\sigma_{my} \quad (3.5a)$$

$$\sigma_2 = \frac{1}{2}(SCF_{AC} + SFC_{AS})\sigma_x - \frac{1}{2}\sqrt{2}(SCF_{MOP})\sigma_{mz} \quad (3.5b)$$

$$\sigma_3 = (SCF_{AS})\sigma_x - (SCF_{MOP})\sigma_{mz} \quad (3.5c)$$

$$\sigma_4 = \frac{1}{2}(SCF_{AC} + SFC_{AS})\sigma_x - \frac{1}{2}\sqrt{2}(SCF_{MOP})\sigma_{mz} \quad (3.5d)$$

$$\sigma_5 = (SCF_{AC})\sigma_x - (SCF_{MIP})\sigma_{my} \quad (3.5e)$$

$$\sigma_6 = \frac{1}{2}(SCF_{AC} + SFC_{AS})\sigma_x + \frac{1}{2}\sqrt{2}(SCF_{MOP})\sigma_{mz} \quad (3.5f)$$

$$\sigma_7 = (SCF_{AS})\sigma_x + (SCF_{MOP})\sigma_{mz} \quad (3.5g)$$

$$\sigma_8 = \frac{1}{2}(SCF_{AC} + SFC_{AS})\sigma_x + \frac{1}{2}\sqrt{2}(SCF_{MOP})\sigma_{mz} \quad (3.5h)$$

Here, SCF_{MOP} is the stress concentration factor for out-of-plane action, while SCF_{MIP} is the stress concentration factor for in-plane-bending. SCF_{AS} is the stress concentration factor at the saddle points, and SCF_{AC} is the stress concentration factor at the crown. σ_{my} and σ_{mz} are the stresses due to in-plane and out-of-plane bending respectively, while σ_x is the stress induced due to axial loading of the brace.

NOTE

The combination of stresses and SCFs in equations 3.5 assume an orientation of the

element where the z-axis goes through crown points of the joint. For certain analyzed elements this is not the case, and it is the y-axis that goes through the crown points. In this case equations 3.5 must be modified to account for this. This is done by multiplying σ_{my} with SCF_{MOP} , and σ_{mz} with SCF_{MIP} for all instances.

Stresses induced by bending are dependant on the cross-section's moment of inertia. The analyzed parts are all CHS elements, whose moment of inertia can be calculated by the following formula:

$$I_{CHS} = \frac{\pi}{4}(R^4 - R_i^4) = \frac{\pi}{64}(D^4 - D_i^4) \quad (3.6)$$

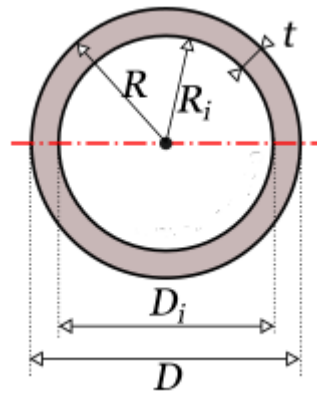


Figure 3.9 Geometrical definitions of a hollow cylinder's cross-section

SCFs

SCFs for simple tubular joints and overlap joints can be derived using parametric equations found in Table B-1 in the Recommended Practice. The equations are based on the equations developed by Efthymiou found in [15]. These vary depending on the classification of the joint (T/Y, K, or X), and are functions of the geometrical relationships given in equation 3.7. Relevant parametric formulae are given under the section "Parametric Formulae" below. The analyzed points in the structure include five different types of tubular joints. These are presented one by one under the section "Joint Types in the Structure", accompanied by relevant SCF calculations.

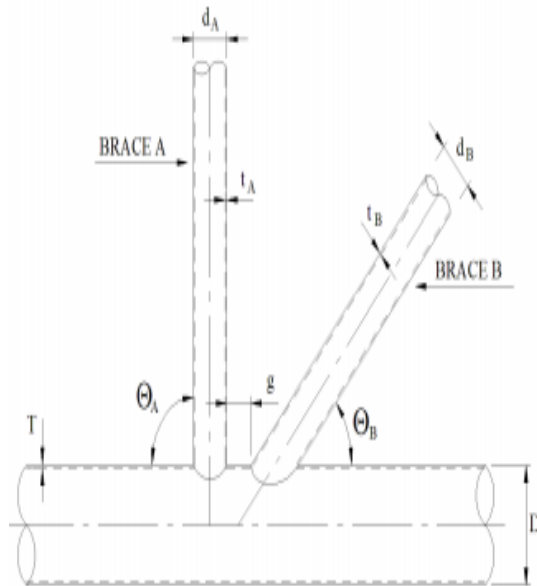


Figure 3.10 Definitions of geometrical parameters for K-joints

$$\beta = \frac{d}{D} \quad (3.7a)$$

$$\alpha = \frac{2L}{D} \quad (3.7b)$$

$$\gamma = \frac{D}{2T} \quad (3.7c)$$

$$\tau = \frac{t}{T} \quad (3.7d)$$

$$\zeta = \frac{g}{D} \quad (3.7e)$$

Tubular joints are classified as either T/Y, K, or X joints. These classifications are simply based on what letter the geometrical shape of the joint resembles. In the case of this thesis, only joint types T/Y and K are analyzed. Figure 4.11 and 4.11 illustrates examples of such joints, as well as the relevant parameters.

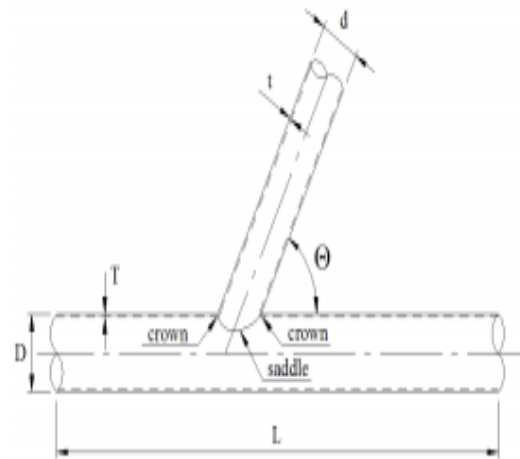


Figure 3.11 Definitions of geometrical parameters for T-joint

Valid Range:

Parameter	α	β	γ	τ	ζ
Valid range	$4 \leq \alpha \leq 40$	$0.2 \leq \beta \leq 1.0$	$8 \leq \gamma \leq 32$	$0.2 \leq \tau \leq 1.0$	$\frac{-0.6\beta}{\sin\theta} \leq \zeta \leq 1.0$

Parametric Formulae**T-Joint**

SCF_{AC}	$3 + \gamma^{1.2}(0.12 \exp(-4\beta) + 0.011\beta^2 - 0.045) + \beta\tau(0.1\alpha - 1.2)$
SCF_{AS}	$1.3 + \gamma\tau^{0.52}\alpha^{0.1}(0.187 - 1.25\beta^{1.1}(\beta - 0.96))(\sin\theta)^{2.7-0.01\alpha}$
SCF_{MIP}	$1 + 0.65\beta\tau^{0.4}\gamma^{1.09-0.77\beta}(\sin(\theta))^{0.06\gamma-1.16}$
SCF_{MOP}	$\tau^{-0.54}\gamma^{-0.05}(0.99 - 0.47\beta + 0.08\beta^4)(\gamma\tau(1.7 - 1.05\beta^3)(\sin(\theta))^{1.6})$

K-Joint

SCF_{AC}	$1 + (1.97 - 1.57\beta^{0.25})\tau^{-0.14}(\sin\theta)^{0.7}\tau^{0.9}\gamma^{0.5}(0.67\beta^2 + 1.16\beta)$
SCF_{AS}	same as SCF_{AC}
SCF_{MIP}	$1 + 0.65\beta\tau^{0.4}\gamma^{1.09-0.77\beta}(\sin(\theta))^{0.06\gamma-1.16}$
SCF_{MOP}	$\tau^{-0.54}\gamma^{-0.05}(0.99 - 0.47\beta + 0.08\beta^4)$ $(\tau_A^{-0.54}\gamma^{-0.05}(0.99 - 0.47\beta_A + 0.08\beta_A^4)(\gamma\tau_A\beta_A(1.7 - 1.05\beta_A^3)(\sin(\theta_A))^{1.6})$ $(1-0.08(\beta_B\gamma)^{0.5} \exp(-0.8x)) +$ $\tau_B^{-0.54}\gamma^{-0.05}(0.99 - 0.47\beta_B + 0.08\beta_B^4)(\gamma\tau_B\beta_B(1.7 - 1.05\beta_B^3)(\sin(\theta_B))^{1.6})$ $(1-0.08(\beta_B\gamma)^{0.5} \exp(-0.8x))(2.05\beta_{max}^{0.5} \exp(-1.3x))$

where

$$x = 1 + \frac{\zeta \sin\theta}{\beta}$$

Joint Types in the Structure

Type 1 joint (T)

Elements included in the joint geometry (see table 3.1):

- Strut (Front Vertical/Horizontal)
- Shaft

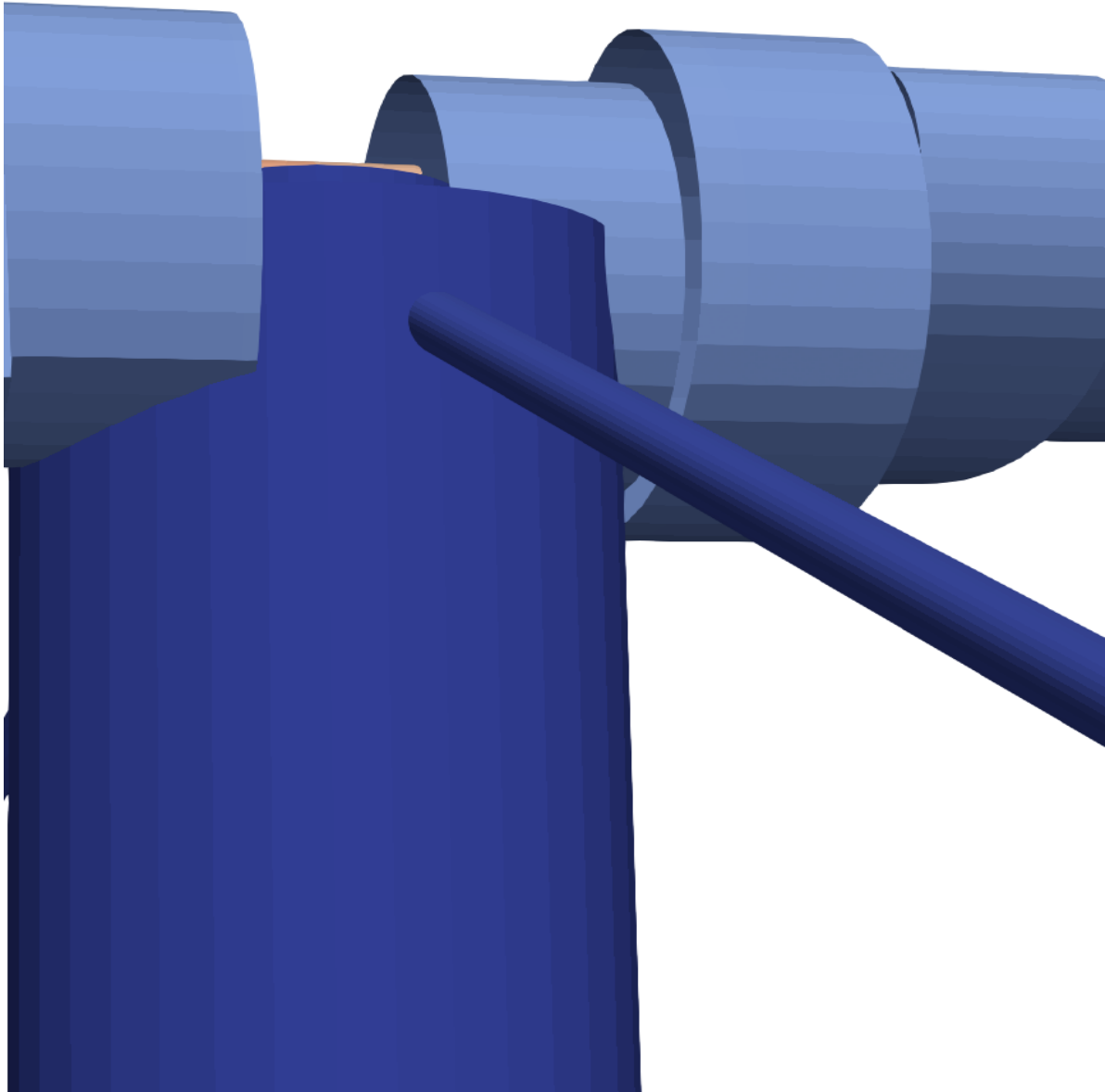


Figure 3.12 Example of a Type 1 joint

Parameter	α	β	γ	τ	θ
Value	6.0	0.1	12.5	0.1	61.39

SCF	SCF_{AC}	SCF_{AS}	SCF_{MIP}	SCF_{MOP}
Value	3.73	2.09	1.36	1

Type 2 joint (T)

Elements included in the joint geometry (see table 3.1):

- Strut (Front Vertical/Horizontal, Rear Vertical)
- Frame (Vertical/Horizontal)

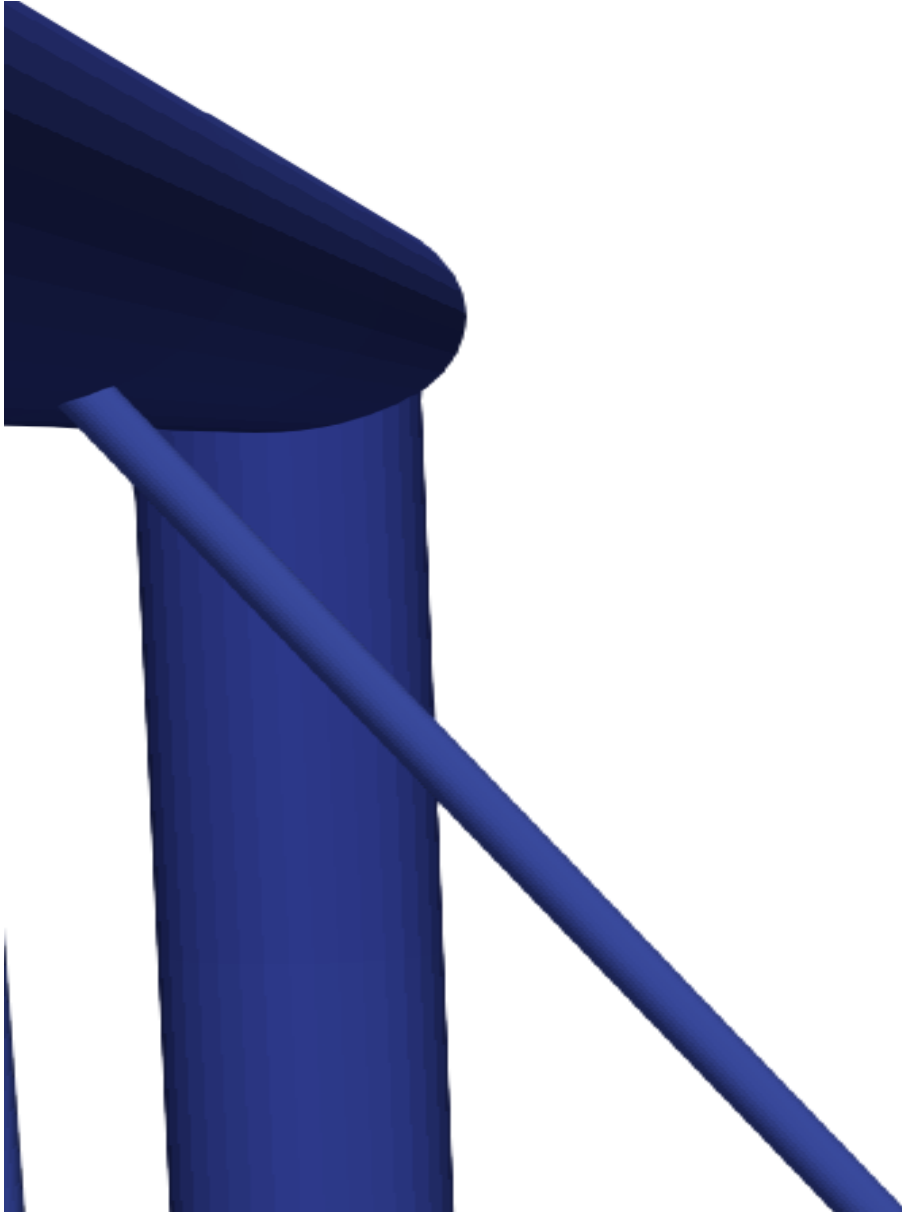


Figure 3.13 Example of a Type 2 joint

Parameter	α	β	γ	τ	θ
Value	33.33	0.13	46.88	0.5	90.00

SCF	SCF_{AC}	SCF_{AS}	SCF_{MIP}	SCF_{MOP}
Value	5.73	15.21	3.93	5.90

Type 3 joint (T)

Elements included in the joint geometry (see table 3.1):

- Strut (Rear Horizontal)
- Strut (Rear Vertical)

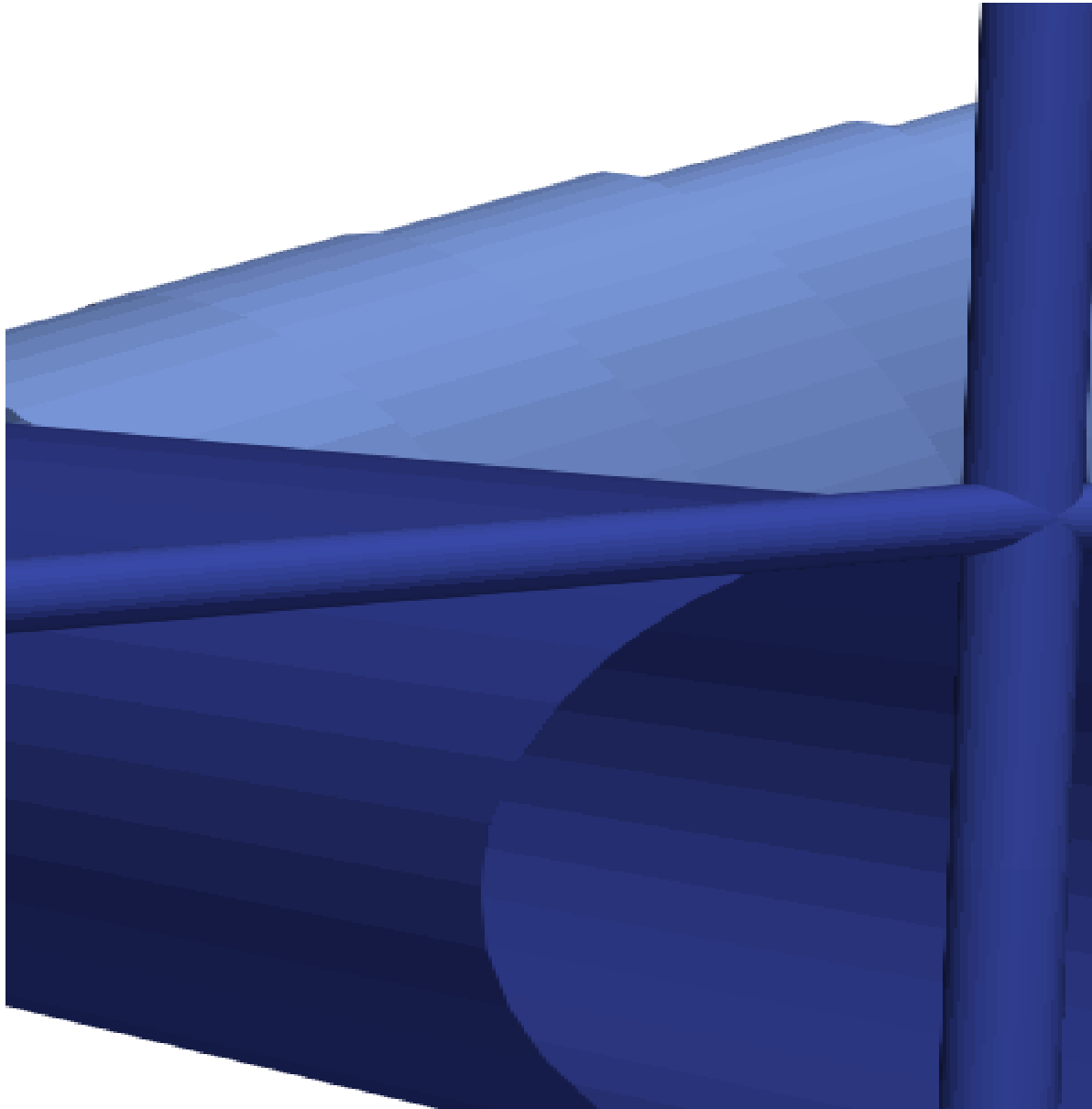


Figure 3.14 Example of a Type 3 joint

Parameter	α	β	γ	τ	θ
Value	250	1.00	12.50	1.00	84.8

SCF	SCF_{AC}	SCF_{AS}	SCF_{MIP}	SCF_{MOP}
Value	26.14	4.27	2.46	4.27

Type 4 joint (K)

Elements included in the joint geometry (see table 3.1):

- Strut (rear horizontal) x 2
- Frame (Vertical)

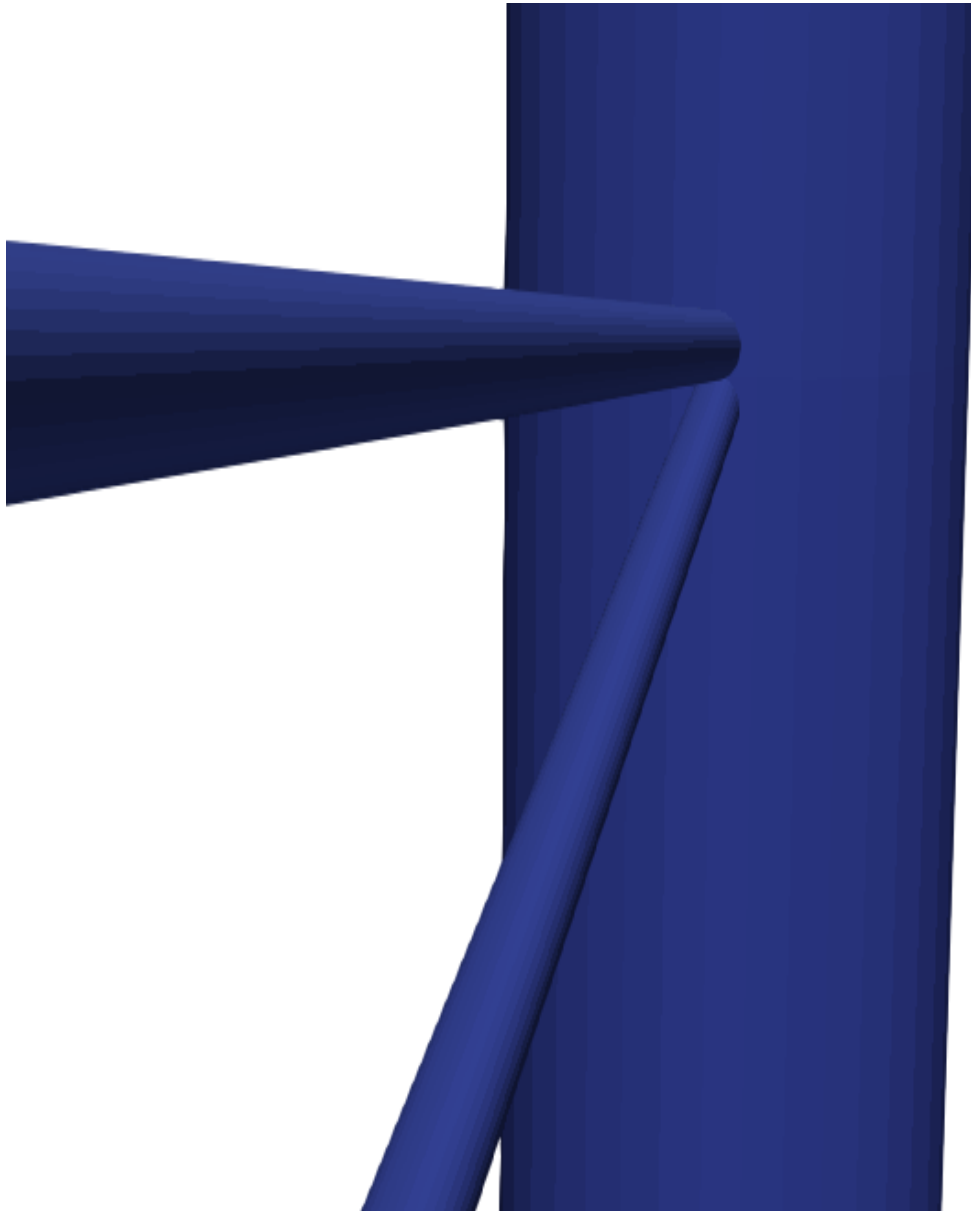


Figure 3.15 Example of a Type 4 joint

Parameter	d_A	t_A	θ_A	d_B	t_B	θ_B	D	L	T	g
Value	0.2	0.008	84.8	0.2	0.008	84.8	1.5	2	0.016	0.136

SCF	SCF_{AC}	SCF_{AS}	SCF_{MIP}	SCF_{MOP}
Value	7.72	7.72	3.91	6.03

Design Fatigue Factor (DFF)

DFFs are related to partial safety factors used in code-based design of any structure, or its component, to account for uncertainties associated with the design process. DNVGL-ST-0119 - Floating wind turbine structures [16] provides guidelines for appropriate DFFs based on:

- Location of the structural detail
- Accessibility for inspection or repair

As the analyzed structure in this thesis may be considered as an “External structure, accessible for regular inspection and repair in dry and clean conditions.”, it is advised to set $DFF = 1$ according to table 7-5 in the standard [16].

3.4 Buckling Calculations

The four front struts are long and slender, and will be under significant compressive forces due to axial loading and moments. These members are therefore to be verified against buckling. Calculations will be carried out according to EN 1993-1-1 (2005): Eurocode 3: Design of steel structures - Part 1-1: General rules and rules for buildings practice (EC3) [17]. See table 3.1 and 3.2 for relevant properties of the front struts.

3.4.1 Eurocode

Classification of geometrical properties

Cross-section classification

The calculation of a member’s resistance always starts with cross-section classification. EC3 features the concept of cross-section classification to account for the influence of local instability. For Class 1 and 2 cross-sections, the design resistance is taken as the full plastic cross-section capacity. For Class 3 cross-sections, only the elastic cross-section resistance is utilized due to premature local buckling. Class 4 cross-sections are those in which local buckling will occur before the effect of yield stress in one or more parts of the cross-section. Formulae for cross-section classification for CHS members are presented in table 3.5

Class	Section in bending and/or compression					
1	$d/t \leq 50\epsilon^2$					
2	$d/t \leq 70\epsilon^2$					
3	$d/t \leq 90\epsilon^2$					
$\epsilon = \sqrt{235/f_y}$	f_y	235	275	355	420	460
	ϵ	1.00	0.92	0.81	0.75	0.71
	ϵ^2	1.00	0.85	0.66	0.56	0.51

Table 3.5 Classification of CHS according to EC3

Imperfection factor

To account for any defects that might be present in the analyzed member, i.e. geometric imperfections, eccentricities of applied loads, and residual stresses, EC3 introduces an imperfection factor α_i . The imperfection factor is based on deciding whichever buckling curve is appropriate from table 3.6

Cross section	Finish	Buckling about axis	Buckling curve	
			S 235 S 275 S 355 S 420	S 460
CHS	Hot finished	any	a	a_0
	Cold formed	any	c	c

Table 3.6 Assessment of buckling curve for CHS members

when the appropriate buckling curve is assessed, the imperfection factor, α_i , may be obtained from table 3.7

Buckling curve	a_0	a	b	c	d
Imperfection factor α_i	0.13	0.21	0.34	0.49	0.76

Table 3.7 Relationship between buckling curve and imperfection factor α

Buckling Length

The buckling length, L_{cr} , of a hollow section brace member without cropping or flattening, welded around its perimeter to hollow section chords, may be generally taken as $0.75L$ for both in-plane and out-of-plane buckling.

The analyzed members have the following properties:

Parameter	d [m]	t [m]	f_y [MPa]	L_{cr} [m]	$class$	ϵ	α_i	Finish
Value	0.2	0.008	355	10.4/9.4	a	0.81	0.21	Hot finished

Table 3.8 Buckling related properties of the structure's front struts

Buckling verification - compression and uniaxial bending

EC3 presents the following governing inequalities for which the member should be verified against:

$$\frac{N_{Ed}}{N_{b,Rd}} + k_{yy} \frac{M_{y,Ed} + \Delta M_{y,Ed}}{M_{by,Rd}} + k_{yz} \frac{M_{z,Ed} + \Delta M_{z,Ed}}{M_{bz,Rd}} \leq 1.0 \quad (3.8)$$

and

$$\frac{N_{Ed}}{N_{b,Rd}} + k_{zy} \frac{M_{y,Ed} + \Delta M_{y,Ed}}{M_{by,Rd}} + k_{zz} \frac{M_{z,Ed} + \Delta M_{z,Ed}}{M_{bz,Rd}} \leq 1.0 \quad (3.9)$$

where

- N_{Ed} is the design value for the compression force
- $N_{b,Rd}$ is the design buckling force of the compression member
- $M_{y,Ed}$, $M_{z,Ed}$ are the design value of the moments about the y-y and z-z axes respectively
- $M_{by,Rd}$, $M_{bz,Rd}$ are the design buckling resistance moment about the y-y and z-z axis respectively
- $\Delta M_{y,Ed}$, $\Delta M_{z,Ed}$ represents the moments caused by shift in the position of the neutral axis due to buckling, for class 4 sections
- k_{yy} , k_{yz} , k_{zy} , and k_{zz} are the interaction factors

According to [18, p. 17], a CHS beam-column, owing to its symmetric geometry, biaxial bending is treated in the same manner as uniaxial bending, leading to the simplified interaction formula 3.10:

$$\frac{N_{Ed}}{N_{b,Rd}} + k \frac{M_{Ed}}{M_{b,Rd}} \leq 1.0 \quad (3.10)$$

where the interaction factor, k , can be calculated according to Table B.1 in EC3 as such:

$$k = C_m \left(1 + (\bar{\lambda} - 0.2) \frac{N_{Ed}}{N_{b,Rd}} \right) \text{ for } \bar{\lambda} \leq 1 \quad (3.11)$$

and

$$k = C_m \left(1 + 0.8 \frac{N_{Ed}}{N_{b,Rd}} \right) \text{ for } \bar{\lambda} > 1 \quad (3.12)$$

where C_m is the equivalent uniform moment factor taken as unity in this study, assuming uniform bending.

The design buckling resistance of a compression member is calculated as:

$$N_{b,Rd} = \frac{\chi A f_y}{\gamma_{M0}} \quad (3.13)$$

where χ is the reduction factor for the associated buckling mode

For axial compression in members the value of χ for the appropriate relative slenderness, $\bar{\lambda}$, is to be determined according to:

$$\chi = \frac{1}{\Phi + \sqrt{\Phi^2 - \bar{\lambda}^2}} \quad (3.14)$$

where

$$\Phi = 0.5[1 + \alpha(\bar{\lambda} - 0.2) + \bar{\lambda}^2] \quad (3.15)$$

and

$$\bar{\lambda} = \sqrt{\frac{Af_y}{N_{cr}}} \quad (3.16)$$

N_{cr} is calculated similar to P_{cr} from equation 2.2.1, and f_y is obtained from table 3.2 of material properties.

The buckling resistance moment is calculated as follows:

$$M_{b,Rd} = \chi_{LT} \frac{W_y f_y}{\gamma_{M0}} \quad (3.17)$$

where W_y is the appropriate section modulus. CHS members are not susceptible to torsional deformation, and in such cases $\chi_{LT} = 1$

The analyzed members have the following properties:

Property	A [mm^2]	$\bar{\lambda}$	Φ	χ	$N_{b,Rd}$ [kN]	$M_{b,Rd}$ [kNm]
Value	4825	3.41	6.666	0.081	131.7	75.3

Table 3.9 More buckling related properties of the structure's front struts

3.4.2 Buckling Algorithm

As mentioned previously, each strut element is divided into eight equally sized elements along its length. Monitors set on each end of every front strut outputs values for axial forces as well as moments about all three axes for every time step in the simulation. The data is imported to python for analysis, where a calculation is made for the left-hand side of inequality 3.10 for every time step. See appendix D for the employed python routine.

3.5 Von Mises Equivalent Stress Assessment

The structural members are joined together using a (CJP) groove weld. A CJP is a groove weld that extends completely through the thickness of components joined. The primary purpose for the use of the CJP groove welds is to transmit the full load-carrying capacity of the structural components they join. Von Mises stress assessments are thus not necessary in the welds.

Von Mises equivalent stress Calculations are carried out according to theory presented in 2.3 with a safety factor $\gamma = 1.05$, leading to the following criterion:

$$\sigma_{eq} \leq \frac{f_y}{\gamma} = \frac{355}{1.05} = 338.1MPa \quad (3.18)$$

Investigated elements are similar to that of buckling and fatigue. Shear and axial/bending stresses are calculated at 8 points around the circumference of the cross-section similar to that of the fatigue calculations. The material's yield strength is according to table 3.2.

3.6 Adjusting Cross-Section Parameters

Results yielded from the fatigue analysis identifies members in the structure that need changes in order to meet the design requirements. Some members might also be overdimensioned, driving the cost of producing the turbine unnecessarily high. Cross-sectional properties of these members should be altered such as to guarantee a reasonable fatigue life, while also minimizing the material cost. For the first step towards a support structure of acceptable properties, the aim is to get all elements within 30-100 years of fatigue life. This way, a lower limit is assessed in terms of the entire structure's fatigue life. As long as any further adjustments to the members are of a *strengthening* nature, i.e. thicker walls, a fatigue life of more than 30 years is assured.

A CHS is defined by its diameter and wall thickness. A fatigue life versus wall thickness analysis is conducted for each element to find a cross-section that is within the range of acceptable fatigue life (see appendix E). If adjustments outside of what is reasonable must be made to the wall thickness to achieve this, a change in diameter will be made.

In the second stage, the turbine is subjected to extreme wind conditions, and Von Mises stresses are to be assessed for the two structures. This time it assessed with the adjusted cross-sectional parameters based on the fatigue analysis. Since a lower limit is already set for an acceptable fatigue life for each element, adjustments will only be made to those who exhibit too high of a Von Mises stress value. A wall thickness versus Von Mises stress value iteration is carried out to find a value which satisfies the Von Mises Yield criterion.

Finally, to investigate the necessity for further adjustments, a buckling analysis is carried out for both structures in both normal and extreme conditions.

4. Results

Due to simplicity in post-processing, elements contain abbreviations and labelling:

- vert: vertical
- hor: horizontal
- 1: the end of the strut closest to the center of the structure
- 2: the end of the strut farthest away from the center of the structure
- s1 -> s8: point on the circumference according to figure 3.8

If not specified through labelling, the most critical value available in that category has been identified and is being presented.

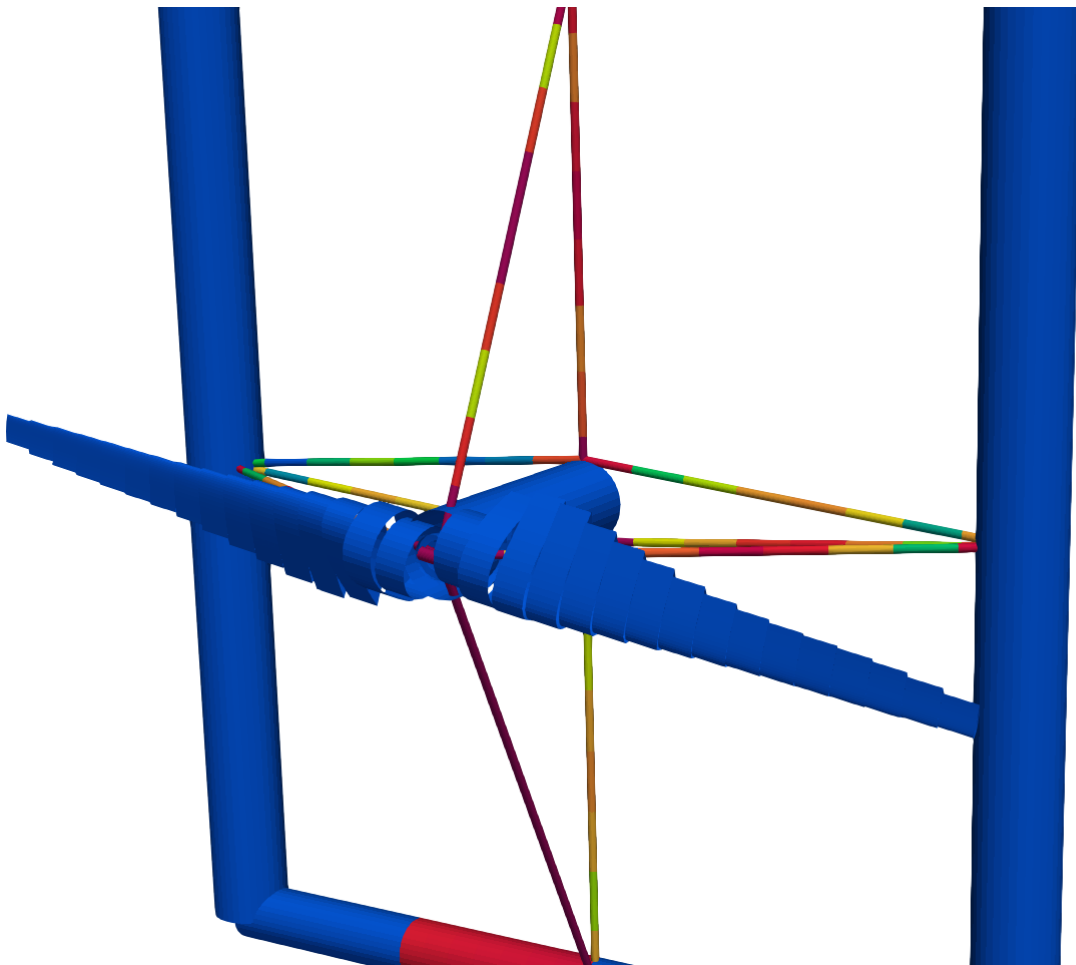


Figure 4.1 Stress distribution for extreme wind conditions from ParaView

4.1 Fatigue

NOTE

The fatigue life was assessed at eight different points of the cross-section for every monitored element. The most critical point for every element has been identified and presented in table 4.1. Elements in which the calculated fatigue life was more than 100 000 years have been labeled 'infinite'.

Element	Design Fatigue Life 2 blades (years)	Design Fatigue Life 4 blades (years)
front_hor_left_strut_1	1084	34081
front_hor_left_strut_2	1	10
front_hor_right_strut_1	1003	10660
front_hor_right_strut_2	1	26
front_vert_down_strut_1	3592	50793
front_vert_down_strut_2	4	56
front_vert_up_strut_1	3617	32551
front_vert_up_strut_2	2	27
rear_left_down_strut_1	5	21
rear_left_down_strut_2	35	299
rear_left_up_strut_1	36	486
rear_left_up_strut_2	810	8947
rear_right_down_strut_1	39	512
rear_right_down_strut_2	892	10519
rear_right_up_strut_1	5	23
rear_right_up_strut_2	104	593
rear_vert_down_strut	11	191
rear_vert_up_strut	13	557
shaft	infinite	infinite
blade root	infinite	infinite
rotor	infinite	infinite

Table 4.1 Assessed fatigue life for the structure's elements

4.2 Preliminary adjustment of Cross-sectional Properties

NOTE

Due to adjustments made to the wall thickness of the rear vertical struts, the SCFs changed significantly for the rear horizontal struts. Consequently, some of them ended up within acceptable fatigue life without any further adjustments.

4.2.1 2-bladed

Element	Diameter [mm]	Wall Thickness [mm]	Fatigue Life [Years]
Front_hor_left	300	24	33
Front_hor_right	300	24	32
Front_vert_down	300	12	31
Front_vert_up	300	16	38
Rear_left_down	200	8	71
Rear_left_up	200	2	72
Rear_right_down	200	2	77
Rear_right_up	200	8	64
Rear_vert	200	16	45
Blade_root	1200	2.8	95
Shaft	1000	4	98
Rotor	500	4	59

Table 4.2 Element properties and estimated fatigue life after 1st stage adjustments

4.2.2 4-bladed

Element	Diameter [mm]	Wall Thickness [mm]	Fatigue Life [Years]
Front_hor_left	200	16	33
Front_hor_right	200	12	57
Front_vert_down	200	8	56
Front_vert_up	200	12	57
Rear_left_down	200	8	60
Rear_left_up	100	2	86
Rear_right_down	100	2	44
Rear_right_up	200	8	65
Rear_vert	200	12	1465
Blade_root	1200	1.8	65
Shaft	1000	1.5	35
Rotor	500	3	45

Table 4.3 Element properties and estimated fatigue life after 1st stage adjustments

4.3 Extreme wind Von Mises Analysis

4.3.1 2-bladed

Element	Diameter [mm]	Wall Thickness [mm]	Von Mises Stress [MPa]
Front_hor_left	300	24	101
Front_hor_right	300	24	102
Front_vert_down	300	12	152
Front_vert_up	300	16	109
Rear_left_down	200	8	372
Rear_left_up	200	2	1456
Rear_right_down	200	2	1513
Rear_right_up	200	8	349
Rear_vert	200	16	238
Blade_root	1200	2.8	2266
Shaft	1000	4	630
Rotor	500	4	3005

Table 4.4 Element properties and Von Mises stress after 1st stage adjustments

4.3.2 4-bladed

Element	Diameter [mm]	Wall Thickness [mm]	Von Mises Stress [MPa]
Front_hor_left	200	16	300
Front_hor_right	200	12	386
Front_vert_down	200	8	378
Front_vert_up	200	12	315
Rear_left_down	200	8	372
Rear_left_up	100	2	5380
Rear_right_down	100	2	5320
Rear_right_up	200	8	349
Rear_vert	200	12	303
Blade_root	1200	1.8	1419
Shaft	1000	1.5	1637
Rotor	500	3	3005

Table 4.5 Element properties and Von Mises stress after 1st stage adjustments

4.4 Final Adjustments and verifying against buckling

NOTE (Buckling)

Negative values indicates that the member is under compression. Consequently, the combined fractional buckling load of inequality 3.10 must have values of < 1 in order to buckle. On the contrary, a positive value indicates that the member is under tension and is therefore not prone to buckling.

Element	Value of equation 3.10	Buckling (Yes/No)
front_hor_left_1	-5.97e-02	No
front_hor_left_2	-6.84e-02	No
front_hor_right_1	-6.01e-02	No
front_hor_right_2	-6.74e-02	No
front_vert_down_1	-1.79e-01	No
front_vert_down_2	-1.86e-01	No
front_vert_up_1	2.49e-02	No
front_vert_up2	2.42e-02	No

Table 4.6 Buckling assessment for 2-bladed structure in normal conditions

Element	Value of equation 3.10	Buckling (Yes/No)
front_hor_left_1	-6.66e-02	No
front_hor_left_2	-7.97e-02	No
front_hor_right_1	-6.77e-02	No
front_hor_right_2	-8.04e-02	No
front_vert_down_1	-1.80e-01	No
front_vert_down_2	-1.91e-01	No
front_vert_up_1	1.50e-02	No
front_vert_up2	6.34e-03	No

Table 4.7 Buckling assessment for 2-bladed structure in extreme conditions

Element	Diameter [mm]	Wall Thickness [mm]	Von Mises Stress [MPa]	Buckling (Yes/No)	Fatigue
Front_hor_left	300	24	101	No	Acceptable
Front_hor_right	300	24	102	No	Acceptable
Front_vert_down	300	12	152	No	Acceptable
Front_vert_up	300	16	109	No	Acceptable
Rear_left_down	200	10	305	No	Acceptable
Rear_left_up	200	10	324	No	Acceptable
Rear_right_down	200	10	334	No	Acceptable
Rear_right_up	200	10	287	No	Acceptable
Rear_vert	200	16	238	No	Acceptable
Blade_root	1200	8	324	No	Acceptable
Shaft	1000	8	330	No	Acceptable
Rotor	500	40	275	No	Acceptable

Table 4.8 Complete overview for the 2nd stage adjusted 2-bladed structure

Element	Value of equation 3.10	Buckling (Yes/No)
front_hor_left_1	-2.30e-01	No
front_hor_left_2	-2.59e-01	No
front_hor_right_1	-2.30e-01	No
front_hor_right_2	-2.55e-01	No
front_vert_down_1	-7.11e-01	No
front_vert_down_2	-7.40e-01	No
front_vert_up_1	2.11e-01	No
front_vert_up2	2.15e-01	No

Table 4.9 Buckling assessment for 4-bladed structure in normal conditions

Element	Value of equation 3.10	Buckling (Yes/No)
front_hor_left_1	-3.15e-01	No
front_hor_left_2	-3.59e-01	No
front_hor_right_1	-4.02e-01	No
front_hor_right_2	-4.57e-01	No
front_vert_down_1	-8.73e-01	No
front_vert_down_2	-9.18e-01	No
front_vert_up_1	7.73e-02	No
front_vert_up2	5.71e-02	No

Table 4.10 Buckling assessment for 4-bladed structure in extreme conditions

Element	Diameter [mm]	Wall Thickness [mm]	Von Mises Stress [MPa]	Buckling (Yes/No)	Fatigue
Front_hor_left	200	16	300	No	Acceptable
Front_hor_right	200	16	304	No	Acceptable
Front_vert_down	200	10	307	No	Acceptable
Front_vert_up	200	12	315	No	Acceptable
Rear_left_down	200	10	305	No	Acceptable
Rear_left_up	200	10	324	No	Acceptable
Rear_right_down	200	10	334	No	Acceptable
Rear_right_up	200	10	287	No	Acceptable
Rear_vert	200	12	303	No	Acceptable
Blade_root	1200	8	324	No	Acceptable
Shaft	1000	8	330	No	Acceptable
Rotor	500	40	275	No	Acceptable

Table 4.11 Complete overview for the 2nd stage adjusted 4-bladed structure

4.5 Comparison and Summary

Element	2-Bladed			
	Diameter [mm]	Wall Thickness [mm]	Mass [kg]	Limited by
Front_hor_left	300	24	2251	Fatigue
Front_hor_right	300	24	2251	Fatigue
Front_vert_down	300	12	1061	Fatigue
Front_vert_up	300	16	1395	Fatigue
Rear_left_down	200	10	514	Von Mises
Rear_left_up	200	10	514	Von Mises
Rear_right_down	200	10	514	Von Mises
Rear_right_up	200	10	514	Von Mises
Rear_vert	200	16	1804	Fatigue
Blade_root	1200	8	N/A	Von Mises
Shaft	1000	8	10	Von Mises
Rotor	500	40	45	Von Mises
Sum			10873	
	4-bladed			
Front_hor_left	200	16	904	Fatigue
Front_hor_right	200	16	904	Fatigue
Front_vert_down	200	10	646	Fatigue
Front_vert_up	200	12	768	Fatigue
Rear_left_down	200	10	514	Von Mises
Rear_left_up	200	10	514	Von Mises
Rear_right_down	200	10	514	Von Mises
Rear_right_up	200	10	514	Von Mises
Rear_vert	200	12	1382	Fatigue
Blade_root	1200	8	N/A	Von Mises
Shaft	1000	8	10	Von Mises
Rotor	500	40	45	Von Mises
Sum			6715	

Table 4.12 Summary and comparison between 2-bladed and 4-bladed rotor support structure

5. Discussion

5.1 Dynamic Behavior, Structural Response, and Fatigue Life Assessment

General observations made from the fatigue analysis results include the following:

- The fatigue life is consistently higher for the members in the four-bladed structure
- Average stress levels in the same members are not consistently higher for either of the structures
- Similar trend in the sense that both exhibit proportionally similar fatigue lives relative to other members in the same structure. For instance, the horizontal front struts are the most vulnerable for both, while the rotors are estimated to have the longest fatigue life.

Examining the stress histories for the '*front_vert_down_strut_1*' elements from appendix F might help explain some of the above observations. They share a similar average stress of about 85MPa, and a similar stress range spanning from 70-105 MPa. However, there is an obvious difference in the *thickness* of the stress history; evidence of more rapid fluctuations between stresses with a high degree of dissimilarity. The employed fatigue life estimations do not account for the average stress of the stress history, but are functions of stress ranges and number of cycles, explaining the discrepancy of the fatigue life estimations for this particular case.

The higher fluctuations experienced by the two-bladed structure is likely related to *balance*, as a two-bladed rotor is dynamically imbalanced. When the rotor is in such a configuration that the blades are in a completely vertical position, the turbine blades are subjected to different wind speeds. This results in a bending moment that needs to be balanced by the support structure. The four-bladed structure also has a blade configuration in which two of the blades are in the complete vertical position. However, these blades are smaller compared to those of the two-bladed rotor, thus experiencing smaller, but more rapid load variations.

Two-bladed rotors also suffer from complications related to yawing, since the turbine's moment of inertia changes as the position of the blades changes. When the blades are in a completely horizontal configuration, the turbine will experience a higher inertia to the yawing motion than when they are in the vertical configuration. If the turning force is

kept constant, the rotational speed of the nacelle will oscillate in line with the rotation of the turbine blades, resulting in stress fluctuations in the support structure.

5.2 Von Mises stress Assessment

Von Mises stress assessment was carried out for the fatigue-adjusted structure. The yielded results identified members in which the Von Mises stress was the limiting factor. Yielding was predicted in the shaft, blade roots, rotor, and all horizontal rear struts for both of the structures. The Von Mises stresses were calculated for a case of extreme wind conditions. When the rotor blades are not rotating, the responses of the different structures are very similar. The dynamic responses of the system is out of the equation. Consequently, despite having a different number of blades, the blades are designed such that the surface area facing the wind they cover, are identical (see 3.2.3). This results in very similar drag forces acting on the system, as well as similar stresses developing in the support structure. Modifications made to ensure the structure was verified against the Von Mises yield criterion were therefore the same for both structures.

5.3 Buckling Assessment

The last stage of the analysis involved verifying the front struts against buckling. None of the members were predicted to buckle under any of the load cases. The *front_vert_down_strut_2* element showed the highest probability of buckling, with a fractional buckle value of -0.918 on the four-bladed structure for the extreme wind load case. Essentially, this means that the strut is under 91.8% of its allowable buckle load. This is likely due to its position right beneath the nacelle, having to carry a big part of its weight.

5.4 Validity of the Results

The methods and formulae employed in this thesis are largely based on empirical data, with inherent errors and deviations. Both fatigue and buckling analyses are carried out according to design codes. Design codes usually account for imperfections in real life situations for their calculations. The structural members were verified against the Von Mises yield criterion with a safety factor of 1.05. While the Von Mises calculations themselves were not according to any design code, Eurocode typically divides the material's yield stress with 1.05 in their comparison criterions.

Throughout this thesis, assumptions and simplifications possibly affecting the results has been made. Arguments in favor of and against the vailidity of the results are presented below:

- The effect of the mean stress value is not accounted for in the fatigue analysis
- The S-N curves are shifted downwards by two standard deviations, and are thus associated with a 97.7% of survival
- Von Mises Stresses were only calculated at certain points in the structure and might be higher elsewhere
- Not all SCFs are within valid range 4.11, and are encouraged to be analyzed more carefully with Finite-Element-Analysis (FEA) for higher accuracy by DNVGL-RP-C203.
- The fatigue calculations can be viewed as being conservative considering the wind speeds from the *normal case* are relatively high. The average wind speed experienced by the rotor is 15 *m/s*
- In reality, the module will be a part of a bigger structure where forces may interfere with each other

6. Conclusion

Comparing the results from the structural analysis, it is clear that the two-bladed structure requires a more robust design in order to meet the design requirements. Examining the stress histories from the fatigue analysis of the different structures revealed differences in terms of stability. The more rapidly fluctuating stresses observed in the two-bladed structure reduces the estimated fatigue life compared to that of the four-bladed structure. A Von Mises stress analysis identified that certain members in the structures were not limited by fatigue loading, but were predicted to yield due to development of high stress levels, if the turbine was to be subjected to extreme wind speeds of up to 58 m/s . Adjustments to the cross-sectional properties were made accordingly. Finally, the structures' columns were verified against buckling. However, further adjustments were not needed at this stage.

The analyses culminated in the design of two support structures which have been verified against buckling and the Von Mises stress criterion, as well as being designed to have a fatigue life of more than 30 years.

6.1 Further Work

This work is not a complete analysis of the MRWT support structure. Many different simulation cases as well as more detailed structural analyses has been left for the future due to lack of time. Some ideas for future work include:

- Analysis of the support structure with flexible blades. The current analysis assumed the blades to have high stiffness.
- Run simulations with different load paths
- Analysis of the entire MRWT with tower and multiple rotor assembly.
- Cost analyses

References

- [1] “Wwea global capacity.” <https://library.wwindea.org/global-statistics/>. Accessed: 2021-01-23.
- [2] “Gwec global wind report 2019.” <https://gwec.net/global-wind-report-2019/>. Accessed: 2021-01-23.
- [3] “Wind energy in europe in 2019.” <https://windeurope.org/intelligence-platform/product/wind-energy-in-europe-in-2019-trends-and-statistics/>. Accessed: 2021-01-24.
- [4] J. F. M. et al., *WIND ENERGY EXPLAINED - Theory, Design and Application Second Edition*. Wiley, 2009.
- [5] A. Ashgar and X. Liu, “Online estimation of wind turbine tip speed ratio by adaptive neuro-fuzzy algorithm,” *Int. J. Adv. Comput. Sci. Appl*, vol. 9, no. 3, 2018.
- [6] “Offshore pioneers.” <https://www.siemensgamesa.com/products-and-services/offshore>. Accessed: 2021-04-23.
- [7] N. E. Dowling, *MECHANICAL BEHAVIORS OF MATERIALS - Engineering, Methods for Deformation, Fracture, and Fatigue Fourth Edition*. Pearson Education, 2013.
- [8] R. C. Hibbeler, *Mechanics of Materials Ninth Edition*. Pearson Education, 2014.
- [9] A. International, *Elements of Metallurgy and Engineering Alloys*. 2008.
- [10] Siemens, “What is a sn-curve?.” <https://community.sw.siemens.com/s/article/what-is-a-sn-curve>. Accessed: 2021-02-15.
- [11] G. Terjesen, *2. S-N kurven*. NMBU, 2020.
- [12] G. Terjesen, *1. Spenningsanalyse*. NMBU, 2020.
- [13] T. A. N. et al, *Development, Verification and Validation of 3DFloat; Aero-Servo-Hydro-Elastic Computations of Offshore Structures*. Science Direct, 2016.
- [14] D. G. AS, *DNVGL-RP-C203, Fatigue Design of Offshore Structures*. 2019.
- [15] H. BOOKS, “Stress concentration factors for simple tubular joints assessment of existing and development of new parametric,”
- [16] A. L. H. Hopstad, K. Argyriadis, A. Manjock, J. Goldsmith, and K. O. Ronold, “Dnv gl standard for floating wind turbines,” in *ASME 2018 1st International*

Offshore Wind Technical Conference, American Society of Mechanical Engineers Digital Collection, 2018.

- [17] european committee for standardization, *Eurocode 3: Design of steel structures -Part 1-1: General rules and rules for buildings*. 2005.
- [18] X. Meng, L. Gardner, A. J. Sadowski, and J. M. Rotter, “Elasto-plastic behaviour and design of semi-compact circular hollow sections,” *Thin-Walled Structures*, vol. 148, p. 106486, 2020.

Appendices

A. Python Routine For Partial Fatigue Damage Calculation

```
1 import numpy as np
2 import pylab as plt
3 import scipy
4
5 #####
6 ## Functions to calculate partial fatigue damage for welded ##
7 ## steel structures from timeseries of stress given in Pa ##
8 ## and two-sloped SN-curves as defined in DNV-OS-C203 Fatigue##
9 ## Design of Offshore Steel Structures. ##
10 ## ##
11 ## Stress concentration factors for hot-spot stress must be ##
12 ## included in the stress timeseries before these functions ##
13 ## are used. ##
14 ## ##
15 ## Stress cycles are counted using Rainflow counting. ##
16 ## ##
17 ## See example_fatigue_funcs.py for example of how to use ##
18 ## ##
19 ## Marit Kvittem Feb 2015 ##
20 #####
21
22 def turningpoints(x):
23
24     ## Find the amplitude at turning points of a 1D numpy array x
25
26     dx = np.diff(x)
27     Np = np.sum( dx[1:] * dx[:-1] < 0)
28     ind = np.where(dx[1:] * dx[:-1] < 0)
29     tp_m = x[ind]
30
31     ## add end points
32     tp = [x[0]]
33     tp.extend(tp_m)
34     tp.extend([x[-1]])
35
36     return tp
37
```

```

38 def turningpoints_steffen(x,amp): #Written by Steffen Aasen, April 2016
39     #save indexes of turning points
40     turningpoints=[]
41     indexes=[]
42     for i in range(1,len(x)-1):
43         if x[i-1]>x[i] and x[i+1]>x[i] or x[i-1]<x[i] and x[i+1]<x[i]:
44             indexes.append(i)
45     #make array with turningpoints
46     for element in indexes:
47         if abs(x[element-1]-x[element])>amp and abs(x[element+1]-x[eleme
48             nt])>amp:
49             turningpoints.append(x[element])
50     #delete points that are not turningpoints (due to numerical error)
51     indexes=[]
52     for i in range(len(turningpoints)-2):
53         if turningpoints[i+1]>turningpoints[i] and turningpoints[i+2]>
54             turningpoints[i+1] or turningpoints[i+1]<turningpoints[i] and
55             turningpoints[i+2]<turningpoints[i+1]:
56             indexes.append(i+1)
57     turningpoints_2=[]
58     for i in range(len(turningpoints)):
59         if i not in indexes:
60             turningpoints_2.append(turningpoints[i])
61     return turningpoints_2
62 def findrfc_wafo(x):
63
64     ## Rainflow counting of 1D list of turning points x
65     ## based on matlab wafo's tp2arfc4p and default values given in tp2rfc
66
67     def_time=0.
68     res0 = []
69     T = len(x)
70     ARFC = np.zeros((int(np.floor(T/2)),2))
71     N = -1
72
73     res = np.zeros(max([200,len(res0)]))
74
75     nres = -1
76
77     for i in range(0,T):
78         nres = nres+1
79         res[nres] = x[i]
80         cycleFound = 1
81         while cycleFound ==1 and nres >=4:
82             if res[nres-1] < res[nres-2]:
83                 A = [res[nres-1], res[nres-2]]

```

```

84     else:
85         A = [res[nres-2], res[nres-1]]
86
87     if res[nres] < res[nres-3]:
88         B = [res[nres], res[nres-3]]
89     else:
90         B = [res[nres-3], res[nres]]
91
92     if A[0] >= B[0] and A[1] <= B[1]:
93         N=N+1
94         arfc = [[res[nres-2]],[res[nres-1]]]
95         ARFC[N] = [res[nres-2],res[nres-1]]
96         res[nres-2] = res[nres]
97         nres = nres-2
98     else:
99         cycleFound = 0
100     ## residual
101     res = res[0:nres+1]
102
103     def res2arfc(res):
104         nres = len(res)
105         ARFC = []
106         if nres < 2:
107             return
108         ## count min to max cycles, gives correct number of upcrossings
109         if (res[1]-res[0]) > 0.:
110             i_start = 0
111         else:
112             i_start=1
113         I = range(i_start, nres-1,2)
114         Ip1 = range(i_start+1,nres,2)
115         ## def_time = 0
116
117         for ii in range(len(I)):
118             ARFC.append( [res[I[ii]], res[Ip1[ii]]] )
119
120         ARFC = np.array(ARFC)
121
122         return ARFC
123
124     ARFC_res = res2arfc(res)
125
126     ARFC = np.concatenate((ARFC,ARFC_res))
127
128     ## make symmetric
129     [N,M] = np.shape(ARFC)
130     I = []
131     J=0
132     RFC = ARFC

```

```

133
134 for ii in range(N):
135     if ARFC[ii ,0] > ARFC[ii ,1]:
136         ## Swap variables
137         RFC[ii ,J],RFC[ii ,J+1] = RFC[ii ,J+1],RFC[ii ,J]
138
139 cc = RFC
140
141 rfcamp = (RFC[:,1] - RFC[:,0]) /2.
142
143 return rfcamp
144
145
146
147 def fatiguedamage_twoslope(time ,stress ,m1,loga1 ,m2,loga2 ,Nlim ,th=25E-3,
    tref=25E-3,k=0.25):
148     ## stress: stressvector , unit: Pa
149     ## m1, loga1 , m2, loga2: Parameters from table 2.2 in RP-C203
150     ## Note that the parameters in RP C203 are given for stress ranges in
    MPa
151     ## tref , k: perameters from point 2.4 in RP-C203
152     ## th: structural detail thickness
153     ## Calculates fatigue damage for bilinear SN curves
154     ## hist: true/false parameter, wether or not to plot histogram
155
156     stress = stress*1.E-6
157
158
159     tp = turningpoints_steffen(stress ,0.0) ## Find turning points
160     mm = findrfc_wafo(tp) ## Rainflow cycles as by the routine in matlab wafo
161
162     Nbins = len(mm)
163
164     if th<tref:
165         th = tref
166
167     a1 = 10.**loga1
168     K1 = 2.0**m1/a1*(th/tref)**(k*m1)
169     beta1 = m1
170
171     a2 = 10.**loga2
172     K2 = 2.0**m2/a2*(th/tref)**(k*m2)
173     beta2 = m2
174
175
176
177     alim = (1.0/(K1*Nlim))**(1./m1)
178     alim2 = (1.0/(K2*Nlim))**(1./m2)
179

```

```
180
181 avalid = (1.0/(K2*1.0E7))**(1./m2)
182
183 if not np.round(alim,0) == np.round(alim2,0):
184     print(loga1)
185     print(loga2)
186     print(m1)
187     print(m2)
188     print(K1)
189     print(K2)
190     print(alim)
191     print(alim2)
192     print('alim not the same as alim2, check SN curve values')
193
194 dd = 0.0
195
196 amp = abs(mm)
197
198 for aa in amp:
199     if aa > alim:
200         dd = dd + K1*aa**beta1
201
202     elif aa <= alim:
203         if aa < avalid:
204             key = True
205
206         dd = dd + K2*aa**beta2
207     else:
208         dd = dd + K2*aa**beta2
209
210 D_T = dd
211
212 return D_T
```

B. Python Routine For Calculating stresses around the circumference of a CHS

```
1  ## coding: utf-8 ##
2  '''
3  Created on Fri Apr 16 13:10:24 2021
4
5  @author: Vetle Birkeland Aass
6  '''
7
8  from pylab import *
9  from numpy import loadtxt
10
11
12
13 def cyl_beam_stresses(t, fx, fy, fz, mx, my, mz, d, twall, SCF_AS, SCF_AC, SCF_MIP
    , SCF_MOP):
14     #
15     # Stresses for box beam
16     # Signs are for monitors set for node 2. Forces can be seen as acting
       on node 2
17     # from the neighbor element
18     # Calculation points in element coordinate system:
19     # Point 1 on y axis, then positive rotation about x axis
20     #
21     # Point      y      z
22     # 1          0.5*d   0
23     # 2          0.5*d*cos45  0.5*d*sin45
24     # 3          0      b/2
25     # and so on
26     # iy means bending stiffness ABOUT y axis, integral of z^2dA
27
28     di = d - 2.*twall
29     a = .25*pi*(d**2-di**2)
30     icyl = pi*(d**4-di**4)/64.
31
32     # normal stresses for points 1-8
33     sax = fx/a # axial stress, positive tension
```

```

34  sbendy = my*.5*d/icyl # positive moment gives tension (positive for
      pos z)
35  sbendz = -mz*.5*d/icyl # positive moment gives tension (positive
      for neg y)
36
37  sin45 = sin(pi*45./180.) # z coordinate
38  cos45 = cos(pi*45./180.) # y coordinate
39
40
41  s1      = SCF_AC*sax                                     +
      SCF_MOP*sbendz
42  s2      = 0.5*(SCF_AC+SCF_AS)*sax + SCF_MIP*sbendy*sin45 +
      SCF_MOP*sbendz*cos45
43  s3      = SCF_AS*sax + sbendy
44  s4      = 0.5*(SCF_AC+SCF_AS)*sax + SCF_MIP*sbendy*sin45 -
      SCF_MOP*sbendz*cos45
45  s5      = SCF_AS*sax                                     -
      SCF_MOP*sbendz
46  s6      = 0.5*(SCF_AC+SCF_AS)*sax - SCF_MIP*sbendy*sin45 -
      SCF_MOP*sbendz*cos45
47  s7      = SCF_AS*sax                                     - SCF_MIP*sbendy
48  s8      = 0.5*(SCF_AC+SCF_AS)*sax - SCF_MIP*sbendy*sin45 +
      SCF_MOP*sbendz*cos45
49
50  # shear stress
51  symax = fy/a*(4./3)*(d**2 + d*di + di**2)/(d**2 + di**2) # on z axis
52  szmax = fz/a*(4./3)*(d**2 + d*di + di**2)/(d**2 + di**2) # on y axis
53  # torsion stress, thin wall approx
54  dm = .5*(d+di)
55  sx = mx/(2.*twall*.25*pi*dm**2)
56
57  # shear stresses, positive along section, positive x rotation
58  ssh1 = sx + szmax
59  ssh2 = sx # ignore shear stress due to shear forces here
60  ssh3 = sx - symax
61  ssh4 = sx # ignore shear stress due to shear forces here
62  ssh5 = sx - szmax
63  ssh6 = sx
64  ssh7 = sx + symax
65  ssh8 = sx
66
67  # von Mises stress
68  svm1 = sqrt(s1**2 + 3.*ssh1**2)
69  svm2 = sqrt(s2**2 + 3.*ssh2**2)
70  svm3 = sqrt(s3**2 + 3.*ssh3**2)
71  svm4 = sqrt(s4**2 + 3.*ssh4**2)
72  svm5 = sqrt(s5**2 + 3.*ssh5**2)
73  svm6 = sqrt(s6**2 + 3.*ssh6**2)
74  svm7 = sqrt(s7**2 + 3.*ssh7**2)

```



```
75     svm8 = sqrt(s8**2 + 3.*ssh8**2)
76
77
78     return [sax , sbendy , sbendz , \
79             s1 ,  s2 ,  s3 ,  s4 ,  s5 ,  s6 ,  s7 ,  s8 ,  \
80             ssh1 , ssh2 , ssh3 , ssh4 , ssh5 , ssh6 , ssh7 , ssh8 , \
81             svm1 , svm2 , svm3 , svm4 , svm5 , svm6 , svm7 , svm8 , \
82             sx , symax , szmax]
```

C. 3DFloat input

C.0.1 2-bladed rotor

data.txt

```
!  
! 3Dfloat input file.  
!  
! From C:\Users\toran\OneDrive - Institutt for Energiteknikk\project2\oc4_semisub_2013\revisit_2020\load_cases\3.2_flex\r5  
!  
! Single rotor/rna/attachments for WCS  
  
r2: resolved blades  
r3: Danwin180-like blades  
r4: tuned twist and chord, as C:\Users\toran\OneDrive - Institutt for Energiteknikk\project2\wcs_2020\rigid_rotor\r2  
r5: added generator, test run 15 - 20m/s  
r6: trimmed back root chord  
r7: const lambda to check cp and ct  
r8: new generator, more sensors, variable speed run, effects of gravity and shear  
r9: mores sensors, turbulence  
  
! 2 bladed rotor, from wcs_2020\rotor\r9  
! r1: from 4 to 2 blades, double chord length and mass: same overall rotor mass  
! r2: turbulence, as 4 bladed r9, new generator  
  
\vetle_sjekk_2021_4_30\wcsrotor-2blad\r1 som mottatt  
r2: med tyngde !  
r3: extra element and sensor for nonrotating system, shaft  
r4: with turbulence  
  
use_sparse_system_matrices  
  
set_environment  
rho_water: 1025.  
rho_air: 1.225  
nu_water: 1.e-5  
nu_air: 1.e-6  
gravity: on !TAN skal v on  
buoyancy: off on  
hydro_force: none morison  
wind_force: drag  
wind: mean_profile  
wind_ref_height: 30.  
wind_speed: 0.  
wind_direction: 0.  
wind_exponent: 0.14  
waves: extrapolated_airy !irregular_waves  
norder: -1  
wave_amplitude: 1.e-6  
wave_direction: 0.  
spectral_peak: 19.2  
depth: 200.  
surface_option: 2ndorder  
current_speed: 0.  
current_direction: 0.  
current_exponent: 0.  
wave_ramps: 0.  
wave_rampe: 200.  
tshift_waves: 0.  
tshift_wind: 0.  
kinematics_option: updated  
tc_kin: 0.  
dtwkin: -1.  
scale_wind: 1.  
roughness_length -99.  
pre_computed_waves_file: off
```

```

!jonswap_wavelets_constant
hs: 6.
tp: 10.
tstart: -1.
tcut: -1.
gamma: 2.87
tperiodic: 3600.
depth: 200.0
spread -99.
file: wave_table_hs6_tp10_gam2.87.txt

!wavelets
scale_amplitude: 1.
file: wave_table_hs6_tp10_gam2.87.txt

!test_irreg_airy
dt: .1
nstep: 3600
x: 0. 0. 0.
filename: waveheight_wave_table_hs6_tp10_gam2.87.txt

wave_forces
cd_morison: 0.7
cm_morison: 1.63

!
!!! Defining materials needed to construct model:
!

define_material name s355 rho 7850.0 e 210.e9 g 81.e9
define_material name blade_dummy rho: 0.0001 e: 210.e9 g: 80.8e9
define_material name generator_material rho 0.0001 e 210.e9 g 80.8e9

define_material name semisub rho 7800.0 e 2.1e+11 g 81000000000.0
define_material name HPmaterial rho 3900.0 e 2.1e+13 g 8.1e+12
define_material name connect_mat rho 1e-08 e 2.1e+15 g 8.1e+14
define_material name connect_mat0 rho 1e-08 e 2.1e+13 g 8.1e+12
define_material name masslessStiff1 rho 0.0001 e 2.1e+14 g 8.08e+13
define_material name wirewb_oc3 rho 11189.705669 e 60399180000.0 g 22300000000.0
define_material name chain0 rho 7904.5462588 e 56000000000.0 g 22300000000.0
define_material name chain1 rho 12743.9021729 e 56000000000.0 g 22300000000.0
define_material name chain2 rho 23196.1475439 e 56000000000.0 g 22300000000.0
define_material name chain3 rho 7875.98230769 e 34609101925.5 g 13781897867.8
define_material name OC4semisub rho 7850.0 e 2.1e+13 g 7.89473684211e+12
define_material name wire_OC4 rho 24596.5447981 e 1.63528506042e+11 g 65119666311.8
define_material name s35grout rho 9153.29723 e 210.e9 g 80.8e9
define_material name s35grout2 rho 3425.024110 e 210.e9 g 80.8e9
define_material name massless1 rho 0.0001 e 210.e9 g 80.8e9
define_material name massless2 rho 0.0001 e 210.e9 g 80.8e9 ! pitch actuator elements
define_material name trans_piece rho 1725.2536669 e 210.e9 g 80.8e9
define_material name mainshaftStiff rho 0.0001 e 210.e12 g 1056.10143e9
define_material name mainshaft rho 0.0001 e 210.e9 g 1056.10143e6
define_material name mainshaft2 rho 0.0001 e 210.e9 g 503745286.6525068 ! tuned to new shaft element
define_material name mainshaft_stiff rho 0.0001 e 210.e9 g 80.8e9
define_material name s355j2g3stf rho 7850. e 210.e9 g 81.e9
define_material name steeltowerMassless rho 0.00001 e 2.1e+11 g 8.08e+11
! g tuned in mainshaft to give correct drivetrain torsional stiffness

!
! Geometry definition
!

! Dummy element to provide reference node to body

new_elements_beam1 nelem 1 material s355 connect_mode no_connect
x1 -1. 0. 30. dcy11 1. tcyl1 .1
x2 0. 0. 30. dcy12 1. tcyl2 .1 btype beamwb cdn 0.

new_body
name: rotor_1
parent: inertial
reftype: refnode
xfind: 0. 0. 30.
rotorder: 123
rotation: 0. 0. 0.
axes: normal
tc_kin: 0. # geometry def from here to next body def will be relative to xfind pos

```

```

!/ frame with cc h = 25m, b = 22m

new_elements_beam1 nelelem 4 material s355 connect_mode no_connect
x1 0. -11. -12.5 dcy11 1.5 tcy11 0.016 pre_strain_long 0.0
x2 0. 11. -12.5 dcy12 1.5 tcy12 0.016 btype beam
frac_stiff 1.0 cdn 0.

new_elements_beam1 nelelem 4 material s355 connect_mode no_connect
x1 0. -11. 12.5 dcy11 1.5 tcy11 0.016 pre_strain_long 0.0
x2 0. 11. 12.5 dcy12 1.5 tcy12 0.016 btype beam
frac_stiff 1.0 cdn 0.

new_elements_beam1 nelelem 4 material s355 connect_mode connect_both
x1 0. -11. -12.5 dcy11 2. tcy11 0.02 pre_strain_long 0.0
x2 0. -11. 12.5 dcy12 2. tcy12 0.02 btype beam
frac_stiff 1.0 cdn 0.

new_elements_beam1 nelelem 4 material s355 connect_mode connect_both
x1 0. 11. -12.5 dcy11 2. tcy11 0.02 pre_strain_long 0.0
x2 0. 11. 12.5 dcy12 2. tcy12 0.02 btype beam
frac_stiff 1.0 cdn 0.

!/ nacelle

new_elements_beam1 nelelem 1 material s355 connect_mode no_connect
x1 -6. 0. 0. dcy11 2. tcy11 0.08 pre_strain_long 0.0
x2 0.01 0. 0. dcy12 2. tcy12 0.08 btype beam
frac_stiff 1.0 cdn 0.

!TAN force sensor, nonrotating system
new_elements_beam1 nelelem 1 material s355 connect_mode connect_2
x1 -6.05 0. 0. dcy11 1. tcy11 0.1 pre_strain_long 0.0
x2 -6. 0. 0. dcy12 1. tcy12 0.1 btype beam
frac_stiff 1.0 cdn 0.

new_elements_beam1 nelelem 1 material generator_material connect_mode connect_2
x1 -6.1 0. 0. dcy11 1. tcy11 0.1 pre_strain_long 0.0
x2 -6.05 0. 0. dcy12 1. tcy12 0.1 btype beam
frac_stiff 1.0 cdn 0. generator: var_rpm_2bl !const_rpm !

!/ force sensor, rotor
new_elements_beam1 nelelem 1 material s355 connect_mode connect_2
x1 -6.2 0. 0. dcy11 1. tcy11 0.03 pre_strain_long 0.0
x2 -6.1 0. 0. dcy12 1. tcy12 0.03 btype beam
frac_stiff 1.0 cdn 0.

! this is used only when forcing constant rpm on rotor
!! [rad/s] [Nm] ! mgen here is mechanical. El power = omegagenerator name: const_rpm gen_eta: 1. npoints_ave: 5 max_trq_rate: 2000.e3 ! Spin up in 100s
scale_omega: 1. scale_mgen: 1.
omega: mgen:
0. -48.e7
10. -48.e7
omega_eta: eta:
0. 1.
10. 1.

! 4-bladed rotor
! generator for variable speed at tsr 6 cp 0.49680306]
! [rad/s] [Nm] ! mgen here is mechanical. El power = omegagenerator name: var_rpm gen_eta: 1. npoints_ave: 2 max_trq_rate: 20000.e3
scale_omega: 1. scale_mgen: 0.9454522618354058
omega: mgen:
0. 0.
2.4 0. 7779.631483486891 ! Zero moment also at 5m/s for easier starting. So valid only above 6m/s
2.88 11202.669336221124
3.36 15248.077707634307
3.84 19915.856597726444
4.32 25206.00600649753
4.8 31118.525933947563
5.28 37653.41638007656
5.76 44810.6773448845
6.24 52590.30882837139
6.72 60992.31083053723
7.2 70016.68335138202
7.68 79663.42639090578
8.16 89932.53994910847
8.64 100824.02402599012
9.12 112337.87862155074
9.6 124474.10373579025
10.08 137232.69936870877

```

```

10.56 150613.66552030624
11.04 164617.00219058266
11.52 179242.709379538
12.0 194490.78708717227
12.48 210361.23531348555
12.96 226854.05405847775
13.44 243969.24332214892
13.92 261706.80310449903
14.4 280066.73340552807
omega_eta: eta:
0. 1.
10. 1.

! 2-bladed rotor
! generator for variable speed at tsr 6 cp 0.4568
!
generator name: var_rpm_2bl gen_eta: 1. npoints_ave: 2 max_trq_rate: 20000.e3
scale_omega: 1. scale_mgen: 1.
omega: mgen:
0. 0.
2.4 0. 7153.2699999999995
2.88 10300.7088
3.36 14020.409200000002
3.84 18312.371200000005
4.32 23176.594800000003
4.8 28613.079999999998
5.28 34621.8268
5.76 41202.8352
6.24 48356.105200000005
6.72 56081.636800000001
7.2 64379.43
7.68 73249.484800000002
8.16 82691.8012
8.64 92706.379200000001
9.12 103293.218800000002
9.6 114452.31999999999
10.08 126183.6828
10.56 138487.3072
11.04 151363.1932
11.52 164811.3408
12.0 178831.75
12.48 193424.420800000002
12.96 208589.3532
13.44 224326.547200000003
13.92 240636.002800000002
14.4 257517.72
omega_eta: eta:
0. 1.
10. 1.

! this is used only when forcing constant rpm on rotor
!material_specific_damping material generator_material type rayleigh alpha 0. beta 0.
! damping_matrix: add_matrix_noscaling ! terminate with blank line
! 4 4 10.e7
! 10 10 10.e7
! 4 10 -10.e7
! 10 4 -10.e7 ! Should give 48/10 = 4.8 rad/s

! / torque arms to torque struts

new_elements_beam1 nelem 1 material massless connect_mode connect_1
x1 0.01 0. 0. dcy11 2. tcy11 0.08 pre_strain_long 0.0
x2 0.01 0. 1. dcy12 2. tcy12 0.08 btype beamwb
frac_stiff 1.0 cdn 1.

new_elements_beam1 nelem 1 material massless connect_mode connect_1
x1 0.01 0. 0. dcy11 2. tcy11 0.08 pre_strain_long 0.0
x2 0.01 0. -1. dcy12 2. tcy12 0.08 btype beamwb
frac_stiff 1.0 cdn 1.

! / struts nacelle to frame

mass tag struts start

```

```

! front horizontal
! front_hor_left
new_elements_beam1 nelem 9 material s355 connect_mode connect_both
x1 -6. 0. 0. dcy11 .2 tcyl1 0.008 pre_strain_long 0.0
x2 0. -11. 0. dcy12 .2 tcyl2 0.008 btype beam
frac_stiff 1.0 cdn 1. ! member midpoint in body system: -3.0000E+00 -5.5000E+00 0.0000E+00

! front_hor_right
new_elements_beam1 nelem 9 material s355 connect_mode connect_both
x1 -6. 0. 0. dcy11 .2 tcyl1 0.008 pre_strain_long 0.0
x2 0. 11. 0. dcy12 .2 tcyl2 0.008 btype beam
frac_stiff 1.0 cdn 1. ! member midpoint in body system: -3.0000E+00 5.5000E+00 0.0000E+00

! front vertical
! front_vert_down
new_elements_beam1 nelem 9 material s355 connect_mode connect_both
x1 -6. 0. 0. dcy11 .2 tcyl1 0.008 pre_strain_long 0.0
x2 0. 0. -12.5 dcy12 .2 tcyl2 0.008 btype beam
frac_stiff 1.0 cdn 1. ! member midpoint in body system: -3.0000E+00 0.0000E+00 -6.2500E+00

!front_vert_up
new_elements_beam1 nelem 9 material s355 connect_mode connect_both
x1 -6. 0. 0. dcy11 .2 tcyl1 0.008 pre_strain_long 0.0
x2 0. 0. 12.5 dcy12 .2 tcyl2 0.008 btype beam
frac_stiff 1.0 cdn 1. ! member midpoint in body system: -3.0000E+00 0.0000E+00 6.2500E+00

!rear horizontal avoid connection to dummy node in 0,0,0
! rear_left_up
new_elements_beam1 nelem 8 material s355 connect_mode connect_both
x1 0.01 0. 1. dcy11 .2 tcyl1 0.008 pre_strain_long 0.0
x2 0. -11. 0. dcy12 .2 tcyl2 0.008 btype beam
frac_stiff 1.0 cdn 1. ! member midpoint in body system: 5.0000E-03 -5.5000E+00 5.0000E-01

! rear_left_down
new_elements_beam1 nelem 8 material s355 connect_mode connect_both
x1 0.01 0. -1. dcy11 .2 tcyl1 0.008 pre_strain_long 0.0
x2 0. -11. 0. dcy12 .2 tcyl2 0.008 btype beam
frac_stiff 1.0 cdn 1. ! member midpoint in body system: 5.0000E-03 -5.5000E+00 -5.0000E-01

! rear_right_up
new_elements_beam1 nelem 8 material s355 connect_mode connect_both
x1 0.01 0. 1. dcy11 .2 tcyl1 0.008 pre_strain_long 0.0
x2 0. 11. 0. dcy12 .2 tcyl2 0.008 btype beam
frac_stiff 1.0 cdn 1. ! member midpoint in body system: 5.0000E-03 5.5000E+00 5.0000E-01

! rear_right_down
new_elements_beam1 nelem 8 material s355 connect_mode connect_both
x1 0.01 0. -1. dcy11 .2 tcyl1 0.008 pre_strain_long 0.0
x2 0. 11. 0. dcy12 .2 tcyl2 0.008 btype beam
frac_stiff 1.0 cdn 1. ! member midpoint in body system: 5.0000E-03 5.5000E+00 -5.0000E-01

! rear vertical
! rear_vert_down
new_elements_beam1 nelem 8 material s355 connect_mode connect_both
x1 0.01 0. 0. dcy11 .2 tcyl1 0.008 pre_strain_long 0.0
x2 0. 0. -12.5 dcy12 .2 tcyl2 0.008 btype beam
frac_stiff 1.0 cdn 1. ! member midpoint in body system: 5.0000E-03 0.0000E+00 -6.2500E+00

! rear_vert_up
new_elements_beam1 nelem 8 material s355 connect_mode connect_both
x1 0.01 0. 0. dcy11 .2 tcyl1 0.008 pre_strain_long 0.0
x2 0. 0. 12.5 dcy12 .2 tcyl2 0.008 btype beam
frac_stiff 1.0 cdn 1. ! member midpoint in body system: 5.0000E-03 0.0000E+00 6.2500E+00

mass tag struts end

! -----
! WCS 12.5m blade v0: STRUCTURAL AND AERODYNAMIC PROPERTIES
! -----
!
! blade element structural coordinate system:
!
! y out TE along flap principal axis
! z out suction side along lag principal axis
! x = y x z ! NB CAN THEREFORE BE TOWARDS TIP OR HUB
!
! structural twist is positive around x
! mass center and shear center offsets are given in blade element structural

```

```

! coordinate system
!
!
! r[M] EI_f[Nm!
old_bl_table blname: wcs_v0_blade
0. 6.00000000e+07 6.00000000e+07 6.00000000e+10 2.E+10 3.e2 2.050e+01 0. 0. 0. 0. 0. 0. 0. 0.
0.375 6.00000000e+07 6.00000000e+07 6.00000000e+10 2.E+10 3.e2 2.050e+01 0. 0. 0. 0. 0. 0. 0. 0.
0.75 6.00000000e+07 6.00000000e+07 6.00000000e+10 2.E+10 3.e2 2.050e+01 0. 0. 0. 0. 0. 0. 0. 0.
1.13362832e+00 6.00000000e+07 6.00000000e+07 6.00000000e+10 2.E+10 2.28333333e+02 2.050e+01 0. 0. 0. 0. 0. 0. 0. 0.
1.67345133e+00 6.00000000e+07 6.00000000e+07 6.00000000e+10 2.E+10 1.28666667e+02 2.060e+01 0. 0. 0. 0. 0. 0. 0. 0.
2.26725664e+00 5.00000000e+07 6.50000000e+07 6.50000000e+10 2.E+10 1.07666667e+02 2.010e+01 0. 0. 0. 0. 0. 0. 0. 0.
2.91504425e+00 3.50000000e+07 6.80000000e+07 6.80000000e+10 2.E+10 8.45000000e+01 1.700e+01 0. 0. 0. 0. 0. 0. 0. 0.
3.56283186e+00 2.50000000e+07 5.00000000e+07 5.00000000e+10 2.E+10 7.11666667e+01 1.210e+01 0. 0. 0. 0. 0. 0. 0. 0.
4.21061947e+00 1.71000000e+07 3.70000000e+07 3.70000000e+10 2.E+10 6.20000000e+01 8.470e+00 0. 0. 0. 0. 0. 0. 0. 0.
4.85840708e+00 1.10000000e+07 3.10000000e+07 3.10000000e+10 2.E+10 5.80000000e+01 6.040e+00 0. 0. 0. 0. 0. 0. 0. 0.
5.50619469e+00 7.00000000e+06 2.50000000e+07 2.50000000e+10 2.E+10 5.33333333e+01 4.370e+00 0. 0. 0. 0. 0. 0. 0. 0.
6.15398230e+00 4.50000000e+06 2.20000000e+07 2.20000000e+10 2.E+10 4.80000000e+01 3.040e+00 0. 0. 0. 0. 0. 0. 0. 0.
6.80176991e+00 3.40000000e+06 1.80000000e+07 1.80000000e+10 2.E+10 4.45000000e+01 2.050e+00 0. 0. 0. 0. 0. 0. 0. 0.
7.44955752e+00 2.40000000e+06 1.58000000e+07 1.58000000e+10 2.E+10 4.15000000e+01 1.330e+00 0. 0. 0. 0. 0. 0. 0. 0.
8.09734513e+00 1.70000000e+06 1.30000000e+07 1.30000000e+10 2.E+9 3.95000000e+01 7.800e-01 0. 0. 0. 0. 0. 0. 0. 0.
8.74513274e+00 1.10000000e+07 1.10000000e+07 1.10000000e+10 2.E+9 3.70000000e+01 4.200e-01 0. 0. 0. 0. 0. 0. 0. 0.
9.39292035e+00 8.00000000e+05 9.11000000e+06 9.11000000e+09 2.E+9 3.50000000e+01 2.200e-01 0. 0. 0. 0. 0. 0. 0. 0.
1.00407080e+01 6.50000000e+05 6.50000000e+06 6.50000000e+09 2.E+9 3.02500000e+01 1.100e-01 0. 0. 0. 0. 0. 0. 0. 0.
1.05805310e+01 6.50000000e+05 5.30000000e+06 5.30000000e+09 2.E+9 2.32500000e+01 0.000e+00 0. 0. 0. 0. 0. 0. 0. 0.
1.10123894e+01 6.50000000e+05 4.90000000e+06 4.90000000e+09 2.E+9 2.25000000e+01 0.000e+00 0. 0. 0. 0. 0. 0. 0. 0.
1.15522124e+01 6.50000000e+05 4.60000000e+06 4.60000000e+09 2.E+8 2.65833333e+01 0.000e+00 0. 0. 0. 0. 0. 0. 0. 0.
1.22000000e+01 6.50000000e+05 4.50000000e+06 4.50000000e+09 2.E+8 2.70000000e+01 0.000e+00 0. 0. 0. 0. 0. 0. 0. 0.
1.25100000e+01 6.50000000e+05 4.50000000e+06 4.50000000e+09 2.E+8 0. 0.00e+00 0. 0. 0. 0. 0. 0. 0. 0.

blade_table blname: wcs_v0_blade scale_mass: 2. scale_stiff: 1.
0. 6.00000000e+10 6.00000000e+10 6.00000000e+10 2.E+10 3.e2 2.05000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
0.375 6.00000000e+10 6.00000000e+10 6.00000000e+10 2.E+10 3.e2 2.05000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
0.75 6.00000000e+10 6.00000000e+10 6.00000000e+10 2.E+10 3.e2 2.05000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
1.13362832e+00 6.00000000e+10 6.00000000e+10 6.00000000e+10 2.E+10 2.28333333e+02 2.05000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
1.67345133e+00 6.00000000e+10 6.00000000e+10 6.00000000e+10 2.E+10 1.28666667e+02 2.06000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
2.26725664e+00 5.00000000e+10 6.50000000e+10 6.50000000e+10 2.E+10 1.07666667e+02 2.01000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
2.91504425e+00 3.50000000e+10 6.80000000e+10 6.80000000e+10 2.E+10 8.45000000e+01 1.70000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
3.56283186e+00 2.50000000e+10 5.00000000e+10 5.00000000e+10 2.E+10 7.11666667e+01 1.21000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
4.21061947e+00 1.71000000e+10 3.70000000e+10 3.70000000e+10 2.E+10 6.20000000e+01 8.47000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
4.85840708e+00 1.10000000e+10 3.10000000e+10 3.10000000e+10 2.E+10 5.80000000e+01 6.04000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
5.50619469e+00 7.00000000e+10 2.50000000e+10 2.50000000e+10 2.E+10 5.33333333e+01 4.37000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
6.15398230e+00 4.50000000e+10 2.20000000e+10 2.20000000e+10 2.E+10 4.80000000e+01 3.04000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
6.80176991e+00 3.40000000e+10 1.80000000e+10 1.80000000e+10 2.E+10 4.45000000e+01 2.05000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
7.44955752e+00 2.40000000e+10 1.58000000e+10 1.58000000e+10 2.E+10 4.15000000e+01 1.33000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
8.09734513e+00 1.70000000e+10 1.30000000e+10 1.30000000e+10 2.E+9 3.95000000e+01 7.80000000e-01 0. 0. 0. 0. 0. 0. 0. 0.
8.74513274e+00 1.10000000e+10 1.10000000e+10 1.10000000e+10 2.E+9 3.70000000e+01 4.20000000e-01 0. 0. 0. 0. 0. 0. 0. 0.
9.39292035e+00 8.00000000e+10 9.11000000e+10 9.11000000e+09 2.E+9 3.50000000e+01 2.20000000e-01 0. 0. 0. 0. 0. 0. 0. 0.
1.00407080e+01 6.50000000e+10 6.50000000e+10 6.50000000e+09 2.E+9 3.02500000e+01 1.10000000e-01 0. 0. 0. 0. 0. 0. 0. 0.
1.05805310e+01 6.50000000e+10 5.30000000e+10 5.30000000e+09 2.E+9 2.32500000e+01 0.00000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
1.10123894e+01 6.50000000e+10 4.90000000e+10 4.90000000e+09 2.E+9 2.25000000e+01 0.00000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
1.15522124e+01 6.50000000e+10 4.60000000e+10 4.60000000e+09 2.E+8 2.65833333e+01 0.00000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
1.22000000e+01 6.50000000e+10 4.50000000e+10 4.50000000e+09 2.E+8 2.70000000e+01 0.00000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
1.25100000e+01 6.50000000e+10 4.50000000e+10 4.50000000e+09 2.E+8 0. 0.00000000e+00 0. 0. 0. 0. 0. 0. 0. 0.

!aero_blade_table blname: wcs_v0_blade scale_chord: 2. ! r/R c/R aero twist[deg] t/c airfol, from r3
0.0 0.04352 20.5 0.8288658 Cylinder1
0.090690 0.04352 20.5 0.8288658 Cylinder1
0.133876 0.0672 20.6 0.5748849 interpolate
0.181380 0.0864 20.1 0.4460525 interpolate
0.233204 0.09408 17.0 0.3809175 interpolate
0.285027 0.08896 12.1 0.3488297 interpolate
0.336849 0.08512 8.47 0.3291967 interpolate
0.388673 0.08128 6.04 0.3095637 NACA63_2XX_INNER
0.440495 0.07744 4.37 0.2935702 interpolate
0.492319 0.0736 3.04 0.2804608 interpolate
0.544142 0.06976 2.05 0.2673515 interpolate
0.595965 0.06592 1.33 0.2410318 interpolate
0.647788 0.06208 0.78 0.2213989 NACA63_221
0.699611 0.05824 0.42 0.2017659 interpolate
0.751434 0.05440 0.22 0.1849139 interpolate
0.803257 0.05056 0.11 0.1725000 NACA63_218
0.846442 0.04736 0.0 0.1592134 interpolate
0.880991 0.04480 0.0 0.1571405 interpolate
0.924177 0.0416 0.0 0.1545493 interpolate
0.976 0.0378 0.0 0.1514400 NACA63_215
1.01 0.0378 0.0 0.1514400 NACA63_215

```

```

aero_blade_table blname: wcs_v0_blade scale_chord: 2. ! r/R c/R aero twist[deg] t/c airfoil, from r3
0.0      0.04      40.      1.          Cylinder1
0.03     0.04      40.      1.          Cylinder1
0.06     0.04      40.      1.          Cylinder1
0.090690 0.1       40.      0.8288658 interpolate
0.133876 0.1       27.      0.5748849 interpolate
0.181380 0.113     20.1     0.4460525 FFA_W3_360_RE_2_3E6
0.233204 0.102     16.      0.3809175 interpolate
0.285027 0.0915    12.5     0.3488297 FFA_W3_360_RE_2_3E6
0.336849 0.082     9.7      0.3291967 interpolate
0.388673 0.074     7.7      0.3095637 interpolate
0.440495 0.067     6.       0.2935702 FFA_W3_301_RE_2_3E6
0.492319 0.059     4.5     0.2804608 interpolate
0.544142 0.053     3.5     0.2673515 interpolate
0.595965 0.048     2.5     0.2410318 FFA_W3_241_RE_2_3E6
0.647788 0.045     0.       0.2213989 NACA63_221
0.699611 0.044     -1.     0.2017659 interpolate
0.751434 0.042     -1.     0.1849139 interpolate
0.803257 0.04     -1.     0.1725000 NACA63_218
0.846442 0.0385    -1.     0.1592134 interpolate
0.880991 0.036     -1.     0.1571405 interpolate
0.924177 0.032     -2.     0.1545493 interpolate
0.976     0.025     -1.     0.1514400 NACA63_215
1.0001    0.025     -1.     0.1514400 NACA63_215

read_airfoil
file NACA63_215.dat      afname NACA63_215 thick .15

read_airfoil
file NACA63_218.dat      afname NACA63_218 thick .18

read_airfoil
file NACA63_221.dat      afname NACA63_221 thick .21

read_airfoil
file Cylinder1.dat      afname Cylinder1 thick 1.

read_airfoil
file FFA_W3_360_RE_2_3E6.DAT afname FFA_W3_360_RE_2_3E6 thick .36

read_airfoil
file FFA_W3_301_RE_2_3E6.DAT afname FFA_W3_301_RE_2_3E6 thick .3

read_airfoil
file FFA_W3_241_RE_2_3E6.DAT afname FFA_W3_241_RE_2_3E6 thick .24

new_body
name:      hub
parent:    rotor_1
reftype:   refnode
xfind:     -6.2 0. 0.
rotorder:  123
rotation:  0. 0. 0.
axes:      normal
tc_kin:    0.

new_body
name:      blade_1_root
parent:    hub
reftype:   refnode
xfind:     0. 0. 0.
rotorder:  123
rotation:  1. 0. 0. ! 0. cone_angle_0
axes:      normal
tc_kin:    0.

new_body
name:      blade_1_pitch
parent:    blade_1_root
reftype:   refnode
xfind:     0. 0. 0.
rotorder:  123
rotation:  0. 0. 0. ! 0. 0. pre_pitch neg towards feather
axes:      normal
tc_kin:    0.

new_elements_beam1 nelelem 1 material s355 connect_mode auto ! scale to hub mass
x1 0. 0. 0.35 dcy11 0.3 tcy11 0.03
x2 0. 0. 0. dcy12 0.3 tcy12 0.03 btype beam

```



```

blade mass tag start ! 875.5 kg for first version

new_elements_beam1 nelem 16 material blade_dummy connect_mode auto
x1 0. 0. 12.5 dcyl1 1. 0.175 tcyl1 0.1 0.01
x2 0. 0. 0.35 dcyl2 3. 0.625 tcyl2 0.3 0.01 btype beam
blname wcs_v0_blade scale_mass 1. frac_stiff 1.e3 !TAN Scale mass is now set in blade_table

blade mass tag end

new_body
name: blade_2_root
parent: hub
reftype: refnode
xfind: 0. 0. 0.
rotorder: 123
rotation: 181. 0. 0. !180.001 0. 0. ! 0. cone_angle 0
axes: normal
tc_kin: 0.

new_body
name: blade_2_pitch
parent: blade_2_root
reftype: refnode
xfind: 0. 0. 0.
rotorder: 123
rotation: 0. 0. 0. ! 0. 0. pre_pitch neg towards feather
axes: normal
tc_kin: 0.

copy_elements
from_body: blade_1_pitch
to_body: blade_2_pitch ! can replace new_element statements above

rotor_wake
name: rotor_1_wcs
shaft_element_body: rotor_1
shaft_element_position: -6.075 0. 0.
rotating_node: 1
induction: all
logfile: rotor_1.txt
nblades: 2
bladenames: blade_1_pitch
blade_2_pitch
pitch_control_name: stall_1
pitch_actuator_bodies: blade_1_pitch
blade_2_pitch
pitch_actuator_positons: 0. 0. .175
0. 0. .175

pitchcontrol
control_type: none !iea_oc3_floating
name: stall_1
scale_omega: 1.
scale_vaxial: 1.
scale_pitchcoll: -1.0
omeganom: 1.2671
ti: 10.
gain: 1.
pitchref: 0.
aux: 1. 2. 3. 4. 5. 6. 7. 8. 9. 10.

!/monitors

set_max_monitors
nsensmax: 300

monitor_element_b1 type wind_fixed find 0. 0. 30. nmonitor 1 node 2
file wind.txt body_name inertial label WindVxi WindVyi WindVzi

```

```

!Vetles monitorer

monitor_element_b1 type forces file front_hor_left_strut_force_x1.txt nmonitor 1
find -5.6250E+00 -6.8750E-01 0.0000E+00 node 2 body_name rotor_1 label front_hor_left_strut_fx_1 front_hor_left_strut_fy_1 front_hor_left_strut_fz_1

monitor_element_b1 type moments file front_hor_left_strut_moment_x1.txt nmonitor 1
find -5.6250E+00 -6.8750E-01 0.0000E+00 node 2 body_name rotor_1 label mx_fhls_1 my_fhls_1 mz_fhls_1

monitor_element_b1 type forces file front_hor_left_strut_force_x2.txt nmonitor 1
find -3.7500E-01 -1.0312E+01 0.0000E+00 node 2 body_name rotor_1 label front_hor_left_strut_fx_2 front_hor_left_strut_fy_2 front_hor_left_strut_fz_2

monitor_element_b1 type moments file front_hor_left_strut_moment_x2.txt nmonitor 1
find -3.7500E-01 -1.0312E+01 0.0000E+00 node 2 body_name rotor_1 label mx_fhls_2 my_fhls_2 mz_fhls_2

monitor_element_b1 type forces file front_hor_right_strut_force_x1.txt nmonitor 1
find -5.6250E+00 6.8750E-01 0.0000E+00 node 2 body_name rotor_1 label front_hor_right_strut_fx_1 front_hor_right_strut_fy_1 front_hor_right_strut_fz_1

monitor_element_b1 type moments file front_hor_right_strut_moment_x1.txt nmonitor 1
find -5.6250E+00 6.8750E-01 0.0000E+00 node 2 body_name rotor_1 label mx_fhrs_1 my_fhrs_1 mz_fhrs_1

monitor_element_b1 type forces file front_hor_right_strut_force_x2.txt nmonitor 1
find -3.7500E-01 1.0312E+01 0.0000E+00 node 2 body_name rotor_1 label front_hor_right_strut_fx_2 front_hor_right_strut_fy_2 front_hor_right_strut_fz_2

monitor_element_b1 type moments file front_hor_right_strut_moment_x2.txt nmonitor 1
find -3.7500E-01 1.0312E+01 0.0000E+00 node 2 body_name rotor_1 label mx_fhrs_2 my_fhrs_2 mz_fhrs_2

monitor_element_b1 type forces file front_vert_down_strut_force_x1.txt nmonitor 1
find -5.6250E+00 0.0000E+00 -7.8125E-01 node 2 body_name rotor_1 label front_vert_down_strut_fx_1 front_vert_down_strut_fy_1 front_vert_down_strut_fz_1

monitor_element_b1 type moments file front_vert_down_strut_moment_x1.txt nmonitor 1
find -5.6250E+00 0.0000E+00 -7.8125E-01 node 2 body_name rotor_1 label mx_fvds_1 my_fvds_1 mz_fvds_1

monitor_element_b1 type forces file front_vert_down_strut_force_x2.txt nmonitor 1
find -3.7500E-01 0.0000E+00 -1.1719E+01 node 2 body_name rotor_1 label front_vert_down_strut_fx_2 front_vert_down_strut_fy_2 front_vert_down_strut_fz_2

monitor_element_b1 type moments file front_vert_down_strut_moment_x2.txt nmonitor 1
find -3.7500E-01 0.0000E+00 -1.1719E+01 node 2 body_name rotor_1 label mx_fvds_2 my_fvds_2 mz_fvds_2

monitor_element_b1 type forces file front_vert_up_strut_force_x1.txt nmonitor 1
find -5.6250E+00 0.0000E+00 7.8125E-01 node 2 body_name rotor_1 label front_vert_up_strut_fx_1 front_vert_up_strut_fy_1 front_vert_up_strut_fz_1

monitor_element_b1 type moments file front_vert_up_strut_moment_x1.txt nmonitor 1
find -5.6250E+00 0.0000E+00 7.8125E-01 node 2 body_name rotor_1 label mx_fvus_1 my_fvus_1 mz_fvus_1

monitor_element_b1 type forces file front_vert_up_strut_force_x2.txt nmonitor 1
find -3.7500E-01 0.0000E+00 1.1719E+01 node 2 body_name rotor_1 label front_vert_up_strut_fx_2 front_vert_up_strut_fy_2 front_vert_up_strut_fz_2

monitor_element_b1 type moments file front_vert_up_strut_moment_x2.txt nmonitor 1
find -3.7500E-01 0.0000E+00 1.1719E+01 node 2 body_name rotor_1 label mx_fvus_2 my_fvus_2 mz_fvus_2

monitor_element_b1 type forces file rear_left_up_strut_force_x1.txt nmonitor 1
find 9.3750E-03 -6.8750E-01 9.3750E-01 node 2 body_name rotor_1 label rear_left_up_strut_fx_1 rear_left_up_strut_fy_1 rear_left_up_strut_fz_1

monitor_element_b1 type moments file rear_left_up_strut_moment_x1.txt nmonitor 1
find 9.3750E-03 -6.8750E-01 9.3750E-01 node 2 body_name rotor_1 label mx_rlus_1 my_rlus_1 mz_rlus_1

monitor_element_b1 type forces file rear_left_up_strut_force_x2.txt nmonitor 1
find 6.2500E-04 -1.0312E+01 6.2500E-02 node 2 body_name rotor_1 label rear_left_up_strut_fx_2 rear_left_up_strut_fy_2 rear_left_up_strut_fz_2

monitor_element_b1 type moments file rear_left_up_strut_moment_x2.txt nmonitor 1
find 6.2500E-04 -1.0312E+01 6.2500E-02 node 2 body_name rotor_1 label mx_rlus_2 my_rlus_2 mz_rlus_2

monitor_element_b1 type forces file rear_left_down_strut_force_x1.txt nmonitor 1
find 9.3750E-03 -6.8750E-01 -9.3750E-01 node 2 body_name rotor_1 label rear_left_down_strut_fx_1 rear_left_down_strut_fy_1 rear_left_down_strut_fz_1

monitor_element_b1 type moments file rear_left_down_strut_moment_x1.txt nmonitor 1
find 9.3750E-03 -6.8750E-01 -9.3750E-01 node 2 body_name rotor_1 label mx_rlds_1 my_rlds_1 mz_rlds_1

monitor_element_b1 type forces file rear_left_down_strut_force_x2.txt nmonitor 1
find 6.2500E-04 -1.0312E+01 -6.2500E-02 node 2 body_name rotor_1 label rear_left_down_strut_fx_2 rear_left_down_strut_fy_2 rear_left_down_strut_fz_2

monitor_element_b1 type moments file rear_left_down_strut_moment_x2.txt nmonitor 1
find 6.2500E-04 -1.0312E+01 -6.2500E-02 node 2 body_name rotor_1 label mx_rlds_2 my_rlds_2 mz_rlds_2

monitor_element_b1 type forces file rear_right_up_strut_force_x1.txt nmonitor 1
find 9.3750E-03 6.8750E-01 9.3750E-01 node 2 body_name rotor_1 label rear_right_up_strut_fx_1 rear_right_up_strut_fy_1 rear_right_up_strut_fz_1

monitor_element_b1 type moments file rear_right_up_strut_moment_x1.txt nmonitor 1
find 9.3750E-03 6.8750E-01 9.3750E-01 node 2 body_name rotor_1 label mx_rrus_1 my_rrus_1 mz_rrus_1

monitor_element_b1 type forces file rear_right_up_strut_force_x2.txt nmonitor 1
find 6.2500E-04 1.0312E+01 6.2500E-02 node 2 body_name rotor_1 label rear_right_up_strut_fx_2 rear_right_up_strut_fy_2 rear_right_up_strut_fz_2

```

```

monitor_element_b1 type moments file rear_right_up_strut_moment_x2.txt nmonitor 1
find 6.2500E-04 1.0312E+01 6.2500E-02 node 2 body_name rotor_1 label mx_rrus_2 my_rrus_2 mz_rrus_2

monitor_element_b1 type forces file rear_right_down_strut_force_x1.txt nmonitor 1
find 9.3750E-03 6.8750E-01 -9.3750E-01 node 2 body_name rotor_1 label rear_right_down_strut_fx_1 rear_right_down_strut_fy_1 rear_right_down_strut_fz_1

monitor_element_b1 type moments file rear_right_down_strut_moment_x1.txt nmonitor 1
find 9.3750E-03 6.8750E-01 -9.3750E-01 node 2 body_name rotor_1 label mx_rrds_1 my_rrds_1 mz_rrds_1

monitor_element_b1 type forces file rear_right_down_strut_force_x2.txt nmonitor 1
find 6.2500E-04 1.0312E+01 -6.2500E-02 node 2 body_name rotor_1 label rear_right_down_strut_fx_2 rear_right_down_strut_fy_2 rear_right_down_strut_fz_2

monitor_element_b1 type moments file rear_right_down_strut_moment_x2.txt nmonitor 1
find 6.2500E-04 1.0312E+01 -6.2500E-02 node 2 body_name rotor_1 label mx_rrds_2 my_rrds_2 mz_rrds_2

monitor_element_b1 type forces file rear_vert_down_strut_force_x1.txt nmonitor 1
find 9.3750E-03 0.0000E+00 -7.8125E-01 node 2 body_name rotor_1 label rear_vert_down_strut_fx_1 rear_vert_down_strut_fy_1 rear_vert_down_strut_fz_1

monitor_element_b1 type moments file rear_vert_down_strut_moment_x1.txt nmonitor 1
find 9.3750E-03 0.0000E+00 -7.8125E-01 node 2 body_name rotor_1 label mx_rvds_1 my_rvds_1 mz_rvds_1

monitor_element_b1 type forces file rear_vert_down_strut_force_x2.txt nmonitor 1
find 6.2500E-04 0.0000E+00 -1.1719E+01 node 2 body_name rotor_1 label rear_vert_down_strut_fx_2 rear_vert_down_strut_fy_2 rear_vert_down_strut_fz_2

monitor_element_b1 type moments file rear_vert_down_strut_moment_x2.txt nmonitor 1
find 6.2500E-04 0.0000E+00 -1.1719E+01 node 2 body_name rotor_1 label mx_rvds_2 my_rvds_2 mz_rvds_2

monitor_element_b1 type forces file rear_vert_up_strut_force_x1.txt nmonitor 1
find 9.3750E-03 0.0000E+00 7.8125E-01 node 2 body_name rotor_1 label rear_vert_up_strut_fx_1 rear_vert_up_strut_fy_1 rear_vert_up_strut_fz_1

monitor_element_b1 type moments file rear_vert_up_strut_moment_x1.txt nmonitor 1
find 9.3750E-03 0.0000E+00 7.8125E-01 node 2 body_name rotor_1 label mx_rvus_1 my_rvus_1 mz_rvus_1

monitor_element_b1 type forces file rear_vert_up_strut_force_x2.txt nmonitor 1
find 6.2500E-04 0.0000E+00 1.1719E+01 node 2 body_name rotor_1 label rear_vert_up_strut_fx_2 rear_vert_up_strut_fy_2 rear_vert_up_strut_fz_2

monitor_element_b1 type moments file rear_vert_up_strut_moment_x2.txt nmonitor 1
find 6.2500E-04 0.0000E+00 1.1719E+01 node 2 body_name rotor_1 label mx_rvus_2 my_rvus_2 mz_rvus_2

monitor_element_b1 type forces file rotor_shaft_force.txt nmonitor 1
find -6.1500E+00 0.0000E+00 0.0000E+01 node 1 body_name rotor_1 label shaft_fx shaft_fy shaft_fz

monitor_element_b1 type moments file rotor_shaft_moment.txt nmonitor 1
find -6.1500E+00 0.0000E+00 0.0000E+01 node 1 body_name rotor_1 label shaft_mx shaft_my shaft_mz

!TAN force sensor nonrotating system, note element has been added

monitor_element_b1 type forces file rotor_force.txt nmonitor 1
find -6.0250E+00 0.0000E+00 0.0000E+00 node 1 body_name rotor_1 label rotor_fx rotor_fy rotor_fz

monitor_element_b1 type moments file rotor_moment.txt nmonitor 1
find -6.0250E+00 0.0000E+00 0.0000E+00 node 1 body_name rotor_1 label rotor_mx rotor_my rotor_mz

monitor_element_b1 type forces file blade_1_root_force.txt nmonitor 1
find 0. 0. 0.175 node 2 body_name blade_1_root label fx_bl_1_root fy_bl_1_root fz_bl_1_root

monitor_element_b1 type moments file blade_1_root_moment.txt nmonitor 1
find 0. 0. 0.175 node 2 body_name blade_1_root label mx_bl_1_root my_bl_1_root mz_bl_1_root

monitor_element_b1 type forces file blade_2_root_force.txt nmonitor 1
find 0. 0. 0.175 node 2 body_name blade_2_root label fx_bl_2_root fy_bl_2_root fz_bl_2_root

monitor_element_b1 type moments file blade_2_root_moment.txt nmonitor 1
find 0. 0. 0.175 node 2 body_name blade_2_root label mx_bl_2_root my_bl_2_root mz_bl_2_root

!midpoints for front struts to calculate buckling

monitor_element_b1 type forces file front_hor_left_strut_force_x3.txt nmonitor 1
find -3.0000E+00 -5.5000E+00 0.0000E+00 node 2 body_name rotor_1 label front_hor_left_strut_fx_3 front_hor_left_strut_fy_3 front_hor_left_strut_fz_3

monitor_element_b1 type moments file front_hor_left_strut_moment_x3.txt nmonitor 1
find -3.0000E+00 -5.5000E+00 0.0000E+00 node 2 body_name rotor_1 label mx_fhls_3 my_fhls_3 mz_fhls_3

monitor_element_b1 type forces file front_hor_right_strut_force_x3.txt nmonitor 1
find -3.0000E+00 5.5000E+00 0.0000E+00 node 2 body_name rotor_1 label front_hor_right_strut_fx_3 front_hor_right_strut_fy_3 front_hor_right_strut_fz_3

monitor_element_b1 type moments file front_hor_right_strut_moment_x3.txt nmonitor 1
find -3.0000E+00 5.5000E+00 0.0000E+00 node 2 body_name rotor_1 label mx_fhrs_3 my_fhrs_3 mz_fhrs_3

monitor_element_b1 type forces file front_vert_down_strut_force_x3.txt nmonitor 1
find -3.0000E+00 0.0000E+00 -6.2500E+00 node 2 body_name rotor_1 label front_vert_down_strut_fx_3 front_vert_down_strut_fy_3 front_vert_down_strut_fz_3

```

```

monitor_element_b1 type moments file front_vert_down_strut_moment_x3.txt nmonitor 1
find -3.0000E+00 0.0000E+00 -6.2500E+00 node 2 body_name rotor_1 label mx_fvds_3 my_fvds_3 mz_fvds_3

monitor_element_b1 type forces file front_vert_up_strut_force_x3.txt nmonitor 1
find -3.0000E+00 0.0000E+00 6.2500E+00 node 2 body_name rotor_1 label front_vert_up_strut_fx_3 front_vert_up_strut_fy_3 front_vert_up_strut_fz_3

monitor_element_b1 type moments file front_vert_up_strut_moment_x3.txt nmonitor 1
find -3.0000E+00 0.0000E+00 6.2500E+00 node 2 body_name rotor_1 label mx_fvds_3 my_fvds_3 mz_fvds_3

monitor_element_b1 find 0. 0. -20. nmonitor 1 node 1 type gather
file sensors.txt label wcs_gather toffset 10.
sensors:
nfact: 1 source: WindVxi scale: 1. unit: [m/s] ! header as source label
nfact: 1 source: WindVyi scale: 1. unit: [m/s] !
nfact: 1 source: WindVzi scale: 1. unit: [m/s] !
nfact: 1 source: Azimuth_1 scale: 57.295779513 unit: [deg] !
nfact: 1 source: AeroThr_1 scale: 1.e-3 unit: [kN] !
nfact: 1 source: AeroTq_1 scale: 1.e-3 unit: [kNm] !
nfact: 1 source: RotSpeed_1 scale: 9.549296585513721 unit: [rpm] header: RotSpeed ! 30./pi
nfact: 1 source: RotSpeed_1 scale: 1 unit: [rad/s] header: Angular_vel ! 30./pi
nfact: 1 source: Vaxial_1 scale: 1. unit: [m/s] !
nfact: 1 source: shaft_fx scale: 1.e-3 unit: [kN] !
nfact: 1 source: shaft_fy scale: 1.e-3 unit: [kN] !
nfact: 1 source: shaft_fz scale: 1.e-3 unit: [kN] !
nfact: 1 source: shaft_mx scale: 1.e-3 unit: [kNm] !
nfact: 1 source: shaft_my scale: 1.e-3 unit: [kNm] !
nfact: 1 source: shaft_mz scale: 1.e-3 unit: [kNm] !
nfact: 1 source: fx_bl_1_root scale: 1.e-3 unit: [kN]
nfact: 1 source: fy_bl_1_root scale: 1.e-3 unit: [kN]
nfact: 1 source: fz_bl_1_root scale: 1.e-3 unit: [kN]
nfact: 1 source: mx_bl_1_root scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_bl_1_root scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_bl_1_root scale: 1.e-3 unit: [kNm]
nfact: 1 source: front_hor_left_strut_fx_1 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_fhls_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_fhls_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_fhls_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: front_hor_left_strut_fx_2 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_fhls_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_fhls_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_fhls_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: front_hor_right_strut_fx_1 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_fhrs_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_fhrs_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_fhrs_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: front_hor_right_strut_fx_2 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_fhrs_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_fhrs_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_fhrs_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: front_vert_down_strut_fx_1 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_fvds_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_fvds_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_fvds_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: front_vert_down_strut_fx_2 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_fvds_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_fvds_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_fvds_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: front_vert_up_strut_fx_1 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_fvus_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_fvus_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_fvus_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: front_vert_up_strut_fx_2 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_fvus_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_fvus_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_fvus_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: rear_left_down_strut_fx_1 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_rlids_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_rlids_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_rlids_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: rear_left_down_strut_fx_2 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_rlids_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_rlids_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_rlids_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: rear_left_up_strut_fx_1 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_rlus_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_rlus_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_rlus_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: rear_left_up_strut_fx_2 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_rlus_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_rlus_2 scale: 1.e-3 unit: [kNm]

```



```

nfact: 1 source: my_fhrs_3      scale: 1.e-3      unit: [kNm]
nfact: 1 source: mz_fhrs_3      scale: 1.e-3      unit: [kNm]
nfact: 1 source: front_vert_down_strut_fx_3  scale: 1.e-3      unit: [kN]  !
nfact: 1 source: front_vert_down_strut_fy_3  scale: 1.e-3      unit: [kN]  !
nfact: 1 source: front_vert_down_strut_fz_3  scale: 1.e-3      unit: [kN]  !
nfact: 1 source: mx_fvds_3      scale: 1.e-3      unit: [kNm]
nfact: 1 source: my_fvds_3      scale: 1.e-3      unit: [kNm]
nfact: 1 source: mz_fvds_3      scale: 1.e-3      unit: [kNm]
nfact: 1 source: front_vert_up_strut_fx_3    scale: 1.e-3      unit: [kN]  !
nfact: 1 source: front_vert_up_strut_fy_3    scale: 1.e-3      unit: [kN]  !
nfact: 1 source: front_vert_up_strut_fz_3    scale: 1.e-3      unit: [kN]  !
nfact: 1 source: mx_fvus_3      scale: 1.e-3      unit: [kNm]
nfact: 1 source: my_fvus_3      scale: 1.e-3      unit: [kNm]
nfact: 1 source: mz_fvus_3      scale: 1.e-3      unit: [kNm]
nfact: 1 source: fx_bl_2_root  scale: 1.e-3      unit: [kN]
nfact: 1 source: fy_bl_2_root  scale: 1.e-3      unit: [kN]
nfact: 1 source: fz_bl_2_root  scale: 1.e-3      unit: [kN]
nfact: 1 source: mx_bl_2_root  scale: 1.e-3      unit: [kNm]
nfact: 1 source: my_bl_2_root  scale: 1.e-3      unit: [kNm]
nfact: 1 source: mz_bl_2_root  scale: 1.e-3      unit: [kNm]
nfact: 1 source: rotor_fx      scale: 1.e-3      unit: [kN]  !
nfact: 1 source: rotor_fy      scale: 1.e-3      unit: [kN]  !
nfact: 1 source: rotor_fz      scale: 1.e-3      unit: [kN]  !
nfact: 1 source: rotor_mx      scale: 1.e-3      unit: [kN]  !
nfact: 1 source: rotor_my      scale: 1.e-3      unit: [kN]  !
nfact: 1 source: rotor_mz      scale: 1.e-3      unit: [kN]  !
nfact: 1 source: dummy_end      scale: 1.e-3      unit: [kN]  !

! old junk
nfact: 1 source: AeroTq_1      scale: 1.e-3      unit: [kNm]  !
nfact: 1 source: GenTq_S_1     scale: 1.e-3      unit: [kNm]  !
nfact: 1 source: GenTq_C_1     scale: 1.e-3      unit: [kNm]  !
nfact: 1 source: BldPitch1_1   scale: -57.295779513082323  unit: [deg]  header: BldPitch1 ! from control system
nfact: 1 source: BldPitch1_1   scale: 57.295779513      unit: [deg]  !
nfact: 1 source: GenPwr_1      scale: 1.e-3      unit: [kW]  !

! more old junk
nfact: 1 source: front_hor_left_strut_fx     scale: 1.e-3      unit: [kN]  !
nfact: 1 source: front_hor_right_strut_fx    scale: 1.e-3      unit: [kN]  !
nfact: 1 source: front_vert_low_strut_fx     scale: 1.e-3      unit: [kN]  !
nfact: 1 source: front_vert_up_strut_fx      scale: 1.e-3      unit: [kN]  !
nfact: 1 source: front_vert_up_strut_stress  scale: 1.e-6      unit: [MPa]  !
nfact: 1 source: shaft_fx                    scale: 1.e-3      unit: [kN]  !
nfact: 1 source: shaft_fy                    scale: 1.e-3      unit: [kN]  !
nfact: 1 source: shaft_fz                    scale: 1.e-3      unit: [kN]  !
nfact: 1 source: dummy_end                    scale: 1.e-3      unit: [kN]  !

!
! Boundary conditions
!

apply_nodal_bc find 0. 0. 30.  apply_displ idofs 1 idofe 6 displ 0. 0. 0. 0. 0. 0. 0. ! Hold dummy element

apply_nodal_bc find 0. -11. 17.5  rotor_1 apply_displ idofs 1 idofe 6 displ 0. 0. 0. 0. 0. 0. 0. ! lower left, in inertial system

apply_nodal_bc find 0. 11. 17.5  apply_displ idofs 1 idofe 6 displ 0. 0. 0. 0. 0. 0. 0. ! lower right

apply_nodal_bc find 0. -11. 42.5  apply_displ idofs 1 idofe 6 displ 0. 0. 0. 0. 0. 0. 0. ! upper left

apply_nodal_bc find 0. 11. 42.5  apply_displ idofs 1 idofe 6 displ 0. 0. 0. 0. 0. 0. 0. ! upper right

!apply_nodal_bc find 0. 11. 17.5  apply_displ idofs 1 idofe 1 displ 0. ! lower right
!apply_nodal_bc find 0. 11. 17.5  apply_displ idofs 3 idofe 3 displ 0. ! lower right
!apply_nodal_bc find 0. -11. 42.5  apply_displ idofs 1 idofe 1 displ 0. ! upper left
!apply_nodal_bc find 0. 11. 42.5  apply_displ idofs 1 idofe 1 displ 0. ! upper right

cg_cb
  blade_1_pitch

wind_transient
  0. 15. 0.

! C:\Users\toran\OneDrive - Institutt for Energiteknikk\project2\wcs_2020\turbsim\15ms_ti10

turbulence_box

```

```
turbulence:          file
turbulence_info:    turbsim\15ms_ti10\t15ms_ti10.bts
turb_scale: none ! spatial_scaling !
turb_tstart:        0.
turb_intens:        .1
reference_node:     0. 0. 0. ! box moves slowly horizontally with SWL node
reference_height:   30. ! hub height, center of box is placed here

damping type rayleigh alpha 0. beta .002

#damping type ratio2rayl ratio_1 .01 ratio_2 .01 omega_1 0.1 omega_2 1.

newmark
dt:          0.01
nassemble:   1
nnewton:     20
nsubmin:     2
rwilson:     .9
relax:       0.
resid_newton: 1e-8
nmonitor:    100

step nstep 370000 method step9

blade_status
ir: 1 ! rotor #
ib: 1 ! blade #
blade_file: rotor_1_blade_1_status.txt

blade_status
ir: 1 ! rotor #
ib: 2 ! blade #
blade_file: rotor_1_blade_2_status.txt

monitor_element_b1 type tecplot nmonitor 10
file turb1.dat

step nstep 1000 method step9

! Eigen analysis
! Aim minimum 5.5Hz for nacelle system

!eigen_analysis amplitude 5. nfreq 20 filename plotall

end
```

C.0.2 4-bladed rotor

data.txt

```

!
! 3Dfloat input file.
!
! From C:\Users\toran\OneDrive - Institutt for Energiteknikk\project2\oc4_semisub_2013\revisit_2020\load_cases\3.2_flex\r5
!
! Single rotor/rna/attachments for WCS

r2: resolved blades
r3: Danwin180-like blades
r4: tuned twist and chord, as C:\Users\toran\OneDrive - Institutt for Energiteknikk\project2\wcs_2020\rigid_rotor\r2
r5: added generator, test run 15 - 20m/s
r6: trimmed back root chord
r7: const lambda to check cp and ct
r8: new generator, more sensors, variable speed run, effects of gravity and shear
r9: mores sensors

vetle_check
r1: remove bugs, more sensors
r2: with turbulence

use_sparse_system_matrices

set_environment
rho_water: 1025.
rho_air: 1.225
nu_water: 1.e-5
nu_air: 1.e-6
gravity: on
buoyancy: off on
hydro_force: none morison
wind_force: drag
wind: mean_profile
wind_ref_height: 30.
wind_speed: 0. ! Turb box has mean Ux of -0.266 m/s !
wind_direction: 0.
wind_exponent: 0.14
waves: extrapolatedairy !irregular_waves
norder: -1
wave_amplitude: 1.e-6
wave_direction: 0.
spectral_peak: 19.2
depth: 200.
surface_option: 2ndorder
current_speed: 0.
current_direction: 0.
current_exponent: 0.
wave_ramps: 0.
wave_rampe: 200.
tshift_waves: 0.
tshift_wind: 0.
kinematics_option: updated
tc_kin: 0.
dtwkin: -1.
scale_wind: 1.
roughness_length -99.
pre_computed_waves_file: off

!jonswap_wavelets_constant
hs: 6.
tp: 10.
tstart: -1.
tcut: -1.
gamma: 2.87
tperiodic: 3600.
depth: 200.0
spread -99.
file: wave_table_hs6_tp10_gam2.87.txt

!wavelets
scale_amplitude: 1.
file: wave_table_hs6_tp10_gam2.87.txt

!test_irregairy
dt: .1
nstep: 3600

```



```

x: 0. 0. 0.
filename: waveheight_wave_table_hs6_tp10_gam2.87.txt

wave_forces
  cd_morison: 0.7
  cm_morison: 1.63

!
!!! Defining materials needed to construct model:
!

define_material name s355 rho 7850.0 e 210.e9 g 81.e9
define_material name: blade_dummy rho: 0.0001 e: 210.e9 g: 80.8e9
define_material name generator_material rho 0.0001 e 210.e9 g 80.8e9

define_material name semisub rho 7800.0 e 2.1e+11 g 81000000000.0
define_material name HPmaterial rho 3900.0 e 2.1e+13 g 8.1e+12
define_material name connect_mat rho 1e-08 e 2.1e+15 g 8.1e+14
define_material name connect_mat0 rho 1e-08 e 2.1e+13 g 8.1e+12
define_material name masslessStiff1 rho 0.0001 e 2.1e+14 g 8.08e+13
define_material name wirewb_oc3 rho 11189.705669 e 60399180000.0 g 22300000000.0
define_material name chain0 rho 7904.5462588 e 56000000000.0 g 22300000000.0
define_material name chain1 rho 12743.9021729 e 56000000000.0 g 22300000000.0
define_material name chain2 rho 23196.1475439 e 56000000000.0 g 22300000000.0
define_material name chain3 rho 7875.98230769 e 34609101925.5 g 13781897867.8
define_material name OC4semisub rho 7850.0 e 2.1e+13 g 7.89473684211e+12
define_material name wire_OC4 rho 24596.5447981 e 1.63528506042e+11 g 65119666311.8
define_material name s35grout rho 9153.29723 e 210.e9 g 80.8e9
define_material name s35grout2 rho 3425.024110 e 210.e9 g 80.8e9
define_material name massless1 rho 0.0001 e 210.e9 g 80.8e9
define_material name massless2 rho 0.0001 e 210.e9 g 80.8e9 ! pitch actuator elements
define_material name trans_piece rho 1725.2536669 e 210.e9 g 80.8e9
define_material name mainshaftStiff rho 0.0001 e 210.e12 g 1056.10143e9
define_material name mainshaft rho 0.0001 e 210.e9 g 1056.10143e6
define_material name mainshaft2 rho 0.0001 e 210.e9 g 503745286.6525068 ! tuned to new shaft element
define_material name mainshaft_stiff rho 0.0001 e 210.e9 g 80.8e9
define_material name s35j2g3stf rho 7850. e 210.e9 g 81.e9
define_material name steeltowerMassless rho 0.00001 e 2.1e+11 g 8.08e+11
! g tuned in mainshaft to give correct drivetrain torsional stiffness

!
! Geometry definition
!

! Dummy element to provide reference node to body

new_elements_beam1 nelem 1 material s355 connect_mode no_connect
x1 -1. 0. 30. dcy11 1. tcy11 .1
x2 0. 0. 30. dcy12 1. tcy12 .1 btype beamwb cdn 0.

new_body
  name: rotor_1
  parent: inertial
  reftype: refnode
  xfind: 0. 0. 30.
  rotorder: 123
  rotation: 0. 0. 0.
  axes: normal
  tc_kin: 0. # geometry def from here to next body def will be relative to xfind pos

! / frame with cc h = 25m, b = 22m

new_elements_beam1 nelem 4 material s355 connect_mode no_connect
x1 0. -11. -12.5 dcy11 1.5 tcy11 0.016 pre_strain_long 0.0
x2 0. 11. -12.5 dcy12 1.5 tcy12 0.016 btype beam
frac_stiff 1.0 cdn 0.

new_elements_beam1 nelem 4 material s355 connect_mode no_connect
x1 0. -11. 12.5 dcy11 1.5 tcy11 0.016 pre_strain_long 0.0
x2 0. 11. 12.5 dcy12 1.5 tcy12 0.016 btype beam
frac_stiff 1.0 cdn 0.

new_elements_beam1 nelem 4 material s355 connect_mode connect_both
x1 0. -11. -12.5 dcy11 2. tcy11 0.02 pre_strain_long 0.0
x2 0. -11. 12.5 dcy12 2. tcy12 0.02 btype beam
frac_stiff 1.0 cdn 0.

new_elements_beam1 nelem 4 material s355 connect_mode connect_both

```

```

x1 0. 11. -12.5 dcy11 2. tcy11 0.02 pre_strain_long 0.0
x2 0. 11. 12.5 dcy12 2. tcy12 0.02 btype beam
frac_stiff 1.0 cdn 0.

!// nacelle

new_elements_beam1 nelem 1 material s355 connect_mode no_connect
x1 -6. 0. 0. dcy11 2. tcy11 0.08 pre_strain_long 0.0
x2 0.01 0. 0. dcy12 2. tcy12 0.08 btype beam
frac_stiff 1.0 cdn 0.

!TAN force sensor, nonrotating system
new_elements_beam1 nelem 1 material s355 connect_mode connect_2
x1 -6.05 0. 0. dcy11 1. tcy11 0.1 pre_strain_long 0.0
x2 -6. 0. 0. dcy12 1. tcy12 0.1 btype beam
frac_stiff 1.0 cdn 0.

new_elements_beam1 nelem 1 material generator_material connect_mode connect_2
x1 -6.1 0. 0. dcy11 1. tcy11 0.1 pre_strain_long 0.0
x2 -6.05 0. 0. dcy12 1. tcy12 0.1 btype beam
frac_stiff 1.0 cdn 0. generator: var_rpm !const_rpm !

!// force sensor, rotor
new_elements_beam1 nelem 1 material s355 connect_mode connect_2
x1 -6.2 0. 0. dcy11 1. tcy11 0.03 pre_strain_long 0.0
x2 -6.1 0. 0. dcy12 1. tcy12 0.03 btype beam
frac_stiff 1.0 cdn 0.

! this us used only when forcing constant rpm on rotor
! ! [rad/s] [Nm] ! mgen here is mechanical. El power = omega!generator name: const_rpm gen_eta: 1. npoints_ave: 5 max_trq_rate: 2000.e3 ! Spin up in 100s
scale_omega: 1. scale_mgen: 1.
omega: mgen:
0. -48.e7
10. -48.e7
omega_eta: eta:
0. 1.
10. 1.

! generator for variable speed at tsr 6 cp 0.49680306]
! ! [rad/s] [Nm] ! mgen here is mechanical. El power = omegagenerator name: var_rpm gen_eta: 1. npoints_ave: 2 max_trq_rate: 20000.e3
scale_omega: 1. scale_mgen: 1.
omega: mgen:
0. 0.
2.4 0. 7779.631483486891 ! Zero moment also at 5m/s for easier starting. So valid only above 6m/s
2.88 11202.669336221124
3.36 15248.077707634307
3.84 19915.856597726444
4.32 25206.00600649753
4.8 31118.525933947563
5.28 37653.41638007656
5.76 44810.6773448845
6.24 52590.30882837139
6.72 60992.31083053723
7.2 70016.68335138202
7.68 79663.42639090578
8.16 89932.53994910847
8.64 100824.02402599012
9.12 112337.87862155074
9.6 124474.10373579025
10.08 137232.69936870877
10.56 150613.66552030624
11.04 164617.00219058266
11.52 179242.709379538
12.0 194490.78708717227
12.48 210361.23531348555
12.96 226854.05405847775
13.44 243969.24332214892
13.92 261706.80310449903
14.4 280066.73340552807
omega_eta: eta:
0. 1.
10. 1.

! this is used only when forcing constant rpm on rotor
!material_specific_damping material generator_material type rayleigh alpha 0. beta 0.
damping_matrix: add_matrix_noscaling ! terminate with blank line
4 4 10.e7

```

```

10 10 10.e7
4 10 -10.e7
10 4 -10.e7 ! Should give 48/10 = 4.8 rad/s

! / torque arms to torque struts

new_elements_beam1 nelem 1 material massless connect_mode connect_1
x1 0.01 0. 0. dcy11 2. tcy11 0.08 pre_strain_long 0.0
x2 0.01 0. 1. dcy12 2. tcy12 0.08 btype beamwb
frac_stiff 1.0 cdn 1.

new_elements_beam1 nelem 1 material massless connect_mode connect_1
x1 0.01 0. 0. dcy11 2. tcy11 0.08 pre_strain_long 0.0
x2 0.01 0. -1. dcy12 2. tcy12 0.08 btype beamwb
frac_stiff 1.0 cdn 1.

! / struts nacelle to frame

mass tag struts start

! front horizontal
! front_hor_left
new_elements_beam1 nelem 9 material s355 connect_mode connect_both
x1 -6. 0. 0. dcy11 .2 tcy11 0.008 pre_strain_long 0.0
x2 0. -11. 0. dcy12 .2 tcy12 0.008 btype beam
frac_stiff 1.0 cdn 1. ! member midpoint in body system: -3.0000E+00 -5.5000E+00 0.0000E+00

! front_hor_right
new_elements_beam1 nelem 9 material s355 connect_mode connect_both
x1 -6. 0. 0. dcy11 .2 tcy11 0.008 pre_strain_long 0.0
x2 0. 11. 0. dcy12 .2 tcy12 0.008 btype beam
frac_stiff 1.0 cdn 1. ! member midpoint in body system: -3.0000E+00 5.5000E+00 0.0000E+00

! front vertical
! front_vert_down
new_elements_beam1 nelem 9 material s355 connect_mode connect_both
x1 -6. 0. 0. dcy11 .2 tcy11 0.008 pre_strain_long 0.0
x2 0. 0. -12.5 dcy12 .2 tcy12 0.008 btype beam
frac_stiff 1.0 cdn 1. ! member midpoint in body system: -3.0000E+00 0.0000E+00 -6.2500E+00

!front_vert_up
new_elements_beam1 nelem 9 material s355 connect_mode connect_both
x1 -6. 0. 0. dcy11 .2 tcy11 0.008 pre_strain_long 0.0
x2 0. 0. 12.5 dcy12 .2 tcy12 0.008 btype beam
frac_stiff 1.0 cdn 1. ! member midpoint in body system: -3.0000E+00 0.0000E+00 6.2500E+00

!rear horizontal avoid connection to dummy node in 0,0,0
! rear_left_up
new_elements_beam1 nelem 8 material s355 connect_mode connect_both
x1 0.01 0. 1. dcy11 .2 tcy11 0.008 pre_strain_long 0.0
x2 0. -11. 0. dcy12 .2 tcy12 0.008 btype beam
frac_stiff 1.0 cdn 1. ! member midpoint in body system: 5.0000E-03 -5.5000E+00 5.0000E-01

! rear_left_down
new_elements_beam1 nelem 8 material s355 connect_mode connect_both
x1 0.01 0. -1. dcy11 .2 tcy11 0.008 pre_strain_long 0.0
x2 0. -11. 0. dcy12 .2 tcy12 0.008 btype beam
frac_stiff 1.0 cdn 1. ! member midpoint in body system: 5.0000E-03 -5.5000E+00 -5.0000E-01

! rear_right_up
new_elements_beam1 nelem 8 material s355 connect_mode connect_both
x1 0.01 0. 1. dcy11 .2 tcy11 0.008 pre_strain_long 0.0
x2 0. 11. 0. dcy12 .2 tcy12 0.008 btype beam
frac_stiff 1.0 cdn 1. ! member midpoint in body system: 5.0000E-03 5.5000E+00 5.0000E-01

```

```

! rear_right_down
new_elements_beam1 nelelem 8 material s355 connect_mode connect_both
x1 0.01 0. -1. dcy11 .2 tcyl1 0.008 pre_strain_long 0.0
x2 0. 0. 11. 0. dcy12 .2 tcyl2 0.008 btype beam
frac_stiff 1.0 cdn 1. ! member midpoint in body system: 5.0000E-03 5.5000E+00 -5.0000E-01

! rear vertical
! rear_vert_down
new_elements_beam1 nelelem 8 material s355 connect_mode connect_both
x1 0.01 0. 0. dcy11 .2 tcyl1 0.008 pre_strain_long 0.0
x2 0. 0. -12.5 dcy12 .2 tcyl2 0.008 btype beam
frac_stiff 1.0 cdn 1. ! member midpoint in body system: 5.0000E-03 0.0000E+00 -6.2500E+00

! rear_vert_up
new_elements_beam1 nelelem 8 material s355 connect_mode connect_both
x1 0.01 0. 0. dcy11 .2 tcyl1 0.008 pre_strain_long 0.0
x2 0. 0. 12.5 dcy12 .2 tcyl2 0.008 btype beam
frac_stiff 1.0 cdn 1. ! member midpoint in body system: 5.0000E-03 0.0000E+00 6.2500E+00

```

```
mass tag struts end
```

```

! -----
! WCS 12.5m blade v0: STRUCTURAL AND AERODYNAMIC PROPERTIES
! -----
!
! blade element structural coordinate system:
!
! y out TE along flap principal axis
! z out suction side along lag principal axis
! x = y x z ! NB CAN THEREFORE BE TOWARDS TIP OR HUB
!
! structural twist is positive around x
! mass center and shear center offsets are given in blade element structural
! coordinate system
!
!
! r[m] EI_f[Nm] dy dz dy dz dy
blade_table_old blname: wcs_v0_blade
0. 6.00000000e+07 6.00000000e+07 6.00000000e+10 2.E+10 3.e2 2.05000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
0.375 6.00000000e+07 6.00000000e+07 6.00000000e+10 2.E+10 3.e2 2.05000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
0.75 6.00000000e+07 6.00000000e+07 6.00000000e+10 2.E+10 3.e2 2.05000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
1.13362832e+00 6.00000000e+07 6.00000000e+07 6.00000000e+10 2.E+10 2.28333333e+02 2.05000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
1.67345133e+00 6.00000000e+07 6.00000000e+07 6.00000000e+10 2.E+10 1.28666667e+02 2.06000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
2.26725664e+00 5.00000000e+07 6.50000000e+07 6.50000000e+10 2.E+10 1.07666667e+02 2.01000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
2.91504425e+00 3.50000000e+07 6.80000000e+07 6.80000000e+10 2.E+10 8.45000000e+01 1.70000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
3.56283186e+00 2.50000000e+07 5.00000000e+07 5.00000000e+10 2.E+10 7.11666667e+01 1.21000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
4.21061947e+00 1.71000000e+07 3.70000000e+07 3.70000000e+10 2.E+10 6.20000000e+01 8.47000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
4.85840708e+00 1.10000000e+07 3.10000000e+07 3.10000000e+10 2.E+10 5.80000000e+01 6.04000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
5.50619469e+00 7.00000000e+06 2.50000000e+07 2.50000000e+10 2.E+10 5.33333333e+01 4.37000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
6.15398230e+00 4.50000000e+06 2.20000000e+07 2.20000000e+10 2.E+10 4.80000000e+01 3.04000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
6.80176991e+00 3.40000000e+06 1.80000000e+07 1.80000000e+10 2.E+10 4.45000000e+01 2.05000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
7.44955752e+00 2.40000000e+06 1.58000000e+07 1.58000000e+10 2.E+10 4.15000000e+01 1.33000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
8.09734513e+00 1.70000000e+06 1.30000000e+07 1.30000000e+10 2.E+9 3.95000000e+01 7.80000000e-01 0. 0. 0. 0. 0. 0. 0. 0.
8.74513274e+00 1.10000000e+06 1.10000000e+07 1.10000000e+10 2.E+9 3.70000000e+01 4.20000000e-01 0. 0. 0. 0. 0. 0. 0. 0.
9.39292035e+00 8.00000000e+05 9.11000000e+06 9.11000000e+09 2.E+9 3.50000000e+01 2.20000000e-01 0. 0. 0. 0. 0. 0. 0. 0.
1.00407080e+01 6.50000000e+05 6.50000000e+06 6.50000000e+09 2.E+9 3.02500000e+01 1.10000000e-01 0. 0. 0. 0. 0. 0. 0. 0.
1.05805310e+01 6.50000000e+05 5.30000000e+06 5.30000000e+09 2.E+9 2.32500000e+01 0.00000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
1.10123894e+01 6.50000000e+05 4.90000000e+06 4.90000000e+09 2.E+9 2.25000000e+01 0.00000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
1.15522124e+01 6.50000000e+05 4.60000000e+06 4.60000000e+09 2.E+8 2.65833333e+01 0.00000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
1.22000000e+01 6.50000000e+05 4.50000000e+06 4.50000000e+09 2.E+8 2.70000000e+01 0.00000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
1.25100000e+01 6.50000000e+05 4.50000000e+06 4.50000000e+09 2.E+8 0. 0.00000000e+00 0. 0. 0. 0. 0. 0. 0. 0.

blade_table blname: wcs_v0_blade
0. 6.00000000e+10 6.00000000e+10 6.00000000e+10 2.E+10 3.e2 2.05000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
0.375 6.00000000e+10 6.00000000e+10 6.00000000e+10 2.E+10 3.e2 2.05000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
0.75 6.00000000e+10 6.00000000e+10 6.00000000e+10 2.E+10 3.e2 2.05000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
1.13362832e+00 6.00000000e+10 6.00000000e+10 6.00000000e+10 2.E+10 2.28333333e+02 2.05000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
1.67345133e+00 6.00000000e+10 6.00000000e+10 6.00000000e+10 2.E+10 1.28666667e+02 2.06000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
2.26725664e+00 5.00000000e+10 6.50000000e+10 6.50000000e+10 2.E+10 1.07666667e+02 2.01000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
2.91504425e+00 3.50000000e+10 6.80000000e+10 6.80000000e+10 2.E+10 8.45000000e+01 1.70000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
3.56283186e+00 2.50000000e+10 5.00000000e+10 5.00000000e+10 2.E+10 7.11666667e+01 1.21000000e+01 0. 0. 0. 0. 0. 0. 0. 0.
4.21061947e+00 1.71000000e+10 3.70000000e+10 3.70000000e+10 2.E+10 6.20000000e+01 8.47000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
4.85840708e+00 1.10000000e+10 3.10000000e+10 3.10000000e+10 2.E+10 5.80000000e+01 6.04000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
5.50619469e+00 7.00000000e+10 2.50000000e+10 2.50000000e+10 2.E+10 5.33333333e+01 4.37000000e+00 0. 0. 0. 0. 0. 0. 0. 0.
6.15398230e+00 4.50000000e+10 2.20000000e+10 2.20000000e+10 2.E+10 4.80000000e+01 3.04000000e+00 0. 0. 0. 0. 0. 0. 0. 0.

```

```

6.80176991e+00      3.40000000e+10      1.80000000e+10      1.80000000e+10      2.E+10      4.45000000e+01      2.05000000e+00      0. 0. 0.
7.44955752e+00      2.40000000e+10      1.58000000e+10      1.58000000e+10      2.E+10      4.15000000e+01      1.33000000e+00      0. 0. 0.
8.09734513e+00      1.70000000e+10      1.30000000e+10      1.30000000e+10      2.E+9      3.95000000e+01      7.80000000e-01      0. 0. 0.
8.74513274e+00      1.10000000e+10      1.10000000e+10      1.10000000e+10      2.E+9      3.70000000e+01      4.20000000e-01      0. 0. 0.
9.39292035e+00      8.00000000e+10      9.11000000e+10      9.11000000e+09      2.E+9      3.50000000e+01      2.20000000e-01      0. 0. 0.
1.00407080e+01      6.50000000e+10      6.50000000e+10      6.50000000e+09      2.E+9      3.02500000e+01      1.10000000e-01      0. 0. 0.
1.05805310e+01      6.50000000e+10      5.30000000e+10      5.30000000e+09      2.E+9      2.32500000e+01      0.00000000e+00      0. 0. 0.
1.10123894e+01      6.50000000e+10      4.90000000e+10      4.90000000e+09      2.E+9      2.25000000e+01      0.00000000e+00      0. 0. 0.
1.15522124e+01      6.50000000e+10      4.60000000e+10      4.60000000e+09      2.E+8      2.65833333e+01      0.00000000e+00      0. 0. 0.
1.22000000e+01      6.50000000e+10      4.50000000e+10      4.50000000e+09      2.E+8      2.70000000e+01      0.00000000e+00      0. 0. 0.
1.25100000e+01      6.50000000e+10      4.50000000e+10      4.50000000e+09      2.E+8      0. 0.00000000e+00      0. 0. 0.

```

```

!aero_blade_table blname: wcs_v0_blade ! r/R c/R aero twist[deg] t/c airfol, from r3
0.0      0.04352  20.5  0.8288658      Cylinder1
0.090690 0.04352  20.5  0.8288658      Cylinder1
0.133876 0.0672   20.6  0.5748849      interpolate
0.181380 0.0864   20.1  0.4460525      interpolate
0.233204 0.09408  17.0  0.3809175      interpolate
0.285027 0.08896  12.1  0.3488297      interpolate
0.336849 0.08512  8.47  0.3291967      interpolate
0.388673 0.08128  6.04  0.3095637      NACA63_2XX_INNER
0.440495 0.07744  4.37  0.2935702      interpolate
0.492319 0.0736   3.04  0.2804608      interpolate
0.544142 0.06976  2.05  0.2673515      interpolate
0.595965 0.06592  1.33  0.2410318      interpolate
0.647788 0.06208  0.78  0.2213989      NACA63_221
0.699611 0.05824  0.42  0.2017659      interpolate
0.751434 0.05440  0.22  0.1849139      interpolate
0.803257 0.05056  0.11  0.1725000      NACA63_218
0.846442 0.04736  0.0   0.1592134      interpolate
0.880991 0.04480  0.0   0.1571405      interpolate
0.924177 0.0416   0.0   0.1545493      interpolate
0.976    0.0378  0.0   0.1514400      NACA63_215
1.01     0.0378  0.0   0.1514400      NACA63_215

```

```

aero_blade_table blname: wcs_v0_blade ! r/R c/R aero twist[deg] t/c airfoil, from r3
0.0      0.04   40.  1.      Cylinder1
0.03     0.04   40.  1.      Cylinder1
0.06     0.04   40.  1.      Cylinder1
0.090690 0.1     40.  0.8288658      interpolate
0.133876 0.1     27.  0.5748849      interpolate
0.181380 0.113  20.1  0.4460525      FFA_W3_360_RE_2_3E6
0.233204 0.102  16.  0.3809175      interpolate
0.285027 0.0915 12.5  0.3488297      FFA_W3_360_RE_2_3E6
0.336849 0.082   9.7  0.3291967      interpolate
0.388673 0.074   7.7  0.3095637      interpolate
0.440495 0.067   6.   0.2935702      FFA_W3_301_RE_2_3E6
0.492319 0.059   4.5  0.2804608      interpolate
0.544142 0.053   3.5  0.2673515      interpolate
0.595965 0.048   2.5  0.2410318      FFA_W3_241_RE_2_3E6
0.647788 0.045   0.   0.2213989      NACA63_221
0.699611 0.044  -1.  0.2017659      interpolate
0.751434 0.042  -1.  0.1849139      interpolate
0.803257 0.04   -1.  0.1725000      NACA63_218
0.846442 0.0385 -1.  0.1592134      interpolate
0.880991 0.036  -1.  0.1571405      interpolate
0.924177 0.032  -2.  0.1545493      interpolate
0.976    0.025  -1.  0.1514400      NACA63_215
1.0001   0.025  -1.  0.1514400      NACA63_215

```

```

read_airfoil
file NACA63_215.dat      afname NACA63_215  thick .15

read_airfoil
file NACA63_218.dat      afname NACA63_218  thick .18

read_airfoil
file NACA63_221.dat      afname NACA63_221  thick .21

read_airfoil
file Cylinder1.dat      afname Cylinder1  thick 1.

read_airfoil
file FFA_W3_360_RE_2_3E6.DAT  afname FFA_W3_360_RE_2_3E6  thick .36

read_airfoil
file FFA_W3_301_RE_2_3E6.DAT  afname FFA_W3_301_RE_2_3E6  thick .3

```

```

read_airfoil
file FFA_W3_241_RE_2_3E6.DAT  afname FFA_W3_241_RE_2_3E6  thick .24

new_body
name:      hub
parent:    rotor_1
reftype:   refnode
xfind:     -6.2 0. 0.
rotorder:  123
rotation:  0. 0. 0.
axes:      normal
tc_kin:    0.

new_body
name:      blade_1_root
parent:    hub
reftype:   refnode
xfind:     0. 0. 0.
rotorder:  123
rotation:  .001 0. 0.  ! 0. cone_angle_0
axes:      normal
tc_kin:    0.

new_body
name:      blade_1_pitch
parent:    blade_1_root
reftype:   refnode
xfind:     0. 0. 0.
rotorder:  123
rotation:  0. 0. 0.  ! 0. 0. pre_pitch neg towards feather
axes:      normal
tc_kin:    0.

new_elements_beam1  nelem 1 material s355  connect_mode auto  ! scale to hub mass
x1  0. 0. 0. 0.35  dcy11  0.3  tcy11  0.03
x2  0. 0. 0. 0.    dcy12  0.3  tcy12  0.03  btype beam

blade mass tag start  ! 875.5 kg for first version

new_elements_beam1  nelem 16 material blade_dummy connect_mode auto
x1  0. 0. 12.5 dcy11  1.  0.175  tcy11  0.1  0.01
x2  0. 0. 0.35 dcy12  3.  0.625  tcy12  0.3  0.01  btype beam
blname wcs_v0_blade  scale_mass 1. frac_stiff 1.e6  ! TODO check stiffness, rigid for now

blade mass tag end

new_body
name:      blade_2_root
parent:    hub
reftype:   refnode
xfind:     0. 0. 0.
rotorder:  123
rotation:  90.001 0. 0.  ! 0. cone_angle 0
axes:      normal
tc_kin:    0.

new_body
name:      blade_2_pitch
parent:    blade_2_root
reftype:   refnode
xfind:     0. 0. 0.
rotorder:  123
rotation:  0. 0. 0.  ! 0. 0. pre_pitch neg towards feather
axes:      normal
tc_kin:    0.

copy_elements
from_body: blade_1_pitch
to_body:   blade_2_pitch

new_body
name:      blade_3_root
parent:    hub

```

```

reftype: refnode
xfind: 0. 0. 0.
rotorder: 123
rotation: 180.001 0. 0.    ! 0. cone_angle 0
axes: normal
tc_kin: 0.

new_body
name: blade_3_pitch
parent: blade_3_root
reftype: refnode
xfind: 0. 0. 0.
rotorder: 123
rotation: 0. 0. 0.    ! 0. 0. pre_pitch neg towards feather
axes: normal
tc_kin: 0.

copy_elements
from_body: blade_2_pitch
to_body: blade_3_pitch

new_body
name: blade_4_root
parent: hub
reftype: refnode
xfind: 0. 0. 0.
rotorder: 123
rotation: 270.001 0. 0.    ! 0. cone_angle_0
axes: normal
tc_kin: 0.

new_body
name: blade_4_pitch
parent: blade_4_root
reftype: refnode
xfind: 0. 0. 0.
rotorder: 123
rotation: 0. 0. 0.    ! 0. 0. pre_pitch neg towards feather
axes: normal
tc_kin: 0.

copy_elements
from_body: blade_3_pitch
to_body: blade_4_pitch

rotor_wake
name: rotor_1_wcs
shaft_element_body: rotor_1
shaft_element_position: -6.075 0. 0.
rotating_node: 1
induction: all
logfile: rotor_1.txt
nblades: 4
bladenames: blade_1_pitch
blade_2_pitch
blade_3_pitch
blade_4_pitch
pitch_control_name: stall_1
pitch_actuator_bodies: blade_1_pitch
blade_2_pitch
blade_3_pitch
blade_4_pitch
pitch_actuator_positons: 0. 0. .175
0. 0. .175
0. 0. .175
0. 0. .175

pitchcontrol
control_type: none !iea_oc3_floating
name: stall_1
scale_omega: 1.
scale_vaxial: 1.
scale_pitchcoll: -1.0
omeganom: 1.2671
ti: 10.
gain: 1.
pitchref: 0.
aux: 1. 2. 3. 4. 5. 6. 7. 8. 9. 10.

!/monitors

```

```

set_max_monitors
  nsensmax: 300

monitor_element_b1 type wind_fixed find 0. 0. 30. nmonitor 1 node 2
file wind.txt body_name inertial label WindVxi WindVyi WindVzi

!Vetles monitorer

monitor_element_b1 type forces file front_hor_left_strut_force_x1.txt nmonitor 1
find -5.6250E+00 -6.8750E-01 0.0000E+00 node 2 body_name rotor_1 label front_hor_left_strut_fx_1 front_hor_left_strut_fy_1 front_hor_left_strut_fz_1

monitor_element_b1 type moments file front_hor_left_strut_moment_x1.txt nmonitor 1
find -5.6250E+00 -6.8750E-01 0.0000E+00 node 2 body_name rotor_1 label mx_fhls_1 my_fhls_1 mz_fhls_1

monitor_element_b1 type forces file front_hor_left_strut_force_x2.txt nmonitor 1
find -3.7500E-01 -1.0312E+01 0.0000E+00 node 2 body_name rotor_1 label front_hor_left_strut_fx_2 front_hor_left_strut_fy_2 front_hor_left_strut_fz_2

monitor_element_b1 type moments file front_hor_left_strut_moment_x2.txt nmonitor 1
find -3.7500E-01 -1.0312E+01 0.0000E+00 node 2 body_name rotor_1 label mx_fhls_2 my_fhls_2 mz_fhls_2

monitor_element_b1 type forces file front_hor_right_strut_force_x1.txt nmonitor 1
find -5.6250E+00 6.8750E-01 0.0000E+00 node 2 body_name rotor_1 label front_hor_right_strut_fx_1 front_hor_right_strut_fy_1 front_hor_right_strut_fz_1

monitor_element_b1 type moments file front_hor_right_strut_moment_x1.txt nmonitor 1
find -5.6250E+00 6.8750E-01 0.0000E+00 node 2 body_name rotor_1 label mx_fhrs_1 my_fhrs_1 mz_fhrs_1

monitor_element_b1 type forces file front_hor_right_strut_force_x2.txt nmonitor 1
find -3.7500E-01 1.0312E+01 0.0000E+00 node 2 body_name rotor_1 label front_hor_right_strut_fx_2 front_hor_right_strut_fy_2 front_hor_right_strut_fz_2

monitor_element_b1 type moments file front_hor_right_strut_moment_x2.txt nmonitor 1
find -3.7500E-01 1.0312E+01 0.0000E+00 node 2 body_name rotor_1 label mx_fhrs_2 my_fhrs_2 mz_fhrs_2

monitor_element_b1 type forces file front_vert_down_strut_force_x1.txt nmonitor 1
find -5.6250E+00 0.0000E+00 -7.8125E-01 node 2 body_name rotor_1 label front_vert_down_strut_fx_1 front_vert_down_strut_fy_1 front_vert_down_strut_fz_1

monitor_element_b1 type moments file front_vert_down_strut_moment_x1.txt nmonitor 1
find -5.6250E+00 0.0000E+00 -7.8125E-01 node 2 body_name rotor_1 label mx_fvds_1 my_fvds_1 mz_fvds_1

monitor_element_b1 type forces file front_vert_down_strut_force_x2.txt nmonitor 1
find -3.7500E-01 0.0000E+00 -1.1719E+01 node 2 body_name rotor_1 label front_vert_down_strut_fx_2 front_vert_down_strut_fy_2 front_vert_down_strut_fz_2

monitor_element_b1 type moments file front_vert_down_strut_moment_x2.txt nmonitor 1
find -3.7500E-01 0.0000E+00 -1.1719E+01 node 2 body_name rotor_1 label mx_fvds_2 my_fvds_2 mz_fvds_2

monitor_element_b1 type forces file front_vert_up_strut_force_x1.txt nmonitor 1
find -5.6250E+00 0.0000E+00 7.8125E-01 node 2 body_name rotor_1 label front_vert_up_strut_fx_1 front_vert_up_strut_fy_1 front_vert_up_strut_fz_1

monitor_element_b1 type moments file front_vert_up_strut_moment_x1.txt nmonitor 1
find -5.6250E+00 0.0000E+00 7.8125E-01 node 2 body_name rotor_1 label mx_fvus_1 my_fvus_1 mz_fvus_1

monitor_element_b1 type forces file front_vert_up_strut_force_x2.txt nmonitor 1
find -3.7500E-01 0.0000E+00 1.1719E+01 node 2 body_name rotor_1 label front_vert_up_strut_fx_2 front_vert_up_strut_fy_2 front_vert_up_strut_fz_2

monitor_element_b1 type moments file front_vert_up_strut_moment_x2.txt nmonitor 1
find -3.7500E-01 0.0000E+00 1.1719E+01 node 2 body_name rotor_1 label mx_fvus_2 my_fvus_2 mz_fvus_2

monitor_element_b1 type forces file rear_left_up_strut_force_x1.txt nmonitor 1
find 9.3750E-03 -6.8750E-01 9.3750E-01 node 2 body_name rotor_1 label rear_left_up_strut_fx_1 rear_left_up_strut_fy_1 rear_left_up_strut_fz_1

monitor_element_b1 type moments file rear_left_up_strut_moment_x1.txt nmonitor 1
find 9.3750E-03 -6.8750E-01 9.3750E-01 node 2 body_name rotor_1 label mx_rlus_1 my_rlus_1 mz_rlus_1

monitor_element_b1 type forces file rear_left_up_strut_force_x2.txt nmonitor 1
find 6.2500E-04 -1.0312E+01 6.2500E-02 node 2 body_name rotor_1 label rear_left_up_strut_fx_2 rear_left_up_strut_fy_2 rear_left_up_strut_fz_2

monitor_element_b1 type moments file rear_left_up_strut_moment_x2.txt nmonitor 1
find 6.2500E-04 -1.0312E+01 6.2500E-02 node 2 body_name rotor_1 label mx_rlus_2 my_rlus_2 mz_rlus_2

monitor_element_b1 type forces file rear_left_down_strut_force_x1.txt nmonitor 1
find 9.3750E-03 -6.8750E-01 -9.3750E-01 node 2 body_name rotor_1 label rear_left_down_strut_fx_1 rear_left_down_strut_fy_1 rear_left_down_strut_fz_1

monitor_element_b1 type moments file rear_left_down_strut_moment_x1.txt nmonitor 1
find 9.3750E-03 -6.8750E-01 -9.3750E-01 node 2 body_name rotor_1 label mx_rlds_1 my_rlds_1 mz_rlds_1

monitor_element_b1 type forces file rear_left_down_strut_force_x2.txt nmonitor 1
find 6.2500E-04 -1.0312E+01 -6.2500E-02 node 2 body_name rotor_1 label rear_left_down_strut_fx_2 rear_left_down_strut_fy_2 rear_left_down_strut_fz_2

monitor_element_b1 type moments file rear_left_down_strut_moment_x2.txt nmonitor 1
find 6.2500E-04 -1.0312E+01 -6.2500E-02 node 2 body_name rotor_1 label mx_rlds_2 my_rlds_2 mz_rlds_2

monitor_element_b1 type forces file rear_right_up_strut_force_x1.txt nmonitor 1
find 9.3750E-03 6.8750E-01 9.3750E-01 node 2 body_name rotor_1 label rear_right_up_strut_fx_1 rear_right_up_strut_fy_1 rear_right_up_strut_fz_1

```



```

monitor_element_b1 type moments file rear_right_up_strut_moment_x1.txt nmonitor 1
find 9.3750E-03 6.8750E-01 9.3750E-01 node 2 body_name rotor_1 label mx_rrus_1 my_rrus_1 mz_rrus_1

monitor_element_b1 type forces file rear_right_up_strut_force_x2.txt nmonitor 1
find 6.2500E-04 1.0312E+01 6.2500E-02 node 2 body_name rotor_1 label rear_right_up_strut_fx_2 rear_right_up_strut_fy_2 rear_right_up_strut_fz_2

monitor_element_b1 type moments file rear_right_up_strut_moment_x2.txt nmonitor 1
find 6.2500E-04 1.0312E+01 6.2500E-02 node 2 body_name rotor_1 label mx_rrus_2 my_rrus_2 mz_rrus_2

monitor_element_b1 type forces file rear_right_down_strut_force_x1.txt nmonitor 1
find 9.3750E-03 6.8750E-01 -9.3750E-01 node 2 body_name rotor_1 label rear_right_down_strut_fx_1 rear_right_down_strut_fy_1 rear_right_down_strut_fz_1

monitor_element_b1 type moments file rear_right_down_strut_moment_x1.txt nmonitor 1
find 9.3750E-03 6.8750E-01 -9.3750E-01 node 2 body_name rotor_1 label mx_rrds_1 my_rrds_1 mz_rrds_1

monitor_element_b1 type forces file rear_right_down_strut_force_x2.txt nmonitor 1
find 6.2500E-04 1.0312E+01 -6.2500E-02 node 2 body_name rotor_1 label rear_right_down_strut_fx_2 rear_right_down_strut_fy_2 rear_right_down_strut_fz_2

monitor_element_b1 type moments file rear_right_down_strut_moment_x2.txt nmonitor 1
find 6.2500E-04 1.0312E+01 -6.2500E-02 node 2 body_name rotor_1 label mx_rrds_2 my_rrds_2 mz_rrds_2

monitor_element_b1 type forces file rear_vert_down_strut_force_x1.txt nmonitor 1
find 9.3750E-03 0.0000E+00 -7.8125E-01 node 2 body_name rotor_1 label rear_vert_down_strut_fx_1 rear_vert_down_strut_fy_1 rear_vert_down_strut_fz_1

monitor_element_b1 type moments file rear_vert_down_strut_moment_x1.txt nmonitor 1
find 9.3750E-03 0.0000E+00 -7.8125E-01 node 2 body_name rotor_1 label mx_rvds_1 my_rvds_1 mz_rvds_1

monitor_element_b1 type forces file rear_vert_down_strut_force_x2.txt nmonitor 1
find 6.2500E-04 0.0000E+00 -1.1719E+01 node 2 body_name rotor_1 label rear_vert_down_strut_fx_2 rear_vert_down_strut_fy_2 rear_vert_down_strut_fz_2

monitor_element_b1 type moments file rear_vert_down_strut_moment_x2.txt nmonitor 1
find 6.2500E-04 0.0000E+00 -1.1719E+01 node 2 body_name rotor_1 label mx_rvds_2 my_rvds_2 mz_rvds_2

monitor_element_b1 type forces file rear_vert_up_strut_force_x1.txt nmonitor 1
find 9.3750E-03 0.0000E+00 7.8125E-01 node 2 body_name rotor_1 label rear_vert_up_strut_fx_1 rear_vert_up_strut_fy_1 rear_vert_up_strut_fz_1

monitor_element_b1 type moments file rear_vert_up_strut_moment_x1.txt nmonitor 1
find 9.3750E-03 0.0000E+00 7.8125E-01 node 2 body_name rotor_1 label mx_rvus_1 my_rvus_1 mz_rvus_1

monitor_element_b1 type forces file rear_vert_up_strut_force_x2.txt nmonitor 1
find 6.2500E-04 0.0000E+00 1.1719E+01 node 2 body_name rotor_1 label rear_vert_up_strut_fx_2 rear_vert_up_strut_fy_2 rear_vert_up_strut_fz_2

monitor_element_b1 type moments file rear_vert_up_strut_moment_x2.txt nmonitor 1
find 6.2500E-04 0.0000E+00 1.1719E+01 node 2 body_name rotor_1 label mx_rvus_2 my_rvus_2 mz_rvus_2

monitor_element_b1 type forces file rotor_shaft_force.txt nmonitor 1
find -6.1500E+00 0.0000E+00 0.0000E+01 node 1 body_name rotor_1 label shaft_fx shaft_fy shaft_fz

monitor_element_b1 type moments file rotor_shaft_moment.txt nmonitor 1
find -6.1500E+00 0.0000E+00 0.0000E+01 node 1 body_name rotor_1 label shaft_mx shaft_my shaft_mz

!TAN force sensor nonrotating system, note element has been added

monitor_element_b1 type forces file rotor_force.txt nmonitor 1
find -6.0250E+00 0.0000E+00 0.0000E+00 node 1 body_name rotor_1 label rotor_fx rotor_fy rotor_fz

monitor_element_b1 type moments file rotor_moment.txt nmonitor 1
find -6.0250E+00 0.0000E+00 0.0000E+00 node 1 body_name rotor_1 label rotor_mx rotor_my rotor_mz

monitor_element_b1 type forces file blade_1_root_force.txt nmonitor 1
find 0. 0. 0.175 node 2 body_name blade_1_root label fx_bl_1_root fy_bl_1_root fz_bl_1_root

monitor_element_b1 type moments file blade_1_root_moment.txt nmonitor 1
find 0. 0. 0.175 node 2 body_name blade_1_root label mx_bl_1_root my_bl_1_root mz_bl_1_root

!midpoints for front struts to calculate buckling

monitor_element_b1 type forces file front_hor_left_strut_force_x3.txt nmonitor 1
find -3.0000E+00 -5.5000E+00 0.0000E+00 node 2 body_name rotor_1 label front_hor_left_strut_fx_3 front_hor_left_strut_fy_3 front_hor_left_strut_fz_3

monitor_element_b1 type moments file front_hor_left_strut_moment_x3.txt nmonitor 1
find -3.0000E+00 -5.5000E+00 0.0000E+00 node 2 body_name rotor_1 label mx_fhls_3 my_fhls_3 mz_fhls_3

monitor_element_b1 type forces file front_hor_right_strut_force_x3.txt nmonitor 1
find -3.0000E+00 5.5000E+00 0.0000E+00 node 2 body_name rotor_1 label front_hor_right_strut_fx_3 front_hor_right_strut_fy_3 front_hor_right_strut_fz_3

monitor_element_b1 type moments file front_hor_right_strut_moment_x3.txt nmonitor 1
find -3.0000E+00 5.5000E+00 0.0000E+00 node 2 body_name rotor_1 label mx_fhrs_3 my_fhrs_3 mz_fhrs_3

monitor_element_b1 type forces file front_vert_down_strut_force_x3.txt nmonitor 1
find -3.0000E+00 0.0000E+00 -6.2500E+00 node 2 body_name rotor_1 label front_vert_down_strut_fx_3 front_vert_down_strut_fy_3 front_vert_down_strut_fz_3

```

```

monitor_element_b1 type moments file front_vert_down_strut_moment_x3.txt nmonitor 1
find -3.0000E+00 0.0000E+00 -6.2500E+00 node 2 body_name rotor_1 label mx_fvds_3 my_fvds_3 mz_fvds_3

monitor_element_b1 type forces file front_vert_up_strut_force_x3.txt nmonitor 1
find -3.0000E+00 0.0000E+00 6.2500E+00 node 2 body_name rotor_1 label front_vert_up_strut_fx_3 front_vert_up_strut_fy_3 front_vert_up_strut_fz_3

monitor_element_b1 type moments file front_vert_up_strut_moment_x3.txt nmonitor 1
find -3.0000E+00 0.0000E+00 6.2500E+00 node 2 body_name rotor_1 label mx_fvds_3 my_fvds_3 mz_fvds_3

monitor_element_b1 find 0. 0. -20. nmonitor 1 node 1 type gather
file sensors.txt label wcs_gather toffset 10. 180.
sensors:
nfact: 1 source: WindVxi scale: 1. unit: [m/s] ! header as source label
nfact: 1 source: WindVyi scale: 1. unit: [m/s] !
nfact: 1 source: WindVzi scale: 1. unit: [m/s] !
nfact: 1 source: Azimuth_1 scale: 57.295779513 unit: [deg] !
nfact: 1 source: AeroThr_1 scale: 1.e-3 unit: [kN] !
nfact: 1 source: AeroTq_1 scale: 1.e-3 unit: [kNm] !
nfact: 1 source: RotSpeed_1 scale: 9.549296585513721 unit: [rpm] header: RotSpeed ! 30./pi
nfact: 1 source: RotSpeed_1 scale: 1 unit: [rad/s] header: Angular_vel ! 30./pi
nfact: 1 source: Vaxial_1 scale: 1. unit: [m/s] !
nfact: 1 source: shaft_fx scale: 1.e-3 unit: [kN] !
nfact: 1 source: shaft_fy scale: 1.e-3 unit: [kN] !
nfact: 1 source: shaft_fz scale: 1.e-3 unit: [kN] !
nfact: 1 source: shaft_mx scale: 1.e-3 unit: [kNm] !
nfact: 1 source: shaft_my scale: 1.e-3 unit: [kNm] !
nfact: 1 source: shaft_mz scale: 1.e-3 unit: [kNm] !
nfact: 1 source: fx_bl_1_root scale: 1.e-3 unit: [kN]
nfact: 1 source: fy_bl_1_root scale: 1.e-3 unit: [kN]
nfact: 1 source: fz_bl_1_root scale: 1.e-3 unit: [kN]
nfact: 1 source: mx_bl_1_root scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_bl_1_root scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_bl_1_root scale: 1.e-3 unit: [kNm]
nfact: 1 source: front_hor_left_strut_fx_1 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_fhls_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_fhls_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_fhls_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: front_hor_left_strut_fx_2 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_fhls_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_fhls_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_fhls_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: front_hor_right_strut_fx_1 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_fhrs_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_fhrs_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_fhrs_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: front_hor_right_strut_fx_2 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_fhrs_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_fhrs_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_fhrs_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: front_vert_down_strut_fx_1 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_fvds_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_fvds_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_fvds_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: front_vert_down_strut_fx_2 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_fvds_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_fvds_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_fvds_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: front_vert_up_strut_fx_1 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_fvus_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_fvus_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_fvus_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: front_vert_up_strut_fx_2 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_fvus_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_fvus_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_fvus_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: rear_left_down_strut_fx_1 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_rlids_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_rlids_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_rlids_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: rear_left_down_strut_fx_2 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_rlids_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_rlids_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_rlids_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: rear_left_up_strut_fx_1 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_rlus_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_rlus_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: mz_rlus_1 scale: 1.e-3 unit: [kNm]
nfact: 1 source: rear_left_up_strut_fx_2 scale: 1.e-3 unit: [kN] !
nfact: 1 source: mx_rlus_2 scale: 1.e-3 unit: [kNm]
nfact: 1 source: my_rlus_2 scale: 1.e-3 unit: [kNm]

```



```

nfact: 1 source: my_fhrs_3      scale: 1.e-3      unit: [kNm]
nfact: 1 source: mz_fhrs_3      scale: 1.e-3      unit: [kNm]
nfact: 1 source: front_vert_down_strut_fx_3  scale: 1.e-3      unit: [kN] !
nfact: 1 source: front_vert_down_strut_fy_3  scale: 1.e-3      unit: [kN] !
nfact: 1 source: front_vert_down_strut_fz_3  scale: 1.e-3      unit: [kN] !
nfact: 1 source: mx_fvds_3      scale: 1.e-3      unit: [kNm]
nfact: 1 source: my_fvds_3      scale: 1.e-3      unit: [kNm]
nfact: 1 source: mz_fvds_3      scale: 1.e-3      unit: [kNm]
nfact: 1 source: front_vert_up_strut_fx_3    scale: 1.e-3      unit: [kN] !
nfact: 1 source: front_vert_up_strut_fy_3    scale: 1.e-3      unit: [kN] !
nfact: 1 source: front_vert_up_strut_fz_3    scale: 1.e-3      unit: [kN] !
nfact: 1 source: mx_fvus_3      scale: 1.e-3      unit: [kNm]
nfact: 1 source: my_fvus_3      scale: 1.e-3      unit: [kNm]
nfact: 1 source: mz_fvus_3      scale: 1.e-3      unit: [kNm]
nfact: 1 source: rotor_fx      scale: 1.e-3      unit: [kN] !
nfact: 1 source: rotor_fy      scale: 1.e-3      unit: [kN] !
nfact: 1 source: rotor_fz      scale: 1.e-3      unit: [kN] !
nfact: 1 source: rotor_mx      scale: 1.e-3      unit: [kN] !
nfact: 1 source: rotor_my      scale: 1.e-3      unit: [kN] !
nfact: 1 source: rotor_mz      scale: 1.e-3      unit: [kN] !
nfact: 1 source: dummy_end      scale: 1.e-3      unit: [kN] !

! old junk
!nfact: 1 source: AeroTq_1      scale: 1.e-3      unit: [kNm] !
!nfact: 1 source: GenTq_S_1     scale: 1.e-3      unit: [kNm] !
!nfact: 1 source: GenTq_C_1     scale: 1.e-3      unit: [kNm] !
!nfact: 1 source: BldPitch1_1   scale: -57.295779513082323 unit: [deg] header: BldPitch1 ! from control system
!nfact: 1 source: BldPitch1_1   scale: 57.295779513 unit: [deg] !
!nfact: 1 source: GenPwr_1      scale: 1.e-3      unit: [kW] !

! more old junk
!nfact: 1 source: front_hor_left_strut_fx    scale: 1.e-3      unit: [kN] !
!nfact: 1 source: front_hor_right_strut_fx   scale: 1.e-3      unit: [kN] !
!nfact: 1 source: front_vert_low_strut_fx    scale: 1.e-3      unit: [kN] !
!nfact: 1 source: front_vert_up_strut_fx     scale: 1.e-3      unit: [kN] !
!nfact: 1 source: front_vert_up_strut_stress scale: 1.e-6      unit: [MPa] !
!nfact: 1 source: shaft_fx                  scale: 1.e-3      unit: [kN] !
!nfact: 1 source: shaft_fy                  scale: 1.e-3      unit: [kN] !
!nfact: 1 source: shaft_fz                  scale: 1.e-3      unit: [kN] !
!nfact: 1 source: dummy_end                  scale: 1.e-3      unit: [kN] !

! more strut midpoints
!5.0000E-03  0.0000E+00  3.6250E+01
!5.0000E-03  0.0000E+00  2.3750E+01
!5.0000E-03  5.5000E+00  2.9500E+01
!5.0000E-03  5.5000E+00  3.0500E+01
!5.0000E-03 -5.5000E+00  2.9500E+01
!5.0000E-03 -5.5000E+00  3.0500E+01
!-3.0000E+00  0.0000E+00  3.6250E+01
!-3.0000E+00  0.0000E+00  2.3750E+01
!-3.0000E+00  5.5000E+00  3.0000E+01
!-3.0000E+00 -5.5000E+00  3.0000E+01

!
! Boundary conditions
!

apply_nodal_bc find 0. 0. 30. apply_displ idofs 1 idofe 6 displ 0. 0. 0. 0. 0. 0. ! Hold dummy element

apply_nodal_bc find 0. -11. 17.5 rotor_1 apply_displ idofs 1 idofe 6 displ 0. 0. 0. 0. 0. 0. ! lower left, in inertial system

apply_nodal_bc find 0. 11. 17.5 apply_displ idofs 1 idofe 6 displ 0. 0. 0. 0. 0. 0. ! lower right

apply_nodal_bc find 0. -11. 42.5 apply_displ idofs 1 idofe 6 displ 0. 0. 0. 0. 0. 0. ! upper left

apply_nodal_bc find 0. 11. 42.5 apply_displ idofs 1 idofe 6 displ 0. 0. 0. 0. 0. 0. ! upper right

!apply_nodal_bc find 0. 11. 17.5 apply_displ idofs 1 idofe 1 displ 0. ! lower right
!apply_nodal_bc find 0. 11. 17.5 apply_displ idofs 3 idofe 3 displ 0. ! lower right
!apply_nodal_bc find 0. -11. 42.5 apply_displ idofs 1 idofe 1 displ 0. ! upper left
!apply_nodal_bc find 0. 11. 42.5 apply_displ idofs 1 idofe 1 displ 0. ! upper right

cg_cb
blade_1_pitch

```

```
wind_transient
0. 15. 0.
100. 15. 0.

0. 10. 0. ! Needs a boost to started
100. 10. 0.
101. 7. 0.
200. 7. 0.
3000. 30. 0.

! C:\Users\toran\OneDrive - Institutt for Energiteknikk\project2\wcs_2020\turbsim\15ms_ti10

turbulence_box
turbulence: file
turbulence_info: ..\..\t15ms_ti10.bts
turb_scale: none ! spatial_scaling !
turb_tstart: 0.
turb_intens: .1
reference_node: 0. 0. 0. ! box moves slowly horizontally with SWL node
reference_height: 30. ! hub height, center of box is placed here

damping type rayleigh alpha 0. beta .002

#damping type ratio2rayl ratio_1 .01 ratio_2 .01 omega_1 0.1 omega_2 1.

newmark
dt: 0.01
nassemble: 1
nnewton: 20
nsubmin: 2
rwilson: .9
relax: 0.
resid_newton: 1e-8
nmonitor: 100

step nstep 370000 method step9

monitor_element_b1 type tecplot nmonitor 5
file turb1.dat

step nstep 500 method step9

! Eigen analysis
! Aim minimum 5.5Hz for nacelle system

!eigen_analysis amplitude 5. nfreq 20 filename plotall

end
```

D. Python Routine For Assessing Occurences Of Buckling

```
1  ## coding: utf-8 ##
2  """
3  Created on Thu May 20 14:44:57 2021
4
5  @author: veaa
6  """
7
8  #Buckling of front struts
9  # EUROCODE 3
10 # Note to self: remember to check if the moments are added such that you'
    re investigating the point that leads to compression from both of the
    moments
11
12 from pylab import *
13 from numpy import loadtxt
14 import numpy as np
15
16
17 #importing data
18 a = loadtxt('sensors.txt',skiprows=50000,dtype='float',delimiter=';') #
    skip 500 s
19
20 pi = 3.14159 #
21
22 #cross-sectional properties and length
23 d = 0.2 #outer diameter of strut [m]
24 twall = 0.008 #wall thickness [m]
25 di = d - 2.*twall #inner diameter of strut [m]
26 A = .25*pi*(d**2-di**2) #area of cross-section (cylinder) [m^2]
27 icyl = pi*(d**4-di**4)/64. #second moment of inertia for cylinder [m^4]
28 W_pl = (1/6)*(d**3-di**3) #plastic section modulus [m^3]
29 W_el = icyl/(d/2) #elastic section modulus [m^3]
30 L_cr = 10 #Buckle length [m]
31
32 #Tverrsnittsklasse
33 epsilon = sqrt(235/355)
34
```

```

35 #tv.klasse 1
36 if d/twall <= 50*epsilon**2:
37     W_el = W_pl
38 #tv.klasse 2
39 if d/twall <= 70*epsilon**2:
40     W_el = W_pl
41 #tv.klasse 3
42 d/twall <= 90*epsilon**2
43
44
45
46 #Material properties
47 gamma_M1 = 1.05    # [Dimensionless]
48 E = 210*10**9      # [Pa]
49 f_y = 355*10**6    # [Pa]
50
51
52 #Euler's critical load
53 N_cr = (pi**2*E*I)/(L_cr**2) # [N]
54
55 #Relative slenderness
56 lambda_ybar = sqrt((A*f_y)/N_cr)
57
58 #form deviation factor from table: cyrve a = 0.21
59 alpha = 0.21
60
61 #reduction factor for buckling (khi)
62 phi = 0.5*(1+alpha*(lambda_ybar-0.2)+(lambda_ybar)**2)
63
64 khi = (1)/(phi+sqrt((phi)**2-(lambda_ybar)**2))
65
66 #Design criterias
67 N_bxRd = (khi*f_y*A)/(gamma_M1)
68
69 M_elxRd = (f_y*W_el)/(gamma_M1)
70
71
72
73 K = []
74 KK = []
75 KKK = []
76 KKKK = []
77
78
79 #front_hor_left_strut_x1
80 front_hor_left_strut_fx_1 = a[:,22]*1.e3; #[N]
81 my_fhls_1 = a[:,24]*1.e3; #[Nm]
82 mz_fhls_1 = a[:,25]*1.e3; #[Nm]
83

```

```

84
85
86 for i in range(0, len(front_hor_left_strut_fx_1)):
87     S = (front_hor_left_strut_fx_1[i]/N_bxRd) + ((-abs(my_fhls_1[i])-abs(
           mz_fhls_1[i]))/M_elxRd)
88     K.append(S)
89
90 K_min = (min(K))
91
92 if abs(K_min)>1:
93     print('%.2e, and front_hor_left_strut will likely buckle under the
           current circumstances' %K_min)
94 else :
95     print('%.2e, front_hor_left_strut is not prone to buckling under the
           current circumstances' %K_min)
96
97
98 #front_hor_right_strut_x1
99 front_hor_right_strut_fx_1 = a[:,30]*1.e3;
100 my_fhrs_1 = a[:,32]*1.e3;
101 mz_fhrs_1 = a[:,33]*1.e3;
102
103 for i in range(0, len(front_hor_right_strut_fx_1)):
104     SS = (front_hor_right_strut_fx_1[i]/N_bxRd) + ((-abs(my_fhrs_1[i])-abs
           (mz_fhrs_1[i]))/M_elxRd)
105     KK.append(SS)
106
107 KK_min = (min(KK))
108
109 if abs(KK_min)>1:
110     print('%.2e, and front_hor_right_strut will likely buckle under the
           current circumstances' %KK_min)
111 else :
112     print('%.2e, front_hor_right_strut is not prone to buckling under the
           current circumstances' %KK_min)
113
114 #front_vert_down_strut_x1
115 front_vert_down_strut_fx_1 = a[:,38]*1.e3;
116 my_fvds_1 = a[:,40]*1.e3;
117 mz_fvds_1 = a[:,41]*1.e3;
118
119 for i in range(0, len(front_vert_down_strut_fx_1)):
120     SSS = (front_vert_down_strut_fx_1[i]/N_bxRd) + ((-abs(my_fvds_1[i])-
           abs(mz_fvds_1[i]))/M_elxRd)
121     KKK.append(SSS)
122
123 KKK_min = (min(KKK))
124
125 if abs(KKK_min)>1:

```



```

126     print( '%.2e, and front_vert_down_strut will likely buckle under the
           current circumstances' %KKK_min)
127 else :
128     print( '%.2e, front_vert_down_strut is not prone to buckling under the
           current circumstances' %KKK_min)
129
130 #front_vert_up_strut_x1
131 front_vert_up_strut_fx_1 = a[:,46]*1.e3;
132 my_fvus_1 = a[:,48]*1.e3;
133 mz_fvus_1 = a[:,49]*1.e3;
134
135 for i in range(0,len(front_hor_left_strut_fx_1)):
136     SSSS = (front_vert_up_strut_fx_1[i]/N_bxRd) + ((-abs(my_fvus_1[i])-abs
           (mz_fvus_1[i]))/M_elxRd)
137     KKKK.append(SSSS)
138
139 KKKK_min = (min(KKKK))
140
141 if abs(KKKK_min)>1:
142     print( '%.2e, and front_vert_up_strut will likely buckle under the
           current circumstances' %KKKK_min)
143 else :
144     print( '%.2e, front_vert_up_strut is not prone to buckling under the
           current circumstances' %KKKK_min)
145
146
147 #-----
148
149 B = []
150 BB = []
151 BBB = []
152 BBBB = []
153
154 #front_hor_left_strut_x2
155 front_hor_left_strut_fx_2 = a[:,26]*1.e3;
156 my_fhls_2 = a[:,28]*1.e3;
157 mz_fhls_2 = a[:,29]*1.e3;
158
159
160
161 for i in range(0,len(front_hor_left_strut_fx_1)):
162     T = (front_hor_left_strut_fx_2[i]/N_bxRd) + ((-abs(my_fhls_2[i])-abs(
           mz_fhls_2[i]))/M_elxRd)
163     B.append(T)
164
165 B_min = (min(B))
166
167 if abs(B_min)>1:
168     print( '%.2e, and front_hor_left_strut will likely buckle under the

```

```

        current circumstances' %B_min)
169 else :
170     print('%.2e, front_hor_left_strut is not prone to buckling under the
        current circumstances' %B_min)
171
172
173 #front_hor_right_strut_x2
174 front_hor_right_strut_fx_2 = a[:,34]*1.e3;
175 my_fhrs_2 = a[:,36]*1.e3;
176 mz_fhrs_2 = a[:,37]*1.e3;
177
178 for i in range(0,len(front_hor_left_strut_fx_1)):
179     TT = (front_hor_right_strut_fx_2[i]/N_bxRd) + ((-abs(my_fhrs_2[i])-abs
        (mz_fhrs_2[i]))/M_elxRd)
180     BB.append(TT)
181
182 BB_min = (min(BB))
183
184 if abs(BB_min)>1:
185     print('%.2e, and front_hor_right_strut will likely buckle under the
        current circumstances' %BB_min)
186 else :
187     print('%.2e, front_hor_right_strut is not prone to buckling under the
        current circumstances' %BB_min)
188
189 #front_vert_down_strut_x2
190 front_vert_down_strut_fx_2 = a[:,42]*1.e3;
191 my_fvds_2 = a[:,44]*1.e3;
192 mz_fvds_2 = a[:,45]*1.e3;
193
194 for i in range(0,len(front_hor_left_strut_fx_1)):
195     TTT = (front_vert_down_strut_fx_2[i]/N_bxRd) + ((-abs(my_fvds_2[i])-
        abs(mz_fvds_2[i]))/M_elxRd)
196     BBB.append(TTT)
197
198 BBB_min = (min(BBB))
199
200 if abs(BBB_min)>1:
201     print('%.2e, and front_vert_down_strut will likely buckle under the
        current circumstances' %BBB_min)
202 else :
203     print('%.2e, front_vert_down_strut is not prone to buckling under the
        current circumstances' %BBB_min)
204
205 #front_vert_up_strut_x2
206 front_vert_up_strut_fx_2 = a[:,50]*1.e3;
207 my_fvus_2 = a[:,52]*1.e3;
208 mz_fvus_2 = a[:,53]*1.e3;
209

```

```
210 for i in range(0, len(front_hor_left_strut_fx_1)):
211     TTTT = (front_vert_up_strut_fx_2[i]/N_bxRd) + ((-abs(my_fvus_2[i])-abs
           (mz_fvus_2[i]))/M_elxRd)
212     BBBB.append(TTTT)
213
214 BBBB_min = (min(BBBB))
215
216 if abs(BBBB_min)>1:
217     print( '%.2e, and front_vert_up_strut will likely buckle under the
           current circumstances' %BBBB_min)
218 else :
219     print( '%.2e, front_vert_up_strut is not prone to buckling under the
           current circumstances' %BBBB_min)
```

E. Fatigue Life vs. Wall Thickness

2-Bladed Structure

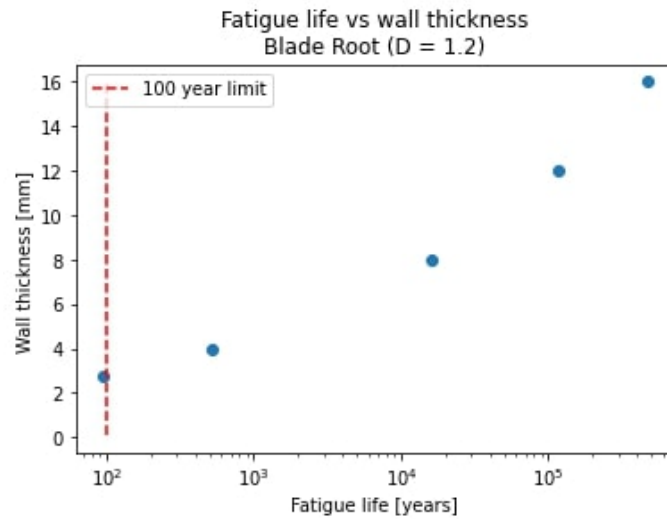


Figure E.1

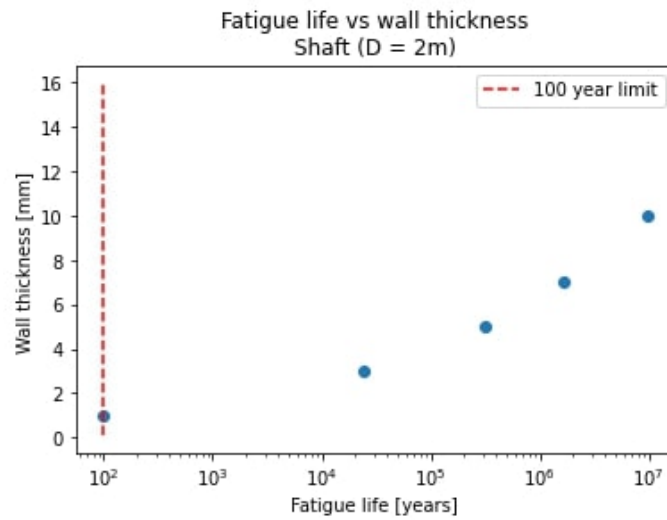


Figure E.2

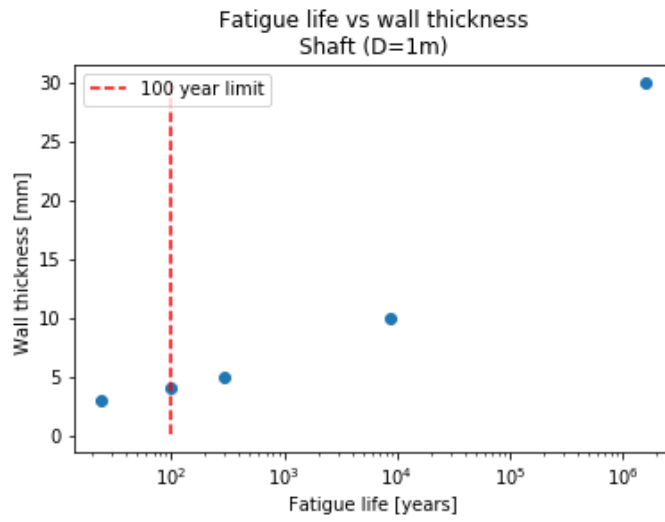


Figure E.3

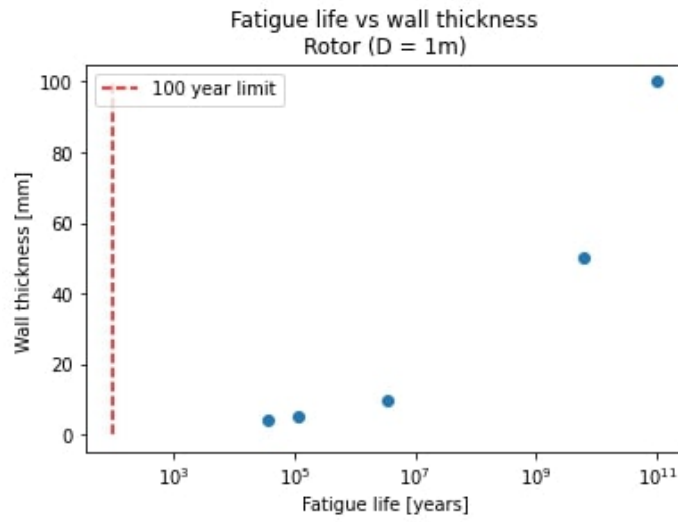


Figure E.4

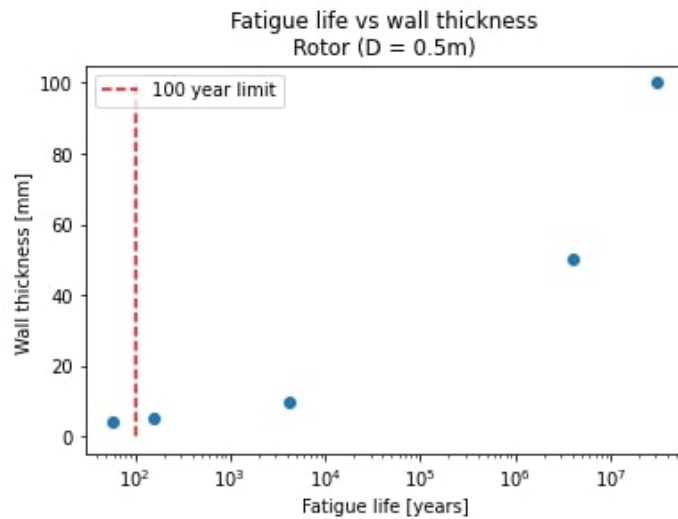


Figure E.5

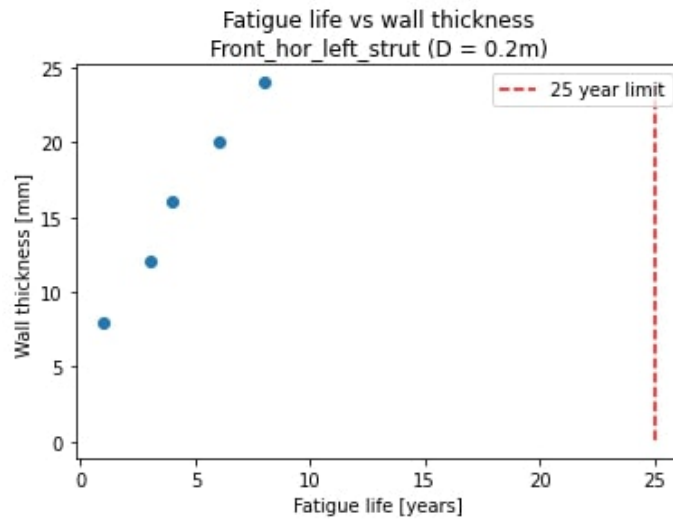


Figure E.6

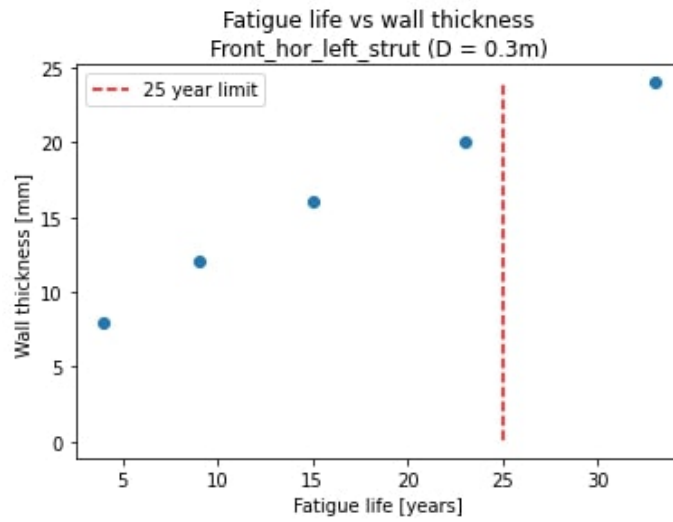


Figure E.7

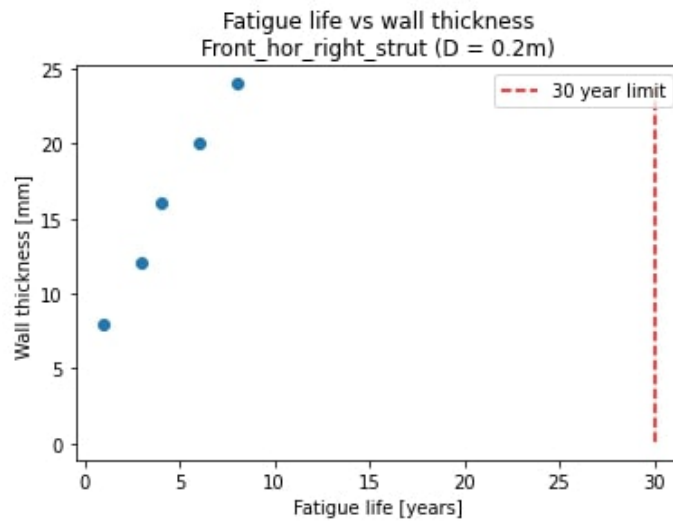


Figure E.8

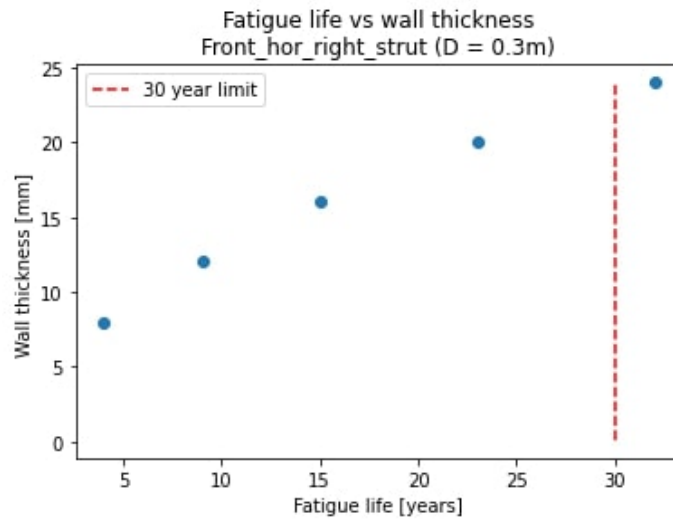


Figure E.9

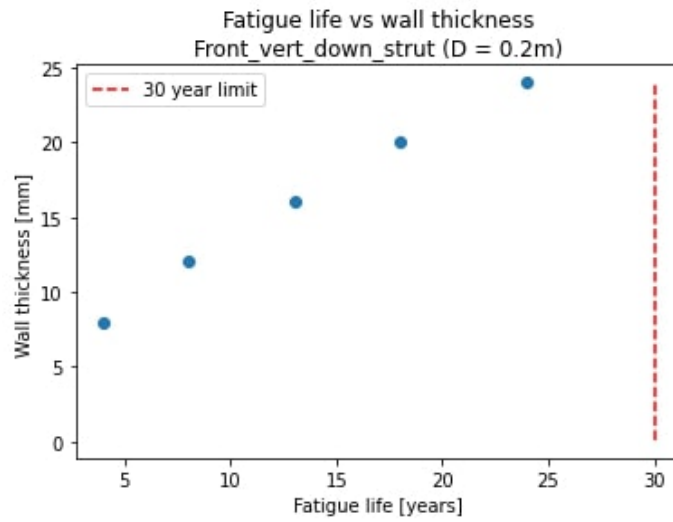


Figure E.10

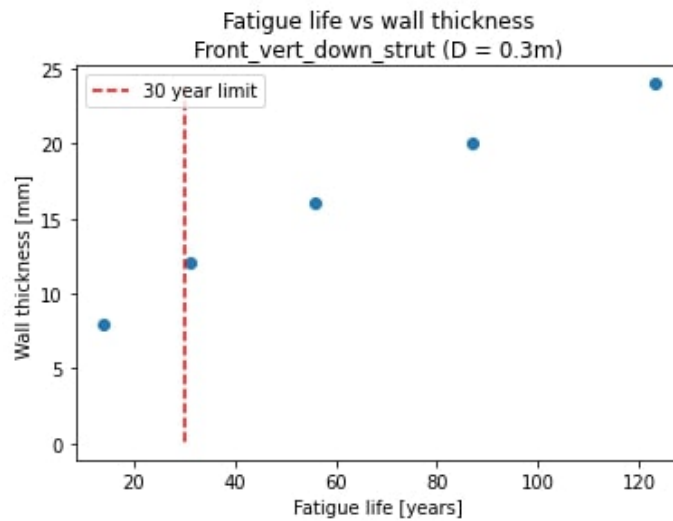


Figure E.11

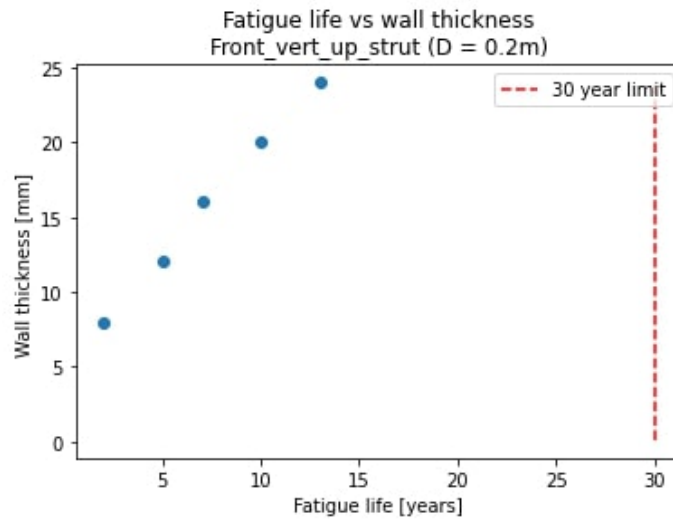


Figure E.12

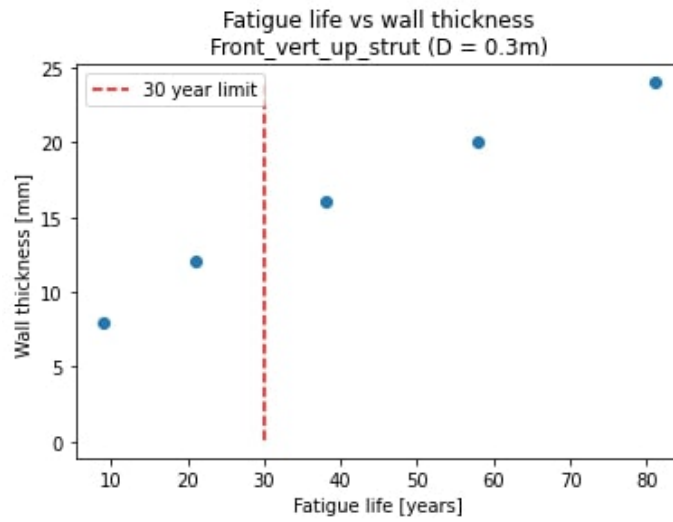


Figure E.13

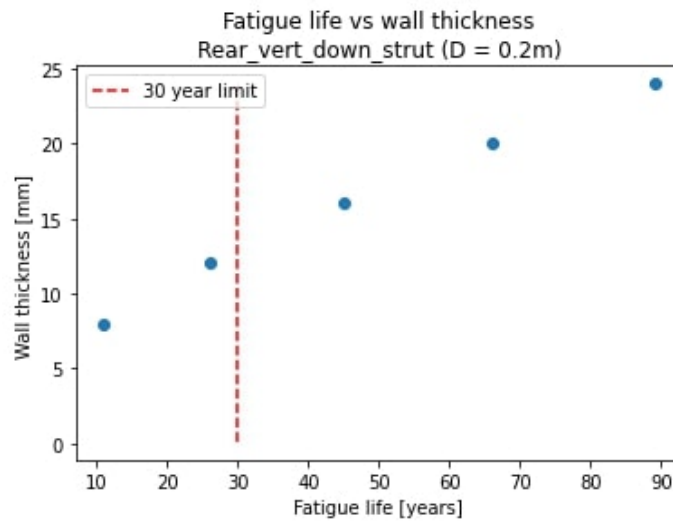


Figure E.14

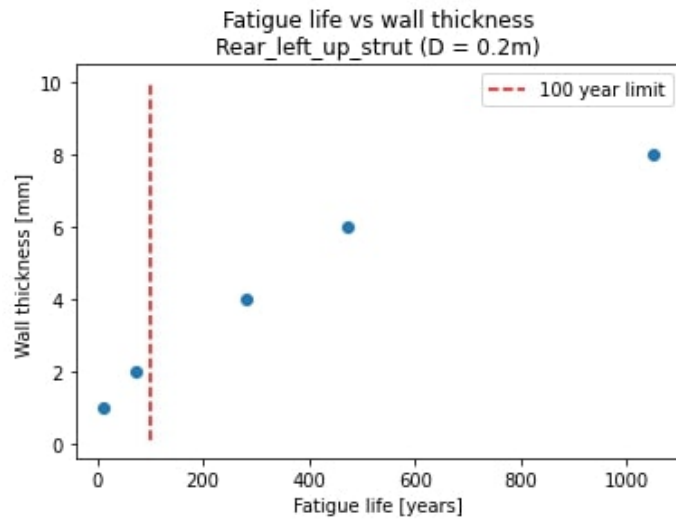


Figure E.15

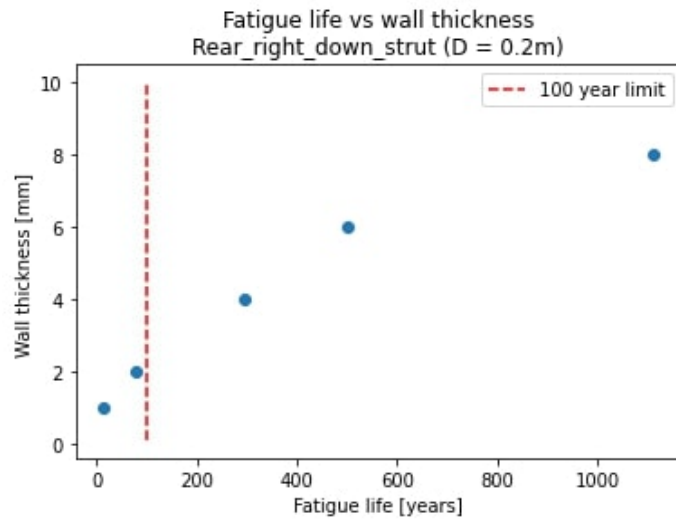


Figure E.16

4-Bladed Structure

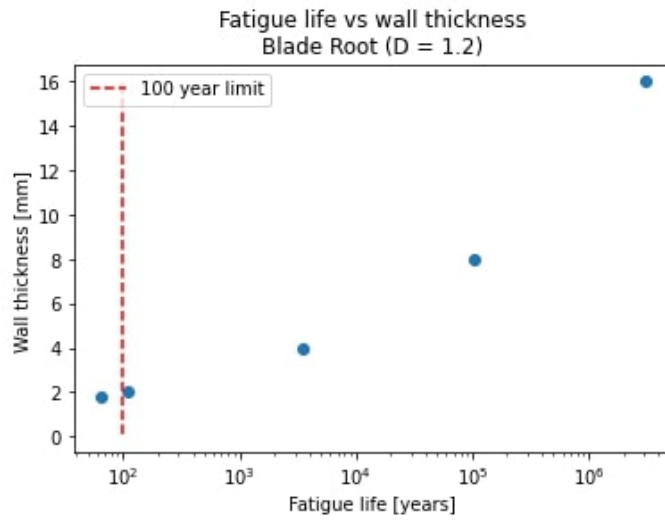


Figure E.17

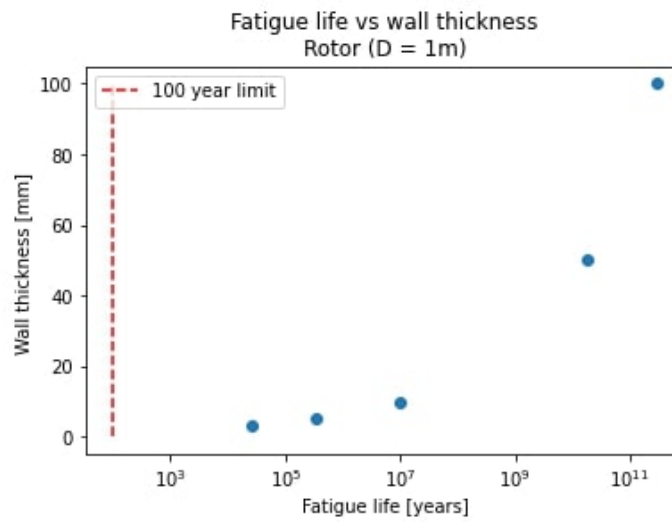


Figure E.18

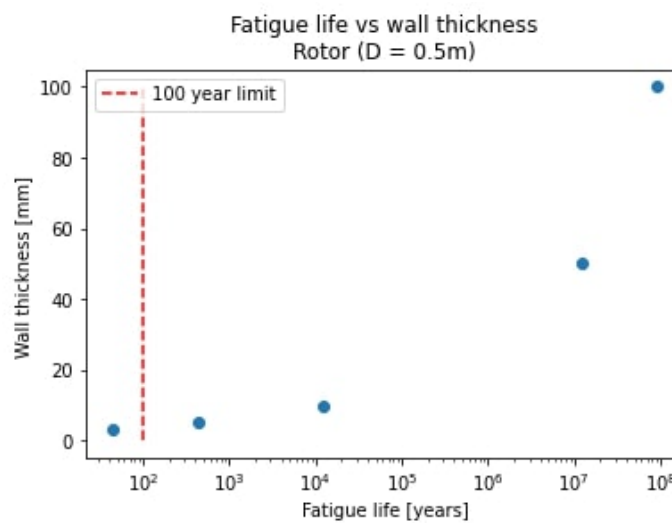


Figure E.19

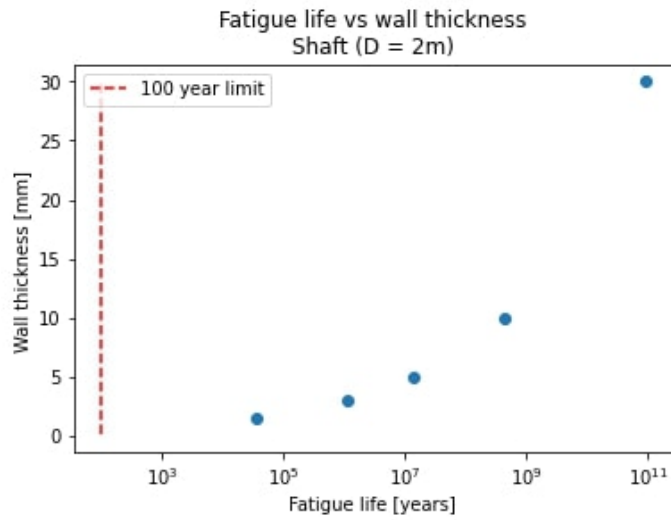


Figure E.20

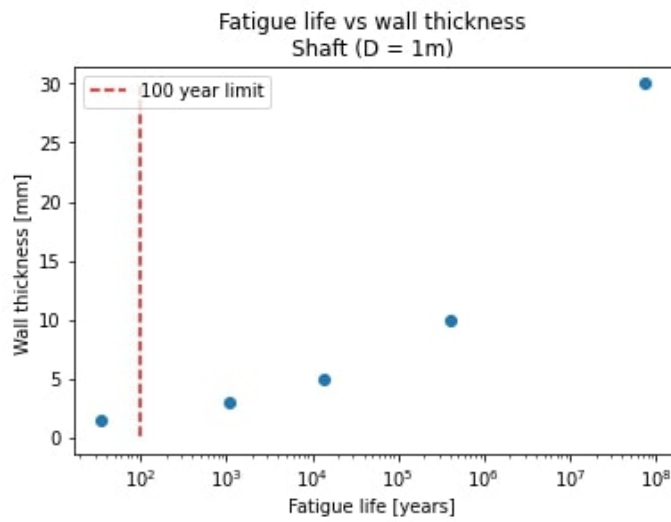


Figure E.21

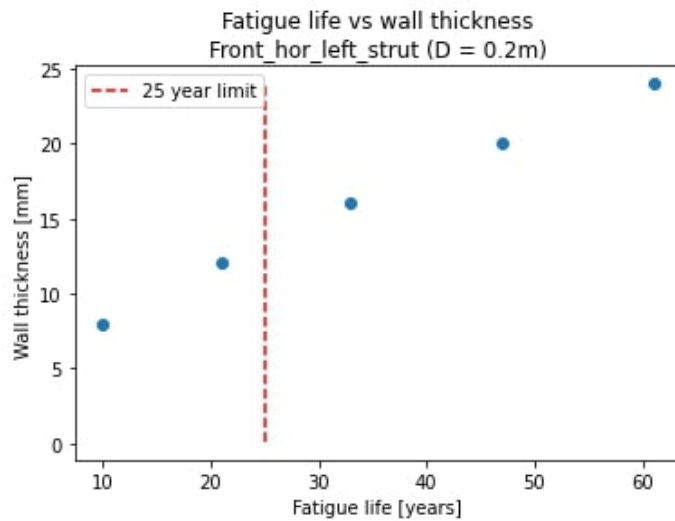


Figure E.22

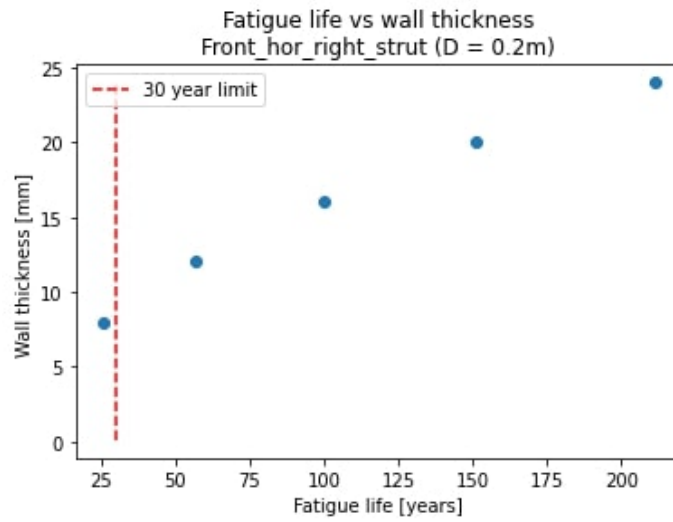


Figure E.23

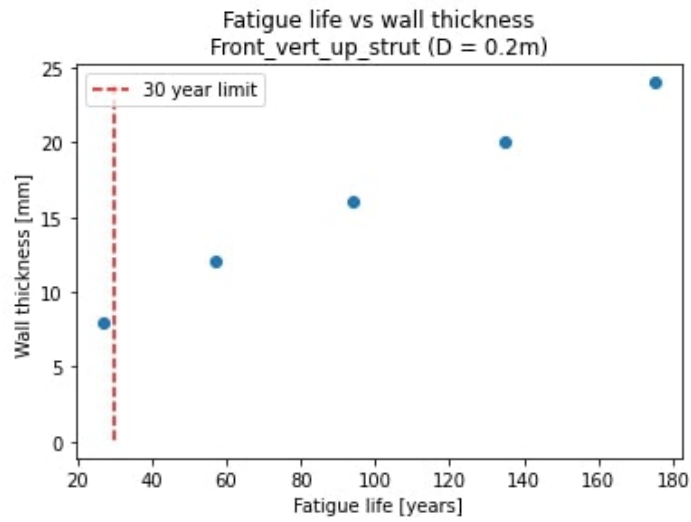


Figure E.24

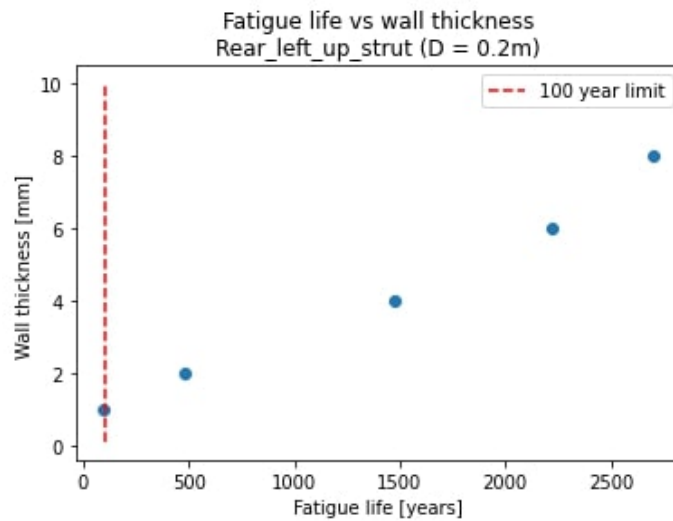


Figure E.25

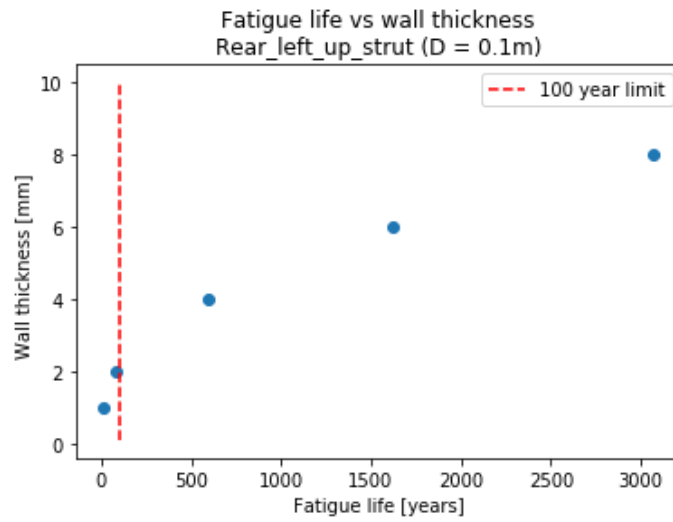


Figure E.26

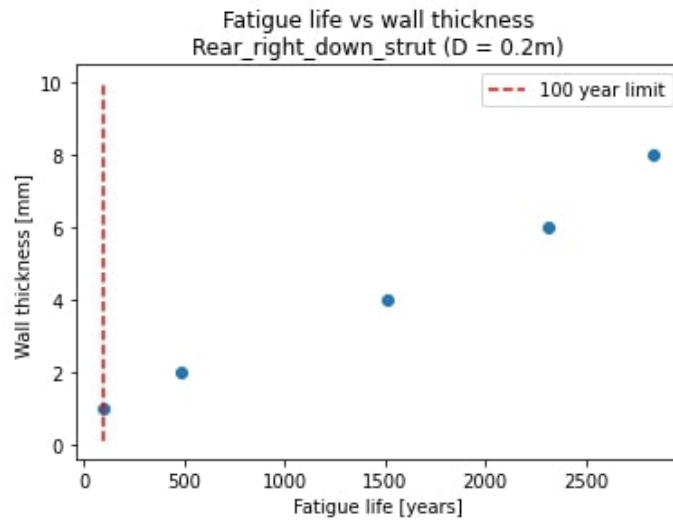


Figure E.27

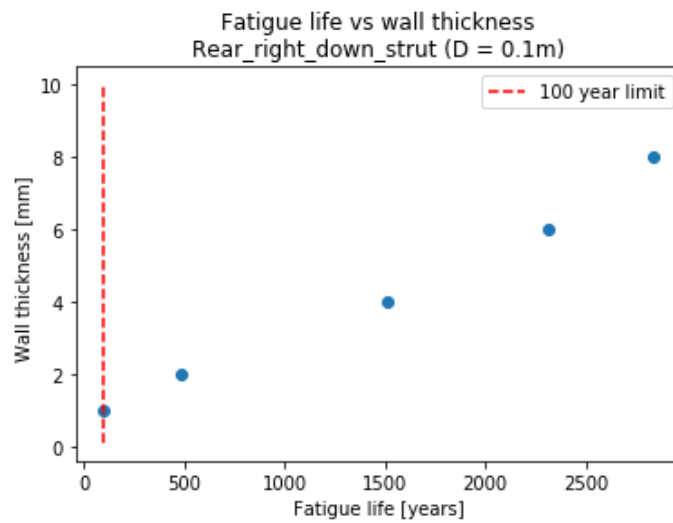


Figure E.28

F. Stress histories from Fatigue Life Assessment

2-Bladed Structure

Front hor left strut 1

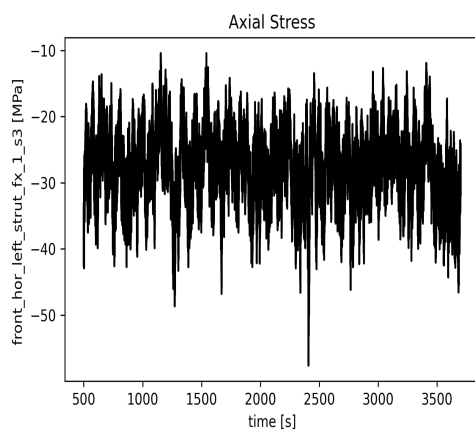


Figure F.1

4-Bladed Structure

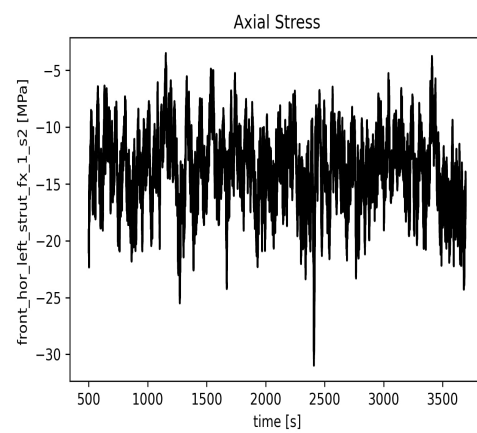


Figure F.2

Front hor left strut 2

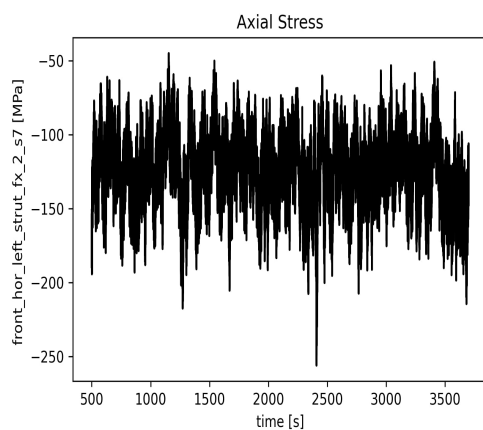


Figure F.3

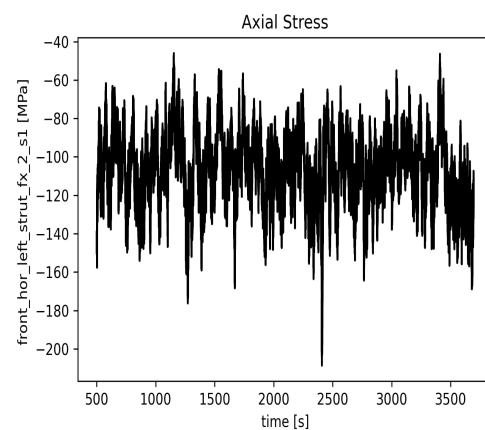


Figure F.4

Front hor right strut 1

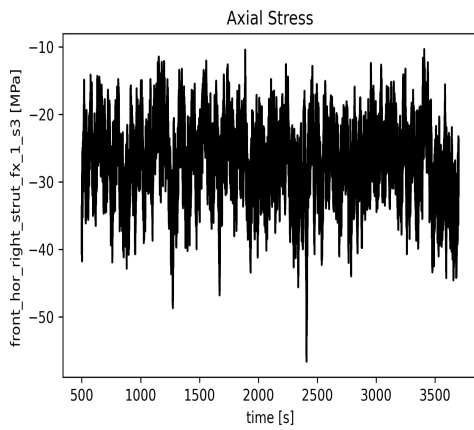


Figure F.5

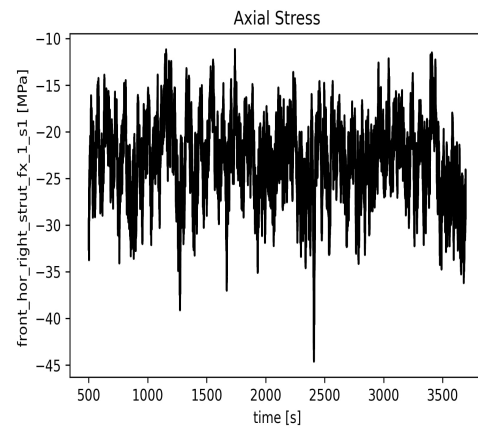


Figure F.6

Front hor right strut 2

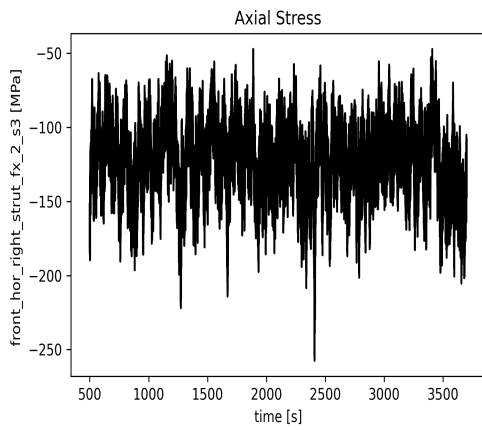


Figure F.7

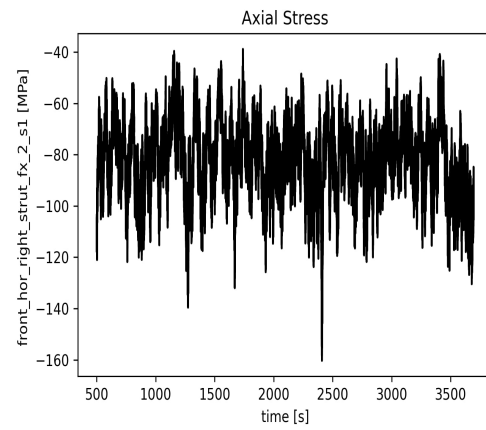


Figure F.8

Front vert down strut 1

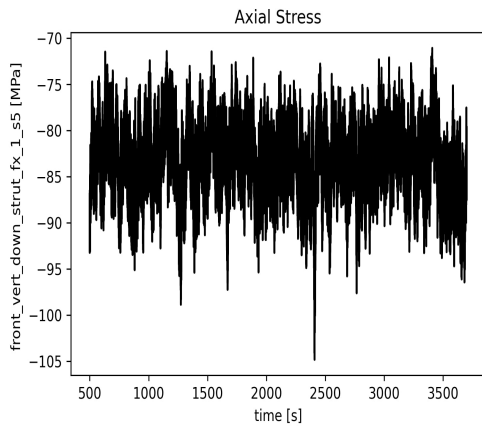


Figure F.9

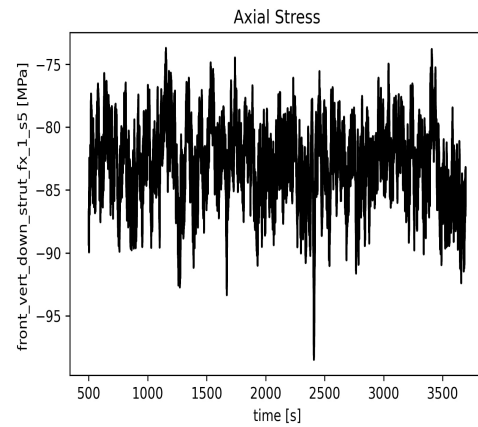


Figure F.10

Front vert down strut 2

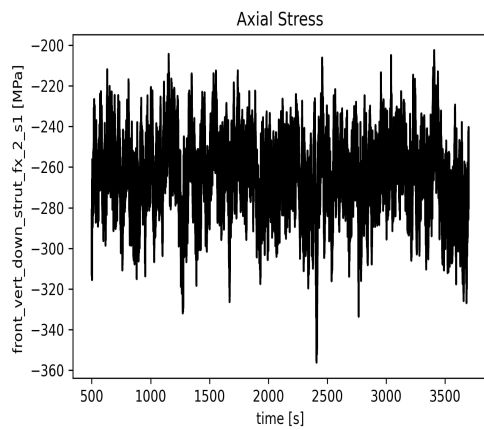


Figure F.11

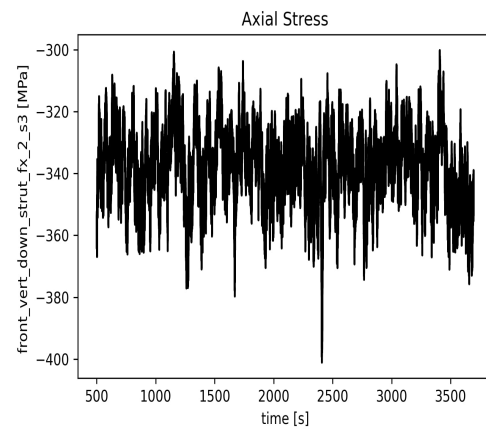


Figure F.12

Front vert up strut 1

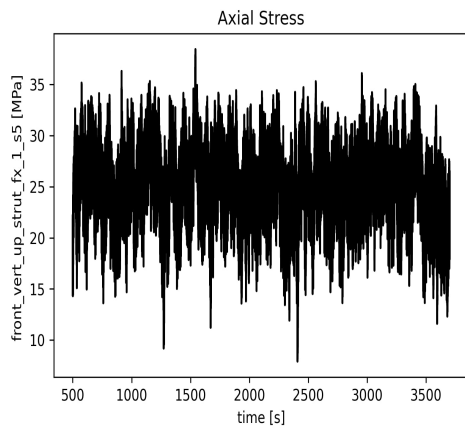


Figure F.13

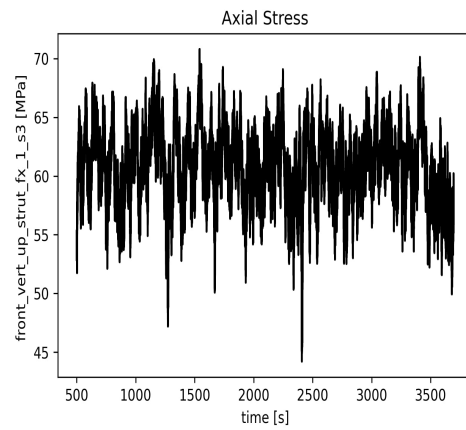


Figure F.14

Front vert up strut 2

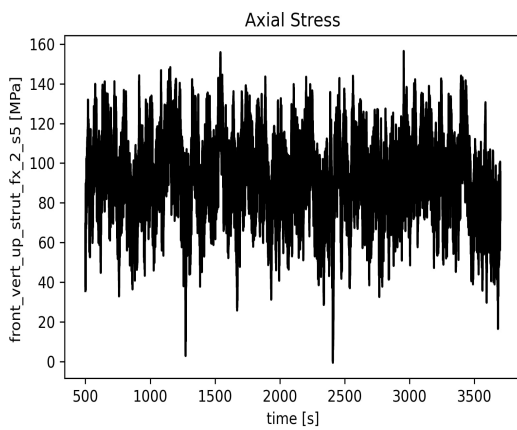


Figure F.15

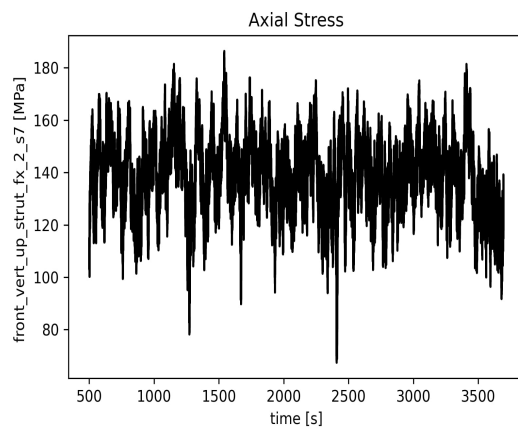


Figure F.16

Rear left down strut 1

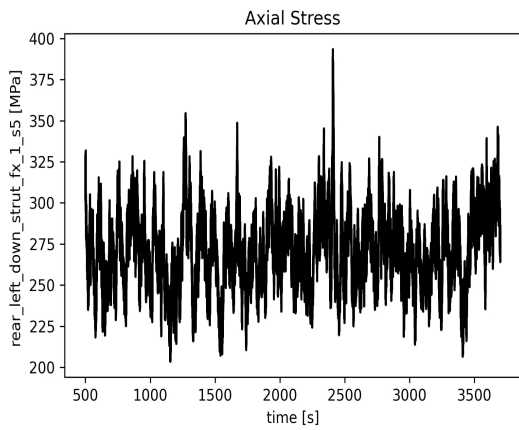


Figure F.17

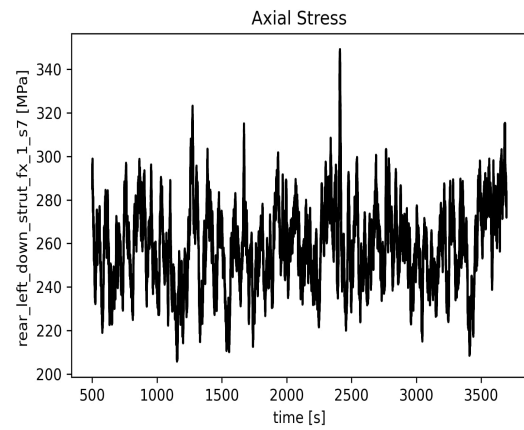


Figure F.18

Rear left down strut 2

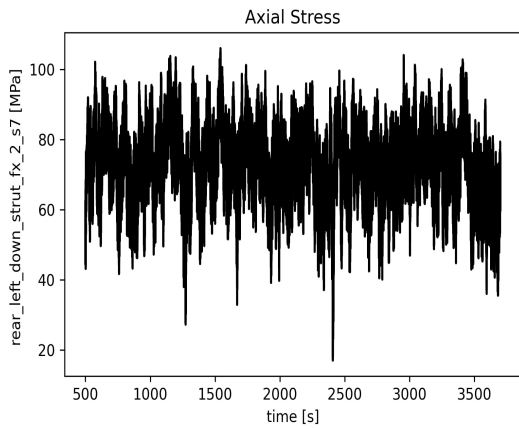


Figure F.19

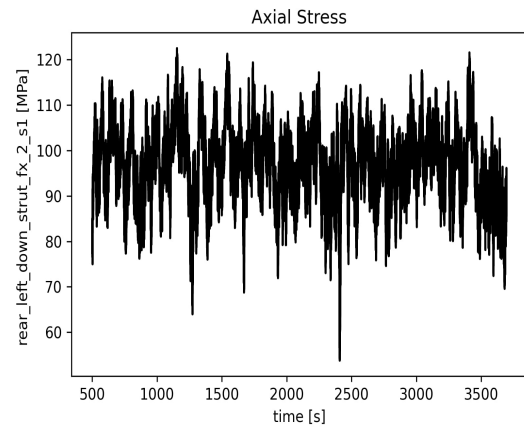


Figure F.20

Rear left up strut 1

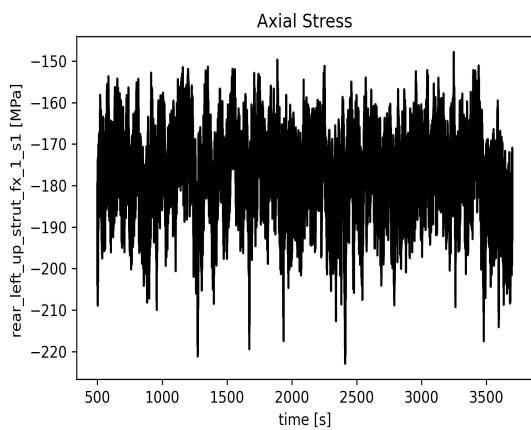


Figure F.21

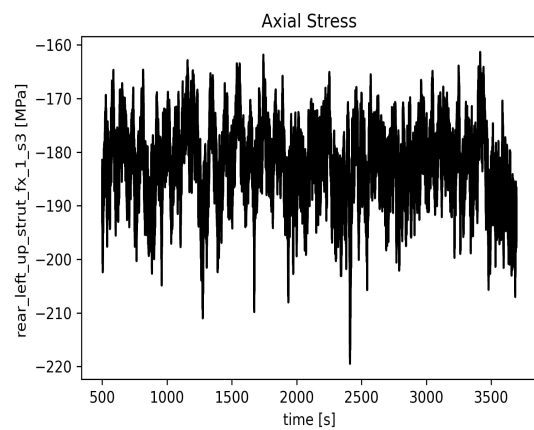


Figure F.22

Rear left up strut 2

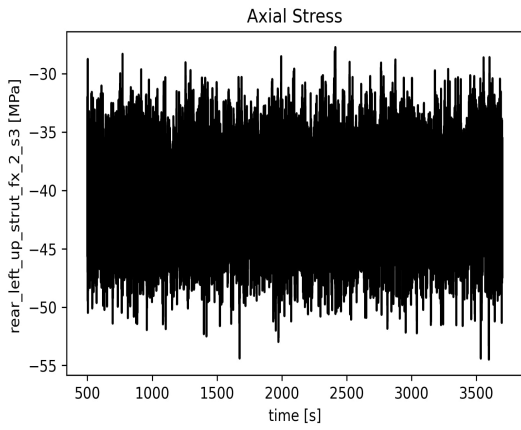


Figure F.23

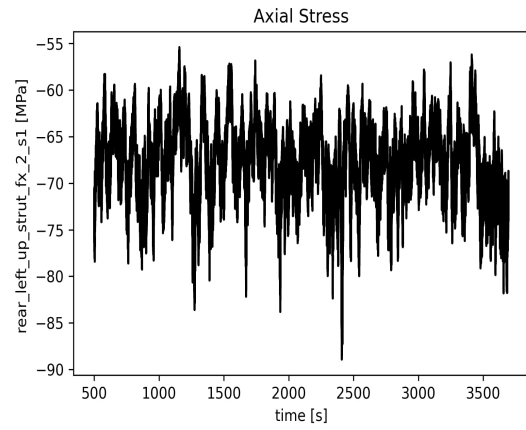


Figure F.24

Rear right down strut 1

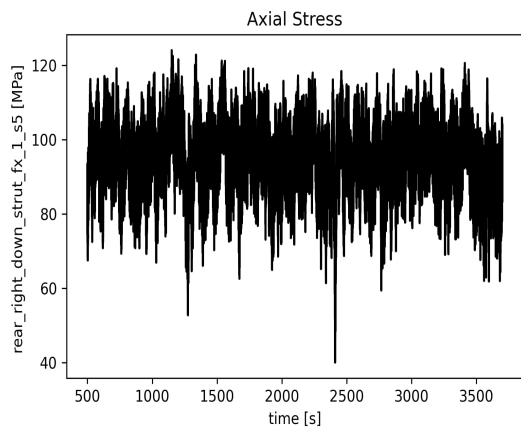


Figure F.25

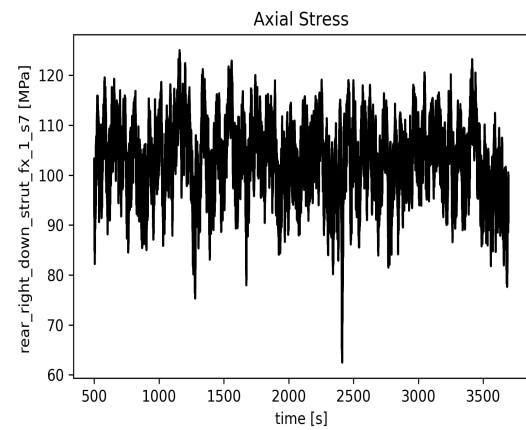


Figure F.26

Rear right down strut 2

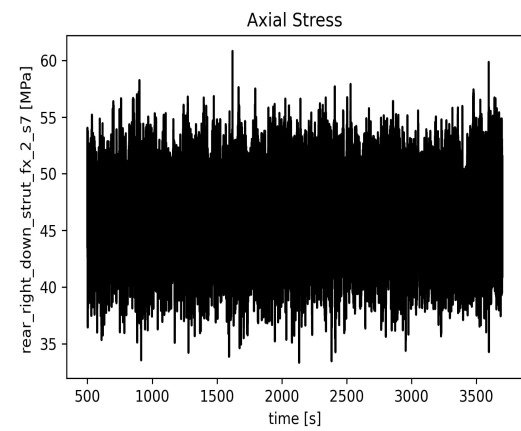


Figure F.27

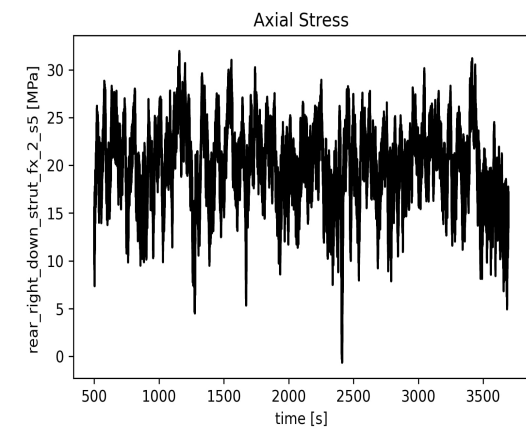


Figure F.28

Rear right up 1

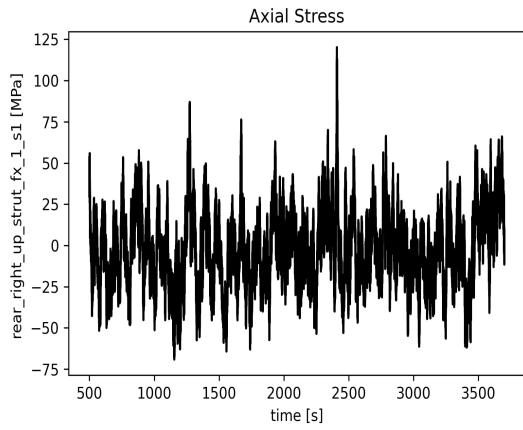


Figure F.29

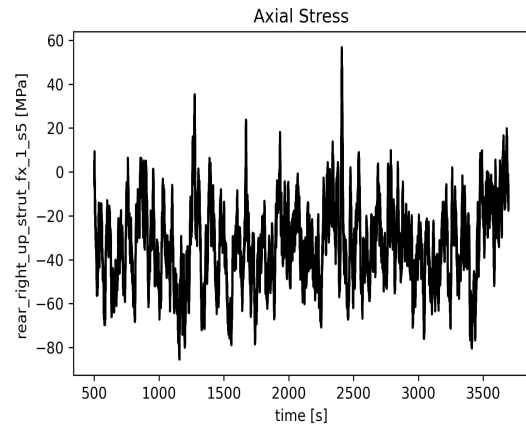


Figure F.30

Rear right up 2

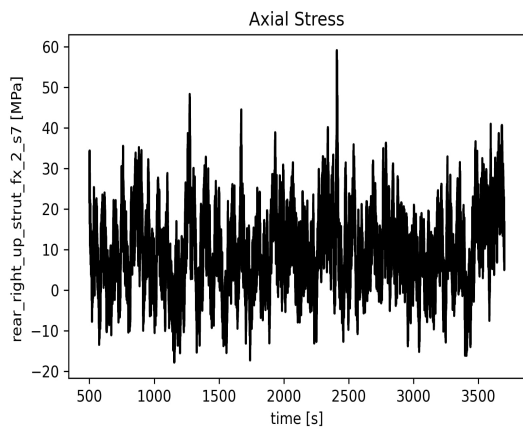


Figure F.31

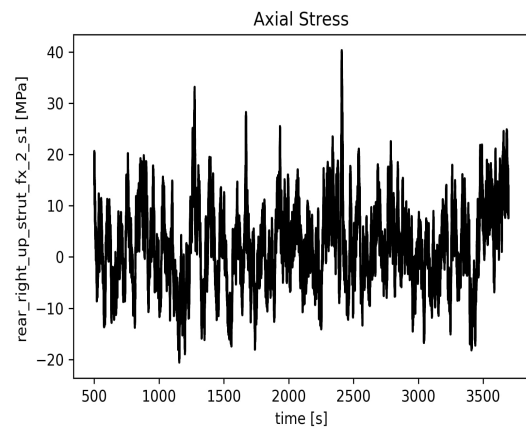


Figure F.32

Rear vert down strut

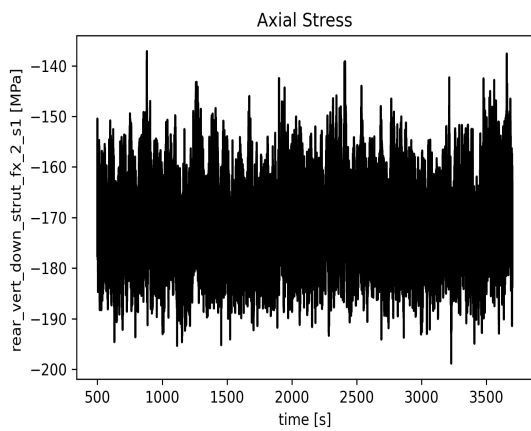


Figure F.33

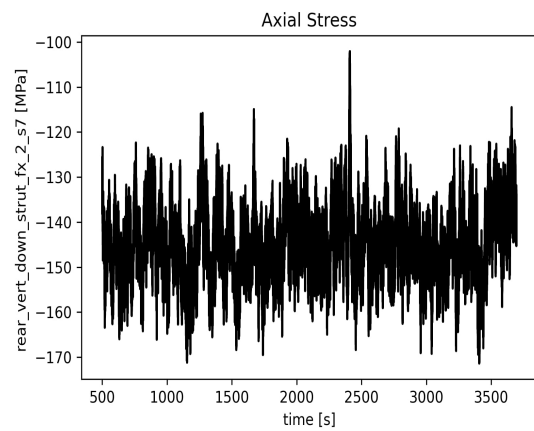


Figure F.34

Rear vert up strut

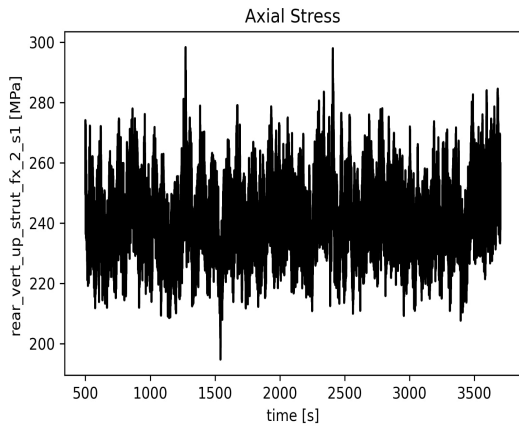


Figure F.35

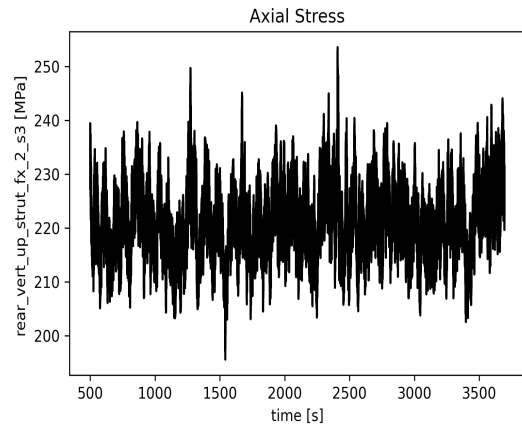


Figure F.36

Shaft

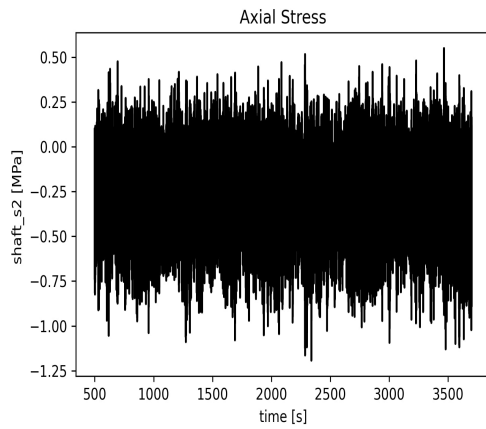


Figure F.37

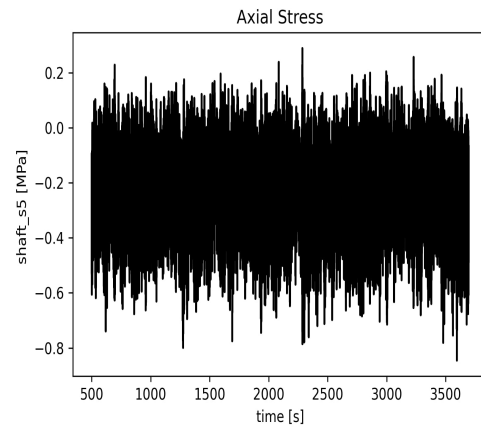


Figure F.38

Blade root

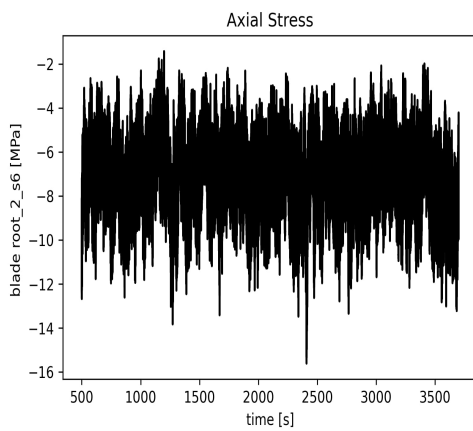


Figure F.39

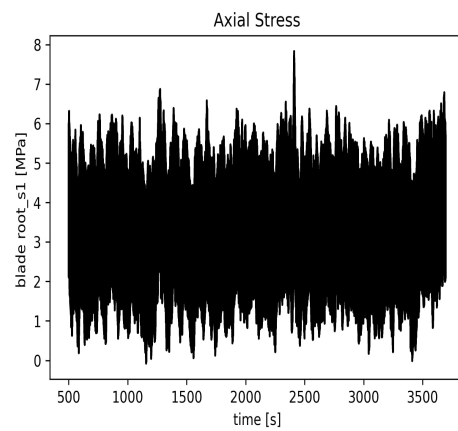


Figure F.40

Rotor

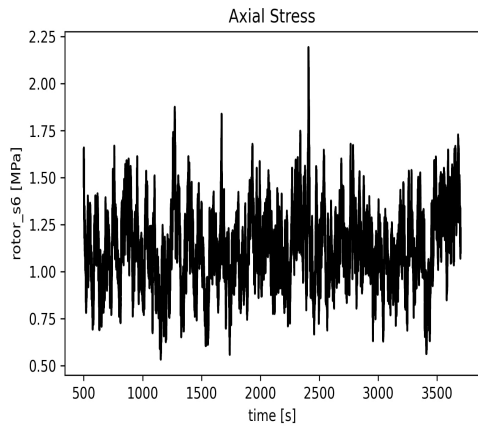


Figure F.41

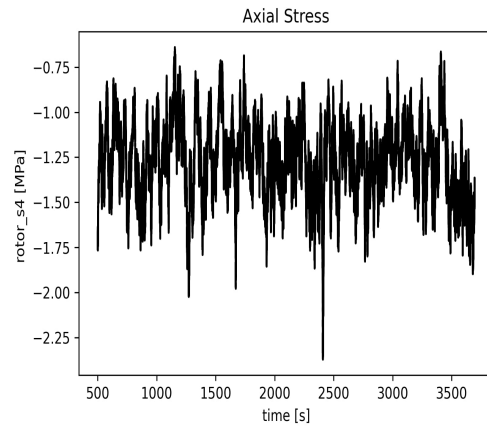


Figure F.42

G. Python Routine for Assessing Fatigue Life in the Structure and element stress history generation

```
1  ## -*- coding: utf-8 -*-
2  """
3  Created on Thu Apr 29 13:19:16 2021
4
5  @author: Vetle Birkeland Aass
6  """
7
8  import numpy as np
9  from mik_fatigue_funcs import turningpoints, turningpoints_steffen,
    fatiguedamage_twoslope
10 from atte_punkter import cyl_beam_stresses
11
12
13 from pylab import *
14 from numpy import loadtxt
15 from scipy import interpolate
16
17
18 ## STANDARD INPUT for tubular joints
19 # SN-curve T "air"
20 th = 32E-3 # plate thickness base material (m)
21 ## SN-curve parameters from DNV-GL-C203:
22 m1 = 3.0 # slope 1
23 loga1 = 12.479 # intercept 1
24 m2 = 5.0 # slope high cycle region
25 loga2 = 16.132 # intercept high-cycle region
26 Nlim = 1.0E7 # limit for high-cycle region
27 tref=32E-3 # reference thickness (32E-3 for tubular joints)
28 k=0.25 # thickness exponent
29 DFF = 1 #Design Fatigue Factor
30
31 ## STANDARD INPUT for unwelded tubular member
32 th_st = 25E-3 # plate thickness base material (m)
33 ## SN-curve parameters from DNV-GL-C203:
34 m1_st = 4.0 # slope 1
```

```
35 loga1_st = 15.117 # intercept 1
36 m2_st = 5.0 # slope high cycle region
37 loga2_st = 17.146 # intercept high-cycle region
38 Nlim_st = 1.0E7 # limit for high-cycle region
39 tref= 25E-3 # reference thickness (25E-3 for tubular joints)
40 k_st=0.25 # thickness exponent
41 DFF_st = 1 #Design Fatigue Factor
42
43
44
45 #struts
46 d      = 0.2
47 twall = 0.008
48
49 #shaft
50 D_shaft = 2
51 T_shaft = 0.002
52
53 #Blade root
54 D = 1.2 # element midpoint
55 TWALL = 0.0028
56
57 #Rotor
58 D_Rotor = 1 #[m]
59 Twall_Rotor = 0.008 #[m]
60
61
62 #importing data
63 a  = loadtxt('sensors.txt',skiprows=50000,dtype='float',delimiter=';') #
        skip 500 s
64 t      = a[:,0];
65
66
67 #Joint types and their corresponding stress concentration factors
68
69 #Type 1 joint
70 SCF_AC_1 = 3.73          #stress concentration factor for joint type 1,
        brace crown
71 SCF_AS_1 = 1.99          #stress concentration factor for joint type 1,
        brace saddle
72 SCF_IPB_1 = 1.36          #stress concentration factor for joint type 1, in
        plane bending
73 SCF_OPB_1 = 1           #stress concentration factor for joint type 1, out
        of plane bending
74
75 #Type 2 joint
76 SCF_AC_2 = 5.73          #stress concentration factor for joint type 2, brace
        crown
77 SCF_AS_2 = 15.21          #stress concentration factor for joint type 2, brace
```

```

    saddle
78 SCF_IPB_2 = 3.93      #stress concentration factor for joint type 2, in
    plane bending
79 SCF_OPB_2 = 5.90      #stress concentration factor for joint type 2, out of
    plane bending
80
81 #Type 3 joint
82 SCF_AC_3 = 26.14      #stress concentration factor for joint type 3,
    brace crown
83 SCF_AS_3 = 4.27      #stress concentration factor for joint type 3,
    brace saddle
84 SCF_IPB_3 = 2.46      #stress concentration factor for joint type 3, in
    plane bending
85 SCF_OPB_3 = 4.27      #stress concentration factor for joint type 3, out
    of plane bending
86
87
88 #Type 4 joint
89 SCF_AC_4 = 7.72      #stress concentration factor for joint type 4, brace
    crown
90 SCF_AS_4 = 7.72      #stress concentration factor for joint type 4, brace
    saddle
91 SCF_IPB_4 = 3.91      #stress concentration factor for joint type 4, in
    plane bending
92 SCF_OPB_4 = 6.03      #stress concentration factor for joint type 4, out of
    plane bending
93
94
95 #Type 5 joint
96
97
98
99
100 #

```

```

101
102 #front_hor_left_strut_x1
103 #Joint type 1
104 #saddle in z-dir
105 front_hor_left_strut_fx_1 = a[:,22]*1.e3;
106 front_hor_left_strut_fy_1 = a[:,102]*1.e3;
107 front_hor_left_strut_fz_1 = a[:,103]*1.e3;
108 mx_fhls_1 = a[:,23]*1.e3;
109 my_fhls_1 = a[:,24]*1.e3;
110 mz_fhls_1 = a[:,25]*1.e3;
111
112 [sax ,sbendy ,sbendz ,s1 ,s2 ,s3 ,s4 ,s5 ,s6 ,s7 ,s8 ,ssh1 ,ssh2 ,ssh3 ,ssh4 ,ssh5 ,ssh6 ,
    ssh7 ,ssh8 ,svm1 ,svm2 ,svm3 ,svm4 ,svm5 ,svm6 ,svm7 ,svm8 ,sx ,sy ,sz ] = \

```



```

113     cyl_beam_stresses(t, front_hor_left_strut_fx_1,
114                       front_hor_left_strut_fy_1, front_hor_left_strut_fz_1, mx_fhls_1,
115                       my_fhls_1, mz_fhls_1, d, twall, SCF_AS_1, SCF_AC_1, SCF_OPB_1, SCF_IPB_1)
116
117 figure()
118 plot(t, s1/1.e6, 'k-', label='')
119 xlabel('time [s]')
120 ylabel('front_hor_left_strut_fx_1_s1 [MPa]')
121 title('Axial Stress')
122 #axis([-16., 90., -1.5, 1.7])
123 #legend(loc='best')
124 savefig('front_hor_left_strut_fx_1_s1', dpi=300)
125
126 damage = fatiguedamage_twoslope(t, s2, m1, loga1, m2, loga2, Nlim, th, tref=25E-3,
127                                 k=0.25)
128 print('Partial fatigue damage for front_hor_left_strut_fx_1_s2 calculated
129        by script: %.5e (includes residual cycles)' %damage)
130 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
131        (damage*60*60*24*365)))
132
133 figure()
134 plot(t, s2/1.e6, 'k-', label='')
135 xlabel('time [s]')
136 ylabel('front_hor_left_strut_fx_1_s2 [MPa]')
137 title('Axial Stress')
138 #axis([-16., 90., -1.5, 1.7])
139 #legend(loc='best')
140 savefig('front_hor_left_strut_fx_1_s2', dpi=300)
141
142 damage = fatiguedamage_twoslope(t, s3, m1, loga1, m2, loga2, Nlim, th, tref=25E-3,
143                                 k=0.25)
144 print('Partial fatigue damage for front_hor_left_strut_fx_1_s3 calculated
145        by script: %.5e (includes residual cycles)' %damage)
146 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
147        (damage*60*60*24*365)))
148
149 figure()
150 plot(t, s3/1.e6, 'k-', label='')
151 xlabel('time [s]')
152 ylabel('front_hor_left_strut_fx_1_s3 [MPa]')
153 title('Axial Stress')
154 #axis([-16., 90., -1.5, 1.7])
155 #legend(loc='best')
156 savefig('front_hor_left_strut_fx_1_s3', dpi=300)
157
158 damage = fatiguedamage_twoslope(t, s4, m1, loga1, m2, loga2, Nlim, th, tref=25E-3,
159                                 k=0.25)
160 print('Partial fatigue damage for front_hor_left_strut_fx_1_s4 calculated
161        by script: %.5e (includes residual cycles)' %damage)

```

```

152 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365))
153
154 figure()
155 plot(t,s4/1.e6,'k-',label='')
156 xlabel('time [s]')
157 ylabel('front_hor_left_strut_fx_1_s4 [MPa] ')
158 title('Axial Stress')
159 #axis([-16.,90.,-1.5,1.7])
160 #legend(loc='best')
161 savefig('front_hor_left_strut_fx_1_s4',dpi=300)
162
163 damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
164 print('Partial fatigue damage for front_hor_left_strut_fx_1_s5 calculated
      by script: %.5e (includes residual cycles)' %damage)
165 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365))
166
167 figure()
168 plot(t,s5/1.e6,'k-',label='')
169 xlabel('time [s]')
170 ylabel('front_hor_left_strut_fx_1_s5 [MPa] ')
171 title('Axial Stress')
172 #axis([-16.,90.,-1.5,1.7])
173 #legend(loc='best')
174 savefig('front_hor_left_strut_fx_1_s5',dpi=300)
175
176 damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
177 print('Partial fatigue damage for front_hor_left_strut_fx_1_s6 calculated
      by script: %.5e (includes residual cycles)' %damage)
178 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365))
179
180 figure()
181 plot(t,s6/1.e6,'k-',label='')
182 xlabel('time [s]')
183 ylabel('front_hor_left_strut_fx_1_s6 [MPa] ')
184 title('Axial Stress')
185 #axis([-16.,90.,-1.5,1.7])
186 #legend(loc='best')
187 savefig('front_hor_left_strut_fx_1_s6',dpi=300)
188
189 damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
190 print('Partial fatigue damage for front_hor_left_strut_fx_1_s7 calculated
      by script: %.5e (includes residual cycles)' %damage)
191 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/

```

```

        damage*60*60*24*365)))
192
193 figure()
194 plot(t,s7/1.e6,'k-',label='')
195 xlabel('time [s]')
196 ylabel('front_hor_left_strut_fx_1_s7 [MPa] ')
197 title('Axial Stress')
198 #axis([-16.,90.,-1.5,1.7])
199 #legend(loc='best')
200 savefig('front_hor_left_strut_fx_1_s7',dpi=300)
201
202 damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
203 print('Partial fatigue damage for front_hor_left_strut_fx_1_s8 calculated
        by script: %.5e (includes residual cycles)' %damage)
204 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365)))
205
206 figure()
207 plot(t,s8/1.e6,'k-',label='')
208 xlabel('time [s]')
209 ylabel('front_hor_left_strut_fx_1_s8 [MPa] ')
210 title('Axial Stress')
211 #axis([-16.,90.,-1.5,1.7])
212 #legend(loc='best')
213 savefig('front_hor_left_strut_fx_1_s8',dpi=300)
214
215 #

```

```

216
217 #front_hor_left_strut_x2
218 #Joint type 2
219 #crown in z-dir
220 front_hor_left_strut_fx_2 = a[:,26]*1.e3;
221 front_hor_left_strut_fy_2 = a[:,104]*1.e3;
222 front_hor_left_strut_fz_2 = a[:,105]*1.e3;
223 mx_fhls_2 = a[:,27]*1.e3;
224 my_fhls_2 = a[:,28]*1.e3;
225 mz_fhls_2 = a[:,29]*1.e3;
226
227 [sax,sbendy,sbendz,s1,s2,s3,s4,s5,s6,s7,s8,ssh1,ssh2,ssh3,ssh4,ssh5,ssh6,
        ssh7,ssh8,svm1,svm2,svm3,svm4,svm5,svm6,svm7,svm8,sx,sy,sz] = \
228     cyl_beam_stresses(t,front_hor_left_strut_fx_2,
        front_hor_left_strut_fy_2,front_hor_left_strut_fz_2,mx_fhls_2,
        my_fhls_2,mz_fhls_2,d,twall,SCF_AC_2,SCF_AS_2,SCF_IPB_2,SCF_OPB_2)
229
230 damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)

```

```

231 print('Partial fatigue damage front_hor_left_strut_fx_2_s1 calculated by
      script: %.5e (includes residual cycles)' %damage)
232 print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/
      damage*60*60*24*DFF))
233 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFF))
234
235 figure()
236 plot(t,s1/1.e6,'k-',label='')
237 xlabel('time [s]')
238 ylabel('front_hor_left_strut_fx_2_s1 [MPa] ')
239 title('Axial Stress')
240 #axis([-16.,90.,-1.5,1.7])
241 #legend(loc='best')
242 savefig('front_hor_left_strut_fx_2_s1',dpi=300)
243
244 damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
245 print('Partial fatigue damage for front_hor_left_strut_fx_2_s2 calculated
      by script: %.5e (includes residual cycles)' %damage)
246 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFF))
247
248 figure()
249 plot(t,s2/1.e6,'k-',label='')
250 xlabel('time [s]')
251 ylabel('front_hor_left_strut_fx_2_s2 [MPa] ')
252 title('Axial Stress')
253 #axis([-16.,90.,-1.5,1.7])
254 #legend(loc='best')
255 savefig('front_hor_left_strut_fx_2_s2',dpi=300)
256
257 damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
258 print('Partial fatigue damage for front_hor_left_strut_fx_2_s3 calculated
      by script: %.5e (includes residual cycles)' %damage)
259 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFF))
260
261 figure()
262 plot(t,s3/1.e6,'k-',label='')
263 xlabel('time [s]')
264 ylabel('front_hor_left_strut_fx_2_s3 [MPa] ')
265 title('Axial Stress')
266 #axis([-16.,90.,-1.5,1.7])
267 #legend(loc='best')
268 savefig('front_hor_left_strut_fx_2_s3',dpi=300)
269
270 damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,

```

```

    k=0.25)
271 print('Partial fatigue damage for front_hor_left_strut_fx_2_s4 calculated
        by script: %.5e (includes residual cycles)' %damage)
272 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF))
273
274 figure()
275 plot(t,s4/1.e6,'k-',label='')
276 xlabel('time [s]')
277 ylabel('front_hor_left_strut_fx_2_s4 [MPa] ')
278 title('Axial Stress')
279 #axis([-16.,90.,-1.5,1.7])
280 #legend(loc='best')
281 savefig('front_hor_left_strut_fx_2_s4',dpi=300)
282
283 damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
284 print('Partial fatigue damage for front_hor_left_strut_fx_2_s5 calculated
        by script: %.5e (includes residual cycles)' %damage)
285 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF))
286
287 figure()
288 plot(t,s5/1.e6,'k-',label='')
289 xlabel('time [s]')
290 ylabel('front_hor_left_strut_fx_2_s5 [MPa] ')
291 title('Axial Stress')
292 #axis([-16.,90.,-1.5,1.7])
293 #legend(loc='best')
294 savefig('front_hor_left_strut_fx_2_s5',dpi=300)
295
296 damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
297 print('Partial fatigue damage for front_hor_left_strut_fx_2_s6 calculated
        by script: %.5e (includes residual cycles)' %damage)
298 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF))
299
300 figure()
301 plot(t,s6/1.e6,'k-',label='')
302 xlabel('time [s]')
303 ylabel('front_hor_left_strut_fx_2_s6 [MPa] ')
304 title('Axial Stress')
305 #axis([-16.,90.,-1.5,1.7])
306 #legend(loc='best')
307 savefig('front_hor_left_strut_fx_2_s6',dpi=300)
308
309 damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)

```

```

310 print('Partial fatigue damage for front_hor_left_strut_fx_2_s7 calculated
      by script: %.5e (includes residual cycles)' %damage)
311 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
      damage*60*60*24*365*DFE)))
312
313 figure()
314 plot(t,s7/1.e6,'k-',label='')
315 xlabel('time [s]')
316 ylabel('front_hor_left_strut_fx_2_s7 [MPa]')
317 title('Axial Stress')
318 #axis([-16.,90.,-1.5,1.7])
319 #legend(loc='best')
320 savefig('front_hor_left_strut_fx_2_s7',dpi=300)
321
322 damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
323 print('Partial fatigue damage for front_hor_left_strut_fx_2_s8 calculated
      by script: %.5e (includes residual cycles)' %damage)
324 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
      damage*60*60*24*365*DFE)))
325
326 figure()
327 plot(t,s8/1.e6,'k-',label='')
328 xlabel('time [s]')
329 ylabel('front_hor_left_strut_fx_2_s8 [MPa]')
330 title('Axial Stress')
331 #axis([-16.,90.,-1.5,1.7])
332 #legend(loc='best')
333 savefig('front_hor_left_strut_fx_2_s8',dpi=300)
334
335 #

```

```

336
337 #front_hor_right_strut_x1
338 #Joint type 1
339 #saddle in z-dir
340 front_hor_right_strut_fx_1 = a[:,30]*1.e3;
341 front_hor_right_strut_fy_1 = a[:,106]*1.e3;
342 front_hor_right_strut_fz_1 = a[:,107]*1.e3;
343 mx_fhrs_1 = a[:,31]*1.e3;
344 my_fhrs_1 = a[:,32]*1.e3;
345 mz_fhrs_1 = a[:,33]*1.e3;
346
347
348 [sax,sbendy,sbendz,s1,s2,s3,s4,s5,s6,s7,s8,ssh1,ssh2,ssh3,ssh4,ssh5,ssh6,
      ssh7,ssh8,svm1,svm2,svm3,svm4,svm5,svm6,svm7,svm8,sx,sy,sz] = \
349     cyl_beam_stresses(t,front_hor_right_strut_fx_1,
      front_hor_right_strut_fy_1,front_hor_right_strut_fz_1,mx_fhrs_1,

```

```
my_fhrs_1 , mz_fhrs_1 , d , twall , SCF_AS_1 , SCF_AC_1 , SCF_OPB_1 , SCF_IPB_1)
350
351 damage = fatiguedamage_twoslope(t , s1 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
    k=0.25)
352 print('Partial fatigue damage for front_hor_right_strut_fx_1_s1 calculated
    by script: %.5e (includes residual cycles)' %damage)
353 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
354
355 figure()
356 plot(t , s1/1.e6 , 'k-' , label='')
357 xlabel('time [s]')
358 ylabel('front_hor_right_strut_fx_1_s1 [MPa] ')
359 title('Axial Stress')
360 #axis([-16. , 90. , -1.5 , 1.7])
361 #legend(loc='best')
362 savefig('front_hor_right_strut_fx_1_s1' , dpi=300)
363
364 damage = fatiguedamage_twoslope(t , s2 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
    k=0.25)
365 print('Partial fatigue damage for front_hor_right_strut_fx_1_s2 calculated
    by script: %.5e (includes residual cycles)' %damage)
366 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
367
368 figure()
369 plot(t , s2/1.e6 , 'k-' , label='')
370 xlabel('time [s]')
371 ylabel('front_hor_right_strut_fx_1_s2 [MPa] ')
372 title('Axial Stress')
373 #axis([-16. , 90. , -1.5 , 1.7])
374 #legend(loc='best')
375 savefig('front_hor_right_strut_fx_1_s2' , dpi=300)
376
377 damage = fatiguedamage_twoslope(t , s3 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
    k=0.25)
378 print('Partial fatigue damage for front_hor_right_strut_fx_1_s3 calculated
    by script: %.5e (includes residual cycles)' %damage)
379 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
380
381 figure()
382 plot(t , s3/1.e6 , 'k-' , label='')
383 xlabel('time [s]')
384 ylabel('front_hor_right_strut_fx_1_s3 [MPa] ')
385 title('Axial Stress')
386 #axis([-16. , 90. , -1.5 , 1.7])
387 #legend(loc='best')
388 savefig('front_hor_right_strut_fx_1_s3' , dpi=300)
```

```

389
390 damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
391 print('Partial fatigue damage for front_hor_right_strut_fx_1_s4 calculated
      by script: %.5e (includes residual cycles)' %damage)
392 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFF))
393
394 figure()
395 plot(t,s4/1.e6,'k-',label='')
396 xlabel('time [s]')
397 ylabel('front_hor_right_strut_fx_1_s4 [MPa] ')
398 title('Axial Stress')
399 #axis([-16.,90.,-1.5,1.7])
400 #legend(loc='best')
401 savefig('front_hor_right_strut_fx_1_s4',dpi=300)
402
403 damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
404 print('Partial fatigue damage for front_hor_right_strut_fx_1_s5 calculated
      by script: %.5e (includes residual cycles)' %damage)
405 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFF))
406
407 figure()
408 plot(t,s5/1.e6,'k-',label='')
409 xlabel('time [s]')
410 ylabel('front_hor_right_strut_fx_1_s5 [MPa] ')
411 title('Axial Stress')
412 #axis([-16.,90.,-1.5,1.7])
413 #legend(loc='best')
414 savefig('front_hor_right_strut_fx_1_s5',dpi=300)
415
416 damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
417 print('Partial fatigue damage for front_hor_right_strut_fx_1_s6 calculated
      by script: %.5e (includes residual cycles)' %damage)
418 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFF))
419
420 figure()
421 plot(t,s6/1.e6,'k-',label='')
422 xlabel('time [s]')
423 ylabel('front_hor_right_strut_fx_1_s6 [MPa] ')
424 title('Axial Stress')
425 #axis([-16.,90.,-1.5,1.7])
426 #legend(loc='best')
427 savefig('front_hor_right_strut_fx_1_s6',dpi=300)
428

```



```

429 damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
430 print('Partial fatigue damage for front_hor_right_strut_fx_1_s7 calculated
    by script: %.5e (includes residual cycles)' %damage)
431 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
432
433 figure()
434 plot(t,s7/1.e6,'k-',label='')
435 xlabel('time [s]')
436 ylabel('Sfront_hor_right_strut_fx_1_s7 [MPa] ')
437 title('Axial Stress')
438 #axis([-16.,90.,-1.5,1.7])
439 #legend(loc='best')
440 savefig('front_hor_right_strut_fx_1_s7',dpi=300)
441
442 damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
443 print('Partial fatigue damage for front_hor_right_strut_fx_1_s8 calculated
    by script: %.5e (includes residual cycles)' %damage)
444 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
445
446 figure()
447 plot(t,s8/1.e6,'k-',label='')
448 xlabel('time [s]')
449 ylabel('front_hor_right_strut_fx_1_s8 [MPa] ')
450 title('Axial Stress')
451 #axis([-16.,90.,-1.5,1.7])
452 #legend(loc='best')
453 savefig('front_hor_right_strut_fx_1_s8',dpi=300)
454
455 #

```

```

456
457
458 #front_hor_right_strut_x2
459 #Joint type 2
460 #crown in z-dir
461 front_hor_right_strut_fx_2 = a[:,34]*1.e3;
462 front_hor_right_strut_fy_2 = a[:,108]*1.e3;
463 front_hor_right_strut_fz_2 = a[:,109]*1.e3;
464 mx_fhrs_2 = a[:,35]*1.e3;
465 my_fhrs_2 = a[:,36]*1.e3;
466 mz_fhrs_2 = a[:,37]*1.e3;
467
468 [sax,sbendy,sbendz,s1,s2,s3,s4,s5,s6,s7,s8,ssh1,ssh2,ssh3,ssh4,ssh5,ssh6,
    ssh7,ssh8,svm1,svm2,svm3,svm4,svm5,svm6,svm7,svm8,sx,sy,sz] = \

```

```
469     cyl_beam_stresses(t, front_hor_right_strut_fx_1 ,
        front_hor_right_strut_fy_2 , front_hor_right_strut_fz_2 , mx_fhrs_2 ,
        my_fhrs_2 , mz_fhrs_2 , d , twall , SCF_AC_2 , SCF_AS_2 , SCF_IPB_2 , SCF_OPB_2)
470
471 damage = fatiguedamage_twoslope(t , s1 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
        k=0.25)
472 print('Partial fatigue damage for front_hor_right_strut_fx_2_s1 calculated
        by script: %.5e (includes residual cycles)' %damage)
473 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365))
474
475 figure()
476 plot(t , s1 /1.e6 , 'k-' , label='')
477 xlabel('time [s]')
478 ylabel('front_hor_right_strut_fx_2_s1 [MPa] ')
479 title('Axial Stress')
480 #axis([-16., 90., -1.5, 1.7])
481 #legend(loc='best')
482 savefig('front_hor_right_strut_fx_2_s1' , dpi=300)
483
484 damage = fatiguedamage_twoslope(t , s2 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
        k=0.25)
485 print('Partial fatigue damage for front_hor_right_strut_fx_2_s2 calculated
        by script: %.5e (includes residual cycles)' %damage)
486 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365))
487
488 figure()
489 plot(t , s2 /1.e6 , 'k-' , label='')
490 xlabel('time [s]')
491 ylabel('front_hor_right_strut_fx_2_s2 [MPa] ')
492 title('Axial Stress')
493 #axis([-16., 90., -1.5, 1.7])
494 #legend(loc='best')
495 savefig('front_hor_right_strut_fx_2_s2' , dpi=300)
496
497 damage = fatiguedamage_twoslope(t , s3 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
        k=0.25)
498 print('Partial fatigue damage for front_hor_right_strut_fx_2_s3 calculated
        by script: %.5e (includes residual cycles)' %damage)
499 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365))
500
501 figure()
502 plot(t , s3 /1.e6 , 'k-' , label='')
503 xlabel('time [s]')
504 ylabel('front_hor_right_strut_fx_2_s3 [MPa] ')
505 title('Axial Stress')
506 #axis([-16., 90., -1.5, 1.7])
```

```
507 #legend(loc='best')
508 savefig('front_hor_right_strut_fx_2_s3',dpi=300)
509
510 damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
511 print('Partial fatigue damage for front_hor_right_strut_fx_2_s4 calculated
    by script: %.5e (includes residual cycles)' %damage)
512 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365))
513
514 figure()
515 plot(t,s4/1.e6,'k-',label='')
516 xlabel('time [s]')
517 ylabel('front_hor_right_strut_fx_2_s4 [MPa]')
518 title('Axial Stress')
519 #axis([-16.,90.,-1.5,1.7])
520 #legend(loc='best')
521 savefig('front_hor_right_strut_fx_2_s4',dpi=300)
522
523 damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
524 print('Partial fatigue damage for front_hor_right_strut_fx_2_s5 calculated
    by script: %.5e (includes residual cycles)' %damage)
525 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365))
526
527 figure()
528 plot(t,s5/1.e6,'k-',label='')
529 xlabel('time [s]')
530 ylabel('front_hor_right_strut_fx_2_s5 [MPa]')
531 title('Axial Stress')
532 #axis([-16.,90.,-1.5,1.7])
533 #legend(loc='best')
534 savefig('front_hor_right_strut_fx_2_s5',dpi=300)
535
536 damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
537 print('Partial fatigue damage for front_hor_right_strut_fx_2_s6 calculated
    by script: %.5e (includes residual cycles)' %damage)
538 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365))
539
540 figure()
541 plot(t,s6/1.e6,'k-',label='')
542 xlabel('time [s]')
543 ylabel('front_hor_right_strut_fx_2_s6 [MPa]')
544 title('Axial Stress')
545 #axis([-16.,90.,-1.5,1.7])
546 #legend(loc='best')
```

```
547 savefig('front_hor_right_strut_fx_2_s6',dpi=300)
548
549 damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
550 print('Partial fatigue damage for front_hor_right_strut_fx_2_s7 calculated
    by script: %.5e (includes residual cycles)' %damage)
551 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365))
552
553 figure()
554 plot(t,s7/1.e6,'k-',label='')
555 xlabel('time [s]')
556 ylabel('front_hor_right_strut_fx_2_s7 [MPa] ')
557 title('Axial Stress')
558 #axis([-16.,90.,-1.5,1.7])
559 #legend(loc='best')
560 savefig('front_hor_right_strut_fx_2_s7',dpi=300)
561
562 damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
563 print('Partial fatigue damage for front_hor_right_strut_fx_2_s8 calculated
    by script: %.5e (includes residual cycles)' %damage)
564 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365))
565
566 figure()
567 plot(t,s8/1.e6,'k-',label='')
568 xlabel('time [s]')
569 ylabel('front_hor_right_strut_fx_2_s8 [MPa] ')
570 title('Axial Stress')
571 #axis([-16.,90.,-1.5,1.7])
572 #legend(loc='best')
573 savefig('front_hor_right_strut_fx_2_s8',dpi=300)
574
575 #
```

```
576
577
578 #front_vert_down_strut_x1
579 #Joint type 1
580 #crown in z-dir
581 front_vert_down_strut_fx_1 = a[:,38]*1.e3;
582 front_vert_down_strut_fy_1 = a[:,110]*1.e3;
583 front_vert_down_strut_fz_1 = a[:,111]*1.e3;
584 mx_fvds_1 = a[:,39]*1.e3;
585 my_fvds_1 = a[:,40]*1.e3;
586 mz_fvds_1 = a[:,41]*1.e3;
587
```

```

588 [sax , sbendy , sbendz , s1 , s2 , s3 , s4 , s5 , s6 , s7 , s8 , ssh1 , ssh2 , ssh3 , ssh4 , ssh5 , ssh6 ,
      ssh7 , ssh8 , svm1 , svm2 , svm3 , svm4 , svm5 , svm6 , svm7 , svm8 , sx , sy , sz ] = \
589     cyl_beam_stresses(t , front_vert_down_strut_fx_1 ,
      front_vert_down_strut_fy_1 , front_vert_down_strut_fz_1 , mx_fvds_1 ,
      my_fvds_1 , mz_fvds_1 , d , twall , SCF_AC_1 , SCF_AS_1 , SCF_IPB_1 , SCF_OPB_1)
590
591 damage = fatiguedamage_twoslope(t , s1 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
      k=0.25)
592 print('Partial fatigue damage for front_vert_down_strut_fx_1_s1 calculated
      by script: %.5e (includes residual cycles)' %damage)
593 print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
      damage*60*60*24*DFE)))
594 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
      damage*60*60*24*365*DFE)))
595
596 figure()
597 plot(t , s1/1.e6 , 'k-' , label='')
598 xlabel('time [s]')
599 ylabel('front_vert_down_strut_fx_1_s1 [MPa] ')
600 title('Axial Stress')
601 #axis([-16. , 90. , -1.5 , 1.7])
602 #legend(loc='best')
603 savefig('front_vert_down_strut_fx_1_s1' , dpi=300)
604
605 damage = fatiguedamage_twoslope(t , s2 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
      k=0.25)
606 print('Partial fatigue damage for front_vert_down_strut_fx_2_s2 calculated
      by script: %.5e (includes residual cycles)' %damage)
607 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
      damage*60*60*24*365*DFE)))
608
609 figure()
610 plot(t , s2/1.e6 , 'k-' , label='')
611 xlabel('time [s]')
612 ylabel('front_vert_down_strut_fx_1_s2 [MPa] ')
613 title('Axial Stress')
614 #axis([-16. , 90. , -1.5 , 1.7])
615 #legend(loc='best')
616 savefig('front_vert_down_strut_fx_1_s2' , dpi=300)
617
618 damage = fatiguedamage_twoslope(t , s3 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
      k=0.25)
619 print('Partial fatigue damage for front_vert_down_strut_fx_2_s3 calculated
      by script: %.5e (includes residual cycles)' %damage)
620 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
      damage*60*60*24*365*DFE)))
621
622 figure()
623 plot(t , s3/1.e6 , 'k-' , label='')

```

```

624 xlabel('time [s]')
625 ylabel('front_vert_down_strut_fx_1_s3 [MPa] ')
626 title('Axial Stress')
627 #axis([-16.,90.,-1.5,1.7])
628 #legend(loc='best')
629 savefig('front_vert_down_strut_fx_1_s3',dpi=300)
630
631 damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
632 print('Partial fatigue damage for front_vert_down_strut_fx_2_s4 calculated
    by script: %.5e (includes residual cycles)' %damage)
633 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
634
635 figure()
636 plot(t,s4/1.e6,'k-',label='')
637 xlabel('time [s]')
638 ylabel('front_vert_down_strut_fx_1_s4 [MPa] ')
639 title('Axial Stress')
640 #axis([-16.,90.,-1.5,1.7])
641 #legend(loc='best')
642 savefig('front_vert_down_strut_fx_1_s4',dpi=300)
643
644 damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
645 print('Partial fatigue damage for front_vert_down_strut_fx_2_s5 calculated
    by script: %.5e (includes residual cycles)' %damage)
646 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
647
648 figure()
649 plot(t,s5/1.e6,'k-',label='')
650 xlabel('time [s]')
651 ylabel('front_vert_down_strut_fx_1_s5 [MPa] ')
652 title('Axial Stress')
653 #axis([-16.,90.,-1.5,1.7])
654 #legend(loc='best')
655 savefig('front_vert_down_strut_fx_1_s5',dpi=300)
656
657 damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
658 print('Partial fatigue damage for front_vert_down_strut_fx_2_s6 calculated
    by script: %.5e (includes residual cycles)' %damage)
659 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
660
661 figure()
662 plot(t,s6/1.e6,'k-',label='')
663 xlabel('time [s]')

```

```
664 ylabel('front_vert_down_strut_fx_1_s6 [MPa] ')
665 title('Axial Stress')
666 #axis([-16., 90., -1.5, 1.7])
667 #legend(loc='best')
668 savefig('front_vert_down_strut_fx_1_s6', dpi=300)
669
670 damage = fatiguedamage_twoslope(t, s7, m1, loga1, m2, loga2, Nlim, th, tref=25E-3,
    k=0.25)
671 print('Partial fatigue damage for front_vert_down_strut_fx_2_s7 calculated
    by script: %.5e (includes residual cycles)' %damage)
672 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
673
674 figure()
675 plot(t, s7/1.e6, 'k-', label='')
676 xlabel('time [s]')
677 ylabel('front_vert_down_strut_fx_1_s7 [MPa] ')
678 title('Axial Stress')
679 #axis([-16., 90., -1.5, 1.7])
680 #legend(loc='best')
681 savefig('front_vert_down_strut_fx_1_s7', dpi=300)
682
683 damage = fatiguedamage_twoslope(t, s8, m1, loga1, m2, loga2, Nlim, th, tref=25E-3,
    k=0.25)
684 print('Partial fatigue damage for front_vert_down_strut_fx_2_s8 calculated
    by script: %.5e (includes residual cycles)' %damage)
685 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
686
687 figure()
688 plot(t, s8/1.e6, 'k-', label='')
689 xlabel('time [s]')
690 ylabel('front_vert_down_strut_fx_1_s8 [MPa] ')
691 title('Axial Stress')
692 #axis([-16., 90., -1.5, 1.7])
693 #legend(loc='best')
694 savefig('front_vert_down_strut_fx_1_s8', dpi=300)
695
696 #

```

```
697
698 #front_vert_down_strut_x2
699 #Joint type 2
700 #saddle in z-dir
701 front_vert_down_strut_fx_2 = a[:, 42]*1.e3;
702 front_vert_down_strut_fy_2 = a[:, 112]*1.e3;
703 front_vert_down_strut_fz_2 = a[:, 113]*1.e3;
704 mx_fvds_2 = a[:, 43]*1.e3;
```

```

705 my_fvds_2 = a[:,44]*1.e3;
706 mz_fvds_2 = a[:,45]*1.e3;
707
708 [sax , sbendy , sbendz , s1 , s2 , s3 , s4 , s5 , s6 , s7 , s8 , ssh1 , ssh2 , ssh3 , ssh4 , ssh5 , ssh6 ,
    ssh7 , ssh8 , svm1 , svm2 , svm3 , svm4 , svm5 , svm6 , svm7 , svm8 , sx , sy , sz ] = \
709     cyl_beam_stresses(t , front_vert_down_strut_fx_2 ,
    front_vert_down_strut_fy_2 , front_vert_down_strut_fz_2 , mx_fvds_2 ,
    my_fvds_2 , mz_fvds_2 , d , twall , SCF_AS_2 , SCF_AC_2 , SCF_OPB_2 , SCF_IPB_2)
710
711 damage = fatiguedamage_twoslope(t , s1 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
    k=0.25)
712 print('Partial fatigue damage for front_vert_down_strut_fx_2_s1 calculated
    by script: %.5e (includes residual cycles)' %damage)
713 print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/
    damage*60*60*24*DFE))
714 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
715
716 figure()
717 plot(t , s1/1.e6 , 'k-' , label='')
718 xlabel('time [s]')
719 ylabel('front_vert_down_strut_fx_2_s1 [MPa] ')
720 title('Axial Stress')
721 #axis([-16. , 90. , -1.5 , 1.7])
722 #legend(loc='best')
723 savefig('front_vert_down_strut_fx_2_s1' , dpi=300)
724
725 damage = fatiguedamage_twoslope(t , s2 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
    k=0.25)
726 print('Partial fatigue damage for front_vert_down_strut_fx_2_s2 calculated
    by script: %.5e (includes residual cycles)' %damage)
727 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
728
729 figure()
730 plot(t , s2/1.e6 , 'k-' , label='')
731 xlabel('time [s]')
732 ylabel('front_vert_down_strut_fx_2_s2 [MPa] ')
733 title('Axial Stress')
734 #axis([-16. , 90. , -1.5 , 1.7])
735 #legend(loc='best')
736 savefig('front_vert_down_strut_fx_2_s2' , dpi=300)
737
738 damage = fatiguedamage_twoslope(t , s3 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
    k=0.25)
739 print('Partial fatigue damage for front_vert_down_strut_fx_2_s3 calculated
    by script: %.5e (includes residual cycles)' %damage)
740 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))

```



```
741
742 figure()
743 plot(t,s3/1.e6,'k-',label='')
744 xlabel('time [s]')
745 ylabel('front_vert_down_strut_fx_2_s3 [MPa]')
746 title('Axial Stress')
747 #axis([-16.,90.,-1.5,1.7])
748 #legend(loc='best')
749 savefig('front_vert_down_strut_fx_2_s3',dpi=300)
750
751 damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
752 print('Partial fatigue damage for front_vert_down_strut_fx_2_s4 calculated
    by script: %.5e (includes residual cycles)' %damage)
753 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFF)))
754
755 figure()
756 plot(t,s4/1.e6,'k-',label='')
757 xlabel('time [s]')
758 ylabel('front_vert_down_strut_fx_2_s4 [MPa]')
759 title('Axial Stress')
760 #axis([-16.,90.,-1.5,1.7])
761 #legend(loc='best')
762 savefig('front_vert_down_strut_fx_2_s4',dpi=300)
763
764 damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
765 print('Partial fatigue damage for front_vert_down_strut_fx_2_s5 calculated
    by script: %.5e (includes residual cycles)' %damage)
766 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFF)))
767
768 figure()
769 plot(t,s5/1.e6,'k-',label='')
770 xlabel('time [s]')
771 ylabel('front_vert_down_strut_fx_2_s5 [MPa]')
772 title('Axial Stress')
773 #axis([-16.,90.,-1.5,1.7])
774 #legend(loc='best')
775 savefig('front_vert_down_strut_fx_2_s5',dpi=300)
776
777 damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
778 print('Partial fatigue damage for front_vert_down_strut_fx_2_s6 calculated
    by script: %.5e (includes residual cycles)' %damage)
779 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFF)))
780
```

```
781 figure()
782 plot(t,s6/1.e6,'k-',label='')
783 xlabel('time [s]')
784 ylabel('front_vert_down_strut_fx_2_s6 [MPa] ')
785 title('Axial Stress')
786 #axis([-16.,90.,-1.5,1.7])
787 #legend(loc='best')
788 savefig('front_vert_down_strut_fx_2_s6',dpi=300)
789
790 damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
791 print('Partial fatigue damage for front_vert_down_strut_fx_2_s7 calculated
    by script: %.5e (includes residual cycles)' %damage)
792 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
793
794 figure()
795 plot(t,s7/1.e6,'k-',label='')
796 xlabel('time [s]')
797 ylabel('front_vert_down_strut_fx_2_s7 [MPa] ')
798 title('Axial Stress')
799 #axis([-16.,90.,-1.5,1.7])
800 #legend(loc='best')
801 savefig('front_vert_down_strut_fx_2_s7',dpi=300)
802
803 damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
804 print('Partial fatigue damage for front_vert_down_strut_fx_2_s8 calculated
    by script: %.5e (includes residual cycles)' %damage)
805 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
806
807 figure()
808 plot(t,s8/1.e6,'k-',label='')
809 xlabel('time [s]')
810 ylabel('front_vert_down_strut_fx_2_s8 [MPa] ')
811 title('Axial Stress')
812 #axis([-16.,90.,-1.5,1.7])
813 #legend(loc='best')
814 savefig('front_vert_down_strut_fx_2_s8',dpi=300)
815 #

```

```
816
817 #front_vert_up_strut_x1
818 #Joint type 1
819 #crown in z-dir
820 front_vert_up_strut_fx_1 = a[:,46]*1.e3;
821 front_vert_up_strut_fy_1 = a[:,114]*1.e3;
```

```

822 front_vert_up_strut_fz_1 = a[:,115]*1.e3;
823 mx_fvus_1 = a[:,47]*1.e3;
824 my_fvus_1 = a[:,48]*1.e3;
825 mz_fvus_1 = a[:,49]*1.e3;
826
827 [sax , sbendy , sbendz , s1 , s2 , s3 , s4 , s5 , s6 , s7 , s8 , ssh1 , ssh2 , ssh3 , ssh4 , ssh5 , ssh6 ,
    ssh7 , ssh8 , svm1 , svm2 , svm3 , svm4 , svm5 , svm6 , svm7 , svm8 , sx , sy , sz ] = \
828     cyl_beam_stresses(t , front_vert_up_strut_fx_1 , front_vert_up_strut_fy_1 ,
        front_vert_up_strut_fz_1 , mx_fvds_2 , my_fvds_2 , mz_fvds_2 , d , twall ,
        SCF_AC_1 , SCF_AS_1 , SCF_IPB_1 , SCF_OPB_1)
829
830 damage = fatiguedamage_twoslope(t , s1 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
    k=0.25)
831 print('Partial fatigue damage for front_vert_up_strut_fx_1_s1 calculated
    by script: %.5e (includes residual cycles)' %damage)
832 print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFE)))
833 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFE)))
834
835 figure()
836 plot(t , s1/1.e6 , 'k-' , label='')
837 xlabel('time [s]')
838 ylabel('front_vert_up_strut_fx_1_s1 [MPa] ')
839 title('Axial Stress')
840 #axis([-16.,90., -1.5,1.7])
841 #legend(loc='best')
842 savefig('front_vert_up_strut_fx_1_s1' , dpi=300)
843
844 damage = fatiguedamage_twoslope(t , s2 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
    k=0.25)
845 print('Partial fatigue damage for front_vert_up_strut_fx_1_s2 calculated
    by script: %.5e (includes residual cycles)' %damage)
846 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFE)))
847
848 figure()
849 plot(t , s2/1.e6 , 'k-' , label='')
850 xlabel('time [s]')
851 ylabel('front_vert_up_strut_fx_1_s2 [MPa] ')
852 title('Axial Stress')
853 #axis([-16.,90., -1.5,1.7])
854 #legend(loc='best')
855 savefig('front_vert_up_strut_fx_1_s2' , dpi=300)
856
857 damage = fatiguedamage_twoslope(t , s3 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
    k=0.25)
858 print('Partial fatigue damage for front_vert_up_strut_fx_1_s3 calculated
    by script: %.5e (includes residual cycles)' %damage)

```

```

859 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFF))
860
861 figure()
862 plot(t,s3/1.e6,'k-',label='')
863 xlabel('time [s]')
864 ylabel('front_vert_up_strut_fx_1_s3 [MPa] ')
865 title('Axial Stress')
866 #axis([-16.,90.,-1.5,1.7])
867 #legend(loc='best')
868 savefig('front_vert_up_strut_fx_1_s3',dpi=300)
869
870 damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
871 print('Partial fatigue damage for front_vert_up_strut_fx_1_s4 calculated
      by script: %.5e (includes residual cycles)' %damage)
872 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFF))
873
874 figure()
875 plot(t,s4/1.e6,'k-',label='')
876 xlabel('time [s]')
877 ylabel('front_vert_up_strut_fx_1_s4 [MPa] ')
878 title('Axial Stress')
879 #axis([-16.,90.,-1.5,1.7])
880 #legend(loc='best')
881 savefig('front_vert_up_strut_fx_1_s4',dpi=300)
882
883 damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
884 print('Partial fatigue damage for front_vert_up_strut_fx_1_s5 calculated
      by script: %.5e (includes residual cycles)' %damage)
885 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFF))
886
887 figure()
888 plot(t,s5/1.e6,'k-',label='')
889 xlabel('time [s]')
890 ylabel('front_vert_up_strut_fx_1_s5 [MPa] ')
891 title('Axial Stress')
892 #axis([-16.,90.,-1.5,1.7])
893 #legend(loc='best')
894 savefig('front_vert_up_strut_fx_1_s5',dpi=300)
895
896 damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
897 print('Partial fatigue damage for front_vert_up_strut_fx_1_s6 calculated
      by script: %.5e (includes residual cycles)' %damage)
898 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/

```

```
        damage*60*60*24*365*DFF)))
899
900 figure()
901 plot(t,s6/1.e6,'k-',label='')
902 xlabel('time [s]')
903 ylabel('front_vert_up_strut_fx_1_s6 [MPa]')
904 title('Axial Stress')
905 #axis([-16.,90.,-1.5,1.7])
906 #legend(loc='best')
907 savefig('front_vert_up_strut_fx_1_s6',dpi=300)
908
909 damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
910 print('Partial fatigue damage for front_vert_up_strut_fx_1_s7 calculated
        by script: %.5e (includes residual cycles)' %damage)
911 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF)))
912
913 figure()
914 plot(t,s7/1.e6,'k-',label='')
915 xlabel('time [s]')
916 ylabel('front_vert_up_strut_fx_1_s7 [MPa]')
917 title('Axial Stress')
918 #axis([-16.,90.,-1.5,1.7])
919 #legend(loc='best')
920 savefig('front_vert_up_strut_fx_1_s7',dpi=300)
921
922 damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
923 print('Partial fatigue damage for front_vert_up_strut_fx_1_s8 calculated
        by script: %.5e (includes residual cycles)' %damage)
924 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF)))
925
926 figure()
927 plot(t,s8/1.e6,'k-',label='')
928 xlabel('time [s]')
929 ylabel('front_vert_up_strut_fx_1_s8 [MPa]')
930 title('Axial Stress')
931 #axis([-16.,90.,-1.5,1.7])
932 #legend(loc='best')
933 savefig('front_vert_up_strut_fx_1_s8',dpi=300)
934
935
936 #


---


937
938 #front_vert_up_strut_x2
```

```

939 #Joint type 2
940 #saddle in z-dir
941 front_vert_up_strut_fx_2 = a[:,50]*1.e3;
942 front_vert_up_strut_fy_2 = a[:,116]*1.e3;
943 front_vert_up_strut_fz_2 = a[:,117]*1.e3;
944 mx_fvus_2 = a[:,51]*1.e3;
945 my_fvus_2 = a[:,52]*1.e3;
946 mz_fvus_2 = a[:,53]*1.e3;
947
948 [sax , sbendy , sbendz , s1 , s2 , s3 , s4 , s5 , s6 , s7 , s8 , ssh1 , ssh2 , ssh3 , ssh4 , ssh5 , ssh6 ,
    ssh7 , ssh8 , svm1 , svm2 , svm3 , svm4 , svm5 , svm6 , svm7 , svm8 , sx , sy , sz ] = \
949     cyl_beam_stresses(t , front_vert_up_strut_fx_2 , front_vert_up_strut_fy_2 ,
        front_vert_up_strut_fz_2 , mx_fvus_2 , my_fvus_2 , mz_fvus_2 , d , twall ,
        SCF_AS_2 , SCF_AC_2 , SCF_OPB_2 , SCF_IPB_2)
950
951 damage = fatiguedamage_twoslope(t , s1 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
    k=0.25)
952 print('Partial fatigue damage front_vert_up_strut_fx_2_s1 calculated by
    script: %.5e (includes residual cycles)' %damage)
953 print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFE)))
954 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFE)))
955
956 figure()
957 plot(t , s1/1.e6 , 'k-' , label='')
958 xlabel('time [s]')
959 ylabel('front_vert_up_strut_fx_2_s1 [MPa] ')
960 title('Axial Stress')
961 #axis([-16. , 90. , -1.5 , 1.7])
962 #legend(loc='best')
963 savefig('front_vert_up_strut_fx_2_s1' , dpi=300)
964
965 damage = fatiguedamage_twoslope(t , s2 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
    k=0.25)
966 print('Partial fatigue damage for front_vert_up_strut_fx_2_s2 calculated
    by script: %.5e (includes residual cycles)' %damage)
967 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFE)))
968
969 figure()
970 plot(t , s2/1.e6 , 'k-' , label='')
971 xlabel('time [s]')
972 ylabel('front_vert_up_strut_fx_2_s2 [MPa] ')
973 title('Axial Stress')
974 #axis([-16. , 90. , -1.5 , 1.7])
975 #legend(loc='best')
976 savefig('front_vert_up_strut_fx_2_s2' , dpi=300)
977

```

```
978 damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
979 print('Partial fatigue damage for front_vert_up_strut_fx_2_s3 calculated
    by script: %.5e (includes residual cycles)' %damage)
980 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
981
982 figure()
983 plot(t,s3/1.e6,'k-',label='')
984 xlabel('time [s]')
985 ylabel('front_vert_up_strut_fx_2_s3 [MPa]')
986 title('Axial Stress')
987 #axis([-16.,90.,-1.5,1.7])
988 #legend(loc='best')
989 savefig('front_vert_up_strut_fx_2_s3',dpi=300)
990
991 damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
992 print('Partial fatigue damage for front_vert_up_strut_fx_2_s4 calculated
    by script: %.5e (includes residual cycles)' %damage)
993 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
994
995 figure()
996 plot(t,s4/1.e6,'k-',label='')
997 xlabel('time [s]')
998 ylabel('front_vert_up_strut_fx_2_s4 [MPa]')
999 title('Axial Stress')
1000 #axis([-16.,90.,-1.5,1.7])
1001 #legend(loc='best')
1002 savefig('front_vert_up_strut_fx_2_s4',dpi=300)
1003
1004 damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1005 print('Partial fatigue damage for front_vert_up_strut_fx_2_s5 calculated
    by script: %.5e (includes residual cycles)' %damage)
1006 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
1007
1008 figure()
1009 plot(t,s5/1.e6,'k-',label='')
1010 xlabel('time [s]')
1011 ylabel('front_vert_up_strut_fx_2_s5 [MPa]')
1012 title('Axial Stress')
1013 #axis([-16.,90.,-1.5,1.7])
1014 #legend(loc='best')
1015 savefig('front_vert_up_strut_fx_2_s5',dpi=300)
1016
1017 damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
```

```

    k=0.25)
1018 print('Partial fatigue damage for front_vert_up_strut_fx_2_s6 calculated
        by script: %.5e (includes residual cycles)' %damage)
1019 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF))
1020
1021 figure()
1022 plot(t,s6/1.e6,'k-',label='')
1023 xlabel('time [s]')
1024 ylabel('front_vert_up_strut_fx_2_s6 [MPa]')
1025 title('Axial Stress')
1026 #axis([-16.,90.,-1.5,1.7])
1027 #legend(loc='best')
1028 savefig('front_vert_up_strut_fx_2_s6',dpi=300)
1029
1030 damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1031 print('Partial fatigue damage for front_vert_up_strut_fx_2_s7 calculated
        by script: %.5e (includes residual cycles)' %damage)
1032 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF))
1033
1034 figure()
1035 plot(t,s7/1.e6,'k-',label='')
1036 xlabel('time [s]')
1037 ylabel('front_vert_up_strut_fx_2_s7 [MPa]')
1038 title('Axial Stress')
1039 #axis([-16.,90.,-1.5,1.7])
1040 #legend(loc='best')
1041 savefig('front_vert_up_strut_fx_2_s7',dpi=300)
1042
1043 damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1044 print('Partial fatigue damage for front_vert_up_strut_fx_2_s8 calculated
        by script: %.5e (includes residual cycles)' %damage)
1045 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF))
1046
1047 figure()
1048 plot(t,s8/1.e6,'k-',label='')
1049 xlabel('time [s]')
1050 ylabel('front_vert_up_strut_fx_2_s8 [MPa]')
1051 title('Axial Stress')
1052 #axis([-16.,90.,-1.5,1.7])
1053 #legend(loc='best')
1054 savefig('front_vert_up_strut_fx_2_s8',dpi=300)
1055
1056
1057 #

```



```

1058
1059 #rear_left_down_strut_x1
1060 #Joint type 3
1061 #crown in z-dir
1062 rear_left_down_strut_fx_1 = a[:,54]*1.e3;
1063 rear_left_down_strut_fy_1 = a[:,118]*1.e3;
1064 rear_left_down_strut_fz_1 = a[:,119]*1.e3;
1065 mx_rlds_1 = a[:,55]*1.e3;
1066 my_rlds_1 = a[:,56]*1.e3;
1067 mz_rlds_1 = a[:,57]*1.e3;
1068
1069
1070 [ sax , sbendy , sbendz , s1 , s2 , s3 , s4 , s5 , s6 , s7 , s8 , ssh1 , ssh2 , ssh3 , ssh4 , ssh5 , ssh6 ,
      ssh7 , ssh8 , svm1 , svm2 , svm3 , svm4 , svm5 , svm6 , svm7 , svm8 , sx , sy , sz ] = \
1071     cyl_beam_stresses(t , rear_left_down_strut_fx_1 ,
      rear_left_down_strut_fy_1 , rear_left_down_strut_fz_1 , mx_rlds_1 ,
      my_rlds_1 , mz_rlds_1 , d , twall , SCF_AC_3 , SCF_AS_3 , SCF_IPB_3 , SCF_OPB_3)
1072
1073 damage = fatiguedamage_twoslope(t , s1 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
      k=0.25)
1074 print('Partial fatigue damage for rear_left_down_strut_fx_1_s1 calculated
      by script: %.5e (includes residual cycles)' %damage)
1075 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365))
1076
1077 figure()
1078 plot(t , s1/1.e6 , 'k-' , label='')
1079 xlabel('time [s]')
1080 ylabel('rear_left_down_strut_fx_1_s1 [MPa] ')
1081 title('Axial Stress')
1082 #axis([-16. , 90. , -1.5 , 1.7])
1083 #legend(loc='best')
1084 savefig('rear_left_down_strut_fx_1_s1' , dpi=300)
1085
1086 damage = fatiguedamage_twoslope(t , s2 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
      k=0.25)
1087 print('Partial fatigue damage for rear_left_down_strut_fx_1_s2 calculated
      by script: %.5e (includes residual cycles)' %damage)
1088 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365))
1089
1090 figure()
1091 plot(t , s2/1.e6 , 'k-' , label='')
1092 xlabel('time [s]')
1093 ylabel('rear_left_down_strut_fx_1_s2 [MPa] ')
1094 title('Axial Stress')
1095 #axis([-16. , 90. , -1.5 , 1.7])

```

```

1096 #legend(loc='best')
1097 savefig('rear_left_down_strut_fx_1_s2',dpi=300)
1098
1099 damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1100 print('Partial fatigue damage for rear_left_down_strut_fx_1_s3 calculated
    by script: %.5e (includes residual cycles)' %damage)
1101 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365))
1102
1103 figure()
1104 plot(t,s3/1.e6,'k-',label='')
1105 xlabel('time [s]')
1106 ylabel('rear_left_down_strut_fx_1_s3 [MPa]')
1107 title('Axial Stress')
1108 #axis([-16.,90.,-1.5,1.7])
1109 #legend(loc='best')
1110 savefig('rear_left_down_strut_fx_1_s3',dpi=300)
1111
1112 damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1113 print('Partial fatigue damage for rear_left_down_strut_fx_1_s4 calculated
    by script: %.5e (includes residual cycles)' %damage)
1114 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365))
1115
1116 figure()
1117 plot(t,s4/1.e6,'k-',label='')
1118 xlabel('time [s]')
1119 ylabel('rear_left_down_strut_fx_1_s4 [MPa]')
1120 title('Axial Stress')
1121 #axis([-16.,90.,-1.5,1.7])
1122 #legend(loc='best')
1123 savefig('rear_left_down_strut_fx_1_s4',dpi=300)
1124
1125 damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1126 print('Partial fatigue damage for rear_left_down_strut_fx_1_s5 calculated
    by script: %.5e (includes residual cycles)' %damage)
1127 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365))
1128
1129 figure()
1130 plot(t,s5/1.e6,'k-',label='')
1131 xlabel('time [s]')
1132 ylabel('rear_left_down_strut_fx_1_s5 [MPa]')
1133 title('Axial Stress')
1134 #axis([-16.,90.,-1.5,1.7])
1135 #legend(loc='best')

```

```

1136 savefig('rear_left_down_strut_fx_1_s5',dpi=300)
1137
1138 damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1139 print('Partial fatigue damage for rear_left_down_strut_fx_1_s6 calculated
    by script: %.5e (includes residual cycles)' %damage)
1140 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365))
1141
1142 figure()
1143 plot(t,s6/1.e6,'k-',label='')
1144 xlabel('time [s]')
1145 ylabel('rear_left_down_strut_fx_1_s6 [MPa] ')
1146 title('Axial Stress')
1147 #axis([-16.,90.,-1.5,1.7])
1148 #legend(loc='best')
1149 savefig('rear_left_down_strut_fx_1_s6',dpi=300)
1150
1151 damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1152 print('Partial fatigue damage for rear_left_down_strut_fx_1_s7 calculated
    by script: %.5e (includes residual cycles)' %damage)
1153 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365))
1154
1155 figure()
1156 plot(t,s7/1.e6,'k-',label='')
1157 xlabel('time [s]')
1158 ylabel('rear_left_down_strut_fx_1_s7 [MPa] ')
1159 title('Axial Stress')
1160 #axis([-16.,90.,-1.5,1.7])
1161 #legend(loc='best')
1162 savefig('rear_left_down_strut_fx_1_s7',dpi=300)
1163
1164 damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1165 print('Partial fatigue damage for rear_left_down_strut_fx_1_s8 calculated
    by script: %.5e (includes residual cycles)' %damage)
1166 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365))
1167
1168 figure()
1169 plot(t,s8/1.e6,'k-',label='')
1170 xlabel('time [s]')
1171 ylabel('rear_left_down_strut_fx_1_s8 [MPa] ')
1172 title('Axial Stress')
1173 #axis([-16.,90.,-1.5,1.7])
1174 #legend(loc='best')
1175 savefig('rear_left_down_strut_fx_1_s8',dpi=300)

```

```

1176
1177 #

```

```

1178
1179
1180 #rear_left_down_strut_x2
1181 #Joint type 4
1182 #crown in z-dir
1183 rear_left_down_strut_fx_2 = a[:,58]*1.e3;
1184 rear_left_down_strut_fy_2 = a[:,120]*1.e3;
1185 rear_left_down_strut_fz_2 = a[:,121]*1.e3;
1186 mx_rlds_2 = a[:,59]*1.e3;
1187 my_rlds_2 = a[:,60]*1.e3;
1188 mz_rlds_2 = a[:,61]*1.e3;
1189
1190 [sax , sbendy , sbendz , s1 , s2 , s3 , s4 , s5 , s6 , s7 , s8 , ssh1 , ssh2 , ssh3 , ssh4 , ssh5 , ssh6 ,
    ssh7 , ssh8 , svm1 , svm2 , svm3 , svm4 , svm5 , svm6 , svm7 , svm8 , sx , sy , sz ] = \
1191     cyl_beam_stresses(t , front_vert_up_strut_fx_2 , rear_left_down_strut_fy_2
    , rear_left_down_strut_fy_2 , mx_rlds_2 , my_rlds_2 , mz_rlds_2 , d , twall ,
    SCF_AC_4 , SCF_AS_4 , SCF_IPB_4 , SCF_OPB_4)
1192
1193 damage = fatiguedamage_twoslope(t , s1 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
    k=0.25)
1194 print('Partial fatigue damage for rear_left_down_strut_fx_2_s1 calculated
    by script: %.5e (includes residual cycles)' %damage)
1195 print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFE)))
1196 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFE)))
1197
1198 figure()
1199 plot(t , s1/1.e6 , 'k-' , label='')
1200 xlabel('time [s]')
1201 ylabel('rear_left_down_strut_fx_2_s1 [MPa] ')
1202 title('Axial Stress')
1203 #axis([-16. , 90. , -1.5 , 1.7])
1204 #legend(loc='best')
1205 savefig('rear_left_down_strut_fx_2_s1' , dpi=300)
1206
1207 damage = fatiguedamage_twoslope(t , s2 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
    k=0.25)
1208 print('Partial fatigue damage for rear_left_down_strut_fx_2_s2 calculated
    by script: %.5e (includes residual cycles)' %damage)
1209 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFE)))
1210
1211 figure()
1212 plot(t , s2/1.e6 , 'k-' , label='')

```

```
1213 xlabel('time [s]')
1214 ylabel('rear_left_down_strut_fx_2_s2 [MPa] ')
1215 title('Axial Stress')
1216 #axis([-16.,90.,-1.5,1.7])
1217 #legend(loc='best')
1218 savefig('rear_left_down_strut_fx_2_s2',dpi=300)
1219
1220 damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1221 print('Partial fatigue damage for rear_left_down_strut_fx_2_s3 calculated
    by script: %.5e (includes residual cycles)' %damage)
1222 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
1223
1224 figure()
1225 plot(t,s3/1.e6,'k-',label='')
1226 xlabel('time [s]')
1227 ylabel('rear_left_down_strut_fx_2_s3 [MPa] ')
1228 title('Axial Stress')
1229 #axis([-16.,90.,-1.5,1.7])
1230 #legend(loc='best')
1231 savefig('rear_left_down_strut_fx_2_s3',dpi=300)
1232
1233 damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1234 print('Partial fatigue damage for rear_left_down_strut_fx_2_s4 calculated
    by script: %.5e (includes residual cycles)' %damage)
1235 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
1236
1237 figure()
1238 plot(t,s4/1.e6,'k-',label='')
1239 xlabel('time [s]')
1240 ylabel('rear_left_down_strut_fx_2_s4 [MPa] ')
1241 title('Axial Stress')
1242 #axis([-16.,90.,-1.5,1.7])
1243 #legend(loc='best')
1244 savefig('rear_left_down_strut_fx_2_s4',dpi=300)
1245
1246 damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1247 print('Partial fatigue damage for rear_left_down_strut_fx_2_s5 calculated
    by script: %.5e (includes residual cycles)' %damage)
1248 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
1249
1250 figure()
1251 plot(t,s5/1.e6,'k-',label='')
1252 xlabel('time [s]')
```

```

1253 ylabel('rear_left_down_strut_fx_2_s5 [MPa] ')
1254 title('Axial Stress')
1255 #axis([-16.,90.,-1.5,1.7])
1256 #legend(loc='best')
1257 savefig('rear_left_down_strut_fx_2_s5',dpi=300)
1258
1259 damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1260 print('Partial fatigue damage for rear_left_down_strut_fx_2_s6 calculated
    by script: %.5e (includes residual cycles)' %damage)
1261 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
1262
1263 figure()
1264 plot(t,s6/1.e6,'k-',label='')
1265 xlabel('time [s]')
1266 ylabel('rear_left_down_strut_fx_2_s6 [MPa] ')
1267 title('Axial Stress')
1268 #axis([-16.,90.,-1.5,1.7])
1269 #legend(loc='best')
1270 savefig('rear_left_down_strut_fx_2_s6',dpi=300)
1271
1272 damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1273 print('Partial fatigue damage for rear_left_down_strut_fx_2_s7 calculated
    by script: %.5e (includes residual cycles)' %damage)
1274 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
1275
1276 figure()
1277 plot(t,s7/1.e6,'k-',label='')
1278 xlabel('time [s]')
1279 ylabel('rear_left_down_strut_fx_2_s7 [MPa] ')
1280 title('Axial Stress')
1281 #axis([-16.,90.,-1.5,1.7])
1282 #legend(loc='best')
1283 savefig('rear_left_down_strut_fx_2_s7',dpi=300)
1284
1285 damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1286 print('Partial fatigue damage for rear_left_down_strut_fx_2_s8 calculated
    by script: %.5e (includes residual cycles)' %damage)
1287 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
1288
1289 figure()
1290 plot(t,s8/1.e6,'k-',label='')
1291 xlabel('time [s]')
1292 ylabel('rear_left_down_strut_fx_2_s8 [MPa] ')

```

```

1293 title('Axial Stress')
1294 #axis([-16.,90.,-1.5,1.7])
1295 #legend(loc='best')
1296 savefig('rear_left_down_strut_fx_2_s8',dpi=300)
1297
1298
1299
1300 #

```

```

1301
1302
1303 #rear_left_up_strut_x1
1304 #Joint type 3
1305 #crown in z-dir
1306 rear_left_up_strut_fx_1 = a[:,62]*1.e3;
1307 rear_left_up_strut_fy_1 = a[:,122]*1.e3;
1308 rear_left_up_strut_fz_1 = a[:,123]*1.e3;
1309 mx_rlus_1 = a[:,63]*1.e3;
1310 my_rlus_1 = a[:,64]*1.e3;
1311 mz_rlus_1 = a[:,65]*1.e3;
1312
1313 [sax, sbendy, sbendz, s1, s2, s3, s4, s5, s6, s7, s8, ssh1, ssh2, ssh3, ssh4, ssh5, ssh6,
    ssh7, ssh8, svm1, svm2, svm3, svm4, svm5, svm6, svm7, svm8, sx, sy, sz] = \
1314     cyl_beam_stresses(t, rear_left_up_strut_fx_1, rear_left_up_strut_fy_1,
        rear_left_up_strut_fz_1, mx_rlus_1, my_rlus_1, mz_rlus_1, d, twall,
        SCF_AC_3, SCF_AS_3, SCF_IPB_3, SCF_OPB_3)
1315
1316 damage = fatiguedamage_twoslope(t, s1, m1, loga1, m2, loga2, Nlim, th, tref=25E-3,
    k=0.25)
1317 print('Partial fatigue damage for rear_left_up_strut_fx_1_s1 calculated by
    script: %.5e (includes residual cycles)' %damage)
1318 print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFE)))
1319 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFE)))
1320
1321 figure()
1322 plot(t, s1/1.e6, 'k-', label='')
1323 xlabel('time [s]')
1324 ylabel('rear_left_up_strut_fx_1_s1 [MPa]')
1325 title('Axial Stress')
1326 #axis([-16.,90.,-1.5,1.7])
1327 #legend(loc='best')
1328 savefig('rear_left_up_strut_fx_1_s1',dpi=300)
1329
1330 damage = fatiguedamage_twoslope(t, s2, m1, loga1, m2, loga2, Nlim, th, tref=25E-3,
    k=0.25)
1331 print('Partial fatigue damage for rear_left_up_strut_fx_1_s2 calculated by

```

```

        script: %.5e (includes residual cycles)' %damage)
1332 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFE)))
1333
1334 figure()
1335 plot(t,s2/1.e6,'k-',label='')
1336 xlabel('time [s]')
1337 ylabel('rear_left_up_strut_fx_1_s2 [MPa] ')
1338 title('Axial Stress')
1339 #axis([-16.,90.,-1.5,1.7])
1340 #legend(loc='best')
1341 savefig('rear_left_up_strut_fx_1_s2',dpi=300)
1342
1343 damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
1344 print('Partial fatigue damage for rear_left_up_strut_fx_1_s3 calculated by
        script: %.5e (includes residual cycles)' %damage)
1345 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFE)))
1346
1347 figure()
1348 plot(t,s3/1.e6,'k-',label='')
1349 xlabel('time [s]')
1350 ylabel('rear_left_up_strut_fx_1_s3 [MPa] ')
1351 title('Axial Stress')
1352 #axis([-16.,90.,-1.5,1.7])
1353 #legend(loc='best')
1354 savefig('rear_left_up_strut_fx_1_s3',dpi=300)
1355
1356 damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
1357 print('Partial fatigue damage for rear_left_up_strut_fx_1_s4 calculated by
        script: %.5e (includes residual cycles)' %damage)
1358 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFE)))
1359
1360 figure()
1361 plot(t,s4/1.e6,'k-',label='')
1362 xlabel('time [s]')
1363 ylabel('rear_left_up_strut_fx_1_s4 [MPa] ')
1364 title('Axial Stress')
1365 #axis([-16.,90.,-1.5,1.7])
1366 #legend(loc='best')
1367 savefig('rear_left_up_strut_fx_1_s4',dpi=300)
1368
1369 damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
1370 print('Partial fatigue damage for rear_left_up_strut_fx_1_s5 calculated by
        script: %.5e (includes residual cycles)' %damage)

```



```

1371 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF))
1372
1373 figure()
1374 plot(t,s5/1.e6,'k-',label='')
1375 xlabel('time [s]')
1376 ylabel('rear_left_up_strut_fx_1_s5 [MPa]')
1377 title('Axial Stress')
1378 #axis([-16.,90.,-1.5,1.7])
1379 #legend(loc='best')
1380 savefig('rear_left_up_strut_fx_1_s5',dpi=300)
1381
1382 damage = fatiguedamage_twslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
1383 print('Partial fatigue damage for rear_left_up_strut_fx_1_s6 calculated by
        script: %.5e (includes residual cycles)' %damage)
1384 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF))
1385
1386 figure()
1387 plot(t,s6/1.e6,'k-',label='')
1388 xlabel('time [s]')
1389 ylabel('rear_left_up_strut_fx_1_s6 [MPa]')
1390 title('Axial Stress')
1391 #axis([-16.,90.,-1.5,1.7])
1392 #legend(loc='best')
1393 savefig('rear_left_up_strut_fx_1_s6',dpi=300)
1394
1395 damage = fatiguedamage_twslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
1396 print('Partial fatigue damage for rear_left_up_strut_fx_1_s7 calculated by
        script: %.5e (includes residual cycles)' %damage)
1397 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF))
1398
1399 figure()
1400 plot(t,s7/1.e6,'k-',label='')
1401 xlabel('time [s]')
1402 ylabel('rear_left_up_strut_fx_1_s7 [MPa]')
1403 title('Axial Stress')
1404 #axis([-16.,90.,-1.5,1.7])
1405 #legend(loc='best')
1406 savefig('rear_left_up_strut_fx_1_s7',dpi=300)
1407
1408 damage = fatiguedamage_twslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
1409 print('Partial fatigue damage for rear_left_up_strut_fx_1_s8 calculated by
        script: %.5e (includes residual cycles)' %damage)
1410 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/

```

```

        damage*60*60*24*365*DFF)))
1411
1412 figure()
1413 plot(t,s8/1.e6,'k-',label='')
1414 xlabel('time [s]')
1415 ylabel('rear_left_up_strut_fx_1_s8 [MPa] ')
1416 title('Axial Stress')
1417 #axis([-16.,90.,-1.5,1.7])
1418 #legend(loc='best')
1419 savefig('rear_left_up_strut_fx_1_s8',dpi=300)
1420
1421
1422
1423
1424 #

```

```

1425
1426 #rear_left_up_strut_x2
1427 #Joint type 4
1428 #crown in z-dir
1429 rear_left_up_strut_fx_2 = a[:,66]*1.e3;
1430 rear_left_up_strut_fy_2 = a[:,124]*1.e3;
1431 rear_left_up_strut_fz_2 = a[:,125]*1.e3;
1432 mx_rlus_2 = a[:,67]*1.e3;
1433 my_rlus_2 = a[:,68]*1.e3;
1434 mz_rlus_2 = a[:,69]*1.e3;
1435
1436 [sax,sbendy,sbendz,s1,s2,s3,s4,s5,s6,s7,s8,ssh1,ssh2,ssh3,ssh4,ssh5,ssh6,
    ssh7,ssh8,svm1,svm2,svm3,svm4,svm5,svm6,svm7,svm8,sx,sy,sz] = \
1437     cyl_beam_stresses(t,rear_left_up_strut_fx_2,rear_left_up_strut_fy_2,
        rear_left_up_strut_fz_2,mx_rlus_2,my_rlus_2,mz_rlus_2,d,twall,
        SCF_AC_4,SCF_AS_4,SCF_IPB_4,SCF_OPB_4)
1438
1439 damage = fatiguedamage_twslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1440 print('Partial fatigue damage for rear_left_down_strut_fx_2_s1 calculated
    by script: %.5e (includes residual cycles)' %damage)
1441 print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/
    damage*60*60*24*DFF)))
1442 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFF)))
1443
1444 figure()
1445 plot(t,s1/1.e6,'k-',label='')
1446 xlabel('time [s]')
1447 ylabel('rear_left_up_strut_fx_2_s1 [MPa] ')
1448 title('Axial Stress')
1449 #axis([-16.,90.,-1.5,1.7])

```

```
1450 #legend(loc='best')
1451 savefig('rear_left_up_strut_fx_2_s1',dpi=300)
1452
1453 damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1454 print('Partial fatigue damage for rear_left_down_strut_fx_2_s2 calculated
    by script: %.5e (includes residual cycles)' %damage)
1455 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
1456
1457 figure()
1458 plot(t,s2/1.e6,'k-',label='')
1459 xlabel('time [s]')
1460 ylabel('rear_left_up_strut_fx_2_s2 [MPa]')
1461 title('Axial Stress')
1462 #axis([-16.,90.,-1.5,1.7])
1463 #legend(loc='best')
1464 savefig('rear_left_up_strut_fx_2_s2',dpi=300)
1465
1466 damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1467 print('Partial fatigue damage for rear_left_down_strut_fx_2_s3 calculated
    by script: %.5e (includes residual cycles)' %damage)
1468 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
1469
1470 figure()
1471 plot(t,s3/1.e6,'k-',label='')
1472 xlabel('time [s]')
1473 ylabel('rear_left_up_strut_fx_2_s3 [MPa]')
1474 title('Axial Stress')
1475 #axis([-16.,90.,-1.5,1.7])
1476 #legend(loc='best')
1477 savefig('rear_left_up_strut_fx_2_s3',dpi=300)
1478
1479 damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1480 print('Partial fatigue damage for rear_left_down_strut_fx_2_s4 calculated
    by script: %.5e (includes residual cycles)' %damage)
1481 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
1482
1483 figure()
1484 plot(t,s4/1.e6,'k-',label='')
1485 xlabel('time [s]')
1486 ylabel('rear_left_up_strut_fx_2_s4 [MPa]')
1487 title('Axial Stress')
1488 #axis([-16.,90.,-1.5,1.7])
1489 #legend(loc='best')
```

```
1490 savefig('rear_left_up_strut_fx_2_s4',dpi=300)
1491
1492 damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1493 print('Partial fatigue damage for rear_left_down_strut_fx_2_s5 calculated
    by script: %.5e (includes residual cycles)' %damage)
1494 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
1495
1496 figure()
1497 plot(t,s5/1.e6,'k-',label='')
1498 xlabel('time [s]')
1499 ylabel('rear_left_up_strut_fx_2_s5 [MPa]')
1500 title('Axial Stress')
1501 #axis([-16.,90.,-1.5,1.7])
1502 #legend(loc='best')
1503 savefig('rear_left_up_strut_fx_2_s5',dpi=300)
1504
1505 damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1506 print('Partial fatigue damage for rear_left_down_strut_fx_2_s6 calculated
    by script: %.5e (includes residual cycles)' %damage)
1507 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
1508
1509 figure()
1510 plot(t,s6/1.e6,'k-',label='')
1511 xlabel('time [s]')
1512 ylabel('rear_left_up_strut_fx_2_s6 [MPa]')
1513 title('Axial Stress')
1514 #axis([-16.,90.,-1.5,1.7])
1515 #legend(loc='best')
1516 savefig('rear_left_up_strut_fx_2_s6',dpi=300)
1517
1518 damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1519 print('Partial fatigue damage for rear_left_down_strut_fx_2_s7 calculated
    by script: %.5e (includes residual cycles)' %damage)
1520 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
1521
1522 figure()
1523 plot(t,s7/1.e6,'k-',label='')
1524 xlabel('time [s]')
1525 ylabel('rear_left_up_strut_fx_2_s7 [MPa]')
1526 title('Axial Stress')
1527 #axis([-16.,90.,-1.5,1.7])
1528 #legend(loc='best')
1529 savefig('rear_left_up_strut_fx_2_s7',dpi=300)
```

```

1530
1531 damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
1532 print('Partial fatigue damage for rear_left_down_strut_fx_2_s8 calculated
      by script: %.5e (includes residual cycles)' %damage)
1533 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFE))

1534
1535 figure()
1536 plot(t,s8/1.e6,'k-',label='')
1537 xlabel('time [s]')
1538 ylabel('rear_left_up_strut_fx_2_s8 [MPa] ')
1539 title('Axial Stress')
1540 #axis([-16.,90.,-1.5,1.7])
1541 #legend(loc='best')
1542 savefig('rear_left_up_strut_fx_2_s8',dpi=300)
1543
1544 #



---



1545
1546 #rear_right_down_strut_x1
1547 #Joint type 3
1548 #crown in z-dir
1549 rear_right_down_strut_fx_1 = a[:,70]*1.e3;
1550 rear_right_down_strut_fy_1 = a[:,126]*1.e3;
1551 rear_right_down_strut_fz_1 = a[:,127]*1.e3;
1552 mx_rrds_1 = a[:,71]*1.e3;
1553 my_rrds_1 = a[:,72]*1.e3;
1554 mz_rrds_1 = a[:,73]*1.e3;
1555
1556 [sax,sbendy,sbendz,s1,s2,s3,s4,s5,s6,s7,s8,ssh1,ssh2,ssh3,ssh4,ssh5,ssh6,
      ssh7,ssh8,svm1,svm2,svm3,svm4,svm5,svm6,svm7,svm8,sx,sy,sz] = \
1557 cyl_beam_stresses(t,rear_right_down_strut_fx_1,
      rear_right_down_strut_fy_1,rear_right_down_strut_fy_1,mx_rrds_1,
      my_rrds_1,mz_rrds_1,d,twall,SCF_AC_3,SCF_AS_3,SCF_IPB_3,SCF_OPB_3)
1558
1559 damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
1560 print('Partial fatigue damage for rear_right_down_strut_fx_1_s1 calculated
      by script: %.5e (includes residual cycles)' %damage)
1561 print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/
      damage*60*60*24*DFE))
1562 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFE))

1563
1564 figure()
1565 plot(t,s1/1.e6,'k-',label='')
1566 xlabel('time [s]')

```

```
1567 ylabel('rear_right_down_strut_fx_1_s1 [MPa] ')
1568 title('Axial Stress')
1569 #axis([-16.,90.,-1.5,1.7])
1570 #legend(loc='best')
1571 savefig('rear_right_down_strut_fx_1_s1',dpi=300)
1572
1573 damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1574 print('Partial fatigue damage for rear_right_down_strut_fx_1_s2 calculated
    by script: %.5e (includes residual cycles)' %damage)
1575 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
1576
1577 figure()
1578 plot(t,s2/1.e6,'k-',label='')
1579 xlabel('time [s]')
1580 ylabel('rear_right_down_strut_fx_1_s2 [MPa] ')
1581 title('Axial Stress')
1582 #axis([-16.,90.,-1.5,1.7])
1583 #legend(loc='best')
1584 savefig('rear_right_down_strut_fx_1_s2',dpi=300)
1585
1586 damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1587 print('Partial fatigue damage for rear_right_down_strut_fx_1_s3 calculated
    by script: %.5e (includes residual cycles)' %damage)
1588 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
1589
1590 figure()
1591 plot(t,s3/1.e6,'k-',label='')
1592 xlabel('time [s]')
1593 ylabel('rear_right_down_strut_fx_1_s3 [MPa] ')
1594 title('Axial Stress')
1595 #axis([-16.,90.,-1.5,1.7])
1596 #legend(loc='best')
1597 savefig('rear_right_down_strut_fx_1_s3',dpi=300)
1598
1599 damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1600 print('Partial fatigue damage for rear_right_down_strut_fx_1_s4 calculated
    by script: %.5e (includes residual cycles)' %damage)
1601 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
1602
1603 figure()
1604 plot(t,s4/1.e6,'k-',label='')
1605 xlabel('time [s]')
1606 ylabel('rear_right_down_strut_fx_1_s4 [MPa] ')

```

```
1607 title('Axial Stress')
1608 #axis([-16.,90.,-1.5,1.7])
1609 #legend(loc='best')
1610 savefig('rear_right_down_strut_fx_1_s4',dpi=300)
1611
1612 damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1613 print('Partial fatigue damage for rear_right_down_strut_fx_1_s5 calculated
    by script: %.5e (includes residual cycles)' %damage)
1614 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFF)))
1615
1616 figure()
1617 plot(t,s5/1.e6,'k-',label='')
1618 xlabel('time [s]')
1619 ylabel('rear_right_down_strut_fx_1_s5 [MPa]')
1620 title('Axial Stress')
1621 #axis([-16.,90.,-1.5,1.7])
1622 #legend(loc='best')
1623 savefig('rear_right_down_strut_fx_1_s5',dpi=300)
1624
1625 damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1626 print('Partial fatigue damage for rear_right_down_strut_fx_1_s6 calculated
    by script: %.5e (includes residual cycles)' %damage)
1627 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFF)))
1628
1629 figure()
1630 plot(t,s6/1.e6,'k-',label='')
1631 xlabel('time [s]')
1632 ylabel('rear_right_down_strut_fx_1_s6 [MPa]')
1633 title('Axial Stress')
1634 #axis([-16.,90.,-1.5,1.7])
1635 #legend(loc='best')
1636 savefig('rear_right_down_strut_fx_1_s6',dpi=300)
1637
1638 damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1639 print('Partial fatigue damage for rear_right_down_strut_fx_1_s7 calculated
    by script: %.5e (includes residual cycles)' %damage)
1640 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFF)))
1641
1642 figure()
1643 plot(t,s7/1.e6,'k-',label='')
1644 xlabel('time [s]')
1645 ylabel('rear_right_down_strut_fx_1_s7 [MPa]')
1646 title('Axial Stress')
```

```

1647 #axis([-16.,90.,-1.5,1.7])
1648 #legend(loc='best')
1649 savefig('rear_right_down_strut_fx_1_s7',dpi=300)
1650
1651 damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1652 print('Partial fatigue damage for rear_right_down_strut_fx_1_s8 calculated
    by script: %.5e (includes residual cycles)' %damage)
1653 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFE)))
1654
1655 figure()
1656 plot(t,s8/1.e6,'k-',label='')
1657 xlabel('time [s]')
1658 ylabel('rear_right_down_strut_fx_1_s8 [MPa]')
1659 title('Axial Stress')
1660 #axis([-16.,90.,-1.5,1.7])
1661 #legend(loc='best')
1662 savefig('rear_right_down_strut_fx_1_s8',dpi=300)
1663
1664 #

```

```

1665
1666 #rear_right_down_strut_x2
1667 #Joint type 4
1668 #crown in z-dir
1669 rear_right_down_strut_fx_2 = a[:,74]*1.e3;
1670 rear_right_down_strut_fy_2 = a[:,128]*1.e3;
1671 rear_right_down_strut_fz_2 = a[:,129]*1.e3;
1672 mx_rrds_2 = a[:,75]*1.e3;
1673 my_rrds_2 = a[:,76]*1.e3;
1674 mz_rrds_2 = a[:,77]*1.e3;
1675
1676 [sax,sbendy,sbendz,s1,s2,s3,s4,s5,s6,s7,s8,ssh1,ssh2,ssh3,ssh4,ssh5,ssh6,
    ssh7,ssh8,svm1,svm2,svm3,svm4,svm5,svm6,svm7,svm8,sx,sy,sz] = \
1677 cyl_beam_stresses(t,rear_right_down_strut_fx_2,
    rear_right_down_strut_fy_2,rear_right_down_strut_fz_2,mx_rrds_2,
    my_rrds_2,mz_rrds_2,d,twall,SCF_AC_4,SCF_AS_4,SCF_IPB_4,SCF_OPB_4)
1678
1679 damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1680 print('Partial fatigue damage for rear_right_down_strut_fx_2_s1 calculated
    by script: %.5e (includes residual cycles)' %damage)
1681 print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFE)))
1682 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFE)))
1683

```



```
1684 figure()
1685 plot(t,s1/1.e6,'k-',label='')
1686 xlabel('time [s]')
1687 ylabel('rear_right_down_strut_fx_2_s1 [MPa] ')
1688 title('Axial Stress')
1689 #axis([-16.,90.,-1.5,1.7])
1690 #legend(loc='best')
1691 savefig('rear_right_down_strut_fx_2_s1',dpi=300)
1692
1693 damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1694 print('Partial fatigue damage for rear_right_down_strut_fx_2_s2 calculated
    by script: %.5e (includes residual cycles)' %damage)
1695 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFE)))
1696
1697 figure()
1698 plot(t,s2/1.e6,'k-',label='')
1699 xlabel('time [s]')
1700 ylabel('rear_right_down_strut_fx_2_s2 [MPa] ')
1701 title('Axial Stress')
1702 #axis([-16.,90.,-1.5,1.7])
1703 #legend(loc='best')
1704 savefig('rear_right_down_strut_fx_2_s2',dpi=300)
1705
1706 damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1707 print('Partial fatigue damage for rear_right_down_strut_fx_2_s3 calculated
    by script: %.5e (includes residual cycles)' %damage)
1708 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFE)))
1709
1710 figure()
1711 plot(t,s3/1.e6,'k-',label='')
1712 xlabel('time [s]')
1713 ylabel('rear_right_down_strut_fx_2_s3 [MPa] ')
1714 title('Axial Stress')
1715 #axis([-16.,90.,-1.5,1.7])
1716 #legend(loc='best')
1717 savefig('rear_right_down_strut_fx_2_s3',dpi=300)
1718
1719 damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1720 print('Partial fatigue damage for rear_right_down_strut_fx_2_s4 calculated
    by script: %.5e (includes residual cycles)' %damage)
1721 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFE)))
1722
1723 figure()
```

```
1724 plot(t,s4/1.e6,'k-',label='')
1725 xlabel('time [s]')
1726 ylabel('rear_right_down_strut_fx_2_s4 [MPa] ')
1727 title('Axial Stress')
1728 #axis([-16.,90.,-1.5,1.7])
1729 #legend(loc='best')
1730 savefig('rear_right_down_strut_fx_2_s4',dpi=300)
1731
1732 damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1733 print('Partial fatigue damage for rear_right_down_strut_fx_2_s5 calculated
    by script: %.5e (includes residual cycles)' %damage)
1734 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFE)))
1735
1736 figure()
1737 plot(t,s5/1.e6,'k-',label='')
1738 xlabel('time [s]')
1739 ylabel('rear_right_down_strut_fx_2_s5 [MPa] ')
1740 title('Axial Stress')
1741 #axis([-16.,90.,-1.5,1.7])
1742 #legend(loc='best')
1743 savefig('rear_right_down_strut_fx_2_s5',dpi=300)
1744
1745 damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1746 print('Partial fatigue damage for rear_right_down_strut_fx_2_s6 calculated
    by script: %.5e (includes residual cycles)' %damage)
1747 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFE)))
1748
1749 figure()
1750 plot(t,s6/1.e6,'k-',label='')
1751 xlabel('time [s]')
1752 ylabel('rear_right_down_strut_fx_2_s6 [MPa] ')
1753 title('Axial Stress')
1754 #axis([-16.,90.,-1.5,1.7])
1755 #legend(loc='best')
1756 savefig('rear_right_down_strut_fx_2_s6',dpi=300)
1757
1758 damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1759 print('Partial fatigue damage for rear_right_down_strut_fx_2_s7 calculated
    by script: %.5e (includes residual cycles)' %damage)
1760 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFE)))
1761
1762 figure()
1763 plot(t,s7/1.e6,'k-',label='')
```

```

1764 xlabel('time [s]')
1765 ylabel('rear_right_down_strut_fx_2_s7 [MPa] ')
1766 title('Axial Stress')
1767 #axis([-16.,90.,-1.5,1.7])
1768 #legend(loc='best')
1769 savefig('rear_right_down_strut_fx_2_s7',dpi=300)
1770
1771 damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1772 print('Partial fatigue damage for rear_right_down_strut_fx_2_s8 calculated
    by script: %.5e (includes residual cycles)' %damage)
1773 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
1774
1775 figure()
1776 plot(t,s8/1.e6,'k-',label='')
1777 xlabel('time [s]')
1778 ylabel('rear_right_down_strut_fx_2_s8 [MPa] ')
1779 title('Axial Stress')
1780 #axis([-16.,90.,-1.5,1.7])
1781 #legend(loc='best')
1782 savefig('rear_right_down_strut_fx_2_s8',dpi=300)
1783
1784 #

```

```

1785
1786 #rear_right_up_strut_x1
1787 #Joint type 3
1788 #crown in z-dir
1789 rear_right_up_strut_fx_1 = a[:,78]*1.e3;
1790 rear_right_up_strut_fy_1 = a[:,130]*1.e3;
1791 rear_right_up_strut_fz_1 = a[:,131]*1.e3;
1792 mx_rrus_1 = a[:,79]*1.e3;
1793 my_rrus_1 = a[:,80]*1.e3;
1794 mz_rrus_1 = a[:,81]*1.e3;
1795
1796 [sax,sbendy,sbendz,s1,s2,s3,s4,s5,s6,s7,s8,ssh1,ssh2,ssh3,ssh4,ssh5,ssh6,
    ssh7,ssh8,svm1,svm2,svm3,svm4,svm5,svm6,svm7,svm8,sx,sy,sz] = \
1797     cyl_beam_stresses(t,rear_right_up_strut_fx_1,rear_right_up_strut_fy_1,
    rear_right_up_strut_fz_1,mx_rrus_1,my_rrus_1,mz_rrus_1,d,twall,
    SCF_AC_3,SCF_AS_3,SCF_IPB_3,SCF_OPB_3)
1798
1799 damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1800 print('Partial fatigue damage for rear_right_up_strut_fx_1_s1 calculated
    by script: %.5e (includes residual cycles)' %damage)
1801 print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/
    damage*60*60*24*DFE))

```

```

1802 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF))
1803
1804 figure()
1805 plot(t,s1/1.e6,'k-',label='')
1806 xlabel('time [s]')
1807 ylabel('rear_right_up_strut_fx_1_s1 [MPa] ')
1808 title('Axial Stress')
1809 #axis([-16.,90.,-1.5,1.7])
1810 #legend(loc='best')
1811 savefig('rear_right_up_strut_fx_1_s1',dpi=300)
1812
1813 damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
1814 print('Partial fatigue damage for rear_right_up_strut_fx_1_s2 calculated
        by script: %.5e (includes residual cycles)' %damage)
1815 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF))
1816
1817 figure()
1818 plot(t,s2/1.e6,'k-',label='')
1819 xlabel('time [s]')
1820 ylabel('rear_right_up_strut_fx_1_s2 [MPa] ')
1821 title('Axial Stress')
1822 #axis([-16.,90.,-1.5,1.7])
1823 #legend(loc='best')
1824 savefig('rear_right_up_strut_fx_1_s2',dpi=300)
1825
1826 damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
1827 print('Partial fatigue damage for rear_right_up_strut_fx_1_s3 calculated
        by script: %.5e (includes residual cycles)' %damage)
1828 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF))
1829
1830 figure()
1831 plot(t,s3/1.e6,'k-',label='')
1832 xlabel('time [s]')
1833 ylabel('rear_right_up_strut_fx_1_s3 [MPa] ')
1834 title('Axial Stress')
1835 #axis([-16.,90.,-1.5,1.7])
1836 #legend(loc='best')
1837 savefig('rear_right_up_strut_fx_1_s3',dpi=300)
1838
1839 damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
1840 print('Partial fatigue damage for rear_right_up_strut_fx_1_s4 calculated
        by script: %.5e (includes residual cycles)' %damage)
1841 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/

```

```

        damage*60*60*24*365*DFF))
1842
1843 figure()
1844 plot(t,s4/1.e6,'k-',label='')
1845 xlabel('time [s]')
1846 ylabel('rear_right_up_strut_fx_1_s4 [MPa]')
1847 title('Axial Stress')
1848 #axis([-16.,90.,-1.5,1.7])
1849 #legend(loc='best')
1850 savefig('rear_right_up_strut_fx_1_s4',dpi=300)
1851
1852 damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
1853 print('Partial fatigue damage for rear_right_up_strut_fx_1_s5 calculated
        by script: %.5e (includes residual cycles)' %damage)
1854 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF))
1855
1856 figure()
1857 plot(t,s5/1.e6,'k-',label='')
1858 xlabel('time [s]')
1859 ylabel('rear_right_up_strut_fx_1_s5 [MPa]')
1860 title('Axial Stress')
1861 #axis([-16.,90.,-1.5,1.7])
1862 #legend(loc='best')
1863 savefig('rear_right_up_strut_fx_1_s5',dpi=300)
1864
1865 damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
1866 print('Partial fatigue damage for rear_right_up_strut_fx_1_s6 calculated
        by script: %.5e (includes residual cycles)' %damage)
1867 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF))
1868
1869 figure()
1870 plot(t,s6/1.e6,'k-',label='')
1871 xlabel('time [s]')
1872 ylabel('rear_right_up_strut_fx_1_s6 [MPa]')
1873 title('Axial Stress')
1874 #axis([-16.,90.,-1.5,1.7])
1875 #legend(loc='best')
1876 savefig('rear_right_up_strut_fx_1_s6',dpi=300)
1877
1878 damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
1879 print('Partial fatigue damage for rear_right_up_strut_fx_1_s7 calculated
        by script: %.5e (includes residual cycles)' %damage)
1880 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF))

```

```

1881
1882 figure()
1883 plot(t,s7/1.e6,'k-',label='')
1884 xlabel('time [s]')
1885 ylabel('rear_right_up_strut_fx_1_s7 [MPa]')
1886 title('Axial Stress')
1887 #axis([-16.,90.,-1.5,1.7])
1888 #legend(loc='best')
1889 savefig('rear_right_up_strut_fx_1_s7',dpi=300)
1890
1891 damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
1892 print('Partial fatigue damage for rear_right_up_strut_fx_1_s8 calculated
      by script: %.5e (includes residual cycles)' %damage)
1893 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFE))
1894
1895 figure()
1896 plot(t,s8/1.e6,'k-',label='')
1897 xlabel('time [s]')
1898 ylabel('rear_right_up_strut_fx_1_s8 [MPa]')
1899 title('Axial Stress')
1900 #axis([-16.,90.,-1.5,1.7])
1901 #legend(loc='best')
1902 savefig('rear_right_up_strut_fx_1_s8',dpi=300)
1903
1904 #

```

```

1905
1906 #rear_right_up_strut_x2
1907 #Joint type 4
1908 #crown in z-dir
1909 rear_right_up_strut_fx_2 = a[:,82]*1.e3;
1910 rear_right_up_strut_fy_2 = a[:,132]*1.e3;
1911 rear_right_up_strut_fz_2 = a[:,133]*1.e3;
1912 mx_rrus_2 = a[:,83]*1.e3;
1913 my_rrus_2 = a[:,84]*1.e3;
1914 mz_rrus_2 = a[:,85]*1.e3;
1915
1916 [sax,sbendy,sbendz,s1,s2,s3,s4,s5,s6,s7,s8,ssh1,ssh2,ssh3,ssh4,ssh5,ssh6,
      ssh7,ssh8,svm1,svm2,svm3,svm4,svm5,svm6,svm7,svm8,sx,sy,sz] = \
1917     cyl_beam_stresses(t,rear_right_up_strut_fx_2,rear_right_up_strut_fy_2,
      rear_right_up_strut_fz_2,mx_rrus_2,my_rrus_2,mz_rrus_2,d,twall,
      SCF_AC_4,SCF_AS_4,SCF_IPB_4,SCF_OPB_4)
1918
1919 damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
1920 print('Partial fatigue damage for rear_right_up_strut_fx_2_s1 calculated

```

```
    by script: %.5e (includes residual cycles)' %damage)
1921 print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFF)))
1922 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))
1923
1924 figure()
1925 plot(t,s1/1.e6,'k-',label='')
1926 xlabel('time [s]')
1927 ylabel('rear_right_up_strut_fx_2_s1 [MPa] ')
1928 title('Axial Stress')
1929 #axis([-16.,90.,-1.5,1.7])
1930 #legend(loc='best')
1931 savefig('rear_right_up_strut_fx_2_s1',dpi=300)
1932
1933 damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1934 print('Partial fatigue damage for rear_right_up_strut_fx_2_s2 calculated
    by script: %.5e (includes residual cycles)' %damage)
1935 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))
1936
1937 figure()
1938 plot(t,s2/1.e6,'k-',label='')
1939 xlabel('time [s]')
1940 ylabel('rear_right_up_strut_fx_2_s2 [MPa] ')
1941 title('Axial Stress')
1942 #axis([-16.,90.,-1.5,1.7])
1943 #legend(loc='best')
1944 savefig('rear_right_up_strut_fx_2_s2',dpi=300)
1945
1946 damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
1947 print('Partial fatigue damage for rear_right_up_strut_fx_2_s3 calculated
    by script: %.5e (includes residual cycles)' %damage)
1948 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))
1949
1950 figure()
1951 plot(t,s3/1.e6,'k-',label='')
1952 xlabel('time [s]')
1953 ylabel('rear_right_up_strut_fx_2_s3 [MPa] ')
1954 title('Axial Stress')
1955 #axis([-16.,90.,-1.5,1.7])
1956 #legend(loc='best')
1957 savefig('rear_right_up_strut_fx_2_s3',dpi=300)
1958
1959 damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
```

```
1960 print('Partial fatigue damage for rear_right_up_strut_fx_2_s4 calculated
      by script: %.5e (includes residual cycles)' %damage)
1961 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFF))
1962
1963 figure()
1964 plot(t,s4/1.e6,'k-',label='')
1965 xlabel('time [s]')
1966 ylabel('rear_right_up_strut_fx_2_s4 [MPa]')
1967 title('Axial Stress')
1968 #axis([-16.,90.,-1.5,1.7])
1969 #legend(loc='best')
1970 savefig('rear_right_up_strut_fx_2_s4',dpi=300)
1971
1972 damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
1973 print('Partial fatigue damage for rear_right_up_strut_fx_2_s5 calculated
      by script: %.5e (includes residual cycles)' %damage)
1974 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFF))
1975
1976 figure()
1977 plot(t,s5/1.e6,'k-',label='')
1978 xlabel('time [s]')
1979 ylabel('rear_right_up_strut_fx_2_s5 [MPa]')
1980 title('Axial Stress')
1981 #axis([-16.,90.,-1.5,1.7])
1982 #legend(loc='best')
1983 savefig('rear_right_up_strut_fx_2_s5',dpi=300)
1984
1985 damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
1986 print('Partial fatigue damage for rear_right_up_strut_fx_2_s6 calculated
      by script: %.5e (includes residual cycles)' %damage)
1987 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFF))
1988
1989 figure()
1990 plot(t,s6/1.e6,'k-',label='')
1991 xlabel('time [s]')
1992 ylabel('rear_right_up_strut_fx_2_s6 [MPa]')
1993 title('Axial Stress')
1994 #axis([-16.,90.,-1.5,1.7])
1995 #legend(loc='best')
1996 savefig('rear_right_up_strut_fx_2_s6',dpi=300)
1997
1998 damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
1999 print('Partial fatigue damage for rear_right_up_strut_fx_2_s7 calculated
```



```

    by script: %.5e (includes residual cycles)' %damage)
2000 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFE))
2001
2002 figure()
2003 plot(t,s7/1.e6,'k-',label='')
2004 xlabel('time [s]')
2005 ylabel('rear_right_up_strut_fx_2_s7 [MPa] ')
2006 title('Axial Stress')
2007 #axis([-16.,90.,-1.5,1.7])
2008 #legend(loc='best')
2009 savefig('rear_right_up_strut_fx_2_s7',dpi=300)
2010
2011 damage = fatiguedamage_twslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
2012 print('Partial fatigue damage for rear_right_up_strut_fx_2_s8 calculated
        by script: %.5e (includes residual cycles)' %damage)
2013 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFE))
2014
2015 figure()
2016 plot(t,s8/1.e6,'k-',label='')
2017 xlabel('time [s]')
2018 ylabel('rear_right_up_strut_fx_2_s8 [MPa] ')
2019 title('Axial Stress')
2020 #axis([-16.,90.,-1.5,1.7])
2021 #legend(loc='best')
2022 savefig('rear_right_up_strut_fx_2_s8',dpi=300)
2023
2024 #

```

```

2025
2026 #EXCLUDE?
2027
2028 #rear_vert_down_strut_x1
2029
2030 #N/A
2031 rear_vert_down_strut_fx_1 = a[:,86]*1.e3;
2032 rear_vert_down_strut_fy_1 = a[:,134]*1.e3;
2033 rear_vert_down_strut_fz_1 = a[:,135]*1.e3;
2034 mx_rvds_1 = a[:,87]*1.e3;
2035 my_rvds_1 = a[:,88]*1.e3;
2036 mz_rvds_1 = a[:,89]*1.e3;
2037
2038 [sax,sbendy,sbendz,s1,s2,s3,s4,s5,s6,s7,s8,ssh1,ssh2,ssh3,ssh4,ssh5,ssh6,
        ssh7,ssh8,svm1,svm2,svm3,svm4,svm5,svm6,svm7,svm8,sx,sy,sz] = \
2039     cyl_beam_stresses(t,rear_vert_down_strut_fx_1,
        rear_vert_down_strut_fy_1,rear_vert_down_strut_fz_1,mx_rvds_1,

```

```

2040         my_rvds_1,mz_rvds_1,d,twall,1,1,1,1)
2041 damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
2042     k=0.25)
2043 print('Partial fatigue damage for rear_vert_down_strut_fx_1_s1 calculated
2044     by script: %.5e (includes residual cycles)' %damage)
2045 print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/
2046     (damage*60*60*24*DFF)))
2047 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
2048     (damage*60*60*24*365*DFF)))
2049
2050 figure()
2051 plot(t,s1/1.e6,'k-',label='')
2052 xlabel('time [s]')
2053 ylabel('rear_vert_down_strut_fx_1_s1 [MPa] ')
2054 title('Axial Stress')
2055 #axis([-16.,90.,-1.5,1.7])
2056 #legend(loc='best')
2057 savefig('rear_vert_down_strut_fx_1_s1',dpi=300)
2058
2059 damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
2060     k=0.25)
2061 print('Partial fatigue damage for rear_vert_down_strut_fx_1_s2 calculated
2062     by script: %.5e (includes residual cycles)' %damage)
2063 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
2064     (damage*60*60*24*365*DFF)))
2065
2066 figure()
2067 plot(t,s2/1.e6,'k-',label='')
2068 xlabel('time [s]')
2069 ylabel('rear_vert_down_strut_fx_1_s2 [MPa] ')
2070 title('Axial Stress')
2071 #axis([-16.,90.,-1.5,1.7])
2072 #legend(loc='best')
2073 savefig('rear_vert_down_strut_fx_1_s2',dpi=300)
2074
2075 damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
2076     k=0.25)
2077 print('Partial fatigue damage for rear_vert_down_strut_fx_1_s3 calculated
2078     by script: %.5e (includes residual cycles)' %damage)
2079 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
2080     (damage*60*60*24*365*DFF)))
2081
2082 figure()
2083 plot(t,s3/1.e6,'k-',label='')
2084 xlabel('time [s]')
2085 ylabel('rear_vert_down_strut_fx_1_s3 [MPa] ')
2086 title('Axial Stress')
2087 #axis([-16.,90.,-1.5,1.7])

```

```
2078 #legend(loc='best')
2079 savefig('rear_vert_down_strut_fx_1_s3',dpi=300)
2080
2081 damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
2082 print('Partial fatigue damage for rear_vert_down_strut_fx_1_s4 calculated
    by script: %.5e (includes residual cycles)' %damage)
2083 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
2084
2085 figure()
2086 plot(t,s4/1.e6,'k-',label='')
2087 xlabel('time [s]')
2088 ylabel('rear_vert_down_strut_fx_1_s4 [MPa]')
2089 title('Axial Stress')
2090 #axis([-16.,90.,-1.5,1.7])
2091 #legend(loc='best')
2092 savefig('rear_vert_down_strut_fx_1_s4',dpi=300)
2093
2094 damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
2095 print('Partial fatigue damage for rear_vert_down_strut_fx_1_s5 calculated
    by script: %.5e (includes residual cycles)' %damage)
2096 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
2097
2098 figure()
2099 plot(t,s5/1.e6,'k-',label='')
2100 xlabel('time [s]')
2101 ylabel('rear_vert_down_strut_fx_1_s5 [MPa]')
2102 title('Axial Stress')
2103 #axis([-16.,90.,-1.5,1.7])
2104 #legend(loc='best')
2105 savefig('rear_vert_down_strut_fx_1_s5',dpi=300)
2106
2107 damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
2108 print('Partial fatigue damage for rear_vert_down_strut_fx_1_s6 calculated
    by script: %.5e (includes residual cycles)' %damage)
2109 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
2110
2111 figure()
2112 plot(t,s6/1.e6,'k-',label='')
2113 xlabel('time [s]')
2114 ylabel('rear_vert_down_strut_fx_1_s6 [MPa]')
2115 title('Axial Stress')
2116 #axis([-16.,90.,-1.5,1.7])
2117 #legend(loc='best')
```

```

2118 savefig('rear_vert_down_strut_fx_1_s6',dpi=300)
2119
2120 damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
2121 print('Partial fatigue damage for rear_vert_down_strut_fx_1_s7 calculated
    by script: %.5e (includes residual cycles)' %damage)
2122 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFF))
2123
2124 figure()
2125 plot(t,s7/1.e6,'k-',label='')
2126 xlabel('time [s]')
2127 ylabel('rear_vert_down_strut_fx_1_s7 [MPa] ')
2128 title('Axial Stress')
2129 #axis([-16.,90.,-1.5,1.7])
2130 #legend(loc='best')
2131 savefig('rear_vert_down_strut_fx_1_s7',dpi=300)
2132
2133 damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
2134 print('Partial fatigue damage for rear_vert_down_strut_fx_1_s8 calculated
    by script: %.5e (includes residual cycles)' %damage)
2135 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFF))
2136
2137 figure()
2138 plot(t,s8/1.e6,'k-',label='')
2139 xlabel('time [s]')
2140 ylabel('rear_vert_down_strut_fx_1_s8 [MPa] ')
2141 title('Axial Stress')
2142 #axis([-16.,90.,-1.5,1.7])
2143 #legend(loc='best')
2144 savefig('rear_vert_down_strut_fx_1_s8',dpi=300)
2145
2146 #

```

```

2147
2148 #rear_vert_down_strut_x2
2149 #Joint type 2
2150 #saddle in z-dir
2151 rear_vert_down_strut_fx_2 = a[:,90]*1.e3;
2152 rear_vert_down_strut_fy_2 = a[:,136]*1.e3;
2153 rear_vert_down_strut_fz_2 = a[:,137]*1.e3;
2154 mx_rvds_2 = a[:,91]*1.e3;
2155 my_rvds_2 = a[:,92]*1.e3;
2156 mz_rvds_2 = a[:,93]*1.e3;
2157
2158 [sax,sbendy,sbendz,s1,s2,s3,s4,s5,s6,s7,s8,ssh1,ssh2,ssh3,ssh4,ssh5,ssh6,

```

```

2159     ssh7 , ssh8 , svm1 , svm2 , svm3 , svm4 , svm5 , svm6 , svm7 , svm8 , sx , sy , sz ] = \
        cyl_beam_stresses(t , rear_vert_down_strut_fx_2 ,
            rear_vert_down_strut_fy_2 , rear_vert_down_strut_fz_2 , mx_rvds_2 ,
            my_rvds_2 , mz_rvds_2 , d , twall , SCF_AS_2 , SCF_AC_2 , SCF_OPB_2 , SCF_IPB_2)
2160
2161 damage = fatiguedamage_tvoslope(t , s1 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
        k=0.25)
2162 print('Partial fatigue damage for rear_vert_down_strut_fx_2_s1 calculated
        by script: %.5e (includes residual cycles)' %damage)
2163 print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
        damage*60*60*24*DFE)))
2164 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
        damage*60*60*24*365*DFE)))
2165
2166 figure()
2167 plot(t , s1/1.e6 , 'k-' , label='')
2168 xlabel('time [s]')
2169 ylabel('rear_vert_down_strut_fx_2_s1 [MPa] ')
2170 title('Axial Stress')
2171 #axis([-16. , 90. , -1.5 , 1.7])
2172 #legend(loc='best')
2173 savefig('rear_vert_down_strut_fx_2_s1' , dpi=300)
2174
2175 damage = fatiguedamage_tvoslope(t , s2 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
        k=0.25)
2176 print('Partial fatigue damage for rear_vert_down_strut_fx_2_s2 calculated
        by script: %.5e (includes residual cycles)' %damage)
2177 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
        damage*60*60*24*365*DFE)))
2178
2179 figure()
2180 plot(t , s2/1.e6 , 'k-' , label='')
2181 xlabel('time [s]')
2182 ylabel('rear_vert_down_strut_fx_2_s2 [MPa] ')
2183 title('Axial Stress')
2184 #axis([-16. , 90. , -1.5 , 1.7])
2185 #legend(loc='best')
2186 savefig('rear_vert_down_strut_fx_2_s2' , dpi=300)
2187
2188 damage = fatiguedamage_tvoslope(t , s3 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
        k=0.25)
2189 print('Partial fatigue damage for rear_vert_down_strut_fx_2_s3 calculated
        by script: %.5e (includes residual cycles)' %damage)
2190 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
        damage*60*60*24*365*DFE)))
2191
2192 figure()
2193 plot(t , s3/1.e6 , 'k-' , label='')
2194 xlabel('time [s]')

```

```
2195 ylabel('rear_vert_down_strut_fx_2_s3 [MPa] ')
2196 title('Axial Stress')
2197 #axis([-16., 90., -1.5, 1.7])
2198 #legend(loc='best')
2199 savefig('rear_vert_down_strut_fx_2_s3', dpi=300)
2200
2201 damage = fatiguedamage_twoslope(t, s4, m1, loga1, m2, loga2, Nlim, th, tref=25E-3,
2202                                 k=0.25)
2203 print('Partial fatigue damage for rear_vert_down_strut_fx_2_s4 calculated
2204       by script: %.5e (includes residual cycles)' %damage)
2205 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
2206       damage*60*60*24*365*DFE)))
2207
2208 figure()
2209 plot(t, s4/1.e6, 'k-', label='')
2210 xlabel('time [s]')
2211 ylabel('rear_vert_down_strut_fx_2_s4 [MPa] ')
2212 title('Axial Stress')
2213 #axis([-16., 90., -1.5, 1.7])
2214 #legend(loc='best')
2215 savefig('rear_vert_down_strut_fx_2_s4', dpi=300)
2216
2217 damage = fatiguedamage_twoslope(t, s5, m1, loga1, m2, loga2, Nlim, th, tref=25E-3,
2218                                 k=0.25)
2219 print('Partial fatigue damage for rear_vert_down_strut_fx_2_s5 calculated
2220       by script: %.5e (includes residual cycles)' %damage)
2221 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
2222       damage*60*60*24*365*DFE)))
2223
2224 figure()
2225 plot(t, s5/1.e6, 'k-', label='')
2226 xlabel('time [s]')
2227 ylabel('rear_vert_down_strut_fx_2_s5 [MPa] ')
2228 title('Axial Stress')
2229 #axis([-16., 90., -1.5, 1.7])
2230 #legend(loc='best')
2231 savefig('rear_vert_down_strut_fx_2_s5', dpi=300)
2232
2233 damage = fatiguedamage_twoslope(t, s6, m1, loga1, m2, loga2, Nlim, th, tref=25E-3,
2234                                 k=0.25)
2235 print('Partial fatigue damage for rear_vert_down_strut_fx_2_s6 calculated
2236       by script: %.5e (includes residual cycles)' %damage)
2237 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
2238       damage*60*60*24*365*DFE)))
2239
2240 figure()
2241 plot(t, s6/1.e6, 'k-', label='')
2242 xlabel('time [s]')
2243 ylabel('rear_vert_down_strut_fx_2_s6 [MPa] ')
2244
```

```
2235 title('Axial Stress')
2236 #axis([-16.,90.,-1.5,1.7])
2237 #legend(loc='best')
2238 savefig('rear_vert_down_strut_fx_2_s6',dpi=300)
2239
2240 damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
2241 print('Partial fatigue damage for rear_vert_down_strut_fx_2_s7 calculated
    by script: %.5e (includes residual cycles)' %damage)
2242 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFE)))
2243
2244 figure()
2245 plot(t,s7/1.e6,'k-',label='')
2246 xlabel('time [s]')
2247 ylabel('rear_vert_down_strut_fx_2_s7 [MPa]')
2248 title('Axial Stress')
2249 #axis([-16.,90.,-1.5,1.7])
2250 #legend(loc='best')
2251 savefig('rear_vert_down_strut_fx_2_s7',dpi=300)
2252
2253 damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
2254 print('Partial fatigue damage for rear_vert_down_strut_fx_2_s8 calculated
    by script: %.5e (includes residual cycles)' %damage)
2255 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFE)))
2256
2257 figure()
2258 plot(t,s8/1.e6,'k-',label='')
2259 xlabel('time [s]')
2260 ylabel('rear_vert_down_strut_fx_2_s8 [MPa]')
2261 title('Axial Stress')
2262 #axis([-16.,90.,-1.5,1.7])
2263 #legend(loc='best')
2264 savefig('rear_vert_down_strut_fx_2_s8',dpi=300)
2265
2266 #

```

```
2267
2268 #EXCLUDE?
2269
2270 #rear_vert_up_strut_x1
2271
2272 #N/A
2273 rear_vert_up_strut_fx_1 = a[:,94]*1.e3;
2274 rear_vert_up_strut_fy_1 = a[:,138]*1.e3;
2275 rear_vert_up_strut_fz_1 = a[:,139]*1.e3;
```

```

2276 mx_rvus_1 = a[:,95]*1.e3;
2277 my_rvus_1 = a[:,96]*1.e3;
2278 mz_rvus_1 = a[:,97]*1.e3;
2279
2280 [sax ,sbendy ,sbendz ,s1 ,s2 ,s3 ,s4 ,s5 ,s6 ,s7 ,s8 ,ssh1 , ssh2 , ssh3 , ssh4 , ssh5 , ssh6 ,
      ssh7 , ssh8 ,svm1 ,svm2 ,svm3 ,svm4 ,svm5 ,svm6 ,svm7 ,svm8 ,sx ,sy , sz ] = \
2281     cyl_beam_stresses(t ,rear_vert_up_strut_fx_1 ,rear_vert_up_strut_fy_1 ,
      rear_vert_up_strut_fz_1 ,mx_rvus_1 ,my_rvus_1 ,mz_rvus_1 ,d ,twall
      ,1 ,1 ,1 ,1)
2282
2283 damage = fatiguedamage_twoslope(t ,s1 ,m1 ,loga1 ,m2 ,loga2 ,Nlim ,th ,tref=25E-3,
      k=0.25)
2284 print('Partial fatigue damage for rear_vert_up_strut_fx_1_s1 calculated by
      script: %.5e (includes residual cycles)' %damage)
2285 print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
      damage*60*60*24*DFE)))
2286 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
      damage*60*60*24*365*DFE)))
2287
2288 figure()
2289 plot(t ,s1 /1.e6 , 'k-' ,label='')
2290 xlabel('time [s]')
2291 ylabel('rear_vert_up_strut_fx_1_s1 [MPa] ')
2292 title('Axial Stress')
2293 #axis([-16. ,90. , -1.5 ,1.7])
2294 #legend(loc='best')
2295 savefig('rear_vert_up_strut_fx_1_s1' ,dpi=300)
2296
2297 damage = fatiguedamage_twoslope(t ,s2 ,m1 ,loga1 ,m2 ,loga2 ,Nlim ,th ,tref=25E-3,
      k=0.25)
2298 print('Partial fatigue damage for rear_vert_up_strut_fx_1_s2 calculated by
      script: %.5e (includes residual cycles)' %damage)
2299 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
      damage*60*60*24*365*DFE)))
2300
2301 figure()
2302 plot(t ,s2 /1.e6 , 'k-' ,label='')
2303 xlabel('time [s]')
2304 ylabel('rear_vert_up_strut_fx_1_s2 [MPa] ')
2305 title('Axial Stress')
2306 #axis([-16. ,90. , -1.5 ,1.7])
2307 #legend(loc='best')
2308 savefig('rear_vert_up_strut_fx_1_s2' ,dpi=300)
2309
2310 damage = fatiguedamage_twoslope(t ,s3 ,m1 ,loga1 ,m2 ,loga2 ,Nlim ,th ,tref=25E-3,
      k=0.25)
2311 print('Partial fatigue damage for rear_vert_up_strut_fx_1_s3 calculated by
      script: %.5e (includes residual cycles)' %damage)
2312 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(

```



```

        damage*60*60*24*365*DFF)))
2313
2314 figure()
2315 plot(t,s3/1.e6,'k-',label='')
2316 xlabel('time [s]')
2317 ylabel('rear_vert_up_strut_fx_1_s3 [MPa]')
2318 title('Axial Stress')
2319 #axis([-16.,90.,-1.5,1.7])
2320 #legend(loc='best')
2321 savefig('rear_vert_up_strut_fx_1_s3',dpi=300)
2322
2323 damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
2324 print('Partial fatigue damage for rear_vert_up_strut_fx_1_s4 calculated by
        script: %.5e (includes residual cycles)' %damage)
2325 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF)))
2326
2327 figure()
2328 plot(t,s4/1.e6,'k-',label='')
2329 xlabel('time [s]')
2330 ylabel('rear_vert_up_strut_fx_1_s4 [MPa]')
2331 title('Axial Stress')
2332 #axis([-16.,90.,-1.5,1.7])
2333 #legend(loc='best')
2334 savefig('rear_vert_up_strut_fx_1_s4',dpi=300)
2335
2336 damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
2337 print('Partial fatigue damage for rear_vert_up_strut_fx_1_s5 calculated by
        script: %.5e (includes residual cycles)' %damage)
2338 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF)))
2339
2340 figure()
2341 plot(t,s5/1.e6,'k-',label='')
2342 xlabel('time [s]')
2343 ylabel('rear_vert_up_strut_fx_1_s5 [MPa]')
2344 title('Axial Stress')
2345 #axis([-16.,90.,-1.5,1.7])
2346 #legend(loc='best')
2347 savefig('rear_vert_up_strut_fx_1_s5',dpi=300)
2348
2349 damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
2350 print('Partial fatigue damage for rear_vert_up_strut_fx_1_s6 calculated by
        script: %.5e (includes residual cycles)' %damage)
2351 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF)))

```

```
2352
2353 figure()
2354 plot(t,s6/1.e6,'k-',label='')
2355 xlabel('time [s]')
2356 ylabel('rear_vert_up_strut_fx_1_s6 [MPa] ')
2357 title('Axial Stress')
2358 #axis([-16.,90.,-1.5,1.7])
2359 #legend(loc='best')
2360 savefig('rear_vert_up_strut_fx_1_s6',dpi=300)
2361
2362 damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
2363 print('Partial fatigue damage for rear_vert_up_strut_fx_1_s7 calculated by
    script: %.5e (includes residual cycles)' %damage)
2364 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFE)))
2365
2366 figure()
2367 plot(t,s7/1.e6,'k-',label='')
2368 xlabel('time [s]')
2369 ylabel('rear_vert_up_strut_fx_1_s7 [MPa] ')
2370 title('Axial Stress')
2371 #axis([-16.,90.,-1.5,1.7])
2372 #legend(loc='best')
2373 savefig('rear_vert_up_strut_fx_1_s7',dpi=300)
2374
2375 damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
2376 print('Partial fatigue damage for rear_vert_up_strut_fx_1_s8 calculated by
    script: %.5e (includes residual cycles)' %damage)
2377 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFE)))
2378
2379 figure()
2380 plot(t,s8/1.e6,'k-',label='')
2381 xlabel('time [s]')
2382 ylabel('rear_vert_up_strut_fx_1_s8 [MPa] ')
2383 title('Axial Stress')
2384 #axis([-16.,90.,-1.5,1.7])
2385 #legend(loc='best')
2386 savefig('rear_vert_up_strut_fx_1_s8',dpi=300)
2387
2388 #


---


2389
2390 #rear_vert_up_strut_x2
2391 #Joint type 2
2392 #saddle in z-dir
```

```

2393 rear_vert_up_strut_fx_2 = a[:,98]*1.e3;
2394 rear_vert_up_strut_fy_2 = a[:,140]*1.e3;
2395 rear_vert_up_strut_fz_2 = a[:,141]*1.e3;
2396 mx_rvus_2 = a[:,99]*1.e3;
2397 my_rvus_2 = a[:,100]*1.e3;
2398 mz_rvus_2 = a[:,101]*1.e3;
2399
2400 [sax , sbendy , sbendz , s1 , s2 , s3 , s4 , s5 , s6 , s7 , s8 , ssh1 , ssh2 , ssh3 , ssh4 , ssh5 , ssh6 ,
      ssh7 , ssh8 , svm1 , svm2 , svm3 , svm4 , svm5 , svm6 , svm7 , svm8 , sx , sy , sz ] = \
2401     cyl_beam_stresses(t , rear_vert_up_strut_fx_2 , rear_vert_up_strut_fy_2 ,
      rear_vert_up_strut_fz_2 , mx_rvus_2 , my_rvus_2 , mz_rvus_2 , d , twall ,
      SCF_AS_2 , SCF_AC_2 , SCF_OPB_2 , SCF_IPB_2)
2402
2403 damage = fatiguedamage_twslope(t , s1 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
      k=0.25)
2404 print('Partial fatigue damage for rear_vert_up_strut_fx_2_s1 calculated by
      script: %.5e (includes residual cycles)' %damage)
2405 print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/
      damage*60*60*24*DFE))
2406 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFE))
2407
2408 figure()
2409 plot(t , s1/1.e6 , 'k-' , label='')
2410 xlabel('time [s]')
2411 ylabel('rear_vert_up_strut_fx_2_s1 [MPa] ')
2412 title('Axial Stress')
2413 #axis([-16. , 90. , -1.5 , 1.7])
2414 #legend(loc='best')
2415 savefig('rear_vert_up_strut_fx_2_s1' , dpi=300)
2416
2417 damage = fatiguedamage_twslope(t , s2 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
      k=0.25)
2418 print('Partial fatigue damage for rear_vert_up_strut_fx_2_s2 calculated by
      script: %.5e (includes residual cycles)' %damage)
2419 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFE))
2420
2421 figure()
2422 plot(t , s2/1.e6 , 'k-' , label='')
2423 xlabel('time [s]')
2424 ylabel('rear_vert_up_strut_fx_2_s2 [MPa] ')
2425 title('Axial Stress')
2426 #axis([-16. , 90. , -1.5 , 1.7])
2427 #legend(loc='best')
2428 savefig('rear_vert_up_strut_fx_2_s2' , dpi=300)
2429
2430 damage = fatiguedamage_twslope(t , s3 , m1 , loga1 , m2 , loga2 , Nlim , th , tref=25E-3,
      k=0.25)

```

```
2431 print('Partial fatigue damage for rear_vert_up_strut_fx_2_s3 calculated by
      script: %.5e (includes residual cycles)' %damage)
2432 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFF))
2433
2434 figure()
2435 plot(t,s3/1.e6,'k-',label='')
2436 xlabel('time [s]')
2437 ylabel('rear_vert_up_strut_fx_2_s3 [MPa]')
2438 title('Axial Stress')
2439 #axis([-16.,90.,-1.5,1.7])
2440 #legend(loc='best')
2441 savefig('rear_vert_up_strut_fx_2_s3',dpi=300)
2442
2443 damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
2444 print('Partial fatigue damage for rear_vert_up_strut_fx_2_s4 calculated by
      script: %.5e (includes residual cycles)' %damage)
2445 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFF))
2446
2447 figure()
2448 plot(t,s4/1.e6,'k-',label='')
2449 xlabel('time [s]')
2450 ylabel('rear_vert_up_strut_fx_2_s4 [MPa]')
2451 title('Axial Stress')
2452 #axis([-16.,90.,-1.5,1.7])
2453 #legend(loc='best')
2454 savefig('rear_vert_up_strut_fx_2_s4',dpi=300)
2455
2456 damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
2457 print('Partial fatigue damage for rear_vert_up_strut_fx_2_s5 calculated by
      script: %.5e (includes residual cycles)' %damage)
2458 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFF))
2459
2460 figure()
2461 plot(t,s5/1.e6,'k-',label='')
2462 xlabel('time [s]')
2463 ylabel('rear_vert_up_strut_fx_2_s5 [MPa]')
2464 title('Axial Stress')
2465 #axis([-16.,90.,-1.5,1.7])
2466 #legend(loc='best')
2467 savefig('rear_vert_up_strut_fx_2_s5',dpi=300)
2468
2469 damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
      k=0.25)
2470 print('Partial fatigue damage for rear_vert_up_strut_fx_2_s6 calculated by
```

```
        script: %.5e (includes residual cycles)' %damage)
2471 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF))
2472
2473 figure()
2474 plot(t,s6/1.e6,'k-',label='')
2475 xlabel('time [s]')
2476 ylabel('rear_vert_up_strut_fx_2_s6 [MPa] ')
2477 title('Axial Stress')
2478 #axis([-16.,90.,-1.5,1.7])
2479 #legend(loc='best')
2480 savefig('rear_vert_up_strut_fx_2_s6',dpi=300)
2481
2482 damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
2483 print('Partial fatigue damage for rear_vert_up_strut_fx_2_s7 calculated by
        script: %.5e (includes residual cycles)' %damage)
2484 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF))
2485
2486 figure()
2487 plot(t,s7/1.e6,'k-',label='')
2488 xlabel('time [s]')
2489 ylabel('rear_vert_up_strut_fx_2_s7 [MPa] ')
2490 title('Axial Stress')
2491 #axis([-16.,90.,-1.5,1.7])
2492 #legend(loc='best')
2493 savefig('rear_vert_up_strut_fx_2_s7',dpi=300)
2494
2495 damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
        k=0.25)
2496 print('Partial fatigue damage for rear_vert_up_strut_fx_2_s8 calculated by
        script: %.5e (includes residual cycles)' %damage)
2497 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
        damage*60*60*24*365*DFF))
2498
2499 figure()
2500 plot(t,s8/1.e6,'k-',label='')
2501 xlabel('time [s]')
2502 ylabel('rear_vert_up_strut_fx_2_s8 [MPa] ')
2503 title('Axial Stress')
2504 #axis([-16.,90.,-1.5,1.7])
2505 #legend(loc='best')
2506 savefig('rear_vert_up_strut_fx_2_s8',dpi=300)
2507
2508 #
```

2509

```

2510 #Shaft
2511
2512 D_shaft = 2
2513 T_shaft = 0.001
2514
2515
2516 shaft_fx = a[:,10]*1.e3;
2517 shaft_fy = a[:,11]*1.e3;
2518 shaft_fz = a[:,12]*1.e3;
2519 shaft_mx = a[:,13]*1.e3;
2520 shaft_my = a[:,14]*1.e3;
2521 shaft_mz = a[:,15]*1.e3;
2522
2523 [sax , sbendy , sbendz , s1 , s2 , s3 , s4 , s5 , s6 , s7 , s8 , ssh1 , ssh2 , ssh3 , ssh4 , ssh5 , ssh6 ,
      ssh7 , ssh8 , svm1 , svm2 , svm3 , svm4 , svm5 , svm6 , svm7 , svm8 , sx , sy , sz ] = \
2524     cyl_beam_stresses(t , shaft_fx , shaft_fy , shaft_fz , shaft_mx , shaft_my ,
      shaft_mz , D_shaft , T_shaft , 1 , 1 , 1 , 1)
2525
2526 damage = fatiguedamage_twoslope(t , s1 , m1_st , loga1_st , m2_st , loga2_st , Nlim ,
      th_st , tref=25E-3 , k=0.25)
2527 print('Partial fatigue damage for shaft_s1 calculated by script: %.5e (
      includes residual cycles)' %damage)
2528 print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
      damage*60*60*24*DFE)))
2529 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
      damage*60*60*24*365*DFE)))
2530
2531 figure()
2532 plot(t , s1/1.e6 , 'k-' , label='')
2533 xlabel('time [s]')
2534 ylabel('shaft_s1 [MPa] ')
2535 title('Axial Stress')
2536 #axis([-16. , 90. , -1.5 , 1.7])
2537 #legend(loc='best')
2538 savefig('shaft_s1_s1' , dpi=300)
2539
2540 damage = fatiguedamage_twoslope(t , s2 , m1_st , loga1_st , m2_st , loga2_st , Nlim ,
      th_st , tref=25E-3 , k=0.25)
2541 print('Partial fatigue damage for shaft_s2 calculated by script: %.5e (
      includes residual cycles)' %damage)
2542 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
      damage*60*60*24*365*DFE)))
2543
2544 figure()
2545 plot(t , s2/1.e6 , 'k-' , label='')
2546 xlabel('time [s]')
2547 ylabel('shaft_s2 [MPa] ')
2548 title('Axial Stress')
2549 #axis([-16. , 90. , -1.5 , 1.7])

```

```

2550 #legend(loc='best')
2551 savefig('shaft_s2',dpi=300)
2552
2553 damage = fatiguedamage_twslope(t,s3,m1_st,loga1_st,m2_st,loga2_st,Nlim,
      th_st,tref=25E-3,k=0.25)
2554 print('Partial fatigue damage for shaft_s3 calculated by script: %.5e (
      includes residual cycles)' %damage)
2555 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFF))
2556
2557 figure()
2558 plot(t,s3/1.e6,'k-',label='')
2559 xlabel('time [s]')
2560 ylabel('shaft_s3 [MPa]')
2561 title('Axial Stress')
2562 #axis([-16.,90.,-1.5,1.7])
2563 #legend(loc='best')
2564 savefig('shaft_s3',dpi=300)
2565
2566 damage = fatiguedamage_twslope(t,s4,m1_st,loga1_st,m2_st,loga2_st,Nlim,
      th_st,tref=25E-3,k=0.25)
2567 print('Partial fatigue damage for shaft_s4 calculated by script: %.5e (
      includes residual cycles)' %damage)
2568 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFF))
2569
2570 figure()
2571 plot(t,s4/1.e6,'k-',label='')
2572 xlabel('time [s]')
2573 ylabel('shaft_s4 [MPa]')
2574 title('Axial Stress')
2575 #axis([-16.,90.,-1.5,1.7])
2576 #legend(loc='best')
2577 savefig('shaft_s4',dpi=300)
2578
2579 damage = fatiguedamage_twslope(t,s5,m1_st,loga1_st,m2_st,loga2_st,Nlim,
      th_st,tref=25E-3,k=0.25)
2580 print('Partial fatigue damage for shaft_s5 calculated by script: %.5e (
      includes residual cycles)' %damage)
2581 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
      damage*60*60*24*365*DFF))
2582
2583 figure()
2584 plot(t,s5/1.e6,'k-',label='')
2585 xlabel('time [s]')
2586 ylabel('shaft_s5 [MPa]')
2587 title('Axial Stress')
2588 #axis([-16.,90.,-1.5,1.7])
2589 #legend(loc='best')

```

```
2590 savefig('shaft_s5',dpi=300)
2591
2592 damage = fatiguedamage_twslope(t,s6,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
2593 print('Partial fatigue damage for shaft_s6 calculated by script: %.5e (
    includes residual cycles)' %damage)
2594 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFF))
2595
2596 figure()
2597 plot(t,s6/1.e6,'k-',label='')
2598 xlabel('time [s]')
2599 ylabel('shaft_s6 [MPa] ')
2600 title('Axial Stress')
2601 #axis([-16.,90.,-1.5,1.7])
2602 #legend(loc='best')
2603 savefig('shaft_s6',dpi=300)
2604
2605 damage = fatiguedamage_twslope(t,s7,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
2606 print('Partial fatigue damage for shaft_s7 calculated by script: %.5e (
    includes residual cycles)' %damage)
2607 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFF))
2608
2609 figure()
2610 plot(t,s7/1.e6,'k-',label='')
2611 xlabel('time [s]')
2612 ylabel('shaft_s7 [MPa] ')
2613 title('Axial Stress')
2614 #axis([-16.,90.,-1.5,1.7])
2615 #legend(loc='best')
2616 savefig('shaft_s7',dpi=300)
2617
2618 damage = fatiguedamage_twslope(t,s8,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
2619 print('Partial fatigue damage for shaft_s8 calculated by script: %.5e (
    includes residual cycles)' %damage)
2620 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFF))
2621
2622 figure()
2623 plot(t,s8/1.e6,'k-',label='')
2624 xlabel('time [s]')
2625 ylabel('shaft_s8 [MPa] ')
2626 title('Axial Stress')
2627 #axis([-16.,90.,-1.5,1.7])
2628 #legend(loc='best')
2629 savefig('shaft_s8',dpi=300)
```



```

2630
2631
2632
2633 #

```

```

2634 #blade roots
2635 # only one blade analyzed due to equilibrium for 4 blades
2636
2637
2638 fx_bl_1_root = a[:,16]*1.e3;
2639 fy_bl_1_root = a[:,17]*1.e3;
2640 fz_bl_1_root = a[:,18]*1.e3;
2641 mx_bl_1_root = a[:,19]*1.e3;
2642 my_bl_1_root = a[:,20]*1.e3;
2643 mz_bl_1_root = a[:,21]*1.e3;
2644
2645 [sax ,sbendy ,sbendz ,s1 ,s2 ,s3 ,s4 ,s5 ,s6 ,s7 ,s8 ,ssh1 ,ssh2 ,ssh3 ,ssh4 ,ssh5 ,ssh6 ,
    ssh7 ,ssh8 ,svm1 ,svm2 ,svm3 ,svm4 ,svm5 ,svm6 ,svm7 ,svm8 ,sx ,sy ,sz ] = \
2646     cyl_beam_stresses(t ,fx_bl_1_root ,fy_bl_1_root ,fz_bl_1_root ,
        mx_bl_1_root ,my_bl_1_root ,mz_bl_1_root ,D,TWALL,1 ,1 ,1 ,1)
2647
2648 damage = fatiguedamage_twoslope(t ,s1 ,m1_st ,loga1_st ,m2_st ,loga2_st ,Nlim ,
    th_st ,tref=25E-3,k=0.25)
2649 print('Partial fatigue damage for blade root_s1 calculated by script: %.5e
    (includes residual cycles)' %damage)
2650 print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFE)))
2651 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFE)))
2652
2653 figure()
2654 plot(t ,s1/1.e6 ,'k-' ,label='')
2655 xlabel('time [s]')
2656 ylabel('blade root_s1 [MPa]')
2657 title('Axial Stress')
2658 #axis([-16. ,90. , -1.5 ,1.7])
2659 #legend(loc='best')
2660 savefig('blade root_s1' ,dpi=300)
2661
2662 damage = fatiguedamage_twoslope(t ,s2 ,m1_st ,loga1_st ,m2_st ,loga2_st ,Nlim ,
    th_st ,tref=25E-3,k=0.25)
2663 print('Partial fatigue damage for blade root_s2 calculated by script: %.5e
    (includes residual cycles)' %damage)
2664 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFE)))
2665
2666 figure()
2667 plot(t ,s2/1.e6 ,'k-' ,label='')

```

```

2668 xlabel('time [s]')
2669 ylabel('blade root_s2 [MPa] ')
2670 title('Axial Stress')
2671 #axis([-16.,90.,-1.5,1.7])
2672 #legend(loc='best')
2673 savefig('blade root_s2',dpi=300)
2674
2675 damage = fatiguedamage_twslope(t,s3,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
2676 print('Partial fatigue damage for blade root_s3 calculated by script: %.5e
    (includes residual cycles)' %damage)
2677 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
2678
2679 figure()
2680 plot(t,s3/1.e6,'k-',label='')
2681 xlabel('time [s]')
2682 ylabel('blade root_s3 [MPa] ')
2683 title('Axial Stress')
2684 #axis([-16.,90.,-1.5,1.7])
2685 #legend(loc='best')
2686 savefig('blade root_s3',dpi=300)
2687
2688 damage = fatiguedamage_twslope(t,s4,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
2689 print('Partial fatigue damage for blade root_s4 calculated by script: %.5e
    (includes residual cycles)' %damage)
2690 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
2691
2692 figure()
2693 plot(t,s4/1.e6,'k-',label='')
2694 xlabel('time [s]')
2695 ylabel('blade root_s4 [MPa] ')
2696 title('Axial Stress')
2697 #axis([-16.,90.,-1.5,1.7])
2698 #legend(loc='best')
2699 savefig('blade root_s4',dpi=300)
2700
2701 damage = fatiguedamage_twslope(t,s5,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
2702 print('Partial fatigue damage for blade root_s5 calculated by script: %.5e
    (includes residual cycles)' %damage)
2703 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
2704
2705 figure()
2706 plot(t,s5/1.e6,'k-',label='')
2707 xlabel('time [s]')

```

```
2708 ylabel('blade root_s5 [MPa] ')
2709 title('Axial Stress')
2710 #axis([-16.,90.,-1.5,1.7])
2711 #legend(loc='best')
2712 savefig('blade root_s5',dpi=300)
2713
2714 damage = fatiguedamage_twoslope(t,s6,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
2715 print('Partial fatigue damage for blade root_s6 calculated by script: %.5e
    (includes residual cycles)' %damage)
2716 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
2717
2718 figure()
2719 plot(t,s6/1.e6,'k-',label='')
2720 xlabel('time [s]')
2721 ylabel('blade root_s6 [MPa] ')
2722 title('Axial Stress')
2723 #axis([-16.,90.,-1.5,1.7])
2724 #legend(loc='best')
2725 savefig('blade root_s6',dpi=300)
2726
2727 damage = fatiguedamage_twoslope(t,s7,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
2728 print('Partial fatigue damage for blade root_s7 calculated by script: %.5e
    (includes residual cycles)' %damage)
2729 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
2730
2731 figure()
2732 plot(t,s7/1.e6,'k-',label='')
2733 xlabel('time [s]')
2734 ylabel('blade root_s7 [MPa] ')
2735 title('Axial Stress')
2736 #axis([-16.,90.,-1.5,1.7])
2737 #legend(loc='best')
2738 savefig('blade root_s7',dpi=300)
2739
2740 damage = fatiguedamage_twoslope(t,s8,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
2741 print('Partial fatigue damage for blade root_s8 calculated by script: %.5e
    (includes residual cycles)' %damage)
2742 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
2743
2744 figure()
2745 plot(t,s8/1.e6,'k-',label='')
2746 xlabel('time [s]')
2747 ylabel('blade root_s8 [MPa] ')
```

```

2748 title('Axial Stress')
2749 #axis([-16.,90.,-1.5,1.7])
2750 #legend(loc='best')
2751 savefig('blade_root_s8',dpi=300)
2752
2753 #

```

```

2754
2755
2756 #For two blades uncomment this
2757 fx_bl_2_root = a[:,166]*1.e3;
2758 fy_bl_2_root = a[:,167]*1.e3;
2759 fz_bl_2_root = a[:,168]*1.e3;
2760 mx_bl_2_root = a[:,168]*1.e3;
2761 my_bl_2_root = a[:,170]*1.e3;
2762 mz_bl_2_root = a[:,171]*1.e3;
2763
2764 [sax , sbendy , sbendz , s1 , s2 , s3 , s4 , s5 , s6 , s7 , s8 , ssh1 , ssh2 , ssh3 , ssh4 , ssh5 , ssh6 ,
    ssh7 , ssh8 , svm1 , svm2 , svm3 , svm4 , svm5 , svm6 , svm7 , svm8 , sx , sy , sz ] = \
2765     cyl_beam_stresses(t , fx_bl_2_root , fy_bl_2_root , fz_bl_2_root ,
    mx_bl_2_root , my_bl_2_root , mz_bl_2_root , D , TWALL , 1 , 1 , 1 , 1)
2766
2767 damage = fatiguedamage_twoslope(t , s1 , m1_st , loga1_st , m2_st , loga2_st , Nlim , th
    , tref=25E-3 , k=0.25)
2768 print('Partial fatigue damage for blade root_2_s1 calculated by script :
    %.5e (includes residual cycles)' %damage)
2769 print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFE)))
2770 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFE)))
2771
2772 figure()
2773 plot(t , s1 / 1.e6 , 'k-' , label='')
2774 xlabel('time [s]')
2775 ylabel('blade root_2_s1 [MPa] ')
2776 title('Axial Stress')
2777 #axis([-16.,90.,-1.5,1.7])
2778 #legend(loc='best')
2779 savefig('blade_root_2_s1',dpi=300)
2780
2781 damage = fatiguedamage_twoslope(t , s2 , m1_st , loga1_st , m2_st , loga2_st , Nlim , th
    , tref=25E-3 , k=0.25)
2782 print('Partial fatigue damage for blade root_2_s2 calculated by script :
    %.5e (includes residual cycles)' %damage)
2783 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFE)))
2784
2785 figure()

```

```

2786 plot(t,s2/1.e6,'k-',label='')
2787 xlabel('time [s]')
2788 ylabel('blade root_2_s2 [MPa] ')
2789 title('Axial Stress')
2790 #axis([-16.,90.,-1.5,1.7])
2791 #legend(loc='best')
2792 savefig('blade root_2_s2',dpi=300)
2793
2794 damage = fatiguedamage_twoslope(t,s3,m1_st,loga1_st,m2_st,loga2_st,Nlim,th
    ,tref=25E-3,k=0.25)
2795 print('Partial fatigue damage for blade root_2_s3 calculated by script:
    %.5e (includes residual cycles)' %damage)
2796 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFE)))
2797
2798 figure()
2799 plot(t,s3/1.e6,'k-',label='')
2800 xlabel('time [s]')
2801 ylabel('blade root_2_s3 [MPa] ')
2802 title('Axial Stress')
2803 #axis([-16.,90.,-1.5,1.7])
2804 #legend(loc='best')
2805 savefig('blade root_2_s3',dpi=300)
2806
2807 damage = fatiguedamage_twoslope(t,s4,m1_st,loga1_st,m2_st,loga2_st,Nlim,th
    ,tref=25E-3,k=0.25)
2808 print('Partial fatigue damage for blade root_2_s4 calculated by script:
    %.5e (includes residual cycles)' %damage)
2809 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFE)))
2810
2811 figure()
2812 plot(t,s4/1.e6,'k-',label='')
2813 xlabel('time [s]')
2814 ylabel('blade root_2_s4 [MPa] ')
2815 title('Axial Stress')
2816 #axis([-16.,90.,-1.5,1.7])
2817 #legend(loc='best')
2818 savefig('blade root_2_s4',dpi=300)
2819
2820 damage = fatiguedamage_twoslope(t,s5,m1_st,loga1_st,m2_st,loga2_st,Nlim,th
    ,tref=25E-3,k=0.25)
2821 print('Partial fatigue damage for blade root_2_s5 calculated by script:
    %.5e (includes residual cycles)' %damage)
2822 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFE)))
2823
2824 figure()
2825 plot(t,s5/1.e6,'k-',label='')

```

```

2826 xlabel('time [s]')
2827 ylabel('blade root_2_s5 [MPa] ')
2828 title('Axial Stress')
2829 #axis([-16.,90.,-1.5,1.7])
2830 #legend(loc='best')
2831 savefig('blade root_2_s5',dpi=300)
2832
2833 damage = fatiguedamage_twslope(t,s6,m1_st,loga1_st,m2_st,loga2_st,Nlim,th
    ,tref=25E-3,k=0.25)
2834 print('Partial fatigue damage for blade root_2_s6 calculated by script:
    %.5e (includes residual cycles)' %damage)
2835 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
2836
2837 figure()
2838 plot(t,s6/1.e6,'k-',label='')
2839 xlabel('time [s]')
2840 ylabel('blade root_2_s6 [MPa] ')
2841 title('Axial Stress')
2842 #axis([-16.,90.,-1.5,1.7])
2843 #legend(loc='best')
2844 savefig('blade root_2_s6',dpi=300)
2845
2846 damage = fatiguedamage_twslope(t,s7,m1_st,loga1_st,m2_st,loga2_st,Nlim,th
    ,tref=25E-3,k=0.25)
2847 print('Partial fatigue damage for blade root_2_s7 calculated by script:
    %.5e (includes residual cycles)' %damage)
2848 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
2849
2850 figure()
2851 plot(t,s7/1.e6,'k-',label='')
2852 xlabel('time [s]')
2853 ylabel('blade root_2_s7 [MPa] ')
2854 title('Axial Stress')
2855 #axis([-16.,90.,-1.5,1.7])
2856 #legend(loc='best')
2857 savefig('blade root_2_s7',dpi=300)
2858
2859 damage = fatiguedamage_twslope(t,s8,m1_st,loga1_st,m2_st,loga2_st,Nlim,th
    ,tref=25E-3,k=0.25)
2860 print('Partial fatigue damage for blade root_2_s8 calculated by script:
    %.5e (includes residual cycles)' %damage)
2861 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
2862
2863 figure()
2864 plot(t,s8/1.e6,'k-',label='')
2865 xlabel('time [s]')

```

```

2866 ylabel('blade root_2_s8 [MPa] ')
2867 title('Axial Stress')
2868 #axis([-16.,90.,-1.5,1.7])
2869 #legend(loc='best')
2870 savefig('blade root_2_s8',dpi=300)
2871
2872 #-----#
2873
2874 #Rotor
2875 rotor_fx = a[:,172]*1.e3;
2876 rotor_fy = a[:,173]*1.e3;
2877 rotor_fz = a[:,174]*1.e3;
2878 rotor_mx = a[:,175]*1.e3;
2879 rotor_my = a[:,176]*1.e3;
2880 rotor_mz = a[:,177]*1.e3;
2881
2882 [sax , sbendy , sbendz , s1 , s2 , s3 , s4 , s5 , s6 , s7 , s8 , ssh1 , ssh2 , ssh3 , ssh4 , ssh5 , ssh6 ,
      ssh7 , ssh8 , svm1 , svm2 , svm3 , svm4 , svm5 , svm6 , svm7 , svm8 , sx , sy , sz ] = \
2883     cyl_beam_stresses(t , rotor_fx , rotor_fy , rotor_fz , rotor_mx , rotor_my , rotor_mz ,
      rotor_fx , D_Rotor , Twall_Rotor , 1 , 1 , 1 , 1)
2884
2885 damage = fatiguedamage_twslope(t , s1 , m1_st , loga1_st , m2_st , loga2_st , Nlim ,
      th_st , tref=25E-3 , k=0.25)
2886 print('Partial fatigue damage for rotor_s1 calculated by script: %.5e (
      includes residual cycles)' % damage)
2887 print('which corresponds to a fatigue life of %.0f days' % ((t[-1]-t[1]) / (
      damage*60*60*24*DFE)))
2888 print('which corresponds to a fatigue life of %.0f years' % ((t[-1]-t[1]) / (
      damage*60*60*24*365*DFE)))
2889
2890 figure()
2891 plot(t , s1 / 1.e6 , 'k-' , label='')
2892 xlabel('time [s]')
2893 ylabel('rotor_s1 [MPa] ')
2894 title('Axial Stress')
2895 #axis([-16.,90.,-1.5,1.7])
2896 #legend(loc='best')
2897 savefig('rotor_s1' , dpi=300)
2898
2899 damage = fatiguedamage_twslope(t , s2 , m1_st , loga1_st , m2_st , loga2_st , Nlim ,
      th_st , tref=25E-3 , k=0.25)
2900 print('Partial fatigue damage for rotor_s2 calculated by script: %.5e (
      includes residual cycles)' % damage)
2901 print('which corresponds to a fatigue life of %.0f years' % ((t[-1]-t[1]) / (
      damage*60*60*24*365*DFE)))
2902
2903 figure()
2904 plot(t , s2 / 1.e6 , 'k-' , label='')
2905 xlabel('time [s]')

```

```

2906 ylabel('rotor_s2 [MPa] ')
2907 title('Axial Stress')
2908 #axis([-16.,90.,-1.5,1.7])
2909 #legend(loc='best')
2910 savefig('rotor_s2',dpi=300)
2911
2912 damage = fatiguedamage_twoslope(t,s3,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
2913 print('Partial fatigue damage for rotor_s3 calculated by script: %.5e (
    includes residual cycles)' %damage)
2914 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
2915
2916 figure()
2917 plot(t,s3/1.e6,'k-',label='')
2918 xlabel('time [s]')
2919 ylabel('rotor_s3 [MPa] ')
2920 title('Axial Stress')
2921 #axis([-16.,90.,-1.5,1.7])
2922 #legend(loc='best')
2923 savefig('rotor_s3',dpi=300)
2924
2925 damage = fatiguedamage_twoslope(t,s4,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
2926 print('Partial fatigue damage for rotor_s4 calculated by script: %.5e (
    includes residual cycles)' %damage)
2927 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
2928
2929 figure()
2930 plot(t,s4/1.e6,'k-',label='')
2931 xlabel('time [s]')
2932 ylabel('rotor_s4 [MPa] ')
2933 title('Axial Stress')
2934 #axis([-16.,90.,-1.5,1.7])
2935 #legend(loc='best')
2936 savefig('rotor_s4',dpi=300)
2937
2938 damage = fatiguedamage_twoslope(t,s5,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
2939 print('Partial fatigue damage for rotor_s5 calculated by script: %.5e (
    includes residual cycles)' %damage)
2940 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    damage*60*60*24*365*DFE))
2941
2942 figure()
2943 plot(t,s5/1.e6,'k-',label='')
2944 xlabel('time [s]')
2945 ylabel('rotor_s5 [MPa] ')

```



```
2946 title('Axial Stress')
2947 #axis([-16.,90.,-1.5,1.7])
2948 #legend(loc='best')
2949 savefig('rotor_s5',dpi=300)
2950
2951 damage = fatiguedamage_twoslope(t,s6,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
2952 print('Partial fatigue damage for rotor_s6 calculated by script: %.5e (
    includes residual cycles)' %damage)
2953 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFE)))
2954
2955 figure()
2956 plot(t,s6/1.e6,'k-',label='')
2957 xlabel('time [s]')
2958 ylabel('rotor_s6 [MPa]')
2959 title('Axial Stress')
2960 #axis([-16.,90.,-1.5,1.7])
2961 #legend(loc='best')
2962 savefig('rotor_s6',dpi=300)
2963
2964 damage = fatiguedamage_twoslope(t,s7,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
2965 print('Partial fatigue damage for rotor_s7 calculated by script: %.5e (
    includes residual cycles)' %damage)
2966 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFE)))
2967
2968 figure()
2969 plot(t,s7/1.e6,'k-',label='')
2970 xlabel('time [s]')
2971 ylabel('rotor_s7 [MPa]')
2972 title('Axial Stress')
2973 #axis([-16.,90.,-1.5,1.7])
2974 #legend(loc='best')
2975 savefig('rotor_s7',dpi=300)
2976
2977 damage = fatiguedamage_twoslope(t,s8,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
2978 print('Partial fatigue damage for rotor_s8 calculated by script: %.5e (
    includes residual cycles)' %damage)
2979 print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/
    (damage*60*60*24*365*DFE)))
2980
2981 figure()
2982 plot(t,s8/1.e6,'k-',label='')
2983 xlabel('time [s]')
2984 ylabel('rotor_s8 [MPa]')
2985 title('Axial Stress')
```

```
2986 #axis([-16.,90.,-1.5,1.7])  
2987 #legend(loc='best')  
2988 savefig('rotor_s8',dpi=300)
```

H. Python Routine for Assessing Von Mises Stresses at Extreme Wind Conditions

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Thu Apr 29 13:19:16 2021
4
5 @author: Vetle Birkeland Aass
6 """
7
8 import numpy as np
9 from atte_punkter import cyl_beam_stresses
10
11
12 from pylab import *
13 from numpy import loadtxt
14 from scipy import interpolate
15
16
17 SCF_AS_1 = 1
18 SCF_AC_1 = 1
19 SCF_OPB_1 = 1
20 SCF_IPB_1 = 1
21
22 SCF_AS_2 = 1
23 SCF_AC_2 = 1
24 SCF_OPB_2 = 1
25 SCF_IPB_2 = 1
26
27 SCF_AS_3 = 1
28 SCF_AC_3 = 1
29 SCF_OPB_3 = 1
30 SCF_IPB_3 = 1
31
32 SCF_AS_4 = 1
33 SCF_AC_4 = 1
34 SCF_OPB_4 = 1
35 SCF_IPB_4 = 1
```

```
36
37
38 #Yield strength
39
40 gamma = 1.05
41 f_y = 355/gamma
42
43
44 #importing data
45 a = loadtxt('sensors.txt',skiprows=500,dtype='float',delimiter=';') #
    skip 500 s
46 t = a[:,0];
47
48
49
50
51 #

```

```
52 d = 0.3
53 twall = 0.024
54
55 #front_hor_left_strut_x1
56 #Joint type 1
57 #saddle in z-dir
58 front_hor_left_strut_fx_1 = a[:,22]*1.e3;
59 front_hor_left_strut_fy_1 = a[:,102]*1.e3;
60 front_hor_left_strut_fz_1 = a[:,103]*1.e3;
61 mx_fhls_1 = a[:,23]*1.e3;
62 my_fhls_1 = a[:,24]*1.e3;
63 mz_fhls_1 = a[:,25]*1.e3;
64
65 [sax ,sbendy ,sbendz ,s1 ,s2 ,s3 ,s4 ,s5 ,s6 ,s7 ,s8 ,ssh1 ,ssh2 ,ssh3 ,ssh4 ,ssh5 ,ssh6 ,
    ssh7 ,ssh8 ,svm1 ,svm2 ,svm3 ,svm4 ,svm5 ,svm6 ,svm7 ,svm8 ,sx ,sy ,sz ] = \
66     cyl_beam_stresses(t ,front_hor_left_strut_fx_1 ,
        front_hor_left_strut_fy_1 ,front_hor_left_strut_fz_1 ,mx_fhls_1 ,
        my_fhls_1 ,mz_fhls_1 ,d ,twall ,SCF_AS_1,SCF_AC_1,SCF_OPB_1,SCF_IPB_1)
67
68
69 b = (max(svm1))*10e-6
70 print('front_hor_left_x1_svm1 = %.0f' %b)
71 if b > f_y:
72     print('This element is under too much stress')
73
74 b = (max(svm2))*10e-6
75 print('front_hor_left_x1_svm2 = %.0f' %b)
76 if b > f_y:
77     print('This element is under too much stress')
78
```

```
79 b = (max(svm3))*10e-6
80 print('front_hor_left_x1_svm3 = %.0f' %b)
81 if b > f_y:
82     print('This element is under too much stress')
83
84 b = (max(svm4))*10e-6
85 print('front_hor_left_x1_svm4 = %.0f' %b)
86 if b > f_y:
87     print('This element is under too much stress')
88
89 b = (max(svm5))*10e-6
90 print('front_hor_left_x1_svm5 = %.0f' %b)
91 if b > f_y:
92     print('This element is under too much stress')
93
94 b = (max(svm6))*10e-6
95 print('front_hor_left_x1_svm6 = %.0f' %b)
96 if b > f_y:
97     print('This element is under too much stress')
98
99 b = (max(svm7))*10e-6
100 print('front_hor_left_x1_svm7 = %.0f' %b)
101 if b > f_y:
102     print('This element is under too much stress')
103
104 b = (max(svm8))*10e-6
105 print('front_hor_left_x1_svm8 = %.0f' %b)
106 if b > f_y:
107     print('This element is under too much stress')
108
109
110 figure()
111 plot(t,svml/1.e6,'k-',label='')
112 xlabel('time [s]')
113 ylabel('front_hor_left_strut_fx_1_s1 [MPa] ')
114 title('Axial Stress')
115 #axis([-16.,90.,-1.5,1.7])
116 #legend(loc='best')
117 savefig('front_hor_left_strut_fx_1_s1',dpi=300)
118
119
120
121 figure()
122 plot(t,svm2/1.e6,'k-',label='')
123 xlabel('time [s]')
124 ylabel('front_hor_left_strut_fx_1_s2 [MPa] ')
125 title('Axial Stress')
126 #axis([-16.,90.,-1.5,1.7])
127 #legend(loc='best')
```

```
128 savefig('front_hor_left_strut_fx_1_s2',dpi=300)
129
130
131
132 figure()
133 plot(t,svm3/1.e6,'k-',label='')
134 xlabel('time [s]')
135 ylabel('front_hor_left_strut_fx_1_s3 [MPa]')
136 title('Axial Stress')
137 #axis([-16.,90.,-1.5,1.7])
138 #legend(loc='best')
139 savefig('front_hor_left_strut_fx_1_s3',dpi=300)
140
141
142 figure()
143 plot(t,svm4/1.e6,'k-',label='')
144 xlabel('time [s]')
145 ylabel('front_hor_left_strut_fx_1_s4 [MPa]')
146 title('Axial Stress')
147 #axis([-16.,90.,-1.5,1.7])
148 #legend(loc='best')
149 savefig('front_hor_left_strut_fx_1_s4',dpi=300)
150
151
152
153 figure()
154 plot(t,svm5/1.e6,'k-',label='')
155 xlabel('time [s]')
156 ylabel('front_hor_left_strut_fx_1_s5 [MPa]')
157 title('Axial Stress')
158 #axis([-16.,90.,-1.5,1.7])
159 #legend(loc='best')
160 savefig('front_hor_left_strut_fx_1_s5',dpi=300)
161
162
163 figure()
164 plot(t,svm6/1.e6,'k-',label='')
165 xlabel('time [s]')
166 ylabel('front_hor_left_strut_fx_1_s6 [MPa]')
167 title('Axial Stress')
168 #axis([-16.,90.,-1.5,1.7])
169 #legend(loc='best')
170 savefig('front_hor_left_strut_fx_1_s6',dpi=300)
171
172
173
174 figure()
175 plot(t,svm7/1.e6,'k-',label='')
176 xlabel('time [s]')
```

```

177 ylabel('front_hor_left_strut_fx_1_s7 [MPa] ')
178 title('Axial Stress')
179 #axis([-16., 90., -1.5, 1.7])
180 #legend(loc='best')
181 savefig('front_hor_left_strut_fx_1_s7', dpi=300)
182
183
184
185 figure()
186 plot(t, svm8/1.e6, 'k-', label='')
187 xlabel('time [s]')
188 ylabel('front_hor_left_strut_fx_1_s8 [MPa] ')
189 title('Axial Stress')
190 #axis([-16., 90., -1.5, 1.7])
191 #legend(loc='best')
192 savefig('front_hor_left_strut_fx_1_s8', dpi=300)
193
194 #

```

```

195
196 #front_hor_left_strut_x2
197 #Joint type 2
198 #crown in z-dir
199 front_hor_left_strut_fx_2 = a[:, 26]*1.e3;
200 front_hor_left_strut_fy_2 = a[:, 104]*1.e3;
201 front_hor_left_strut_fz_2 = a[:, 105]*1.e3;
202 mx_fhls_2 = a[:, 27]*1.e3;
203 my_fhls_2 = a[:, 28]*1.e3;
204 mz_fhls_2 = a[:, 29]*1.e3;
205
206 [sax, sbendy, sbendz, s1, s2, s3, s4, s5, s6, s7, s8, ssh1, ssh2, ssh3, ssh4, ssh5, ssh6,
    ssh7, ssh8, svm1, svm2, svm3, svm4, svm5, svm6, svm7, svm8, sx, sy, sz] = \
207     cyl_beam_stresses(t, front_hor_left_strut_fx_2,
        front_hor_left_strut_fy_2, front_hor_left_strut_fz_2, mx_fhls_2,
        my_fhls_2, mz_fhls_2, d, twall, SCF_AC_2, SCF_AS_2, SCF_IPB_2, SCF_OPB_2)
208
209
210 b = (max(svm1))*10e-6
211 print('front_hor_left_x2_svm1 = %.0f' %b)
212 if b > f_y:
213     print('This element is under too much stress')
214
215 b = (max(svm2))*10e-6
216 print('front_hor_left_x2_svm2 = %.0f' %b)
217 if b > f_y:
218     print('This element is under too much stress')
219
220 b = (max(svm3))*10e-6

```

```
221 print( 'front_hor_left_x2_svm3 = %.0f' %b)
222 if b > f_y:
223     print( 'This element is under too much stress' )
224
225 b = (max(svm4))*10e-6
226 print( 'front_hor_left_x2_svm4 = %.0f' %b)
227 if b > f_y:
228     print( 'This element is under too much stress' )
229
230 b = (max(svm5))*10e-6
231 print( 'front_hor_left_x2_svm5 = %.0f' %b)
232 if b > f_y:
233     print( 'This element is under too much stress' )
234
235 b = (max(svm6))*10e-6
236 print( 'front_hor_left_x2_svm6 = %.0f' %b)
237 if b > f_y:
238     print( 'This element is under too much stress' )
239
240 b = (max(svm7))*10e-6
241 print( 'front_hor_left_x2_svm7 = %.0f' %b)
242 if b > f_y:
243     print( 'This element is under too much stress' )
244
245 b = (max(svm8))*10e-6
246 print( 'front_hor_left_x2_svm8 = %.0f' %b)
247 if b > f_y:
248     print( 'This element is under too much stress' )
249
250
251
252 figure()
253 plot(t,svm1/1.e6, 'k-', label='')
254 xlabel( 'time [s]' )
255 ylabel( 'front_hor_left_strut_fx_2_s1 [MPa] ' )
256 title( 'Axial Stress' )
257 #axis( [-16., 90., -1.5, 1.7])
258 #legend( loc='best' )
259 savefig( 'front_hor_left_strut_fx_2_s1', dpi=300)
260
261
262
263 figure()
264 plot(t,svm2/1.e6, 'k-', label='')
265 xlabel( 'time [s]' )
266 ylabel( 'front_hor_left_strut_fx_2_s2 [MPa] ' )
267 title( 'Axial Stress' )
268 #axis( [-16., 90., -1.5, 1.7])
269 #legend( loc='best' )
```



```
270 savefig('front_hor_left_strut_fx_2_s2',dpi=300)
271
272
273
274 figure()
275 plot(t,svm3/1.e6,'k-',label='')
276 xlabel('time [s]')
277 ylabel('front_hor_left_strut_fx_2_s3 [MPa]')
278 title('Axial Stress')
279 #axis([-16.,90.,-1.5,1.7])
280 #legend(loc='best')
281 savefig('front_hor_left_strut_fx_2_s3',dpi=300)
282
283
284 figure()
285 plot(t,svm4/1.e6,'k-',label='')
286 xlabel('time [s]')
287 ylabel('front_hor_left_strut_fx_2_s4 [MPa]')
288 title('Axial Stress')
289 #axis([-16.,90.,-1.5,1.7])
290 #legend(loc='best')
291 savefig('front_hor_left_strut_fx_2_s4',dpi=300)
292
293
294
295 figure()
296 plot(t,svm5/1.e6,'k-',label='')
297 xlabel('time [s]')
298 ylabel('front_hor_left_strut_fx_2_s5 [MPa]')
299 title('Axial Stress')
300 #axis([-16.,90.,-1.5,1.7])
301 #legend(loc='best')
302 savefig('front_hor_left_strut_fx_2_s5',dpi=300)
303
304
305
306 figure()
307 plot(t,svm6/1.e6,'k-',label='')
308 xlabel('time [s]')
309 ylabel('front_hor_left_strut_fx_2_s6 [MPa]')
310 title('Axial Stress')
311 #axis([-16.,90.,-1.5,1.7])
312 #legend(loc='best')
313 savefig('front_hor_left_strut_fx_2_s6',dpi=300)
314
315
316
317 figure()
318 plot(t,svm7/1.e6,'k-',label='')
```

```

319 xlabel('time [s]')
320 ylabel('front_hor_left_strut_fx_2_s7 [MPa] ')
321 title('Axial Stress')
322 #axis([-16.,90.,-1.5,1.7])
323 #legend(loc='best')
324 savefig('front_hor_left_strut_fx_2_s7',dpi=300)
325
326
327
328 figure()
329 plot(t,svm8/1.e6,'k-',label='')
330 xlabel('time [s]')
331 ylabel('front_hor_left_strut_fx_2_s8 [MPa] ')
332 title('Axial Stress')
333 #axis([-16.,90.,-1.5,1.7])
334 #legend(loc='best')
335 savefig('front_hor_left_strut_fx_2_s8',dpi=300)
336
337 #

```

```

338 d = 0.2
339 twall = 0.016
340
341 #front_hor_right_strut_x1
342 #Joint type 1
343 #saddle in z-dir
344 front_hor_right_strut_fx_1 = a[:,30]*1.e3;
345 front_hor_right_strut_fy_1 = a[:,106]*1.e3;
346 front_hor_right_strut_fz_1 = a[:,107]*1.e3;
347 mx_fhrs_1 = a[:,31]*1.e3;
348 my_fhrs_1 = a[:,32]*1.e3;
349 mz_fhrs_1 = a[:,33]*1.e3;
350
351
352 [sax ,sbendy ,sbendz ,s1 ,s2 ,s3 ,s4 ,s5 ,s6 ,s7 ,s8 ,ssh1 ,ssh2 ,ssh3 ,ssh4 ,ssh5 ,ssh6 ,
    ssh7 ,ssh8 ,svm1 ,svm2 ,svm3 ,svm4 ,svm5 ,svm6 ,svm7 ,svm8 ,sx ,sy ,sz ] = \
353     cyl_beam_stresses(t ,front_hor_right_strut_fx_1 ,
        front_hor_right_strut_fy_1 ,front_hor_right_strut_fz_1 ,mx_fhrs_1 ,
        my_fhrs_1 ,mz_fhrs_1 ,d ,twall ,SCF_AS_1,SCF_AC_1,SCF_OPB_1,SCF_IPB_1)
354
355
356 b = (max(svm1))*10e-6
357 print('front_hor_right_x1_svm1 = %.0f' %b)
358 if b > f_y:
359     print('This element is under too much stress')
360
361 b = (max(svm2))*10e-6
362 print('front_hor_right_x1_svm2 = %.0f' %b)

```

```
363 if b > f_y:
364     print('This element is under too much stress')
365
366 b = (max(svm3))*10e-6
367 print('front_hor_right_x1_svm3 = %.0f' %b)
368 if b > f_y:
369     print('This element is under too much stress')
370
371 b = (max(svm4))*10e-6
372 print('front_hor_right_x1_svm4 = %.0f' %b)
373 if b > f_y:
374     print('This element is under too much stress')
375
376 b = (max(svm5))*10e-6
377 print('front_hor_right_x1_svm5 = %.0f' %b)
378 if b > f_y:
379     print('This element is under too much stress')
380
381 b = (max(svm6))*10e-6
382 print('front_hor_right_x1_svm6 = %.0f' %b)
383 if b > f_y:
384     print('This element is under too much stress')
385
386 b = (max(svm7))*10e-6
387 print('front_hor_right_x1_svm7 = %.0f' %b)
388 if b > f_y:
389     print('This element is under too much stress')
390
391 b = (max(svm8))*10e-6
392 print('front_hor_right_x1_svm8 = %.0f' %b)
393 if b > f_y:
394     print('This element is under too much stress')
395
396
397 figure()
398 plot(t,svm1/1.e6,'k-',label='')
399 xlabel('time [s]')
400 ylabel('front_hor_right_strut_fx_1_s1 [MPa] ')
401 title('Axial Stress')
402 #axis([-16.,90.,-1.5,1.7])
403 #legend(loc='best')
404 savefig('front_hor_right_strut_fx_1_s1',dpi=300)
405
406
407
408 figure()
409 plot(t,svm2/1.e6,'k-',label='')
410 xlabel('time [s]')
411 ylabel('front_hor_right_strut_fx_1_s2 [MPa] ')

```

```
412 title('Axial Stress')
413 #axis([-16.,90.,-1.5,1.7])
414 #legend(loc='best')
415 savefig('front_hor_right_strut_fx_1_s2',dpi=300)
416
417
418
419 figure()
420 plot(t,svm3/1.e6,'k-',label='')
421 xlabel('time [s]')
422 ylabel('front_hor_right_strut_fx_1_s3 [MPa]')
423 title('Axial Stress')
424 #axis([-16.,90.,-1.5,1.7])
425 #legend(loc='best')
426 savefig('front_hor_right_strut_fx_1_s3',dpi=300)
427
428
429
430 figure()
431 plot(t,svm4/1.e6,'k-',label='')
432 xlabel('time [s]')
433 ylabel('front_hor_right_strut_fx_1_s4 [MPa]')
434 title('Axial Stress')
435 #axis([-16.,90.,-1.5,1.7])
436 #legend(loc='best')
437 savefig('front_hor_right_strut_fx_1_s4',dpi=300)
438
439
440
441 figure()
442 plot(t,svm5/1.e6,'k-',label='')
443 xlabel('time [s]')
444 ylabel('front_hor_right_strut_fx_1_s5 [MPa]')
445 title('Axial Stress')
446 #axis([-16.,90.,-1.5,1.7])
447 #legend(loc='best')
448 savefig('front_hor_right_strut_fx_1_s5',dpi=300)
449
450
451 figure()
452 plot(t,svm6/1.e6,'k-',label='')
453 xlabel('time [s]')
454 ylabel('front_hor_right_strut_fx_1_s6 [MPa]')
455 title('Axial Stress')
456 #axis([-16.,90.,-1.5,1.7])
457 #legend(loc='best')
458 savefig('front_hor_right_strut_fx_1_s6',dpi=300)
459
460
```

```

461
462 figure()
463 plot(t,svm7/1.e6,'k-',label='')
464 xlabel('time [s]')
465 ylabel('Sfront_hor_right_strut_fx_1_s7 [MPa] ')
466 title('Axial Stress')
467 #axis([-16.,90.,-1.5,1.7])
468 #legend(loc='best')
469 savefig('front_hor_right_strut_fx_1_s7',dpi=300)
470
471
472
473 figure()
474 plot(t,svm8/1.e6,'k-',label='')
475 xlabel('time [s]')
476 ylabel('front_hor_right_strut_fx_1_s8 [MPa] ')
477 title('Axial Stress')
478 #axis([-16.,90.,-1.5,1.7])
479 #legend(loc='best')
480 savefig('front_hor_right_strut_fx_1_s8',dpi=300)
481
482 #

```

```

483
484
485 #front_hor_right_strut_x2
486 #Joint type 2
487 #crown in z-dir
488 front_hor_right_strut_fx_2 = a[:,34]*1.e3;
489 front_hor_right_strut_fy_2 = a[:,108]*1.e3;
490 front_hor_right_strut_fz_2 = a[:,109]*1.e3;
491 mx_fhrs_2 = a[:,35]*1.e3;
492 my_fhrs_2 = a[:,36]*1.e3;
493 mz_fhrs_2 = a[:,37]*1.e3;
494
495 [sax,sbendy,sbendz,s1,s2,s3,s4,s5,s6,s7,s8,ssh1,ssh2,ssh3,ssh4,ssh5,ssh6,
    ssh7,ssh8,svm1,svm2,svm3,svm4,svm5,svm6,svm7,svm8,sx,sy,sz] = \
496     cyl_beam_stresses(t,front_hor_right_strut_fx_1,
    front_hor_right_strut_fy_2,front_hor_right_strut_fz_2,mx_fhrs_2,
    my_fhrs_2,mz_fhrs_2,d,twall,SCF_AC_2,SCF_AS_2,SCF_IPB_2,SCF_OPB_2)
497
498
499 b = (max(svm1))*10e-6
500 print('front_hor_right_x2_svm1 = %.0f' %b)
501 if b > f_y:
502     print('This element is under too much stress')
503
504 b = (max(svm2))*10e-6

```

```
505 print('front_hor_right_x2_svm2 = %.0f' %b)
506 if b > f_y:
507     print('This element is under too much stress')
508
509 b = (max(svm3))*10e-6
510 print('front_hor_right_x2_svm3 = %.0f' %b)
511 if b > f_y:
512     print('This element is under too much stress')
513
514 b = (max(svm4))*10e-6
515 print('front_hor_right_x2_svm4 = %.0f' %b)
516 if b > f_y:
517     print('This element is under too much stress')
518
519 b = (max(svm5))*10e-6
520 print('front_hor_right_x2_svm5 = %.0f' %b)
521 if b > f_y:
522     print('This element is under too much stress')
523
524 b = (max(svm6))*10e-6
525 print('front_hor_right_x2_svm6 = %.0f' %b)
526 if b > f_y:
527     print('This element is under too much stress')
528
529 b = (max(svm7))*10e-6
530 print('front_hor_right_x2_svm7 = %.0f' %b)
531 if b > f_y:
532     print('This element is under too much stress')
533
534 b = (max(svm8))*10e-6
535 print('front_hor_right_x2_svm8 = %.0f' %b)
536 if b > f_y:
537     print('This element is under too much stress')
538
539
540
541 figure()
542 plot(t,svm1/1.e6,'k-',label='')
543 xlabel('time [s]')
544 ylabel('front_hor_right_strut_fx_2_s1 [MPa] ')
545 title('Axial Stress')
546 #axis([-16.,90.,-1.5,1.7])
547 #legend(loc='best')
548 savefig('front_hor_right_strut_fx_2_s1',dpi=300)
549
550
551
552 figure()
553 plot(t,svm2/1.e6,'k-',label='')
```

```
554 xlabel('time [s]')
555 ylabel('front_hor_right_strut_fx_2_s2 [MPa] ')
556 title('Axial Stress')
557 #axis([-16.,90.,-1.5,1.7])
558 #legend(loc='best')
559 savefig('front_hor_right_strut_fx_2_s2',dpi=300)
560
561
562
563 figure()
564 plot(t,svm3/1.e6,'k-',label='')
565 xlabel('time [s]')
566 ylabel('front_hor_right_strut_fx_2_s3 [MPa] ')
567 title('Axial Stress')
568 #axis([-16.,90.,-1.5,1.7])
569 #legend(loc='best')
570 savefig('front_hor_right_strut_fx_2_s3',dpi=300)
571
572
573
574 figure()
575 plot(t,svm4/1.e6,'k-',label='')
576 xlabel('time [s]')
577 ylabel('front_hor_right_strut_fx_2_s4 [MPa] ')
578 title('Axial Stress')
579 #axis([-16.,90.,-1.5,1.7])
580 #legend(loc='best')
581 savefig('front_hor_right_strut_fx_2_s4',dpi=300)
582
583
584
585 figure()
586 plot(t,svm5/1.e6,'k-',label='')
587 xlabel('time [s]')
588 ylabel('front_hor_right_strut_fx_2_s5 [MPa] ')
589 title('Axial Stress')
590 #axis([-16.,90.,-1.5,1.7])
591 #legend(loc='best')
592 savefig('front_hor_right_strut_fx_2_s5',dpi=300)
593
594
595
596 figure()
597 plot(t,svm6/1.e6,'k-',label='')
598 xlabel('time [s]')
599 ylabel('front_hor_right_strut_fx_2_s6 [MPa] ')
600 title('Axial Stress')
601 #axis([-16.,90.,-1.5,1.7])
602 #legend(loc='best')
```

```

603 savefig('front_hor_right_strut_fx_2_s6',dpi=300)
604
605
606
607 figure()
608 plot(t,svm7/1.e6,'k-',label='')
609 xlabel('time [s]')
610 ylabel('front_hor_right_strut_fx_2_s7 [MPa] ')
611 title('Axial Stress')
612 #axis([-16.,90.,-1.5,1.7])
613 #legend(loc='best')
614 savefig('front_hor_right_strut_fx_2_s7',dpi=300)
615
616
617
618 figure()
619 plot(t,svm8/1.e6,'k-',label='')
620 xlabel('time [s]')
621 ylabel('front_hor_right_strut_fx_2_s8 [MPa] ')
622 title('Axial Stress')
623 #axis([-16.,90.,-1.5,1.7])
624 #legend(loc='best')
625 savefig('front_hor_right_strut_fx_2_s8',dpi=300)
626
627 #

```

```

628 d = 0.3
629 twall = 0.012
630
631 #front_vert_down_strut_x1
632 #Joint type 1
633 #crown in z-dir
634 front_vert_down_strut_fx_1 = a[:,38]*1.e3;
635 front_vert_down_strut_fy_1 = a[:,110]*1.e3;
636 front_vert_down_strut_fz_1 = a[:,111]*1.e3;
637 mx_fvds_1 = a[:,39]*1.e3;
638 my_fvds_1 = a[:,40]*1.e3;
639 mz_fvds_1 = a[:,41]*1.e3;
640
641 [sax, sbendy, sbendz, s1, s2, s3, s4, s5, s6, s7, s8, ssh1, ssh2, ssh3, ssh4, ssh5, ssh6,
    ssh7, ssh8, svm1, svm2, svm3, svm4, svm5, svm6, svm7, svm8, sx, sy, sz] = \
642     cyl_beam_stresses(t, front_vert_down_strut_fx_1,
        front_vert_down_strut_fy_1, front_vert_down_strut_fz_1, mx_fvds_1,
        my_fvds_1, mz_fvds_1, d, twall, SCF_AC_1, SCF_AS_1, SCF_IPB_1, SCF_OPB_1)
643
644
645 b = (max(svm1))*10e-6
646 print('front_vert_down_x1_svm1 = %.0f' %b)

```



```
647 if b > f_y:
648     print('This element is under too much stress')
649
650 b = (max(svm2))*10e-6
651 print('front_vert_down_x1_svm2 = %.0f' %b)
652 if b > f_y:
653     print('This element is under too much stress')
654
655 b = (max(svm3))*10e-6
656 print('front_vert_down_x1_svm3 = %.0f' %b)
657 if b > f_y:
658     print('This element is under too much stress')
659
660 b = (max(svm4))*10e-6
661 print('front_vert_down_x1_svm4 = %.0f' %b)
662 if b > f_y:
663     print('This element is under too much stress')
664
665 b = (max(svm5))*10e-6
666 print('front_vert_down_x1_svm5 = %.0f' %b)
667 if b > f_y:
668     print('This element is under too much stress')
669
670 b = (max(svm6))*10e-6
671 print('front_vert_down_x1_svm6 = %.0f' %b)
672 if b > f_y:
673     print('This element is under too much stress')
674
675 b = (max(svm7))*10e-6
676 print('front_vert_down_x1_svm7 = %.0f' %b)
677 if b > f_y:
678     print('This element is under too much stress')
679
680 b = (max(svm8))*10e-6
681 print('front_vert_down_x1_svm8 = %.0f' %b)
682 if b > f_y:
683     print('This element is under too much stress')
684
685
686
687 figure()
688 plot(t,svm1/1.e6,'k-',label='')
689 xlabel('time [s]')
690 ylabel('front_vert_down_strut_fx_1_s1 [MPa] ')
691 title('Axial Stress')
692 #axis([-16.,90.,-1.5,1.7])
693 #legend(loc='best')
694 savefig('front_vert_down_strut_fx_1_s1',dpi=300)
695
```

```
696
697
698 figure()
699 plot(t,svm2/1.e6,'k-',label='')
700 xlabel('time [s]')
701 ylabel('front_vert_down_strut_fx_1_s2 [MPa] ')
702 title('Axial Stress')
703 #axis([-16.,90.,-1.5,1.7])
704 #legend(loc='best')
705 savefig('front_vert_down_strut_fx_1_s2',dpi=300)
706
707
708
709 figure()
710 plot(t,svm3/1.e6,'k-',label='')
711 xlabel('time [s]')
712 ylabel('front_vert_down_strut_fx_1_s3 [MPa] ')
713 title('Axial Stress')
714 #axis([-16.,90.,-1.5,1.7])
715 #legend(loc='best')
716 savefig('front_vert_down_strut_fx_1_s3',dpi=300)
717
718
719
720 figure()
721 plot(t,svm4/1.e6,'k-',label='')
722 xlabel('time [s]')
723 ylabel('front_vert_down_strut_fx_1_s4 [MPa] ')
724 title('Axial Stress')
725 #axis([-16.,90.,-1.5,1.7])
726 #legend(loc='best')
727 savefig('front_vert_down_strut_fx_1_s4',dpi=300)
728
729
730
731 figure()
732 plot(t,svm5/1.e6,'k-',label='')
733 xlabel('time [s]')
734 ylabel('front_vert_down_strut_fx_1_s5 [MPa] ')
735 title('Axial Stress')
736 #axis([-16.,90.,-1.5,1.7])
737 #legend(loc='best')
738 savefig('front_vert_down_strut_fx_1_s5',dpi=300)
739
740
741
742 figure()
743 plot(t,svm6/1.e6,'k-',label='')
744 xlabel('time [s]')
```

```

745 ylabel('front_vert_down_strut_fx_1_s6 [MPa] ')
746 title('Axial Stress')
747 #axis([-16.,90.,-1.5,1.7])
748 #legend(loc='best')
749 savefig('front_vert_down_strut_fx_1_s6',dpi=300)
750
751
752
753 figure()
754 plot(t,svm7/1.e6,'k-',label='')
755 xlabel('time [s]')
756 ylabel('front_vert_down_strut_fx_1_s7 [MPa] ')
757 title('Axial Stress')
758 #axis([-16.,90.,-1.5,1.7])
759 #legend(loc='best')
760 savefig('front_vert_down_strut_fx_1_s7',dpi=300)
761
762
763
764 figure()
765 plot(t,svm8/1.e6,'k-',label='')
766 xlabel('time [s]')
767 ylabel('front_vert_down_strut_fx_1_s8 [MPa] ')
768 title('Axial Stress')
769 #axis([-16.,90.,-1.5,1.7])
770 #legend(loc='best')
771 savefig('front_vert_down_strut_fx_1_s8',dpi=300)
772
773 #

```

```

774
775 #front_vert_down_strut_x2
776 #Joint type 2
777 #saddle in z-dir
778 front_vert_down_strut_fx_2 = a[:,42]*1.e3;
779 front_vert_down_strut_fy_2 = a[:,112]*1.e3;
780 front_vert_down_strut_fz_2 = a[:,113]*1.e3;
781 mx_fvds_2 = a[:,43]*1.e3;
782 my_fvds_2 = a[:,44]*1.e3;
783 mz_fvds_2 = a[:,45]*1.e3;
784
785 [sax ,sbendy ,sbendz ,s1 ,s2 ,s3 ,s4 ,s5 ,s6 ,s7 ,s8 ,ssh1 ,ssh2 ,ssh3 ,ssh4 ,ssh5 ,ssh6 ,
    ssh7 ,ssh8 ,svm1 ,svm2 ,svm3 ,svm4 ,svm5 ,svm6 ,svm7 ,svm8 ,sx ,sy ,sz ] = \
786     cyl_beam_stresses(t,front_vert_down_strut_fx_2 ,
    front_vert_down_strut_fy_2 ,front_vert_down_strut_fy_2 ,mx_fvds_2 ,
    my_fvds_2 ,mz_fvds_2 ,d ,twall ,SCF_AS_2,SCF_AC_2,SCF_OPB_2,SCF_IPB_2)
787
788

```

```
789
790
791 b = (max(svm1))*10e-6
792 print('front_vert_down_x2_svm1 = %.0f' %b)
793 if b > f_y:
794     print('This element is under too much stress')
795
796 b = (max(svm2))*10e-6
797 print('front_vert_down_x2_svm2 = %.0f' %b)
798 if b > f_y:
799     print('This element is under too much stress')
800
801 b = (max(svm3))*10e-6
802 print('front_vert_down_x2_svm3 = %.0f' %b)
803 if b > f_y:
804     print('This element is under too much stress')
805
806 b = (max(svm4))*10e-6
807 print('front_vert_down_x2_svm4 = %.0f' %b)
808 if b > f_y:
809     print('This element is under too much stress')
810
811 b = (max(svm5))*10e-6
812 print('front_vert_down_x2_svm5 = %.0f' %b)
813 if b > f_y:
814     print('This element is under too much stress')
815
816 b = (max(svm6))*10e-6
817 print('front_vert_down_x2_svm6 = %.0f' %b)
818 if b > f_y:
819     print('This element is under too much stress')
820
821 b = (max(svm7))*10e-6
822 print('front_vert_down_x2_svm7 = %.0f' %b)
823 if b > f_y:
824     print('This element is under too much stress')
825
826 b = (max(svm8))*10e-6
827 print('front_vert_down_x2_svm8 = %.0f' %b)
828 if b > f_y:
829     print('This element is under too much stress')
830
831
832
833
834 figure()
835 plot(t, svm1/1.e6, 'k-', label='')
836 xlabel('time [s]')
837 ylabel('front_vert_down_strut_fx_2_s1 [MPa]')
```

```
838 title('Axial Stress')
839 #axis([-16.,90.,-1.5,1.7])
840 #legend(loc='best')
841 savefig('front_vert_down_strut_fx_2_s1',dpi=300)
842
843
844
845 figure()
846 plot(t,svm2/1.e6,'k-',label='')
847 xlabel('time [s]')
848 ylabel('front_vert_down_strut_fx_2_s2 [MPa]')
849 title('Axial Stress')
850 #axis([-16.,90.,-1.5,1.7])
851 #legend(loc='best')
852 savefig('front_vert_down_strut_fx_2_s2',dpi=300)
853
854
855
856 figure()
857 plot(t,svm3/1.e6,'k-',label='')
858 xlabel('time [s]')
859 ylabel('front_vert_down_strut_fx_2_s3 [MPa]')
860 title('Axial Stress')
861 #axis([-16.,90.,-1.5,1.7])
862 #legend(loc='best')
863 savefig('front_vert_down_strut_fx_2_s3',dpi=300)
864
865
866
867 figure()
868 plot(t,svm4/1.e6,'k-',label='')
869 xlabel('time [s]')
870 ylabel('front_vert_down_strut_fx_2_s4 [MPa]')
871 title('Axial Stress')
872 #axis([-16.,90.,-1.5,1.7])
873 #legend(loc='best')
874 savefig('front_vert_down_strut_fx_2_s4',dpi=300)
875
876
877
878 figure()
879 plot(t,svm5/1.e6,'k-',label='')
880 xlabel('time [s]')
881 ylabel('front_vert_down_strut_fx_2_s5 [MPa]')
882 title('Axial Stress')
883 #axis([-16.,90.,-1.5,1.7])
884 #legend(loc='best')
885 savefig('front_vert_down_strut_fx_2_s5',dpi=300)
886
```

```
887
888
889 figure()
890 plot(t,svm6/1.e6,'k-',label='')
891 xlabel('time [s]')
892 ylabel('front_vert_down_strut_fx_2_s6 [MPa]')
893 title('Axial Stress')
894 #axis([-16.,90.,-1.5,1.7])
895 #legend(loc='best')
896 savefig('front_vert_down_strut_fx_2_s6',dpi=300)
897
898
899
900 figure()
901 plot(t,svm7/1.e6,'k-',label='')
902 xlabel('time [s]')
903 ylabel('front_vert_down_strut_fx_2_s7 [MPa]')
904 title('Axial Stress')
905 #axis([-16.,90.,-1.5,1.7])
906 #legend(loc='best')
907 savefig('front_vert_down_strut_fx_2_s7',dpi=300)
908
909
910
911 figure()
912 plot(t,svm8/1.e6,'k-',label='')
913 xlabel('time [s]')
914 ylabel('front_vert_down_strut_fx_2_s8 [MPa]')
915 title('Axial Stress')
916 #axis([-16.,90.,-1.5,1.7])
917 #legend(loc='best')
918 savefig('front_vert_down_strut_fx_2_s8',dpi=300)
919 #
```

```
920
921 d = 0.3
922 twall = 0.016
923
924 #front_vert_up_strut_x1
925 #Joint type 1
926 #crown in z-dir
927 front_vert_up_strut_fx_1 = a[:,46]*1.e3;
928 front_vert_up_strut_fy_1 = a[:,114]*1.e3;
929 front_vert_up_strut_fz_1 = a[:,115]*1.e3;
930 mx_fvus_1 = a[:,47]*1.e3;
931 my_fvus_1 = a[:,48]*1.e3;
932 mz_fvus_1 = a[:,49]*1.e3;
933
```

```
934 [sax , sbendy , sbendz , s1 , s2 , s3 , s4 , s5 , s6 , s7 , s8 , ssh1 , ssh2 , ssh3 , ssh4 , ssh5 , ssh6 ,
    ssh7 , ssh8 , svm1 , svm2 , svm3 , svm4 , svm5 , svm6 , svm7 , svm8 , sx , sy , sz ] = \
935     cyl_beam_stresses(t , front_vert_up_strut_fx_1 , front_vert_up_strut_fy_1 ,
    front_vert_up_strut_fz_1 , mx_fvds_2 , my_fvds_2 , mz_fvds_2 , d , twall ,
    SCF_AC_1 , SCF_AS_1 , SCF_IPB_1 , SCF_OPB_1)
936
937
938 b = (max(svm1))*10e-6
939 print('front_vert_up_x1_svm1 = %.0f' %b)
940 if b > f_y:
941     print('This element is under too much stress')
942
943 b = (max(svm2))*10e-6
944 print('front_vert_up_x1_svm2 = %.0f' %b)
945 if b > f_y:
946     print('This element is under too much stress')
947
948 b = (max(svm3))*10e-6
949 print('front_vert_up_x1_svm3 = %.0f' %b)
950 if b > f_y:
951     print('This element is under too much stress')
952
953 b = (max(svm4))*10e-6
954 print('front_vert_up_x1_svm4 = %.0f' %b)
955 if b > f_y:
956     print('This element is under too much stress')
957
958 b = (max(svm5))*10e-6
959 print('front_vert_up_x1_svm5 = %.0f' %b)
960 if b > f_y:
961     print('This element is under too much stress')
962
963 b = (max(svm6))*10e-6
964 print('front_vert_up_x1_svm6 = %.0f' %b)
965 if b > f_y:
966     print('This element is under too much stress')
967
968 b = (max(svm7))*10e-6
969 print('front_vert_up_x1_svm7 = %.0f' %b)
970 if b > f_y:
971     print('This element is under too much stress')
972
973 b = (max(svm8))*10e-6
974 print('front_vert_up_x1_svm8 = %.0f' %b)
975 if b > f_y:
976     print('This element is under too much stress')
977
978
979
```

```
980 figure()
981 plot(t,svm1/1.e6,'k-',label='')
982 xlabel('time [s]')
983 ylabel('front_vert_up_strut_fx_1_s1 [MPa]')
984 title('Axial Stress')
985 #axis([-16.,90.,-1.5,1.7])
986 #legend(loc='best')
987 savefig('front_vert_up_strut_fx_1_s1',dpi=300)
988
989
990
991 figure()
992 plot(t,svm2/1.e6,'k-',label='')
993 xlabel('time [s]')
994 ylabel('front_vert_up_strut_fx_1_s2 [MPa]')
995 title('Axial Stress')
996 #axis([-16.,90.,-1.5,1.7])
997 #legend(loc='best')
998 savefig('front_vert_up_strut_fx_1_s2',dpi=300)
999
1000
1001
1002 figure()
1003 plot(t,svm3/1.e6,'k-',label='')
1004 xlabel('time [s]')
1005 ylabel('front_vert_up_strut_fx_1_s3 [MPa]')
1006 title('Axial Stress')
1007 #axis([-16.,90.,-1.5,1.7])
1008 #legend(loc='best')
1009 savefig('front_vert_up_strut_fx_1_s3',dpi=300)
1010
1011
1012
1013 figure()
1014 plot(t,svm4/1.e6,'k-',label='')
1015 xlabel('time [s]')
1016 ylabel('front_vert_up_strut_fx_1_s4 [MPa]')
1017 title('Axial Stress')
1018 #axis([-16.,90.,-1.5,1.7])
1019 #legend(loc='best')
1020 savefig('front_vert_up_strut_fx_1_s4',dpi=300)
1021
1022
1023
1024 figure()
1025 plot(t,svm5/1.e6,'k-',label='')
1026 xlabel('time [s]')
1027 ylabel('front_vert_up_strut_fx_1_s5 [MPa]')
1028 title('Axial Stress')
```



```
1029 #axis([-16.,90.,-1.5,1.7])
1030 #legend(loc='best')
1031 savefig('front_vert_up_strut_fx_1_s5',dpi=300)
1032
1033
1034
1035 figure()
1036 plot(t,svm6/1.e6,'k-',label='')
1037 xlabel('time [s]')
1038 ylabel('front_vert_up_strut_fx_1_s6 [MPa]')
1039 title('Axial Stress')
1040 #axis([-16.,90.,-1.5,1.7])
1041 #legend(loc='best')
1042 savefig('front_vert_up_strut_fx_1_s6',dpi=300)
1043
1044
1045
1046 figure()
1047 plot(t,svm7/1.e6,'k-',label='')
1048 xlabel('time [s]')
1049 ylabel('front_vert_up_strut_fx_1_s7 [MPa]')
1050 title('Axial Stress')
1051 #axis([-16.,90.,-1.5,1.7])
1052 #legend(loc='best')
1053 savefig('front_vert_up_strut_fx_1_s7',dpi=300)
1054
1055
1056
1057 figure()
1058 plot(t,svm8/1.e6,'k-',label='')
1059 xlabel('time [s]')
1060 ylabel('front_vert_up_strut_fx_1_s8 [MPa]')
1061 title('Axial Stress')
1062 #axis([-16.,90.,-1.5,1.7])
1063 #legend(loc='best')
1064 savefig('front_vert_up_strut_fx_1_s8',dpi=300)
1065
1066
1067 #
```

```
1068
1069 #front_vert_up_strut_x2
1070 #Joint type 2
1071 #saddle in z-dir
1072 front_vert_up_strut_fx_2 = a[:,50]*1.e3;
1073 front_vert_up_strut_fy_2 = a[:,116]*1.e3;
1074 front_vert_up_strut_fz_2 = a[:,117]*1.e3;
1075 mx_fvus_2 = a[:,51]*1.e3;
```

```
1076 my_fvus_2 = a[:,52]*1.e3;
1077 mz_fvus_2 = a[:,53]*1.e3;
1078
1079 [sax , sbendy , sbendz , s1 , s2 , s3 , s4 , s5 , s6 , s7 , s8 , ssh1 , ssh2 , ssh3 , ssh4 , ssh5 , ssh6 ,
1080     ssh7 , ssh8 , svm1 , svm2 , svm3 , svm4 , svm5 , svm6 , svm7 , svm8 , sx , sy , sz ] = \
1081     cyl_beam_stresses(t , front_vert_up_strut_fx_2 , front_vert_up_strut_fy_2 ,
1082     front_vert_up_strut_fz_2 , mx_fvus_2 , my_fvus_2 , mz_fvus_2 , d , twall ,
1083     SCF_AS_2 , SCF_AC_2 , SCF_OPB_2 , SCF_IPB_2)
1084
1085 b = (max(svm1))*10e-6
1086 print('front_vert_up_x2_svm1 = %.0f' %b)
1087 if b > f_y:
1088     print('This element is under too much stress')
1089
1090 b = (max(svm2))*10e-6
1091 print('front_vert_up_x2_svm2 = %.0f' %b)
1092 if b > f_y:
1093     print('This element is under too much stress')
1094
1095 b = (max(svm3))*10e-6
1096 print('front_vert_up_x2_svm3 = %.0f' %b)
1097 if b > f_y:
1098     print('This element is under too much stress')
1099
1100 b = (max(svm4))*10e-6
1101 print('front_vert_up_x2_svm4 = %.0f' %b)
1102 if b > f_y:
1103     print('This element is under too much stress')
1104
1105 b = (max(svm5))*10e-6
1106 print('front_vert_up_x2_svm5 = %.0f' %b)
1107 if b > f_y:
1108     print('This element is under too much stress')
1109
1110 b = (max(svm6))*10e-6
1111 print('front_vert_up_x2_svm6 = %.0f' %b)
1112 if b > f_y:
1113     print('This element is under too much stress')
1114
1115 b = (max(svm7))*10e-6
1116 print('front_vert_up_x2_svm7 = %.0f' %b)
1117 if b > f_y:
1118     print('This element is under too much stress')
1119
1120 b = (max(svm8))*10e-6
1121 print('front_vert_up_x2_svm8 = %.0f' %b)
1122 if b > f_y:
1123     print('This element is under too much stress')
```

```
1122
1123
1124 figure()
1125 plot(t,svm1/1.e6,'k-',label='')
1126 xlabel('time [s]')
1127 ylabel('front_vert_up_strut_fx_2_s1 [MPa]')
1128 title('Axial Stress')
1129 #axis([-16.,90.,-1.5,1.7])
1130 #legend(loc='best')
1131 savefig('front_vert_up_strut_fx_2_s1',dpi=300)
1132
1133
1134
1135 figure()
1136 plot(t,svm2/1.e6,'k-',label='')
1137 xlabel('time [s]')
1138 ylabel('front_vert_up_strut_fx_2_s2 [MPa]')
1139 title('Axial Stress')
1140 #axis([-16.,90.,-1.5,1.7])
1141 #legend(loc='best')
1142 savefig('front_vert_up_strut_fx_2_s2',dpi=300)
1143
1144
1145
1146 figure()
1147 plot(t,svm3/1.e6,'k-',label='')
1148 xlabel('time [s]')
1149 ylabel('front_vert_up_strut_fx_2_s3 [MPa]')
1150 title('Axial Stress')
1151 #axis([-16.,90.,-1.5,1.7])
1152 #legend(loc='best')
1153 savefig('front_vert_up_strut_fx_2_s3',dpi=300)
1154
1155
1156
1157 figure()
1158 plot(t,svm4/1.e6,'k-',label='')
1159 xlabel('time [s]')
1160 ylabel('front_vert_up_strut_fx_2_s4 [MPa]')
1161 title('Axial Stress')
1162 #axis([-16.,90.,-1.5,1.7])
1163 #legend(loc='best')
1164 savefig('front_vert_up_strut_fx_2_s4',dpi=300)
1165
1166
1167
1168 figure()
1169 plot(t,svm5/1.e6,'k-',label='')
1170 xlabel('time [s]')
```

```
1171 ylabel( 'front_vert_up_strut_fx_2_s5 [MPa] ' )
1172 title( 'Axial Stress ' )
1173 #axis( [-16., 90., -1.5, 1.7])
1174 #legend( loc='best ' )
1175 savefig( 'front_vert_up_strut_fx_2_s5 ', dpi=300)
1176
1177
1178
1179 figure()
1180 plot( t, svm6/1.e6, 'k-', label='' )
1181 xlabel( 'time [s] ' )
1182 ylabel( 'front_vert_up_strut_fx_2_s6 [MPa] ' )
1183 title( 'Axial Stress ' )
1184 #axis( [-16., 90., -1.5, 1.7])
1185 #legend( loc='best ' )
1186 savefig( 'front_vert_up_strut_fx_2_s6 ', dpi=300)
1187
1188
1189
1190 figure()
1191 plot( t, svm7/1.e6, 'k-', label='' )
1192 xlabel( 'time [s] ' )
1193 ylabel( 'front_vert_up_strut_fx_2_s7 [MPa] ' )
1194 title( 'Axial Stress ' )
1195 #axis( [-16., 90., -1.5, 1.7])
1196 #legend( loc='best ' )
1197 savefig( 'front_vert_up_strut_fx_2_s7 ', dpi=300)
1198
1199
1200
1201 figure()
1202 plot( t, svm8/1.e6, 'k-', label='' )
1203 xlabel( 'time [s] ' )
1204 ylabel( 'front_vert_up_strut_fx_2_s8 [MPa] ' )
1205 title( 'Axial Stress ' )
1206 #axis( [-16., 90., -1.5, 1.7])
1207 #legend( loc='best ' )
1208 savefig( 'front_vert_up_strut_fx_2_s8 ', dpi=300)
1209
1210
1211 #

```

```
1212
1213 d = 0.2
1214 twall = 0.008
1215
1216 #rear_left_down_strut_x1
1217 #Joint type 3

```

```

1218 #crown in z-dir
1219 rear_left_down_strut_fx_1 = a[:,54]*1.e3;
1220 rear_left_down_strut_fy_1 = a[:,118]*1.e3;
1221 rear_left_down_strut_fz_1 = a[:,119]*1.e3;
1222 mx_rlds_1 = a[:,55]*1.e3;
1223 my_rlds_1 = a[:,56]*1.e3;
1224 mz_rlds_1 = a[:,57]*1.e3;
1225
1226 [sax , sbendy , sbendz , s1 , s2 , s3 , s4 , s5 , s6 , s7 , s8 , ssh1 , ssh2 , ssh3 , ssh4 , ssh5 , ssh6 ,
      ssh7 , ssh8 , svm1 , svm2 , svm3 , svm4 , svm5 , svm6 , svm7 , svm8 , sx , sy , sz ] = \
1227     cyl_beam_stresses(t , front_vert_up_strut_fx_1 , rear_left_down_strut_fy_1
      , rear_left_down_strut_fy_1 , mx_rlds_1 , my_rlds_1 , mz_rlds_1 , d , twall ,
      SCF_AC_4 , SCF_AS_4 , SCF_IPB_4 , SCF_OPB_4)
1228
1229
1230
1231 b = (max(svm1))*10e-6
1232 print('rear_left_down_x1_svm1 = %.0f' %b)
1233 if b > f_y:
1234     print('This element is under too much stress')
1235
1236 b = (max(svm2))*10e-6
1237 print('rear_left_down_x1_svm2 = %.0f' %b)
1238 if b > f_y:
1239     print('This element is under too much stress')
1240
1241 b = (max(svm3))*10e-6
1242 print('rear_left_down_x1_svm3 = %.0f' %b)
1243 if b > f_y:
1244     print('This element is under too much stress')
1245
1246 b = (max(svm4))*10e-6
1247 print('rear_left_down_x1_svm4 = %.0f' %b)
1248 if b > f_y:
1249     print('This element is under too much stress')
1250
1251 b = (max(svm5))*10e-6
1252 print('rear_left_down_x1_svm5 = %.0f' %b)
1253 if b > f_y:
1254     print('This element is under too much stress')
1255
1256 b = (max(svm6))*10e-6
1257 print('rear_left_down_x1_svm6 = %.0f' %b)
1258 if b > f_y:
1259     print('This element is under too much stress')
1260
1261 b = (max(svm7))*10e-6
1262 print('rear_left_down_x1_svm7 = %.0f' %b)
1263 if b > f_y:

```

```
1264     print('This element is under too much stress')
1265
1266 b = (max(svm8))*10e-6
1267 print('rear_left_down_x1_svm8 = %.0f' %b)
1268 if b > f_y:
1269     print('This element is under too much stress')
1270
1271
1272
1273 figure()
1274 plot(t,svml/1.e6,'k-',label='')
1275 xlabel('time [s]')
1276 ylabel('rear_left_down_strut_fx_1_s1 [MPa] ')
1277 title('Axial Stress')
1278 #axis([-16.,90.,-1.5,1.7])
1279 #legend(loc='best')
1280 savefig('rear_left_down_strut_fx_1_s1',dpi=300)
1281
1282
1283 figure()
1284 plot(t,svm2/1.e6,'k-',label='')
1285 xlabel('time [s]')
1286 ylabel('rear_left_down_strut_fx_1_s2 [MPa] ')
1287 title('Axial Stress')
1288 #axis([-16.,90.,-1.5,1.7])
1289 #legend(loc='best')
1290 savefig('rear_left_down_strut_fx_1_s2',dpi=300)
1291
1292
1293
1294 figure()
1295 plot(t,svm3/1.e6,'k-',label='')
1296 xlabel('time [s]')
1297 ylabel('rear_left_down_strut_fx_1_s3 [MPa] ')
1298 title('Axial Stress')
1299 #axis([-16.,90.,-1.5,1.7])
1300 #legend(loc='best')
1301 savefig('rear_left_down_strut_fx_1_s3',dpi=300)
1302
1303
1304 figure()
1305 plot(t,svm4/1.e6,'k-',label='')
1306 xlabel('time [s]')
1307 ylabel('rear_left_down_strut_fx_1_s4 [MPa] ')
1308 title('Axial Stress')
1309 #axis([-16.,90.,-1.5,1.7])
1310 #legend(loc='best')
1311 savefig('rear_left_down_strut_fx_1_s4',dpi=300)
1312
```

```
1313
1314
1315 figure()
1316 plot(t,svm5/1.e6,'k-',label='')
1317 xlabel('time [s]')
1318 ylabel('rear_left_down_strut_fx_1_s5 [MPa] ')
1319 title('Axial Stress')
1320 #axis([-16.,90.,-1.5,1.7])
1321 #legend(loc='best')
1322 savefig('rear_left_down_strut_fx_1_s5',dpi=300)
1323
1324
1325
1326 figure()
1327 plot(t,svm6/1.e6,'k-',label='')
1328 xlabel('time [s]')
1329 ylabel('rear_left_down_strut_fx_1_s6 [MPa] ')
1330 title('Axial Stress')
1331 #axis([-16.,90.,-1.5,1.7])
1332 #legend(loc='best')
1333 savefig('rear_left_down_strut_fx_1_s6',dpi=300)
1334
1335
1336 figure()
1337 plot(t,svm7/1.e6,'k-',label='')
1338 xlabel('time [s]')
1339 ylabel('rear_left_down_strut_fx_1_s7 [MPa] ')
1340 title('Axial Stress')
1341 #axis([-16.,90.,-1.5,1.7])
1342 #legend(loc='best')
1343 savefig('rear_left_down_strut_fx_1_s7',dpi=300)
1344
1345
1346
1347 figure()
1348 plot(t,svm8/1.e6,'k-',label='')
1349 xlabel('time [s]')
1350 ylabel('rear_left_down_strut_fx_1_s8 [MPa] ')
1351 title('Axial Stress')
1352 #axis([-16.,90.,-1.5,1.7])
1353 #legend(loc='best')
1354 savefig('rear_left_down_strut_fx_1_s8',dpi=300)
1355
1356 #
```

```
1357
1358
1359 #rear_left_down_strut_x2
```

```

1360 #Joint type 4
1361 #crown in z-dir
1362 rear_left_down_strut_fx_2 = a[:,58]*1.e3;
1363 rear_left_down_strut_fy_2 = a[:,120]*1.e3;
1364 rear_left_down_strut_fz_2 = a[:,121]*1.e3;
1365 mx_rlds_2 = a[:,59]*1.e3;
1366 my_rlds_2 = a[:,60]*1.e3;
1367 mz_rlds_2 = a[:,61]*1.e3;
1368
1369 [sax ,sbendy ,sbendz ,s1 ,s2 ,s3 ,s4 ,s5 ,s6 ,s7 ,s8 ,ssh1 ,ssh2 ,ssh3 ,ssh4 ,ssh5 ,ssh6 ,
    ssh7 ,ssh8 ,svm1 ,svm2 ,svm3 ,svm4 ,svm5 ,svm6 ,svm7 ,svm8 ,sx ,sy ,sz ] = \
1370     cyl_beam_stresses(t ,front_vert_up_strut_fx_2 ,rear_left_down_strut_fy_2
    ,rear_left_down_strut_fy_2 ,mx_rlds_2 ,my_rlds_2 ,mz_rlds_2 ,d ,twall ,
    SCF_AC_4,SCF_AS_4,SCF_IPB_4,SCF_OPB_4)
1371
1372
1373 b = (max(svm1))*10e-6
1374 print('rear_left_down_x2_svm1 = %.0f' %b)
1375 if b > f_y:
1376     print('This element is under too much stress')
1377
1378 b = (max(svm2))*10e-6
1379 print('rear_left_down_x2_svm2 = %.0f' %b)
1380 if b > f_y:
1381     print('This element is under too much stress')
1382
1383 b = (max(svm3))*10e-6
1384 print('rear_left_down_x2_svm3 = %.0f' %b)
1385 if b > f_y:
1386     print('This element is under too much stress')
1387
1388 b = (max(svm4))*10e-6
1389 print('rear_left_down_x2_svm4 = %.0f' %b)
1390 if b > f_y:
1391     print('This element is under too much stress')
1392
1393 b = (max(svm5))*10e-6
1394 print('rear_left_down_x2_svm5 = %.0f' %b)
1395 if b > f_y:
1396     print('This element is under too much stress')
1397
1398 b = (max(svm6))*10e-6
1399 print('rear_left_down_x2_svm6 = %.0f' %b)
1400 if b > f_y:
1401     print('This element is under too much stress')
1402
1403 b = (max(svm7))*10e-6
1404 print('rear_left_down_x2_svm7 = %.0f' %b)
1405 if b > f_y:

```



```
1406     print('This element is under too much stress')
1407
1408 b = (max(svm8))*10e-6
1409 print('rear_left_down_x2_svm8 = %.0f' %b)
1410 if b > f_y:
1411     print('This element is under too much stress')
1412
1413
1414
1415 figure()
1416 plot(t,svml/1.e6,'k-',label='')
1417 xlabel('time [s]')
1418 ylabel('rear_left_down_strut_fx_2_s1 [MPa] ')
1419 title('Axial Stress')
1420 #axis([-16.,90.,-1.5,1.7])
1421 #legend(loc='best')
1422 savefig('rear_left_down_strut_fx_2_s1',dpi=300)
1423
1424
1425
1426 figure()
1427 plot(t,svm2/1.e6,'k-',label='')
1428 xlabel('time [s]')
1429 ylabel('rear_left_down_strut_fx_2_s2 [MPa] ')
1430 title('Axial Stress')
1431 #axis([-16.,90.,-1.5,1.7])
1432 #legend(loc='best')
1433 savefig('rear_left_down_strut_fx_2_s2',dpi=300)
1434
1435
1436
1437 figure()
1438 plot(t,svm3/1.e6,'k-',label='')
1439 xlabel('time [s]')
1440 ylabel('rear_left_down_strut_fx_2_s3 [MPa] ')
1441 title('Axial Stress')
1442 #axis([-16.,90.,-1.5,1.7])
1443 #legend(loc='best')
1444 savefig('rear_left_down_strut_fx_2_s3',dpi=300)
1445
1446
1447
1448 figure()
1449 plot(t,svm4/1.e6,'k-',label='')
1450 xlabel('time [s]')
1451 ylabel('rear_left_down_strut_fx_2_s4 [MPa] ')
1452 title('Axial Stress')
1453 #axis([-16.,90.,-1.5,1.7])
1454 #legend(loc='best')
```

```
1455 savefig('rear_left_down_strut_fx_2_s4',dpi=300)
1456
1457
1458
1459 figure()
1460 plot(t,svm5/1.e6,'k-',label='')
1461 xlabel('time [s]')
1462 ylabel('rear_left_down_strut_fx_2_s5 [MPa] ')
1463 title('Axial Stress')
1464 #axis([-16.,90.,-1.5,1.7])
1465 #legend(loc='best')
1466 savefig('rear_left_down_strut_fx_2_s5',dpi=300)
1467
1468
1469
1470 figure()
1471 plot(t,svm6/1.e6,'k-',label='')
1472 xlabel('time [s]')
1473 ylabel('rear_left_down_strut_fx_2_s6 [MPa] ')
1474 title('Axial Stress')
1475 #axis([-16.,90.,-1.5,1.7])
1476 #legend(loc='best')
1477 savefig('rear_left_down_strut_fx_2_s6',dpi=300)
1478
1479
1480
1481 figure()
1482 plot(t,svm7/1.e6,'k-',label='')
1483 xlabel('time [s]')
1484 ylabel('rear_left_down_strut_fx_2_s7 [MPa] ')
1485 title('Axial Stress')
1486 #axis([-16.,90.,-1.5,1.7])
1487 #legend(loc='best')
1488 savefig('rear_left_down_strut_fx_2_s7',dpi=300)
1489
1490
1491
1492 figure()
1493 plot(t,svm8/1.e6,'k-',label='')
1494 xlabel('time [s]')
1495 ylabel('rear_left_down_strut_fx_2_s8 [MPa] ')
1496 title('Axial Stress')
1497 #axis([-16.,90.,-1.5,1.7])
1498 #legend(loc='best')
1499 savefig('rear_left_down_strut_fx_2_s8',dpi=300)
1500
1501
1502
1503 #
```

```
1504
1505 d = 0.2
1506 twall = 0.002
1507
1508 d = 0.2
1509 twall = 0.010
1510
1511 #rear_left_up_strut_x1
1512 #Joint type 3
1513 #crown in z-dir
1514 rear_left_up_strut_fx_1 = a[:,62]*1.e3;
1515 rear_left_up_strut_fy_1 = a[:,122]*1.e3;
1516 rear_left_up_strut_fz_1 = a[:,123]*1.e3;
1517 mx_rlus_1 = a[:,63]*1.e3;
1518 my_rlus_1 = a[:,64]*1.e3;
1519 mz_rlus_1 = a[:,65]*1.e3;
1520
1521 [sax , sbendy , sbendz , s1 , s2 , s3 , s4 , s5 , s6 , s7 , s8 , ssh1 , ssh2 , ssh3 , ssh4 , ssh5 , ssh6 ,
1522   ssh7 , ssh8 , svm1 , svm2 , svm3 , svm4 , svm5 , svm6 , svm7 , svm8 , sx , sy , sz ] = \
1523   cyl_beam_stresses(t , rear_left_up_strut_fx_1 , rear_left_up_strut_fy_1 ,
1524     rear_left_up_strut_fz_1 , mx_rlus_1 , my_rlus_1 , mz_rlus_1 , d , twall ,
1525     SCF_AC_3 , SCF_AS_3 , SCF_IPB_3 , SCF_OPB_3)
1526
1527 b = (max(svm1))*10e-6
1528 print('rear_left_up_x1_svm1 = %.0f' %b)
1529 if b > f_y:
1530     print('This element is under too much stress')
1531
1532 b = (max(svm2))*10e-6
1533 print('rear_left_up_x1_svm2 = %.0f' %b)
1534 if b > f_y:
1535     print('This element is under too much stress')
1536
1537 b = (max(svm3))*10e-6
1538 print('rear_left_up_x1_svm3 = %.0f' %b)
1539 if b > f_y:
1540     print('This element is under too much stress')
1541
1542 b = (max(svm4))*10e-6
1543 print('rear_left_up_x1_svm4 = %.0f' %b)
1544 if b > f_y:
1545     print('This element is under too much stress')
1546
1547 b = (max(svm5))*10e-6
1548 print('rear_left_up_x1_svm5 = %.0f' %b)
1549 if b > f_y:
```

```
1548     print('This element is under too much stress')
1549
1550 b = (max(svm6))*10e-6
1551 print('rear_left_up_x1_svm6 = %.0f' %b)
1552 if b > f_y:
1553     print('This element is under too much stress')
1554
1555 b = (max(svm7))*10e-6
1556 print('rear_left_up_x1_svm7 = %.0f' %b)
1557 if b > f_y:
1558     print('This element is under too much stress')
1559
1560 b = (max(svm8))*10e-6
1561 print('rear_left_up_x1_svm8 = %.0f' %b)
1562 if b > f_y:
1563     print('This element is under too much stress')
1564
1565
1566
1567 figure()
1568 plot(t,svm1/1.e6,'k-',label='')
1569 xlabel('time [s]')
1570 ylabel('rear_left_up_strut_fx_1_s1 [MPa] ')
1571 title('Axial Stress')
1572 #axis([-16.,90.,-1.5,1.7])
1573 #legend(loc='best')
1574 savefig('rear_left_up_strut_fx_1_s1',dpi=300)
1575
1576
1577 figure()
1578 plot(t,svm2/1.e6,'k-',label='')
1579 xlabel('time [s]')
1580 ylabel('rear_left_up_strut_fx_1_s2 [MPa] ')
1581 title('Axial Stress')
1582 #axis([-16.,90.,-1.5,1.7])
1583 #legend(loc='best')
1584 savefig('rear_left_up_strut_fx_1_s2',dpi=300)
1585
1586
1587
1588 figure()
1589 plot(t,svm3/1.e6,'k-',label='')
1590 xlabel('time [s]')
1591 ylabel('rear_left_up_strut_fx_1_s3 [MPa] ')
1592 title('Axial Stress')
1593 #axis([-16.,90.,-1.5,1.7])
1594 #legend(loc='best')
1595 savefig('rear_left_up_strut_fx_1_s3',dpi=300)
1596
```

```
1597
1598
1599 figure()
1600 plot(t,svm4/1.e6,'k-',label='')
1601 xlabel('time [s]')
1602 ylabel('rear_left_up_strut_fx_1_s4 [MPa] ')
1603 title('Axial Stress')
1604 #axis([-16.,90.,-1.5,1.7])
1605 #legend(loc='best')
1606 savefig('rear_left_up_strut_fx_1_s4',dpi=300)
1607
1608
1609
1610 figure()
1611 plot(t,svm5/1.e6,'k-',label='')
1612 xlabel('time [s]')
1613 ylabel('rear_left_up_strut_fx_1_s5 [MPa] ')
1614 title('Axial Stress')
1615 #axis([-16.,90.,-1.5,1.7])
1616 #legend(loc='best')
1617 savefig('rear_left_up_strut_fx_1_s5',dpi=300)
1618
1619
1620
1621 figure()
1622 plot(t,svm6/1.e6,'k-',label='')
1623 xlabel('time [s]')
1624 ylabel('rear_left_up_strut_fx_1_s6 [MPa] ')
1625 title('Axial Stress')
1626 #axis([-16.,90.,-1.5,1.7])
1627 #legend(loc='best')
1628 savefig('rear_left_up_strut_fx_1_s6',dpi=300)
1629
1630
1631
1632 figure()
1633 plot(t,svm7/1.e6,'k-',label='')
1634 xlabel('time [s]')
1635 ylabel('rear_left_up_strut_fx_1_s7 [MPa] ')
1636 title('Axial Stress')
1637 #axis([-16.,90.,-1.5,1.7])
1638 #legend(loc='best')
1639 savefig('rear_left_up_strut_fx_1_s7',dpi=300)
1640
1641
1642
1643 figure()
1644 plot(t,svm8/1.e6,'k-',label='')
1645 xlabel('time [s]')
```

```

1646 ylabel('rear_left_up_strut_fx_1_s8 [MPa] ')
1647 title('Axial Stress')
1648 #axis([-16.,90.,-1.5,1.7])
1649 #legend(loc='best')
1650 savefig('rear_left_up_strut_fx_1_s8',dpi=300)
1651
1652
1653
1654
1655 #

```

```

1656
1657 #rear_left_up_strut_x2
1658 #Joint type 4
1659 #crown in z-dir
1660 rear_left_up_strut_fx_2 = a[:,66]*1.e3;
1661 rear_left_up_strut_fy_2 = a[:,124]*1.e3;
1662 rear_left_up_strut_fz_2 = a[:,125]*1.e3;
1663 mx_rlus_2 = a[:,67]*1.e3;
1664 my_rlus_2 = a[:,68]*1.e3;
1665 mz_rlus_2 = a[:,69]*1.e3;
1666
1667 [sax, sbendy, sbendz, s1, s2, s3, s4, s5, s6, s7, s8, ssh1, ssh2, ssh3, ssh4, ssh5, ssh6,
    ssh7, ssh8, svm1, svm2, svm3, svm4, svm5, svm6, svm7, svm8, sx, sy, sz] = \
1668     cyl_beam_stresses(t, rear_left_up_strut_fx_2, rear_left_up_strut_fy_2,
    rear_left_up_strut_fz_2, mx_rlus_2, my_rlus_2, mz_rlus_2, d, twall,
    SCF_AC_4, SCF_AS_4, SCF_IPB_4, SCF_OPB_4)
1669
1670
1671
1672 b = (max(svm1))*10e-6
1673 print('rear_left_up_x2_svm1 = %.0f' %b)
1674 if b > f_y:
1675     print('This element is under too much stress')
1676
1677 b = (max(svm2))*10e-6
1678 print('rear_left_up_x2_svm2 = %.0f' %b)
1679 if b > f_y:
1680     print('This element is under too much stress')
1681
1682 b = (max(svm3))*10e-6
1683 print('rear_left_up_x2_svm3 = %.0f' %b)
1684 if b > f_y:
1685     print('This element is under too much stress')
1686
1687 b = (max(svm4))*10e-6
1688 print('rear_left_up_x2_svm4 = %.0f' %b)
1689 if b > f_y:

```

```
1690     print('This element is under too much stress')
1691
1692 b = (max(svm5))*10e-6
1693 print('rear_left_up_x2_svm5 = %.0f' %b)
1694 if b > f_y:
1695     print('This element is under too much stress')
1696
1697 b = (max(svm6))*10e-6
1698 print('rear_left_up_x2_svm6 = %.0f' %b)
1699 if b > f_y:
1700     print('This element is under too much stress')
1701
1702 b = (max(svm7))*10e-6
1703 print('rear_left_up_x2_svm7 = %.0f' %b)
1704 if b > f_y:
1705     print('This element is under too much stress')
1706
1707 b = (max(svm8))*10e-6
1708 print('rear_left_up_x2_svm8 = %.0f' %b)
1709 if b > f_y:
1710     print('This element is under too much stress')
1711
1712
1713
1714
1715 figure()
1716 plot(t,svm1/1.e6,'k-',label='')
1717 xlabel('time [s]')
1718 ylabel('rear_left_up_strut_fx_2_s1 [MPa] ')
1719 title('Axial Stress')
1720 #axis([-16.,90.,-1.5,1.7])
1721 #legend(loc='best')
1722 savefig('rear_left_up_strut_fx_2_s1',dpi=300)
1723
1724
1725
1726 figure()
1727 plot(t,svm2/1.e6,'k-',label='')
1728 xlabel('time [s]')
1729 ylabel('rear_left_up_strut_fx_2_s2 [MPa] ')
1730 title('Axial Stress')
1731 #axis([-16.,90.,-1.5,1.7])
1732 #legend(loc='best')
1733 savefig('rear_left_up_strut_fx_2_s2',dpi=300)
1734
1735
1736
1737 figure()
1738 plot(t,svm3/1.e6,'k-',label='')
```

```
1739 xlabel('time [s]')
1740 ylabel('rear_left_up_strut_fx_2_s3 [MPa] ')
1741 title('Axial Stress')
1742 #axis([-16.,90.,-1.5,1.7])
1743 #legend(loc='best')
1744 savefig('rear_left_up_strut_fx_2_s3',dpi=300)
1745
1746
1747 figure()
1748 plot(t,svm4/1.e6,'k-',label='')
1749 xlabel('time [s]')
1750 ylabel('rear_left_up_strut_fx_2_s4 [MPa] ')
1751 title('Axial Stress')
1752 #axis([-16.,90.,-1.5,1.7])
1753 #legend(loc='best')
1754 savefig('rear_left_up_strut_fx_2_s4',dpi=300)
1755
1756
1757
1758 figure()
1759 plot(t,svm5/1.e6,'k-',label='')
1760 xlabel('time [s]')
1761 ylabel('rear_left_up_strut_fx_2_s5 [MPa] ')
1762 title('Axial Stress')
1763 #axis([-16.,90.,-1.5,1.7])
1764 #legend(loc='best')
1765 savefig('rear_left_up_strut_fx_2_s5',dpi=300)
1766
1767
1768
1769 figure()
1770 plot(t,svm6/1.e6,'k-',label='')
1771 xlabel('time [s]')
1772 ylabel('rear_left_up_strut_fx_2_s6 [MPa] ')
1773 title('Axial Stress')
1774 #axis([-16.,90.,-1.5,1.7])
1775 #legend(loc='best')
1776 savefig('rear_left_up_strut_fx_2_s6',dpi=300)
1777
1778
1779
1780 figure()
1781 plot(t,svm7/1.e6,'k-',label='')
1782 xlabel('time [s]')
1783 ylabel('rear_left_up_strut_fx_2_s7 [MPa] ')
1784 title('Axial Stress')
1785 #axis([-16.,90.,-1.5,1.7])
1786 #legend(loc='best')
1787 savefig('rear_left_up_strut_fx_2_s7',dpi=300)
```



```

1788
1789
1790
1791 figure()
1792 plot(t,svm8/1.e6,'k-',label='')
1793 xlabel('time [s]')
1794 ylabel('rear_left_up_strut_fx_2_s8 [MPa]')
1795 title('Axial Stress')
1796 #axis([-16.,90.,-1.5,1.7])
1797 #legend(loc='best')
1798 savefig('rear_left_up_strut_fx_2_s8',dpi=300)
1799
1800 #

```

```

1801
1802 #rear_right_down_strut_x1
1803 #Joint type 3
1804 #crown in z-dir
1805 rear_right_down_strut_fx_1 = a[:,70]*1.e3;
1806 rear_right_down_strut_fy_1 = a[:,126]*1.e3;
1807 rear_right_down_strut_fz_1 = a[:,127]*1.e3;
1808 mx_rrds_1 = a[:,71]*1.e3;
1809 my_rrds_1 = a[:,72]*1.e3;
1810 mz_rrds_1 = a[:,73]*1.e3;
1811
1812 [sax, sbendy, sbendz, s1, s2, s3, s4, s5, s6, s7, s8, ssh1, ssh2, ssh3, ssh4, ssh5, ssh6,
    ssh7, ssh8, svm1, svm2, svm3, svm4, svm5, svm6, svm7, svm8, sx, sy, sz] = \
1813 cyl_beam_stresses(t, rear_right_down_strut_fx_1,
    rear_right_down_strut_fy_1, rear_right_down_strut_fy_1, mx_rrds_1,
    my_rrds_1, mz_rrds_1, d, twall, SCF_AC_3, SCF_AS_3, SCF_IPB_3, SCF_OPB_3)
1814
1815
1816 b = (max(svm1))*10e-6
1817 print('rear_right_down_x1_svm1 = %.0f' %b)
1818 if b > f_y:
1819     print('This element is under too much stress')
1820
1821 b = (max(svm2))*10e-6
1822 print('rear_right_down_x1_svm2 = %.0f' %b)
1823 if b > f_y:
1824     print('This element is under too much stress')
1825
1826 b = (max(svm3))*10e-6
1827 print('rear_right_down_svm3 = %.0f' %b)
1828 if b > f_y:
1829     print('This element is under too much stress')
1830
1831 b = (max(svm4))*10e-6

```

```
1832 print('rear_right_down_x1_svm4 = %.0f' %b)
1833 if b > f_y:
1834     print('This element is under too much stress')
1835
1836 b = (max(svm5))*10e-6
1837 print('rear_right_down_x1_svm5 = %.0f' %b)
1838 if b > f_y:
1839     print('This element is under too much stress')
1840
1841 b = (max(svm6))*10e-6
1842 print('rear_right_down_x1_svm6 = %.0f' %b)
1843 if b > f_y:
1844     print('This element is under too much stress')
1845
1846 b = (max(svm7))*10e-6
1847 print('rear_right_down_x1_svm7 = %.0f' %b)
1848 if b > f_y:
1849     print('This element is under too much stress')
1850
1851 b = (max(svm8))*10e-6
1852 print('rear_right_down_x1_svm8 = %.0f' %b)
1853 if b > f_y:
1854     print('This element is under too much stress')
1855
1856
1857
1858 figure()
1859 plot(t,svml/1.e6,'k-',label='')
1860 xlabel('time [s]')
1861 ylabel('rear_right_down_strut_fx_1_s1 [MPa] ')
1862 title('Axial Stress')
1863 #axis([-16.,90.,-1.5,1.7])
1864 #legend(loc='best')
1865 savefig('rear_right_down_strut_fx_1_s1',dpi=300)
1866
1867
1868
1869 figure()
1870 plot(t,svm2/1.e6,'k-',label='')
1871 xlabel('time [s]')
1872 ylabel('rear_right_down_strut_fx_1_s2 [MPa] ')
1873 title('Axial Stress')
1874 #axis([-16.,90.,-1.5,1.7])
1875 #legend(loc='best')
1876 savefig('rear_right_down_strut_fx_1_s2',dpi=300)
1877
1878
1879
1880 figure()
```

```
1881 plot(t,svm3/1.e6,'k-',label='')
1882 xlabel('time [s]')
1883 ylabel('rear_right_down_strut_fx_1_s3 [MPa] ')
1884 title('Axial Stress')
1885 #axis([-16.,90.,-1.5,1.7])
1886 #legend(loc='best')
1887 savefig('rear_right_down_strut_fx_1_s3',dpi=300)
1888
1889
1890
1891 figure()
1892 plot(t,svm4/1.e6,'k-',label='')
1893 xlabel('time [s]')
1894 ylabel('rear_right_down_strut_fx_1_s4 [MPa] ')
1895 title('Axial Stress')
1896 #axis([-16.,90.,-1.5,1.7])
1897 #legend(loc='best')
1898 savefig('rear_right_down_strut_fx_1_s4',dpi=300)
1899
1900
1901
1902 figure()
1903 plot(t,svm5/1.e6,'k-',label='')
1904 xlabel('time [s]')
1905 ylabel('rear_right_down_strut_fx_1_s5 [MPa] ')
1906 title('Axial Stress')
1907 #axis([-16.,90.,-1.5,1.7])
1908 #legend(loc='best')
1909 savefig('rear_right_down_strut_fx_1_s5',dpi=300)
1910
1911
1912
1913 figure()
1914 plot(t,svm6/1.e6,'k-',label='')
1915 xlabel('time [s]')
1916 ylabel('rear_right_down_strut_fx_1_s6 [MPa] ')
1917 title('Axial Stress')
1918 #axis([-16.,90.,-1.5,1.7])
1919 #legend(loc='best')
1920 savefig('rear_right_down_strut_fx_1_s6',dpi=300)
1921
1922
1923
1924 figure()
1925 plot(t,svm7/1.e6,'k-',label='')
1926 xlabel('time [s]')
1927 ylabel('rear_right_down_strut_fx_1_s7 [MPa] ')
1928 title('Axial Stress')
1929 #axis([-16.,90.,-1.5,1.7])
```

```

1930 #legend(loc='best')
1931 savefig('rear_right_down_strut_fx_1_s7',dpi=300)
1932
1933
1934
1935 figure()
1936 plot(t,svm8/1.e6,'k-',label='')
1937 xlabel('time [s]')
1938 ylabel('rear_right_down_strut_fx_1_s8 [MPa]')
1939 title('Axial Stress')
1940 #axis([-16.,90.,-1.5,1.7])
1941 #legend(loc='best')
1942 savefig('rear_right_down_strut_fx_1_s8',dpi=300)
1943
1944 #

```

```

1945
1946 #rear_right_down_strut_x2
1947 #Joint type 4
1948 #crown in z-dir
1949 rear_right_down_strut_fx_2 = a[:,74]*1.e3;
1950 rear_right_down_strut_fy_2 = a[:,128]*1.e3;
1951 rear_right_down_strut_fz_2 = a[:,129]*1.e3;
1952 mx_rrds_2 = a[:,75]*1.e3;
1953 my_rrds_2 = a[:,76]*1.e3;
1954 mz_rrds_2 = a[:,77]*1.e3;
1955
1956 [sax,sbendy,sbendz,s1,s2,s3,s4,s5,s6,s7,s8,ssh1,ssh2,ssh3,ssh4,ssh5,ssh6,
    ssh7,ssh8,svm1,svm2,svm3,svm4,svm5,svm6,svm7,svm8,sx,sy,sz] = \
1957     cyl_beam_stresses(t,rear_right_down_strut_fx_2,
        rear_right_down_strut_fy_2,rear_right_down_strut_fz_2,mx_rrds_2,
        my_rrds_2,mz_rrds_2,d,twall,SCF_AC_4,SCF_AS_4,SCF_IPB_4,SCF_OPB_4)
1958
1959
1960 b = (max(svm1))*10e-6
1961 print('rear_right_down_x2_svm1 = %.0f' %b)
1962 if b > f_y:
1963     print('This element is under too much stress')
1964
1965 b = (max(svm2))*10e-6
1966 print('rear_right_down_x2_svm2 = %.0f' %b)
1967 if b > f_y:
1968     print('This element is under too much stress')
1969
1970 b = (max(svm3))*10e-6
1971 print('rear_right_down_x2_svm3 = %.0f' %b)
1972 if b > f_y:
1973     print('This element is under too much stress')

```

```
1974
1975 b = (max(svm4))*10e-6
1976 print('rear_right_down_x2_svm4 = %.0f' %b)
1977 if b > f_y:
1978     print('This element is under too much stress')
1979
1980 b = (max(svm5))*10e-6
1981 print('rear_right_down_x2_svm5 = %.0f' %b)
1982 if b > f_y:
1983     print('This element is under too much stress')
1984
1985 b = (max(svm6))*10e-6
1986 print('rear_right_down_x2_svm6 = %.0f' %b)
1987 if b > f_y:
1988     print('This element is under too much stress')
1989
1990 b = (max(svm7))*10e-6
1991 print('rear_right_down_x2_svm7 = %.0f' %b)
1992 if b > f_y:
1993     print('This element is under too much stress')
1994
1995 b = (max(svm8))*10e-6
1996 print('rear_right_down_x2_svm8 = %.0f' %b)
1997 if b > f_y:
1998     print('This element is under too much stress')
1999
2000
2001
2002
2003 figure()
2004 plot(t,svm1/1.e6,'k-',label='')
2005 xlabel('time [s]')
2006 ylabel('rear_right_down_strut_fx_2_s1 [MPa] ')
2007 title('Axial Stress')
2008 #axis([-16.,90.,-1.5,1.7])
2009 #legend(loc='best')
2010 savefig('rear_right_down_strut_fx_2_s1',dpi=300)
2011
2012
2013
2014 figure()
2015 plot(t,svm2/1.e6,'k-',label='')
2016 xlabel('time [s]')
2017 ylabel('rear_right_down_strut_fx_2_s2 [MPa] ')
2018 title('Axial Stress')
2019 #axis([-16.,90.,-1.5,1.7])
2020 #legend(loc='best')
2021 savefig('rear_right_down_strut_fx_2_s2',dpi=300)
2022
```

```
2023
2024
2025 figure()
2026 plot(t,svm3/1.e6,'k-',label='')
2027 xlabel('time [s]')
2028 ylabel('rear_right_down_strut_fx_2_s3 [MPa]')
2029 title('Axial Stress')
2030 #axis([-16.,90.,-1.5,1.7])
2031 #legend(loc='best')
2032 savefig('rear_right_down_strut_fx_2_s3',dpi=300)
2033
2034
2035
2036 figure()
2037 plot(t,svm4/1.e6,'k-',label='')
2038 xlabel('time [s]')
2039 ylabel('rear_right_down_strut_fx_2_s4 [MPa]')
2040 title('Axial Stress')
2041 #axis([-16.,90.,-1.5,1.7])
2042 #legend(loc='best')
2043 savefig('rear_right_down_strut_fx_2_s4',dpi=300)
2044
2045
2046
2047 figure()
2048 plot(t,svm5/1.e6,'k-',label='')
2049 xlabel('time [s]')
2050 ylabel('rear_right_down_strut_fx_2_s5 [MPa]')
2051 title('Axial Stress')
2052 #axis([-16.,90.,-1.5,1.7])
2053 #legend(loc='best')
2054 savefig('rear_right_down_strut_fx_2_s5',dpi=300)
2055
2056
2057 figure()
2058 plot(t,svm6/1.e6,'k-',label='')
2059 xlabel('time [s]')
2060 ylabel('rear_right_down_strut_fx_2_s6 [MPa]')
2061 title('Axial Stress')
2062 #axis([-16.,90.,-1.5,1.7])
2063 #legend(loc='best')
2064 savefig('rear_right_down_strut_fx_2_s6',dpi=300)
2065
2066
2067
2068 figure()
2069 plot(t,svm7/1.e6,'k-',label='')
2070 xlabel('time [s]')
2071 ylabel('rear_right_down_strut_fx_2_s7 [MPa]')
```

```

2072 title('Axial Stress')
2073 #axis([-16.,90.,-1.5,1.7])
2074 #legend(loc='best')
2075 savefig('rear_right_down_strut_fx_2_s7',dpi=300)
2076
2077
2078
2079 figure()
2080 plot(t,svm8/1.e6,'k-',label='')
2081 xlabel('time [s]')
2082 ylabel('rear_right_down_strut_fx_2_s8 [MPa]')
2083 title('Axial Stress')
2084 #axis([-16.,90.,-1.5,1.7])
2085 #legend(loc='best')
2086 savefig('rear_right_down_strut_fx_2_s8',dpi=300)
2087
2088 #

```

```

2089
2090 d = 0.2
2091 twall = 0.008
2092
2093 #rear_right_up_strut_x1
2094 #Joint type 3
2095 #crown in z-dir
2096 rear_right_up_strut_fx_1 = a[:,78]*1.e3;
2097 rear_right_up_strut_fy_1 = a[:,130]*1.e3;
2098 rear_right_up_strut_fz_1 = a[:,131]*1.e3;
2099 mx_rrus_1 = a[:,79]*1.e3;
2100 my_rrus_1 = a[:,80]*1.e3;
2101 mz_rrus_1 = a[:,81]*1.e3;
2102
2103 [sax,sbendy,sbendz,s1,s2,s3,s4,s5,s6,s7,s8,ssh1,ssh2,ssh3,ssh4,ssh5,ssh6,
    ssh7,ssh8,svm1,svm2,svm3,svm4,svm5,svm6,svm7,svm8,sx,sy,sz] = \
2104     cyl_beam_stresses(t,rear_right_up_strut_fx_1,rear_right_up_strut_fy_1,
    rear_right_up_strut_fz_1,mx_rrus_1,my_rrus_1,mz_rrus_1,d,twall,
    SCF_AC_3,SCF_AS_3,SCF_IPB_3,SCF_OPB_3)
2105
2106
2107 b = (max(svm1))*10e-6
2108 print('rear_right_up_x1_svm1 = %.0f' %b)
2109 if b > f_y:
2110     print('This element is under too much stress')
2111
2112 b = (max(svm2))*10e-6
2113 print('rear_right_up_x1_svm2 = %.0f' %b)
2114 if b > f_y:
2115     print('This element is under too much stress')

```

```
2116
2117 b = (max(svm3))*10e-6
2118 print('rear_right_up_svm3 = %.0f' %b)
2119 if b > f_y:
2120     print('This element is under too much stress')
2121
2122 b = (max(svm4))*10e-6
2123 print('rear_right_up_x1_svm4 = %.0f' %b)
2124 if b > f_y:
2125     print('This element is under too much stress')
2126
2127 b = (max(svm5))*10e-6
2128 print('rear_right_up_x1_svm5 = %.0f' %b)
2129 if b > f_y:
2130     print('This element is under too much stress')
2131
2132 b = (max(svm6))*10e-6
2133 print('rear_right_up_x1_svm6 = %.0f' %b)
2134 if b > f_y:
2135     print('This element is under too much stress')
2136
2137 b = (max(svm7))*10e-6
2138 print('rear_right_up_x1_svm7 = %.0f' %b)
2139 if b > f_y:
2140     print('This element is under too much stress')
2141
2142 b = (max(svm8))*10e-6
2143 print('rear_right_up_x1_svm8 = %.0f' %b)
2144 if b > f_y:
2145     print('This element is under too much stress')
2146
2147
2148
2149 figure()
2150 plot(t,svml/1.e6,'k-',label='')
2151 xlabel('time [s]')
2152 ylabel('rear_right_up_strut_fx_1_s1 [MPa] ')
2153 title('Axial Stress')
2154 #axis([-16.,90.,-1.5,1.7])
2155 #legend(loc='best')
2156 savefig('rear_right_up_strut_fx_1_s1',dpi=300)
2157
2158
2159
2160 figure()
2161 plot(t,svm2/1.e6,'k-',label='')
2162 xlabel('time [s]')
2163 ylabel('rear_right_up_strut_fx_1_s2 [MPa] ')
2164 title('Axial Stress')
```



```
2165 #axis([-16.,90.,-1.5,1.7])
2166 #legend(loc='best')
2167 savefig('rear_right_up_strut_fx_1_s2',dpi=300)
2168
2169
2170 figure()
2171 plot(t,svm3/1.e6,'k-',label='')
2172 xlabel('time [s]')
2173 ylabel('rear_right_up_strut_fx_1_s3 [MPa]')
2174 title('Axial Stress')
2175 #axis([-16.,90.,-1.5,1.7])
2176 #legend(loc='best')
2177 savefig('rear_right_up_strut_fx_1_s3',dpi=300)
2178
2179
2180
2181 figure()
2182 plot(t,svm4/1.e6,'k-',label='')
2183 xlabel('time [s]')
2184 ylabel('rear_right_up_strut_fx_1_s4 [MPa]')
2185 title('Axial Stress')
2186 #axis([-16.,90.,-1.5,1.7])
2187 #legend(loc='best')
2188 savefig('rear_right_up_strut_fx_1_s4',dpi=300)
2189
2190
2191
2192 figure()
2193 plot(t,svm5/1.e6,'k-',label='')
2194 xlabel('time [s]')
2195 ylabel('rear_right_up_strut_fx_1_s5 [MPa]')
2196 title('Axial Stress')
2197 #axis([-16.,90.,-1.5,1.7])
2198 #legend(loc='best')
2199 savefig('rear_right_up_strut_fx_1_s5',dpi=300)
2200
2201
2202
2203 figure()
2204 plot(t,svm6/1.e6,'k-',label='')
2205 xlabel('time [s]')
2206 ylabel('rear_right_up_strut_fx_1_s6 [MPa]')
2207 title('Axial Stress')
2208 #axis([-16.,90.,-1.5,1.7])
2209 #legend(loc='best')
2210 savefig('rear_right_up_strut_fx_1_s6',dpi=300)
2211
2212
2213
```

```

2214 figure()
2215 plot(t,svm7/1.e6,'k-',label='')
2216 xlabel('time [s]')
2217 ylabel('rear_right_up_strut_fx_1_s7 [MPa]')
2218 title('Axial Stress')
2219 #axis([-16.,90.,-1.5,1.7])
2220 #legend(loc='best')
2221 savefig('rear_right_up_strut_fx_1_s7',dpi=300)
2222
2223
2224
2225 figure()
2226 plot(t,svm8/1.e6,'k-',label='')
2227 xlabel('time [s]')
2228 ylabel('rear_right_up_strut_fx_1_s8 [MPa]')
2229 title('Axial Stress')
2230 #axis([-16.,90.,-1.5,1.7])
2231 #legend(loc='best')
2232 savefig('rear_right_up_strut_fx_1_s8',dpi=300)
2233
2234 #

```

```

2235
2236 #rear_right_up_strut_x2
2237 #Joint type 4
2238 #crown in z-dir
2239 rear_right_up_strut_fx_2 = a[:,82]*1.e3;
2240 rear_right_up_strut_fy_2 = a[:,132]*1.e3;
2241 rear_right_up_strut_fz_2 = a[:,133]*1.e3;
2242 mx_rrus_2 = a[:,83]*1.e3;
2243 my_rrus_2 = a[:,84]*1.e3;
2244 mz_rrus_2 = a[:,85]*1.e3;
2245
2246 [sax,sbendy,sbendz,s1,s2,s3,s4,s5,s6,s7,s8,ssh1,ssh2,ssh3,ssh4,ssh5,ssh6,
    ssh7,ssh8,svm1,svm2,svm3,svm4,svm5,svm6,svm7,svm8,sx,sy,sz] = \
2247     cyl_beam_stresses(t,rear_right_up_strut_fx_2,rear_right_up_strut_fy_2,
    rear_right_up_strut_fz_2,mx_rrus_2,my_rrus_2,mz_rrus_2,d,twall,
    SCF_AC_4,SCF_AS_4,SCF_IPB_4,SCF_OPB_4)
2248
2249
2250 b = (max(svm1))*10e-6
2251 print('rear_right_up_x2_svm1 = %.0f' %b)
2252 if b > f_y:
2253     print('This element is under too much stress')
2254
2255 b = (max(svm2))*10e-6
2256 print('rear_right_up_x2_svm2 = %.0f' %b)
2257 if b > f_y:

```

```
2258     print('This element is under too much stress')
2259
2260 b = (max(svm3))*10e-6
2261 print('rear_right_up_x2_svm3 = %.0f' %b)
2262 if b > f_y:
2263     print('This element is under too much stress')
2264
2265 b = (max(svm4))*10e-6
2266 print('rear_right_up_x2_svm4 = %.0f' %b)
2267 if b > f_y:
2268     print('This element is under too much stress')
2269
2270 b = (max(svm5))*10e-6
2271 print('rear_right_up_x2_svm5 = %.0f' %b)
2272 if b > f_y:
2273     print('This element is under too much stress')
2274
2275 b = (max(svm6))*10e-6
2276 print('rear_right_up_x2_svm6 = %.0f' %b)
2277 if b > f_y:
2278     print('This element is under too much stress')
2279
2280 b = (max(svm7))*10e-6
2281 print('rear_right_up_x2_svm7 = %.0f' %b)
2282 if b > f_y:
2283     print('This element is under too much stress')
2284
2285 b = (max(svm8))*10e-6
2286 print('rear_right_up_x2_svm8 = %.0f' %b)
2287 if b > f_y:
2288     print('This element is under too much stress')
2289
2290
2291 figure()
2292 plot(t,svm1/1.e6,'k-',label='')
2293 xlabel('time [s]')
2294 ylabel('rear_right_up_strut_fx_2_s1 [MPa]')
2295 title('Axial Stress')
2296 #axis([-16.,90.,-1.5,1.7])
2297 #legend(loc='best')
2298 savefig('rear_right_up_strut_fx_2_s1',dpi=300)
2299
2300
2301
2302 figure()
2303 plot(t,svm2/1.e6,'k-',label='')
2304 xlabel('time [s]')
2305 ylabel('rear_right_up_strut_fx_2_s2 [MPa]')
2306 title('Axial Stress')
```

```
2307 #axis([-16.,90.,-1.5,1.7])
2308 #legend(loc='best')
2309 savefig('rear_right_up_strut_fx_2_s2',dpi=300)
2310
2311
2312
2313 figure()
2314 plot(t,svm3/1.e6,'k-',label='')
2315 xlabel('time [s]')
2316 ylabel('rear_right_up_strut_fx_2_s3 [MPa]')
2317 title('Axial Stress')
2318 #axis([-16.,90.,-1.5,1.7])
2319 #legend(loc='best')
2320 savefig('rear_right_up_strut_fx_2_s3',dpi=300)
2321
2322
2323
2324 figure()
2325 plot(t,svm4/1.e6,'k-',label='')
2326 xlabel('time [s]')
2327 ylabel('rear_right_up_strut_fx_2_s4 [MPa]')
2328 title('Axial Stress')
2329 #axis([-16.,90.,-1.5,1.7])
2330 #legend(loc='best')
2331 savefig('rear_right_up_strut_fx_2_s4',dpi=300)
2332
2333
2334 figure()
2335 plot(t,svm5/1.e6,'k-',label='')
2336 xlabel('time [s]')
2337 ylabel('rear_right_up_strut_fx_2_s5 [MPa]')
2338 title('Axial Stress')
2339 #axis([-16.,90.,-1.5,1.7])
2340 #legend(loc='best')
2341 savefig('rear_right_up_strut_fx_2_s5',dpi=300)
2342
2343
2344 figure()
2345 plot(t,svm6/1.e6,'k-',label='')
2346 xlabel('time [s]')
2347 ylabel('rear_right_up_strut_fx_2_s6 [MPa]')
2348 title('Axial Stress')
2349 #axis([-16.,90.,-1.5,1.7])
2350 #legend(loc='best')
2351 savefig('rear_right_up_strut_fx_2_s6',dpi=300)
2352
2353
2354 figure()
2355 plot(t,svm7/1.e6,'k-',label='')
```

```

2356 xlabel('time [s]')
2357 ylabel('rear_right_up_strut_fx_2_s7 [MPa] ')
2358 title('Axial Stress')
2359 #axis([-16.,90.,-1.5,1.7])
2360 #legend(loc='best')
2361 savefig('rear_right_up_strut_fx_2_s7',dpi=300)
2362
2363
2364
2365 figure()
2366 plot(t,svm8/1.e6,'k-',label='')
2367 xlabel('time [s]')
2368 ylabel('rear_right_up_strut_fx_2_s8 [MPa] ')
2369 title('Axial Stress')
2370 #axis([-16.,90.,-1.5,1.7])
2371 #legend(loc='best')
2372 savefig('rear_right_up_strut_fx_2_s8',dpi=300)
2373
2374 #

```

```

2375
2376
2377
2378 #

```

```

2379 d = 0.2
2380 twall = 0.016
2381
2382 #rear_vert_down_strut_x2
2383 #Joint type 2
2384 #saddle in z-dir
2385 rear_vert_down_strut_fx_2 = a[:,90]*1.e3;
2386 rear_vert_down_strut_fy_2 = a[:,136]*1.e3;
2387 rear_vert_down_strut_fz_2 = a[:,137]*1.e3;
2388 mx_rvds_2 = a[:,91]*1.e3;
2389 my_rvds_2 = a[:,92]*1.e3;
2390 mz_rvds_2 = a[:,93]*1.e3;
2391
2392 [sax ,sbendy ,sbendz ,s1 ,s2 ,s3 ,s4 ,s5 ,s6 ,s7 ,s8 ,ssh1 ,ssh2 ,ssh3 ,ssh4 ,ssh5 ,ssh6 ,
    ssh7 ,ssh8 ,svm1 ,svm2 ,svm3 ,svm4 ,svm5 ,svm6 ,svm7 ,svm8 ,sx ,sy ,sz ] = \
2393     cyl_beam_stresses(t ,rear_vert_down_strut_fx_2 ,
    rear_vert_down_strut_fy_2 ,rear_vert_down_strut_fz_2 ,mx_rvds_2 ,
    my_rvds_2 ,mz_rvds_2 ,d ,twall ,SCF_AS_2 ,SCF_AC_2 ,SCF_OPB_2 ,SCF_IPB_2)
2394
2395 b = (max(svm1))*10e-6
2396 print('rear_vert_down_svm1 = %.0f' %b)
2397 if b > f_y:

```

```
2398     print('This element is under too much stress')
2399
2400 b = (max(svm2))*10e-6
2401 print('rear_vert_down_svm2 = %.0f' %b)
2402 if b > f_y:
2403     print('This element is under too much stress')
2404
2405 b = (max(svm3))*10e-6
2406 print('rear_vert_down_svm3 = %.0f' %b)
2407 if b > f_y:
2408     print('This element is under too much stress')
2409
2410 b = (max(svm4))*10e-6
2411 print('rear_vert_down_svm4 = %.0f' %b)
2412 if b > f_y:
2413     print('This element is under too much stress')
2414
2415 b = (max(svm5))*10e-6
2416 print('rear_vert_down_svm5 = %.0f' %b)
2417 if b > f_y:
2418     print('This element is under too much stress')
2419
2420 b = (max(svm6))*10e-6
2421 print('rear_vert_down_svm6 = %.0f' %b)
2422 if b > f_y:
2423     print('This element is under too much stress')
2424
2425 b = (max(svm7))*10e-6
2426 print('rear_vert_down_svm7 = %.0f' %b)
2427 if b > f_y:
2428     print('This element is under too much stress')
2429
2430 b = (max(svm8))*10e-6
2431 print('rear_vert_down_svm8 = %.0f' %b)
2432 if b > f_y:
2433     print('This element is under too much stress')
2434
2435
2436 figure()
2437 plot(t, svm1/1.e6, 'k-', label='')
2438 xlabel('time [s]')
2439 ylabel('rear_vert_down_strut_fx_2_s1 [MPa] ')
2440 title('Axial Stress')
2441 #axis([-16., 90., -1.5, 1.7])
2442 #legend(loc='best')
2443 savefig('rear_vert_down_strut_fx_2_s1', dpi=300)
2444
2445
2446
```

```
2447 figure()
2448 plot(t,svm2/1.e6,'k-',label='')
2449 xlabel('time [s]')
2450 ylabel('rear_vert_down_strut_fx_2_s2 [MPa] ')
2451 title('Axial Stress')
2452 #axis([-16.,90.,-1.5,1.7])
2453 #legend(loc='best')
2454 savefig('rear_vert_down_strut_fx_2_s2',dpi=300)
2455
2456
2457
2458 figure()
2459 plot(t,svm3/1.e6,'k-',label='')
2460 xlabel('time [s]')
2461 ylabel('rear_vert_down_strut_fx_2_s3 [MPa] ')
2462 title('Axial Stress')
2463 #axis([-16.,90.,-1.5,1.7])
2464 #legend(loc='best')
2465 savefig('rear_vert_down_strut_fx_2_s3',dpi=300)
2466
2467
2468
2469 figure()
2470 plot(t,svm4/1.e6,'k-',label='')
2471 xlabel('time [s]')
2472 ylabel('rear_vert_down_strut_fx_2_s4 [MPa] ')
2473 title('Axial Stress')
2474 #axis([-16.,90.,-1.5,1.7])
2475 #legend(loc='best')
2476 savefig('rear_vert_down_strut_fx_2_s4',dpi=300)
2477
2478
2479
2480 figure()
2481 plot(t,svm5/1.e6,'k-',label='')
2482 xlabel('time [s]')
2483 ylabel('rear_vert_down_strut_fx_2_s5 [MPa] ')
2484 title('Axial Stress')
2485 #axis([-16.,90.,-1.5,1.7])
2486 #legend(loc='best')
2487 savefig('rear_vert_down_strut_fx_2_s5',dpi=300)
2488
2489
2490
2491 figure()
2492 plot(t,svm6/1.e6,'k-',label='')
2493 xlabel('time [s]')
2494 ylabel('rear_vert_down_strut_fx_2_s6 [MPa] ')
2495 title('Axial Stress')
```

```

2496 #axis([-16.,90.,-1.5,1.7])
2497 #legend(loc='best')
2498 savefig('rear_vert_down_strut_fx_2_s6',dpi=300)
2499
2500
2501
2502 figure()
2503 plot(t,svm7/1.e6,'k-',label='')
2504 xlabel('time [s]')
2505 ylabel('rear_vert_down_strut_fx_2_s7 [MPa]')
2506 title('Axial Stress')
2507 #axis([-16.,90.,-1.5,1.7])
2508 #legend(loc='best')
2509 savefig('rear_vert_down_strut_fx_2_s7',dpi=300)
2510
2511
2512
2513 figure()
2514 plot(t,svm8/1.e6,'k-',label='')
2515 xlabel('time [s]')
2516 ylabel('rear_vert_down_strut_fx_2_s8 [MPa]')
2517 title('Axial Stress')
2518 #axis([-16.,90.,-1.5,1.7])
2519 #legend(loc='best')
2520 savefig('rear_vert_down_strut_fx_2_s8',dpi=300)
2521
2522 #

```

```

2523
2524
2525 #

```

```

2526
2527 #rear_vert_up_strut_x2
2528 #Joint type 2
2529 #saddle in z-dir
2530 rear_vert_up_strut_fx_2 = a[:,98]*1.e3;
2531 rear_vert_up_strut_fy_2 = a[:,140]*1.e3;
2532 rear_vert_up_strut_fz_2 = a[:,141]*1.e3;
2533 mx_rvus_2 = a[:,99]*1.e3;
2534 my_rvus_2 = a[:,100]*1.e3;
2535 mz_rvus_2 = a[:,101]*1.e3;
2536
2537 [sax,sbendy,sbendz,s1,s2,s3,s4,s5,s6,s7,s8,ssh1,ssh2,ssh3,ssh4,ssh5,ssh6,
    ssh7,ssh8,svm1,svm2,svm3,svm4,svm5,svm6,svm7,svm8,sx,sy,sz] = \
2538     cyl_beam_stresses(t,rear_vert_up_strut_fx_2,rear_vert_up_strut_fy_2,
    rear_vert_up_strut_fz_2,mx_rvus_2,my_rvus_2,mz_rvus_2,d,twall,

```



```
SCF_AS_2,SCF_AC_2,SCF_OPB_2,SCF_IPB_2)
2539
2540
2541 b = (max(svm1))*10e-6
2542 print('rear_vert_up_svm1 = %.0f' %b)
2543 if b > f_y:
2544     print('This element is under too much stress')
2545
2546 b = (max(svm2))*10e-6
2547 print('rear_vert_up_svm2 = %.0f' %b)
2548 if b > f_y:
2549     print('This element is under too much stress')
2550
2551 b = (max(svm3))*10e-6
2552 print('rear_vert_up_svm3 = %.0f' %b)
2553 if b > f_y:
2554     print('This element is under too much stress')
2555
2556 b = (max(svm4))*10e-6
2557 print('rear_vert_up_svm4 = %.0f' %b)
2558 if b > f_y:
2559     print('This element is under too much stress')
2560
2561 b = (max(svm5))*10e-6
2562 print('rear_vert_up_svm5 = %.0f' %b)
2563 if b > f_y:
2564     print('This element is under too much stress')
2565
2566 b = (max(svm6))*10e-6
2567 print('rear_vert_up_svm6 = %.0f' %b)
2568 if b > f_y:
2569     print('This element is under too much stress')
2570
2571 b = (max(svm7))*10e-6
2572 print('rear_vert_up_svm7 = %.0f' %b)
2573 if b > f_y:
2574     print('This element is under too much stress')
2575
2576 b = (max(svm8))*10e-6
2577 print('rear_vert_up_svm8 = %.0f' %b)
2578 if b > f_y:
2579     print('This element is under too much stress')
2580
2581
2582
2583 figure()
2584 plot(t,svm1/1.e6,'k-',label='')
2585 xlabel('time [s]')
2586 ylabel('rear_vert_up_strut_fx_2_s1 [MPa]')
```

```
2587 title('Axial Stress')
2588 #axis([-16.,90.,-1.5,1.7])
2589 #legend(loc='best')
2590 savefig('rear_vert_up_strut_fx_2_s1',dpi=300)
2591
2592
2593
2594 figure()
2595 plot(t,svm2/1.e6,'k-',label='')
2596 xlabel('time [s]')
2597 ylabel('rear_vert_up_strut_fx_2_s2 [MPa]')
2598 title('Axial Stress')
2599 #axis([-16.,90.,-1.5,1.7])
2600 #legend(loc='best')
2601 savefig('rear_vert_up_strut_fx_2_s2',dpi=300)
2602
2603
2604
2605 figure()
2606 plot(t,svm3/1.e6,'k-',label='')
2607 xlabel('time [s]')
2608 ylabel('rear_vert_up_strut_fx_2_s3 [MPa]')
2609 title('Axial Stress')
2610 #axis([-16.,90.,-1.5,1.7])
2611 #legend(loc='best')
2612 savefig('rear_vert_up_strut_fx_2_s3',dpi=300)
2613
2614
2615
2616 figure()
2617 plot(t,svm4/1.e6,'k-',label='')
2618 xlabel('time [s]')
2619 ylabel('rear_vert_up_strut_fx_2_s4 [MPa]')
2620 title('Axial Stress')
2621 #axis([-16.,90.,-1.5,1.7])
2622 #legend(loc='best')
2623 savefig('rear_vert_up_strut_fx_2_s4',dpi=300)
2624
2625
2626
2627 figure()
2628 plot(t,svm5/1.e6,'k-',label='')
2629 xlabel('time [s]')
2630 ylabel('rear_vert_up_strut_fx_2_s5 [MPa]')
2631 title('Axial Stress')
2632 #axis([-16.,90.,-1.5,1.7])
2633 #legend(loc='best')
2634 savefig('rear_vert_up_strut_fx_2_s5',dpi=300)
2635
```

```
2636
2637
2638 figure()
2639 plot(t,svm6/1.e6,'k-',label='')
2640 xlabel('time [s]')
2641 ylabel('rear_vert_up_strut_fx_2_s6 [MPa] ')
2642 title('Axial Stress')
2643 #axis([-16.,90.,-1.5,1.7])
2644 #legend(loc='best')
2645 savefig('rear_vert_up_strut_fx_2_s6',dpi=300)
2646
2647
2648
2649 figure()
2650 plot(t,svm7/1.e6,'k-',label='')
2651 xlabel('time [s]')
2652 ylabel('rear_vert_up_strut_fx_2_s7 [MPa] ')
2653 title('Axial Stress')
2654 #axis([-16.,90.,-1.5,1.7])
2655 #legend(loc='best')
2656 savefig('rear_vert_up_strut_fx_2_s7',dpi=300)
2657
2658
2659
2660 figure()
2661 plot(t,svm8/1.e6,'k-',label='')
2662 xlabel('time [s]')
2663 ylabel('rear_vert_up_strut_fx_2_s8 [MPa] ')
2664 title('Axial Stress')
2665 #axis([-16.,90.,-1.5,1.7])
2666 #legend(loc='best')
2667 savefig('rear_vert_up_strut_fx_2_s8',dpi=300)
2668
2669 #
```

```
2670
2671 #Shaft
2672
2673 D_shaft = 1
2674 T_shaft = 0.004
2675
2676 D_shaft = 1
2677 T_shaft = 0.008
2678
2679
2680 shaft_fx = a[:,10]*1.e3;
2681 shaft_fy = a[:,11]*1.e3;
2682 shaft_fz = a[:,12]*1.e3;
```

```
2683 shaft_mx = a[:,13]*1.e3;
2684 shaft_my = a[:,14]*1.e3;
2685 shaft_mz = a[:,15]*1.e3;
2686
2687 [sax , sbendy , sbendz , s1 , s2 , s3 , s4 , s5 , s6 , s7 , s8 , ssh1 , ssh2 , ssh3 , ssh4 , ssh5 , ssh6 ,
      ssh7 , ssh8 , svm1 , svm2 , svm3 , svm4 , svm5 , svm6 , svm7 , svm8 , sx , sy , sz ] = \
2688     cyl_beam_stresses(t , shaft_fx , shaft_fy , shaft_fz , shaft_mx , shaft_my ,
      shaft_mz , D_shaft , T_shaft , 1 , 1 , 1 , 1)
2689
2690 b = (max(svm1))*10e-6
2691 print('shaft_svm1 = %.0f' %b)
2692 if b > f_y:
2693     print('This element is under too much stress')
2694
2695 b = (max(svm2))*10e-6
2696 print('shaft_svm2 = %.0f' %b)
2697 if b > f_y:
2698     print('This element is under too much stress')
2699
2700 b = (max(svm3))*10e-6
2701 print('shaft_svm3 = %.0f' %b)
2702 if b > f_y:
2703     print('This element is under too much stress')
2704
2705 b = (max(svm4))*10e-6
2706 print('shaft_svm4 = %.0f' %b)
2707 if b > f_y:
2708     print('This element is under too much stress')
2709
2710 b = (max(svm5))*10e-6
2711 print('shaft_svm5 = %.0f' %b)
2712 if b > f_y:
2713     print('This element is under too much stress')
2714
2715 b = (max(svm6))*10e-6
2716 print('shaft_svm6 = %.0f' %b)
2717 if b > f_y:
2718     print('This element is under too much stress')
2719
2720 b = (max(svm7))*10e-6
2721 print('shaft_svm7 = %.0f' %b)
2722 if b > f_y:
2723     print('This element is under too much stress')
2724
2725 b = (max(svm8))*10e-6
2726 print('shaft_svm8 = %.0f' %b)
2727 if b > f_y:
2728     print('This element is under too much stress')
2729
```

```
2730
2731
2732 figure()
2733 plot(t,svm1/1.e6,'k-',label='')
2734 xlabel('time [s]')
2735 ylabel('shaft_s1 [MPa] ')
2736 title('Axial Stress')
2737 #axis([-16.,90.,-1.5,1.7])
2738 #legend(loc='best')
2739 savefig('shaft_s1_s1',dpi=300)
2740
2741
2742
2743 figure()
2744 plot(t,svm2/1.e6,'k-',label='')
2745 xlabel('time [s]')
2746 ylabel('shaft_s2 [MPa] ')
2747 title('Axial Stress')
2748 #axis([-16.,90.,-1.5,1.7])
2749 #legend(loc='best')
2750 savefig('shaft_s2',dpi=300)
2751
2752
2753
2754 figure()
2755 plot(t,svm3/1.e6,'k-',label='')
2756 xlabel('time [s]')
2757 ylabel('shaft_s3 [MPa] ')
2758 title('Axial Stress')
2759 #axis([-16.,90.,-1.5,1.7])
2760 #legend(loc='best')
2761 savefig('shaft_s3',dpi=300)
2762
2763
2764
2765 figure()
2766 plot(t,svm4/1.e6,'k-',label='')
2767 xlabel('time [s]')
2768 ylabel('shaft_s4 [MPa] ')
2769 title('Axial Stress')
2770 #axis([-16.,90.,-1.5,1.7])
2771 #legend(loc='best')
2772 savefig('shaft_s4',dpi=300)
2773
2774
2775
2776 figure()
2777 plot(t,svm5/1.e6,'k-',label='')
2778 xlabel('time [s]')
```

```
2779 ylabel('shaft_s5 [MPa] ')
2780 title('Axial Stress ')
2781 #axis([-16.,90.,-1.5,1.7])
2782 #legend(loc='best ')
2783 savefig('shaft_s5',dpi=300)
2784
2785
2786
2787 figure()
2788 plot(t,svm6/1.e6,'k-',label='')
2789 xlabel('time [s] ')
2790 ylabel('shaft_s6 [MPa] ')
2791 title('Axial Stress ')
2792 #axis([-16.,90.,-1.5,1.7])
2793 #legend(loc='best ')
2794 savefig('shaft_s6',dpi=300)
2795
2796
2797
2798 figure()
2799 plot(t,svm7/1.e6,'k-',label='')
2800 xlabel('time [s] ')
2801 ylabel('shaft_s7 [MPa] ')
2802 title('Axial Stress ')
2803 #axis([-16.,90.,-1.5,1.7])
2804 #legend(loc='best ')
2805 savefig('shaft_s7',dpi=300)
2806
2807
2808
2809 figure()
2810 plot(t,svm8/1.e6,'k-',label='')
2811 xlabel('time [s] ')
2812 ylabel('shaft_s8 [MPa] ')
2813 title('Axial Stress ')
2814 #axis([-16.,90.,-1.5,1.7])
2815 #legend(loc='best ')
2816 savefig('shaft_s8',dpi=300)
2817
2818
2819
2820 #

```

```
2821 #blade roots
2822 # only one blade analyzed due to equilibrium for 4 blades
2823 D = 1.2
2824 TWALL = 0.0028
2825
```

```
2826 D = 1.2
2827 TWALL = 0.008
2828
2829 fx_bl_1_root = a[:,16]*1.e3;
2830 fy_bl_1_root = a[:,17]*1.e3;
2831 fz_bl_1_root = a[:,18]*1.e3;
2832 mx_bl_1_root = a[:,19]*1.e3;
2833 my_bl_1_root = a[:,20]*1.e3;
2834 mz_bl_1_root = a[:,21]*1.e3;
2835
2836 [sax, sbendy, sbendz, s1, s2, s3, s4, s5, s6, s7, s8, ssh1, ssh2, ssh3, ssh4, ssh5, ssh6,
    ssh7, ssh8, svm1, svm2, svm3, svm4, svm5, svm6, svm7, svm8, sx, sy, sz] = \
2837     cyl_beam_stresses(t, fx_bl_1_root, fy_bl_1_root, fz_bl_1_root,
    mx_bl_1_root, my_bl_1_root, mz_bl_1_root, D, TWALL, 1, 1, 1, 1)
2838
2839
2840 b = (max(svm1))*10e-6
2841 print('blade_svm1 = %.0f' %b)
2842 if b > f_y:
2843     print('This element is under too much stress')
2844
2845 b = (max(svm2))*10e-6
2846 print('blade_svm2 = %.0f' %b)
2847 if b > f_y:
2848     print('This element is under too much stress')
2849
2850 b = (max(svm3))*10e-6
2851 print('blade_svm3 = %.0f' %b)
2852 if b > f_y:
2853     print('This element is under too much stress')
2854
2855 b = (max(svm4))*10e-6
2856 print('blade_svm4 = %.0f' %b)
2857 if b > f_y:
2858     print('This element is under too much stress')
2859
2860 b = (max(svm5))*10e-6
2861 print('blade_svm5 = %.0f' %b)
2862 if b > f_y:
2863     print('This element is under too much stress')
2864
2865 b = (max(svm6))*10e-6
2866 print('blade_svm6 = %.0f' %b)
2867 if b > f_y:
2868     print('This element is under too much stress')
2869
2870 b = (max(svm7))*10e-6
2871 print('blade_svm7 = %.0f' %b)
2872 if b > f_y:
```

```
2873     print('This element is under too much stress')
2874
2875 b = (max(svm8))*10e-6
2876 print('blade_svm8 = %.0f' %b)
2877 if b > f_y:
2878     print('This element is under too much stress')
2879
2880
2881 figure()
2882 plot(t,svm1/1.e6,'k-',label='')
2883 xlabel('time [s]')
2884 ylabel('blade root_s1 [MPa] ')
2885 title('Axial Stress')
2886 #axis([-16.,90.,-1.5,1.7])
2887 #legend(loc='best')
2888 savefig('blade root_s1',dpi=300)
2889
2890
2891
2892 figure()
2893 plot(t,svm2/1.e6,'k-',label='')
2894 xlabel('time [s]')
2895 ylabel('blade root_s2 [MPa] ')
2896 title('Axial Stress')
2897 #axis([-16.,90.,-1.5,1.7])
2898 #legend(loc='best')
2899 savefig('blade root_s2',dpi=300)
2900
2901
2902
2903 figure()
2904 plot(t,svm3/1.e6,'k-',label='')
2905 xlabel('time [s]')
2906 ylabel('blade root_s3 [MPa] ')
2907 title('Axial Stress')
2908 #axis([-16.,90.,-1.5,1.7])
2909 #legend(loc='best')
2910 savefig('blade root_s3',dpi=300)
2911
2912
2913
2914 figure()
2915 plot(t,svm4/1.e6,'k-',label='')
2916 xlabel('time [s]')
2917 ylabel('blade root_s4 [MPa] ')
2918 title('Axial Stress')
2919 #axis([-16.,90.,-1.5,1.7])
2920 #legend(loc='best')
2921 savefig('blade root_s4',dpi=300)
```



```
2922
2923
2924
2925 figure()
2926 plot(t,svm5/1.e6,'k-',label='')
2927 xlabel('time [s]')
2928 ylabel('blade root_s5 [MPa]')
2929 title('Axial Stress')
2930 #axis([-16.,90.,-1.5,1.7])
2931 #legend(loc='best')
2932 savefig('blade root_s5',dpi=300)
2933
2934
2935
2936 figure()
2937 plot(t,svm6/1.e6,'k-',label='')
2938 xlabel('time [s]')
2939 ylabel('blade root_s6 [MPa]')
2940 title('Axial Stress')
2941 #axis([-16.,90.,-1.5,1.7])
2942 #legend(loc='best')
2943 savefig('blade root_s6',dpi=300)
2944
2945
2946
2947 figure()
2948 plot(t,svm7/1.e6,'k-',label='')
2949 xlabel('time [s]')
2950 ylabel('blade root_s7 [MPa]')
2951 title('Axial Stress')
2952 #axis([-16.,90.,-1.5,1.7])
2953 #legend(loc='best')
2954 savefig('blade root_s7',dpi=300)
2955
2956
2957
2958 figure()
2959 plot(t,svm8/1.e6,'k-',label='')
2960 xlabel('time [s]')
2961 ylabel('blade root_s8 [MPa]')
2962 title('Axial Stress')
2963 #axis([-16.,90.,-1.5,1.7])
2964 #legend(loc='best')
2965 savefig('blade root_s8',dpi=300)
2966
2967 #
```

2968

```
2969
2970 #-----#
2971 D_Rotor = 0.5
2972 Twall_Rotor = 0.004
2973
2974 D_Rotor = 0.5
2975 Twall_Rotor = 0.04
2976
2977 #Rotor
2978 rotor_fx = a[:,172]*1.e3;
2979 rotor_fy = a[:,173]*1.e3;
2980 rotor_fz = a[:,174]*1.e3;
2981 rotor_mx = a[:,175]*1.e3;
2982 rotor_my = a[:,176]*1.e3;
2983 rotor_mz = a[:,177]*1.e3;
2984
2985 [sax , sbendy , sbendz , s1 , s2 , s3 , s4 , s5 , s6 , s7 , s8 , ssh1 , ssh2 , ssh3 , ssh4 , ssh5 , ssh6 ,
    ssh7 , ssh8 , svm1 , svm2 , svm3 , svm4 , svm5 , svm6 , svm7 , svm8 , sx , sy , sz ] = \
2986     cyl_beam_stresses(t , rotor_fx , rotor_fy , rotor_fz , rotor_mx , rotor_my , rotor_mz ,
    rotor_fx , D_Rotor , Twall_Rotor , 1 , 1 , 1 , 1)
2987
2988
2989 b = (max(svm1))*10e-6
2990 print('rotor_svm1 = %.0f' %b)
2991 if b > f_y:
2992     print('This element is under too much stress')
2993
2994 b = (max(svm2))*10e-6
2995 print('rotor_svm2 = %.0f' %b)
2996 if b > f_y:
2997     print('This element is under too much stress')
2998
2999 b = (max(svm3))*10e-6
3000 print('rotor_svm3 = %.0f' %b)
3001 if b > f_y:
3002     print('This element is under too much stress')
3003
3004 b = (max(svm4))*10e-6
3005 print('rotor_svm4 = %.0f' %b)
3006 if b > f_y:
3007     print('This element is under too much stress')
3008
3009 b = (max(svm5))*10e-6
3010 print('rotor_svm5 = %.0f' %b)
3011 if b > f_y:
3012     print('This element is under too much stress')
3013
3014 b = (max(svm6))*10e-6
3015 print('rotor_svm6 = %.0f' %b)
```

```
3016 if b > f_y:
3017     print('This element is under too much stress')
3018
3019 b = (max(svm7))*10e-6
3020 print('rotor_svm7 = %.0f' %b)
3021 if b > f_y:
3022     print('This element is under too much stress')
3023
3024 b = (max(svm8))*10e-6
3025 print('rotor_svm8 = %.0f' %b)
3026 if b > f_y:
3027     print('This element is under too much stress')
3028
3029
3030
3031 figure()
3032 plot(t,svm1/1.e6,'k-',label='')
3033 xlabel('time [s]')
3034 ylabel('rotor_s1 [MPa] ')
3035 title('Axial Stress')
3036 #axis([-16.,90.,-1.5,1.7])
3037 #legend(loc='best')
3038 savefig('rotor_s1',dpi=300)
3039
3040
3041 figure()
3042 plot(t,svm2/1.e6,'k-',label='')
3043 xlabel('time [s]')
3044 ylabel('rotor_s2 [MPa] ')
3045 title('Axial Stress')
3046 #axis([-16.,90.,-1.5,1.7])
3047 #legend(loc='best')
3048 savefig('rotor_s2',dpi=300)
3049
3050
3051
3052 figure()
3053 plot(t,svm3/1.e6,'k-',label='')
3054 xlabel('time [s]')
3055 ylabel('rotor_s3 [MPa] ')
3056 title('Axial Stress')
3057 #axis([-16.,90.,-1.5,1.7])
3058 #legend(loc='best')
3059 savefig('rotor_s3',dpi=300)
3060
3061
3062
3063 figure()
3064 plot(t,svm4/1.e6,'k-',label='')
```

```
3065 xlabel('time [s]')
3066 ylabel('rotor_s4 [MPa] ')
3067 title('Axial Stress')
3068 #axis([-16.,90.,-1.5,1.7])
3069 #legend(loc='best')
3070 savefig('rotor_s4',dpi=300)
3071
3072
3073
3074 figure()
3075 plot(t,svm5/1.e6,'k-',label='')
3076 xlabel('time [s]')
3077 ylabel('rotor_s5 [MPa] ')
3078 title('Axial Stress')
3079 #axis([-16.,90.,-1.5,1.7])
3080 #legend(loc='best')
3081 savefig('rotor_s5',dpi=300)
3082
3083
3084
3085 figure()
3086 plot(t,svm6/1.e6,'k-',label='')
3087 xlabel('time [s]')
3088 ylabel('rotor_s6 [MPa] ')
3089 title('Axial Stress')
3090 #axis([-16.,90.,-1.5,1.7])
3091 #legend(loc='best')
3092 savefig('rotor_s6',dpi=300)
3093
3094
3095
3096 figure()
3097 plot(t,svm7/1.e6,'k-',label='')
3098 xlabel('time [s]')
3099 ylabel('rotor_s7 [MPa] ')
3100 title('Axial Stress')
3101 #axis([-16.,90.,-1.5,1.7])
3102 #legend(loc='best')
3103 savefig('rotor_s7',dpi=300)
3104
3105
3106
3107 figure()
3108 plot(t,svm8/1.e6,'k-',label='')
3109 xlabel('time [s]')
3110 ylabel('rotor_s8 [MPa] ')
3111 title('Axial Stress')
3112 #axis([-16.,90.,-1.5,1.7])
3113 #legend(loc='best')
```

```
3114 savefig('rotor_s8',dpi=300)
```



Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitenskapelige universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway