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 rotorer på én enkelt bærende struktur. Denne avhandlingen undersøker den strukturelle integriteten i en foreslått støttestruktur for én slik rotor i en multirotor vindturbin. Støttestrukturen vil bli analysert for både to og fire blader. Den strukturelle analysen inkluderer utmatting, 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. Knekkeberegninger 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

List of Figures

List of Tables

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 chrod 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_{logN} 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

v_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 worlds 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.

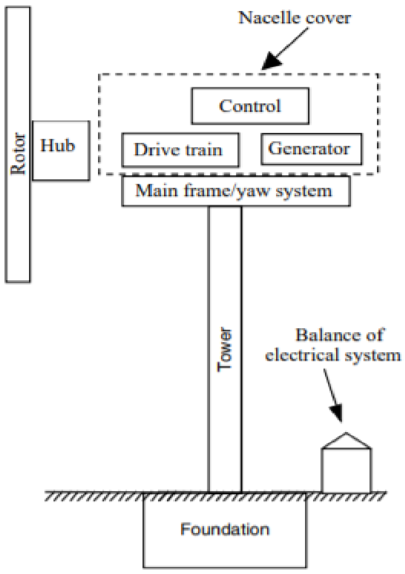

Figure 1.1 Major components of a conventional wind turbine

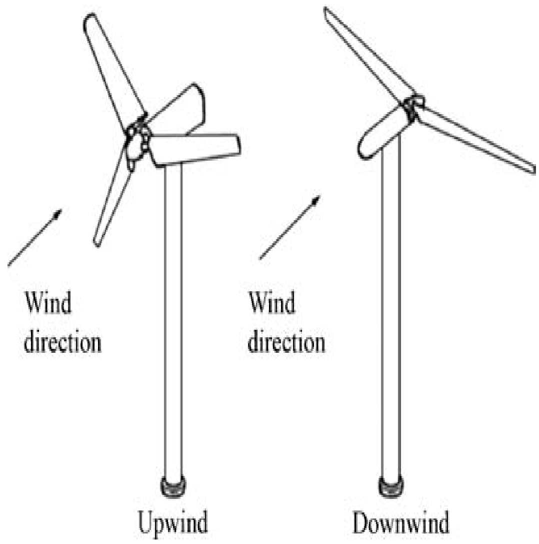

Figure 1.2 The different orientations of a HAWT

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, Pt, from the wind can be derived.

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

Where ρ and v are the wind's density and speed respectively, and R is the radius of the turbine. Cp 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 Cp 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.

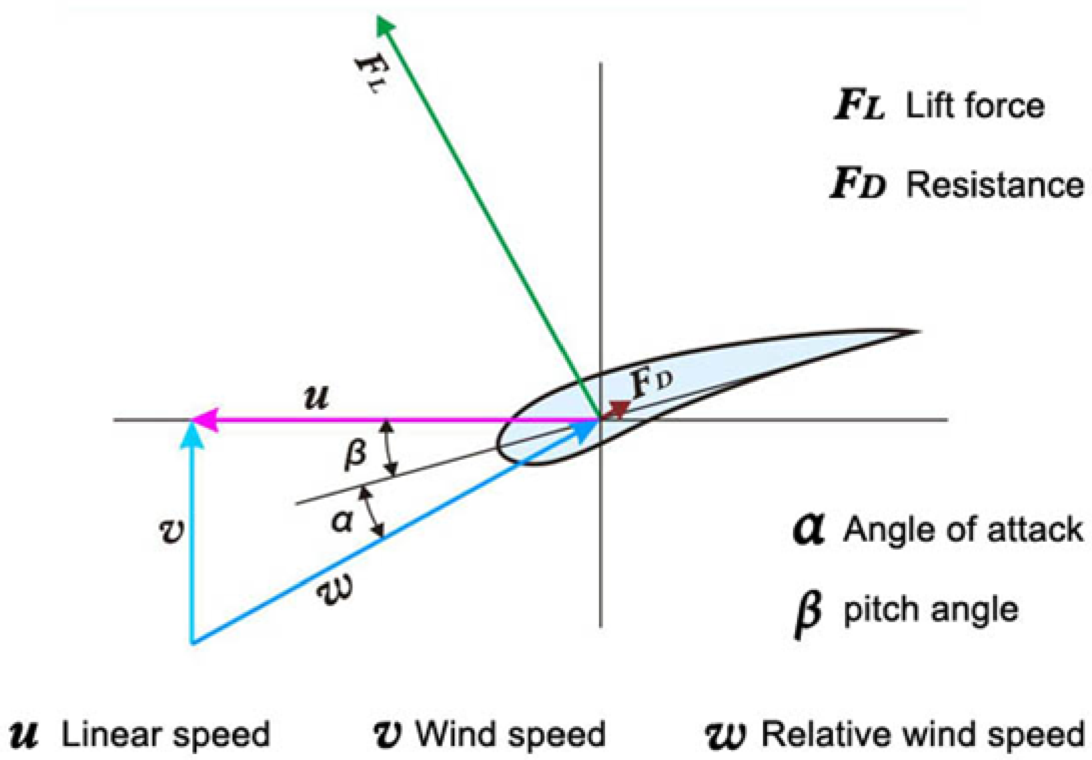

Figure 1.3 Parametric definitions related to a turbine blade in motion

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:

$$C_T = \frac{Thrust\,force}{Dynamic\,force} = \frac{T}{\frac{1}{2}\rho v^2 A} \tag{1.2}$$

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:

$$\lambda_{tsr} = \frac{\omega R}{v} = \frac{tip\ speed\ of\ blade}{free\ stream\ wind\ speed} \tag{1.3}$$

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 Cp is at its highest.

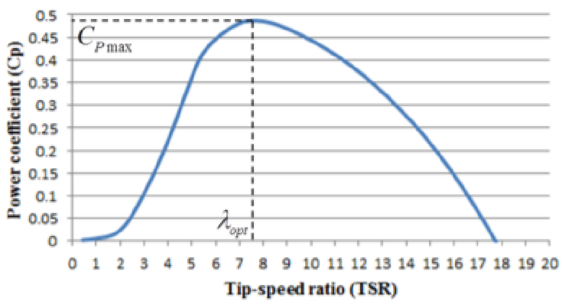

Figure 1.4 $C_P - TSR$ curve for NREL offshore 5 MW baseline wind turbine at pitch angle (β) of 0

[5]

1.1.4 Power Curve

The power coefficient, Cp, 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:

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

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

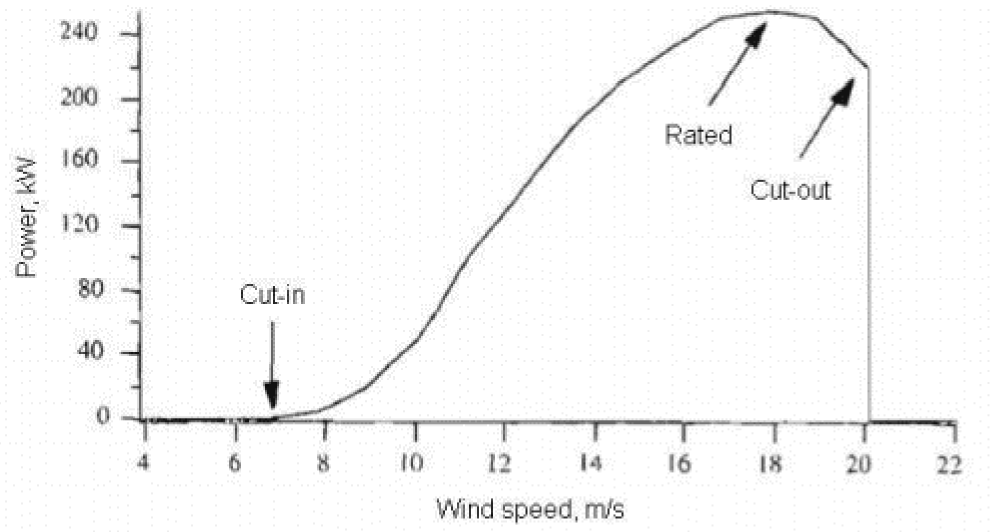

Figure 1.5 Key points of a wind turbine's power curve

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 fatige 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:

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

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.

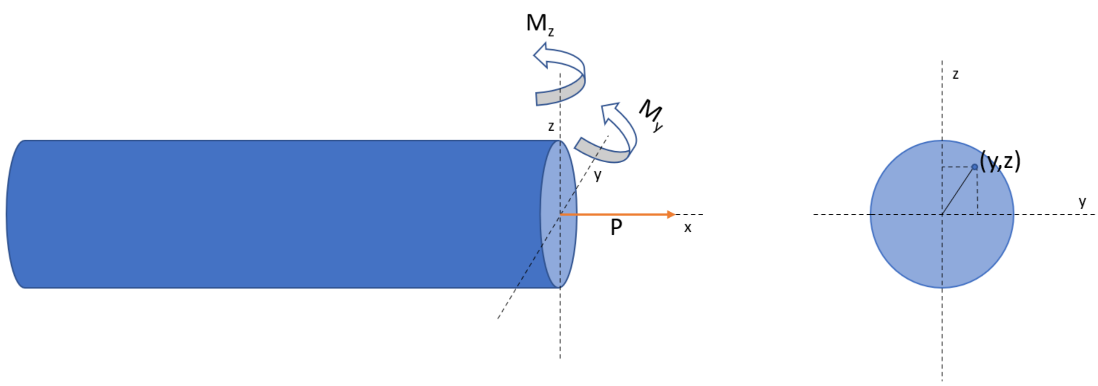

Figure 2.1 Cylinder subject to combined loading

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:

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

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

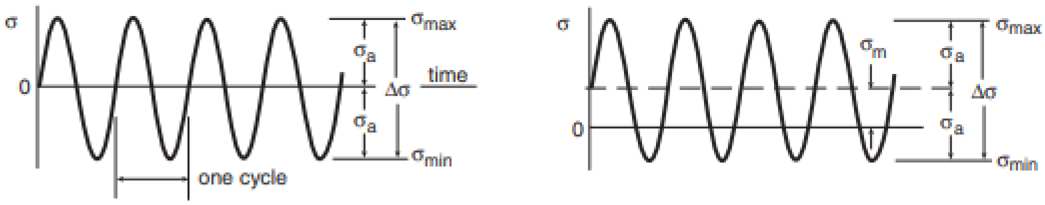

Figure 2.2 Constant amplitude cycling and the associated nomenclature.

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.

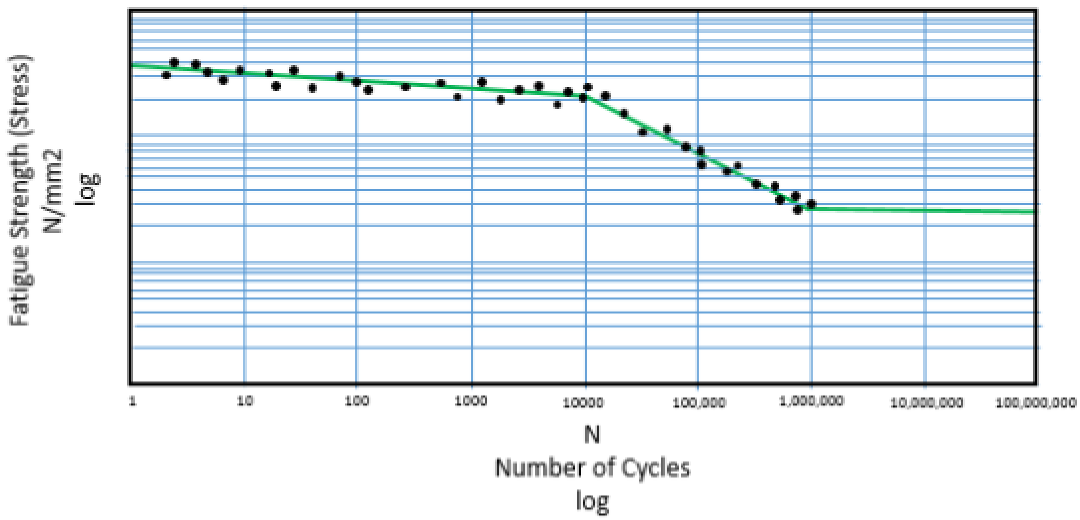

Figure 2.3 Best-fit curves for the S-N data points [10]

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

$$N = C\Delta\sigma^{-m} \tag{2.4}$$

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:

$$\log N = \log C - m \log \Delta\sigma \tag{2.5}$$

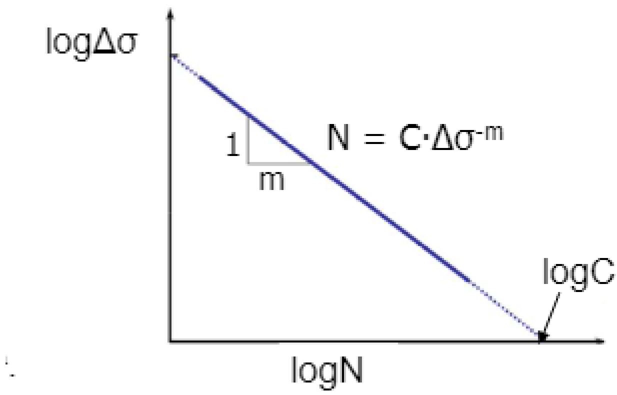

Figure 2.4 Illustration of an S-N curve with the associated nomenclature [11]

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 N1 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 $N1/N_{f1}$ 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:

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

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 $\Delta\sigma_i$.

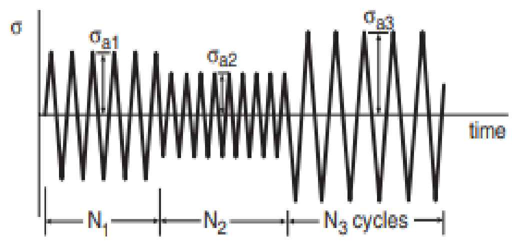

Figure 2.5 Variable amplitude loading

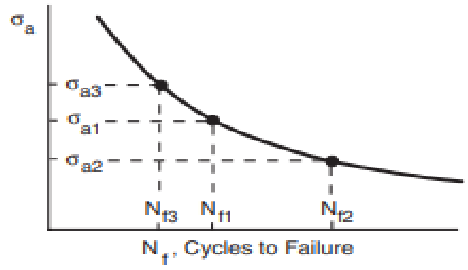

Figure 2.6 Relating different stress ranges to number of cycles to failure on the S-N curve

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.

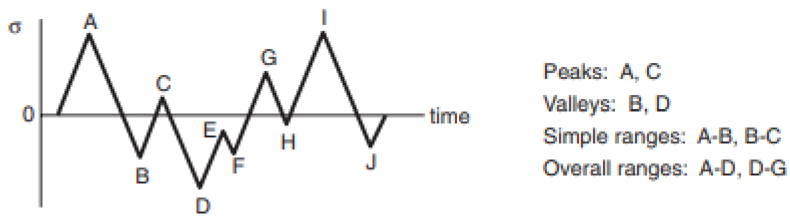

Figure 2.7 Example of an irregular loading situation with relevant definitions for the rainflow counting method.

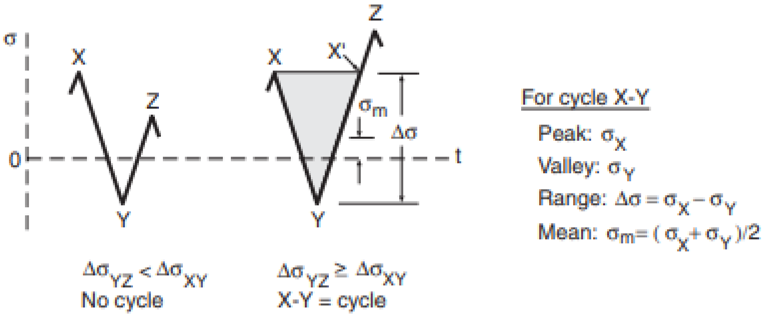

Figure 2.8 Definition of a cycle in Rainflow Cycle Counting with related definitions

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}:

$$\sigma_{XY} \leq \sigma_{YZ} \tag{2.7}$$

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.

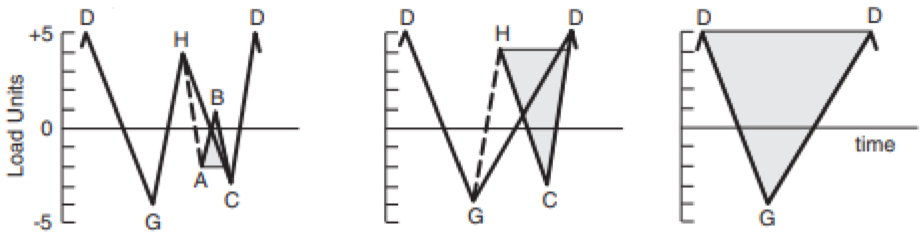

Figure 2.9 Procedure of Rainflow Cycle Counting

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:

$$P_{cr} = \frac{\pi^2 E I_{min}}{L_e^2} \tag{2.8}$$

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.

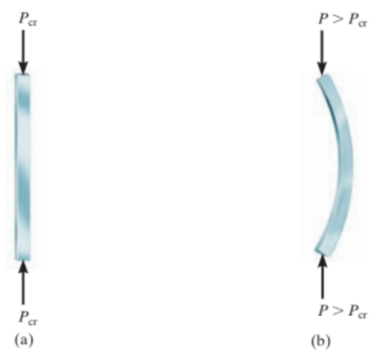

Figure 2.10 Side by side comparison between a column before and after buckling

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:

$$L_e = KL \tag{2.9}$$

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:

$$\lambda = \frac{L_e}{i} \tag{2.10}$$

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

$$i = \sqrt{\frac{I}{A}} \tag{2.11}$$

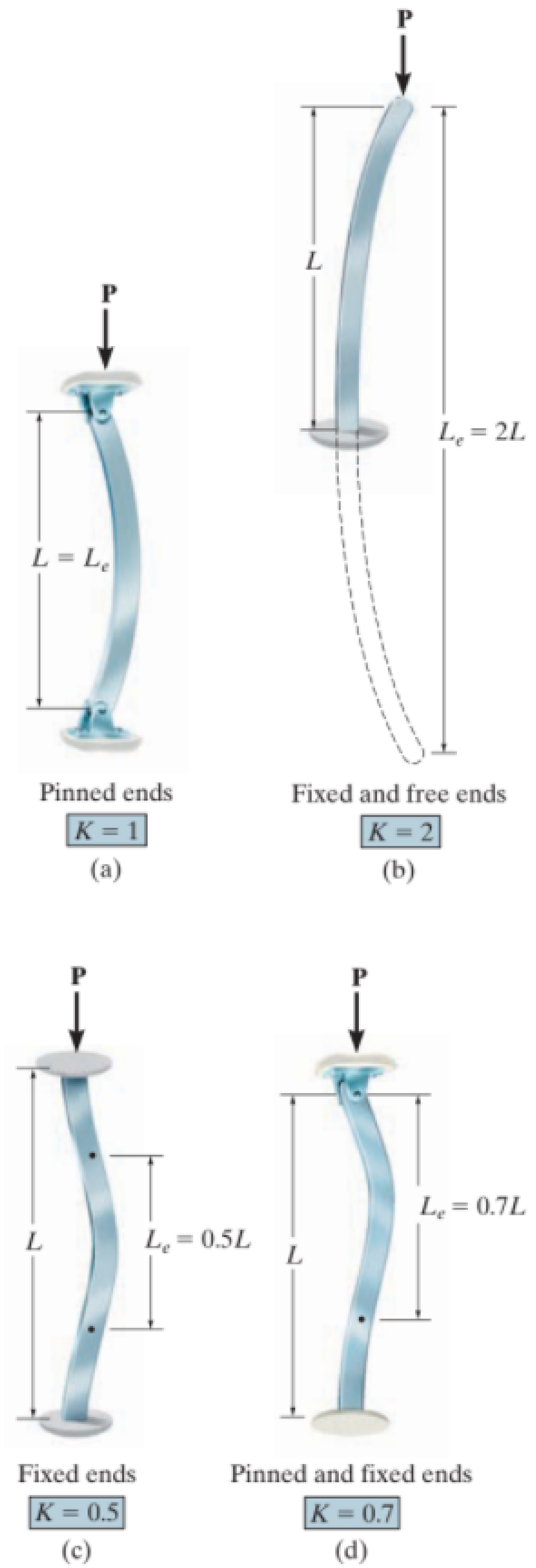

Figure 2.11 Column supports and associated buckling lengths

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 distorion energy, u_d is equal to, or higher, than the distorion energy, $u_{y,d}$, at yielding in a uniaxial tensile test:

$$u_d \geq u_{y,d} \tag{2.12}$$

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:

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

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

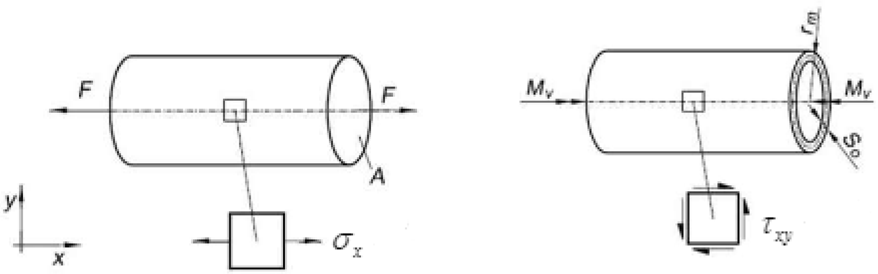

Figure 2.12 Plane stress for a cylinder [12]

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:

$$\sigma_{eq} \leq f_y \tag{2.14}$$

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.

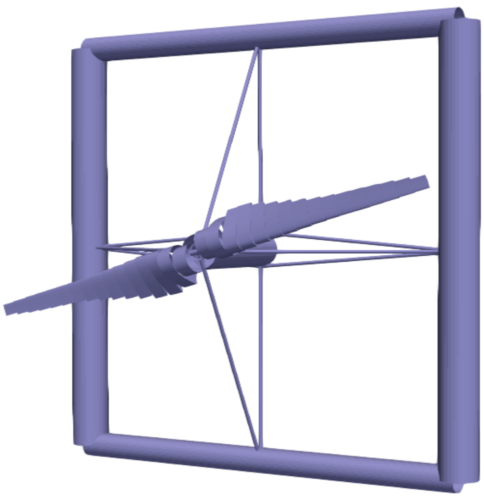

Figure 3.1 A single MRWT module featuring two blades

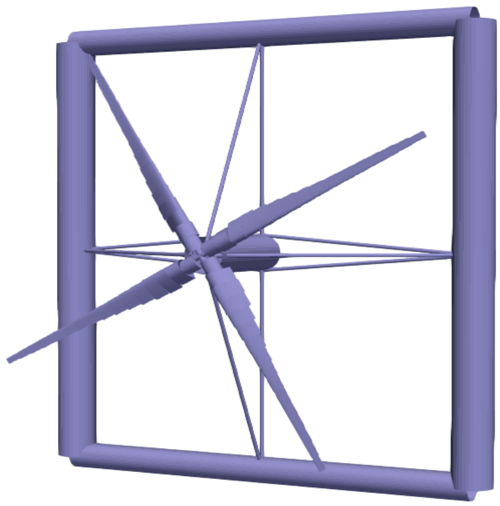

Figure 3.2 A single MRWT module featuring four blades

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 struts[1] 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.

[1]A 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

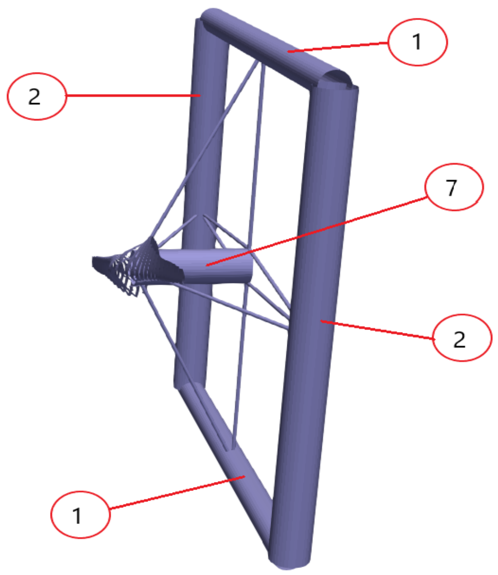

Figure 3.3 Elements in the table labelled according to table 3.1

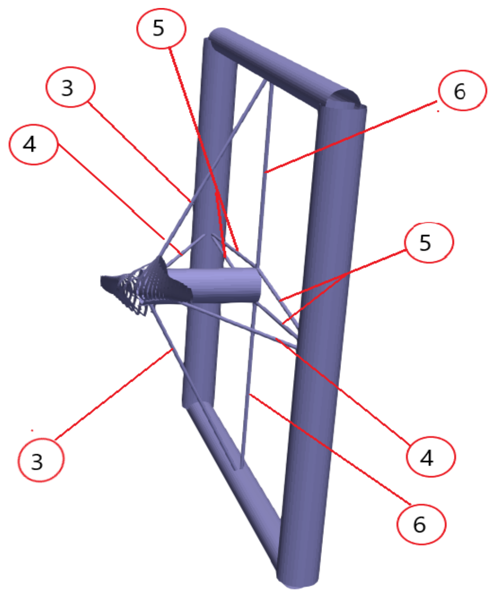

Figure 3.4 Elements in the table labelled according to table 3.1

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, v_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 length[2].

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.

[2] The width of the blade

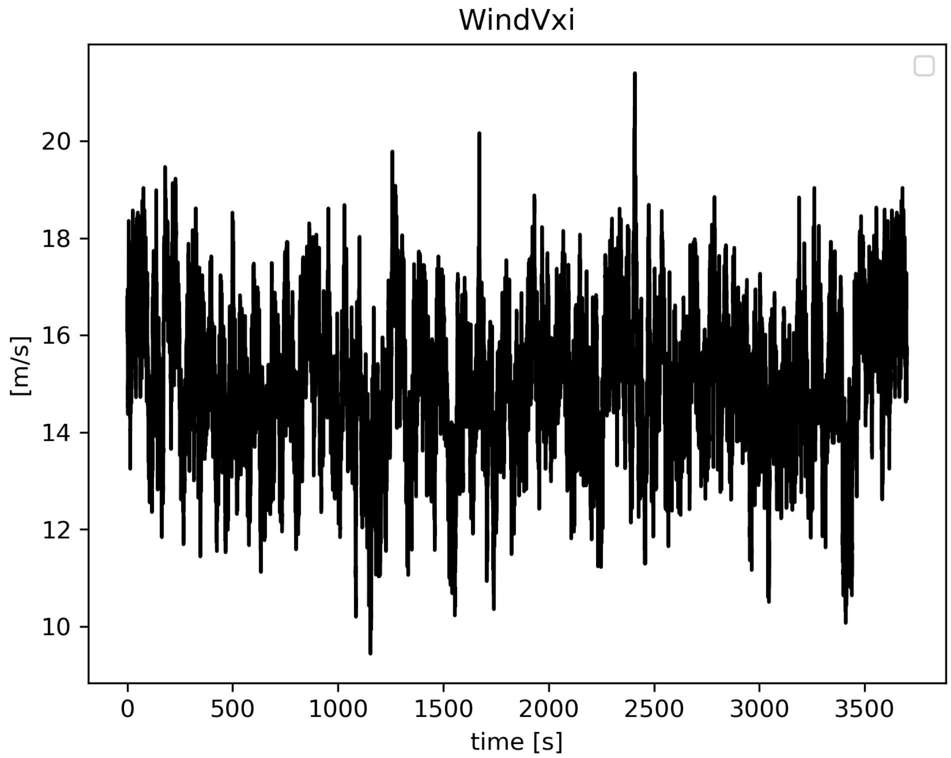

Figure 3.5 Wind speed as a function of time normal to the rotor at normal wind conditions

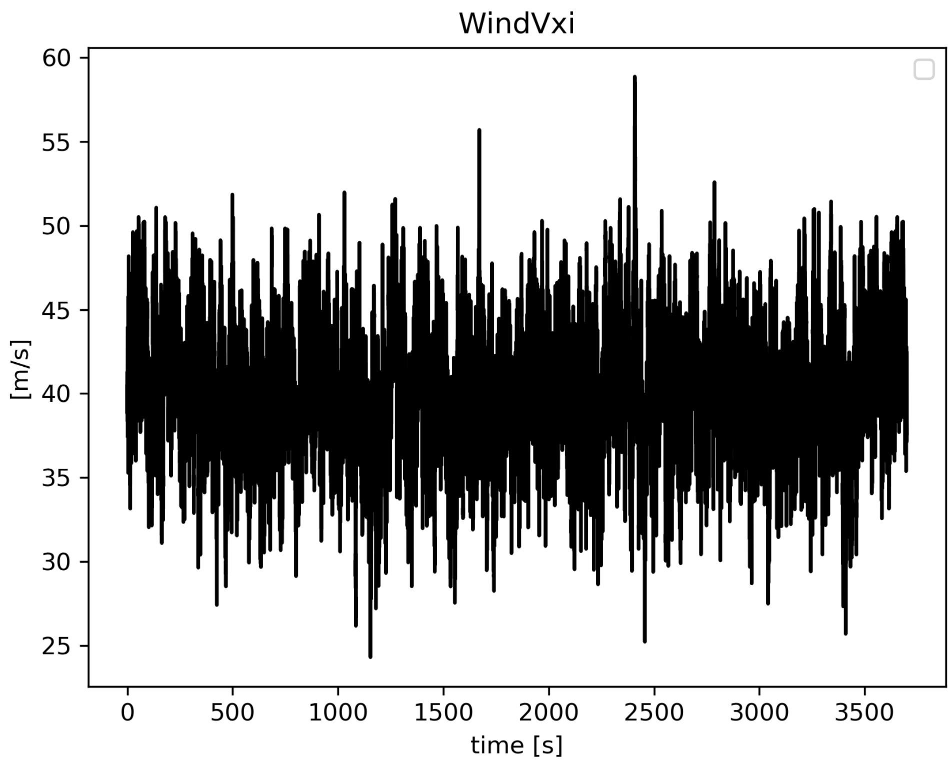

Figure 3.6 Wind speed as a function of time normal to the rotor at extreme wind conditions

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:

$$\log \bar{a} = \log C - 2s_{\log N} \tag{3.1}$$

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:

$$\log N = \log \bar{a} - m \log \Delta\sigma \tag{3.2}$$

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:

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

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 $\leq 10^7$ 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:

$$\sigma_{hot\ spot} = SCF \sigma_{nominal} \tag{3.4}$$

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.

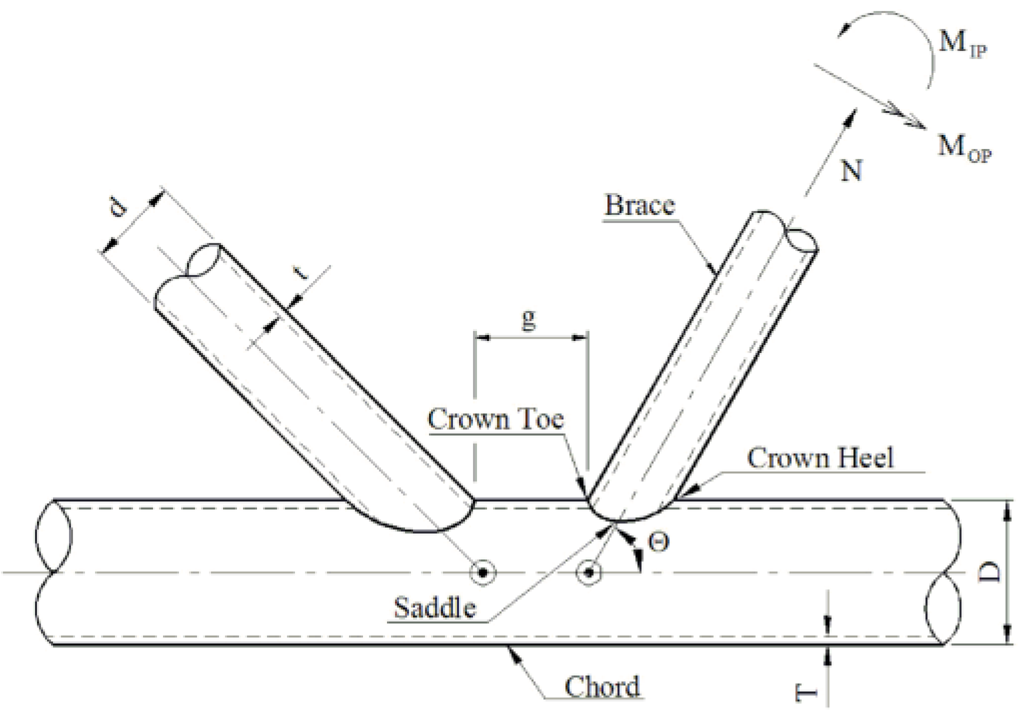

Figure 3.7 Geometrical definitions for tubular joints

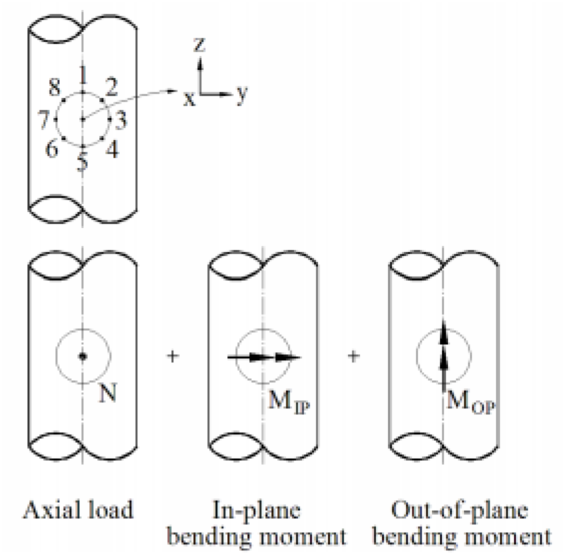

Figure 3.8 Superposition of stresses around the circumference due to combined loading

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:

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

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

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

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

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

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

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

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

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:

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

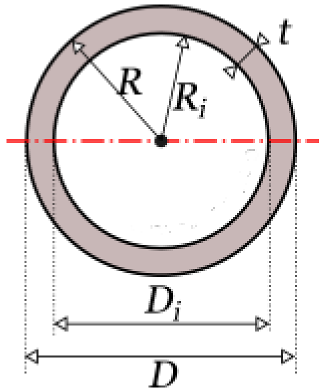

Figure 3.9 Geometrical definitions of a hollow cylinder's cross-section

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.

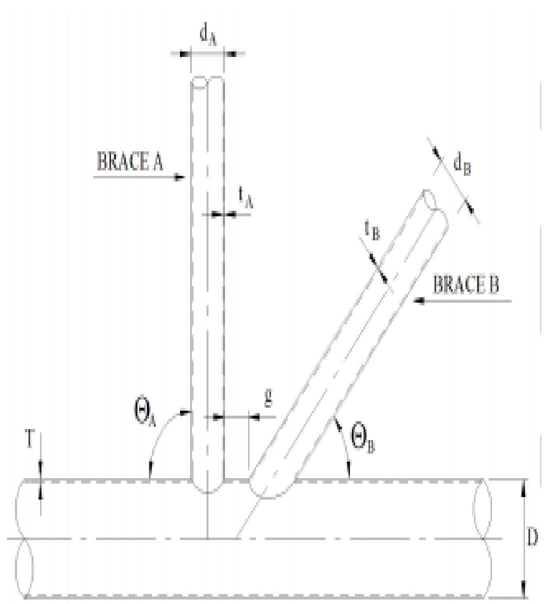

Figure 3.10 Definitions of geometrical parameters for K-joints

$$\beta = \frac{d}{D} \tag{3.7a}$$
$$\alpha = \frac{2L}{D} \tag{3.7b}$$
$$\gamma = \frac{D}{2T} \tag{3.7c}$$
$$\tau = \frac{t}{T} \tag{3.7d}$$
$$\zeta = \frac{g}{D} \tag{3.7e}$$

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.

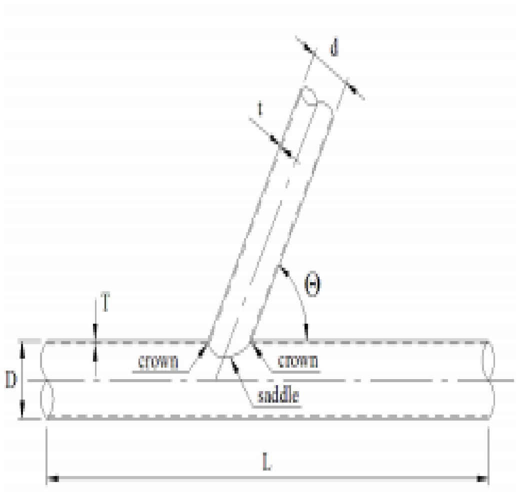

Figure 3.11 Definitions of geometrical parameters for T-joint

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}	$1+(1.97-1.57\beta^{0.25})\tau^{-0.14}(sin\theta)^{0.7}\tau^{0.9}\gamma^{0.5}(0.67\beta^2+1.16\beta)$ $\sin\theta(\frac{sin\theta_{max}}{sin\theta_{min}})^{0.30}(\frac{\beta_{max}}{\beta_{min}})^{0.30}(1.64+0.29\beta^{-0.38}arctan(8\zeta))$
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$(\beta_B\gamma)^{0.5}\exp(-0.8x))+$ $\tau_B^{-0.54}\gamma^{-0.05}(0.99-0.47\beta_B+0.08\beta_B^4)(\gamma\tau_A\beta_B(1.7-1.05\beta_B^3)(sin(\theta_B))^{1.6})$ (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

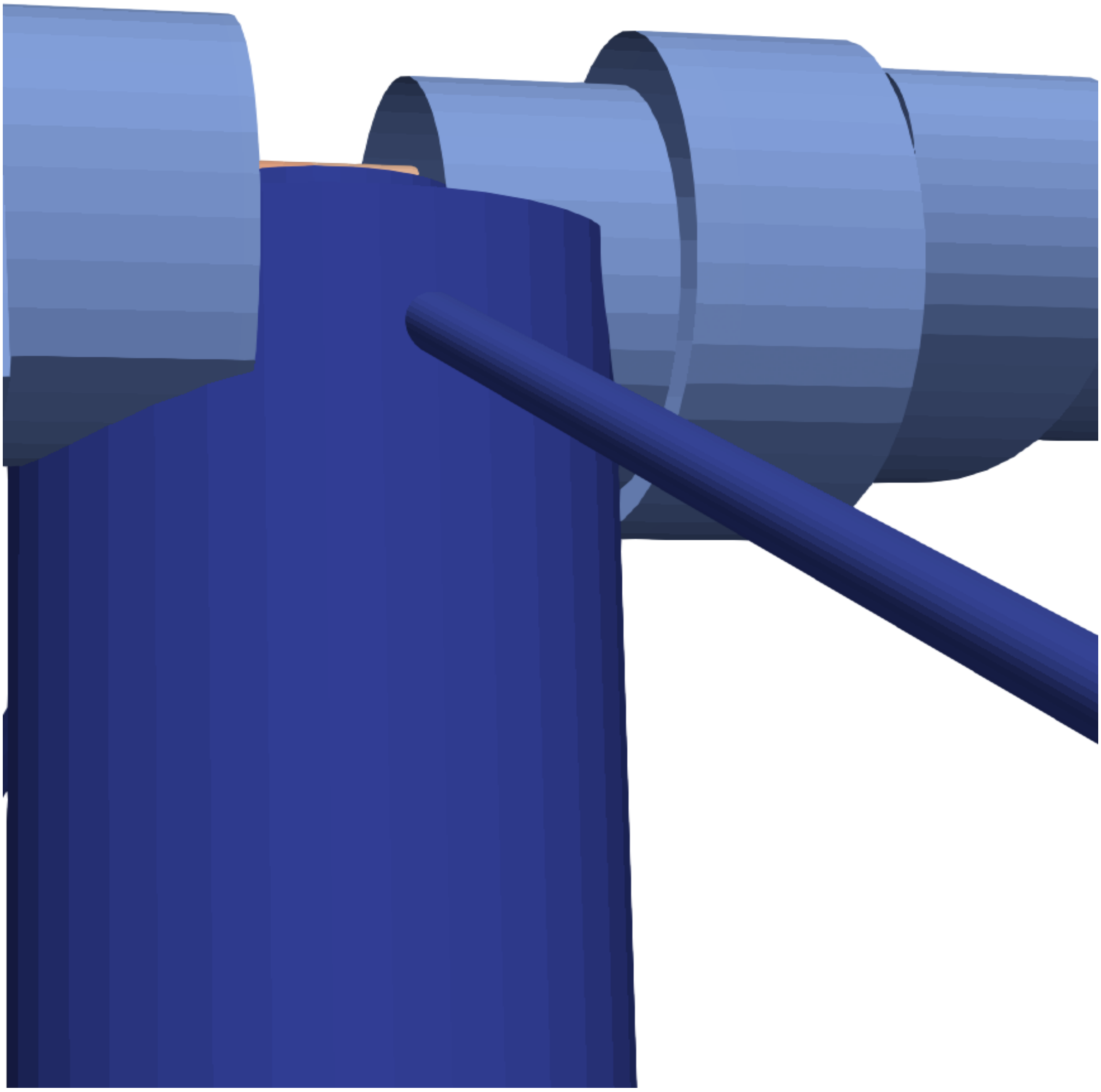

Figure 3.12 Example of a Type 1 joint

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)

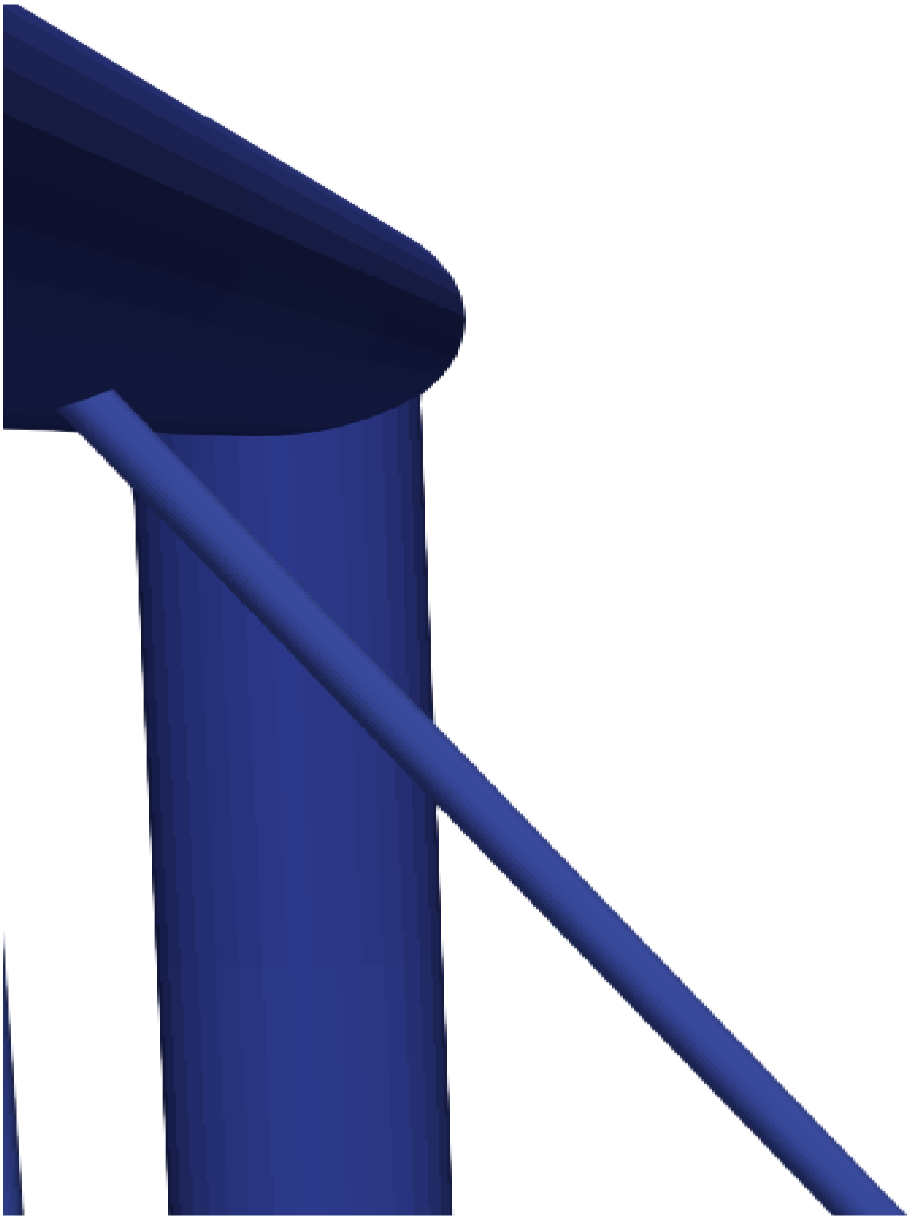

Figure 3.13 Example of a Type 2 joint

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)

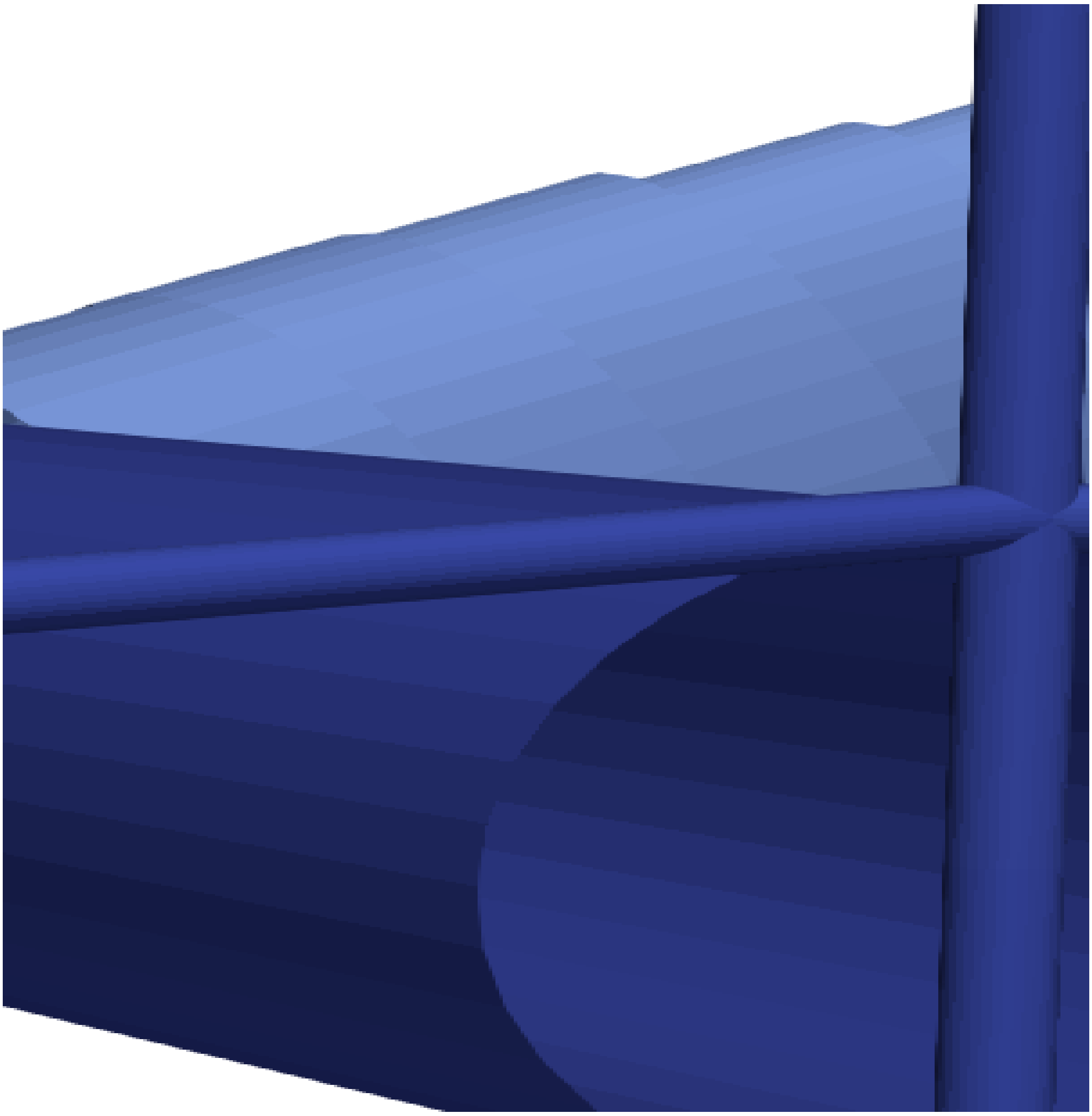

Figure 3.14 Example of a Type 3 joint

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)

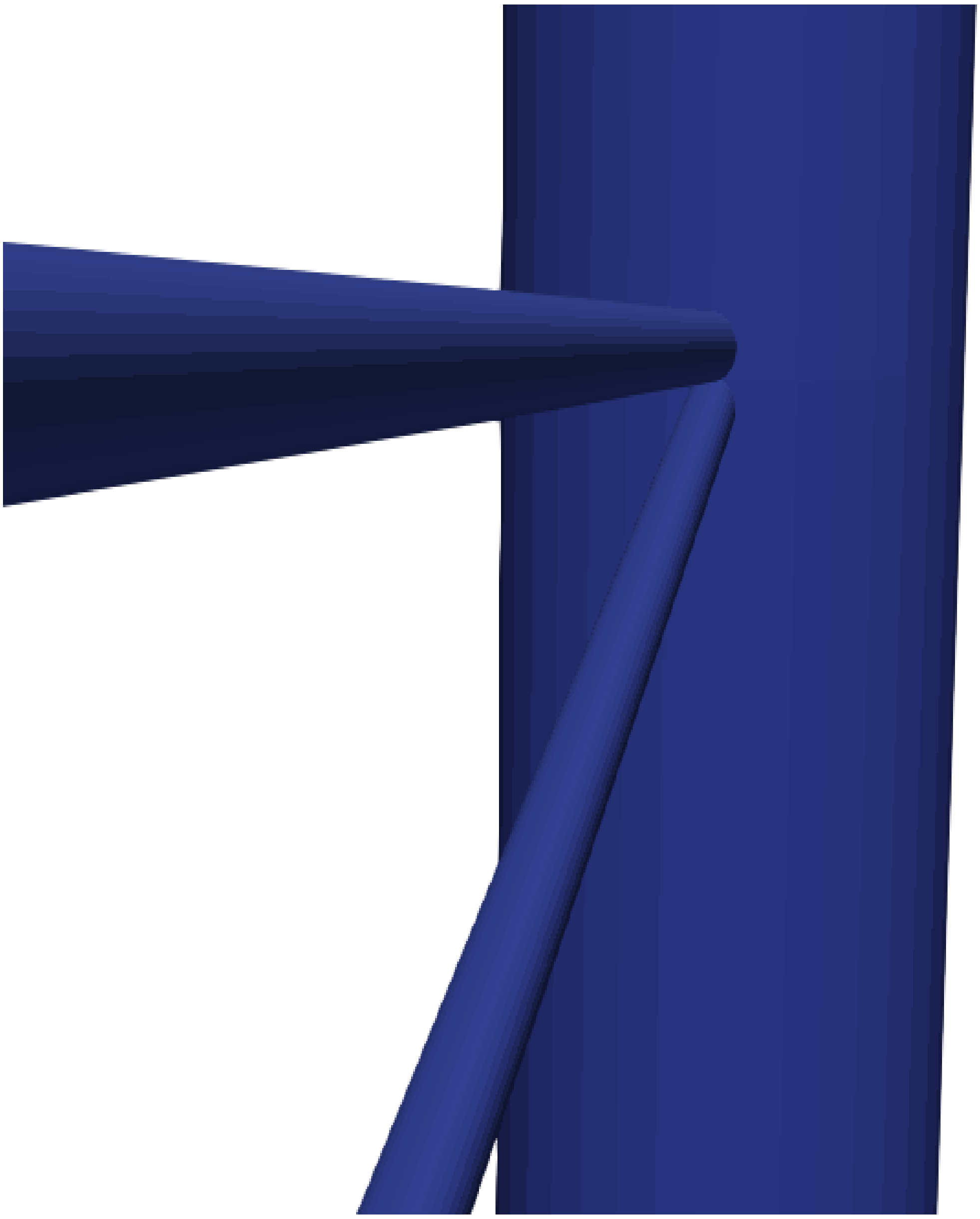

Figure 3.15 Example of a Type 4 joint

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 steelstructures - 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

<table>
<tr><td>Class</td><td colspan="6">Section in bending and/or compression</td></tr>
<tr><td>1</td><td colspan="6">$d/t \leq 50\epsilon^2$</td></tr>
<tr><td>2</td><td colspan="6">$d/t \leq 70\epsilon^2$</td></tr>
<tr><td>3</td><td colspan="6">$d/t \leq 90\epsilon^2$</td></tr>
<tr><td rowspan="3">$\epsilon = \sqrt{235/f_y}$</td><td>$f_y$</td><td>235</td><td>275</td><td>355</td><td>420</td><td>460</td></tr>
<tr><td>ϵ</td><td>1.00</td><td>0.92</td><td>0.81</td><td>0.75</td><td>0.71</td></tr>
<tr><td>ϵ^2</td><td>1.00</td><td>0.85</td><td>0.66</td><td>0.56</td><td>0.51</td></tr>
</table>

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

<table>
<tr><td rowspan="2">Cross section</td><td rowspan="2">Finish</td><td rowspan="2">Buckling about axis</td><td colspan="2">Buckling curve</td></tr>
<tr><td>S 235
S 275
S 355
S 420</td><td>S 460</td></tr>
<tr><td rowspan="2">CHS</td><td>Hot finished</td><td>any</td><td>a</td><td>a_0</td></tr>
<tr><td>Cold formed</td><td>any</td><td>c</td><td>c</td></tr>
</table>

Table 3.6 Assessment of buckling curve for CHS members

when the appropriate buckling curve is assesed, 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 inequalites for which the member should be verified against:

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

and

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

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
- $M_{by,Rd}$, $M_{by,Rd}$ are the design buckling resistance moment about the y-y and z-z axis respectivley
- $\Delta M_{y,Ed}$, $\Delta M_{z,Ed}$ represents the moments caused by shift in the postition 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:

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

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

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

and

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

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:

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

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:

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

where

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

and

$$\bar{\lambda} = \sqrt{\frac{Af_y}{N_{cr}}} \tag{3.16}$$

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:

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

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:

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

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 acoording 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 strucure
- 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.

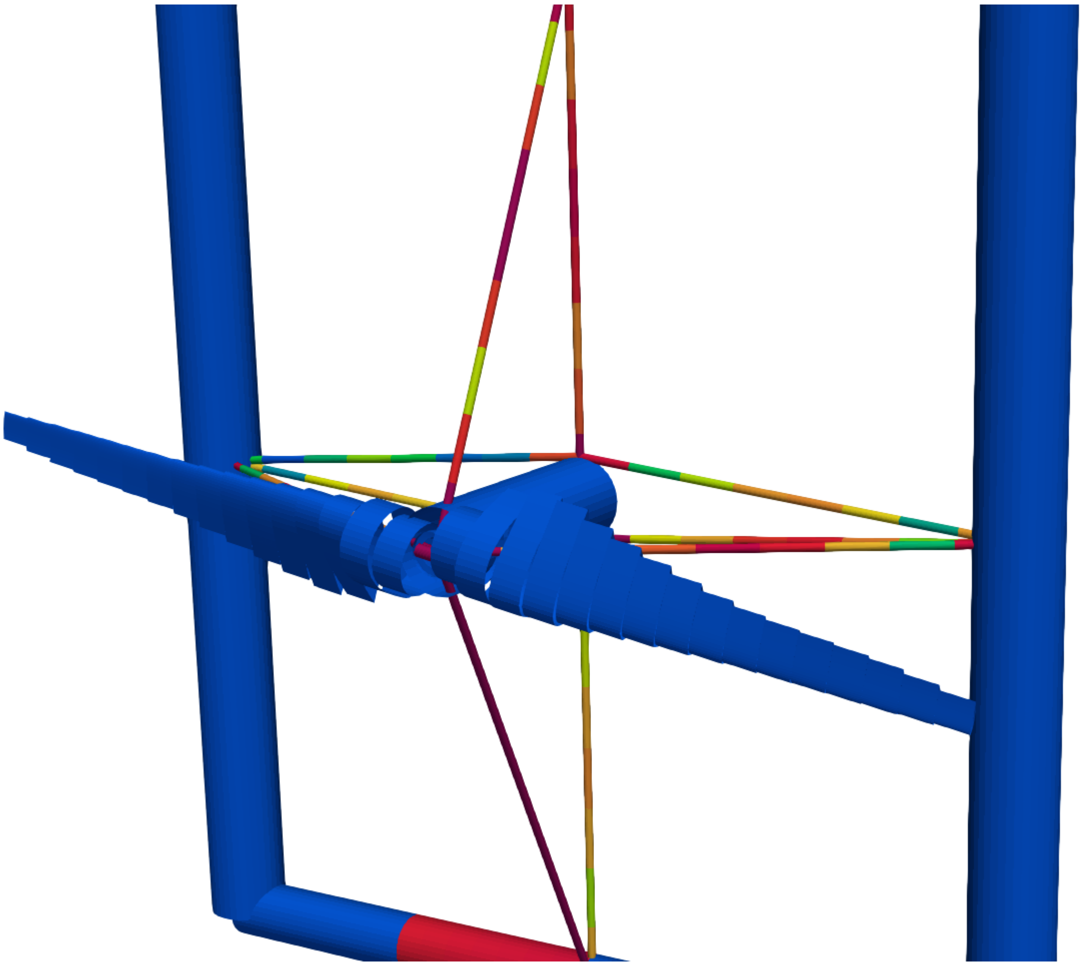

Figure 4.1 Stress distribution for extreme wind conditions from ParaView

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 eachother

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

```
import numpy as np
import pylab as plt
import scipy

##################################################################
## Functions to calculate partial fatigue damage for welded     ##
## steel structures from timeseries of stress given in Pa       ##
## and two-sloped SN-curves as defined in DNV-OS-C203 Fatigue##
## Design of Offshore Steel Structures.                         ##
##                                                              ##
## Stress concentration factors for hot-spot stress must be ##
## included in the stress timeseries before these functions    ##
## are used.                                                    ##
##                                                              ##
## Stress cycles are counted using Rainflow counting.          ##
##                                                              ##
## See example_fatigue_funcs.py for example of how to use      ##
##                                                              ##
## Marit Kvittem Feb 2015                                       ##
##################################################################

def turningpoints(x):

    ## Find the amplitude at turning points of a 1D numpy array x

    dx = np.diff(x)
    Np = np.sum( dx[1:] * dx[:-1] < 0)
    ind = np.where(dx[1:] * dx[:-1] < 0)
    tp_m = x[ind]

    ## add end points
    tp = [x[0]]
    tp.extend(tp_m)
    tp.extend([x[-1]])

    return tp

```

```python
def turningpoints_steffen(x,amp): #Written by Steffen Aasen, April 2016
    #save indexes of turning points
    turningpoints=[]
    indexes=[]
    for i in range(1,len(x)-1):
        if x[i-1]>x[i] and x[i+1]>x[i] or x[i-1]<x[i] and x[i+1]<x[i]:
            indexes.append(i)
    #make array with turningpoints
    for element in indexes:
        if abs(x[element-1]-x[element])>amp and abs(x[element+1]-x[element])>amp:
            turningpoints.append(x[element])

    #delete points that are not turningpoints (due to numerical error)
    indexes=[]
    for i in range(len(turningpoints)-2):
        if turningpoints[i+1]>turningpoints[i] and turningpoints[i+2]>turningpoints[i+1] or turningpoints[i+1]<turningpoints[i] and turningpoints[i+2]<turningpoints[i+1]:
            indexes.append(i+1)
    turningpoints_2=[]
    for i in range(len(turningpoints)):
        if i not in indexes:
            turningpoints_2.append(turningpoints[i])

    return turningpoints_2

def findrfc_wafo(x):

    ## Rainflow counting of 1D list of turning points x
 ## based on matlab wafo's tp2arfc4p and default values given in tp2rfc

 def_time=0.
 res0 = []
 T = len(x)
 ARFC = np.zeros((int(np.floor(T/2)),2))
 N = -1

 res = np.zeros(max([200,len(res0)]))

 nres = -1

 for i in range(0,T):
  nres = nres+1
  res[nres] = x[i]
  cycleFound = 1
  while cycleFound ==1 and nres >=4:
   if res[nres-1] < res[nres-2]:
    A = [res[nres-1], res[nres-2]]
```

```
        else:
            A = [res[nres-2], res[nres-1]]

        if res[nres] < res[nres-3]:
            B = [res[nres], res[nres-3]]
        else:
            B = [res[nres-3], res[nres]]

        if A[0] >= B[0] and A[1] <= B[1]:
            N=N+1
            arfc = [[res[nres-2]],[res[nres-1]]]
            ARFC[N] = [res[nres-2],res[nres-1]]
            res[nres-2] = res[nres]
            nres = nres-2
        else:
            cycleFound = 0
## residual
res = res[0:nres+1]

def res2arfc(res):
    nres = len(res)
    ARFC = []
    if nres < 2:
        return
    ## count min to max cycles, gives correct number of upcrossings
    if (res[1]-res[0]) > 0.:
        i_start = 0
    else:
        i_start=1
    I = range(i_start,nres-1,2)
    Ip1 = range(i_start+1,nres,2)
    ## def_time = 0

    for ii in range(len(I)):
        ARFC.append( [res[I[ii]], res[Ip1[ii]]] )

    ARFC = np.array(ARFC)

    return ARFC

ARFC_res = res2arfc(res)

ARFC = np.concatenate((ARFC,ARFC_res))

## make symmetric
[N,M] = np.shape(ARFC)
I = []
J=0
RFC = ARFC
```

```
    for ii in range(N):
        if ARFC[ii,0] > ARFC[ii,1]:
            ## Swap variables
            RFC[ii,J],RFC[ii,J+1] = RFC[ii,J+1],RFC[ii,J]

    cc = RFC

    rfcamp = (RFC[:,1] - RFC[:,0])/2.

    return rfcamp



def fatiguedamage_twoslope(time,stress,m1,loga1,m2,loga2,Nlim,th=25E-3,tref=25E-3,k=0.25):
        ## stress: stressvector, unit: Pa
        ## m1, loga1, m2, loga2: Parameters from table 2.2 in RP-C203
        ## Note that the parameters in RP C203 are given for stress ranges in MPa
        ## tref, k: perameters from point 2.4 in RP-C203
        ## th: structural detail thickness
        ## Calculates fatigue damage for bilinear SN curves
        ## hist: true/false parameter, wether or not to plot histogram

    stress = stress*1.E-6


    tp = turningpoints_steffen(stress,0.0) ## Find turning points
    mm = findrfc_wafo(tp) ## Rainflow cycles as by the routine in matlab wafo

    Nbins = len(mm)

    if th<tref:
        th = tref

    a1 = 10.**loga1
    K1 = 2.0**m1/a1*(th/tref)**(k*m1)
    beta1 = m1

    a2 = 10.**loga2
    K2 = 2.0**m2/a2*(th/tref)**(k*m2)
    beta2 = m2



    alim = (1.0/(K1*Nlim))**(1./m1)
    alim2 =  (1.0/(K2*Nlim))**(1./m2)

```

```
avalid = (1.0/(K2*1.0E7))**(1./m2)

if not np.round(alim,0) == np.round(alim2,0):
    print(loga1)
    print(loga2)
    print(m1)
    print(m2)
    print(K1)
    print(K2)
    print(alim)
    print(alim2)
    print('alim not the same as alim2, check SN curve values')

dd = 0.0

amp = abs(mm)

for aa in amp:
    if aa > alim:
        dd = dd + K1*aa**beta1

    elif aa <= alim:
        if aa < avalid:
            key = True

            dd = dd + K2*aa**beta2
        else:
            dd = dd + K2*aa**beta2

D_T = dd

return D_T
```

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

```
# -*- coding: utf-8 -*-
"""
Created on Fri Apr 16 13:10:24 2021

@author: Vetle Birkeland Aass
"""

from pylab import *
from numpy import loadtxt



def cyl_beam_stresses(t,fx,fy,fz,mx,my,mz,d,twall, SCF_AS, SCF_AC, SCF_MIP
    , SCF_MOP):
    #
    # Stresses for box beam
    # Signs are for monitors set for node 2. Forces can be seen as acting
        on node 2
    # from the neighbor element
    # Calculation ponts in element coordinate system:
    # Point 1 on y axis, then positive rotation about x axis
    #
    #  Point     y         z
    #  1         0.5*d     0
    #  2         0.5*d*cos45   0.5d*sin45
    #  3         0         b/2
    #  and so on
    #  iy means bending stiffness ABOUT y axis,  integral of z^2dA

    di = d - 2.*twall
    a = .25*pi*(d**2-di**2)
    icyl = pi*(d**4-di**4)/64.

    # normal stresses for points 1-8
    sax     =  fx/a  # axial stress, positive tension
```

```
    sbendy =  my*.5*d/icyl  # positive moment gives tension (positive for pos z)
    sbendz = -mz*.5*d/icyl    # positive moment gives tension (positive for neg y)

    sin45 = sin(pi*45./180.)  # z coordinate
    cos45 = cos(pi*45./180.)  # y coordinate


    s1    = SCF_AC*sax                                              + SCF_MOP*sbendz
    s2    =  0.5*(SCF_AC+SCF_AS)*sax     + SCF_MIP*sbendy*sin45     + SCF_MOP*sbendz*cos45
    s3    = SCF_AS*sax  + sbendy
    s4    =  0.5*(SCF_AC+SCF_AS)*sax     + SCF_MIP*sbendy*sin45     - SCF_MOP*sbendz*cos45
    s5    = SCF_AS*sax                                              - SCF_MOP*sbendz
    s6    =  0.5*(SCF_AC+SCF_AS)*sax     - SCF_MIP*sbendy*sin45     - SCF_MOP*sbendz*cos45
    s7    = SCF_AS*sax                   - SCF_MIP*sbendy
    s8    =  0.5*(SCF_AC+SCF_AS)*sax     - SCF_MIP*sbendy*sin45     + SCF_MOP*sbendz*cos45

    # shear stress
    symax = fy/a*(4./3)*(d**2 + d*di + di**2)/(d**2 + di**2) # on z axis
    szmax = fz/a*(4./3)*(d**2 + d*di + di**2)/(d**2 + di**2) # on y axis
    # torsion stress, thin wall approx
    dm = .5*(d+di)
    sx = mx/(2.*twall*.25*pi*dm**2)

    # shear stresses, positive along section, positive x rotation
    ssh1 = sx + szmax
    ssh2 = sx            # ignore shear stress due to shear forces here
    ssh3 = sx - symax
    ssh4 = sx            # ignore shear stress due to shear forces here
    ssh5 = sx - szmax
    ssh6 = sx
    ssh7 = sx + symax
    ssh8 = sx

    # von Mises stress
    svm1 = sqrt(s1**2 + 3.*ssh1**2)
    svm2 = sqrt(s2**2 + 3.*ssh2**2)
    svm3 = sqrt(s3**2 + 3.*ssh3**2)
    svm4 = sqrt(s4**2 + 3.*ssh4**2)
    svm5 = sqrt(s5**2 + 3.*ssh5**2)
    svm6 = sqrt(s6**2 + 3.*ssh6**2)
    svm7 = sqrt(s7**2 + 3.*ssh7**2)
```

```
    svm8 = sqrt(s8**2 + 3.*ssh8**2)


    return [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,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.      dcyl1 1.   tcyl1 .1
x2  0. 0. 30.      dcyl2 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  nelem 4  material s355  connect_mode no_connect
x1 0. -11. -12.5  dcyl1 1.5  tcyl1 0.016  pre_strain_long  0.0
x2 0.  11. -12.5  dcyl2 1.5  tcyl2 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  dcyl1 1.5  tcyl1 0.016  pre_strain_long  0.0
x2 0.  11.  12.5  dcyl2 1.5  tcyl2 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  dcyl1 2.  tcyl1 0.02  pre_strain_long  0.0
x2 0. -11.  12.5  dcyl2 2.  tcyl2 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  dcyl1 2.  tcyl1 0.02  pre_strain_long  0.0
x2 0.  11.  12.5  dcyl2 2.  tcyl2 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.   dcyl1 2.  tcyl1 0.08  pre_strain_long  0.0
x2   0.01  0.  0. dcyl2 2.  tcyl2 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. dcyl1 1.  tcyl1 0.1  pre_strain_long  0.0
x2  -6.    0.  0. dcyl2 1.  tcyl2 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.   dcyl1 1.  tcyl1 0.1  pre_strain_long  0.0
x2  -6.05   0.  0.   dcyl2 1.  tcyl2 0.1  btype beam
frac_stiff 1.0   cdn 0.  generator: var_rpm_2bl  !const_rpm !


!/ force sensor, rotor
new_elements_beam1  nelem 1  material s355  connect_mode connect_2
x1  -6.2   0.  0.    dcyl1 1.  tcyl1 0.03  pre_strain_long  0.0
x2  -6.1   0.  0.   dcyl2 1.  tcyl2 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 = 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     56081.63680000001
7.2     64379.43
7.68     73249.48480000002
8.16     82691.8012
8.64     92706.37920000001
9.12     103293.21880000002
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     193424.42080000002
12.96     208589.3532
13.44     224326.54720000003
13.92     240636.00280000002
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.  dcyl1 2.  tcyl1 0.08  pre_strain_long  0.0
x2  0.01  0.  1.  dcyl2 2.  tcyl2 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.  dcyl1 2.  tcyl1 0.08  pre_strain_long  0.0
x2  0.01  0. -1.  dcyl2 2.  tcyl2 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.  dcyl1  .2  tcyl1 0.008  pre_strain_long  0.0
x2   0. -11.  0.  dcyl2  .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.  dcyl1  .2  tcyl1 0.008  pre_strain_long  0.0
x2   0. 11.   0.  dcyl2   .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.    dcyl1  .2  tcyl1 0.008  pre_strain_long  0.0
x2   0.  0.  -12.5  dcyl2  .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.    dcyl1  .2  tcyl1 0.008  pre_strain_long  0.0
x2   0.  0.  12.5  dcyl2  .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.  dcyl1  .2  tcyl1 0.008  pre_strain_long  0.0
x2   0. -11.  0.  dcyl2  .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.  dcyl1  .2  tcyl1 0.008  pre_strain_long  0.0
x2   0. -11.  0.  dcyl2  .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.  dcyl1  .2  tcyl1 0.008  pre_strain_long  0.0
x2   0. 11.  0.  dcyl2   .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.  dcyl1  .2  tcyl1 0.008  pre_strain_long  0.0
x2   0. 11.     0.  dcyl2   .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.    dcyl1  .2  tcyl1 0.008  pre_strain_long  0.0
x2   0.  0.  -12.5  dcyl2  .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.    dcyl1  .2  tcyl1 0.008  pre_strain_long  0.0
x2   0.  0.  12.5  dcyl2  .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
!
!                                                                                                                   dy        dz        dy        dz        dy
!   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
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
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.
8.74513274e+00             1.10000000e+06           1.10000000e+07      1.10000000e+10          2.E+9     3.70000000e+01    4.200e-01        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.
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.
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.
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.
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.
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.
1.251000000e+01       6.50000000e+05           4.50000000e+06      4.50000000e+09          2.E+8      0.                 0.00e+00         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.
0.375                 6.00000000e+10      6.00000000e+10      6.00000000e+10      2.E+10     3.e2              2.05000000e+01    0.   0.   0.   0.     0.   0.   0.
0.75                  6.00000000e+10      6.00000000e+10      6.00000000e+10            2.E+10     3.e2              2.05000000e+01    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.
1.67345133e+00             6.00000000e+10           6.00000000e+10      6.00000000e+10          2.E+10     1.28666667e+02    2.06000000e+01         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.
2.91504425e+00             3.50000000e+10           6.80000000e+10      6.80000000e+10          2.E+10     8.45000000e+01    1.70000000e+01         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.
4.21061947e+00             1.71000000e+10           3.70000000e+10      3.70000000e+10          2.E+10     6.20000000e+01    8.47000000e+00         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.
5.50619469e+00             7.00000000e+10           2.50000000e+10      2.50000000e+10          2.E+10     5.33333333e+01    4.37000000e+00         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.
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.
1.251000000e+01       6.50000000e+10           4.50000000e+10      4.50000000e+09          2.E+8      0.                 0.00000000e+00        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  nelem 1 material s355  connect_mode auto   ! scale to hub mass
x1    0.  0.  0.35      dcyl1    0.3     tcyl1    0.03
x2    0.  0.  0.        dcyl2    0.3     tcyl2    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_rlds_1       scale: 1.e-3         unit: [kNm]
nfact: 1 source: my_rlds_1       scale: 1.e-3         unit: [kNm]
nfact: 1 source: mz_rlds_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_rlds_2       scale: 1.e-3         unit: [kNm]
nfact: 1 source: my_rlds_2       scale: 1.e-3         unit: [kNm]
nfact: 1 source: mz_rlds_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: mz_rlus_2      scale: 1.e-3        unit: [kNm]
nfact: 1 source: rear_right_down_strut_fx_1    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: mx_rrds_1      scale: 1.e-3        unit: [kNm]
nfact: 1 source: my_rrds_1      scale: 1.e-3        unit: [kNm]
nfact: 1 source: mz_rrds_1      scale: 1.e-3        unit: [kNm]
nfact: 1 source: rear_right_down_strut_fx_2    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: mx_rrds_2      scale: 1.e-3        unit: [kNm]
nfact: 1 source: my_rrds_2      scale: 1.e-3        unit: [kNm]
nfact: 1 source: mz_rrds_2      scale: 1.e-3        unit: [kNm]
nfact: 1 source: rear_right_up_strut_fx_1      scale: 1.e-3        unit: [kN]  !
nfact: 1 source: mx_rrus_1      scale: 1.e-3        unit: [kNm]
nfact: 1 source: my_rrus_1      scale: 1.e-3        unit: [kNm]
nfact: 1 source: mz_rrus_1      scale: 1.e-3        unit: [kNm]
nfact: 1 source: rear_right_up_strut_fx_2      scale: 1.e-3        unit: [kN]  !
nfact: 1 source: mx_rrus_2      scale: 1.e-3        unit: [kNm]
nfact: 1 source: my_rrus_2      scale: 1.e-3        unit: [kNm]
nfact: 1 source: mz_rrus_2      scale: 1.e-3        unit: [kNm]
nfact: 1 source: rear_vert_down_strut_fx_1     scale: 1.e-3           unit: [kN]  !
nfact: 1 source: mx_rvds_1      scale: 1.e-3        unit: [kNm]
nfact: 1 source: my_rvds_1      scale: 1.e-3        unit: [kNm]
nfact: 1 source: mz_rvds_1      scale: 1.e-3        unit: [kNm]
nfact: 1 source: rear_vert_down_strut_fx_2     scale: 1.e-3           unit: [kN]  !
nfact: 1 source: mx_rvds_2     scale: 1.e-3        unit: [kNm]
nfact: 1 source: my_rvds_2      scale: 1.e-3        unit: [kNm]
nfact: 1 source: mz_rvds_2      scale: 1.e-3        unit: [kNm]
nfact: 1 source: rear_vert_up_strut_fx_1      scale: 1.e-3           unit: [kN]  !
nfact: 1 source: mx_rvus_1      scale: 1.e-3        unit: [kNm]
nfact: 1 source: my_rvus_1      scale: 1.e-3        unit: [kNm]
nfact: 1 source: mz_rvus_1      scale: 1.e-3        unit: [kNm]
nfact: 1 source: rear_vert_up_strut_fx_2      scale: 1.e-3           unit: [kN]  !
nfact: 1 source: mx_rvus_2      scale: 1.e-3        unit: [kNm]
nfact: 1 source: my_rvus_2      scale: 1.e-3        unit: [kNm]
nfact: 1 source: mz_rvus_2      scale: 1.e-3        unit: [kNm]
nfact: 1 source: front_hor_left_strut_fy_1    scale: 1.e-3        unit: [kN]  !  Fy and Fz for strut endpoints x1 and x2
nfact: 1 source: front_hor_left_strut_fz_1    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_hor_left_strut_fy_2    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_hor_left_strut_fz_2    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_hor_right_strut_fy_1   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_hor_right_strut_fz_1   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_hor_right_strut_fy_2   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_hor_right_strut_fz_2   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_vert_down_strut_fy_1   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_vert_down_strut_fz_1   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_vert_down_strut_fy_2   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_vert_down_strut_fz_2   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_vert_up_strut_fy_1     scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_vert_up_strut_fz_1     scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_vert_up_strut_fy_2     scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_vert_up_strut_fz_2     scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_left_down_strut_fy_1    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_left_down_strut_fz_1    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_left_down_strut_fy_2    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_left_down_strut_fz_2    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_left_up_strut_fy_1      scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_left_up_strut_fz_1      scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_left_up_strut_fy_2      scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_left_up_strut_fz_2      scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_right_down_strut_fy_1    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_right_down_strut_fz_1    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_right_down_strut_fy_2    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_right_down_strut_fz_2    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_right_up_strut_fy_1      scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_right_up_strut_fz_1      scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_right_up_strut_fy_2      scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_right_up_strut_fz_2      scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_vert_down_strut_fy_1     scale: 1.e-3           unit: [kN]  !
nfact: 1 source: rear_vert_down_strut_fz_1     scale: 1.e-3           unit: [kN]  !
nfact: 1 source: rear_vert_down_strut_fy_2     scale: 1.e-3           unit: [kN]  !
nfact: 1 source: rear_vert_down_strut_fz_2     scale: 1.e-3           unit: [kN]  !
nfact: 1 source: rear_vert_up_strut_fy_1      scale: 1.e-3           unit: [kN]  !
nfact: 1 source: rear_vert_up_strut_fz_1      scale: 1.e-3           unit: [kN]  !
nfact: 1 source: rear_vert_up_strut_fy_2      scale: 1.e-3           unit: [kN]  !
nfact: 1 source: rear_vert_up_strut_fz_2      scale: 1.e-3           unit: [kN]  !
nfact: 1 source: front_hor_left_strut_fx_3    scale: 1.e-3        unit: [kN]  ! midpoints at front struts
nfact: 1 source: front_hor_left_strut_fy_3    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_hor_left_strut_fz_3    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: mx_fhls_3      scale: 1.e-3        unit: [kNm]
nfact: 1 source: my_fhls_3      scale: 1.e-3        unit: [kNm]
nfact: 1 source: mz_fhls_3      scale: 1.e-3        unit: [kNm]
nfact: 1 source: front_hor_right_strut_fx_3   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_hor_right_strut_fy_3   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_hor_right_strut_fz_3   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: mx_fhrs_3      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.  ! 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.


! 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: 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.      dcyl1 1.   tcyl1 .1
x2  0. 0. 30.      dcyl2 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  nelem 4  material s355  connect_mode no_connect
x1 0. -11. -12.5  dcyl1 1.5  tcyl1 0.016  pre_strain_long  0.0
x2 0.  11. -12.5  dcyl2 1.5  tcyl2 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  dcyl1 1.5  tcyl1 0.016  pre_strain_long  0.0
x2 0.  11.  12.5  dcyl2 1.5  tcyl2 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  dcyl1 2.  tcyl1 0.02  pre_strain_long  0.0
x2 0. -11.  12.5  dcyl2 2.  tcyl2 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  dcyl1 2.  tcyl1 0.02  pre_strain_long  0.0
x2 0.  11.  12.5  dcyl2 2.  tcyl2 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.   dcyl1 2.  tcyl1 0.08  pre_strain_long  0.0
x2   0.01  0.  0. dcyl2 2.  tcyl2 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. dcyl1 1.  tcyl1 0.1  pre_strain_long  0.0
x2  -6.    0.  0. dcyl2 1.  tcyl2 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.   dcyl1 1.  tcyl1 0.1  pre_strain_long  0.0
x2  -6.05 0.  0.   dcyl2 1.  tcyl2 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.    dcyl1 1.  tcyl1 0.03  pre_strain_long  0.0
x2  -6.1   0.  0.   dcyl2 1.  tcyl2 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.  dcyl1 2.  tcyl1 0.08  pre_strain_long  0.0
x2  0.01  0.  1.  dcyl2 2.  tcyl2 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.  dcyl1 2.  tcyl1 0.08  pre_strain_long  0.0
x2  0.01  0. -1.  dcyl2 2.  tcyl2 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.  dcyl1  .2  tcyl1 0.008  pre_strain_long  0.0
x2   0. -11.  0.  dcyl2  .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.  dcyl1  .2  tcyl1 0.008  pre_strain_long  0.0
x2   0. 11.   0.  dcyl2   .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.    dcyl1  .2  tcyl1 0.008  pre_strain_long  0.0
x2   0.  0.  -12.5  dcyl2  .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.    dcyl1  .2  tcyl1 0.008  pre_strain_long  0.0
x2   0.  0.  12.5  dcyl2  .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.  dcyl1  .2  tcyl1 0.008  pre_strain_long  0.0
x2   0. -11.  0.  dcyl2  .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.  dcyl1  .2  tcyl1 0.008  pre_strain_long  0.0
x2   0. -11.  0.  dcyl2  .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.  dcyl1  .2  tcyl1 0.008  pre_strain_long  0.0
x2   0. 11.  0.  dcyl2   .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.  dcyl1  .2  tcyl1 0.008  pre_strain_long  0.0
x2   0. 11.     0.  dcyl2   .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.    dcyl1  .2  tcyl1 0.008  pre_strain_long  0.0
x2   0.  0.  -12.5  dcyl2  .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.    dcyl1  .2  tcyl1 0.008  pre_strain_long  0.0
x2   0.  0.  12.5  dcyl2  .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
!
!                                                                                                             dy         dz        dy        dz        dy
!   r[m]    EI_f[Nm!
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.
0.375                 6.00000000e+07      6.00000000e+07      6.00000000e+10      2.E+10     3.e2              2.05000000e+01    0.   0.   0.   0.     0.   0.   0.
0.75                  6.00000000e+07      6.00000000e+07      6.00000000e+10            2.E+10     3.e2              2.05000000e+01    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.
1.67345133e+00             6.00000000e+07           6.00000000e+07      6.00000000e+10          2.E+10     1.28666667e+02    2.06000000e+01         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.
2.91504425e+00             3.50000000e+07           6.80000000e+07      6.80000000e+10          2.E+10     8.45000000e+01    1.70000000e+01         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.
4.21061947e+00             1.71000000e+07           3.70000000e+07      3.70000000e+10          2.E+10     6.20000000e+01    8.47000000e+00         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.
5.50619469e+00             7.00000000e+06           2.50000000e+07      2.50000000e+10          2.E+10     5.33333333e+01    4.37000000e+00         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.
6.80176991e+00             3.40000000e+06           1.80000000e+07      1.80000000e+10          2.E+10     4.45000000e+01    2.05000000e+00         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.
8.09734513e+00             1.70000000e+06           1.30000000e+07      1.30000000e+10          2.E+9     3.95000000e+01    7.80000000e-01         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.
9.39292035e+00             8.00000000e+05           9.11000000e+06      9.11000000e+09          2.E+9     3.50000000e+01    2.20000000e-01         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.
1.05805310e+01             6.50000000e+05           5.30000000e+06      5.30000000e+09          2.E+9     2.32500000e+01    0.00000000e+00         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.
1.15522124e+01             6.50000000e+05           4.60000000e+06      4.60000000e+09          2.E+8     2.65833333e+01    0.00000000e+00         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.
1.251000000e+01       6.50000000e+05           4.50000000e+06      4.50000000e+09          2.E+8      0.                 0.00000000e+00         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.
0.375                 6.00000000e+10      6.00000000e+10      6.00000000e+10      2.E+10     3.e2              2.05000000e+01    0.   0.   0.   0.     0.   0.   0.
0.75                  6.00000000e+10      6.00000000e+10      6.00000000e+10            2.E+10     3.e2              2.05000000e+01    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.
1.67345133e+00             6.00000000e+10           6.00000000e+10      6.00000000e+10          2.E+10     1.28666667e+02    2.06000000e+01         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.
2.91504425e+00             3.50000000e+10           6.80000000e+10      6.80000000e+10          2.E+10     8.45000000e+01    1.70000000e+01         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.
4.21061947e+00             1.71000000e+10           3.70000000e+10      3.70000000e+10          2.E+10     6.20000000e+01    8.47000000e+00         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.
5.50619469e+00             7.00000000e+10           2.50000000e+10      2.50000000e+10          2.E+10     5.33333333e+01    4.37000000e+00         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.
```

```
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.
1.251000000e+01      6.50000000e+10          4.50000000e+10     4.50000000e+09        2.E+8     0.                 0.00000000e+00       0.   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.  0.35      dcyl1    0.3     tcyl1    0.03
x2    0.  0.  0.        dcyl2    0.3     tcyl2    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.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_rlds_1       scale: 1.e-3         unit: [kNm]
nfact: 1 source: my_rlds_1       scale: 1.e-3         unit: [kNm]
nfact: 1 source: mz_rlds_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_rlds_2       scale: 1.e-3         unit: [kNm]
nfact: 1 source: my_rlds_2       scale: 1.e-3         unit: [kNm]
nfact: 1 source: mz_rlds_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: mz_rlus_2       scale: 1.e-3        unit: [kNm]
nfact: 1 source: rear_right_down_strut_fx_1    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: mx_rrds_1       scale: 1.e-3        unit: [kNm]
nfact: 1 source: my_rrds_1       scale: 1.e-3        unit: [kNm]
nfact: 1 source: mz_rrds_1       scale: 1.e-3        unit: [kNm]
nfact: 1 source: rear_right_down_strut_fx_2    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: mx_rrds_2       scale: 1.e-3        unit: [kNm]
nfact: 1 source: my_rrds_2       scale: 1.e-3        unit: [kNm]
nfact: 1 source: mz_rrds_2       scale: 1.e-3        unit: [kNm]
nfact: 1 source: rear_right_up_strut_fx_1      scale: 1.e-3        unit: [kN]  !
nfact: 1 source: mx_rrus_1       scale: 1.e-3        unit: [kNm]
nfact: 1 source: my_rrus_1       scale: 1.e-3        unit: [kNm]
nfact: 1 source: mz_rrus_1       scale: 1.e-3        unit: [kNm]
nfact: 1 source: rear_right_up_strut_fx_2      scale: 1.e-3        unit: [kN]  !
nfact: 1 source: mx_rrus_2       scale: 1.e-3        unit: [kNm]
nfact: 1 source: my_rrus_2       scale: 1.e-3        unit: [kNm]
nfact: 1 source: mz_rrus_2       scale: 1.e-3        unit: [kNm]
nfact: 1 source: rear_vert_down_strut_fx_1     scale: 1.e-3           unit: [kN]  !
nfact: 1 source: mx_rvds_1       scale: 1.e-3        unit: [kNm]
nfact: 1 source: my_rvds_1       scale: 1.e-3        unit: [kNm]
nfact: 1 source: mz_rvds_1       scale: 1.e-3        unit: [kNm]
nfact: 1 source: rear_vert_down_strut_fx_2     scale: 1.e-3           unit: [kN]  !
nfact: 1 source: mx_rvds_2      scale: 1.e-3        unit: [kNm]
nfact: 1 source: my_rvds_2       scale: 1.e-3        unit: [kNm]
nfact: 1 source: mz_rvds_2       scale: 1.e-3        unit: [kNm]
nfact: 1 source: rear_vert_up_strut_fx_1      scale: 1.e-3           unit: [kN]  !
nfact: 1 source: mx_rvus_1       scale: 1.e-3        unit: [kNm]
nfact: 1 source: my_rvus_1       scale: 1.e-3        unit: [kNm]
nfact: 1 source: mz_rvus_1       scale: 1.e-3        unit: [kNm]
nfact: 1 source: rear_vert_up_strut_fx_2      scale: 1.e-3           unit: [kN]  !
nfact: 1 source: mx_rvus_2       scale: 1.e-3        unit: [kNm]
nfact: 1 source: my_rvus_2       scale: 1.e-3        unit: [kNm]
nfact: 1 source: mz_rvus_2       scale: 1.e-3        unit: [kNm]
nfact: 1 source: front_hor_left_strut_fy_1    scale: 1.e-3        unit: [kN]  !  Fy and Fz for strut endpoints x1 and x2
nfact: 1 source: front_hor_left_strut_fz_1    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_hor_left_strut_fy_2    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_hor_left_strut_fz_2    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_hor_right_strut_fy_1   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_hor_right_strut_fz_1   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_hor_right_strut_fy_2   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_hor_right_strut_fz_2   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_vert_down_strut_fy_1   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_vert_down_strut_fz_1   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_vert_down_strut_fy_2   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_vert_down_strut_fz_2   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_vert_up_strut_fy_1     scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_vert_up_strut_fz_1     scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_vert_up_strut_fy_2     scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_vert_up_strut_fz_2     scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_left_down_strut_fy_1    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_left_down_strut_fz_1    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_left_down_strut_fy_2    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_left_down_strut_fz_2    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_left_up_strut_fy_1      scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_left_up_strut_fz_1      scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_left_up_strut_fy_2      scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_left_up_strut_fz_2      scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_right_down_strut_fy_1    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_right_down_strut_fz_1    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_right_down_strut_fy_2    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_right_down_strut_fz_2    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_right_up_strut_fy_1      scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_right_up_strut_fz_1      scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_right_up_strut_fy_2      scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_right_up_strut_fz_2      scale: 1.e-3        unit: [kN]  !
nfact: 1 source: rear_vert_down_strut_fy_1     scale: 1.e-3           unit: [kN]  !
nfact: 1 source: rear_vert_down_strut_fz_1     scale: 1.e-3           unit: [kN]  !
nfact: 1 source: rear_vert_down_strut_fy_2     scale: 1.e-3           unit: [kN]  !
nfact: 1 source: rear_vert_down_strut_fz_2     scale: 1.e-3           unit: [kN]  !
nfact: 1 source: rear_vert_up_strut_fy_1      scale: 1.e-3           unit: [kN]  !
nfact: 1 source: rear_vert_up_strut_fz_1      scale: 1.e-3           unit: [kN]  !
nfact: 1 source: rear_vert_up_strut_fy_2      scale: 1.e-3           unit: [kN]  !
nfact: 1 source: rear_vert_up_strut_fz_2      scale: 1.e-3           unit: [kN]  !
nfact: 1 source: front_hor_left_strut_fx_3    scale: 1.e-3        unit: [kN]  ! midpoints at front struts
nfact: 1 source: front_hor_left_strut_fy_3    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_hor_left_strut_fz_3    scale: 1.e-3        unit: [kN]  !
nfact: 1 source: mx_fhls_3       scale: 1.e-3        unit: [kNm]
nfact: 1 source: my_fhls_3       scale: 1.e-3        unit: [kNm]
nfact: 1 source: mz_fhls_3       scale: 1.e-3        unit: [kNm]
nfact: 1 source: front_hor_right_strut_fx_3   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_hor_right_strut_fy_3   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: front_hor_right_strut_fz_3   scale: 1.e-3        unit: [kN]  !
nfact: 1 source: mx_fhrs_3       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

```python
# -*- coding: utf-8 -*-
"""
Created on Thu May 20 14:44:57 2021

@author: veaa
"""

#Buckling of front struts
# EUROCODE 3
# 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

from pylab import *
from numpy import loadtxt
import numpy as np


#importing data
a    = loadtxt('sensors.txt',skiprows=50000,dtype='float',delimiter=';')  # skip 500 s

pi = 3.14159        #

#cross-sectional properties and length
d = 0.2               #outer diameter of strut [m]
twall = 0.008         #wall thickness [m]
di = d - 2.*twall  #inner diameter of strut [m]
A = .25*pi*(d**2-di**2)  #area of cross-section (cylinder) [m^2]
icyl = pi*(d**4-di**4)/64. #second moment of inertia for cylinder [m^4]
W_pl = (1/6)*(d**3-di**3)  #plastic section modulus [m^3]
W_el = icyl/(d/2)  #elastic section modulus [m^3]
L_cr = 10             #Buckle length [m]

#Tverrsnittsklasse
epsilon = sqrt(235/355)

```

```
#tv.klasse 1
if d/twall <= 50*epsilon**2:
    W_el = W_pl
#tv.klasse 2
if d/twall <= 70*epsilon**2:
    W_el = W_pl
#tv.klasse 3
d/twall <= 90*epsilon**2



#Material properties
gamma_M1 = 1.05      # [Dimensionless]
E = 210*10**9        # [Pa]
f_y = 355*10**6      # [Pa]


#Euler's critical load
N_cr = (pi*E*icyl)/((L_cr)**2) # [N]

#Relative slenderness
lambda_ybar = sqrt((A*f_y)/N_cr)

#form deviation factor from table: cyrve a = 0.21
alpha = 0.21

#reduction factor for buckling (khi)
phi = 0.5*(1+alpha*(lambda_ybar-0.2)+(lambda_ybar)**2)

khi = (1)/(phi+sqrt((phi)**2-(lambda_ybar)**2))

#Design criterias
N_bxRd = (khi*f_y*A)/(gamma_M1)

M_elxRd = (f_y*W_el)/(gamma_M1)



K = []
KK = []
KKK = []
KKKK = []


#front_hor_left_strut_x1
front_hor_left_strut_fx_1 = a[:,22]*1.e3; #[N]
my_fhls_1 = a[:,24]*1.e3; #[Nm]
mz_fhls_1 = a[:,25]*1.e3; #[Nm]

```

```
for i in range(0,len(front_hor_left_strut_fx_1)):
    S = (front_hor_left_strut_fx_1[i]/N_bxRd) + ((-abs(my_fhls_1[i])-abs(mz_fhls_1[i]))/M_elxRd)
    K.append(S)

K_min = (min(K))

if abs(K_min)>1:
    print('%.2e, and front_hor_left_strut will likely buckle under the current circumstances' %K_min)
else :
    print('%.2e, front_hor_left_strut is not prone to buckling under the current circumstances' %K_min)


#front_hor_right_strut_x1
front_hor_right_strut_fx_1 = a[:,30]*1.e3;
my_fhrs_1 = a[:,32]*1.e3;
mz_fhrs_1 = a[:,33]*1.e3;

for i in range(0,len(front_hor_left_strut_fx_1)):
    SS = (front_hor_right_strut_fx_1[i]/N_bxRd) + ((-abs(my_fhrs_1[i])-abs(mz_fhrs_1[i]))/M_elxRd)
    KK.append(SS)

KK_min = (min(KK))

if abs(KK_min)>1:
    print('%.2e, and front_hor_right_strut will likely buckle under the current circumstances' %KK_min)
else :
    print('%.2e, front_hor_right_strut is not prone to buckling under the current circumstances' %KK_min)

#front_vert_down_strut_x1
front_vert_down_strut_fx_1 = a[:,38]*1.e3;
my_fvds_1 = a[:,40]*1.e3;
mz_fvds_1 = a[:,41]*1.e3;

for i in range(0,len(front_hor_left_strut_fx_1)):
    SSS = (front_vert_down_strut_fx_1[i]/N_bxRd) + ((-abs(my_fvds_1[i])-abs(mz_fvds_1[i]))/M_elxRd)
    KKK.append(SSS)

KKK_min = (min(KKK))

if abs(KKK_min)>1:
```

```
    print('%.2e, and front_vert_down_strut will likely buckle under the current circumstances' %KKK_min)
else :
    print('%.2e, front_vert_down_strut is not prone to buckling under the current circumstances' %KKK_min)

#front_vert_up_strut_x1
front_vert_up_strut_fx_1 = a[:,46]*1.e3;
my_fvus_1 = a[:,48]*1.e3;
mz_fvus_1 = a[:,49]*1.e3;

for i in range(0,len(front_hor_left_strut_fx_1)):
    SSSS = (front_vert_up_strut_fx_1[i]/N_bxRd) + ((-abs(my_fvus_1[i])-abs(mz_fvus_1[i]))/M_elxRd)
    KKKK.append(SSSS)

KKKK_min = (min(KKKK))

if abs(KKKK_min)>1:
    print('%.2e, and front_vert_up_strut will likely buckle under the current circumstances' %KKKK_min)
else :
    print('%.2e, front_vert_up_strut is not prone to buckling under the current circumstances' %KKKK_min)


#-----------------------------

B = []
BB = []
BBB = []
BBBB = []

#front_hor_left_strut_x2
front_hor_left_strut_fx_2 = a[:,26]*1.e3;
my_fhls_2 = a[:,28]*1.e3;
mz_fhls_2 = a[:,29]*1.e3;



for i in range(0,len(front_hor_left_strut_fx_1)):
    T = (front_hor_left_strut_fx_2[i]/N_bxRd) + ((-abs(my_fhls_2[i])-abs(mz_fhls_2[i]))/M_elxRd)
    B.append(T)

B_min = (min(B))

if abs(B_min)>1:
    print('%.2e, and front_hor_left_strut will likely buckle under the
```

```
        current circumstances' %B_min)
else :
    print('%.2e, front_hor_left_strut is not prone to buckling under the
        current circumstances' %B_min)



#front_hor_right_strut_x2
front_hor_right_strut_fx_2 = a[:,34]*1.e3;
my_fhrs_2 = a[:,36]*1.e3;
mz_fhrs_2 = a[:,37]*1.e3;

for i in range(0,len(front_hor_left_strut_fx_1)):
    TT = (front_hor_right_strut_fx_2[i]/N_bxRd) + ((-abs(my_fhrs_2[i])-abs
        (mz_fhrs_2[i]))/M_elxRd)
    BB.append(TT)

BB_min = (min(BB))

if abs(BB_min)>1:
    print('%.2e, and front_hor_right_strut will likely buckle under the
        current circumstances' %BB_min)
else :
    print('%.2e, front_hor_right_strut is not prone to buckling under the
        current circumstances' %BB_min)

#front_vert_down_strut_x2
front_vert_down_strut_fx_2 = a[:,42]*1.e3;
my_fvds_2 = a[:,44]*1.e3;
mz_fvds_2 = a[:,45]*1.e3;

for i in range(0,len(front_hor_left_strut_fx_1)):
    TTT = (front_vert_down_strut_fx_2[i]/N_bxRd) + ((-abs(my_fvds_2[i])-
        abs(mz_fvds_2[i]))/M_elxRd)
    BBB.append(TTT)

BBB_min = (min(BBB))

if abs(BBB_min)>1:
    print('%.2e, and front_vert_down_strut will likely buckle under the
        current circumstances' %BBB_min)
else :
    print('%.2e, front_vert_down_strut is not prone to buckling under the
        current circumstances' %BBB_min)

#front_vert_up_strut_x2
front_vert_up_strut_fx_2 = a[:,50]*1.e3;
my_fvus_2 = a[:,52]*1.e3;
mz_fvus_2 = a[:,53]*1.e3;

```

```
for i in range(0,len(front_hor_left_strut_fx_1)):
    TTTT = (front_vert_up_strut_fx_2[i]/N_bxRd) + ((-abs(my_fvus_2[i])-abs(mz_fvus_2[i]))/M_elxRd)
    BBBB.append(TTTT)

BBBB_min = (min(BBBB))

if abs(BBBB_min)>1:
    print('%.2e, and front_vert_up_strut will likely buckle under the current circumstances' %BBBB_min)
else :
    print('%.2e, front_vert_up_strut is not prone to buckling under the current circumstances' %BBBB_min)
```

E. Fatigue Life vs. Wall Thickness

<u>**2-Bladed Structure**</u>

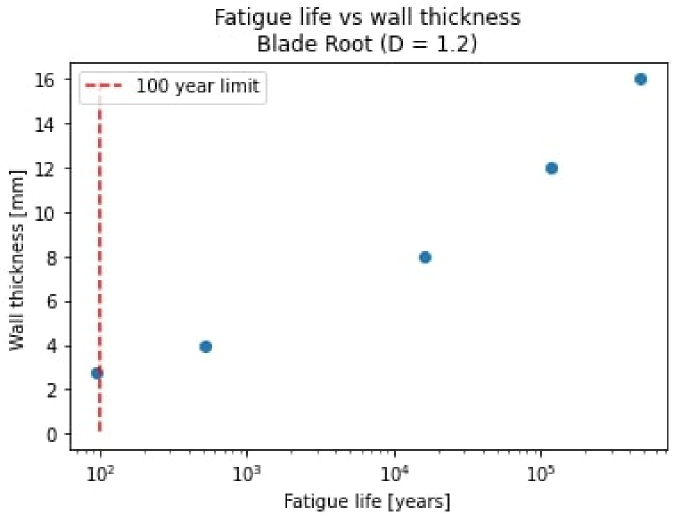

Figure E.1

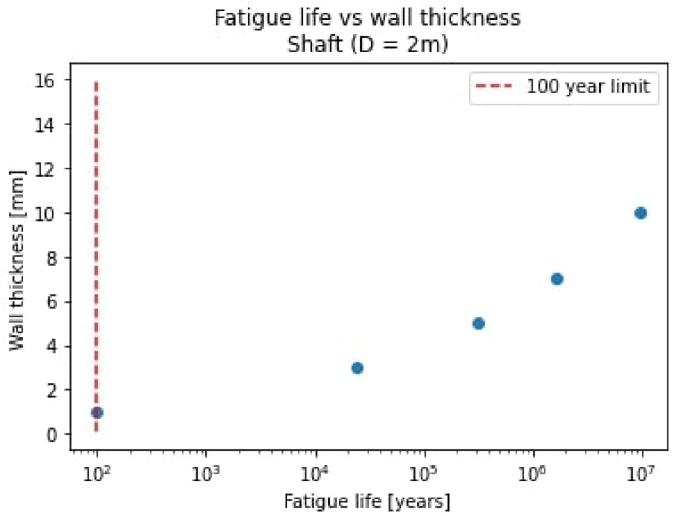

Figure E.2

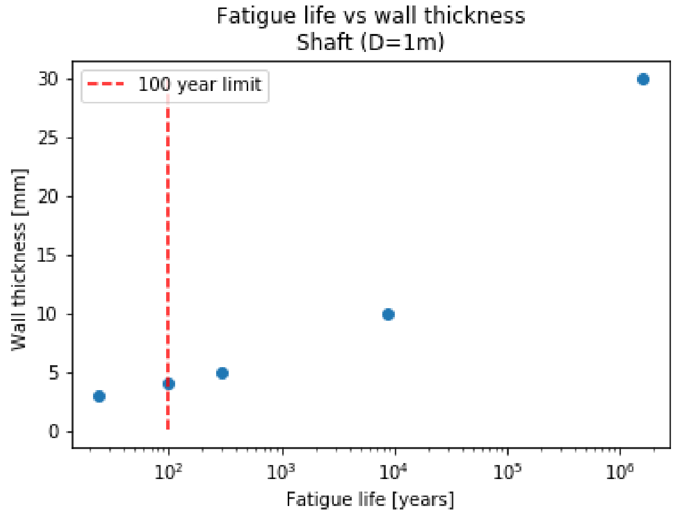

Figure E.3

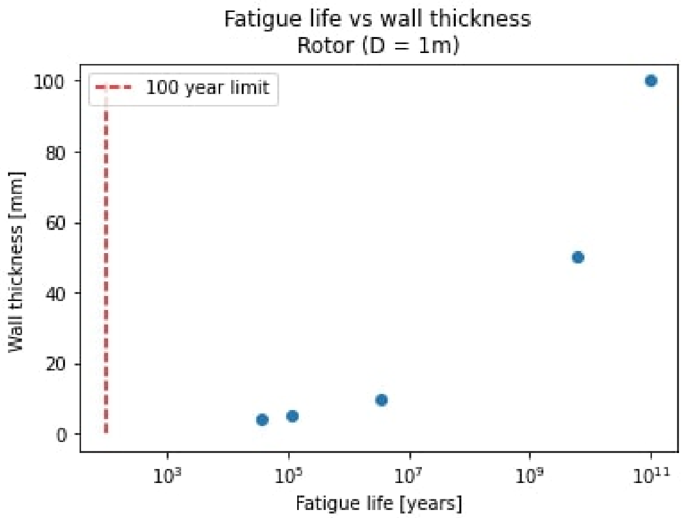

Figure E.4

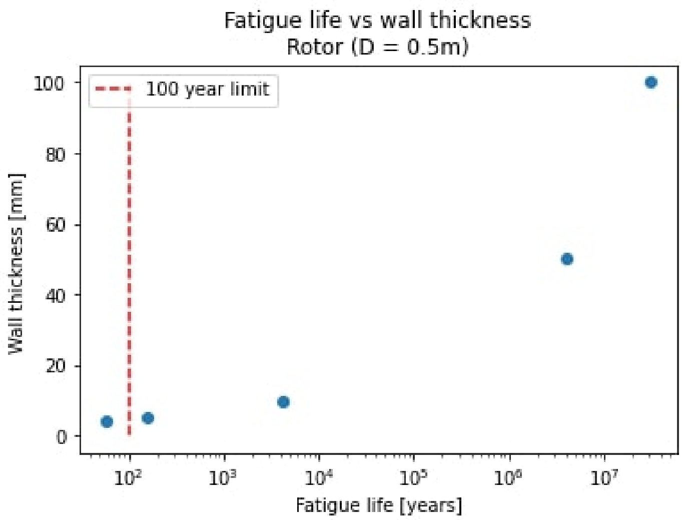

Figure E.5

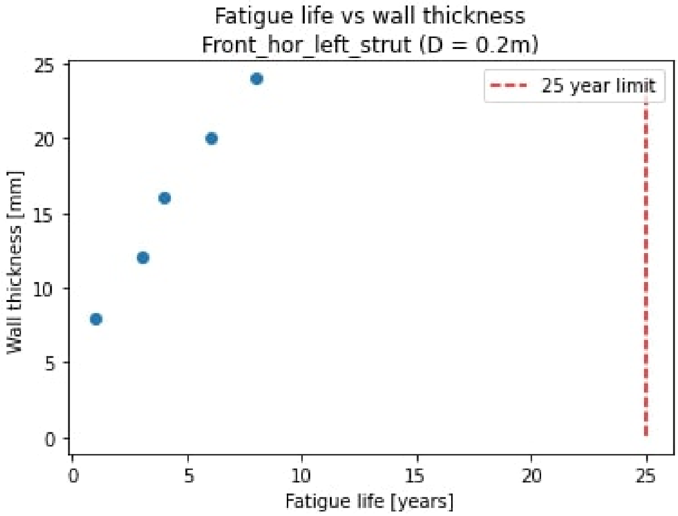

Figure E.6

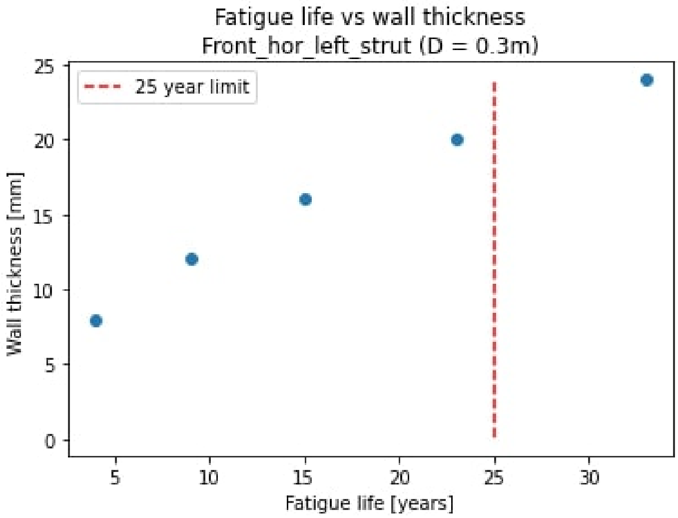

Figure E.7

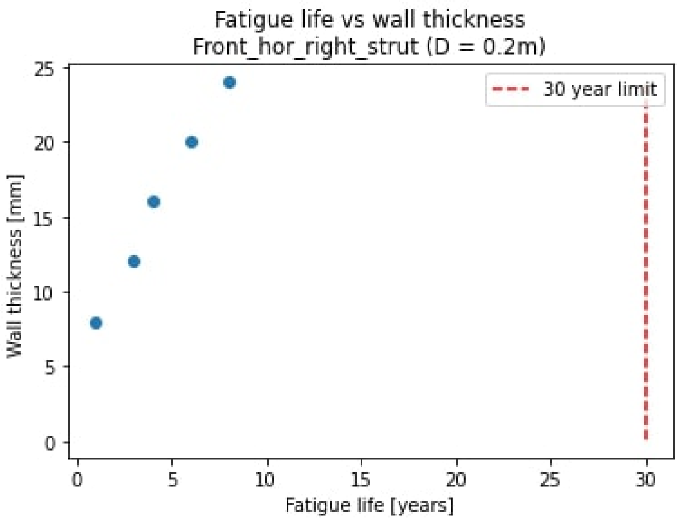

Figure E.8

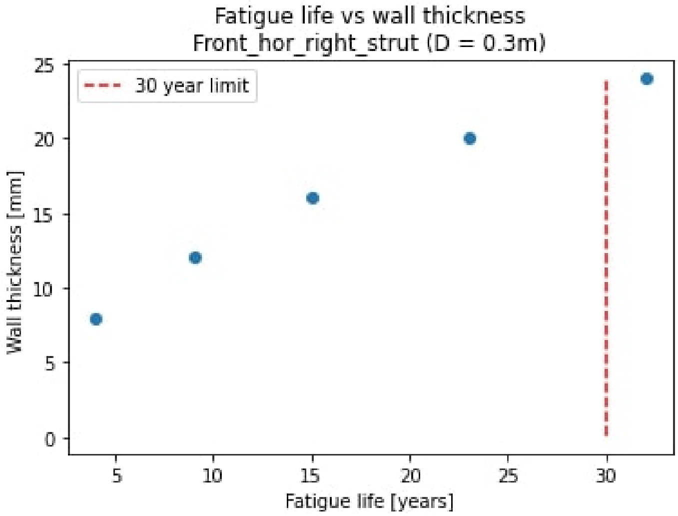

Figure E.9

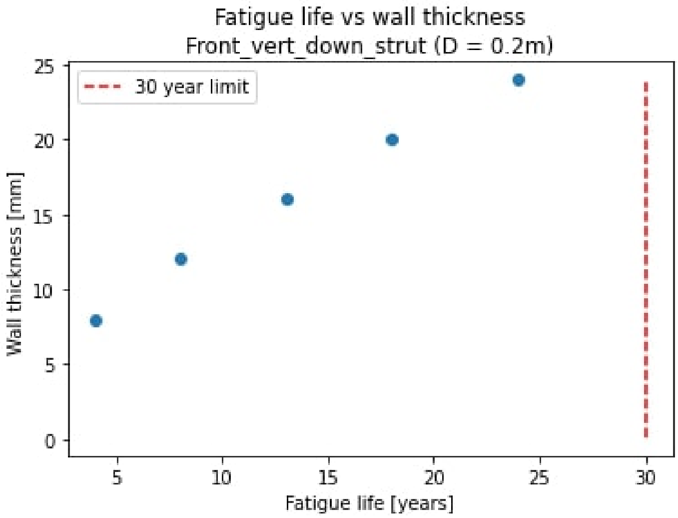

Figure E.10

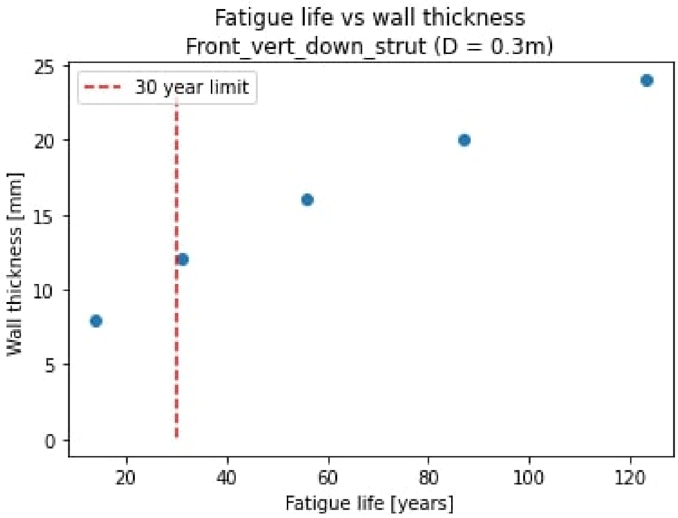

Figure E.11

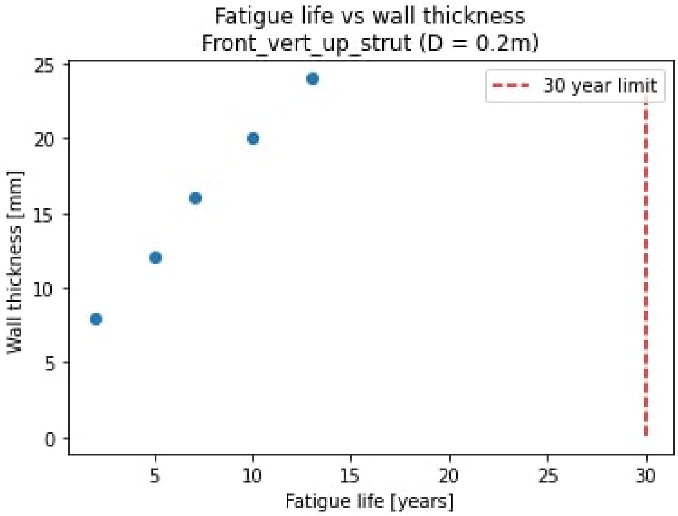

Figure E.12

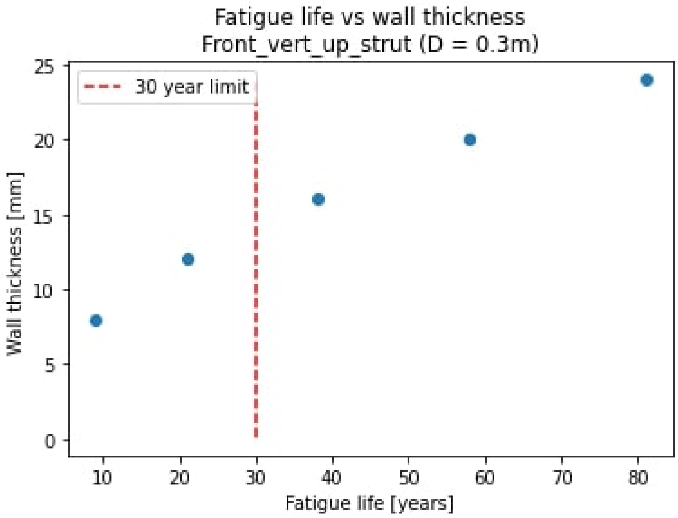

Figure E.13

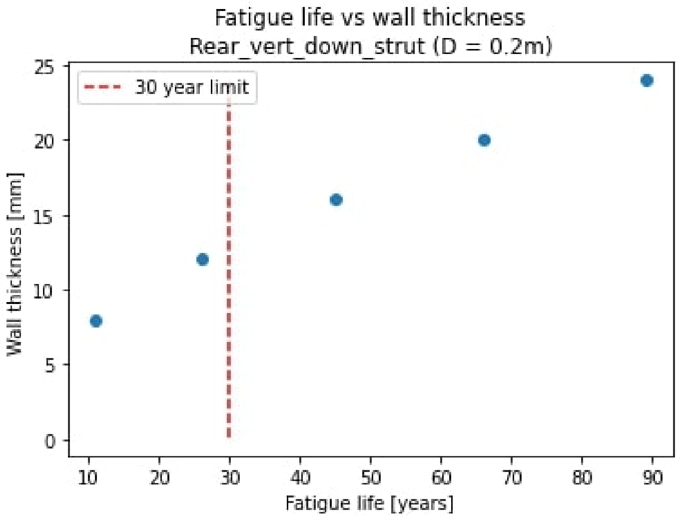

Figure E.14

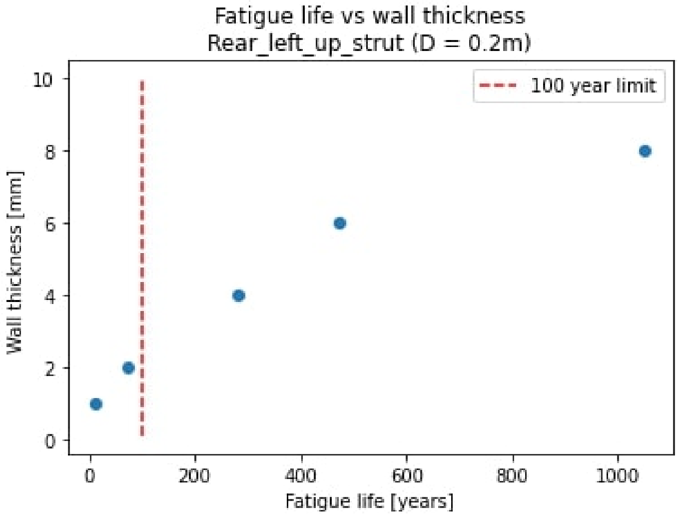

Figure E.15

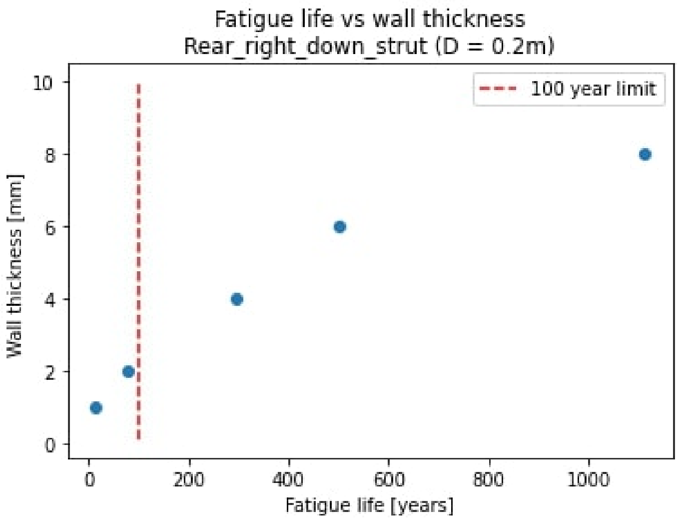

Figure E.16

4-Bladed Structure

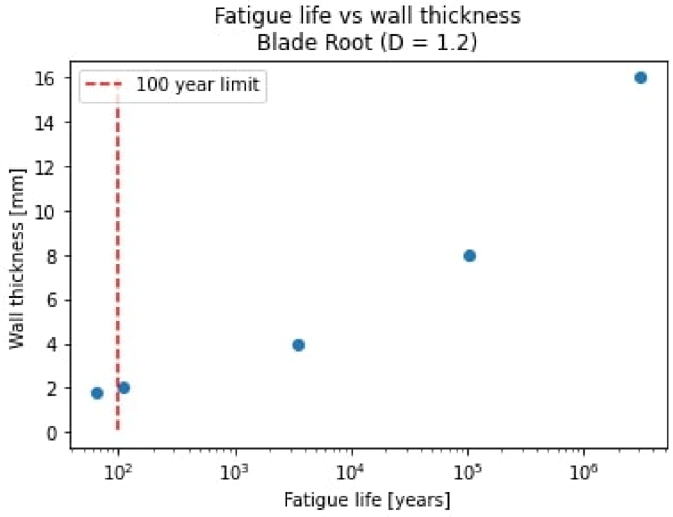

Figure E.17

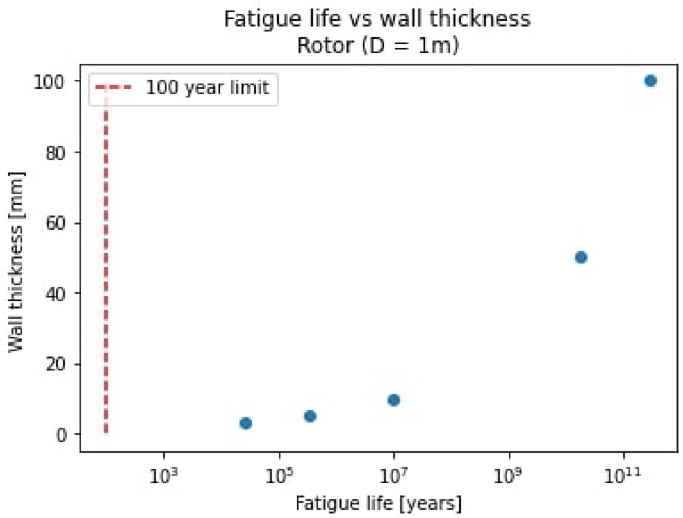

Figure E.18

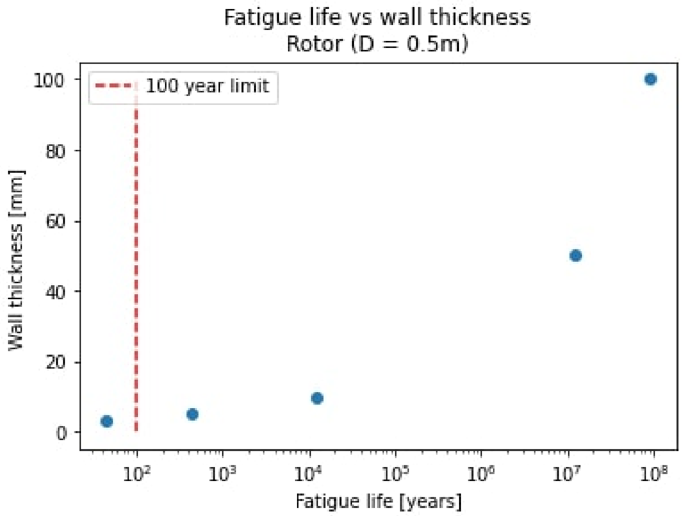

Figure E.19

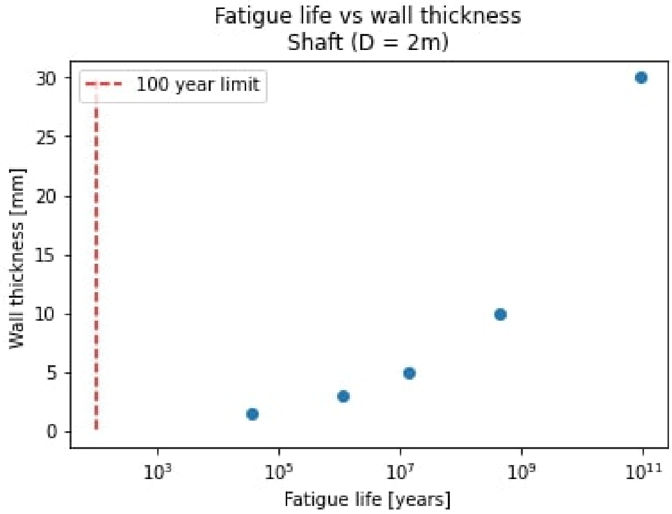

Figure E.20

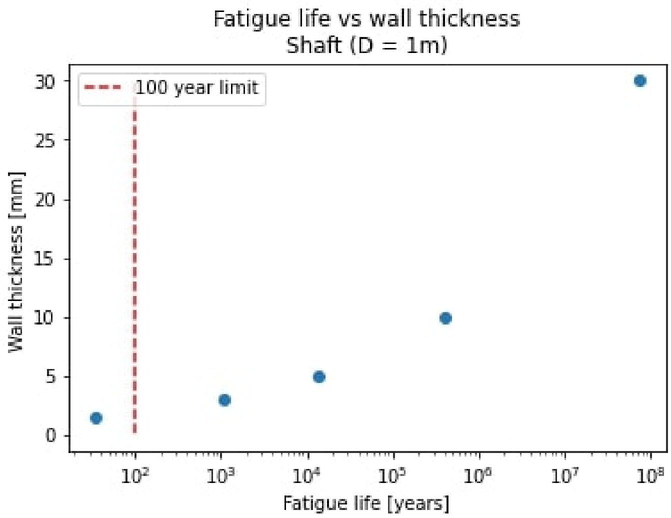

Figure E.21

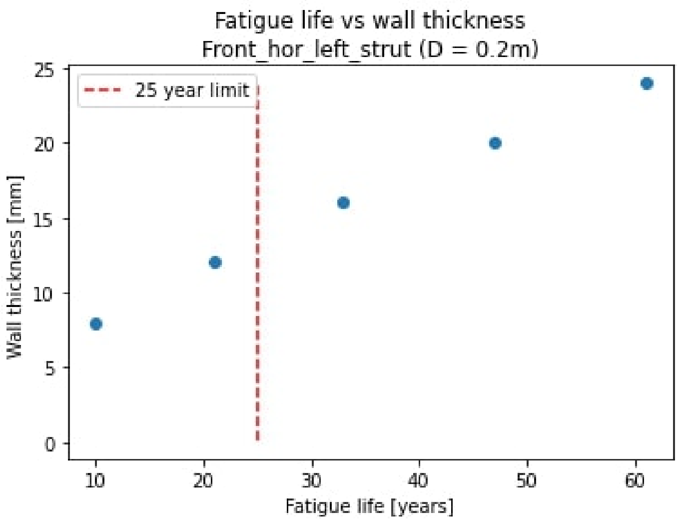

Figure E.22

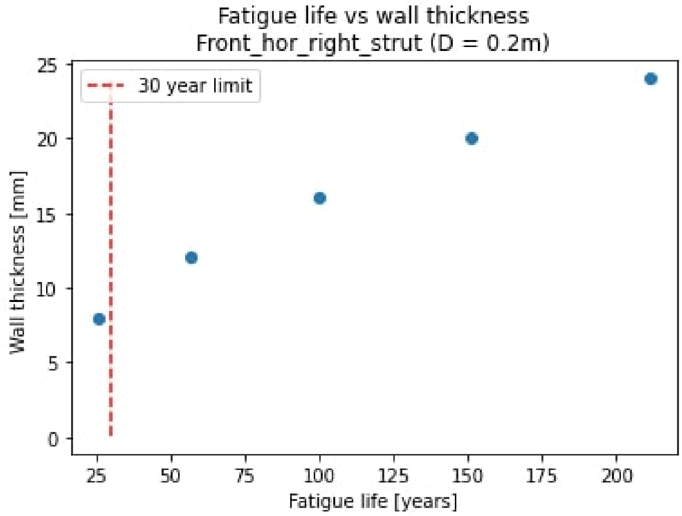

Figure E.23

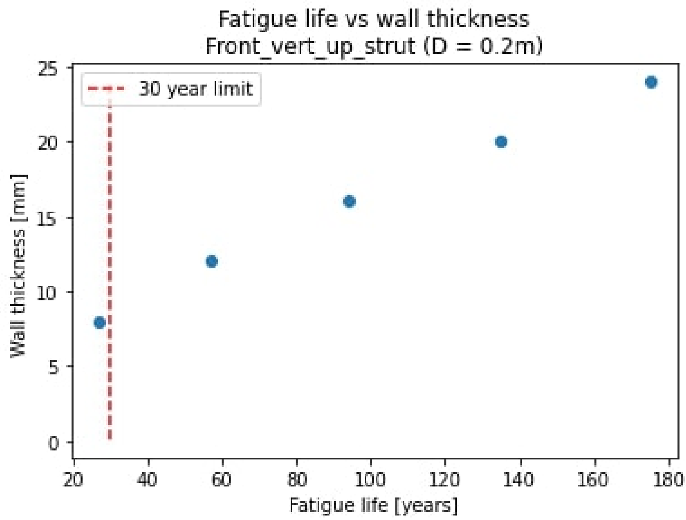

Figure E.24

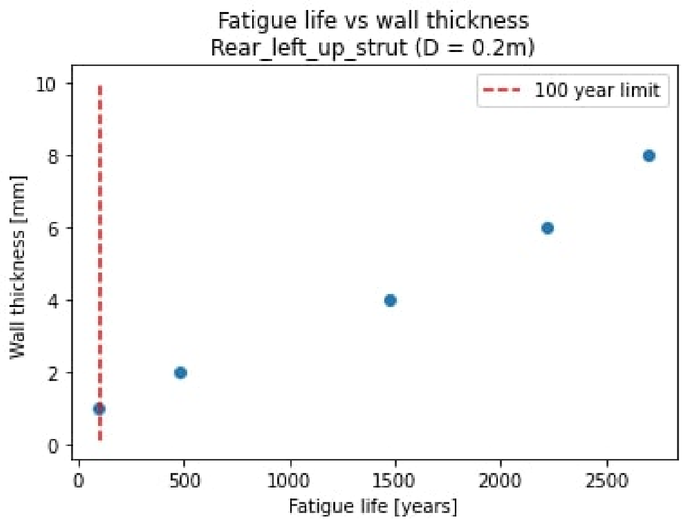

Figure E.25

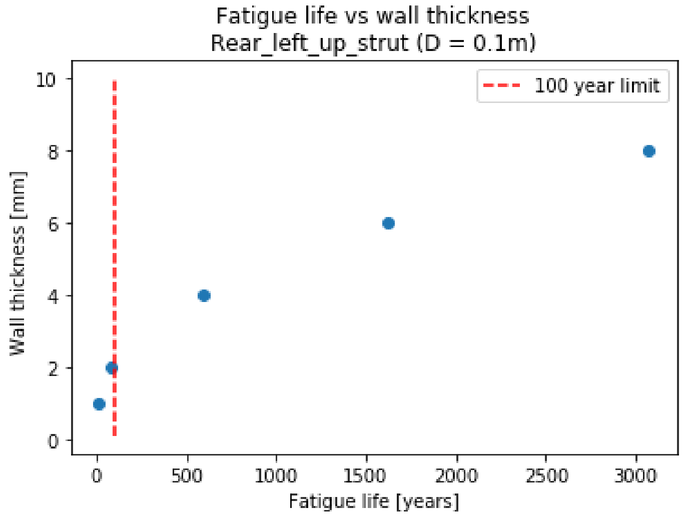

Figure E.26

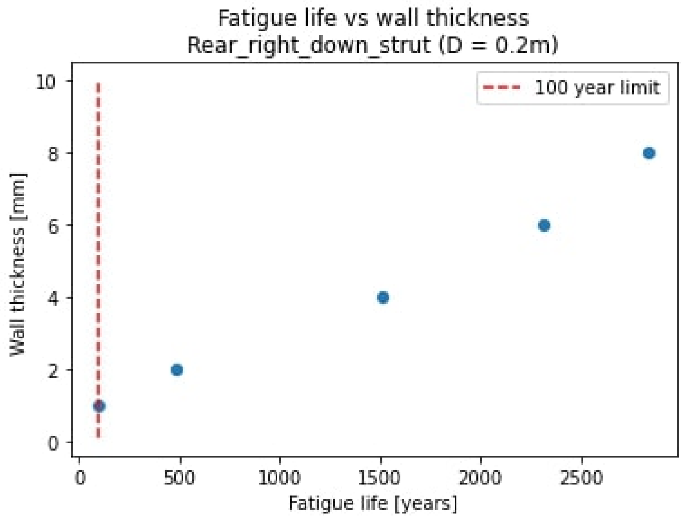

Figure E.27

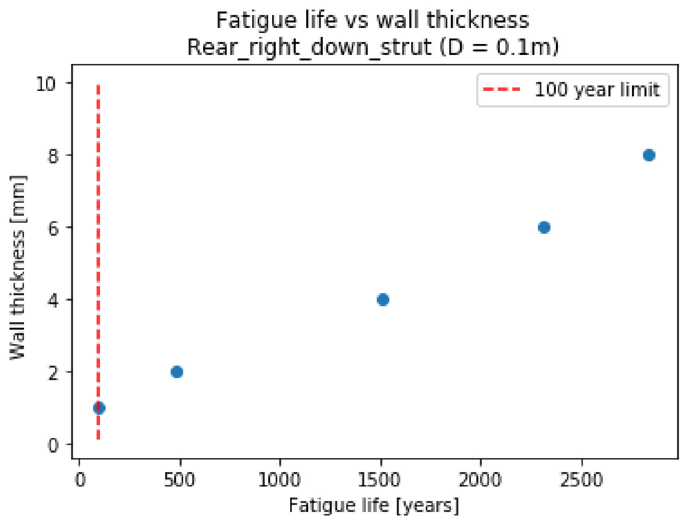

Figure E.28

F. Stress histories from Fatigue Life Assessment

2-Bladed Structure **4-Bladed Structure**

Front hor left strut 1

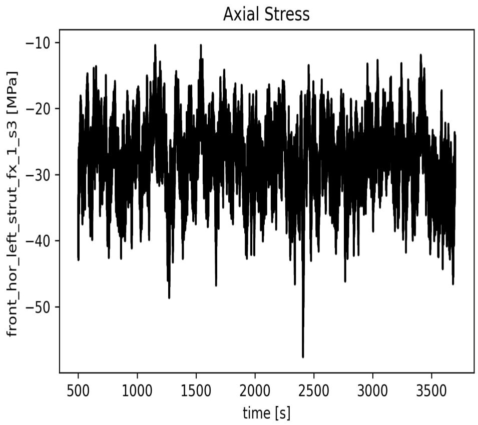

Figure F.1

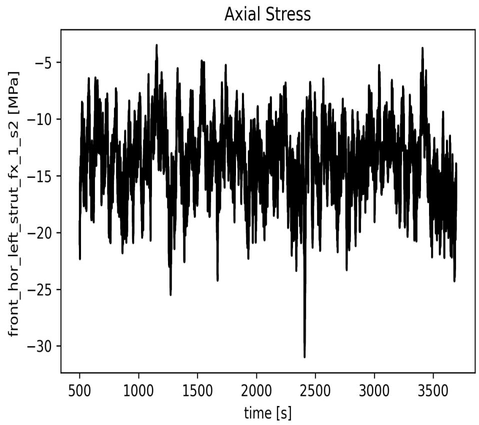

Figure F.2

Front hor left strut 2

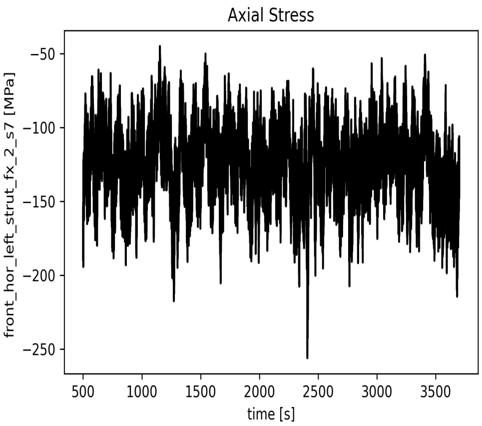

Figure F.3

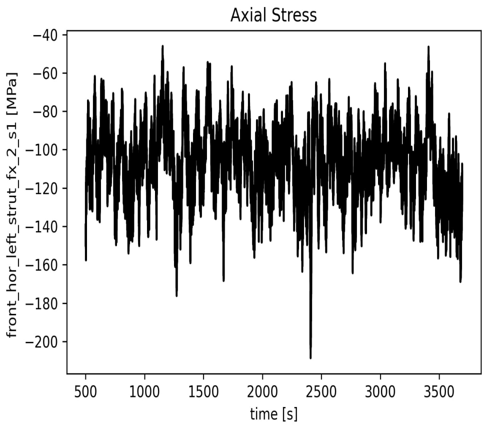

Figure F.4

Front hor right strut 1

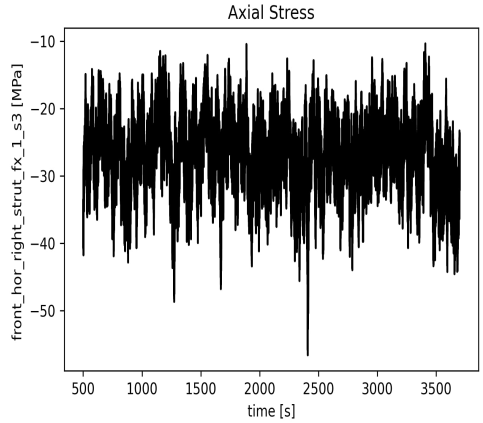

Figure F.5

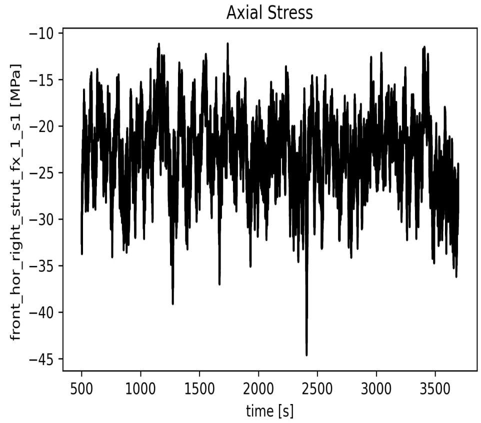

Figure F.6

Front hor right strut 2

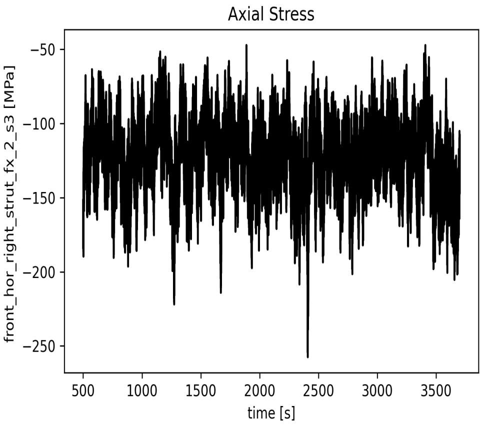

Figure F.7

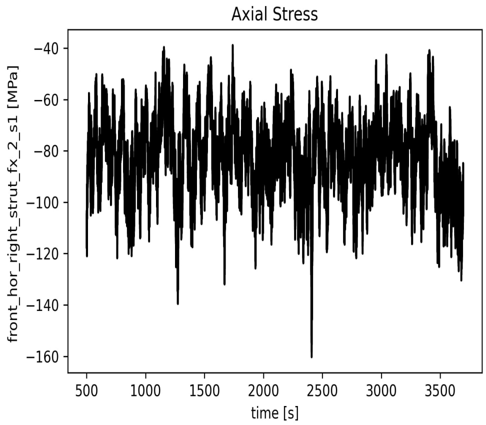

Figure F.8

Front vert down strut 1

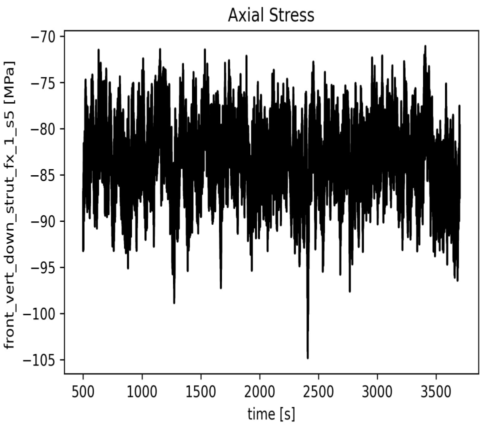

Figure F.9

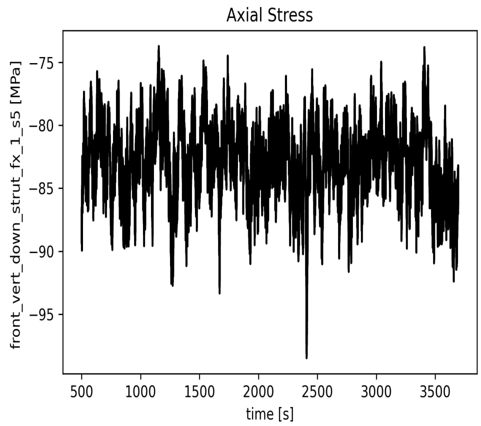

Figure F.10

Front vert down strut 2

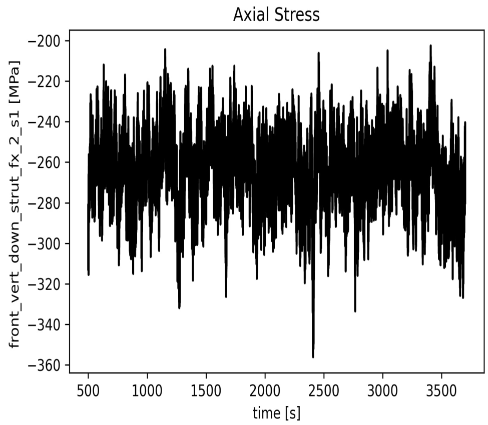

Figure F.11

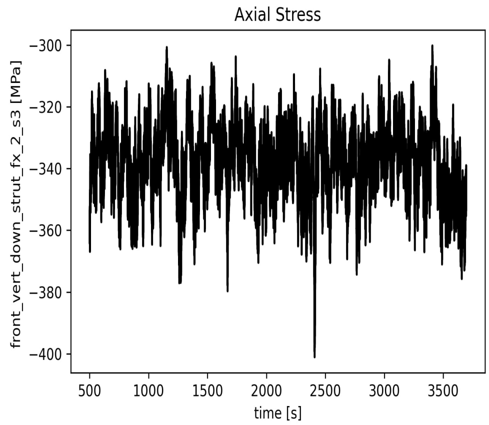

Figure F.12

Front vert up strut 1

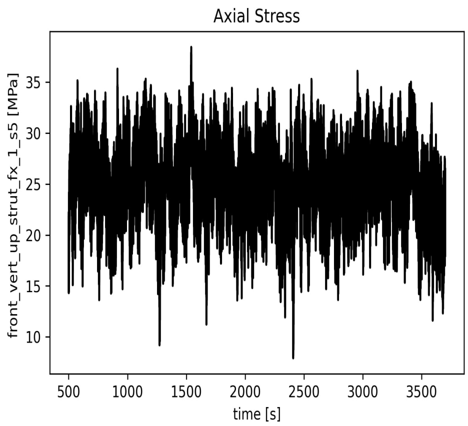

Figure F.13

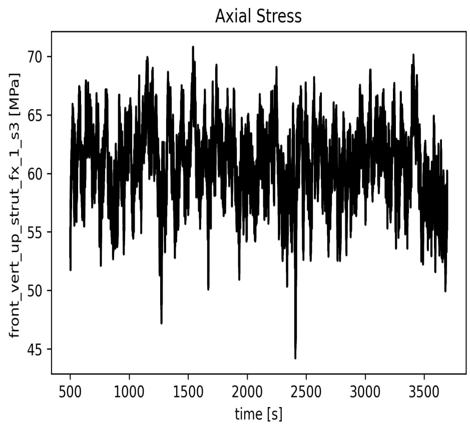

Figure F.14

Front vert up strut 2

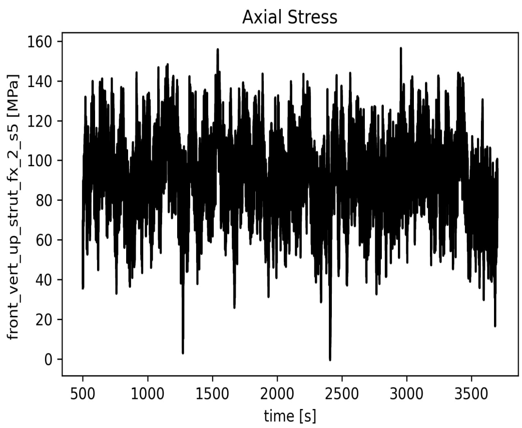

Figure F.15

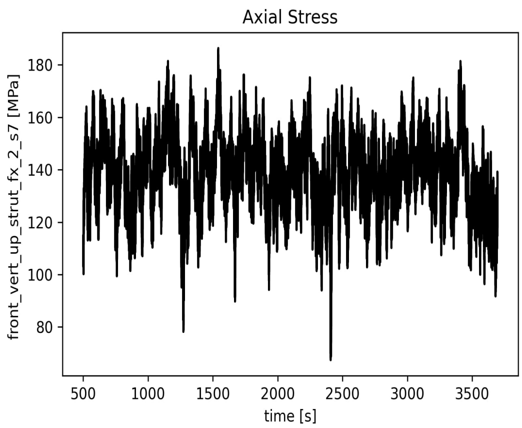

Figure F.16

Rear left down strut 1

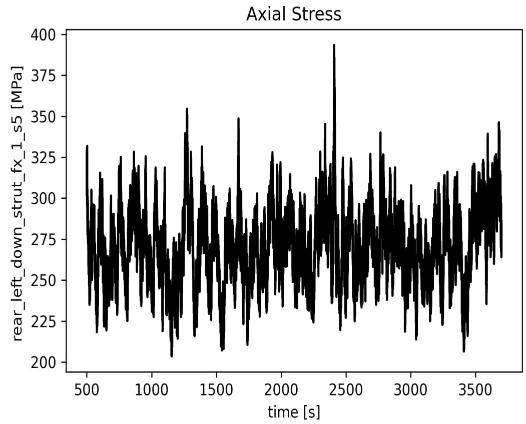

Figure F.17

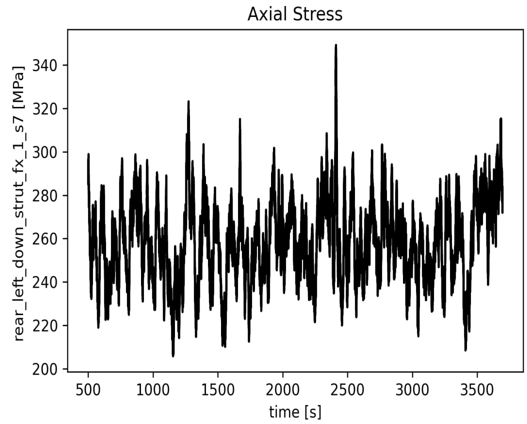

Figure F.18

Rear left down strut 2

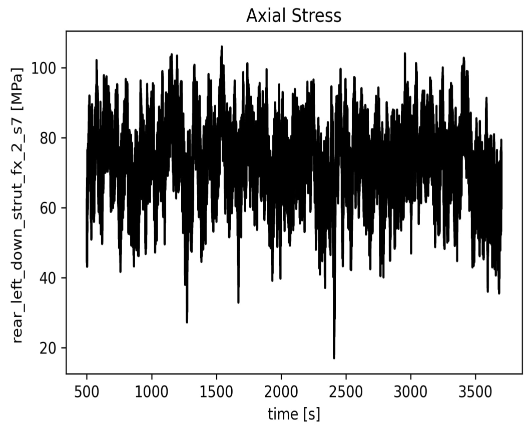

Figure F.19

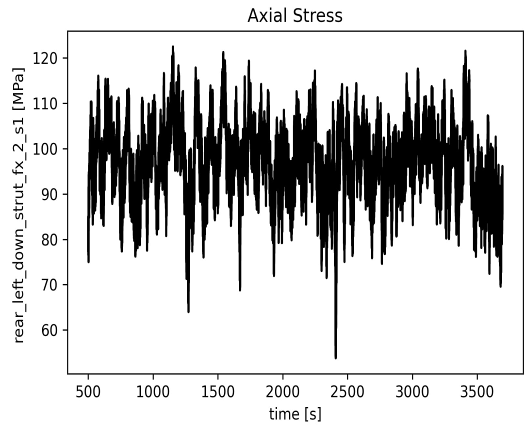

Figure F.20

Rear left up strut 1

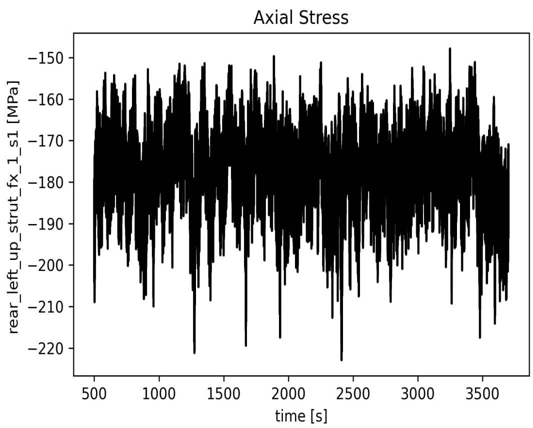

Figure F.21

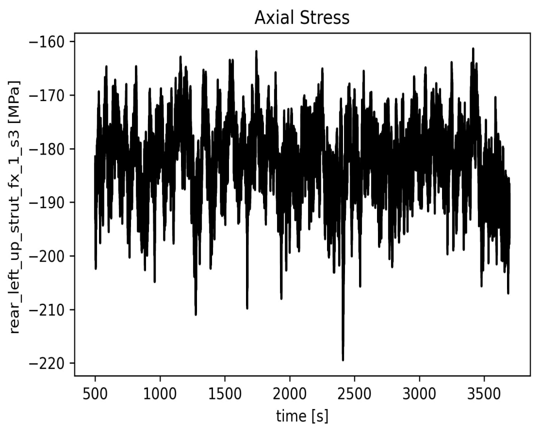

Figure F.22

Rear left up strut 2

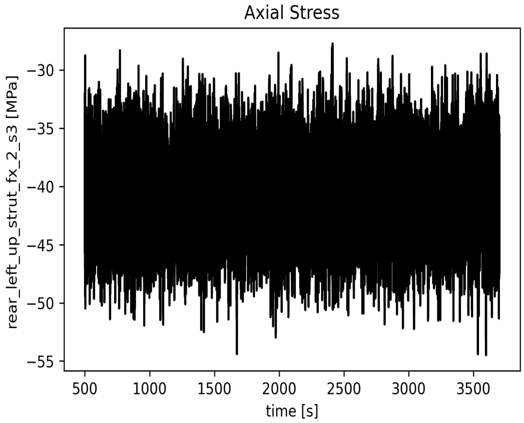

Figure F.23

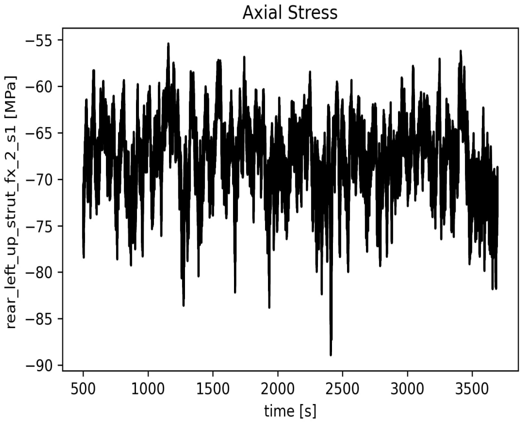

Figure F.24

Rear right down strut 1

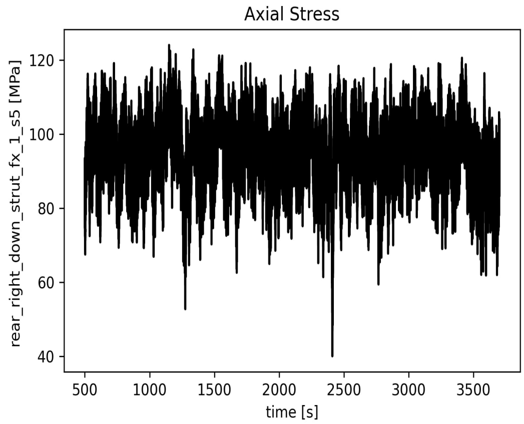

Figure F.25

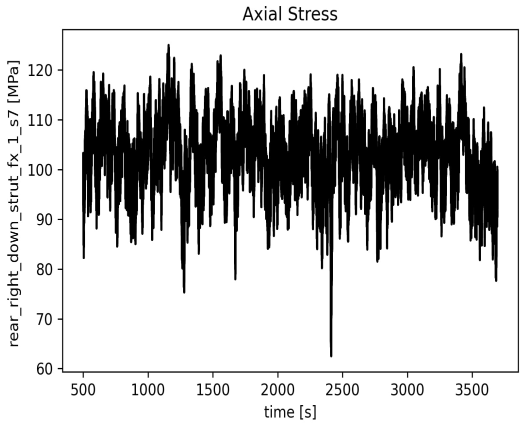

Figure F.26

Rear right down strut 2

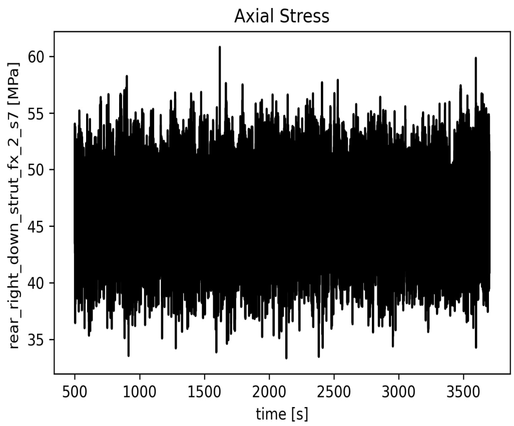

Figure F.27

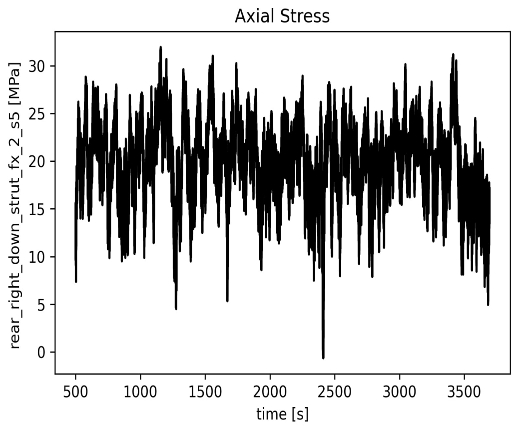

Figure F.28

Rear right up 1

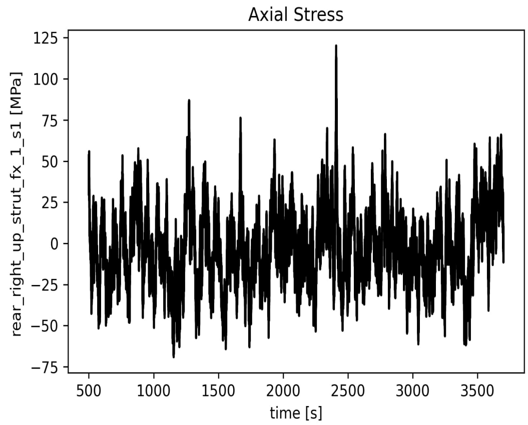

Figure F.29

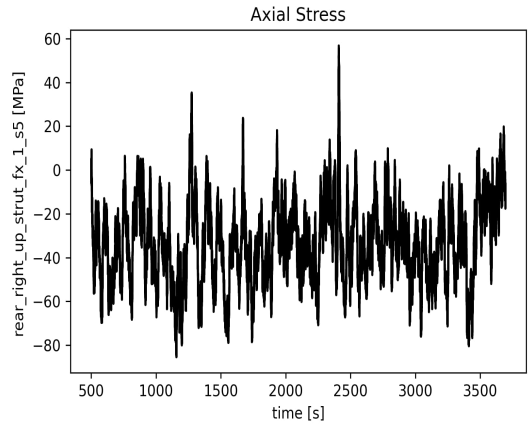

Figure F.30

Rear right up 2

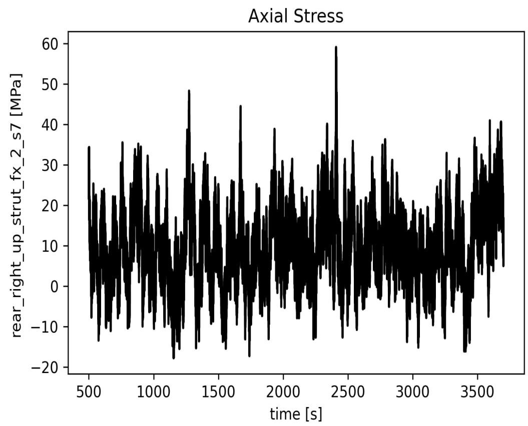

Figure F.31

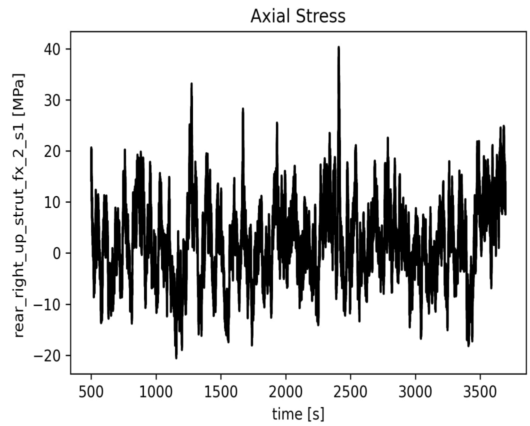

Figure F.32

Rear vert down strut

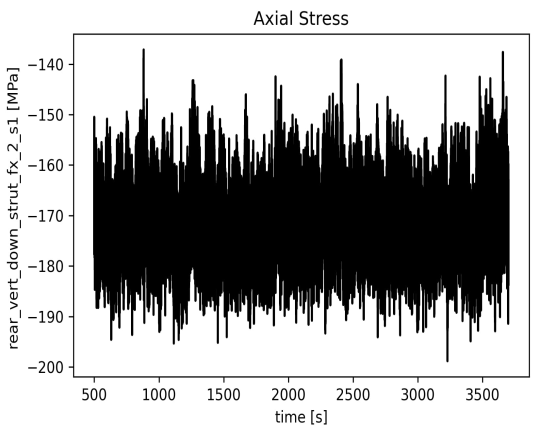

Figure F.33

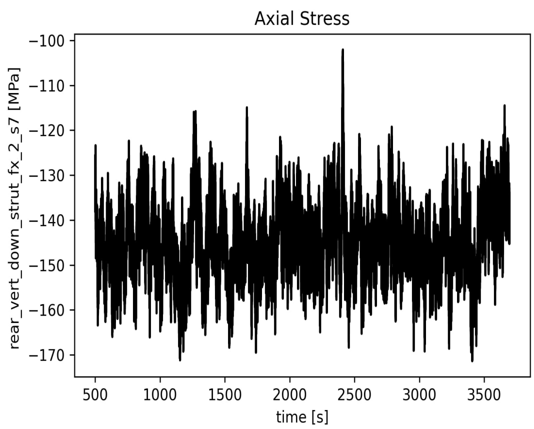

Figure F.34

Rear vert up strut

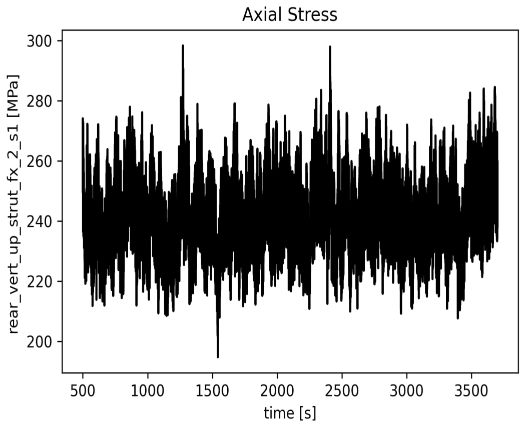

Figure F.35

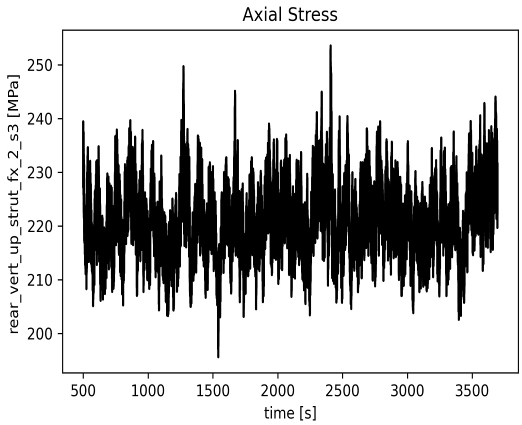

Figure F.36

Shaft

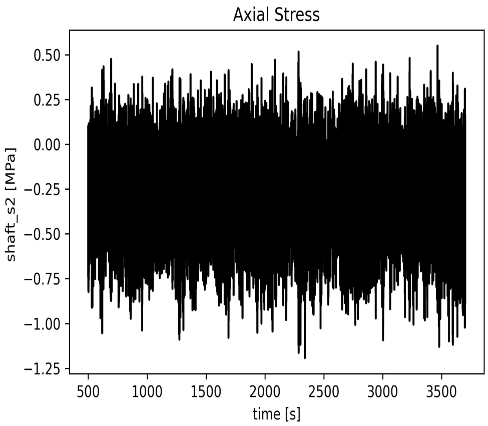

Figure F.37

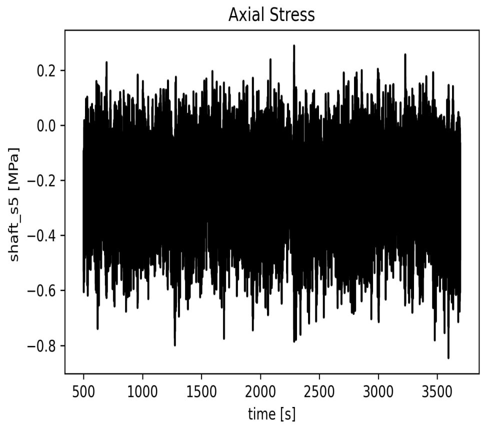

Figure F.38

Blade root

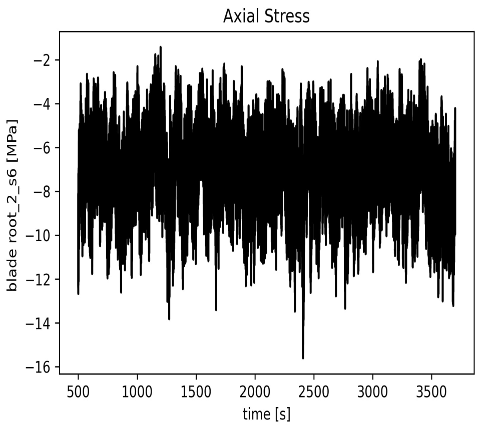

Figure F.39

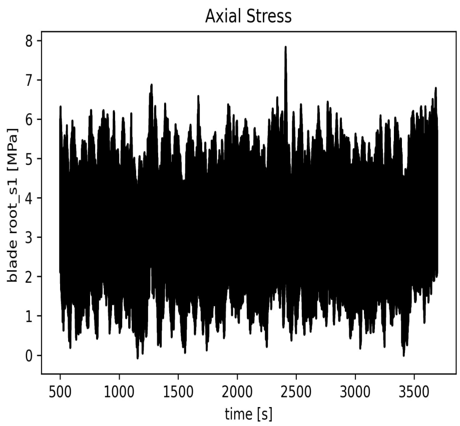

Figure F.40

Rotor

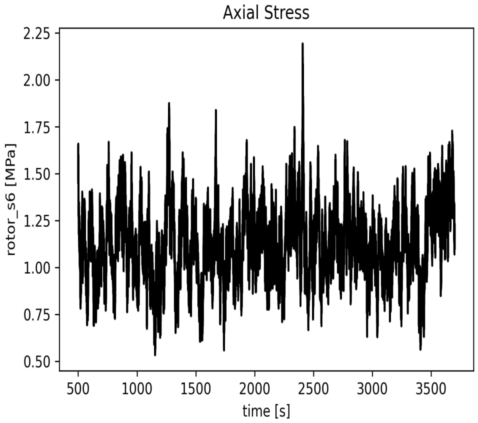

Figure F.41

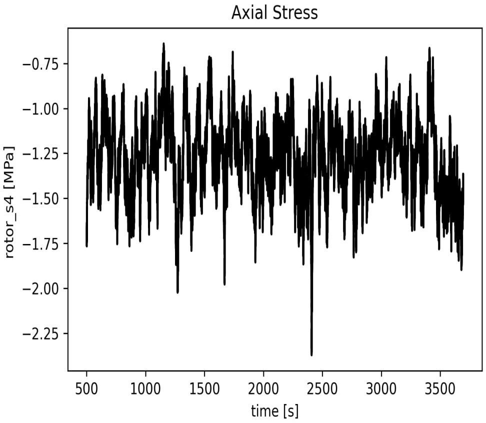

Figure F.42

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

```
# -*- coding: utf-8 -*-
"""
Created on Thu Apr 29 13:19:16 2021

@author: Vetle Birkeland Aass
"""

import numpy as np
from mik_fatigue_funcs import turningpoints,turningpoints_steffen, fatiguedamage_twoslope
from atte_punkter import cyl_beam_stresses


from pylab import *
from numpy import loadtxt
from scipy import interpolate


## STANDARD INPUT for tubular joints
# SN-curve T "air"
th = 32E-3 # plate thickness base material (m)
## SN-curve parameters from DNV-GL-C203:
m1 = 3.0 # slope 1
loga1 = 12.479 # intercept 1
m2 = 5.0 # slope high cycle region
loga2 = 16.132 # intercept high-cycle region
Nlim = 1.0E7 # limit for high-cycle region
tref=32E-3 # reference thickness (32E-3 for tubular joints)
k=0.25 # thickness exponent
DFF = 1 #Design Fatigue Factor

## STANDARD INPUT for unwelded tubular member
th_st = 25E-3 # plate thickness base material (m)
## SN-curve parameters from DNV-GL-C203:
m1_st = 4.0 # slope 1
```

```
loga1_st = 15.117 # intercept 1
m2_st = 5.0 # slope high cycle region
loga2_st = 17.146 # intercept high-cycle region
Nlim_st = 1.0E7 # limit for high-cycle region
tref= 25E-3 # reference thickness (25E-3 for tubular joints)
k_st=0.25 # thickness exponent
DFF_st = 1 #Design Fatigue Factor



#struts
d     = 0.2
twall = 0.008

#shaft
D_shaft = 2
T_shaft = 0.002

#Blade root
D = 1.2 # element midpoint
TWALL = 0.0028

#Rotor
D_Rotor = 1 #[m]
Twall_Rotor = 0.008 #[m]


#importing data
a    = loadtxt('sensors.txt',skiprows=50000,dtype='float',delimiter=';')  #
    skip 500 s
t        = a[:,0];


#Joint types and their corresponding stress concentration factors

#Type 1 joint
SCF_AC_1 = 3.73          #stress concentration factor for joint type 1,
    brace crown
SCF_AS_1 = 1.99          #stress concentration factor for joint type 1,
    brace saddle
SCF_IPB_1 = 1.36         #stress concentration factor for joint type 1, in
    plane bending
SCF_OPB_1 = 1          #stress concentration factor for joint type 1, out
    of plane bending

#Type 2 joint
SCF_AC_2 = 5.73      #stress concentration factor for joint type 2, brace
    crown
SCF_AS_2 = 15.21     #stress concentration factor for joint type 2, brace
```

```
saddle
SCF_IPB_2 = 3.93      #stress concentration factor for joint type 2, in plane bending
SCF_OPB_2 = 5.90      #stress concentration factor for joint type 2, out of plane bending

#Type 3 joint
SCF_AC_3 = 26.14          #stress concentration factor for joint type 3, brace crown
SCF_AS_3 = 4.27          #stress concentration factor for joint type 3, brace saddle
SCF_IPB_3 = 2.46          #stress concentration factor for joint type 3, in plane bending
SCF_OPB_3 = 4.27          #stress concentration factor for joint type 3, out of plane bending


#Type 4 joint
SCF_AC_4 = 7.72       #stress concentration factor for joint type 4, brace crown
SCF_AS_4 = 7.72       #stress concentration factor for joint type 4, brace saddle
SCF_IPB_4 = 3.91      #stress concentration factor for joint type 4, in plane bending
SCF_OPB_4 = 6.03      #stress concentration factor for joint type 4, out of plane bending


#Type 5 joint




#----------------------------------------------------------------------------------------

#front_hor_left_strut_x1
#Joint type 1
#saddle in z-dir
front_hor_left_strut_fx_1 = a[:,22]*1.e3;
front_hor_left_strut_fy_1 = a[:,102]*1.e3;
front_hor_left_strut_fz_1 = a[:,103]*1.e3;
mx_fhls_1 = a[:,23]*1.e3;
my_fhls_1 = a[:,24]*1.e3;
mz_fhls_1 = a[:,25]*1.e3;

[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] = \
```

```
    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)

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_1_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_1_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_hor_left_strut_fx_1_s2 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_1_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_1_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_hor_left_strut_fx_1_s3 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_1_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_1_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_hor_left_strut_fx_1_s4 calculated
    by script: %.5e (includes residual cycles)' %damage)
```

```
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_1_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_1_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for front_hor_left_strut_fx_1_s5 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_1_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_1_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for front_hor_left_strut_fx_1_s6 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_1_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_1_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for front_hor_left_strut_fx_1_s7 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
```

```
    damage*60*60*24*365)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_1_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_1_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_hor_left_strut_fx_1_s8 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_1_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_1_s8',dpi=300)

#
    ------------------------------------------------------------------------

#front_hor_left_strut_x2
#Joint type 2
#crown in z-dir
front_hor_left_strut_fx_2 = a[:,26]*1.e3;
front_hor_left_strut_fy_2 = a[:,104]*1.e3;
front_hor_left_strut_fz_2 = a[:,105]*1.e3;
mx_fhls_2 = a[:,27]*1.e3;
my_fhls_2 = a[:,28]*1.e3;
mz_fhls_2 = a[:,29]*1.e3;

[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] = \
     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)

damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
```

```
print('Partial fatigue damage front_hor_left_strut_fx_2_s1 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(damage*60*60*24*DFF)))
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_2_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_2_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for front_hor_left_strut_fx_2_s2 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_2_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_2_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for front_hor_left_strut_fx_2_s3 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_2_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
```

```
k=0.25)
print('Partial fatigue damage for front_hor_left_strut_fx_2_s4 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_2_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for front_hor_left_strut_fx_2_s5 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_2_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for front_hor_left_strut_fx_2_s6 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_2_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_2_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
```

```
print('Partial fatigue damage for front_hor_left_strut_fx_2_s7 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_2_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_2_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for front_hor_left_strut_fx_2_s8 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_2_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_2_s8',dpi=300)

#
----------------------------------------------------------------------

#front_hor_right_strut_x1
#Joint type 1
#saddle in z-dir
front_hor_right_strut_fx_1 = a[:,30]*1.e3;
front_hor_right_strut_fy_1 = a[:,106]*1.e3;
front_hor_right_strut_fz_1 = a[:,107]*1.e3;
mx_fhrs_1 = a[:,31]*1.e3;
my_fhrs_1 = a[:,32]*1.e3;
mz_fhrs_1 = a[:,33]*1.e3;


[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] = \
    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)

damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_hor_right_strut_fx_1_s1 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_1_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_1_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_hor_right_strut_fx_1_s2 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_1_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_1_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_hor_right_strut_fx_1_s3 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_1_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_1_s3',dpi=300)
```

```
damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_hor_right_strut_fx_1_s4 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_1_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_1_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_hor_right_strut_fx_1_s5 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_1_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_1_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_hor_right_strut_fx_1_s6 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_1_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_1_s6',dpi=300)

```

```
damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_hor_right_strut_fx_1_s7 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('Sfront_hor_right_strut_fx_1_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_1_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_hor_right_strut_fx_1_s8 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_1_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_1_s8',dpi=300)

#
    ______________________________________________________________________________



#front_hor_right_strut_x2
#Joint type 2
#crown in z-dir
front_hor_right_strut_fx_2 = a[:,34]*1.e3;
front_hor_right_strut_fy_2 = a[:,108]*1.e3;
front_hor_right_strut_fz_2 = a[:,109]*1.e3;
mx_fhrs_2 = a[:,35]*1.e3;
my_fhrs_2 = a[:,36]*1.e3;
mz_fhrs_2 = a[:,37]*1.e3;

[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] = \
```

```
    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)

damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_hor_right_strut_fx_2_s1 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_2_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_2_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_hor_right_strut_fx_2_s2 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_2_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_2_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_hor_right_strut_fx_2_s3 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
```

```
#legend(loc='best')
savefig('front_hor_right_strut_fx_2_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_hor_right_strut_fx_2_s4 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_2_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_hor_right_strut_fx_2_s5 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_2_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_hor_right_strut_fx_2_s6 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_2_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
```

```
savefig('front_hor_right_strut_fx_2_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_hor_right_strut_fx_2_s7 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_2_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_2_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_hor_right_strut_fx_2_s8 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_2_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_2_s8',dpi=300)

#
    ------------------------------------------------------------------------------



#front_vert_down_strut_x1
#Joint type 1
#crown in z-dir
front_vert_down_strut_fx_1 = a[:,38]*1.e3;
front_vert_down_strut_fy_1 = a[:,110]*1.e3;
front_vert_down_strut_fz_1 = a[:,111]*1.e3;
mx_fvds_1 = a[:,39]*1.e3;
my_fvds_1 = a[:,40]*1.e3;
mz_fvds_1 = a[:,41]*1.e3;

```

```
[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] = \
    cyl_beam_stresses(t,front_vert_down_strut_fx_1,front_vert_down_strut_fy_1,front_vert_down_strut_fy_1,mx_fvds_1,my_fvds_1,mz_fvds_1,d,twall,SCF_AC_1,SCF_AS_1,SCF_IPB_1,SCF_OPB_1)

damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for front_vert_down_strut_fx_1_s1 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(damage*60*60*24*DFF)))
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_1_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_1_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for front_vert_down_strut_fx_2_s2 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_1_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_1_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for front_vert_down_strut_fx_2_s3 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s3/1.e6,'k-',label='')
```

```
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_1_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_1_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_down_strut_fx_2_s4 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_1_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_1_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_down_strut_fx_2_s5 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_1_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_1_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_down_strut_fx_2_s6 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
```

```
ylabel('front_vert_down_strut_fx_1_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_1_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3, k=0.25)
print('Partial fatigue damage for front_vert_down_strut_fx_2_s7 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_1_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_1_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3, k=0.25)
print('Partial fatigue damage for front_vert_down_strut_fx_2_s8 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_1_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_1_s8',dpi=300)

#
    ________________________________________________________________

#front_vert_down_strut_x2
#Joint type 2
#saddle in z-dir
front_vert_down_strut_fx_2 = a[:,42]*1.e3;
front_vert_down_strut_fy_2 = a[:,112]*1.e3;
front_vert_down_strut_fz_2 = a[:,113]*1.e3;
mx_fvds_2 = a[:,43]*1.e3;
```

```
my_fvds_2 = a[:,44]*1.e3;
mz_fvds_2 = a[:,45]*1.e3;

[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] = \
     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)

damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_down_strut_fx_2_s1 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFF)))
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_2_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_2_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_down_strut_fx_2_s2 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_2_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_2_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_down_strut_fx_2_s3 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))
```

```
figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_2_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_down_strut_fx_2_s4 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_2_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_down_strut_fx_2_s5 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_2_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_down_strut_fx_2_s6 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

```

```
figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_2_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_2_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_down_strut_fx_2_s7 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_2_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_2_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_down_strut_fx_2_s8 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_2_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_2_s8',dpi=300)
#
    ------------------------------------------------------------------------------------------------

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

```
front_vert_up_strut_fz_1 = a[:,115]*1.e3;
mx_fvus_1 = a[:,47]*1.e3;
my_fvus_1 = a[:,48]*1.e3;
mz_fvus_1 = a[:,49]*1.e3;

[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] = \
    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)

damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for front_vert_up_strut_fx_1_s1 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(damage*60*60*24*DFF)))
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_1_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_1_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for front_vert_up_strut_fx_1_s2 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_1_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_1_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for front_vert_up_strut_fx_1_s3 calculated by script: %.5e (includes residual cycles)' %damage)
```

```
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_1_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_1_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_up_strut_fx_1_s4 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_1_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_1_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_up_strut_fx_1_s5 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_1_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_1_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_up_strut_fx_1_s6 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
```

```
    damage*60*60*24*365*DFF)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_1_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_1_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_up_strut_fx_1_s7 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_1_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_1_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_up_strut_fx_1_s8 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_1_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_1_s8',dpi=300)



#
    ------------------------------------------------------------------------------

#front_vert_up_strut_x2
```

```
#Joint type 2
#saddle in z-dir
front_vert_up_strut_fx_2 = a[:,50]*1.e3;
front_vert_up_strut_fy_2 = a[:,116]*1.e3;
front_vert_up_strut_fz_2 = a[:,117]*1.e3;
mx_fvus_2 = a[:,51]*1.e3;
my_fvus_2 = a[:,52]*1.e3;
mz_fvus_2 = a[:,53]*1.e3;

[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] = \
     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)

damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage front_vert_up_strut_fx_2_s1 calculated by
    script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFF)))
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_2_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_2_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_up_strut_fx_2_s2 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_2_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_2_s2',dpi=300)

```

```
damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_up_strut_fx_2_s3 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_2_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_up_strut_fx_2_s4 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_2_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_up_strut_fx_2_s5 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_2_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
```

```
    k=0.25)
print('Partial fatigue damage for front_vert_up_strut_fx_2_s6 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_2_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_2_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_up_strut_fx_2_s7 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_2_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_2_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for front_vert_up_strut_fx_2_s8 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_2_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_2_s8',dpi=300)


#
```

```
#rear_left_down_strut_x1
#Joint type 3
#crown in z-dir
rear_left_down_strut_fx_1 = a[:,54]*1.e3;
rear_left_down_strut_fy_1 = a[:,118]*1.e3;
rear_left_down_strut_fz_1 = a[:,119]*1.e3;
mx_rlds_1 = a[:,55]*1.e3;
my_rlds_1 = a[:,56]*1.e3;
mz_rlds_1 = a[:,57]*1.e3;


[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] = \
     cyl_beam_stresses(t,rear_left_down_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_3,SCF_AS_3,SCF_IPB_3,SCF_OPB_3)

damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_1_s1 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_1_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_1_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_1_s2 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_1_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
```

```
#legend(loc='best')
savefig('rear_left_down_strut_fx_1_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_1_s3 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_1_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_1_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_1_s4 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_1_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_1_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_1_s5 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_1_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
```

```
savefig('rear_left_down_strut_fx_1_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_1_s6 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_1_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_1_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_1_s7 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_1_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_1_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_1_s8 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_1_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_1_s8',dpi=300)
```

```

#



#rear_left_down_strut_x2
#Joint type 4
#crown in z-dir
rear_left_down_strut_fx_2 = a[:,58]*1.e3;
rear_left_down_strut_fy_2 = a[:,120]*1.e3;
rear_left_down_strut_fz_2 = a[:,121]*1.e3;
mx_rlds_2 = a[:,59]*1.e3;
my_rlds_2 = a[:,60]*1.e3;
mz_rlds_2 = a[:,61]*1.e3;

[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] = \
     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)

damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_2_s1 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFF)))
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_2_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_2_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_2_s2 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s2/1.e6,'k-',label='')
```

```
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_2_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_2_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_2_s3 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_2_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_2_s4 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_2_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_2_s5 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
```

```
ylabel('rear_left_down_strut_fx_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_2_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3, k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_2_s6 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_2_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_2_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3, k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_2_s7 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_2_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_2_s7',dpi=300)

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

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_2_s8 [MPa] ')
```

```
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_2_s8',dpi=300)



# ------------------------------------------------------------------------------------------------


#rear_left_up_strut_x1
#Joint type 3
#crown in z-dir
rear_left_up_strut_fx_1 = a[:,62]*1.e3;
rear_left_up_strut_fy_1 = a[:,122]*1.e3;
rear_left_up_strut_fz_1 = a[:,123]*1.e3;
mx_rlus_1 = a[:,63]*1.e3;
my_rlus_1 = a[:,64]*1.e3;
mz_rlus_1 = a[:,65]*1.e3;

[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] = \
     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)

damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_up_strut_fx_1_s1 calculated by
     script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFF)))
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_1_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_1_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_up_strut_fx_1_s2 calculated by
```

```
    script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_1_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_1_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_up_strut_fx_1_s3 calculated by
    script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_1_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_1_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_up_strut_fx_1_s4 calculated by
    script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_1_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_1_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_up_strut_fx_1_s5 calculated by
    script: %.5e (includes residual cycles)' %damage)
```

```
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_1_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_1_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_up_strut_fx_1_s6 calculated by
     script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_1_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_1_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_up_strut_fx_1_s7 calculated by
     script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_1_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_1_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_up_strut_fx_1_s8 calculated by
     script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
```

```
damage*60*60*24*365*DFF)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_1_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_1_s8',dpi=300)




#
    ----------------------------------------------------------------------------------------------------

#rear_left_up_strut_x2
#Joint type 4
#crown in z-dir
rear_left_up_strut_fx_2 = a[:,66]*1.e3;
rear_left_up_strut_fy_2 = a[:,124]*1.e3;
rear_left_up_strut_fz_2 = a[:,125]*1.e3;
mx_rlus_2 = a[:,67]*1.e3;
my_rlus_2 = a[:,68]*1.e3;
mz_rlus_2 = a[:,69]*1.e3;

[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] = \
     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)

damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_2_s1 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFF)))
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_2_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
```

```
#legend(loc='best')
savefig('rear_left_up_strut_fx_2_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_2_s2 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_2_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_2_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_2_s3 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_2_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_2_s4 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
```

```
savefig('rear_left_up_strut_fx_2_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_2_s5 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_2_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_2_s6 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_2_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_2_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_left_down_strut_fx_2_s7 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_2_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_2_s7',dpi=300)
```

```python

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

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_2_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_2_s8',dpi=300)

#
    ------------------------------------------------------------------------------------------------

#rear_right_down_strut_x1
#Joint type 3
#crown in z-dir
rear_right_down_strut_fx_1 = a[:,70]*1.e3;
rear_right_down_strut_fy_1 = a[:,126]*1.e3;
rear_right_down_strut_fz_1 = a[:,127]*1.e3;
mx_rrds_1 = a[:,71]*1.e3;
my_rrds_1 = a[:,72]*1.e3;
mz_rrds_1 = a[:,73]*1.e3;

[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] = \
     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)

damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_down_strut_fx_1_s1 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFF)))
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

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

```
ylabel('rear_right_down_strut_fx_1_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_1_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_down_strut_fx_1_s2 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_1_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_1_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_down_strut_fx_1_s3 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_1_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_1_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_down_strut_fx_1_s4 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_1_s4 [MPa] ')
```

```
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_1_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for rear_right_down_strut_fx_1_s5 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_1_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_1_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for rear_right_down_strut_fx_1_s6 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_1_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_1_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for rear_right_down_strut_fx_1_s7 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_1_s7 [MPa] ')
title('Axial Stress')
```

```
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_1_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_down_strut_fx_1_s8 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_1_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_1_s8',dpi=300)

#
    ______________________________________________________________________________


#rear_right_down_strut_x2
#Joint type 4
#crown in z-dir
rear_right_down_strut_fx_2 = a[:,74]*1.e3;
rear_right_down_strut_fy_2 = a[:,128]*1.e3;
rear_right_down_strut_fz_2 = a[:,129]*1.e3;
mx_rrds_2 = a[:,75]*1.e3;
my_rrds_2 = a[:,76]*1.e3;
mz_rrds_2 = a[:,77]*1.e3;

[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] = \
     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)

damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_down_strut_fx_2_s1 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFF)))
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

```

```
figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_2_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_2_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_down_strut_fx_2_s2 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_2_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_2_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_down_strut_fx_2_s3 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_2_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_down_strut_fx_2_s4 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
```

```
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_2_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for rear_right_down_strut_fx_2_s5 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_2_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for rear_right_down_strut_fx_2_s6 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_2_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_2_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for rear_right_down_strut_fx_2_s7 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s7/1.e6,'k-',label='')
```

```
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_2_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_2_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_down_strut_fx_2_s8 calculated
     by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_2_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_2_s8',dpi=300)

#
    ------------------------------------------------------------------------------

#rear_right_up_strut_x1
#Joint type 3
#crown in z-dir
rear_right_up_strut_fx_1 = a[:,78]*1.e3;
rear_right_up_strut_fy_1 = a[:,130]*1.e3;
rear_right_up_strut_fz_1 = a[:,131]*1.e3;
mx_rrus_1 = a[:,79]*1.e3;
my_rrus_1 = a[:,80]*1.e3;
mz_rrus_1 = a[:,81]*1.e3;

[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] = \
     cyl_beam_stresses(t,rear_right_up_strut_fx_1,rear_right_up_strut_fy_1,
         rear_right_up_strut_fy_1,mx_rrus_1,my_rrus_1,mz_rrus_1,d,twall,
         SCF_AC_3,SCF_AS_3,SCF_IPB_3,SCF_OPB_3)

damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_up_strut_fx_1_s1 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFF)))
```

```
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_1_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_1_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_up_strut_fx_1_s2 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_1_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_1_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_up_strut_fx_1_s3 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_1_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_1_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_up_strut_fx_1_s4 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
```

```
    damage*60*60*24*365*DFF)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_1_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_1_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_up_strut_fx_1_s5 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_1_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_1_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_up_strut_fx_1_s6 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_1_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_1_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_up_strut_fx_1_s7 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))
```

```

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_1_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_1_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_up_strut_fx_1_s8 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_1_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_1_s8',dpi=300)

#
    ------------------------------------------------------------------------------


#rear_right_up_strut_x2
#Joint type 4
#crown in z-dir
rear_right_up_strut_fx_2 = a[:,82]*1.e3;
rear_right_up_strut_fy_2 = a[:,132]*1.e3;
rear_right_up_strut_fz_2 = a[:,133]*1.e3;
mx_rrus_2 = a[:,83]*1.e3;
my_rrus_2 = a[:,84]*1.e3;
mz_rrus_2 = a[:,85]*1.e3;

[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] = \
     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)

damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_up_strut_fx_2_s1 calculated
```

```
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFF)))
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_2_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_2_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_up_strut_fx_2_s2 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_2_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_2_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_up_strut_fx_2_s3 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_2_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
```

```
print('Partial fatigue damage for rear_right_up_strut_fx_2_s4 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_2_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for rear_right_up_strut_fx_2_s5 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_2_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for rear_right_up_strut_fx_2_s6 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_2_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_2_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for rear_right_up_strut_fx_2_s7 calculated
```

```
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_2_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_2_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_right_up_strut_fx_2_s8 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_2_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_2_s8',dpi=300)

#
    ------------------------------------------------------------------------------------------


#EXLUDE?

#rear_vert_down_strut_x1

#N/A
rear_vert_down_strut_fx_1 = a[:,86]*1.e3;
rear_vert_down_strut_fy_1 = a[:,134]*1.e3;
rear_vert_down_strut_fz_1 = a[:,135]*1.e3;
mx_rvds_1 = a[:,87]*1.e3;
my_rvds_1 = a[:,88]*1.e3;
mz_rvds_1 = a[:,89]*1.e3;

[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] = \
     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,
```

```
        my_rvds_1,mz_rvds_1,d,twall,1,1,1,1)

damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for rear_vert_down_strut_fx_1_s1 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(damage*60*60*24*DFF)))
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_1_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_1_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for rear_vert_down_strut_fx_1_s2 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_1_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_1_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for rear_vert_down_strut_fx_1_s3 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_1_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
```

```
#legend(loc='best')
savefig('rear_vert_down_strut_fx_1_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_down_strut_fx_1_s4 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_1_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_1_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_down_strut_fx_1_s5 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_1_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_1_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_down_strut_fx_1_s6 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_1_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
```

```
savefig('rear_vert_down_strut_fx_1_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_down_strut_fx_1_s7 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_1_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_1_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_down_strut_fx_1_s8 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_1_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_1_s8',dpi=300)

#
    ________________________________________________________________________


#rear_vert_down_strut_x2
#Joint type 2
#saddle in z-dir
rear_vert_down_strut_fx_2 = a[:,90]*1.e3;
rear_vert_down_strut_fy_2 = a[:,136]*1.e3;
rear_vert_down_strut_fz_2 = a[:,137]*1.e3;
mx_rvds_2 = a[:,91]*1.e3;
my_rvds_2 = a[:,92]*1.e3;
mz_rvds_2 = a[:,93]*1.e3;

[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] = \
     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)

damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_down_strut_fx_2_s1 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFF)))
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_2_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_2_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_down_strut_fx_2_s2 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_2_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_2_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_down_strut_fx_2_s3 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
```

```
ylabel('rear_vert_down_strut_fx_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_2_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_down_strut_fx_2_s4 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_2_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_down_strut_fx_2_s5 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_2_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_down_strut_fx_2_s6 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_2_s6 [MPa] ')
```

```
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_2_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_down_strut_fx_2_s7 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_2_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_2_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_down_strut_fx_2_s8 calculated
    by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_2_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_2_s8',dpi=300)

#

#EXCLUDE?

#rear_vert_up_strut_x1

#N/A
rear_vert_up_strut_fx_1 = a[:,94]*1.e3;
rear_vert_up_strut_fy_1 = a[:,138]*1.e3;
rear_vert_up_strut_fz_1 = a[:,139]*1.e3;
```

```
mx_rvus_1 = a[:,95]*1.e3;
my_rvus_1 = a[:,96]*1.e3;
mz_rvus_1 = a[:,97]*1.e3;

[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] = \
     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)

damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_up_strut_fx_1_s1 calculated by
     script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFF)))
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_1_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_1_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_up_strut_fx_1_s2 calculated by
     script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_1_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_1_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_up_strut_fx_1_s3 calculated by
     script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
```

```
    damage*60*60*24*365*DFF)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_1_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_1_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_up_strut_fx_1_s4 calculated by
     script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_1_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_1_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_up_strut_fx_1_s5 calculated by
     script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_1_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_1_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_up_strut_fx_1_s6 calculated by
     script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))
```

```

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_1_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_1_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3, k=0.25)
print('Partial fatigue damage for rear_vert_up_strut_fx_1_s7 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_1_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_1_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3, k=0.25)
print('Partial fatigue damage for rear_vert_up_strut_fx_1_s8 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_1_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_1_s8',dpi=300)

#
--------------------------------------------------------------------------------

#rear_vert_up_strut_x2
#Joint type 2
#saddle in z-dir
```

```
rear_vert_up_strut_fx_2 = a[:,98]*1.e3;
rear_vert_up_strut_fy_2 = a[:,140]*1.e3;
rear_vert_up_strut_fz_2 = a[:,141]*1.e3;
mx_rvus_2 = a[:,99]*1.e3;
my_rvus_2 = a[:,100]*1.e3;
mz_rvus_2 = a[:,101]*1.e3;

[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] = \
     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)

damage = fatiguedamage_twoslope(t,s1,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_up_strut_fx_2_s1 calculated by
     script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFF)))
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_2_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_2_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_up_strut_fx_2_s2 calculated by
     script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_2_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_2_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
```

```
print('Partial fatigue damage for rear_vert_up_strut_fx_2_s3 calculated by
    script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_2_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_up_strut_fx_2_s4 calculated by
    script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_2_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_up_strut_fx_2_s5 calculated by
    script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_2_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_up_strut_fx_2_s6 calculated by
```

```
    script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_2_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_2_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_up_strut_fx_2_s7 calculated by
    script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_2_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_2_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1,loga1,m2,loga2,Nlim,th,tref=25E-3,
    k=0.25)
print('Partial fatigue damage for rear_vert_up_strut_fx_2_s8 calculated by
    script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_2_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_2_s8',dpi=300)

#
```

```
#Shaft

D_shaft = 2
T_shaft = 0.001


shaft_fx = a[:,10]*1.e3;
shaft_fy = a[:,11]*1.e3;
shaft_fz = a[:,12]*1.e3;
shaft_mx = a[:,13]*1.e3;
shaft_my = a[:,14]*1.e3;
shaft_mz = a[:,15]*1.e3;

[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] = \
     cyl_beam_stresses(t,shaft_fx,shaft_fy,shaft_fz,shaft_mx,shaft_my,
         shaft_mz,D_shaft,T_shaft,1,1,1,1)

damage = fatiguedamage_twoslope(t,s1,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for shaft_s1 calculated by script: %.5e (
    includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFF)))
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('shaft_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('shaft_s1_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for shaft_s2 calculated by script: %.5e (
    includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('shaft_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
```

```
#legend(loc='best')
savefig('shaft_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for shaft_s3 calculated by script: %.5e (
    includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('shaft_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('shaft_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for shaft_s4 calculated by script: %.5e (
    includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('shaft_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('shaft_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for shaft_s5 calculated by script: %.5e (
    includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('shaft_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
```

```
savefig('shaft_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for shaft_s6 calculated by script: %.5e (
    includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('shaft_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('shaft_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for shaft_s7 calculated by script: %.5e (
    includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('shaft_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('shaft_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for shaft_s8 calculated by script: %.5e (
    includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('shaft_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('shaft_s8',dpi=300)
```

```
# ---------------------------------------------------------------------------

#blade roots
# only one blade analyzed due to equilibrium for 4 blades


fx_bl_1_root = a[:,16]*1.e3;
fy_bl_1_root = a[:,17]*1.e3;
fz_bl_1_root = a[:,18]*1.e3;
mx_bl_1_root = a[:,19]*1.e3;
my_bl_1_root = a[:,20]*1.e3;
mz_bl_1_root = a[:,21]*1.e3;

[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] = \
     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)

damage = fatiguedamage_twoslope(t,s1,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for blade root_s1 calculated by script: %.5e
     (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFF)))
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('blade root_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for blade root_s2 calculated by script: %.5e
     (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s2/1.e6,'k-',label='')
```

```
xlabel('time [s]')
ylabel('blade root_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for blade root_s3 calculated by script: %.5e
     (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('blade root_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for blade root_s4 calculated by script: %.5e
     (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('blade root_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for blade root_s5 calculated by script: %.5e
     (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
```

```
ylabel('blade root_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1_st,loga1_st,m2_st,loga2_st,Nlim,th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for blade root_s6 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('blade root_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1_st,loga1_st,m2_st,loga2_st,Nlim,th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for blade root_s7 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('blade root_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1_st,loga1_st,m2_st,loga2_st,Nlim,th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for blade root_s8 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('blade root_s8 [MPa] ')
```

```
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_s8',dpi=300)

# ------------------------------------------------------------------------------------------



#For two blades uncomment this
fx_bl_2_root = a[:,166]*1.e3;
fy_bl_2_root = a[:,167]*1.e3;
fz_bl_2_root = a[:,168]*1.e3;
mx_bl_2_root = a[:,168]*1.e3;
my_bl_2_root = a[:,170]*1.e3;
mz_bl_2_root = a[:,171]*1.e3;

[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] = \
    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)

damage = fatiguedamage_twoslope(t,s1,m1_st,loga1_st,m2_st,loga2_st,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for blade root_2_s1 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(damage*60*60*24*DFF)))
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('blade root_2_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_2_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1_st,loga1_st,m2_st,loga2_st,Nlim,th,tref=25E-3,k=0.25)
print('Partial fatigue damage for blade root_2_s2 calculated by script: %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(damage*60*60*24*365*DFF)))

figure()
```

```
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('blade root_2_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_2_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1_st,loga1_st,m2_st,loga2_st,Nlim,th
    ,tref=25E-3,k=0.25)
print('Partial fatigue damage for blade root_2_s3 calculated by script:
    %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('blade root_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_2_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1_st,loga1_st,m2_st,loga2_st,Nlim,th
    ,tref=25E-3,k=0.25)
print('Partial fatigue damage for blade root_2_s4 calculated by script:
    %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('blade root_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_2_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1_st,loga1_st,m2_st,loga2_st,Nlim,th
    ,tref=25E-3,k=0.25)
print('Partial fatigue damage for blade root_2_s5 calculated by script:
    %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
```

```
xlabel('time [s]')
ylabel('blade root_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_2_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1_st,loga1_st,m2_st,loga2_st,Nlim,th
    ,tref=25E-3,k=0.25)
print('Partial fatigue damage for blade root_2_s6 calculated by script:
    %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('blade root_2_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_2_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1_st,loga1_st,m2_st,loga2_st,Nlim,th
    ,tref=25E-3,k=0.25)
print('Partial fatigue damage for blade root_2_s7 calculated by script:
    %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('blade root_2_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_2_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1_st,loga1_st,m2_st,loga2_st,Nlim,th
    ,tref=25E-3,k=0.25)
print('Partial fatigue damage for blade root_2_s8 calculated by script:
    %.5e (includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
```

```
ylabel('blade root_2_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_2_s8',dpi=300)

#——————————————————————————————————————#

#Rotor
rotor_fx = a[:,172]*1.e3;
rotor_fy = a[:,173]*1.e3;
rotor_fz = a[:,174]*1.e3;
rotor_mx = a[:,175]*1.e3;
rotor_my = a[:,176]*1.e3;
rotor_mz = a[:,177]*1.e3;

[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] = \
     cyl_beam_stresses(t,rotor_fx,rotor_fx,rotor_fx,rotor_fx,rotor_fx,
         rotor_fx,D_Rotor,Twall_Rotor,1,1,1,1)

damage = fatiguedamage_twoslope(t,s1,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for rotor_s1 calculated by script: %.5e (
    includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f days' %((t[-1]-t[1])/(
    damage*60*60*24*DFF)))
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rotor_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rotor_s1',dpi=300)

damage = fatiguedamage_twoslope(t,s2,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for rotor_s2 calculated by script: %.5e (
    includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s2/1.e6,'k-',label='')
xlabel('time [s]')
```

```
ylabel('rotor_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rotor_s2',dpi=300)

damage = fatiguedamage_twoslope(t,s3,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for rotor_s3 calculated by script: %.5e (
    includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rotor_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rotor_s3',dpi=300)

damage = fatiguedamage_twoslope(t,s4,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for rotor_s4 calculated by script: %.5e (
    includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rotor_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rotor_s4',dpi=300)

damage = fatiguedamage_twoslope(t,s5,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for rotor_s5 calculated by script: %.5e (
    includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rotor_s5 [MPa] ')
```

```
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rotor_s5',dpi=300)

damage = fatiguedamage_twoslope(t,s6,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for rotor_s6 calculated by script: %.5e (
    includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel(' rotor_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rotor_s6',dpi=300)

damage = fatiguedamage_twoslope(t,s7,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for rotor_s7 calculated by script: %.5e (
    includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rotor_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rotor_s7',dpi=300)

damage = fatiguedamage_twoslope(t,s8,m1_st,loga1_st,m2_st,loga2_st,Nlim,
    th_st,tref=25E-3,k=0.25)
print('Partial fatigue damage for rotor_s8 calculated by script: %.5e (
    includes residual cycles)' %damage)
print('which corresponds to a fatigue life of %.0f years' %((t[-1]-t[1])/(
    damage*60*60*24*365*DFF)))

figure()
plot(t,s8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rotor_s8 [MPa] ')
title('Axial Stress')
```

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

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

```
# -*- coding: utf-8 -*-
"""
Created on Thu Apr 29 13:19:16 2021

@author: Vetle Birkeland Aass
"""

import numpy as np
from atte_punkter import cyl_beam_stresses


from pylab import *
from numpy import loadtxt
from scipy import interpolate


SCF_AS_1 = 1
SCF_AC_1 = 1
SCF_OPB_1 = 1
SCF_IPB_1 = 1

SCF_AS_2 = 1
SCF_AC_2 = 1
SCF_OPB_2 = 1
SCF_IPB_2 = 1

SCF_AS_3 = 1
SCF_AC_3 = 1
SCF_OPB_3 = 1
SCF_IPB_3 = 1

SCF_AS_4 = 1
SCF_AC_4 = 1
SCF_OPB_4 = 1
SCF_IPB_4 = 1
```

```


#Yield strength

gamma  = 1.05
f_y = 355/gamma


#importing data
a    = loadtxt('sensors.txt',skiprows=500,dtype='float',delimiter=';')  # skip 500 s
t        = a[:,0];




#------------------------------------------------------------------------------------------
d = 0.3
twall = 0.024

#front_hor_left_strut_x1
#Joint type 1
#saddle in z-dir
front_hor_left_strut_fx_1 = a[:,22]*1.e3;
front_hor_left_strut_fy_1 = a[:,102]*1.e3;
front_hor_left_strut_fz_1 = a[:,103]*1.e3;
mx_fhls_1 = a[:,23]*1.e3;
my_fhls_1 = a[:,24]*1.e3;
mz_fhls_1 = a[:,25]*1.e3;

[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] = \
    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)


b = (max(svm1))*10e-6
print('front_hor_left_x1_svm1 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm2))*10e-6
print('front_hor_left_x1_svm2 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

```

```
b = (max(svm3))*10e-6
print('front_hor_left_x1_svm3 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm4))*10e-6
print('front_hor_left_x1_svm4 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('front_hor_left_x1_svm5 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('front_hor_left_x1_svm6 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('front_hor_left_x1_svm7 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('front_hor_left_x1_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')


figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_1_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_1_s1',dpi=300)



figure()
plot(t,svm2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_1_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
```

```
savefig('front_hor_left_strut_fx_1_s2',dpi=300)
figure()
plot(t,svm3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_1_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_1_s3',dpi=300)
figure()
plot(t,svm4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_1_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_1_s4',dpi=300)
figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_1_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_1_s5',dpi=300)
figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_1_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_1_s6',dpi=300)
figure()
plot(t,svm7/1.e6,'k-',label='')
xlabel('time [s]')
```

```
ylabel('front_hor_left_strut_fx_1_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_1_s7',dpi=300)



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_1_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_1_s8',dpi=300)

#------------------------------------------------------------------------------------------

#front_hor_left_strut_x2
#Joint type 2
#crown in z-dir
front_hor_left_strut_fx_2 = a[:,26]*1.e3;
front_hor_left_strut_fy_2 = a[:,104]*1.e3;
front_hor_left_strut_fz_2 = a[:,105]*1.e3;
mx_fhls_2 = a[:,27]*1.e3;
my_fhls_2 = a[:,28]*1.e3;
mz_fhls_2 = a[:,29]*1.e3;

[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] = \
    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)


b = (max(svm1))*10e-6
print('front_hor_left_x2_svm1 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm2))*10e-6
print('front_hor_left_x2_svm2 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm3))*10e-6
```

```
print('front_hor_left_x2_svm3 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm4))*10e-6
print('front_hor_left_x2_svm4 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('front_hor_left_x2_svm5 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('front_hor_left_x2_svm6 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('front_hor_left_x2_svm7 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('front_hor_left_x2_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')



figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_2_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_2_s1',dpi=300)



figure()
plot(t,svm2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_2_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
```

```
savefig('front_hor_left_strut_fx_2_s2',dpi=300)



figure()
plot(t,svm3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_2_s3',dpi=300)


figure()
plot(t,svm4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_2_s4',dpi=300)



figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_2_s5',dpi=300)



figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_2_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_2_s6',dpi=300)



figure()
plot(t,svm7/1.e6,'k-',label='')
```

```
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_2_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_2_s7',dpi=300)



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_left_strut_fx_2_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_left_strut_fx_2_s8',dpi=300)

#
d = 0.2
twall = 0.016

#front_hor_right_strut_x1
#Joint type 1
#saddle in z-dir
front_hor_right_strut_fx_1 = a[:,30]*1.e3;
front_hor_right_strut_fy_1 = a[:,106]*1.e3;
front_hor_right_strut_fz_1 = a[:,107]*1.e3;
mx_fhrs_1 = a[:,31]*1.e3;
my_fhrs_1 = a[:,32]*1.e3;
mz_fhrs_1 = a[:,33]*1.e3;


[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] = \
    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)


b = (max(svm1))*10e-6
print('front_hor_right_x1_svm1 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm2))*10e-6
print('front_hor_right_x1_svm2 = %.0f' %b)
```

```
if b > f_y:
    print('This element is under too much stress')

b = (max(svm3))*10e-6
print('front_hor_right_x1_svm3 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm4))*10e-6
print('front_hor_right_x1_svm4 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('front_hor_right_x1_svm5 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('front_hor_right_x1_svm6 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('front_hor_right_x1_svm7 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('front_hor_right_x1_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')


figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_1_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_1_s1',dpi=300)



figure()
plot(t,svm2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_1_s2 [MPa] ')
```

```
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_1_s2',dpi=300)



figure()
plot(t,svm3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_1_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_1_s3',dpi=300)



figure()
plot(t,svm4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_1_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_1_s4',dpi=300)



figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_1_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_1_s5',dpi=300)


figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_1_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_1_s6',dpi=300)
```

```

figure()
plot(t,svm7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('Sfront_hor_right_strut_fx_1_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_1_s7',dpi=300)



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_1_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_1_s8',dpi=300)

#
    ----------------------------------------------------------------------------------------------------


#front_hor_right_strut_x2
#Joint type 2
#crown in z-dir
front_hor_right_strut_fx_2 = a[:,34]*1.e3;
front_hor_right_strut_fy_2 = a[:,108]*1.e3;
front_hor_right_strut_fz_2 = a[:,109]*1.e3;
mx_fhrs_2 = a[:,35]*1.e3;
my_fhrs_2 = a[:,36]*1.e3;
mz_fhrs_2 = a[:,37]*1.e3;

[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] = \
     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)


b = (max(svm1))*10e-6
print('front_hor_right_x2_svm1 = %.0f' %b)
if b > f_y:
     print('This element is under too much stress')

b = (max(svm2))*10e-6
```

```
print('front_hor_right_x2_svm2 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm3))*10e-6
print('front_hor_right_x2_svm3 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm4))*10e-6
print('front_hor_right_x2_svm4 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('front_hor_right_x2_svm5 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('front_hor_right_x2_svm6 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('front_hor_right_x2_svm7 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('front_hor_right_x2_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')



figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_2_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_2_s1',dpi=300)



figure()
plot(t,svm2/1.e6,'k-',label='')
```

```
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_2_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_2_s2',dpi=300)



figure()
plot(t,svm3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_2_s3',dpi=300)



figure()
plot(t,svm4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_2_s4',dpi=300)



figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_2_s5',dpi=300)



figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_2_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
```

```
savefig('front_hor_right_strut_fx_2_s6',dpi=300)



figure()
plot(t,svm7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_2_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_2_s7',dpi=300)



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_hor_right_strut_fx_2_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_hor_right_strut_fx_2_s8',dpi=300)

# ------------------------------------------------------------------------------
d = 0.3
twall = 0.012

#front_vert_down_strut_x1
#Joint type 1
#crown in z-dir
front_vert_down_strut_fx_1 = a[:,38]*1.e3;
front_vert_down_strut_fy_1 = a[:,110]*1.e3;
front_vert_down_strut_fz_1 = a[:,111]*1.e3;
mx_fvds_1 = a[:,39]*1.e3;
my_fvds_1 = a[:,40]*1.e3;
mz_fvds_1 = a[:,41]*1.e3;

[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] = \
    cyl_beam_stresses(t,front_vert_down_strut_fx_1,front_vert_down_strut_fy_1,front_vert_down_strut_fy_1,mx_fvds_1,my_fvds_1,mz_fvds_1,d,twall,SCF_AC_1,SCF_AS_1,SCF_IPB_1,SCF_OPB_1)


b = (max(svm1))*10e-6
print('front_vert_down_x1_svm1 = %.0f' %b)
```

```
if b > f_y:
    print('This element is under too much stress')

b = (max(svm2))*10e-6
print('front_vert_down_x1_svm2 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm3))*10e-6
print('front_vert_down_x1_svm3 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm4))*10e-6
print('front_vert_down_x1_svm4 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('front_vert_down_x1_svm5 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('front_vert_down_x1_svm6 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('front_vert_down_x1_svm7 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('front_vert_down_x1_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')




figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_1_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_1_s1',dpi=300)

```

```
figure()
plot(t,svm2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_1_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_1_s2',dpi=300)



figure()
plot(t,svm3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_1_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_1_s3',dpi=300)



figure()
plot(t,svm4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_1_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_1_s4',dpi=300)



figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_1_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_1_s5',dpi=300)



figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
```

```
ylabel('front_vert_down_strut_fx_1_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_1_s6',dpi=300)



figure()
plot(t,svm7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_1_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_1_s7',dpi=300)



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_1_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_1_s8',dpi=300)

# ----------------------------------------------------------------------------------------

#front_vert_down_strut_x2
#Joint type 2
#saddle in z-dir
front_vert_down_strut_fx_2 = a[:,42]*1.e3;
front_vert_down_strut_fy_2 = a[:,112]*1.e3;
front_vert_down_strut_fz_2 = a[:,113]*1.e3;
mx_fvds_2 = a[:,43]*1.e3;
my_fvds_2 = a[:,44]*1.e3;
mz_fvds_2 = a[:,45]*1.e3;

[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] = \
     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)


```

```
b = (max(svm1))*10e-6
print('front_vert_down_x2_svm1 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm2))*10e-6
print('front_vert_down_x2_svm2 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm3))*10e-6
print('front_vert_down_x2_svm3 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm4))*10e-6
print('front_vert_down_x2_svm4 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('front_vert_down_x2_svm5 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('front_vert_down_x2_svm6 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('front_vert_down_x2_svm7 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('front_vert_down_x2_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')




figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_2_s1 [MPa] ')
```

```
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_2_s1',dpi=300)



figure()
plot(t,svm2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_2_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_2_s2',dpi=300)



figure()
plot(t,svm3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_2_s3',dpi=300)



figure()
plot(t,svm4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_2_s4',dpi=300)



figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_2_s5',dpi=300)

```

```
figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_2_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_2_s6',dpi=300)



figure()
plot(t,svm7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_2_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_2_s7',dpi=300)



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_down_strut_fx_2_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_down_strut_fx_2_s8',dpi=300)
#
------------------------------------------------------------------------

d = 0.3
twall = 0.016

#front_vert_up_strut_x1
#Joint type 1
#crown in z-dir
front_vert_up_strut_fx_1 = a[:,46]*1.e3;
front_vert_up_strut_fy_1 = a[:,114]*1.e3;
front_vert_up_strut_fz_1 = a[:,115]*1.e3;
mx_fvus_1 = a[:,47]*1.e3;
my_fvus_1 = a[:,48]*1.e3;
mz_fvus_1 = a[:,49]*1.e3;

```

```
[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] = \
     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)


b = (max(svm1))*10e-6
print('front_vert_up_x1_svm1 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm2))*10e-6
print('front_vert_up_x1_svm2 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm3))*10e-6
print('front_vert_up_x1_svm3 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm4))*10e-6
print('front_vert_up_x1_svm4 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('front_vert_up_x1_svm5 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('front_vert_up_x1_svm6 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('front_vert_up_x1_svm7 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('front_vert_up_x1_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')
```

```
figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_1_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_1_s1',dpi=300)



figure()
plot(t,svm2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_1_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_1_s2',dpi=300)



figure()
plot(t,svm3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_1_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_1_s3',dpi=300)



figure()
plot(t,svm4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_1_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_1_s4',dpi=300)



figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_1_s5 [MPa] ')
title('Axial Stress')
```

```
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_1_s5',dpi=300)



figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_1_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_1_s6',dpi=300)



figure()
plot(t,svm7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_1_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_1_s7',dpi=300)



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_1_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_1_s8',dpi=300)


#

#front_vert_up_strut_x2
#Joint type 2
#saddle in z-dir
front_vert_up_strut_fx_2 = a[:,50]*1.e3;
front_vert_up_strut_fy_2 = a[:,116]*1.e3;
front_vert_up_strut_fz_2 = a[:,117]*1.e3;
mx_fvus_2 = a[:,51]*1.e3;
```

```
my_fvus_2 = a[:,52]*1.e3;
mz_fvus_2 = a[:,53]*1.e3;

[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] = \
    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)


b = (max(svm1))*10e-6
print('front_vert_up_x2_svm1 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm2))*10e-6
print('front_vert_up_x2_svm2 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm3))*10e-6
print('front_vert_up_x2_svm3 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm4))*10e-6
print('front_vert_up_x2_svm4 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('front_vert_up_x2_svm5 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('front_vert_up_x2_svm6 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('front_vert_up_x2_svm7 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('front_vert_up_x2_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')
```

```
figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_2_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_2_s1',dpi=300)



figure()
plot(t,svm2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_2_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_2_s2',dpi=300)



figure()
plot(t,svm3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_2_s3',dpi=300)



figure()
plot(t,svm4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_2_s4',dpi=300)



figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
```

```
ylabel('front_vert_up_strut_fx_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_2_s5',dpi=300)



figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_2_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_2_s6',dpi=300)



figure()
plot(t,svm7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_2_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_2_s7',dpi=300)



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('front_vert_up_strut_fx_2_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('front_vert_up_strut_fx_2_s8',dpi=300)


#
    ----------------------------------------------------------------------------

d = 0.2
twall = 0.008

#rear_left_down_strut_x1
#Joint type 3
```

```
#crown in z-dir
rear_left_down_strut_fx_1 = a[:,54]*1.e3;
rear_left_down_strut_fy_1 = a[:,118]*1.e3;
rear_left_down_strut_fz_1 = a[:,119]*1.e3;
mx_rlds_1 = a[:,55]*1.e3;
my_rlds_1 = a[:,56]*1.e3;
mz_rlds_1 = a[:,57]*1.e3;

[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] = \
        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)



b = (max(svm1))*10e-6
print('rear_left_down_x1_svm1 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm2))*10e-6
print('rear_left_down_x1_svm2 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm3))*10e-6
print('rear_left_down_x1_svm3 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm4))*10e-6
print('rear_left_down_x1_svm4 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('rear_left_down_x1_svm5 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('rear_left_down_x1_svm6 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('rear_left_down_x1_svm7 = %.0f' %b)
if b > f_y:
```

```
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('rear_left_down_x1_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')



figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_1_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_1_s1',dpi=300)


figure()
plot(t,svm2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_1_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_1_s2',dpi=300)



figure()
plot(t,svm3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_1_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_1_s3',dpi=300)


figure()
plot(t,svm4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_1_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_1_s4',dpi=300)

```

```
figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_1_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_1_s5',dpi=300)



figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_1_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_1_s6',dpi=300)


figure()
plot(t,svm7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_1_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_1_s7',dpi=300)



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_1_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_1_s8',dpi=300)

#----------------------------------------------------------------------------------------------------


#rear_left_down_strut_x2
```

```
#Joint type 4
#crown in z-dir
rear_left_down_strut_fx_2 = a[:,58]*1.e3;
rear_left_down_strut_fy_2 = a[:,120]*1.e3;
rear_left_down_strut_fz_2 = a[:,121]*1.e3;
mx_rlds_2 = a[:,59]*1.e3;
my_rlds_2 = a[:,60]*1.e3;
mz_rlds_2 = a[:,61]*1.e3;

[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] = \
    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)


b = (max(svm1))*10e-6
print('rear_left_down_x2_svm1 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm2))*10e-6
print('rear_left_down_x2_svm2 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm3))*10e-6
print('rear_left_down_x2_svm3 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm4))*10e-6
print('rear_left_down_x2_svm4 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('rear_left_down_x2_svm5 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('rear_left_down_x2_svm6 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('rear_left_down_x2_svm7 = %.0f' %b)
if b > f_y:
```

```
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('rear_left_down_x2_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')



figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_2_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_2_s1',dpi=300)



figure()
plot(t,svm2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_2_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_2_s2',dpi=300)



figure()
plot(t,svm3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_2_s3',dpi=300)



figure()
plot(t,svm4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
```

```
savefig('rear_left_down_strut_fx_2_s4',dpi=300)



figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_2_s5',dpi=300)



figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_2_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_2_s6',dpi=300)



figure()
plot(t,svm7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_2_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_2_s7',dpi=300)



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_down_strut_fx_2_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_down_strut_fx_2_s8',dpi=300)



#
```

```
d = 0.2
twall = 0.002

d = 0.2
twall = 0.010

#rear_left_up_strut_x1
#Joint type 3
#crown in z-dir
rear_left_up_strut_fx_1 = a[:,62]*1.e3;
rear_left_up_strut_fy_1 = a[:,122]*1.e3;
rear_left_up_strut_fz_1 = a[:,123]*1.e3;
mx_rlus_1 = a[:,63]*1.e3;
my_rlus_1 = a[:,64]*1.e3;
mz_rlus_1 = a[:,65]*1.e3;

[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] = \
    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)


b = (max(svm1))*10e-6
print('rear_left_up_x1_svm1 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm2))*10e-6
print('rear_left_up_x1_svm2 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm3))*10e-6
print('rear_left_up_x1_svm3 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm4))*10e-6
print('rear_left_up_x1_svm4 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('rear_left_up_x1_svm5 = %.0f' %b)
if b > f_y:
```

```
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('rear_left_up_x1_svm6 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('rear_left_up_x1_svm7 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('rear_left_up_x1_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')



figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_1_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_1_s1',dpi=300)


figure()
plot(t,svm2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_1_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_1_s2',dpi=300)



figure()
plot(t,svm3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_1_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_1_s3',dpi=300)
```

```
figure()
plot(t,svm4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_1_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_1_s4',dpi=300)



figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_1_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_1_s5',dpi=300)



figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_1_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_1_s6',dpi=300)



figure()
plot(t,svm7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_1_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_1_s7',dpi=300)



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
```

```
ylabel('rear_left_up_strut_fx_1_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_1_s8',dpi=300)




#

#rear_left_up_strut_x2
#Joint type 4
#crown in z-dir
rear_left_up_strut_fx_2 = a[:,66]*1.e3;
rear_left_up_strut_fy_2 = a[:,124]*1.e3;
rear_left_up_strut_fz_2 = a[:,125]*1.e3;
mx_rlus_2 = a[:,67]*1.e3;
my_rlus_2 = a[:,68]*1.e3;
mz_rlus_2 = a[:,69]*1.e3;

[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] = \
    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)



b = (max(svm1))*10e-6
print('rear_left_up_x2_svm1 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm2))*10e-6
print('rear_left_up_x2_svm2 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm3))*10e-6
print('rear_left_up_x2_svm3 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm4))*10e-6
print('rear_left_up_x2_svm4 = %.0f' %b)
if b > f_y:
```

```
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('rear_left_up_x2_svm5 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('rear_left_up_x2_svm6 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('rear_left_up_x2_svm7 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('rear_left_up_x2_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')




figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_2_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_2_s1',dpi=300)



figure()
plot(t,svm2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_2_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_2_s2',dpi=300)



figure()
plot(t,svm3/1.e6,'k-',label='')
```

```
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_2_s3',dpi=300)


figure()
plot(t,svm4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_2_s4',dpi=300)



figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_2_s5',dpi=300)



figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_2_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_2_s6',dpi=300)



figure()
plot(t,svm7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_2_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_2_s7',dpi=300)
```

```python



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_left_up_strut_fx_2_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_left_up_strut_fx_2_s8',dpi=300)

# --------------------------------------------------------------------------------

#rear_right_down_strut_x1
#Joint type 3
#crown in z-dir
rear_right_down_strut_fx_1 = a[:,70]*1.e3;
rear_right_down_strut_fy_1 = a[:,126]*1.e3;
rear_right_down_strut_fz_1 = a[:,127]*1.e3;
mx_rrds_1 = a[:,71]*1.e3;
my_rrds_1 = a[:,72]*1.e3;
mz_rrds_1 = a[:,73]*1.e3;

[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] = \
    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)


b = (max(svm1))*10e-6
print('rear_right_down_x1_svm1 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm2))*10e-6
print('rear_right_down_x1_svm2 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm3))*10e-6
print('rear_right_down_svm3 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm4))*10e-6
```

```
print('rear_right_down_x1_svm4 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('rear_right_down_x1_svm5 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('rear_right_down_x1_svm6 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('rear_right_down_x1_svm7 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('rear_right_down_x1_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')



figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_1_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_1_s1',dpi=300)



figure()
plot(t,svm2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_1_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_1_s2',dpi=300)



figure()
```

```
plot(t,svm3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_1_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_1_s3',dpi=300)



figure()
plot(t,svm4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_1_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_1_s4',dpi=300)



figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_1_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_1_s5',dpi=300)



figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_1_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_1_s6',dpi=300)



figure()
plot(t,svm7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_1_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
```

```
#legend(loc='best')
savefig('rear_right_down_strut_fx_1_s7',dpi=300)



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_1_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_1_s8',dpi=300)

#
    ----------------------------------------------------------------------------

#rear_right_down_strut_x2
#Joint type 4
#crown in z-dir
rear_right_down_strut_fx_2 = a[:,74]*1.e3;
rear_right_down_strut_fy_2 = a[:,128]*1.e3;
rear_right_down_strut_fz_2 = a[:,129]*1.e3;
mx_rrds_2 = a[:,75]*1.e3;
my_rrds_2 = a[:,76]*1.e3;
mz_rrds_2 = a[:,77]*1.e3;

[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] = \
     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)


b = (max(svm1))*10e-6
print('rear_right_down_x2_svm1 = %.0f' %b)
if b > f_y:
     print('This element is under too much stress')

b = (max(svm2))*10e-6
print('rear_right_down_x2_svm2 = %.0f' %b)
if b > f_y:
     print('This element is under too much stress')

b = (max(svm3))*10e-6
print('rear_right_down_x2_svm3 = %.0f' %b)
if b > f_y:
     print('This element is under too much stress')
```

```
b = (max(svm4))*10e-6
print('rear_right_down_x2_svm4 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('rear_right_down_x2_svm5 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('rear_right_down_x2_svm6 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('rear_right_down_x2_svm7 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('rear_right_down_x2_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')




figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_2_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_2_s1',dpi=300)



figure()
plot(t,svm2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_2_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_2_s2',dpi=300)

```

```
figure()
plot(t,svm3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_2_s3',dpi=300)



figure()
plot(t,svm4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_2_s4',dpi=300)



figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_2_s5',dpi=300)


figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_2_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_2_s6',dpi=300)



figure()
plot(t,svm7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_2_s7 [MPa] ')
```

```
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_2_s7',dpi=300)



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_down_strut_fx_2_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_down_strut_fx_2_s8',dpi=300)

#
    ------------------------------------------------------------------------------------------------

d = 0.2
twall = 0.008

#rear_right_up_strut_x1
#Joint type 3
#crown in z-dir
rear_right_up_strut_fx_1 = a[:,78]*1.e3;
rear_right_up_strut_fy_1 = a[:,130]*1.e3;
rear_right_up_strut_fz_1 = a[:,131]*1.e3;
mx_rrus_1 = a[:,79]*1.e3;
my_rrus_1 = a[:,80]*1.e3;
mz_rrus_1 = a[:,81]*1.e3;

[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] = \
    cyl_beam_stresses(t,rear_right_up_strut_fx_1,rear_right_up_strut_fy_1,
        rear_right_up_strut_fy_1,mx_rrus_1,my_rrus_1,mz_rrus_1,d,twall,
        SCF_AC_3,SCF_AS_3,SCF_IPB_3,SCF_OPB_3)


b = (max(svm1))*10e-6
print('rear_right_up_x1_svm1 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm2))*10e-6
print('rear_right_up_x1_svm2 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')
```

```
b = (max(svm3))*10e-6
print('rear_right_up_svm3 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm4))*10e-6
print('rear_right_up_x1_svm4 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('rear_right_up_x1_svm5 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('rear_right_up_x1_svm6 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('rear_right_up_x1_svm7 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('rear_right_up_x1_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')



figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_1_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_1_s1',dpi=300)



figure()
plot(t,svm2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_1_s2 [MPa] ')
title('Axial Stress')
```

```
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_1_s2',dpi=300)


figure()
plot(t,svm3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_1_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_1_s3',dpi=300)



figure()
plot(t,svm4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_1_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_1_s4',dpi=300)



figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_1_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_1_s5',dpi=300)



figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_1_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_1_s6',dpi=300)
```

```
figure()
plot(t,svm7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_1_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_1_s7',dpi=300)



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_1_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_1_s8',dpi=300)

#
    ______________________________________________________________________


#rear_right_up_strut_x2
#Joint type 4
#crown in z-dir
rear_right_up_strut_fx_2 = a[:,82]*1.e3;
rear_right_up_strut_fy_2 = a[:,132]*1.e3;
rear_right_up_strut_fz_2 = a[:,133]*1.e3;
mx_rrus_2 = a[:,83]*1.e3;
my_rrus_2 = a[:,84]*1.e3;
mz_rrus_2 = a[:,85]*1.e3;

[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] = \
     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)


b = (max(svm1))*10e-6
print('rear_right_up_x2_svm1 = %.0f' %b)
if b > f_y:
     print('This element is under too much stress')

b = (max(svm2))*10e-6
print('rear_right_up_x2_svm2 = %.0f' %b)
if b > f_y:
```

```
    print('This element is under too much stress')

b = (max(svm3))*10e-6
print('rear_right_up_x2_svm3 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm4))*10e-6
print('rear_right_up_x2_svm4 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('rear_right_up_x2_svm5 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('rear_right_up_x2_svm6 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('rear_right_up_x2_svm7 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('rear_right_up_x2_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')


figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_2_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_2_s1',dpi=300)



figure()
plot(t,svm2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_2_s2 [MPa] ')
title('Axial Stress')
```

```
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_2_s2',dpi=300)



figure()
plot(t,svm3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_2_s3',dpi=300)



figure()
plot(t,svm4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_2_s4',dpi=300)


figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_2_s5',dpi=300)


figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_2_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_2_s6',dpi=300)


figure()
plot(t,svm7/1.e6,'k-',label='')
```

```
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_2_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_2_s7',dpi=300)



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_right_up_strut_fx_2_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_right_up_strut_fx_2_s8',dpi=300)

# ------------------------------------------------------------



# ------------------------------------------------------------
d = 0.2
twall = 0.016

#rear_vert_down_strut_x2
#Joint type 2
#saddle in z-dir
rear_vert_down_strut_fx_2 = a[:,90]*1.e3;
rear_vert_down_strut_fy_2 = a[:,136]*1.e3;
rear_vert_down_strut_fz_2 = a[:,137]*1.e3;
mx_rvds_2 = a[:,91]*1.e3;
my_rvds_2 = a[:,92]*1.e3;
mz_rvds_2 = a[:,93]*1.e3;

[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] = \
    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)

b = (max(svm1))*10e-6
print('rear_vert_down_svm1 = %.0f' %b)
if b > f_y:
```

```
    print('This element is under too much stress')

b = (max(svm2))*10e-6
print('rear_vert_down_svm2 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm3))*10e-6
print('rear_vert_down_svm3 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm4))*10e-6
print('rear_vert_down_svm4 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('rear_vert_down_svm5 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('rear_vert_down_svm6 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('rear_vert_down_svm7 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('rear_vert_down_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')


figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_2_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_2_s1',dpi=300)



```

```
figure()
plot(t,svm2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_2_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_2_s2',dpi=300)



figure()
plot(t,svm3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_2_s3',dpi=300)



figure()
plot(t,svm4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_2_s4',dpi=300)



figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_2_s5',dpi=300)



figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_2_s6 [MPa] ')
title('Axial Stress')
```

```
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_2_s6',dpi=300)



figure()
plot(t,svm7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_2_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_2_s7',dpi=300)



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_down_strut_fx_2_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_down_strut_fx_2_s8',dpi=300)

# ----------------------------------------------------------------------------------------------------------------


# ----------------------------------------------------------------------------------------------------------------

#rear_vert_up_strut_x2
#Joint type 2
#saddle in z-dir
rear_vert_up_strut_fx_2 = a[:,98]*1.e3;
rear_vert_up_strut_fy_2 = a[:,140]*1.e3;
rear_vert_up_strut_fz_2 = a[:,141]*1.e3;
mx_rvus_2 = a[:,99]*1.e3;
my_rvus_2 = a[:,100]*1.e3;
mz_rvus_2 = a[:,101]*1.e3;

[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] = \
     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)


b = (max(svm1))*10e-6
print('rear_vert_up_svm1 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm2))*10e-6
print('rear_vert_up_svm2 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm3))*10e-6
print('rear_vert_up_svm3 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm4))*10e-6
print('rear_vert_up_svm4 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('rear_vert_up_svm5 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('rear_vert_up_svm6 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('rear_vert_up_svm7 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('rear_vert_up_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')



figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_2_s1 [MPa] ')
```

```
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_2_s1',dpi=300)



figure()
plot(t,svm2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_2_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_2_s2',dpi=300)



figure()
plot(t,svm3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_2_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_2_s3',dpi=300)



figure()
plot(t,svm4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_2_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_2_s4',dpi=300)



figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_2_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_2_s5',dpi=300)

```

```


figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_2_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_2_s6',dpi=300)



figure()
plot(t,svm7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_2_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_2_s7',dpi=300)



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rear_vert_up_strut_fx_2_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rear_vert_up_strut_fx_2_s8',dpi=300)

#
    ------------------------------------------------------------------------

#Shaft

D_shaft = 1
T_shaft = 0.004

D_shaft = 1
T_shaft = 0.008


shaft_fx = a[:,10]*1.e3;
shaft_fy = a[:,11]*1.e3;
shaft_fz = a[:,12]*1.e3;
```

```
shaft_mx = a[:,13]*1.e3;
shaft_my = a[:,14]*1.e3;
shaft_mz = a[:,15]*1.e3;

[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] = \
     cyl_beam_stresses(t,shaft_fx,shaft_fy,shaft_fz,shaft_mx,shaft_my,
         shaft_mz,D_shaft,T_shaft,1,1,1,1)

b = (max(svm1))*10e-6
print('shaft_svm1 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm2))*10e-6
print('shaft_svm2 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm3))*10e-6
print('shaft_svm3 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm4))*10e-6
print('shaft_svm4 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('shaft_svm5 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('shaft_svm6 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('shaft_svm7 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('shaft_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')
```

```
figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('shaft_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('shaft_s1_s1',dpi=300)



figure()
plot(t,svm2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('shaft_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('shaft_s2',dpi=300)



figure()
plot(t,svm3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('shaft_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('shaft_s3',dpi=300)



figure()
plot(t,svm4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('shaft_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('shaft_s4',dpi=300)



figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
```

```
ylabel('shaft_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('shaft_s5',dpi=300)



figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('shaft_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('shaft_s6',dpi=300)



figure()
plot(t,svm7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('shaft_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('shaft_s7',dpi=300)



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('shaft_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('shaft_s8',dpi=300)



#
#blade roots
# only one blade analyzed due to equilibrium for 4 blades
D = 1.2
TWALL = 0.0028

```

```
D = 1.2
TWALL = 0.008

fx_bl_1_root = a[:,16]*1.e3;
fy_bl_1_root = a[:,17]*1.e3;
fz_bl_1_root = a[:,18]*1.e3;
mx_bl_1_root = a[:,19]*1.e3;
my_bl_1_root = a[:,20]*1.e3;
mz_bl_1_root = a[:,21]*1.e3;

[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] = \
     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)


b = (max(svm1))*10e-6
print('blade_svm1 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm2))*10e-6
print('blade_svm2 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm3))*10e-6
print('blade_svm3 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm4))*10e-6
print('blade_svm4 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('blade_svm5 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('blade_svm6 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('blade_svm7 = %.0f' %b)
if b > f_y:
```

```
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('blade_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')


figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('blade root_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_s1',dpi=300)



figure()
plot(t,svm2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('blade root_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_s2',dpi=300)



figure()
plot(t,svm3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('blade root_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_s3',dpi=300)



figure()
plot(t,svm4/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('blade root_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_s4',dpi=300)
```

```
figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('blade root_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_s5',dpi=300)



figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('blade root_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_s6',dpi=300)



figure()
plot(t,svm7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('blade root_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_s7',dpi=300)



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('blade root_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('blade root_s8',dpi=300)

#
```

```

#----------------------------------------------------------------------------#
D_Rotor = 0.5
Twall_Rotor = 0.004

D_Rotor = 0.5
Twall_Rotor = 0.04

#Rotor
rotor_fx = a[:,172]*1.e3;
rotor_fy = a[:,173]*1.e3;
rotor_fz = a[:,174]*1.e3;
rotor_mx = a[:,175]*1.e3;
rotor_my = a[:,176]*1.e3;
rotor_mz = a[:,177]*1.e3;

[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] = \
     cyl_beam_stresses(t,rotor_fx,rotor_fx,rotor_fx,rotor_fx,rotor_fx,
         rotor_fx,D_Rotor,Twall_Rotor,1,1,1,1)


b = (max(svm1))*10e-6
print('rotor_svm1 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm2))*10e-6
print('rotor_svm2 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm3))*10e-6
print('rotor_svm3 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm4))*10e-6
print('rotor_svm4 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm5))*10e-6
print('rotor_svm5 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm6))*10e-6
print('rotor_svm6 = %.0f' %b)
```

```
if b > f_y:
    print('This element is under too much stress')

b = (max(svm7))*10e-6
print('rotor_svm7 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')

b = (max(svm8))*10e-6
print('rotor_svm8 = %.0f' %b)
if b > f_y:
    print('This element is under too much stress')



figure()
plot(t,svm1/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rotor_s1 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rotor_s1',dpi=300)


figure()
plot(t,svm2/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rotor_s2 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rotor_s2',dpi=300)



figure()
plot(t,svm3/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rotor_s3 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rotor_s3',dpi=300)



figure()
plot(t,svm4/1.e6,'k-',label='')
```

```
xlabel('time [s]')
ylabel('rotor_s4 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rotor_s4',dpi=300)



figure()
plot(t,svm5/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rotor_s5 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rotor_s5',dpi=300)



figure()
plot(t,svm6/1.e6,'k-',label='')
xlabel('time [s]')
ylabel(' rotor_s6 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rotor_s6',dpi=300)



figure()
plot(t,svm7/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rotor_s7 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
savefig('rotor_s7',dpi=300)



figure()
plot(t,svm8/1.e6,'k-',label='')
xlabel('time [s]')
ylabel('rotor_s8 [MPa] ')
title('Axial Stress')
#axis([-16.,90.,-1.5,1.7])
#legend(loc='best')
```

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

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

Postboks 5003
NO-1432 Ås
Norway