ECN-I--05-005

PHATAS Release "NOV-2003" and "APR-2005" USER'S MANUAL

Program for Horizontal Axis wind Turbine Analysis and Simulation

C. Lindenburg

May 2005

Preface

This manual is related to the use of PHATAS release "APR-2005" and "NOV-2003".
Release "APR-2005" and the former release "NOV-2003" were developed as result of activities making the code suitable for load-set calculations (following G.L., IEC, or NVN) wind turbine and rotor blade manufacturers, ECN project no. 7.9421. This included among others some modifications of the input structure, using a global coordinate system (to describe e.g. the wind loading) that corresponds with the G.L. '*Vorschriften und Richtlinien*', and improvements of the control algorithm.

In performing these modifications it was kept in mind that this PHATAS version should be running under the rotor blade design tool *FOCUS*, ECN project no. 7.9413.

In addition to the User's manual (the underlying report) release "APR-2005" is documented in a validation report (as non-formal ECN note) that focuses on the structural dynamic modelling of the tower and the turbine-tower interactions.

The most significant improvements issued for release "APR-2005" compared to "NOV-2003" are an extension to the variables in the binary datafile with the forces in the blade root, adding some inertia properties of the nacelle, and allowing analysis of towers with a tripod base. Small improvements on release "APR-2005" may be issued as release "APR-2005b", "APR-2005c", etcetera. For those releases the input dataset and the format of the binary datafile will remain the same, which implies that this manual will still be applicable.

Acknowledgement

Writing User's manual requires special attention because it should be compliant with the computer program. In this respect the author would like to thank his colleagues of ECN, Gerben D. de Winkel of WMC, and Edo Kuipers of CTC for their comments on the PHATAS code and the manual.

ABSTRACT

The computer program PHATAS, ”Program for Horizontal Axis wind Turbine Analysis and Simulation”, is developed for the time-domain calculation of the dynamic behaviour and the corresponding loads on a Horizontal Axis wind Turbine, HAT. The program PHATAS is available for use as ”console application” on a PC operating under MS-Windows, on a UNIX work station, or under LINUX.

The most significant improvement of PHATAS release "APR-2005" and "NOV-2003" compared to the former releases "OCT-2002" and "JAN-2002" is that they were made more suitable for load-set calculations, either if used as single application of under the design package Focus. This manual gives a global description of the type of calculations (PV-curve, dynamic response) that can be done with PHATAS, while a detailed description is given on all input items of the turbine and of the tower model. Special attention is paid to the description of the controller while some explanation is given on the structure for linking additional algorithms, based on a simple yaw controller.

Descriptions of additional programs provided with PHATAS are added in appendices. Among these programs are a pre-processor for load-case input files following IEC or GL, and postprocessors to retrieve load-time series or to select extreme loads of a complete load set.

Keywords

Aeroelastic behaviour, Computer program, Design load calculations
User’s manual, Variable speed control, Wind turbine.

CONTENTS

LIST OF SYMBOLS

a	[.]	Parameter used to describe the turbulence distribution (App B).
C_{a}	[.]	Apparent mass coefficient, used for tower accelerations in water.
C_{Dax}	[.]	Axial force coefficient $= \frac{\text{Axial force}}{1/2 \cdot \rho \cdot V_{\text{steady}}^2 \cdot \pi \cdot R^2}$.
C_{K}	[.]	Aerodynamic power coefficient $= \frac{\text{Aerodynamic power}}{1/2 \cdot \rho \cdot V_{\text{steady}}^3 \cdot \pi \cdot R^2}$.
C_{M}	[.]	Water mass coefficient, for a tower in accelerating water $= 1 + C_{\mathrm{a}}$.
D	[m]	Diameter of the tower cross section, see chapter 5.
e_{xt}	[m]	Location of teeter hinge behind the rotor centre, see Figure 3.
$e_{\mathrm{z.root}}$	[m]	Spanwise location of the blade root; '**blade_root_radius**'.
g	[m/s^2]	Gravitational constant (= 9.81m/s^2).
I_{15}	[.]	Turbulence intensity at 15m/s wind, used for the load case description.
KC	[.]	Keulegan-Carpenter number, used for wave loads, see chapter 5.
L	[m]	Length of a rotor blade measured from the root; '**blade_span**'.
P_{rated}	[W]	Nominal power.
Q	[N·m]	Torque.
r	[m]	Spanwise co-ordinate along the blade, measured from rotor centre.
R	[m]	Radius of the tip $= (L + e_{\mathrm{z.root}}) \cos \alpha_{\mathrm{c}}$.
t	[s]	Time.
T	[s]	Time period of the modelled wind gust, section 3.7. or period of oscillating wave loads, see chapter 5.
V_{ref}	[m/s]	Reference wind speed, used for the load case description.
$V_{\mathrm{steady}}(Y,Z)$	[m/s]	Steady wind velocity at position (Y,Z) .
$V_{\mathrm{wind}}(Y,Z,t)$	[m/s]	Actual wind speed at position (Y,Z) at time t .
X, Y, Z	[m]	Inertial coordinate axes following Germanischer Lloyd, Figure 3.
$X_{\mathrm{r}}, Y_{\mathrm{r}}, Z_{\mathrm{r}}$	[m]	Co-ordinates in the rotor system.
$X_{\mathrm{w}}, Y_{\mathrm{w}}$	[m]	Wind-direction co-ordinate system, see Figure 10.
Z	[m]	Height above ground level.
$Y_{\mathrm{hub}}, Z_{\mathrm{hub}}$	[m]	Position of the rotor hub.
α_{c}	[°]	'**cone_angle**' of the blades, see Figure 3.
α_{n}	[°]	'**tilt_angle**' of the rotor shaft, see Figure 3.
Δt	[s]	Time increment in the calculation.
$\delta_{3\mathrm{t}}$	[°]	Orientation of the teeter hinge, Figure 3.
δ_{tow}	[°]	Torsional deformation of the tower, Figure 3.
ϕ	[°]	Azimuth in the rotor plane $= \arctan(\frac{Y_{\mathrm{hub}} - Y}{Z - Z_{\mathrm{hub}}})$.
ϕ_{r}	[°]	Rotor azimuth, see Figure 3.
ϕ_{t}	[°]	Teeter angle, see Figure 3.
ϕ_{y}	[°]	Yaw angle, see Figure 3 and Figure 10.
ν	[rad/s]	Natural frequency, section 3.2 and 8.2.
Λ_1	[m]	Turbulence length scale used for wind description (App B).
λ	[.]	Tip speed ratio $= \Omega \cdot R / V_{\mathrm{steady}}(Y_{\mathrm{hub}}, Z_{\mathrm{hub}})$.
θ_{p}	[°]	Pitch angle, see Figure 8.
θ_{tw}	[°]	Blade twist angle, see Figure 8.
Ω	[rad/s]	Rotor speed.
Ω_{rated}	[rad/s]	Nominal rotor speed '**rated_rotorspeed**'.
Ω_{sh}	[rad/s]	Rotational speed of slow shaft, see Figure 4.
ρ	[kg/m^3]	Air density.

TERMINOLOGY

In the description of the blade properties (stiffnesses, deformations, velocities) the directions are described among others with the following terms.

Expression	Description
edge-wise	Along the local chord.
flap-wise	Perpendicular to the plane of rotation, positive downwind.
flat-wise	Perpendicular to the local chord, positive downwind.
lead-wise	In the plane of rotation, positive in rotational direction (opp. to 'lag-wise').

1 INTRODUCTION

The computer program PHATAS –"Program for Horizontal Axis wind Turbine Analysis and Simulation"– is developed for the time-domain calculation of the dynamic behaviour and the corresponding loads in the main components of a Horizontal Axis wind Turbine, 'HAT' .
The program PHATAS-IV is in particular capable for the analysis of offshore turbines, by having an improved model for tower dynamics which allows wind and wave loading.

Modifications of release "APR-2005" with respect to the former release "NOV-2003" are:

- Write the shear forces in the blade root to the binary datafile;
- Add some inertia properties of the nacelle;
- Add the bending stiffness of guy wires (-bars) of the tower model.

Modifications of release "NOV-2003" with respect to its predecessor "OCT-2002" are:

- Improvements on the implementation of the tip-loss factor in the B.E.M. solution;
- Allow blades with a lag-wise pre-bend ('aft-swept') shape, in addition to flap-wise;
- Add transmission loss to the table with generator torque as function of rotor speed;
- A straightforward P-D pitch controller for variable speed wind turbines;
- Model for rotor speed sensor by pulse-counting, with eccentricity;
- Options to simulate starts and faulted conditions following IEC;
- Making the program more robust, such that it can be applied for load-set calculations.

In the remainder of this manual PHATAS release "APR-2005" (or "NOV-2003") is simply referred to as 'PHATAS'.

Chapter 2 of this manual contains a global description of the functionality of PHATAS, and the structure of the input and output files with relation to the other programs.

The input variables of PHATAS are described in chapter 3, while the definitions of all output properties are described in chapter 4. The modular structure of PHATAS with respect to the tower model allows separate development of other tower models that may be linked to the code. For the current PHATAS version three tower models are available for which a description of the input and output is given in chapter 5.

In general a wind turbine controller has several options and can be rather complicated. The structure of the controller available in PHATAS was organised such that it has a high level of agreement with most of the modern large size wind turbine controllers. This controller with the options to simulate faulted conditions following the IEC requirements is described in chapter 6. The PHATAS input files for IEC load cases can be generated fast using the load-case preprocessor *lcprep* that is described in Appendix B. These load-case input files apply to release "NOV-2003" and "APR-2005".

Instead of using the algorithms that are present in PHATAS (such as the P-D controller) the user can also link its own routine, provided that a FORTRAN compiler is available. For this purpose routines are available in which e.g. the pitch angle, generator torque, and yaw actions can be modelled. The structure of the solution process in terms of subroutines and some include files with common blocks are described in chapter 7, which gives an example of a yaw controller.

Finally chapter 8 contains some remarks on the limitations of PHATAS and on some numerical aspects of the calculations.

Appendix A contains a description of some tools provided together with PHATAS that can be used for selecting properties from the quasi-steady turbine characteristics, and for integrating the annual power production.

Appendix B contains a description of the **L**oad **C**ase **Prep**rocessor *lcprep* that is provided to generate the PHATAS input files to perform dynamic load calculations following the IEC [8], G.L. [5], or NVN [17] design recommendations. These load-case input files apply to release "NOV-2003" and "APR-2005".

For retrieving time series of the dynamic properties from the binary datafiles generated with PHATAS, the postprocessor *phpost* is provided. A description is given in Appendix C.

For more specific processing of the PHATAS output, the tools *loadex* and *bin* are provided. Descriptions of *bin* and *loadex* are given in Appendix D and E. With *bin* the time series can be sorted on basis of the rotor azimuth, also called 'bin analysis'. The tool *bin* can be applied to PHATAS output time series as well as to general ASCII time series, as long as the rotor azimuth is in any of the columns. The tool *loadex* can be used to find the extreme sectional loads in the blade, rotor shaft, or tower for a complete load set. In addition to this also the sectional loads sorted on basis of the Time-At-Level of the bending moment and/or torsional moment can be generated. The Time-At-Level results of the bending moment can be used for the design of bearings, while the Time-At-Level results of the torsional moment can be used for the design of the yaw or pitch drive.

For the development of postprocessing tools with specific requirements on the (combinations of) output properties or output format, the content of the binary datafile of PHATAS is described in Appendix F.

2 USING THE PROGRAM

2.1 General

The command for invoking the 'standard' PHATAS executable is `phatas` followed by up to three `CHARACTER*64` command line arguments for the names of files with:

1 Input specifications ('phatinp');

2 Either stochastic SWIFT wind [25], an ASCII wind time series,
or a wind field from WAKEFARM output [24];

3 Dynamic wave time series as generated with ROWS [3] or *Streamfunction* [18].

If no arguments are given the program reads the input from file 'phatinp'. If only one argument is given and the input file includes specifications to read wind c.q. wave velocities and accelerations from a file, these file-specifications will be valid.

In chapter 3 the description of the input variables is given for each of the (SUB)MENUs.
When many load cases are to be analysed for the same configuration one may use PHATAS as one of the wind turbine design tools under the 'driver' ProgSeq [6] or with the tool FOCUS developed at the Knowledge Centre WMC. The global "data flow diagram" for the design tools under ProgSeq is shown in Figure 1.

The source code of PHATAS is written in the FORTRAN F77 programming language and has the following three origins:

- The main program and the input structure; that are generated with the tool GENMENU. This input structure contains the name of the default input file, the include files with input properties and the routines for reading the input files.
- The part for the calculations on the wind turbine dynamics of which the subroutine on the highest level is '*phmain*', see Figure 23.
- The part in which the mode shapes and the natural frequencies of the tower model are solved and in which also the dynamic response is calculated. This part is developed separate from the PHATAS code and can be replaced on a modular basis.
For the current PHATAS version, 2 tower models are available and described in chapter 5.

When using PHATAS one is not aware of these different parts of the code.

2.2 Input Files

The data flow diagram with the files that are directly related to PHATAS is presented in Figure 2. This data flow diagram does not show input files that are read from the user-reserved routines '*conrol*', '*tipsol*', '*torgen*', '*upyaw*' or their initialisation routines, see chapter 7.

The calculation of the design loads of a wind turbine deals with several load cases and various types of input data, for which the pre-processor *lcprep* has been developed, see Appendix B.

2.2.1 Menu default file

The definition of the input data structure, the comments, and the default values of the input variables are stored in the default file named 'defphat'.
Most of the structural dynamic properties in this default file have a neutral or a "most meaningless" value, while the values of some geometric properties describe the former 25m HAT rotor test facility of ECN, Petten. This default file also contains the default name of the input file which is 'phatinp'. *The default file 'defphat' should not not be modified!*

2.2.2 Turbine input file(s)

When executing PHATAS the default file 'defphat' is read first after which the file indicated by the first command line argument is read as input file. An additional file with input specifications can be used by specifying its name using the input item '**FILE**' in **MENU menu_param**, see section 3.1.
It is recommended to use the file with specifications of each of the load cases as command line argument when invoking PHATAS. The file with of geometric and structural turbine specifications must be referred ('included') then in each 'load case' file using the '**FILE**' input item.

2.2.3 Tower input file

The model for solving the dynamic response of the tower in PHATAS is stored in a separate module. Because different tower models can be developed within this modular structure the input for the tower dynamics can have different forms. This implies that the input properties for the tower dynamics depend on the tower model, and are thus read from a separate file to be specified in the PHATAS input.

2.2.4 Input file for elastic tailored blades

For investigations into bending-torsion coupling of rotor blades by means of elastic tailoring, the model for blade torsional deformation has terms for coupling from the tensile force and from the blade bending moments. The coupling-terms are defined following the theory of Karaolis (later used by Kooijman [10]) and are read by PHATAS from file '*karaolis.dat*'.
If the file '*karaolis.dat*' is not present, the coupling terms are set to zero.
If a file '*karaolis.dat*' is present, a table is read with in the first column the spanwise location measured from '**blade_root_radius**' (see Figure 7) and in the next three columns the values of 'h_{41}', 'h_{42}', and 'h_{43}' for torsion due to tension, flap-bending, and lead-bending respectively. For the definition of these terms, see the description of 'CROSTAB', Lindenburg [13].
In the current PHATAS releases only the torsional deformation due to a bending moment is modelled and not the bending deformation due to blade torsion. This means that one cannot fully rely on PHATAS for calculation of the aeroelastic stability of tailored blades.

2.2.5 Input file with blade damping properties

Within the joint European research project DAMPBLADE, investigations were done into materials and the structural layout of rotor blades that provide enhanced damping behaviour. Within this project the spanwise distribution of the damping properties for rotor blades had to be used in analyses with PHATAS. For this purpose an algorithm was added that reads the spanwise distribution of the damping for blade bending from the file named ’dampbeam.dat’. If the file ’dampbeam.dat’ is not present the existing input items ’**flat_damping**’ and ‘**edge_damping**’ are initialised by PHATAS with the dimensionless damping properties of the first bending modes.

2.2.6 Airfoil coefficient file(s)

For each of the airfoils used to describe the aerodynamic loads, the dimensionless aerodynamic coefficients have to be available in PHATAS. Because the list of airfoil coefficients may be long and because they only depend on the type of airfoil and on the Reynolds number, the aerodynamic coefficients are read from files of which only the names need to be specified.
The files with airfoil data must contain tables with four columns containing:

1 The aerodynamic angle of attack [^{o}].

2 Lift coefficient $= \frac{\text{lift force per unit span}}{1/2 \cdot \rho \cdot V_{\text{wind}}^2 \cdot chord}$.
The lift force is the aerodynamic force component perpendicular to the relative airflow.

3 Drag coefficient $= \frac{\text{drag force per unit span}}{1/2 \cdot \rho \cdot V_{\text{wind}}^2 \cdot chord}$.
The drag force is the aerodynamic force component in the direction of the relative airflow.

4 Moment coefficient $= \frac{\text{moment per unit span}}{1/2 \cdot \rho \cdot V_{\text{wind}}^2 \cdot chord^2}$.
The aerodynamic moment is the ‘nose up’ moment (opposite to the pitch direction, Figure 8) about the point on the blade chord specified by **aero_centre** in table **blade_geometry**, see section 3.6. In general ‘**aero_centre**‘ is located at 25% chord length behind the leading edge.

The aerodynamic airfoil coefficients must be specified for angles of attack ranging at least from zero-lift to 90^o. If the coefficients at large angles of attack (in deep stall) are not available while the airfoil geometry is known, the program *StC* (see chapter 2 of [15]) can be used to generate the aerodynamic coefficients for all angles of attack. Each airfoil datafile may contain the coefficients for a maximum of 150 angles of attack. A file with airfoil coefficients is read as long as **1**) the file does not end, **2**) the number of coefficients is smaller than 150, and **3**) the subsequent angles of attack form an increasing series.
Comment lines can be written as heading above the table with coefficients and should start with one of the characters ’#’, ’$<$’, or ’/’. Comments can also be written after the end of the coefficients file for which no ’comment-line’ characters are required.

2.2.7 Wind datafile

In case the wind is read from a file with ASCII time series, from WAKEFARM output [24], or from a SWIFT wind-data file [25] the name of this file needs to be specified as second command line argument or in the PHATAS input file. The value of the second command line argument (if given) overwrites the name of **wind_data_file**, see section 3.7.

2.2.8 Wave datafile

For offshore wind turbines the water level and also the wave kinematics can be applied from a file with wave data. The name of this file can be specified in the input file of PHATAS, but is overruled by the value of the third command line argument (if given). The routine for the tower dynamics, see chapter 5, uses the water level from this file to calculate the "wet" tower bending frequencies. Finally the routine for the dynamic response of the tower uses the wave loads (drag and inertia terms) for the modal loading and damping from the water.

2.3 Modes of Application

The program PHATAS can be used to calculate either **1**) the aerodynamic rotor characteristics, **2**) a power curve, or **3**) the non-linear dynamic response in time. This mode of application depends on the input items **time_history_flag** and **delta_lambda**, see MENU **job_param** and SUBMENU **turbine_coeff** in section 3.2. In the code the main branches for these modes of application are in routine '*phmain*', see also the 'Program Structural Diagram' in Figure 23.

2.3.1 Aerodynamic rotor characteristics

The calculation of aerodynamic rotor characteristics is invoked by the input specification '**time_history_flag OFF**' in combination with a positive value for **delta_lambda**.

For each of the blade pitch angles specified with array **pitch_series** (SUBMENU **turbine_coeff**) the aerodynamic rotor characteristics are calculated for a sequence of tip speed ratios. For each value of the pitch angle and tip speed ratio, the dynamic response is calculated until the difference in aerodynamic power and rotor speed between 2 subsequent revolutions is less than 0.5%.
For calculation of the rotor characteristics, the turbine geometry is included completely while the structural dynamic aspects of the rotor are described following the specifications in the input file. This also holds for the model of rotational effects ('3D correction' on lift) and for the dynamic stall model. For this mode of application, the wind speed is constant in time (although shear and misalignment can be modelled), the control algorithm is not active, the generator speed (expressed at the rotor shaft) has the constant value '**rated_rotorspeed**', and yawing (free or controlled) is not modelled.

2.3.2 Power curve calculations

The calculation of an aerodynamic power curve is invoked by the input specification '**time_history_flag OFF**' in combination with a negative value for **delta_lambda**.

In this mode the quasi-steady performance of the turbine is calculated for an increasing series of wind speed values. Compared to the calculation of aerodynamic rotor characteristics, the generator characteristics are included, if present the pitch control is included and/or modelled, and the yaw motion is solved. If no pitch algorithm is linked and '**control_type 0**' is specified the blade pitch angles have the (initial) value as specified by **pitch_angle** in SUBMENU **time_history**.

The output files from the calculation of the power curve and the aerodynamic rotor characteristics have the same format and can both be processed with the tool '*selchar*', see Appendix A.

2.3.3 Dynamic response

The calculation of the dynamic response in time is invoked by '**time_history_flag ON**'.
For this mode all specifications in SUBMENU **turbine_coeff** can be omitted. For some degrees of freedom (e.g. yaw angle, rotor azimuth and -speed, pitch, teeter) the initial values can be specified in SUBMENU **time_history**.
The initial values for teeter motion and tower torsion c.q. free yawing are set to steady equilibrium values. For the remaining degrees of freedom, such as blade bending, a quasi-steady equilibrium state is calculated at $t = 0$s. For the initial equilibrium solution all first and second time derivatives of the degrees of freedom are set to zero, except for e.g. the teeter rate and the rotor speed. For the rotor speed the initial rotor angular acceleration is solved such that equilibrium of rotor and drive train torque is satisfied.

(Branches in the source code for the initial solution at t=0s
are present in routine '*phiter*' and in its subroutines, see Figure 24
and use the INTEGER variable for the time step number `istep` .)

2.4 Output Files

The data flow diagram in Figure 2 shows that PHATAS generates several output files.

2.4.1 File 'STATUS'

Execution of the program PHATAS does not start in case of missing or incorrect input files and in case of syntax errors in the input. Messages related to these errors are written in the file named 'STATUS'.
In case of errors the '*status variable*' is set to 1, otherwise the '*status variable*' is set to 0.

2.4.2 File 'RESTART'

After reading the default file 'defphat', the input file (default name: 'phatinp') and –if used– additional input files, all input specifications with the values that are finally used are written in the file named 'RESTART'. The file 'RESTART' can be used as input file for PHATAS.

2.4.3 Message file

The file '**message_file**' is used for messages from checks on the input variables, for messages from the controller, and from run-time messages e.g. in case of bad convergence, see section 3.2. The controller messages deal with changes of the state of the turbine controller, such as activation of a brake, or generator (dis-)connection. At the end of a dynamic response calculation, PHATAS writes some statistical properties to this message file, such as average power, maximum, minimum and average rotor speed, and extreme blad-tip deformations.

It may still occur that the execution of PHATAS interrupts (mostly for numerical reasons) in which case PHATAS stops with the status variable set to 2.
After a successful calculation PHATAS stops while setting the status variable to 0.

2.4.4 Model file

When specifying '**output_level 2**' or higher in the PHATAS input file the overall properties of the turbine model are written in the file 'model file' specified by **model_file**, see section 3.2. The contents of this file include:

- Integrated mass properties of the rotor and the blades;
- System properties (mass, stiffness, damping, limits, ...) for all degrees of freedom;
- Properties of the elements or of the nodes of the blade model;
- Bending stiffness properties of blade elements in rotor plane directions;
- Mode shapes and frequencies of the blades and of the turbine for a number of rotational speeds, which are eigenvalue solutions without aerodynamic contribution;
- A table which can be used to plot the geometrical contour, including the mass centre line and the 25% chord line.

2.4.5 Output file

Output from the PHATAS calculations is written to ile '**output_file**', see section 3.2.
For the calculation of the non-linear dynamic response in time the output file contains a table with time series of properties that are to be selected in the input file, see chapter 4.
For the calculation of aerodynamic rotor characteristics or a power curve this file contains (for each pitch angle) a table with rotor characteristics as a function of tip speed ratio.

2.4.6 Binary datafile

When using PHATAS for calculation of the non-linear dynamic response in time, a binary datafile can (optionally) be written. The time series of nearly all output properties can be retrieved from this file using the postprocessor '*phpost*', see Appendix C.

A description of the contents of this binary datafile is given in Appendix F.

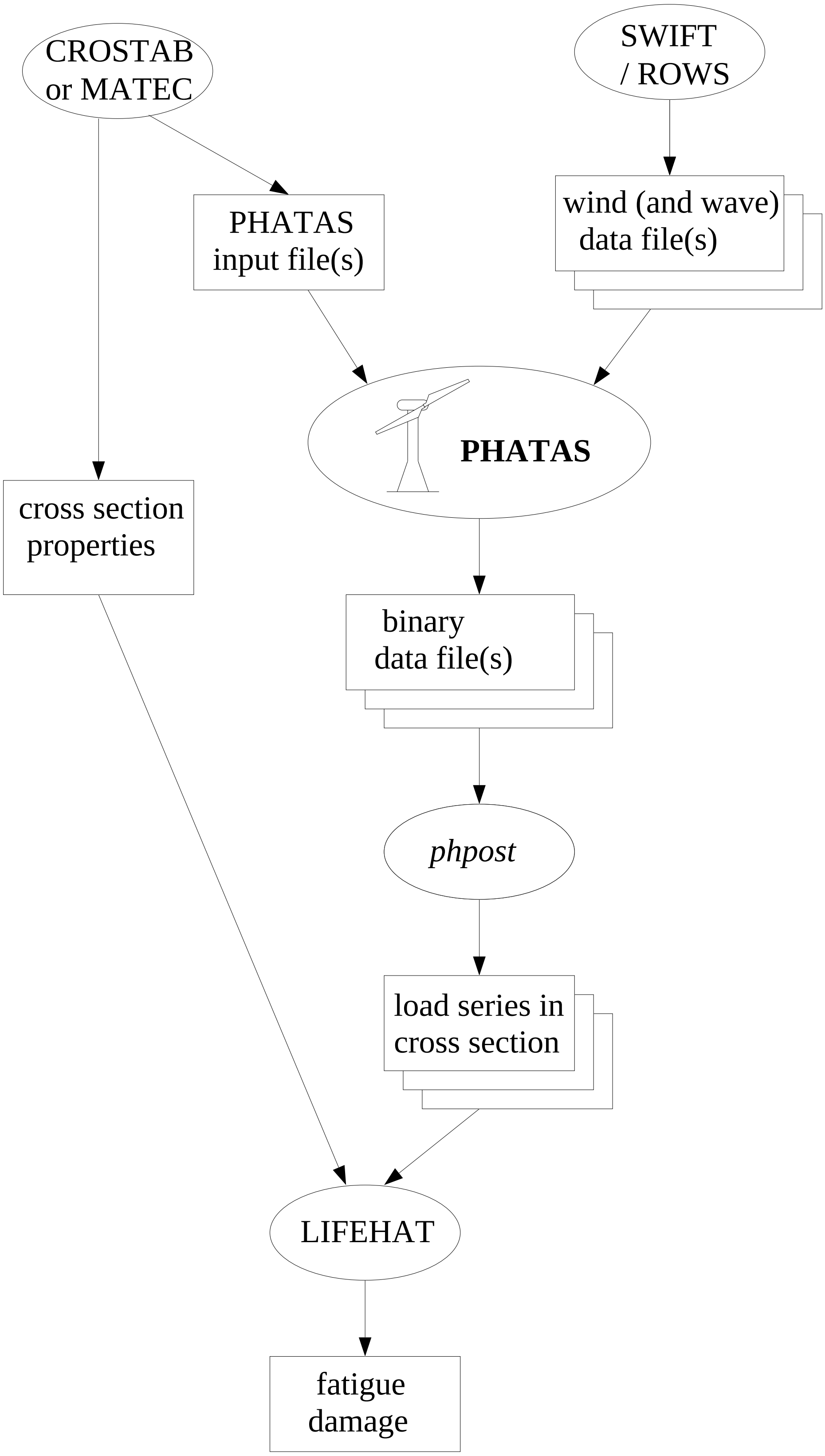


Figure 1: Data flow diagram of tools within ProgSeq

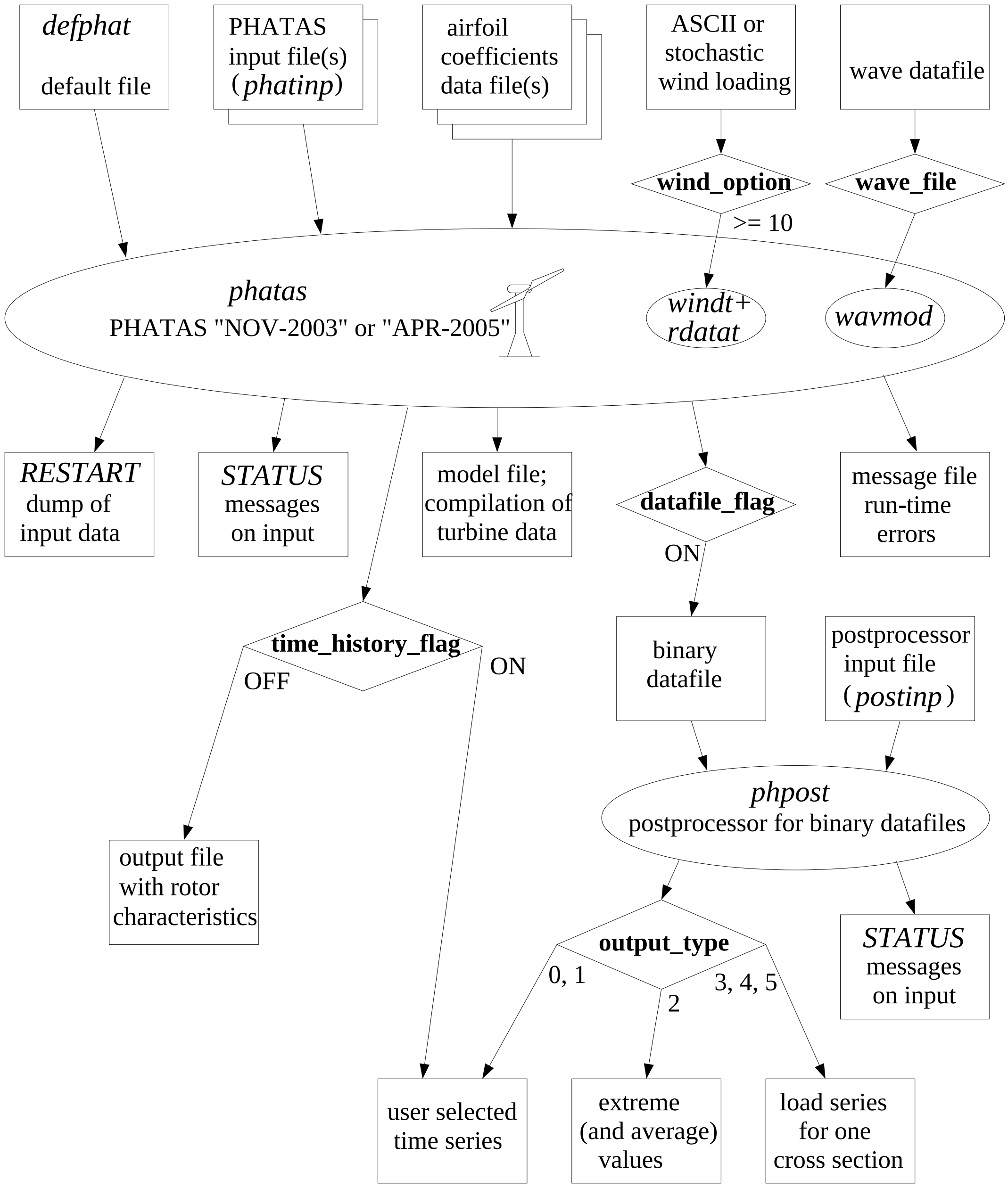


Figure 2: PHATAS file structure

3 DESCRIPTION OF INPUT VARIABLES

This chapter gives a description of the variables of the turbine input file(s) for PHATAS release "NOV-2003" and "APR-2005". Because of the modular structure of the code for the tower dynamics, the input items of the tower are described separately in chapter 5.

Modifications of the input values with respect to the default file 'defphat' can be written in the input file(s). The specifications in the default file are read first. The names of all files from which input specifications are read as well as messages from syntax checks are listed in file 'STATUS'.

A description of the syntax of the specifications in the input file(s) is given below.
The parts of the format description written between square brackets [ ] are optional.

1 The input specifications are read as strings with a length up to 80 characters.
Lines with specifications longer than 80 characters are partially read and may lead to erroneous input or syntax errors. These errors are reported in file 'STATUS'.

2 Specification of a MENU: **MENU** *keyname*$_{\mathrm{MENU}}$
After the specification of a MENU its input items can be specified.
For the specification of items in another (SUB)MENU the (SUB)MENU keyname needs to be specified first.

3 Specification of a SUBMENU: **SUBMENU** *keyname*$_{\mathrm{SUBMENU}}$
After this specification, the SUBMENU items can be specified.

4 Assigning a value to a single item: *keyname*$_{\mathrm{item}}$ *value*
When a specified value is of a wrong type a message is written to file 'STATUS'.

5 Modification of an array:
$keyname_{\mathrm{array}}$
$[value[,\ value\ .....],]$
$[value[,\ value\ .....],]$
.....
$value[,\ value\ .....]$

The array is specified starting from the first element. Multi-dimensional arrays are modified starting with columns, which means that the first index varies first, next the second index etc.
A comma "," is the delimiter for the specified values of subsequent array elements. Spaces may be written between the specification of different elements. TABs however are not allowed!
A comma ending a line in the input file(s) indicates that more specifications of the array elements follow on the next line. A *value* not followed by a comma indicates the last specified array element.

6 Skipping the specification of elements of an array (example):
$keyname_{\mathrm{array}}$
$value,\ *,\ value\ .....,\ value$
The "$*$" indicates that the array element (the 2^{nd} in this case) is not modified.

7 Assigning values to specific array elements:
$keyname_{\mathrm{array}}$
$(index[, index\ .....])\ value[, value\ .....]$
The array elements are modified starting from the element indicated by $(index[, index\ .....])$.
The optional index needs to be used for the specification of a multi-dimensional array.

8 Modification of the elements of a *table*:

$keyname_{\rm table}$
$[value\ value[\ value],]$
$[value\ value[\ value],]$
.....
$value\ value[\ value]$

The values of the arrays in the table are modified starting from the first element. Every row may only contain values that refer to that specific row. Each line can contain not more than the number of rows in the table. A space " " is the delimiter for the specified values in the different rows. A comma "," indicates that more modifications follow on the next line. An empty line after a specified *value* indicates that the modification of the table is finished.

9 Skipping table elements in a specification (example):

$keyname_{\rm table}$
$value * value$

The element in location "$*$" will not be modified.

10 Modification of specific table elements (example):

$keyname_{\rm table}$
$(index[, column])\ value\ value[\ value]$

The elements in the table are modified starting from $(index[, column])$.
Because tables only contain one-dimensional arrays the specification of the element needs only one *index*.

11 Specification of files with additional modifications: **FILE** *filename*
After specification of **FILE** under MENU **menu_param** the file with the name *filename* is opened. Reading the input specifications continues from this file, after which the remaining specifications in the current file are read. Using additional input files makes it possible to have a "baseline" input file with general properties of the turbine for a set of calculations and refer to this file from each input file that contains the specifications for the calculation of one load case.

12 Adding comments. Comments in the input files must be preceded by the character "$<$". Lines starting with "$<$" and blank lines are skipped. Comments between the keyname and the value of an item are not allowed!

The following sections contain a description of the input variables "items" following the MENU structure. For each (SUB)MENU item the description contains:

- the **keyname** in **bold face** ;
- the `type` in `typewriter font` in accordance with the FORTRAN variable type ;
- the (default values) between ordinary ( ) brackets ;
- if applicable, the [unit] between square [ ] brackets ;
- if applicable, the $<$allowed range$>$ between angled $<>$ brackets ;
- an explanation in normal font .

3.1 MENU menu_param

MENU **menu_param** contains some general information about the input of the program.

Screen_option `LOGICAL` (OFF). For UNIX-versions only!
For '**Screen_option ON**' PHATAS displays the menu structure on the screen for interactive modification of the input data. This interactive mode is a guide for the user through the menu structure with comments on the input variables.
For '**Screen_option OFF**' PHATAS starts calculating immediately after reading the specifications from the input file(s).

Default_file `CHARACTER*20` ('defphat')
Name of the file with default PHATAS specifications. Specification of this filename has no effect since the **Default_file** is already read at the time of reading the input files.

Input_file `CHARACTER*20` ('phatinp')
Default name of the PHATAS input file.
For the same reason as for **Default_file** specification has no effect.

FILE `CHARACTER*20`
Name of an additional input file. The input data in an additional input file are read as if this file is inserted at the place of its specification. The 'FILE' reference can be nested.

In the case of interactive input (for UNIX only), another input file can be specified in MENU **menu_param** as **Input_file**. The value for **Input_file**, 'phatinp', is read from file 'defphat'.

If PHATAS is invoked with (`CHARACTER*64`) command line arguments the first argument is read as name of the input file, the second argument is read as the name of the wind datafile, and the third argument is read as the name of the wave datafile, see section 3.8.

If the input file(s) contains multiple specifications of a variable the last value is assigned to that variable. Knowing that different input files are used for calculating the dynamic response of each load-case, which all use the same turbine description, it is recommended to use the '**FILE**' specification of the turbine data at the beginning of each load-case input file.

3.2 MENU job_param

MENU **job_param** contains general information of the calculations.
The variables of this MENU are stored in include file 'PNCLM2'.

model_identif `CHARACTER*20` ('PHATAS_default')
Identification of the turbine configuration which is written at the beginning of the files '**output_file**' and '**model_file**'.

time_history_flag `LOGICAL` (OFF)

= **ON** Calculation of the dynamic response in time domain.
Initial values for the calculation of the dynamic response are described in 'SUBMENU **time_history**'.

= **OFF** Calculation of the aerodynamic rotor characteristics or a power curve, depending on the value of '**lambda_increment**' in 'SUBMENU **turbine_coeff**'.
The conditions for this calculation are described in 'SUBMENU **turbine_coeff**' and in 'SUBMENU **time_history**', see also section 2.3.

message_file `CHARACTER*20` ('phatdef.err')
Name of the file for messages on the values of the input variables, messages (from controller) on the state of the turbine, messages on bad convergence, and overall statistics of the complete calculation.

model_file `CHARACTER*20` ('phatdef.mdl')
Name of the file in which the properties of the model are written.
This file is only created if **output_level** is **2** or larger.

output_file `CHARACTER*20` ('phatdef.pht')
Name of the file with selected PHATAS output.

comment_mark `CHARACTER*20 (' ')`
Variable of which the first character is written in the first column of the heading of the files **output_file** and **model_file**.

output_level `INTEGER` (2) $<0, ... >$
Indicates the amount of information that is written during execution.

- $=$ **0** Output is only written to file '**output_table**' and file '**binary_datafile**' with all run time messages and job-progress to screen suppressed.
- $>=$ **1** As for '**output_level 0**' including job-progress to the console (screen) and run time messages written to file '**message_file**' if a structural dynamic degree of freedom did not converge.
- $>=$ **2** Writes general properties of the turbine and blade model including natural frequencies to file '**model_file**'.
- $>=$ **3** If a SWIFT or WAKEFARM wind file is used and if wave loading is applied, the global properties of the wind or wave distribution are read from the heading of the datafiles and are written to file '*STATUS*'. The job progress is echoed for some more time steps.
- $>=$ **4** For each time step the initial guess and the last value of the degrees of freedom are written to file '**message_file**'. The job-progress is echoed to the screen for each time step. For the calculation of aerodynamic rotor characteristics, also the aerodynamic coefficients for each of the elements are written.
- $>=$ **5** As for '**output_level 4**' including the values of the degrees of freedom for each iteration and messages for bad convergence (if so) of the iteration for the axial induction factors. Because the induction factors are calculated in the "inner" iteration of the algorithm the messages for '**output_level 5**' may result in a large file.

Recommended values are **1** (no file '**model_file**') or **2** (write file '**model_file**')
Values of **output_level** larger than **3** may be used for debugging actions.

time_increment `REAL` (0.01) [s] $<t\_1\mathrm{rev}/60000, ..., t\_1\mathrm{rev}>$
Time increment in the calculation.
A recommended value for **time_increment** is $0.27/\nu$ where ν is the highest frequency [rad/s] that is calculated with a numerical damping of maximum 1%.
When a dynamic stall model is used (**dynamic_stall** $>$ **0**) the time increment must be smaller than the time in which the relative airflow at e.g. 80% span travels half the chord length: **time_increment** $< 0.5\, chord/\,(0.8\,\Omega\, R)$.
A smaller '**time_increment**' results in a larger computing time.

3.2.1 SUBMENU turbine_coeff

SUBMENU **turbine_coeff** contains variables for the calculation of quasi-steady turbine characteristics for a constant wind speed. This means that the specifications in this SUBMENU are only used for '**time_history_flag OFF**'. The turbine characteristics are calculated as average of the calculated response over one revolution. These calculations are performed with a time increment specified with '**time_increment**' in MENU '**job_param**'.
The variables in SUBMENU '**turbine_coeff**' are stored in include file 'PNCLS2'.

The input items are:

delta_lambda `REAL` (1.0) [.]
: Increment in tip speed ratio λ (or wind speed value) for the calculation of quasi-steady characteristics. A negative value of **delta_lambda** indicates that turbine characteristics are calculated for wind speed increments in [m/s].

lambda_min `REAL` (1.0) [.] $<$0.5, ... $>$
: Smallest tip speed ratio λ for the calculation of turbine characteristics:
The tip speed ratio is $\lambda = \Omega \cdot (R \cdot \cos\alpha_{\rm c})/V_{\rm steady}({\rm hub})$.
For a negative value of **delta_lambda**, '**lambda_min**' is the smallest value of the wind speed at the hub, $V_{\rm steady}(Y_{\rm hub}, Z_{\rm hub})$ [m/s].

nr_lambda `INTEGER` (15) $<$1, ... $>$
: Number of increments for λ or wind speed in the calculation of turbine characteristics.

pitch_series `REAL(1:40)` (0.0, 5.0, -180.0, -180.0, ...) [deg]
: Only if '**delta_lambda**' is positive, a table with aerodynamic rotor characteristics is calculated for each of the pitch angles indicated. A value smaller than- or equal to -90.0 indicates the end of the specified pitch angles.

The quasi-steady turbine characteristics can be calculated in two ways depending on the sign of the input item '**delta_lambda**':

Positive: specifies lambda increments for the calculation of aerodynamic rotor characteristics.
: In this case a table with rotor characteristics is calculated for each of the pitch angles specified by '**pitch_series**'. This means that the control algorithm '*conrol*', see chapter 6, is *not* referred ('called'), the rotor speed is kept constant at '**rated_rotorspeed**', and the yaw motion is not modelled.

Negative: specifies wind speed increments for a power curve calculation.
: In this case the quasi-steady performance of the turbine is calculated including the algorithm in routine '*conrol*' (standard or user defined). The values of '**pitch_angle**' in SUBMENU '**time_history**' are used as initial value for the blade pitch angles. As far as modelled or specified in the input, the pitch algorithm is active, the rotor-speed is solved according to the generator model, and the yaw motion (free or controlled) is solved.

3.2.2 SUBMENU time_history

SUBMENU **time_history** contains specifications for the calculation of the non-linear dynamic response while some initial values (such as **pitch_angle**, **rotor_speed** etc.) are also used for a power curve calculation.

The variables in this SUBMENU are stored in include file 'PNCLS1'.

maximum_time `REAL` (1.0) [s]

Time limit for the calculation of the non-linear dynamic response.

output_table `table(`**signal_nr , location**`)`
$$\begin{pmatrix} 2 & 0.0, \\ 5 & 0.0, \\ 12 & 0.0, \\ 999 & 0.0, \\ .... & .... \end{pmatrix}$$

Table with specifications of selected output properties.
This table only needs to be specified for '**time_history_flag ON**'.
The columns contain the following arrays:

signal_nr `INTEGER(1:22)` Numbers indicating the output properties.
The default signal numbers 2, 5, and 12 indicate wind speed, generator power, and rotor speed respectively. An explanation of these numbers is given in chapter 4.
A **signal_nr** of **999** indicates the end of the specified output properties.

location `REAL(1:22)` [m] Coordinates of the properties, if '**signal_nr**' refers to a property for a cross section of one of the main components:

blade numbers 102 to 179, 202 to 279, or 302 to 379:
Spanwise coordinate measured from the blade root (see **blade_root_radius**)

rotor shaft or **drive train** numbers 51 to 71:
Position on the rotor shaft measured downwind from the rotor centre.

tower numbers 21 to 49: Height above the tower-base or foundation.

table_increment `INTEGER` (1) $<1, ... >$

Increment in time steps for which the output is printed.

datafile_flag `LOGICAL` (OFF)

If 'ON' a binary datafile is generated. The binary datafile contains all properties of the wind turbine and its response with all the blade moment distributions (in compressed form). From this file the time series of most output properties can be retrieved with the postprocessor, see Appendix C.

datafile_name `CHARACTER*20` ('phatdef.tbh')

Name of the binary file to which PHATAS writes the dynamic response.

data_increment `INTEGER` (1) $<1, ... >$

After every '**data_increment**' time steps data are written to the binary output file.

pitch_angle `REAL(1:3)` (0.0, 0.0, 0.0) [deg]

Initial values of the pitch angle of the blades. '**pitch_angle**' is positive if it reduces the angle of attack ("nose-down") Figure 8. Note that the values of '**pitch_angle**' are also used as initial values for the calculation of aerodynamic power curves, section 2.3.2.

rotor_azimuth `REAL` (0.0) [deg]

Initial value of the rotor azimuth. '**rotor_azimuth**' is zero if blade 1 has the "12 o'clock" position and is positive for clockwise rotation, looking along the rotor axis, Figure 3.

rotor_speed REAL (0.0) [rpm]
For **generator_model** > 0 (SUBMENU **generator_data**) or **generator_cut_off** ≤ 0.0 (see SUBMENU **control**): Initial value of the rotor rotational speed.

teeter_angle REAL (0.0) [deg]
Initial value of the teeter angle, measured positive if blade 1 has moved downwind, Figure 3 (assuming '**delta_3_angle**' $< 90^o$; MENU **teeter_data**).

teeter_velocity REAL (0.0) [deg/s]
Initial value of the first time derivative of the teeter angle.
For '**teeter_flag OFF**' the values of **teeter_angle** and **teeter_velocity** are set to zero.

yaw_angle REAL (0.0) [deg]
Initial yaw angle, positive if the nacelle is yawed from north to west, see Figure 10.
The initial yaw angle is also used in the standard controller as 'yaw tracking error'.

3.2.3 SUBMENU control

SUBMENU **control** contains input items for a P-D pitch-controller of the power or rotor speed, that can be used in the calculation of the dynamic response, or of a power curve. This SUBMENU also contains input items for start and stop actions, generator disconnection, and the simulation of faulted conditions such as required for design load calculations, see Appendix B. The variables in this SUBMENU are stored in include file 'PNCLS3'.

Following items enable the modelling of stop actions either at a specified time '**brake_time**' or after exceeding a specified rotor speed '**brake_overspeed**' which may be triggered after generator disconnection at '**generator_cut_off**'.
Stop actions can be a combination of brake activation, pitch rate, and yaw action and are only used for the calculation of the dynamic response: '**time_history_flag ON**'.

brake_time REAL (99000.0) [s].
Time at which the stop action (brake and pitch) is activated. In the program PHATAS this time is set to the current time if the rotor speed exceeds '**brake_overspeed**', if the generator is disconnected during a calculation, or if the difference between the pitch angle of any of the blades exceeds '**max_pitch_error**'.

generator_cut_off REAL (99000.0) [s].
Time at which the generator is disconnected. For idling or a start this should be negative.

brake_overspeed REAL (99000.0) [rpm].
Rotor speed (of slow rotating shaft) at which a stop action is triggered.

brake_torque REAL (0.0) [N·m].
Full torque of the mechanical brake measured at the generator shaft, Figure 5.

brake_delay_time REAL (0.0) [s].
Time between triggering a stop action and activation of the mechanical brake c.q. pitch.

brake_ramp_time REAL (0.0) [s].
The brake torque increases linearly from zero at '**brake_time**' to '**brake_torque**' during '**brake_ramp_time**'.

brake_pitch_rate REAL (0.0) [deg/s].
Rate of pitch motion used for a stop action.

Following are the input items for the power or rotor speed controller, which are only used for calculation of the dynamic response or of a power curve. If the major input item **control_type** has a value zero the input items described below have no meaning, except for **pitch_error**.

control_increment `REAL` (0.01) [s].
Time increment for calling the control algorithm, see chapter 6.
This can also be used for user-written control routines, see chapter 7.

control_type `INTEGER` (0) $<$0, 1, 2, ...$>$.
For '**control_type 0**' the P-D controller is not called;
For '**control_type 1**' the power is controlled by pitch actions (*not fully tested*);
For '**control_type 2**' the rotor speed is controlled by pitch actions.
Higher values may be used for turbine-specific controllers (e.g. by a dynamic link library).

controller_target `REAL` (0.0) [W] or [rpm].
Depending on **control_type** this is the target power [W] or the target rotor speed [rpm]. For variable speed wind turbines a realistic value of **controller_target** is 2% to 3% above **rated_rotorspeed**.

pulses_per_rev `INTEGER` (900 000) $<$ 100, ... $>$.
Number of pulses on the ring for the rotor speed sensor as if mounted on the rotor shaft. For 100 000 pulses per revolution or more, the rotor speed signal is nearly continuous.

pulse_ring_eccentr `REAL` (0.0) $<$ -0.1, ..., +0.1 $>$.
Eccentricity of the pulse ring for the rotor speed sensor as if mounted on the rotor shaft. A value 0.001 (0.1%) already gives a small 1 per-rev disturbance.

average_time `REAL` (1.0) [s].
Time over which the proportional input is averaged and the differential input is calculated. In the PHATAS controller a 'running average' is calculated from $(1 - factor)$ times the current measured value (speed or power) plus $factor$ times the previous running average, where $factor$ is calculated as $1/\exp(\textbf{control\_increment}/\textbf{average\_time})$.
For variable-speed pitch-to-vane controlled wind turbines the disturbances from blade-tower interaction are filtered out if '**average_time**' is about the time for 1/6 of a revolution. '**average_time**' is also used as time-constant for the partial-load pitch actions.

proportional_gain `REAL` (0.0) [(rad/s)/W] or [(rad/s)/(rad/s)].
Gain factor related to the averaged/filtered power or rotor speed. For a variable-speed pitch-to-vane controlled 2.5MW wind turbine **proportional_gain** is in the order of 0.30.

differential_gain `REAL` (0.0) [(rad/s)/(W/s)] or [(rad/s)/(rad/s^2)].
Gain factor on difference of the power or averaged/filtered rotor speed over a time span of **control_increment**. For a variable-speed pitch-to-vane controlled 2.5MW wind turbine **differential_gain** is in the order of 2.0.

gain_scheduling `REAL` (0.0) [1/^{o}].
Decrease of gain and threshold value with pitch angle, applied as factor:
$1/(1 + \textbf{gain\_scheduling}\,(\theta_{\rm p} - \theta_{\rm p\,min}))$.
For pitch-to-vane controlled variable speed wind turbines a realistic value is 0.16.

pitch_time_lag `REAL` (0.1) [s].
Time constant in the (e.g. hydraulic or electric) pitch actuator. In PHATAS, the value of **pitch_time_lag** is minimised to half the value of **time_increment**.

threshold_value `REAL` (0.0) [deg/s].
Minimum value of the desired pitch rate for actuation in full load. The threshold aims to reduce the amount of pitch motions, while it adds some delay to the controller. For a large pitch-to-vane controlled variable-speed wind turbine a realistic value is 1.0^{o}/s.

control_pitch_rate `REAL` (0.0) [deg/s].
Maximum pitch rate for control during normal operation.

fuzzy_control_limit `REAL` (9.E+9) [W] or [rpm].
Power or rotor speed value above which the blades are pitched unconditional to vane, aiming to avoid over-power or over-speed.

fuzzy_pitch_rate `REAL` (1.0) [deg/s].
Pitch rate if power or rotor speed exceeds '**fuzzy_control_limit**' *and* is also increasing. A too high value of **fuzzy_pitch_rate** induces strong blade loads.

generator_emergency `REAL` (9.E+9) [W] or [rpm].
Emergency level for generator cut-off. For variable-speed pitch-to-vane controlled wind turbines the generator may be disconnected to avoid overloading or damage of the electrical conversion system.

pitch_error `REAL` (0.0) [deg].
Value with which the pitch angle of blade 2 (and 3) differ from that of blade 1.
For a positive '**pitch_error**', blade 2 has a larger (and blade 3 a smaller) pitch angle.

peak_shave_begin `REAL` (8.E+9) [W/rpm].
Power- or rotor speed value at which the linear! 'peak-shaving' pitch strategy start.

peak_shave_end `REAL` (9.E+9) [W/rpm].
Power- or rotor speed value of the end of the 'peak-shaving' interval, usually nominal-.

peak_shave_pitch `REAL` (1.0) [deg].
Pitch angle (minimum!) beyond **peak_shave_end**. The values of **peak_shave_begin**, **peak_shave_end**, and **peak_shave_pitch** have the purpose to reduce peak values of the (quasi-steady!) axial aerodynamic loads on the rotor. For large size wind turbines the peak-shaving pitch angle may be in the order of 5^{o} and issued over a rotor-speed range of about 6% of the rated rotor speed.

wait_time `REAL` (0.0) [s].
Time before simulation of a start or a fault, see **failure_type**.

failure_type `INTEGER` (0) $<0, 1, 2, 3, 4, 5>$.
Type of failure of the wind turbine, to be used for IEC load calculations:
- **= 0** No failure;
- **= 1** Pitch control of all blades is suppressed;
- **= 2** All blades pitch with '**start_pitch_rate**' (failed sensor);
- **= 3** Blade 2 does not pitch;
- **= 4** Blade 2 pitches (uncontrolled) with '**start_pitch_rate**';
- **= 5** Generator short-circuit, with '**maximum_torque**' (MENU **generator_data**).

All these failure modes are simulated after '**wait_time**'.

start_pitch_rate `REAL` (-1.0) [deg/s].
Pitch rate for start-up condition. This is active after **wait_time**.
The controller enters the 'START' mode, if the calculation starts with a disconnected generator (**generator_cut_off** < 0.0), a rotor speed smaller than 0.9 * **start_speed**, the initial pitch angleis larger than (0.6 * **min_pitch_angle** + 0.4 * **max_pitch_angle**).

The value of **start_pitch_rate** is also used to simulate pitch failure for '**failure_type 2**' and '**failure_type 4**'.

start_speed `REAL` (0.0) [rpm].
If the rotor speed exceeds **start_speed** and the controller is in the 'START' mode, the generator is connected and the controller gets in the mode for normal production. If the rotor is in the normal production mode 'NORM' and a stop is activated, then the generator is disconnected if the rotor speed gets below 0.9 times **start_speed**.

max_pitch_error `REAL` (180.0) [deg].
If the difference in pitch angle between any of the blades exceeds '**max_pitch_error**' (excluding '**pitch_error**') a stop action is initiated, which implies pitching the blades with '**brake_pitch_rate**' and applying the mechanical brake. If the turbine stops without mechanical brake, one should specify a zero value for **brake_torque** for the governing load-case.

3.3 MENU geometry

MENU **geometry** contains overall mass and geometric properties of the rotor and turbine.
The variables of this MENU are stored in include file 'PNCLM3'.
The input items **static_moment_z**, **rolling_inertia**, **tilting_inertia**, and **cross_inertia** were not present in the former release "NOV-2003" while **static_moment_x** was formerly named **nacelle_unbalance**.

tilt_angle `REAL` (0.0) [deg] Tilt angle of the rotor shaft, see Figure 3.

rotor_distance `REAL` (4.0) [m].
Distance from hub to tower axis measured along the rotor shaft, see Figure 3.

hub_location `REAL` (0.0) [m] Location of the hub above the tower top, see Figure 3.

rotor_eccentricity `REAL` (0.0) [m] $< -0.2 \cdot$ '**blade_span**', ..., $0.2 \cdot$ '**blade_span**' $>$.
Lateral distance between rotor axis and tower axis, positive if the rotor axis is left from the tower axis in front view, see Figure 10. This eccentricity can be used to eliminate the yaw misalignment of free yawing wind turbines resulting from the tilt angle component of the shaft torque.

cone_angle `REAL` (0.0) [deg].
Angle of the rotor blades with respect to a plane perpendicular to the rotor shaft, see Figure 3.

blade_root_radius `REAL` (0.0) [m].
Spanwise location of the blade root measured from the rotor centre along the blade axis, see Figure 7. When modelling blade deformation, this is described from '**blade_root_radius**' to the blade tip. For hinged blades the hinges are modelled at '**blade_root_radius**'.

nr_blades `INTEGER` (1) $<$1, 2, 3$>$. Number of rotor blades.

blade_span `REAL` (11.5) [m].
Length of the blades measured from '**blade_root_radius**' to the tip, see Figure 7.

blade_elements `INTEGER` (15) $<$3, ..., 24$>$.
Number of elements (of equal length) into which the part of the blades outside of '**blade_root_radius**' is divided, see Figure 7. The number of blade elements used for the aerodynamic BEM model is 1 larger than **blade_elements** because two aerodynamic annuli are used for the 'tip-element'.

pitch_location REAL (0.0) [m] <0.0, ..., '**blade_span**'>.
Spanwise distance from '**blade_root_radius**' of the pitch bearing.
In PHATAS the pitch bearing is modelled at the element intersection that is nearest to '**pitch_location**' but not beyond '**blade_elements**'-2, see Figure 7.
All elements outside of node '**pitch_location**' will rotate with pitch actions.

feather_axis REAL (0.0) [m].
Chordwise location of the blade pitch axis, measured from the blade axis to the trailing edge, see Figure 7 and 8. The item **feather_axis** has been used in the past for the classical stall controlled wind turbine rotors of which only the blade tips undergo pitch motions. Here **feather_axis** can be used to reduce the torsional moment in the tip pitch mechanism. The validity of '**feather_axis**' (non-zero values) is not verified recently.

radial_motion REAL (0.0) [m/rad].
Radial motion of the blade (-tip) connected with pitch rotation. This has been used for rotor concepts with passive blade tip control, driven by centrifugal loads.

hub_mass REAL (0.0) [kg].
Mass of the hub measured up to '**blade_root_radius**'.

hub_static_moment REAL (0.0) [kg·m].
First static moment of hub mass w.r.t. the rotor centre, positive downwind along the rotor axis.

hub_inertia REAL (0.0) [kg·m^2].
Rotational moment of inertia of the hub and the parts of the drive train that are on the rotor-side of the modelled flexibility in the rotor shaft, see Figure 5.

nacelle_mass REAL (1000.) [kg].
Mass of the nacelle including drive train components, excluding the mass of the hub and rotor blades, see Figure 4.

static_moment_x REAL (0.0) [kg·m]. *In release "NOV-2003" this is '***nacelle_unbalance***'.*
Static moment of inertia of the nacelle mass (excl. hub) w.r.t. the tower axis, positive downwind see Figure 4.

static_moment_z REAL (0.0) [kg·m]. *Not in release "NOV-2003".*
Static moment of inertia of the nacelle mass (excl. hub) w.r.t. the tower top, positive upward.

rolling_inertia REAL (0.0) [kg·m^2]. *Not in release "NOV-2003".*
Inertia of the nacelle (excluding hub) about the tower-top X-axis.

tilting_inertia REAL (0.0) [kg·m^2]. *Not in release "NOV-2003".*
Inertia of the nacelle (excluding hub) for tilting about the tower top.

cross_inertia REAL (0.0) [kg·m^2]. *Not in release "NOV-2003".*
Cross inertia for roll- and yaw motion of the nacelle excluding hub. Positive if the nacelle mass centre is located clearly above the tower top and downwind of the tower axis.

yawing_inertia REAL (0.0) [kg·m^2].
Inertia about the tower axis of the nacelle excluding hub and rotor blades, see Figure 4.

nacelle_side_drag REAL (0.0) [m^2] <0.0, ... >
Aerodynamic drag area (= $C_{\text{drag}} \cdot area$) of the nacelle for the transverse wind component.

nacelle_front_drag `REAL` (0.0) [m^2] <0.0, ... >
Aerodynamic drag area of the nacelle for wind on the front or back, similar as for '**nacelle_side_drag**'. In fact 'drag area' is the product of drag coefficient and area.
For a flat plate (rectangular or circular) the drag coefficient is 1.2. With the rounded edges as for the nacelle of some turbines, the drag coefficient may reduce to 1.0 or 0.9.

3.4 MENU configuration

MENU **configuration** contains flags for the degrees of freedom of the model. The input items for each d.o.f. are listed following the order of the corresponding SUBMENUs.
The variables of this MENU are stored in include file 'PNCLM4'.

tower_input `CHARACTER*20` ('towdef.inp').
Name of the file with input properties for the tower model, see chapter 5.

free_yawing `LOGICAL` (OFF)
If 'ON' the free yawing motion is solved, see section 3.4.1.
For the calculation of aerodynamic rotor characteristics, section 2.3.1, this is set OFF.

gearbox_support `LOGICAL` (OFF)
If 'ON' the torsional deformation of the gearbox support is modelled, see section 3.4.2.

shaft_torsion `LOGICAL` (OFF)
If 'ON' the elastic torsional deformation of the rotor shaft is modelled, see section 3.4.2.

teeter_flag `LOGICAL` (OFF)
If 'ON' the teeter motion is solved as degree of freedom, see section 3.4.3.

hinge_flag `LOGICAL` (OFF)
If 'ON' the blades have flapping hinges as a degree of freedom with stiffness, damping and bumpers, see section 3.4.4.
These hinges are located at a distance '**blade_root_radius**' from the hub, see Figure 3.

pitch_flag `LOGICAL` (OFF)
If 'ON' passive pitch motion is solved in routine '*tipsol*', see section 3.4.5.
If 'OFF' the pitch algorithm in routine '*conrol*' is active, see chapter 7.
For the calculation of aerodynamic rotor characteristics, section 2.3.1, this is set OFF.

flapping_flag `LOGICAL` (OFF)
If 'ON' continuous flapping deformation of the blades is modelled.

lagging_flag `LOGICAL` (OFF)
If 'ON' continuous lead-lag deformation of the blades is modelled.
Because of the interaction between flapwise and lead-lag bending for non-zero blade pitch angles, it is strongly recommended to set always **flapping_flag** and **lagging_flag** both 'ON' or both 'OFF.

blade_torsion `LOGICAL` (OFF)
If 'ON' continuous blade torsional deformation is modelled, for which the table '**torsion_data**' (MENU *blade_data*) needs to be specified, see section 3.6.
If specified in file '*karaolis.dat*', the coupling from bending or tension is included'.

The properties for each of the degrees of freedom are described in the following subsections.

3.4.1 SUBMENU yaw_data

SUBMENU **yaw_data** contains items to specify the prescribed periodic yawing or to solve the free yawing response of the nacelle. These items are stored in include file 'PNCLS4'.

The options to simulate the yaw behaviour are described in section 7.1.

For '**free_yawing OFF**' the following items specify a prescribed periodic yaw motion, which starts at **yaw_start_time** after which it follows a ramp-shaped motion between '**yaw_angle**' and '**yaw_angle** + **yaw_ramp_time** * **yaw_rate**'. This motion is plotted in Figure 9.

yaw_start_time REAL (99000.0) [s].
Time at which the periodic alternating yaw motion starts.

yaw_rate REAL (0.0) [deg/s].
Rate of periodic alternating yaw motion, starting with the sign of '**yaw_rate**' as given.

yaw_ramp_time REAL (0.0) [s].
Duration of each periodic yaw action, which is both for a 'positive' and for a 'negative' action with rate **yaw_rate**.

yaw_repeat_time REAL (99000.0) [s].
Full cycle time of each periodic yaw action, which includes a 'positive' and 'negative' action with rate **yaw_rate**.
The time between each action is: **yaw_repeat_time**/2 - **yaw_ramp_time**.

yaw_time_lag REAL (1.0) [s].
Time lag between yaw control and effectuation of yaw motion. Using a non-zero value for **yaw_time_lag** avoids exessive inertia loads from large yaw accelerations.

For '**free_yawing ON**' the following items are used to solve passive yaw motion.

yaw_damping REAL (0.0) [N·m/(rad/s)]. Viscous damping for a free yawing nacelle.

yaw_friction REAL (0.0) [N·m]. Coulomb friction for free yawing.

yawdrive_inertia REAL (0.0) [kg·m^2]. Inertia of yaw motor, expressed w.r.t. yaw motion.

3.4.2 SUBMENU generator_data

SUBMENU **generator_data** contains items for the drive train model.
The variables in this SUBMENU are stored in include file 'PNCLS5'.

Because the drive train describes the interaction between the dynamics of the rotor and the generator some emphasis is laid on its modelling. The PHATAS configuration of the drive train is shown in Figure 5. The set of non-linear differential equations is solved in terms of:

- Rotor speed;
- Shaft torsional deformation rate.
  The elastic deformation of the rotor shaft is optional ('**shaft_torsion ON**');
- Torsional deformation rate of the flexible gearbox support.
  A flexible gearbox support is optional ('**gearbox_support ON**').

The generator characteristics are described with an expression for the torque as function of the generator rotational speed, see **generator_model**. The generator speed is the result of the rotor speed minus the torsional deformation rate of the rotor shaft and the gearbox support.

Items for the generator

The input item used to select the generator torque model is:

generator_model `INTEGER` (0) <0, ..., 5>
Indicates the type of model used for the torque characteristics of the generator.
For the calculation of aerodynamic rotor characteristics (see section 2.3.1)
generator_model is set to **0** in PHATAS. For other applications its value specifies:

= 0 The generator speed is kept constant at the value; '**rated_rotorspeed**'.
The flexibility of the rotor shaft and gearbox support can still be modelled.

= 1 Synchronous generator characteristics of which the synchronous rotor speed is '**rated_rotorspeed**'. The load angle for nominal power '**rated_power**' measured at the generator shaft is '**generator_slope**'. The generator torque is evaluated with
$Q_{\rm gen} = Q_{\rm nom} \cdot ((\phi_{\rm gen} - \phi_{\rm synch}) + \mathbf{tau\_gen} \cdot (\Omega_{\rm gen} - \Omega_{\rm synch})) / \mathbf{generator\_slope}$.

= 2 Asynchronous generator characteristics with a nominal power '**rated_power**' at '**rated_rotorspeed**'. The slip at '**rated_rotorspeed**' is '**generator_slope**' which implies that the synchronous rotorspeed is
'**rated_rotorspeed**'/(1 + '**generator_slope**') .
The generator torque follows the relation of Kloss:
$Q_{\rm gen} + \mathbf{tau\_gen} \cdot \partial Q_{\rm gen}/\partial t = 2 \cdot Q_{\rm k}/(S/S_{\rm k} + S_{\rm k}/S)$.
With $Q_{\rm k}$ the maximum torque '**maximum_torque**' at a slip of $S_{\rm k}$ and '**tau_gen**' the time constant for dynamic behaviour of the generator.

= 3 The generator is described with a constant λ control below- and a constant power control above '**rated_rotorspeed**', see Figure 6.

= 4 The generator torque and transmission loss are read from table **generator_curve** as piece-wise linearised function.

= 5 The routine '*torgen*' for generator characteristics, see section 7.2 and Figure 24, is not called. This means that (if present) the statements for the generator torque in routine '*conrol*' are valid. In '*conrol*' the generator torque and its derivative with respect to the slow shaft speed must be assigned to the source code variables `qgen` and `dqdomg` see section 7.6.

For all values of **generator_model** larger than 1, the generator torque is expressed with a time-lag function of the rotational speed, see input item **tau_gen**. The properties used for the description of the generator model(s) are:

rated_rotorspeed `REAL` (60.0) [rpm] $< \ldots -0.001 >$ or $< 0.001 \ldots >$
The interpretation of **rated_rotorspeed** depends on the type of calculation:

For '**time_history_flag OFF**' and '**delta_lambda** $>$ **0**':
the aerodynamic rotor characteristics are calculated with '**rated_rotorspeed**';

For '**time_history_flag ON**' or '**delta_lambda** $<$ **0**':
'**rated_rotorspeed**' is used for the description of the generator torque.

The value of **rated_rotorspeed** is also used to simulate short-circuit (see **failure_type 5** in MENU **control**), based on an expression for the 'kip' torque of an asynchronous generator.

rated_power `REAL` (300.E+3) [W]
Rated (or "nominal") generator power, used for '**generator_model 0, 1, 2**', and '**3**', and used for the simulation of a short-circuit.

generator_slope `REAL` (0.01)
For ’**generator_model 1**’ the load angle [deg] at the generator shaft for ‘**rated_power**‘;
For ’**generator_model 2**’ the slip [.] of the generator at ‘**rated_rotorspeed**‘;
The value of **generator_slope** is also used to simulate a generator short-circuit.

maximum_torque `REAL` (9.E+9) [Nm]
The maximum torque (at the generator shaft) of an asynchronous generator used in the relation of Kloss, see ’**generator_model 2**’, and for the simulation of generator short-circuit, see ’**failure_type 5**’ of SUBMENU **control**.

tau_gen `REAL` (0.0) [s]
Time constant in the expressions for the generator torque. For a speed-dependent torque (all except ’**generator_model 1**’) the expression for the unsteady generator torque is:
$Q_{\mathrm{gen}} + \mathbf{tau\_gen} \cdot \partial Q_{\mathrm{gen}}/\partial t = Q_{\mathrm{gen,\,steady}}$.

generator_curve `table(`**ordinate , generator_torque, loss_torque**`)` $\begin{pmatrix} 0.9901 & 0.0 & 0.0 \\ 1.0 & 2387.3 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ .... & .... & .... \end{pmatrix}$

Describes the piece-wise linearised relationship for the generator torque and transmission loss. This table is only used for ’**generator_model 4**’. The columns are:

ordinate `REAL(1:24)` [.] Slow shaft speed divided by ‘**rated_rotorspeed**‘.
A decrease of two subsequent values indicates the end of the specifications;

generator_torque `REAL(1:24)` [N·m]
Torque values measured at the generator shaft.

loss_torque `REAL(1:24)` [N·m]
Transmission loss values measured at the generator shaft.

The default input describes 300kW generator power for 1% slip at 60rpm without loss.

Items for drive train components

The configuration of the drive train modelled in PHATAS includes flexibility of the gearbox support while the generator and brake are each mounted on the nacelle frame, see Figure 5.

gear_ratio `REAL` (1.0) [.]
Transmission ratio of the gearbox. The value of ‘**gear_ratio**‘ can be either positive or negative depending on the rotational direction of the generator shaft. ‘**gear_ratio**‘ is negative if the generator shaft rotates opposite to the rotor shaft. The rotational direction is of importance to solve the correct nacelle and tower moments when speeding-up or -down and when yawing.

constant_loss `REAL` (0.0) [N·m]
The losses in the drive train are described with a linear function of the generator shaft torque. ‘**constant_loss**‘ is the loss of torque for an unloaded generator expressed at the slow shaft, Figure 5. *Not used for* ’**generator_model 4**’.

proportional_loss `REAL` (0.0) [.] $< 0.0 \ldots 1.0 >$
Fraction of the drive train loss proportional to the generator shaft torque of the gearbox. See also ‘**constant_loss**’. *Not used for* ’**generator_model 4**’.

slow_shaft_inertia `REAL` (0.0) [kg·m^2]
Rotational moment of inertia of the slow rotating parts of the drive train, Figure 5.
See also **fast_shaft_inertia**.

fast_shaft_inertia `REAL` (0.0) [kg·m^2]
Rotational moment of inertia of the fast rotating parts of the drive train (generator, brake, generator shaft, see Figure 5) expressed at the generator shaft. See also **slow_shaft_inertia**.

shaft_stiffness `REAL` (9.E+9) [N·m/rad]
Torsional stiffness of (flexible-) rotor shaft. For '**shaft_torsion ON**' this flexibility is modelled between the inertia of the hub and the inertia of the slow shaft, see Figure 5.

gearbox_inertia `REAL` (0.0) [kg·m^2]
Rotational moment of inertia of the gearbox and all connected parts, see Figure 5.

support_stiffness `REAL` (9.E+9) [N·m/rad]
Torsional stiffness of the flexible gearbox support, see Figure 5.

support_damping `REAL` (0.0) [N·m/(rad/s)]
Viscous damping of the flexible gearbox support, see Figure 5.

3.4.3 SUBMENU teeter_data

SUBMENU **teeter_data** contains items of the model for the dynamics of the teeter mechanism. (for '**teeter_flag ON**'). The variables in this SUBMENU are stored in include file 'PNCLS6'. For 2-bladed rotors the teeter hinge model has been used to analyse the shaft bending dynamics, using the **teeter_spring** for the shaft bending stiffness.

delta_3_angle `REAL` (0.0) [deg]
Orientation of the teeter hinge in the rotor axis system.
'**delta_3_angle**' is positive if the motion of blade 1 for an increasing teeter angle results in a reduction of the aerodynamic inflow angle, see Figure 3.

teeter_offset `REAL` (0.0) [m]
Location of the teeter hinge on the rotor shaft measured downwind from the hub, Figure 3.

teeter_spring `table(`**teeter_spring_angle , teeter_moment**`)` $\begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 0.0 \\ .... & .... \end{pmatrix}$
Describes the piece-wise linearised stiffness of the teeter spring:

teeter_spring_angle `REAL(1:10)` [deg]
Increasing series of teeter angles, which must be positive.
A decrease of two subsequent teeter angles indicates the end of the specifications.

teeter_moment `REAL(1:10)` [N·m]
Restoring moments about the teeter hinge.

teeter_friction `REAL` (0.0) [N·m]
Hysteresis or 'Coulomb' friction in the teeter hinge.

teeter_viscosity `REAL` (0.0) [N·m/(rad/s)]
Viscous damping of the teeter motion.

3.4.4 SUBMENU hinge_data

SUBMENU **hinge_data** contains items for flapping hinges ’**hinge_flag ON**’.
For ’**hinge_flag OFF**’ the items in this SUBMENU are not significant.
The flapping hinges are located at ‘**blade_root_radius**‘ and oriented in the rotor plane perpendicular to the blade-axes, see Figure 3.
The variables in this SUBMENU are stored in include file ’PNCLS7’.

hinge_stiffness `REAL` (0.0) $<$ 0.0 .... $>$ [N·m/rad]
Stiffness of the flapping hinge spring.

hinge_damping `REAL` (0.0) [N·m/(rad/s)]
Viscous damping of the hinge flapping motion.

min_flap_angle `REAL` (-90.0) [deg]
Minimum hinge flapping angle when the blade just hits the upwind bumper.

max_flap_angle `REAL` (90.0) [deg]
Maximum hinge flapping angle when the blade just hits the downwind bumper.

bumper_stiffness `REAL` (0.0) $<$ ‘**hinge_stiffness**‘ .... $>$ [N·m/rad]
Stiffness of the bumper for angles beyond **min_flap_angle** or **max_flap_angle**.

3.4.5 SUBMENU pitch_data

SUBMENU **pitch_data** contains properties of the model for the pitch mechanism.
For ’**pitch_flag OFF**’ the input items in this SUBMENU are not significant, except for **min_pitch_angle** and **max_pitch_angle**.
The variables in this SUBMENU are stored in include file ’PNCLS8’.

coupled_pitch `LOGICAL` (OFF)
If ’ON’ the passive pitch motion of all blades is coupled.

min_pitch_angle `REAL` (-90.0) [deg]
Stop at minimum pitch angle.
This is not used for the calculation of aerodynamic rotor characteristics, see 2.3.1.

max_pitch_angle `REAL` (90.0) [deg]
Stop at maximum pitch angle.
This is not used for the calculation of aerodynamic rotor characteristics, see 2.3.1.

pitch_stiffness `REAL` (9.E+9) [N·m/rad]
Stiffness of a torsional spring in the pitch bearing of one blade.

pre_stress_moment `REAL` (0.0) [N·m]
Pre-stress moment of the torsional spring for one blade at zero pitch angle.

pitch_damping `REAL` (0.0) [N·m/(rad/s)]
Viscous damping of the pitch motion for one blade.

pitch_friction `REAL` (0.0) [N·m]
Coulomb friction in the pitch motion of one blade.

The stops for the pitch angles are active for solving passive pitch motion in routine ‘*tipsol*’ (’**pitch_flag ON**’) and if pitch control is modelled in routine ‘*conrol*’ (’**pitch_flag OFF**’).

3.5 MENU profiles

MENU **profiles** contains items for the aerodynamic properties of the blades and for the description of the rotor aerodynamics.
The variables of this MENU are stored in include file 'PNCLM5'.

airfoil_table `table`(**airfoil_span, airfoil_file**) $\begin{pmatrix} 0.0 & '\mathrm{NAC23018}' \\ 0.0 & '\mathrm{none}' \\ .... & .... \end{pmatrix}$

Contains the spanwise distribution of files with aerodynamic coefficient-tables.

airfoil_span `REAL(1:12)` Spanwise coordinate [m] measured from '**blade_root_radius**'.

airfoil_file `CHARACTER*20(1:12)` Name of the file with the airfoil coefficients table that has to be used outboard of the span '**airfoil_span**'.

Based on this table the aerodynamic properties of the blade elements are assigned using the spanwise location at the middle of each of the elements.
If the first value of '**airfoil_span**' is larger than 0.0 the aerodynamic properties of the elements of which the middle is within '**airfoil_span**' are set to zero.

3D_correction `LOGICAL` (OFF)
If 'ON' a correction for the effects of rotation is applied to the lift coefficients of blade sections up to 80% radius. This correction is based on the method of Snel, Houwink, and Bosschers [21]. The dependency of rotational speed (speed ratio) is described with factor $(\Omega\, r)^{\,2}/V_{\mathrm{eff}}^2$, see also section 4.4 of [15].

dynamic_stall `INTEGER` (0) $< 0, 1, 2 >$
= **0**: the quasi-steady airfoil properties are used (if specified: with **3D_correction**).

= **1**: the dynamic stall behaviour is described with the linear part (with $\Delta\, c_{l\,1}$) of the dynamic stall model of Snel [22], which describes the dynamic lift coefficient based on the description by Truong. The first-order part of this model is described in section 4.4 of [16].

= **2**: *not fully tested!* the unsteady stall behaviour is described with the second order (non-linear) dynamic stall model of Snel, in terms of $\Delta\, c_{l\,1} \mathrm{and} \Delta\, c_{l\,2}$. Due to the non-linear terms in this model, the solution requires an iteration process which uses more CPU time. Because this dynamic stall model is developed using wind tunnel measurements on a single blade for pitch variations with only one frequency and amplitude, this model still needs validation. Moreover the wind turbulence and rotor blade dynamics (flap bending) of a real wind turbine have more the character of a 'heave' motion instead of pitch.

hup1 `REAL` (0.0)
Variable that is reserved for additional or alternative algorithms, to be applied in research applications. In the governing routine the file 'PNCLM5' should be included.

hup2 `REAL` (0.0) (see description of **hup1**).

hup3 `REAL` (0.0) (see description of **hup1**).

The format of the contents of the airfoil datafiles is described in section 2.2.6. Note that the moment coefficient is positive in the direction of decreasing pitch angle, "nose-up", which is in the direction of larger angles of attack, see Figure 8.

The formulation of the '3D correction' for rotational effects is based on the angle of attack for zero lift $\alpha_{(Cl=0)}$ which means that the coefficients near zero lift must have reliable values.

3.6 MENU blade_data

MENU **blade_data** contains items for the geometric-, mass-, and stiffness properties of the blades. The geometric blade properties (chord, aerodynamic centre, and twist) are used for the calculation of the dynamic response as well as for the aerodynamic rotor characteristics. All tabulated blade properties that are a function of the spanwise location are read as piece-wise linearised distributions between the specified values. The spanwise coordinates of the tabulated blade properties need to form an increasing series. If the blade properties are not specified over the entire blade span the distribution is extrapolated linearly to the rotor centre and/or to the tip. For blades shorter than 11.5m (range of default blade) it is strongly recommended to specify the blade properties over the entire blade span to avoid that incorrect (default) values are used.
The variables of this MENU are stored in include file 'PNCLM6'.

Items for the blade shape

The blade model in PHATAS allows a blade axis with a so-called 'pre-bend' geometry of the unloaded state. This 'pre-bend' blade axis is modelled with a uniform curvature over a part of the span, between locations **pre_bend_begin** and **pre_bend_end**. The amount of 'pre-bend' curvature is specified as the unloaded! position of the blade tip and can have a component in flap-wise and in lagwise direction (defined for zero pitch).

flap_pre_bend `REAL` (0.0) [m], $< -0.25(R-(\mathbf{pre\_bend\_begin}+\mathbf{pre\_bend\_end})/2), ..., 0.25(R-(\mathbf{pre\_bend\_begin}+\mathbf{pre\_bend\_end})/2) >$

Geometric flapwise blade curvature of the tip, downwind positive.

lag_pre_bend `REAL` (0.0) [m], $< -0.25(R-(\mathbf{pre\_bend\_begin}+\mathbf{pre\_bend\_end})/2), ..., 0.25(R-(\mathbf{pre\_bend\_begin}+\mathbf{pre\_bend\_end})/2) >$

Similar as **flap_pre_bend**, but in lag-wise direction.
Both **flap_pre_bend** and **lag_pre_bend** are defined for a zero blade pitch angle.

pre_bend_begin `REAL` (0.0) [m], $< 0.0, ..., R >$

Location from the blade root at which the uniform 'pre-bend' blade curvature starts.

pre_bend_end `REAL` (0.0) [m], $< 0.0, ..., R >$

Location from the blade root at which the uniform 'pre-bend' blade curvature ends.

Items for the geometric blade properties

The geometrical blade properties are vital for the calculation of the aerodynamic loads for which reason they need to be specified for all applications of PHATAS.

blade_geometry `table(`**s_geometry , blade_chord , aero_centre**`)`

The default values are $\begin{pmatrix} 0.0 & 0.6 & 0.0 \\ 0.20 & 0.6 & 0.0 \\ 0.375 & 1.813 & 0.0 \\ 11.5 & 0.3 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ .... & .... & .... \end{pmatrix}$

s_geometry `REAL(1:25)` [m] Spanwise coordinates, see Figure 7.

blade_chord `REAL(1:25)` [m] Blade chord values, see Figure 7.

aero_centre `REAL(1:25)` [m]

Chordwise location of the aerodynamic centre, measured from the blade axis to the trailing edge of the cross section, see Figure 8. The aerodynamic centre is the point at which the specified aerodynamic moment coefficients are defined (usually 25% of the chord from the leading edge). The position of the aerodynamic centre can be used to calculate the aeroelastic behaviour of wind turbines with blades that have a curvature in lag-wise direction. In this case also the orientation of the aerodynamic centre line is included in the decomposition of the relative airflow over the blade chord.

blade_twist `table(`**satw1 , twist_angle**`)` $\begin{pmatrix} 0.0 & 0.0 \\ 11.5 & 0.0 \\ 0.0 & 0.0 \\ ... & ... \end{pmatrix}$

satw1 `REAL(1:25)` [m] Spanwise coordinates.

twist_angle `REAL(1:25)` [deg] Blade twist angle, see Figure 8.

Items for the blade mass properties

The blade mass properties are essential for the gravitational loads and the dynamic loading. These properties need to be specified at least for a power curve calculation and for dynamic design load calculations.

symass `REAL(0:24)` [m]

Spanwise coordinates for the blade mass distributions.

ymass `REAL(0:24,1:3)` [kg/m]

Distribution of the blade mass per unit length. The first index corresponds with the index of the spanwise coordinate. The second index is the blade number.
If the mass properties are specified from a span larger than '**blade_root_radius**' then the mass distribution is extrapolated to the blade root.
If the mass properties are specified from a negative span (inboard of '**blade_root_radius**') then this mass is added to the hub.

sxcg `REAL(0:24)` (0.0 11.5 0.0 ...) [m]

Spanwise coordinates of the mass centre line.

xcg `REAL(0:24,1:3)` (0.0 0.0 0.0 ...) [m]

Chordwise location of the mass centre line measured from the blade axis to the trailing edge. The indices are defined identical as for **ymass**.

radius_of_gyration `table(`**srdel , rdel**`)`

The default values are $\begin{pmatrix} 0.0 & 0.3118 \\ 1.5 & 0.3039 \\ 3.5 & 0.2416 \\ .... & .... \\ 10.5 & 0.0892 \\ 11.5 & 0.05678 \\ 0.0 & 0.0 \\ .... & .... \end{pmatrix}$

srdel `REAL(0:24)` [m] Spanwise coordinates.

rdel `REAL(0:24)` [m] Radius of gyration with respect to the mass centre line **xcg**.

Items for the blade stiffness properties

The blade stiffness properties are needed for modelling of the structural dynamic response of the rotor blades. These properties only need to be specified if blade deformation is modelled;
’**flapping_flag ON**’, ’**lagging_flag ON**’, or ’**blade_torsion ON**’.
The stiffnesses are expressed in the axis system of the local blade chord as shown in Figure 8.

s_stiffness `REAL(0:24)` [m]
Spanwise coordinates of the specified blade bending stiffnesses.

flat_stiffness `REAL(0:24)` [N·m^2]
Values of the flatwise bending stiffnesses of the blades.

edge_stiffness `REAL(0:24)` [N·m^2]
Values of the edgewise bending stiffnesses of the blades.

cross_stiffness `REAL(0:24)` (0.0 0.0 ...) [N·m^2]
Values of the coupling stiffnesses between flatwise and edgewise bending, positive if a downwind flap bending moment results in a deformation with a leadwise component.

torsion_data `table(`**sgjt , gjt , shear_centre**`)`

The default values are

$$\begin{pmatrix} 0.0 & 16.9E+6 & 0.0 \\ 1.5 & 4.71E+6 & 0.0 \\ .... & .... & .... \\ 9.5 & 0.19E+6 & 0.0 \\ 11.5 & 0.044E+6 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ .... & .... & .... \end{pmatrix}$$

sgjt `REAL(0:24)` [m] Spanwise coordinates;

gjt `REAL(0:24)` [N·m/(rad/m)] Values of the torsional stiffnesses;

shear_centre `REAL(0:24)` [m] Chordwise locations of the shear centre measured from the blade axis to the trailing edge.

flat_damping `REAL` (0.0) [.]
Structural damping for flatwise blade bending deformation. This is expressed as fraction of the critical damping for the first symmetric flatwise bending mode of blade 1 when rotating with ‘**rated_rotorspeed**‘. For other flatwise bending modes the structural damping (in dimensionless terms) is proportional to the natural frequency.

edge_damping `REAL` (0.0) [.]
Structural damping for edgewise blade bending deformation.
The amount of edgewise damping is defined similar as for flatwise bending.

torsion_damping `REAL` (0.0) [.]
Structural damping for blade torsional deformation.
The amount of torsional damping is defined similar as for bending deformation.

For rotor blades with elastic tailoring, the torsional deformation is not only caused by a torsional moment but also by the bending moments and/or the spanwise force in the cross section. The dynamic response of these rotor blades can be solved with PHATAS if the coupling terms following the notation of Karaolis are present in a file named ‘*karaolis.dat*’, see section 2.2.4. If this file is not present these coupling effects are set to zero.

The default distributions of the chord, the sectional mass, the radius of gyration, and the stiffnesses of the blades are those of the former 25m HAT rotor test facility of ECN, Petten.

3.7 MENU wind

MENU **wind** contains input items for the wind loading on the rotor.
The variables of this MENU are stored in include file 'PNCLM7'.

In PHATAS the aerodynamic loads on the blade elements are calculated from the difference between the undisturbed wind velocity vector and the motion of the rotor blade due to all degrees of freedom and structural deformations. For each blade element the wind loading is evaluated at the deformed position.
For all descriptions of the wind loading the aerodynamic stagnation of the (lattice- or tubular-) tower is described in one of the routines of the tower module, see chapter 5.

load_case `CHARACTER*20` ('none')
String that is written in the output files and can be used to identify the wind conditions.

Several options for wind loading are available in PHATAS that can be selected using the input item **wind_option**. For all options of the wind loading, the wind direction and wind elevation are applied in the following sequence:

$$\vec{V}_{(\mathrm{X,Y,Z,t})} = \bar{S}_{(\mathbf{wind\_direction})} \cdot \bar{S}_{(\mathbf{wind\_elevation})} \cdot \vec{V}_{(\mathrm{wind,t})} \; .$$

Here $\bar{S}$ denotes a transformation matrix.
The wind velocity on the rotor plane may be the wind as read from a SWIFT wind datafile.
It may also be a wind loading as modelled with the other 'options':
$\vec{V}_{(\mathrm{wind,t})} = \bar{S}_{\mathbf{wind\_twist}} \cdot [\,(V_{(\mathrm{ambient})} + vertical\ shear) \cdot (1 + horiz.\,shear) + gust(t)\,].$
Here $S_{\mathbf{wind\_twist}}$ is the transformation matrix for the vertical variation in wind direction.

The item **wind_option** is described here first.

wind_option `INTEGER` (0) <0, 1, 2, 3, 4, 5, 6, 10, 11, 12, 13, 20>
Indicates the type of wind loading on the rotor:

= **0** A constant wind speed with time.
For rotor characteristics or power curve calculation **wind_option** is set to '0'.

= **1** A stepwise increasing wind speed and wind direction.
This wind option can be used to evaluate the response of the control algorithm. The discontinuous! steps all have a subsequent additional wind speed amplitude **amplitude** and wind direction increments of **direction_change**. These steps start at **start_time**, last for **time_period** seconds each, while the total sequence continues for **duration** seconds.

= **2** A 'ramp shaped' wind gust and direction change.
The length of the increasing and the decreasing part of the ramp is '**time_period**' while the maximum wind speed lasts for '**duration**' seconds, see Figure 11.a.

= **3** A gust and direction change following the description of Frost [4]:
$V_{\mathrm{wind}}(t) = V_{\mathrm{wind}}(t=0) \; + \mathbf{amplitude} \cdot 1.581977 \cdot [1 - \exp\left(-(\sin arg)^{1/3}\right)] \; .$
For the increasing part of the gust $arg = (\pi/2) \cdot (\mathrm{t} - \mathbf{start\_time})/(\mathrm{aa} \cdot \mathbf{time\_period})$
and for the decreasing part of the gust

$$arg = \frac{\pi}{2} \cdot \left(\frac{\mathbf{start\_time} + \mathbf{time\_period} + \mathbf{duration} - t}{(1-aa) \cdot \mathbf{time\_period}}\right) \; .$$

Here the factor aa is given by: $aa = 0.117 + 0.048 \cdot \log\left(Z_{\mathrm{hub}}\right)$.
The maximum wind speed holds for '**duration**' seconds, see Figure 11.b.

= **4** A sinusoidal wind speed variation and direction change with the specified time '**time_period**', which acts continuously for '**duration**' seconds, see Figure 11.c.

$= \mathbf{5}$ Options for a (1 - cos)/2 shaped gust, direction change, and shear variation following the IEC 61400-1 Safety requirements, [8] and the G.L. recommendations [5]. Using this option, (1 - cos)/2 variations are available for:

- Wind speed with an amplitude '**amplitude**';
- Wind direction with an amplitude '**direction_change**' [°];
- Linear vertical shear with an amplitude '**vert_shear_change**' [(m/s)/Radius];
- Linear horizontal shear with an amplitude '**hor_shear_change**' [(m/s)/Radius].

These definitions of **time_period** *and* **amplitude** *are not consistent with those in [8] and [5]!*

The (1 - cos)/2 variations are of the form of Figure 11.d. For this option vertical twist in wind direction can not be applied while vertical shear applies to the specifications for **shear_type**.

$= \mathbf{6}$ An 'Extreme Operating Gust' as in section 6.3.2 of IEC 61400-1 [8] and G.L. [5]:
"$-\,\mathbf{amplitude} \cdot 0.5 \cdot \sin(3\,\pi\,t/\mathbf{time\_period}) \cdot (1 - \cos(2\,\pi\,t/\mathbf{time\_period}))$"
One should be aware of the factor 0.37 *in [8] and [5].*
This function can also be used for a '**direction_change**', although this is not required for wind turbine design.

$= \mathbf{10}$ (Measured) wind data are read from the ASCII file '**wind_datafile**'.
The variables of which the time series are read (in routine '*windt*') depend on the value of **shear_type**:

For '**shear_type 0**' every record must contain:

- time [s];
- wind speed [m/s] at hub height $-$ '**lower_shear**' [m];
- wind speed [m/s] at hub height;
- wind speed [m/s] at hub height + '**upper_shear**' [m];
- wind direction [°], defined similar as **wind_direction**.

The vertical shear has a bilinear distribution through the wind speed values read from the file.

For '**shear_type** $>$ **0**' every record must contain:

- time [s];
- wind speed [m/s] at hub height;
- wind direction [°], defined similar as **wind_direction**.

The vertical shear is in accordance with '**shear_type**', where the shear is related to the actual wind speed and *not* to the ambient value: '**wind_velocity**'

The measured wind time series are applied at '**start_time**' seconds later than the time read from the wind file. If the time series in the ASCII wind file starts later or ends earlier than the time series of the calculation (after adding '**start_time**'), the wind is kept constant at the value at the beginning c.q. end of the file. The actual wind data used in PHATAS are obtained by linear interpolation in time between the data read from file. The horizontal shear and the wind twist are added as function of the position in the rotor plane.

= **11** The longitudinal components of the wind distribution on the rotor plane are read from a file as generated with SWIFT [25]. The wind generated with SWIFT is calculated for a given rotor geometry, spectral density, and coherence. In PHATAS checks are performed on the hub height, rotor diameter, wind speed, vertical shear type, and on its shear parameter to ensure that the correct wind data file is used. If the time interval in the wind data file does not match with '**time_increment**' of the PHATAS calculation the wind data are interpolated to the time steps needed. If the time in the calculation exceeds the time span of the SWIFT wind datafile, reading of the wind data is continued from the beginning of the file. Because the SWIFT wind datafiles are periodic this gives no discontinuities.

= **12** As for '**wind_option 11**' including lateral turbulence.
If the SWIFT wind datafile does not contain lateral turbulence a question is written to the screen whether to perform the calculation for '**wind_option 11**'.

= **13** As for '**wind_option 11**' including lateral and vertical turbulence.
If the SWIFT wind datafile does not contain the full 3D turbulence components a question is written to the screen whether to perform the calculation for '**wind_option 12**' or '**11**'.

= **20** The steady wind distribution is read as acting in a plane perpendicular to the ambient wind direction using the WAKEFARM output, [24]. The WAKEFARM output-tables with u, v, and w components can be copied below each other where the table with u components must come first. After reading the u components the program PHATAS looks for a table with the v and/or w components, which are zero if these components are not present.
If during the calculation with PHATAS the blade tips move out of the "WAKEFARM" grid, the applied wind on the tips is that at the edges of the grid.
Knowing that the WAKEFARM output usually describes a uniform wind out of the wake, it is recommended to generate a WAKEFARM output that encloses the complete wake of the up-wind rotor. Then the analysed turbine is always exposed to a realistic wind distribution, whether or not in partial wake operation.

For '**output_level 3**' or higher the global properties of the grid of the SWIFT or WAKEFARM file are written to file '*STATUS*'.

For '**wind_option 1, 2, 3, 4, 5,** or **6**' the gusts starts at '**start_time**' .

For all options for the wind loading a time series of the wind speed and the wind direction at the hub can be obtained as output property **2** and **6** respectively, see chapter 4 and Appendix A.

The remaining input items for wind loading are described in alphabetic order.

air_density `REAL` (1.225) [kg/m^3] $< 0.0 \ldots.\ >$ Density of the air.

amplitude `REAL` (0.0) [m/s]
Amplitude of the gust (for ’**wind_option 1 - 6**), see Figure 11

direction_change `REAL` (0.0) [^{o}]
Amplitude of the wind direction change. The direction change is modelled for **wind_option 1** through **6** with a similar function as the wind gust (with the input item **amplitude**).

duration `REAL` (0.0) [s]
The definition depends on the value of **wind_option**, see Figure 11.

$=$ **1** : Overall time during which the wind velocity increases stepwise.

$=$ **2, 3, 5, 6** : Maximum wind speed of the modelled gust is kept constant for ‘**duration**‘ seconds.

$=$ **4** : The sinusoidal wind acts for ‘**duration**‘ seconds.

hor_shear `REAL` (0.0) [Radius^{-1}]
Linear horizontal shear over the rotor plane, positive if the wind speed on the left side of the rotor is higher than on the right side, see Figure 10. The specification ’**hor_shear 0.01**’ desribes a wind speed on the left edge of the rotor plane is 1% higher than the speed at hub. The horizontal shear is proportional with the wind speed at the given height.

hor_shear_change `REAL` (0.0) [(m/s)/Radius]
For ’**wind_option 5**’: Amplitude of the (1 - cos)/2 variation in horizontal shear, as described in the IEC Safety requirements [8].

lower_shear `REAL` (0.0)

For ’**wind_option 10**’ and ’**shear_type 0**’: Distance [m] below the hub of the first wind speed (2-nd column) on the records from the file.

For ’**shear_type 1**’: Relative wind shear [(m/s)/Radius] below hub height, positive for a wind speed that increases with height.

shear_parameter `REAL` (0.0) [m] $<$ 1.E-8, ... $>$
Parameter in the description of the vertical shear profile, which depends on **shear_type**.

$=$ **2:** Exponent [-] used in the power law;

$=$ **3** and **4:** Terrain roughness length z_0 [m] in the expression for a logarithmic shear.

shear_type `INTEGER` (0) $<$0, 1, 2, 3, 4$>$
Indicates which vertical wind shear profile is used:

$=$ **0:** No vertical shear. For ’**wind_option 10**’ (wind from a file) this indicates that a bilinear shear is calculated through the values of the wind speed from file.

$=$ **1:** A bilinear shear. A linear shear above hub height:
$V_{\mathrm{steady}}(Y_{\mathrm{hub}}, Z) = [\,\mathbf{wind\_velocity} + \mathbf{upper\_shear} \cdot (\frac{Z - Z_{\mathrm{hub}}}{R})\,]$
and a linear shear below hub height:
$V_{\mathrm{steady}}(Y_{\mathrm{hub}}, Z) = [\,\mathbf{wind\_velocity} + \mathbf{lower\_shear} \cdot (\frac{Z - Z_{\mathrm{hub}}}{R})\,]$.

$=$ **2:** A power function for the vertical shear:
$V_{\mathrm{steady}}(Y_{\mathrm{hub}}, Z) = \mathbf{wind\_velocity} \cdot (\frac{Z}{Z_{\mathrm{hub}}})^{\mathbf{shear\_parameter}}$.

= **3:** A logarithmic vertical wind shear:
$V_{\rm steady}(Y_{\rm hub}, Z) = \textbf{wind\_velocity} \cdot \frac{\ln(Z/\textbf{shear\_parameter})}{\ln(Z_{\rm hub}/\textbf{shear\_parameter})}$.
A logarithmic wind shear may be used for the wind profile at low altitudes.

= **4:** A logarithmic vertical wind shear as for **3** with an additional term '0.6 m/s' for heights above 25m (see Handbook version 3 [23]):
$V_{\rm steady}(Y_{\rm hub}, Z) = [\,\textbf{wind\_velocity} \cdot \frac{\ln(Z/\textbf{shear\_parameter})}{\ln(Z_{\rm hub}/\textbf{shear\_parameter})} + 0.6{\rm m/s} \cdot \ln(Z/Z_{\rm hub})\,]$.

Here $Y_{\rm hub}$ and $Z_{\rm hub}$ describe the location of the hub.
In the logarithmic shear profiles '**shear_parameter**' is the so-called 'roughness length'.
The stationary wind velocity in any point of the rotor plane is then
$V_{\rm steady}(Y, Z) = V_{\rm steady}(Y_{\rm hub}, Z) \cdot [\,1 + \textbf{hor\_shear} \cdot (\frac{Y - Y_{\rm hub}}{R})\,]$.
In PHATAS the gust is a uniform wind variation over the rotor disk that is added to the stationary wind distribution, independent from the vertical and horizontal wind shear.

start_time `REAL` (0.0) [s]
Time at which a modelled gust starts or the data read from file are applied.
For '**wind_option 10**' the wind time series (from file) are applied to the rotor at '**start_time**' seconds later than the time read from the file.

time_period `REAL` (10.0) [s]
For '**wind_option 1**' the time interval between the stepwise wind increments.
For '**wind_option 2, 3, 4, 5**, and **6**' the time period of the modelled gust, see Figure 11.

upper_shear `REAL` (0.0)

For '**wind_option 10**' and '**shear_type 0**': Distance [m] above the hub of the third wind speed (4-th column) read from the records of the file.

For '**shear_type 1**': Relative wind shear [(m/s)/Radius] above hub height related to the undisturbed wind speed at hub height.

vert_shear_change `REAL` (0.0) [(m/s)/Radius]
For '**wind_option 5**': Amplitude of the (1 - cos)/2 variation in vertical shear, as described in the IEC Safety requirements.

wind_datafile `CHARACTER*20` ('none')
For '**wind_option 10, 11, 12, 13** or **20**': Name of the file with wind data. If PHATAS is invoked with at least two command line arguments, the second is read as **wind_datafile**.

wind_direction `REAL` (0.0) [°]
Direction of the wind with respect to the global X-axis, positive if the wind comes from the east and the turbine is oriented to the north, see Figure 10. This is not valid for wind read from a file.

wind_elevation `REAL` (0.0) [°]
The angle of the undisturbed wind direction with respect to the horizontal plane.
Positive if the wind is upward. This is valid for all wind options.

wind_twist `REAL` (0.0) [°/Radius]
Vertical gradient of the wind direction, positive if the wind direction, see Figure 10, increases with height. This is not applied in combination with wind from a file.

wind_velocity `REAL` (0.0) [m/s]
Undisturbed (without gust) wind speed at hub height.

3.8 MENU waves

MENU **waves** contains input items for wave loading on the tower.
The variables of this MENU are stored in include file 'PNCLM8'.

The wave loads on the tower in the PHATAS model are calculated from the difference between the (external) wave velocities and the motion of the tower due to bending. A description of the modules for tower dynamics is given in chapter 5.

With the item **wave_option** the user can specify which type of description is used for the wave loading. This item is described here first.

wave_option `INTEGER` (0) <0, 1, 2>
Indicates the type of wave loading

= **0** : No wave loading;

= **1** : Wave loading read from a '*ROWS*' file, in binary format;

= **2** : Wave loading read from an ASCII file, as generated with *Streamfunction* [18].

water_density `REAL` (1025.0) [kg/m^3]
Density of the water, used for the wave loading on an offshore tower.

water_depth `REAL` (0.0) [m]
Average depth of the water surrounding the tower.
If the wave loading is read from a file (e.g. generated with *ROWS* or an ASCII file, **wave_option 1** or larger) the value of **water_depth** must correspond with the data in this file.

wave_direction `REAL` (0.0) [deg]
Direction of the waves with respect to the global X-axis of the turbine, positive if the waves come from the east and the turbine is oriented to the north, similar as for '**wind_direction**', see Figure 10.

wave_datafile `CHARACTER*20` ('none')
Name of the file with wave data, in ASCII form or as generated with *ROWS*, [3]. If PHATAS is invoked with three (`CHARACTER*64`) command line arguments the third argument is read as name of the wave datafile.
This file is not looked-up for **wave_option 0**.

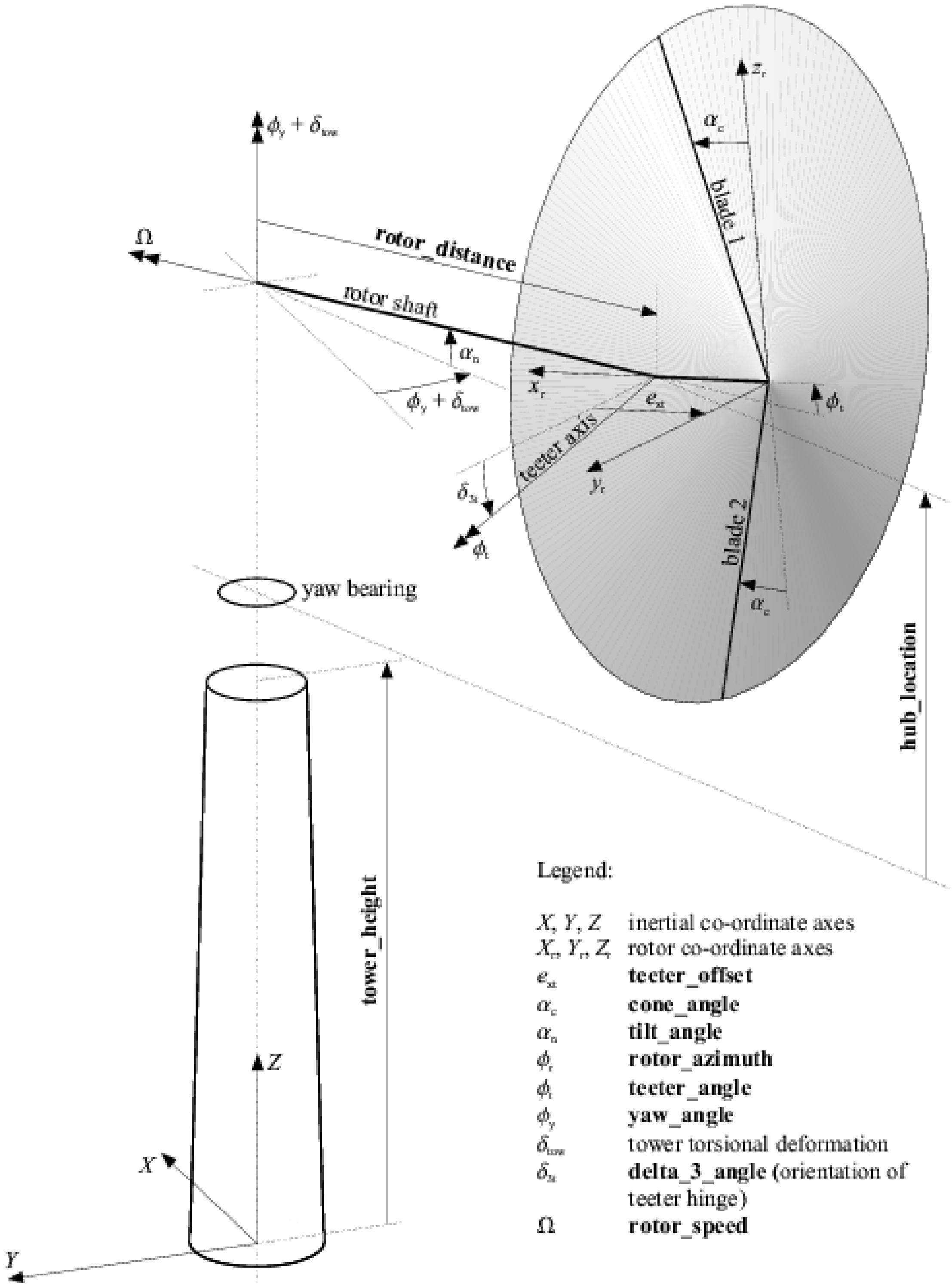


Figure 3: Geometric properties of the wind turbine model (drawn for zero rotor azimuth)

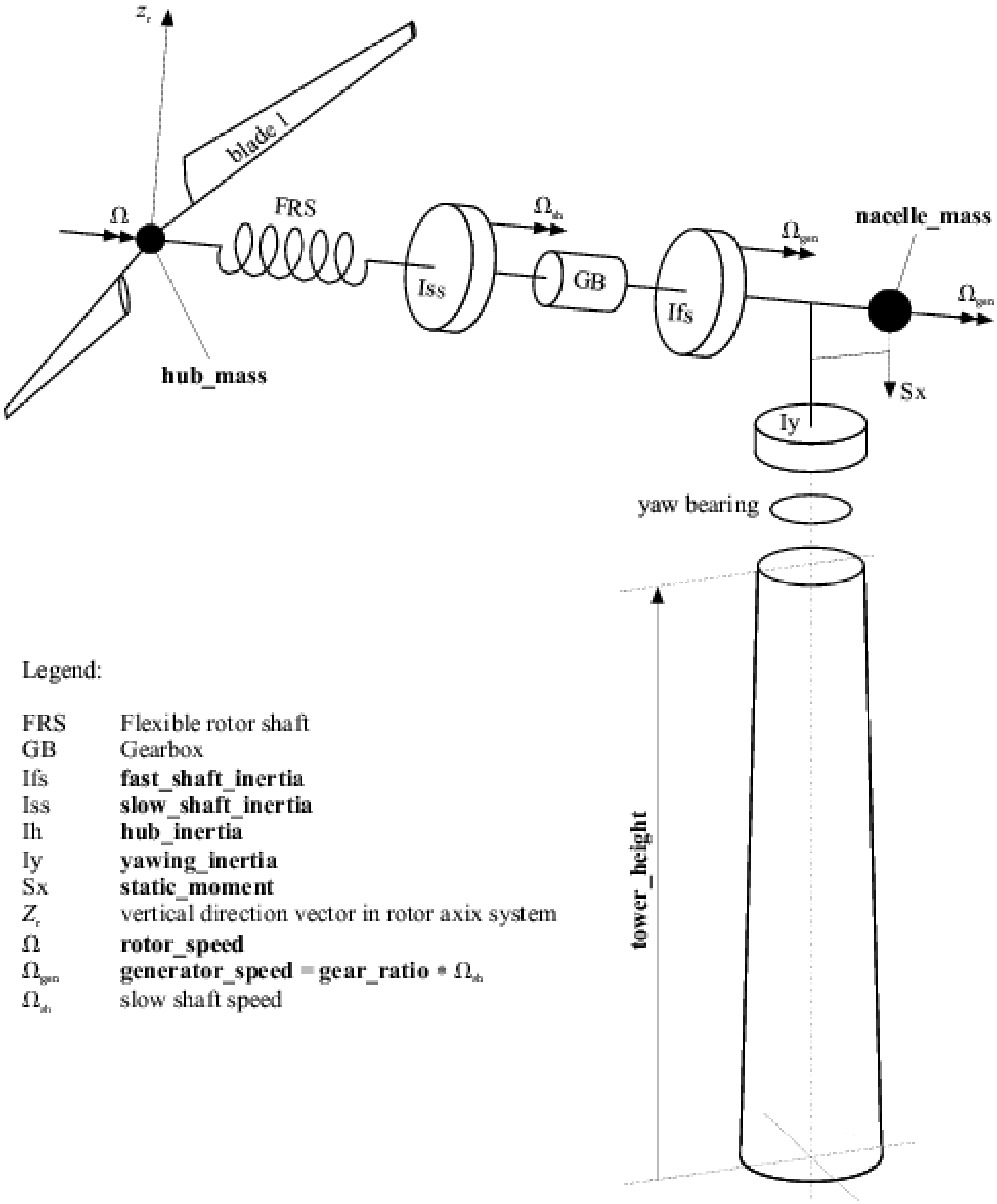


Figure 4: Global mass properties of the wind turbine model

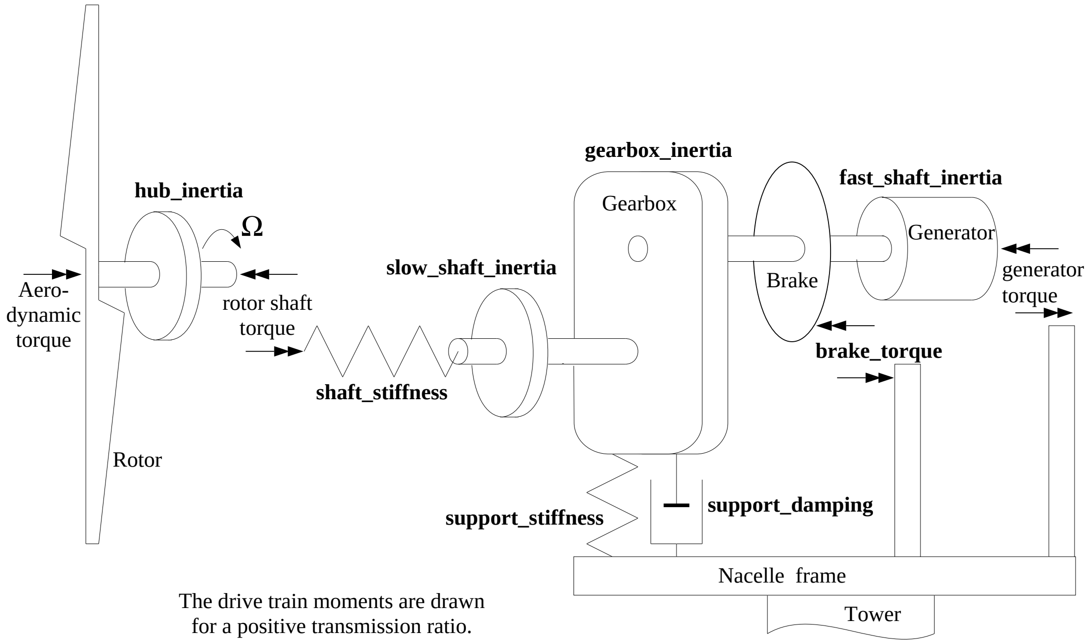


Figure 5: Configuration of the drive train model

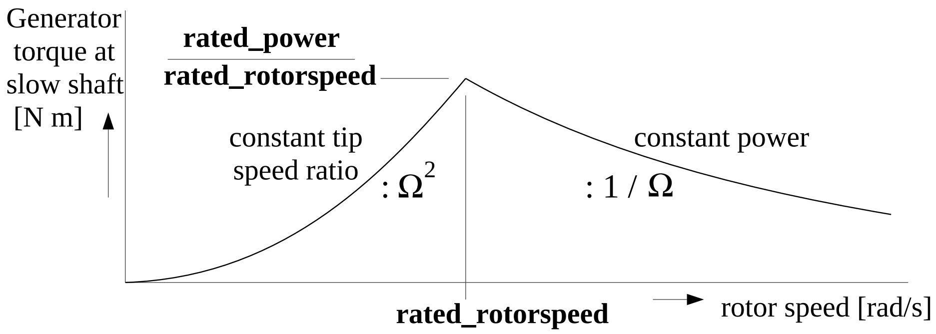


Figure 6: Torque characteristics for generator model 3

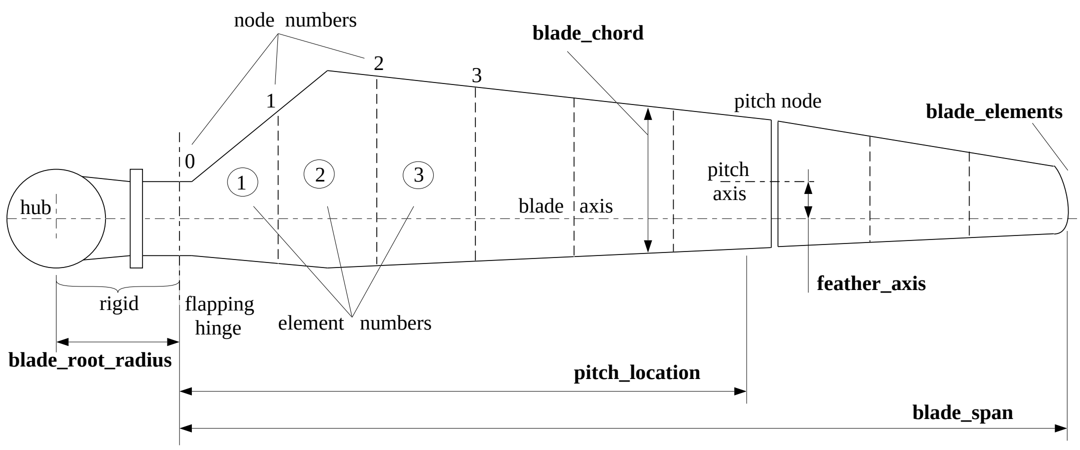


Figure 7: Geometric properties of the blade model

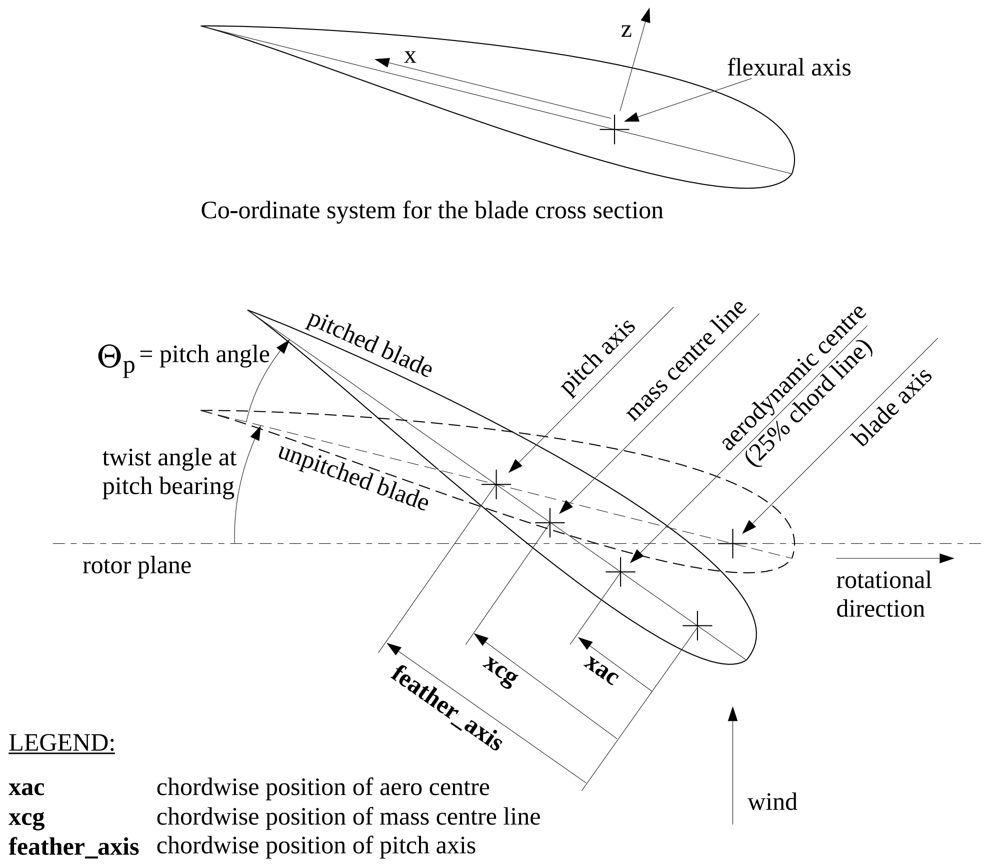


Figure 8: Co-ordinate system used in the blade cross section

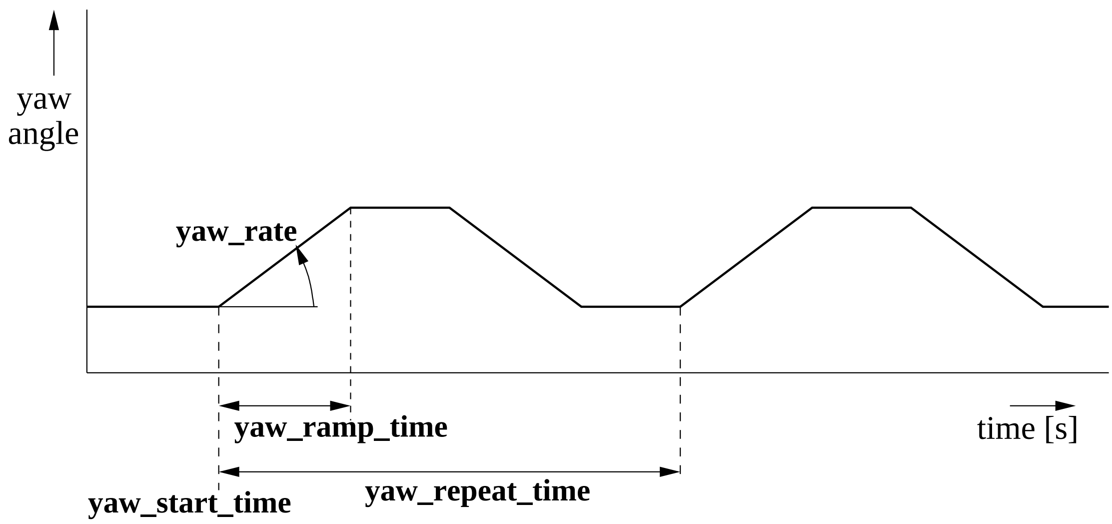


Figure 9: Pre-defined periodic yaw motion in PHATAS

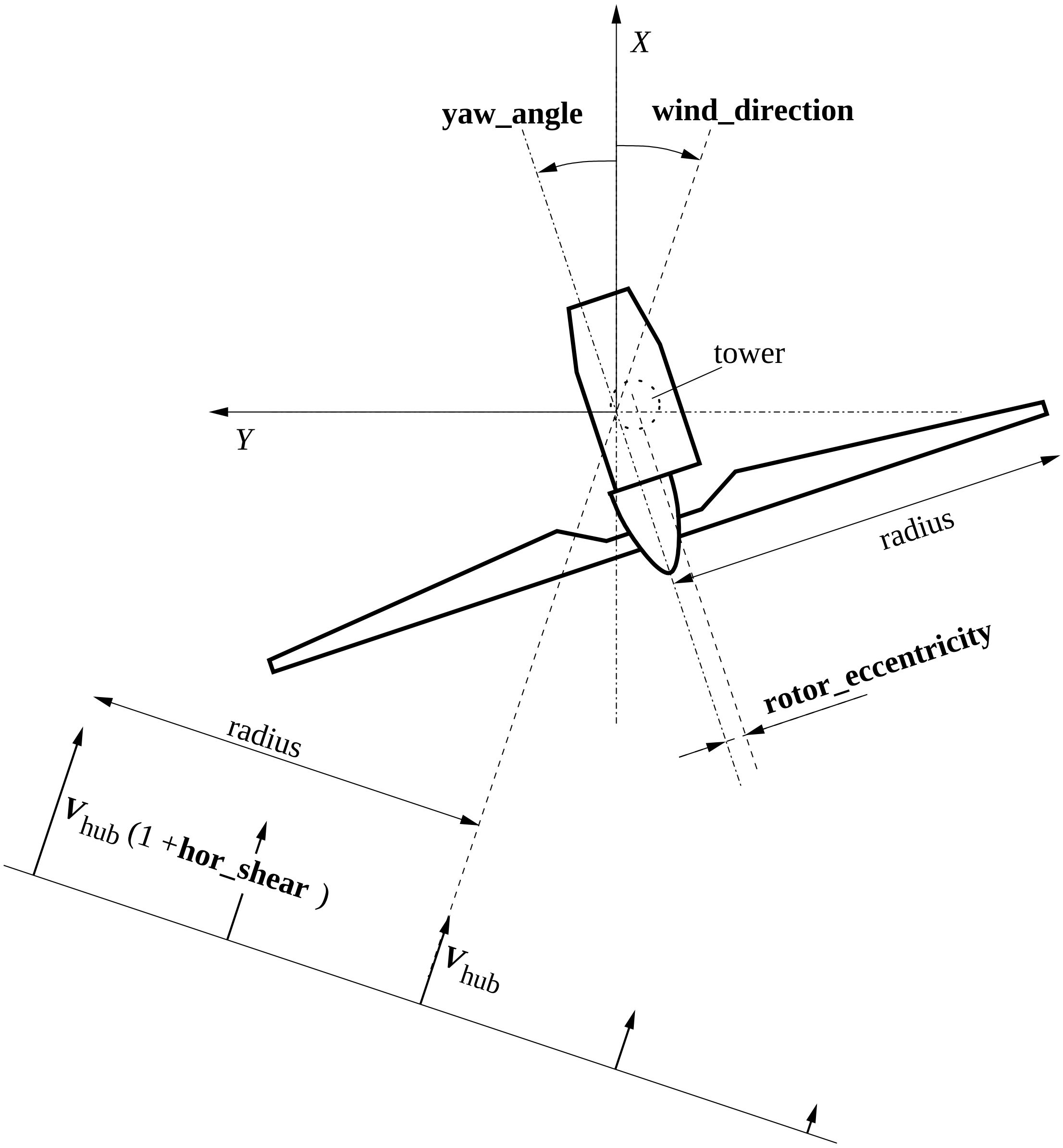
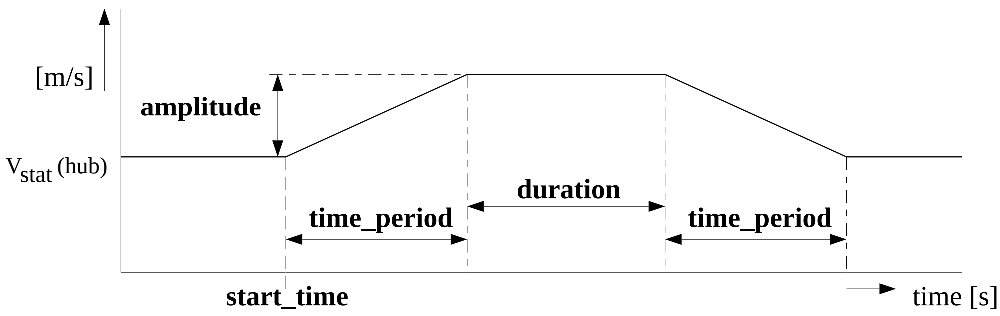
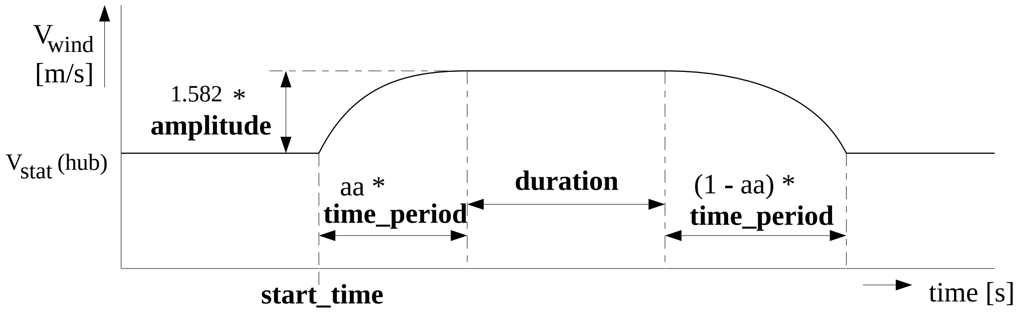


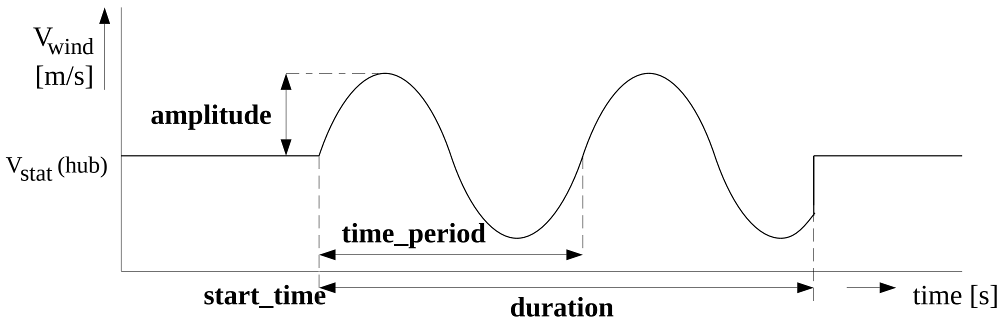
Figure 10: Orientation of the turbine with respect to the wind loading



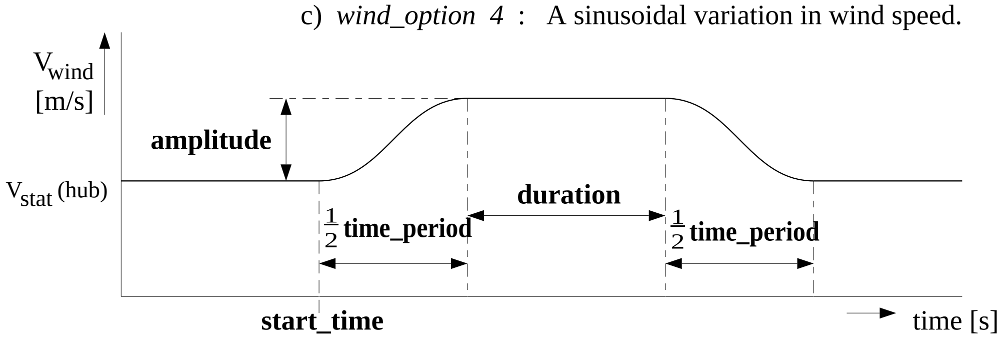
a) *wind_option 2* : A ramp shaped gust.



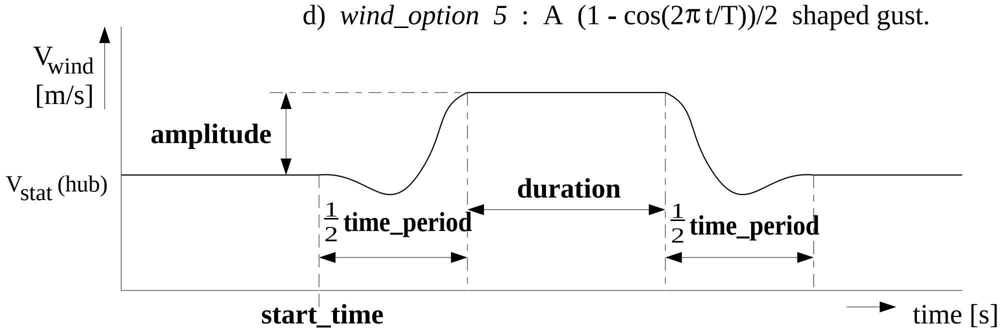
b) *wind_option 3* : A ’Frost’ shaped gust.



c) *wind_option 4* : A sinusoidal variation in wind speed.



d) *wind_option 5* : A $(1 - \cos(2\pi t/T))/2$ shaped gust.



e) *wind_option 6* : A $- \sin(3\pi t/T)\,(1 - \cos(2\pi t/T))/2$ shaped gust.

Figure 11: Some wind gust models available

4 OUTPUT PROPERTIES

When using PHATAS for the calculation of the dynamic (time-) response an output file is generated containing a table with time series of selected properties. The name of this file can be specified with the input item **output_file**. The columns of the table in this file contain the time `REAL`, the number of iterations `INTEGER`, the rotor azimuth in degrees `REAL`, followed by properties specified in the input file by the item **output_table**.
With the postprocessor it is also possible to generate an output file with another format, see the postprocessor input item **output_type** in Appendix C.
The item **output_table** (see 3.2.2 SUBMENU **time_history**) is a `table` type of input item with two arrays:

signal_nr indicates the output properties that are to be printed;

location indicating the coordinate in the main component of the structure.

If a negative value of **signal_number** is specified the corresponding property is written with the opposite sign. An absolute value of **signal_number** higher than 100*(‘**nr_blades**‘ + 1) (such as the default values 999) indicates the end of the specified output properties.

Following is a description of the output properties corresponding to each value of **signal_nr**.
Table 4.1. *General output properties of the wind turbine.*

signal_number	Description of the output property	Unit
0	Time step number (record number for ‘*phpost*’).	
1	Actual time.	[s]
2	Length of the wind speed vector at hub. This is including lateral and vertical components.	[m/s]
3	Aerodynamic power subtracted from the air.	[W]
4	Axial aerodynamic force on the rotor.	[N]
5	Power through the generator shaft.	[W]
6	Actual wind direction at hub, positive, defined following Figure 10. For measured wind or for stochastic wind this includes lateral turbulence at the hub.	[°]
7	Tower bending rate in X direction.	[m/s]
8	Tower bending rate in Y direction.	[m/s]
9	Yaw angle with respect to tower top, defined following Figure 10.	[°]
10	Yaw rate.	[°/s]
11	Rotor azimuth, zero at 12 o’clock.	[°]
12	Rotor (angular) speed.	[rpm]
13	Teeter angle.	[°]
14	Teeter rate.	[°/s]
15	Torsional deformation of the rotor shaft.	[°]
16	First time derivative of (15).	[°/s]
17	Rotor angular acceleration.	[°/s^2]
18	Sum of yaw angle and tower torsion at top.	[°]
19	Sum of yaw rate and rate of tower torsion.	[°/s]
20	Component in rotor-shaft direction of the tower top ’deformation’ velocity.	[rpm]

Table 4.2. *Output properties for the tower and the rotor shaft*

signal_number	Description of the output property	Unit
21	Bending moment in the tower about the inertial X axis.	[N·m]
22	Bending moment in the tower about the inertial Y axis.	[N·m]
23	Torsional (yawing) moment in the tower.	[N·m]
24	Shear force in the tower in the inertial X direction.	[N]
25	Shear force in the tower in the inertial Y direction.	[N]
26	Normal (tensile) force in the tower.	[N]
27	Displacement of the nacelle in the inertial X direction.	[m]
28	Displacement of the nacelle in the inertial Y direction.	[m]
29	Torsional deformation of the tower.	[^{o}]
41	Sideways rolling moment in the tower along the horizontal component of the rotation vector.	[N·m]
42	Backward tilting moment in the tower.	[N·m]
43	Torsional (yawing) moment in the tower top (as 23).	[N·m]
44	Shear force in the tower along the horizontal component of the rotation vector.	[N]
45	Shear force in the tower perpendicular to the rotor axis positive to the left when looking downwind.	[N]
46	Normal tensile force in the tower.	[N]
47	Displacement of the nacelle (direction as 44).	[m]
48	Displacement of the nacelle (direction as 45).	[m]
49	Torsional deformation of the tower.	[^{o}]
51	Torque on the drive train support.	[N·m]
52	Tilting moment on the rotor, about $Y_{\rm GL}$ axis.	[N·m]
53	Yawing moment on the drive train.	[N·m]
54	Axial (compressive) force on the drive train.	[N]
55	Lateral force (in Y direction) on drive train.	[N]
56	Vertical (upwards) shear force on the drive train, Figure 14.	[N]
57	Torsional deformation of the flexible gearbox support.	[^{o}]
61	Torque on the rotor shaft.	[N·m]
67	Torsional deformation of the flexible rotor shaft.	[^{o}]
71	Rotor shaft torque transmitted by one gearbox tooth. Here '**location**' indicates the rotor azimuth [^{o}] for which the particular tooth is in contact. This can be used to analyse fatigue of gearbox teeth.	[N·m]
91 - 99	User defined (e.g. from '*conrol*') output variables.	[?]

The output properties 21 to 49 are for a cross-section of the tower at a height '**location**' [m] above the tower base. The positive directions of the properties 21 to 29 are displayed in Figure 12.

The properties 41 to 49 are similar to the properties 21 to 29 but defined with respect to the nacelle reference system; after yawing and tower torsional deformation, see Figure 13.

The properties 51 to 56 are for the loads from the rotor to the drive train defined in the non-rotating coordinate system at a distance '**location**' The properties 51 to 56 are for the loads in the non-rotating rotor shaft at a distance '**location**' behind the hub, as is displayed in Figure 14.

The properties 62 to 66 are the loads in the rotating rotor shaft of which the directions are similar to those of properties 52 to 56 for a zero rotor azimuth, see Figure 15. The difference between the torque '61' and '51' comes from the angular acceleration of the drive train components.

If the user-defined properties '91' through '99' are not assigned in routine *conrol*, they have the following 'default' values:

91	0.0	
92	For (**control_type** 1) measured power	[W]
	For (**control_type** 2) the filtered rotor speed	[rpm]
93	Target pitch rate from the controller	[°/s]
94	Lead moment Mx in blade root 3 (for 3 blades)	[Nm]
95	Flap moment My in blade root 3 (for 3 blades)	[Nm]
96	Lead moment Mx in blade root 2	[Nm]
97	Flap moment My in blade root 2	[Nm]
98	Lead moment Mx in blade root 1	[Nm]
99	Flap moment My in blade root 1	[Nm]

The properties with numbers larger than 100 are blade related properties. The first digit of these numbers denotes the blade number of the specific output property. Blade 1 is used to define the rotor azimuth, blade 2 is following blade 1 which means that it's azimuth is always smaller. For a 3-bladed rotor blade 3 is following blade 1. The last two digits are significant for the type of output property which is the same for all blades. The properties in table 4.3 refer to blade 1. Blade properties of which the last two digits are larger than 10 are a function of the location '**location**' which is measured from the blade root: '**blade_root_radius**'.

Table 4.3. *Blade related output properties*

signal_number	Description of output property	Unit
102	Hinge flapping angle.	[°]
103	Rate of hinge flapping motion.	[°/s]
104	Pitch angle.	[°]
105	Pitch rate.	[°/s]
111	Axial induced velocity on the element; not in '*phpost*'.	[m/s]
112	Aerodynamic angle of attack between the relative wind and the blade chord; not in '*phpost*	[°]
113	Time derivative of the angle of attack made dimensionless by: (blade_chord)/(rel. wind speed); not in '*phpost*'.	[.]
114	Relative wind speed on a blade section; not in '*phpost*'.	[m/s]
115	Strength of the bound vorticity; not in '*phpost*'.	[m^2/s]
116	Value of the lift coefficient c_l; not in '*phpost*'.	[.]
117	First time derivative of the lift coeffic.; not in '*phpost*'.	[1/s]
118	Normal force distribution N ; not in '*phpost*'.	[N/m]
119	Tangential force distribution T ; not in '*phpost*'.	[N/m]
137	Flap displacement in the rotor plane reference system. This is including the 'pre-bend' in the blade, see 3.6.	[m]
139	Lead displacement in the rotor plane reference system. This is including the 'pre-bend' in the blade, see 3.6.	[m]
141	Lead moment in the blade.	[N·m]
142	Torsional moment in the blade.	[N·m]
143	Upwind bending moment in the blade.	[N·m]
144	Flapwise shear force in the blade.	[N]
145	Spanwise force in the blade.	[N]
146	Leadwise shear force in the blade.	[N]
148	Torsional blade deformation '*delta*'.	[°]
999	Indicates the end of the output table (999 = default).	

The positive directions of the loads with respect to the deformed blade cross-section (not accounting for pitch, twist and torsional deformation), 141 to 146 are displayed in Figure 17.

The loads 131 to 136 are similar to the loads 141 to 146 but defined with respect to the undeformed rotor plane reference system, see Figure 16.

The loads 151 to 156 are in directions similar as for 141 to 146 but expressed in the local blade chord reference system, see Figure 18. Compared to the loads 141 to 146 these load vectors are also transformed for twist, for blade torsion and if '**location**' is larger than '**pitch_location**' also for blade pitch. The flatwise and edgewise deformations 157 and 159 are obtained from the deformations 147 and 149 after rotation to the local blade chord reference system, see Figure 18.

The loads 161 to 166 are in directions similar as for 151 to 156 but expressed in the coordinate system through the blade pitch axis, that may have an offset '**feather_axis**' from the blade axis.

The blade torsional deformation is only defined in the deformed blade reference system, indicated by the numbers 148, 158, 168, or 178.

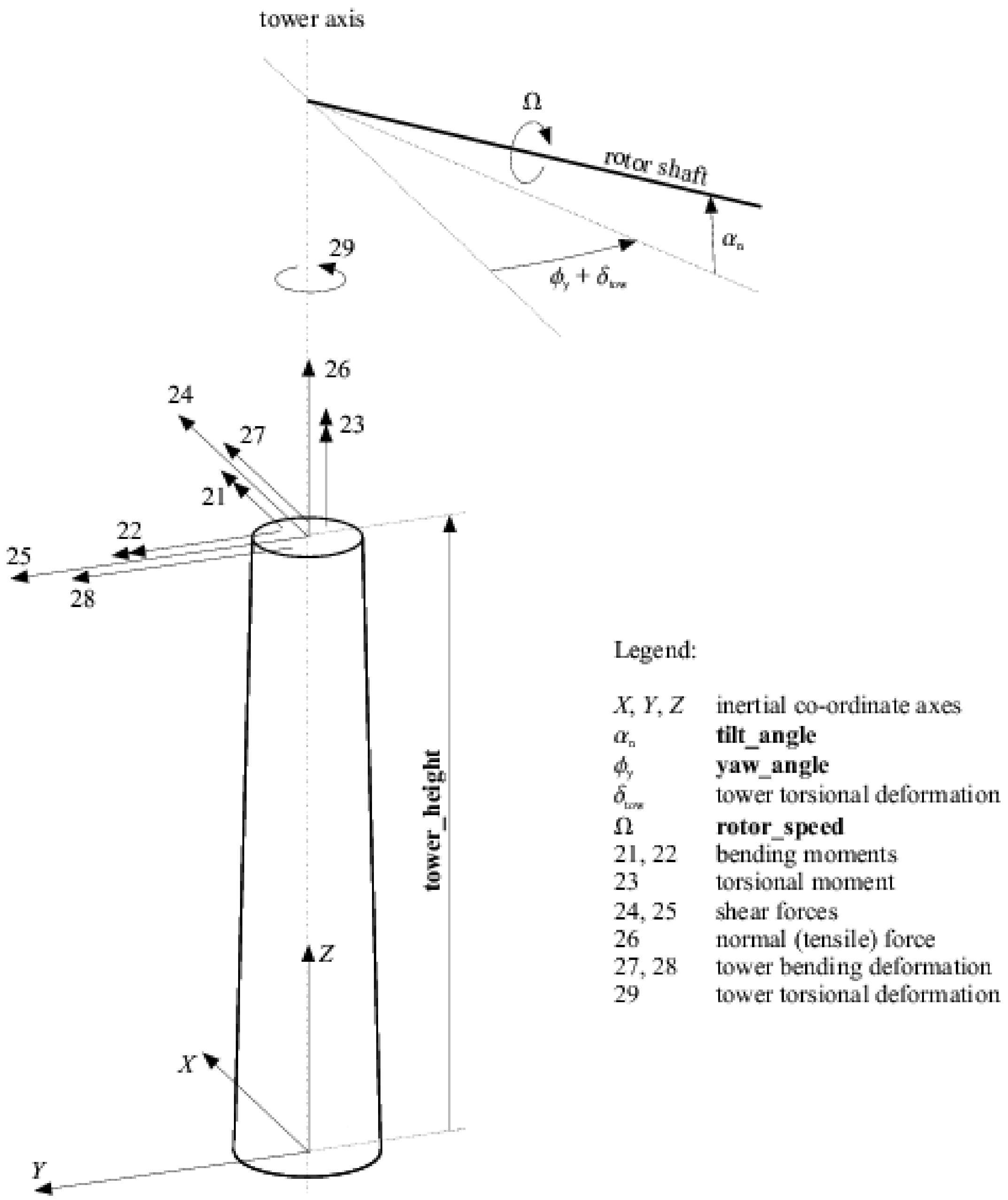


Figure 12: Tower loads in the inertial reference system

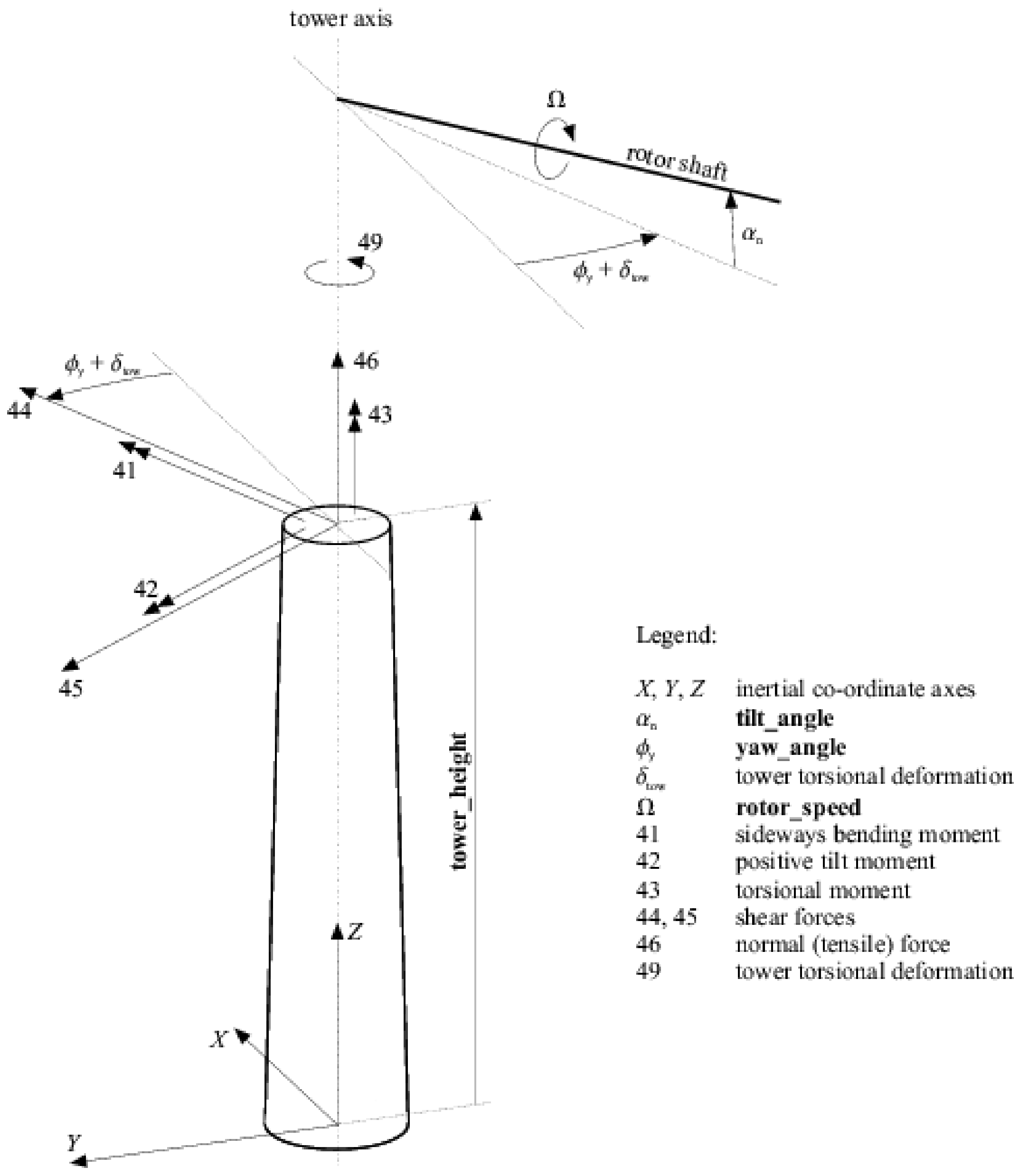


Figure 13: Tower loads in the ‘nacelle’ reference system

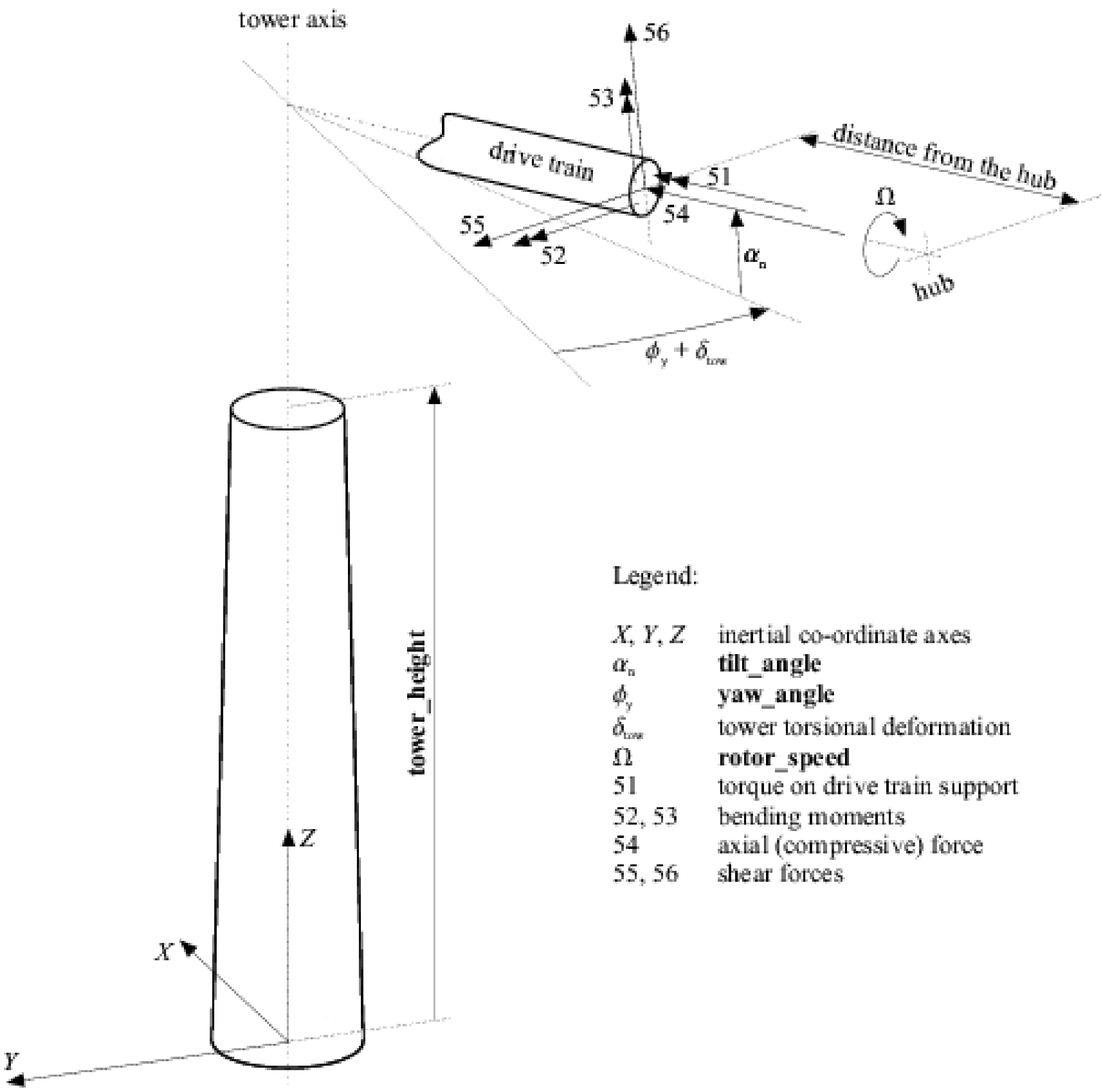


Figure 14: Loads on the drive train in the non-rotating reference system

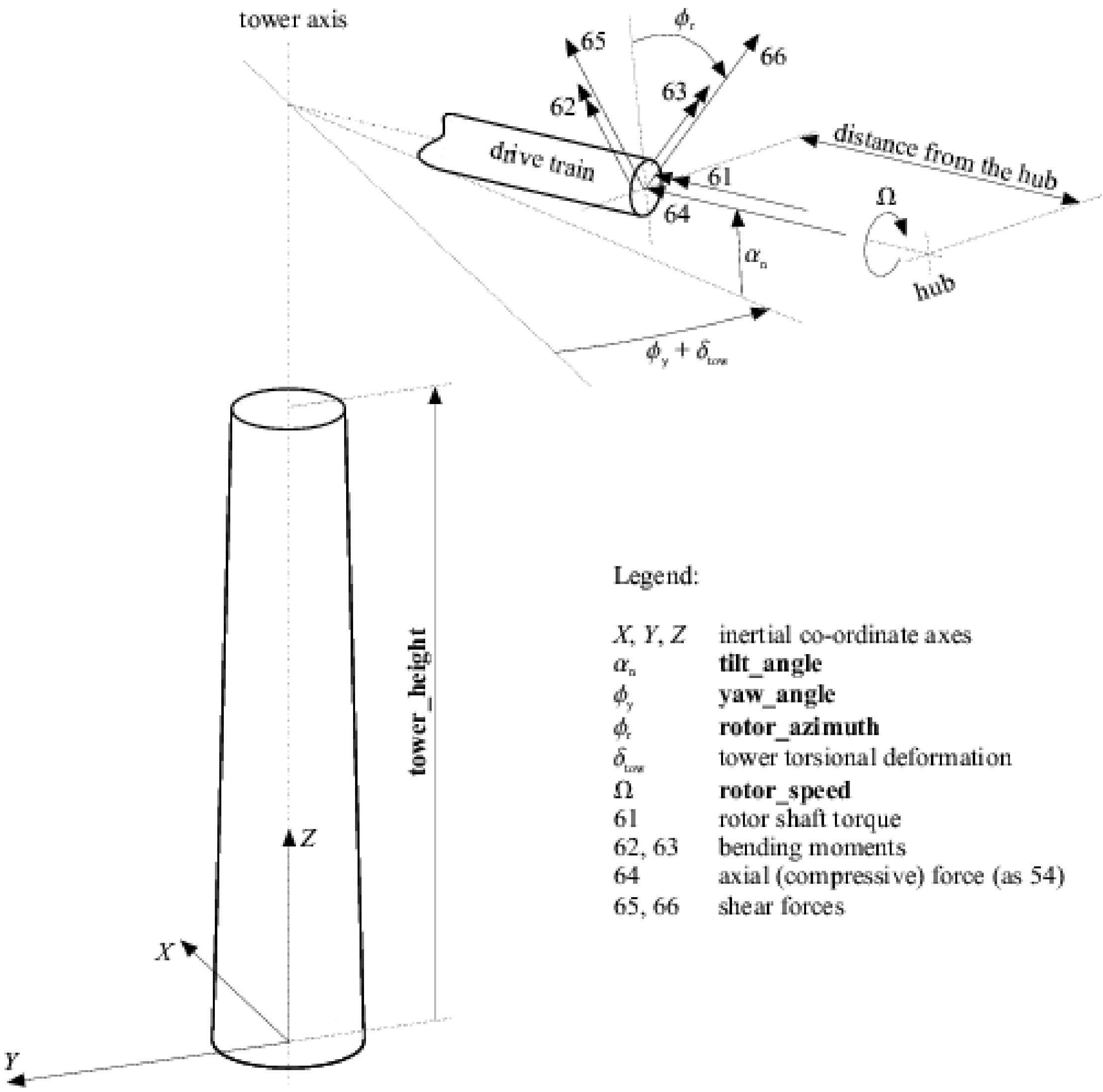
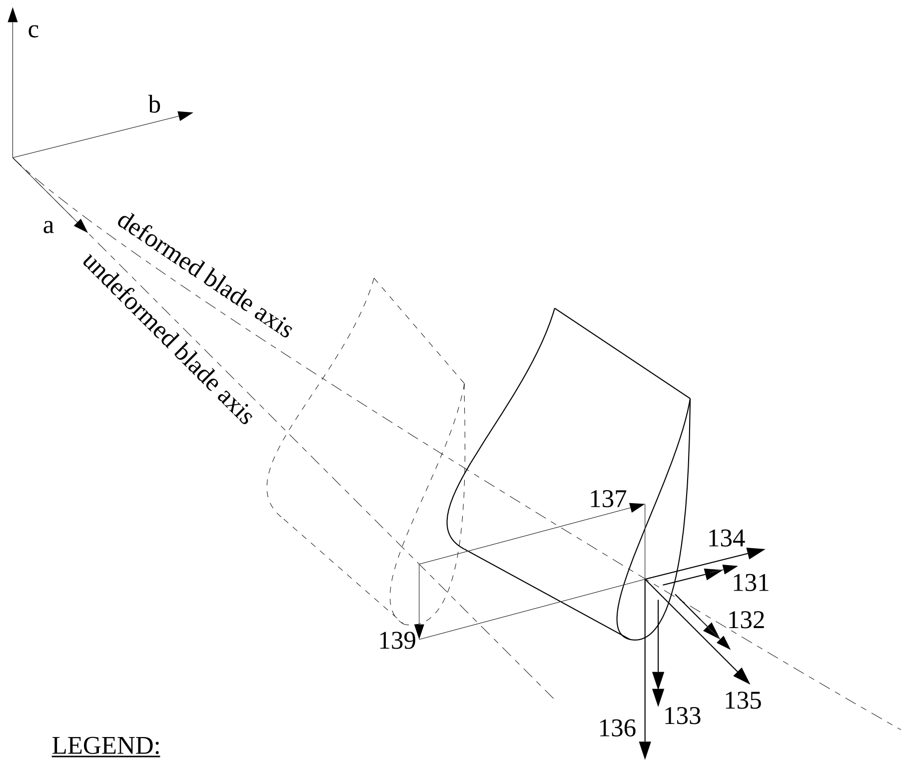


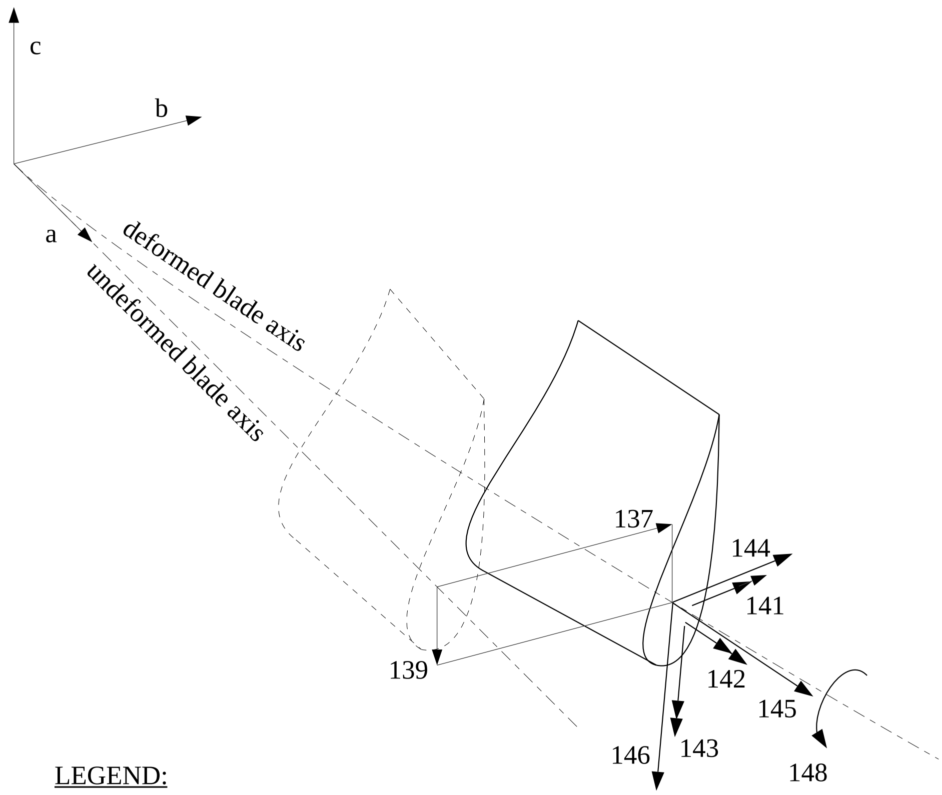
Figure 15: Loads on the rotating rotor shaft



LEGEND:

a, b, c local blade direction vectors
131 leadwise bending moment
132 blade torsional moment
133 - flapwise bending moment
134 flapwise shear force
135 spanwise (tensile) force
136 leadwise shear force
137 flapwise deformation
139 leadwise deformation

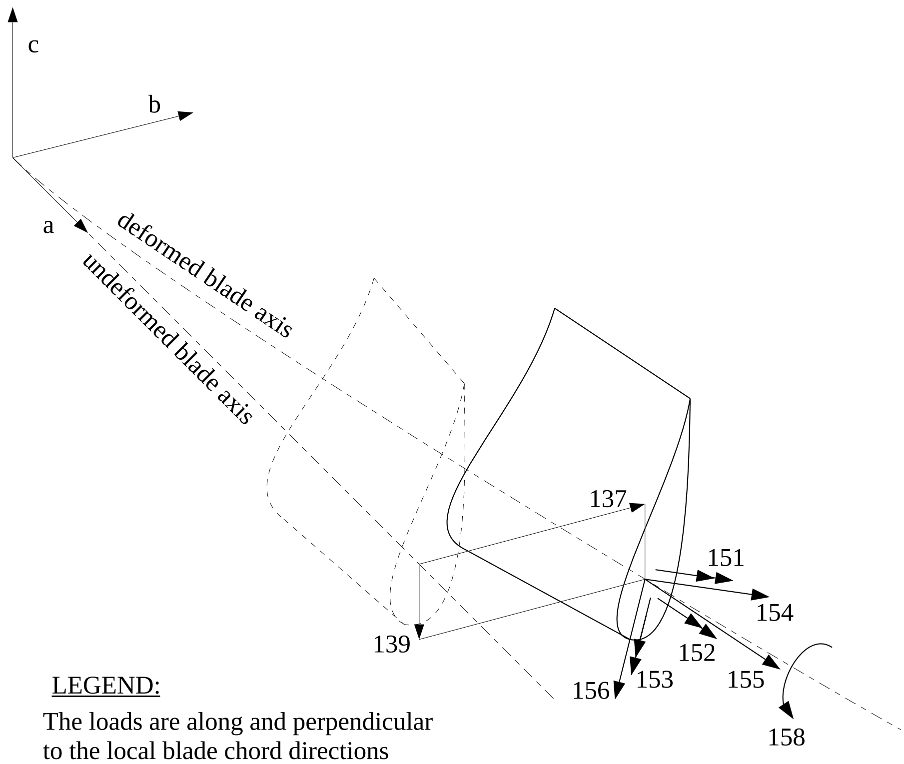
Figure 16: Blade properties in the undeformed rotor plane reference system



LEGEND:

a, b, c local blade direction vectors
141 leadwise bending moment
142 blade torsional moment
143 - flapwise bending moment
144 flapwise shear force
145 blade axial (tensile) force
146 leadwise shear force
148 torsional deformation
137 flapwise deformation
139 leadwise deformation

Figure 17: Blade loads in the deformed rotor plane reference system



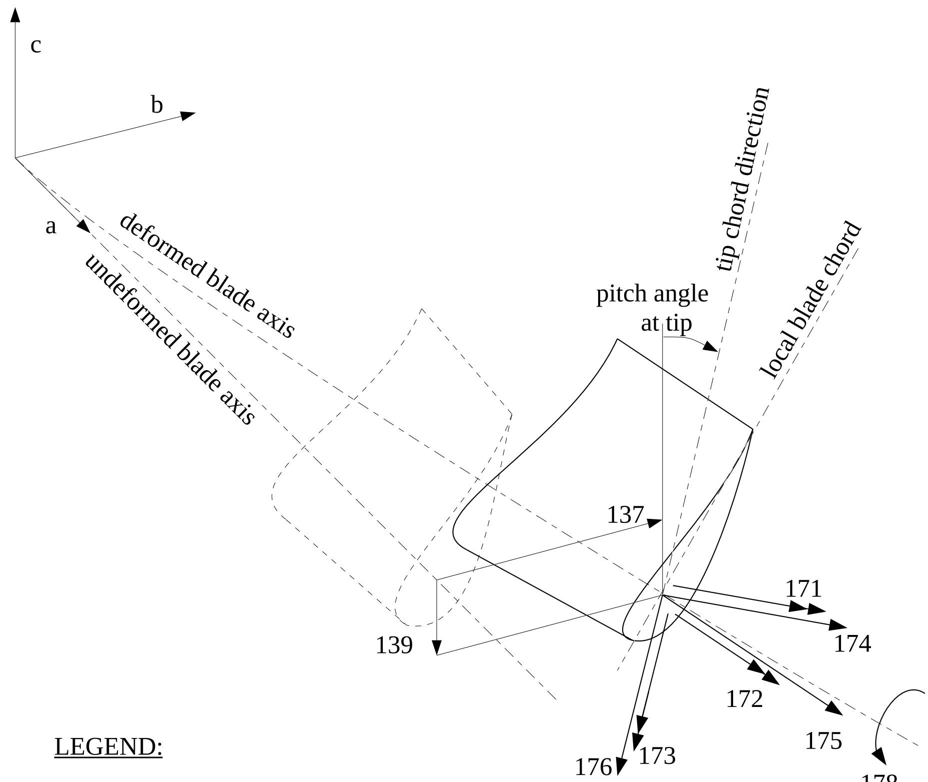
<u>LEGEND:</u>

The loads are along and perpendicular to the local blade chord directions

a, b, c local blade direction vectors

151 edgewise bending moment
152 blade torsional moment
153 - flatwise bending moment
154 flatwise shear force
155 blade axial (tensile) force
156 edgewise shear force
158 torsional deformation
137 flapwise deformation
139 leadwise deformation

Figure 18: Blade loads in the local blade chord reference system



LEGEND:

a, b, c local blade direction vectors
171 edgewise bending moment
172 blade torsional moment
173 - flatwise bending moment
174 flatwise shear force
175 blade axial (tensile) force
176 edgewise shear force
178 torsional deformation
137 flapwise deformation
139 leadwise deformation

Figure 19: Blade loads in the tip chord reference system

5 MODULE FOR TOWER DYNAMICS

5.1 Model Description

In the program PHATAS the tower dynamics are described in terms of modal response that is solved in separate routines. These routines are grouped in the parts:

1 Routines that read or derive the modal properties for tower deformation.

2 Routines that solve the dynamic response in terms of modal amplitudes.
In these routines also the loads from wind and waves are calculated, as well as the aerodynamic tower stagnation used for the blade loads.

3 Routines that read the 'wave field' in the format as generated with the program *ROWS*, '**R**andom **O**cean **W**ave **S**imulation' or with the program *Streamfunction*. With *ROWS* one can generate a stochastic wave field in time, based on the JONSWAP spectrum [3]. With *Stream function* one can generate a deterministic non-linear wave [18].

For the first part of the 'tower module', two versions are available that can be linked to the PHATAS turbine model. For both versions the dynamic tower deformation is described following the Craig-Bampton method: with 6 'constraint modes' (for static tower top translation and -rotation) and up to 12 internal modes. The latter modes are eigenmodes with the 'interface' (tower top) fixed.
The executable names of the PHATAS versions with the different tower modules are:

phatas The tower specified in terms of sectional properties of a monopile tower, with input in terms of engineering properties of beams or with input in terms of tubular sections;

phatmod For more complicated tower structures of which a finite element model has been made, the modal properties (modal mass, modal stiffness etc.) defined following the Craig-Bampton method [2] can be used as input for *phatmod*.

For both modules the name of the input file for the tower module should be specified with the PHATAS input item **tower_input** in MENU **configuration**.

In the following sections the input properties are described for both these models.

5.2 Input file for a Monopile Tower

This module is developed for the analysis of the dynamic response of a monopile tower that can be partially submerged, such as for offshore support structures. The foundation of the tower can be flexible in five directions: three rotations and two horizontal translations. The wave loading and inertia of the surrounding water is included in the calculation of the dynamic loads.

The modes for steady tower top deformation ('constraint modes') and the internal modes are calculated during initialisation of the tower model in a routine based on integration of the bending and torsion equations. The calculation of the modes is including the effect of the weight of the tower and turbine and including transverse shear deformation. For these modes, the mass and stiffness matrices are calculated and used in PHATAS to solve the structural dynamic response.

For this mass and stiffness matrix and with the mass and inertia of the turbine on the tower top, the eigenvalues are calculated and written in file '*towmod.out*'. For the condition with disconnected generator (idling) or for wind turbines with a variable speed generator characteristics these eigenvalues are also calculated without the rotor rotational inertia.

For monopile towers (in the standard version *'phatas'*) the sectional properties can be defined by either the properties of conical tubular sections, or by the engineering properties of a beam.

If the tower is given in terms of conical sections this has to be specified in a TABLE after the key-name **tower_segments**. The subsequent records following **tower_segments** should contain the properties from the base to the top, with on each record the following seven REAL variables of a tube segment (see also Figure 20):

1 Height [m] of the segment upper end, measured from the tower base;

2 Outer diameter [m] of the segment lower end;

3 Outer diameter [m] of the segment upper end;

4 Wall thickness [m], for conical segments measured perpendicular to the wall;

5 Young's modulus [N/m^2];

6 Shear modulus [N/m^2]. The shear modulus is used for the torsional mode
and for the shear deformation in the bending modes;

7 Mass density [kg/m^3]. The mass of interior parts of the tower can be included by
a mass density for steel 8000 kg/m^3 .... 9000 kg/m^3 instead of 7850 kg/m^3.

All records in this table should end with a comma "," except for the last record.

If the tower is given in terms of engineering properties, these should be specified in a TABLE after the key-name **sectional_data**. The subsequent records following **sectional_data** should contain the properties from the base to the top, with on each record the following eight REAL variables:

1 Height [m] (cumulative) of the upper segment end, measured from the tower base;

2 Outer diameter or width [m] of the lower end of the segment;

3 Outer diameter or width [m] of the upper end of the segment.
This diameter is used to calculate the aerodynamic tower stagnation
and the loads from the wind and waves;

4 Sectional mass [kg/m];

5 Bending stiffness [N m^2];

6 Torsional stiffness [N m^2], used for the torsional mode;

7 Longitudinal tensile stiffness [N];

8 Transverse shear stiffness [N].
Especially for the higher bending modes the transverse shear flexibility reduces the bending frequencies. If transverse shear flexibility is not to be modelled, one should specify a value much larger than the bending stiffness times the beam-diameter squared.
For a tubular tower with radius R and wall thickness t the shear stiffness is $\pi\ G\ t\ R$.

The tower model uses the properties from the last table that is read from the input file (either **tower_segments** or **sectional_data**). For these tables a maximum number of 48 records or 'tower segments' is allowed. The diameter of the tower sections is also used to describe the additional mass from the interior and surrounding water. For the torsional inertia of the tower cross sections the sectional mass is assumed to be located/concentrated at the tower diameter.

Following is a description of the remaining input items of the monopile tower module including the **key-name**, the `type`, the (default value), the [unit], and the <allowable range>.

tower_dynamics `LOGICAL` (OFF)
: If '*ON*' the tower structural dynamic deformation is solved.
If '*OFF*' only the tower geometry is used for the aerodynamic stagnation, while still 6 'rigid' modes are used that describe the sectional loads in the tower due to the forces and moments from the turbine.

root_flexibility `LOGICAL` (OFF)
: If '*ON*' the elasticity of the foundation is modelled using '**root_stiffness**'.

root_stiffness `TABLE`
: On the record following this key-name four or five `REAL` values are read:

1 Rotation stiffness k_ϕ (1.E+30) [N·m/rad] for bending ('rolling') of the base;

2 Linear stiffness k_u (1.E+30) [N/m] for (fore-aft) translation of the base;

3 Coupling f (0.0) [m] between 'rolling' and translation of the base;

4 Torsional stiffness (1.E+30) [N·m/rad] of the base;

5 Linear stiffness k_v (1.E+30) [N/m] for side translation at the base.
This 'fifth' value was added to analyse the tower frequency of the MEXICO wind tunnel model. If the record following after **root_stiffness** contains only four values, the side-translation stiffness is set equal to the fore-aft translation stiffness.

With the root stiffnesses the deformations *u* and ϕ are related to the loads by

$$\begin{pmatrix} \phi \\ u \end{pmatrix} = \begin{pmatrix} 1/k_\phi & f/k_\phi \\ f/k_\phi & 1/k_\mathrm{u} \end{pmatrix} \cdot \begin{pmatrix} M_\mathrm{bend} \\ F_\mathrm{fore-aft} \end{pmatrix} .$$

The root stiffnesses are only used for '**root_flexibility ON**' and '**tower_dynamics ON**'.
For monopile wind turbine towers on a soft soil Stig Øye found that the rotational ('rolling') stiffness k_ϕ of the foundation has roughly the same effect on the bending frequency as a 1 diameter elongation below the base.

bending_modes `INTEGER` (0) <0, ..., 6>
: Number of internal ('elastic') modes for bending in both lateral and fore-aft direction.
In addition to this, the deformation is also described with the 6 'constraint modes'.
For a realistic description of the higher-order bending modes and/or for a good representation of the internal load distribution in the tower it is recommended to use 2 (or more) internal modes. For off-shore wind turbines subjected to wave loading at the lower part of the tower, it is recommended to use about 5 or 6 internal bending modes in each direction.

damprat_bend `REAL` (0.0) <-0.1, ..., 1.1>
: Damping of the eigen-modes for bending, divided by the critical damping.

damprat_tors `REAL` (0.0) <-0.1, ..., 1.1>
: Damping of the torsion mode, divided by the critical damping.

guy_wires `INTEGER` (0) <0, 2, ...>
: Number of guy wires or support-bars that give lateral support to the tower.
For a non-zero tension ('**wire_tension**') the minimum for **guy_wires** is 2.

connect_height `REAL` (1.0) [m] $<$0.0, ...$>$
Height above the foundation at which the guy wires are connected to the tower.

connect_radius `REAL` (0.0) [m] $<$0.0, ...$>$
Radius at which the guy wires are connected to the tower.

wire_angle `REAL` (45.0) [^{o}] $<$0.0, ...$>$
Angle of the guy wires with respect to the horizontal, positive if the wires go down.

wire_stiffness `REAL` (0.0) [N/m] $<$0.0, ...$>$
Axial stiffness of one guy wire.

guy_bend_stiff `REAL` (0.0) [N/m] $<$0.0, ...$>$ *Not for release "NOV-2003".*
Bending stiffness at the tower connection of one guy-wire.
This can be used to model towers with e.g. a tripod base, where one would speak about 'support bars'. For a support bar with length L that is not clamped at its end, with a sectional bending stiffness EI, the stiffness at the tower connection is $K = 3 \cdot EI/L^2$.

wire_tension `REAL` (0.0) [N] $<$0.0, ...$>$
Tension in the wire-direction of an individual wire, for undeformed tower.

lattice_tower `LOGICAL` (OFF)
If 'OFF' the aerodynamic stagnation is calculated for a tubular tower.
If 'ON' the aerodynamic stagnation is calculated for a lattice tower while the inertia terms in the Morison equation for wave loading are not included.

air_drag `REAL` (0.7)
Drag coefficient used to calculate the wind force on the tower.

water_drag `REAL` (1.2)
Drag coefficient for the waves on the tower.
The drag of a cylinder depends on Reynolds number $Re_\mathrm{D} = v_\mathrm{wave} \cdot D/\nu$
where the kinematic viscosity $\nu = 1.5 \cdot 10^{-5}$:

Re_D	Drag coefficient
$< 10^4$	roughly 2
$10^4 ... 10^5$	1.2
10^6	0.3
$\geq 10^7$	0.7

See also Hoerner [7] p.3-9 and Anderson [1] p.229.

water_mass_coeff `REAL` (1.0) $<$0.0, ..., 2.0 $>$
Coefficient of the "added mass" of the accelerating water around the tower; C_a .
This is also called 'hydrodynamic mass'. In PHATAS the inertia loads of the moving tower are calculated with the 'inertia coefficient' of added mass $C_\mathrm{M} = C_\mathrm{a} + 1$.
The difference of 1 is theoretically and experimentally determined. The theoretical value of the coefficient C_a for a potential flow is 1. Because of flow separation this coefficient may have a smaller value where negative values are considered unrealistic. It appears that C_a is a function of the so-called Keulegan Carpenter number $KC = v_\mathrm{a} \cdot T/D$, where v_a is the amplitude of wave motion, D the diameter, and T the oscillating flow period.
For lattice towers (**lattice_tower ON**) the inertia loads from water mass are not included.

An example of the input file for a 98m high tower with a tripod base is listed in figure 20. This file contains most input items with comments.
The character `<` , `#` , or `/` in the input file indicates that comments follow on that line.

```
<  PHATAS tower input file for a 98m high tower with tripod base.
<
tower_dynamics    ON
bending_modes     3     <  Maximum 6 internal elastic modes.
<
<  Damping of the bending modes as fraction of critical.
damprat_bend     0.00159  < decrement = 1%  = 2*pi*0.00159
damprat_tors     0.00159  < 1%
<
lattice_tower    OFF
air_drag         0.7
water_drag       0.7   < 0.7 is valid for Re_D > 3.5E6,
water_mass_coeff 1.0   < The amount of moving-water has the same
                       < volume as the submerged part of the tower.
<
root_flexibility   ON
< The first element will reserved for flexibility at root
< for which the stiffness properties are given by  "root_stiffness".
<
< Stiffnesses of the tower base:
<   1.  Rotation stiffness [Nm/rad] for bending of root;
<       2.  Linear stiffness [N/m] for translation of root;
<           3.  Coupling [m] between rotation and translation;
<               4.  Torsional rotation (bending) at root [Nm/rad].
root_stiffness
 10.0E+9  1.0E+30  0.0  1.0E+30  < Low value, for centre of tripod
<
<  Specification of 3 guy bars with 1.5m diameter and 30mm wall.
<  Axial stiffness = 951.337E+6N/m
<  Bending stiffness of a cross section:  EI = 7.71225E+9Nm^2
<  Stiffness measured at the end of 30m long bar:  771.225E+6Nm/rad
guy_wires         3
connect_height   15.0    < [m] Above tower base.
wire_angle       30.0    < [deg] With respect to the horizontal.
connect_radius    2.25   < [m] Radius at which guys are connected.
wire_stiffness  951.337E+6 < [N/m] axial stiffness.
guy_bend_stiff  771.225E+6 < [Nm/rad] bending stiffness of one guy.
<
< Table with specification of the segments of a tubular tower.
<       Dia  Dia
< Hcum  low  high  Thick EI_young G_shear  density
<  [m]  [m]  [m]   [m]   [N.m^2]  [N.m^2] [kg/m^3]
tower_segments
  5.0   2.5  3.5   0.020  206.E+9 80.E+9 8100.0,
 10.0   3.5  4.5   0.030  206.E+9 80.E+9 8100.0,
 13.5   4.5  4.5   0.040  206.E+9 80.E+9 8100.0,
 16.5   4.5  4.5   0.045  206.E+9 80.E+9  8200.0,
 20.0   4.5  4.5   0.038  206.E+9 80.E+9 8150.0,
 25.0   4.5  4.5   0.034  206.E+9 80.E+9 8150.0,
 30.0   4.5  4.5   0.032  206.E+9 80.E+9 8150.0,
 35.0   4.5  4.5   0.030  206.E+9 80.E+9 8150.0,
 40.0   4.5  4.5   0.028  206.E+9 80.E+9 8150.0,
 45.0   4.5  4.5   0.026  206.E+9 80.E+9 8150.0,
 49.94  4.5  4.5   0.024  206.E+9 80.E+9 8150.0,
 50.06  4.5  4.5   0.250  206.E+9 80.E+9  8500.0,
 62.0   4.5  4.0   0.022  206.E+9 80.E+9 8150.0,
 73.94  4.0  3.5   0.020  206.E+9 80.E+9 8150.0,
 74.06  3.5  3.5   0.200  206.E+9 80.E+9  8500.0,
 83.6   3.5  3.1   0.018  206.E+9 80.E+9 8150.0,
 90.8   3.1  2.8   0.016  206.E+9 80.E+9 8150.0,
 97.92  2.8  2.5   0.014  206.E+9 80.E+9 8150.0,
 98.0   2.5  2.5   0.150  206.E+9 80.E+9  8500.0
<
```

Figure 20: Input file for a 98m high tower with tripod base

5.3 Input in terms of mass and stiffness matrices

The version of the tower module with input in terms of mass and stiffness matrices is linked in the PHATAS executable '*phatmod*', because of the following differences with respect to the other module:

- The input for this module represents only the dynamic interaction at the tower top, so most other aspects (foundation properties, guy wires/bars are only represented indirectly in the mass and stiffness matrices.
- Following the 'basic' approach of the Craig-Bampton method the mode shapes are not known, although they are needed if the wind and/or wave loads on the tower are to be included.

The PHATAS version '*phatmod*' has been used (with success) for the dynamic load calculations of turbines with a lattice tower for land locations. For this tower and location the wind and wave loads were less significant, so that the wind loading could be calculated with 'estimated' mode shapes. For more solid tower structures and for offshore towers the program '*phatmod*' probably has to be modified.

The version '*phatmod*' was developed for tower input in terms of mass and stiffness matrices. In this version the mode shapes of the 6 constraint modes are rough estimations. For application of this version to offshore wind turbines it is foreseen that this module can be extended with regard to wind and wave loading. The input items of this tower module are:

tower_height `REAL` (19.85) [m]
Height of the tower top, or the so-called 'interface point' between tower and turbine.

tower_dynamics `LOGICAL` (OFF)
If '*ON*' the tower structural dynamic deformation is solved.
If '*OFF*' only the tower geometry (**tower_diameter** and **tower_tapering**) is used for the aerodynamic stagnation.

mass_matrix `TABLE`
This key-name is to be followed by a 6*6 table with REAL variables. This table represents the mass matrix for the steady tower top deformations. The elements are ordered:
1 Unit translation in *X*-direction (G.L.);
2 Unit translation in *Y*-direction (G.L.);
3 Unit translation in *Z*-direction (G.L.) (vertical);
4 Unit rotation about *X*-axis (G.L.);
5 Unit rotation about *Y*-axis (G.L.);
6 Unit rotation about *Z*-axis (G.L.) (yaw axis).
Because of the symmetry of the mass matrix, the tower module reads only the lower half.

stiffness_matrix `TABLE`
This key-name is to be followed by a 6*6 table with REAL variables. This table should represent the stiffness matrix for the steady tower top deformations, where the elements are ordered similar as for the mass matrix.

internal_modes `TABLE`
This key-name should be followed by a table with on each record the properties of an internal mode. Following the Craig-Bampton method the internal modes are eigenmodes of the structure (tower) calculated for fixed interface points (here a fixed tower top). As result of chosing internal eigenmodes, the part of the mass and stiffness matrix related to these modes only have diagonal terms because eigenmodes are orthogonal.

Each record contains the following REAL properties of an internal elastic mode.

1 Eigen-mass [kg*m^2]. If the internal mode is mass-normalised, this value is 1.0;

2 Eigen-stiffness [N*m]. If the internal mode is mass-normalised, the eigen-stiffness is the frequency squared [(rad/s)^2];

3 Mass coupling term with constraint-mode 1 (for top-translation in X-direction);

4 Mass coupling term with constraint-mode 2 (for top-translation in Y-direction);

5 Mass coupling term with constraint-mode 3 (for top-translation in Z-direction);

6 Mass coupling term with constraint-mode 4 (for top-rotation about X-axis);

7 Mass coupling term with constraint-mode 5 (for top-rotation about Y-axis);

8 Mass coupling term with constraint-mode 6 (for top-rotation about Z-axis).

damprat_bend `REAL` (0.0) <-0.1, ..., 1.1>
Damping of the eigen-modes for bending, divided by the critical damping.

damprat_tors `REAL` (0.0) <-0.1, ..., 1.1>
Damping of the torsion mode, divided by the critical damping.

lattice_tower `LOGICAL` (OFF)
If ’OFF’ the aerodynamic stagnation is calculated for a tubular tower.
If ’ON’ the aerodynamic stagnation is calculated for a lattice tower while the inertia terms in the Morison equation for wave loading are not included.

tower_diameter `REAL` (0.0) [m]
Diameter of the upper end of the tower. Together with **tower_tapering** and **lattice_tower** this is used to calculate the aerodynamic tower stagnation.

tower_tapering `REAL` (0.0) [^{o}]
Half the taper-angle of the upper part of the tower. Together with **tower_diameter** and **lattice_tower** this is used to calculate the aerodynamic tower stagnation.

air_drag `REAL` (0.7)
Drag coefficient used to calculate the wind force on the tower.

water_drag `REAL` (1.2)
Drag coefficient for the waves on the tower, see also section 5.2.

water_mass_coeff `REAL` (1.0) <0.0, ..., 2.0 >
Coefficient of the "added mass" of the accelerating water around the tower: C_{a}.
This is also called ’hydrodynamic mass’. For more details see section 5.2.

The solution method for the tower dynamics is a linear method, which is the basis for describing the tower deformation with modes. Basically this means that the combination of axial compression (from turbine weight) and a tower top side force gives the same side-deformation as a tower top side force only, which is not the case. Knowing that the compressive ’weight’ loading of the nacelle on the tower is nearly constant, one may treat this weight as a pre-stress load on the FEM model of the tower in the analysis of the mass and stiffness matrices. This approach avoids an over-estimation of the tower frequency, which may be a few % for large offshore towers.

5.4 Output

File towmod.out

In addition to the output file written by PHATAS, the tower module writes some of its model properties to file '*towmod.out*' which contains:

- Number of elements, below and above the water surface;
- For input in terms of tubular sections: the engineering sectional properties of a beam;
- The stiffnesses at the top;
- Mass and stiffness matrices following the Craig-Bampton method, not for '*phatmod*';
- Damping properties for each mode;
- Bending frequencies with top-displacements and -rotations;
- Table with tower geometry and the mode shapes of the 2 smallest fore-aft bending modes as function of the vertical coordinate.

Figure 21 shows a (rotated!) plot of file '*towmod.out*' with the geometry and two mode shapes of a 98m high offshore tower with tripod base.

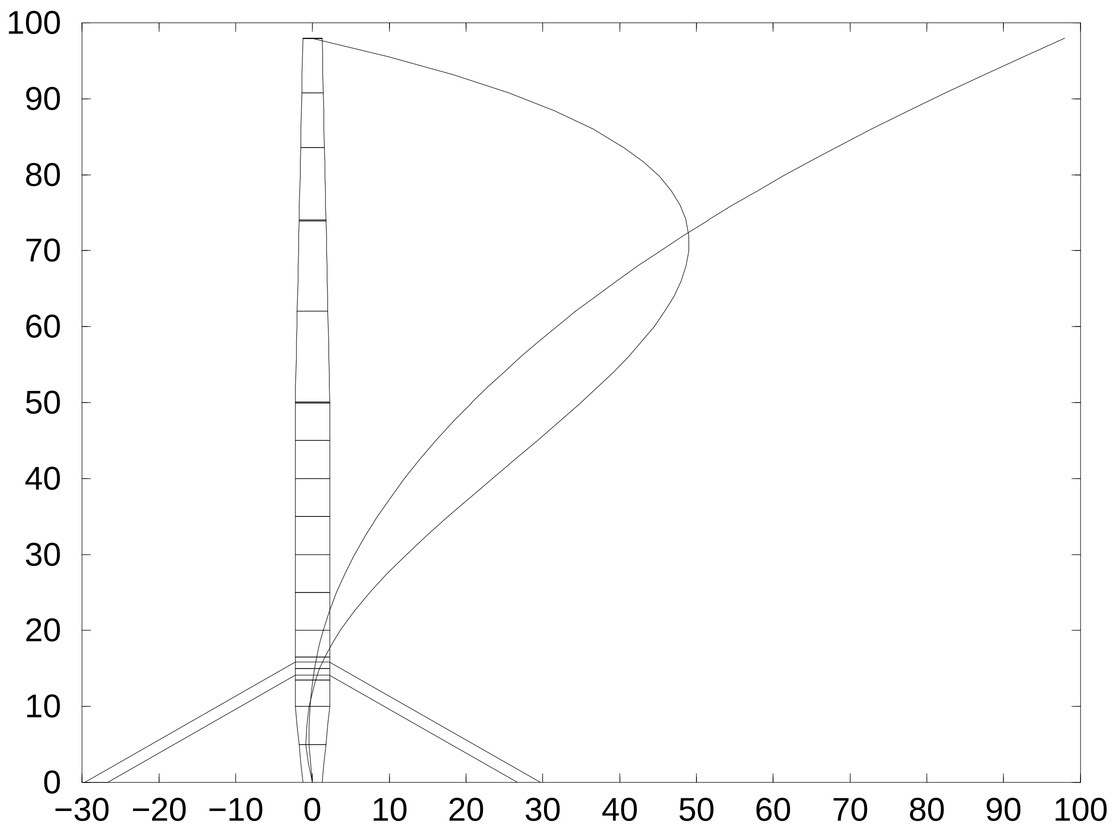


Figure 21: Geometry and bending modes for a 98m high tower with tripod base

6 CONTROLLER

Controllers for wind turbines do not have a 'standard' concept. The classical wind turbines from the 90's (with a nominal power less than e.g. 1MW) have an asynchronous generator and are passive stall controlled or active pitch-to-vane controlled, while most of the modern large wind turbines are variable-speed pitch-to-vane controlled. In addition to these configurations wind turbines have also been built with active pitch-to-stall control. It was found hard to implement a controller in PHATAS that serves all existing turbine concepts. However, since most of the modern large size wind turbines have a variable-speed pitch-to-vane controller, and in the scope of implementation in the design package FOCUS, a P-D pitch controller was developed for the PHATAS code. This controller can be used for speed control or for power control and is embedded in an algorithm with options to simulate starts, stops, and the failed conditions that are required for IEC load-set calculations.

The program-structural-diagram (PSD) of the controller algorithm is shown in Figure 22, which shows only the most significant statements of the controller. In Figure 22 the variables in normal 'Times Roman' font correspond as much as possible with the source code variable names in the PHATAS controller algorithm. Some of the variables in Figure 22 such as rotor speed and pitch angle are represented by mathematical symbols similar as used here. The strings `"NORM"` and `"STAR"` are values of the 4-character variable for the 'state' of the controller, which refer to either **norm**al operation or a **star**t. In the PHATAS source code the blade pitch angles and rotor speed values are defined in [rad] and [rad/s].

6.1 Operational control

Although the PHATAS controller can be used for both rotor-speed and power control (specified with input item **control_type**, see 3.2.3) this section is written for rotor-speed control.

Filtered rotor speed signal

For variable-speed pitch controlled wind turbines (**control_type 2**) the control algorithm uses a 'filtered rotor speed' signal, which is calculated from the 'measured rotor speed'. This 'measured rotor speed' follows from the difference in rotor azimuth between 2 subsequent controller-calls. The rotor azimuth values are calculated in PHATAS as the result of pulse-counting from a ring with '**pulses_per_rev**' pulses per revolution and eccentricity '**pulse_ring_eccentricity**'. Both the pulse-counting and the eccentricity give a disturbance of the rotor speed signal.

In addition to the scatter from pulse-counting the rotor speed signal also has high frequency variations from edgewise blade vibrations and from the aerodynamic blade-tower interaction. The latter occurs B times per rev (with B the number of blades) where each blade-tower passage takes place in a short time. The 'filtered rotor speed' used by the PHATAS controller is obtained with the first-order expression in the rotor speed Ω_{meas} from pulse counting;
$\Omega_{\mathrm{filt}} = filfac\, \Omega_{\mathrm{filt,prev}} + (1 - filfac)\, \Omega_{\mathrm{meas}}$

Here the factor $filfac$ is expressed with the controller **average_time** with:
$filfac = 1./\exp\left(\mathbf{control\_increment}/\mathbf{average\_time}\right)$.
For **control_increment** and **average_time**, see section 3.2.3.

A large value of **average_time** gives an inefficient controller that tends to give pitch-speed oscillations because of its time delay. A small value of **average_time** introduces sensitivity for all kind of rotor vibrations. A value of **average_time** that is about 1/6 of the time for one rotor revolution is practically sufficient to filter-out the collective edgewise blade vibrations and the aerodynamic blade-tower interaction.

Rotor speed control in full load

In full load the desired pitch rate is expressed with:

$$\dot{\theta}_{\text{set}} = \left(k_{\text{P}} \cdot (\Omega_{\text{filt}} - \Omega_{\text{target}})\right) + k_{\text{D}} \cdot (\Omega_{\text{filt}} - \Omega_{\text{filt,prev}})) \,/ (1 + \textbf{gain\_scheduling} \cdot (\theta_{\text{p}} - \theta_{\text{p,min}}))$$

with k_{P} = **proportional_gain** and k_{D} = **differential_gain**.
The target rotor speed Ω_{target} is the input item **controller_target**.

The source code variables for the gain factors are:
`kprop` k_{P} with unit [(rad/s)/rad];
`kdiff` k_{D} with unit [(rad/s)/rpm].

The unit for the differential gain factor looks strange because it describes the difference in (filtered) rotor speed over 0.1s which has the unit [rpm] rather than [rpm/s]. Pitch actions are only issued if the desired pitch rate $\dot{\theta}_{\text{set}}$ is larger than the **threshold_value**.

The denominator $(1 + \textbf{gain\_scheduling} \cdot (\theta_{\text{p}} - \theta_{\text{p,min}}))$ in the previous expression is simply represented in Figure 22 by '$gain\,scheduling$' . This 'gain scheduling' is a reduction of the proportional and differential gains and of the threshold value with increasing pitch angle. A '**gain_scheduling**' value of about 0.16 is reasonable. The minimum pitch angle $\theta_{\text{p,min}}$ used in the term for gain-scheduling is the pitch angle for which the rotor has its maximum power coefficient at optimum tip speed ratio.

Fuzzy actions

If the rotor speed (or power) is above the value **fuzzy_control_limit** and also increasing, the blades are pitched unconditional to vane, with a rate of **fuzzy_pitch_rate**.

For power control **fuzzy_control_limit** is in terms of power [W] instead of [rpm].

Pitch actuation

The 'control pitch rate' $\dot{\theta}_{\text{set}}$ is applied to the blades with a time delay of **pitch_time_lag** for the actuator. This delay is applied with a first-order differential expression.

Peak-shaving pitch strategy

In order to reduce the structural loads on the turbine such as blade root bending moment and aerodynamic thrust, the blades are pitched gradually to working position even before the nominal speed is reached. An evident advantage is that the quasi-steady aerodynamic thrust is limited. When these actions are smooth they do not introduce additional dynamic structural loads. In addition to avoiding pitch-induced dynamic loads, the pitch actions from the PHATAS controller around (and below) nominal conditions are performed relatively slow because with an increased steady-state pitch angle the required pitch actions for gusts that just exceed the nominal speed are smaller. This strategy just below nominal conditions with the (usually steep) positive slope of the generator torque-speed relation (stabilising) were the basis to use a simple (slow) first-order expression for the target pitch rate. The control algorithm in PHATAS is such that below nominal conditions the pitch-motion is not 'halted' by the threshold value.

Although not mentioned in Figure 22, it should be noted that in full load the minimum pitch angle is the value at the end of the 'peak shaving' interval.

6.2 Starts and Stops

Starts

In the program PHATAS a start process is simulated if the "state variable" has the value ’STAR’, otherwise this is ’NORM’.
This state variable is initiated as ’STAR’ for the combination of the following conditions:

1 The generator is disconnected (**generator_cut_off** $<$ **0**);
2 The initial rotor speed is smaller than 0.9***start_speed**;
3 The brake is not applied (**brake_time** is large);
4 The initial pitch angle is larger than (0.6***min_pitch_angle** + 0.4***max_pitch_angle**).

These conditions are applied to distinguish a start procedure from an ’idling’ or ’stand-by’ state. The statements in the input files generated with the program *lcprep* (see Appendix B) account for these conditions.

In the simulation of a start process, the blades are pitched with ‘**start_pitch_rate**‘ to working position after ‘**wait_time**‘ seconds. If the rotor speed (not the filtered value) exceeds ‘**start_speed**‘ then the generator is connected and the controller state is set to ’NORM’.

Stops

Stops can be simulated relatively easy using a prescribed time of brake activation **brake_time**, using **brake_overspeed** (eventually together with generator disconnection **generator_cut_off**) or from a more complicated sequence of events and/or failure modes (see ’6.3 Models for Failed Conditions’).

In the PHATAS controller a ’stop procedure’ involves both applying the brake **brake_torque** and pitching the blades with **brake_pitch_rate**. According to the governing turbine and the load case one may set the value of **brake_pitch_rate** or **brake_torque** to zero.

6.3 Models for Failed Conditions

The control algorithm has some options for failed conditions. It was attempted to implement these options at the location where they also occur in the real wind turbine. Simulation of failed conditions is also possible if no operational control is modelled (**control_type 0**).

A list of failure modes (**failure_type**) in the PHATAS controller is given in section 3.2.3. The implementation of the governing statements of the failure modes is indicated with the source-code variable ’*ifail*’ in the Program Structural Diagram of Figure 22. All failed conditions are simulated after ‘**wait_time**‘ seconds.

For failure modes of e.g. the yaw controller, the types of failure can be extended with higher numbers for **failure_type**.

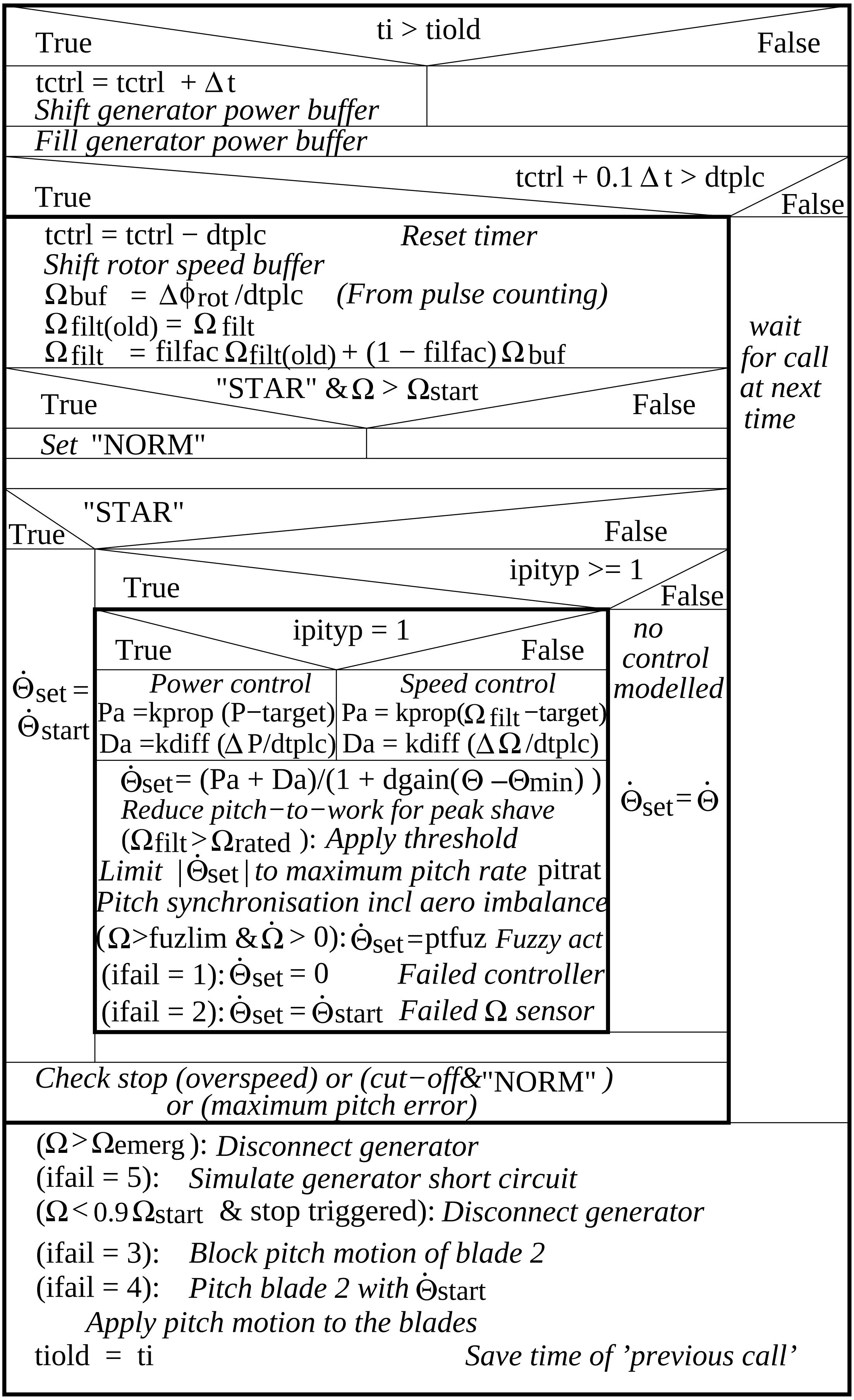


Figure 22: Program Structural Diagram of the PHATAS controller

7 ADDING SPECIFIC ALGORITHMS

Most of the dynamic behaviour of a horizontal axis wind turbine can be described with the modelling that is implemented in PHATAS. For the degrees of freedom of which the dynamics are- or can be influenced by some kind of controller, the modelling depends on the particular turbine configuration. These degrees of freedom include among others:

- Rotor rotation/speed, with generator characteristics;
- Yaw motion: controlled or (partially-) free;
- Blade pitch angle.

For the solution of each of these degrees of freedom (and also for others such as a flapping hinge or teeter hinge) PHATAS has an individual routine, see Figure 24. If a FORTRAN compiler is available it is possible to modify and link each of the routines as long as one does only assign those variables that are 'allowed' to be modified, see section 7.6.

For the solution of the rotor-speed with generator torque-speed relation basically 3 types of models are available, depending on the input item **generator_model**, see section 3.4.2:

0 Fixed generator speed (not much of a model);
1 - 4 Prescribed torque-speed relation;
5 Generator torque setting (user defined) from controller routine *'conrol'*.

The solution for the blade pitch angle can be either **1:** passive pitch (coupled or individual-, for **pitch_flag ON**) or **2:** controlled pitch (from routine *conrol*).

The options for describing and/or solving yaw motion are summarised in section 7.1.

With the fact that several models are available for the generator torque-speed relation and models are available for rotor speed or power control by pitch, this chapter was kept to a limited size by describing only the structure and the available options for yaw control. This choice was also based on the fact that the options for controlled yaw require a user-written algorithm.

7.1 Yaw control

In the structure of PHATAS release "APR-2005" and "NOV-2003" the yaw motion is expressed- or solved in routine '*upyaw*' which is called twice in each iterative solution, see Figure 24. The first call for '*upyaw*' is issued if the input item **free_yawing** is 'OFF', while the second call is issued if the input item **free_yawing** is 'ON', see Figure 25. These two calls and the call for the (user reserved) controller routine *'conrol'* provide the following options for yaw control:

1 free_yawing OFF and no statements in routine '*conrol*':
With this PHATAS option a prescribed yaw motion such as plotted in Figure 9, using 'standard' input items described in section 3.4.1.;

2 free_yawing ON and no statements in routine '*conrol*':
With this 'standard' PHATAS option the free yawing motion of the nacelle is solved, with the influence of friction, viscous damping, and yaw drive inertia, see section 3.4.1.

3 free_yawing OFF and an algorithm for yaw motion in routine '*conrol*':
This can be used for straightforward yaw control on basis of e.g. time-averages of the misalignment. In the control routine *'conrol'* a buffer (array) is available for the yaw misalignment, where the initial yaw angle is used as systematic (yaw tracking) error.

4 free_yawing ON and an algorithm for the yawdrive torque in routine '*conrol*':
This can be used to solve the dynamic yaw behaviour, including yaw drive dynamics.

Calculations without any yaw motion can be performed (independent of the value for **free_yawing**) by specification of a very large value of **yaw_friction** together with a zero value of **yaw_rate**.

These statements are written by '*lcprep*' for the simulation of e.g. parked or maintenance load cases, see Appendix B.

7.2 Solution Process

Guidelines for editing and linking user-defined routines of PHATAS are presented here, with an example for controlled yaw. For most applications, the user-defined parts of the code are written in routine '*conrol*', although one can modify any other routine such as '*tipsol*', '*torgen*', or '*upyaw*'. If the user-defined algorithms solve the dynamics of a degree of freedom, equilibrium of the solutions at each time is only realised if this algorithm fits the implicit solution method for the integrated structural dynamic response.

In PHATAS the non-linear set of dynamic equations is expressed in terms of the variables (and their time derivatives) to be solved at the current time, known as "implicit formulation". At every time step equilibrium is obtained (or approached) iteratively in which the equations for each of the degrees of freedom are solved subsequently treating the most recent solutions for the other degrees of freedom as constants. The reference structure of this iteration process is shown in Figure 23 and Figure 24.
As can be seen from Figure 24 the routines '*conrol*', '*tipsol*', '*torgen*', and '*upyaw*' are called in the iteration loop for which reason it is recommended to add only statements that assign (the solution of) a degree of freedom at the current time. These statements can be a function of the current values of other variables or of the values of the variable at the previous times.

Not 'shifting' variables in time!

Before starting the iterative solution at each time the current values are assigned to the variables for the degrees of freedom at the previous time which describe a converged solution. The variables are shifted in routine '*tguess*' in which also a guess for the values at the current time is calculated using a second- or third order extrapolation through solutions at previous time(s). For the yaw motion, a second- order extrapolation is used for '**free_yawing ON**' and a linear extrapolation is used for '**free_yawing OFF**'. When adding statements in routine '*conrol*', '*tipsol*', '*torgen*', or '*upyaw*' one must be aware not to increment or 'shift' the values of the degrees of freedom and of not assigning variables for the state at the previous times.

If for any reason an iterative (implicit) solution scheme is not used it is recommended to `SAVE` the variable for the time '`ti`' and assign only the additional statements when this time is increased, indicating a new time step. In this case:

- The solution is no longer an implicit solution which means that the cut-off frequency of the dynamic response that is described, is smaller unless the time increment of the calculation is reduced strongly;

- It is recommended to assign the statements at the 3-rd iteration, because a minimum of at least three iterations is performed while the structural dynamic response is solved after calling routine '*conrol*'. This approach is a (poor) kind of "predictor-corrector" method.

7.3 Controlled yaw from routine *conrol*

Routine '*conrol*' was initially reserved for algorithms such as for pitch motion, start and stop actions, failed conditions etcetera. For the latest PHATAS releases also the statements for yaw control are written in '*conrol*'.

Because pitch- and yaw actions have a direct influence on the aerodynamic rotor loads, routine '*conrol*' is called first in each iteration, see Figure 24.

If the expressions for transients (e.g. generator disconnection or brake activation) or for pitch control are very sensitive to the current value of the slow shaft speed `omega` the result may be an alternating sequence of solutions that do not converge. For these cases it is recommended to evaluate the control algorithm with the rotational speed at the previous time `omegam` which is at each time a converged equilibrium variable.

The statements for initialisation of parameters (e.g. when read from file) can be written in routine '*inicon*' which is called once from routine '*phmain*' before the calculation of the structural dynamic response. The standard routine '*inicon*' contains read statements for passive pitch (from file 'tipdata.txt'). It contains also read statements from file 'yawcont.txt' for the limits (start- and stop-) for yaw control and the average time for the yaw misalignment. An example of the file 'yawcont.txt' is listed in Figure 26.

The 'standard' routine *conrol* contains the algorithm described in chapter 6. An 'improved' or 'extended' version of '*conrol*' has been developed in with which also the yaw control is modelled on basis of the average measured misalignment. The additional parts of the source code of this extended routine are listed in Figure 27. These statements describe a yaw activation if the average misalignment over some period exceeds a given ('trigger') value, while the yaw motion is de-activated if the average misalignment gets below another 'trigger' value. These variables are read in routine '*inicon*' from file 'yawcont.txt' and communicated by `COMMON /yawdat/`. The statements in Figure 27 also model a 'yaw tracking error' for which the initial yaw angle is used.

The yaw motion (either prescribed or free yaw) is assigned in routine *upyaw*, of which the Program Structural Diagram is shown in Figure 25.

7.4 Unit Numbers

Initialisation of properties used in the algorithms in '*conrol*', '*tipsol*', '*torgen*', and '*upyaw*' can be done in routine '*inicon*' which is called once from routine '*phmain*', see Figure 23. When reading data from files in routine '*inicon*' the following unit numbers are already opened:

`iout`	**2**	Output file;
`iodat`	**4**	Binary data file;
`iomesg`	**8**	Message file;
	22	(if used) ASCII- or SWIFT wind data file;
	23	(if used) wave data file.

These numbers and the numbers **0** (standard error), **5** (standard input), and **6** (standard output) must not be used as unit numbers for reading files in routines '*conrol*', '*tipsol*', or '*torgen*'.

For reading of the input files a "SCRATCH"-file is used with unit number **1**.

The PHATAS input file is read using unit number **10** while the unit numbers used for the additional input files are **11**, **12**, ... etc.

The airfoil coefficients files, and the tower input file are read using unit number `iodat` = **4**.

The output files of the tower model are written using unit numbers **32**, **34**, and **35**.

When calling routines '*conrol*', '*tipsol*', '*torgen*', and '*upyaw*' the files 'STATUS', 'RESTART', and the file '**model_file**' are already closed but were opened with unit numbers **8**, **3**, and **2** respectively.

When solving the passive pitch motion, '**pitch_flag ON**' see section 3.4, some additional input data are read from file '*tipdata.txt*' using unit number **17**.

When modelling controlled yaw such as described in section 7.3, the input values are read from 'yawcont.txt' (if present) using unit number **17**.

7.5 Time Derivatives of Variables

In general the equations for the non-linear dynamic behaviour do not only depend on the degrees of freedom themselves but also on their first and second time derivatives. For the dynamic response in PHATAS the values of the variables and their time derivatives are related to each other using backward differences. As a result, each degree of freedom of the structural dynamic model is in fact also a single degree of freedom in the set of equations. To have a single equation with only one variable for each degree of freedom, the time derivatives of this variable are expressed in the variable itself at the current and the previous time and in the derivatives of this variable at the previous time (or the inverse expression) using backward differences.

For the time derivatives of the yaw angle for example these backward differences describe a second order yaw motion and are of the form:

$$\dot{\phi}_{\mathrm{yaw}}(t) = 2\,(\phi_{\mathrm{yaw}}(t) - \phi_{\mathrm{yaw}}(t-\Delta t))/\Delta t - \dot{\phi}_{\mathrm{yaw}}(t-\Delta t)\ ;$$
$$\ddot{\phi}_{\mathrm{yaw}}(t) = (\dot{\phi}_{\mathrm{yaw}}(t) - \dot{\phi}_{\mathrm{yaw}}(t-\Delta t))/\Delta t\ .$$

When writing a control algorithm with statements for the yaw angle it is recommended to assign the first and second time derivative with the FORTRAN statements:

```
dfiyt  = 2*(fiyaw - fiym)/delt - dfydtm
d2fiy  = (dfiydt - dfydtm)/delt
```

When writing a control algorithm with statements for the yaw rate it is recommended to assign the yaw angle and its second time derivative in routine '*conrol*' with the FORTRAN statements:

```
fiyaw  = fiyaw + (dfiydt + dfydtm)*delt/2.
d2fiy  = (dfiydt - dfydtm)/delt
```

The time derivatives for the pitch angle are calculated with the same backward time difference schemes as for the yaw angle.

Warning! When adding a control algorithm with statements for pitch and/or yaw motion that have a discontinuous 'rate', the corresponding second time derivatives of these variables will be very high. A high second time derivative for pitch and/or yaw leads to peak values of pitch or yaw moment at these time steps. In the example of yaw control listed in Figure 27 these 'peak values' for yaw acceleration are avoided by using the **yaw_time_lag** (source code variable '*tauyaw*') as a first-order time delay function.

For such kind of control algorithms these peak-moments can be avoided by always setting the second time derivative to zero instead of using the expressions mentioned above, which however also neglects some of the structural dynamics.

7.6 Source Code Variables

Following is a description of source code variables in the most relevant include files. For the first character of most of the FORTRAN variable names the following convention is used:

- `INTEGER` variable names start with `i` , except for some 'DO-loop' indices `j` and `k` and some dimensions that start with `n` .
  Parameters for array dimensions start with `n` or with `i` ;
- `LOGICAL` variable names start with `l` ;
- `CHARACTER` variable names start with `c` ;
- `REAL` variable names start with a letter in the range < `a-h, o-z` >.

This convention is not pre-defined but is used to make the source code readable.
Note for instance that variables with names starting with ' `c` ' can be `REAL` or `CHARACTER`.
The length of the variable- and subroutine names is maximised to 6 characters.

In the source of PHATAS every angle is expressed in [rad] and every angular velocity in [rad/s] except for the aerodynamic angles of attack (from the tables with airfoil coefficients) which are expressed in [^{o}].

The descriptions of source code variables in the following tables contain;

- the FORTRAN **variable name** in **bold face** with an *italic S!* if assigning is allowed and effective, or with an *italic N!* if these values may not be assigned;
- the `type` in `typewriter font` ;
- for parameters, the (values) between ordinary ( ) brackets;
- if applicable, the [unit] between square [ ] brackets;
- some explanation of their definition as normal text.

Table 1: Contents of include file *COMPALL*

The parameters **nnbl** and **nnel** stand for the dimension of the arrays used in the code. Therefore file '*COMPALL*' must be included in the (user written) routines before the INCLUDE statements for '*CONTRLC*' or any other array with e.g. the number of elements or number of blades as index.

None of the variables and parameters of '*COMPALL*' may be changed except for setting '**lstop**' to `.TRUE.` in case of a serious error

Parameters:		
nnbl	`INTEGER` (3)	Maximum number of rotor blades.
nnel	`INTEGER` (24)	Maximum number of elements per blade.
iodat	`INTEGER` (4)	Unit number of the binary datafile.
iomesg	`INTEGER` (8)	Unit number for messages.
iout	`INTEGER` (2)	Unit number of the ASCII output file.
cgr	`REAL` (.01745329252)	Conversion factor = π/180 [rad/^{o}].
grav	`REAL` (9.81)	Gravitational constant [m/s^2].
pi	`REAL` (3.1415926536)	Trigonometric constant π.
Variables:		
istep	`INTEGER`	Number of the time step.
inwmsg	`INTEGER`	Number of non-convergence messages written. *Not for release "NOV-2003".*
lstop	`LOGICAL`	If '.TRUE.' an error has occurred.
dels	`REAL`	Spanwise size [m] of one blade element.
epsing	`REAL`	5 times round off error of a REAL number.
ti	`REAL`	The current time [s].

Table 2: Variables of include file *CONTRLC*

CONTRLC contains properties that may be used in control algorithms.

lincon *S!*	`LOGICAL`		**lincon** is initialised `.TRUE.` before. *conrol* is called for the first time.
tipmas *N!*	`REAL(1:nnbl)`	[kg]	Mass of the pitching part of the blade.
smtip *N!*	`REAL(1:nnbl)`	[kg·m]	Static mass moment of the pitching part of the blade.
ajtip *N!*	`REAL(1:nnbl)`	[kg·m^2]	Moment of inertia about the pitch axis of the pitching part of the blade.
bldmas *N!*	`REAL(1:nnbl)`	[kg]	Mass of the blade measured from the blade root.
sblad *N!*	`REAL(1:nnbl)`	[kg·m]	Static mass moment of inertia of the blade with respect to the blade root.
ajhing *N!*	`REAL(1:nnbl)`	[kg·m^2]	Second moment of inertia of the blade.
rotmas *N!*	`REAL`	[kg]	Mass of the rotor including the hub.
ajrot *N!*	`REAL`	[kg·m^2]	Rotor rotational inertia, including hub.
ajhuby *N!*	`REAL`	[kg·m^2]	Hub inertia about the rotor Y-axis.
ajhubz *N!*	`REAL`	[kg·m^2]	Hub inertia about the rotor Z-axis (teeter axis).
ajhbyz *N!*	`REAL`	[kg·m^2]	Coupling inertia of the hub in case 'δ_3' teeter hinge.
ajtet *N!*	`REAL`	[kg·m^2]	Inertia of the rotor about teeter hinge.
coan *N!*	`REAL`	[.]	Cosine of the tilt angle of the rotor shaft.
sian *N!*	`REAL`	[.]	Sine of the tilt angle of the rotor shaft.
coalc *N!*	`REAL`	[.]	Cosine of the blade cone angle.
sialc *N!*	`REAL`	[.]	Sine of the blade cone angle.
hhub *N!*	`REAL`	[m]	Height of the rotor centre above ambient water level.
ht *N!*	`REAL`	[m]	Height of the tower top above ambient water level.
pelec *N!*	`REAL`	[W]	Power transmitted through the generator shaft.
qelec *N!*	`REAL`	[Nm]	Generator shaft torque expressed w.r.t. slow shaft.
qelecm *N!*	`REAL`	[Nm]	Idem at the previous time.
qplus *N!*	`REAL`	[Nm]	Additional generator torque from controller.
qyawdr *S!*	`REAL`	[Nm]	Torque of a yaw drive, valid for '**free_yawing ON**'.
firot *N!*	`REAL(1:4)`	[rad]	Rotor azimuth. The index refers to time step '4' for the current time, and '$3, 2, 1$' for 1, 2, and 3 time steps ago.
dfirdt *N!*	`REAL`	[rad/s]	The rotor angular speed.
dfrdtm *N!*	`REAL`	[rad/s]	Idem at the previous time step.
d2firo *N!*	`REAL`	[rad/s^2]	The rotor angular acceleration.
d2firm *N!*	`REAL`	[rad/s^2]	Idem at the previous time step.
fiteet *N!*	`REAL`	[rad]	Teeter angle of the rotor.
fitm *N!*	`REAL`	[rad]	Teeter angle at previous time step.
dfitdt *N!*	`REAL`	[rad/s]	Teeter rate.
dftdtm *N!*	`REAL`	[rad/s]	Idem at the previous time step.
d2fite *N!*	`REAL`	[rad/s^2]	Teeter angular acceleration.
d2fitm *N!*	`REAL`	[rad/s^2]	Idem at previous time step.
fiyaw *S!*	`REAL`	[rad]	Yaw angle of the nacelle.
fiym *N!*	`REAL`	[rad]	Yaw angle at previous time step.
dfiydt *S!*	`REAL`	[rad/s]	Yaw rate.
dfydtm *N!*	`REAL`	[rad/s]	Idem at the previous time step.
d2fiy *S!*	`REAL`	[rad/s^2]	Yaw angular acceleration.
pitcha *S!*	`REAL(1:nnbl,j)`	[rad]	Pitch angles of the blades. The first index denotes the blade number and the second index j refers to the time step similar as for the rotor azimuth **firot**.
dpitdt *S!*	`REAL(1:nnbl)`	[rad/s]	Time derivative of the pitch angles.
dpitm *N!*	`REAL(1:nnbl)`	[rad/s]	Idem, at the previous time step.
d2pit *S!*	`REAL(1:nnbl)`	[rad/s^2]	Second time derivative of *pitcha*.
uwrite *S!*	`REAL(1:9)`	[?]	User reserved output properties.

Table 3: Some variables of include file *PNCLM2*

PNCLM2 contains properties of MENU **job_param**.
The values of none of these properties may be changed.

chmark	`CHARACTER*20`		Character written on first pos. of comment lines.
iecho	`INTEGER`		Value of '**output_level**' for the amount of messages.
dt	`REAL`	[s]	Time increment.
ltime	`LOGICAL`		If `.TRUE.` PHATAS calculates the dynamic response.

Table 4: All variables of include file *PNCLM3*

PNCLM3 contains properties of MENU **geometry**.
The values of none of these properties may be changed.

inob	`INTEGER`		Number of rotor blades.
inel	`INTEGER`		Number of elements of one blade counting from '1' for the root element to '*inel*' at the tip. The nodes are numbered from '0' at the root to '*inel*' at the tip.
alna	`REAL`	[rad]	Tilt angle of the rotor shaft.
dn	`REAL`	[m]	Distance between the tower axis and the rotor centre (or teeter axis), measured along the rotor shaft.
ezhub	`REAL`	[m]	Location of the hub above the tower top.
erotor	`REAL`	[m]	Lateral offset of rotor-axis left of the tower (Y-dir.).
alfcon	`REAL`	[rad]	Cone angle of the rotor blades.
ezroot	`REAL`	[m]	Spanwise location of the blade root.
rlb	`REAL`	[m]	Blade span measured from rotor centre to tip.
yspit	`REAL`	[m]	Spanwise location of pitch bearing, from blade root.
xcpit	`REAL`	[m]	Chordwise location of the pitch axis w.r.t. blade axis.
dsdtet	`REAL`	[m/rad]	Radial tip motion for pitch actions.
rmhub	`REAL`	[kg]	Mass of the rotor hub.
sxhub	`REAL`	[kg·m]	Static mass moment (downwind) of the hub.
ajhubx	`REAL`	[kg·m^2]	Rotational moment of inertia of the hub.
rmnac	`REAL`	[kg]	Mass of the nacelle, without hub and rotor.
sxnac	`REAL`	[kg·m]	Downwind static moment of nacelle.
sznac	`REAL`	[kg·m]	Vertical static moment of nacelle.
ajxxnc	`REAL`	[kg·m^2]	Nacelle inertia about the roll axis, w.r.t. tower top.
ajyync	`REAL`	[kg·m^2]	Nacelle inertia about the tilt axis, w.r.t. tower top.
ajzznc	`REAL`	[kg·m^2]	Nacelle inertia about the yaw axis.
ajzxnc	`REAL`	[kg·m^2]	Nacelle coupling inertia betweem roll and yaw.
cxnac	`REAL`	[m^2]	Drag-area of the nacelle for front or rear wind.
cynac	`REAL`	[m^2]	Drag-area of the nacelle for sideways wind.

Table 5: All variables of include file *PNCLM4*

PNCLM4 contains flags for the degrees of freedom (MENU **configuration**).
The values of none of these properties may be changed.

ctowin	`CHAR*20`	Name of the input file for the tower module.
lyaw	`LOGICAL`	If `.TRUE.` free yawing is modelled.
lgear	`LOGICAL`	If `.TRUE.` torsion of the gearbox support is solved.
lshaft	`LOGICAL`	If `.TRUE.` the rotor shaft has torsional flexibility.
lteet	`LOGICAL`	If `.TRUE.` a teetered rotor-hub connection is modelled.
lhing	`LOGICAL`	If `.TRUE.` a discrete flapping hinge is modelled at the blade root.
ltip	`LOGICAL`	If `.TRUE.` routine '*tipsol*' is called.
llag	`LOGICAL`	If `.TRUE.` blade lead-lag deformation is solved.
lflap	`LOGICAL`	If `.TRUE.` blade flapping deformation is solved.
ltors	`LOGICAL`	If `.TRUE.` blade torsional deformation is solved.

Table 6: All variables of include file *PNCLS3*

PNCLS3 contains properties of SUBMENU **control**.
The values of these properties may be changed.

ipityp	`INTEGER`		Indicates the type of operational controller.
ifail	`INTEGER`		Indicates the type of failure.
tbrake	`REAL`	[s]	Time of activating a stop: brake and/or pitch.
tcutof	`REAL`	[s]	Time of disconnecting the generator.
ombrak	`REAL`	[rad/s]	Slow shaft speed for triggering stop action.
qbrake	`REAL`	[N·m]	Brake torque expressed at slow shaft.
tdelay	`REAL`	[s]	Time interval between triggering and activation of the brake/pitch/yaw stop action.
tbramp	`REAL`	[s]	Time needed to obtain full brake torque.
ptbrak	`REAL`	[rad/s]	Pitch rate for stop action.
dtplc	`REAL`	[s]	Time increment for the control algorithm.
inpuls	`INTEGER`		Number of pulses on speed sensor ring.
ecsens	`REAL`	[.]	Eccentricity of speed sensor ring.
target	`REAL`	[W] / [rad/s]	Target of power or rotor speed control.
fuzlim	`REAL`	[W] / [rad/s]	Power or speed for unconditional pitch action.
ptfuz	`REAL`	[rad/s]	Pitch rate of unconditional ('fuzzy') actions.
tauav	`REAL`	[s]	Time for averaging controller input.
kprop	`REAL`	[(rad/s)/('**target**')]	Proportional gain factor in controller.
kdiff	`REAL`	[(rad/s)/('**target**'/s)]	Differential gain factor in controller.
dgain	`REAL`	[1/rad]	Decrease of target rate with pitch.
thresh	`REAL`	[rad/s]	Threshold value for output of P-D algorithm.
taupit	`REAL`	[s]	Time delay for pitch actuator.
pitrat	`REAL`	[rad/s]	Pitch rate for normal control.
piterr	`REAL`	[rad]	Difference in pitch angle between the blades.
emergn	`REAL`	[W] / [rad/s]	Emergency speed or power for generator disconnect.
pitnom	`REAL`	[rad]	Pitch angle at end of peak-shaving interval.
psbeg	`REAL`	[W] / [rad/s]	Begin of peak-shaving interval.
psend	`REAL`	[W] / [rad/s]	End of peak-shaving interval.
dtwait	`REAL`	[s]	'Wait' time before starts or faults take place.
ptstar	`REAL`	[rad]	Pitch rate for a start procedure.
omstar	`REAL`	[rad/s]	Rotor speed at which the generator is connected.
ptermx	`REAL`	[rad]	Maximum pitch error, triggers stop if exceeded.

Table 7: All variables of include file *PNCLS4*

PNCLS4 contains properties of SUBMENU **yaw_data**.

These properties can be used for a yaw-algorithm and may all be changed.

tystar	`REAL`	[s]	Time at which controlled yaw action start.
yawrat	`REAL`	[rad/s]	Rate for yaw actions.
tyramp	`REAL`	[s]	Ramp or rise time for yaw actions.
typeri	`REAL`	[s]	Recurrence time of full prescribed yaw cycle.
tauyaw	`REAL`	[s]	Time constant for yaw rate.
fyfric	`REAL`	[N·m]	Coulomb friction in the yaw mechanism.
cvyaw	`REAL`	[N·m/(rad/s)]	Viscous damping in the yaw mechanism.
yawdri	`REAL`	[kg·m^2]	Yaw drive inertia, with respect to yaw motion.

Table 8: Some variables of include file *PNCLS5*

PNCLS5 contains properties of SUBMENU **generator_data**

The values of none of these properties may be changed.

igen	`INTEGER`		Indicates the type of generator model.
wrated	`REAL`	[rad/s]	’**rated_rotorspeed**’ as specified (in [rpm]).
prated	`REAL`	[W]	’**rated_power**’ as specified.
gslip	`REAL`	[.]	’**generator_slope**’, the "slip" in case of an asynchronous generator.
torkip	`REAL`	[Nm]	Maximum ’kip’ torque of a (asynchronous) generator.
gr	`REAL`	[.]	Transmission ratio of the gearbox.
glos0	`REAL`	[N·m]	Constant loss of torque in the gearbox.
glos1	`REAL`	[.]	Gearbox loss linear with generator torque.
ajslow	`REAL`	[kg·m^2]	Moment of inertia of low speed shaft.
ajfast	`REAL`	[kg·m^2]	Moment of inertia of high speed shaft.
ajgbs	`REAL`	[kg·m^2]	Moment of inertia of the gearbox.
qksh	`REAL`	[Nm/rad]	Torsional stiffness of rotor shaft.

Table 9: All variables of INCLUDE file *PNCLS8*

PNCLS8 contains properties of SUBMENU **pitch_data**.

These variables can be used to model e.g. pitch failure.

lcoupl	`LOGICAL`		If `.TRUE.` passive pitch motion of all blades is coupled.
qkpit	`REAL`	[N·m/rad]	Stiffness of a torsional pitch spring.
qpre	`REAL`	[N·m]	Pre-stressing moment of the spring.
qhpit	`REAL`	[N·m]	Coulomb friction for pitch.
cvpit	`REAL`	[N·m/(rad/s)]	Viscous damping of the pitch motion.
pitmin	`REAL`	[rad]	Minimum value of the pitch angles.
pitmax	`REAL`	[rad]	Maximum value of the pitch angles.

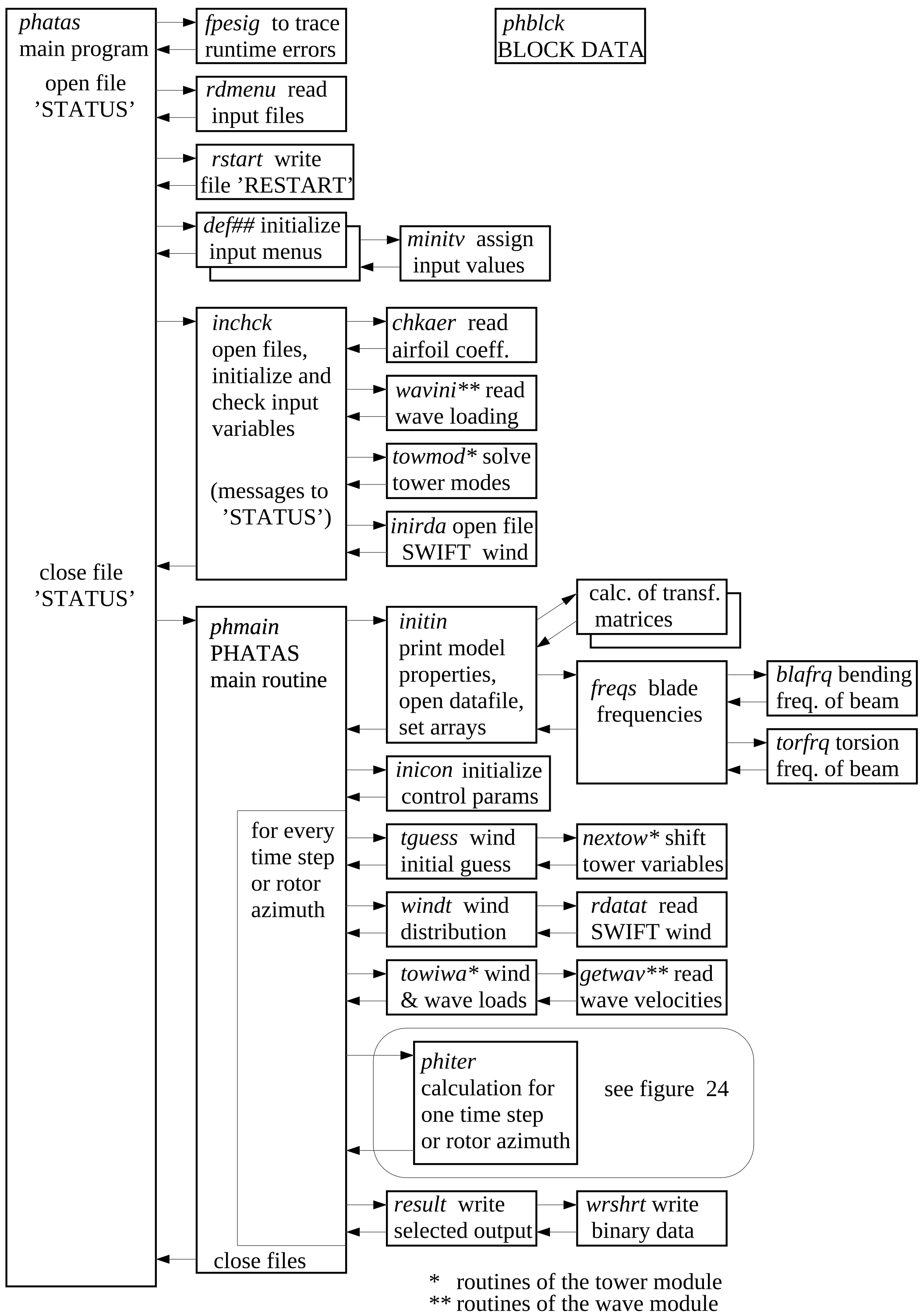


Figure 23: Global reference structure of the program PHATAS

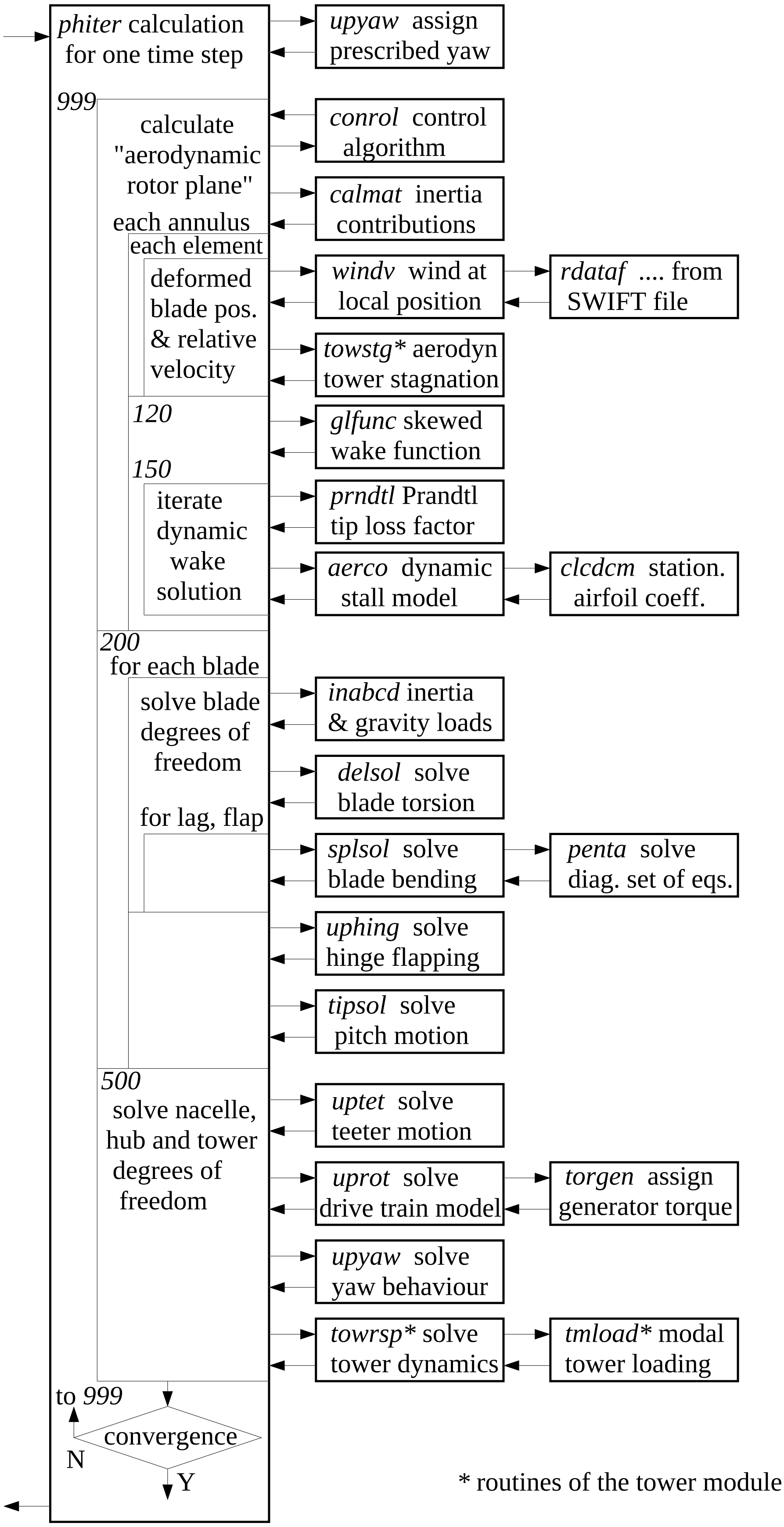


Figure 24: Reference structure of the PHATAS iteration process

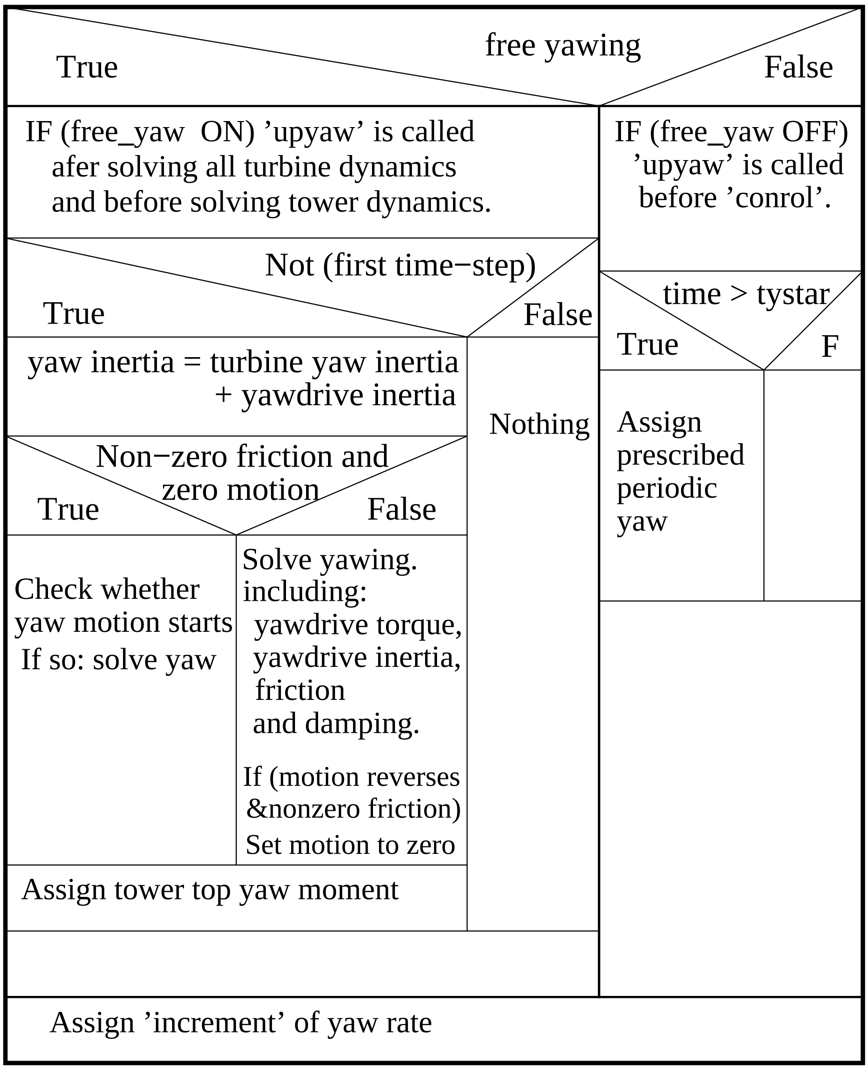


Figure 25: Program Structural Diagram of the standard routine '*upyaw*'

```
/
/  Input specifications for yaw control.
yaw_start_limit  10.0  < [deg] Misalignment to start yawing.
yaw_stop_limit    6.0  < [deg] Misalignment to stop yawing.
yaw_average_time 15.0  < [s] Average time for yaw start trigger.
yaw_short_average 5.0  < [s] Average time for yaw stop trigger.
/
```

Figure 26: Example of input file for user-written yaw control

```
*DECK conrol
      SUBROUTINE conrol(omega,omegam,qgen,dqdomg,delt)
CM
CM ******************************************************************
CM
CM     s u b r o u t i n e   c o n r o l
CM
CM ‘conrol’ contains an algorithm with a P−D pitch controller.
CM This algorithm is invoked every ‘dtplc’ seconds (from PNCLS3)
CM by which the process within a PLC controller is simulated.
CM
CM Common properties:
CM PNCLM4:
CM    lyaw    LOGICAL If .TRUE. free yawing is modelled.
CM PNCLS4:
CM    yawrat  REAL    Yaw rate [rad/s] for yaw control.
CM    tauyaw  REAL    Time constant [s] of yaw actuator.
CM
CM Latest modification was done by  C. Lindenburg;  May  22,  2005.
CM Add statements for yaw control.
CM In the implementation used here, the buffer ’fiwbuf’ contains wind
CM direction only, as if the yaw angle is subtracted from misalignment.
CM
CM ******************************************************************
CM
      INTEGER nnbuf
      PARAMETER (nnbuf=512)
C
      INCLUDE ’COMPALL’
      INCLUDE ’CONTRLC’
      INCLUDE ’PNCLM4’
      INCLUDE ’PNCLS4’
C
      REAL    velwin, fiwhub, cwndir, swndir, cofwup, sifwup
      COMMON /wintim/ velwin, fiwhub, cwndir, swndir, cofwup, sifwup
C
      REAL    ombuf(0:nnbuf),  powbuf(0:nnbuf), pitbuf(0:nnbuf)
      REAL    winbuf(0:nnbuf), fiwbuf(0:nnbuf), yawbuf(0:nnbuf), tctrl
      COMMON /conbuf/ ombuf,powbuf, pitbuf, winbuf,fiwbuf,yawbuf,tctrl
C
C  Properties read from file ’yawcont.txt’ in routine ’inicon’.
      REAL    fystar, fystop, tyave
      COMMON /yawdat/ fystar, fystop, tyave
C
C  ******************************************************************
C  Argument declarations:
C
      REAL    omega, omegam, qgen, dqdomg, delt
C
C  ******************************************************************
C  Local declarations:
C
      REAL    yawerr, fiyave
      INTEGER inbuf, iactiv
C
      SAVE    yawerr
      SAVE    iactiv
C
C  ******************************************************************
C  Start body:
C
      IF (lincon) THEN
C
C  ******************************************************************
C     I N I T I A L I Z E   C O N T R O L   V A R I A B L E S.
C
C  Use the initial yaw angle from PHATAS input as yaw error.
         yawerr = fiyaw
         fiyave = fiwhub − yawerr
         iactiv = 0
         DO 20 ibuf = 0,nnbuf
            ombuf(ibuf)  = omega
            powbuf(ibuf) = qelec*omega
            fiwbuf(ibuf) = fiwhub − yawerr
  20     CONTINUE
C
         lincon = .FALSE.
C  END IF (lincon)
      END IF
C
```

Figure 27 Statements for yaw control in routine ’*conrol*’ (part 1 of 2)

```
      fiwbuf(0) = fiwhub + fztow - yawerr
C
C  The contribution of '0.1*delt' is to cover round-off errors.
      IF (tctrl + 0.1*delt .GT. dtplc) THEN
C
C  *****************************************************************
C  E N T E R   T H E   C O N T R O L L E R   A L G O R I T H M
C
C  'tctrl' is the time [s] since the previous controller evaluation.
C  Re-set timer 'tctrl' for the following controller evaluation.
         tctrl = tctrl - dtplc
C
C  Shift the previous rotor speed and wind-direction values.
         DO 50 ibuf = nnbuf,1,-1
            ombuf(ibuf)  = ombuf(ibuf-1)
            fiwbuf(ibuf) = fiwbuf(ibuf-1)
  50     CONTINUE
C
         IF (fiwbuf(1) .GT. fiwbuf(2) + 3.1416) THEN
            DO 54 ibuf = 2,nnbuf
               fiwbuf(ibuf) = fiwbuf(ibuf) + 6.283185307
  54        CONTINUE
         END IF
         IF (fiwbuf(1) .LT. fiwbuf(2) - 3.1416) THEN
            DO 56 ibuf = 2,nnbuf
               fiwbuf(ibuf) = fiwbuf(ibuf) - 6.283185307
  56        CONTINUE
         END IF
C
C  Calculate the misalignment average over 'tyave' s.
         inbuf = int(tyave/dtplc)
         IF (inbuf .GT. nnbuf) inbuf = nnbuf
         fiyave = fiwbuf(1)
         DO 60 ibuf = 2, inbuf
            fiyave = fiyave + fiwbuf(ibuf)
  60     CONTINUE
         fiyave = fiyaw + fiyave/real(inbuf)
         IF (fiyave .GT.  3.1416) fiyave = fiyave - 6.283185307
         IF (fiyave .LT. -3.1416) fiyave = fiyave + 6.283185307
C
C  *****************************************************************
C  Trigger or stop yaw action if necessary.
         IF (.NOT. lyaw) THEN
           IF (iactiv .NE. 0) THEN
             IF (real(iactiv)*fiyave .GE. -cgr*fystop) THEN
C  Stop yawing (and reset misalignment buffer).
               WRITE (UNIT=iomesg,FMT='(A,F9.3)')
     &           ' Stop yaw action, time: ', ti
               iactiv = 0
             END IF
           ELSE
             IF (ti .GE. dtwait  .AND.  fiyave .GT. cgr*fystar) THEN
               WRITE (UNIT=iomesg,FMT='(A,F9.3)')
     &            ' Start yaw action in negative dir, time: ', ti
               iactiv = -1
             END IF
             IF (ti .GE. dtwait  .AND.  fiyave .LT. -cgr*fystar) THEN
               WRITE (UNIT=iomesg,FMT='(A,F9.3)')
     &            ' Start yaw action in positive dir, time: ', ti
               iactiv = 1
             END IF
           END IF
         END IF
C
C  *****************************************************************
C     E X I T   F R O M   T H E   C O N T R O L L E R
C
C  END IF (tctrl + 0.01*delt .GT. dtplc) THEN
      END IF
C
C  *****************************************************************
C  Apply yaw actions with rate 'yawrat',
C  use time-lag 'tauyaw' to 'smoothen' begin and end.
      IF (.NOT. lyaw) THEN
         dfiydt = (dt*real(iactiv)*yawrat +tauyaw*dfydtm)/(dt +tauyaw)
         d2fiy  = (dfiydt - dfydtm)/dt
         fiyaw  = fiym + (dfiydt + dfydtm)*dt/2.
      END IF
C
      RETURN
      END
```

Figure 27: Statements for yaw control in routine '*conrol*' (last part)

8 REMARKS

The purpose of the PHATAS development is to obtain a code with which the non-linear dynamic response and the corresponding loads in the structure can be calculated following the requirements of research and design. The strength of PHATAS is that it describes the integrated aerodynamic and structural dynamic behaviour of the turbine and tower, the response of the controller, and the options to simulate starts, stops, and failed conditions.

This chapter contains some remarks and limitations on the PHATAS model and on the accuracy of the calculated results. Most remarks follow from the assumptions on which the modelling of the structure and the aerodynamics are based.

8.1 Remarks on the modelling

Structural dynamic aspects

For the current generation of large size wind turbines (pitch-to-vane controlled variable speed) the program PHATAS shows to be well capable of analysing the structural dynamic response. In chosing the degrees of freedom to be included in dynamic analyses with PHATAS, the following should be taken into account.

Blade bending Blade bending deformation is described with degrees of freedom for the out-of-plane and for the in-plane displacement. Because rotors with a strong blade twist and rotors with pitch control have a strong coupling between flap-wise and lag-wise bending it makes *no sense* to solve only flap-wise or only lag-wise blade bending.

Flexible gearbox support The optional model for the dynamics of a flexible gearbox support has not been used recently, so it is not clear whether this will give convergence in combination with the current model for tower dynamics. The structural dynamics of the latter model has improved seriously with the development of a modular structure.

Options for former wind turbine configurations In the program PHATAS several options are still available that have been added to solve the dynamic response of former rotor concepts. Among these options are: **1:** Flapping hinge, **2:** Teeter hinge, **3:** Passive pitch motion of blade tips about an off-set axis, **4:** Synchronous generator, **5:** Active stall control, **6:** Bending-torsion coupling. These options (and the model for drive train dynamics) most probably still work well, but were not validated recently.

Aerodynamic modelling aspects

Some remarks on the models for rotor aerodynamics are listed below:

Use relative velocities For the calculation of the relative flow velocity at the blade sections, the motion of the sections due to the dynamics of all degrees of freedom, also from tower dynamics, is included. This implies that all structural dynamics have an appropriate aerodynamic damping.

Combined implementation of BEM aspects Much effort is spent on a realistic formulation of the aerodynamic momentum equations with a correct representation of the combined effects of: Oblique inflow; Turbulent wake state; Tip- and root loss factor; Dynamic inflow; Aerodynamic tower stagnation; Tangential induced velocity; Wind distribution (vertical shear). Most steady-state aspects of the PHATAS rotor aerodynamics model apply to the description in chapter 5 of [15].

Aerodynamic tower stagnation For a tubular tower the influence of the tower on the airflow ('stagnation' or 'tower shadow') is described with the three-dimensional potential flow around a semi-infinite dipole.
In former joint research projects it was concluded that using a potential-flow model gives an over-estimation of the tower stagnation (some other computer codes multiply this influence with 0.75.) It is more realistic to describe the aerodynamic drag of the tower with a double trailing vortex 'street'. In this case the dipole-model should have a smaller strength for the same tower diameter. A model in which this drag-influence is included has a smaller dipole strength for a given tower diameter and may give a smoother dip in the aerodynamic blade loading.
The (instationary) interaction from the blades passing the tower is *not* described.

Implementation of rotational effects Since a couple of years the correction models for the effects of rotation on the aerodynamic loads account for the 'local speed ratio' which represents the ratio between the driving forces (pressure) for rotational effects versus the dynamic pressure of the basic airfoil loads. The rotational 'correction' is thus multiplied with the factor $(\Omega\, r)^2/V_{\rm eff}^2$ which nearly equals former relation $\lambda^2/(\lambda^2+1)$. In the latest release "APR-2005" this 'local speed ratio' is evaluated for each position in the rotor plane, and includes the lateral flow component due to yawed operation.

Dynamic stall model The dynamic stall behaviour is described in terms of a first-order and second-order correction on the quasi-steady lift coefficient, following the heuristic model of H. Snel, [22]. The first-order correction is similar to other dynamic stall models. Because the development of this dynamic stall model was partly based on measurements on a single wind turbine with pitch variations only (while wind turbine rotors have flapping blades), the second-order correction of this model has no general validity. In wind engineering practise, the second order models are not used so much, which also does not contribute to the validity of these models for an arbitrary wind turbine.

8.2 Numerical Aspects of the Analysis

Following are remarks on numerical and modelling aspects of PHATAS.

Number of blade elements For an accurate representation the blade bending dynamics up to the second mode and the first blade torsional mode, a model with 15 elements (the default) appears to be sufficient.

The 'time_increment' The time increment Δt used in the calculation needs to be chosen following the requirement on the numerical damping and the deviation in natural frequencies for the dynamic behaviour with the highest frequency ν [rad/s] that is to be resolved, with the expression: $\rm max.\ allowable\ numerical\ damping = 1/2 \cdot (\nu \cdot \Delta t)^3$.

Here the damping is expressed as fraction of the critical damping.
If the dynamic stall behaviour is solved (**dynamic_stall** $>$ **0**) the time increment must be smaller than the time in which the relative airflow travels half a chord length at e.g. 80% span: $\textbf{time\_increment} < 1/2 \cdot {\rm chord}/(0.8\,\Omega \cdot R)$.

Operation at 90 degrees yaw The aerodynamic routines appear to converge bad for operation at turbulent wind with 90°yaw angle. For load cases at which this occurs (usually parked or idling), it is recommended to use a larger time increment.

Blade tip flexibility Rotor blades of multi-megawatt wind turbines in general have a pronounced tapered geometry, which aims to reduce the forces on the blade tip and likewise the moments in the blade root. Tapered blades also have a mass and stiffness distribution that decrease strongly towards the tip. If the bending stiffness decreases much stronger than the mass distribution, the outer part of the blade appears to be relatively flexible and may result in large (local) tip deflections. Disregarding the fact whether this stiffness-decrease is realistic, the numerical (convergence) problems in PHATAS may be avoided by using a constant bending stiffness over the outer blade element. Near the tip a stiffness correction has a negligible influence on the bending frequencies.

Convergence Slow convergence may be expected for a model with:

- Tower bending and drive train flexibility;
- Blade flap bending and a teetered hub or flapping hinges;
- Blade lead bending and drive train flexibility c.q. soft generator characteristics.

Nacelle mass properties If the rolling and tilting moments of inertia of the nacelle are not in compliance (too small) with the vertical position of the nacelle mass above the tower top (**static_moment_z**) then bad convergence may be expected.

A detailed discussion on the influence of the numerical solution on the equations solved is given in chapter 8 of [14].

8.3 Developments on PHATAS

In addition to the optional developments based on the modularity of the tower dynamics model, further developments of PHATAS may be addressed to:

Discrete tower damper Large wind turbines with a high tower appear to have a tower bending frequency that is close to the operational rotor speed. For many of these wind turbines the tower vibrations are reduced by installing a damper inside of the tower. With a small modification of the PHATAS tower module, such a damper can be implemented relatively easy as a mass-spring-damper system in both X and Y direction. Within the concept of the Craig-Bampton method the X- and Y- displacement of the damper is represented with another internal mode.

A similar extension is using the physical damping of the foundation (e.g. for rotation) in the tower damping matrix instead of the dimensionless damping of the first bending mode. This is based on the fact that the foundation has a dominant contribution to the tower dynamics. With the physical foundation damping the damping of each of the modes is then a result of the damping from the foundation.

Yaw control In the current release "APR-2005" a relatively simple but practical P-D model is present for rotor-speed (or power) control by pitch actions, see chapter 6. The dynamic loads on a wind turbine rotor are very sensitive to oblique inflow conditions. This means that a good dynamic model for yaw control (passive, active, or combined) may be of benefit for the design of a wind turbine to reduce fatigue loads.

Individual periodic pitch Wind turbine rotors with large diameter are sensitive to a wind gradient in any direction. This fact and the increasing relative material costs of large wind turbines are the reason for some manufacturers to apply individual periodic pitch of the rotor blades. At ECN individual periodic pitch is included in the development of wind turbine controllers that can be implemented in the PHATAS code.

Unsteady airfoil loads The unsteady aerodynamic loads on a rotor blade do not only comprise the effects of dynamic stall, but are also caused by the influence of local wake effects of the blade. The latter can be described in terms of 'apparent mass' and of 'shed vorticity'. For pitch-to-vane controlled variable speed wind turbines the latter 'unsteady aerodynamic loads' play an important role in the aerodynamic damping of in-plane blade vibrations. The 'flow curvature' effects from blade-torsion, pitch actions, and from local gradients of 'turbulence crossing' are already included in the model for blade aerodynamics.

The 'ONERA' model has been implemented in a former version (1992) of PHATAS, which may still be activated, tested, and issued for future releases. A more fundamental model for unsteady aerodynamics may also be implemented in following releases.

Vortex Wake model Although serious effort was spent (and will be spend) on a realistic implementation of several aspects of the Blade Element Momentum theory, a Vortex Wake model gives a more detailed description of the influence of the rotor wake on the rotor. Primarily a Vortex Wake model gives a more realistic description of the 'tip-loss', the effects of oblique inflow, and the effects of instationary blade loading. Reliable aerodynamic coefficients and a realistic model for the effects of rotation are still required for a Vortex Wake model.

REFERENCES

[1] Anderson, J.D. Jr. ;
'*FUNDAMENTALS OF AERODYNAMICS*'.
McGraw-Hill Inc. Singapore, ISBN 0-07-100767-9, 1991.

[2] Craig, Roy R. Jr. (University of Texas, Austin)
and Bampton, Mervin C.C. (Boeing company, Seattle);
'*Coupling of Substructures for Dynamic Analysis*'.
AIAA Journal, Vol.6, No.7, July 1968.

[3] Eecen, P.J. ;
'*WIND WAVES, Forces due to waves on offshore wind turbines*'.
ECN-C--03-097, Petten, September 2003.

[4] Frost, W., Long, B.H., and Turner, R.E. ;
'*ATMOSPHERIC ENVIRONMENTS GUIDELINES FOR USE IN WIND TURBINE GENERATOR DEVELOPMENT*'.
December 1978, NASA TP 1359.

[5] Germanischer Lloyd ;
'*Vorschriften und Richtlinien Nicht Maritime Technik, Teil 1 - Richtlinie für die Zertifizierung von Windenergieanlagen*'.
Ausgabe 2003.

[6] Hendriks, H.B. and Oudshoorn, H.L. ;
'*PROGSEQ-I USER'S MANUAL, Driver for Wind Turbine Design Tools version I*'.
ECN-C--94-108, Petten, May 1995.

[7] Hoerner, Dr. Ing. S.F. ;
'*FLUID DYNAMIC DRAG*'.
Published by the author, 1965.

[8] International Electrotechnical Commission ;
'*Wind Turbine Generator Systems, Part 1: Safety Requirements*'.
International Standard, IEC 61400-1:1999(E), Second edition 1999-02.

[9] International Electrotechnical Commission ;
'*INTERNATIONAL STANDARD, Wind turbines - Part 1: design requirements*'.
IEC 61400-1 Third edition, CDV, 2004.

[10] Kooijman, H.J.T. ;
'*BENDING-TORSION COUPLING OF A WIND TURBINE ROTOR BLADE*'.
ECN-I--96-060 , Petten, December 1996.

[11] Lange, D.F. de ;
'*Modelling of the wave load on offshore wind turbines*'.
ECN - Z&W Memo-99-017, Petten, March 1999.

[12] Lindenburg, C. ;
'*PHATAS-II PROGRAM STRUCTURE AND DERIVATION OF FORMULAE, Program for Horizontal Axis wind Turbine Analysis and Simulation version II*'.
ECN-C--93-038, Petten, June 1993.

[13] Lindenburg, C. ;
'*STABLAD, Stability analysis Tool for Anisotropic rotor BLADe panels*'.
ECN-CX--99-031 (confidential), Petten, June 1999.

[14] Lindenburg, C. ;
'*PHATAS-IV AEROELASTIC MODELLING, Release "DEC-1999" and "NOV-2000"*'.
ECN-CX--00-027, Petten, April 2001.

[15] Lindenburg, C. ;
'*INVESTIGATION INTO ROTOR BLADE AERODYNAMICS, Analysis of the stationary measurements on the UAE phase-VI rotor in the NASA-Ames wind tunnel*'.
ECN-C--03-025, Petten, June 2003.

[16] Lindenburg, C. ;
'*BLADMODE, Program for Rotor Blade Mode Analysis*'.
ECN-C--02-050, Petten, July 2003.

[17] Nederlands Normalisatie-instituut ;
'*Dutch Prestandard NVN 11400-0, Wind turbines - Part 0: Criteria for type-certification - technical criteria*'.
ICS 27.1980, 1-st edition, April 1999.

[18] Peeringa, J.M. ;
'*Streamfunction Wave Program, User's manual*'.
ECN-C--05-028, Petten, February 2005.

[19] Rosen, A. and Sheinmann Y. ;
(Faculty of Aerospace Engineering, Technion, Israel Institute of Technology, Haifa 32000)
'*The average output of a wind turbine in a turbulent wind*'.
In '*Journal of Wind Engineering and Industrial Aerodynamics*', Vol.51, pp.287-302, 1994.

[20] Schepers, J.G. and Vermeer, L.J. ;
'*EEN ENGINEERING MODEL VOOR SCHEEFSTAND OP BASIS VAN WINDTUNNELMETINGEN*'.
ECN-CX--98-070 (confidential), Petten, July 1998.

[21] Snel, H., Houwink, R., and Bosschers J. ;
'*SECTIONAL PREDICTION OF LIFT COEFFICIENTS ON ROTATING WIND TURBINE BLADES IN STALL*'.
ECN-C--93-052, Petten, December 1994.

[22] Snel, H. (ECN);
'*Heuristic modelling of dynamic stall characteristics*'.
European Wind Energy Conference, pp.429-433, Dublin Castle, Ireland, October 1997.

[23] Verheij, F.J., Föllings, F.J., and Curvers, A.P.W.M. ;
'*HANDBOOK WIND DATA FOR WIND TURBINE DESIGN - VERSION 3*'.
TNO - Institute of Environmental and Energy Technology, Apeldoorn, January 1991.

[24] Verhoef, J.P. and Schepers, J.G. ;
'*WAKEFARM, Gebruikershandleiding plus beknopte theorie*' (In Dutch).
ECN Memo, GR-74248-01, Rev. 1.0, Petten, March 1998.

[25] Winkelaar, D. ;
'*Onwikkeling en validatie van een stochastisch windbelastingmodel, Deel 1: Windmodellering*' (In Dutch).
ECN-C--96-096, Petten, December 1996.

A PROCESSING OF ROTOR CHARACTERISTICS

A.1 Selection of Rotor Characteristics

The PHATAS output file can be of three different types, with results from either (see section 2.3):

- Calculation of aerodynamic rotor characteristics;
- Power curve calculations;
- Calculation of the dynamic response.

The calculation of aerodynamic rotor characteristics is performed for a (specified) series of pitch angles and for a (specified) range of tip speed ratios. If a control algorithm is used (chapter 6) or modelled (chapter 7) the calculation of a power curve is including the quasi-steady response of this pitch controller.

The output files for all three types of calculations have a heading starting with an identification of the PHATAS release, and with identifications of the wind turbine model and load case.

With the program '*selchar*' tables with rotor characteristics can be selected from the output of aerodynamic rotor characteristics or power curve calculations.
The program *selchar* is invoked by typing its name followed by up to three `CHARACTER*64` arguments for the names of the files:

1 With input specifications for selchar ('selchar.in');
2 PHATAS-results with rotor characteristics or a power curve ('phatdef.pht');
3 For the output table with selected properties ('selchar.out').

If a file name is not specified as argument the name between brackets is used. Note that if an argument is used for a file-name specification, all preceding command line arguments also need to be given.

Figure 28 shows an example of the file with specifications for *selchar* for selection of a table with aerodynamic power as function of wind speed as used for a 2.7MW 90.5m diameter wind turbine.

The output file of *selchar* contains records with a property denoted as '*X*' variable in the first column (if '**x_type**' > 0) and in the following columns the properties denoted as '*Y*' variables. If the columns for the '*Y*' variables are related to the different pitch angles, the file is ended with an empty record and a record containing the corresponding tip speed ratios.

Following is a description of the input items for *selchar*:

heading `LOGICAL` ('*OFF*') $<ON, OFF>$
    If '*ON*' a heading is written.

comment_mark `CHARACTER*1` (' ')
    Character that is written in the first column of the heading lines.

phatas_file `CHARACTER*64` ('phatdef.pht')
    Name of the PHATAS output file with rotor characteristics.
    This filename can be overridden by a second command line argument.

output_file `CHARACTER*64` ('selchar.out')
    Name of file to which the selected properties are written.
    This filename can be overridden by a third command line argument.

x_type `INTEGER` (0) <0, 1, 2, 3, 11>

Indicates which property is written in the 'X' (first-) column:

- **= 0** No column with an 'X'-property,
- **= 1** Tip speed ratio: $\lambda = \Omega \cdot R/V_{\rm steady}$;
- **= 2** Inverse of the tip speed ratio: $1/\lambda$;
- **= 3** Wind speed at hub height [m/s]: $V_{\rm steady}$;
- **= 11** Element number. This gives the characteristics for each blade element. This is only available if the PHATAS output file is created for '**output_level 3** ' or higher, see chapter 3.2.

y_type `INTEGER` (1) <1, ..., 11>

Indicates which 'Y' property (for each pitch angle) is written in the following columns

- **= 1** Aerodynamic power coefficient: $C_{\rm K} = P_{\rm aero}/(\rho/2 \cdot V_{\rm steady}^3 \cdot \pi \cdot R^2)$;
- **= 2** Dimensionless aerodynamic power: $K_{\rm P} = C_{\rm K}/\lambda^3$;
- **= 3** Value of the generator power: $P_{\rm gen}$ [W];
- **= 4** Axial aerodynamic force coefficient (thrust): $C_{\rm Dax} = D_{\rm ax}/(\rho/2 \cdot V_{\rm steady}^2 \cdot \pi \cdot R^2)$;
- **= 5** Dimensionless aerodynamic axial force on the rotor: $K_{\rm D} = C_{\rm Dax}/\lambda^2$;
- **= 6** Axial aerodynamic force (thrust) on the rotor: $D_{\rm ax}$ [N];
- **= 7** Aerodynamic torque coefficient: $C_{\rm Q} = Q_{\rm aero}/(\rho/2 \cdot V_{\rm steady}^2 \cdot \pi \cdot R^3)$.
  Note that $C_{\rm Q} = C_{\rm K}/\lambda$ since $Q_{\rm aero} = P_{\rm aero}/\Omega$;
- **= 8** Dimensionless aerodynamic torque on the rotor: $K_{\rm Q} = K_{\rm P}$;
- **= 9** Aerod. torque on the rotor [N$*$m];
- **= 10** Aerodyn. blade pitch moment coefficient: $C_{\rm m\,pit} = \Sigma M_{\rm pitch}/(\rho/2 \cdot V_{\rm steady}^2 \cdot \pi \cdot R^3)$;
- **= 11** Performance of each blade element for the pitch angle indicated with '**pitch_number**' expressed as the aerodynamic power divided by the axial force times the local span. This can only be used for **x_type 11**.

y_factor `REAL` (1.0) [.]

Scaling factor for the 'Y' values in the table.

pitch_number `INTEGER` (1)

For **x_type 11**: number of the 'lambda-table' (pitch angle) following the order in the PHATAS output.

```
< Input file for program 'selchar' that is provided with PHATAS.
<
< Specifications consist of a key-name followed by the value.
< All lines starting with one of the characters  <  /  #
< are interpreted as comment lines.
< The specifications in this example do not correspond with
< the default values but describe a table for power curves.
<
heading      ON  <  A heading is written in the output table.
<
comment_mark '#' < Text lines in the heading start with
<                   the character '#' in the first column.
<
x_type       1   < Tip-speed ratio for X-column.
x_factor     1.0 < The property in the 'X'-column is multiplied with 1.
<
y_type       1   < Aero power coefficient selected in the 'Y' columns.
y_factor     1.0 < Not scale the power coefficients (multiply with 1)
<
pitch_number  1  <  Properties per blade element are selected
<                    for the first pitch angle.
< The calculated characteristics over the blade span
< of the first ('1') pitch angle are to be written.
< This is only significant for 'x_type 11' and 'y_type 10'
<
```

Figure 28: Input file for '*selchar*' for the selection of power coefficients

A.2 Annual Energy Capture

For integration of the energy capture of a wind turbine from P-V data the tool '*eprod*' is provided. If applied to PHATAS output with aerodynamic rotor characteristics or a power curve, the tool *eprod* reads the first table from the PHATAS output with aerodynamic rotor characteristics or a power curve and integrates the annual energy capture for a given Weibull wind distribution.

A.2.1 Input

When invoking *eprod* three `CHARACTER*64` arguments can be given for the filenames of:

1 *eprod* input specifications ('eprod.in');
2 File with aerodynamic rotor characteristics or a power curve ('phatdef.pht');
3 Filename for output of the calculated energy production.
If this name is not specified the results are written to the console.

The default values for the file names are written between brackets. Note that if an argument is used for a file-name specification, all preceding command line arguments also need to be given.

Figure 29 shows an example of the *eprod* input file with specifications that correspond with the default values. The syntax of the input specifications is similar to that for PHATAS.

Following are the input items for *eprod*:

phatas_file `CHARACTER*64` ('phatdef.pht')
Name of the PHATAS output file with rotor characteristics or a power curve.
If two or more command line arguments are given the second argument is used as file name.

weibull_mean `REAL` (6.67) [m/s] <0.01, ...>
Average wind speed of the Weibull wind distribution (not the "scale factor").

weibull_shape `REAL` (2.0) <1.0, ..., 8.0>
Shape factor *k* of the Weibull wind distribution.
The probability that the wind speed is between V_{low} and V_{high} is calculated with:

$$Probability = exp^{-(\Gamma_{(1+1/k)} \cdot V_{\mathrm{low}} / \mathbf{weibull\_mean})^k} - exp^{-(\Gamma_{(1+1/k)} \cdot V_{\mathrm{high}} / \mathbf{weibull\_mean})^k} .$$

For a Rayleigh wind distribution ($k = 2$) the function $\Gamma_{(1+1/k)} = \sqrt{\pi/4}$,
in which case the Weibull scale factor is $\sqrt{4/\pi} = 1.1283$ times '**weibull_mean**'.

v_in `REAL` (0.0) [m/s] <0.0, ...>
Lowest wind speed at which the turbine is operational c.q. the energy capture is integrated.

v_out `REAL` (100.0) [m/s] <'**v_in**', ...>
Highest wind speed at which the turbine is operational c.q. the energy capture is integrated.

minimum_power `REAL` (0.0) [W]
Minimum value of the power for which the turbine is operational. A negative value of **minimum_power** may result in "motoring" and thus give a small loss of energy capture.

rated_power `REAL` (1000.E+6) [W] <0.0, ...>
Nominal power. This is used as upper power limit in the integration of the energy capture.

availability `REAL` (1.0) <0.0, ..., 1.0>
Fraction of time during which the turbine is operational.

turbulence_15 REAL (0.0) [%] <0.0, ..., 50.0>
Turbulence intensity in percent of the wind velocity at 15m/s, following the IEC description.

slope_parameter REAL (3.0) <0.0, ..., 1.0E+7>
The turbulence intensity described with **turbulence_15** and **slope_parameter** is used to estimate the 'turbulent power'; the power including turbulence following the expression by Rosen et al. [19].

control_effect REAL (1.0) <0.0, ..., 1.0>
Efficiency of the controller near nominal power, applied only on the 'turbulent power'.

A.2.2 Output

An example of the calculated annual energy capture for a 2.7MW turbine is listed in Figure 30. The integration scheme used is analytically exact for the condition that the rotor characteristics in the PHATAS output are linear between two subsequent tip speed ratios.
The output also includes the energy production for a rotor with the Betz optimum; $C_{\mathrm{K}} = 16/27$, and the energy output with a correction for turbulence. For the example in Figure 30 the turbulence was zero.

```
<  Input specifications for the program 'eprod'
<
<  Name of the PHATAS output file:
phatas_file   turb91_PV.pht
<
weibull_mean    7.0     < [m/s] Average of wind distribution.
weibull_shape   2.0     < A factor 2 gives a Rayleigh distribution.
<
v_in            3.0     < [m/s] Lower bound for energy integration.
v_out          25.0     < [m/s] Upper bound for energy integration.
<
minimum_power   0.0     < No energy integration below 0W.
<
<  If a controller is modelled, power-limitation is not needed.
rated_power  2750.0E+3  < [W] Used for limitation of power.
availability    1.0     < Turbine is always (100%) available.
<
```

Figure 29: Input file for '*eprod*'

```
# Annual energy production from program 'eprod'.
# For Weibull mean  7.000 m/s   scale  7.898 m/s   shape  2.000
# Integrated from   3.000 m/s  to   25.000 m/s.
# IEC turbulence intensity I−15:   0.000 %  and slope param. 3.00
# PHATAS output file:  turb91_PV.pht
# Model_identif:  turb91m
# Load_case:  Turbine_Power_Curve
# Energy capture [kWh] for pitch angle:     0.000
# V_wind  Lambda  Stat_Power  Turb_Power    E_Betz     E_stat     E_turbul
   3.000  14.278  2.8431E+04  2.8431E+04  0.0000E+00  0.0000E+00  0.0000E+00
   3.500  12.303  6.3686E+04  6.3686E+04  2.7469E+04  1.7858E+04  1.7858E+04
   4.000  10.839  1.0952E+05  1.0952E+05  4.8272E+04  3.6512E+04  3.6512E+04
   4.500   9.712  1.6546E+05  1.6546E+05  7.6044E+04  6.1522E+04  6.1522E+04
   5.000   8.866  2.2977E+05  2.2977E+05  1.1119E+05  9.1873E+04  9.1873E+04
   5.500   8.651  3.0575E+05  3.0575E+05  1.5351E+05  1.2693E+05  1.2693E+05
   6.000   8.498  3.9610E+05  3.9610E+05  2.0245E+05  1.6677E+05  1.6677E+05
   6.500   8.378  5.0235E+05  5.0235E+05  2.5695E+05  2.1072E+05  2.1072E+05
   7.000   8.279  6.2495E+05  6.2495E+05  3.1484E+05  2.5725E+05  2.5725E+05
   7.500   8.201  7.6691E+05  7.6691E+05  3.7454E+05  3.0491E+05  3.0491E+05
   8.000   8.137  9.2874E+05  9.2874E+05  4.3424E+05  3.5207E+05  3.5207E+05
   8.500   8.087  1.1131E+06  1.1131E+06  4.9153E+05  3.9696E+05  3.9696E+05
   9.000   7.771  1.3029E+06  1.3029E+06  5.4466E+05  4.3475E+05  4.3475E+05
   9.500   7.452  1.4955E+06  1.4955E+06  5.8980E+05  4.6087E+05  4.6087E+05
  10.000   7.158  1.6739E+06  1.6739E+06  6.2474E+05  4.7255E+05  4.7255E+05
  10.500   6.887  1.8439E+06  1.8439E+06  6.4978E+05  4.6979E+05  4.6979E+05
  11.000   6.637  2.0101E+06  2.0101E+06  6.4563E+05  4.5624E+05  4.5624E+05
  11.500   6.406  2.1703E+06  2.1703E+06  5.7181E+05  4.3425E+05  4.3425E+05
  12.000   6.194  2.3302E+06  2.3302E+06  4.9674E+05  4.0612E+05  4.0612E+05
  12.500   5.997  2.4878E+06  2.4878E+06  4.2730E+05  3.7400E+05  3.7400E+05
  13.000   5.814  2.6430E+06  2.6430E+06  3.6403E+05  3.3931E+05  3.3931E+05
  13.500   5.749  2.7003E+06  2.7003E+06  3.0719E+05  2.9834E+05  2.9834E+05
  14.000   5.539  2.7004E+06  2.7004E+06  2.5678E+05  2.5214E+05  2.5214E+05
  14.500   5.349  2.7004E+06  2.7004E+06  2.1265E+05  2.0881E+05  2.0881E+05
  15.000   5.172  2.7003E+06  2.7003E+06  1.7449E+05  1.7134E+05  1.7134E+05
  15.500   5.006  2.7003E+06  2.7003E+06  1.4187E+05  1.3931E+05  1.3931E+05
  16.000   4.862  2.7003E+06  2.7003E+06  1.1430E+05  1.1224E+05  1.1224E+05
  16.500   4.708  2.7003E+06  2.7003E+06  9.1266E+04  8.9616E+04  8.9616E+04
  17.000   4.571  2.7003E+06  2.7003E+06  7.2222E+04  7.0917E+04  7.0917E+04
  17.500   4.440  2.7003E+06  2.7003E+06  5.6646E+04  5.5622E+04  5.5622E+04
  18.000   4.316  2.7003E+06  2.7003E+06  4.4038E+04  4.3242E+04  4.3242E+04
  18.500   4.199  2.7003E+06  2.7003E+06  3.3936E+04  3.3322E+04  3.3322E+04
  19.000   4.089  2.7003E+06  2.7003E+06  2.5923E+04  2.5454E+04  2.5454E+04
  19.500   3.984  2.7003E+06  2.7003E+06  1.9630E+04  1.9275E+04  1.9275E+04
  20.000   3.884  2.7003E+06  2.7003E+06  1.4736E+04  1.4470E+04  1.4470E+04
  20.500   3.789  2.7003E+06  2.7003E+06  1.0967E+04  1.0769E+04  1.0769E+04
  21.000   3.699  2.7003E+06  2.7003E+06  8.0921E+03  7.9459E+03  7.9459E+03
  21.500   3.613  2.7003E+06  2.7003E+06  5.9197E+03  5.8127E+03  5.8127E+03
  22.000   3.531  2.7003E+06  2.7003E+06  4.2935E+03  4.2160E+03  4.2160E+03
  22.500   3.446  2.7004E+06  2.7004E+06  3.0876E+03  3.0318E+03  3.0318E+03
  23.000   3.372  2.7003E+06  2.7003E+06  2.2016E+03  2.1618E+03  2.1618E+03
  23.500   3.301  2.7003E+06  2.7003E+06  1.5565E+03  1.5284E+03  1.5284E+03
  24.000   3.233  2.7003E+06  2.7003E+06  1.0912E+03  1.0715E+03  1.0715E+03
  24.500   3.168  2.7003E+06  2.7003E+06  7.5852E+02  7.4482E+02  7.4482E+02
  25.000   3.105  2.7003E+06  2.7003E+06  5.2284E+02  5.1340E+02  5.1340E+02
   7.000          8.4911E+05  8.4911E+05 1.80194E+07 7.44314E+06 7.44314E+06
```

Figure 30: Energy capture as integrated with '*eprod*'

B LOAD CASE PREPROCESSOR

The design load calculations following e.g. the G.L. Rules and Regulations [5], the IEC Recommendations [8], or the NVN 11400-0 Technical criteria (prestandard) [17] include the analysis of over hundred load cases. Preparing the input (error free) for all these load cases is a laborious task, especially because of the various types of wind loading and failure modes.

For variable-speed pitch-to-vane controlled wind turbines a tool *lcprep* (**l**oad **c**ase **prep**rocessor) has been developed with which a set of input files for each load case can be prepared relatively easy. Despite serious attempts in the development of *lcprep*, the input that is generated will not be fully complete because:

- Many turbine conditions such as faulted situations depend on the safety and control system. For this reason it is hard to define 'multi-purpose' input for the design load cases.
- The G.L. and IEC documents are presented as 'guidelines' or 'recommendations', and not as 'requirements'. This implies that for the certification different load cases may have to be considered, depending on e.g. the environmental and/or grid conditions.

In the following a description is given of the tool *lcprep* and its input. Also a description is given of the process of design load calculations with the steps that require special attention.

B.1 Input of *lcprep*

The load case pre-processor is invoked by typing it's name `lcprep` followed by a up to two `CHARACTER*64` command line arguments for the names of the files with (If a file name is not specified as command line argument the name between brackets is used):

1. Load case input ('lcprep.inp')
Input specifications of *lcprep* with design conditions (such as wind) and initial conditions (such as rotor speed and pitch). A description of these input items is given below.
An examples of this input file for a 2MW turbine is given in Figure 31.

2. Base input ('basecase.inp')
A file with the complete specifications of the turbine, the tower, the rotor blades, and the controller. The input file for each load case is constructed by copying this 'base' input file and adding the input specifications for each load case. So for simplicity it is recommended to make this file compact by references to other files using the *FILE* input item of PHATAS. In addition to a complete turbine description, this file should contain the numerical items (such as '**time_increment**') and the environmental conditions:

air_density	Density of the air. Following G.L. [5] and IEC [8]: 1.225kg/m^3;
vertical shear model	Following many recommendations: a power law with exponent 0.2;
wind_elevation	Elevation ('upflow') of the undisturbed wind direction;
mass unbalance	Difference in blade mass distribution to cover mass imbalance;
time_increment	Time increment used for most calculations;
data_increment	Increment in time steps for writing to the binary data datafile;
table_increment	Increment in time steps for writing to the ASCII output table.

When running *lcprep* some checks are performed on the syntax and the type of the input values before the load case input files are generated. Messages from these checks are written to file 'STATUS'. The specifications in the input file for *lcprep* have the same 'keyname - value' syntax as used in PHATAS. Following is a description of each of the input items by their key-name:

GL_additions `LOGICAL` (.FALSE.)
If '.TRUE.' the input for some additional load cases is generated as prescribed by Germanischer Lloyd [5] and not by IEC [8]. This includes the load cases DLC1.2, DLC1.13, DLC6.3, and DLC6.5.

rotor_positions `INTEGER` (1) <1, ..., 9>
Following the G.L. 'Rules and Regulations' [5] some of the extreme load cases have to be analysed with different initial rotor azimuth values. In *lcprep* input files for different rotor positions are generated for DLC1.3, DLC1.5, DLC1.6, DLC1.7, DLC1.8, DLC1.9, and DLC2.2. Although not recommended by G.L. different rotor positions are issued for generator short circuit, DLC2.2, because this load case is often design driving for the extreme blade loads. Depending on the rotor azimuth the gravity loads have another contribution to these design driving blade moments. The input files generated with *lcprep* always include at least 2 load cases for DLC2.2.
When using 2 or more **rotor_positions** the last digit is the 'rotor azimuth number'. If the loads in all 3 blades are included in processing of the results a load-set with 5 rotor positions for the extreme load cases cover blade positions with intervals of 360°/15 = 24°, which is smaller than the maximum interval of 30° recommended by G.L. [5]. For a 3-bladed rotor it is recommended *not* to use 3 or 6 rotor positions because this does not give substantially different blade positions.

rotor_diameter `REAL` (90.0) [m]
Outer diameter of the rotor, which is used in combination with the turbulence length scale Λ_1 **turblengt_scale** in the expressions for the extreme gusts and direction changes.

turbulence_15 `REAL` (0.16) [.]
The turbulence intensity I_{15} at an ambient wind velocity of 15m/s. This is used in combination with the turbulence parameter a in the expressions of the turbulences at the different wind velocities, see **turbulence_a**.

turbulence_a `REAL` (3.0) [.]
Parameter a used in the distribution of the turbulence with wind velocity:
$\sigma = I_{15}\,(15\,\mathrm{m/s} + a\,V_{\mathrm{hub}})/(1+a)$.

turblengt_scale `REAL` (21.0) [m]
Turbulence length scale parameter Λ_1 used for extreme gusts and direction changes.
Following G.L. [5] (and draft IEC ed.3 [9]): Λ_1 is the smallest of ($0.7\,z_{\mathrm{hub}}$) and (42.0m).
Following IEC ed.2 [8] and NVN [17] (and a draft G.L. proposal):
Λ_1 is the smallest of ($0.7\,z_{\mathrm{hub}}$) and (21.0m).

skip_time `REAL` (30.0) [s]
Time that has to be skipped in the processing of the results. This time is written as start time in the 'occurrences' file: *lcprep.occ*. This time is also added to the duration of a simulation ('**production_time**' and '**simulation_time**') which results in the total PHATAS calculation time.
For pitch controlled wind turbines it is recommended to use for **skip_time** the time for about 5 to 6 revolutions such that a quasi-steady operation is obtained before the gust or faulted condition starts.

wait_time `REAL` (35.0) [s]
Time at which events such as gusts and/or internal or external faults are modelled, which should be larger than **skip_time**. This is similar to the PHATAS input item **wait_time**.

simulation_time REAL (60.0) [s]
Time span for of the calculation of the response on a gust or a faulted situation, excluding **skip_time**. The input files for load cases with direction changes have a calculation time that is ((3 * **simulation_time**) + **skip_time**) such that yaw actions as result of the direction change are included. It should be noted that yaw actions will only take place if they are described in the PHATAS input or modelled in the controller.
The response of load cases that deal with a start procedure is calculated for 1.5 times **simulation_time** (or 2 times-, at cut-in wind).

production_time REAL (600.0) [s]
Time span for of the calculation of a the dynamic response of e.g. normal production, idling conditions, or operation with ice or failed yaw. After adding **skip_time** this time is written as 'end time' in file *lcprep.occ*.

misalignment REAL (8.0) [deg]
The misalignment or yaw tracking error of the wind turbine, in positive sense. The input for most of the load cases is generated with specification of the negative value of the misalignment, because for a high tip-speed ratio a negative misalignment usually gives the largest periodic load variations. The input for some load cases such as for production and for extreme direction change are generated for negative and positive direction change, see DLC1.3 and DLC1.8. In these input files the direction change has the same sign as the misalignment specification such that they add up to each other.

pitch_error REAL (0.0) [deg]
Difference in blade pitch angle, which is used to represent the aerodynamic rotor imbalance. For the load cases with an idling rotor this **pitch_error** is subtracted from the pitch angle of blade 2 and added to the pitch angle of blade 3.

start_wind REAL (3.0) [m/s]
Smallest wind velocity at which the turbine is in power production. This is used for starts and stops at the 'cut-in' wind speed. Due to the misalignment and the elevation (8^o) of the undisturbed wind direction the electric power of the turbine may still be negative, in which case one may think of using a slightly higher **start_wind**.

sub_rated_wind REAL (99000.0) [m/s]
Wind velocity for which the input files of extreme load cases are generated.
For 'classical' wind turbines the power curve has a clear 'edge' at the rated wind speed, where the distribution of the steady-state axial aerodynamic rotor thrust has its maximum. For these turbines one may expect the maximum loads to occur at the rated wind speed or at the cut-out wind speed, which means that analysing the gust responses at these wind speeds may suffice.
Modern large size wind turbines however have a pitch-to-vane strategy that starts below the speed for rated power. The effect of this pitch strategy is that some of the large loads near rated power are reduced, also called 'peak shaving'. A similar strategy can be applied to the torque-speed relation of the generator. The effect of these 'load reducing' strategies is that the assessment of the ultimate loads on the wind turbine requires analyses of the extreme gust responses for more wind velocities than only 'rated' and 'cut-out' For this purpose the input item **sub_rated_wind** can be used.
If the value of **sub_rated_wind** is larger than the value of **rated_wind** (such as the default values) then no input files are generated for 'sub-rated' wind conditions.

rated_wind REAL (12.0) [m/s]
Wind velocity used for the (extremes) load-case input files.

cutout_wind REAL (25.0) [m/s]
Wind velocity used for the (extremes) load-case input files.

reference_wind REAL (42.5) [m/s]
In many design recommendations the 'reference wind speed' is 5 times the annual average wind speed of the design load spectrum. The input for DLC6.1 (idling) is generated with 1.4 times the reference wind. The input for DLC7.1 (idling with a fault) is generated with 1.05 times the reference wind.

start_omega REAL (10.0) [rpm]
Smallest rotor speed at which the turbine is in operation. This item is used for the initial conditions of some input files.

rated_omega REAL (16.0) [rpm]
Initial value of the rotor speed used in the input for load cases that start at **sub_rated_wind** and at **rated_wind**.

cutout_omega REAL (18.0) [rpm]
Initial value of the rotor speed used in the input for load cases that start at **cutout_wind**. *The value of* **cutout_omega** *should be smaller than the PHATAS input specifications for* **brake_overspeed** *and for* **generator_emergency**.

wait_pitch REAL (60.0) [deg]
Pitch angle below the cut-in wind speed for a rotor in stand-by condition. This is used in the input for starts at the cut-in wind, sub-rated wind, and rated wind speed.
The controller enters the start-mode **1**: if the initial rotor speed is smaller than 90% of **start_speed** (section 3.2.3), **2**: if the generator is disconnected, **3**: if no stop is specified, and **4**: if the pitch angle is larger than (0.6 * **min_pitch_angle** + 0.4 * **max_pitch_angle**) The latter implies that for a successful start procedure **wait_pitch** should be large enough.

minimum_pitch REAL (0.0) [deg]
Initial pitch angle used in the input for load cases that start in partial load, so below the sub-rated wind speed. *The value of* **minimum_pitch** *should be between the PHATAS input values for* **min_pitch_angle** *and* **max_pitch_angle**.

sub_rated_pitch REAL (2.0) [deg]
Initial pitch angle used in the input for load cases that start at the sub-rated wind speed.

rated_pitch REAL (2.0) [deg]
Initial pitch angle for the load cases that start at the rated wind speed.

cutout_pitch REAL (23.0) [deg]
Initial pitch angle used in the input for load cases that start at the cut-out wind.

idling_pitch REAL (90.0) [deg]
Pitch angle used for the load cases with an idling rotor, which includes the 'start' load cases at rated and cut-out wind. *The value of* **idling_pitch** *should be between the PHATAS input values for* **min_pitch_angle** *and* **max_pitch_angle**.

short_increment REAL (0.01) [s]
Time increment used for the calculation of the response on a generator short-circuit as in DLC2.2. This time increment has to be small because a short-circuit usually leads to a fast torque variation. This may be about 2/3 of the time increment for the other calculations.

B.2 Output of *lcprep*

After successful use of *lcprep* the following files are generated.

Input files A set of input files for each load case calculation. The names of these files include the G.L. load case number followed by a digit related to the wind velocity (e.g. rated or cut-out). Except for DLC1.10 and DLC1.13, the load-case names are ending with a digit that is a counter for the initial rotor azimuth (extreme load cases) or for the random seed and/or sign of the misalignment (production load cases). These input files contain references to the turbine input (a copy from the 'base input file') followed by specifications of initial conditions (rotor speed, yaw angle, pitch), the state of the turbine, and external (wind) conditions.

'fatigue.bat' (Unix or Linux: **'fatiguejob')** Script for calculating the response of all fatigue load cases.

'extreme.bat' (Unix or Linux: **'extremejob')** Script for calculating the response of all extreme load cases.

'lcprep.occ' A file in which all load cases are listed, including occurrences for the fatigue load cases, the start- and end- time for processing the calculated dynamic behaviour, and the partial safety factors for each of the load cases. This file can be read when selecting extreme sectional loads with '*loadex*', see Appendix D.

Syntax of input file names

The input files for each of the load cases have a name that starts with a number and have the extension `.inp` . For the input files generated with *lcprep* this number has 4 digits. The number in the load case file name is constructed as follows:

digit 1 and 2 Load case numbers following IEC [8] and G.L. [5] for load cases DLC1.3 through DLC8.2. For load cases DLC1.10 (with ice) and DLC1.13 (grid loss), the first three digits are used for the load-case number.

digit 3 For some load cases the recommendations prescribe a calculation for different wind velocities. The third digit in the load case name (for load case DLC1.10 or DLC1.13 the fourth digit) is related to this wind velocity as follows:

`0`	**start_wind**
`1` (and `2` and `3`)	**sub_rated_wind**
`4` (and `5` and `6`)	**rated_wind**
`7` (and `8` and `9`)	**cutout_wind**

Here a value `2` and `3` also denote a calculation at the sub-rated wind speed, but either with different sign of the extreme direction change, with different direction of extreme wind shear variation, or with different combination of gust and start or stop or grid loss. The same holds for a value `5` and `6` at the rated wind and for `8` and `9` at the cut-out wind.

digit 4 Except for DLC1.10 and DCL1.13 the fourth digit is reserved to vary the random seed for turbulent wind (SWIFT) or for different initial rotor azimuth values of extreme load cases, see **rotor_positions**.

Description of the load case input files

Following is a list of the load case input files generated with *lcprep* with a description of what they simulate. Depending on the safety and protection system of the turbine and of the external conditions (special wind climate, weak grid) the description of some of these load cases may require some modifications, while it may also be necessary to add some load cases.

In general this means that pre-processing of the load-case input always requires the attention of the user (designer), for which purpose a short description of the load cases generated with *lcprep* is given below based on the dynamic load case number following the G.L. Rules and Regulations [5]. The 'occurrences' mentioned are for 20 years of operation. The 'Partial Safety Factor' for the load cases that are written in file 'lcprep.occ' are 1.35 for Normal conditions, 1.1 for Abnormal conditions, and 1.5 for Transport and erection. In this description, the last digit in the file names (for different random seed, or different initial rotor azimuth) is denoted with an italic '*i*'.

Description of fatigue load cases

Following is a description of the fatigue load cases, for which a non-zero value for 'occurrences' is written in file 'lcprep.occ'. The fatigue load cases can be calculated with the batch job 'fatigue.bat', see section B.4.

DLC1.2 The input files for normal production start with the digit '0'. This is done to use the 2-nd and 3-rd digit for the average wind velocity and still have unique load-case names. For the current release of *lcprep* input files for normal production are generated with wind intervals of 1m/s. For each '1m/s value' of the average wind velocity two input files are generated, for negative and for positive misalignment. The 4-th digit of the file names is '1' and '2' respectively. The random seed for the turbulent wind generation with SWIFT is also written in these input files following the (comment) string <Swift_random_seed .
The occurrences written to file 'lcprep.occ' are calculated for 20 years of operation (= 631136000s) using the **production_time** and assuming that the annual average wind velocity is 1/5 of the value **reference_wind**.

The G.L. 'Rules and Regulations' also require the calculation of the response for a slow wind velocity variations. The input files for these 'slow' transients are 1210.inp and 1240.inp for a wind-speed variation from cut-in to rated and from (sub-)rated to cut-out respectively. These wind speed variations cover the low-frequency fatigue cycles due to changes in average wind velocity. The input files for these load cases describe a normal wind velocity NWP that increases and decreases slowly in two periods of 250s. The wind variations from cut-in to rated occur 6000 times and the wind variations from (sub-)rated to cutout occur 1000 times.

DLC1.4 The input files for grid loss are generated for sub-rated, rated, and cut-out wind, see files 1410.inp, 1440.inp, and 1470.inp respectively. Grid loss is simulated by generator disconnection after **wait_time**. The occurrences for these load cases are set to 1.

DLC1.10 The input for the load cases with ice is generated for rated, for cut-out, and if applicable also for sub-rated conditions, see files (1104.inp, 1105,inp, 1107.inp, 1108.inp, and eventually 1101.inp and 1102.inp). Here the files (1101.inp,) 1104.inp, and 1107.inp describe the state of all blades iced except blade 1, and the input is assumed to be given (e.g. by FOCUS) in file `twobladice.inp`. The files (1102.inp,) 1105.inp, and 1108.inp describe the state with all blades iced, and the input is assumed to be given (e.g. by FOCUS) in file `allbladice.inp`.
The occurrences for these load cases are set to 1.

DLC2.3 The input files for a control or protection system fault are generated for 3 fault types:

Operation with increased (failed) yaw (of -20^o)	2310.inp, 2340.inp, 2370.inp
Pitch of blade 2 blocks	2320.inp, 2350.inp, 2380.inp
Blade 2 pitches (with **start_pitch_rate**)	2330.inp, 2360.inp, 2390.inp

Operation with increased yaw occurs 960hr/20y at (sub-)rated wind [17] and 40min/20y at cut-out wind. Stop-actions as result of pitch fault occur 100 times at (sub-)rated wind, and 100 times at cut-out wind.
The faulted situations that may occur depend strongly on the safety and control system of the turbine, for which reason these input files and the occurrences will probably require modification, see also the Germanischer Lloyd 'Richtlinie .....', [5].

DLC3.1 The normal start procedures of the turbine are simulated with pitching the blades (after **skip_time**) to working position. For the cut-in wind speed, the sub-rated wind speed and the idling wind speed the initial pitch angle is **wait_pitch** while for the cut-out wind velocity the initial pitch angle is **idling_pitch**. Because of the yaw misalignment and the elevation of the undisturbed wind (of 8^o) it may be necessary to increase the value of **start_wind**. The occurrences for starts are 20000 times at cut-in wind [5, 17], 1000 times at (sub-)rated wind [5], and 1200 times at cut-out wind [17].

DLC4.1 The input files for the normal stop procedures contain simply the specification of **brake_time** which activates a mechanical brake and/or a pitch action, following the turbine specifications in the 'base input file'. The occurrences for stops are 20000 times at cut-in wind [5, 17], 1000 times at (sub-)rated wind [5, 17], and 1000 times at cut-out wind [5].

DLC5.1 The input files for an emergency shutdown are generated for a simultaneous generator disconnection and activation of a stop procedure. If the turbine undertakes specific actions during emergency situations, these should be specified in files 5110.inp, 5140.inp, and 5170.inp for sub-rated, rated, and cut-out wind.
The emergency stops occur 200 times at (sub-)rated and 200 times at cut-out wind speed, see [17].

DLC6.4 (IEC ed.2: DLC6.2) The input files for stand-by above the cut-out wind are generated for an idling rotor in downwind position, at a turbulent wind with an average that is 1m/s above the cut-out wind (6410.inp) and also for a turbulent wind with an average of 0.7 V_{ref} (6420.inp). The occurrences of these load cases in the fatigue load set should cover the time that the rotor is idling above the cut-out wind for the design wind distribution.
The occurrences for these load cases depend on the average design wind speed, so they probably have to be corrected.

Description of extreme load cases

Following is a description of the fatigue load cases, for which a non-zero value for 'occurrences' is written in file '`lcprep.occ`'. The extreme load cases can be calculated with the batch job '`extreme.bat`', see section B.4.

DLC1.3 The input files for the extreme coherent gust with direction change DLC1.3 are generated for a negative (131*i*.inp and 134*i*.inp) and for a positive (132*i*.inp and 135*i*.inp) direction change. Here the direction change is added to the 'normal' misalignment.

DLC1.5 The input files for grid loss in combination with a 1-year extreme operating gust are generated for 3 different time intervals between the grid loss occurrence and the start of the gust. Following the recommendations of G.L. [5] the grid loss occurs **1**: at the first minimum value of the wind velocity, **2**: at the strongest wind gradient, and **3**: at the maximum of the wind velocity. This is defined in files 151*i*.inp through 153*i*.inp for the sub-rated wind, in files 154*i*.inp through 156*i*.inp for the rated wind, and in files 157*i*.inp through 159*i*.inp for the cut-out wind. It is recommended to verify whether the extreme loads tend to occur at (or near) the 'middle' load cases 152*i*, 155*i*, and 158*i*. If not, the time of the grid loss should be modified.

DLC1.6 The input files for the 50-year extreme operating gust are written in files 161*i*.inp, 164*i*.inp, and 167*i*.inp for sub-rated wind, rated wind, and for cut-out wind.
Probably these iput files require no modification.

DLC1.7 The input files for the extreme wind shear variation are written in files 171*i*.inp through 173*i*.inp for sub-rated wind, in files 174*i*.inp through 176*i*.inp for rated wind, and in files 177*i*.inp through 179*i*.inp for the cut-out wind. The different files for each wind velocity are for a wind shear variation in vertical direction, to the right, and to the left. The load case description generated with *lcprep* also simulate the 'return' to the normal wind profile.
Probably these input files require no modification.

DLC1.8 The input files for the 50-year extreme direction change are generated for different wind velocities and for both negative (181*i*.inp, 184*i*.inp, and 187*i*.inp) and positive (182*i*.inp, 185*i*.inp, and 188*i*.inp) direction change. The direction change is added to the 'normal' yaw misalignment. One may consider to model yaw control.

DLC1.9 The input files for the 50-year extreme coherent gust are generated for the sub-rated wind speed (191*i*.inp and 192*i*.inp) and for the rated wind speed (194*i*.inp and 195*i*.inp). Here 191*i*.inp and 194*i*.inp are for a negative yaw tracking error and 192*i*.inp and 195*i*.inp for a positive yaw tracking error. Besides the fact that an extreme coherent gust at rated wind speed may finally lead to a shut-down because of exceedance of the cut-out wind, these load case input files require no modification.

DLC1.13 (G.L. only) The input for the load case with grid-loss is generated for rated (1131.inp) and cut-out wind (1137.inp), and if applicable also for sub-rated (1134.inp) conditions. The input files specify a generator disconnection. These load cases occur 200 times at (sub-)rated wind and 200 times at cut-out wind.

DLC2.1 The input files for 'control or protection system fault' (2110.inp, 2140.inp, and 2170.inp) are generated for a grid loss together with pitch failure of one of the blades.
Additional files are generated for the case that the yaw controller drives to an additional negative misalignment of -50^o (2120.inp, 2150.inp, and 2180.inp), or to an additional positive misalignment of +50^o (2130.inp, 2160.inp, and 2190.inp).

DLC2.2 The input files for a short circuit (221*i*.inp, 224*i*.inp, and 227*i*.inp) include the specifications of a generator torque transient. Because this torque transient has a frequency of the grid (or twice this frequency) a shorter calculation increment is used while the output is written each time step. If the number of rotor positions **rotor_positions** is 1 then the input files are still generated for 2 initial rotor positions.
For this '*Abnormal*' condition the partial safety factor is 1.1.

DLC3.2 The start procedures with the occurrence of a 1-year extreme operating gust are simulated with different times at which the gust interferes with the start procedure, see e.g. files 3240.inp, 3250.inp, and 3260.inp for the rated wind velocity. For the same reason as for DLC1.5 it is recommended to verify whether the start time of the gust approaches the most unfavourable loads.

DLC3.3 The start procedures with the occurrence of a 1-year direction change are simulated with different times at which this direction change takes place. The input files for this load case only deal with a negative direction change, assuming that in combination with a positive vertical shear a positive direction change gives the strongest periodic load variations. For the same reason as for DLC1.5 and DLC3.2 it is recommended to verify whether the start time of the direction change approaches the most unfavourable loads.

DLC4.2 The stop procedures with the occurrence of a 1-year extreme operating gust are simulated with different times at which the gust interferes with the stop procedure, see e.g. files 4240.inp, 4250.inp, and 4260.inp for the rated wind velocity. For the same reason as for DLC3.2 it is recommended to verify whether the stop time of the gust approaches the most unfavourable loads.

DLC6.1 The input for the 50-year extreme wind is generated for the turbine idling downwind and a slow sinusoidal wind direction variation with 15^o amplitude: file 6110.inp.

DLC6.2 (IEC ed.2: DLC6.1) The input file for grid loss at the 50-year extreme wind is generated for the wind turbine in downwind orientation, which is assumed to be the 'parked state' at high wind. The consequence of grid loss is simulated as if the yaw controller is not active for which the wind direction is varied with 1.25^o/s over 360^{o}and back, see file 6210.inp. If the backup system of the wind turbine is sufficient to operate for hours or if the wind turbine may be free-waying in the 'parked state', this load case has to be re-defined according to the reaction of the wind turbine.
For this '*Abnormal*' condition the partial safety factor is 1.1.

DLC6.3 (G.L. only) The input for the 1-year Extreme Windspeed model with large misalignment is generated for the wind turbine idling in downwind position with a slow sinusoidal wind direction variation with a 30^o amplitude: file 6310.inp.

DLC6.5 (G.L. only) The input for the 50-year extreme direction change at the reference wind speed is generated for a negative and positive direction change: files 6510.inp and 6520.inp respectively. The initial orientation of the turbine is assumed downwind, with the 'normal' misalignment specified with **misalignment**. The rotor has to be modelled with 30mm ice (700kg/m^3) on the entire blade surface (in fact all non-rotating parts) which has to be specified in file `idlebladice.inp` .

DLC7.1 The input file for stand-by after occurrence of a fault is generated for an idling rotor that operates 1m/s above the cut-out wind. Here blade 2 remains in working position (partial load) while the other blades are in vane position. For the blade 2 in working position, no pitch error is applied.
For some wind turbines this load-case may contribute to the fatigue loads.
For this '*Abnormal*' condition the partial safety factor is 1.1.

DLC8.1 The input files for the maintenance load cases are generated for the sub-rated, the rated, and the cut-out wind (files: 8110.inp, 8140.inp, and 8170.inp). The state of maintenance is defined as for a fixed rotor position. However, in order to cover all possible rotor positions the rotor is rotating slowly with a given rotor speed of 0.2rpm, which approaches the dynamics of a fixed rotor position. A 0.2rpm rotor speed gives 2 complete revolutions during the 10 minute simulation time for turbulent wind.
For these '*Transport and erection*' conditions the partial safety factor is 1.5.

DLC8.2 The input file for the maintenance with a halted rotor (8210.inp) is generated for a wind velocity of 1.05 times V_{ref}. Here the rotor is halted by a strong brake torque. In the simulation the rotor is rotating with the slow speed of 0.2rpm such that every rotor azimuth is covered. The wind direction varies slowly over 360$^{\mathrm{o}}$ and back. In case of a free yawing (with friction) nacelle this load case input has to be redefined.
For this '*Abnormal*' condition the partial safety factor is 1.1.

B.3 Actions before running PHATAS

Directory structure

The input files and the scripts that are generated for the load-set calculations assume the following sub-directories for the PHATAS files:

Unix / Linux	MS-Windows	Contents of directory
`/swftwind`	`\SwiftWind`	SWIFT input files and wind datafiles;
`/phat_err`	`\Messages`	Files with run-time messages;
`/phdata`	`\Loadfiles`	Binary files with the dynamic response;
`/models`	`\Models`	Files with global properties of the turbine model;
`/phatout`	`\phatout`	ASCII output table (with time series).

Input for iced rotor blades

Load case DLC1.10 deals with operation with iced blades. The input files for this load case 1104.inp, 1105.inp, 1107.inp, and 1108.inp and if applicable also 1101.inp and 1102.inp are generated with a reference to either the file `allbladice.inp` or `twobladice.inp`. These files must contain the rotor blade mass distribution for all blades iced and for all-except-1 blades iced. For load case DLC6.5 (following G.L. [5]) the file named `idlebladice.inp` has to be present with the description of 30mm ice (700kg/m^3) on all surfaces.

SWIFT input files

For performing the PHATAS calculations of the fatigue load cases and of the extreme load cases, the scripts `fatigue.bat` and `extreme.bat` (PC version) are generated by *lcprep*. Before running these scripts the input files for the program SWIFT have to be prepared for the load case files:

1101.inp, 1102.inp, 1104.inp, 1105.inp, 1107.inp, 1108.inp (Production with ice)
231*i*.inp through 239*i*.inp (Operation with occurrence of fault)
641*i*.inp, 642*i*.inp (Stand by above the cut-out wind)
711*i*.inp (Parked after occurrence of a fault)
811*i*.inp, 814*i*.inp, 817*i*.inp (Maintenance)

and eventually for load cases that have been added by hand. In the names as written here the letter *i* may indicate the index for different sign of misalignment, random seed, or initial rotor-azimuth.

A large part of the input files for SWIFT describes the geometry of the rotor and the grid of the wind files which is thus identical for each load case. The input files for each individual load case can be obtained by modification of:

- Ambient wind velocity at hub height;
- Turbulence intensity (IEC class or **S**pecial), if applicable including park effects.
  Without park effects, the SWIFT input value is the same for the antire load-set;
- Random seed, which should be different for each wind file and for which prime numbers larger than 1000 are recommended. The PHATAS input files for each of the load cases with turbulent wind contain a unique prime number larger than 1000 after the comment string `<Swift_random_seed` ;
- The name of the wind datafile, which should correspond with the file name mentioned in the load case input file.

B.4 Checks after running PHATAS

The dynamic response of the fatigue and extreme load cases can be calculated simply by using the PC batch files `fatigue.bat` and `extreme.bat` ,
or the Unix or Linux scripts `fatiguejob` and `extremejob` .
If load cases have been added ('by hand') in addition to those already generated with *lcprep*, the commands for PHATAS response calculation have to be added to these scripts, or these calculations have to be started by console-commands of the user.

In general it is recommended to verify the simulation of faults and transients by plotting the calculated response of each load case. The time series for these plots can be generated with the postprocessor '*phpost*', see Appendix C.

Special attention has to be paid to the load cases in which a start, stop, grid loss, or any type of fault interferes with the time at which a gust is simulated. This deals with the load cases for which the input is generated with *lcprep*: 1511.inp through 1591.inp, 3210.inp through 3290.inp, 3310.inp through 3390.inp, and 4210.inp through 4290.inp.

B.5 Actions for postprocessing

After performing all fatigue and extreme load-case calculations the loads can be processed with e.g. *loadex* to extreme sectional loads and Time-At-Level counting for the bearing design. In particular for the processing of the fatigue loads, the number of occurrences for each load case has to be present in a file (usually named `loadset.occ` . This can be done by copying the file `lcprep.occ` (generated by *lcprep*) to `loadset.occ` . This file contains the occurrences for 20 years of operation. It is recommended to verify the occurrences for some of the load cases because they depend on the operational control of the wind turbine. This holds in particular for the load cases with ice (110*i*.inp), the load cases with a grid loss (1410.inp, 1440.inp, and 1470.inp) and the load cases with a control or protection system fault (23*i*0.inp). It may be the case that load case 7110.inp contributes to the fatigue loading.

Finally one should not forget to add the names/numbers of the load cases that were added ('by hand') to the set of files generated with *lcprep*.

```
<
<  Input file for 'LoadCase PREProcessor'  with parameters
<  of the design wind conditions and the operational
<  conditions of the turbine at cut-in, rated, and cut-out.
<
<  *** Input items that describe the load-case conditions. ***
<   The turbulence distribution is following  IEC class II-B.
reference_wind   42.5   < 5 times the annual average design speed.
turbulence_15     0.16  < Turbulence level at 15m/s
turbulence_a      3.0   < Turbulence distribution foll IEC-IIB
turblengt_scale  21.0   < [m] follow. IEC (G.L.:  42.0m)
GL_additions     ON     < For additional load cases, to comply with G.L.
rotor_positions   2     < 2 Rotor positions for extremes and transients.
<
<  *** Turbine-related input items. ***
rotor_diameter   77.4   < [m]
misalignment      8.0   < [deg] Used for some of the load cases.
pitch_error       0.3   < [deg] Represents aerodynamic imbalance.
start_wind        3.5   < Used for start and stop case.
sub_rated_wind   11.0   < [m/s] Also for loads just below rated.
rated_wind       13.0   < Used for many load-cases.
cutout_wind      25.0   < [m/s] Highest wind for normal operation.
idling_pitch     88.0   < Idling at high wind.
<
<  *** Numerical items used for the calculations. ***
skip_time        25.0   < [s] is skipped in processing results.
wait_time        30.0   < [s] when gusts and transients start.
production_time 615.0   < [s] normal operation at turbulent wind.
simulation_time  60.0   < only for short events (gusts, stops).
short_increment   0.01  < [s] Used for short-circuit (DLC2.2).
<
<  *** Used for initial conditions of the calculations. ***
start_omega      10.0   < Rotor speed at smallest wind.
rated_omega      19.1   < Speed at rated wind.
cutout_omega     19.5   < Speed at cut-out wind.
wait_pitch       50.0   < Pitch angle awaiting start procedure.
minimum_pitch     0.5   < Smallest pitch angle (optimum lambda).
sub_rated_pitch   4.0   < [deg] Just below rated (e.g. peak shaving).
rated_pitch       4.5   < Pitch angle at rated- (peak shaving!).
cutout_pitch     23.0   < Pitch angle at cut-out wind.
/
```

Figure 31: Input file for LCPREP for a 2MW turbine

C POSTPROCESSOR *'PHPOST'*

This section contains a description of the postprocessor '*phpost*'. This postprocessor can be used to retrieve time series or extreme values of the properties (e.g. structural loads) of dynamic response from binary files written by PHATAS-IV release "JUL-2001", "JAN-2002", "OCT-2002", "NOV-2003", or "APR-2005". Compared to the PHATAS output, the time series generated with the postprocessor '*phpost'* can not include the aerodynamic sectional properties (with **signal_number** 111 to 119) see chapter 4.

In addition to using the postprocessor that is provided, one can also read the contents of the binary datafiles with a self-written processor. For this purpose the contents of the datafiles are listed in Appendix F.

C.1 Using the Postprocessor

The postprocessor is invoked by typing it's name `phpost` followed by a up to three `CHARACTER*64` command line arguments for the names of the files with:

1 Input specifications of the results be selected ('postinp');
2 Binary datafile with time response from PHATAS ('phatdef.tbh');
3 Output of the postprocessor ('phatdef.tim').

If a file name is not specified as command line argument the name between brackets is used.

The postprocessor performs some checks on the syntax and on the type of the input values before the program starts processing. Messages of these checks are written to file 'STATUS'.

C.2 Description of Input Variables

Following are the input items for the postprocessor *phpost*.

FILE `CHARACTER*64`
Name of an additional input file. The input data in (an) additional input file(s) are read as if this file is inserted at the position of its specification.

datafile_name `CHARACTER*64` ('phatdef.tbh')
Name of the PHATAS binary data file that is to be read. If two or more command line arguments are specified the second command line argument overrides this specification.

output_file `CHARACTER*64` ('phatdef.tim')
Name of the ASCII output file with time series. If three (or more) command line arguments are specified the third command line argument overrides this specification.

comment_mark `CHARACTER*1` (' ')
Character that is written in the first column of the heading of the file '**output_file**'.

output_type `INTEGER` (1) $<0, ..., 5>$
Type or format of the ASCII output file written by '*phpost*':

$=$ **0** An output table without heading, of which all columns can be specified, see **output_table**.

$=$ **1** A user-defined output file similar to the PHATAS output file with a heading of 7 records. The first three columns contain the time [s], the number of iterations, and the rotor azimuth [^{o}] respectively.

$=$ **2** The minimum, average, and maximum values of the properties selected.

$= \mathbf{3}$ A file with load time series of a specified blade cross section.
The file has a heading similar to the PHATAS output file.
The sixth record of the heading contains (after the '**comment_mark**'):
- blade number ;
- spanwise location [m] measured from the blade root '**blade_root_radius**' .

The seventh record in the heading contains (here for blade 1):

```
time  it azimuth   151   152  -153   154   155   156
```

The numbers refer to the output properties and are explained in chapter 4.
Each of the following records contains for every time step:

time actual time [s] ;
it number of iterations ;
azimuth rotor azimuth [°] ;
151 edgewise bending moment [N·m] (pos. in lead direction) ;
152 blade torsional moment [N·m] (opposite to pitch) ;
-153 flatwise (backwards) bending moment [N·m] ;
154 flatwise shear force [N] (positive in downwind direction) ;
155 spanwise (tensile) force [N] in the blade cross section ;
156 edgewise shear force [N] (positive in lead direction).

$= \mathbf{4}$ A file with load time series for a cross section of the rotor shaft.
This file has the same format as for '**output_type 3**' except that on the sixth record a '0' is written instead of the blade number and that the distance behind the rotor centre is written instead of the spanwise location.
The seventh line in the heading of the table looks like:

```
time  it azimuth  -62    61    63   -65   -64   -66
```

$= \mathbf{5}$ A file with load time series of a cross section of the tower.
This file has the same format as for '**output_type 3**' except that on the sixth record '-1' is written instead of the blade number and that the distance above the tower base is written instead of the spanwise location.
The seventh line with the heading of the table looks like:

```
time  it azimuth  -22    23    21   -25   -26    24
```

GL_names `LOGICAL` (OFF)
If 'ON' the output file contains a line with variable names instead of numbers in the heading of '**output_table**', see Figure 33. This is also available for **output_type 0** and **2**.

first_time `REAL` (0.0) [s]
Time from which (and including) the time series are written, as far as available.

last_time `REAL` (1000000.0) [s]
Time until which (and including) the time series are written, as far as available.

time_reset `LOGICAL` (OFF)
If 'ON' the output property for time is expressed with respect to '**first_time**'.

table_increment `INTEGER` (1)
Increment by which the data file contents are to be read and processed.

output_table `table(` variable nr. , location`)`
$$\begin{pmatrix} 151 & 0.0, \\ 152 & 0.0, \\ -153 & 0.0, \\ 154 & 0.0, \\ 155 & 0.0, \\ 156 & 0.0, \\ 999 & 0.0, \\ ... & ... \end{pmatrix}$$

Table with specifications of maximum 128 output properties.
Each record contains the specification of one variable with the numbers:

- `INTEGER` Property number as used for PHATAS.
  This does not include the aerodynamic blade properties 111 to 119, see chapter 4.
- `REAL` Coordinates [m] of the output properties.
  These are valid if they refer to loads in one of the main components:

  Blade numbers 112 to 179, 212 to 279, 312 to 379 etc.:
  Spanwise coordinate [m] measured from the blade root '**blade_root_radius**';

  Rotor shaft or **drive train** numbers 51 to 71:
  Location [m] in the rotor shaft measured downwind from the rotor centre;

  Tower numbers 21 to 49: Height [m] above the tower base.

Example

Figure 32 contains a listing of the postprocessor input file for selection of extreme values of the normal production load case of a 2.7MW 90.5m diameter turbine at 15m/s turbulent wind.

The postprocessor output using the input file in Figure 32 is listed in Figure 33.

```
<  Input file for the postprocessor  'phpost'
<  for retrieving the average and extreme values of
<  some operational properties and structural loads.
<
MENU  specification
datafile_name  .\phdata\015.tbh  < Name of the PHATAS binary data file.
output_file    015_ex.txt   < Output file name.
comment_mark     '#'        < Character in column 1 of the heading lines.
GL_names         ON         < Names in the heading instead of numbers.
output_type       2         < Extreme values of the properties.
first_time       30.0       < The first 30 seconds are skipped.
last_time     99000.0       < The data file is read until 99000s.
table_increment   1    < Each time step from 'first_time' to 'last_time'.
<
output_table   <  Following is the selection of the output properties.
   12   0.0,              <  Rotor rotational speed.
    5   0.0,              <  Generator power [W] on the rotor.
   64   0.0,              <  Axial force in the rotor shaft [N].
 -143   0.0,              <  Flap bending moment [Nm] in blade 1.
  141   0.0,              <  Lead bending moment [Nm] in blade 1.
   22  15.0,              <  Tower tilt moment [Nm] 15m above the base.
  999   0.0               <  End of the specified properties
<
```

Figure 32: Input file for '*phpost*' for selecting extreme values

```
# PHATAS-4
# Postprocessor output from datafile: .\phdata\015.tbh
# Model_identif: turb91m
# Load_case: NORM_PROD_15mps
# Type    Omega      P_elec      Fx_shaft    Mflap1_R    Mlead1_R    My_tow_15
 MIN    1.5391E+01  1.9910E+06  1.6141E+05  2.5422E+05 -1.4514E+06  8.0676E+06
 AVE    1.6311E+01  2.6469E+06  2.7896E+05  2.1061E+06  5.1182E+05  1.9641E+07
 MAX    1.7262E+01  2.7006E+06  3.8900E+05  3.8085E+06  2.4408E+06  2.9908E+07
```

Figure 33: Output from '*phpost*' with extreme values of the properties

D SELECTION OF DESIGN LOADS '*LOADEX*'

Introduction

The design of a wind turbine requires the calculation of a large number of load cases, each resulting in dynamic load time series. Selecting the extreme loads in the main components of a wind turbine (blade, shaft, tower) and retrieving the loads for fatigue-design of bearings requires investigation of all these load time series.

For the selection of extreme loads in a wind turbine component from the set of dynamic load time series, the program '*loadex* ' has been developed. Initially '*loadex* ' was a tool for selecting extreme load components and was later extended with options to retrieve extreme bending moments as function of load-direction, and to perform 'Time-At-Level' counting of the loads on a bearing and on the yaw or pitch drive. The program *loadex* uses the same routines as the postprocessor for opening the PHATAS datafiles and for reading the properties during a given time span and for a given location in the tower, shaft, or blade.

D.1 Input

The program '*loadex*' is invoked by typing its name followed by up to three `CHARACTER*64` command line arguments for the names of:

1 '*loadex*' input file with specification of the blade cross section ('loadex.in');

2 File with a table of all load cases, the so-called 'occurrences' file. ('loadset.occ').

3 File for output tables of extreme values of the sectional loads, and optionally from Time-At-Level counting. If this name is not specified the results are written to the screen.

D.1.1 Input specifications for *loadex*

Figure 34 shows an input file for *loadex* of which most specifications have the default values. Following is a description of the input items for *loadex*:

FILE `CHARACTER*64`
Name of an additional input file. The input data in (an) additional input file(s) are read as if this file is included at the location of its specification.

part `CHARACTER*20` ('*blades*') <'*blades*', '*shaft*', '*tower*'>
This character variable indicates the turbine main component (tower, shaft or blade).
If **part** contains the characters '*bl*' or '*BL*'
the extreme loads in the cross section of the selected rotor *blades* are searched.
If **part** contains the characters '*sh*' or '*SH*'
the extreme loads in the rotor *shaft* cross section are searched.
If **part** contains the characters '*to*', '*TO*', '*tw*', or '*TW*'
the extreme loads in the *tower* cross section are searched.
For other character combinations *loadex* stops with a message to the screen.

blade1 `LOGICAL` (OFF) <ON,OFF>
If 'ON' and **part** is '*blades*' the loads in blade 1 are included in searching the extremes.

blade2 `LOGICAL` (OFF) <ON,OFF> Idem for blade 2, if the rotor has at least 2 blades.

blade3 `LOGICAL` (OFF) <ON,OFF> Idem for blade 3, if the rotor has 3 blades.

location `REAL` [m] (0.0) <0 ....>
If **part** is '*tower*': height above the tower base.
If **part** is '*shaft*': distance behind rotor centre along the shaft.
If **part** is '*blades*': spanwise location measured from the blade root.

rotate `LOGICAL` (ON) <ON,OFF>
If 'ON' the loads are expressed in a reference system that is yawing with the nacelle (**part** = '*tower*'), rotating with the rotor (**part** is '*shaft*'), or pitching (**part** is '*blades*').

twist `REAL` [^{o}] Only if **part** is '*blades*': The extreme loads are evaluated in a reference system that has an angle '**twist**' with respect to the rotor plane, for zero pitch.
If '**twist**' is not specified the reference system has the twist of the blade chord at span '**location**' as is present in the PHATAS model.

detfactor `REAL` [.] (1.0)
Factor where the *det*erministic part of the loads are multiplied with. The deterministic part of the loads are the loads from gravity and from centrifugal effects, evaluated with the geometry for a rigid rotor. In general **detfator** has the value 1.0.

stofactor `REAL` [.] (1.0)
Factor where the *sto*chastic part of the loads are multiplied with. The stochastic part of the loads are obtained by subtracting the deterministic part of the loads from the total loads. In the binary datafiles of PHATAS only the stochastic part of the blade loads are written. Different factors for the deterministic part and for the stochastic part of the blade loads may be applied when using an approach similar to that in section 7.6.2 of the IEC 61400-1 Safety requirements, [8] or section 4.3.5 of the G.L. Richtlinie [5].
For hinged blades or a teetered hub the blade root moment is inherently very small. For this case it makes no sense to use different factors for deterministic- and stochastic loads.

average `LOGICAL` (OFF) <ON,OFF>
If 'ON' the average loads are included in the output table.
Realise that the output file for a complete load set may be long, in which case one may prefer not to write the average loads.

sectors `INTEGER` (0) <0, ..., 180>
If larger than zero, the extreme bending moments are calculated in '**sectors**' directions of the cross section, and written in the output file. Together with these extreme moments per sector, also the corresponding tensile force is listed, giving a close approximation of the extreme loads for e.g. buckling of the cross section.

generator_mass `INTEGER` (0) <0, ... >
The mass of the generator-rotor, viz the parts that are attached to the shaft.
The weight loads of this mass are added to the rotor shaft loads, which applies to the direct-drive wind turbines with a single shaft bearing. See also **gen_mass_centre**.

gen_mass_centre `INTEGER` (0)
Position along the rotor shaft of the generator mass **generator_mass**, measured downwind from the rotor centre. This position together with **generator_mass** are used to calculate the moments on the shaft support of single-bearing direct-drive configurations. In this case **location** should be the position of the main shaft bearing, measured downwind from the rotor centre.

bend_tal_levels `INTEGER` (0) <0, ..., 64>

Number of 'levels' for Time-At-Level counting on basis of the Number of 'levels' for Time-At-Level counting based on the resulting bending moment on the section (bearing). The result of this Time-At-Level counting is used for the design of the bearings, such as pitch bearing, yaw bearing, and rotor shaft main bearing.

tors_tal_levels `INTEGER` (0) <0, ..., 64>

Number of 'levels' for Time-At-Level counting based on the torsional moment from the (yaw or pitch) drive to the structure. The result of this Time-At-Level counting is used for the design of the yaw drive or pitch drive.

IMO_friction `LOGICAL` (OFF) <ON,OFF>

If 'ON' (and '**tors_tal_levels**' > 1) the bearing friction is added to the torsional moment that is used for Time-At-Level counting for the pitch- and yaw-drive. This friction is calculated with the expression for bearings of IMO:

$$M_{\text{tors.fric}} = M_{\text{tors.stat}} + 0.003\,(4.4\,M_{\text{res}} + D_{\text{bear}}(4\,F_{\text{res}} + |F_{\text{tens}}|))\,.$$

Here M_{res} and F_{res} are the resultant bending moment and -shear force respectively.

stat_friction `REAL` [Nm] (0.0)

Static friction $M_{\text{tors.stat}}$ (torsional moment) for an unloaded bearing.
This is only used for '**IMO_friction ON**' and '**tors_tal_levels**' > 1.

bearing_diameter `REAL` [m] (0.0)

Diameter D_{bear} used in the expression for bearing friction on the torsional moment.
This is only used for '**IMO_friction ON**' and '**tors_tal_levels**' > 1.

D.1.2 File with occurrences of load cases

The collection of load cases that have to be included in the selection of extreme (and fatigue) loads are read by *loadex* from the so-called 'occurrences' file, which is read as second command line argument. (This name was used because this file also contains the number of 'occurrences' of each load case in the fatigue spectrum.)

After a number of comment lines, starting with '`#`' or '`<`' or '`/`', this 'occurrences' file contains a table with on each record the specifications of one file by the properties:

- `INTEGER` Load case number.
- `REAL` Number of occurrences in the load set.
  If this number is negative the dynamic loads in this file are skipped in the selection of extreme loads. If *loadex'* is used for Time-At-Level counting, the contribution of each load case is multiplied with this 'number of occurrences' (if positive). This means that a load case is excluded if the number of occurrences is zero or negative.
- `REAL` First time [s] from which the blade loads must be read from file;
- `REAL` Last time [s] until which the blade loads must be read from file.
  If the dynamic behaviour in the PHATAS file is shorter than 'last time' the loads are read as far as available;
- `REAL` So-called 'Partial Safety Factor' with which the loads from the governing load case are to be multiplied. This is done for the selection of extreme loads and also for the 'Time-At-Level' counting.

This occurrences file is also generated when using the program '*lcprep*' as `lcprep.occ` , see Appendix B.

D.2 Output

The output of running loadex contains a table with extreme load components in the cross section that is selected, and depending on the input specifications also some other tables. In this section the tables in the '*loadex*' output file are described for the load-set of a 2.7MW 90.5m diameter wind turbine, using the input file of Figure 34. Parts of the resulting output of *loadex*, for the load-set of a 2.7MW 90.5m diameter wind turbine, using the input file of Figure 34, are listed in Figure 35.

D.2.1 Extremes of all load cases

The blade loads in the columns of the table in Figure 35 are defined as follows (considering zero twist and a non-pitching reference system):

Fflat	Flat wise force [N] perpendicular to the coned rotor plane, positive for a downwind tip force;
Fedge	Edge wise force [N], positive for a tip load towards the trailing edge;
Fspan	Span wise force [N], positive in tension;
Medge	Lead wise moment [Nm], positive for normal power production;
Mflat	Flap wise moment [Nm], positive when the blade is bent downwind;
Mtors	Torsional moment [Nm], nose-up positive (= towards stall);
Fres	Resultant vector [N] of the shear force components;
Fi_Fres	Angle [°] of resultant shear force vector w.r.t the reference system (=rotor plane), calculated as `atan2(Flag,Fflap)`;
Mres	Resultant vector [Nm] of the bending moment components;
Fi_Mres	Angle [°] of resultant bending moment vector w.r.t. the reference system (=rotor plane), calculated as `atan2(Mflat,Mlead)`.

For the rotor shaft the extreme load components in the non-rotating reference system are:

Fside	Side force [N], positive for an in-plane rotor load to the right looking downwind;
Fvert	Vertical shear force [N], positive for an upward load;
Ftens	Tensile force [N], negative for thrust during normal operation;
-Mtilt	Tilt moment [Nm], positive for a downwind load on the lower half of the rotor;
Myaw	Yaw moment [Nm], positive around the upward vertical axis;
-Qshaft	Shaft torsion [Nm], negative for normal power production;
Fres	Resultant vector [N] of the shear force components;
Fi_Fres	Angle [°] of resultant force vector, calculated as `atan2(Fvert,Fside)`;
Mres	Resultant vector [Nm] of the bending moment components;
Fi_Mres	Angle [°] of resultant moment vector w.r.t the vertical, calculated as `atan2(Myaw,-Mtilt)`.

For the extreme shaft loads in the rotating reference system the load components are identical to those in the non-rotating reference system if the rotor azimuth is zero, for which blade 1 is pointing to the right horizon when looking down wind.

For the tower the extreme load components in the non-yawing reference system are:

Fax	Downwind shear force [N];
Fside	Side force [N], positive for an in-plane rotor load to the right looking downwind;
Ftens	Tensile force [N], negative for compressive load (weight);
Mside	Side moment [Nm], positive for a positive shaft torque;
Mtilt	Tilt moment [Nm], pos. for a downwind load on the rotor;
Myaw	Yaw moment [N], positive about the vertical (up) axis;
Fres	Resultant vector [N] of the shear forces;
Fi_Fres	Angle [^{o}] of resultant shear force vector, calculated as `atan2(Fside,Fax)`;
Mres	Resultant vector [Nm] of the bending moments;
Fi_Mres	Angle [^{o}] of resultant bending moment vector, calculated as `atan2(Mtilt,Mside)`.

D.2.2 Load components for each extreme

Also a table is written that contains for each extreme (minimum and maximum) value of the load component, also the other load components at that time. These loads are listed without load factor. The load case and load-factor specified for that load case are included in the table. The format of this table is similar to that described in table 4.B.1 of the G.L. Rules and Recommendations [5].

D.2.3 Extreme moment as function of direction

If the specification of **sectors** is larger than 0 the resulting bending moments are selected for '**sectors**' load directions. The loads are included in the extreme selection for a 'sector' if:

1 The resultant moment vector 'falls' within a sector;

2 The 'moment time trace' since the previous solution passes sector boundaries.

For the blades, the resultant moment is traced for each of the blades specified in the input.

The 'direction' listed in the table is the median of each sector. In addition to the extreme resultant moment, the table also contains the other loads for the time at which the extreme occurs.

The intention of the 'extreme moments per sector' table is that these moments may be applied in e.g. a Finite-Element analysis of the blade that are representative for the extreme loads in the structure.

D.2.4 Time-At-Level tables for bearing loads

If the specification of **bend_tal_levels** is larger than zero the Time-At-Level counting is performed using the resultant moment vector of a cross section. In the program *loadex'* this moment is for the cross section specified in the input. This means that if the results of the Time-At-Level counting are used for fatigue calculations of the bearings, the user has to specify the correct bearing location. For the pitch bearing this radius may be smaller than the blade root.

The Time-At-Level counting is a sorting procedure not only based on the loads (here resultant bending moment) but also depending on whether the bearing rotates in positive direction, negative direction or stands still. Likewise the output of *loadex'* contains separate tables for the Positive, Negative, and 'Zero' motion and finally a table independent of motion. For each load case, the contribution to the 'Time-At-Level' counting is multiplied with the number of times as specified in the 'occurrences' file. This means that a load case with a 'zero' time in the occurrences file is not included (not even read) for the Time-At-Level counting while it is included in the selection of extreme loads. A load case with negative time in the occurrences file is not included in any of the processes.

D.2.5 Time-At-Level tables for drive loads

If the specification of **tors_tal_levels** is larger than zero the Time-At-Level counting is performed based on the moment about the bearing axis, such as the yaw drive of the pitch drive.

```
#  Input file for 'loadex', which reads PHATAS datafiles and performs:
#    Extreme load selection in the blades, shaft, or tower.
/    Time-At-Level counting for bearing and/or drive loads.
/
/  Comment lines are indicated with '/' '#' or '<'
<
part       'blade' /  The substring 'bl' indicates  extreme blade loads.
#
blade1       ON    <  Include loads in blade 1
blade2       ON    <  Not include loads in blade 2
blade3       ON    // Do not include loads in blade 3
#
rotate       OFF   <  If 'ON' rotate blades for pitch (c.q. rotor azimuth)
                   <  Note that the default is 'ON' (or .TRUE. or 1)
twist        0.0   <  [deg] In the 'zero twist' directions.
<
location    -0.126 /  At the pitch bearing (slightly inside blade root).
<
<  The default refers to the twist of the PHATAS model at span 'location'.
<
detfactor    1.0   #  Multiply the deterministic loads with 1.0
stofactor    1.0   #/ Multiply the stochastic loads with 1.0
<
average      OFF   <  If 'ON' write also averages of the loads.
/
sectors      18    <  List the extreme moments in 20deg direction intervals.
/
bend_tal_levels   0  < Levels for Time-At-Level counting of bearing loads.
tors_tal_levels   0  < Levels for Time-At-Level counting of drive loads.
IMO_friction    OFF  < If 'ON' add friction for the drive loads TAL counting.
stat_friction 1000.0 < [Nm] Friction (torsional moment) for unloaded bearing.
bearing_diameter 1.0 < [m] Diameter of bearing used for bearing friction.
<
```

Figure 34: Input file for '*loadex*'

```
  Output of the program 'loadex' for model: turb91m
  for the load set specified in file: loadset.occ
  Extreme blade loads at span-wise location    0.000 m
  For twist angle   0.000 deg  Non-pitching system
   Case        Fflap      Flead      Fspan      Mlead       Mflap      Mtors    Fres Fi_Fr    Mres Fi_Mr
 1210 MIN      18993    -121362      34168   -1313652      490400     -23292
 1210 MAX     111218      97675     527799    1873004     2771608      10523   161903 -47   3331849  56
 1240 MIN      39974    -121026     279338    -941448      515638     -28043
 1240 MAX     117952      83768     537545    1896909     2930515      -2001   166280 -46   3463182  57
 1101 MIN      31938    -136962     233914   -1430996      976975     -26761
 1101 MAX     146770     103784     549902    2421981     3636953      -1437   192253 -40   4142255  61


 8210 MIN     -33905    -166086     -95983   -2930912     -450172    -101553
 8210 MAX      34056     161961     106793    3039618      384292     101620   167513 -97   3057008  -6
 Overall extreme loads.
MINIMUM 1    -694386   -1121384    -182285   -8155773   -19388088    -464023
 Case         7120       2241       6510       2211        7120       7120
 Factor       1.100      1.100      1.350      1.100       1.100      1.100
MINIMUM 2    -222947   -1120934    -171356   -8072191    -5801248    -189030
 Case         1682       2211       6520       2241        1682       6210
 Factor       1.350      1.100      1.350      1.100       1.350      1.100
MINIMUM 3    -201387   -1117671    -163089   -7755627    -5543222    -184816
 Case         1681       2212       7120       2272        1672       6110
 Factor       1.350      1.100      1.100      1.100       1.350      1.350
MAXIMUM 1     680509    1104987    1227485    8423786    18405144     857591  1126059 -85  19479632 -84
 Case         7120       2211       1671       2212        7120       7120     2241          7120
 Factor       1.100      1.100      1.350      1.100       1.100      1.100    1.100         1.100
MAXIMUM 2     237637    1099115    1225291    8311575     5754419     204141  1125357 -85   8836282  18
 Case         2370       2241       1672       2242        1342       6210     2211          2212
 Factor       1.350      1.100      1.350      1.100       1.350      1.100    1.100         1.100
MAXIMUM 3     227791    1087231    1207225    8186063     5694463     146244  1122344 -85   8802048  19
 Case         1342       2272       1682       2241        1312       6110     2212          2242
 Factor       1.350      1.100      1.350      1.100       1.350      1.350    1.100         1.100
ABS. MAX.     694386    1121384    1227485    8423786    19388088     857591
 Case         7120       2241       1671       2212        7120       7120
 Factor       1.100      1.100      1.350      1.100       1.100      1.100

 Load components including load factor for each extreme value.
        Case    Factor   Fx(flap)   Fy(lead)   Fz(span)    Mx(lead)    My(flap)  Mz(tors)  dPitch/dt
 F_x min 7120    1.1000    -694386     -59855    -156724     1886287   -19388088   -201308     0.000
 F_x max 7120    1.1000     680509     143572     125892    -1022131    17594621   -379720     0.000
 F_y min 2241    1.1000     102507   -1121384     416217     6515374     2749626    -59543     1.866
 F_y max 2211    1.1000      89197    1104987     320844    -6800569     2338160     10057     2.326
 F_z min 6510    1.3500     -28061    -120173    -182285     2223836     -273874   -104538     0.000
 F_z max 1671    1.3500     -32751     -28236    1227485      -48634    -1914874     15053     4.000
 F_r max 2241    1.1000     102507   -1121384     416217     6515374     2749626    -59543     1.866
 M_x min 2211    1.1000     113602     858108     505258    -8155773     2607162    -17623     2.408
 M_x max 2212    1.1000     103896    -930163     264587     8423786     2668279    -29094     1.965
 M_y min 7120    1.1000    -694386     -59855    -156724     1886287   -19388088   -201308     0.000
 M_y max 7120    1.1000     668885    -203850     -68690     3935864    18405143   -376019     0.000
 M_z min 7120    1.1000     614879     -76144     -40382     1385894    16248956   -464023     0.000
 M_z max 7120    1.1000    -577960     118120      75199    -3765194   -16101488    857591     0.000
 M_r max 7120    1.1000    -694386     -59855    -156724     1886287   -19388088   -201308     0.000

 Directions:  0deg = Medge_lead  90deg = Mflat_downwind  180deg = Medge_lag  270deg = Mflat_upwind.

 Sector  Extreme mom     Mx           My        Mtorsion      Ftens      Case    Factor
   0.0  6.8740E+06  6.8731E+06  1.0929E+05 -6.9633E+03  3.5830E+05  2271    1.1000
  20.0  8.8020E+06  8.3116E+06  2.8972E+06 -3.2433E+04  2.8138E+05  2242    1.1000
  40.0  6.7533E+06  3.8678E+06  4.8326E+06 -1.2971E+04 -7.2560E+04  7120    1.1000
  60.0  9.3127E+06  3.9632E+06  8.2859E+06 -3.3712E+04 -6.8997E+04  7120    1.1000
  80.0  1.8346E+07  3.2327E+06  1.8059E+07 -4.0607E+05 -5.1561E+04  7120    1.1000
 100.0  1.2358E+07 -1.9087E+06  1.2207E+07 -2.9718E+05  1.1222E+05  7120    1.1000


 320.0  6.4136E+06  3.4633E+06 -4.5975E+06 -6.6946E+04 -9.4378E+04  7120    1.1000
 340.0  5.5027E+06  3.5648E+06 -2.3268E+06 -4.2467E+04 -8.7251E+04  7120    1.1000
```

Figure 35: Extreme blade loads as calculated with ‘*loadex*’

E ROTOR-AZIMUTH BIN ANALYSIS

E.1 Introduction

Most of the dynamic loads in a wind turbine are partly dependent on the rotor azimuth, especially the loads in the rotor blades. In some research applications the rotor-azimuth related part of the dynamic loads are investigated. This is done by sorting the load time series based on the rotor azimuth after which the averages are calculated. For this process, the program '*bin*' has been developed.

In some wind turbine research projects it was noticed that when properties are "binned" against the rotor azimuth following the basic definition, the resulting bin- averages may not form a smooth series. This was caused by the fact that the azimuth-interval of the data series and the bin- interval do not always correspond and may result in some irregularities, with the nature of interference. Assuming that either the calculated- or the measured time series of the wind turbine dynamics are a 'sampling' of the continuous response, the bin- average can be calculated in which each bin- interval contains one data value of each revolution. Using this method the resulting average in each bin is the average from sampled data of all revolutions.

Based on this last method of "re-sampling" of the linearised data series a program '*bin*' is developed. The program *bin* is invoked by typing its name followed by a maximum of four `CHARACTER*64` arguments for the names of the files with:

1 Input specifications for *bin* ('bin.in');
2 Input data series containing records with data at subsequent times ('phatout');
3 Results of the bin- averaging ('bin.out');
4 Data series from which the bin- averages are subtracted based on the rotor azimuth. If no name is specified for this file, it will not be created.

If a file name is not specified as argument the name between brackets is used. Note that if an argument is specified that all preceding command line arguments also need to be present.

Following is description of the input items and examples of input and output files.

E.2 Input of 'bin'

An example of the file with specifications for *bin* is given in Figure 36, containing the specifications of the default values. Following are the input items for *bin*:

FILE `CHARACTER*64`
Name of an additional input file. The input data in (an) additional input file(s) are read as if this file is included at the position of its specification.

input_file `CHARACTER*64` ('phatdef.pht')
Name of the file with input time series. If two command line arguments are given at the call for *bin* the second argument overrides this **input_file** specification.

output_file `CHARACTER*64` ('bin.out')
Name of the file to which the bin- averages are written. If three command line arguments are given at the call for *bin* the third argument is read as name of the 'output_file'.

cor_file `CHARACTER*64` ('none')
Name of the file to which the data series with the bin-averages subtracted are written. These properties are sometimes denoted with "stochastic parts".
If this name is not specified or 'none' such file will not be created.

output_type `INTEGER` (0) <0, 1, 2>

Indicates how the bin- averaged properties are written as output:

= 0 The output file contains one table with records for each bin.
Each record contains the rotor azimuth followed by the bin- averages of each property specified in '**output_table**'.

= 1 The output file contains for each output value one table.
These tables contain on each record the properties for one bin:
- rotor azimuth [°] in the centre of the bin;
- maximum of the data values;
- average plus the standard deviation in the bin;
- average of the data values in the bin;
- average minus the standard deviation in the bin;
- minimum of the data values in the bin.

= 2 The output file contains for each output value one table.
These tables contain on each record the properties for one bin:
- rotor azimuth [°] in the centre of the bin;
- average value in the bin;
- maximum value in the bin;
- minimum value in the bin;
- standard deviation in the bin.

heading `LOGICAL` (ON) <ON,OFF>

If 'ON' a heading is written in the file with binned properties and (if present) in the file with "stochastic parts" of the properties, see **cor_file**.

comment_mark `CHARACTER*1` (' ')

Character that is written in the first column of the heading lines.

nr_bins `INTEGER` (60) <2, ... , 360>

The number of bin intervals over one revolution.

centered `LOGICAL` (ON) <ON,OFF>

If 'ON' the properties are 're-sampled' and averaged at the centre of the bin-intervals. IF 'OFF' the properties are 're-sampled' and averaged at the edges of the bin intervals.

azimuth_column `INTEGER` (3)

Number of the column in the input time series that contains the rotor azimuth.

azimuth_offset `REAL` (0.0)

Offset that is added to the rotor azimuth as it is read.

azimuth_factor `REAL` (1.0)

Multiplication factor for the rotor azimuth in order to give it the unit [°].
The azimuth that is used for bin-analysis is:

$$\{\textbf{azimuth\_factor}\} \cdot (\{\text{value in } \textbf{azimuth\_column}\} + \{\textbf{azimuth\_offset}\})$$

In the PHATAS output files the rotor azimuth is written in column 3 while it does not need to be scaled (factor 1.0) since it is already expressed in degrees.

time_column `INTEGER` (1)

Number of the column that contains the time. This is only used when writing a file with the "stochastic parts" of the properties, see **cor_file**.

first_record `INTEGER` (8) <1, ... >
Number of the first record in the input time series that has to be processed.
The PHATAS output files contain a heading of 7 records.

last_record `INTEGER` (1000000)
Number of the last data record that has to be processed.
If less than **last_record** data records are present the file with time series is processed as far as readable.

output_table The keyname '**output_table**' must be followed by a table with on each record (maximum 60) the specification of one property to be binned.
Each record must contain the following three properties:

`INTEGER`	Column number in the input time series of the property;
`REAL`	Offset that is added to the property in that column;
`REAL`	Multiplication factor for the property in that column.

The files with input time series must apply to the following requirements:

1 The records in the input time series must be written for increasing time. (the time intervals between the records do not need to be constant!).
2 The rotor azimuth must be continuously increasing, in the sense that it always describes a positive rotational speed.
3 The azimuth of two subsequent data records must differ less than 180°.
4 The input time series must contain at least one full revolution.

E.3 Output of 'bin'

Output files may contain an optional heading to be specified by the user. This heading contains some statistical properties of the entire data series and the Fourier components of the bin- averages. By allowing a "comment mark" in the first column of the heading the output files can be read in a plotting package.
Figure 37 contains an output file that is generated with the example input specifications listed in Figure 36 and the time series calculated for a 2.7MW wind turbine operating at 15m/s turbulent wind. For the three formats available a plot is made and displayed in Figure 38 to 40 (using 60 bins).

```
< Input file for the program ’bin’ with which a "BIN−analysis"
< can be performed of properties from a file with time series.
< The input specifications listed below are the default settings
/ except for ’table’ which specifies the properties to be analysed.
< Comments should be preceded by either  /  <  or  #
#
input_file     phatdef.pht < Name of the file with input time series
/
output_file    bin.out <  Name of the file to which output is written
/
cor_file       none    <  A value ’none’ indicates that no file
                       <  with stochastic parts is written.
/
output_type    1       <  For each property that is processed a
                       <  table is written with columns containing:
             <  averaged rotor azimuth [deg]
             <  maximum of the values in the bin
             <  average + standard deviation of the values in the bin
             <  average of the values in the bin
             <  average − standard deviation of the values in the bin
             <  minimum of the values in the bin
output_type    2       <   For each property that is processed a
                       < table is written with columns containing:
             <  averaged rotor azimuth [deg]
             <  average of the values in the bin
             <  minimum of the values in the bin
             <  maximum of the values in the bin
             <  standard deviation of the values in the bin
output_type    0       <  The output of ’bin’ contains one table
                       <  with the rotor azimuth−averages of
                       <  each property selected under ‘table’.
#  ’output_type 0’ is active because this is the last specification.
#
heading         ON     <  A heading is written in the output of ’bin’
comment_mark   ’#’    <  A ‘#’ (tuinhekje) is written in column 1 of heading.
#
nr_bins         60     <  One revolution is split in 60 bins.
centered        OFF    <  The ’bin’−analyses on the edges of the intervals.
first_record   208     <  Skip the heading and the first 200 records = 10s.
                       <  (the PHATAS heading has 7 records).
last_record   1000000  <  The default for the last records is large.
azimuth_column  3      <  The numbers in column 3 are read as azimuth
azimuth_offset  0.0    <  Number with which the azimuth is increased.
azimuth_factor  1.0    <  Multiplication factor to get an azimuth in [deg].
/
/  The columns with properties to be analysed should be listed
/  in a table after the specification of ’table’.
/  Note that this should be included at the end of the input file.
output_table
  4  0.0  1.E−6,  < Col 4 (flap moment) with offset 0.0 and converted to MNm
  5  0.0  1.E−6,  < Col 5 (lead moment) with offset 0.0 and converted to MNm
  6  0.0  1.0,    < Col 6 (pitch rate) with offset 0.0 and factor 1.0
  9  0.0  1.E−6,  < Col 9 (shaft moment) with offset 0.0 and converted to MNm
 10  0.0  1.E−6  < Col 10 (shaft moment) with offset 0.0 and converted to MNm
<
```

Figure 36: Example of input file of the program *bin* (output type 0)

```
# Rotor−azimuth bin analysis of data from file:
#    015_bin.txt
# Number of complete revolutions analysed:  167
#
#Column:      4           5           6           9          10
#Average  2.1020E+00  5.1158E−01  1.0472E−02  3.5824E−03  1.0006E−01
#Minimum  2.5422E−01 −1.4514E+00 −2.6930E+00 −1.8945E+00 −2.4550E+00
#Maximum  3.8085E+00  2.4408E+00  2.8444E+00  2.4846E+00  1.9421E+00
#Std.dev  5.4237E−01  9.7573E−01  8.9715E−01  5.5043E−01  5.4324E−01
# Fourier components
#1P cos:  2.6789E−01  7.4597E−02  2.3093E−03 −1.1305E−01  2.0801E−01
#1P sin:  1.4059E−01  1.3502E+00  3.7590E−02  2.2035E−01  7.5984E−02
#2P cos: −4.3721E−02 −1.4878E−02  1.5786E−02 −6.5857E−02  3.4317E−02
#2P sin: −2.5597E−02 −1.0332E−02 −5.6129E−03 −3.7656E−02 −5.0308E−02
#3P cos:  1.9298E−02  3.5318E−04 −3.4236E−02  4.1308E−03  3.1435E−03
#3P sin:  1.7602E−02 −2.9851E−03 −9.0488E−03 −1.0275E−02 −1.6276E−03
#4P cos:  1.1959E−03 −1.4486E−03 −5.4409E−06 −8.8355E−04 −1.6823E−02
#4P sin: −1.3645E−02  8.5206E−06 −8.8325E−03 −2.1120E−02  2.3659E−04
#5P cos: −6.8731E−03  1.6856E−02 −3.0632E−04 −1.0643E−02 −1.7578E−02
#5P sin:  1.0804E−02 −4.8869E−03  1.6809E−03  1.5754E−02 −4.1898E−03
#6P cos: −5.5341E−03 −2.4877E−04 −3.3318E−03 −1.5530E−03  4.3960E−03
#6P sin: −9.2914E−03 −1.2694E−03  3.6543E−03 −1.1881E−03 −1.2462E−03
#    5   61
#Azimuth      4           5           6           9          10
   0.00   2.3277E+00  5.8565E−01 −9.2692E−03 −1.9549E−01  3.3313E−01
   6.00   2.3455E+00  7.2047E−01 −6.8027E−03 −1.6293E−01  3.3209E−01
  12.00   2.3638E+00  8.4806E−01 −1.5110E−03 −1.3344E−01  3.2267E−01
  18.00   2.3733E+00  9.7254E−01  7.9144E−03 −1.1517E−01  3.0974E−01
  24.00   2.3780E+00  1.0949E+00  1.6037E−02 −9.2715E−02  2.9547E−01
  30.00   2.3835E+00  1.2157E+00  2.8150E−02 −5.6906E−02  2.7983E−01
  36.00   2.3821E+00  1.3323E+00  3.0646E−02 −2.1833E−02  2.7643E−01
  42.00   2.3664E+00  1.4458E+00  3.2733E−02  5.0709E−03  2.8328E−01
  48.00   2.3507E+00  1.5497E+00  4.2283E−02  2.9999E−02  2.9308E−01
  54.00   2.3456E+00  1.6431E+00  6.0842E−02  7.2700E−02  2.8569E−01
  60.00   2.3391E+00  1.7262E+00  7.5897E−02  1.3425E−01  2.3501E−01
  66.00   2.3272E+00  1.7963E+00  7.7996E−02  1.9508E−01  1.6910E−01
  72.00   2.3112E+00  1.8440E+00  7.1939E−02  2.4027E−01  1.2427E−01
  78.00   2.3020E+00  1.8705E+00  6.4686E−02  2.7056E−01  1.1186E−01
  84.00   2.2940E+00  1.8798E+00  6.2536E−02  2.8844E−01  1.2042E−01
  90.00   2.2856E+00  1.8721E+00  5.8989E−02  3.0052E−01  1.1774E−01
  96.00   2.2665E+00  1.8486E+00  3.7236E−02  3.1063E−01  1.2743E−01
 102.00   2.2358E+00  1.8138E+00 −2.5233E−03  3.1730E−01  1.4629E−01
 108.00   2.2063E+00  1.7729E+00 −1.9794E−02  3.2188E−01  1.4713E−01
 114.00   2.1716E+00  1.7244E+00 −6.1372E−03  3.1861E−01  1.2577E−01
 120.00   2.1342E+00  1.6606E+00  4.4322E−03  3.1100E−01  1.0447E−01
    .
    .
    .
 330.00   2.2625E+00 −1.0653E−01  4.7618E−02 −1.7831E−01  3.1326E−01
 336.00   2.2908E+00  2.8644E−02  1.7984E−02 −1.7850E−01  3.1583E−01
 342.00   2.3107E+00  1.6770E−01 −3.3417E−03 −1.8026E−01  3.1792E−01
 348.00   2.3120E+00  3.0495E−01 −8.9196E−03 −1.9819E−01  3.1829E−01
 354.00   2.3128E+00  4.4499E−01 −1.0648E−02 −2.1020E−01  3.2429E−01
 360.00   2.3277E+00  5.8565E−01 −9.2692E−03 −1.9549E−01  3.3313E−01
```

Figure 37: Example of output file of the program *bin* (output type 0)

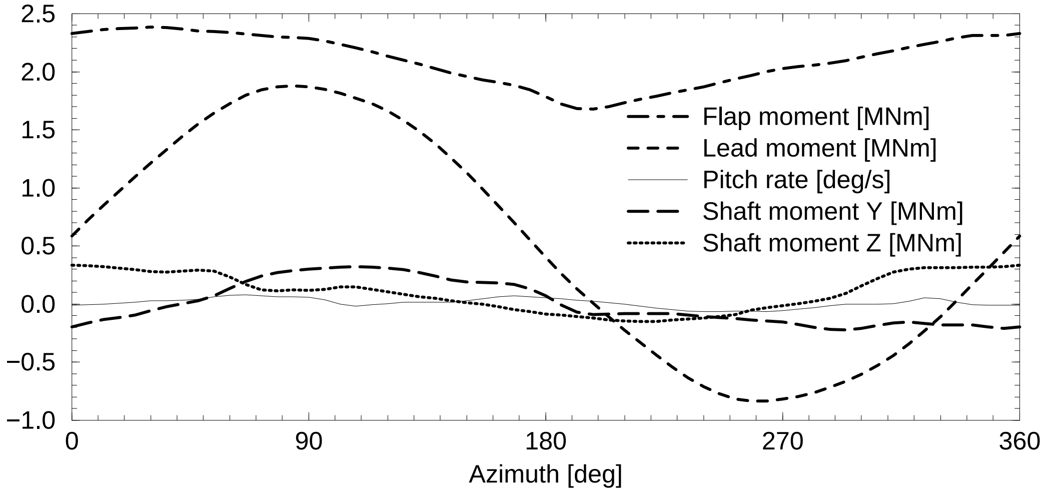


Figure 38: Plot of bin- averaged properties from *bin* for output type 0

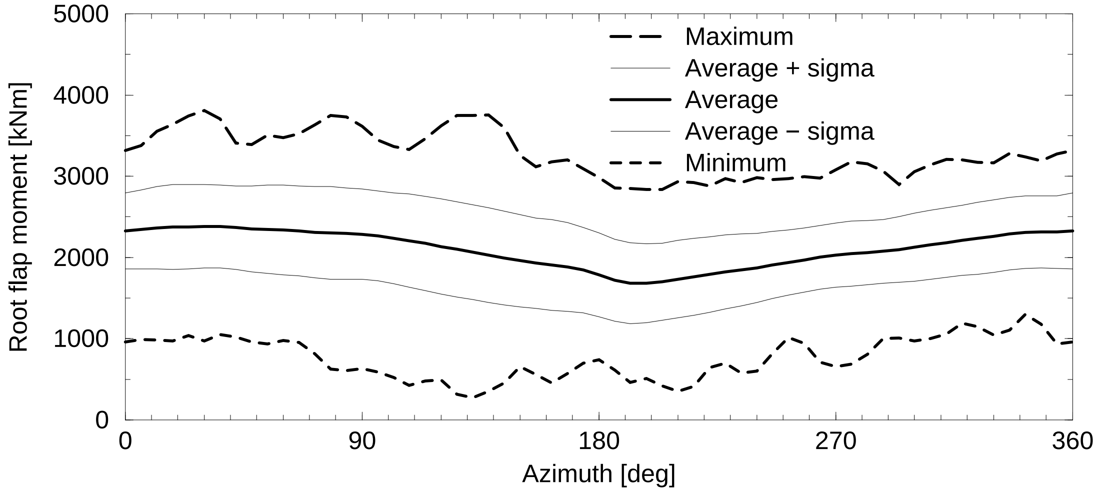


Figure 39: Plot of bin- averaged flap moment from *bin* for output type 1

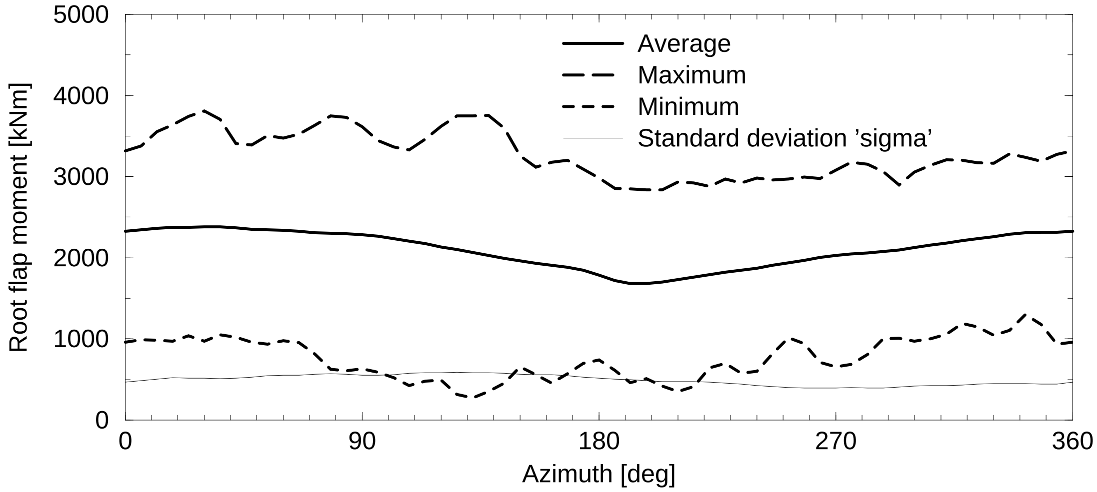


Figure 40: Plot of bin- averaged flap moment from *bin* for output type 2

F DESCRIPTION OF THE BINARY DATAFILES

The most complete output file of PHATAS is the so-called ’binary datafile’, which contains most properties of the structural dynamic response, except for the aerodynamic properties of the blade elements (such as angle of attack and aerodynamic coefficients). During the development of PHATAS, more information may be added to the binary datafiles, for which reason the contents of these files are strictly related to the PHATAS release. With the postprocessor *phpost*, see Appendix C most of the properties can be retrieved, and presented as time series of extremes.

For some design or research purposes the set of output properties that are available by the postprocessor, chapter 4, may be insufficient. For those applications a simplified version of the postprocessor is available by its FORTRAN source code. Here ’simplified’ means that reconstruction of the sectional loads in the outboard locations of the blade is not included, which are stored after being ’compressed’ with a complex algorithm.

As far as the user-defined postprocessing requirements are limited to expressions in terms of the loads in the blade root and in the rotor shaft (rotating or non-rotating) this open-source processor is available. To support tailor-made modifications of this open-source postprocessor the contents of the records are described below. This description holds for binary datafiles of PHATAS release "OCT-2002", "NOV-2003", and "APR-2005".

The ’binary datafiles’ are written by PHATAS as `UNFORMATTED , DIRECT ACCESS` files with a record length of at least 28. The record length of the binary datafile depends on the model of the turbine and of the blades, e.g. number of elements, and is written on the first record. The record length can be read after (’initial’) opening of the governing file with record length 28. The record length is expressed in ’words’, which is the length of a 4-byte variable.

F.1 Contents of first 5 records

Contents of Record 1

Variable type	Name	Description
`CHARACTER*8`	*cprog*	Program label; should contain the string ’PHATAS’.
`CHARACTER*16`	*crel*	Release label; either ’OCT-2002’, ’NOV-2003’ or ’APR-2005’.
`CHARACTER*20`	*cmodel*	Label of turbine model, from PHATAS input file.
`CHARACTER*20`	*cload*	Label of the load case, from PHATAS input file.
`INTEGER*4`	*irecl*	Record length of the binary datafile in ’words’.
`INTEGER*4`	*irfrst*	Record number with results at time = 0.
`INTEGER*4`	*indat*	Number of records with results of a time-step.
`REAL*4`	*delt*	Time increment between the records on the datafile.
`REAL*4`	*ht*	Height [m] of the rotor centre.
`REAL*4`	*ezroot*	Radius [m] of the blade root.
`REAL*4`	*rlb*	Length [m] of a rotor blade, from rotor centre to blade tip.
`INTEGER*4`	*intowl*	Number of which the last 2 digits give the number of tower elements, and the hundreds are the number of bending modes.
`INTEGER*4`	*inel*	Number of which the last 2 digits give the number of blade elements N, and the ’hundred’s the number of rotor blades B.
`INTEGER*4`	*indof*	Number of variables for discrete degrees of freedom.

The time span of the results in the datafile is (’*indat*’ -1) · ’*delt*’.

Contents of Record 2

If opened with record length '*irec1*' the contents of record 2 can be read as:

Variable type	Name	Description
INTEGER*4	*istat*	Exit-status: '0' for successful- and '1' for aborted calculation.
CHARACTER*6	*cdate*	Date [YYMMDD] at which calculation is completed.
CHARACTER*6	*ctime*	Time [HHMMSS] at which calculation is completed.
REAL*4	*t5rev*	Time [s] of the time that was skipped for statistics.
REAL*4	*rnrev*	Total number of revolutions after '*t5rev*'.
REAL*4	*vave*	Average wind speed [m/s] after '*t5rev*'.
REAL*4	*fiwav*	Average misalignment [rad] after '*t5rev*'.
REAL*4	*omgav*	Average rotor speed [rad/s] after '*t5rev*'.
REAL*4	*daxav*	Average aerodynamic thrust [N] after '*t5rev*'.
REAL*4	*pelav*	Average generator power [W] after '*t5rev*'.
REAL*4	*pitave*	Average pitch angle [rad] after '*t5rev*'.
REAL*4	*ommin*	Minimum rotor speed [rad/s] after '*t5rev*'.
REAL*4	*ommax*	Maximum rotor speed [rad/s] after '*t5rev*'.
REAL*4	*tmaxom*	Time [s] at which maximum rotor speed is found.
REAL*4	*umax*	Maximum tip displacement [m].
INTEGER*4	*ibmax*	Blade with maximum tip displacement.
REAL*4	*tmaxu*	Time [s] at which maximum tip displacement is found.
REAL*4	*azmax*	Blade azimuth [rad] at which max. tip displacement is found.
REAL*4	*umin*	Minimum tip-tower distance [m].
INTEGER*4	*ibmin*	Blade with minimum tip-tower distance.
REAL*4	*tminu*	Time [s] with minimum tip-tower distance.
REAL*4	*diatow*	Tower (outer!) radius [m] at lowest blad-tip height.
REAL*4	*ptomax*	Pitch angle [rad] at occurrence of the maximum rotor speed.

The latter property *ptomax* is not present in files from release "OCT-2002".

All statistical properties of the calculation (average, minimum and maximum values) are traced after skipping '*t5rev*' seconds. This is for a net time span of : ('*indat*' -1) · '*delt*' - '*t5rev*'. The time '*t5rev*' is the time needed for the first five revolutions, or for short calculations 10% of the analysis time.

Contents of Record 3

Variable type	Name	Description
`REAL*4`	*ezhub*	Distance [m] from tower top to shaft tower-axis intersection.
`REAL*4`	*erotor*	Lateral eccentricity [m] of the rotor hub.
`REAL*4`	*alna*	Tilt angle [rad] of the rotor shaft.
`REAL*4`	*dn*	Distance [m] along rotor shaft from rotor centre to tower axis.
`REAL*4`	*extet*	Location [m] of teeter hinge behind rotor centre.
`REAL*4`	*del3t*	Orientation [rad] of teeter hinge.
`REAL*4`	*alfcon*	Cone angle [rad] of the nacelle.
`REAL*4`	*yspit*	Radial position [m] of pitch bearing from PHATAS input file.
`REAL*4`	*xcpit*	Chordwise position [m] of pitch bearing.
`REAL*4`	*dsdtet*	Radial motion [m/rad] of (screw-spindle) pitch mechanism.
`REAL*4`	*upreb*	Flap-wise pre-bend deformation [m] at tip.
`REAL*4`	*vpreb*	Lag-wise pre-bend deformation [m] at tip (not for "OCT-2002").
`INTEGER*4`	*ibbend*	Node number where pre-bend starts.
`INTEGER*4`	*iebend*	Node number where pre-bend ends.
`REAL*4(1:N+1)`	*bch*	Chord [m] of each aerodynamic blade element.
`REAL*4(0:N)`	*teta*	Twist [rad] at the nodes of the blade.
`REAL*4(1:N,1:B)`	*xcg*	Chordwise location [m] of mass centre of each element.

Here `N` is the number of blade elements, written in the last 2 digits of variable '*inel*' on record 1 and `B` is the number of rotor blades, given by the 'hundred's of variable '*inel*'.

Contents of Record 4

In the PHATAS model the blade mass distribution is concentrated to values at the nodes, where each node *iel* has a mass *ymau* from the inboard element *iel*-1, and a mass *ymal* from the outboard element *iel*. These masses are stored as REAL*4 variables in the order

```
((ymal(iel,ibl),iel=1,inel),ibl=1,inob), ((ymau(iel,ibl),iel=1,inel),ibl=1
```

Contents of Record 5

Record 5 contains some inertia properties of the turbine and the blade bending stiffnesses. The first part of these properties is:

Type	Name	Description
`REAL*4`	*qksh*	Torsional stiffness [N/rad] of the rotor shaft.
`REAL*4`	*rmnac*	Total mass of nacelle [kg], excluding hub.
`REAL*4`	*sxnac*	Static moment of the nacelle mass [kg*m], downwind.
`REAL*4`	*sznac*	Idem, above tower top (not for "OCT-2002" and "NOV-2003").
`REAL*4`	*ajxxnc*	Nacelle rolling inertia (not for "OCT-2002" and "NOV-2003").
`REAL*4`	*ajyync*	Nacelle tilting inertia (not for "OCT-2002" and "NOV-2003").
`REAL*4`	*ajxznc*	Cross-inertia term (not for "OCT-2002" and "NOV-2003").
`REAL*4`	*ajzznc*	Yawing inertia of nacelle [kg*m*m].
`REAL*4`	*cxnac*	Aerodynamic drag area of nacelle, for longitudinal wind.
`REAL*4`	*cynac*	Aerodynamic drag area of nacelle, for side wind.
`REAL*4`	*gr*	Gear ratio of the drive train.
`REAL*4`	*ajslow*	Rotational inertia of the 'slow' rotating parts'.
`REAL*4`	*ajfast*	Rotational inertia of the 'fast' rotating parts', w.r.t. slow shaft.
`REAL*4`	*ajgbs*	Rotational inertia of the gearbox.
`REAL*4`	*rotmas*	Overall mass of the complete rotor.
`REAL*4`	*ajrot*	Overall rotational inertia of the rotor.

F.2 Contents of Records for Each Time Step

The records up to and including ('*irfrst*'-1) contain the properties of the turbine and the blade model. The records from '*irfrst*' and following, each contain the results of the solution at one time step. This means that record '*irfrst*' contains the results for time = 0s, record ('*irfrst*'+1) the results at time = '*delt*' etc. The record number for the results on time *t* is: '*irfrst*' + *t*/'*delt*'. The contents of each record are:

Variable type	Name	Description
`INTEGER*4`	*iter*	Number of iterations used for this solution.
`REAL*4`	*velwin*	Length [m/s] of wind velocity vector.
`REAL*4`	*windir*	Wind direction [rad] in the horizontal plane.
`REAL*4`	*aertor*	Aerodynamic torque [Nm].
`REAL*4`	*axfor*	Aerodynamic thrust [N].
`REAL*4`	*elpow*	Generator power [W].
`REAL*4`	*fiyaw*	Yaw angle [rad].
`REAL*4`	*dfiydt*	Yaw rate [rad/s].
`REAL*4`	*d2fiy*	Yaw acceleration [rad/s^2].
`REAL*4`	*firot*	Rotor azimuth [rad] (zero for blade 1 at 12 o'clock).
`REAL*4`	*dfirdt*	Rotor rotational speed [rad/s].
`REAL*4`	*d2firo*	Rotor acceleration [rad/s^2].
`REAL*4(1:9)`	*uwrite*	Values of 9 user-reserved output variables.
`REAL*4(1:3)`	*rotmom*	Three moment components in the rotor shaft.
`REAL*4(1:3)`	*rotfor*	Three force components in the rotor shaft.
`REAL*4(1:B)`	*shflap*	Flap-wise shear force in the blade root (only for "APR-2005").
`REAL*4(1:B)`	*shlead*	Lead-wise shear force in the blade root (only for "APR-2005").
`REAL*4`	*pitan(1)*	Pitch angle [rad] of blade 1.
`REAL*4`	*dpitdt(1)*	Pitch rate [rad/s] of blade 1.
`REAL*4`	*pitan(2)*	Pitch angle [rad] of blade 2.
`REAL*4`	*dpitdt(2)*	Pitch rate [rad/s] of blade 2.
`REAL*4`	....	Following loads are even complicated to describe.

The remaining properties depend among others on the number of degrees of freedom while the blade-properties are converted to use less disk space.

When using the 'standard' PHATAS version (no routines linked) the last elements of array 'uwrite' contain the blade root bending moments. The flapwise moment in blade '*ibl*' are stored in '*uwrite*(11-2**ibl*)'. For release "APR-2005") the leadwise moment in blade '*ibl*' of a '*B*' bladed rotor are stored in '*uwrite*(10-2**ibl*)', see also chapter 4.