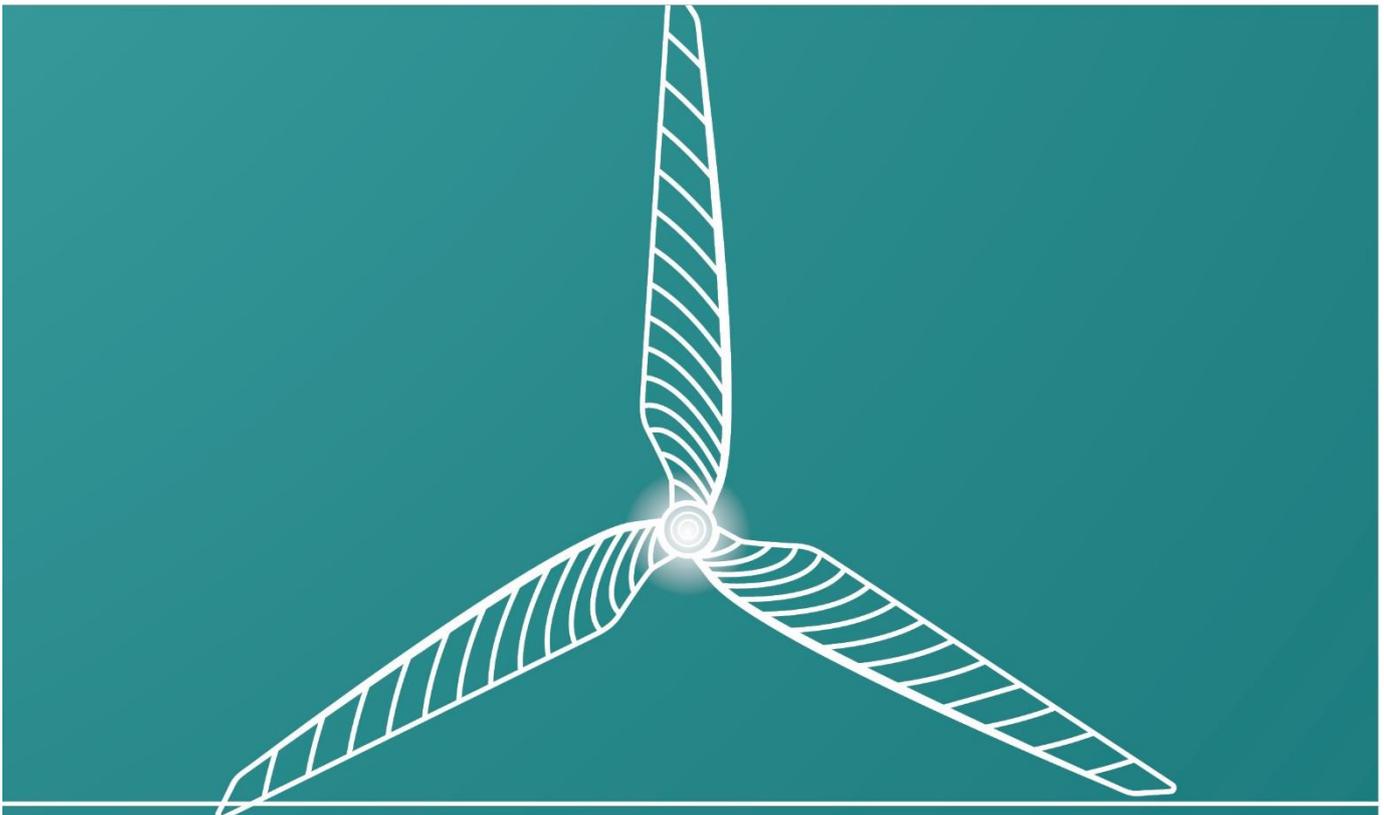


USER MANUAL

BLADED

DNV GL - Energy



SAFER, SMARTER, GREENER



***Bladed* User Manual**

Version 4.9

DISCLAIMER

Acceptance of this document by the client is on the basis that Garrad Hassan & Partners Ltd is not in any way to be held responsible for the application or use made of the findings of the results from the analysis and that such responsibility remains with the client.

COPYRIGHT

All rights reserved. Duplications of this document in any form are not allowed unless agreed in writing by Garrad Hassan & Partners Ltd.

© 2018 Garrad Hassan & Partners Ltd.

DNV GL

One Linear Park, Avon Street, Temple Quay, Bristol, BS2 0PS, UK

www.dnvgl.com

bladed@dnvgl.com

1 INTRODUCTION

Bladed is an integrated software package for wind turbine performance and loading calculations.

The **Bladed** theory manual gives full background information and details of the calculation methods used. This user manual complements the theory manual by describing the operation of the user interface. Much of this information is also available in the on-line Help facility.

1.1 Installation

Bladed is designed to run on PC with the following minimum specification:

- OS: Windows 7
- RAM: 4 GB
- Cores: 4
- Processor 2.5GHz
- SIMD instruction set SSE3
- Resolution: 1280 x 1024 (small fonts)
- HDD: 2GB required for installation files

RAM

1GB per core; high bandwidth RAM recommended (e.g. DDR3). If insufficient memory is available it will reduce calculation performance.

Processor

Fast single-threaded performance: In general, higher the clock speed the faster an individual calculation will run.

Cores

One core per calculation; one further core per instance of the user interface.

HDD

SSD for fast IO performance.

Bladed is supplied as a download or on DVD and is installed by means of a setup program. The setup procedure suggests an installation directory for the software, but allows the user to change this if desired. A directory on the local hard disk is recommended.

Except for demonstration and educational versions, a software key or USB dongle is provided. A network dongle can also be supplied on request. The setup procedure can install the appropriate device driver and network licence manager.

The setup procedure also allows the installation of the Acrobat Reader if desired, which is used to view the User and Theory Manuals on screen. It also installs the Matlab runtime libraries which are used by several modules within **Bladed**.

1.2 Bladed modules

The **Bladed** package consists of a Base module and a number of optional modules, for which licences are separately available. The Base Module consists of three parts:

Base Module	Capabilities
Main module	Steady state calculations, Saving project files, Writing reports, Graphics.
Simulation module	Modal analysis, Simulations, Turbulence, Waves, Post-processing.
Batch module	Running calculations in batch mode.

The following optional modules are available in addition:

Optional Modules	Capabilities
Control	Generation of linearised turbine models.
Seismic	Earthquake simulation.
Hardware test	Test a real turbine controller by interfacing it to a Bladed simulation running in real time.
Offshore support structure	Multiple member tower and support structure.
Advanced Transmission Interface	Allows user-defined gearbox/transmission dynamics to be integrated into Bladed simulations through a DLL interface.
Advanced Pitch actuator Interface	Allows user-defined pitch actuator dynamics to be integrated into Bladed simulations through a DLL interface.
Electrical dynamics	Electrical dynamics models of generators and power converters; network voltage and frequency transients.
WindFarmer link	Site-specific load calculations using <i>WindFarmer</i> output.

1.3 The **Bladed** Educational version

Bladed Educational is available solely for academic and teaching purposes. Only the Base Module is available, and there are a number of further restrictions, as follows:

Turbine model:

- The number of blade radial stations is limited to 10 maximum;
- The number of tower stations is limited to 5 maximum;
- The pitch actuator model is fixed to a first order lag response with 0.3s time constant. No other pitch actuator models are allowed.
- The external controller facility is not available.
- The use of encrypted data is not supported.

Environment:

- Simulation periods are limited to 60s maximum.
- The "seed" used to initialise the random number generators used to synthesise wind turbulence and wave time series is fixed.
- Only the longitudinal component of wind turbulence can be used in simulations. (The other components can be synthesised, but will be ignored in simulations.)
- Although the Seismic module is not available, a single seismic calculation is available for demonstration purposes.

Calculations and post-processing:

- Calculations cannot be run in Batch mode.
- Model linearisation is not available, except as a demo calculation.
- The following post-processing calculations are not available, except as demo calculations:
 - ultimate loads
 - ultimate load cases
 - linear model
- The Extreme Load Extrapolation calculation is not available.

1.4 The **Bladed** Demonstration version

A demonstration version of **Bladed** is available.

A demo project, **demo_a.prj**, is supplied with **Bladed** and will be found in the installation folder. This contains simplified details of a representative but artificial 3-bladed 2MW offshore variable speed pitch regulated turbine. This can be used to run most of the available calculations.

The demonstration version is restricted as follows:

Operation

Modal analysis calculation
Turbulent wind generation

Restrictions

Turbine details may not be changed
Turbulence details may not be changed

Steady calculations and simulations	Turbine details may not be changed; other calculation details may be changed
Post processing calculations	Calculation details and input data may not be changed. The Extreme Load Extrapolation calculation is not available.
Batch system	Calculations cannot be run in Batch mode.
Saving project files	Not allowed
Writing reports	Not allowed, but a sample report may be viewed.
Encrypted data	Not supported
Graphics: viewing calculation results	Unrestricted

When running a calculation, a message box warns that the appropriate demo data will be used for the calculation.

Other project files are also supplied for demonstration purposes. Although these can be viewed, they cannot be used for running calculations.

1.5 Support

Bladed is supplied with a one-year maintenance and support agreement, which can be renewed for further periods. This support includes a 'hot-line' help service by telephone, fax or e-mail:

Telephone: +44 (0)117 972 9900
 User portal: <https://renewableenergysoftwareportal.dnvgl.com/>
 E-mail: bladed@dnvgl.com

It is DNV GL policy to work with clients to respond to their needs so that the software can be constantly improved. As with any software of this complexity, a total absence of bugs cannot be guaranteed, and any reports of bugs, along with comments or suggestions on any aspect of the software, are welcomed. The maintenance and support agreement includes free provision of any revisions or upgrades of the software during the period of the agreement.

Modifications to the software to meet the individual needs of specific clients can be made by arrangement. Such work will be charged for at commercial rates.

1.6 Documentation

Much of the information contained in this manual is also available in the on-line help facility.

There is also a **Bladed** Theory Manual which explains in more detail the theory behind the calculation methods used.

Section 2 of this User Manual gives an overview of how **Bladed** can be used. The later sections then provide more detail on each function. Sections [3](#) to [5](#) explain in detail how to set up a model of a turbine. Section [6](#) covers the specification of the wind field, and the sea state for offshore turbines. Section [7](#) then explains how to set up and run wind turbine calculations. The post-processing, graphics and reporting facilities are described in Sections [8](#), [9](#) and [10](#) respectively.

1.7 Acknowledgements

Bladed was developed with assistance from the Commission of the European Communities under the JOULE II programme, project no. JOU2-CT92-0198.

2 USING BLADED

This chapter gives an overview of the ways in which **Bladed** can be used. It covers the following topics:

- General description and layout of the user interface
- Entering data
- Using project files
- Performing calculations
- Viewing results
- Compiling reports

More detailed descriptions follow in later chapters.

2.1 General description and layout of the user interface

On starting **Bladed**, the main Toolbar appears together with the Calculations screen. The Toolbar consists of a set of graphical icons and a number of pull-down menus. The Calculations screen allows the user to select, define and execute a particular calculation.

There are various ways of opening further screens which allow the user to define the characteristics of the various parts of the turbine, as well as the characteristics of the wind and various parameters which control the execution of calculations. There is also a graphics facility for viewing results.

2.1.1 Main toolbar - pull-down menus

The pull-down menu may be used as follows:

File: use this menu to create, open and save project files, and to import modules from other project files. A project file (.prj) contains wind turbine information and/or parameters defining calculations. Using the file type selector on the File Open dialogue box, it is also possible to open a project backup file (.prx), or a file containing all the details relating to a calculation previously carried out (.\$pj). This is useful for re-running the calculation with or without modifications.

Use the **Import** facility to import individual modules from other project or calculation details files into the currently active project.

The project file header information may be entered or edited using **Project Info**. Use **Protect project** to enter a security password to prevent the file being modified by unauthorised users. The **Encrypt** facility allows any part of the turbine model to be encrypted with a password, so that the data can still be used for calculations but is completely invisible to the user. If necessary, different parts of the turbine can be encrypted by different users, each with a separate password. To make data visible again, use **Decrypt** and enter the correct password.

Specify: this menu allows the user to move directly to a particular screen for specifying any part of the turbine, or any calculation or load case. These screens are also accessible via the Toolbar Icons described below.

Calculation: this menu allows particular calculations to be carried out. See also the Calculation icon on the main toolbar described below. It also allows unwanted calculation results to be deleted.

Batch: this is used to control batch processing (see [7.2.7](#)) of multiple runs. It also gives access to the WindFarmer Link screen, for which a WindFarmer Link licence is required.

Reports: This menu also offers the possibility to write a project report or a calculation report, to append graphs to a report, and to edit or print existing reports. It also gives a choice of report format, which may be ASCII or Microsoft Word.

Tools: **Copy results** allows calculation results to be copied from one location to another. In doing so, the results can be converted between binary and ascii formats if required. There is also a facility to **Delete results**. **Compare Projects** allows two project and/or calculation details files to be compared, and can generate a detailed report of the differences. **Create Header Files** launches a tool to create header files for multiple ASCII data files so that **Bladed** post processing and data view can be used.

There is also a facility for configuring the current printer, and to specify user preferences for certain option settings (see [2.8](#)).

Windows: switches between any of the currently open windows, and gives access to the turbine summary information window and to the 3D graphical display (which can be animated using the double-arrow icon).

Help: activates the on-line Help facility, which contains detailed information on **Bladed** and how to use it. It also gives on-line access to the **Bladed** User and Theory manuals, and to a facility for upgrading the dongle or security device by entering an appropriate password.

2.1.2 Toolbar icons

Beneath the pull-down menus on the toolbar are the Toolbar Icons. Clicking on any one of these opens up the corresponding screen, as follows:

- **Blades** (see [3](#)) to define the blade properties.
- **Aerofoil sections** (see [3.8](#)) to access a database of aerofoil section data.
- **Rotor** (see [4.1](#)) to define the properties of the rotor and overall turbine configuration.
- **Tower** (see [4.5](#)) to define the tower properties.
- **Power Train** (see [4.7](#)) to define the drive train, generator, energy losses and electrical network.
- **Nacelle** (see [4.13](#)) to define the nacelle geometry and mass.
- **Control** (see [5](#)) to define both power production and supervisory control systems.
- **The wind input** (see [6.1](#)) to define the wind speed and direction including spatial and temporal variations.
- **Sea state** to define the waves (see [6.12](#)) currents (see [6.13](#)) and tide height (see [6.14](#)) for offshore turbines.
- **Flexibility Modeller** (see [7.1](#)) to specify the modal analysis of blade and tower vibrations.
- **Calculations** (see [7.2](#)) to select, specify and execute any particular calculation.
- **Data View** (see [9](#)) to view graphs or generate tables of results.
- **Analyse** (see [8](#)) to specify post-processing of results.

2.1.3 The calculation window

The **Calculations** window lists all the available calculations, any one of which may be selected by clicking on it with the mouse. Next to each calculation is an indicator light. If this is green, the calculation may be performed. A red light indicates that no data is available to perform the selected calculation, while a yellow light indicates that some of the required data has been defined but not all.

Below the calculations is a small window showing all the data modules which need to be defined in order for the selected calculation to be performed. Double-clicking on one of these data modules immediately opens up the relevant screen in which the data for that module is defined. It is therefore a simple matter, having selected the calculation to be performed, to work through all required data modules which are still undefined and assign the relevant data.

2.1.4 Sequence of operations

Before any turbine calculations can be done, it is necessary to specify the principal characteristics of the blades and rotor. Open the relevant screens by one of the means described above, i.e. from the **Specify** pull-down menu, from the relevant Toolbar icon, or from the **Calculations** screen with one of the turbine calculations selected.

In order to be able to define the aerodynamic characteristics of the blades, it is necessary to enter or import the relevant aerofoil datasets into the aerofoils database if it is not already present, using the **Aerofoil** icon on the toolbar, or from the **Specify** pull-down menu.

It is also necessary to define the following modules:

- Physical constants (although standard values will have been defined by default)
- Aerodynamics control
- Calculation definition depending on which calculation is selected (further parameters specific to the selected calculation)

These are all on the Calculation Parameters screen, accessible from the **Calculations** screen or from the **Specify** pull-down menu on the toolbar.

Having defined this fundamental data, it should then be possible to perform the following steady state turbine calculations at a fixed rotational speed and pitch angle defined on the Calculation definition screen specific to each calculation:

- Aerodynamic information (to examine the aerodynamics at each blade station, such as lift and drag, inflow, tip loss etc. at a specified wind speed)
- Performance coefficients (dimensionless power, torque and thrust coefficients as a function of tip speed ratio)
- Power curve (at specified fixed rotor speed and pitch angle).

These calculations are quick and the results can instantly be examined graphically using the **Data View** icon on the toolbar. From the power curve, the annual energy yield can also be calculated. This is one of the post-processing calculations, and again the results can be seen using **Data View**.

The **Outputs** button on the **Calculations** screen allows the user to specify in some detail which loads and other outputs are desired. Once this is done, the Steady-state operational and parked loads calculations can also be carried out (at specified fixed rotor speed and pitch angle), and **Data View** used to see the results.

At the preliminary design stage, various turbine parameters can be adjusted and all these calculations can be repeated rapidly until satisfactory results are obtained.

As further data is defined, more complex calculations can be carried out. For example, by defining suitable combinations of drive train, generator and control characteristics, it becomes possible to calculate the steady power curve and steady operational loads with the rotor speed and/or pitch angle varying with wind speed as appropriate.

After defining the numbers of vibrational modes on the **Flexibility Modeller** screen, along with the necessary mass and stiffness information as indicated on the **Calculations** screen, it becomes possible to perform a Modal analysis calculation. Blade and tower bending can then be taken into account in the calculations.

Simulations can also be performed once the Simulation control and Time varying windseastate modules are defined. As always, the minimum data required is always indicated on the **Calculations** screen, making it easy to ensure that the necessary data is defined for each calculation to be done.

Note also the **Show options** button at the bottom of the **Calculations** screen. This provides a rapid means of switching particular features off or on for a particular calculation. For example, provided the energy losses module has been defined, the energy losses can be switched off or on very easily using this feature, for all calculations where energy losses are actually relevant.

2.2 Entering data

In each window, data is entered by typing in the required information in the fields provided. Where there is a choice of alternative options, selection buttons or pull-down selectors are provided. In some cases check boxes are provided for enabling or disabling particular items (the item is selected when a tick appears in the box).

Where numerical values are entered, except in the case of dimensionless numbers, the units in which the data is to be specified are shown. Double-clicking on the entry brings up a units conversion box, allowing the user to enter values in a choice of different units.

Once data in a window has been edited, it is necessary to assign the changes by clicking the **OK** button. Alternatively, any changes made since the data was last assigned may be reversed by clicking the **Cancel** button.

2.3 Using project files

A project file stores all the currently defined information relating to the turbine, the wind field information, and calculation parameters. Click **OK** on any open windows to ensure that the data is assigned before saving to a project file. Project files have a .prj extension.

When a calculation is performed, all the information which is of relevance to that calculation is stored in a special calculation project file which is stored with the calculation results. This file will have a .\$pj extension, and can be loaded just like a standard project file.

A project report may be generated from the **Reports** pull-down menu on the main toolbar.

2.4 Performing calculations

Calculations may be initiated either from the **Calculations** pull-down menu on the toolbar, or from the **Calculations** window by clicking the **Run Now** button. For some of the steady state calculations and all the simulations, the **Calculations** window also gives access to facilities for switching various calculation options on or off, and for specifying the calculation outputs required. Post-processing calculations may also be initiated using the **Execute** button in the post-processing window obtained by clicking the **Analyse** icon on the toolbar.

Long calculations may be stacked up using the **Run in Batch** button in the **Calculations** window. The whole batch of calculations may then be started so that they run one after the other, for example overnight. This is done using the **Batch** menu item on the toolbar.

Except for some auxiliary calculations, the user will be asked to specify where the calculation outputs are to be sent. A dialogue box allows the drive and directory to be selected, and a run name must also be specified. This is used to identify all the output files produced by that calculation. A new output directory may be created by editing the directory name.

While the calculation is running, a window appears which displays a progress bar showing how far the calculation has progressed, together with any warnings which may arise during the calculation. An 'Abort' button allows the user to stop a calculation which is in progress. Any part-written output files which have already been produced should be available. This means that a long simulation can be stopped without losing the results which have so far been produced.

2.5 Viewing results

After a calculation has been completed, the results can be viewed by clicking the **Data View** icon on the toolbar, and selecting the desired data for each channel. Up to six channels may be plotted on one graph. The outputs from any calculation just completed will normally be selected by default. Otherwise, the dialogue box allows any drive and directory to be selected. If the results of several calculations are available within the directory, a pull-down selector allows the user to choose the run name which was specified when the calculation was initiated.

The output files available from that calculation are shown in the Data Group window. Selecting one of these displays a contents list for the file from which the user chooses the variable of interest. Click **OK** to assign the data to the selected graph channel.

Often there is just one independent variable, for example Time in the case of dynamic simulations. Sometimes there is a choice of independent variables: for example a blade bending moment from a simulation could be plotted either against time or radius. Double-click to change the choice of independent variable.

If two independent variables are defined, a three-dimensional graph will result. Three-dimensional plots are most useful for small datasets.



The **View Messages** button displays any warning or error messages generated during the run. In a few cases, some additional information is available by clicking the **Further Info** button. Other buttons allow whole runs or individual data groups to be deleted.

2.6 Compiling reports

If Microsoft Word is installed, it is possible to generate neatly formatted project and calculation reports in Word format. It is also possible to insert calculation results into these reports, either as tables or as graphs. See also User Preferences (see [2.8](#)).

If Word is not available, change the report format to ASCII using the **Reports** pull-down menu. This will cause reports to be generated in tab-delimited ASCII format, suitable for reading into other packages if required. This option does not allow graphs to be inserted into the report.

Reports are generated from the **Reports** pull-down menu on the main toolbar. There are two types of reports:

- **Project reports:** these may contain any currently assigned modules defining the turbine itself, as well as calculation parameters and wind field details if required.
- **Calculation reports:** these may be generated for any calculation which has been carried out, or which is awaiting execution in the Batch queue. A dialogue box allows the desired calculation to be selected. Calculation reports may include any turbine details which are relevant to the calculation, as well as details of the calculation itself and the wind field used if applicable. They may be generated as stand-alone reports, or appended to another report such as a project report.

When reports are generated, they may be appended to existing report files if desired. Thus it is possible to create a project report, and subsequently append calculation reports to it whenever calculations are done, and also append calculation results either as graphs or tables.

2.7 Data Encryption

Any part of the turbine data may be encrypted, using a password entered by the user. Encrypted data becomes invisible to the user, but can still be used to run calculations. It can only be decrypted by entering the correct password. The encrypt data, either click the **Encrypt** button on the appropriate data entry screen, or use the **File ... Encrypt** pull-down menu on the toolbar and tick the modules you wish to encrypt. You will first be prompted for an encryption group name, which identifies the encrypted data. Each encryption group has its own password, so it is possible for different users, perhaps different component suppliers, to encrypt their own data. When a new encryption group is created, the user is prompted to enter a password, and then to verify it. Passwords are case-sensitive. If an existing group is selected, the user must be able to enter the correct password in order to add more data to the group.

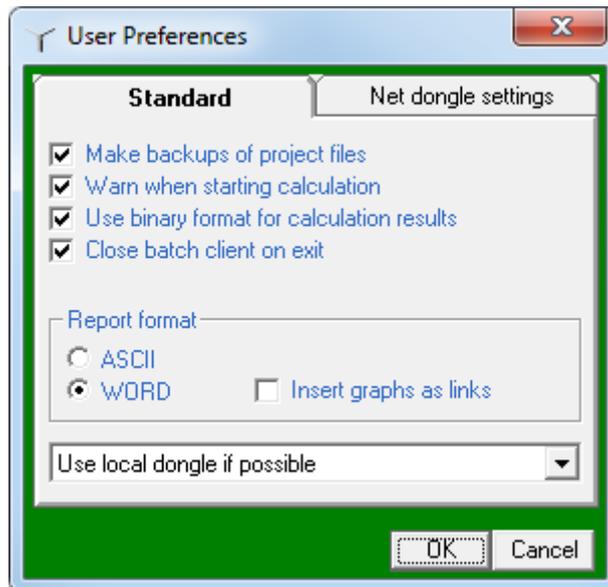
Bladed then offers the user the possibility to specify to disable certain simulation outputs, in case this output data is thought to give too much away. For example a blade manufacturer may choose to encrypt the blade data, and might decide to prevent subsequent calculation results from including the blade loads or deflections, or to allow blade loads only the root station for example, or deflections only at the tip. Of course it is important not to restrict the calculation outputs so far that the calculations are no longer useful. Encryption of the blade data automatically causes the rotor mode shapes to be encrypted, and encryption of tower data causes the tower mode shapes to be encrypted. Modal frequencies remain visible to the user.

To decrypt data, either click the **Decrypt** button on the appropriate data entry screen, or use the **File ... Decrypt** pull-down menu on the toolbar and choose which encryption group to decrypt. The correct password then needs to be entered.

2.8 User preferences

Select **Tools** and **Preferences** from the main toolbar menu, and select desired options as follows:

2.8.1 Standard tab



Make backups of project files: to make a backup file (.prx) whenever a project file (.prj) is opened.

Warn when starting calculation: to display a warning when a shelled calculation is about to start.

Use binary format for calculation results: binary output is faster and results in smaller files.

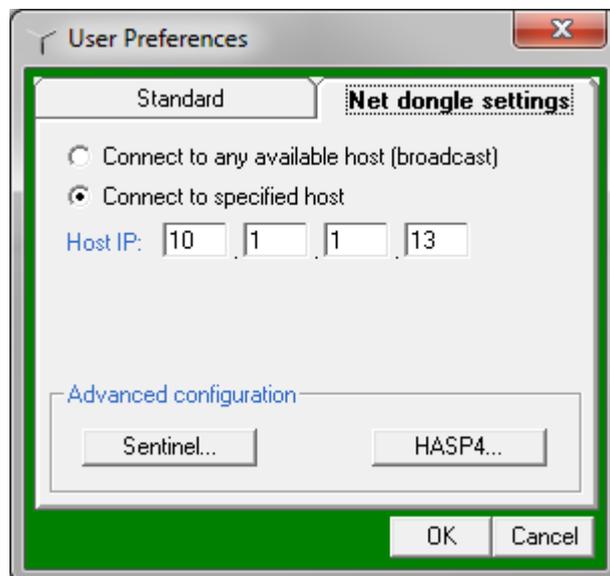
Close batch client on exit: On exit of **Bladed**: Close Batch Viewer, and unless this machine is the manager of the batch, also close the Batch Framework, resulting in aborting running jobs and removing this machine as a runner of the batch.

Report format: Select ASCII or WORD.

Insert graphs as links: to append graphs to WORD reports as links to external metafiles. If this option is not selected, graphs will be fully imported into the WORD document.

Dongle search order combo: specifies whether to use a local dongle, network dongle or favour a local dongle over a network dongle (in the latter case if a local dongle is not found, **Bladed** will search for a network dongle). If you have a network dongle you can change the way in which the network is searched by clicking the Net dongle settings tab.

2.8.2 Net dongle settings tab

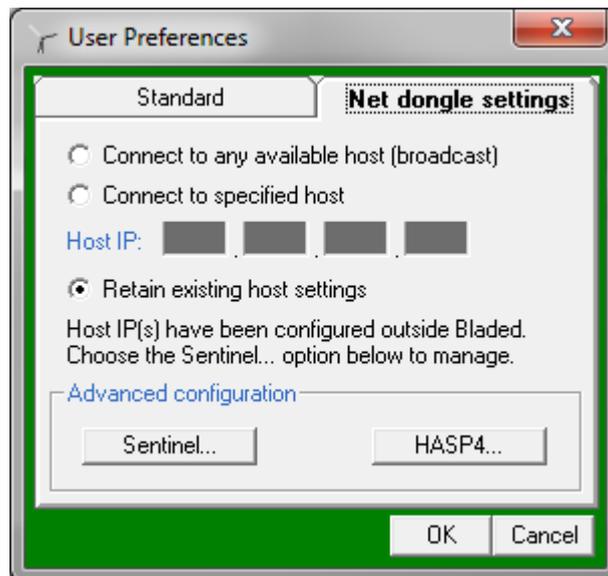


Connect to any available host: Allows the licence manager on the local machine to perform a broadcast search over the entire network for available network hosts. This setting is not recommended as it significantly increases the amount of network traffic and may result in the incorrect licence being reserved if multiple network dongles are found.

Connect to specified host: Restricts connection to a network dongle hosted on the named IP. This is the recommended option.

Advanced Configuration: Allows advanced settings to be entered for HASP4 and Sentinel HASP network licences using the vendor's configuration tools. However, the default settings are usually fine.

If **Bladed** detects that multiple hosts have been defined in Sentinel Admin Control Center (ACC), the option to retain existing settings becomes available. Configuration of multiple hosts should be performed using ACC.



If the option to connect to a specified host is selected with multiple hosts already defined, **Bladed** will prompt the user whether to overwrite the existing configuration.

2.9 Context-sensitive help

For context-sensitive help, press the F1 key at any time.

2.10 Dongles

Full **Bladed** licences are activated using a security device or dongle. Only demo use and Educational licences do not require a dongle. Two types of dongle are used:

Stand-alone dongle

This is plugged into the USB port of the computer on which **Bladed** is installed and run. The Sentinel HASP device driver must be installed on the computer. This dongle contains a single licence for the **Bladed** base module and any optional modules which have been purchased.

Network dongle

This is plugged into the USB port of a computer on the network, for example a network server (this computer is referred to here as the 'server'). Both the Sentinel HASP run-time environment and the HASP4 Licence Manager must be installed and running on the server, and are available to install from within the **Bladed** installer. The HASP4 Licence Manager enables **Bladed** versions pre-v.3.85 to be run using the same network dongle, and is also required for running **Bladed** v3.85 and later. The Sentinel HASP Licence Manager is automatically installed as part of the Sentinel HASP run-time environment installation.

Note: When installing the HASP4 Licence Manager, choose the option to install as a service.



The network dongle may contain more than one licence for the **Bladed** base module and for any optional modules which have been purchased. The number of licences determines the maximum number of users who can simultaneously use each module. The licences are accessible to users on other computers on the network, here referred to as 'client' computers. The client computers must have Bladed and the Sentinel HASP run-time environment installed, and do not need to install the HASP4 Licence Manager. To connect to the server only, the option "Use network dongle" should be selected from the combo at the foot of the Standard tab of the Preferences dialog, available from the **Tools ... Preferences** menu on the **Bladed** toolbar (see [2.8.1](#)). The server IP address should be entered in the Net dongle settings tab (see [2.8.2](#)).

Local use of network dongle

The network dongle may also be used as a stand-alone dongle on the licence server. Licences are checked out in the same way as a remote network dongle. The option "Use local dongle" should be selected in the Standard tab of the Preferences dialog (see [2.8.1](#)).

Bladed versions pre-v.3.85 must not be started on the server if remote clients wish to use network licences as this would disable the network dongle.

Note: Before removing the network dongle from the server, all licences should be freed and the HASP4 service stopped as described below.

Remote updating of dongles

Remote update codes may be issued to you from time to time by Garrad Hassan. To apply these, select **Help... Security device upgrade** from the Toolbar, paste the codes into the box provided, and click the Upgrade button.

For network dongles, it is important to free all HASP4 licences before applying remote update codes by closing all **Bladed** instances running on client computers and stopping the HASP4 Licence Manager (service name "HASP Loader"). The service should be restarted after the update has been applied. There is no need to stop the Sentinel HASP Licence Manager service when applying a remote update.

After successfully applying the dongle update, **Bladed** will generate a C2V file which contains all the licensing data on the dongle. Please attach this file to an email and send to bladed@dnvgl.com.

Company code

From **Bladed** version 4.1 a unique company code is stored on the dongle. This allows third party suppliers of **Bladed** project files to authenticate a request for a project file against the dongle. To see the company code, select the Help About menu item.

2.11 Component download

The component downloader allows users to download **Bladed** project files containing components from third party suppliers with whom they have an agreement. To launch the component downloader select the Tools "Launch component downloader" menu item. The component downloader can also be accessed from some of the other screens. A list of providers can be seen in the drop down list. Select the provider required, enter your username and password as previously agreed with the provider and select Fetch components. This will then produce a list of available components. Type a download path into the download location box and select Download next to the required component.

3 DEFINING THE TURBINE BLADES

This section describes how to construct a model of the turbine blades, including geometrical, mass and stiffness properties as well as the aerodynamic characteristics of the aerofoil sections.

Click the **Blades** icon on the toolbar to open the **Blade Properties** screen.

Before any calculations can be carried out, the user must define the geometry (see [3.2](#)) and aerodynamic characteristics (see [3.8](#)) of the rotor blades on the **Blade Geometry** tab.

If any dynamic calculations are required, check the **Mass** checkbox on the **Mass and Stiffness** tab to allow the blade mass distribution (see [3.3](#)) to be defined.

If blade flexibility (see [7.1](#)) is to be modelled, also check the **Stiffness** checkbox to allow the blade stiffness distribution (see [3.5](#)) to be entered. An axial degree of freedom, a torsional degree of freedom and shear stiffness can also be defined by checking the appropriate checkboxes.

Holding the mouse over any of the blade property labels will bring up additional tool tip text which describes what the property is in more detail.

The blade properties are defined at a number of stations (see [3.1](#)) along the blade, although additional point masses (see [3.3.2](#)) may also be defined anywhere on the blade by clicking the **Add** button on the **Additional Mass** tab.

The **Graph** menu controls the graphical display of blade properties. The planform graph and the cross section plot are both updated as soon as any new values are assigned, giving an instant indication if faulty values are entered. Controls are provided to change the format of stiffness graphs, and to print the graph or save it to a metafile, which can subsequently be incorporated into a report.

Click the **Add** button on the **Blade Geometry** tab to add a new blade station. Stations are automatically sorted by radial position. To highlight a station, click on the station number, or double-click on a graph point. Click **Delete** to remove a highlighted station.

To enter or edit data, highlight the data item by clicking on it, or by moving to it using the arrow keys and then pressing the Return key. Press Escape to restore the previous value.

Select a block of cells by dragging the mouse. Whole rows may be selected by clicking or dragging on the row descriptions. The selected cells may be copied to the clipboard, or the contents of the clipboard may be pasted in. In this way, data from a spreadsheet or a tab-delimited ASCII file may be directly pasted in. Click **Undo** to reverse a paste operation.

Note that the Moving/Fixed and Foil Section entries may not be pasted in.

Click on Moving/Fixed to specify which parts of the blade are pitch or aileron controlled. Click on Foil Section to select an aerofoil section (see [3.8](#)) or to define a new section.

If a sharp discontinuity (see [3.1](#)) in blade characteristics is required, split a blade station into an inboard and an outboard station at the same radial position. This is especially useful to mark the discontinuity between a fixed and a pitchable or an aileron-controlled part of the blade. Use the **Split** and **Join** buttons to split or re-join highlighted blade stations. The first and last stations may not be split.

On the **Blade Information** tab, a name may be entered for the blade. The entire blade may also be made pitchable or fixed, depending on whether the blade is mounted to the hub by means of a pitch bearing. Furthermore it is possible to insert a wedge outboard of this pitch bearing by specifying:

Encrypt Blade: The blade geometry, mass and stiffness data may be encrypted, allowing other users to run calculations without being able to see the details of the blade. See [2.7](#) for further information on encryption.

The aerofoil data used on the blade may also be encrypted in the same way. The aerofoil sections to encrypt are selected from the list on the **Blade Information** tab.

Aerofoil conflicts: The list of aerofoils on the **Blade Information** tab will indicate if any of the (non-encrypted) aerofoil datasets used on the blade conflict with aerofoils in the database. In this case, click **Resolve conflict** to see the differences in detail and to decide whether to replace or rename conflicting aerofoil datasets.

3.1 Choosing blade stations

Blade stations are the points along the blade at which blade geometry, aerofoil, mass and stiffness data are defined. They are also the points at which blade loads will be calculated, if desired - see Defining outputs ([7.2.4](#)).

The blade stations must include the root and tip stations. Blade station positions are measured from the blade root, which is the point at which the blade is attached to the blade root section (see [4.2](#)). Thus the first blade station must be given a radial position of zero, and the position of the last station defines the length of the blade. This will be less than the rotor radius if a blade root section is defined, or if the blade is not straight.

A reasonably even spacing between blade stations is recommended. Try to define a blade station close to or preferably at the same position as any point mass which may be required - see Blade mass distribution ([3.3](#)).

Discontinuities

The blade may have sharp discontinuities in the mass distribution, the stiffness distribution, and the aerofoil section. There will also be a discontinuity where a blade changes from fixed to pitchable, or at the start or end of any aileron or airbrake section.

To specify such a discontinuity, highlight the blade station at the radial position where the discontinuity lies, and click the **Split** button to split the blade station into an inboard and an outboard station at the same radial position. The data may then be specified differently on either side of the discontinuity.

A split station may be re-joined if necessary to remove the discontinuity, by clicking the **Join** button.

3.2 Blade geometry

The blade geometry is defined at each blade station (see [3.1](#)) clicking on the data item to be defined or changed. The following data is required at each station:

Distance: This can be entered as a distance along the blade or as a distance along the blade root Z-axis. Select the appropriate option at the base of the screen.

- **Distance along blade:** the distance from the blade root to the current blade station, along the blade neutral axis, which does not have to be a straight line. It must be zero for the first station. If distance is entered along the blade root Z-axis, this value is calculated based on the distance along the blade root Z-axis and the neutral axis.
- **Distance along blade root Z-axis:** the distance of the blade station along the nominal pitch axis (with no pre-sweep or pre-cone). If distance is entered along the blade, this value is calculated based on the distance along the blade and the neutral axis.
- **Chord:** the distance from the leading edge to the trailing edge, i.e. along the chord line.
- **Aerodynamic Twist:** the local angle of the chord line. More positive values of twist and set angle push the leading edge further upwind. (See diagram below)
- **Thickness:** the thickness of the blade as a percentage of the chord at that station.
- **Neutral axis (x):** the distance from the blade root Z-axis to the neutral axis in the x direction. This would be non-zero if for example the blade was pre-bent. (See diagram below)

- **Neutral axis (y):** the distance from the blade root Z-axis to the neutral axis in the y direction. (See diagram below)
- **Neutral axis, local (x')**: the perpendicular distance from the chord line to the neutral axis in local coordinates, as a percentage of the chord. (See diagram below)
- **Neutral axis, local (y')**: the distance along the chord line from the leading edge to the neutral axis in local coordinates, as a percentage of the chord. (See diagram below)
- **Foil section:** an index number defining the aerofoil section (see [3.8](#)) at that station
- **Moving/Fixed:** differentiates between a fixed part of the blade and a part which is movable to achieve aerodynamic regulation or braking, either by bodily changing the pitch of that part of the blade, or by deploying an aileron, flap or other aerodynamic control (see [4.1](#)) surfaces.

Blade icing can also be specified, as discussed in section ([3.3.4](#)).

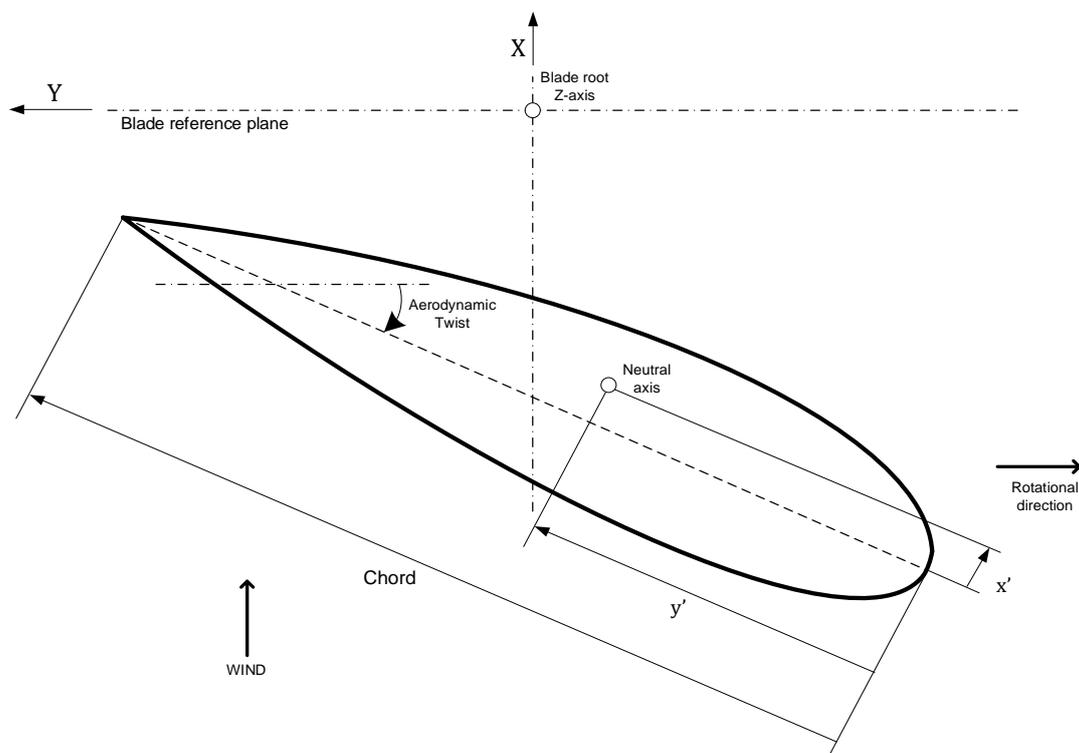


Figure 3-1: Diagram showing the Chord, the position of the Neutral axis relative to the blade root Z-axis and the leading edge, and the aerodynamic twist (seen from above). Note that this figure shows a Neutral axis with negative x and y co-ordinates with respect to the blade root co-ordinate system.

3.3 Blade mass distribution

For dynamic calculations it is necessary to define the mass of the rotor blades. Click the **Mass** checkbox **off** if static calculations are desired and the mass data has not been defined.

3.3.1 Distributed mass

For each blade station (see [3.1](#)) enter:

- **Centre of mass (x')**: the perpendicular distance from the chord line to the centre of mass in local coordinates, as a percentage of the chord. (See diagram below)
- **Centre of mass (y')**: the distance along the chord line from the leading edge to the centre of mass in local coordinates, as a percentage of the chord. (See diagram below)

- **Mass per unit length:** the gradient of the blade mass distribution at each station, with respect to the actual distance along the blade.
- **Polar inertia per unit length:** the gradient of the blade polar mass moment of inertia at each station about the local z_m -axis..
- **Radii of gyration ratio:** the radius of gyration of mass about y_m divided by the radius of gyration of mass about x_m . This defaults to the relative profile thickness but can be over-written by un-checking the "Use default radii of gyration ratio" checkbox.
- **Mass axis orientation:** the orientation of the principle axis of inertia. This defaults to the orientation of aerodynamic twist, but can be over-written by un-checking the "Use default mass axis orientation" checkbox. (See diagram below)

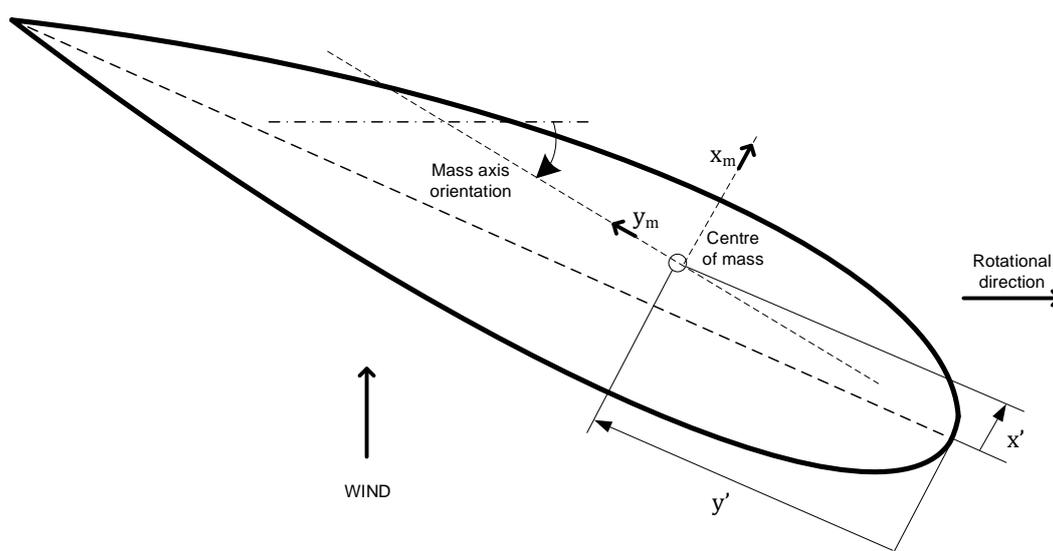


Figure 3-2: Diagram showing the position of the Centre of mass relative to the leading edge, and the mass axis orientation

Click the **Additional Mass** tab to specify any point masses (see [3.3.2](#)) and/or vibration dampers (see [3.3.3](#)) on the blades.

3.3.2 Point masses

If additional point masses are required at specific locations along the blade, click the **Additional Mass** tab and enter the point masses in the table. Click the **Add** button to add each point mass, and then enter the data required by clicking on the appropriate entry. Note that masses are automatically sorted by radial position. To remove a mass, highlight it by clicking on its number, and click **Delete**.

For each mass, the following data is required:

- **Mass:** The Mass required.
- **Distance along blade:** The position of the mass along the blade, measured from the blade root.
- **Chordwise position (x'):** the position of the mass perpendicular to the chord as a percentage of the chord at that point. (See diagram below)
- **Chordwise position (y'):** The chordwise position of the mass, measured backwards from the leading edge as a percentage of the chord at that point. (See diagram below)

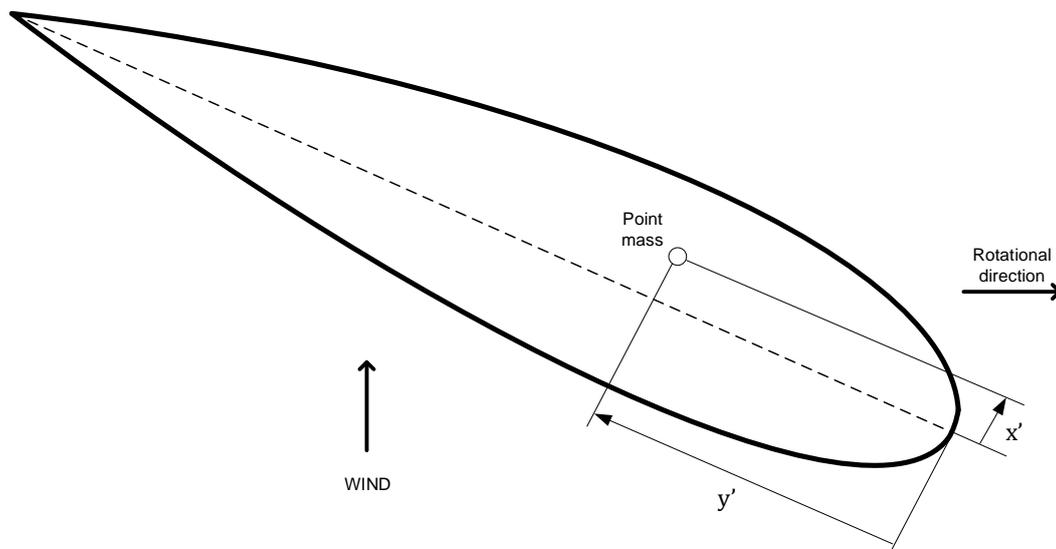


Figure 3-3: Diagram showing the position of point masses relative to the leading edge

Additional pitching inertia should also be specified on this screen.

3.3.3 Vibration dampers

Particularly when operating in stall, the aerodynamic damping of the edgewise vibrational modes of the blades may be very low or even negative, causing large vibrations and corresponding oscillatory loads. To prevent this, some blades are fitted with vibration dampers. These consist of mass-spring-damper system tuned to vibrate at the resonant frequency. Enter the mass, frequency and damping factor (fraction of critical damping), as well as the damper position and the direction of vibration relative to the local chord line. For the direction, the sign convention is the same as for blade twist (see 3.2).

3.3.4 Blade icing

Bladed represents blade ice accretion by adding mass to the blade at the aerofoil leading edge. This additional mass will modify the blade mass totals and mass distribution properties such as polar inertia and centre of mass.

There are two options for the calculation of ice mass on the turbine blades. The first calculation option is according to the specification of Germanischer Lloyd (GL 2010) [9], the second is according to the draft IEC 61400-1 edition 4 standard.

The methodology used for the GL 2010 approach is as follows. The mass distribution increases linearly from zero at the rotor axis to the value μ_e at half the radius and then remains constant up the blade tip. The value μ_e is calculated as follows:

$$\mu_e = \rho c_{min}(c_{max} + c_{min})(0.00675 + \exp(-0.32R))$$

where R is the rotor radius [m], c_{max} is the maximum chord [m], c_{min} is tip chord [m] and ρ is the ice density with default value 700 [kg/m³]. The chord length at the blade tip is an input in Bladed. The maximum chord value is computed using from the blade geometry (see 3.2) information input by the user.

The methodology used for the IEC Ed 4 is to apply a mass distribution that increases linearly from zero at the rotor axis to the maximum value at the blade tip. The ice mass distribution is calculated using:

$$M(r) = \rho C_{85} r$$

where, $M(r)$ is the mass distribution on the leading edge of the rotor blade [kg/m], ρ is a constant parameter with default value of 0.125 [kg/m³], C_{85} is the chord length at 85% rotor radius, and r is the radial position measured from the rotor axis [m]. The parameter C_{85} is calculated by Bladed based on the geometry information input by the user.

For both ice models, the radial distance from the rotor axis is assumed to be a nominal radius i.e. ignoring the effect of rotor cone, blade prebend and blade root mounting angle. The GL 2010 rotor radius R is always half of the "nominal rotor diameter" as shown in the "Turbine and Rotor" screen in the Bladed GUI. Each blade station radial position r can be calculated using the blade root length added to the blade station "distance along blade root Z axis" at each blade station.

Blade icing can be defined on the "Blade Information" tab of the blade screen.

Note that, unlike in version 4.8 and earlier, the change in mass due to blade icing is no-longer reflected in the Turbine Information screen (when clicking on the Mass Totals button). Instead, the mass distribution and blade mass with ice is provided in the verification (\$VE) file that is output when simulations are run.

Ice accretion can also modify the aerodynamic properties of aerofoils. No modification is applied to the aerofoil data that has been input into Bladed. If design standards require these modifications to be made then it is recommended that the user applies these corrections manually to the aerofoil input data.

3.4 Blade mounting orientation

Blade mounting sweep angle: A positive sweep angle means that the blade tip is swept back from the pitch axis, away from the direction of rotation.

Blade mounting cone angle: This has the effect of adding a contribution to the pre-bend deflection which increases linearly with distance along the blade. The blade root Z-axis is coned *upwind* from the pitch axis.

Note: when no (zero) mounting sweep or cone angles are specified, the "blade root Z-axis" is the same as the pitch axis. If there are non-zero mounting sweep and cone angles specified, then there is an orientation difference between the pitch axis and blade root Z-axis. All subsequent blade properties are then defined in relation to the blade root Z-axis rather than the pitch axis.

3.5 Blade stiffness distribution

For calculations involving blade flexing or vibrational dynamics (see [7.1](#)) it is necessary to define the stiffness distribution of the rotor blades. Click the **Stiffness** checkbox **off** if calculations not involving blade flexing are desired and the stiffness data has not been defined.

The stiffness must be defined about the principal axis of inertia at each blade station (see [3.1](#)). The stiffness is the product of Young's Modulus for the material and the second moment of area for the x_p or y_p directions as appropriate. The principal axis orientation is defined as an input, and defaults to the aerodynamic twist. In this case it is assumed to be parallel and perpendicular to the chord line. If the principal axis orientation is different from the aerodynamic twist, click the **Use default principal axis orientation** to **off**. (see diagram below)

To enter or edit data, click on the appropriate entry, or move to it with the arrow keys and press Return.

3.6 Additional degrees of freedom

To enable an axial degree of freedom, torsional degree of freedom or shear stiffness to be defined, click the appropriate check boxes. The following data then needs to be defined:

- **Torsional stiffness:** Required if a torsional degree of freedom is included. Defined with respect to the shear centre.
- **Axial stiffness:** Required if an axial degree of freedom is included.

- **Shear centre (x'):** Required if a torsional degree of freedom or shear stiffness is included. This is the perpendicular distance from chord line to shear centre in local coordinates as a percentage of the chord. (See diagram below)
- **Shear centre (y'):** Required if a torsional degree of freedom or shear stiffness is included. This is the distance along the chord from the leading edge to the shear centre in local coordinates, as a percentage of the chord. (See diagram below)
- **Shear stiffness along x_p :** Required if shear stiffness is included. Defined in the principal axis direction as specified by the principal axis orientation. (See diagram below)
- **Shear stiffness along y_p :** Required if shear stiffness is included. Defined in the principal axis direction as specified by the principal axis orientation. (See diagram below)

To enter or edit data, click on the appropriate entry, or move to it with the arrow keys and press Return.

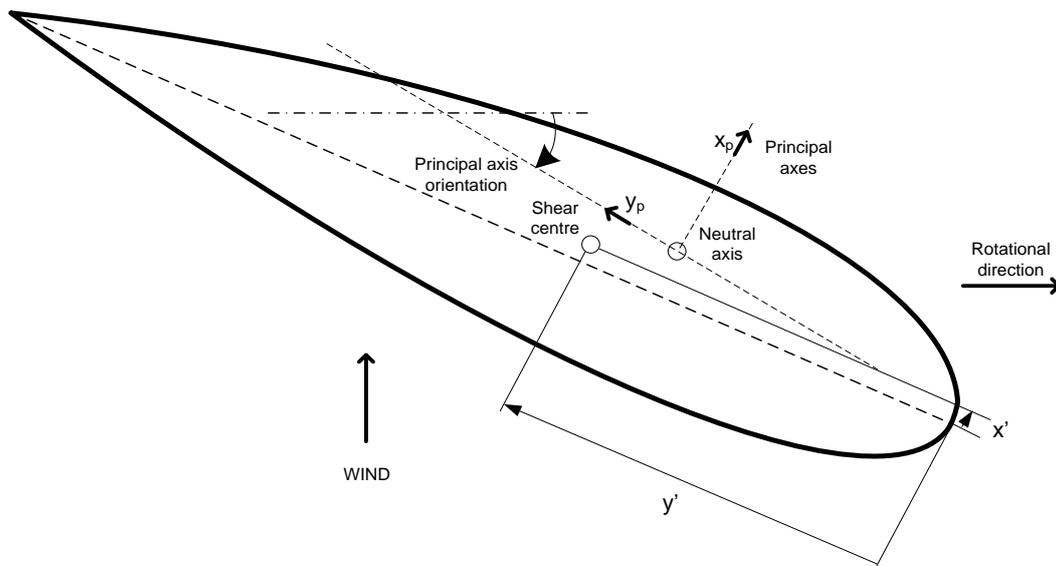


Figure 3-4: Diagram showing the orientation of the principal axes and the location of the shear centre relative to the leading edge

3.7 Conversion from older versions of *Bladed*

If you open a **Bladed** project file from a version prior to version 4.0, **Bladed** will convert the blade into the new format according to the following rules:

Version 4.0 or later	Pre-version 4.0
Distance along blade root Z-axis	Distance from root
Neutral axis (x)	$\text{Chord} * (\text{mass axis} - \text{twist axis}) * \sin(\text{twist})/100 - \text{prebend}$
Neutral axis (y)	$-\text{Chord} * (\text{pitch axis} - \text{twist axis} - (\text{mass axis} - \text{twist axis}) * \cos(\text{twist}))/100$
Neutral axis, local (x')	0
Neutral axis, local (y')	Mass axis
Centre of mass (x')	0
Centre of mass (y')	Mass axis
Shear centre (x')	0
Shear centre (y')	Elastic axis

The neutral axis is assumed to follow the centre of gravity for this conversion

The new blade description should be checked to make sure that the new blade data is correct.

3.8 Aerofoil sections

At each blade station (see [3.1](#)), the aerodynamic characteristics of the blade section must be defined so that lift and drag coefficients may be calculated. Pitch moment coefficients may also be calculated if desired.

Aerodynamic characteristics are defined empirically, by means of look-up tables of lift, drag and pitch moment coefficients tabulated against angle of attack. One such look-up table is referred to here as an aerofoil dataset (see [3.9](#)). A number of aerofoil datasets may be combined to define one **aerofoil section**. Two types of aerofoil section may be defined:

Normal Sections: (see [3.11](#)) for any fixed or pitchable part of blade with no flaps or ailerons. Separate aerofoil datasets may be provided for up to three fixed Reynolds numbers. Interpolation between these datasets will occur depending on the precise Reynolds number at any point in the calculation. For each of the fixed Reynolds numbers, aerofoil datasets may be provided for two different thickness/chord ratios, one either side of the actual thickness/chord ratio for the blade station in question. In this case, a further interpolation will be carried out to give aerodynamic coefficients appropriate to the actual thickness/chord ratio.

Aileron Sections: (see [3.12](#)) for any parts of the blade which have flaps, ailerons, airbrakes or other aerodynamic control (see [4.1](#)) surfaces which alter the aerodynamic profile of the blade section. Separate aerofoil datasets may be provided for a number of different deployment angles of the control surface. Interpolation between these datasets will be used to give aerodynamic coefficients appropriate to the actual deployment angle at any point in the calculation.

3.9 Aerofoil datasets

An aerofoil dataset is a look-up table of Lift coefficient, Drag coefficient and Pitching moment coefficient (optionally) as a function of angle of attack. Such tables are usually obtained empirically for particular aerofoil sections at particular Reynolds numbers, for example in wind tunnel tests.

3.9.1 Defining aerofoil datasets

Bladed maintains a database of aerofoil datasets. Click the **Aerofoil** icon on the toolbar for access to the database. As well as the table of aerodynamic coefficients, each aerofoil dataset is characterised by a set of general data giving the characteristics of the aerofoil section to which it applies, namely:

- **Name:** this is an identifier for referencing the dataset within **Bladed**.
- **Comments:** to identify the source of the data and the full name of the aerofoil section.
- **Thickness to chord ratio** for the aerofoil section.
- **Reynolds number** for which the dataset applies.
- **Pitching moment centre:** the point about which the aerodynamic pitching moment is defined, as a percentage of the chord backwards from the leading edge.
- **Deployment angle** (for aileron sections see [3.12](#)).
- **Include pitching moment:** set to **No** if the pitching moment data is not available. If any dataset used on the turbine does not have pitching moment data defined, then no pitching moments will be calculated for that turbine.

If possible, give angles of attack ranging from -180° up to 180° as the full range will be needed for some calculations.

3.9.2 Importing a dataset

Aerofoil datasets may be imported from an ASCII file of aerofoil data by clicking the **Import** button (see [3.10](#) for the required format of the ASCII file). A standard file selection screen appears. Select the drive, directory and file where the data is stored. If the identifier specified in the file duplicates one which exists in the database, you will be prompted for a different identifier to be used.

A simpler method is to copy and paste the data into **Bladed** as follows.

3.9.3 Adding a new dataset manually or via the clipboard

Click the **New** button, fill in the **General data** to characterise the aerofoil dataset, add comments as required, and give the dataset an appropriate identifier in the **Name** box.

The aerofoil data itself may be pasted in from the clipboard by clicking the **Paste** button. The data should be copied from a spreadsheet or tab-delimited file, and should consist of three or four columns: angle of attack in degrees, Lift coefficient, Drag coefficient, and Pitching moment coefficient if included.

Alternatively, type the data for each angle of attack in the coefficients box, clicking **Add** after each one.

Then press **Save**. New datasets may also be created by editing existing ones, as described below.

3.9.4 Editing an existing dataset

Click the **Load** button to list the names of all currently available datasets. Clicking one of these names displays the main characteristics of that dataset, which can then be selected by double-clicking or pressing **OK**. The data for that aerofoil may then be edited. To add a new angle of attack, enter it in the coefficients window and press **Add**. To modify a datapoint, click on it and press **Edit** to make it appear in the coefficients window where it can be edited. To remove a datapoint, click on it and press **Delete**. When editing is finished, click the **Save** button. To save it as a new aerofoil dataset, modify the General data as appropriate, change or insert any comments in the **Comments** box, change the **Name** as required, and then click **Save**.

The **Copy** button may also be used to transfer aerofoil data via the clipboard to another application.

3.9.5 Removing a dataset

Click the **Delete dataset** button.

3.9.6 Viewing aerofoil data graphically

Select the view required, change the range of angles of attack if desired, and click **View Data**. The graph may be printed or saved as a Metafile or copied to the clipboard for subsequent inclusion in a report.

3.10 Format of ASCII aerofoil files

Aerofoil data is normally pasted into **Bladed** as explained in Section [3.9.3](#). It can also be read in directly from an ASCII file, but it is important to format the file correctly, as follows.

The file consists of a header section describing the aerofoil and the conditions to which the data relate, followed by a table of the lift, drag and (optionally) pitch moment coefficients against angle of attack, and terminated by a label which marks the end of the dataset.

Apart from comment line(s), the header lines each contain an identifier, which must be in upper case, followed by an identifier consisting of up to 32 alphanumeric characters:

REFNUM	up to 32 characters	A unique identifier for this dataset
*		Any comment lines here, starting with an asterisk (*), will be incorporated in the database
*		
*		
*		
XA	<real> %	Pitch moment centre, back from leading edge
THICK	<real> %	Thickness/chord ratio
REYN	<real>	Reynolds number
DEPANG	<real> degrees	Aileron deployment angle
NALPHA	<integer>	Number of angles of attack listed in the table
NVALS	<integer> 2 or 3	Number of coefficients: 2 if just lift and drag coefficients are available, 3 if pitch moment coefficients are also present

The table of coefficients follows, consisting of **NALPHA** lines of data each containing:

Angle of attack (degrees), Lift coefficient, Drag coefficient [, moment coefficient if NVALS=3]

with the values on the line separated by commas, spaces or tabs. Finally, there should be a line consisting of the termination string:

ENDSECTION

Example ASCII aerofoil dataset

A very simple example for an ASCII aerofoil database file, just to illustrate the format required, is given here:

```

REFNUM      SET1
* Example dataset
* Use these lines for a description
XA          25.0
THICK       18.0
REYN        3.E6
DEPANG      0.0
NALPHA      12
NVALS       3
-180        0          0          0
-20         -1         0.3        0.037
-12         -0.88      0.03       -0.067
-8          -0.48      0.0107    -0.078
-4          -0.02      0.0073    -0.089
0           0.45       0.0061    -0.104
4           0.891      0.0069    -0.121
8           1.323      0.016     -0.119
12          1.728      0.0218    -0.116
16          2.119      0.07      -0.111
20          2.478      0.225     -0.112
180.0       0.0        0.0        0.0
ENDSECTION

```

3.11 Defining normal aerofoil sections

A normal aerofoil section should be defined for every blade station which is either:

- **Fixed** (see Blade geometry (3.2)) or
- **Moving** (see Blade geometry (3.2)) and **pitchable** (see Control surfaces (4.1)).

Normal aerofoil sections allow interpolation on up to three Reynolds numbers and up to two thickness/chord ratios - see Aerofoil sections (3.8). A very flexible interpolation scheme is used, so for example it is not necessary for the individual datasets to have matching angles of attack.

A number of different aerofoil sections may be defined, for use at different blade stations.

On the **Blade properties** window, click on the foil section to be defined. A drop-down list allows an already-defined section to be selected, or select **Define...** to open the **Define Aerofoil Sections** window. This allows the characteristics of already-defined sections to be viewed or edited, or new ones created by clicking **New**.

To view or edit an existing aerofoil section:

In the **Define Aerofoil Sections** window, select the section number required.

To set up a new aerofoil section:

In the **Define Aerofoil Sections** window, press **New** to start a new foil section.

To edit an aerofoil section:

Up to six aerofoil Dataset (see 3.9) names may be entered in the boxes provided. Click on any white or red box to open the **Aerofoil Dataset Selection** window. This presents a list of all suitable datasets from the database (they must be in ascending order of Reynolds number and thickness). If desired, the selection criteria may be modified to further restrict the list displayed. Select a dataset from the list and click **OK**.

If fewer than six boxes are required, the boxes used must form a rectangular pattern so that the interpolation scheme is fully defined. The top left hand box must always be used.

To remove an aerofoil section:



There is no need to delete an aerofoil section if it is no longer needed at any blade station. The sections which are still in use will be renumbered when the **Blade Properties** window is closed, starting from 1. The aerofoil datasets themselves of course remain in the database.

3.12 Defining aileron sections

An aileron section should be defined for every blade station which is defined as **Moving** (see Blade geometry (3.2)) if **ailerons** are specified for the aerodynamic control surfaces (see 4.1). Flaps and airbrakes can be treated as if they are ailerons.

Aileron sections allow interpolation between a set of aerofoil datasets (see 3.9) defined for different aileron deployment angles.

A number of different aileron sections may be defined if required, for use at different blade stations.

On the **Blade properties** window, click on the foil section to be defined. A drop-down list allows an already-defined section to be selected, or select **Define...** to open the **Define Aerofoil Sections** window. This allows the characteristics of already-defined sections to be viewed or edited, or new ones created by clicking **New**.

To view or edit an existing aerofoil section:

In the **Define Aerofoil Sections** window, select the section number required.

To set up a new aerofoil section:

In the **Define Aerofoil Sections** window, press **New** to start a new foil section.

To edit an aesection:

Click **Add** or **Insert** to increase the number of aerofoil datasets (see 3.9) to interpolate between. Click **Delete** to remove one. Double-click on any entry to open the **Aerofoil Dataset Selection** window. This presents a list of all suitable datasets from the database (they must be in ascending order of deployment angle). If desired, the selection criteria may be modified to further restrict the list displayed. Select a dataset from the list and click **OK**.

No entries may be left blank.

To remove an aerofoil section:

There is no need to delete an aerofoil section if it is no longer needed at any blade station. The sections which are still in use will be renumbered when the **Blade Properties** window is closed, starting from 1. The aerofoil datasets themselves of course remain in the database.

4 DEFINING THE REST OF THE TURBINE

Having defined the turbine blades, this section describes how to build up a model of the complete turbine structure including the rotor and hub, power train, tower and nacelle.

4.1 Defining the rotor

Click the **Rotor** icon on the toolbar to define the basic characteristics of the turbine rotor and hub. The following data must be defined:

- **Rotor diameter**
- **Number of blades**
- **Hub height:** from the ground to the centre of the rotor (i.e. the intersection of the blade and shaft axes).
- **Tower height:** from the ground or sea surface to the yaw bearing (only needed if the tower itself has not been defined). See also: Tower ([4.5](#)).
- **Hub vertical offset:** the vertical distance from the yaw bearing to the hub.
- **Blade set angle:** the angle at which the blade is mounted onto the hub. More positive values of set angle push the leading edge further upstream. It is usually convenient to define the set angle as zero, but particularly for stall regulated machines, the set angle provides a simple way of rotating the whole blade (both pitching and non-pitching sections) without having to re-define the twist distribution.
- **Cone angle:** the angle between the blade axis and the rotor plane.
- **Tilt angle:** the angle between the shaft and the horizontal (normally positive).
- **Overhang:** the horizontal distance between the rotor centre and the tower centreline.
- **Lateral offset:** the horizontal offset between the shaft and tower axes.
- **Rotational sense:** the turbine may rotate clockwise or anti-clockwise when viewed from upstream.
- **Position:** the rotor may be upwind or downwind of the tower in normal operation.
- **Speed type:** the turbine may be defined as a fixed or a variable speed turbine. See also Generator ([4.10](#)), Control ([5](#)).
- **Control surfaces:**
Specifies whether the blade is fixed, pitchable (includes full or partial span pitch) or aileron controlled (includes any type of flap or airbrake).
See also Blade Geometry ([3.2](#)), Control ([5](#)), Start-up ([5.13](#)), Normal stop ([5.14](#)), Emergency stop ([5.15](#)).
- **Transmission:** select Gearbox or Direct Drive. For a direct drive system, either set the gearbox ratio (see [4.8](#)) to 1, or select the Direct Drive option here. If the Direct Drive option is selected, the generator mass can be separately identified on the Hub screen (see [4.2](#)), otherwise it must be included in the Nacelle mass (see [4.13](#)).
- **Cut-in wind speed:** the steady wind speed at which the turbine is brought on or off line in low wind speeds.
- **Cut-out wind speed:** the steady wind speed at which the turbine is brought on or off line in high wind speeds.

Clicking the **Hub** tab allows the hub configuration to be defined, together with generator mass information if the Direct Drive option is selected. This is described in Section [4.2](#).

4.2 Blade and rotor coordinate summary

In this section, the relationship between blade and rotor properties and coordinate systems are displayed, using views from the side and in front of the turbine.

These views are shown for a clockwise upwind turbine.

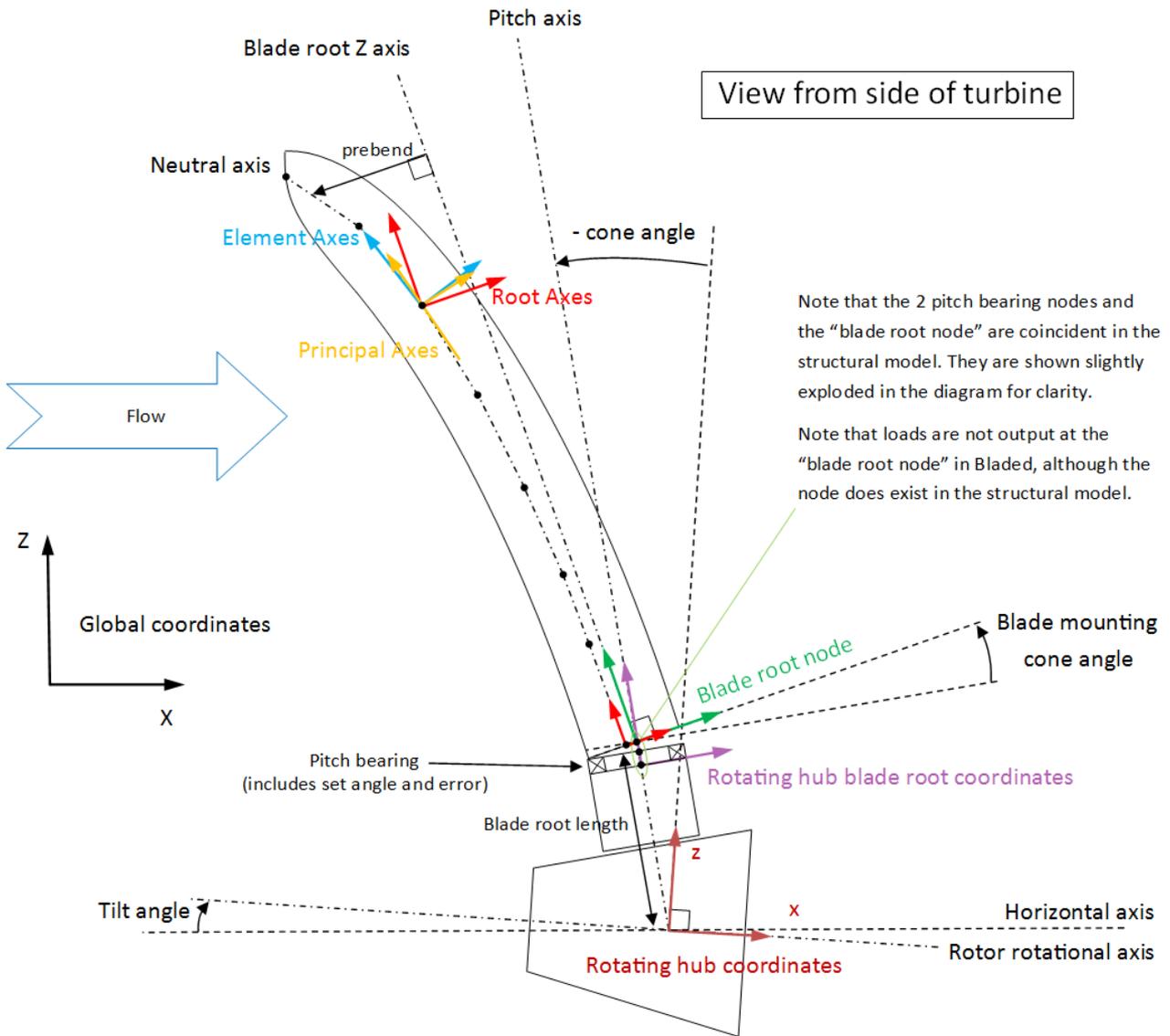


Figure 4-1: Hub and blade coordinate system summary (view from the side of the turbine)

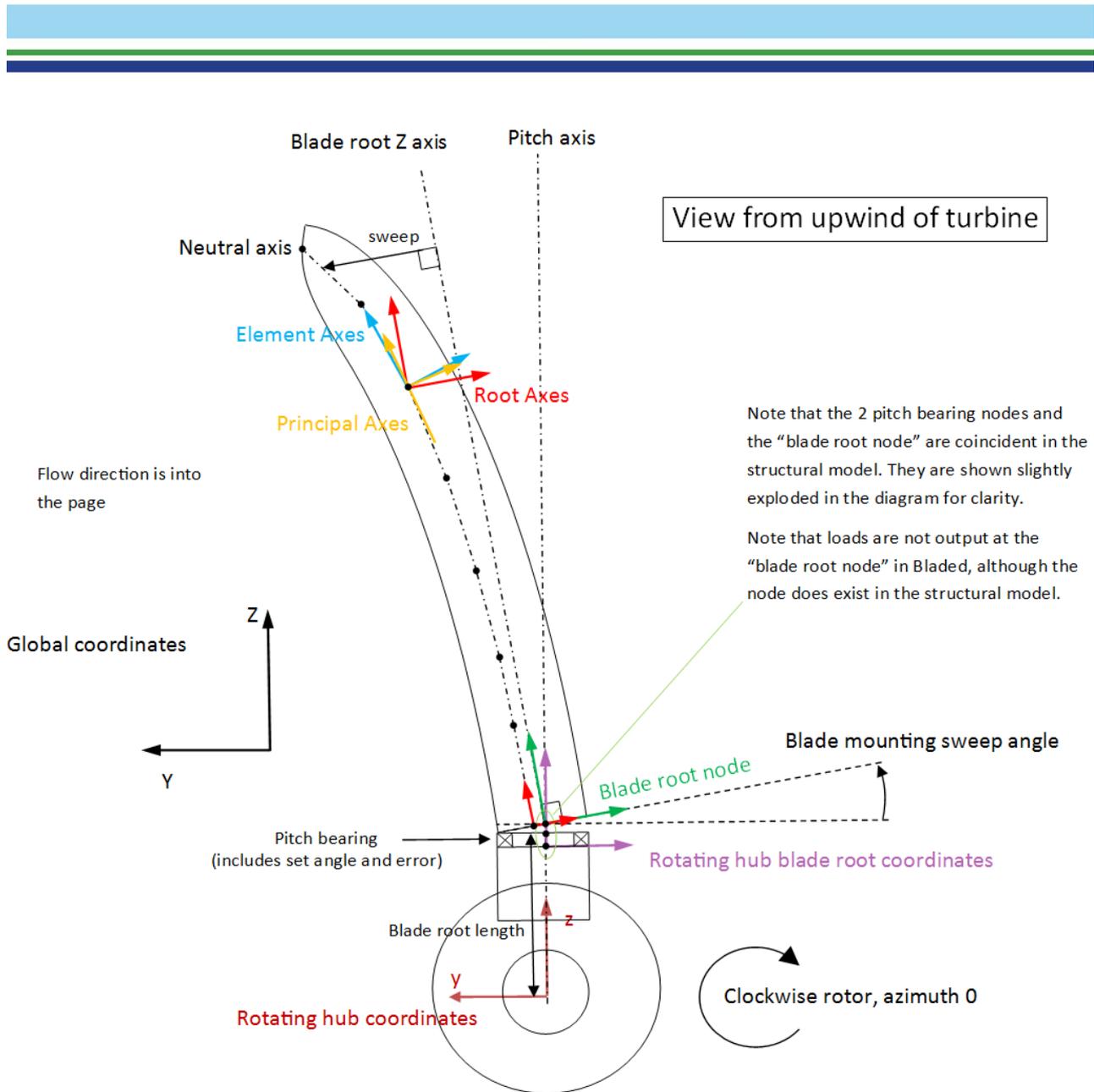


Figure 4-2: Hub and blade coordinate system summary (view from in front of the turbine)

4.3 Defining the hub

From the **Rotor** (see [4.1](#)) screen, click the **Hub** tab to define the hub characteristics.

Enter the **Spinner diameter**. This is the diameter of any spinner or nose-cone, within which the blades themselves experience no aerodynamic forces.

Blade root section

The blade root section connects the blade root to the shaft. It is assumed to be cylindrical in section. It is defined by:

- **Length:** this defines the actual radius of the first blade station (see [3](#)). Set this to zero if there is no root section. The rotor diameter (see [4.1](#)) must be twice the sum of the root section length and the blade length, defined by the radial position of the last blade station.
- **Diameter:** the diameter of the cylindrical tube which forms the blade root section.
- **Drag Coefficient:** the drag coefficient to be used for any part of the blade root section which is outside the spinner. Zero lift coefficient is assumed.

If the spinner entirely covers the root section, the diameter and drag coefficient values will not be used.

For one and two bladed rotors, click the check box to specify a teetered hub if required. See also Options (7.2.3). For teetered hubs, enter a **Delta-3 angle** if pitch-teeter coupling is required. A positive delta-3 angle acts to stabilise teeter by increasing the angle of attack when the blade teeters into the wind. A **Special** setting is available in case other forms of pitch-teeter coupling are required, by special arrangement with Garrad Hassan.

Teeter restraint

If a teetered hub is selected, a teeter restraint system may be specified by clicking the Teeter restraint button (see 4.4).

Mass information

If the Direct Drive option is selected, enter the **moment of inertia of the generator rotor** here.

For dynamic calculations, check the **Mass** checkbox to allow the necessary masses and inertias to be specified as follows:

- **Hub mass:** the mass of the hub, including the spinner and any blade root section.
- **Hub mass centre:** the distance from the intersection of the shaft and blade axes to the centre of mass of the hub, in a direction measured away from the tower.
- **Moments of inertia:** the moment of inertia of the hub mass about the shaft axis must be defined. The inertia about an axis perpendicular to the shaft may also be entered with its origin about the hub centre of mass.

For direct drive systems, enter also the generator mass information as follows:

- **Generator mass**
- **Generator mass centre:** the distance between the centre of mass of the generator and the centre of mass of the hub.
- **Moments of inertia:** the moment of inertia of the complete generator may be defined, about the shaft axis and a perpendicular axis.

In this case, ensure that the mass information entered on the Nacelle (see 4.13) screen does not include these contributions.

For one-bladed rotors only, define the counterweight as follows:

- **Mass:** the mass of the counterweight.
- **Inertia about shaft:** the moment of inertia of the counterweight about the shaft axis.

Note

The **Turbine Information** window shows the total masses and inertias of all turbine components whose mass characteristics are defined. Click **Mass Totals**, or use the **Windows** pull-down menu on the main toolbar to open this window.

4.4 Teeter restraint

A spring and damper model of a teeter restraint system is provided. If a teetered rotor has been specified, click the **Teeter restraint** button on the **Hub** screen, select **Standard model**, and enter the following parameters:

- **Free teeter angle:** The teeter restraint only start to act when this teeter angle is exceeded.
- **Spring preload:** The torque required to start moving the teeter restraint.
- **Spring stiffness:** The rotational stiffness of the teeter restraint.
- **Teeter damping:** The rotational damping of the teeter restraint.

If the teeter hinge has a delta-3 angle, the teeter restraint is on the flapwise motion resolved through the delta-3.

Click **Special** if a client-specific teeter restraint model has been provided by arrangement with Garrad Hassan, and enter the appropriate parameters.

See also: Calculation options ([7.2.3](#)).

4.5 Defining the tower

Click the **Tower** button on the toolbar to define the tower characteristics. Two types of tower model are available. These are the tubular axisymmetric (or monopile) and for users with an Offshore Support Structure module, the multi-member model.

4.5.1 Monopile Tower

The tower is modelled by defining its characteristics at a number of stations from the tower base to the tower top. Tower properties must be defined for at least two tower stations (the tower base and tower top), although if tower vibrations (see [7.1](#)) are to be modelled, a minimum of 5 stations is recommended in order to achieve a reasonable degree of accuracy.

Check the **Tower geometry** check-box to define the tower dimensions. These are used to calculate the tower shadow (see [6.3](#)) and windage loads.

Click the **Add** button to add a new tower station. Stations are automatically sorted by height. To highlight a station, click on the station number or a graph point. Click **Delete** to remove a highlighted station.

To enter or edit data, highlight the data item by clicking on it, or by moving to it using the arrow keys and then pressing the Return key. Press Escape to restore the previous value.

Highlight a block of cells by dragging the mouse. These cells may be copied to the clipboard, or the contents of the clipboard may be pasted in. In this way, data from a spreadsheet or a tab-delimited ASCII file may be directly pasted in. Click **Undo** to reverse a paste operation.

Enter the tower station height and tower diameter for each station. In the case of an onshore (see [4.5.5](#)) turbine the station heights are defined relative to ground level, whilst for an offshore (see [4.5.5](#)) turbine, station heights are defined relative to the mean water level.

If the first tower station is above the ground or sea bed level, the tower is assumed to be mounted on a rigid pedestal, as indicated on the **Tower Geometry** diagram. If the height of the first tower station is below the ground or sea bed level, the tower foundation is assumed to be buried as indicated, and no external forces are assumed to act on the buried portion of the tower.

For all cases, the height of the top tower station must correspond to the tower height defined on the Rotor (see [4.1](#)) screen.

The **Show** button controls the graphical display of tower properties. The graph is updated as soon as any new values are assigned, giving an instant indication if faulty values are entered. Controls are provided to print the graph or save it to a Metafile, which can subsequently be incorporated into a report.

Check the **Mass** check-box to allow the mass per unit length to be entered at each tower station. This is necessary for correct calculation of the tower base loads, and also for modelling tower vibrations (see [7.1](#)), in which case it is also necessary to check the **Stiffness** check-box and supply bending stiffness values at each station. These are defined as the product of the Young's modulus and the second moment of area. Check the **Shear flexibility** check-box to include effects of shear flexibility and to allow entering the shear stiffness at each station. These are defined as the product of the Shear modulus and the shear area that equals half of the cross-section area for a circular cross-section. Check the **Torsional degree of freedom** check-box to allow the torsional stiffness and polar moment of inertia to be entered at each station.

Alternatively, for a tower of circular cross-section, the mass and stiffness distributions can be calculated automatically by entering the wall thickness at each station. Type the name of a material and enter its density and Young's modulus in the boxes provided. If the torsional degree of freedom or the shear flexibility is included, the Shear modulus should also be entered. A number of different materials may be entered if desired. Then select the appropriate material for each station.

If a discontinuity is to be defined, for example a step change in wall thickness or other property, simply enter two stations at exactly the same height to define the discontinuity. Do not enter very closely spaced stations.

4.5.2 Multi-member tower

Users with an Offshore Support Structure module licence can define the tower and support structure as a series of nodes, joined by a series of members.

Click the **Add Node** button to add new nodes. The height, and x and y positions of the node should be specified. The x-axis initially points south, but can be rotated by entering a non-zero value in the **X-axis clockwise from south** box. Point masses can be defined at any of the nodes. The node at which the nacelle joins the tower must be selected from the Nacelle node drop-down menu. The selected node must be the highest node on the structure.

Clicking **Add Member** will allow members to be defined to join the nodes. The grid will display two rows per member, one for each end of the member. Most parameters can be different at each end, but some must be the same, for example the material properties and whether or not the member is flooded. Selecting the **Allow tapered wall thickness** option will enable the wall thickness to vary along the member. Clicking on the materials column will bring up a drop down list of all the materials defined. Select the appropriate material for that member. If a diameter, wall thickness and material have been defined, the various mass, stiffness and inertia properties will be calculated automatically. These can be overwritten if desired. The diameter is assumed to be the outside diameter for the purposes of mass, stiffness, inertia and drag calculations.

Marine growth (fouling): For an offshore turbine, enter a value for the thickness in millimetres of marine growth (fouling) at each end of a member, if required. The thickness is assumed to vary linearly along the member. The increase in member diameter due to the marine growth affects both the drag on the tower member and its mass. Note that the additional mass due to marine growth is *not* included in the tower mass reported in the **Turbine Information** window.

When the multi-member tower option is selected, each member is defined with its own local coordinate system. The local x-axis is defined along the member axis, in the direction from End 1 to End 2. Thus Fx is the member axial force and Mx the torsional bending moment. Clicking on the **Member axes** button will bring up the direction cosines for the member z-axis. These will have defaulted to an orientation in line with the convention for other offshore engineering codes. The default orientation in **Bladed** is to set the local member y-axis in the horizontal plane, with the local member z-axis making up a right-handed coordinate system. For a vertical member the default orientation is for the local y-axis to be in the global x-direction, with the local z-axis in the global y-direction. Thus for a vertical member Mz describes the principal overturning moment and My the side-side overturning moment. These orientations can be changed by the user if required.

Clicking on the **Display Structure** button will bring up a 3D drawing of the support structure. It is possible to rotate, pan and zoom using the appropriate toolbar options. Clicking on the green "N" toolbar option will label the nodes in the display, and the "M" will label the members. If members are selected in the **Member** grid whilst the plot is open, the selected members will be highlighted in red and the local coordinate systems displayed.

4.5.3 Geometric Stiffening

The elastic stiffness of the tower is constant. However, in reality the stiffness of a structure will vary depending on the axial and shear loads. The change of stiffness of the structure depending on these loads is the **geometric stiffening**. This effect is important for blades, and is included by default, but is not always important for support structures. Enabling "Use geometric stiffening" will give more accurate results but may result in slower simulations.

4.5.4 Flanges and point masses

Flanges should be modelled as point masses, not by an increase in wall thickness between closely-spaced stations or nodes. For the monopole tower model, Click the **Point masses** button to define the

height and additional mass of each flange or other point mass. For the multi-member tower model, point masses can be added at any node by entering the value directly into the nodes grid.

4.5.5 Vibration dampers

Click the **Vibration damper** button to define a mechanical vibration damper, characterised by its **mass** (M), the **resonant frequency** (ω) to which it is tuned, and the **damping factor** (ζ , fraction of critical damping). Enter also the **damper position**: the number of the node or station at which it is attached. A damper is usually **Omnidirectional**, but a **Unidirectional** damper may also be defined: this would be mounted in the nacelle, and the **Damper angle** specifies its direction of motion relative to the fore-aft direction.

The damper is modelled by the **Bladed** multibody dynamics as a mass-spring-damper (M-K-C) system, with properties defined as follows:

$$\begin{aligned}M &= M \\K &= \omega^2 M \\C &= 2\zeta\sqrt{KM}\end{aligned}$$

The stiffness and damping values can be specified as constants or using a look-up table. The look-up table allows a non-linear vibration damper system to be modelled. Please see the Project Info User Guide for more information [7] on how to specify the look-up tables.

In the "Tower damper" output group, **Bladed** outputs the displacement and velocity of the tower damper *relative* to the tower node to which the damper is attached. Note that as the tower node rotates due to modal deflection or rigid body rotation, the coordinate system in which the displacements and velocities are reported will also rotate.

4.5.6 Support structure superelement

For offshore turbines, the jacket support structure is sometimes modelled as a superelement. The superelement can be included as a component in the multibody framework in a similar way to the other flexible components. Full details of Bladed's superelement functionality can be found in DNV GL document "Superelement Support Structure - User Guide for Bladed" document number 110052-UKBR-T-37

4.5.7 Environment

Click either **Land** or **Sea** in the **Environment** panel to define whether the turbine is sited onshore or offshore. The **Tower Geometry** diagram will then show either the ground level or the sea bed and mean water level, as appropriate. For an offshore turbine, enter the mean sea depth. To change the mean water depth relative to the base of the tower, for the monopile tower model change the **Depth of first tower station** as appropriate, and for the multi-member model, change the **Z-axis origin above mean water level**. Note that for individual calculations, the actual sea depth can be changed by specifying the tide height (see [6.14](#)).

Enter the aerodynamic drag coefficient to enable aerodynamic forces and damping to be calculated. For an offshore turbine, enter also the hydrodynamic drag and inertia coefficients, to enable wave and current forces and hydrodynamic drag to be calculated, or specify them individually for each tower station or member. For a description of wave and current loading, see the Theory Manual.

For an offshore multi-member turbine only, specify also the density of any marine growth. If any members have non-zero marine growth thickness specified, this density is used to calculate the additional mass due to marine growth.

Note

The **Turbine Information** window shows the total masses and inertias of all turbine components whose mass characteristics are defined. Click the **Mass totals** button or use the **Windows** pull-down menu on the main toolbar to open this window. Marine growth is excluded from these mass totals.

4.5.8 Hydrostatics of Isolated Elements

Hydrostatic forces are calculated by considering the pressure forces on structural elements in isolation. Support structure members can be defined as “sealed” or “unsealed”. “Sealed” elements can be defined as “flooded” or “unflooded”. These terms are defined in this section, illustrated in [Figure 4-3](#), and summarised in [Table 4-1](#).

4.5.8.1 Sealed vs. Unsealed members

The “sealed” or “unsealed” options represent whether the ends of the members are sealed with a plate. This results in different pressure forces being applied to the elements.

Pressure forces on “sealed” elements:

- Element end pressure forces act on the whole element end cross-sectional area calculated using member outer radius at each end.
- Distributed forces on the element sides act only on the outside surface of the member
- For an element in isolation, the buoyancy force corresponds to the mass of water displaced by metal and the space inside the members.

Pressure forces on “unsealed” elements:

- Element end pressure forces act on the element wall cross-sectional area at each end, calculated using the element radius and thickness.
- Distributed forces on the element sides act on the inside and outside surface of the member
- For an element in isolation, the buoyancy force corresponds to the mass of water displaced by the metal only.

“Unsealed” elements are assumed to be full of water up to the water line. In this case, the inertia of water inside the element is accounted for by including hydrodynamic *added mass*, which acts only in a direction perpendicular to the element axis.

4.5.8.2 Flooded vs. Unflooded members

“Sealed” elements can be specified as “flooded” or “unflooded”. This determines whether the sealed element contains water. For flooded members, the mass of enclosed water is added directly to the element structural mass.

The hydrostatic pressure forces on “flooded” and “unflooded” elements are identical.

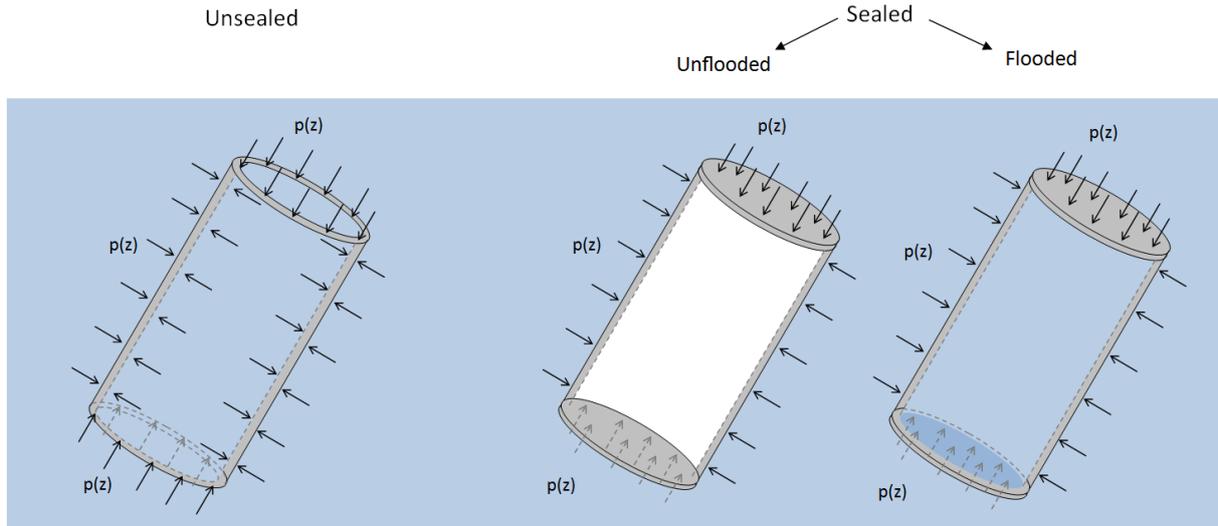


Figure 4-3: Sealed and unsealed structural elements

	Unsealed	Sealed	
		Unflooded	Flooded
End point forces act on	wall area at element end	whole cross-section at element end	

Side wall forces act on	inside and outside of walls	outside wall only	
Enclosed water mass?	Hydrodynamic added mass perpendicular to element axis	None	Structural mass of water added to element
Mass of surrounding water	Included as hydrodynamic added mass		

Table 4-1: Summary of hydrodynamic and water mass assumptions

4.5.8.3 Hydrodynamic mass in simulations

Structural mass and hydrodynamic added mass are treated separately in the Bladed simulations.

For Modal Analysis, the hydrodynamic added mass is not included. For flooded sealed elements, the mass of water enclosed within the structural element is added to the structural mass, so does affect modal analysis.

In a time domain simulation and linearisation calculations, the hydrodynamic added mass is included as modification to the system total mass matrix (i.e. structural + hydrodynamic mass). The hydrodynamic added mass changes at each time step depending on which parts of the structure are submerged.

Note that the additional mass due to enclosed water, and any hydrodynamic added mass, are not included in the tower mass reported in the *Turbine Information* window.

4.5.9 Foundations

The properties of the foundation can have a significant influence on the vibrational properties of the support structure.

Foundations for the monopile tower

For the monopile model, a simple foundation model can be specified at the bottom tower station. Use the check boxes to specify whether translational and/or rotational motions at the tower base are to be taken into account. If translation is selected, enter the foundation mass and translational stiffness in the appropriate boxes. If rotation is selected, enter the foundation moment of inertia and rotational stiffness. If neither is selected, a rigid foundation is assumed.

Foundations for the multi-member tower

For the multi-member tower, a node can be specified as a foundation node by clicking on the appropriate column in the nodes grid. If the entry is left blank, there is no foundation at this node. If **Rigid** is selected, this node will be rigidly connected to ground. If there are any flexible foundation types defined, these will also be available to choose from.

To define these foundation types, click **Foundations...** to open a form which allows you to define an arbitrary number of foundation types (corresponding to different types of rock, soil etc). Start by creating a named foundation type, either by clicking **New**, or **Copy** followed by **Rename**. Each foundation type allows up to 6 degrees of freedom (DOFs) if required, i.e. three translational and three rotational motions. Use the check boxes provided to activate the required degrees of freedom.

The foundation properties are defined by specifying the stiffness, damping and mass matrices. Depending on the selected DOFs, the appropriate matrix cells will be activated. By default, all three grids have their lower-left diagonal greyed out, indicating that the matrices are assumed symmetric. You can change this by clearing the All matrices symmetric checkbox, which will remove the greying-out (at least in the rows and columns that have their DOFs enabled). In this mode you can enter lookup tables or single values on both sides of the diagonal line.

For the mass matrix, enter the appropriate values for any additional mass or inertia which moves as the foundation deflects. For the stiffness and damping matrices, any active cell can contain either a numerical value of stiffness or damping, implying a linear spring or damping characteristic, or a look-up table implying a non-linear relationship between that particular load and that particular deflection or velocity. Use the **Single value / Look-up** table option buttons to define the type of relationship for each cell. If a lookup table is required, the cell will contain a small graph icon, and the look-up table itself will appear in the grid provided, where it can be edited.

Entering P-y curve data in the lookup table

Typically, foundation data is specified as distributed stiffness P-y curves in which P = lateral resistance per unit pile length (N/m), y = lateral displacement (m). This can be interpreted as a set of 'distributed springs' surrounding the pile.

However in **Bladed**, lookup data is entered as equivalent 'point springs' i.e. as pairs of lateral displacement values(m) and resistance values(N). This difference is illustrated in the diagram below:

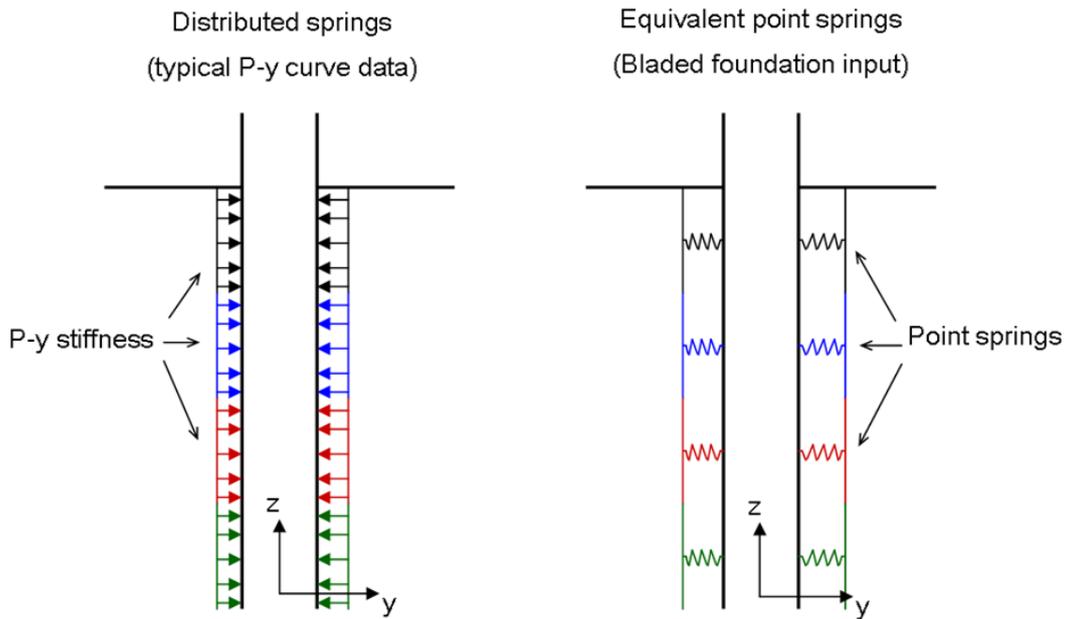


Figure 4-4: Distributed springs vs. equivalent point springs

The 'point spring' lookup data may already be known by the user. Alternatively, approximate 'point spring' lookup table data can be derived from standard 'distributed springs' P-y curve data. A simplified method to perform this conversion is to multiply the P values by the depth of the soil layer. The simplest situation is when each soil layer is represented by a single tower node positioned at the centre of the layer. The conversion method for this situation is illustrated below:

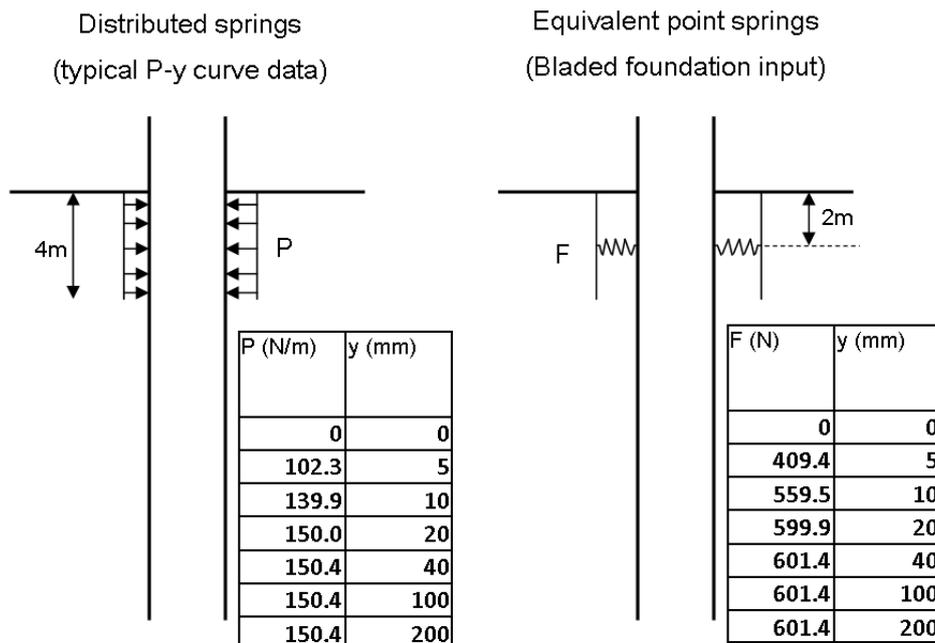


Figure 4-5: Spring conversion method

Particular care should be taken if multiple tower nodes fall within the same soil layer, or if tower nodes are positioned at a boundary between two different soil layers, rather than at the centre of a soil layer. In such situations, the conversion method would be more complex than that shown above.



When using the lookup table, **Bladed** calculates an initial slope in units of N/m based on the lookup table values (F/y). This is different to the initial stiffness (P/y) on the original P-y graph because it takes into consideration the length of pile that has been lumped into each 'point spring'.

Lookup tables usually start from zero deflection or velocity, in which case they are assumed symmetrical about zero. If appropriate, non-symmetrical tables can be defined by entering both positive and negative values of deflection or velocity. In this case, for the diagonal terms in the stiffness matrix, it is usual that an increasing positive displacement causes an increasing positive force and an increasing negative displacement causes an increasing negative force. This ensures that the stiffnesses of the diagonal terms are positive values at all displacements (which should always be the case for a real foundation).

To display the currently-selected lookup table as a graph, click the button below the grid. The graph will also reflect your choice of linear or cubic interpolation, which you can set using the option buttons above, so that you can preview exactly what foundation characteristics will be used in load calculations. The choice of interpolation type is a global setting: it applies to all foundation definitions in the model.

In many cases, the foundation's behaviour will be assumed to be symmetrical in the X and Y directions. If the **Symmetrical in X and Y** box is checked, whenever any additions or edits are made to one cell, other cells are automatically changed where appropriate to maintain the symmetry. The symmetry also means that some of the off-diagonal cells are forced to be zero. (Note: for some of these cells, it is also theoretically possible to maintain symmetry if they are defined as look-up tables with zero initial slope; to achieve this it is necessary to clear the **Symmetrical in X and Y** checkbox).

Foundations are specified in the global coordinate system and do not rotate with the "X-axis clockwise from south" value.

Compatibility of old project files with multi-member tower and non-rigid foundations:

For a turbine model that was created by **Bladed** version 3.80 or earlier, the foundation properties will appear as foundation type 'DEFAULT'. Saving such a model with for use by **Bladed** version 3.80 or earlier is only possible if:

- Only one foundation type is defined
- All damping terms are zero
- X and Y degrees of freedom are defined, but Z degrees of freedom are switched off
- The stiffness and mass matrices are diagonal (no cross-terms)
- The foundation is symmetrical in X and Y.

4.5.10 Moorings

Note: For users of *Bladed* 4.6 and later, if you are creating mooring definitions from scratch, it is recommended that you use the *PreProcessed catenary mooring* rather than the manually defined type. The manual type is still useful if you need to model mooring line damping and/or inertia, but the *PreProcessed* is more accurate and allows a greater variety of mooring configurations to be modelled.

Users with an Offshore Support Structure licence in *Bladed* 4.1 or later versions have the option of modelling a turbine which is floating rather than fixed to the seabed. If the **Sea** option is selected in the **Environment** panel, the **Floating** checkbox will appear in that panel. Tick this box to enter data for a floating turbine. When the **Floating** box is ticked, the **Foundations** column in the node grid changes to a **Moorings** column and the **Foundations...** button changes to **Moorings...** The Modal **reference node** textbox is also present for a floating turbine. This allows you to specify which of the support structure's nodes will be used as a reference for modal analysis. This node will have zero deflection; it must not be the same as the nacelle node. Note also that it is not possible to select the RIGID option in the **Moorings** column of the node grid, as all real moorings will experience some deflections. If you specify the turbine as floating, you must nominate at least one node to be attached to a mooring line (once you have defined moorings as described below). Attaching a node to a mooring line is done by selecting the mooring line from the dropdown in the Moorings column of the grid.

Once you have associated one or more nodes with mooring lines, you need to specify the positions of the mooring anchors on the seabed (version 4.3 onwards). Do this by clicking the **Anchor points...** button, which will bring up a small form containing anchor positions. There is always exactly one anchor per mooring; positions are expressed in global coordinates, with anchor Z being measured relative to the mean sea level (positive Z direction is upwards).

Clicking the **Moorings...** button allows you to create, alter or delete mooring definitions. Any number of moorings can be added. The procedure for this is similar to that for foundations in the non-floating case (see 4.5.8). One difference from the foundations screen is that the degree-of-freedom checkboxes are absent for the moorings case. This is intended to reflect the fact that physical moorings cannot really be constrained in the way that foundations can be. Another difference is that you can choose to enter moorings either as catenary (2 types), tension-leg, or dynamic moorings. It is not currently possible to mix the four types of mooring on the same turbine, so once you have started entering data for one type, *Bladed* will prevent you from switching to any other type. If you really want to switch, you will need to delete all the mooring definitions. The final difference is in the coordinate systems; foundation characteristics are expressed in global coordinates, whereas moorings are not. See below for more details.

Click the **Catenary (manually defined)** option button to enter data for catenary moorings. A catenary mooring definition is entered in much the same way as a foundation definition (subject to the differences indicated above). The **tension-leg** type of mooring definition, by contrast, has no non-floating equivalent. Click the **Tension Leg Moorings** option button to see the data entry screen for this type of mooring. The values which can be entered are axial damping, axial stiffness, torsional damping, torsional stiffness and polar inertia. The textboxes for entering these values will be greyed out until you create the first mooring definition.

From *Bladed* v4.7 onwards it is possible to specify a nominal pre-tension T_{pre} for tension-leg moorings. This results in a reduced value for the unstretched length L_u of the mooring line, according to the following relationship:

$$L_u = \frac{kL_s - T_{pre}}{k}$$

where k is the Hooke's law axial stiffness of the line, and L_s is the length of the line when the turbine is in its nominal starting position, i.e. with the support structure's local origin at the global origin.

Note that the actual tension in the line at the start of the simulation will be less than T_{pre} , because the line tension will make the turbine sit lower in the water, thereby reducing the stretch of the line. However, any desired initial tension can be achieved using an iterative approach, simply by adjusting T_{pre} .

Catenary (PreProcessed) moorings are handled somewhat differently. The underlying data, which is a pair of force / displacement lookup matrices, is not entered here, but requires a separate preprocessing step to generate it. See the PreProcessing User Manual for more details. If you select the



“PreProcessed from file” option button, the Mooring form changes to 2D Lookup configuration. From here you can load in a previously-created mooring definition, by clicking the **Load Mooring** button and then browsing to a *.mooringsoutput file. The shape of the mooring line at equilibrium will be displayed, along with the parameters used to create the tables. The parameters cannot be edited here – if you wish to alter a 2D Lookup mooring you will need to use the PreProcessor. Also displayed is the grid of (x,z) points on which the force lookup matrices are based. To create a new 2D Lookup mooring, click the **Create Mooring** button. This launches the PreProcessing application.

Note that, at present, PreProcessed moorings model only stiffness, not damping or inertia.

Dynamic Moorings are the newest type, intended for modelling catenary-type moorings (including those with inline floats or clump weights), and are available in **Bladed** version 4.8 onwards. They are the most accurate mooring model type, with mooring lines being represented by rigid elements (segments) connected by universal joints. These structural elements form an integral part of the Multibody Dynamics model which is used to represent the entire turbine. In this way the physical properties of the mooring lines are modelled in a fully dynamic manner. In addition, the mooring lines and their interactions with the environment are fully coupled with the rest of the **Bladed** model. The interaction of the lines with the anchors and seabed is explicitly modelled, as are the hydrostatic and hydrodynamic loading on the lines. The hydrodynamic calculations take into account both the motion of the lines themselves, and motion of the water due to waves and currents. It is possible to model complex mooring arrangements featuring sections of different cable properties, as well as floats and clump weights at arbitrary points along the line.

To create a mooring line of this type, click on the **Dynamic Moorings** option button and then click **New**. It is then possible to enter some anchor and seabed properties which apply to the mooring model. Also, one or more line **sections** must be added and then populated. A line section is a region of the line with constant properties such as weight, diameter, etc. Also, an inline float or clump weight – if present - will need to be added at a joint between one section and the next, so at least two sections will be required in this case. The float/weight will appear at the end of the section furthest away from the anchor. Setting the “Buoyancy / weight at end” to a negative value creates a weight, while a positive value creates a float. Increasing the number of segments used to model the section will give better modelling fidelity, but at some cost in terms of simulation time. The default number is 10 segments per section, which is normally a reasonable trade-off.

4.5.10.1 Mooring line coordinate system

This section should be consulted when manually entering data for Catenary moorings.

It is often necessary to model floating turbines which are held in place by several identical mooring lines radiating out in different directions. Previously, catenary mooring descriptions were entered in the global coordinate frame, which normally meant it was necessary for each mooring to have a unique description even when they differed only in their orientation. For this reason, starting in **Bladed** version 4.3, the kinematics and loads set out in the mooring definitions are related to the mooring line’s own coordinate system. This system is illustrated in the diagram below. Each mooring line has an origin at the anchor point with the z-axis of this coordinate system being vertical (as is the global z-axis). The orientation about the z-axis is such that the xz plane of the mooring coordinate system contains both the anchor point and the attachment point (p).

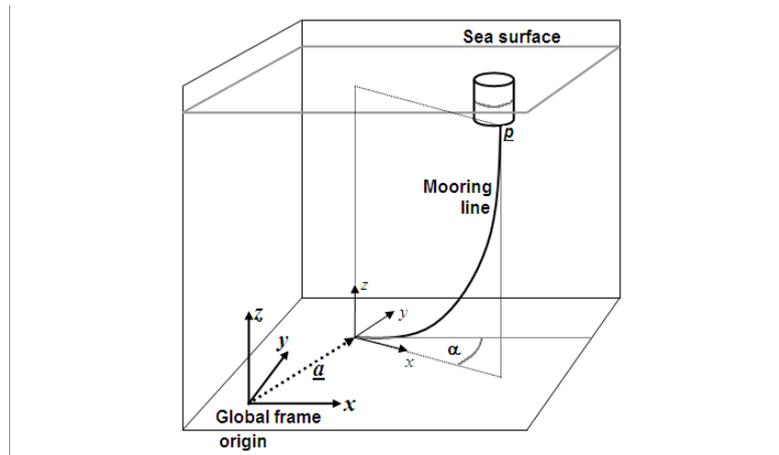


Figure 4-6: The mooring line layout, illustrating the mooring line coordinate system, which has an origin at the line anchor point.

For each mooring line, the displacement and velocity vectors describing the kinematics of the attachment point, \underline{p} , are expressed in the mooring line coordinate system. The displacement vectors are measured relative to \underline{p} 's equilibrium position, which is its steady-state position in the absence of waves. Because the mooring coordinate frame always rotates with the mooring, the y -coordinate of \underline{p} in mooring coordinates is always zero, as is the y -velocity. For this reason the second (Y) column of both stiffness and damping matrices is greyed out.

The use of this coordinate system means that you will only have to enter one mooring definition, unless the moorings are actually different in their characteristics such as line length, line mass etc. You can then use that mooring definition several times over, if there are multiple moorings present in the model. Mooring-based coordinates should also make it more straightforward to calculate the values for populating the lookup tables in the Catenary (manual) case. However, you will have to enter some additional information about anchor positions for each actual mooring. This information is entered via the **Anchor points** form described above.

4.5.10.2 Compatibility of mooring information with pre-4.3 versions

As explained above, moorings in the current version of **Bladed** have anchor positions specified and are expressed in mooring-based coordinates. This is not the case for pre-4.3 versions, so there are some compatibility issues when saving a project as an older version, or when loading in an older project to the current version. When saving as an older version, the anchor points will be discarded, and you will be informed of this. If catenary moorings are used, **Bladed** will also transform the stiffness and damping matrices to global coordinates so that they are appropriate for an older version of **Bladed**. When upgrading an older project file, you will receive a warning message box telling you about the coordinate change, and reminding you that anchor points are currently undefined. The anchor positions will have been given default values, which will generally not be correct, so you will need to edit them. You will almost certainly also need to change the stiffness and damping matrices to reflect the fact that they are now in a different coordinate system. **Bladed** will make no attempt to transform these matrices, because at the time of upgrade it lacks the anchor positions needed to calculate the transformations.

4.5.10.3 Option to auto-populate the stiffness matrix (Catenary manual moorings only)

For version 4.4 and later, users have the option of having the stiffness matrix generated automatically by **Bladed**, after entering a small amount of basic information about the mooring line. This should make it easier to model catenary moorings. **Note: this facility is still available in Bladed 4.6 and later, but it is recommended that you use the more accurate PreProcessed moorings instead.** If you still prefer to use this option, e.g. for backwards compatibility, it can be done as follows: Tick the **Auto-populate stiffness matrix** checkbox in the **Mooring Library** window (if not already ticked). Click the "plus" icon next to the label **Line sections** to create a simple mooring line, and click the arrow to the left of the "1" to expand it. Enter the length, in metres, of the line and its submerged weight (i.e. weight minus buoyancy) per unit length in Newtons. You will also need to specify the horizontal and vertical spans. These are the distances from the anchor to the fairlead (the point at which the end of the line is

attached to the floating structure). Once there are valid values in all four of these boxes, the **Run auto-population** button will become enabled. Click this button and you will see that lookup tables appear in four cells of the stiffness table. (Any lookup tables or single values that were previously in these cells will be overwritten.) Clicking on any of these table icons will show the contents of that table in the **Look-up Table** data grid. You can manually alter this data, but bear in mind that running auto-population again will overwrite it without warning. You can also see the profile of the mooring line you have just defined, by clicking the **View Line Shape** button (which becomes **Refresh Line Shape** when the profile is currently visible). Click the button again after changing any of the line parameters in order to see the effects of the change.

In generating the line shape and the lookup tables, **Bladed** assumes that the line is inelastic. Thus if you enter a line length that is shorter than the straight-line distance from anchor to fairlead, it will fail to give any results. A line which is slightly longer than the straight-line distance will also fail, if it is so short that the line is not lying flat on the seabed at any point. This is because such a line is stretched too taut for the inelastic assumption to be accurate, and thus the results would have been incorrect.

4.5.10.4 Double-accounting of mooring forces (Catenary moorings only)

This is an important difference between version 4.4 and earlier versions. In essence, 4.4 takes care of any double-accounting problems for you, whereas in earlier versions you needed to be aware of the problem and take steps (where necessary) to correct it.

In essence, the problem is as follows. It occurs whenever there is more than one lookup table on a single row of a stiffness or damping matrix. (This will always happen where Auto-populate has been used, for example). It is easiest to follow with the help of an example. Take the case of a stiffness matrix with tables in columns 1 and 3 on the top row, for calculating X forces:

Row 1, column 1:

X Displ (m)	X Force (N)
-100.0	-3.85E+05
0.0	-3.41E+05
100.0	-2.89E+05

Row 1, column 3:

Z Displ (m)	X Force (N)
-50.0	-4.06E+05
0.0	-3.41E+05
60.0	-3.24E+05

The value of X force at zero displacement in both tables should always be the same, because the zero-displacement positions in both tables actually refer to the same situation, where the fairlead is exactly in its starting position. The horizontal force due to the mooring in this situation is -341kN. However, **Bladed** needs to add the contributions from both lookup tables together in order to model the influence of both vertical and horizontal displacements. In pre-4.4 versions it did this by simply adding the force contributions from all the lookup tables in (for instance) row 1, to give a total figure in our example of $-341 + (-341) = -682$ kN horizontal force at zero displacement. This is twice as high as it should be, and the user needed to apply a force offset to all but one of the tables in each row to correct the problem. In version 4.4, however, **Bladed** corrects the problem internally as part of the force calculations. If you try to do the correction yourself, **Bladed** will display one or more warnings in the output window when you run the simulation, of the form ***** WARNING: Not all equilibrium load values are equal in stiffness/damping matrix row N, for mooring at tower node X**.

When upgrading or downgrading a **Bladed** project, you will see a message box alerting you to this problem. So, for example, a 4.3 project loaded into 4.4 may have already had a correcting offset applied to the lookup tables, which you will need to undo. Alternatively, if you save a 4.4 project as 4.3, you should be aware that you may need to apply such an offset in order to get the correct mooring forces when running in 4.3.

4.6 Radiation-Diffraction Hydrodynamic Loading

4.6.1 Application of Advanced Hydrodynamics Module wave loading

Hydrodynamic loading resulting from wave diffraction around and radiation from support structures can be important for towers incorporating bulky, non-slender elements. The effects are most pronounced

when the characteristic length of the structure is of a similar size to the energetic incident wave lengths. Users with the Advanced Hydrodynamics Module can choose to account for radiation and diffraction effects by making use of a boundary element method based hydrodynamics formulation. The incident, diffraction, radiation and hydrostatic loading are calculated from coefficients that are typically provided by a linear boundary element method potential flow solver. The formulation should be used in place of a full Morison model on larger support structure members; although the viscous drag term on all members may be retained, as may the full Morison model for the slender elements not included in the boundary element method data.

4.6.2 Point loading

The advanced hydrodynamics formulation relies on the total distributed wave loading over any stiff segment of the tower being applied as a point load at a specified node. Multiple point loads may be specified for multiple tower segments in order to account for discrete parts of the tower structure interconnected with slender and more flexible 'Morison' elements. This means that multiple 'hydrodynamic bodies' can be defined for the same support structure in **Bladed** (each body contributing a single point load). It is assumed that bulky, diffracting parts of the structure are sufficiently stiff to support the application of a point load rather than a true distributed load as would be required for a more flexible component. If the complete support structure is stiff, a single hydrodynamic load may be applied to represent the amalgamated loading over the entire structure.

Note that in addition to the point loading accounting for the incident, diffraction, radiation and hydrostatic loading, a "hydrodynamic body" may still experience distributed loading from viscous drag (if enabled).

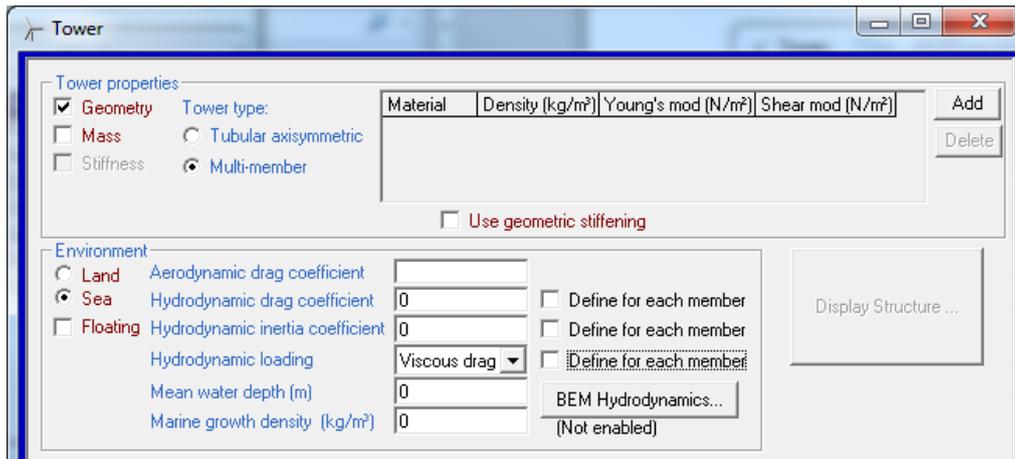
4.6.3 Using a boundary element method potential flow solver

The diffracting parts of the tower geometry are described in a mesh that is provided to the linear boundary element method flow solver. Each segment should be modelled in the flow solver as a distinct isolated body and meshed individually. If the entire tower is sufficiently stiff to be considered a single rigid body then the creation of a single mesh for the complete mean wetted geometry is typical. The flow solver simulation should be set up without any structural properties connecting segments of the tower together (slender connecting elements may be completely omitted) as these will be applied later in **Bladed** and it is useful to be able to account for the relative motion between the tower segments. The output coordinate system for each of the flow solver bodies must be placed at the same physical location as the **Bladed** tower node where the hydrodynamic load is to be applied. Furthermore, the orientation of the output coordinate systems must match that of the **Bladed** Global Coordinate system.

Where multiple bodies are used to model separate segments of the structure, a single flow solver calculation should be completed with all segments placed in the wave field together even though they are structurally disconnected. This allows hydrodynamic data describing the interactions between the tower segments to be supplied to **Bladed**. The tower segments should not be run separately in the potential flow solver as that would prevent the interactions terms from being incorporated.

4.6.4 Defining radiation diffraction loading in the Tower screen

To set up radiation diffraction loading for the tower, opt to either define Hydrodynamic Loading for each member (recommended), or select the 'Viscous Drag' or 'None' hydrodynamic loading in the main tower screen governing the Morison hydrodynamics formulation. Full Morison loading should not be applied to parts of the tower that are incorporated as part of a radiation diffraction hydrodynamic body, as this would result in the inertial part of the wave loading and the hydrostatic forces being double-accounted for. A viscous damping force may still be applied to radiating and diffracting elements however, if a suitable drag coefficient has been determined.



Click on the **BEM Hydrodynamics...** button to add hydrodynamic bodies to the model. Each hydrodynamic body represents a single wave loading point load for a segment of the tower. A tower node must be specified for each body as a location for the applied wave load. The tower node must be located in the same physical position as the flow solver hydrodynamic coefficients output coordinate system origin. All tower nodes have an orientation that is aligned with the **Bladed** global coordinate system and so the flow solver calculation must also be set up with this orientation.

The hydrodynamic properties associated with each hydrodynamic body are described below. The hydrodynamic data for the body will be defined for motion about a fixed point in physical space (the mean global location specified for the body in the flow solver); **Bladed** may be provided with the position and orientation of this point through the parameter 'Hydrodynamic Coordinate System Origin', which is entered in global coordinates; these coordinates will be numerically the same as the flow solver input file body position unless the tower has been built so that the global coordinate system origin in **Bladed** lies in a different location to the flow solver global origin. The tower node where the body is positioned should sit physically on top of its Hydrodynamic Coordinate System origin in the absence of waves and this point acts as the frame of reference for the hydrodynamic forces being applied as the tower moves around it under wave excitation. **Bladed** calculates the hydrodynamic origin of each body by default, setting the 'Hydrodynamic Coordinate System Origin' to the inherent node position as specified in the tower geometry definition. Users wishing to override this option and set the coordinate system manually may do so by selecting the 'Manually Set Hydrodynamic Origin' tick-box

Bladed Hydrodynamic Input Parameters	
Parameter Name	Description
Body Interaction Order	Describes the order in which the inter-body cross-terms appear in the added mass and impulse response function data matrices. This is a list of N strings where N is the total number of wave activated bodies in the system. Each entry should appear on a new line.
Body Hydrodynamic Origin (if option to manually specify has been selected)	(x,y,z,roll,pitch,yaw) vector describing the position and orientation of the body hydrodynamics coordinate system. The initial position of the node where the body is positioned will be used if the user does not opt to manually specify this parameter.
Global viscous drag	Optional 6x1 vector defining six-degree-of-freedom viscous drag characteristics for the body. The drag coefficients form a 6x6 matrix; all except the six diagonal elements are assumed zero, and it is these six that are entered here. The velocities and loads are all expressed in global coordinates, and the loads will be applied at the same support-structure node as the BEM ones.
Excitation Data	
Cartesian Direction Convention	Check box. Tick to indicate that the direction ordinate for the excitation force data is provided in global Cartesian angles (measured clockwise from positive X). Un-tick to specify nautical angles (measured clockwise from North).
North Direction	Cartesian angle of excitation force direction ordinate North axis, measured anti-clockwise from the global X direction.
Excitation Force Data Frequencies	Excitation force data frequency ordinate data provided as a list of scalar numbers.
Excitation Force Data Directions	Excitation force data direction ordinate data provided as a list of scalar numbers in radians if Cartesian angles are being used and degrees if nautical bearings are being used.
Excitation Force Data Amplitudes	Excitation force amplitude and phase data. Lists of matrices, one entry for each Excitation Force Directions value. Each matrix has six rows (Surge, Sway, Heave, Roll, Pitch, Yaw) and a number of columns equal to the number of Excitation Force Frequencies..
Excitation Force Data Phases	
Radiation Data	
Added Mass at Infinite Frequency	Added mass and moments of inertia at infinite frequency. Includes inter-body cross-terms so this data is a 6x6N Matrix for each body where N is the total number of wave activated bodies (total number of hydrodynamic components) in the model.
Time Ordinate Resolution	If the impulse response function time ordinate is equally spaced, then the spacing may be specified here to speed up run-time interpolation. A value of less than 0 should be entered if the spacing is uneven.
Time Ordinate	Impulse response function time ordinate, entered as a scalar list of time values. Bladed will interpolate the Impulse Response Function Matrices for this data.
Impulse Response Function Matrices.	Impulse response function data. Includes local body and inter-body cross-terms in column-wise the order specified by ' <i>Body Interaction Order</i> '. Data is entered as list of 6x6N matrices, one list entry for each Impulse Response Time value. N is the total number of wave activated bodies (hydrodynamic components) in the system. Each new matrix must start on a new line if added manually using the List Editor.
Cut-Off Time	Length of time-history used in the evaluation of the radiation force convolution for the body in question. Short time will speed up the simulation but will result in loss of accuracy. See 5.4.1 for more information.
Independent Radiation Calculation Timestep	Tick to specify a minimum radiation convolution calculation timestep that is coarser than the minimum integrator timestep. This will speed up simulations (as the convolution calculation will be completed less frequently), but may result in loss of accuracy. If this option is not selected, the radiation force calculation will be completed at a timestep determined by the Bladed numerical integrator.
Calculation Timestep	Minimum timestep at which to calculate the radiation convolution force term – only entered if <i>Independent Radiation Calculation Timestep</i> is selected.

Reduce Radiation Convolution Resolution	Tick to manually specify a reduced time resolution for the radiation force convolution calculation. This will speed up simulations but may result in unstable behaviour, particularly if a large resolution is selected. If this option is not selected, the convolution will be completed with a resolution dictated by the numerical integrator timestep.
Convolution Resolution	Time resolution to impose on the radiation force convolution buffer – only entered if <i>Reduce Radiation Convolution Resolution</i> is selected. If the Bladed numerical integrator takes step sizes that are larger than the time resolution specified here then the Bladed integrator step size will be used and the convolution time resolution.
Hydrostatic Data	
Mean Displacement Mass	The mass of water displaced by the body at its starting, equilibrium position.
Centre of Buoyancy	(x,y,z) vector describing the position of the body centre of buoyancy relative to in the hydrodynamics output coordinate system (the orientation of the hydrodynamic coordinate system matches the Bladed global coordinates system)
Hydrostatic Stiffness	6x6 Hydrostatic stiffness matrix for Surge, Sway, Heave, Roll, Pitch, Yaw. Stiffness terms relate to changes in the hydrostatic force on the body as it moves about its Hydrodynamic Coordinate System Origin.

4.6.5 The flow solver data import tool

A large amount of hydrodynamic data must be entered for each hydrodynamic body and so an automatic importing tool has been provided for users of the WAMIT or AQWA potential flow solvers. Once hydrodynamic bodies have been added for all appropriate sections of the tower in **Bladed**, click on the **Import flow solver data...** button to open the importing tool.

The flow solver data import process consists of two stages. New flow solver data must be processed in to the format required by **Bladed**. Following this, the data may be imported to the turbine model. Once flow solver data has been processed once, it may be imported multiple times as **Bladed** saves the processed data in the flow solver data output folder. A single hydrodynamic data file is saved for each body, with a ***.wdnh** extension. The flow solver data import tool presents the option to skip directly to the import itself alongside the option to process data from either WAMIT or AQWA. During the import process, the individual body **.wdnh** files must be matched up with the hydrodynamic bodies applied to the tower nodes.

The parameters required by the processing stage of the flow solver data import tool are described below. The processing tool will write a set of **Bladed** formatted output files alongside the **.wdnh** files and so the effect of the processing on the hydrodynamic data may be easily analysed in the standard Data View tool.

Flow Solver Data Processing Body Input Parameters	
Parameter Name	Description
Data Down-Sampling Factor	Parameter to allow down-sampling of the excitation force data. Use a value of 1 to keep the full flow solver output, 2 to use 1 frequency in 2, 3 to use 1 in 3 etc. This is can be useful because the flow solver simulation frequency resolution tends to be very fine for time-domain simulations to ensure that the impulse response function calculation is numerically accurate. The result is that a large amount of excitation force frequencies are provided. Bladed interpolates linearly between the excitation force data so it is not always necessary to enter the full frequency set for this part of the calculation. Nevertheless, interpolation at high frequencies still requires a high data resolution so a value of 1 is recommended unless simulations are restricted to low frequency wave components and disk space for project files is very limited.
Impulse Response Function Length	The length of impulse response function to calculate. A typical value is 120s as most impulse response functions have decayed to zero by this time. Longer times may be required however, especially if instabilities become noticeable in the simulation and the user should check the processing results to ensure that all significant impulse response functions (all those with appreciable magnitude) have decayed. Investigations may reveal that shorter times (as low as 30s) may be used in some cases, giving significantly higher simulation speeds. A reduced cut-off time may be specified in the hydrodynamic body parameters screen once the data has been imported, so it is advantageous to run for a longer time during this processing step and potentially cut shorter before running a simulation.
Impulse Response Function Resolution	The time resolution at which the impulse response function is evaluated. Bladed will interpolate for timesteps between the data points. A fine resolution allows laded to interpolate the data more accurately on smaller simulation timesteps but is unnecessary for smooth impulse response functions. A resolution of 0.05-0.1s is typical although the user should check the smoothness of the resulting impulse response function data.
Radiation Damping Minimum Cut-off Period	The minimum wave period (highest frequency) to be considered for the impulse response function calculation. This parameter specifies a minimum period below which flow solver radiation damping data is discarded for impulse response function calculation. If the mesh describing the body geometry used in the flows solver calculations is not sufficiently fine (particularly if a low-order mesh is used), then there will be a large amounts of noise in the radiation damping data at high frequencies. The user can (and should) choose to discard this noise by specifying a minimum wave period (maximum frequency) to process. If this means cutting off data frequencies where the radiation damping data, prior to noise disturbance, does not appear to have decayed to zero for all components, then a finer mesh (with a smaller panel size) should be used in the flow solver simulations.
Cartesian Direction Convention	Boolean parameter indicating whether the data processed for Bladed should use Cartesian or Nautical wave directions. Cartesian angles are assumed measured anti-clockwise from positive global X, whilst nautical directions are assumed to be bearings measured clockwise from North. This parameter refers to the format in which the pre-processor provides the excitation data to Bladed , not the format of the direction data supplied by the flow solver (this is typically fixed and handled automatically for WAMIT and AQWA). A Cartesian system is recommended unless the user has a specific reason to for viewing the excitation data in nautical coordinates.
North Direction	The Cartesian direction of North for use in when the Cartesian Direction Convention is not set. The North direction is entered as a direction measured anti-clockwise from positive Global X.
Water Density	Water density to use to dimensionalise the flow solver data.

4.6.6 Flow solver calculation requirements

Any commercial boundary element method potential flow solver may be used to provide **Bladed** inputs as long as it is capable of evaluating the following for each body:

- Hydrodynamic added mass as a function of frequency X. The value at infinite frequency (zero wave period) is also required, although a high frequency value may be used in its place (AQWA).

- Radiation damping as a function of wave frequency if the flow solver processing tool used. Alternatively, the impulse response function must be calculated directly for a suitable length of time (until oscillations in all terms decay to negligible amplitude).
- Wave excitation force amplitude coefficient and phase. The product of the amplitude coefficient and the wave amplitude yields the force amplitude for a given frequency.
- The hydrostatic water-plane moments of area expressed as a hydrostatic stiffness matrix.

In addition, the mean mass of water displaced by the body may be evaluated by the flow solver, or may be determined independently by the user.

All of these quantities must be evaluated for the body in a manner that accounts for the radiation from and diffraction around adjacent bodies in the device or array, or obstacles in the water, so a single flow solver run should be used for all the hydrodynamic bodies in a single **Bladed** simulation. The flow solver quantities should be evaluated for the bodies placed in the positions they would adopt in calm seas, or at a steady operating position (so for example the effect of mooring lines or negatively buoyant bodies pulling a body downwards on the overall draught of the turbine should be accounted for in the body positioning when the flow solver calculation is set up). These physical locations may then be entered in to **Bladed** (converted in to the **Bladed** global coordinate system) as the Hydrodynamic Coordinate System Origin for each body. The option to auto-calculate this parameter using the node locations provided in the tower geometry screen is enabled by default however however.

Where flow solver output data is provided in matrices, containing direct and inter-body (body interaction) cross-terms, the body order in which the cross-terms appear should be noted and entered in the '*Body Interaction Order*' parameter in **Bladed**. Bodies should be listed in the format "Body at node X", with each entry being specified on a new line.

The wave frequency range selected for use in the flow solver should encompass the complete range of wave regular wave frequencies to be run in **Bladed** and the frequencies of all the components used to model the sea states for any irregular wave simulations. **Bladed** will interpolate the hydrodynamic data for sea-state frequencies in between the computed values. To reduce computational time it is advantageous to exclude frequency components with close to zero spectral density from the input.SEA file at both the high and low frequency ends of the spectrum. The SEA file generator tool does this by default.

Flow solver output data should be analysed carefully before use in a **Bladed** model as it is possible that some of the radiation damping matrix terms will not have decayed to zero at either end of the initial frequency range selected. It is also possible that too coarse a frequency range will prevent significant peaks from being correctly resolved. The frequency range may need to be widened or the resolution improved even if it means extension beyond the frequency range where the device is known to be responsive or the range at which there is significant energy in the spectrum. Running at a finer resolution than the input seastate definitions may also be required.

Further to the above, it is important to ensure that the mesh of the body geometries is sufficiently fine to evaluate all quantities up to the highest frequencies required without introducing numerical noise (which will appear as large and apparently random oscillations in the data at adjacent frequency points). Convergence tests on the flow solver simulations are advised to determine when a suitable mesh has been refined. Note that high frequency noise in the data may be ignored but only if it occurs after the radiation damping terms have decayed to zero or the added mass terms have stabilised, as this may never converge. The noise may be cut-off by specifying a **Radiation Damping Minimum Cut-off Period** to be used in the flow solver data processing to produce impulse response functions.

4.6.7 Defining the sea state for radiation diffraction hydrodynamics

The radiation diffraction hydrodynamics formulation requires linear sea state definition in a.SEA file format. For details of how to generate a.SEA file using the SEA file generator tool, see [6.12.5](#).

4.6.8 Current limitations

The radiation diffraction hydrodynamics model released in v4.6 does not incorporate the effects of wave-current interaction. Wave loads are applied assuming the wave field described in the SEA file is undisturbed.

4.7 Defining the power train

Click the **Power Train** button on the toolbar to define the following aspects of the power train:

- the transmission (see [4.8](#)) (gearbox and shafts),
- any flexible mountings (see [4.9](#)),
- the generator (see [4.10](#)),
- the energy losses (see [4.11](#)), and
- the network (see [4.12](#)) to which the turbine is connected.

4.8 Transmission

The **Power Train** button on the toolbar allows the drive train or transmission to be specified. The transmission consists of the gearbox and shafts. Two alternative ways of specifying the drive train are provided:

- Locked speed model (see [4.8.1](#))
- Dynamic model (see [4.8.2](#))

Users with an Advanced Transmission Interface module licence may extend the dynamic model by supplying user-defined gearbox dynamics of arbitrary complexity through a defined DLL interface (see [4.8.3](#)).

4.8.1 Locked speed model

This model is available for simple calculations in which the turbine rotor is assumed to rotate at an absolutely constant speed. The rotor speed is specified along with the gearbox ratio, but no other characteristics of the drive train nor of the generator may be specified. Flexible mountings cannot be modelled either. This model is useful for initial aerodynamic design of fixed speed rotors, but for detailed performance and loading calculations the dynamic model is more appropriate.

4.8.2 Dynamic model

This model includes the rotational degree of freedom of the rotor. Torsional flexibility of the drive train may be included as an option.

Unless the Direct Drive option has been selected on the **Rotor** screen (see [4.1](#)), the gearbox ratio must be specified, along with the moment of inertia of the generator rotor. The additional inertia of the high speed shaft may also be specified along with the inertia of the gearbox which is referred to the high speed shaft. If there is a brake on the high speed shaft, its inertia should be included as shown; depending on the location of the brake, it may be more appropriate to include it in the gearbox inertia. Note that the moment of inertia of the turbine rotor is defined by the blade mass distribution (see [3.3](#)) and the hub inertia (see [4.2](#)).

The rotor speed will be governed by the characteristics of the generator (see [4.10](#)), which must be defined in this case. For a variable speed turbine it is also necessary to define an appropriate controller (see [5](#)) to define the speed schedule. If there is a shaft brake (see [5.12](#)) to be used in stopping and parked calculations, specify where it is located on the high speed or low speed shaft. Click the **Brakes** button to specify the dynamic characteristics of the brakes.

Use the check-boxes to specify if there is significant **torsional flexibility** in the low speed and/or high speed shaft, and supply the torsional stiffness. If both low and high speed shafts are flexible it is necessary to define the gearbox inertia. Torsional damping may also be specified. For fixed speed turbines, the damping is unlikely to have much effect as it will be swamped by the damping provided by the induction generator. In fact the damping may be sufficient to ensure that there is very little drive train torsional motion, in which case it may be advantageous not to specify any shaft torsion, as the simulations will then run faster.

Note: If not including flexibility in the high speed shaft, gearbox or pallet mounting, the LSS stiffness parameter should reflect the total torsional flexibility between the rotor and the tower top. For most geared transmission configurations this will include the series combination of low speed shaft flexibility and equivalent torsional flexibility of the gearbox mounting arrangement (if not defined separately in the Mounting screen) you should also consider whether the rotor hub, gearbox low speed stage and/or yaw bearing roll present significant additional torsional flexibilities about the low speed shaft axis.

Low speed shaft bending can be modelled in **Bladed** with two rigid massless shafts connected at a hinged bending point. Enter the total shaft length and the point of bending. The point of bending is entered as a percentage along the shaft length measured from the hub. So for example, a bending point of 0% would mean the hinge flexibility is at the hub, a bending point of 100% would mean the hinge flexibility is at the gearbox. The hinged connection is modelled as a 2D hinge, with flexibility about the hub 'y' and 'z' axes. The bending stiffness and damping of the hinged connection should also be entered.

If required, a **slipping clutch** may be modelled. This is assumed to be at the generator end of the high speed shaft. The clutch slipping friction should be entered, and any additional stiction which has to be overcome before slipping begins. The high speed shaft inertia must be specified if the slipping clutch is to be used.

The dynamic drive train model may be used in combination with flexible mountings (see [4.9](#)).

Note: including drive train flexibility may reduce the speed of simulations.

4.8.3 External DLL for transmission dynamics

Users with a licence for the Advanced Transmission Interface module may tick the "External DLL" checkbox to indicate that a user-defined DLL will be used to define the gearbox dynamics. Three options are available for the scope of the dynamics defined by the DLL:

Gearbox only: the DLL replaces the gearbox ratio in the **Bladed** model and has a single torsional degree of freedom only. Both the low and high speed shafts in the **Bladed** model must have torsional flexibility defined. If the shaft brake is defined then it is also considered to be within the scope of the DLL. Note that this affects not only the braking torque (see [5.12](#)) but also the mechanical losses (see [4.11.1](#)). Any flexible gearbox mountings must be modelled within the DLL if required.

Gearbox + HSS + generator rotor: the DLL replaces the gearbox ratio, high speed shaft and generator inertia in the **Bladed** model and has a single torsional degree of freedom only. The low speed shaft in the **Bladed** model must have torsional flexibility defined. If the shaft brake is defined then it is also considered to be within the scope of the DLL. Note that this affects not only the braking torque (see [5.12](#)) but also the mechanical losses (see [4.11.1](#)). The slipping clutch model is not available in this case – of course it can be modelled within the DLL if required. Any flexible gearbox mountings must be modelled within the DLL if required.

Gearbox + HSS + generator rotor (6 DOF): the DLL replaces the gearbox ratio, high speed shaft and generator mass and inertia in the **Bladed** model and has full six degrees of freedom. The low speed shaft and gearbox mountings in the **Bladed** model must have six degrees of flexibility defined. If the shaft brake is defined then it is also considered to be within the scope of the DLL. Note that this affects not only the braking torque (see [5.12](#)) but also the mechanical losses (see [4.11.1](#)). The slipping clutch model is not available in this case – of course it can be modelled within the DLL if required.

For the 6 DOF model it is also necessary to specify the length of the low speed shaft and the location about which the bending is assumed. This bending point is specified as a % of the shaft length from the hub side. The stiffness and damping parameters in all 6 degrees of freedom can be specified by clicking on the "LSS Stiffness / Damping Parameters" button.

On the mounting tab, the stiffness and damping parameters of the gearbox mounting in all 6 degrees of freedom have to be specified by clicking on the "Mounting Stiffness / Damping Parameters" button. In addition the dimensions of the gearbox need to be entered by specifying the location of the LSS relative to the centre of the mounting.

For all options:

Within the DLL, the dynamic equations of motion are specified in continuous time in state-space form:

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

where x is a vector of state variables defined within the DLL, u is a vector of torques acting on the DLL shafts, and y is a vector of output variables. Matrices A , B , C and D need not be constant: in other words the gearbox model may be non-linear and time-varying, and discontinuous changes are allowed.

The DLL interface is specified in detail in Appendix C. The DLL may be written in any suitable language, using either of the two most common calling conventions: `__cdecl`, which is the usual default for code written in C or C++, and `__stdcall`, which is the usual default for Fortran and Visual Basic; note however that these are only default settings and particular compilers may be different. The calling convention used must be specified on the Transmission screen, as well as the location of the compiled DLL.

User data: If desired, enter any User Data in the free-format field provided. This data will be placed in a text file whose location is passed to the DLL, so that the DLL may read this information if desired. This is useful for passing parameter values to the DLL which are also stored in the project file or the calculation details (\$pj) file.

Mechanical loss modelling:

If an external dll is used to represent the gearbox, the mechanical losses as specified in the user interface are not applied. Instead, they are calculated and passed through to the dll where they should be applied appropriately. The calculation of these losses depends on whether the brake is defined or not in the powertrain transmission screen:

- No brake defined: the losses are calculated at the HSS (note that this assumes the gearbox ratio as specified in the user interface)
- Brake at LSS: the losses are calculated at the LSS
- Brake at HSS: the losses are calculated at the HSS (note that this assumes the gearbox ratio as specified in the user interface)

4.9 Drive train mountings

The **Power Train** button on the toolbar also allows the drive train mountings to be specified.

If desired, torsional flexibility may be specified in the gearbox mounting and/or between the pallet or bedplate and the tower top. This option is only allowed if the dynamic drive train model (see [4.8.2](#)) is specified without the external user-defined DLL facility (see [4.8.3](#)). If the external DLL facility is used then of course it can include its own flexible mountings.

In either case, specify the following parameters:

- The torsional stiffness of the mounting.
- The damping of the mounting.
- In the case of a flexible gearbox mounting, the moment of inertia of the gearbox casing.

If desired, pallet or bedplate nodding stiffness can also be specified.

If either form of mounting is specified, the direction of rotation of the generator shaft will affect some of the loads. If the low speed and high speed shafts rotate in opposite directions, specify a negative gearbox ratio in the drive train (see [4.8](#)) model.

Flexible mountings introduce a high frequency mode which is liable to result in much slower simulations. If the mountings are relatively stiff, for example if they are intended for noise isolation, then they will not influence loads significantly and are best omitted. If they are intended to modify the rotational dynamics, then they can be approximated by a corresponding reduction in shaft stiffness. This will have little effect on rotor and tower loads, but simulations will run faster.

4.10 Defining the Electrical Model

The **Power Train** button on the toolbar also allows the electrical model of the generator and power converter to be specified. An electrical model must be defined if the dynamic drive train model (see [4.8.2](#)) is specified.

Three categories of electrical model are possible:

- Simple (see [4.10.1](#))
- Electrical Dynamics (see [4.10.3](#))
- External DLL (see [4.10.7](#)) (User defined electrical dynamics model)

Simple models are often sufficient for modelling turbine performance and loading while the Electrical Dynamics models also simulates currents and voltages, active and reactive power flows etc., and can be used to calculate electrical flicker coefficients (see [8.18](#)) and to model the effect of network voltage and frequency variations (see [7.20.2](#)). To use the Electrical Dynamics models, a licence for the Electrical Dynamics module is required.

The external DLL option (see [4.10.7](#)) allows a user-defined model for calculating the air-gap torque and electrical power output. This could be anything from a simple mechanical model to a comprehensive electrical transient model representing the electrical dynamics of the generator and also the power converter and electrical network if appropriate.

See also Defining the rotor ([4.1](#)), Control ([5](#)).

4.10.1 Simple electrical models

4.10.1.1 Fixed speed induction

This model represents an induction generator directly connected to the grid. It is characterised by:

- **Rated slip:** the slip speed at rated power as a percentage of synchronous speed.
- **Rated power:** the power output corresponding to rated slip.
- **Synchronous speed:** the rotational speed of the generator rotor at zero load.
- **Generator time constant:** this is the short-circuit transient time constant for the generator. **Note:** a small time constant may result in slower simulations. If it is very small, specifying a zero time constant will speed up the simulations, without much effect on accuracy.

The following table shows the synchronous speed for some typical cases:

Grid frequency:	50 Hz	60 Hz
2 pole generator:	3000 rpm	3600 rpm
4 pole generator:	1500 rpm	1800 rpm
6 pole generator:	1000 rpm	1200 rpm
8 pole generator:	750 rpm	900 rpm

4.10.1.2 Variable Slip

A variable slip generator is an induction generator with a controlled resistance in series with the generator rotor windings, giving a fixed speed induction generator characteristic below rated and limited range variable speed operation above rated.

The parameters required are a combination of the fixed speed induction generator parameters (see [4.10.1](#)) and the variable speed generator parameters detailed above, although power factor control is not available. The maximum slip should also be supplied.

If a two-speed variable slip generator is selected, torque control is only possible for the main generator.

Note that a variable slip turbine requires a controller (see [5](#)) which generates a torque demand. Many variable speed turbines require drive train damping feedback (see [4.10.8](#)) to help control torsional vibrations. This can be defined here if it is implemented in the generator/converter control. Alternatively it may be defined in the external controller as part of the main turbine controller.

4.10.1.3 Variable speed

For the simple variable speed model, the variable speed generator is modelled as a first order lag from demanded torque to actual generator torque, independently of rotational speed. This is specified by the **Power electronics time constant**. **Note:** a small time constant may result in slower simulations. If it is very small, specifying a zero time constant will speed up the simulations, without much effect on accuracy.

The **minimum and maximum generator torque limits** must be specified. Motoring may occur if a negative minimum torque is specified. The **power factor** defines the phase relationship between the current and voltage from the network side of the frequency converter. It is assumed that the power factor is completely controllable by the frequency converter.

Note that a variable speed turbine requires a controller (see 5) which generates a torque demand. Many variable speed turbines require drive train damping feedback (see 4.10.8) to help control torsional vibrations. This can be defined here if it is implemented in the generator/converter control. Alternatively it may be defined in the external controller as part of the main turbine controller.

4.10.2 Electrical Dynamics – default values and calculated values

While electrical dynamics model parameters are important to predict the behaviour of the electrical drive train, obtaining electrical data at the preliminary design stage of a turbine may be difficult. In situations where data is not known, the Bladed Electrical Dynamics Module guides the user with default values and calculated values. This enables the conceptual design stage of the electrical system and helps to predict the response of the electrical system at an early stage of the turbine design.

4.10.2.1 Default values

When a new Electrical Dynamics system is created by Bladed many of the parameters are set to meaningful default values. These default values have been chosen to represent typical systems used by the wind industry.

4.10.2.2 Calculated values

The design value for many of the parameters can be estimated from other parameter values in the electrical dynamics model. If the user has not entered a value in such cases, Bladed will calculate a suggested value based on the other parameter values. The user can override calculated values, in which case Bladed will stop auto-calculating the value and display it with a yellow background in the UI. Auto-calculation can also be enabled or disabled through the context menu.

4.10.3 Electrical Dynamics – Fixed speed induction

This model allows for a detailed representation which models the generator starting from its equivalent circuit parameters, namely stator and rotor resistance and inductance, mutual inductance, number of poles, grid frequency and voltage. An auxiliary load may be specified in parallel with the generator to represent any power-consuming auxiliary equipment which the turbine may have. Power factor correction capacitors may also be modelled. This representation of the generator allows both active and reactive power to be calculated, as well as voltages and currents.

For starting the generator, a soft starter can be modelled to ramp up the voltage from zero to the working value. It is also possible to specify that the power factor correction capacitors are switched in in stages, with a delay between each stage.

A choice of generator model orders is provided. Choose a high order for higher accuracy, or a low order for faster simulations. Further details are provided in the Theory Manual.

Since this representation includes modelling of the electrical losses in the generator and auxiliaries, electrical losses may not also be specified under Energy losses (see 4.11).

If **Two-speed operation** has been selected, enter all the relevant parameters for both the main generator and the low speed generator. The code automatically determines which generator to use, as follows:

- Steady calculations and simulation initial conditions: if the wind speed is less than 13 m/s, the generator which gives the higher power is used, otherwise the main (high speed) generator is used.

- Start-ups using built-in start logic: the generator whose synchronous speed is closer to the speed specified for generator switch-on (see [5.13](#)) is used.

An external controller (see [5.9](#)) can be used to determine which generator to use, and to implement speed change logic.

4.10.4 Electrical Dynamics – Variable slip generator

This is similar to the directly connected induction generator (see [4.10.1](#)), but the following additional parameters are required to define the variable slip control:

Additional controlled resistance: the additional resistance which is connected in series with the generator rotor, and controlled to vary the slip,

Proportional gain: the proportional gain of the PI controller which adjusts the additional resistance in response to the normalised rotor current error, and

Integral gain: the integral gain of the PI controller.

4.10.5 Electrical Dynamics – Synchronous generator with fully rated converter

An electrical dynamics model of a synchronous generator is available with a fully rated power converter between the generator and the grid.

Generator controller: three types are available. For detailed information about the controller types refer to the Theory Manual.

(a) Steady state controller

- This controller uses algebraic equations to calculate the required stator voltages.
- No additional parameters are required.
- The steady state model is effectively the same as that described in Section [4.10.1.3](#) with the “Power electronic time constant” set to zero.

(b) First order lag:

- If the details of the generator controller and its parameters are unknown, the first order lag is a suitable model to start with.
- This model is effectively the same as that described in Section [4.10.1.3](#) with a non-zero “Power electronic time constant”.
- Two time constants in both d and q axes are available. The time constants T_d and T_q represent the transients associated with flux control (T_d) and torque control (T_q) actions.
- If the time constants are set to zero, the first order lag controller becomes a steady state controller.

(c) PI with de-coupling

- Proportional and integral gains have to be defined.
- Select either reference or measured current as the input to the decoupling loops.

Notes

- If fast electrical transients are not important, a steady state generator model or the first order lag generator control model could be a suitable model in order to speed up the simulation.
- Fast electrical transients are important in some cases such as generator faults. The full order generator model would be desirable in this circumstance with an appropriate generator controller model.
- Steady state or reduced order generator models may not be compatible with a generator controller having PI controllers. These generator models neglect stator transients and the stator currents are used as the feedback for the PI controller, so a suitable solution may not be found in some circumstances.
- Electrical transients are considerably faster than the mechanical transients. Therefore when using transient electrical dynamics models, it is desirable to reduce the integrator tolerance.

PI control limit: select the type of PI control limit.

- **No limit:** this option do not apply any limit to the PI controller

- **Fixed limit:** The difference between the DC link voltage and the rated generator terminal voltage is applied to the PI.
- **Dynamic limit:** It is similar to **Fixed limit**, but it uses actual generator terminal voltage instead of rated voltage.

Grid side controller notes

- In order to speed up the simulation a current source controller can be used if the network transient is not important.

Defining the protection

Protection can be provided through torque demand override or a chopper resistor:

- (a) Torque demand override:
 - This method rapidly reduces the generator torque during a grid fault. In order to change the torque rapidly, a fast torque control scheme is required as the generator controller.
 - Choose either terminal or DC link voltage as the input to the torque control protection scheme.
 - A filter at the input to the torque control protection is given as a first order lag. Setting a zero time constant removes this filter.
 - Rate of change of torque can be limited by enabling the **torque ramp slope limit**.
- (b) Chopper resistor protection:
 - This method dissipates power as heat in order to maintain the DC link voltage below its upper limit. The DC link maximum voltage limit has to be defined.
 - DC link voltage tolerance is used for the switching of the chopper protection scheme.
 - Maximum time limit for the braking chopper can be defined by selecting **Chopper time limit** box and then defining the time.
 - If the braking chopper time limit is exceeded then the torque control method can be activated by selecting **Maintain DC link voltage**

4.10.6 Electrical Dynamics - Doubly fed induction generator model

The doubly-fed induction generator model has a power converter between the rotor windings and the network, while the stator is directly connected.

Defining the generator

The model can be steady state, reduced order (neglecting stator transients) or transient (full order). Stator transients may be particularly important for transient events such as network voltage dips and crowbar operation.

PI control limit: three options for limiting the PI control are available:

- **No limit:** this option do not apply any limit to the PI controller
- **Fixed limit:** The difference between the DC link voltage and the rated generator terminal voltage is applied.
- **Dynamic limit:** It is similar to **Fixed limit**, but it uses actual generator terminal voltage instead of rated voltage.

Defining the protection.

Protection for low voltage ride through can be provided by defining a crowbar.

4.10.7 User-defined generator model

Users with a licence for the Advanced Transmission Interface module may select the "External DLL" option to indicate that a user-defined DLL will be used to define the electrical dynamics. This DLL replaces the model currently used to represent the electrical dynamics of the generator, together with the power converter and network if applicable. The mechanical equations of motion of the generator, for example to describe the rotation of the generator rotor, are not included in the DLL since they are already modelled within **Bladed**. Note that any electrical losses should be included with the DLL.

Within the DLL, the dynamic equations are specified in continuous time in state-space form:

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du_{in}$$

$$y_{out} = C_{out}x + D_{out}u$$

where x is a vector of state variables defined in the DLL, u and u_{in} are vectors of input variables such as generator speed, torque demand, and fractional disturbances in network voltage and frequency. Vector y is output variables such as generator air-gap torque and electrical power. y_{out} is a vector of output variables defined in the DLL (e.g. currents, voltages and reactive power). Matrices A , B , C , D , C_{out} and D_{out} represent terms defined in the DLL, although they need not be defined explicitly as matrices, and need not be constant: in other words the electrical dynamics model may be non-linear and time-varying, and discontinuous changes are allowed.

The DLL interface is specified in detail in Appendix D. The DLL may be written in any suitable language, using either of the two most common calling conventions: `__cdecl`, which is the usual default for code written in C or C++, and `__stdcall`, which is the usual default for Fortran and Visual Basic; note however that these are only default settings and particular compilers may be different. The calling convention used must be specified, as well as the location of the compiled DLL.

User data: If desired, enter any User Data in the free-format field provided. This data will be placed in a text file whose location is passed to the DLL, so that the DLL may read this information if desired. This is useful for passing parameter values to the DLL which are also stored in the project file or the calculation details (\$pj) file.

4.10.8 Drive train damping feedback

The generator torque control may also include a built-in feedback term derived from the generator speed by means of a transfer function. This is superimposed on the torque demand from the controller, and may be used to help damp out drive train torsional vibrations. Specify the transfer function required in terms of numerator and denominator polynomials in the Laplace operator.

NOTE: Do not include the drive train damping feedback here if it is already included in the torque demand from the external controller.

4.11 Energy losses

The **Power Train** button on the toolbar also allows the energy losses to be specified. Power train energy losses are divided into mechanical and electrical losses.

4.11.1 Mechanical losses

A choice of two options is available to specify mechanical losses, for convenience. Select **Expressed as torque** to specify a loss torque in kNm (referred to the low speed shaft), or **Expressed as power** to specify a power loss in kW. Then enter either a constant loss in the box, or click the appropriate **Add** button(s) as required to create a look-up table. This may be a simple look-up table on shaft speed or on shaft torque or power (depending on the option selected), or a two-dimensional look-up table on shaft speed and either torque or power. If a look-up table is used, loss values will be linearly interpolated from the table.

The option to express losses in terms of power has been provided for convenience, since data is sometimes provided in this form. However this model is inappropriate at low rotational speeds. If any calculations at low speed are to be carried out, including start-ups, shut-downs, idling, or parked simulations with slipping brake, then it is advisable to recalculate the losses in terms of torque.

In the **Bladed** model, the mechanical loss torque is applied at the position of the shaft brake (see [5.12](#)). If the brake is on the high speed shaft, the loss torque is scaled appropriately using the gearbox ratio.

When using an external gearbox DLL, the mechanical losses defined in the user interface are not applied and should be applied in the DLL if required.

4.11.2 Electrical losses

Two alternative models are available for the electrical losses:

Linear model:

This requires a no-load loss L_0 and an efficiency ε , where the electrical power output P_e is related to the generator shaft input power P_s by:

$$P_e = \varepsilon (P_s - L_0)$$

Look-up table:

The power loss $L(P_s)$ is specified as a function of generator shaft input power (P_s) by means of a look-up table. The electrical power output (P_e) is given by:

$$P_e = P_s - L(P_s)$$

Use the **Add** button to add points to the look-up table. Double-click on a look-up table entry to edit it.

4.12 The Electrical Network

From the **Power Train** button on the toolbar, click **Network** to define the electrical characteristics of the wind farm network and the connection from the wind farm to the external network. This is useful if the **Electrical dynamics** option has been selected for the fixed speed induction generator, or if the phase angle is specified for the variable speed generator. Then the network currents and voltages, as well as active and reactive power, will be calculated at the turbine, the point of common coupling, and the 'infinite busbar'.

Click **Network as shown**, and enter the resistance and inductance of the line connecting the turbine to the point of common coupling, and of the line connecting the wind farm to the infinite busbar. Enter also the voltage for which these impedance values were calculated. If required, specify the number of additional turbines connected at the point of common coupling. These turbines will be assumed identical to the turbine being modelled, and running at a constant current and voltage equal to that of the turbine being modelled at the start of the simulation. The dynamics of the additional turbines is not modelled, but the steady state voltage rise at the point of common coupling will be calculated.

4.13 The nacelle

Click the **Nacelle** icon on the toolbar to define the nacelle characteristics.

Check the **Geometry** checkbox to define the characteristics required to calculate windage loads. Two models are available:

Drag model: the following information is required:

- **Nacelle length**
- **Nacelle height**
- **Nacelle width**
- **Nacelle drag coefficient.**

And the drag force is then calculated based on the projected nacelle area perpendicular to the wind direction at any time,

Aerofoil model: if the lift and drag forces on the nacelle are known as a function of yaw angle, specify:

- **Nacelle length**
- **Nacelle height**
- **The position of the upwind end of the nacelle relative to the tower (yaw bearing) axis**, and
- Select an **aerofoil from the database** which has been constructed to give the correct variation of lift and drag with angle of attack (yaw angle). The nacelle length will be used as the aerofoil chord, and the aerofoil area will be the product of length and height.

Check the **Mass** checkbox to define the nacelle mass characteristics required for dynamic calculations. This includes the nacelle structure and all the machinery within it. It does not include the rotor blades and hub. If the Direct Drive (see [4.1](#)) option is selected, the mass of the generator (see [4.2](#)) is also excluded. Enter the following:

- **Mass on tower top.**
- **Centre of mass position** - see diagram.
- **Yaw moment of inertia (about the tower axis).**
- **Nodding and rolling moments of inertia** – these are specified about the centre of mass position.

Note

The **Turbine Information** window shows the total masses and inertias of all turbine components whose mass characteristics are defined. Use the **Windows** pull-down menu on the main toolbar to open this window.

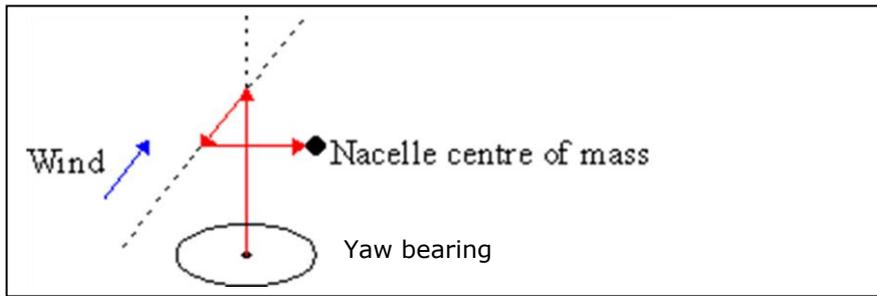


Figure 4-7: Diagram showing positive directions for centre of mass position relative to the top of the tower, showing wind direction during normal running.

5 CONTROL SYSTEMS

Click the **Control** icon on the toolbar to open the **Control Systems** screen. This allows definition of the power production control characteristics as well as supervisory control functions.

Power production control

Power production control models are available for pitch or stall regulated, fixed or variable speed turbines.

Select one of the following options:

1. **Fixed speed stall regulated** (see [5.1](#)): the induction generator model should be selected.
2. **Fixed speed pitch regulated** (see [5.2](#)): the induction generator model and a pitch or aileron controlled rotor should be selected.
3. **Variable speed stall regulated** (see [5.3](#)): the variable speed generator model should be selected.
4. **Variable speed pitch regulated** (see [5.4](#)): the variable speed or variable slip generator model should be selected, and a pitch or aileron controlled rotor.

See also: Generator ([4.10](#)), Drive train ([4.8](#)), Defining the rotor ([4.1](#)).

The steady state parameters define the operating envelope of the turbine, and are required for calculation of the steady power curve and steady operational loads. Click on **Controller Dynamics** to define additional parameters which model the dynamic action of the controller, such as controller gains and transducer and actuator time constants.

The different power production control models are described in Sections [5.2](#) to [5.8](#) below. Further details of the power production controllers is provided in the Theory Manual. It is also possible for user-defined control logic to be used - see [5.9](#).

Supervisory control

The Supervisory control screens may also be accessed by clicking the **Control** icon on the toolbar.

Use the Supervisory control screens to define the characteristics of the following:

- Shaft brake (see [5.12](#))
- Start-up sequence (see [5.13](#))
- Normal stop sequence (see [5.14](#))
- Emergency stop sequence (see [5.15](#))
- Idling conditions (see [5.16](#))
- Parked conditions (see [5.17](#))
- Yaw control (see [5.18](#))
- User-defined controller (see [5.9](#))
- Pitch actuators (see [5.11](#))

In-built supervisory control logic is available for stops and starts, but the user may also supply the required logic for these (see [5.9](#)), as well as for yaw control.

5.1 Fixed Speed Stall Regulated Control

There is no in-built controller for fixed speed stall regulated turbines, so no parameters need be entered in this case.

However, clicking on **Dynamics** will allow the speed and power transducer dynamics to be defined, even though these are not used for power production control. The transducer signals are available so that an external controller can have access to them if desired. Click on **Use external controller for pitch/airbrake control** if an external controller will be used to control any pitchable blades, tips or ailerons. In this case, the aerodynamic control surfaces must be defined appropriately on the **Rotor** and



Blade screens. The software rate pitch limits will be available to the external controller if required, and the pitch actuator dynamics should be specified.

5.2 The fixed speed pitch regulated controller

Use this control model for a fixed speed turbine (i.e. with a directly-connected induction generator (see [4.10.1](#)) which has a pitch- or aileron-controlled rotor (see [4.1](#)).

5.2.1 Steady state parameters

To specify the steady state operating curve, supply the following data:

- **Demanded electrical power:** the power set-point, i.e. the electrical power output required above rated wind speed.
- **Minimum pitch angle:** the lower limit of pitch or aileron deployment angle.
- **Maximum pitch angle:** the upper limit of pitch or aileron deployment angle.
- Select **Pitch feathering** or **Assisted stall**: to specify the direction of pitching. If Pitch Feathering is selected, the Minimum pitch angle will be set for normal operation below rated wind speed, and the pitch (or aileron) angle will increase above rated. If Assisted Stall is selected, the Maximum pitch setting is used below rated, and the pitch (or aileron) angle decreases (or moves to more negative values) above rated.

5.2.2 Dynamic parameters

This controller varies the pitch or aileron angle above rated, to regulate the electrical power output to the set point given above. The following information is needed for simulations which include controller dynamics:

- **Power transducer time constant** See [5.10.1](#). A speed transducer may also be defined if desired.
- **Pitch actuator dynamics** Enter the software rate limits for use in the control algorithm. Click **Actuator...** to specify the pitch actuator details (see [5.10](#)).
- **Control algorithm** Choose between PI (see [5.5](#)) and Discrete External (see [5.9](#)), and click **Define...** to specify the details.

5.3 The variable speed stall regulated controller

Use this control model for a variable speed turbine (i.e. with a variable speed generator (see [4.10](#)) which uses the aerodynamic stall characteristics of the rotor (see [4.1](#)) to control the power above rated wind speed.

5.3.1 Steady state parameters

To specify the steady state operating curve, supply the following data:

- **Torque-speed curve below rated** See Variable speed control below rated ([5.8](#)).
- **Demanded shaft power** The power set-point, i.e. the shaft power required above rated wind speed. See Energy losses ([4.11](#)).
- **Maximum generator speed** Usually the same as the Optimal mode maximum speed, but exceptionally a higher value could be used for high wind speed operation.

5.3.2 Dynamic parameters

This controller varies the generator torque to regulate the speed of the generator and rotor. Above rated, the power can be controlled by adjusting the speed set-point to stall the rotor. The following information is needed for simulations which include controller dynamics:

- **Speed transducer time constant** See transducers (see [5.10.1](#)). A power transducer may also be defined if desired. The in-built control algorithm uses the product of measured speed and generator torque to give a measured shaft power signal.
- **Control algorithm** Choose between PI (see [5.5](#)) and Discrete External (see [5.9](#)), and click **Define...** to specify the details. If you choose PI, you must define both the torque and the power controller details. See also Variable speed control below rated ([5.8](#)).

As for the fixed speed stall regulated controller (see [5.1](#)), it is possible to define an external pitch controller and a pitch actuator if required.

5.4 The variable speed pitch regulated controller

Use this control model for a variable speed turbine (i.e. with a variable speed generator (see [4.10](#))) which has a pitch- or aileron-controlled rotor (see [4.1](#)).

5.4.1 Steady state parameters

To specify the steady state operating curve, supply the following data:

- **Torque-speed curve below rated:** See Variable speed control below rated ([5.8](#)).
- **Pitch feathering or Assisted stall:** Specify the direction of pitching.
- **Minimum pitch angle:** the lower limit of pitch or aileron deployment angle.
- **Maximum pitch angle:** the upper limit of pitch or aileron deployment angle.
If Pitch Feathering is selected, the Minimum pitch angle will be set for normal operation below rated wind speed, and the pitch (or aileron) angle will increase above rated. If Assisted Stall is selected,

the Maximum pitch setting is used below rated, and the pitch (or aileron) angle decreases (or moves to more negative values) above rated.

- **Demanded generator torque:** the maximum generator torque demand, used as a constant torque demand above rated wind speed.
- **Demanded speed above rated:** the generator speed demand above rated. This is usually the same as or slightly above the Optimal mode maximum speed.

5.4.2 Dynamic parameters

This controller varies the generator torque to regulate the speed of the generator and rotor. Above rated, the torque is held constant and the speed is then controlled by adjusting the blade pitch angle. The following information is needed for simulations which include controller dynamics:

- **Generator speed transducer** See [5.10.1](#). A power transducer may also be defined if desired.
- **Pitch actuator dynamics** Enter the software rate limits for use in the control algorithm. Click **Actuator...** to specify the pitch actuator (see [5.10](#)) details.
- **Torque control algorithm** Choose between PI (see [5.5](#)) and Discrete External (see [5.9](#)), and click **Define...** to specify the details. See also Variable speed control below rated.
- **Pitch control algorithm** Choose between PI (see [5.5](#)) and Discrete External (see [5.9](#)), and click **Define...** to specify the details.

5.5 PI control

PI or Proportional plus Integral controllers are widely used for closed loop controllers on turbines. A PI controller with input x and output y has the form:

$$y = K_p \cdot x + K_i \cdot \text{int}(x)$$

where

K_p = Proportional gain

K_i = Integral gain

and $\text{int}(x)$ is the time integral of x .

The input x is the deviation of a measured quantity from its desired level or set-point. The output y is the demanded control action, which is intended to minimise those deviations.

Bladed has the following built-in PI controllers:

Fixed speed pitch regulated	x = deviation of measured power from the set-point y = pitch angle demand*
Variable speed torque control (both stall and pitch regulated)	x = deviation of measured generator speed from the set-point y = generator torque demand
Variable speed stall regulated power control	x = deviation of shaft power from the set-point y = generator speed set-point
Variable speed pitch controller	x = deviation of measured generator speed from the set-point y = pitch angle demand*

*Note: If a pitch rate actuator (see [5.10](#)) is defined, the rate of change of this pitch angle demand is used as the input to the actuator.

Various control design techniques are available to help with the selection of the gains K_p and K_i . However the design of controllers is a specialist task. The main aim is usually to minimise deviations

from the set-point without excessive control action and without causing any instabilities. A non-zero integral gain is important to ensure that the steady-state error is zero. This means that the measured quantity settles at the desired value in steady state conditions. The ratio K_p/K_i is known as the integral time constant.

For very simple cases only, an [automatic calculation of PI gains](#) is available. Click the “**Auto tune ...**” button after having first run a [Steady Operation Loads](#) calculation which generates the required aerodynamic characteristics. This may be useful in the early stages of design when the blade aerodynamic characteristics are defined but structural details are not yet available.

It is often useful or necessary to vary the overall gain of the controller depending on the operating point. This is known as Gain scheduling - see [5.6](#).

Garrad Hassan will be pleased to undertake the design of any turbine controllers.

The **desaturation time constant** is used to prevent “integrator wind-up” when the output y is constrained by limits. If this is too long, the controller may take too long to break away from a limit. For example a pitch controlled machine crossing rated wind speed may take too long before the pitch starts acting. On the other hand a very short time constant may result in slower simulations. If the PI controller being modelled is actually implemented in discrete form, as is usual, then the desaturation time constant should be chosen to be somewhat smaller than the discrete controller timestep. Alternatively, specify a zero time constant for instantaneous desaturation.

5.6 Gain scheduling

The gains of a closed loop controller such as a PI controller (see [5.5](#)) are designed to give good control (i.e. stable operation with good tracking of the set-point) at one particular operating point, for example at one wind speed. At a different operating point, the gains may need to be modified because the relevant characteristics of the system may be different. This is particularly true of pitch controlled turbines, where the aerodynamic gain changes significantly with pitch angle, and hence with wind speed.

The gain scheduling facility allows the overall gain of each PI controller to be modified as a function $f(V)$ of some chosen variable V . Thus the actual proportional and integral gains at any point are given by $K_p/f(V)$ and $K_i/f(V)$, where K_p and K_i are the proportional and integral gains as entered in the boxes provided.

The variable V is selected by clicking the arrow by **Schedule on:** and choosing from the list. The variables to choose from are:

- **Pitch** [in radians] (for pitch regulated controllers only).
- **Power** [in W]
- **Speed** [in rad/s] (generator speed, for variable speed controllers only)
- **Wind speed** [in m/s] (note that in practice, a suitable wind speed signal is unlikely to be available).

There are three possibilities for $f(V)$:

- **Constant:** the gains are fixed, with $f(V)$ constant as defined by **Value**.
- **Look-up table:** $f(V)$ is defined by a look-up table. Click **LUT Data** to open the table. Use **Add** to add points to the table, and double-click on an entry to edit or enter a value. The X values refer to the variable V in the units shown, and the Y values are the corresponding $f(V)$.
- **Polynomial:** $f(V)$ is defined by a polynomial. Click **Coefficients** to define it. Use **Add** to add terms to the polynomial, and double-click on an entry to edit or enter a value. Enter a coefficient for each term of the polynomial. $f(V)$ is the sum of terms of V (in the units defined above) raised to the power of the order n , and multiplied by the coefficient entered for that term.

Recommendations

For pitch regulated turbines, it is usually useful to apply gain scheduling to the above-rated pitch controller. This applies to both fixed speed and variable speed pitch regulated controllers. This is because the sensitivity of aerodynamic torque to changes in pitch angle is much greater at large pitch angles, and hence large wind speeds, than close to rated. The gain schedule may be defined as a function of pitch angle, and it is often appropriate, though conservative, to set $f(\text{pitch angle})$ to be

proportional to the partial derivative of aerodynamic torque with respect to pitch angle. Once the steady state control parameters have been defined, use the steady operating loads calculation to calculate the partial derivatives at each operating point, using the '**Calculate pitch and speed change**' option.

For very simple cases only, if the [automatic calculation of PI gains](#) is used for the pitch controller, an appropriate gain schedule is also calculated at the same

Tip: in practice, for full-span pitch-regulated rotors pitching towards feather, setting $f(\text{pitch angle})$ to be proportional to pitch angle is often a reasonable first approximation, but this is unlikely to apply to partial span pitch or aileron-controlled rotors, or in the assisted stall case.

Gain scheduling is unlikely to be required for below rated controllers on variable speed turbines. For above-rated stall-regulated variable speed controllers, the need for gain scheduling will depend on the stall characteristics.

5.7 Auto tune for internal PI controllers

The tuning of control loops is a specialist task. To tune a [PI controller](#) it is necessary to select appropriate values for the proportional gain and the integral gain, and in the case of the pitch controller these gains will vary with the operating point according to a [gain schedule](#). Classical tuning methods based on open and closed loop analysis of the system and its controller may attempt to achieve a particular cross-over frequency and stability margins for example. It is usually necessary to include a number of filters in series with the PI controller to achieve desired characteristics, and also to filter out signal noise, structural resonance frequencies, and forcing frequencies such as multiples of the blade passing frequency.

By ignoring all structural resonances and forcing frequencies it is possible to formulate an automatic calculation of PI gains. **Bladed** provides such a calculation, which may be appropriate for preliminary calculations at an early stage of turbine design when only the aerodynamic characteristics of the rotor have been defined. The calculation is only available for variable speed pitch regulated turbines with feathering pitch. First, carry out a [Steady Operation Loads](#) calculation, which generates the required aerodynamic characteristics. Then on the Control Systems screen, select the controller dynamics for a variable speed pitch regulated controller, and click one of the PI Control definition buttons. The "Auto tune..." button can then be used to launch the automatic calculation. You will be prompted to locate the results of the Steady Operational Loads calculation, and asked to confirm the required cross-over frequency, for which a default value of 1.0 rad/s is provided. In the case of the pitch PI controller, an appropriate [gain schedule](#) is also calculated.

The resulting controller is not likely to work effectively once any of the following effects are taken into account:

- Tower flexibility
- Blade flexibility
- Drive train flexibility
- Turbulence
- Non-uniformity of the flow field
- Non-axial flows
- Sensor and actuator dynamics
- Noise and delays in control loop signals

Since these effects are important, a professional controller design is required for any detailed calculations.

5.8 Variable speed control below rated

Select **Optimal tip speed ratio** or **Look-up table**, and proceed as follows:

5.8.1 Optimal tip speed ratio

Below rated wind speed, a variable speed turbine may try to stay at its optimum tip speed ratio wherever possible, by changing the rotor speed in proportion to the wind speed. This maximises the power coefficient and hence the aerodynamic power available.

This can be achieved in the steady state by setting the generator torque to be proportional to the square of the rotor or generator speed. The **Optimal mode gain** multiplies the square of the generator speed to give the required generator torque demand. It can be calculated as:

$$K_{opt} = \pi \rho R^5 C_p / 2 \lambda^3 G^3$$

where

K_{opt} = Optimal mode gain
 ρ = air density
 R = rotor radius
 C_p = power coefficient at λ
 λ = desired tip speed ratio
 G = gearbox ratio

The torque demand is then given by

$$Q_d = K_{opt} \Omega^2$$

where

Q_d = generator torque demand
 Ω = measured generator speed

Note that energy output may not be maximised by maximising aerodynamic efficiency, because the energy losses may also vary with the operating point. It may therefore be better to track a slightly different tip speed ratio. The desired tip speed ratio and the corresponding power coefficient should be used in calculating the **Optimal mode gain** as above.

See [Automatic calculation of optimal torque-speed curve](#).

Alternatively the torque-speed curve may be expressed as a [look-up table](#) to account for losses or any other effects

5.8.2 Look-up table

Click **Data**, and enter a look-up table of generator torque demand against generator speed, using the **Add** button to add entries to the table. The **View** button allows the curve to be visualised.

See [Automatic calculation of optimal torque-speed curve](#).

5.8.3 Automatic calculation of optimal torque-speed curve

When operating in the peak region of peak power coefficient, the torque-speed curve can be calculated by means of a [quadratic relationship](#) which can be calculated from the turbine parameters once the peak power coefficient and the corresponding optimum tip speed ratio are known.

First use the [Performance Coefficients](#) calculation to calculate the CP- λ curve for the optimum fine pitch angle. If this is not known, repeat the calculation for different pitch angles to find the angle giving the highest peak CP.

Then on the Control Systems screen with "Variable Speed" selected, click the '**Calculate ...**' button in the below rated section to perform the calculation of the optimal mode gain. The result will either replace the existing value or be used to construct a [look-up table](#) if that option has been selected.

You will be prompted to confirm the air density and to locate the Performance Coefficients calculation results.

5.8.4 Other parameters

In addition to the above, specify the **Minimum generator speed** and the **Optimal mode maximum speed** (this must be no greater than the **Maximum generator speed**).

The controller will adjust torque demand as above using the optimal mode gain or the look-up table provided the generator speed lies between the **Minimum generator speed** and the **Optimal mode maximum speed**. If either of these speed limits is reached, it is used as the speed set-point for the Speed Control by Torque Demand PI controller. The deviation of measured generator speed from this set-point is the input to the controller, and generator torque demand is the output. Thus the generator torque is varied to keep the generator speed at the limit.

Note: This PI controller remains active while the speed is between the limits, but the torque demand output is constrained to lie on the specified torque-speed curve.

In addition to the **Optimal mode gain**, the **Optimal mode maximum speed** must be specified. This must be no greater than the **Maximum generator speed**. The controller will adjust torque demand as above to maintain the desired tip speed ratio provided the generator speed lies between the **Minimum generator speed** and the **Optimal mode maximum speed**. If either of these speed limits is reached, it is used as the speed set-point for the Speed Control by Torque Demand PI controller (see [5.5](#)). The deviation of measured generator speed from this set-point is the input to the controller, and generator torque demand is the output. Thus the generator torque is varied to keep the generator speed at the limit.

Note: This PI controller remains active while the speed is between the limits, but the torque demand output is constrained to lie on the quadratic curve defined by the **Optimal mode gain**.

Note also that energy output may not be maximised by maximising aerodynamic efficiency, because the energy losses (see [4.11](#)) may also vary with the operating point. It may therefore be better to track a slightly different tip speed ratio. The desired tip speed ratio and the corresponding power coefficient should be used in calculating the **Optimal mode gain** as above.

Further details are given in the Theory Manual.

5.8.5 Control in the variable slip case

With a variable slip generator, it is sufficient to set a high torque demand below rated: the generator itself will restrict operation to the nominal slip curve. This can easily be specified using the [look-up table](#) option.

5.9 User-defined controllers

Although **Bladed** has a full set of built-in controllers, both for power production control and for supervisory control, a wide range of different control algorithms is used in practice by different turbine manufacturers. The controller details can have a profound effect on turbine loads and performance, so it is important to allow the flexibility to use any control logic the user wishes to define. Also it is worth considering whether the same code can be used for controlling **Bladed** simulations as is used in the actual turbine controller.

Bladed allows a user-defined controller to be used for any of the following tasks:

- Control of blade pitch and generator torque across the whole range of operation, including power production, normal and emergency stops, starts, idling and parked conditions.
- Control of the shaft brake and generator contactors.
- Control of nacelle yaw.

You may use a mixture of user-defined and built-in controllers for the various different controller functions.

Note that the user-defined controller operates on a discrete timestep, like most real controllers. The built-in controllers operate in continuous time, but can be used to approximate the behaviour of a discrete controller as long as its timestep is not too long.

The user-defined controller may be written in any language, as a 32-bit DLL (dynamic link library).

The user-defined controller is ignored in the case of Hardware Test (see [7.17](#)) simulations.

Bladed versions prior to 4.4 used a "swap array" to pass data between **Bladed** and the external controller. This has been replaced in version 4.4 onwards with a new function style of data communication, making use of "Get" and "Set" functions. The old swap array method is still supported but will not be extended further. Details of the new method can be found in Appendix A with further details available to download from the **Bladed** website. Details of the old "swap array" method are available to download from the **Bladed** website.

5.9.1 Using a user-defined controller

To use your own controller for any of the control functions, select the External Controller option where appropriate on the Control Systems or Yaw Control windows.

To specify your controller, select the External Controller tab on the **Supervisory Control** window; on the Control Systems window, **Define...** buttons are provided for this in appropriate places. Then specify the following information:

Controller code: the compiled.exe or.dll file which runs your controller (a dialogue box allows you to select the appropriate file)

Communication Interval: the controller timestep. At these regular intervals, **Bladed** will send information to the external controller, and expect to receive information in return. In real life the controller will take some time to calculate its outputs, usually a whole timestep, so it may be appropriate for the controller to always return the results of the previous timestep's calculation.

Noise on measured signals: if desired, click **Specify noise** to specify any random noise and discretisation errors (see [5.10.3](#)) on the signals passed to the external controller.

External Controller Data (optional): The information you provide here will be placed in a text file DISCON.IN which your controller is free to open and read when it starts up (no path is required as the file will be in the **Bladed** directory, which is the current directory when the external controller starts. You can use this to pass additional parameters to your controller, e.g. gains, look-up tables, etc. There is no need to pass the set-points etc. which are defined in the steady-state part of the Control Systems window, as these will be passed across in the communication file or array. (Note: to enter a tab in this field, press Control+Tab).

Important note:

When using a user-defined controller, **Bladed** evidently cannot guarantee that the performance defined by the steady-state control parameters will actually be achieved on average during a dynamic simulation.

5.10 Simulation of Measured Signals

Bladed provides the facility to mimic measured signals – that is signals that are measured using a physical transducer. This is of particular relevance to controller design, when the input signals upon which the controller relies (such as the blade pitch angle) may suffer from noise, lag, discretisation or specific faults.

Any signal available to the external controller which represents a value coming from a physical transducer can have the following properties defined:

1. Noise
2. 1st or 2nd order transducer behaviour
3. Discretisation of value (e.g. rotor speed discretised to 0.1 rpm)
4. Sampling period (i.e. discretisation in time)
5. Various inbuilt or time-dependent faults

Signals which can have these properties defined can be identified in the external controller API as they begin "GetMeasured..." (see [APPENDIX A Communication Between Bladed And External Controllers](#) for details of how to write an external controller). In the user interface, these signals can be viewed by clicking 'Measured Signals' in the Control Systems window. For each of the transducer signals listed, it's possible to select any of the following signal quality values:

- Simulated Value** The external controller will receive values exactly as they occur in the simulation.
- With Transducer Lag** Transducer behaviour is added to the signal, along with any faults.
- As Measured** Noise, discretisation, transducer behaviour, and faults are all added to the signal.

5.10.1 Transducer Behaviour

Most physical sensors have some form of time-dependent behaviour. **Bladed** allows the user to simulate both a simple first order lag and second order behaviour:

First Order Lag

Time Constant	
Lag Time Constant (τ)	0.3 s

$$\tau \frac{dx(t)}{dt} + x(t) = u(t)$$

- $u(t)$ = Input (black)
- $x(t)$ = Response (red)
- τ = Time constant

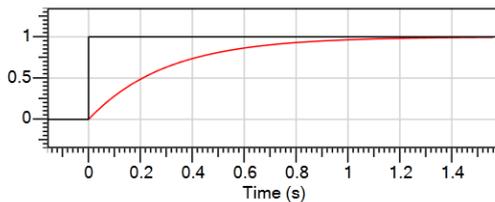


Figure 5-1: A simple lag, where the transducer response tends towards the inputted simulated value.

Second Order Lag

Parameters	
Frequency (f)	1.0000000000000007 Hz
Damping factor (ζ)	0.8 -

$$\frac{d^2x(t)}{dt^2} + 2\zeta\omega \frac{dx(t)}{dt} + \omega^2x(t) = \omega^2u(t)$$

- $u(t)$ = Input (black)
- $x(t)$ = Response (red)
- ω = Frequency (rad/s) = $2\pi f$
- ζ = Damping factor

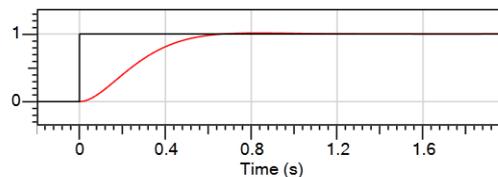


Figure 5-2: A lag with damping, where the transducer response can overshoot the inputted simulated value.

Note that even if transducer behaviour is defined, it will not be added to the signal if the signal quality is set to "Simulated Value".

5.10.2 Noise

Uniform (rectangular distribution) or Gaussian noise can be added to the signal. In order to be reproducible, a random number seed can be provided. For a rectangular distribution, specify the half-width of the distribution (i.e. the maximum error). For a Gaussian distribution specify the standard deviation.

Noise is only applied if the signal quality is "As Measured".

Note that the noise (the only random element of the measured signal) can be turned off for all measured signals at once with an option on the main signal properties screen.

5.10.3 Discretisation

The discretisation step for each signal may also be specified – the signal is discretised after any random noise has been added.

Discretisation is only performed if the signal quality is “As Measured”.

5.10.4 Sampling Period

A constant sampling period can be defined. This represents a signal whose value is only updated periodically – and the transducer will return the last sampled value until it takes another sample.

Periodic sampling is only added if the signal quality is “As Measured”.

5.10.5 Faults

In order to create robust controllers, **Bladed** allows the user to replicate a number of faults occurring on various signals upon which it may rely. These can either occur on the first instance of this type of controller, or on all controllers of this type.

Faults are applied if the signal quality is “With Transducer Lag” or “As Measured”.

1.1.1.1 Constant Faults

These faults represent manufacturing, assembly or installation faults with the sensor.

Installed Backwards? If true, the sensor will always return the negative of its ‘true’ value. This is to represent a sensor that has been installed incorrectly.

Inbuilt Miscalibration This is a constant value offset, to represent a faulty sensor or one that has been installed incorrectly.

1.1.1.2 Time-Dependent Faults

Time-dependent faults occur after a specified amount of time. A note is reported in the message file when the error is activated.

Change in Noise Characteristics Once triggered, this will multiply the noise element of a measured signal by a specified amount. If there is no noise defined, this fault will have no effect.

Constant Fail Value Once triggered, the sensor will return a specified fixed value. A ramp rate to this value can also be specified, in which case the value will linearly ramp between its last ‘good’ value and the constant fail value.

Constant Miscalibration Once triggered, a constant offset is applied to the signal.

Increasing Miscalibration Once triggered, the signal’s value drifts away from its ‘true’ value. The offset increases linearly over time, but the signal will still vary underneath. Note that there is no limit on the value – it can continue increasing or decreasing indefinitely.

Harmonic Interference Once triggered, a sinusoidal element is added to the signal. The properties of this interference can be defined.

5.11 The Pitch Actuator

The dynamics of the pitch actuator are an important part of the control loop dynamics for pitch controlled turbines, and must be defined before dynamic simulations can be carried out for these turbines. Click **Control** on the main toolbar followed by **Pitch Actuator**, or use the **Specify -> Control systems** pull-down menu. (For aileron or air-brake control, the pitch actuator is actually an aileron or air-brake actuator.)

Pitch actuator models can be defined as either passive or active. In the case of passive models, the motion of the pitch freedom is prescribed to follow the output of a Laplace transfer function. In an active model, an actuator torque is calculated using one or two PID controllers, and this is then applied to the pitch freedom.

The **input demand** may be defined as a position or rate demand. This must match with the controller's behaviour. The user can choose to filter the incoming demand by the use of **setpoint trajectory planning** (also known as ramp control). This enables step demands (coming for example from a discrete external controller) to be smoothed according to rate and acceleration limits. Furthermore, the rate and acceleration limits will limit the demand signal over multiple controller time-steps, which may be the case if the setpoint trajectory rate and acceleration limits are within the controller rate and acceleration limits.

Response Types

- **Use setpoint trajectory planning** – This uses the acceleration and rate limits of the setpoint trajectory planning to prescribe the pitch freedom. This can only be used when acceleration limits have been defined, otherwise infinite acceleration would be possible and the actuator torque would be undefined.
- **1st order passive** – 1st order passive dynamics are represented by transfer function.
- **2nd order passive** – 2nd order passive dynamics are represented by a transfer function.
- **Other passive** – The user may enter a custom Laplace transfer function.
- **Closed Loop PID**
 - Position Response: PID dynamics convert the position demand into a rate demand.
 - Rate Response: PID dynamics convert the rate demand into a torque or force demand, depending on the whether the actuator drive type is rotary or linear respectively.

There are three types of position limits:

- The **Standard limit switches** define the minimum and maximum pitch demands.
- When the **Safety limit switches** are tripped, the pitch motor is disengaged so that a rotary actuator drive provides no actuator torque and a linear actuator drive provides no actuator force. The brake is also applied at this time. If the brake has not been defined by the user, an internal brake is used by the calculation. The brake torque will appear in the pitch bearing friction output.
- The **end stops** model a hard physical stop on the bearing and provide a stiffness force if the pitch angle has exceeded the end stop position.

The **bearing friction** model calculates friction based on a constant value plus components based on the magnitudes of the bending moment and axial and radial forces at the blade root. The additional components of the friction value may be defined proportionally to the loads, or as look-up tables for a more detailed model. **Stiction** is also defined, which is an additional friction torque acting when the pitch is stationary.

The user may choose to not define the actuator drive details, or define them as **rotary** or **linear**:

- A **rotary** actuator system models a system of a pitch motor with a gearbox between the motor and blade ring. The user is able to specify the torque of a pitch **brake**, which is applied when a limit switch is activated. **Torque limits** are defined on the motor side of the gear ratio. These can be defined as fixed limits, as a look-up table according to pitch angle (blade side) or as a look-up table according to motor rate (motor side).
- A **linear** actuator models a system where linear motion is converted into rotary pitching motion by the use of a ram. The **pitch angle at the point of maximum torque** is the value of the pitch angle when the ram is perpendicular to the radius from the pitch axis to the connection point of the ram.
- If the user selects no actuator drive details, the drive is implemented as a rotary system with a unity gear ratio and no inertia.

5.11.1 Safety System Pitch Action

Two principle types of pitch action during a safety system event are available. For both of these, the user is able to define **torque or force limits** (depending on whether the actuator drive is rotary or linear respectively). These limits override the limits that are used in normal operation. If the user defines no torque or force limits in the safety system definition, the normal operation torque or force limits will **not** be used.

- In a **rate demand** safety system, the rate demand may be calculated from one of four methods, and this will override the demand from the controller. If the rate is to be passed by the external controller, the external controller must pass the demand by a specific function in the function API.
- In a **torque demand** (or force demand for a linear actuator) safety system, the torque or force demand can either be calculated by the external controller, or by the application of a constant torque plus torque terms dependent on the pitch angle (**spring torque**) and pitch rate (**damper torque**). The torque or force demand is applied directly to the actuator drive so must be defined on the motor side of the gear ratio in a rotary actuator.

5.12 The shaft brake

Click the **Brakes** button, either on the **Transmission** screen or in the Supervisory Control section of the **Control Systems** screen, to define the dynamic characteristics of the shaft brake.

The mechanical shaft brake is used for simulations of stopping or parked rotors. Up to three shaft brake characteristics may be defined, e.g. for different torque settings, or if different numbers of callipers are used. Shaft brake 1 is applied during the in-built normal (see [5.14](#)) and emergency stop calculations (see [5.15](#)), and at the start of a parked simulation (see [5.17](#)). The other brake definitions can be applied through the external controller (see [A.1 The Bladed External Controller API](#)) or by means of the safety system (see [5.19](#)).

When the brake is applied, it usually takes a little time to reach full torque. Two ways are provided for specifying how the brake torque develops. Select **Linear Ramp** or **Look-up Table** by clicking the appropriate button:

Linear ramp model

The brake torque increases linearly from zero up to full braking torque. Enter the following data in the boxes provided:

- **Maximum shaft brake torque:** the full braking torque, and
- **Shaft brake ramp time:** the time to reach full torque starting from zero.

Look-up table

The braking torque may be specified as a time history of braking torque against time since the brake application was initiated. Use the **Add** button to add points to the time history.

See Drive train ([4.8.2](#)) to specify the position of the brake. See also Mechanical Losses ([4.11.1](#)), which are assumed to result in an additional braking torque.

See also: Normal stop sequence ([5.14](#)), Emergency stop sequence ([5.15](#)), Parked conditions ([5.17](#)).

5.13 Start-up sequence

For simulations of start-up, click the **Start** button on the **Control** screen and specify the following parameters:

- **Initial rotor speed:** the rotor speed at the start of the simulation.
- **Initial pitch angle:** the pitch angle at the start of the simulation.

The built-in start-up logic may be used, in which case the following parameters should be entered to define the sequence of events:

- **Initial pitch rate during start-up:** the constant pitch rate which will be used during the start-up sequence.
- **Generator speed at which generator is put on line:** As soon as the generator speed reaches this value, the generator is connected to the grid and the power production control (see [5](#)) starts to act. However any pitch angle demanded by the controller will be ignored until the initial pitch ramp is complete, as defined by:
- **Final pitch angle in start-up mode:** The **Initial pitch rate during start-up** continues to act until this pitch angle is reached, at which point any pitch angle demanded by the power production controller starts to be used.

Alternatively, click **Defined by external controller** to use your own start-up logic.

Notes:

1. The sign of the Initial pitch rate must be consistent with the initial and final pitch angles: if the final pitch angle is less than the initial pitch angle, the pitch rate must be negative, and vice versa.
2. Pitch angles and rates apply also to aileron/flap/airbrake deployment angles. They are ignored if the rotor has no aerodynamic control surfaces (see [4.1](#)).

5.14 Normal stop sequence

Normal stops use aerodynamic braking and/or the shaft brake to stop the rotor from the normal running condition. For simulations of normal stops, click the **Normal stop** button on the **Control** screen. To use the built-in logic for normal stops, specify the following parameters:

- **Pitch rate:** the constant pitch change rate used during the stop sequence.
- **Maximum pitch:** the final pitch angle. If the Pitch rate is negative, the Maximum pitch should actually be the most negative pitch angle. The initial pitch angle will depend on the running conditions before the stop occurs.
- **Rotor speed for cut-in of shaft brake:** when the rotor has slowed down to this value, the shaft brake (see [5.12](#)) is applied to bring the rotor to rest.

The normal stop begins at the **Time to begin a stop** specified in Simulation control (see [7.14](#)).

Alternatively, click **Defined by external controller** to use your own normal stop logic. Note that the **Time to begin a stop** (see [7.14](#)) does not apply in this case.

Notes:

1. Pitch angles and rates apply also to aileron/flap/airbrake deployment angles. They are ignored if the rotor has no aerodynamic control surfaces (see [4.1](#)).
2. User-defined supervisory control may also be defined in an external controller (see [5.9](#)).

5.15 Emergency stop sequence

The emergency stop calculation models the complete shutdown of a turbine by the safety system in the event of a loss of load due to an unexpected grid loss. Modern turbines attempt a "ride through" in the case of grid loss, so this calculation is rarely used in loads calculations unless it is for a legacy project. See section [5.19](#) for the functionality to model a modern safety system.

If this calculation were to be used, it models the safety system's actions to bring the rotor to a full halt. A traditional safety system of this type will attempt to stop the rotor by feathering the blades to utilise aerodynamic braking and/or the use of a brake within the drivetrain. The triggering of these actions can be either directly on the occurrence of a grid loss (the "Grid Loss" trip mode), or by the increase of the rotor speed beyond a certain threshold (the "Overspeed" trip mode).

For simulations of emergency stops, click the **Emergency stop** button on the **Control** screen. To use the built-in logic for emergency stops, specify the following parameters:

- **Emergency pitch trip mode:** Select Overspeed if aerodynamic braking is to be triggered by an overspeed, or Grid loss if it is to be triggered immediately on loss of load.
- **Rotor speed at which pitch motion commences:** The overspeed condition which triggers the aerodynamic braking if Emergency pitch trip mode is set to Overspeed.
- **Emergency pitch rate:** the constant pitch change rate which is used following the Grid loss or Overspeed.
- **Maximum pitch:** the final pitch angle. If the Emergency pitch rate is negative, the Maximum pitch should actually be the most negative pitch angle. The initial pitch angle will depend on the running conditions before the stop occurs.
- **Emergency shaft brake trip mode:** Select Overspeed if the shaft brake is to be triggered by an overspeed, or Grid loss if it is to be triggered immediately on loss of load.
- **Rotor speed for cut-in of brake (speed control):** The overspeed condition which triggers the shaft brake if Emergency shaft brake trip mode is set to Overspeed.
- **Rotor speed for cut-in of brake (for parking):** when the rotor has slowed down to this value, the shaft brake is applied to bring the rotor to rest if it has not already been applied as a result of grid loss or overspeed..

The emergency stop begins at the **Time to begin a stop** specified in Simulation control (see [7.14](#)).

Alternatively, click **Defined by external controller** to use your own emergency stop logic. Note that the **Time to begin a stop** (see [7.14](#)) does not apply in this case.

Notes:

1. Pitch angles and rates apply also to aileron/flap/airbrake deployment angles. They are ignored if the rotor has no aerodynamic control surfaces (see [4.1](#)).
2. User-defined supervisory control may also be defined in an external controller (see [5.9](#)).

5.16 Idling conditions

For simulations of an idling rotor, click the **Idling** button on the **Control** screen and specify the fixed **pitch or aileron angle** to be used while idling. Click **Enable external controller** if any user-defined control logic is to be used.

5.17 Parked conditions

For simulations of a parked rotor, click the **Parked** button on the **Control** screen and enter:

- the fixed **pitch or aileron angle** to be used when parked, and
- the **rotor position** or azimuth angle when parked (note that this position is used at the start of the simulation, but it is possible for the brake to slip, allowing the azimuth to change during the simulation). A zero azimuth angle means that the first blade is pointing vertically upwards.

Click **Enable external controller** if any user-defined control logic is to be used.

5.18 Yaw control

5.18.1 Yaw Dynamics

Four yaw dynamics options are available. Click the **Yaw control** button on the **Control Systems** window and select the appropriate option:

None: the nacelle direction will remain fixed.

Rigid yaw: the nacelle follows the demanded nacelle angle (see [5.18.2](#)) precisely.

Flexible yaw: the nacelle yaws passively with respect to the demanded nacelle angle (see [5.18.2](#)) as a result of aerodynamic forces.

Controlled torque: the yaw actuator torque is specified by the external controller (see [5.18.2](#)).

If any yawing is allowed, specify the **Yaw friction** torque

If **Flexible Yaw** or **Controlled torque** is selected, you may specify any

Additional stiction. The nacelle will only start to yaw if the balance of applied yawing torques at the yaw bearing exceeds the friction plus the stiction torque, and it will stop yawing if the balance of torques falls below the friction torque. The applied torques include aerodynamic and inertial M_z torques and either the spring/damper torque (in the case of **Flexible yaw**) or the **Controlled torque**.

If **Flexible Yaw** is selected, specify the:

Yaw Damping (if any), and either the

Yaw Stiffness (if any) for a linear spring model, or select the hydraulic accumulator model.

Note that using the external controller, it is possible to control the yaw damping and stiffness and also to provide additional friction by applying a yaw brake torque.

The following parameters are required to model yaw compliance provided by a hydraulic accumulator system:

Gas volume (V): the volume of gas in the accumulators

Nominal pressure (P): the equilibrium pressure in the hydraulic system

Pump flow per unit yaw (F): the volume of fluid which must pass through the yaw motor to achieve a one radian change in nacelle angle

Yaw torque per unit pressure (Q): the relationship between the pressure difference across the yaw motor and the torque developed at the yaw bearing

Gas law constant: the constant γ for the gas in the accumulator. In the gas law equation,

$$PV^\gamma = RT$$

Specify $\gamma = 1$ for isothermal conditions.

The yaw torque Y provided by the hydraulic system is then given by:

$$Y = Q.P \left(\left[\frac{V}{v_1} \right]^\gamma - \left[\frac{V}{v_2} \right]^\gamma \right)$$

where $v_1 = V - F.\alpha$, $v_2 = V + F.\alpha$, and α is the difference between the actual and the demanded nacelle angle.

5.18.2 Active Yaw

Three choices are available for defining the demanded nacelle angle in simulations:

- **None:** the demanded nacelle angle will be fixed at at the initial angle. The default is zero, i.e. pointing North.
- **Prescribed manoeuvre:** specify a yaw manoeuvre by entering:
 - **Time to start yaw manoeuvre:** when the demanded nacelle angle starts to move
 - **Required yaw position change:** the angle through which it will move
 - **Yaw rate for yaw manoeuvres:** the yaw rate to be used.
- **External controller:** if **Rigid yaw** or **Flexible yaw** is selected (see [5.18.1](#)), the user-defined controller will generate a yaw rate demand which defines the **demanded nacelle angle** at any instant. If **Controlled torque** is selected, the user-defined controller (see [5.9](#)) will generate a yaw actuator torque demand. This will be the same as the actual yaw actuator torque, since the dynamic response of the yaw torque actuator is assumed to be fast, and is therefore not represented in the model.

5.19 Safety System

The safety system module allows a series of trips and subsequent actions to be specified to model the turbine safety system. The safety system consists of a number of hard-wired trips and relays, arranged in one or more fail-safe circuits which act independently of the turbine controller, and are intended to bring the turbine safely to rest in the event of a problem which the normal control system cannot handle.

To define the safety system from the **Bladed** toolbar, use **Specify... Control Systems... Safety system**.

5.19.1 Safety System Circuits

One or more safety system circuits can be defined. These circuits can be tripped by various safety system trips or by the external controller. For each circuit, specify the actions that take place when the circuit is tripped. These actions include fail-safe pitch action, disconnecting the generator, applying shaft brakes, and disconnecting the yaw drive.

5.19.2 Safety System Pitch Action

Two options are available for the safety system pitch action: **Constant rate demand** and **Torque demand**. These actions override the pitch demand from the controller. There are three possible components to the torque demand:

- **Constant torque**
- **Spring torque**, which takes the form of a look up table of pitch angle to torque.
- **Damping torque**, which takes the form of a look up table of pitch rate to torque.

In addition, a variable torque or force limit can be specified, as a function of the distance moved by the actuator since the start of the safety system pitch action. This can be used to represent, for example, the decrease in hydraulic pressure in a failsafe hydraulic actuator as the accumulator discharges.

5.19.3 Safety System Trips

Six different trips are available. For each trip, specify which safety circuit it will activate.

There are four sensor trips: rotor speed, generator speed, generator power after electrical losses and nacelle vibration, which uses the magnitude of the horizontal nacelle acceleration vector. Specify the sensor trip level and the delay before the circuit is activated.

There is also a generator short-circuit (see [7.20.2](#)) trip, also with a specified delay, and an operator emergency stop button which can be activated at a specified time in the simulation.

Check **Only activate safety system once output logging begins** to prevent safety system activation during any transients at the beginning of the simulation before any output has been recorded.

5.20 LIDAR

LIDAR (light detection, imaging and ranging) is a laser Doppler anemometer system by which laser beams mounted on the turbine measure the wind field ahead of the turbine. This information can be passed to the controller, to implement a feed-forward control system. The instruments collect data by emitting light waves from transducers according to some pre-determined scanning pattern. Particles suspended in the air cause backscatter of the emitted waves. The Doppler shift in the frequency of the backscattered light is used to determine the wind speed along the direction of the LIDAR beam, at a particular focal point. This information can be logged in the control system and used to enhance the control strategy to provide load reduction.

The LIDAR definition can be accessed via **Specify->Control Systems...->LIDAR**. The LIDAR definition allows the specification of multiple beams that take velocity measurements at intervals and pass the data to the external controller. All the options in the LIDAR screen only affect the LIDAR definition and the information passed through to the external controller, and do not affect any other environmental or structural definitions within Bladed.

When using turbulent wind with a LIDAR sensor, the **evolving turbulence** feature can be used so that the wind variations measured by the LIDAR will change before reaching the turbine. This requires **two wind files** to be defined and is more realistic than the usual assumption of "frozen turbulence" (see Section [6.7](#)).

5.20.1 Laser beam direction

This section explains how a user can define the LIDAR beam direction, relative to a mounting component, in the user interface. The LIDAR beam is specified using a LIDAR axis which describes the orientation of the LIDAR system relative to the mounting node. The beam direction is then defined relative to the LIDAR axis and this can vary depending on the scanning pattern; or remains fixed on the LIDAR axis if the fixed scan pattern is specified.

The user can add multiple LIDAR beams in the user interface. These can be mounted on the nacelle, hub or at a blade station. The coordinate system used depends on the component. The rotating hub coordinate system is used if mounted on the Spinner, the yaw bearing coordinate system is used if mounted on the nacelle and the root axes coordinate system is used if mounted on a blade station. A position offset and the nominal (undeflected) LIDAR axis is specified relative to the mounting component. The LIDAR axis is orientated by a rotational quaternion and is specified using two angles; the inclination angle θ and the azimuth angle ψ . The transformation is carried out by first inclining the LIDAR axis about the y axis, then rotating about the x axis by the azimuth angle, as shown in [Figure 5-3](#).

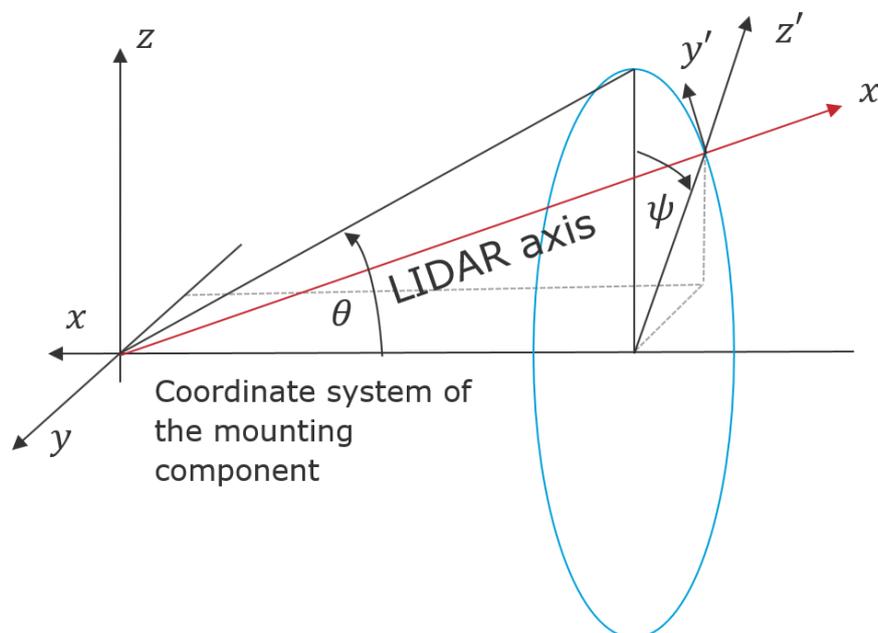


Figure 5-3: Visualisation of the LIDAR axis defined in the mounting component coordinate system

The direction of each beam will change at run-time according to the motion of its mounting component, the LIDAR axis and the scanning pattern defined by the user or external controller. Static and dynamic deflections of the mounting component will also affect the LIDAR beam's orientation and position. For instance, if mounted on the spinner, the LIDAR system will be inclined if the hub has a tilt angle. Likewise, if the nacelle is yawed, the LIDAR system will also be yawed accordingly and if mounted on the blade, the LIDAR system will rotate as the rotor azimuth angle varies.

A laser beam can be steered to scan an area in front of the turbine referred to as a scan pattern. The scan pattern describes the variation in the beam direction relative to the LIDAR axis with respect to time. The beam direction is defined by a spherical coordinate system defined using two angles α and β , as denoted in [Figure 5-4](#). The angles α and β represent the inclination to and the azimuthal rotation about the LIDAR axis.

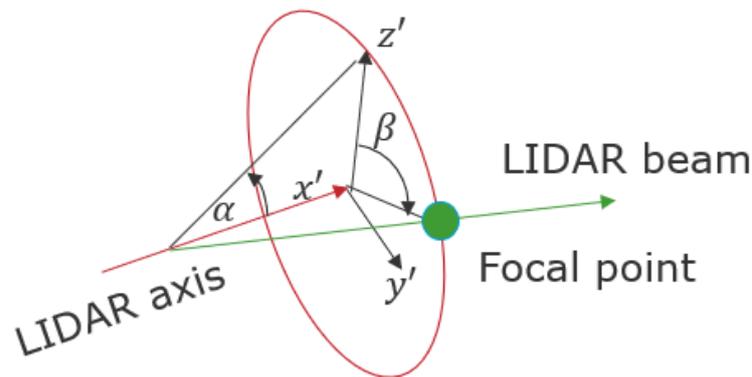


Figure 5-4: LIDAR beam direction defined relative to the LIDAR axis coordinate system

5.20.2 LIDAR scan patterns

The user can apply one of the pre-defined scanning techniques, fixed position, circular scanning or rosette scanning. If the fixed scan pattern is selected, then the beam direction is coincident with the LIDAR axis.

The number of samples per complete scan N_s , should be specified by the user along with the sampling rate T_s . The period of each scan is then given by $T = T_s \cdot N_s$.

If a scanning pattern is selected when there is more than one beam defined, the same scanning pattern is assumed for all beams. This limitation can be avoided, if the scan pattern is defined by the external controller.

For more details on the scan patterns that can be used, please consult the Theory Manual.

5.20.3 Line of sight wind speed calculation

At each sample interval t , the LIDAR beam orientation is updated and then the wind module is interrogated to obtain the wind speed along the direction of the beam. This section outlines the calculation process that is used for each beam. The line of sight speed reported is relative to the motion of the mounting component.

The focal distance(s) can be set directly in the user interface, or by the external controller. This is the distance from the LIDAR instrument along the beam direction, at which the wind module is interrogated to obtain a line of sight wind speed.

In reality, a single precise wind speed at the focal point is not recorded, but rather the weighted average of velocities around the focal point according to a weighting function. For a pulsed beam, this can be specified by the user, as a look-up table. For a continuous beam, the weighting function is defined by the lens area and beam wavelength (see the Theory manual for further details).

If there is more than one LIDAR beam, then the LIDAR beam sampling can be made sequential, in which case a measurement is only taken from a single beam per LIDAR sample. The order upon which the readings are taken is determined by the order in which the lidar beams were inputted in the user interface (top to bottom is first to last). If simultaneous beam sampling is selected, then wind speed measurements will be taken from all beams on every LIDAR sample.

In order to compute the line of sight velocity reading for beam the following algorithm is used:

1. The beam direction is updated relative to the LIDAR axis as per the scanning pattern.

2. The orientation of the mounting component, the LIDAR axis and the beam direction are used to define the direction of the LIDAR beam in the inertial frame (global frame).
3. The focal position of the beam is computed (in the global frame) using the mounting location of the lidar instrument, the distance along the beam and the direction of the beam.
4. The wind velocity is computed by interrogating the wind module at the focal position, the velocity of the mounting component is subtracted from this value.
5. The dot product of the lidar beam direction and the wind velocity is computed to get the line of sight velocity.
6. If a weighting function is defined, then steps 3 to 5 are repeated at each distance from the focal point and the result is multiplied by the corresponding weighting. The product of the line of sight speed and weighting is summated and then normalised to compute a final line of sight velocity.

5.20.4 LIDAR and external controller interface

To make use of the LIDAR feature, the user will need to write an external controller DLL. By default, Bladed will not log any values from the LIDAR system. To output any states or measurements made by the LIDAR system, the relevant channels should be added to the External Controller outputs group. Please consult the external controller documentation regarding this feature. The relevant external controller commands that can be used are documented in the ExternalControllerAPI.chm file, which can be found in the Bladed installation directory. The sections below provide more details on the implementation of the external controller LIDAR commands.

5.20.4.1 Setting up LIDAR system using external controller

The external controller can be used to define the following LIDAR system attributes:

1. LIDAR range: distance to the focal point(s)
2. LIDAR beam angles: the scanning pattern
3. LIDAR sampling rate: interval between successive LIDAR measurements

If the sample rate is defined by the external controller, then it is recommended that the value specified be the same as or greater than the external controller time step. The external controller can also inspect some other properties of the LIDAR system, see the ExternalControllerAPI.chm file for further details. Some aspects of the LIDAR model such as the LIDAR type (pulsed/continuous), the weighting function, beam mounting component, offset and centreline orientation cannot be specified by the controller.

5.20.4.2 LIDAR outputs from external controller

The external controller is commonly used to output the LIDAR beam orientation and line of sight velocity readings. The orientation data is returned using the following coordinate system defined in [Figure 5-5](#). The angle Y is the rotation about the y-axis between the negative x-axis of the LIDAR beam. Similarly, the angle Z is the rotation about the z-axis between the negative x-axis and the LIDAR beam. The line of sight velocity reading is the component of the measured wind speed, at a particular focal point, which is parallel to the LIDAR beam as defined by its measured Y and Z angles.

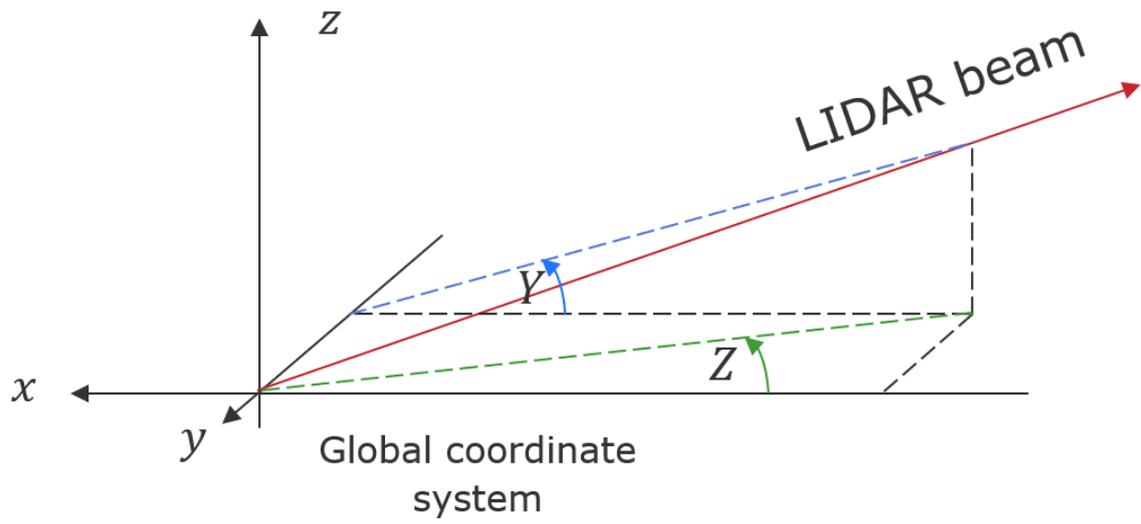


Figure 5-5: Visualisation of the angles Y and Z returned by external controller API calls.

6 DEFINING THE ENVIRONMENT

This section describes how to define the environmental conditions in which the turbine operates. The environmental conditions include wind (Section [6.1](#)), sea state (Sections [6.12](#) to [6.14](#)) and seismic conditions (Section [6.15](#)).

6.1 Defining the wind

Click the **Wind** icon on the toolbar to define the characteristics of the wind field.

For the calculation of steady state parked loads and for all simulations, define the steady-state spatial distribution of the wind. For simulations, define also the time-varying wind characteristics.

The steady-state characteristics of the wind field may include any combination of the following elements:

- Wind shear (see [6.2](#)): the variation of wind speed with height.
- Tower shadow (see [6.3](#)): distortion of the wind flow by the tower.
- Upturbine wake (see [6.4](#)): full or partial immersion in the wake of an upstream turbine.

For simulations, wind speed may vary with time in addition to the steady-state spatial characteristics defined above. Click on **Time Varying Wind** and choose one of the following options:

- No variation (see [6.5](#)): for simulations in which the wind speed and direction are constant in time.
- Single point history (see [6.6](#)): to supply a time history of wind speed and direction which is fully coherent over the whole rotor.
- 3D turbulent wind (see [6.7](#)): use a 3-dimensional turbulent wind field with defined spectral and spatial coherence characteristics. See Generating turbulence fields.
- Transients (see [6.8](#)): to use sinusoidal wind speed and direction transients as defined by certain standards, such as IEC 1400-1.

These options are described below, starting at Section [6.5](#). In each case the wind characteristics are defined at a specified height, and the wind speed at other heights will be modified according to the shear profile. Click **Refer wind speed to hub height** to ensure that the definitions always apply at the hub height of the turbine.

Click on **View wind data** for a graphical representation of the wind field. The options available include:

- Time history plots for selected positions in the rotor plane
- 3D carpet plots and animations showing the wind field over the whole rotor plane

The variables which may be displayed include:

- Longitudinal, lateral and vertical wind speeds
- Resultant horizontal wind speed
- Total resultant wind speed
- Wind direction and upflow angle.

This data may also be tabulated as for other graphical data (see [9.9](#)).

The time history plots optionally include moving-averaged values if a gust average period is specified. Click **Stats** to see the mean wind speed and the maximum and minimum values of the wind speed and the moving-average gust wind speed. Click **Max Gust Table** to view a table of maximum moving-average gust values at a grid of points, after first specifying the spacing of the grid points required. Note that this calculation may be quite slow.

6.2 Wind shear

Wind shear is the variation of steady state mean wind speed with height. If wind shear is required, click the **Wind** icon on the toolbar and select **Wind shear**. Then select either the exponential or the logarithmic wind shear model, or provide a look-up table to specify a user-defined shear profile.

Exponential model:

Enter the wind shear exponent. The wind speed $V(h)$ at height h above the ground is then given by:

$$V(h) = V(h_0) \left(\frac{h}{h_0} \right)^\alpha$$

where

h_0 is the reference height, and

α is the wind shear exponent.

A zero exponent results in no wind speed variation with height.

Logarithmic model:

Enter the ground roughness height parameter. The wind speed $V(h)$ at height h above the ground is then given by:

$$V(h) = V(h_0) \left(\frac{\log(h/z_0)}{\log(h_0/z_0)} \right)$$

where

h_0 is the reference height, and

z_0 is the ground roughness height.

User-defined profile:

Enter a lookup table giving a wind speed multiplication factor as a function of height. The multiplication factor would normally be 1.0 at the reference height.

6.3 Tower shadow

Tower shadow defines the distortion of the steady-state mean wind field due to the presence of the tower. Click the **Windicon** on the toolbar and select **Tower shadow**, and select a model.

Potential flow model:

This model is appropriate for rotors operating upwind of the tower. Enter the **Tower diameter correction** factor, F . The longitudinal wind velocity component around the tower is modified using the assumption of incompressible inviscid flow around a cylinder of diameter $F \cdot D$, where D is the tower diameter at the height where the tower shadow is being calculated.

Empirical model:

For rotors operating downwind of the tower, use this empirical model, which uses a cosine bell-shaped tower wake. Enter the following parameters:

- **Maximum velocity deficit** at the centre of the wake as a fraction of the local wind speed.
- **Width of the tower shadow** as a fraction of the local tower diameter.
- **Reference position:** the distance downwind, as a proportion of the local tower diameter, at which the above parameters are defined. At other distances, the shadow width increases and the velocity deficit decreases with the square root of the distance from the tower.

Combined model:

This model uses the potential flow model at the front and sides of the tower, and the empirical model on the downwind side wherever it gives a deficit greater than the potential flow model deficit, with a smooth transition between the two. This is useful in simulations where the rotor might yaw in and out of the downwind shadow area.

6.4 Upwind turbine wake

There are two ways to model a wake originating from an upwind turbine in Bladed:

- The '*Upwind turbine wake*' tab in the '*Wind*' screen (to be deprecated)

- The 'Dynamic upwind wake' screen in *Specify->Wind*

From these options, the latter is recommended for site-specific calculations as the user-inputs in general do not need to change between runs in a set of site-specific load calculations, because they are independent of operating condition. The former has several user entries relating to the upwind turbine (e.g. tip speed ratio), which are dependent on operating condition. This formulation will be deprecated in a future release.

In order to define simple, static *Gaussian* wake, this must be done in the 'Upwind turbine wake' tab, where the user must enter:

- **Centre line velocity deficit:** as a percentage of the local wind speed.
- **Wake half-width:** the distance from the wake centre line at which the deficit is reduced to $e^{-0.5}$ times the centre line value.

The 'Dynamic upwind wake' screen allows the user to generate a wake at the start of the simulation based on an eddy-viscosity model and also meander the wake centreline throughout the simulation.

The wake deficit profile is calculated at the start of the simulation and is constant throughout. The user must provide a steady operational loads relevant to the upwind turbine over all operating points. The initial wake profile is calculated from the axial induction factors of the aerodynamic information group in these steady operational loads results at the wind speed that corresponds to the mean wind speed of the current simulation. It is therefore important that aerodynamic information is output at every blade station otherwise the initial wake profile will be very coarse. This information can also be obtained from an aerodynamic information simulation but then the information will not be valid for multiple wind speeds.

The initial wake profile is propagated downstream according to the thin shear layer approximation of the Navier-Stokes equations, which is documented in the theory manual. The distance of propagation and the eventual centreline offset at the modelled turbine is determined by the relative global location (North and West) of the upwind turbine and the simulation mean wind direction, which is shown as an example in [Figure 6-1](#).

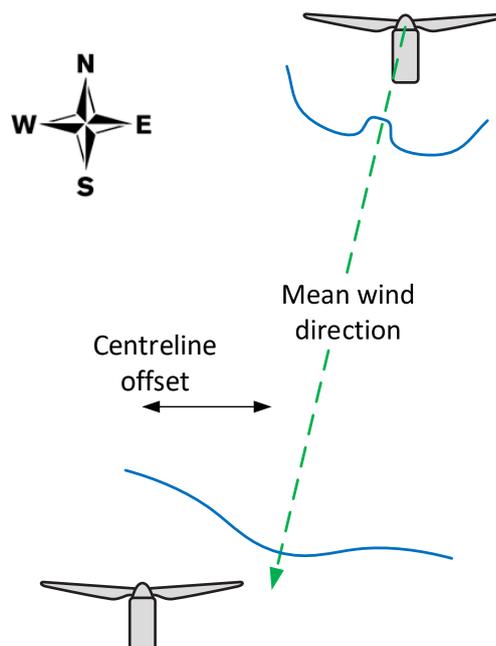


Figure 6-1: Propagation of a wake profile from upwind turbine to modelled turbine

There is also the option for a *meandering wake* where the wake centreline meanders through the simulation. The motion of the wake is calculated by integrating the lateral and vertical components of a user-specified turbulence file. This turbulence file should be low-pass filtered in order to avoid high frequency content in the meandering. The *speed of wake propagation relative to ambient wind speed*

affects the lateral and vertical meander speeds from the file and also affects the length of axis over which the meander is integrated. Therefore, a slower speed of propagation will generally lead to larger meander.

6.5 No time variation of wind speed

Use this option to specify the wind conditions for a simulation when no variation of the wind field with time is required. Any steady-state spatial variations defined by wind shear, tower shadow and an up-turbine wake will apply.

Click the **Wind** icon on the toolbar, select **Time varying wind**, and choose the **No variation** option.

Enter the following parameters:

- **Wind speed:** the steady wind speed to be used in the simulation.
- **Height at which speed is defined:** the reference height to which the **wind speed** applies, unless **Refer wind speed to hub height** has been selected. If wind shear (see [6.2](#)) is defined, the wind speed at any other height will be different.
- **Wind direction:** measured clockwise from North - see wind direction (see [6.9](#)).
- **Flow inclination:** for non-horizontal flows (e.g. on the side of a hill). A positive value indicates a rising flow.

6.6 Single point wind history

Use this option to specify the wind conditions for a simulation when an arbitrary time-varying wind speed and/or direction is required, but no spatial variation is required other than the steady-state characteristics defined by wind shear, tower shadow and an up-turbine wake.

Click the **Wind** icon on the toolbar, select **Time varying wind**, and choose the **Single point history** option.

Enter the following parameters:

- **Height to which speeds relate:** the reference height to which the wind speed time history applies, unless **Refer wind speed to hub height** has been selected. If wind shear (see [6.2](#)) is defined, the wind speed at any other height will be different.
- **Flow inclination:** for non-horizontal wind flows (e.g. on the side of a hill). A positive value indicates a rising wind.

Then use the **Add** button to add points to the time history. For each point, enter

- **Time:** this should start from zero.
- **Speed:** the wind speed at that time.
- **Wind direction:** measured clockwise from North - see wind direction (see [6.9](#)).

6.7 3D turbulent wind

Use this option to perform a simulation in which the turbine is immersed in a turbulent wind field which varies both in space and time. The turbulence is superimposed on any steady-state spatial variations defined by wind shear, tower shadow and any upwind turbine wake. The turbulent variations at any point in the rotor disk have defined spectral characteristics, and the variations at any two points in the rotor disk are correlated by a defined coherence relationship representative of the spatial structure of real atmospheric turbulence. The turbulent wind field may have just the longitudinal component of turbulence, or it may have all three components.

Click the **Wind** icon on the toolbar, select **Time varying wind**, and choose the **3D turbulent wind** option.

Then enter the following information:

- **Turbulent wind file name:** Click here to select a file which contains an appropriate turbulent wind field: see Generating turbulent wind fields (see [6.10](#)). Click **Properties** to display the characteristics of the turbulent wind field. The windfield should be high enough and wide enough to envelop the whole rotor, and long enough to allow the simulation to continue for as long as is required. If U is the average wind speed corrected to hub height according to the wind shear, and the simulation is to run for a time T , the turbulence field length should be $U.T + \text{Turbine diameter}$, or just $U.T$ if the turbulence file is allowed to wrap around (see below). It may be longer than this, although in this case the mean wind speed and turbulence intensity for the simulation may not match the values selected, as they apply only if the whole file is used.
- **Mean wind speed:** the mean wind speed for the turbulent wind at the reference height. If this does not match the wind speed for which the turbulence characteristics were defined, the turbulence field will be scaled appropriately. However, since the dimensionless characteristics of turbulence are not quite invariant with wind speed, this scaling is not strictly valid.
- **Height at which speed is defined:** the reference height to which the **Mean wind speed** applies, unless **Refer wind speed to hub height** has been selected. If wind shear (see [6.2](#)) is defined, the mean wind speed at any other height will be different.
- **Turbulence intensity:** the standard deviation of turbulent wind speed variations as a percentage of the mean wind speed. If using a three-component wind field, specify the turbulence intensity for each component.
- **Wind direction:** measured clockwise from North - see wind direction.
- **Flow inclination:** for non-horizontal flows (e.g. on the side of a hill). A positive value indicates a rising wind.
- **Allow turbulence file to wrap around:** if this is checked, the turbulence file can be recycled indefinitely by looping from the end back to the beginning.

If a LIDAR sensor (see [5.20](#)) is used to provide wind preview information to the external controller, the upstream wind velocities measured by the LIDAR cannot be expected to convect towards the turbine unchanged as implied by Taylor's frozen turbulence hypothesis. Select '**Evolving turbulence**' to allow the turbulence to change as it convects towards the turbine, and then provide the following additional information:

- **Additional wind file name:** Select a second file containing a turbulent wind field, which should be identical to the first turbulent wind field apart from the random number seed - see Generating turbulent wind fields (see [6.10](#)). While the turbulence reaching the turbine is still given by the first turbulent wind field, the turbulence measured by the LIDAR is calculated by combining frequency components from both wind fields depending on the upstream distance of the measurement point. Note that this can result in significantly slower simulations, especially if there are many simultaneous LIDAR measurement points and/or many points are defined for the LIDAR weighting function.

For the **Height of turbulent wind field**, select one of the following options. These do not affect the mean wind speed at any height, only the location of the turbulent wind fluctuations:

- **Centred on hub height:** The centre of the turbulent wind field is placed at hub height.
- **Best fit for rotor and tower:** If the turbulence field vertical dimension is less than the height of the turbine, the top of the turbulence field will be located at the height of the top of the rotor, so that the turbulence field extends as far down the tower as possible. If the turbulence field vertical dimension is greater than the turbine height, the turbulence field will start at ground level and will envelop the whole turbine.

Turbulent variations outside the defined turbulence field will be taken from the nearest defined point.

The turbulence is defined at a number of grid points (see [6.10](#)), both in the rotor plane and in the alongwind direction. In between these points, either linear or cubic interpolation can be used to determine the turbulence. Specify one of the following options for the interpolation scheme:

- **Linear:** uses linear interpolation
- **Cubic in rotor plane only:** uses cubic interpolation laterally and vertically, and linear interpolation in the alongwind direction
- **Fully cubic:** uses three-dimensional cubic interpolation.

- **Cubic interpolation for Lidar measurement:** If 'Evolving turbulence' has been selected, this option specifies that cubic interpolation in the y-z plane will be used for both wind fields when calculating the wind velocity at any LIDAR measurement point, otherwise linear interpolation will be used. Note that evolving turbulence can result in slow simulations, in which case selecting cubic interpolation for the Lidar measurement would make them even slower.

If desired, specify also a superimposed sinusoidal wind direction transient. This is particularly useful if only a single turbulence component is being used. Enter the following parameters:

- **Amplitude of direction change:** the peak-to-peak amplitude of the transient. Note that the specified wind direction (see above) refers to the start of the simulation. The mean wind direction will be different from this if a transient is added.
- **Start time of transient:** the time into the simulation at which the transient starts.
- **Duration of transient:** the duration of the transient from the time it starts to the time it finishes.
- **Type of transient:** half or full wave: see 6.8.

6.8 Transients

Use this option to define sinusoidal wind speed and direction transients such as are defined in certain standards, such as, for example, IEC 1400-1. These transients are superimposed on any steady-state spatial variations defined by wind shear, tower shadow and an up-turbine wake.

Click the **Wind** icon on the toolbar, select **Time varying wind**, and choose the **Transients** option.

First enter the following data:

- **Reference height** at which the wind speed is defined, unless **Refer wind speed to hub height** has been selected. If wind shear (see [6.2](#)) is defined, the wind speed at any other height will be different.
- **Flow inclination:** for non-horizontal wind flows (e.g. on the side of a hill). A positive value indicates a rising wind.

Separate transients may be defined for each of the following variables:

- **Wind speed:** defined at the **reference height**.
- **Wind direction:** measured clockwise from North - see wind direction (see [6.9](#)).
- **Horizontal wind shear:** defined as the difference in wind speed between the two sides of the rotor plane. The wind speed varies linearly with horizontal distance across the rotor plane.
- **Vertical wind shear:** defined as the difference in wind speed between the top and bottom of the rotor plane. The wind speed varies linearly with height.
- **Vertical direction shear:** defined as the difference in wind direction between the top and bottom of the rotor plane. The wind direction varies linearly with height.

All defined transients will be used, but if any is not required, specify an appropriate start value and a zero amplitude. For each transient, enter the following information for the variable in question:

- **Start value:** the value of the variable before the start of the transient.
- **Amplitude of change:** the largest range from maximum to minimum during the transient.
- **Time to start cycle:** the time into the simulation at which the transient starts.
- **Time period of cycle:** the duration of the transient.
- **Type of cycle:** Three types of transient are provided, as specified by various standards. A **half wave** transient begins with the start value and increases sinusoidally by the amount defined by the Amplitude. A **full wave** transient begins with the start value and changes sinusoidally by the amount defined by the Amplitude in the first half of the time period, changing back to the start value by the end of the time period. The **IEC-2** transient is a more complex shape, as described in the Theory Manual.

6.9 Definition of wind direction

The wind direction is the direction from which the wind is coming, measured clockwise from North when looking from above, i.e. East of North.

The default nacelle orientation and the wind direction together define the nacelle yaw angle. This may change during a simulation if any yaw control (see 5.18) is specified, i.e. passive yaw, active yaw control or a yaw manoeuvre. Note that the yaw angle can also be defined explicitly for the steady parked loads (see 7.10) calculation.

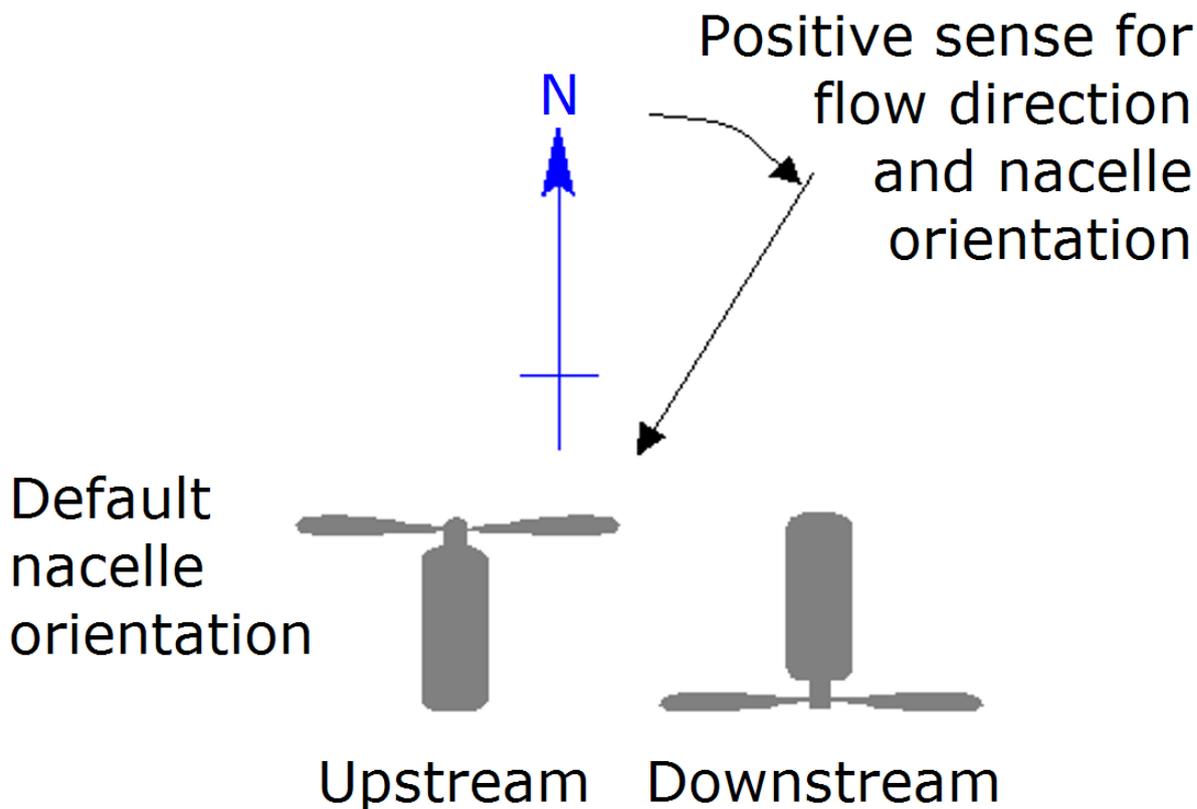


Figure 6-2: Default nacelle orientation

The yaw misalignment angle is defined as:

$$\begin{aligned} \text{Yaw misalignment} &= \text{nacelle angle} - \text{flow direction} && (\text{for } 0^\circ \leq \text{nacelle angle} \leq 180^\circ) \\ \text{Yaw misalignment} &= (\text{nacelle angle} - 360^\circ) - \text{flow direction} && (\text{for } 180^\circ < \text{nacelle angle} \leq 360^\circ) \end{aligned}$$

Note that the calculated "Nacelle yaw displacement" output variable found under "Nacelle motion", is defined in terms of the measured nacelle angle (positive sign convention shown in figure 6.2) as follows:

$$\begin{aligned} \text{Nacelle yaw displacement} &= - \text{nacelle angle} && (\text{for } 0^\circ \leq \text{nacelle angle} \leq 180^\circ) \\ \text{Nacelle yaw displacement} &= 360^\circ - \text{nacelle angle} && (\text{for } 180^\circ < \text{nacelle angle} \leq 360^\circ) \end{aligned}$$

6.10 Generating turbulent wind fields

In order to run a simulation which uses a 3-dimensional turbulent wind field, a suitable turbulence field must first be generated. Once generated, these turbulence fields are stored so that they can be used for future simulations.

Turbulence fields contain time histories of wind speed variations at each of a number of points on a rectangular grid covering the turbine rotor swept area. The time history at each point will have spectral characteristics conforming to one of the available turbulence models. The time histories at any two points in the rotor plane will be correlated with each other in accordance with an appropriate model of the lateral and vertical coherence characteristics of atmospheric turbulence. Linear or cubic interpolation is used between grid points, as well as between successive time points.

To generate a new turbulence field, first define the turbulence characteristics and then generate the turbulence field as described below.

6.10.1 Defining turbulence characteristics

Click the **Wind** icon on the toolbar and select **Define turbulence**. Then define the following:

- **Number of points along Y:** the number of grid points horizontally.
- **Number of points along Z:** the number of grid points vertically.
- **Length of Y:** the width of the turbulent wind field - this should be at least the rotor diameter, or twice the greatest distance from the tower centreline to the blade tip to allow for operation at non-zero yaw angles.
- **Length of Z:** the height of the turbulent wind field - this should be at least the rotor diameter, or hub height plus radius if the turbulence is required all the way down the tower.
- **Duration of wind field:** this should be sufficient for the length of simulation required. The 'frozen' turbulence field moves past the turbine at the specified hub height mean wind speed. The product of the duration and the wind speed gives the length of the turbulent wind field which will be generated. See 3D turbulent wind for details.
- **Frequency along X:** the resolution to be used for the time histories. This is used to calculate the distance between successive points, which move past the turbine at the appropriate wind speed.
- **Mean wind speed:** this should correspond to the hub height mean wind speed to be used for the simulations. In fact the turbulence characteristics are used in dimensionless form which means that they only depend weakly on wind speed. This means that the mean wind speed used here to generate the turbulence need not be the same as the wind speed used in a subsequent simulation which uses the turbulence field, since the turbulence will be re-scaled accordingly. However, for strict conformance to the specified turbulence characteristics, generate a different turbulent wind field, using the correct mean wind speed, for each mean wind speed to be used in the simulations.
- **Turbulence seed:** this is a seed for the random number generator, and should be an integer between 0 and 999. Change the seed to generate a different turbulence field with the same statistical characteristics.
- **Spectrum type:** select Kaimal, von Karman or Mann, and click **Define...** to enter the detailed turbulence characteristics, as follows:

Kaimal model: three models are available:

1-component model: enter the lateral, vertical and longitudinal length scales of the longitudinal component of turbulence.

IEC-2 model: enter the scale parameter (Λ_1 as defined in IEC 61400-1 Edition 2).

General model: enter the longitudinal length scales for the three components of turbulence, the coherence scale parameter L_c , and the coherence decay constant H . Note that this model encompasses the IEC 1400-1 model by setting $^xL_u = 8.1\Lambda_1$, $^xL_v = 2.7\Lambda_1$, $^xL_w = 0.66\Lambda_1$, and in the case of Edition 2 $L_c = 3.5\Lambda_1$, and $H = 8.8$, while for Edition 3 $L_c = 8.1\Lambda_1$, and $H = 12$.

von Karman model: the **improved model** is easy to use. This conforms with ESDU data items 85020 and 86010. Enter:

Number of components: 1, 2 or 3 components of turbulence can be generated.

Height above ground: The hub height is recommended.

Surface roughness length: For the terrain.

Latitude: Turbulence characteristics are only weakly dependent on latitude except near the equator, where the model breaks down.

The appropriate length scales and turbulence intensities are calculated and displayed, although it is possible to specify different turbulence intensities when setting up a simulation run.

The **old von Karman model** allows either 1 or 3 components of turbulence to be generated. All the turbulence length scales must be entered by the user. This model conforms to ESDU 74031.

Mann model: enter the following parameters:

Shear parameter (γ): Zero gives isotropic turbulence; the recommended value for IEC edition 3 is 3.9, which produces a spectrum similar to a Kaimal spectrum.

Scale length (L): The scale length is given by $L = 0.8 \Lambda_1$ (Λ_1 as defined in IEC 61400-1 Edition 3). The recommended value for IEC edition 3 is 33.6m when the turbine hub height $z > 60$ m.

FFT points: The number of points used for the FFT in the lateral and vertical directions, e.g. 32 (must be a power of 2). Cubic interpolation is used to find the wind at the requested grid points.

Max. lateral/vertical wavelength This should be appreciably greater than the rotor diameter to avoid periodicity in the lateral and vertical directions. A value of at least four times the scale length is recommended.

To start from an existing turbulent wind field definition, click **Import details...** to select an existing turbulent wind file. The parameters defining the turbulence in that file will then be loaded.

6.10.2 Advanced options

The **Advanced options** button allows the following additional functionality:

Gust iterator: this option is available only with the 'Improved von Karman' model of turbulence (see [6.10.1](#)). It will carry out an iteration in which the surface roughness is varied until a particular gust is achieved somewhere within the rotor swept area. The turbulence intensity will change with surface roughness in accordance with the model. The following parameters are required:

- **Gust wind speed, tolerance, and averaging time:** the iteration will continue until the specified maximum gust, defined as a block average with the specified averaging time, is achieved within the specified tolerance somewhere within the rotor swept area. Gusts of the specified speed and up to the specified tolerance above this speed are accepted.
- **Hub height and rotor diameter:** these define where the rotor swept area is in relation to the lateral and vertical extent of the wind file. Click **Get current values** to input the values currently available on the **Rotor** (see [4.1](#)) screen.
- **Height of turbulent wind field:** together with the hub height, this defines the vertical position of the rotor within the wind field – see [6.7](#).
- **Minimum and maximum allowed values of surface roughness:** the iteration will fail if the required gust cannot be achieved with a surface roughness between these limits.
- **Minimum and maximum alongwind position of gust:** these can be used to ensure that the maximum gust is not too close to the start or end of the wind file.
- **Wind shear:** Wind shear can be taken into account when finding the maximum gust in the wind file. In addition wind shear can be applied to the target gust speed. If this second option is chosen, the target gust speed will vary with height, such that the required gust is higher near the top of the rotor; otherwise the target gust speed is equal at all heights. The wind shear may be specified in terms of a shear exponent or a roughness length. Click **Get current values** to input the shear model as currently defined on the **Wind** screen.

- **Number of files to generate:** If required, this calculation can generate more than one file with different random number seeds which match the specified criteria. In this case, the resulting wind files will be called **name_n.wnd**, where **name** is the run name selected, and **n** is a sequential number.

Matching a measured single-point time history: this option is available only with the Kaimal and von Karman models of turbulence. It generates a wind file in which the wind speed time history at a specified point in the rotor plane exactly matches a given, perhaps measured, time history. The coherence function associated with the chosen turbulence model is used to generate appropriate time histories at all the other points in the plane, but the single-point spectrum associated with the chosen turbulence model is not used, since the measured data defines this. The measured data may include more than one component of turbulence: any unmeasured components should be provided as zeros, and the program will then synthesise these components entirely, using the single-point spectrum of the chosen turbulence model. The following information must be supplied:

- a file of measured data, with one column for each component specified in the chosen turbulence model (using zeros for any components for which measured data is not available).
- The mean wind speed for the measured data.
- The sampling frequency of the measured data.
- The grid point at which the simulated turbulence should match the measured data.

The duration of the measured dataset must be at least as great as the duration of the wind file which is to be created. The sampling frequency should of course be sufficient to capture the wind speed variations in sufficient detail, but need not match the frequency of points specified for the final turbulence file: the measured data is simply re-sampled as required.

6.10.3 Generating the turbulence field

Having defined the turbulence characteristics as above, click the appropriate button to generate the turbulence field either **Now** or **In Batch**. Alternatively, click the **Calculations** icon on the toolbar to bring up the Calculations screen (if it is not already present). Select **Wind turbulence**, and then click either **Run now**, or **Run in Batch** to store the calculation for later execution using the batch (see [7.2.7](#)) facility.

6.10.4 Turbulence file format

The wind file format is documented in [APPENDIX F TURBULENCE FILE FORMAT](#). This information is useful for users who wish to generate their own turbulence file.

6.11 Annual wind distribution

Some post-processing calculations require the annual distribution of hourly mean speeds to be defined. This includes the Annual Energy Yield calculation as well as any calculations which produce weighted lifetime values, for example of fatigue damage or damage equivalent loads.

Click the **Wind** icon on the toolbar and select **Annual wind distribution**. Select either a Weibull or a user-defined distribution. For the Weibull distribution, enter the annual mean wind speed and shape factor. A shape factor of 2 results in a Rayleigh distribution. For the user-defined distribution, enter the cumulative distribution or exceedance table, starting with an exceedance of 1.0 at zero wind speed. For wind speeds above the last point in the table, the exceedance will be assumed to decay in proportion to $\exp(-v^k)$, where v is the wind speed and k is the number entered as the "Exponent for high wind speeds".

Click either **Cumulative plot** or **Probability plot** to view the wind speed distribution as defined.

To define a joint distribution of wind and waves, select the **wind and wave** option, and click **Define scatter diagram** (see [6.15](#)).

6.12 Defining Waves

Click the **Sea state** icon on the toolbar and select the type of wave model required:

- **None** (see [6.12.1](#)): no waves are required.
- **Regular waves** (see [6.12.4](#)): choose Linear Airy or Stream Function waves of a single frequency.
- **Irregular waves** (see [6.12.2](#)): choose either a standard Jonswap or Pierson Moskowitz spectrum, or enter a user-defined spectrum. This will generate a stochastic wave train as a superposition of linear Airy waves with the appropriate spectral distribution.

For all wave types, enter:

- **Direction of approach (from North)**: the bearing from which waves arrive at the tower. Like wind direction, wave direction is defined as the direction which the waves are coming *from*, and not the direction that the waves are going *to*. The angle increases positively to the East of North.

Then enter the appropriate data for the chosen model as described below.

In the case of irregular waves, it is also possible to add a "Constrained wave" (see [6.12.3](#)), which guarantees that the irregular wave train will contain an extreme wave of a specified height.

Any specified waves will be ignored if the turbine is not offshore - see [4.5.5](#).

6.12.1 None (no waves)

In this case the tower is not subjected to wave loading. However, if the turbine is specified as being offshore, it will still experience hydrodynamic damping and a drag force from any specified currents.

6.12.2 Irregular waves

Irregular waves are generated as a superposition of linear Airy waves with the appropriate spectral distribution. Enter:

- **Random number seed**: an integer to be used as the seed value for the random wave generation. A different seed value will generate a different realisation of the irregular waves, but still conforming to the desired wave spectrum.

To define the wave spectrum, choose either **Jonswap / Pierson-Moskowitz spectrum** or **User-defined spectrum**, and specify the diffraction approximation to use – see below.

Jonswap / Pierson-Moskowitz spectrum

This standard wave spectrum is described in the Theory Manual, and requires the following parameters:

- **Significant wave height**: corresponding to the average height of the highest one third of the waves in the seastate. Given a time-history of wave height, this parameter can be calculated as four times the standard deviation of the water surface elevation.
- **Peak spectral period**: the period of the most energetic component in the wave spectrum.
- **Peakedness parameter**: this parameter controls the width of the frequency band containing most of the energy in the spectrum. It should take a value between 1 and 7. For a Pierson-Moskowitz spectrum, the peakedness parameter should be set to 1.

User-defined spectrum

If the wave spectrum is known at the site of interest, select this option and then click **Define spectrum**. A further window opens which allows up to 100 pairs of values of frequency and power spectral density to be entered. Data entry boxes may be created and deleted by clicking on the **Add** and **Delete** buttons respectively. Data points are automatically sorted by frequency. Clicking on the **Show** button will reveal a plot of the spectrum as defined. The values of power spectral density at the lowest and highest frequencies entered should be zero.

Wave diffraction approximation

Two options are available to account for diffraction effects at small wavelengths comparable to the size of the structure (see Theory Manual):

- MacCamy-Fuchs approximation: enter a representative member diameter, or select Auto-define to allow **Bladed** to select a suitable value.
- Simple cut-off frequency: enter a frequency, or select Auto-define to allow **Bladed** to select a suitable value.

6.12.3 Constrained waves

An irregular wave history can be modified to include a prescribed extreme wave at a particular time. Two methods are available (see Theory Manual):

- **Linear NewWave**: the linear airy waves are constrained in such a way as to produce the desired extreme wave height at the specified time, or
- **Stream Function**: a non-linear stream function wave is blended into the irregular wave history.

In each case, enter:

- **Constrained wave height**: measured from trough to crest; in the case of **Linear NewWave**, the trough elevation is taken as the lower of the troughs on either side of the crest.
- **Time of constrained wave**: determines at what point in the simulation the constrained wave occurs

For the Stream Function method, enter also the

- **Constrained wave time period**.

6.12.4 Regular waves

This option generates a regular wave train, also sometimes used as a way of obtaining extreme deterministic waves.

Choose the type of regular wave model:

- **Linear Airy** gives a simple linear wave of the specified period
- **Stream Function** gives non-linear stream function waves, which are more appropriate for extreme waves.

In either case, the following parameters are required:

- **Wave height**: defined from trough to crest.
- **Wave period**.

6.12.5 SEA file

This option in the "Tide and SEA file" tab defines the sea-state based on a pre-processed SEA file. A SEA file can be generated by clicking the "Generate SEA File" button. It consists of a list of regular wave components which are linearly superimposed to form the irregular sea state. Each regular wave component is specified in terms of four properties:

- **Frequency** – the reciprocal of the period of the wave. The number of frequency components generated depends both on the "duration" variable D , and on the frequency cut off values selected. The frequency resolution $df = 1/D$, so the longer the duration, the smaller the frequency steps between adjacent frequency components. Harmonic components with frequencies that lie outside of the **frequency range** indicated in the user interface will be omitted. If 'Automatic Frequency Range' is selected (this is the default option), 0.05% of spectral energy at each end of the spectrum will be omitted so that only 0.1% of energy is lost overall. Alternatively it is up to the user to specify the range of frequencies to be used. If the user enters their own values, they need to ensure that the most important features of the spectrum are retained, both in terms of frequency range, and energy distribution. In some cases for instance, it may be acceptable to cut off high frequency 'tails'.
- **Amplitude** – With the deterministic **amplitude method**, the amplitude is determined by the spectral density. Random amplitudes are calculated according to a Rayleigh distribution with the scale parameter proportional to the spectral density.

- **Direction** – assigned randomly according to the **directional distribution** about the **mean direction**. More details on the directional distributions are given in the theory manual.
- **Phase** – a uniformly distributed random variable.

The **random number seed** is used to initiate the sequence of random numbers used for the phases, directions and amplitudes (if applicable). If **regular** waves are chosen rather than a JONSWAP spectrum, then there is only one component in the SEA file.

The SEA file format can be edited using standard text editing tools. It is therefore possible to write or even modify existing SEA files externally to Bladed such that constant energy spectra or constrained waves can be simulated using SEA files. While the user is free to select the range and resolution of frequencies used to define the power spectral density of the sea state the user should verify that the required statistics/properties of the input wave spectra are satisfied for the free surface elevation time-history created (for example, spectral shape, Hmax , etc). It would also be important to make sure that the possible frequency responses of the structure will be covered (i.e. not omitting any important frequencies of the structure response).

Definitions of the fields used to define the **JONSWAP** spectrum can also be found in section [6.12.2](#), and detailed theory is given in the theory manual, as well as definitions for the different period parameters.

6.13 Defining Currents

To define currents, click on the **Sea state** icon on the toolbar and then select the Currents tab. Three current components may be specified, separately or in combination:

- near-surface current
- sub-surface current
- near-shore current

If more than one current is selected, the velocity vectors are added linearly. The form of each current profile is described in the Theory Manual. Any specified currents will be ignored if the turbine is not offshore (see [4.5.5](#)).

Click the check boxes to select the required current components, and enter the relevant parameters for each as described below:

6.13.1 Near-surface current

This is a current whose strength decreases linearly with depth. Enter the following parameters:

- **Surface velocity.**
- **Reference depth:** the depth at which the flow velocity reaches zero.
- **Heading (degrees from North):** the direction towards which the current is flowing. Unlike wind and wave directions, current direction is defined as the direction that the current is flowing *to*, and not the direction that the current is coming *from*.

6.13.2 Sub-surface current

This is a current whose strength depends on height above the sea bed according to a power law or a user defined relationship. Enter the following parameters:

- **Surface velocity.**
- **Heading (degrees from North):** the direction towards which the current is flowing. Unlike wind and wave directions, current direction is defined as the direction that the current is flowing *to*, and not the direction that the current is coming *from*.

Normally a 1/7 power law is used, but there is also an option to enter a different power law exponent x such that $V(h) = V(\text{surface}) * (h/\text{sea depth})^x$, where $V(h)$ represents flow speed a distance h above the sea bed. The default is $x = 1/7$.

Alternatively, a custom shear profile can be defined by the user by selecting the "Custom Shear Profile" option. The desired shear profile can then be entered in a look up table.

To define the custom shear profile, the current shear at each height above the sea bed must be specified.

The height above sea bed data must be entered as values normalised by the mean sea depth for the simulation. Note that the mean sea depth for a simulation is the sum of the mean sea depth specified in the support structure definition screen and the tide height for the simulation.

$$\text{Normalised height above sea bed} = \frac{\text{height above sea bed}}{\text{mean sea level} + \text{tide height}}$$

The current shear at each height is defined as the mean current speed at that height divided by the mean sea surface current speed.

$$\text{Current shear} = \frac{\text{mean current speed at each height}}{\text{mean sea surface current speed}}$$

The values in the first row of the look-up table represent the normalised height above sea bed and the current shear values at the sea surface. The values in this first row therefore always each take the value 1. Subsequent data points must be entered in descending order of normalised sea surface height.

6.13.3 Near-shore current

This is a current which is constant with depth. Enter the following parameters:

- **Current velocity.**
- **Heading (degrees from North):** the direction towards which the current is flowing. Unlike wind and wave directions, current direction is defined as the direction that the current is flowing *to*, and not the direction that the current is coming *from*.

6.14 Tide height

The mean sea level is defined on the Tower screen (see [4.5.5](#)), but for any particular calculation the water depth may be changed by specifying the height of the tide. Click the **Sea state** icon on the toolbar, select **Tide** and specify the change in water depth compared to mean sea level.

6.15 Scatter diagram

Users with a licence for the **Advanced Processing** module may use a scatter diagram to set up multiple simulations containing the correct combinations of wind and wave parameters, and to post-process these simulations using the extreme load extrapolation option. To define the scatter diagram, from the toolbar menu select **Specify ... Sea State ... Scatter**.

The scatter diagram is used to define the distribution of wave height and period as a function of wind speed, for use in setting up multiple simulations and extreme load extrapolation of off-shore turbines.

The probability density of each wind speed bin is the summation of the probability density in the appropriate set of wave bins. The total probability density should not be greater than 1.0.

6.16 Earthquakes

Users with a licence for the Seismic module may run simulations which include an earthquake. From the toolbar menu, select **Specify... Earthquake** to bring up the **Earthquake definition** screen. Select the option to specify an earthquake to use in simulations, and browse for an earthquake file. Specify the **time** in the simulation at which the earthquake will strike, and the **principal direction** of the earthquake.

An earthquake file is a text file containing two header lines and a time history of earthquake ground accelerations. The header format is:

```
POINTS n
TSTEP  t
```

where *n* is the number of time points and *t* is the timestep. The header is followed by two columns of numbers representing time histories of ground acceleration in the principal direction and perpendicular to this direction. Each column must contain *n* entries.

6.16.1 Generating earthquake time histories

Users with a licence for the Seismic module may generate earthquake files from a standard earthquake response spectrum.

Open the **Earthquake definition** screen, and select **Generate new earthquake**. Click **Import...** to start from an existing definition.

Click **Define response spectrum** to specify the response spectrum as a look-up table, and then enter the following parameters:

- **Response spectrum damping coefficient:** This is the damping coefficient of the structure for which the response spectrum is defined. The damping used in the earthquake generation is not a physical characteristic of the structure or the soil, but is the damping that was used to calculate the seismic response spectrum and as such it is inherent to the spectrum definition and it is not correct to use different damping values for the same response spectrum. It could be understood as the damping coefficient of the earthquake measuring device.
- **Random number seed:** different values will result in different earthquake realisations, but in each case the resulting earthquake will conform to the specification as defined.
- **Duration of earthquake:** The length of time for which the earthquake will persist.
- **Earthquake time step:** the timestep of the ground acceleration time histories which will be produced.

The accelerations will be in a given fixed direction, the principal direction of the earthquake. However, if desired, an additional component at 90° to the principal direction can be specified. This may be proportional to the acceleration in the principal direction (which simply corresponds to scaling and rotating the accelerations by a fixed amount), or independent.

Certain **advanced parameters** may also be specified, to control in detail the way in which the earthquake is generated:

- **Accuracy of fit:** the number of frequency points can be specified at which the actual response spectrum can be specified, will be compared against the target response spectrum. The allowable deviation from the target response spectrum can also be specified.
- **Filters:** a low and high pass filter can be defined to remove frequencies outside the range of interest. This may help to speed up convergence.
- **Shaping function:** This controls the overall shape of the earthquake. Two models are included. These are typical for **Stiff** soils or **Soft** soils. Some seismic standards specify the length of the **Stationary** part of the earthquake. These can be entered using the **Soft** option, along with a final **Decay constant**.

6.17 Point loading

On the **Point Loading** screen, the user is able to apply a pre-defined loading onto any point on the tower or support structure or the blade. To access the screen, use the **Specify** and then **Point Loading** menu item. **Bladed** uses a file that contains a time-history of loading in a specified format. The time-history of loading can either have a single loading component and be applied at any horizontal direction or have full six directions of loading (3 forces and 3 moments). The load histories must be specified in the global coordinate system. The user can specify more than one point loading to be applied during a simulation.

- **Tower Loading\Blade loading:** If either of these boxes are checked then the currently entered settings will be applied during the simulation.

To add a point loading description, press the **+** symbol next to the **Loading Set**. Then enter the following parameters:

- **Impact data file:** Browse for the location of the file containing the specified time-history of loading. The file must specify the number of points in time at the top of the file. For the single component case, the order of columns is (Time//Force), separated by a tab. For the six component case, the order of columns is (Time//Fx//Fy//Fz//Mx//My//Mz). The values in the time column do not need to be equally spaced. The loads should be in SI units i.e. N and Nm. For example:

NUMPOINTS	5
0.0	0.0
0.1	58.8
0.25	95.1
0.3	65.1
0.4	58.8

NUMPOINTS	7					
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.1	0.3	122.0	122.0	2.1e8	2.1e8	2.1e8
0.2	0.5	122.0	122.0	4.3e8	4.3e8	4.3e8
0.6	0.4	122.0	122.0	3.5e8	3.5e8	3.5e8
1.4	0.2	122.0	122.0	1.1e8	1.1e8	1.1e8
2.5	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0

- **Height:** If the *tower* is defined as *tubular axisymmetric*, then the height of the loading above ground or sea-level is entered here and otherwise not used.
- **Tower member:** If the support-structure is defined as *multi-member*, then this is the member number.
- **Distance along member:** Only for *multi-member* support structures. Distance from *End 1* as defined in the *tower* screen.
- **Number of components:** Either one or six, to correspond with diagrams above.
- **Direction:** Only for single component. Horizontal direction, clockwise from North.
- **Start time:** The time during the simulation that the loading from the impact file starts to be implemented. The left column of the impact file is additional to the start time value.
- **Blade number:** Only for *blade loading*. The blade number that the loading is to be applied to.
- **Distance along blade:** Only for *blade loading*. The distance along the neutral axis that the loading is to be applied to.

6.18 External loads DLL

Bladed 4.9 and higher offers the option to define externally applied loads to the multibody model using an External Loads DLL. This feature allows the user to apply external loads that are a function of time and kinematics of the structure.



The External Loads DLL can be defined under **Specify** and then **Additional Items**.

This feature is shipped with a separate user guide that is included in the installation directory of **Bladed**.

7 EXECUTING WIND TURBINE CALCULATIONS

This section describes how to set up the calculation parameters which define how a particular calculation will be carried out and how to carry out the calculations themselves.

Section 7.1 describes the modal analysis calculation which is used to determine the frequencies and mode shapes of the blade and tower vibration modes. The remaining sections describe the various turbine performance and loading calculations themselves.

7.1 Structural flexibility

The structural formulation in **Bladed** is based on linear finite element bodies, enhanced by non-linear geometric stiffness models that account for the effect of small structural deflections on dynamics. For flexible structures, modal reduction via the Craig-Bampton method is applied to the tower and (optionally) to the blade.

The modal analysis calculates *uncoupled* blade and tower modes. In the subsequent simulations, the modes from different bodies are coupled together by the equations of motion, so that the actual eigenfrequencies of the coupled system are different from the uncoupled mode frequencies. The Model Linearisation (see [7.11](#)) calculation allows the coupled modal frequencies to be calculated.

7.1.1 Multi-part blade model

In **Bladed** 4.7 and earlier, the blade was modelled using a single finite element body as shown in [Figure 7-1](#). Using a single body results in fast simulations as the number of degrees of freedom to model the blade deflections is small, and the freedom frequencies are relatively low. This approach often gives an accurate representation of blade dynamics. However, for this approach to be valid, the deflections of the blade must be small.

Many modern blade designs are very flexible, meaning that the small deflection assumptions in the linear model become less valid. This can lead to inaccuracies in predicting the blade dynamic response, in particular the blade torsion.

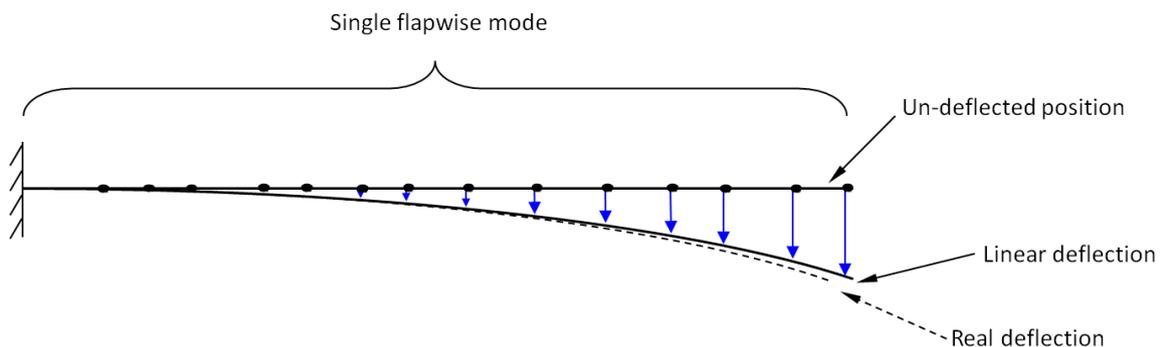


Figure 7-1: Single-part blade model

From **Bladed** 4.8, the blade can be sub-divided into many rigidly connected finite element bodies, in order to more rigorously model large blade deflections.

The outer blade parts can undergo rigid body rotation based on the deflection and rotation of the more inward parts, as well as including linear deflections within the part. The deflections within each linear part are therefore smaller than if using a single linear blade part. Splitting the blade into several parts allows for non-linear load transfer between each linear part, and a more accurate model of the blade deflection.

[Figure 7-4](#) shows a schematic of modelling a blade using two linear parts.

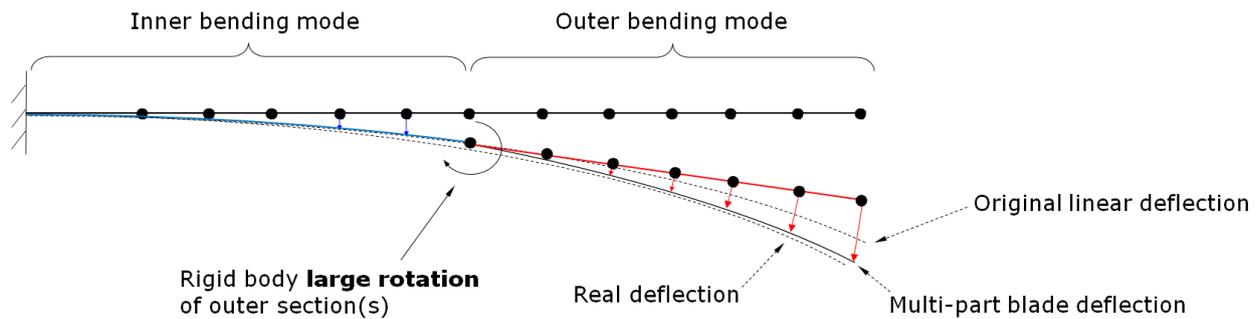


Figure 7-2: Two-part blade model

7.1.2 Flexibility Modeller screen

The structural flexibility model for the blade and support structure is defined in the *Flexibility Modeller* screen, as described in the rest of this section.

7.1.2.1 Flexibility Model inputs

In the *Flexibility Model inputs* section, the user can define inputs in order for **Bladed** to calculate uncoupled modes for the blades and support structure.

Tower

The support structure is modelled as a single finite element body. The user can specify the number of mode shapes to be calculated for the support structure to represent the structural deflection and dynamics. Enough modes should be included to capture the tower dynamics up to the frequencies of interest.

Blade

The blade can be sub-divided into many rigidly connected finite element bodies, in order to rigorously model large blade deflections. For maximum accuracy, enough blade parts should be specified to ensure that deflections remain small (less than $\sim 5-7$ degrees) within each blade part.

The structural deflection within each blade part can be accounted for either by calculating mode shapes, or by using the finite element degrees of freedom directly in the simulation. Modal reduction within the blade parts generally results in faster simulation than using the finite element model directly. If using modal reduction on the blade parts, careful attention should be paid to the number of modes on each part, to ensure that sufficient modes are included to capture the blade dynamic response.

Note that for blades with more than one part, it is recommended to use a fixed step integrator such as Newmark- β or Generalised- α to improve simulation speed. Further details in section [7.15](#).

Running modal analysis

With the Flexibility Model inputs defined, a modal analysis calculation can be performed by clicking "Run modal analysis". This action will automatically close the screen; the screen can be re-opened to view the results once the modal analysis calculation is complete.

If modal reduction is selected for the blade, then mode shapes are calculated for each blade part. If modal reduction is not selected for the blade, then it is not necessary to calculate mode shapes on each part as the finite element model is used directly. In both cases, **Bladed** also performs a subsequent eigen analysis to calculate the coupled mode shapes and frequencies corresponding to the natural modes of the whole blade, based either on the blade part modes or the underlying finite element model. This is useful both for physical interpretation of the blade mode shapes and for applying damping, as explained in the next section.

Note that one way to ensure that you have specified sufficient modes on each blade part is to check that the whole blade mode frequencies are very similar when using modal reduction and the finite element model on the blade parts.

7.1.2.2 Modal analysis results and damping inputs

In the *Modal analysis results and damping inputs* section, the modal analysis results (i.e. modal frequencies and mode shape animations) can be reviewed. A damping ratio can then be specified for each mode.

Tower

The uncoupled tower modes frequencies are displayed and mode shape animations can be viewed. The damping ratio for each uncoupled tower mode can be specified as a percentage of critical damping.

Blade

The user can view the “whole blade mode shapes” and also assign damping ratios for each whole blade mode. **Bladed** will then automatically calculate the damping values to assign to each individual blade part.

The user can choose how many whole blade modes to specify damping ratios for using the “modes with damping defined” field. Any higher frequency modes will be assigned frequency proportional damping, based on the damping of the highest mode with damping defined. This approach applies high damping to high frequency modes and effectively excludes the higher whole blade modes from the blade dynamic response, so that they do not cause erroneous instability. Further explanation is available in Theory Manual section 3.2.1.1 *Specifying blade damping*.

7.1.2.3 Geometric stiffness

Geometric stiffness models in **Bladed** account for the effect of small deflections on structural dynamics. Further details are given in [6]. The geometric stiffness setting is chosen in the *Simulation flexibility settings* section of the *Flexibility Modeller* screen.

Tower

There are two geometric stiffness settings available for the tower.

- **Axial loads only:** The effect of internal forces along the element axis on dynamic response is included. This accounts for “gravitational de-stiffening” in the support structure dynamic response. Geometric stiffness axial and shear forces are included when calculating the internal member loads.
- **Internal loads only:** Geometric stiffness forces are ignored for the purposes of evaluating the dynamic response, but are included when calculating the internal member loads.

Blade

There are two geometric stiffness settings available for the blades.

- **Axial loads only:** The effect of internal forces along the element axis on dynamic response is included. This primarily accounts for “centrifugal stiffening” in the blade dynamic response. Geometric stiffness axial and shear forces are included when calculating the internal member loads.
- **Full model with orientation correction:** The effect of internal axial and shear forces on dynamic response is included. This model can enhance prediction of torsion deflection in the blade. This model should be used with caution as it is only accurate when deflections within each blade part remain small (less than ~5-7 degrees). If several blade parts are used then this model can be activated to improve the accuracy of the solution, and will allow use of fewer blade parts than when using the “axial loads only” model.

7.1.2.4 Multi-part blade parameter selection

The best practice for determining the number of blade parts and modes per part, is to carry out a convergence test with respect to the blade loading and deflections. Not all configurations will lead to a converged blade model. One instance in which this can happen, is if one of the blade parts is significantly smaller than the others. The steady operational loads calculation can be used to determine whether the multi-part blade model has initialised to a suitable condition.

7.2 The calculations screen

Use the **Calculations** screen to select the calculation required, to define any parameters and options required for that calculation, and to execute the calculation. If not already present, click the **Calculations** icon on the toolbar to open this screen.

7.2.1 Calculations available

The calculations are arranged in four groups:

Supporting Calculations for modal analysis (see [7.1](#)) and generating turbulent wind fields (see [6.10](#)).
Steady Calculations (see [7.3](#)) for rapid steady-state wind turbine calculations.
Simulations (see [7.4](#)) for detailed wind turbine simulations.
Post Processing (see [8](#)) for further analysis of results.

First click on the desired calculation to select it.

7.2.2 Data required for calculations

Alongside each calculation is a coloured light showing whether the calculation can be performed:

Red light: No data available: no calculation possible
Yellow light: Some data available: no calculation possible
Green light: Sufficient data has been defined to perform the calculation

The box below shows all the data items required for the calculation, and indicates whether the relevant data has been defined or not. Double-clicking on any data item which is undefined opens the appropriate window so that the required data can be defined.

7.2.3 Calculation options

A number of machine and wind features, once defined, are available as options which can be switched on or off in some of the steady state calculations and simulations. Click the **Show options** button in the bottom left corner. The screen expands to show a list of the available options (see [7.22](#)), with an indicator for each to show whether it can be used, as follows:

No light (grey): This option is not available for the selected calculation
Red light: No data available: option will not be used even if selected
Yellow light: Some data available: option will not be used even if selected
Green light: Sufficient data is defined for the option to be used if selected

Use the check boxes to switch on or off any desired option, provided it is available for the selected calculation. If a red or yellow dot is displayed, the option will not be used even if selected, as the necessary data has not been defined: click on the option name to display all the required data items in the box below. Double-clicking on any data item which is undefined opens the appropriate window so that the required data can be defined.

7.2.4 Specifying outputs

Some of the turbine calculations can potentially generate large quantities of results, and in many cases only a few of these will actually be needed. Click the **Outputs** button on the calculation screen to specify which of the available outputs (see [7.21](#)) will be generated.

7.2.5 Calculation parameters

Click the Calculation parameters button (see [7.5](#)) on the calculation screen to change or define the specific parameters required to define the selected calculation. The following additional data may also be defined:

- Aerodynamics control (see [7.18](#)): to define the aerodynamic modelling parameters.
- Simulation control (see [7.14](#)): to define how a simulation will proceed.
- Imbalances (see [7.19](#)): to specify mass and aerodynamic imbalances.
- Initial conditions (see [7.16](#)): the rotor azimuth and nacelle yaw position at the start of a simulation can be defined.

7.2.6 Executing a calculation

Click on **Run Now** to carry out the selected calculation immediately, or **Run in Batch** to store the calculation for later execution by the batch facility. Except for certain calculations, a window appears in which a directory can be selected into which the calculation outputs will be placed. If the directory does not exist, it will be created. A run name is also required. Thus a number of calculations may be placed in the same directory, each with a different run name.

There is also the option to select ASCII or binary format for the calculation results. Binary format gives much smaller files, and should be faster and more accurate. Graph plotting, post-processing and tabulation all work with either format. If you want to see the raw numbers from binary files, you can still tabulate (see [9.9](#)) the numbers (to the screen, an ASCII file, Excel or Word).

Process Priority: On later Windows versions, selecting Low priority will prevent the batch calculations from slowing down other user actions

If a whole series of calculations is to be performed, for example a set of simulations for different load cases which might include stops and starts as well as normal running and parked conditions in a series of different wind conditions, some thought should be given to the directory structure used, and the file extensions used within each directory. For example a tree structure could be used with a sub-directory for each group of load cases. Each individual load case could then be in its own sub-directory, or different extensions could be used within the directory, for example to indicate different mean wind speeds. Bear in mind also the need for post-processing results to be stored for each load case. Some post-processing results may also be summed over a series of load cases.

Calculations may also be set off from the **Calculations** pull-down menu on the toolbar. Post-processing and supporting calculations may also be set off from the relevant data entry screen.

7.2.7 Batch processing

Some calculations, notably long simulations with high frequency dynamics, may take as long as several hours to run. Use the batch system to store the complete details of one or more calculations which can then be run at a later time, for example overnight. Use the **Run in Batch** button instead of the **Run now** button on the **Calculations** window. When all the required calculations have been added to the batch, the calculations may be started using the **Batch Viewer**, accessed from the **Batch** menu item on the main toolbar.

A number of different batch queues may be maintained. Each batch must be in a separate directory.

Please see the Batch User Manual for further details of the Batch framework.

7.3 Steady calculations

These calculations give quick results by ignoring the effects of any turbine dynamics and wind turbulence. They are thus particularly useful at the preliminary design stage.

- Aerodynamic information (see [7.6](#)): This gives the aerodynamic parameters including local aerodynamic loading at each blade station, assuming uniform steady wind flow.

- Performance coefficients (see 7.7): Power, torque and thrust coefficients are calculated assuming a uniform steady wind field.
- Steady power curve (see 7.8): This gives power, torque and thrust as a function of wind speed, assuming a uniform steady wind field.
- Steady operational loads (see 7.9): This calculates all the loads in a uniform steady wind field.
- Steady parked loads (see 7.10): Loads on the parked turbine are calculated in steady wind. As the blade position is specified, the wind field need not be spatially uniform.
- Model linearisation (see 7.11): Small perturbations are used to generate system responses, from which the dynamics of the fully coupled system can be analysed. A Campbell diagram can be generated, showing how the system frequencies change with rotational speed, or (with an additional licence) the Linear Model (see 8.19) post-processing calculation is then able to derive a linearised model of the turbine in state-space form. This is of particular value for designing controllers.
- Electrical Performance (see 0): This calculates electrical variables for a given torque speed envelope.

Various simplifying assumptions are made in the steady calculations, as summarised in Table 7-1. Note that the “aerodynamic information” and “performance coefficients” calculations only model the rotor (at zero tilt angle), whereas the other steady calculations model the whole turbine.

	Aerodynamic Information	Performance Coefficients	Steady Power Curve	Steady Operational Loads	Steady Parked Loads
Which structural components are included?	Rotor only	Rotor only	Whole Turbine	Whole Turbine	Whole Turbine
Component flexibility?	No	Yes	No	Yes	Yes
Wind direction other than 0deg?	No	No	No	No	No
Load varies with rotor azimuth?	No	No	No	Yes (azimuth = 0)	Yes
Turbine yaw included?	No	No	No	No	Yes
Tilt included in rotor aerodynamics and structure?	No	No	No	Yes	Yes
Wind shear included?	No	No	No	No	Yes
Upflow included?	No	No	No	No	Yes
Gravity load included?	No	No	No	No	Yes
Tower Shadow included?	No	No	No	No	Yes
Blade prebend and sweep included?	No	Yes	Yes	Yes	Yes

Table 7-1: Turbine model assumptions in steady calculations

7.4 Simulations

Simulations are used to calculate the loads, deflections and other parameters of the wind turbine as a function of time, using the full dynamics of the system, driven if desired by a time-varying wind (see 6.1) input.

They may be used, for example, to simulate the behaviour of a turbine for period of, say, 10 minutes at each of a number of wind speeds, so that the resulting loads can be analysed to produce a representative fatigue load spectrum for the turbine.

The following simulations may be run:

- **Power production** (i.e. normal running)
- **Normal stop**
- **Emergency stop**
- **Start-up**
- **Idling**
- **Parked**
- **Hardware test** (for users with a Hardware Test Module licence)

The full dynamics of the turbine will be used for all these simulations, together with the appropriate supervisory control (see [5](#)) parameters which define the stopping and starting logic, and the state of the machine when idling or parked. See also Yaw manoeuvre ([5.18.2](#)) to cause the turbine to yaw during a simulation.

To define the length of the simulation, and other parameters which control how the simulation proceeds, see [7.14](#).

7.5 Calculation parameters

Click the **Calculation parameters** button on the **Calculations** screen to define the parameters for controlling particular calculations. These include **general** parameters as well as parameters for defining **specific** calculations.

General parameters:

- Aerodynamics control (see [7.18](#)): to define the aerodynamic modelling parameters.
- Simulation control (see [7.14](#)): to define how a simulation will proceed.
- Imbalances (see [7.19](#)): to specify mass and aerodynamic imbalances.
- Initial conditions (see [7.16](#)): the rotor azimuth and nacelle yaw position at the start of a simulation can be defined.

Parameters for specific calculations:

- Aerodynamic information (see [7.6](#)).
- Performance coefficients (see [7.7](#)).
- Power curve (see [7.8](#)).
- Steady operational loads (see [7.9](#)).
- Steady parked loads (see [7.10](#)).
- Model linearisation (see [7.11](#)).
- Hardware test (see [7.17](#))

7.6 Aerodynamic information calculation

This is a simple steady-state calculation which outputs the following aerodynamic information at each blade station:

- **Inflow factors:** axial and tangential
- **Inflow angle**
- **Angle of attack**
- **Tip loss factor**
- **Reynolds number**
- **Lift, drag and pitch moment coefficient**
- **Relative wind speed**
- **Aerodynamic loadings:** in plane force, out of plane force, and pitching moment per unit length of blade.

Use the **Aerodynamic information** screen (see [7.5](#)) to define the following parameters for the calculation:

- **Wind speed**
- **Pitch angle**
- **Rotational speed** of the rotor.

7.7 Performance coefficients calculation

This calculation generates dimensionless power, torque and thrust coefficients for the rotor as a function of tip speed ratio, assuming a uniform steady wind.

Use the **Performance coefficients** screen (see [7.5](#)) to define the following parameters for the calculation:

- **Minimum tip speed ratio**
- **Maximum tip speed ratio**
- **Tip speed ratio step:** the tip speed ratio interval required
- **Minimum Pitch Angle:** The minimum (or only) pitch angle at which coefficients are calculated.
- **Maximum Pitch Angle:** The maximum pitch angle at which coefficients are calculated. If performance coefficients at only one pitch angle are required, leave this value the same as the minimum pitch angle.
- **Pitch Angle Step:** The step in pitch angle between the minimum and maximum that the coefficients are calculated for. If performance coefficients at only one pitch angle are required, leave this value at zero.
- **Rotational speed** of the rotor.

The windspeed will be varied to give the required tip speed ratios at the specified rotational speed. Although the performance coefficients are dimensionless, they may depend on the windspeed or rotational speed, and not just on the tip speed ratio, because of Reynolds number effects.

7.8 Steady power curve calculation

Both steady-state and dynamic calculations can be used to generate a power curve. The **Steady power curve** calculation generates the wind turbine power curve as a function of wind speed, assuming uniform steady winds. Shaft and electrical power, gearbox torque and thrust are calculated.

Use the **Power curve** screen (see [7.5](#)) to define the following parameters for the calculation:

- **Minimum wind speed:** the low wind cut-in speed
- **Maximum wind speed:** the high wind cut-out speed
- **wind speed step:** the wind speed interval required
- **Calculate pitch and speed change:** select **Yes** to use the schedule (see [0](#)) defined by the generator (see [4.10](#)) and controller (see [5](#)) to define the pitch angle and rotor speed at each wind speed. Select **No** to calculate a simple power curve at fixed pitch angle and rotor speed.

If **No** is selected, also supply the following:

- **Fixed pitch angle**
- **Fixed rotational speed** of the rotor.

7.9 Steady operational loads calculation

This calculation generates wind turbine loads as a function of wind speed, assuming uniform steady winds. This optionally includes blade, tower and other loads - see Output Control (see [7.21](#)). It also includes the calculation of partial derivatives which are useful for control design, and in particular for gain scheduling (see [5.6](#)).

Use the **Steady loads** screen (see [7.5](#)) to define the following parameters for the calculation:

- **Minimum wind speed:** the low wind cut-in speed
- **Maximum wind speed:** the high wind cut-out speed
- **Wind speed step:** the wind speed interval required
- **Calculate pitch and speed change:** select **Yes** to use the schedule (see [0](#)) defined by the generator (see [4.10](#)) and controller (see [5](#)) to define the pitch angle and rotor speed at each wind speed (see [6.1](#)). Select **No** to calculate a simple power curve at fixed pitch angle and rotor speed:

If **No** is selected, also supply the following:

- **Fixed pitch angle**
- **Fixed rotational speed** of the rotor.

7.10 Steady parked loads calculation

This calculation generates wind turbine loads for a parked rotor in steady winds. This optionally includes blade, tower and other loads - see Output Control (see [7.21](#)).

Use the **Steady parked loads** screen (see [7.5](#)) to define the following parameters for the calculation:

- **Steady wind speed**: note that although the wind speed is steady, it need not be spatially uniform.
- **Azimuth angle**: the rotor azimuth position; zero means that blade 1 is pointing vertically upwards.
- **Yaw angle**: the nacelle angle measured clockwise from North, assuming that the wind is blowing from the North: see wind direction (see [6.9](#)).
- **Wind inclination**: for non-horizontal wind flows (e.g. on the side of a hill). A positive value indicates a rising wind.
- **Pitch angle**: the pitch angle or aileron/flap/airbrake deployment angle - see Rotor (see [4.1](#)).

The calculation also allows a sweep through any one of the four above angles, i.e. azimuth, yaw, wind inclination or pitch angle. Use **Parameter to vary** to define which angle will vary. The sweep start at the value specified above for that parameter. Specify **End value** to define the end of the sweep, and **Step** to define the step size.

7.11 Model linearisation calculation

There are three possible calculations that can be performed:

- Campbell diagram
- Model linearisation – needs a 'Control' module licence
- Blade stability analysis – needs a 'Blade Stability Analysis' module licence

In order to make the linearisation azimuth independent, the following sources of periodic loading are turned off in all linearisation calculations:

- Gravity
- Tilt
- Imbalances

The wind field is uniform horizontal without shear, tower shadow or wake. All wave loading and currents are turned off. The external controller or any internal control dynamics are not used in any linearisation calculations.

7.11.1 Campbell diagram

The Campbell diagram calculates the coupled modes of the complete aeroelastic system and their properties such as frequency, damping and their composition in terms of blade modes, tower modes and other states of the system. The coupled mode frequencies are plotted against rotor speed.

There are three main modes of operation:

- **Power production** – Wind speed can be ramped from cut-in to cut-out and the steady state controller is used to determine the operating conditions at each speed.
- **Idling** – A range of rotor speeds are chosen and a wind speed is found in order to produce these rotor speeds given the idling pitch angle.
- **Parked** – The turbine is analysed with wind speed of zero. This can be useful to determine the pure elastodynamic coupled modes of the turbine.

In cases where there are a lot of high frequency modes in the system (e.g. multi-part blades), high frequency modes can be excluded from the plot by setting the **maximum frequency for plot**. The rotating modes (i.e. blade modes) can be **transformed into the non-rotating frame**. This will generate forward and backward whirling modes in the output.

7.11.2 Model linearisation

Model linearisation generates input and state perturbations, and records the resulting variations in the state derivatives and the selected outputs. This is done for a series of steady-state power production operating points. The Linear Model 8.19 post-processing calculation is then able to derive a linearised model of the turbine in state-space form. This is of particular value for designing controllers.

The possible input perturbations are wind speed, collective pitch angle demand, generator torque. Optionally, linear horizontal and vertical **shear** and **pitch angle demand on each blade** can be perturbed. These are particularly useful for individual pitch control design. In the advanced fields, the user can change **perturbation magnitudes**. This might be useful if it is found that the perturbations are too large and are therefore include too much non-linear response.

Model linearisation can also be performed over a range of **azimuth** angles for each wind speed. This is usually necessary only for one or two bladed turbines where the structural coupling has a strong azimuthal dependence.

7.11.3 Blade stability analysis

The blade stability analysis feature performs a frequency domain analysis of the turbine rotor in steady state. The analysis provides outputs of damping and frequency of all the coupled rotor or blade modes plotted against wind speed. It is a similar analysis to the Campbell diagram but the main differences are:

- Only the rotor is modelled
- Allowing analysis over a wide range of inflow conditions rather than being constrained to normal operating conditions
- Improved initial condition finding suitable for extreme conditions.

For outputs, the blade stability analysis produces a Campbell diagram plot and frequency and damping curves of all coupled modes. It is primarily the damping curves that will be of most interest as this allows the user to detect possible instabilities by finding damping curves that have negative damping.

The coupled mode frequencies and dampings are plotted against wind speed, but can also be plotted against rotor speed by using an output channel as the x-axis in the data viewer.

There are two **modes of operation** for the blade stability analysis: **tip-speed ratio tracking** and **parked**, which are described in the sections below.

7.11.4 Tip-speed ratio tracking

In this setup, the user chooses a **range of wind speeds**, in response to which the rotor will have a certain speed. The pitch angle is usually at fine or at an operational pitch angle in these simulations.

At very high rotor speeds, it often becomes difficult to find initial conditions. If the analysis reaches this point, it will complete without analysing the last few points. This allows the user to set the upper wind speed with some freedom.

The user can specify a '**torque speed gain**' which determines an opposition torque applied against the rotor aerodynamic torque (in a gearless case this is equivalent to a generator torque and otherwise equivalent to a generator torque applied on the high-speed side of the gearbox). The torque speed gain is defined as:

$$\text{Opposition torque} = \text{Torque speed gain} * (\text{rotor speed})^2$$

In order to follow the optimal mode tip-speed ratio, the torque speed gain can be set as:

$$\text{Torque speed gain} = \text{optimal mode gain} * (\text{gearbox ratio})^3$$

If the torque-speed gain is set at zero then the rotor will be in a free spin (i.e. zero generator torque).

7.11.5 Parked analysis

The user can perform a parked analysis where the rotor speed is locked at zero. Because the rotor freedom is disabled, only one blade is analysed as it is assumed that the blades do not couple. The user can select the **wind direction** and a **range of wind speeds** to do the analysis.

7.12 Electrical performance

The turbine generator and converter needs to operate in a wider range around the nominal operating condition. During operation far away from the nominal operating curve, the design limits of the electrical system may be reached. The electrical performance tool can be used to carryout electrical calculations over a wider range of operations considering grid code requirements and converter limitations. Click the **Electrical Performance** button to define the following parameters

- Minimum generator speed
- Knee generator speed
- Rated generator speed
- Rated torque
- Knee torque
- Minimum torque
- Normal operating speed tolerances
- Extreme operating speed tolerances
- Number of points to calculate
- Frequency fractional change
- Voltage fractional change
- Power factor

7.13 Pitch and speed schedule

For Steady Power Curve and Steady Operating Load calculations, the pitch and rotor speed schedule represents the appropriate steady-state pitch (or aileron) angle and rotor speed corresponding to any particular wind speed. The schedule is defined by:

- the steady-state Controller (see [5](#)) for the pitch angle, and also for the rotor speed in the case of a variable speed turbine, and
- the Generator (see [4.10](#)) for the rotor speed in the case of a fixed speed turbine.

7.14 Simulation control

Use the **Simulation control** screen (see [7.5](#)) to define various parameters which control the progress of a simulation. These parameters are defined below:

- **Output time step:** this is the time interval at which the output variables are sampled. For individual simulations these samples are stored in the output files.
- **Time to start writing output:** Some simulations may start with a small transient, if the initial conditions are not calculated to sufficient accuracy. In this case it may be desirable to ignore the first part of the simulation until the transient has settled. This parameter specifies the length of time to ignore at the start of the simulation before output samples start to be taken.

- **Simulation end time:** This parameter gives the maximum time for which the individual simulation will run. Normal and emergency stop simulations may end earlier if the rotor has stopped and any extra time has elapsed.
- **Time to begin a stop:** for normal and emergency stop simulations, this is the length of time for which normal running continues before the built-in stop logic is initiated.
- **Extra simulation time after stopping:** for normal and emergency stop simulations, this is the length of time for which the simulation continues after the rotor has come to rest.
- **Start time for turbulent wind:** this applies to simulations which use a turbulent wind (see 6.7) input. It is the time into the turbulence time history at which the simulation begins. Specify zero to start at the beginning of the turbulence history.

This parameter controls the way the simulation results are output:

- **Length of buffer:** Simulation results are held in a buffer, which is periodically flushed to the disc. Enter the buffer length, in seconds of simulation time. The default is usually satisfactory. If viewing results while the simulation is in progress, a shorter buffer helps to ensure frequent updating, but it will slow down the simulation.

7.15 Integrator settings

Bladed has several methods for time integration of the structural and aerodynamic states. Integrators are classified as either “fixed step” or “variable step” depending on whether the integrator time step can be changed during the simulation. A variable step integrator is practical as the time step can change dynamically in order to maintain simulation accuracy. However, some of the fixed step integrator methods are much faster at integrating systems with high frequency states (such as multi-part blade models).

The different integrator types and their recommended usage are discussed in this section. The integrator type is selected in the *Integrator settings* screen.

7.15.1 Variable step integrator (Runge-Kutta)

The variable step integrator in **Bladed** is a 4/5th order Runge-Kutta integrator. This default setting is generally the best option for turbine models where the blade is modelled as a single finite element body.

For multi-part blade simulations or floating turbines with an explicit mooring line model, a fixed step integrator should be considered to improve the simulation speed.

7.15.2 Fixed step integrators

Bladed has various fixed-step integrators available. These options can be found by selecting “fixed step integrator” in the *Integrator settings* screen.

In all cases a fixed integrator time step must be specified by the user. A larger step size results in faster simulation. It is necessary to ensure that the simulation accuracy is not excessively degraded by choosing too large a step size.

7.15.2.1 Newmark- β and Generalised- α

The Newmark- β and Generalised- α integrators are implemented in **Bladed** to allow efficient simulation of “stiff” structural systems. A stiff structural system includes very high and low frequency degrees of freedom; an example of this is a turbine with a multi-part blade model.

The **Newmark- β** integrator allows accurate integration of high frequency states at a large time step (e.g. 5ms) as long as the high frequency states are well damped. With the appropriate settings ($\beta \geq 0.25$, $\gamma = 0.5$), the Newmark- β integrator is unconditionally stable (i.e. stable regardless of step size) for linear systems. This results in generally good stability of the algorithm for the non-linear wind turbine system.

However, when integrating a non-linear wind turbine system, the system can still go unstable with reasonable time-step sizes. Such instabilities are generally caused by erroneous “blow up” of high-

frequency states, which in fact have little effect on simulation results when stable. It can therefore be desirable to apply algorithmic damping to the high frequency states to avoid numerical blow up. Damping can be added to the Newmark- β algorithm by adjusting the parameter γ so that $\gamma > 0.5$, however this adds damping on low frequency states as well as high frequency states, and can severely degrade the solution accuracy.

The **Generalised- α** method applies damping to high frequency states whilst minimising the damping applied to low frequency states. The level of damping applied is controlled by the “spectral radius at infinite frequency” parameter. Adding damping in this fashion is desirable as the erroneous high frequency instabilities are avoided without significantly degrading the solution accuracy at lower frequencies.

7.15.2.2 Fixed step Runge-Kutta Methods

These integrators offer lack the practicality of variable step Runge-Kutta and rarely offer performance benefits compared to Newmark- β and Generalised- α integrators as the latter are better designed for structural systems. The Midpoint method is a 2nd order Runge-Kutta algorithm, which is the integrator that is used in real time Hardware Test simulations in versions 4.7 and earlier. The Runge-Kutta 4th Order method is the classical Runge-Kutta method, which is widely used for numerical integration of ordinary differential equations.

7.16 Initial conditions

This allows some control over the initial conditions for simulations.

Initial azimuth angle: This is normally zero, but a different value may be specified here. Note that for parked simulations, the initial azimuth angle defined in the Parked Conditions (see [5.17](#)) is used instead.

Initial nacelle angle (from North): This is normally zero, i.e. with the turbine pointing North, but a different value may be specified here.

Define initial mooring position (only enabled for floating turbines): For floating turbines, the initial position of the platform can be defined. The position entered in this screen is the initial displacement of the Modal reference node defined in the tower screen in global coordinates relative to its defined position. If the Define initial mooring position box is unchecked the position of the floating wind turbine will be iterated so that steady state aerodynamic, hydrostatic, gravitational and current loads are in equilibrium.

7.17 Hardware test simulation

This calculation is only available to users with a licence for the Hardware Test Module. It allows real-time simulations to be carried out, using the real turbine controller hardware to control the simulations with the **Bladed** model acting as a virtual turbine. This requires the *Hardware Test* module to be installed from the **Bladed** CD. This acts as an interface between the **Bladed** simulation and the controller. Any external controller defined in the **Bladed** project is ignored.

The Hardware Test screen in the Calculation parameters (see [7.5](#)) is used as follows. First some parameters are required to define the way in which the simulation will proceed:

Real time step length: Real time simulations use a fixed step integrator, unlike normal **Bladed** simulations. Enter the fixed integrator step length which will be used (the default is 0.005s). If the model contains fast dynamics, such as very high frequency bending modes, or some detailed pitch actuator models, a shorter step may be needed in order to achieve sufficiently accurate results. If the step length is too small however, the simulation may fail to run in real time. Every time this happens, a warning message is generated. Run a normal simulation with the Software Performance calculation output (see [7.21.9](#)) enabled to get a good indication of a suitable step length to use.

- **Real time factor:** The simulation can be made to run faster or slower than real time by entering a factor here. For example, a value of 2 will make the simulation run a factor of 2 slower than real time. A negative value will cause the simulation to run as fast as possible (but no faster than real time) without generating warning messages when samples are missed.
- **Circular buffer:** If this is selected, the oldest results are overwritten rather than being written out to the disc. The results buffer is only written to disc at the end of the simulation, so only the results

from the last part of the simulation (as defined by **Length of buffer** as defined in Simulation Control (see 7.14)) will be saved.

- **Maximise process priority:** This option can be used to increase the priority of the simulation with respect to other Windows processes which may be competing for processing resources, and will reduce the risk of the simulation running too slowly to keep up with real time.

The simulation to be carried out is defined in a **Test Plan**. Enter the name of the test plan file, or browse for it. Test plans are created by the Hardware Test Interface: a button is provided to start up the interface, which comes with its own documentation. Within the test plan, a specific procedure defines the particular simulation, and the appropriate **Procedure name** must therefore be entered. Finally, select the **initial state** of the turbine at the start of the simulation, which may be idling, parked or in the power production state.

If desired, tick the box to show the Hardware Test interface during the simulation. You can specify whether it is shown minimised or not, and whether or not it remains visible when the simulation is finished.

7.18 Aerodynamic models

Use the **Aerodynamics control** screen to define various parameters which define how the aerodynamic calculations are carried out.

The aerodynamics model from Bladed 4.8 is a new implementation compared to the default model in Bladed 4.7 and earlier. The default model from Bladed 4.7 and earlier is referred to as the "pre-4.8" aerodynamics model in this section.

7.18.1.1 Momentum

The following options are given:

- **Drag induction:** When enabled, the drag coefficient will be included in the induction calculations. This mainly influences the BEM solution near the root of the blade, where the drag force is significant.
- **Momentum theory:** This allows selection of the way that mass flow is calculated. In axial flow conditions there is no difference between the two methods.
 - **"Glauert"**: considered more accurate for yawed flow
 - **"Axial"**: legacy pre-4.8 sub-model.

7.18.1.2 Dynamic wake

BEM theory is fundamentally a steady flow model. The dynamic wake model introduces a lagged response in induction to changes in operating conditions. The dynamic wake model is applied independently to each annulus of the rotor. The following options are given:

- **Dynamic wake base:**
 - **"Annulus"**: dynamic wake is calculated over the entire annular ring. Induced velocities are averaged over the number of blades.
 - **"Blade element"**: The annulus is divided into segments to which separate dynamic wakes are applied.
- **Dynamic wake model:**
 - **"Øye"**: Recommended default model, including a larger time constant for the induction lag, yielding slower wake recovery.
 - **"Free flow"**: This model will switch off the induction calculations. This can be used for example when the rotor is parked or idling or in cases where the user desires to have close to zero flow speeds.
 - **"Pitt & Peters"**: legacy sub-model. Equivalent to "Dynamic inflow" in the pre-4.8 model.
 - **"Equilibrium"**: legacy sub-model. Calculates the equilibrium inflow velocities at every instant in time.

- **"Frozen"**: legacy sub-model. Uses the equilibrium induction velocities computed during initial conditions for the entire simulation
- **Dynamic tangential induction**: For the Øye dynamic wake model, when disabled, the tangential induction will be solved by iteration on each time step, as per the Pitt & Peters model.

7.18.1.3 Engineering correction models

A traditional BEM code assumes that the turbine operates in a steady and uniform flow field with no yaw misalignment. In reality, wind turbines do operate in yaw, hence a **Skew wake correction model** is included to better predict the aerodynamic loads when operating at high yaw misalignments. Furthermore, it is assumed that the induction in each annulus is invariant with azimuth angle. This assumption does not hold near the tip and root of the blade for a rotor with a finite number of blades, therefore the **Tip/Hub loss** models are used.

- **Skew wake correction model**:
 - **"Glauert"**: Recommended model for all conditions. It is not recommended to disable the Glauert model in yaw misalignments in excess of 10 degrees. For idling or parked rotors this model has no influence.
 - **"Disabled"**: legacy pre-4.8 option.
- **Tip speed ratio thresholds for inflow**: Specify tip speed ratio thresholds beyond which the inflow calculations will be switched off.
- **Tip/Hub loss**: Enable or disable Prandtl corrections for tip and hub loss.
- **Simplified aerofoil orientation**: When selected, it is assumed that the aerofoil remains perpendicular to the blade root Z-axis when the blade is bending. This is similar to what pre-4.8 aerodynamics does. By default it is recommended that this option is disabled such that the aerofoil orientation follows the blade deflection.

7.18.1.4 Dynamic stall

BEM codes rely on measured or computed lookup tables for steady-state lift, drag and moment coefficients as a function of angle of attack. These steady curves are generally not sufficient to model the dynamic response of a flexible rotor to variable inflow conditions. Therefore **dynamic stall** models are implemented in the **Bladed** code. The dynamic stall models are invalid for parked and idling where the blade experiences high angles of attack, and should be switched off. Note that the **dynamic stall model** refers here to the modelling of both the unsteady effects for attached flow as well as the dynamics of trailing-edge separation.

- **Dynamic stall model**:
 - **"Øye"**: The most simple dynamic stall model offered, which only contains terms for modelling the dynamics of trailing edge separation.
 - **"Beddoes-Leishman"**: The incompressible model is considered most appropriate for typical turbines with tip speeds of Mach number smaller than 0.25. The compressible model is suitable for high Mach numbers but is unstable for low Mach numbers.
 - **"Pre-4.8 Beddoes-Leishman"**: This model is equivalent to the Beddoes-Leishman model as implemented in the aerodynamics module used up to Bladed 4.8. The Theory Manual describes in detail in which areas this model differs from the incompressible and compressible Beddoes-Leishman models. This model typically provides the highest aerodynamic damping in parked/idling simulations where the blade experiences large angles of attack.
 - **"No dynamic stall"**: Steady lift, drag and moment coefficient curves will be used.
- **Use Kirchoff equation for normal force coefficient**: This option is only used for the Incompressible and Compressible Beddoes-Leishman models. If selected, the normal force coefficient is computed using the dynamic separation position in Kirchoff's equation directly. If unselected, the dynamic separation position is used to linearly interpolate between fully separated and fully attached flow. The latter is the case by default. In normal operating conditions this setting will not lead to significant differences, it is however found that in parked/idling where the blade experiences high angles of attack this option will improve the aerodynamic damping of the blade. The explanation for the damping differences is given in [\[8\]](#).

- **Dynamic pitching moment coefficient:** This option is enabled by default. It is not recommended to disable this option for blades with a torsional degree of freedom because the so-called “pitch- rate damping” term of the moment coefficient is typically important to keep the blade torsional mode stable.
- **Include impulsive lift and moment contributions:** Both Beddoes-Leishman models contain terms that model the non-circulatory contributions in lift and moment coefficient. Note that some moment contribution terms are not controlled by this option; but are controlled by the “**Dynamic pitching moment coefficient**” option.
- **Starting radius for dynamic stall:** This is the radial position, given as a percentage, outboard of which the dynamic stall model will be used. This allows the use of aerofoil data on the inboard blade sections that has been corrected for the effects of three-dimensional flow, but which is incompatible with the dynamic stall model.
- **Ending radius for dynamic stall:** This is the radial position, given as a percentage, outboard of which the dynamic stall model will be switched off.
- **Time constants:** The default values come from the legacy model and are considered valid for a wide range of situations.
 - **Separation time constant:** This is a dimensionless time constant, given in terms of the time taken to travel half a chord. It defines the lag in the movement of the separation point due to unsteady pressure and boundary layer response. Unlike the pre-4.8 aerodynamics this value should be positive. The value defaults to 3.
 - **Pressure lag time constant:** This time constant describes the lag between the pressure and the lift. The default value is 1.7. This time constant is not needed on the Øye dynamic stall model.
 - **Vortex lift time constant:** This time constant describes the rate of decay of the vortex induced lift that is generated when airfoils undergo a rapid change in angle of attack towards positive or negative stall. As a default value 6.0 is used.

7.18.1.5 Computational cost of different options

There is no significant difference in computational cost anticipated when switching between different options, with the following exceptions:

- **Dynamic stall model:**
 - “**Øye**”: Typically, this model is quicker but captures less of the flow physics compared to the Beddoes-Leishman models.
 - “**Beddoes-Leishman**”: This is the recommended model but will run slower than the Øye model.
 - “**Compressible**”: With this model, enabling impulsive lift and moment contributions can significantly reduce the integrator step, thereby slowing the simulation.
 - “**Incompressible**”: “: With this model, enabling impulsive lift and moment contributions the simulation will run slower but this effect is less significant compared to the compressible model
 - “**No dynamic stall**”: The fastest of the models.
- **Ending radius for dynamic stall:** Near the tip many blades have very small chord lengths. In combination with the high effective inflow speed, this can lead to small time constants in the dynamic stall models requiring small integrator time steps. For example, the non-dimensional time constant for pressure lift defaults to 1.7. In dimensional form this time constant would be:

$$T = T_p \frac{c}{2W_{xy}}$$

Where:

- c is local chord length
- T_p the non-dimensional pressure lift time constant
- W_{xy} the effective wind speed which can be approximated by Ωr

If the chord length is e.g. 20cm and the tip speed 75 m/s then this would result in a time constant of 2.3ms or 441Hz causing the integrator to take steps of less than 2 ms. A rule of thumb is to avoid time constants less than 10ms in the simulations.

7.18.1.6 Aerodynamic settings compared to “pre-4.8” inputs

This section explains the meaning of the aerodynamic inputs, how they relate to the model used in 4.7 model and earlier – referred to as the “pre-4.8” aerodynamics model.

It is possible to set up the aerodynamics model to use the same engineering sub-models as the pre-4.8 model. However, the new implementation of equivalent models often differs from the pre-4.8 code, as new terms based on recent literature have been introduced. The most significant change is that the latest code is completely written in state-space form, which changes the way the dynamic wake and dynamic stall models are integrated, allowing a more realistic simulation of certain dynamics. Therefore differences are expected in turbine response and load levels between results from the pre-4.8 and the latest models, given the same configuration.

The latest model contains some new sub-models and additional options on the existing sub-models. The default settings in the aerodynamic model are considered appropriate for typical turbine environmental and operational conditions. It is possible to enable/disable many of the sub-models, which may be useful in evaluating their performance.

The following table summarises the default settings in the latest and pre-4.8 models. Some further information about these default settings is given later in this section.

Submodel	Default setting in aerodynamics	Default setting in pre-4.8 aerodynamics
Drag induction	On	On*
Momentum theory	Glauert momentum theory	Axial momentum theory*
Dynamic wake base	Dynamic wake on blade element	Dynamic wake on blade element*
Dynamic wake model	Øye dynamic wake	Pitt & Peters dynamic wake
Dynamic tangential induction	On	N/A
Skew wake correction model	Glauert skew wake model	No skew wake model
No inflow below TSR	1	0
Full inflow above TSR	2	0
Tip loss correction	On	On
Hub loss correction	Off	Off
Simplified aerofoil orientation	Off	On*
Dynamic stall model	Incompressible Beddoes-Leishman	Pre - 4.8 Beddoes-Leishman
Dynamic pitching moment coefficient	On	On*
Include impulsive lift and moment contributions	Off	On*
Starting radius for dynamic stall	25	0
Ending radius for dynamic stall	95	100*
Separation time constant	3.0	3.0
Pressure lag constant	1.7	1.7*
Vortex lift constant	6.0	6.0*

* Asterisk indicates options that were hard-coded in the pre-4.8 model.

7.19 Imbalances

Use the **Imbalances** window (see [7.5](#)) to define any mass imbalance of the rotor, or errors in the set angle or the pitch angle of one or more blades. Enter the following data:

- **Imbalance mass:** the mass of the imbalance. **Note:** From version 4.0 onwards, this is an *additional* mass.
- **Radius of imbalance:** the radial position of the centre of gravity of the imbalance mass.
- **Azimuthal position of imbalance:** zero means the imbalance mass centre of gravity is on blade 1.
- **Error in blade set angle:** for each blade, specify the error in set angle. This will rotate both the pitching and non-pitching sections of the blade - see [4.1](#).
- **Error in pitch angle:** for each blade, specify the error in pitch (or aileron/airbrake) angle. This will rotate only the pitching sections of the blade - see [3.2](#).
- **Error in blade azimuth:** for each blade, specify the error in azimuthal position for each blade.

7.20 Turbine Faults

Use the **Turbine Faults** window to specify pitch fault cases, generator or network fault cases, yaw fault cases and transducer fault cases.

From the **Bladed** toolbar, use **Specify... Faults**. An external controller fault can be specified in the Other faults tab. This allows the user to define an integer number which is passed through in swap array number 161 at the specified time. It is up to the user defined external controller dll to interpret this number and take the appropriate action.

7.20.1 Pitch faults

Enter the following data to specify pitch faults (applies also to ailerons or airbrakes if defined):

- **Pitch failure mode:** for each blade, specify the type of failure for the pitch mechanism.
 - **None:** The pitch mechanism is operating normally.
 - **Permanently Stuck** at the **Pitch of failed blade**.
 - **Constant rate runaway.**
 - **Free** or **Constrained** by a constant torque and/or the pitch bearing friction.
 - **Seizure** at a given time or when passing through a given angle after a given time.
- **Recoverable:** specify whether the pitch failure is recoverable. This means that the safety system pitch action will override the pitch fault.
- **Pitch of failed blade:** for each blade where the pitch mechanism is **Permanently Stuck**, specify the angle at which the pitch is stuck
- **Time for pitch failure:** enter the time at which the failure will occur.
- **Constant rate demand** for constant rate runaway failures.
- **Constant torque demand** for **Free/Constrained** failures.
- **Fail when passing through:** angle at which seizure will take place after the **Time for pitch failure** has passed.

7.20.2 Generator and network faults

Four types of generator or network faults can be modelled:

- **Grid loss**
Specify the time from the start of the simulation for the grid loss to occur.
- **Generator short circuit**
Enter the time from the start of the simulation for the short circuit fault.
Different short circuit models are available for synchronous generators as long as a transient model is selected: select the type of fault required. In other cases, a file containing a time history of generator torque during the fault can be provided. Enter the path to the short circuit file. The file should be in ASCII format, and the first line should be an integer specifying the number of points in the time history. The remainder of the file can contain either 2 or 3 columns. Column 1 is the time in seconds from the start of the fault. If the file contains only 2 columns, the second column is the generator torque (Nm). If the file contains 3 columns the second column is the maximum generator torque and the third column is the minimum generator torque (both in Nm). The power production control is allowed to continue to specify a demanded generator torque between these two values.

- **Network voltage disturbance**

If electrical dynamics have been specified in the generator module, a network voltage disturbance can also be specified. Enter the time for the start of the voltage disturbance, and select either **External Input File** or a user defined **Look up table**. The external input file should be an ASCII file containing an integer specifying the number of points on the first line, and then two columns on the subsequent lines giving time in seconds since the start of the disturbance, and the corresponding voltage disturbance factor. The voltage disturbance factor is expressed as a fraction of the nominal network voltage.

- **Network frequency disturbance**

If electrical dynamics have been specified in the generator module, a network frequency disturbance can also be specified. Enter the time for the start of the frequency disturbance, and select either **External Input File** or a user defined **Look up table**. The external input file should be an ASCII file containing an integer specifying the number of points on the first line, and then two columns on the subsequent lines giving time in seconds since the start of the disturbance, and the corresponding frequency disturbance factor. The frequency disturbance factor is expressed as a fraction of the nominal network frequency.

7.20.3 Yaw faults

If a yaw system is defined, three types of yaw fault can be specified:

- **Constant rate**

Enter the constant rate for a yaw runaway, and the position of any end stop if required. The end stop is defined clockwise from north.

- **Constant torque**

Enter the constant yaw failure torque

- **Free/Constrained**

Enter any linear stiffness, linear damping and yaw friction which are active during the yaw failure.

7.20.4 Transducer faults

Transducer faults can be modelled for both the power transducer and the generator speed transducer. Enter the time for failure and the constant value which the transducer reports after it has failed. Note that a transducer failure can be used to cause an overspeed.

If an external controller failure is specified, at the given time channel 161 of the swap array will take the value specified. This can be used to activate a fault within the external controller.

7.21 Coordinate systems and outputs

A large amount of output is potentially available from some simulations. Click the **Outputs** button on the **Calculations** screen to define which of these outputs are required. The outputs are grouped into **blade**, **tower** and **other** outputs.

The Aerodynamic information, Performance coefficients and Power curve calculations are unaffected, as they produce a pre-defined set of outputs.

7.21.1 Blade coordinate systems and outputs

To configure blade outputs, click **Blade outputs** on the Calculation Outputs Specification screen.

Blade outputs include **aerodynamic information** (including distributed aerodynamic loadings), **blade loads** and **motions**. For each of these categories, the information may be generated at any or all of the blade stations (see 3.1). Click **Select Output Stations** to determine which information is required at which stations. Click **Add** to define additional points where interpolated loads will be output.

Having defined the blade stations required, specify for each type of output whether that information is required on the **First blade**, **All blades**, or not at all (**None**).

The loads can be specified independently about the following axes: Principal axes, Root Axes, Aerodynamic Axes and User Axes. Within each group, the loads are labeled M_x , M_y , M_{xy} , M_z , F_x , F_y , F_{xy} , F_z .

It's important to note that the **Principal Axes** and **Root Axes** stations coincide with the underlying finite beam element model nodes. The Principal Axes coordinate vectors align with the finite beam elements. The Root Axes coordinates have the same origin rotated to align with the blade root.

The **Aerodynamic** and **User Axes** potentially have origins that do not coincide with the underlying finite element beam model nodes. These load outputs are found by transforming the finite element loads (in the Principal Axes coordinates) to the Aerodynamic or User Axes coordinate centre. The transformation accounts for additional M_x and M_y bending moments that are generated at the Aerodynamic or User Axis centre by the element axial force F_z in the Principal Axes system, due to the offset in the aerofoil plane between these two coordinate centres. This effectively estimates the load that would have occurred if the load was carried through an axis at the Aerodynamic Axis or User Axis centre. There is some approximation in this method as the Aerodynamic and User Axes loads are not the true load path. Any changes to the the blade dynamics (e.g. deflections or loads) that might result from such a change in load path are not accounted for.

- Principal axes:** The positive z-axis follows the local deflected neutral axis at each blade station towards the blade tip. The positive y axis is defined by the *principal axis orientation*. The positive x axis is orthogonal to the y and z and follows the right hand rule. For output loads, the origin of the axes is on the neutral axis at each local deflected blade station. (see diagram below)

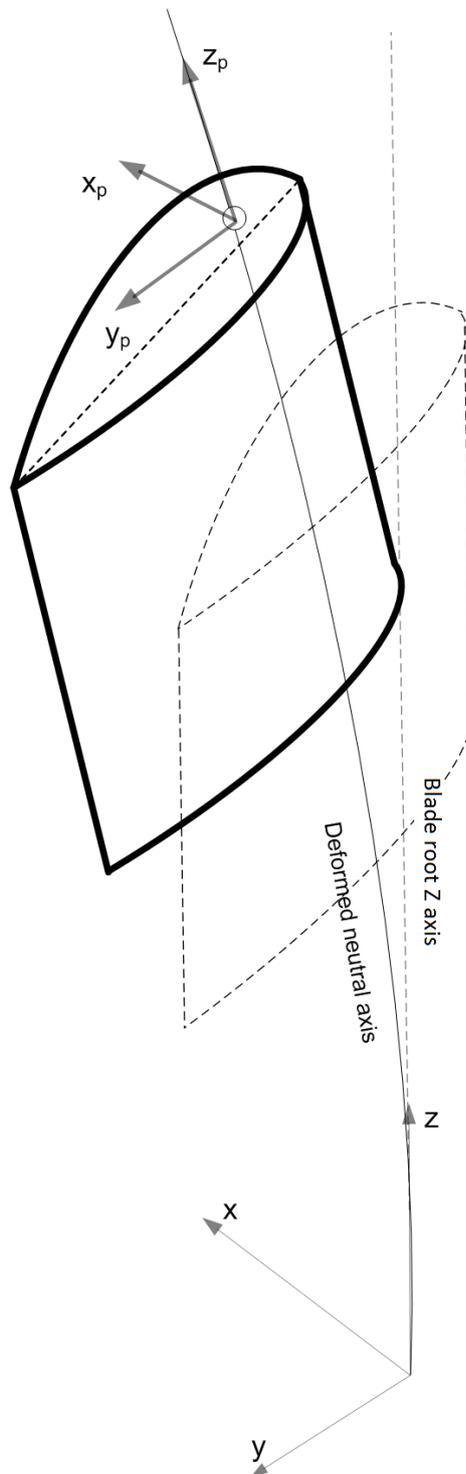


Figure 7-3: Blade principal axes coordinate system

Note that there is a subtle difference between the "principal axis" frame and the "blade local element frame".

The "blade local element frame" is orientated so that the X vector in this coordinate system points directly between adjacent nodes on the blade. The other two coordinate system vector directions are determined by the "principal axis twist" angle. This is explained in more detail in DNV GL technical note UKBR-110052-T-31-A

The "principal axis" frame is used for load output in Bladed. The principal axis orientation is calculated by taking the average orientation of the two blade elements at the node where the elements join. This is illustrated below. The two adjoining local element frames are shown in green and red. The principal axes output frame is shown in blue.

Note that the element local coordinate system has its x direction along the element, unlike the "principal axes" coordinate system which has z along the element axis.

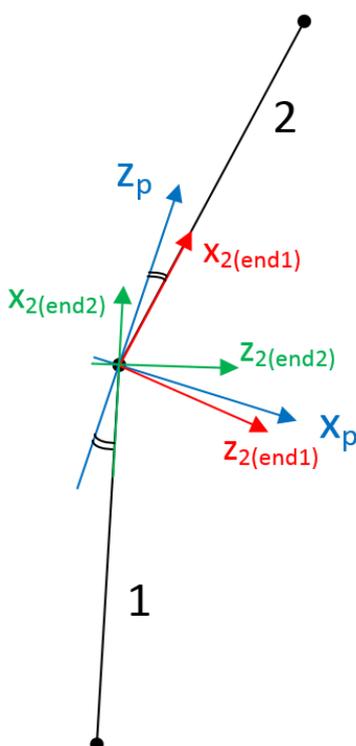


Figure 7-4: Relationship between blade "local element axes" and "principal axes coordinates"

- **Root axes:** The orientation of the axes is fixed to the blade root and does not rotate with either twist or blade deflection. The axis set does rotate about the z axis with pitch. For output loads, the origin of the axes is on the neutral axis at each local deflected blade station.

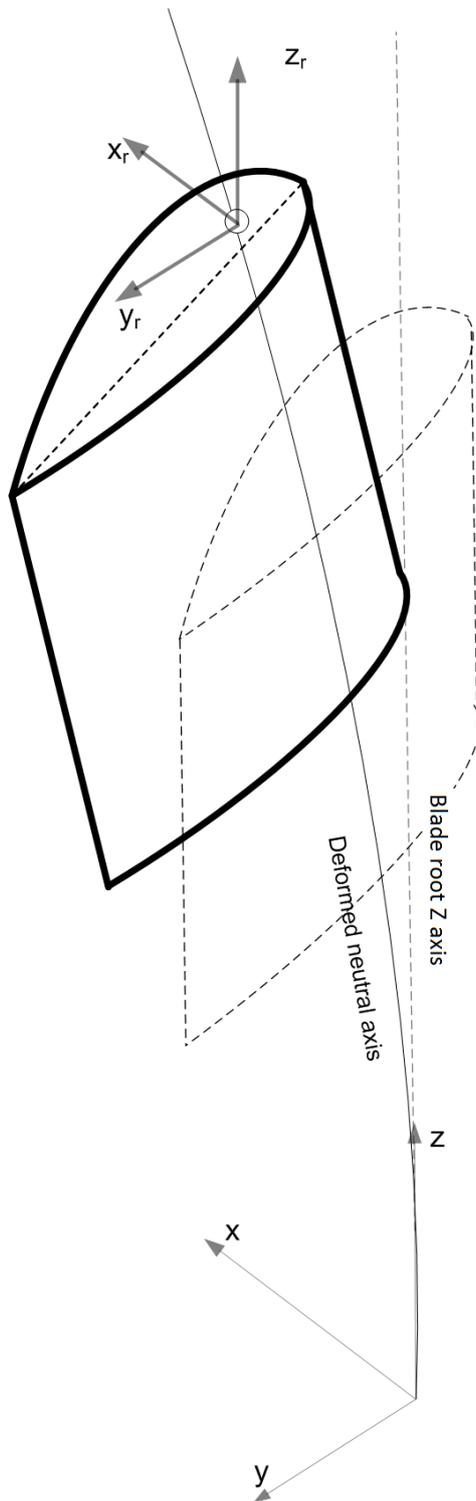


Figure 7-5: Blade root axes coordinate system

- Aerodynamic axes:** The x axis is perpendicular to the local aerodynamic chord line and the positive y axis is aligned along the local aerodynamic chord line from leading edge to trailing edge. The z-axis is parallel to the local deflected neutral axis at each blade station and increases towards the blade tip. For output loads, the origin of the axes is on the chord line at 25% chord from the leading edge at each local deflected blade station.

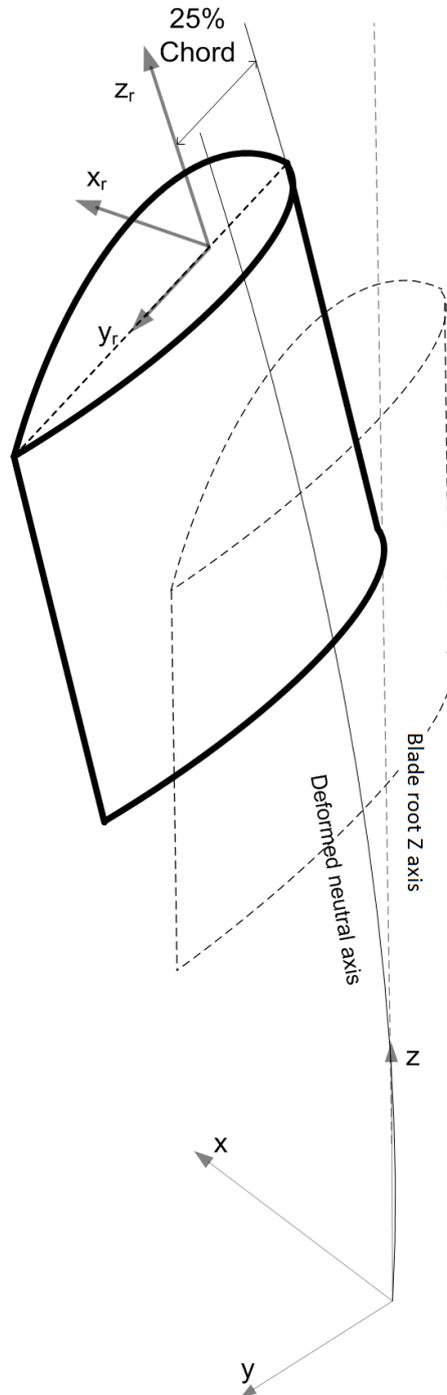


Figure 7-6: Blade Aerodynamic axes coordinate system

- **User axes:** The origin of the axes is specified as percentages of chord, parallel and perpendicular to the chord at each blade station. The user can specify whether the z-axis is parallel to the root axis or the local neutral axis, and independently whether the y-axis is aligned to the principal axis orientation, the aerodynamic twist or the root axis.

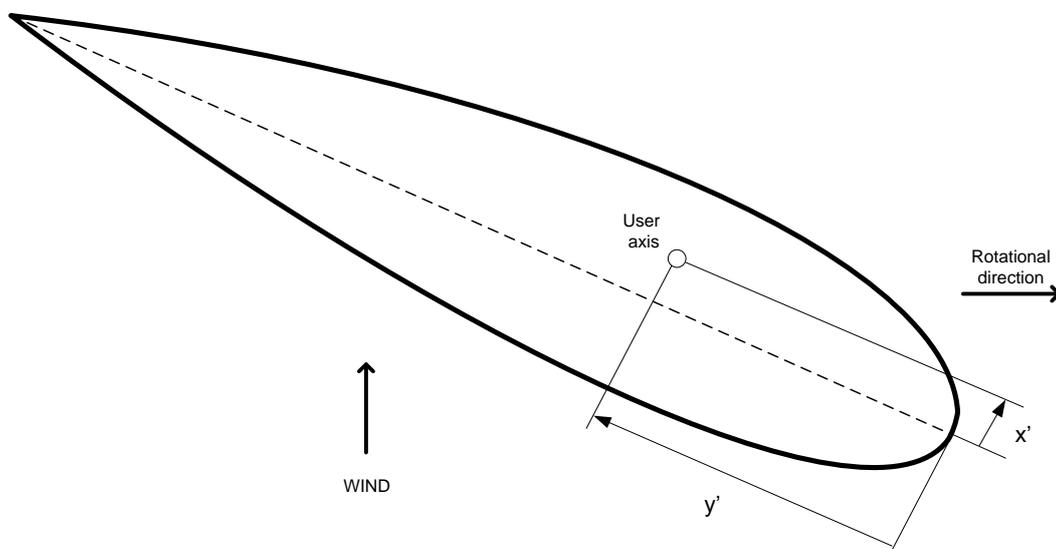


Figure 7-7: Position of the user defined output axis

Note that an additional set of the blade root loads, using a coordinate system fixed to the blade root in the hub (i.e. inboard of the pitch bearing) are output as part of the Rotating Hub output group. This coordinate system is shown in [Figure 7-9](#).

The table below compares the blade outputs between **Bladed** versions 3.x and 4.x

		Origin	y-axis	z-axis
(v3.85)	Blade loads Mx/My	At deformed pitch axis	Fixed to blade root and doesn't rotate with pitch or twist (=always tangential to rotor plane coned)	Parallel to rotor plane (coned)
	Blade loads Flap/edge	At deformed pitch axis	Parallel to chord (twist angle)	Parallel to rotor plane (coned)
(v4.x)	Principal axes	At deformed neutral axis	Determined by principal axis orientation	Parallel to deformed neutral axis. Affected by blade mounting cone / sweep angles
	Root axes	At deformed neutral axis	Parallel to blade root Y axis. (doesn't rotate with twist or deflection, i.e. tangential to rotor plane coned when both pitch angle and blade set angle are 0 and no blade mounting cone or sweep are present. Rotates about z axis with pitch)	Parallel to blade root Z axis. Parallel to rotor plane (coned) when no blade mounting cone or mounting sweep present. Affected by blade mounting cone / sweep angles
	Aero axes	At 25% chord (deformed)	Parallel to chord (aerodynamic twist)	Parallel to deformed neutral axis. Affected by blade mounting cone / sweep angles
	User axes	Anywhere in blade section, user defined	Determined by principal axis orientation, OR	Parallel to deformed neutral axis Affected by blade mounting cone / sweep angles OR
			Fixed to blade root as in root axes (rotates with pitch) , OR	Parallel to rotor plane (coned) when no pre-cone or sweep present. Affected by blade mounting cone / sweep angles
Blade root from rotating hub	At blade root centre (on pitch axis)	Fixed to blade root and doesn't rotate with pitch (=always tangential to rotor plane coned)	Parallel to chord (aerodynamic twist)	Parallel to rotor plane (coned). NOT affected by blade mounting cone / sweep angles

Table 7-2: Blade loads output coordinate systems

The coordinate system for the blade deflections is as follows:

- z-axis Radially along the blade root Z axis
- x-axis Perpendicular to z, and pointing towards the tower for an upwind turbine, or away from the tower for a downwind turbine.
- y-axis Perpendicular to blade axis and shaft axis, to give a right-handed co-ordinate system independent of direction of rotation and rotor location upwind or downwind of the tower.

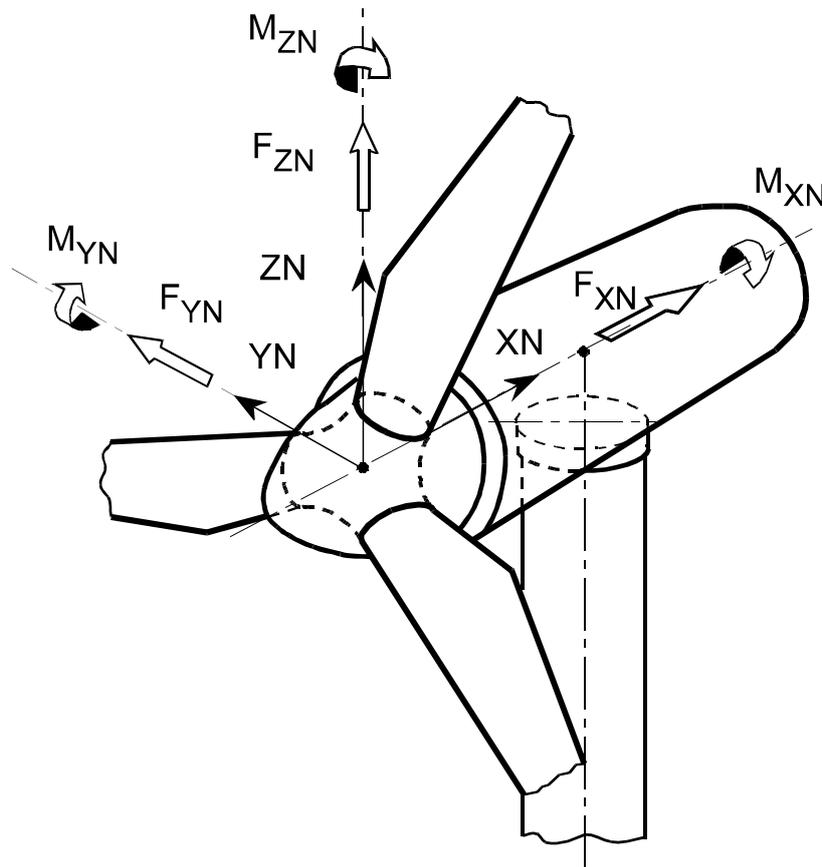
Load configuration file:

If the blade manufacturer has provided a load configuration file, this can be used to produce custom output for that particular blade.

7.21.2 Hub coordinate system

To configure hub outputs, click **Other Outputs** on the Calculation Outputs Specification screen.

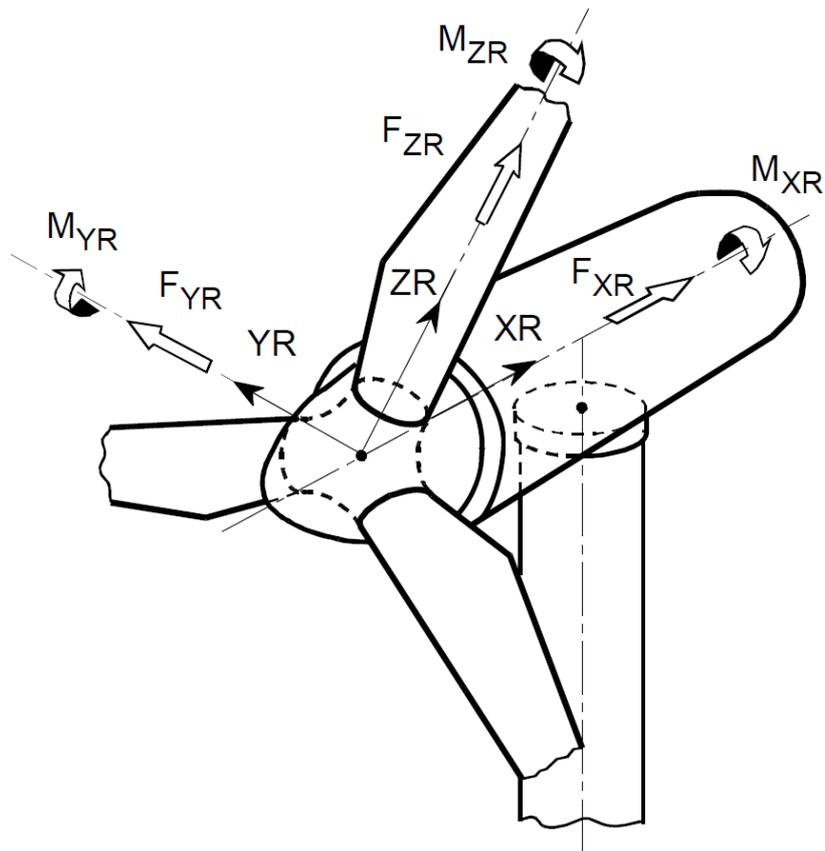
The co-ordinate system for the hub load and deflection outputs from the calculations is based on the 'GL' convention, with some modifications as specified below.



Hub loads in fixed frame of reference:

- XN Along shaft axis, and pointing towards the tower for an upwind turbine, or away from the tower for a downwind turbine (the picture shows an upwind turbine).
- ZN Perpendicular to XN, such that ZN would be vertically upwards if the tilt angle were zero.
- YN Horizontal, to give a right-handed co-ordinate system independent of direction of rotation and rotor location upwind or downwind of the tower.

Figure 7-8: Co-ordinate system for stationary hub loads



Hub loads in rotating frame of reference:

- XN Along shaft axis, and pointing towards the tower for an upwind turbine, or away from the tower for a downwind turbine (the picture shows an upwind turbine).
 - ZN Perpendicular to XN, such that ZN would be aligned with blade 1 axis if the cone angle were zero.
 - YN Perpendicular to XN and ZN, to give a right-handed co-ordinate system independent of direction of rotation and rotor location upwind or downwind of the tower.
- Origin At hub centre (intersection of blade and shaft axes).

Figure 7-9: Co-ordinate system for rotating hub loads

7.21.3 Anti-clockwise and downstream rotor coordinate systems

The rotor and blade output coordinate systems for all permutations of clockwise, anti-clockwise, downstream and upstream rotors are shown in the following four figures.

Note that for simplicity these diagrams do not include blade mounting cone, blade mounting sweep. The blade pitch and set angle are shown at zero in these diagrams.

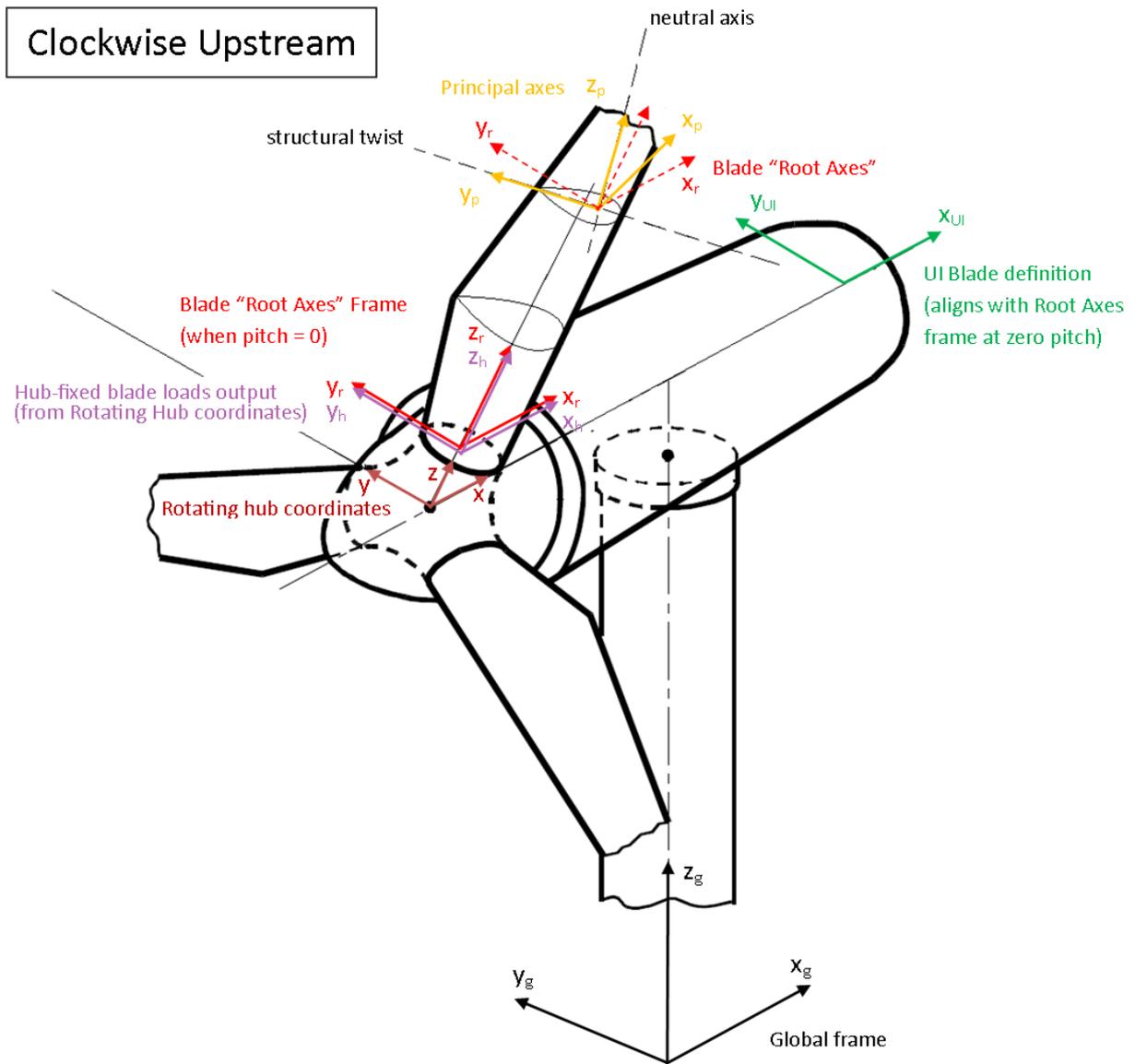


Figure 7-10: Blade and rotor coordinate systems for a clockwise upstream rotor

Anti-Clockwise Upstream

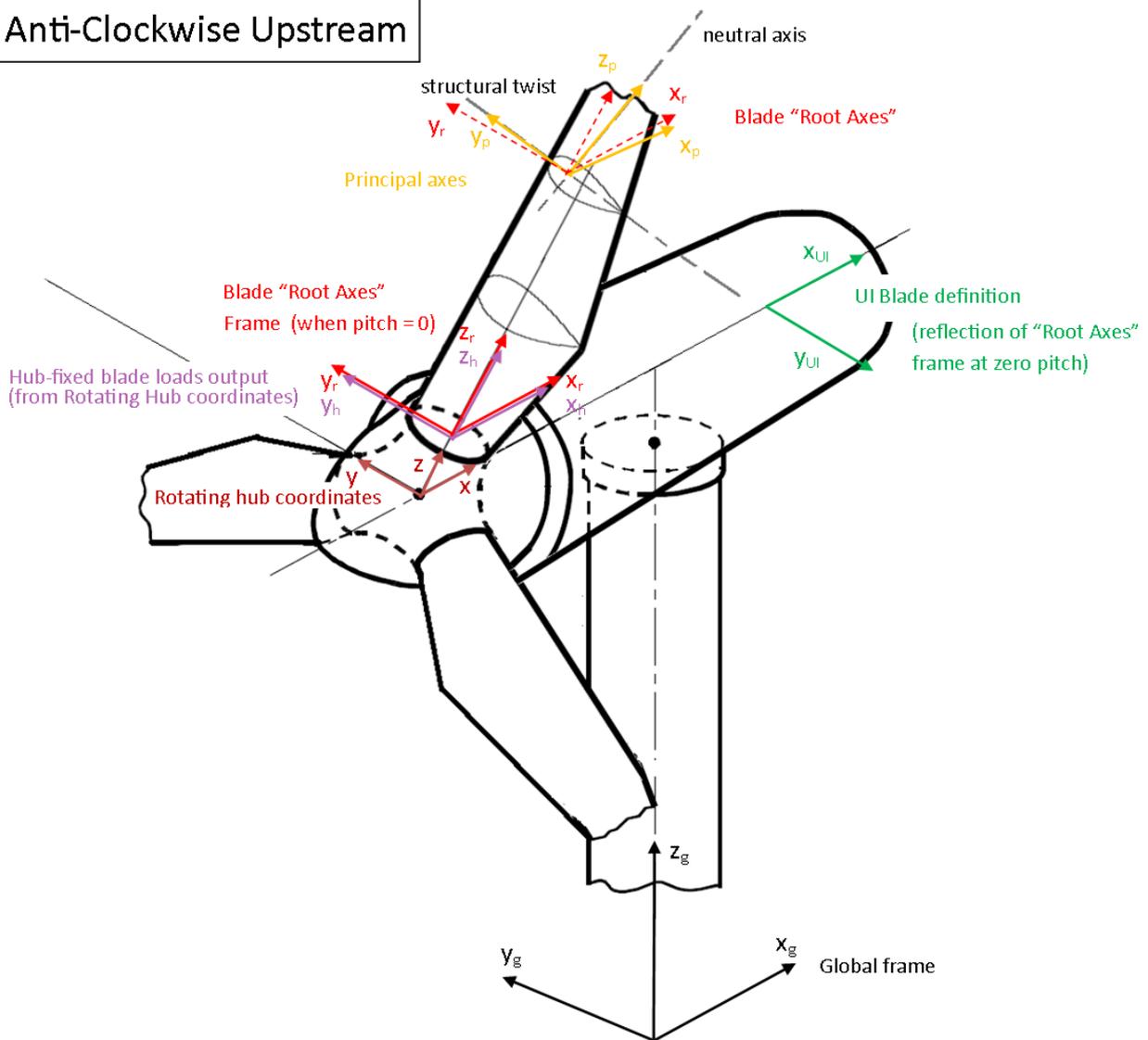


Figure 7-11: Blade and rotor coordinate systems for an anticlockwise upstream rotor

Clockwise Downstream

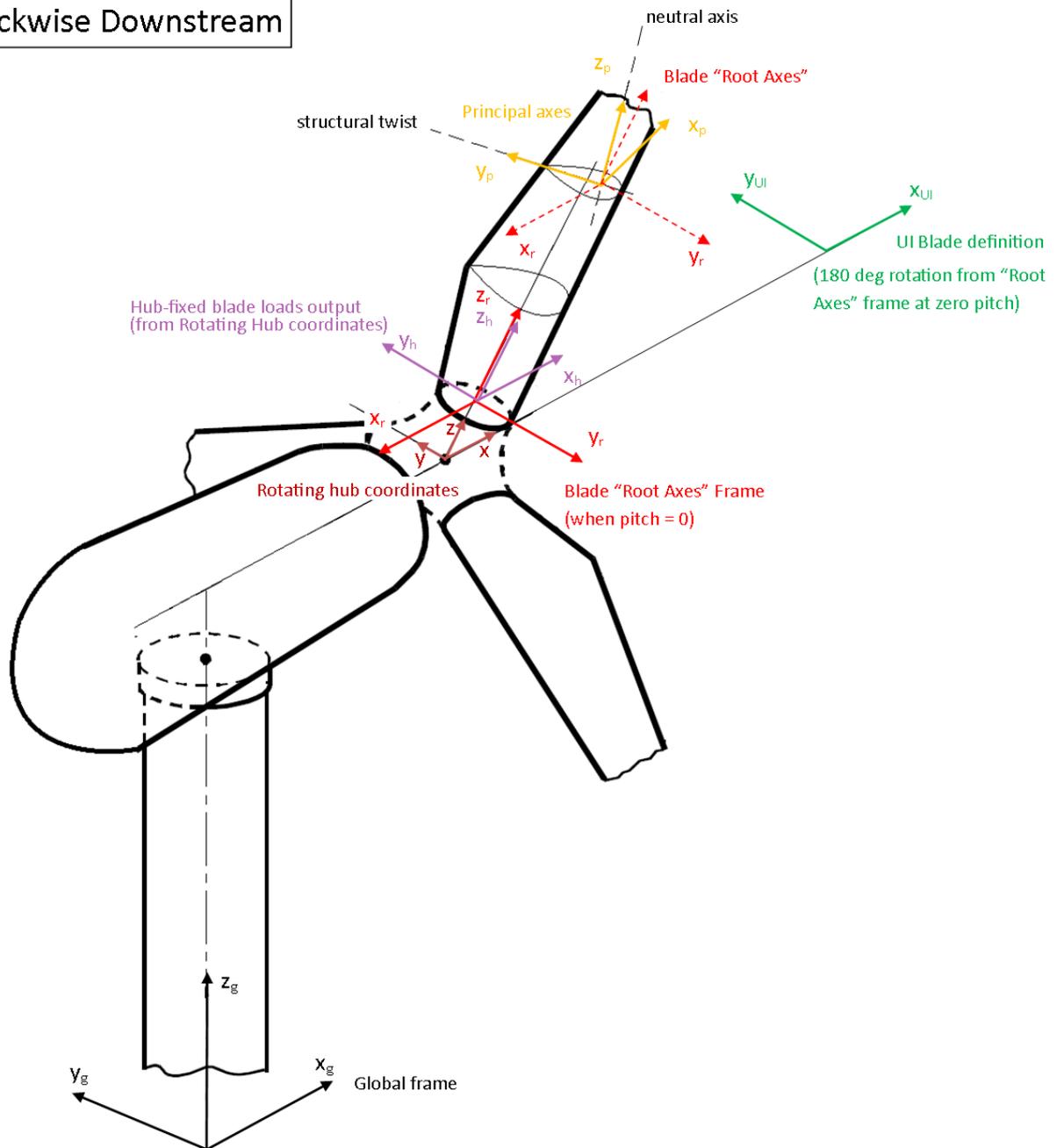


Figure 7-12: Blade and rotor coordinate systems for a clockwise downstream rotor

Anti-Clockwise Downstream

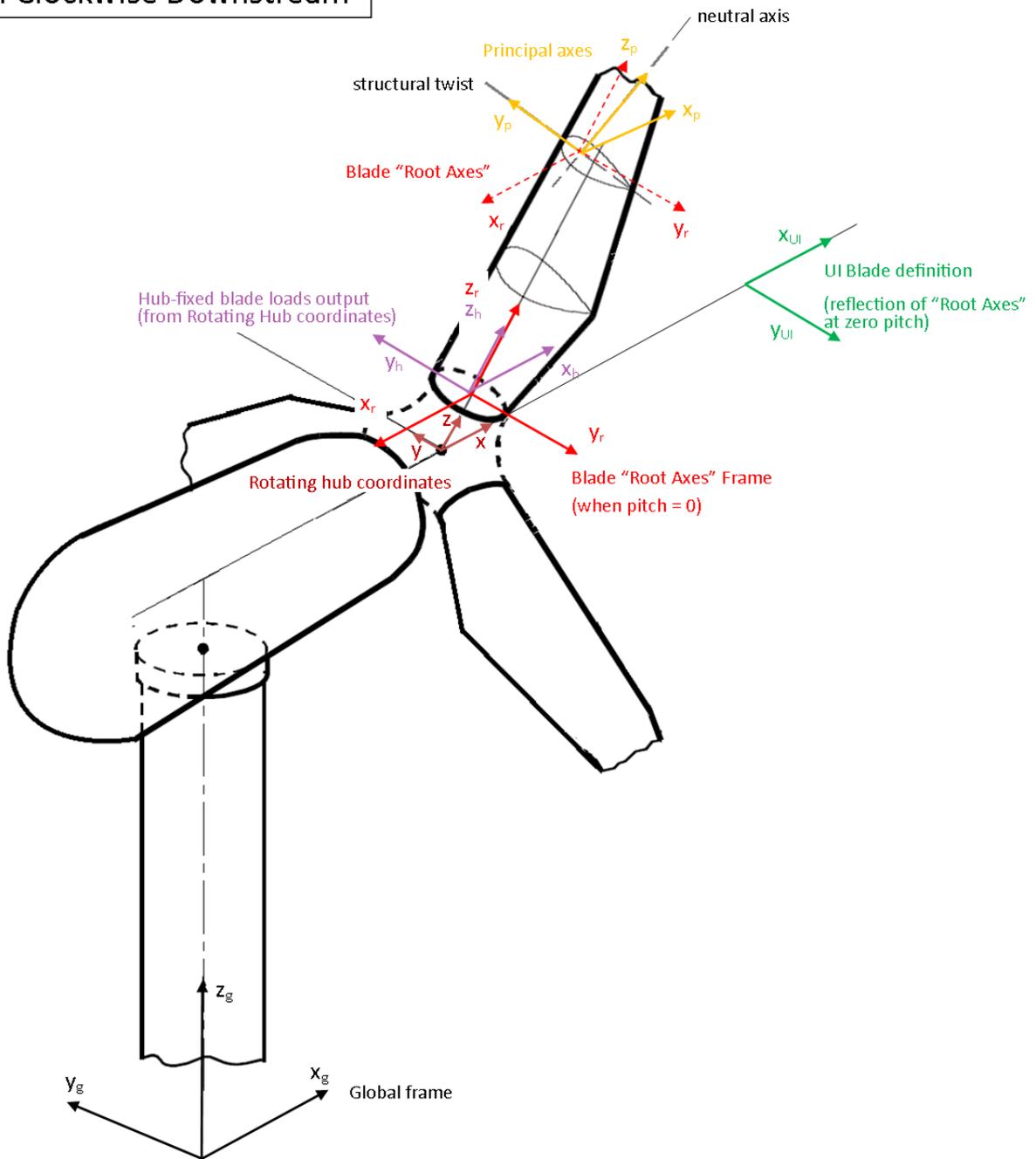


Figure 7-13: Blade and rotor coordinate systems for an anticlockwise downstream rotor

7.21.4 Tower outputs

Click **Tower outputs** on the Calculation Outputs Specification screen.

Tower loads may be generated at any or all of the tower stations (see [4.5](#)). Click **Select Output Stations** to determine at which stations the loads are to be output. For the monopile tower model, click **Add** to define additional points where interpolated loads will be output.

Having defined the tower stations required, specify for each individual output whether that information is required. For the loads, this can be specified independently for each of the forces and moments.

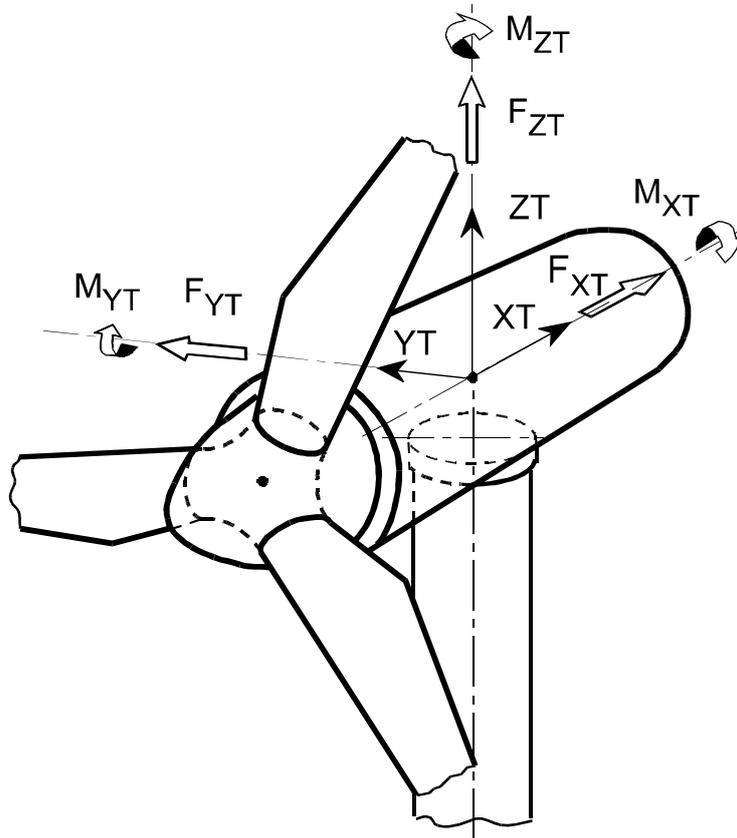
There is also an option to **Refine deflections**. With this option disabled, the tower deflection outputs are modal deflections. Modal deflections give a good estimate of overall tower motion, but may not be very precise in predicting the small deflections at foundation stations. This can lead to a poor estimate of the foundation reaction loads given the applied loads on the rest of the turbine.

With the **Refine deflections** feature enabled, the tower deflections are re-calculated at each output time step using the underlying finite element model with the external loads applied. The refined deflections give a more accurate estimate of the deflections at the foundation nodes, hence the foundation reaction forces are more accurate, especially for non-linear foundations. With **Refine deflections** selected, Bladed will iterate at each output time step to find the foundation applied loads based on the refined deflections. This will ensure that the applied loads, tower deflections and foundation reactions correspond correctly.

Note that the modal deflections are used to estimate the foundation loads when solving the structural system at each integrator time step. To use the refined deflections at every time step would require iteration on each time step, causing a significant increase in simulation time. The assumption with this modelling choice is that error in foundation load estimate due to using modal deflections doesn't affect the overall turbine dynamics significantly. It is therefore reasonable to only calculate the foundation reaction loads based on refined deflection on each output time step.

7.21.5 Tower co-ordinate systems

The co-ordinate system for tower outputs is based on the GL convention, as described below.



XT Pointing South.
ZT Pointing along deflected tower centre line.
YT Pointing East.

Origin At each tower station.

Note that for steady-state calculations the wind is deemed to come from the North.

Figure 7-14: Co-ordinate system for tower loads (monopile only) and deflections

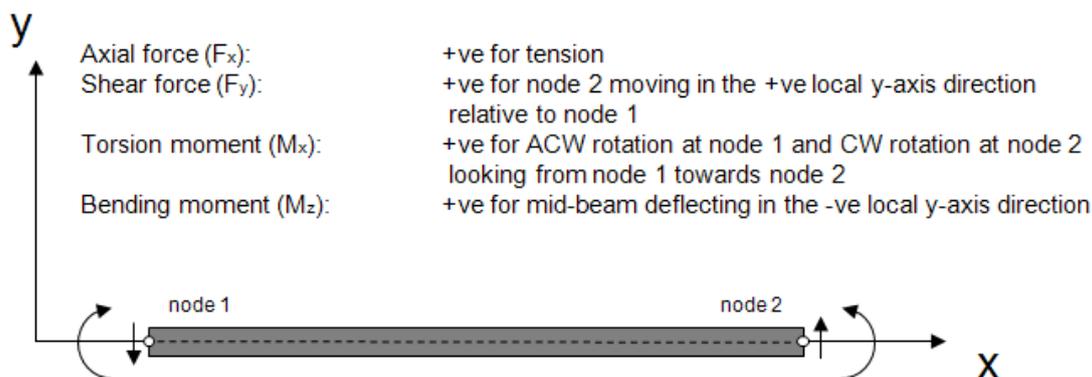
Note that the coordinate system system for tower loads moves with tower modal deflections, so that the coordinate system remains aligned with the deflected member axis.

Loads and deflections for multi-member towers

The loads are output with reference to the local member coordinate system for each member. The member x-axis is always aligned along the member. The member z axis is perpendicular to the member x-axis and aligned according to the direction cosines for the member z-axis as specified in the tower screen. These are the direction cosines of the z-axis relative to the global GL coordinate system. For example, a vertical member with z-axis direction cosine of 0.0 in the x-direction, 1.0 in the y-direction and 0.0 in the z-direction would be a member where the local x-axis corresponds to the GL z-axis, the local y-axis corresponds to the GL x-axis and the local z-axis corresponds to the GL y-axis.

The deflections are output in the GL coordinate system.

Bladed multi-member output convention: local x-y plane:



Bladed multi-member output convention: local x-z plane:

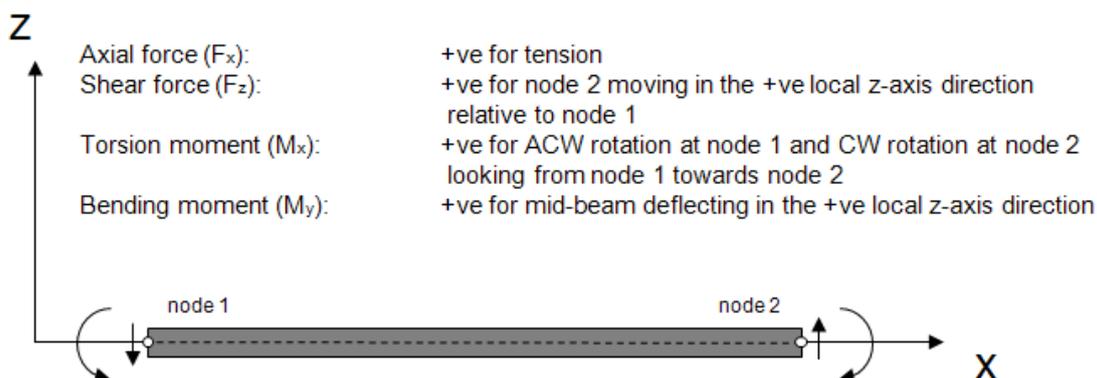


Figure 7-15: Multi-member tower coordinate output convention

7.21.6 Pitch system direction convention

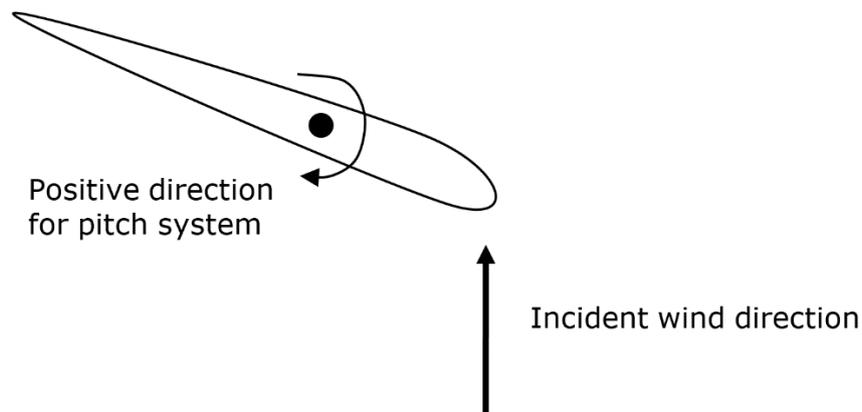


Figure 7-16: Pitch system sign convention

The same sign convention applies to the applied pitching moment, pitch bearing friction and pitch actuator torque. A positive applied pitching moment means that the blade aerodynamic, gravitational and other applied forces act in such a way as to drive the pitch in the positive (feathering) direction.

Pitch system loads and deflections:

For pitch angles, rates and accelerations, the positive direction is the direction in which the leading edge moves upwind, i.e. towards the feathered position:

7.21.7 Yaw bearing output

The yaw bearing is located at the “nacelle node” as specified in the support structure screen. For monopile towers, the yaw bearing is assumed to be located at the top tower station.

The co-ordinate system for yaw bearing loads is the same as for the top tower station except that it rotates with the nacelle yaw angle.

Yaw bearing output is defined opposite to the GL z-axis, with clockwise from North being positive.

7.21.8 Variables that follow “rotor direction is positive” convention

There are a number of other variables for which the positive direction is the ‘normal running’ direction. This applies to controller and drive train variables such as rotor and generator speeds, torques and azimuthal positions, drive train and generator torques etc.

7.21.9 Other general outputs

- **Summary information:** includes principal operational and environmental indicators.
- **Software performance:** useful for diagnosing slow simulations, and finding a suitable step length for Real Time Test simulations.
- **Specific node outputs:** This is an advanced feature, intended for users with some knowledge of the Multibody Dynamics approach used within Bladed. See the description below.

Clicking the **Specific node outputs...** button will open the Node Outputs screen, which allows you to add, delete or edit entries in the node outputs list. This is a list of structural nodes at which you want to see kinematic (position, velocity, acceleration) or loads outputs.

To see a tree diagram of node identifiers and the components they are connected to, run Bladed once without altering the project details (a 1-second simulation is enough), and then open the verification (.\$VE) file in a text editor. Any node and component identifiers you enter must be exactly as they appear in this tree, and the component must be connected directly to the node. For each node, choose the type – Loads or kinematics – and the coordinate system in which you want the outputs to be expressed. Specify a name for the output group; this is the name that will appear in the list of outputs in the dataviewer. You can also choose to have the outputs calculated at a position (Offset) other than that of the node itself. This option should be used with care, as it must represent a position in space that is physically on the turbine structure, otherwise the outputs will be meaningless.

7.21.10 Environmental information

These outputs describe the wind speed and direction, as well as the sea surface elevation and seismic motions, if present. All speeds are relative to the hub or rotor motion, so any motion of the nacelle is reflected in these outputs.

Hub wind speed magnitude: This is the overall relative flow speed magnitude experienced by the hub.

Cup anemometer wind speed: This is the horizontal component of the hub wind speed magnitude.

Hub longitudinal wind speed: The component of the horizontal wind speed aligned with the hub longitudinal direction (includes rotation due to yaw angle).

Hub lateral wind speed: The component of the horizontal wind speed aligned with the hub lateral direction (includes rotation due to yaw angle).

Hub vertical wind speed: The component of the flow speed magnitude in the vertical direction.

Wind direction at hub: Direction, as an angle to North (clockwise looking down).

Wind upflow at hub: Upflow, as an angle to the vector of flow direction in the horizontal plane. Positive angle is upwards.

Rotor average longitudinal wind speed: Component of the overall flow speed along the hub X axis, averaged over the rotor plane. The rotor is effectively treated as a non-rotating disc.

Rotor average longitudinal wind direction: Direction, as an angle to North (clockwise looking down), averaged over the rotor plane as above.

Sea surface elevation: Instantaneous sea surface Z in global coordinates, at global X=0, Y=0.

Ground positions, velocities, accelerations: Seismic motion of the ground at the turbine base, in global coordinates.

7.21.11 Aerodynamic outputs

In this section the outputs of the aerodynamics module are explained. To understand the outputs, the definitions of the rotor axial/tangential direction are defined in the figures below. Note that “axial” indicates in direction of normal vector “n” and tangential indicates in the direction of tangent vector “t”. Further note that the radial vector “r” assumes an un-coned rotor.

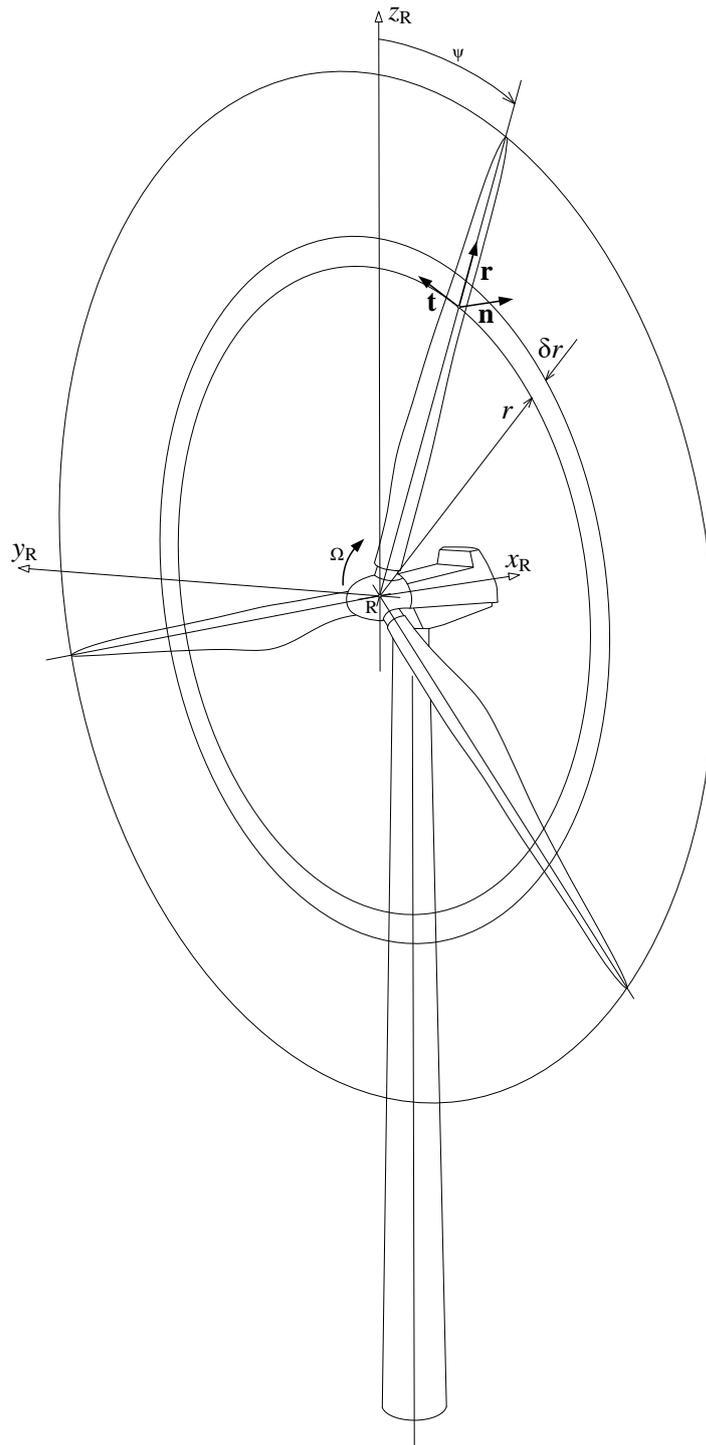


Figure 7-17: Geometry of rotor including rotor and blade coordinate system

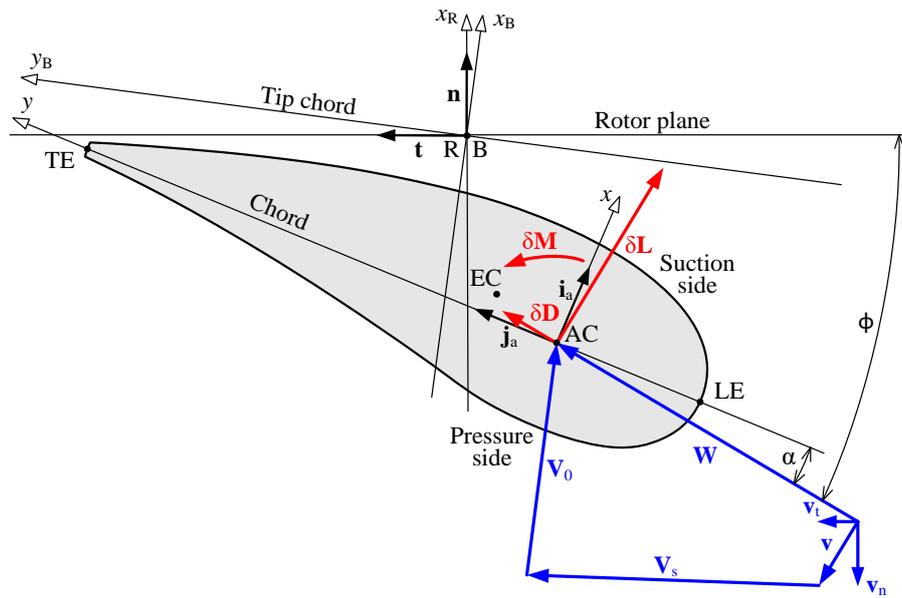


Figure 7-18: Aerofoil section with velocities and aerodynamic loads. The blade is pointing out of the paper and moves towards the right, while the flow is coming from below

Variable name	Explanation	Equivalent variable name in pre-4.8 model
Axial induction factor	Velocity induced by rotor in axial direction normalized with the negative free stream flow speed	Axial inflow factor
Tangential induction factor	Velocity induced by rotor in tangential direction normalized with the negative rotational speed times local radius	Tangential inflow factor
Axial induction velocity	Velocity induced by rotor in axial direction	n.a.
Tangential induction velocity	Velocity induced by rotor in tangential direction	n.a.
Tip/hub loss factor	Magnitude of Prandtl tip/hub loss factor	Tip/hub loss factor
Inflow angle	Angle between effective flow speed velocity vector and tangential direction of rotor plane (φ in the figure)	Inflow angle
Angle of attack	Angle between effective flow speed vector and local chord line (α in the figure) expressed the quarter chord point	Angle of attack
Angle of attack rate of change	Time derivative of angle of attack expressed at the quarter chord point	n.a.
Relative axial flow speed	Axial component of relative flow speed vector.	Incident out of plane velocity
Relative tangential flow speed	Tangential component of relative flow speed vector	Incident in-plane velocity
Axial aerodynamic loading	Axial component of aerodynamic load per unit length.	Out of plane aerodynamic loading
Tangential aerodynamic loading	Tangential component of aerodynamic load per unit length.	In plane aerodynamic loading
Aerodynamic twist loading	Aerodynamic moment about the elastic axis per unit length. Orientation of moment axis is in direction of unit vector "r" in Figure XX	Aerodynamic twist loading
Chordwise Aerodynamic loading	Aerodynamic force in direction of the chordline	n.a.
Normal Aerodynamic loading	Aerodynamic force normal to the chord line	n.a.
Local longitudinal/lateral/vertical flow speed	Components of the flow speed at each blade station in global co-ordinates including tower shadow excluding induction from rotor.	n.a.
Minimum Pressure Coefficient	Minimum pressure coefficient on the suction surface of the hydrofoil at the given angle of attack	n.a.

Table 7-3: Aerodynamic outputs in Bladed

7.21.12 Pitch Actuator outputs

- **Pitch Angle** – The current pitch angle of the blade
- **Pitch rate** – The current rate of pitching of the blade
- **Pitch acceleration** – The current acceleration of pitching of the blade
- **Demand Pitch Angle/Rate** – The input demand from the controller into the actuator system
- **External Pitch Bearing Moment** – This is the load that comes from the blade due to aerodynamic and inertial loading, acting on the bearing. When the rotor is clockwise, the sense of the load is opposite to that of GL Blade z-axis, but is the same sense for an anti-clockwise rotor.
- **Pitch Bearing Friction** – The total friction (kinetic or static) acting on the pitch freedom.
- **Pitching Inertia** – The total pitching inertia consisting of the outboard blade inertia plus the referred motor rotational inertia if a rotary actuator is defined.
- **Pitch actuator torque** (*rotary actuator drive or no drive defined*) –The torque provided by the pitch motor. Defined on the motor side of the gear ratio.
- **Pitch actuator force** (*linear actuator drive*) –The force provided along the line of the actuator ram.
- **Controller demanded pitch angle/rate** (*setpoint trajectory planning defined*) –The input demand from the controller before being limited by setpoint trajectory planning.
- **Limited demanded pitch angle/rate** (*setpoint trajectory planning defined*) –The demand after it has been limited by setpoint trajectory planning.
- **Motor position/rate/acceleration** (*When a flexibility in the actuator drive is defined*) – Kinematics the motor side of the gear ratio and drive flexibility

7.22 Specifying calculation options

A number of machine and wind features, once defined, are available as options which can be switched on or off in some of the steady state calculations and simulations. Click the **Show Options** button in the bottom left corner of the **Calculations** screen to display the options.

Initially, all options are switched on by default, but if any are switched off for a given calculation, the setting will be remembered. An option displaying a yellow or red light will not be used even if selected – see [7.2.3](#). The options available are:

- **Energy losses** (see [4.11](#)).
- **Drive train flexibility** (see [4.8.2](#)).
- **Drive train mounting flexibility** (see [4.9](#)).
- **Teeter restraint** (see [4.4](#)).
- **Teeter motion**: see hub (see [4.2](#)).
- **Passive yaw motion** (see [5.18](#)).
- **Prescribed yaw manoeuvre** (see [5.18.2](#)).
- **Imbalances** (see [7.19](#)) and **Faults** (see [7.20](#)).
- **Nacelle and tower windage loads**.
- **Tower shadow** (see [6.3](#)).
- **Wind shear** (see [6.2](#)).
- **Upturbine wake** (see [6.4](#)).

7.23 Multiple calculation setup

Users can define sets of calculations at one time, varying one or more parameters for each calculation. Select the type of calculation on the **Calculations** screen, and then from the main toolbar select **Calculation ... Multiple calculation**.

Multiple calculation setup is available for the following calculation types:

- Generation of turbulent wind files, for a range of mean wind speeds (using MultiSetup)
- All simulations, for various parameter ranges (using MultiSetup)
- Post processing of variables with multiple locations, such as blade or tower stations (using pre-4.7 Multiple calculation setup)

For the MultiSetup feature released with Bladed 4.7, please see the MultiSetup user manual.

7.23.1 PostProcessing Multiple Calculation Setup

For each parameters, a list of values can be defined, provided the parameter is appropriate for the currently selected calculation, and the module from which the parameter is taken is defined.

At the top of the screen, the list of parameters is displayed along with a status label. The status label can be one of the following:

Not available: The parameter is not appropriate for the currently selected calculation, or the module that contains that parameter has not been defined.

Not active: The parameter is appropriate for the calculation and its module has been defined, but the user has not chosen to vary it as part of the set of multiple calculations. Clicking in the checkbox will activate this parameter.

X Selected: This parameter has X different values that will be used in combination with other selected parameters in the set of multiple calculations. Clicking on the **Select** option will allow the list of values to be edited.

Active but not available: This parameter has a set of values defined by the user, but it is not appropriate for the currently selected calculation or its module is not defined.

Viewing calculation details

Once the lists of parameters have been defined, the set of calculations can be viewed by clicking on the **View calculations details** option. This will bring up a list summarising the set of calculations and presenting the parameters that vary between each one. This screen can also be used to specify the format of the path and runname to be used for the calculations. The parameter value, or a list of letters or numbers, can be used as part of the path or runname. The initial letter or number is displayed in the list box next to the parameter name, and can be changed by selecting it. This letter or number can be used in the path as a folder, or as part of the runname. The first calculation defined will use the initial letter or number displayed, and for subsequent calculations this letter or number will be incremented. It is also possible to choose which calculations to include in the set, and to change some parameters in this screen.

Once the user is happy with the set of calculations, they can be added to the current batch list by clicking on the **Add calculations to Batch** option.

8 POST-PROCESSING

Bladed provides an integral post-processing facility for analysis of calculation results. The following calculations are provided:

- Basic statistics (see [8.1](#)) to calculate the mean, minimum, maximum, standard deviation, skewness and kurtosis of a signal.
- Fourier harmonics (see [8.2](#)) to calculate the fourier components of a signal at multiples of rotor rotational frequency.
- Periodic components (see [8.3](#)) to separate out the periodic and stochastic parts of a signal.
- Extreme predictions (see [8.4](#)) to estimate lifetime extreme loads from a sample time history.
- Auto spectrum (see [8.5](#)) to calculate the auto-spectral density (frequency spectrum) of a signal.
- Cross spectrum (see [8.6](#)) to calculate cross-spectral density, coherence and transfer functions between any two signals.
- Probability density (see [8.7](#)) to calculate the probability distribution of a signal.
- Peak analysis (see [8.8](#)) to calculate the probability density of signal peaks and troughs.
- Level crossing analysis (see [8.9](#)) to calculate the probability of crossing any particular threshold.
- Rainflow cycle count (see [8.10](#)) for cycle counting of a signal for fatigue analysis, and calculation of damage equivalent loads.
- Fatigue analysis (see [8.11](#)) to calculate fatigue damage.
- Annual energy yield (see [8.13](#)) to calculate the annual energy yield as a function of mean wind speed from a power curve.
- Channel combination and tabulation (see [8.14](#)) to combine and scale a number of signals (useful to produce a combined stress signal for fatigue analysis).
- Ultimate loads (see [8.16](#)) to find the maximum and minimum values of loads and concurrent values of other loads.
- Ultimate load cases (see [8.17](#)) to identify the load cases producing maximum and minimum values of specified loads.
- Flicker (see [8.18](#)) to calculate the flicker severity due to voltage variations on the network caused by the turbine.
- Linear model (see [8.19](#)) to convert the output of the Model Linearisation (see [7.11](#)) calculation into a state-space model suitable for control design, for example using Matlab [\[2\]](#).
- Offshore Code Checking (see [8.21](#)) via direct links from **Bladed** to external third party code checking capabilities.

Post-processing calculations may be run from the **Calculation** screen, by selecting the calculation required and pressing **Run now** or **Run in Batch**, or from the **Post-processing** screen (obtained by clicking the **Analyse** icon on the toolbar) by pressing the **Execute now** or **Add to Batch** buttons.

8.1 Basic statistics

This calculation generates the mean, minimum, maximum, standard deviation, skewness and kurtosis of a signal.

Click the **Select...** button (see [8.20](#)) to define the signal to be processed. If you have selected Multiple Channels (see [8.15](#)) you will be able to specify a number of load cases and variables to be processed in a single calculation, and the results for each load case will then be stored as additional outputs of that load case, and/or accumulated over the turbine lifetime.

Note that if the skewness and kurtosis are not required, the remaining statistics may be obtained directly from the **Data View** facility.

8.2 Fourier harmonics

This calculation generates the Fourier components of a signal at multiples of rotor rotational frequency.

Click the **Select...** buttons (see [8.20](#)) to define the signal to be processed and the signal representing rotor azimuth. If you have selected Multiple Channels (see [8.15](#)) you will be able to specify a number of load cases and variables to be processed in a single calculation, and the results for each load case will then be stored as additional outputs of that load case. Then choose either automatic or manual selection

of parameters. Automatic selection is usually adequate, but if entered manually, the parameters required are:

- **Number of bins:** the number of azimuthal bins to be used in the calculation (minimum 4, maximum 144). More bins will generally give a more accurate result, but do not choose more bins than the number of signal samples in one revolution of the rotor.
- **Number of harmonics:** the number of harmonics to be calculated (minimum 1, maximum one quarter of the number of bins).

8.3 Periodic components

This calculation separates out the periodic and stochastic parts of a signal.

Click the **Select...** buttons (see [8.20](#)) to define the signal to be processed and the signal representing rotor azimuth. If you have selected Multiple Channels (see [8.15](#)) you will be able to specify a number of load cases and variables to be processed in a single calculation, and the results for each load case will then be stored as additional outputs of that load case. Then choose either automatic or manual selection of data. Automatic selection is usually adequate, but if entered manually, the parameter required is:

- **Number of bins:** the number of azimuthal bins to be used in the calculation (minimum 4, maximum 144). More bins will generally give a more accurate result, but do not choose more bins than the number of signal samples in one revolution of the rotor.

8.4 Extreme predictions

This obsolete calculation was used to estimate lifetime extreme loads from a sample time history. Use Extreme load extrapolation (see [8.20](#)) instead.

Click the **Select...** buttons (see [8.20](#)) to define the signal to be processed and the signal representing rotor azimuth. If you have selected Multiple Channels (see [8.15](#)) you will be able to specify a number of load cases and variables to be processed in a single calculation, and the results for each load case will then be stored as additional outputs of that load case. Then enter the following information:

- **Time spent at specified condition:** the total amount of time during the turbine lifetime for which the conditions of the sample time history are considered representative.
 - **Number of points per segment**
(power of 2, maximum 4096)
 - **Percentage overlap**
 - **Window** (select)
 - **Remove trends** (check box)
- } See [8.5.1](#)

Then choose either automatic or manual selection of data. Automatic selection is usually adequate, but if entered manually, the parameter required is:

- **Number of bins:** the number of azimuthal bins to be used in the calculation (minimum 4, maximum 144). More bins will generally give a more accurate result, but do not choose more bins than the number of signal samples in one revolution of the rotor.

8.5 Auto spectrum

This calculates the auto-spectral density (frequency spectrum) of a signal.

Click the **Select...** button (see [8.20](#)) to define the signal to be processed. If you have selected Multiple Channels (see [8.15](#)) you will be able to specify a number of load cases and variables to be processed in a single calculation, and the results for each load case will then be stored as additional outputs of that load case, and/or accumulated over the turbine lifetime. Then enter the information required, as follows:

- **Number of points per segment**
(power of 2, maximum 4096)
 - **Percentage overlap**
 - **Window** (select)
 - **Remove trends** (check box)
- } See [8.5.1](#)

8.5.1 Options for spectral analysis

All calculations involving spectral analysis use a Fast Fourier Transform technique with ensemble averaging. To perform the spectral analysis, the signal is divided into a number of segments of equal length, each of which contains a number of points which must be a power of 2. The segments need not be distinct, but may overlap. Each segment is then shaped by multiplying by a 'window' function which tapers the segment to zero at each end. This improves the spectrum particularly at high frequencies. A choice of windowing functions is available. Optionally, each segment may have a linear trend removed before windowing, which can improve the spectral estimation at low frequencies. The final spectrum is obtained by averaging together the resulting spectra from each segment.

The information required is therefore as follows:

- **Number of points:** the number of datapoints per segment. Must be a power of 2, maximum 4096. More points will give better frequency resolution, which may be important especially at low frequencies. However, choosing fewer points may result in a smoother spectrum because there will be more segments. If in doubt, 512 is a good starting point.
- **Percentage overlap:** the overlap between the segments. Must be less than 100%. 50% is often satisfactory, although 0% may be more appropriate if a rectangular window is used.
- **Window:** this may be
 - (a) rectangular (equivalent to not using a window)
 - (b) triangular:

$$1 - |2f - 1|$$

(c) Hanning:

$$(1 - \cos(2\pi f)) / 2$$

(d) Hamming:

$$0.54 - 0.46 \cos(2\pi f)$$

(e) Welch:

$$1 - (2f - 1)^2$$

Where f is the fractional position along the segment (0 at the start, 1 at the end). One of the last three windows (which are all quite similar) is recommended.

- **Trend removal:** If checked, a linear trend is calculated for each segment and removed from it before windowing, this is usually desirable. If left unchecked, the mean is calculated instead for each segment and removed from it before windowing.

8.6 Cross spectrum

This calculates the cross-spectral density, coherence and transfer functions between any two signals.

Click the **Select...** buttons (see [8.15](#)) to define the two signals to be processed. If you have selected Multiple Channels (see [8.15](#)) you will be able to specify a number of load cases and variables to be processed in a single calculation, and the results for each load case will then be stored as additional outputs of that load case, and/or accumulated over the turbine lifetime. Then enter the information required, as follows:

- **Number of points per segment**
(power of 2, maximum 4096)
 - **Percentage overlap**
 - **Window** (select)
 - **Remove trends** (check box)
- } See [8.5.1](#)

8.7 Probability density

This calculation generates the probability distribution of a signal, and also, for comparison, the Gaussian or normal distribution with the same mean and standard deviation.

Using the **Revs at Level** option, it is also possible to bin the signal based upon a signal other than time. In this case the **Azimuth** signal will also need to be defined. If rotor azimuth is selected for this, the load level will be binned against the number of revolutions, which is useful in gearbox and bearing design.

Click the **Select...** button (see [8.20](#)) to define the signal to be processed. If you have selected Multiple Channels (see [8.15](#)) you will be able to specify a number of load cases and variables to be processed in a single calculation, and the results for each load case will then be stored as additional outputs of that load case, and/or accumulated over the turbine lifetime.

For each variable to be processed, enter the following data:

- **Minimum value:** the start of the distribution (should be less than the signal minimum) (see [8.12](#)).
- **Maximum value:** the end of the distribution (should be greater than the signal maximum) (see [8.12](#)).
- **Number of bins:** suggest 20. A higher number (maximum 144) will give finer resolution, provided there is sufficient data.
- **Remove mean** (checkbox): If this is selected, the distribution will be representative of deviations from the mean, and so the distribution mean will be zero. If Multiple Channels is selected, the means will not be removed.

8.8 Peak value analysis

This calculation generates the probability distribution of the peaks and troughs of a signal.

Click the **Select...** button (see [8.20](#)) to define the signal to be processed. button to define the signal to be processed. If you have selected Multiple Channels (see [8.15](#)) you will be able to specify a number of load cases and variables to be processed in a single calculation, and the results for each load case will then be stored as additional outputs of that load case, and/or accumulated over the turbine lifetime.

For each variable to be processed, enter the following data:

- **Minimum value:** the start of the distribution (should be less than the signal minimum) (see [8.12](#)).
- **Maximum value:** the end of the distribution (should be greater than the signal maximum) (see [8.12](#)).
- **Number of bins:** suggest 20. A higher number (maximum 144) will give finer resolution, provided there is sufficient data.

8.9 Level crossing analysis

This calculation generates the frequency with which a signal crosses given levels.

Click the **Select...** button (see [8.20](#)) to define the signal to be processed. button to define the signal to be processed. If you have selected Multiple Channels (see [8.15](#)) you will be able to specify a number of load cases and variables to be processed in a single calculation, and the results for each load case will then be stored as additional outputs of that load case, and/or accumulated over the turbine lifetime.

For each variable to be processed, enter the following data:

- **Minimum value:** the start of the distribution (should be less than the signal minimum) (see [8.12](#)).
- **Maximum value:** the end of the distribution (should be greater than the signal maximum) (see [8.12](#)).
- **Number of bins:** suggest 20. A higher number (maximum 144) will give finer resolution, provided there is sufficient data.

8.10 Rainflow cycle counting

This calculation generates the rainflow cycle count for a stress history. A suitable stress history can be generated from the relevant loads using the Channel combination (see [8.14](#)) calculation.

Click the **Select...** button (see [8.20](#)) to define the stress signal to be processed. If you have selected Multiple Channels (see [8.15](#)) you will be able to specify a number of load cases and variables to be processed in a single calculation, and the results for each load case will then be stored as additional outputs of that load case, and/or accumulated over the turbine lifetime.

For each variable to be processed, enter the following data:

- **Minimum value:** the start of the stress distribution (should be less than the minimum stress) (see [8.12](#)).
- **Maximum value:** the end of the stress distribution (should be greater than the maximum stress) (see [8.12](#)).
- **Number of bins:** suggest 20. A higher number (maximum 128) will give finer resolution, provided there is sufficient data.
- **Minimum range:** the smallest signal range to be counted as a cycle. A non-zero value may be useful to remove the effect of any spurious noise on the signal.

If you select **Calculate Equivalent Loads**, enter up to 10 inverse S-N slopes to use for the calculation of equivalent loads. These are the sinusoidal loads which would produce the same fatigue damage at the specified S-N slope if the frequency of the sinusoid is as specified.

8.11 Fatigue damage estimation

This calculation generates fatigue damage estimates from a stress history or a previously generated rainflow cycle count, by taking account of the fatigue properties of the material. A suitable stress history can be generated from the relevant loads using the Channel combination (see [8.14](#)) calculation.

Click the **Select...** button (see [8.20](#)) to **either** the stress signal to be processed, **or** a previously generated rainflow cycle count (see [8.10](#)) (not available for Multiple Channels). If you have selected Multiple Channels (see [8.15](#)) you will be able to specify a number of load cases and variables to be processed in a single calculation, and the results for each load case will then be stored as additional outputs of that load case, and/or accumulated over the turbine lifetime.

Then enter the material properties as follows:

Fatigue model

- **S-N curve**
Select either the **log-log** or the **look-up table** option. The fatigue properties are then entered as follows:
If **Log-log relationship** is selected: enter the **inverse slope** (m) and **intercept** (c) of the S-N curve, such that the stress range (S) giving N cycles to failure is given by:

$$\log(S) = \log(c) - (1/m) \log(N)$$

If **Look-up table** is selected, click the **define fatigue data look-up table button** for a pop-up window. Use **Add** or **Insert** to add points to the look-up table, and enter the stress values and the corresponding number of cycles to failure by double-clicking on the appropriate table entries. The stress entries must be monotonically decreasing, with the corresponding cycles to failure monotonically increasing.

- **Goodman correction**
Use the check box to enable the Goodman correction if required. If required, enter the ultimate strength of the material, as a stress.

If a **stress history** rather than a previously generated rainflow cycle count is to be processed, enter the following data which is required for the cycle counting:

- **Minimum value:** the start of the stress distribution (should be less than the minimum stress) (see [8.12](#)).
- **Maximum value:** the end of the stress distribution (should be greater than the maximum stress) (see [8.12](#)).
- **Number of bins:** suggest 20. A higher number (maximum 128) will give finer resolution, provided there is sufficient data.
- **Minimum range:** the smallest signal range to be counted as a cycle. A non-zero value may be useful to remove the effect of any spurious noise on the signal.

8.12 Setting bin limits

Several calculations require minimum and maximum values to be entered for the bin range. These values should encompass the entire data range.

For automatic calculation of suitable bin limits, set the minimum and maximum to be equal, or leave them both blank.

Alternatively, the actual minimum and maximum of the signal may be found by running the Basic statistics (see [8.1](#)) calculation.

8.13 Annual energy yield

This calculation generates the annual energy yield of a turbine from either a steady state power curve (see [7.8](#)), a dynamic power curve as output by a previous Annual Energy Yield calculation or a series of power production simulations, as a function of annual mean wind speed. The wind distribution (see [6.11](#)) could be defined either as a Weibull, Rayleigh, or a user defined distribution of wind speeds. The wind and wave scatter diagram option could also be used to define the distribution.

Click the **Select...** button (see [8.20](#)) to define the power curve to be used. Then enter the following data:

- **Minimum wind speed:** the cut-in wind speed for the turbine
- **Maximum wind speed:** the cut-out wind speed for the turbine
- **Turbine availability:** the average availability, assumed to be uncorrelated with wind speed.

Define the wind speed distribution (see [6.11](#)) by clicking the **Define** button. If desired, select the option to scale the wind speed distribution by a series of factors; the energy yield will then be calculated as a function of annual mean wind speed while retaining the shape of the wind speed distribution. In this case, enter

- **Starting annual mean wind speed**
 - **End annual mean wind speed**
 - **Annual mean wind speed step**
- } Energy yield will be calculated for each mean wind speed in this range

8.14 Channel Combination and Tabulation

Use this calculation to combine and scale a number of signals. For example, this can be useful for generating a stress history for a particular point in the turbine structure, by expressing it as a linear combination of the various loads acting on the component. The stress history can then be used to calculate fatigue (see [8.11](#)) damage. The scale factors used for the linear combination should be such that the resulting signal is in the same stress units as are assumed in the fatigue damage calculation.

If **Combine variables across different load cases** is selected, it is possible to combine up to six signals, which can be from different calculations – Section [8.14.5](#). Usually however, the signals to be

combined will all come from the same calculation, so this option should be left unselected. This will give three options:

Channel Combination, which allows the choice of:

- an **Old Style** of channel combination which allows channel combinations from **Bladed** versions 3.51 and older to be re-used (click **Import details...** and select whether to import the channel combination details, the list of load cases, or both), and
- a **new style** of channel combination (if neither check-box is selected) which provides much greater flexibility and ease of use.

Channel Tabulation, which allows a number of variables to be output to a single file.

Matrix Combination, which allows multiple outputs to be produced from multiple inputs through matrix multiplication.

In each case, first define the **Combined Signal File No** (from 1 to 260; up to 260 combined signal files may be generated for any load case), and enter a description for the file.

In the **Tabulation** case, there is a choice to keep the output files with the results of each original calculation, or to place them in a new location. If **New directory** is chosen, select a directory which will be used to replace any part of the file path which is common to all the runs which are to be processed.

Then click **Channels and Load Cases** to define the list of variables to be processed, and the list of runs for which the processing will be repeated. Each run should contain all the variables to be processed.

8.14.1 Multiple processing option – Channel combination

Select **Variables**, and click **Add Variable** to define the variables to be processed. Click **Add Calculation** to set up an equation for processing the variables. Any number of equations may be defined. Each equation will generate a new intermediate or output variable, which will be denoted #1, #2 etc. Each equation may process any of the input variables (which will be denoted \$1, \$2 etc.) together with any of the intermediate variables #1, #2 etc. Any of the intermediate variables which are given an output name will be output to the results file.

To edit an equation, highlight any part of it and use the editing options in the panel on the right. These can be used to enter specific values, such as input (\$) or intermediate (#) variables or constants, to apply binary arithmetic or logical operators, or to apply unary operators. These include the conditional operator IF, and user-defined nonlinear functions, which are entered as look-up tables – the calculation will use linear interpolation between table values, and for values beyond the end of the table the nearest point in the table is used.

To create a calculation that refers to its own previous value simply use the # value from the current equation. You do not need to use the prev() function here as the created variable already refers to its previous value when used in its own equation: So if you use "#1 + 1" in the equation of #1, this will add +1 to its previous value, which is initialised to zero at the start.

For example, in order to find out if and how many times a variable has been greater than a specified value for an uninterrupted period, you can do something of this sort:

In this example, we are calculating how many times the pitch angle exceeds 5 deg (0.0873rad) for an uninterrupted 5 sec (100 time steps).

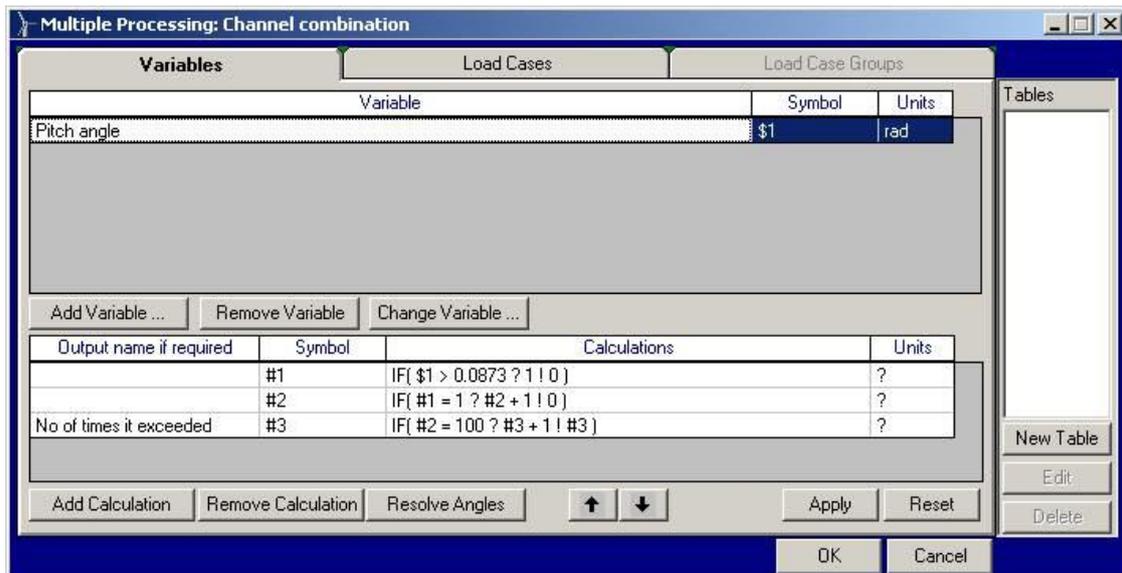


Figure 8-1: Multiple processing: Channel combination

Use **Resolve Angles** to generate a whole set of equations representing the resultant of two orthogonal loads resolved into different angles. Select the 0° and 90° loads from the list on the right of the screen – for example, the hub My and Mz loads could be selected. Then define a range of angles to resolve over, and provide a name for the output variables if they are to be output. If the 0° and 90° loads are V₀ and V₉₀, then for each angle θ an equation is created which generates V₀ cosθ + V₉₀ sinθ. The output name for each variable (if specified) will automatically have the appropriate angle appended.

Finally click **Load Cases** to set up a list of load cases for which the resulting set of channel combinations will be repeated.

8.14.2 Multiple processing option - Tabulation

Select **Variables** to define the list of variables to be tabulated. For each variable, select the desired units to be used for the tabulation.

Finally click **Load Cases** to set up a list of load cases for which the resulting set of channel combinations will be repeated.

When the calculation is run, the output may be chosen to be in ASCII or binary format. Selecting ASCII format will tabulate the selected variables as columns in a tab-delimited text file, with column headings describing each variable.

8.14.3 Multiple processing option – Matrix combination

The matrix combination calculation is available for example to set up an influence matrix, by defining a set of output loads as a linear combination of a set of input loads.

Select **Variables**, and click **Add Variable** to define the variables to be processed. Click **Add Matrix** to set up a matrix for processing the variables. Click **Add Output** to specify the outputs from the matrix combination. Any number of matrices may be defined. The size of the matrix is determined by the number of input variables and the number of outputs.

To edit a matrix, either type directly into the matrix grid or copy and paste into this grid.

Finally click **Load Cases** to set up a list of load cases for which the resulting set of matrix combinations will be repeated.

Once a matrix combination has been set up, the **Vector Combination** option becomes available. This allows outputs from a matrix combination to be processed further as a vector. Click **Define details** to set up the vector combination.

The vector combination is set up in a similar way to the channel combination. Each equation may process the vector of outputs from each matrix (denoted M1, M2 etc) to produce a further vector of outputs (denoted V1, V2 etc). Click **Add Variable** to defined additional variables which may also be used in the equation if required.

8.14.4 Multiple processing option – Old Style channel combination

Select **Variables** and click **New** to define a new output signal, and enter a description for it. Select one of the available units for the signal if appropriate.

For each output variable, click **Add Variable...** to select the required input variables. Specify any factor, offset and other unary operators to be applied to each input signal before it is combined with the other input signals. The Factor is applied first, then the Offset, then any Unary operators are applied. If several unary operators are specified, they must be separated by | (a vertical bar). They will be applied in order, starting from the right-most. Allowed operators are:

SIN	sine (of a value in radians)
COS	cosine (of a value in radians)
ABS	absolute value
SQRT	square root
INV	reciprocal
+n	Add a number n
-n	Subtract n
xn	Multiply by n
/n	Divide by n
^n	Raise to the power n

where n is a real number, which may be negative. For example, **SIN|x-3.2|^-0.2|ABS|+0.1** would calculate **sin(-3.2(|x+0.1|^{-0.2}))** from the input signal **x**.

Then select how the resulting input signals are to be combined. This may be by

- addition,
- squaring and adding, then taking the square root of the result, or
- multiplying.

Finally click **Load Cases** to set up a list of load cases for which the resulting set of channel combinations will be repeated.

8.14.5 Single channel combinations

From **Channel combination and Tabulation**, select **Combine variables across different load cases** to generate a new variable not attached to any existing load case. You will then be asked for a directory and run name for the output when you start the run. In this case, click the numbered channel selection buttons (see 8.20) to define up to six signals to be combined. The signals may be from different runs or load cases if desired. For each signal, enter a scale factor and an offset. Enter a description for the combined signal, and choose units if appropriate.

The scale factors (a_i) are applied before the offsets (b_i), i.e. the result of the calculation can be expressed as:

$$y = \sum (a_i x_i + b_i)$$

if **Simple Addition** is selected. If **Square-Add-Square Root** is selected, the result is

$$y = \sqrt{\sum (a_i x_i + b_i)^2}$$

Enter a description for the combined signal, and select one of the available units for the combined signal if appropriate.

8.15 Multiple Processing

A number of post-processing calculations allow a whole list of variables to be processed across a whole list of load cases. If appropriate, the results are accumulated over the turbine lifetime (or any other desired period) assuming a particular wind speed distribution, and a frequency of occurrence for transient events. Otherwise the results are stored as additional outputs for each load case.

In the window for the required post-processing calculation, click **Multiple Channels** to bring up the **Multiple Processing** window. Use the **Add Load Case** and **Add Variable** buttons to build up the required lists of load cases and variables. **Add Many Load Cases** allows all the calculations in a selected directory (including subdirectories if desired) to be added in one go. In most cases, multiple variables can be selected in one go by dragging over a number of variables, or holding down the **Control** key and clicking on the required variables.

It is important to ensure that the output (see [7.21](#)) specifications for all the load cases are the same, i.e. the same outputs and the same blade and tower stations have been requested.

Load cases tab: If accumulating results over the lifetime, the **Load Cases** window is loaded with all the relevant load cases, and a weighting is applied to each load case to reflect its contribution to the lifetime. For transient (T) load cases such as stops and starts, enter the Occurrences per Year of this load case. For stationary (S) cases, i.e. load cases representing an ongoing situation or state of the turbine, there is a choice three weighting methods:

- "One run per bin": Specify the wind speed bin represented by this load case, by giving the bottom and top of the bin. Eventually the whole wind speed range, starting from 0 m/s, should be represented. The wind regime current (see [6.11](#)) also needs to be defined so that the appropriate weightings can be calculated.
- "Hours per year": In this case, the number of hours per year for which each load case is representative is specified.
- "Pre-defined bins": In this case, the bins are defined independently, as is the wind regime (see [6.11](#)), and the details of each load case are automatically interrogated to allocate it to one of the bins. If a bin contains more than one run, the time in the bin is split equally between each simulation in the bin. For offshore turbines, a scatter diagram (see [6.15](#)) may be used if desired to define bins representing the joint probability distribution for wind speed, wave height and wave period.

At the bottom, enter the turbine lifetime. Note that it is possible to treat a stationary load case as transient, or vice versa, by clicking on the '**S**' or '**T**' indicator. At the bottom, enter the turbine lifetime, and ensure a wind speed distribution (see [6.11](#)) is defined (not required if all the load cases are transient). The length of transient simulations will not be included in the turbine lifetime.

Variables tab: If available, for each variable, enter the number and range of the bins to be used. If the Minimum and Maximum are both left as zero, a bin range will be calculated which is suitable across all the load cases. For rainflow cycle and fatigue analysis, it is possible to specify a minimum range below which cycles are not counted.

8.16 Ultimate Loads

This calculation scans a number of load cases to find the maximum and minimum values of a specified set of variables, and records the simultaneous values of all the other variables in the list.

From the **Ultimate Loads** window, click **Define Channels and Load Cases** to open the Multiple Processing window. Use the **Add Load Case** and **Add Variable** buttons to build up the required lists of load cases and variables.

For users with the **Advanced Post Processing** module, a list of **Load Case Groups** can be set up along with their corresponding safety factors. A check box on the **Load Cases** tab enables these to be assigned to the load cases. During processing the safety factors will then be applied. Safety factors may also be assigned to individual variables, and these will override the group safety factor.



Each load case should have a unique identifier. If **Load Case Groups** have been defined, the identifier will attempt to set itself to a sensible default, which the user is free to change if required. For each variable, specify whether its maximum and minimum are to be found.

If **Load Case Groups** have been used, an additional check box on the **Variables** tab allows the safety factor associated with the extreme load to be reported in the extreme loads tables.

If a blade manufacturer has provided a load configuration file, this can be used to produce custom output for that particular blade.

An additional feature available to users with the **Advanced Post Processing** module is to combine blade loads across all blades. With this option the maximum and minimum loads across all blades will be reported.

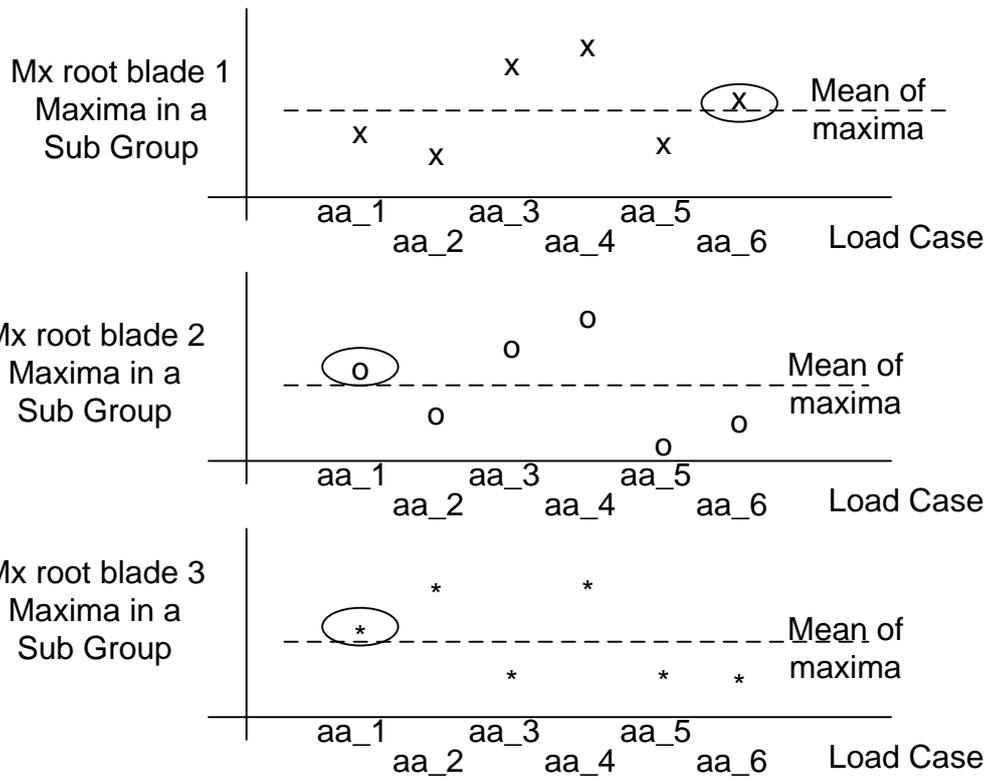
The output from this calculation consists of an ASCII text file with many rows and columns. The file name will be <directory> \ <run name>.\$mx where the directory and run name are entered at the time of running the calculation. The file is suitable for importing directly into a spreadsheet, and can also be tabulated using the **Multiple Plots** (see [9.12](#)) feature. For users with the **Advanced Post Processing** module, additional output is available in the form of histogram plots, which present comparison between the load cases group. A drop down list on the **Load Cases** tab allows the desired type of histogram output to be selected.

For users with the **Advanced Processing** module, an additional check box allows IEC-3 Sub Groups to be turned on. This allows the user to group several simulations together and request that **Bladed** finds a representative simulation that is either closest to the mean of maxima across all simulations within the group, or one that is closest to the mean of the top half across all simulations within the group. **Bladed** will identify the groups by an identification string used within the path or the run name, and these should be entered in the appropriate text box on the **Load Case Groups** tab.

With the IEC-3 Sub Groups box checked, two methods for combining the loads across **Bladed** are offered on the "Variables" tab:

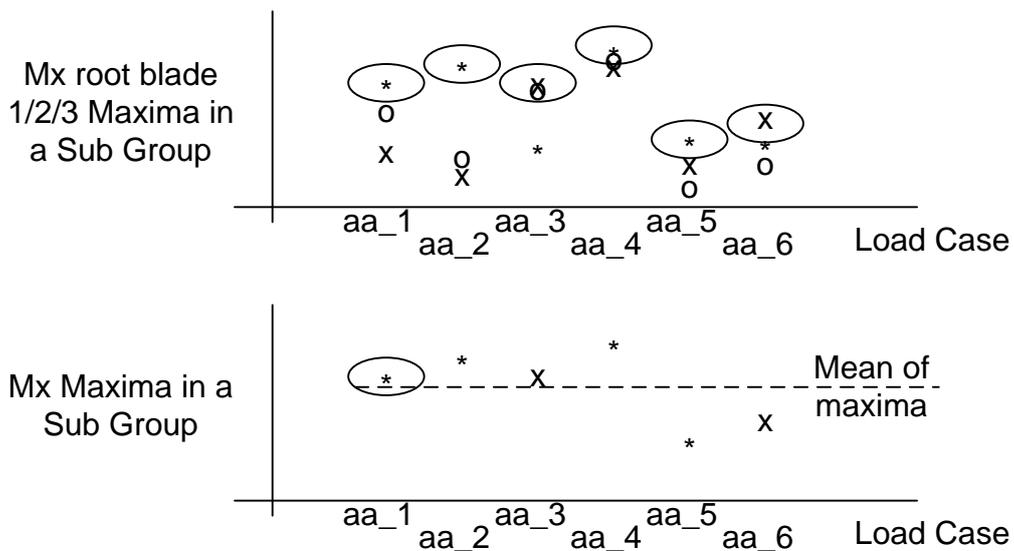
- The default method "Combine after evaluating subgroups" will have **Bladed** first get the representative simulation of a sub group for each blade as explained above, and then pick the largest extreme load across all blades.
- The method "Combine before evaluating subgroups" will have **Bladed** first select the largest extreme simulations across all blades and then get the representative simulation of a subgroup.

Here is an illustration of both combining methods where the maxima of Mx blade root is plotted against the simulation names for a three blades turbine. In this example the sub group consists of six simulations: aa_1 to aa_6. The "_" character in the load case name was defined in the Load case Groups tabs to be the identification text for a MEAN type subgroup.



In this first method, the simulation closest to the mean of maxima is selected as the representative simulation for each blade (circled in the diagrams above). The simulation with the largest maximum between all three of the circled representative simulations is finally selected.

Combine before evaluating subgroups method



In this second method, the simulations are combined across the blades first. That is the blade with the largest maximum is selected for each load case as shown in the top diagram above. Then the simulation closest to the mean of the maxima of these six selected simulations is found and circled in the bottom diagram.

You can also use the Ultimate Load Cases calculation to process one or more Ultimate Loads output files and give a graphical presentation of results.

8.17 Ultimate Load Cases

This calculation generates a list of maxima and minima of loads and the load cases in which they occur.

In the **Ultimate Load Cases** window, click **Add...** to select output files from ultimate loads calculations. Select the required variables from the list of available variables.

To view the results, click the **Data view** icon on the main toolbar and select the **Special** graph type. Then find the calculation results by clicking **Channel 1**. The results will be presented as a histogram.

8.18 Flicker

Flicker is a measure of the severity of voltage fluctuations caused by equipment (such as a turbine) connected to the network. It may be calculated either from a voltage time history or from active and reactive power. Click the **Select...** button(s) (see [8.20](#)) to define the time histories to use.

Range of data to use: if desired, flicker may be calculated from just a part of the time history. To do this, enter a **To** value which is greater than the **From** value.

If flicker is calculated from active and reactive power, it can be calculated for a range of different network short-circuit power levels and impedance angles in a single calculation. Enter as many values as are required. **Set default values** supplies a set of standard impedance angles.

See also:

Generator (see [4.10](#)), Network (see [4.12](#)).

8.19 Linear Model

This calculation is available for users with a licence for the Control module. It converts the output of a **Model Linearisation** (see [7.11](#)) calculation into a state-space model, in a form which is suitable for controller design and is directly compatible with Matlab [\[2\]](#). A state space model has the following form:

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

where u is a vector of inputs, x is a vector of system states, and y is a vector of outputs. A , B , C and D are the state space matrices.

First click **Select...** to define which variables from the **Model linearisation** results are required as model outputs. The model states will depend on the dynamics which were selected in the turbine model for that calculation, and the inputs will include wind speed, pitch angle demand and generator torque demand. The state space matrix coefficients are calculated as the slope of a best fit line through a number of points generated by perturbations of different sizes away from the steady state condition.

The **Minimum acceptable correlation coefficient** defines whether a best fit will be accepted or not. If the correlation is poorer than this, that particular matrix coefficient is set to zero. A value of 0.5 – 0.8 is generally suitable.

Two output files are created in the selected output directory, using the selected run name and the following file extensions:

<RunName>.mat This is a '.mat' file suitable for reading directly into Matlab [\[2\]](#).

<RunName>.\$m2 This is an ASCII text file, also suitable for reading into Excel, containing the same information.

Each file contains the four state space matrices for each operating point, and also wind speeds and rotor azimuths defining the operating points. There are also vectors containing the steady state values of all inputs, states and outputs at each operating point, as well as some additional information which may be

of use in controller design such as the gearbox ratio, number of blades, and nominal speed and torque values.

In the .mat file, the variable names used are fairly self-explanatory, except that the state space matrices and the names of the inputs, states and outputs are all stored in a structure called SYSTURB. The elements of SYSTURB are:

inputname	The names of the input variables (character array)
outputname	The names of the output variables (character array)
statename	The names of the state variables (character array)
A, B, C, D	Arrays of the state-space matrices for the different operating points; thus $A(i,j,k,l)$ is the i,j^{th} element of the A matrix (i.e. row I, column j), for the k^{th} wind speed and the l^{th} rotor azimuth angle.

This structure can readily be converted to a Matlab lti model array using the single Matlab command:

```
SYS = ss (SYSTURB.A, SYSTURB.B, SYSTURB.C, SYSTURB.D, ...  
         'inputname', cellstr(SYSTURB.inputname), ...  
         'outputname', cellstr(SYSTURB.outputname), ...  
         'statename', cellstr(SYSTURB.statename));
```

Diagnostic plots may be generated for all matrix coefficients, or for all matrix coefficients whose correlation coefficients fall within a specified range of values. **Note:** Diagnostic plots cause a much slower calculation, and may generate very large numbers of plot files; these are stored as enhanced metafiles in the selected run output directory, with the following naming convention:

<RunName><M><W>Az<A>_<row>_<col>_<?><cc>.emf

<RunName>	the run name selected for the Linear Model calculation
<M>	which state-space matrix: A, B, C or D the plot refers to
<W>	the operating point wind speed
<A>	the rotor azimuth angle
<row>, <col>	which particular coefficient of the matrix the plot refers to
<?>	OK for a coefficient which was accepted, X for one which was rejected
<cc>	the correlation coefficient obtained for the best fit line.

8.20 Extreme load extrapolation

The extreme load extrapolation is available to users with an **Advanced Processing** licence.

This calculation uses statistical methods to predict the extreme load for a defined return period by extrapolating from the extreme load values obtained from a large number of simulations performed with a range of external conditions.

Typically this calculation is used for extrapolation of loading during power production. It is recommended that for a satisfactory distribution fitting, at least 50 simulations are carried out for each external conditions bin. For onshore calculations each simulation is usually 10 minutes long, but at least 40 minutes is recommended in the case of offshore calculations.

A Least Square fitting method is used and the calculated parameters for each simulation bin are logged in a "*. \$FP" file in the output directory.

Click the **Select** button to define the list of load cases and variables. As each load case is read in, it will be assigned to either just a wind speed bin as defined in the wind bins module, or a wind speed and wave height and wave period bin, as defined in the **scatter diagram** (see [6.15](#)) module. For the latter case, a lower, mid or upper bin approach can be used. Use the option in the **Annual Wind Distribution** (see [6.11](#)) screen to switch between wind only or wind and wave bins.

Enter:

The extrapolation period: Extrapolating from the given simulations, the expected extreme load for this period will be calculated.

The aggregation method: The user can choose to first weight and aggregate all extreme load results across the wind speed bins (or wind and wave bins), onto which a probability distribution is then fitted to give the best fit probability distribution model. The other option is to aggregate after fitting, where probability distributions are fitted first to the extreme load results per wind speed bins (or wind and wave bins), and these fitted distributions are then weighted and summed to give the overall response function for each load.

The type of distribution to be fitted: This can either be Gumbel or Three Parameter Weibull.

The Weighting Factors: The contribution of the extreme load results to the least square data fitting calculation can be influenced by the use of different weighting factors. In particular the weighting factors are useful to ensure that enough contribution is given to the greatest extreme loads values.

Default weighting factors are provided, but the user can visually verify how well the exceedance probability curve fits the data by looking at the exceedance graph, which plots the fitted exceedance curve against the extracted extreme load results.

The weighting factors can then be adjusted to improve on the results.

The weighting factors table expects a range with a corresponding weighting factor. For example an 80% range with a factor of 0.01 means that the first 80% of the extreme loads, sorted in an ascending order, will have their fitting error contribution factored by 0.01. A second weighting range of 95% would correspond to the next 15% of the sorted values, and so forth.

Diagnostic plots: Tick the box to generate plots for each bin individually. Diagnostic plots can slow down the calculation. They will be stored as enhanced metafiles in the selected run output directory. It is only available with the "aggregate after fitting" method.

Confidence Intervals: Two methods, Binomial and Bootstrapping, calculate the confidence intervals for each wind speed bin (or wind and wave bin). For both methods the user needs to specify the confidence interval and the load fractile to be used. These fields accept numbers between [0 - 1].

The bootstrap expects in addition the number of bootstraps and the size of the sample which the user would like to use. Both fields accept any positive integer. If the sample size is not equal to or smaller than number of simulations in each wind speed bin, the sample size is automatically set to total number of simulations in the wind speed bin.

The **peak over threshold** method is activated by the checkbox. This allows several extremes to be used from any one simulation. The threshold can be the same across all bins or alternatively defined for each bin separately. The threshold is the mean plus a user defined number of standard deviations above the mean plus a user defined multiple of (maximum-mean). Zero can be left in these boxes if either or both definitions are not required. In addition to the threshold, the user can specify criteria for statistical independence of points. This means that in order to be considered independent, points must be separated by a certain number of rotor revolutions, a certain amount of time, or both. Again, these entries can be left as zero if not required. The total number of used peaks per simulation using this method is logged in a "\$PT" file in the output directory.

A **Blum test** can be used to check if the local extremes, extracted using the Peak Over Threshold method, are independent. The Blum test statistics B are then calculated and output separately.

8.21 Offshore Code Checking¹

This calculation links **Bladed** load time histories for a given offshore support structure to the ANSYS® or SACS codes. This enables the automatic fatigue and ultimate limit state (code-checking) routines in these codes to be used directly on the **Bladed** load output.

Choose the code you wish to link to (ANSYS® or SACS) from the drop down menu. Click the **Select...** button to define the internal load time histories to be processed. This will take you to the Multiple Processing (see 8.15) module, from which you will be able to specify a number of load cases and variables to be processed in a single calculation together with the associated probabilities for each load case. Select the tower nodes for which you require code checking to be performed, and **Bladed** will automatically enter all members associated with that node.

¹ ANSYS and any and all ANSYS, Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of ANSYS, Inc. or its subsidiaries in the United States or other countries.

8.21.1 *Bladed*-ANSYS® link

If the ANSYS® software package is selected you must first run the Generate Database option. This calculation will store the structural description, internal load time histories and associated probabilities from **Bladed** into an ANSYS® product compatible database. A separate database will be created for each load case in the list. To run this calculation you will need to link to a toolkit.dll file, which can be found in your ANSYS® installation directory. Please note that the toolkit.dll in ANSYS® versions 13 and earlier is limited in the number of time steps it can consider and may cause the **Bladed** code checking link to fail.

An input file is created (ANSYScodecheckingInput.dat) which can be used as an input for the ANSYS® Fatjack or ANSYS® Beamcheck calculations. This file is based on a combination of user-defined inputs (pre-defined by the user in a template file within the **Bladed** installation directory) and automatic input commands from **Bladed**. A detailed explanation of the input commands is given in the ANSYS® Fatjack and ANSYS® Beamcheck user manuals.

The ANSYS® code-checking routines can be run using the Run Calculation option. To run this calculation you will need to link to the .dat file created in the previous step, together with the appropriate .exe file for the routine you wish to use (FATJACK or BEAMCHECK). These can also be found in your ANSYS® installation directory.

8.21.2 *Bladed*-SACS link

If the SACS software package is selected you need only to run a single calculation. This will store the structural description, list of load cases and associated probabilities in a SACS input file (sacs.in). This input file can be read by the SACS software and the internal load time histories processed into a SACS-format database for use by the SACS code-checking routines.

8.21.3 Sesam Fatigue Manager

If the Sesam software package is used then no additional calculations are required from **Bladed** as the code checking of **Bladed** results is all done within the Sesam Fatigue Manager.

This enables the Sesam user to:

- Choose **Bladed** simulations for post processing.
- Convert the **Bladed** loads output to Sesam format.
- Assign a weighting to each simulation for the fatigue life calculation.
- Perform the code checking and fatigue life calculations.

8.22 Data channel selection

This section describes the procedure for selecting calculation results for viewing or post processing.

Click the channel selection button on either the graph parameters (see 9) screen (for viewing data), or on any of the post processing (see 8) screens. Then take the following steps to find the data you require:

1. Select the disk drive and the directory where the data is to be found using the selection boxes provided.
2. Select the run name which identifies the calculation results.
3. Click on the **Data group** containing the required data. The **Variables** box below will show the data available in this group. In some situations, it is possible to select a number of data groups simultaneously.
4. When the correct group has been selected, select the desired variable in the box below. Again in some situations, it is possible to select a number of variables simultaneously.
5. If appropriate, select the independent variable as described below.

8.22.1 Selecting independent variables

Independent variables are shown in the box at the bottom of the channel selection screen. The number of independent variables required depends on the context. For example, a line graph or histogram



requires one independent variable (shown as X-Axis), and a 3D graph requires two (shown as X-Axis and Z-Axis). For post-processing, the last independent variable (usually time) is selected by default and is not listed. Where there are more independent variables than required, particular values must be selected for the remaining variables, by clicking on the variable and selecting a value in the box on the right. Double-click on a variable to change whether or not it will be used as an independent variable.

For example, if the independent variables are **Blade station radius** and **Time**, you may click on **Blade station radius** and select the radius required, leaving **Time** as the X-Axis for the graph. Alternatively, double-click on **Time** to select a particular value of time, and you can then plot the variable at that time as a function of blade station radius. If you make both variables independent, you will get a 3D graph.

For multiple processing of variables defined at different blade or tower stations, it is possible to select intermediate points between defined stations. The variable will be evaluated at that point by linear interpolation.

8.22.2 Messages and further information

The **View Messages** button displays any warning or error messages generated during the run. In a few cases, some additional information is available by clicking the **View Info** button.

8.22.3 Deleting information

Buttons are available on the data channel selector to permanently delete a data group or an entire run.

9 GRAPHICAL DATA VIEW FACILITY

Click the **Data view** icon on the toolbar to open the **Graph Parameters** screen.

First click the selection button for either a two-dimensional or three-dimensional graph. Then click the **Channel 1** button to select the data you wish to view - see Data channel selection (see [8.15](#)) for details of selecting data for two- or three-dimensional graphs. A **Special** graph type is available for results of the Ultimate Load Cases (see [8.17](#)) calculation.

If required, make any changes to the format of the graph on the **Graph Parameters** screen. Then click the **View Graph** button to display the graph. The graph can then be printed if desired, using the **Print** button, or saved to a metafile or bitmap using the **Save** button. The metafile can then be inserted into a report (see [9.10](#)). Alternatively, copy the graph to the clipboard using the **Copy Metafile** or **Copy Bitmap** button. Graphs default to a maximum of 6000 datapoints; the Preferences menu may be used to increase this limit if necessary.

The selected graph data may also be tabulated by clicking the **Tabulate** button (see [9.9](#)). Select whether to view the tabulated data on screen or send it to an ASCII file, a WORD document or an EXCEL spreadsheet.

Click **Stats** to tabulate the mean, minimum, maximum and standard deviation of the selected datasets.

9.1 Graphs of several variables

In the case of two-dimensional graphs, it is possible to plot up to six datasets on one graph. Simply click additional channel selection buttons to define the data required. Click the **Clear** buttons to remove unwanted channels. Any channel may also be disabled temporarily by means of the check-box on the right hand side.

9.2 Graph styles

Two-dimensional graphs may be displayed as line graphs, points or bar charts by clicking the appropriate Style button on the **Graph Parameters** screen. In the same way, three-dimensional graphs may be displayed as surface plots or column plots.

9.3 Grids and logarithmic axes

Use the check-boxes on the **Graph Parameters** screen to specify whether or not to add X and Y grids to the graph, and whether or not to plot on logarithmic axes in the X and/or Y directions. Logarithmic axes are only available on two-dimensional graphs.

The **Darker grids** option may be useful for printing the graph to some black and white printers. The faint grid lines may not always appear, depending on the printer driver.

9.4 Units

A **units** selection box to the right of each channel allows a choice of units for the variable being plotted, where appropriate. Two selection boxes are available for the case of cross-spectra and transfer functions with composite units.

Selection boxes in the lower part of the **Graph Parameters** screen give a choice of units for the X-axis where appropriate, and for the Z-axis in the case of three-dimensional graphs.

9.5 Axis limits

The default axis limits shown in the lower part of the **Graph Parameters** screen may be changed if desired, to alter the limits on the X and Y axes, and also on the Z axis in the case of three-dimensional graphs. The default tick interval may also be changed.

Clicking on an entry with the Shift button depressed will freeze it so that it will no longer be updated automatically as channel selections and units are changed. A second shift-click will unfreeze the entry, so that it will be automatically updated.

9.6 Graph titles

The default axis labels (X, Y and Z if appropriate) and graph titles (top and bottom) may be edited in the lower part of the **Graph Parameters** screen. Use a tilde (~) to start a new line in the graph title.

Clicking on an entry with the Shift button depressed will freeze it so that it will no longer be updated automatically as channel selections and units are changed. A second shift-click will unfreeze the entry, so that it will be automatically updated.

9.7 Graph legends and line styles

For two-dimensional graphs, a legend may be added. This is useful for plots showing more than one dataset, as it allows the different datasets to be identified. Click **Legends and styles** to edit the default legend captions and to specify where the legend is to appear on the graph. One of five line thicknesses may be selected for each line of the graph.

If the default legend captions are too long, use the **Preferences...Label formats** pull-down menu on the **Graph Parameters** screen to customise them by omitting any common paths, run names, etc.

Clicking on a legend entry with the Shift button depressed will freeze it so that it will no longer be updated automatically as channel selections and units are changed. A second shift-click will unfreeze the entry, so that it will be automatically updated.

9.8 Cross-plots

Two dimensional graphs can be presented as cross plots if two or more channels with compatible independent variables have been selected. Use the **X-axis** selector to define which channel is to be plotted along the X axis. The other channels will then be plotted against this one.

9.9 Tabular output of results

Click the **Tabulate** button on the **Graph Parameters** screen to tabulate the current graph data. A dialogue box allows you to select the format:

Screen: the data can be viewed on the screen.

ASCII: to save the table as tab-delimited ASCII text.

Excel: to save the table as a Microsoft Excel workbook (You must have Microsoft Excel installed)

Word: (the default) to save the table in a Microsoft Word document (you must have Microsoft Word installed).

When viewed on the screen, the data may also be copied into the clipboard for transfer to another application. This is sometimes quicker than sending the table directly to EXCEL. For a large table, it is sometimes faster to send it to an ASCII file first and then read the file into EXCEL.

9.10 Refreshing graphs

Click **Refresh Data** to re-read any plot data which has changed since it was last read in, for example in the case of a plot of results from a simulation which is still in progress, or if a calculation has been

repeated, overwriting the old results. Any channel can be forcibly re-read using the **Data** pull-down menu.

Select the **Auto-refresh graph every...** option to re-read the graph data at the specified time intervals. This is useful for a graph to update itself automatically as a simulation progresses. If the interval is set to zero, the graph will not update, the **View** button will now use the same graph window each time it is pressed rather than opening a new window each time.

From the pull-down menu, use **Data -> Change runs** to read new graph data from one or more different runs, for the same variables.

9.11 Graph configurations

The **File** pull-down menu allows you to save and load **Bladed Graph Configurations** (.bgc files) which contain all the necessary details to re-create a graph (the graph size, the channel definitions, styles, and also axis limits and graph annotation whether fixed or automatic). When loading a configuration, a prompt allows the run(s) and variable(s) to be changed. This provides a useful and flexible way of storing and re-using a particular graph configuration for different sets of results.

9.12 Multiple Plotting and Tabulation

Click the **Multiple Plots** button to bring up the Multiple Plotting screen. This feature allows a graph configuration file to be used as a template for a graph or table, with the variable in the graph or table being replaced from lists of variables and runs. Clicking the **Start** button will start the series of plots in the order specified by the **sequence control**. This facility is only available to users with the **Advanced Post Processing** module. The graphs and tables may be viewed on screen and/or sent directly to a report.

9.12.1 Graph configurations

Add one or more graph configurations (see [9.11](#)) to the list, specifying whether any of the runs and/or variables defined in the graph configuration are to be replaced from the lists below.

9.12.2 Replacing variables

If any variables in the graph configuration(s) are to be replaced, they will be replaced by each item in the **Variables** list in turn. **Add** any desired variables to the list, or use **Add Set** to process complete sets of a specific type of graph or table. Clicking on the **Add Set** button will bring up a choice of the type of set required. All the data in a particular run that is part of the specified set will then be processed. The following sets are available:

- Ultimate Loads (histogram output)
- Rainflow Cycle Exceedance
- Rainflow Cycle By Range
- Lifetime Weighted Equivalent Loads

9.12.3 Replacing runs

If any runs in the graph configuration(s) are to be replaced, they will be replaced by each item in the **Runs** list in turn. **Add** any desired runs to the list; click **Add Many** to add all the runs in a directory or a directory tree.

9.12.4 Sequence Control

The sequence control allows the user to specify the order in which the plots or tables will be generated.



9.12.5 Tabulation of Ultimate Loads

Select the **Tabulation of Ultimate Loads** to tabulate the Ultimate Loads from each of the ultimate loads calculations specified in the **Runs** list. There is no graphical output available in this case.

9.12.6 Output

Use the options in the **Output Requirements** to specify what type of output is required, and where it should be sent. Graphs can be displayed on screen, stored as metafiles, and sent to a WORD document. Tabulated data and statistical data can also be sent to a WORD document or to an ASCII text file. If an output file is specified which already exists, data will be appended to the end of the file.

10 REPORTING

The **Bladed** reporting system enables the user to generate reports describing the wind turbine characteristics and the details of any particular calculation. Calculation results can also be entered into the report in tabular or graphical form. The report is prepared directly in Microsoft Word format. Alternatively, ASCII text format is available as an option if required, but this clearly does not allow the inclusion of graphical output.

From the **Reports** menu on the main toolbar, select the required report format from the following options:

ASCII: to save the table as tab-delimited ASCII text.

Word: (the default) to save the table in a Microsoft Word document (you must have Microsoft Word installed).

The current format can be saved as the default using the **Save** item of the **Options** pull-down menu on the main toolbar.

There are two main ways of generating reports:

- **Project reports:** these may contain any currently assigned modules defining the turbine itself, as well as calculation parameters and wind field details if required.
- **Calculation reports:** these may include any turbine details which are currently assigned and which are relevant to the calculation, as well as details of the calculation itself and the wind field used if applicable. They may be generated as stand-alone reports, or appended to another report such as a project report.

As well as adding graphs and tables, the reports may also be edited and printed within **Bladed**.

10.1 Project reports

From the **Reports** menu on the main toolbar, select **Write project report**. A dialogue box allows a document name to be chosen. If it is an existing document, you may choose to overwrite it or to append to it.

The **Report Contents** screen then appears. Select the desired report contents by checking or unchecking the appropriate boxes. By default, this includes any currently assigned modules defining the turbine itself. Calculation parameters and wind field details may also be added. Click on **OK** to generate the report.

10.2 Calculation reports

Calculation reports can be written for any calculation which has been performed with a defined run name. From the **Reports** menu on the main toolbar, select **Write calculation report**. A dialogue box allows a particular run to be chosen.

The **Report Contents** screen then appears. Select the desired report contents by checking or unchecking the appropriate boxes. Select the desired report contents by checking or unchecking the appropriate boxes. If desired, change the **Destination** selector to **Other** to change the report file name. You may append to an existing report if desired. Click on **OK** to generate the report. If an **Other** destination was selected, a dialogue box appears to allow you to select an appropriate file name

10.3 Adding calculation results to a report

When viewing a graph (see [9](#)) click the **Save** button to save the graph to a metafile. Then, or at any later time, go to the **Reports** menu on the main toolbar and select **Append Graph**. Dialogue boxes appear, allowing you to select the metafile required and the Word document to which it will be appended. The graph may be inserted fully or as a link - see [Linked graphs \(see 10.5\)](#).



Alternatively, click **Copy Metafile** on the graph window to copy the graph to the clipboard, and paste it directly into the report.

Graphs cannot be appended to ASCII reports.

To append a table of results rather than a graph, refer to Section [9.9](#).

10.4 Editing and printing reports

An existing report may be edited and printed. From the **Reports** menu on the main toolbar, select **Edit report** or **Print report**.

If **Edit report** is selected and the report format is set to **Word**, Microsoft Word will be opened and any desired editing may be carried out. If the report format is ASCII, the Windows Notepad editor will be opened.

10.5 Linked graphs

From the **Tools -> Preferences** menu on the main toolbar, the **Insert graphs as links** option allows graphics inserted into a Word report using the **Reports** pull-down menu option to be linked to an external metafile. The Word document will be smaller, but not transportable unless the metafile is moved with it.

10.6 LM Uploader

Bladed 4.1 comes with a facility to create a single file of extreme and fatigue loads in the format specified by LM Wind Power. You can launch the uploader from the Tools menu. After entering the data you can save a file which can be uploaded or emailed to LM Wind Power.

11 THE WINDFARMER LINK MODULE

WindFarmer Link is an optional module of **Bladed** which utilises the combined capabilities of **Bladed** (see [3]) and **WindFarmer** (see [4]) for site-specific turbine loading analysis. The Theory Manual gives full background information and details of the calculation methods used.

Before using the **WindFarmer Link** module, a minimum of four input files need to be created:

- A **WindFarmer** output file (see 11.1). This requires **WindFarmer** version 3.4.1.3 or later.
- A **Bladed** wind file template (see 11.2).
- One **Bladed** turbine template for each turbine type defined in the **WindFarmer** model (see 11.3).
- At least one **Bladed** fatigue processing template, or up to one for each turbine type defined in the **WindFarmer** model (see 11.4).

To access the **WindFarmer Link** module from the main **Bladed** toolbar, use the **Batch** pull-down menu and select **WindFarmer link**. Note that a valid 'WindFarmer Link' licence is required to access and run this module.

In **Calculation setup**, click **Project File Creation** and proceed as follows:

- Select the **WindFarmer output file** (see 11.1).
- Specify the **cut-in and cut-out wind speeds** (used in calculating the time spent in each bin).
- Enter a list of **wind speeds to be used for the simulations**. These must be between the cut-in and cut-out wind speeds, and should be entered in ascending order, separated by spaces.
- Specify whether these wind speeds represent the **top** or the **centre** of each wind speed bin.
- Select the **Bladed wind file template** (see 11.2).
- Specify the **wind file sampling frequency and spatial resolution**. These values will override the values used in the wind file template.
- Enter the **number of turbine types** specified in the **WindFarmer** model.
- For each turbine type, select the appropriate **Bladed turbine template** (see 11.3) and **fatigue processing template** (see 11.4).

Options and settings: before setting up the required **Bladed** calculations, a number of options and settings may be specified:

- Use the **Ranking Calculation** (see 11.5) to reduce the number of simulations required by selecting only those turbines which are likely to be the most heavily loaded.
- Choose **Sectors and Options** (see 11.6) to specify how many simulations to use for different wind direction sectors, to limit simulations to certain turbines (for example after running the Ranking Calculation), or if any particular parameters from the **WindFarmer** run need to be overridden.

Then click **Start project file creation** to generate all the required **Bladed** calculation files. Checkboxes are provided to allow the wind file creation, simulation and post-processing calculation files to be generated separately if desired. If **Generate Figures** is checked, diagnostic plots will be created to show the selected simulation wind directions in relation to the wakes.

A dialogue will then prompt for the location of the calculation outputs to be entered. The user is prompted to save the **WindFarmer Link** configuration details in the output directory, as a **Bladed** calculation details file with default name *wfblink.\$PJ*. The calculation outputs will be found in sub-folders, as detailed in Section 11.7. The outputs include the .\$pj files required to run all the wind file creation calculations, simulations and post-processing calculations required to carry out the site-specific load calculations. These should be placed into a standard **Bladed** batch (see 7.2.7): from the main toolbar, select **Batch queue** and from the **Batch List** screen, use **Edit ... Create jobs ... From a set of existing runs** to generate the batch list. Ensure that the wind file creation jobs appear first in the list, and the post-processing calculations last. Then click **Start Batch** to set off the entire set of calculations.

Finally, the resulting loads can be extracted from the Fatigue calculations using standard **Bladed** reporting methods.

11.1 The *WindFarmer* output file

A working knowledge of **WindFarmer** is required to produce a suitable site specification output file. User guidelines related to energy calculations should also be adhered to for site specific load analysis. Furthermore, the following constraints should be adopted if the **WindFarmer Link** is to be used:

- 72 or more direction sectors are recommended (i.e. 5° bin widths or less)
- 0 – 70 m/s mast wind speed range
- Eddy viscosity wake flow model

It is further recommended that the *maximum length of near wake* in the eddy viscosity model options is limited to 10-15D, where D is the diameter of the turbine rotor – see [7.1](#).

The user should check that the ambient *mean* turbulence has been input to the **WindFarmer** model (see [7.1](#)), not the *characteristic* level, since the *mean* turbulence intensity should be used for the wake calculations. If the turbulence intensity required for the *Bladed* simulations is different, for example if the *characteristic* turbulence level is required, the appropriate scaling factor(s) should be introduced through the **Turbulence factor file** (see [11.6](#)). This ensures that only the ambient turbulence is scaled; and not the added turbulence calculated by the wake model.

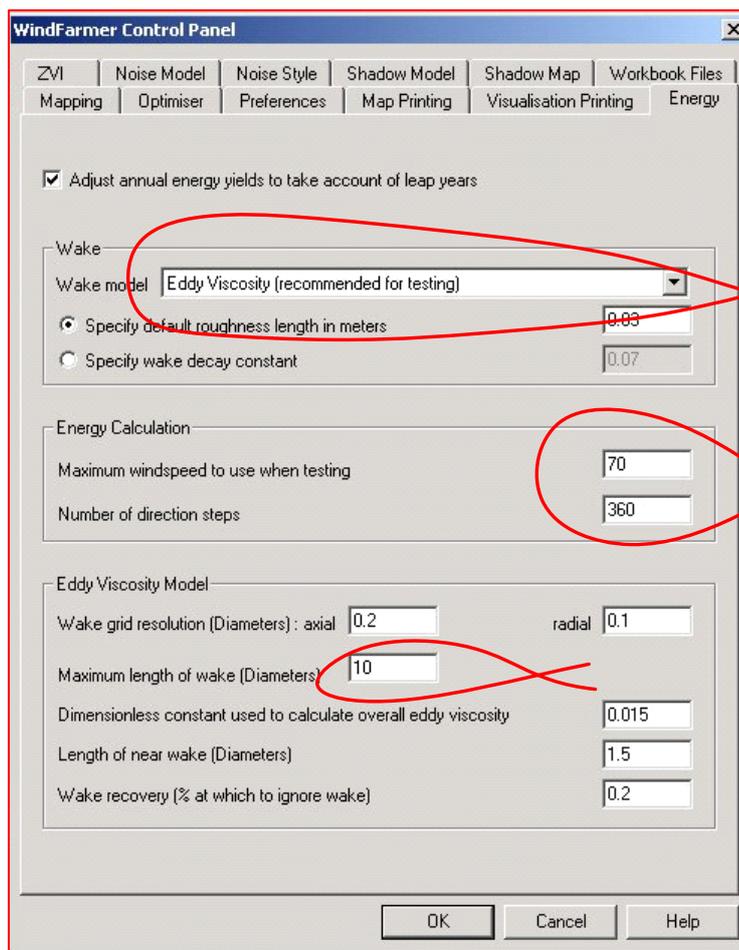


Figure 11.1: WindFarmer data entry

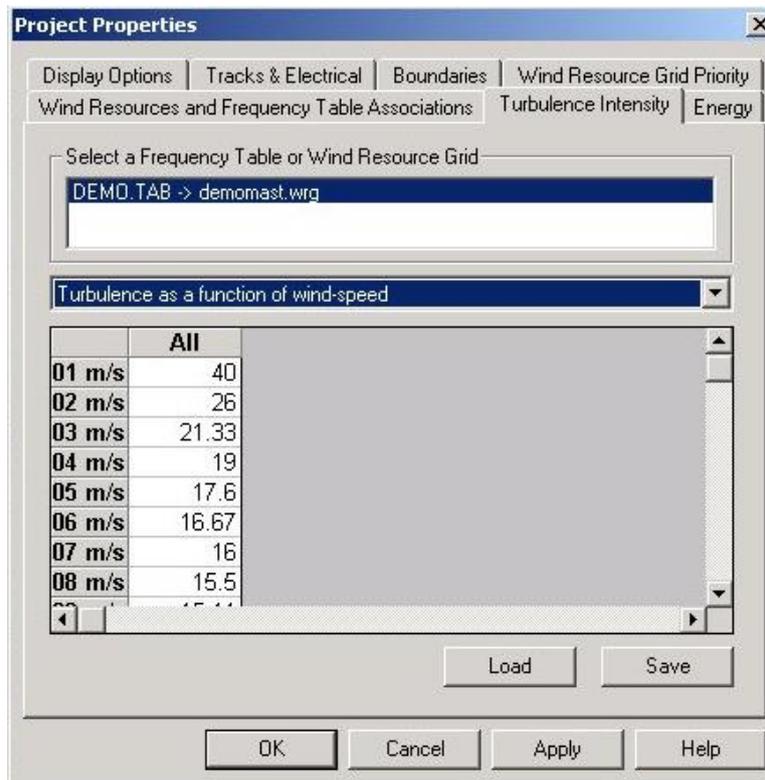


Figure 11.2: Turbulence intensity input to WindFarmer

The model can then be run and the site specification output file created. Upon successful completion of the simulation, select:

File / Export / Export Flow and Performance Matrix

and select the following export channels ([7.1](#)):

- (03) Turbine hub height wind speed (m/s)
- (06) Turbine speed-up
- (07) Turbine incident turbulence (%)
- (12) Turbine incident wake width (m)
- (13) Turbine incident wake offset (m)
- (14) Turbine incident wake centreline depth (m/s)
- (15) Terrain angle (deg)
- (16) Local turbine probability distribution
- (17) WindFarmer Bladed Link summary turbine data

The channels may be specified in any order. The 'WindFarmer Bladed Link summary turbine data' (channel 17) contains details of air density, spatial coordinates, hub height and turbine diameter of each turbine on site.

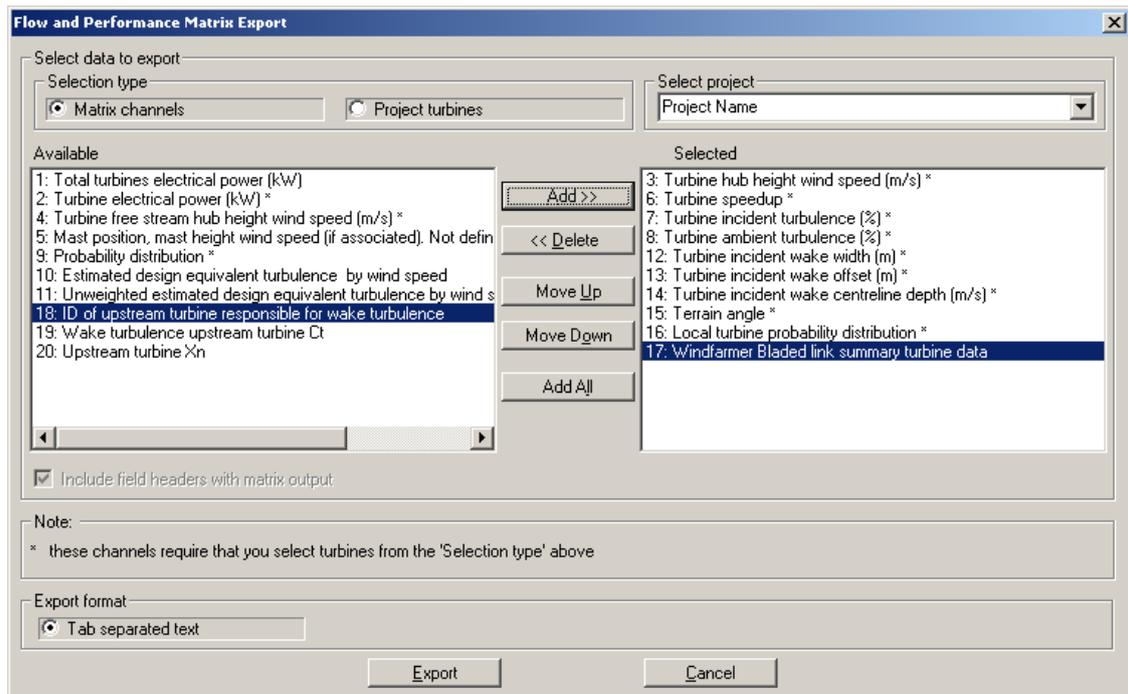


Figure 11.3: Export channel selection in *WindFarmer*

The user should finally select which turbines are required for load assessment. The order of selection is unimportant ([7.1](#)).

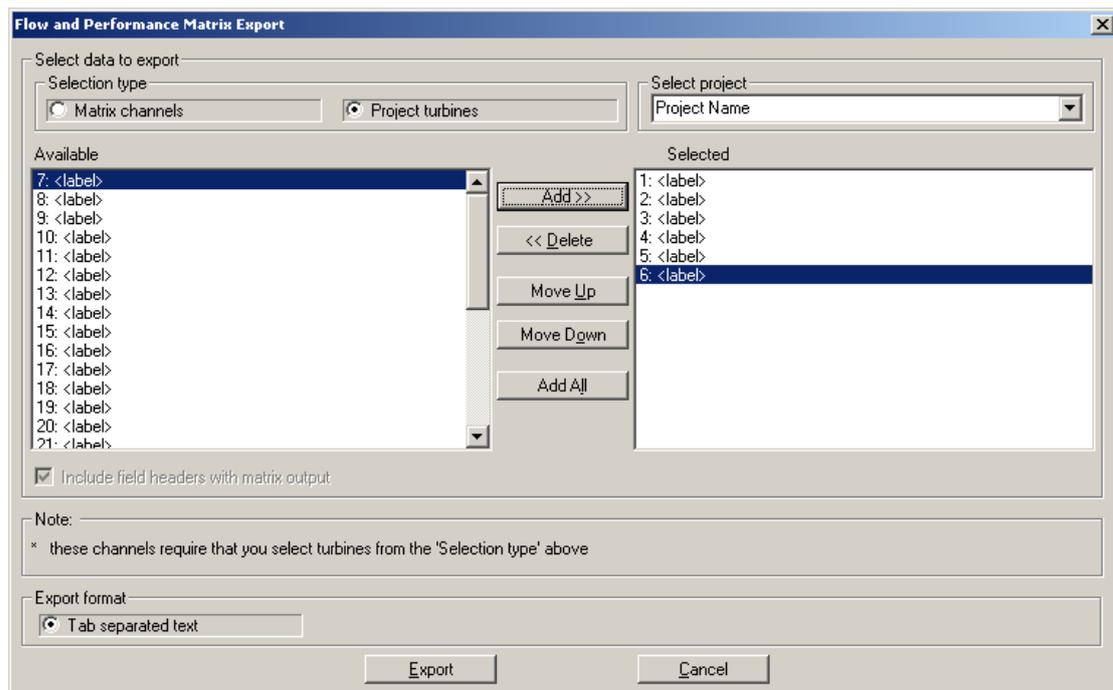


Figure 11.4: Turbine selection in *WindFarmer*

Then press *Export* to save the output to a single.txt file. This file can be saved to any convenient location.

11.2 The wind file template

This is the \$.pj file which is generated by running the *Wind Turbulence* calculation (see [6.10.1](#)) in **Bladed**. The **three-component Improved von Karman model** must be used, but all other values can be arbitrary as they are subsequently imposed by the program. Some of the required values are calculated by the program (for example the lateral and vertical extent of the turbulent wind field will be determined by the turbine diameter). However the wind file spatial resolution should be entered on the **WindFarmer link** screen as follows:

- Longitudinal sample frequency (replaces the Frequency along X defined in Section [6.10](#))
- Horizontal and vertical resolution (replaces the Number of points along Y and Z defined in Section [6.10](#))

11.3 The turbine model template

This is the \$.pj file which is generated by running the *Power Production Loading* calculation in **Bladed**. All the normal requirements for this calculation must be defined, and the following items must be defined in addition:

- **Turbulent wind** option selected (turbulence file details not important)
 - **Exponential Vertical Shear** in the Wind Shear (see [6.2](#)) definition.
 - **Tower Shadow** (see [6.3](#)) if known and applicable.
 - **Gaussian Wake** in the Upturbine Wake (see [6.4](#)) definition.
 - **Conditions for Idling Simulation** (see [5.16](#)).
- although the parameter values for the wind shear and Gaussian wake are immaterial.

11.4 The fatigue processing template

This is the \$.pj file which is generated by running the *Rainflow Cycle Count* (see [8.10](#)) post-processing calculation in **Bladed**. The calculation should be defined using the multiple processing option to select all the required variables from an arbitrary load case simulation, which has been run using the same turbine model as in the turbine model template.

11.5 The Ranking Calculation

This is an optional pre-processing step used to calculate an index of severity of loading for all the turbines, based on the wind conditions calculated by **WindFarmer** and a chosen material fatigue SN slope. This means that only the turbines which are likely to be the most severely loaded can be selected for the simulations. For large wind farms, this can greatly reduce the number of simulations which will be carried out.

Enter up to ten values of inverse SN slope to be used in the Ranking Calculation. An index of severity will be calculated for each turbine and for each inverse SN slope. Note that these values are *not* those used in the actual **Bladed** post-processing calculations, since these are defined separately in the fatigue processing template file (see [11.4](#)).

If desired, the index of severity for each turbine can be calculated relative to the wind conditions for a specified IEC (edition 2) site class (see [5.1](#)). To do this, specify the desired IEC wind speed and turbulence classes. The results will then produce the *margin compared to the selected site class*, which by convention is *negative* if the severity is greater than for the site class, rather than the severity indices themselves.

Check **Generate Figures** to get graphical as well as text file output. Examine the output to decide which turbines to include in the simulations, and enter these manually in **Sectors and Options** (see [11.6](#)). For the location of the output files, see [11.7](#).

11.6 Sectors and Options

Click **Sectors and Options** to define the number of direction sectors to be used for the wake-affected and ambient simulations at each wind speed, to narrow down the number of turbines to use in the simulations, and to override certain values from **WindFarmer** if necessary.

11.6.1 Direction sectors

For wake-affected simulations, select 1 or 2 wake simulations per wake. With one simulation per wake, the wind direction at a particular wind speed will be chosen such that the turbine is on the wake centreline; with two simulations per wake, the directions chosen will be such that the turbine is one turbine radius each side of the centreline.

For ambient simulations, there are two choices (see Theory Manual for more details):

(a) Uniform ambient sectors: enter the total number of direction sectors to use. One simulation will be defined for each sector, provided it contains at some non-wake-affected directions. Usually just one sector would be defined, but the user should choose more if there is a large variation of terrain angle or turbulence intensity by direction around the site.

(b) User Defined ambient sectors: Specify a simulation index number for each 30° direction sector. Use the same index number for sectors which are to be treated as one (usually because they have similar wind conditions). The index numbers must be positive consecutive integers starting at 1 and finishing at the total number of distinct ambient simulation directions to be used for each wind speed. This method is appropriate where the ambient turbulence varies significantly between sectors, but two or more sectors contain similar levels.

11.6.2 Overriding values from *WindFarmer*

Check **Turbines to analyse** if the simulations are to be carried out for only a subset of the turbines on the wind farm, for example those turbines likely to be the most heavily loaded. This might have been determined using the **Ranking calculation** (see [11.5](#)). The turbine ID numbers should be entered as a list, separated by spaces.

Check **Turbulence factor file** to specify a file containing turbulence intensity characteristic factors. This file may have any name but must contain a standard **WindFarmer** output file header followed by one or more turbulence factors. These factors are used to multiply the mean ambient turbulence intensities from **WindFarmer** to define the actual turbulence intensities to be used in the **Bladed** simulations. The turbulence factor may be a single value, or it may vary as a function of direction sector and/or wind speed in which case a matrix of values is entered: the columns represent direction sectors, and the rows represent wind speeds. To match **WindFarmer**, there must be either 1 or 12 columns (the latter representing the 30° wind direction sectors), and either 1 or 16 rows (the latter representing wind speeds from 1 to 16 m/s; the last row will be used for any higher wind speeds). Thus there are four possibilities: a single value, a 12-column row vector, a 16-row column vector, and a 16-row, 12-column matrix (as in the example in [7.1](#)). The header (the first 10 lines) must specify the actual numbers of columns and rows, but must be otherwise identical to the example.

```
#1
TI Factors *
11
Project Name
0
<label>
number of direction bins:
12
number of wind speed bins:
16
1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2
1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2
1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2
1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2
1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2
1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2
1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2
1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2
1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2
1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2
1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2
```

1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2

Figure 11.5: Example turbulence factor file. Entries in bold must not be changed.

The remaining checkboxes allow certain other parameters used for defining the wind field properties to be changed if necessary, overriding the values assumed by **WindFarmer**. This could be useful if, for example, terrain angle data had not been defined in the **WindFarmer** model

11.7 Output folders and file naming convention

Beneath the output directory specified by the user through the interface, the **WindFarmer Link** creates four sub-folders:

- fatigue (contains rainflow cycle count runs)
- internal (contains analysis diagnostics, intermediate outputs and plots)
- simulations (contains the time-history simulations)
- wind (contains the wind files)

Under each of these are individual sub-folders for each turbine selected for analysis. Each sub-folder is identified by the turbine ID number.

The simulations and wind directories are further partitioned into the discrete simulations created for each turbine. Directories of the form *Sabcd* are assigned to denote whether the simulations are ambient or wake-affected and if the latter, which wake is being described.

The first digit *a* states whether the simulation is ambient ($a = 0$), wake-affected ($a = 1$) or no damage ($a = 2$).

For wake-affected simulations ($a = 1$), the second and third digits state the wake number ($bc = 01$ being 1st, $bc = 11$ being the 11th and so on). The fourth digit states whether the simulation is at 0.5D or a centreline (OD) offset as follows:

- $d = 2$ -0.5D offset
- $d = 3$ 0.0D offset
- $d = 4$ +0.5D offset

For example, S1033 corresponds to a centreline (zero offset) wake affected simulation due to the 3rd wake-creating neighbour, as counted moving clockwise from 0 degrees (due north in **WindFarmer** convention).

For ambient or no-damage simulations ($a = 0$ or 2), *bcd* is used as a three-digit sequential number for each simulation. Ambient simulations start with S0001, S0002 etc., and there is just one no-damage simulation, S2001.

Pre-processing turbine ranking histograms are saved in the root of the *Internal* folder. In the *Internal* folder, under each individual turbine folder are two further sub-folders, *Checks* and *Tempstore*. The *Tempstore* folder contains information used internally by the software. It also contains the diagnostic output files. The *Checks* folder contains output files from the software to check that it has run as intended. It also contains a *Plots* folder which holds figures if they are selected as an output: three data plots are shown on each figure, the turbulence intensity, the centre-line velocity deficit (both as percentages) and the centre-line offset (in metres). Each figure represents a single wind speed bin and the x-axis shows the wind direction from north (0 degrees) through 360 degrees.

Graphical output can be viewed using the Image Viewer (see [11.8](#)).

Pre-processing ranking plots are saved in the *Internal* directory; simulation plots are saved in the *Internal/Checks/Plots* directory.



The simulation plots show how turbulence intensity, wind speed deficit and wake offset vary with wind direction, for each selected wind speed bin, for each selected turbine on site. The red dots indicate at which direction bins wake-affected simulations have been imposed by the code; black dots do the same for ambient conditions. They merely identify the *direction bin*; their location on the y-axis in the plots is arbitrary.

11.8 Image viewer

The Image viewer can be accessed from a button at the bottom of the **WindFarmer Link** screen. It includes a folder selector, below which a list of image files in the folder can be selected. When an image file is selected, the image appears in the right hand side of the screen.

APPENDIX A COMMUNICATION BETWEEN BLADED AND EXTERNAL CONTROLLERS

This appendix describes in detail the communication between **Bladed** and the user's external controller code. See also Section [5.9](#).

Appendix B gives some simple examples of external controller code in several languages.

A.1 The **Bladed** External Controller API

Bladed provides the facility for users to create and run their own controllers for the simulated turbine. These controllers can ask for simulation and model details, and 'control' the simulated turbine by setting various operational parameters. For example, the external controller could ask what the current generator speed is, and based on this value, set the demanded pitch actuator rate to try to achieve a different pitch.

The external controller has access to **Bladed** through a published API (Application Programming Interface). The **Bladed** API provides a suite of functions such as `GetNumberOfBlades` and `SetDemandedGeneratorTorque` so that an external routine can interact with the simulation. Documentation is available of the complete API in the accompanying *ExternalControllerAPI.chm*.

The mechanism by which the external controllers are integrated into the simulation is by compiling them as a function called "CONTROLLER" or "CONTROLLER_INIT" within a *dynamic-linked library* (dll). These functions are compiled independently of **Bladed** itself, and are called by the simulation at discrete intervals to allow the controller to modify the turbine's controls.

The function "CONTROLLER_INIT" is called on the first time-step at time zero, if it is found within the dll. If "CONTROLLER_INIT" is not found, the function "CONTROLLER" is called instead, and "CONTROLLER" is called on all subsequent time-steps.

A.2 Simulation Status

The function `GetSimulationStatus` returns the status of the simulation:

Value	Meaning
GH_DISCON_STATUS_INITIALISING or 0	First call to external controller at time zero,
GH_DISCON_STATUS_DISCRETE_STEP or 1	All subsequent time-steps.
GH_DISCON_STATUS_FINALISING or -1	Final call at the end of the simulation.
GH_DISCON_STATUS_REALTIME or 2	Real Time update step (for Real Time Test simulations only). On a call with the status flag set to 2, the DLL must exchange data with the turbine controller.

The controller can return its status as the function's return value:

Return Value	Meaning	Consequence
GH_DISCON_SUCCESS or 0	successful execution	Bladed will continue executing simulation.
integer > 0	execution with warnings	Bladed will report a warning message, quoting the return number.

GH_DISCON_ERROR
or integer < 0

error during
execution

Bladed will exit after having reported an error message to the \$ME and \$TE message files.

A.3 Sending messages to the simulation

The controller may send messages to the simulation, which will then be displayed to the user and written to the **Bladed** message file. These messages can be scaled as to severity, with different prefixes:

API Call	Prefix	Usage
ReportDebugMessage	*** Debug:	For messages to aid development – these should be removed before release.
ReportInfoMessage	*** Note:	For messages to help the user – for instance, progress messages.
ReportWarningMessage	*** WARNING:	For warnings which allow the program to continue.
ReportErrorMessage	*** ERROR:	For messages to be printed before returning - 1 as the error code.

The controller can send any number of messages to the simulation, and they will be displayed instantly.

A.4 Pitch and torque override

If the external controller is used for supervisory control actions such as starts stops, while the built-in continuous-time PI controllers are used for power production control, then it may be necessary for the external controller to specify the instant at which the supervisory control action takes over from the in-built controller action. Calling `SetPitchOverrideStatus(turbine_id, GH_DISCON_OVERRIDE_ON)` whenever the external controller is to control the pitch, overriding the built-in PI controller. Call `SetPitchOverrideStatus(turbine_id, GH_DISCON_OVERRIDE_OFF)` when the built-in PI controller should be controlling the pitch.

For variable speed turbines, use `SetTorqueOverrideStatus` in the same way to determine whether the external or the built-in controller should be controlling the generator torque.

A.5 Sending logging output to Bladed

Additional data may be sent back to **Bladed** for logging in supplementary output files in a similar format to that of other simulation outputs. This data can then be viewed directly using the Data View facility, or post-processed. This is particularly useful for debugging the controller, or for illustrating the details of its operation.

On the first call to "CONTROLLER", or in "CONTROLLER_INIT", the external controller should add the required number of logging values using the function `AddLogValue`. This is called once for each log value, with arguments for the name of the quantity being logged, and the units:

Allowed values for	Meaning (strict SI)
-	(No units specified)
1/T	s ⁻¹ (Hz)
A	rad
A/P	rad/W
A/PT	rad/Ws
A/PTT	rad/Ws ²
A/T	rad/s
A/TT	rad/s ²
F	N
F/L	N/m
F/LL	N/m ²
FL	Nm
FL/A	Nm/rad
FL/L	Nm/m
FLL	Nm ²
FLT/A	Nms/rad
FLTT/AA	Nms ² /rad ²
I	A
L	m
L/T	m/s
L/TT	m/s ²
LLL	m ³
LLL/A	m ³ /rad
M	kg
M/L	kg/m
M/LLL	kg/m ³
M/LT	kg/ms
MLL	kgm ²
N	(No units specified)
P	W
PT	J
Q	VAr
T	s
V	V
VI	VA

Table A.2: Allowed Units

On subsequent calls, the function `setLoggingValue` can be used to set the value in that time-step, and this value will appear in the log files. `setLoggingValue` requires the index of the log value. As the function `AddLogValue` returns the new number of log values, the index for that addition would be 1 minus this size, or else the function `GetLogIndexFromName` can be used to recover the index.

APPENDIX B EXAMPLE EXTERNAL CONTROLLER CODE IN SELECTED LANGUAGES

To assist the user to get started with the coding required for external controllers, this appendix presents a few simple examples. Some simple working examples are also provided in the Code Samples subfolder of your **Bladed** installation.

B.1 Simple example of DLL code written in C

```
#pragma comment(lib, "ExternalControllerApi.lib")

#include "ExternalControllerApi.h" // This defines the C functions that can be called, as well as 'turbine'.

using namespace GHTurbineInterface;

/// <summary>
/// Function to calculate the required pitch demand from the current pitch and wind speed.
/// </summary>
/// <TODO>Write this function.</TODO>
int calcs(const double measured_speed, const double measured_pitch, double pitch_demand)
{ return 0; };

extern "C"
{
    int __declspec( dllexport ) __cdecl CONTROLLER (const turbine turbine_id)
    {
        int result = 0;
        double measured_pitch, measured_speed, pitch_demand;

        measured_pitch = GetMeasuredPitchAngle(turbine_id, 0);
        measured_speed = GetNominalHubFlowSpeed(turbine_id);
        pitch_demand = 0.0;

        // Main calculation (User to supply calcs routine)
        result = calcs(measured_speed, measured_pitch, pitch_demand);
        // -----

        if (result >= 0)
        {
            result = SetDandedPitchAngle(turbine_id, 0, pitch_demand);
            if (result == GH_DISCON_SUCCESS)
            {
                ReportInfoMessage(turbine_id, "Set pitch");
                return GH_DISCON_SUCCESS;
            }
        }

        ReportErrorMessage(turbine_id, "Failed to set pitch");

        return GH_DISCON_ERROR;
    }
}
```

B.2 Simple example of DLL code written in FORTRAN 90

```
INTEGER(4) FUNCTION CONTROLLER [C] (turbine_id)

  USE mExternalControllerApi    ! This defines the C function interfaces

  IMPLICIT NONE

  ! Compiler specific: Tell the compiler that this routine is the entry point
  ! For the DLL. The next line is for the case of the Digital Visual Fortran
  ! compiler.
  !DEC$ ATTRIBUTES DLLEXPORT, ALIAS:'CONTROLLER' :: CONTROLLER

  ! Arguments
  INTEGER(C_SIZE_T), INTENT(IN):: turbine_id[VALUE]

  ! Variables

  INTEGER(4) RESULT
  REAL*8 dMeasuredPitch, dMeasuredSpeed, dPitchDemand

  RESULT = 0;
  dPitchDemand = GetMeasuredPitchAngle(turbine_id, 0)
  dMeasuredSpeed = GetNominalHubFlowSpeed(turbine_id)
  dPitchDemand = 0.0

  ! Main calculation (User to supply calcs routine)

  !RESULT = calcs(iStatus, dMeasuredSpeed, dMeasuredPitch, dPitchDemand)

  ! -----
  IF (RESULT.GT.0) THEN

    RESULT = SetDemandedPitchAngle(turbine_id, 0, dPitchDemand)

    CONTROLLER = RESULT

    IF (RESULT.EQ.GH_DISCON_SUCCESS) THEN
      RESULT = ReportInfoMessage(turbine_id, "Set pitch"//CHAR(0))
      RETURN
    ENDIF
  ENDIF

  RESULT = ReportErrorMessage(turbine_id, "Failed to set pitch"//CHAR(0))
  RETURN
END
```

APPENDIX C GEARBOX DLL INTERFACE SPECIFICATION

Users with a licence for the Advanced Transmission Interface module may run **Bladed** simulations with user-defined gearbox dynamics provided in a Windows Dynamic Link Library (DLL). This appendix specifies the DLL interface in detail. Simple example DLL source codes in Fortran and C++ are provided in the Code Samples folder of your **Bladed** installation.

The gearbox dynamics in the DLL are defined in state-space form:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du_{in} \\ y_{out} &= C_{out}x + D_{out}u\end{aligned}$$

where x is a vector of state variables defined in the DLL, u and u_{in} are vectors of input variables defined below, y is a vector of output variables defined below, and y_{out} is a vector of output variables defined in the DLL. Matrices A , B , C , D , C_{out} and D_{out} represent terms defined in the DLL, although they need not be defined explicitly as matrices, and need not be constant: in other words the gearbox model may be non-linear and time-varying, and discontinuous changes are allowed. The **Bladed** integrator will integrate the states x ; note however that the number of states defined in the DLL may be zero, in which case the DLL may use its own integrator, whether in continuous or discrete time, to integrate any dynamic equations it contains – this is known as co-simulation. If this is done it is important to realise that any dynamic coupling to the **Bladed** equations of motion will be lost, and this should therefore be done only for dynamics which do not couple strongly with the dynamics of the turbine modelled by **Bladed**. It is also important to realise that the variable timestep integrator used by **Bladed** will make DLL calls at varying intervals of simulated time, and that the integrator trial steps, which result in call types 5 and 6 defined below, may step backwards in time as well as forwards, but will never step backwards further than the last completed integrator step which corresponds to call type 9 defined below. It is also possible to use a combination of “**Bladed**” and “co-simulation” states. Using the terms 'displacement' and 'velocity' as the last word in the state name for physical states that are part of a pair will help in the identification of states for the Campbell diagram.

For the DLL interface, all variables obey the GL sign convention, i.e. low speed shaft rotation is positive for a clockwise rotor. A negative gearbox ratio means that the high speed shaft rotates in the opposite direction to the low speed shaft. The low speed shaft torque is the torque exerted by the low speed shaft spring onto the gearbox input shaft. The high speed shaft torque is the torque exerted by the gearbox output shaft onto the high speed shaft spring (if the DLL represents just the gearbox) or by the generator rotor onto the generator stator (if the DLL represents the gearbox, high speed shaft and generator rotor inertia).

The DLL may be written in any language which can be compiled to a Windows DLL. The DLL entry point from **Bladed** is a single procedure called DLL_GBx (upper case) with eight arguments and no return value. Either the `__cdecl` or the `__stdcall` calling convention may be used. The procedure definition is as follows:

C.1 1 DOF Gearbox DLL Interface

Name	DLL_GBX		
Return value	None		
Arguments:	Data type	Array length	Description
Arg1	4-byte integer	10	See below
Arg2	8-byte real ⁺	1	Time (s) since start of simulation
Arg3	4-byte integer	Arg1(3)	See below
Arg4	1-byte char*	Arg1(4)	See below
Arg5	8-byte real ⁺	Arg1(5)	See below
Arg6	8-byte real ⁺	Arg1(6)	See below
Arg7	8-byte real ⁺	Arg1(7)	See below
Arg8	1-byte char*	Arg1(8)	Message returned by DLL

⁺ i.e. double precision float point

*declared as INTEGER*1 in FORTRAN.

All arguments are passed by reference, so for example Arg1 is a pointer to the start of an array of ten 4-byte integers, Arg2 is a pointer to a single 8-byte (double-precision) floating point number, and Arg4 is a pointer to an array of one-byte characters, the number of elements in the array being given by the fourth element of Arg1. The other elements of Arg1 are defined as follows:

Elements of Arg1 INTEGER array		
1	To DLL	DLL interface version number.
2	To DLL	Call type: 1 to 9.
3	To DLL	NI = Size of Arg3
4	To DLL	NC = Size of Arg4
5	To DLL	NR1 = Size of Arg5
6	To DLL	NR2 = Size of Arg6
7	To DLL	NR3 = Size of Arg7
8	To DLL	NC1 = Size of Arg8
9	From DLL	0 means OK, <0 means Abort
10	To DLL	Brake flag: 0 = shaft brake off, 1 = shaft brake 1 on, in general $\sum_i B_i * 2^{i-1}$ where B_i = state of shaft brake i (0=off, 1=on). May be used by the DLL if the Bladed shaft brake position is within the scope of the DLL.

The procedure is called in different ways by **Bladed**, depending on the Call Type given by element 2 of Arg1. Arg2 always contains the time in seconds since the start of the simulation. Arg3 to Arg7 depend on the value of the Call type, Arg1(2), as described below. Any arguments not mentioned are not used in that call, but must still be present.

If Arg1(9) is returned with a non-zero value an explanatory message (terminated by a null character) should be returned in Arg8.

The number of characters returned in Arg4 and Arg8 must never exceed NC and NC1 respectively.

Call type 1: Initialisation (Arg1(2) = 1)

This call initialises the DLL model (using any constants which might come from the **Bladed** model) and returns the required number of integrator states and output variables.

Elements of Arg3 INTEGER array (NI = 2):		
1	From DLL	Nx = No. of states to integrate
2	From DLL	Nout = No. of variables to output
Elements of Arg4 CHAR*1 array:		
1 to NC(max)	To DLL	Name of Parameter File and Verification File, each terminated by a semi-colon (;). If desired the DLL may open and read the Parameter File, which contains any user-defined data. The Verification File must be opened FOR APPENDING ONLY , and can be used to keep a record of the DLL parameters. This will be

		the \$VE file for the simulation, so it is important not to overwrite the data which it already contains. If opened, this file must be closed again before the DLL returns.
Elements of Arg5 REAL*8 array (NR1 = 1):		
1	To and from DLL	Nominal overall gearbox ratio between input and output shaft*. If the value passed by Bladed is not correct, the DLL should communicate this using Arg1(9) and Arg8, and substitute the correct value in the case that Arg1(9) is not set negative.

*This is a nominal ratio only, and is used, for example, to convert losses defined with respect to the low speed shaft if they are actually applied at the high speed shaft brake position. The actual overall ratio may be time-varying.

Call type 2: State definition (Arg1(2) = 2)

This call returns the names and absolute tolerances of the integrator states required by the DLL.

Elements of Arg3 INTEGER array (NI >= Nx+1):		
1	To DLL	Nx
2 to NX+1	From DLL	Auto-initialisation flag for each state: normally 0, but set to 1 if the state derivative must be zero in the steady state and you wish to include this state in the Bladed initial condition refinement step.
Elements of Arg4 CHAR*1 array:		
1 to NC(max)	From DLL	"State name:Units+;" for each state
Elements of Arg5 REAL*8 array (NR1 >= Nx):		
1 to Nx	From DLL	Absolute tolerances of the integrator states

+ Units are defined as for external controller output variables: see [Table A.2](#).

Call type 3: Output variable definition (Arg1(2) = 3)

This call returns the names and units of variables from the DLL to be output in time history output from **Bladed** (i.e. y_{out} from the second state-space equation $y_{out} = C_{out} \cdot x + D_{out} \cdot u$).

Elements of Arg3 INTEGER array (NI >= 1):		
1	To DLL	Nout
Elements of Arg4 CHAR*1 array:		
1 to NC(max)	From DLL	"Variable name:Units+;" for each output variable

+ Units are defined as for external controller output variables: see [Table A.2](#).

Call type 4: Initial conditions (Arg1(2) = 4)

This call executes one step of the initial conditions iteration, and also returns the initial values of the states (assuming everything is in the steady state).

Elements of Arg3 INTEGER array (NI >= Nx+4):		
1	To DLL	Nx
2	To DLL	NinInit = No. of variables to DLL = 5
3	To DLL	NoutInit = No. of variables from DLL = 3
4	-	<i>Reserved for future developments</i>
5 to Nx+4	From DLL	0 = State disabled; 1 = State enabled
Elements of Arg5 REAL*8 array (NR1 >= NinInit):		
1	To DLL	Low speed (input) shaft angular position (rad)
2	To DLL	High speed (output) shaft angular velocity (rad/s)
3	To DLL	High speed (output) shaft torque (Nm)
4	To DLL	Brake applied torque if applicable* (Nm)
5	To DLL	Loss torque if applicable* (Nm)
Elements of Arg6 REAL*8 array (NR2 >= NoutInit):		
1	From DLL	High speed (output) shaft angular position (rad)
2	From DLL	Low speed (input) shaft angular velocity (rad/s)
3	To/From DLL	Low speed (input) shaft torque (Nm)**
Elements of Arg7 REAL*8 array (NR3 >= Nx+10):		
1 to Nx	From DLL	Initial values of states if Arg3(4) = 1
Nx+1 to Nx+10	To/From DLL	User variables 1 to 10***
Nx+11 onwards	To/From DLL	Additional user variables

*May be used by the DLL if the **Bladed** shaft brake position is within the scope of the DLL; the applied brake torque is calculated (as a positive number) by **Bladed** but the DLL may substitute a different brake model; similarly with the loss torque, which is calculated with reference to the brake position.

Normally recalculated by the DLL; however in certain situations this cannot be done, in which case the value from **Bladed can be used. This might be the case if the gearbox rotation is initially stopped by brake friction for example.

***May be used to share information between user-defined DLLs for different turbine components

Call type 5: State derivatives (Arg1(2) = 5)

This call passes the current DLL state values and any required variables derived from **Bladed** states to the DLL, which returns the state derivatives (xdot) calculated by the DLL.

Elements of Arg3 INTEGER array (NI >= Nx+2):		
1	To DLL	Nx
2	To DLL	Nu = No. of input variables to DLL = 4
3 to Nx+2	To DLL	0 = State disabled; 1 = State enabled
Elements of Arg5 REAL*8 array (NR1 >= Nx+10):		
1 to Nx	To DLL	State variables
Nx+1 to Nx+10	To/From DLL	User variables 1 to 10
Nx+11 onwards	To/From DLL	Additional user variables
Elements of Arg6 REAL*8 array (NR2 >= Nu):		
1	To DLL	Low speed (input) shaft torque (Nm)
2	To DLL	High speed (output) shaft torque (Nm)
3	To DLL	Brake applied torque if applicable (Nm)
4	To DLL	Loss torque if applicable (Nm)
Elements of Arg7 REAL*8 array (NR3 >= Nx):		
1 to Nx	From DLL	State derivatives
Nx+1 to Nx+2	From DLL	LSS and HSS acceleration

Call type 6: Variables required by Bladed (Arg1(2) = 6)

This call passes the current DLL state values and any required variables derived from **Bladed** states to the DLL, which returns the values of variables y ($y = Cx + Du_{in}$) which are required by **Bladed**.

Elements of Arg3 INTEGER array (NI >= Nx+3):		
1	To DLL	Nx
2	To DLL	Nin = 2
3	To DLL	Ny = no. of variables required = 4
4 to Nx+3	To DLL	0 = State disabled; 1 = State enabled
Elements of Arg5 REAL*8 array (NR1 >= Nx+10):		
1 to Nx	To DLL	State variables
Nx+1 to Nx+10	To/From DLL	User variables 1 to 10
Nx+11 onwards	To/From DLL	Additional user variables
Elements of Arg6 REAL*8 array (NR2 >= Nin):		
1	To DLL	Brake applied torque if applicable (Nm)
2	To DLL	Loss torque if applicable (Nm)
Elements of Arg7 REAL*8 array (NR3 >= Ny):		
1	From DLL	Low speed (input) shaft angular position (rad)
2	From DLL	Low speed (input) shaft angular velocity (rad/s)
3	From DLL	High speed (output) shaft angular position (rad)
4	From DLL	High speed (output) shaft angular velocity (rad/s)

Call type 7: Output variables (Arg1(2) = 7)

This call passes the current DLL state values and any required variables derived from **Bladed** states to the DLL, which returns the values of the output variables (y_{out}) which will be presented in the simulation time-history output.

Elements of Arg3 INTEGER array (NI >= Nx+3):		
1	To DLL	Nx
2	To DLL	Nu = 4
3	To DLL	Nout
4 to Nx+3	To DLL	0 = State disabled; 1 = State enabled
Elements of Arg5 REAL*8 array (NR1 >= Nx+10):		
1 to Nx	To DLL	State variables
Nx+1 to Nx+10	To/From DLL	User variables 1 to 10
Nx+11 onwards	To/From DLL	Additional user variables
Elements of Arg6 REAL*8 array (NR2 >= Nu):		
1	To DLL	Low speed (input) shaft torque (Nm)
2	To DLL	High speed (output) shaft torque (Nm)
3	To DLL	Brake applied torque if applicable (Nm)
4	To DLL	Loss torque if applicable (Nm)
Elements of Arg7 REAL*8 array (NR3 >= Nout):		
1 to Nout	From DLL	Output variables

Call type 8: Discontinuity check (Arg1(2) = 8)

This call asks the DLL whether a discontinuity has been passed; if so, the DLL has the option of specifying a particular step reduction. These calls may be ignored by the DLL if it does not model any discontinuities.

Elements of Arg3 INTEGER array (NI >= Nx+3):		
1	To DLL	Nx
2	To DLL	Nu = 4
3	From DLL	0 means OK, <0 means need to step back
4 to Nx+3	To DLL	0 = State disabled; 1 = State enabled
Elements of Arg5 REAL*8 array (NR1 >= Nx+10):		
1 to Nx	To DLL	State variables
Nx+1 to Nx+10	To/From DLL	User variables 1 to 10
Nx+11 onwards	To/From DLL	Additional user variables
Elements of Arg6 REAL*8 array (NR2 >= Nu):		
1	To DLL	Low speed (input) shaft torque (Nm)
2	To DLL	High speed (output) shaft torque (Nm)
3	To DLL	Brake applied torque if applicable (Nm)
4	To DLL	Loss torque if applicable (Nm)
Elements of Arg7 REAL*8 array (NR3 >= 1):		
1	From DLL	Suggested time to step back to if Arg3(3) = -2

Call type 9: Completed step (Arg1(2) = 9)

This call tells the DLL that the timestep is complete, so that a discontinuous change to the dynamics can be implemented if required. These calls may be ignored by the DLL if it does not model any discontinuities.

Elements of Arg3 INTEGER array (NI >= Nx+2):		
1	To DLL	Nx
2	To DLL	Nu = 4
3 to Nx+2	To and from DLL	0 = State disabled; 1 = State enabled
Elements of Arg5 REAL*8 array (NR1 >= Nx+10):		
1 to Nx	To/from DLL	State variables
Nx+1 to Nx+10	To/From DLL	User variables 1 to 10
Nx+11 onwards	To/From DLL	Additional user variables
Elements of Arg6 REAL*8 array (NR2 >= Nu):		
1	To DLL	Low speed (input) shaft torque (Nm)
2	To DLL	High speed (output) shaft torque (Nm)
3	To DLL	Brake applied torque if applicable (Nm)
4	To DLL	Loss torque if applicable (Nm)

C.2 6 DOF Gearbox DLL Interface

Name	DLL_GBX		
Return value	None		
Arguments:	Data type	Array length	Description
Arg1	4-byte integer	10	See below
Arg2	8-byte real ⁺	1	Time (s) since start of simulation
Arg3	4-byte integer	Arg1(3)	See below
Arg4	1-byte char*	Arg1(4)	See below
Arg5	8-byte real ⁺	Arg1(5)	See below
Arg6	8-byte real ⁺	Arg1(6)	See below
Arg7	8-byte real ⁺	Arg1(7)	See below
Arg8	1-byte char*	Arg1(8)	Message returned by DLL

⁺ i.e. double precision float point

*declared as INTEGER*1 in FORTRAN.

All arguments are passed by reference, so for example Arg1 is a pointer to the start of an array of ten 4-byte integers, Arg2 is a pointer to a single 8-byte (double-precision) floating point number, and Arg4 is a pointer to an array of one-byte characters, the number of elements in the array being given by the fourth element of Arg1. The other elements of Arg1 are defined as follows:

Elements of Arg1 INTEGER array		
1	To DLL	DLL interface version number.
2	To DLL	Call type: 1 to 9.
3	To DLL	NI = Size of Arg3
4	To DLL	NC = Size of Arg4
5	To DLL	NR1 = Size of Arg5
6	To DLL	NR2 = Size of Arg6
7	To DLL	NR3 = Size of Arg7
8	To DLL	NC1 = Size of Arg8
9	From DLL	0 means OK, <0 means Abort
10	To DLL	Brake flag: 0 = shaft brake off, 1 = shaft brake 1 on, in general $\sum_i B_i * 2^{i-1}$ where B_i = state of shaft brake i (0=off, 1=on). May be used by the DLL if the Bladed shaft brake position is within the scope of the DLL.

The procedure is called in different ways by **Bladed**, depending on the Call Type given by element 2 of Arg1. Arg2 always contains the time in seconds since the start of the simulation. Arg3 to Arg7 depend on the value of the Call type, Arg1(2), as described below. Any arguments not mentioned are not used in that call, but must still be present.

If Arg1(9) is returned with a non-zero value an explanatory message (terminated by a null character) should be returned in Arg8.

The number of characters returned in Arg4 and Arg8 must never exceed NC and NC1 respectively.

Call type 1: Initialisation (Arg1(2) = 1)

This call initialises the DLL model (using any constants which might come from the **Bladed** model) and returns the required number of integrator states and output variables.

Elements of Arg3 INTEGER array (NI = 15):		
1	From DLL	Nx = No. of states to integrate
2	From DLL	Nout = No. of variables to output
3 to 15	-	Reserved for future interface enhancement
Elements of Arg4 CHAR*1 array:		

1 to NC(max)	To DLL	Name of Parameter File and Verification File, each terminated by a semi-colon (;). If desired the DLL may open and read the Parameter File, which contains any user-defined data. The Verification File must be opened FOR APPENDING ONLY , and can be used to keep a record of the DLL parameters. This will be the \$VE file for the simulation, so it is important not to overwrite the data which it already contains. If opened, this file must be closed again before the DLL returns.
--------------	--------	--

Elements of Arg5 REAL*8 array (NR1 = 1):

1	To/From DLL	Nominal overall gearbox ratio between input and output shaft*. If the value passed by Bladed is not correct, the DLL should communicate this using Arg1(9) and Arg8, and substitute the correct value in the case that Arg1(9) is not set negative.
---	-------------	--

*This is a nominal ratio only, and is used, for example, to convert losses defined with respect to the low speed shaft if they are actually applied at the high speed shaft brake position. The actual overall ratio may be time-varying.

Call type 2: State definition (Arg1(2) = 2)

This call returns the names and absolute tolerances of the integrator states required by the DLL.

Elements of Arg3 INTEGER array (NI >= Nx+1):

1	To DLL	Nx
2 to NX+1	From DLL	Auto-initialisation flag for each state: normally 0, but set to 1 if the state derivative must be zero in the steady state and you wish to include this state in the Bladed initial condition refinement step.

Elements of Arg4 CHAR*1 array:

1 to NC(max)	From DLL	"State name:Units+;" for each state
--------------	----------	-------------------------------------

Elements of Arg5 REAL*8 array (NR1 >= Nx):

1 to Nx	From DLL	Absolute tolerances of the integrator states
---------	----------	--

+ Units are defined as for external controller output variables: see [Table A.2](#).

Call type 3: Output variable definition (Arg1(2) = 3)

This call returns the names and units of variables from the DLL to be output in time history output from **Bladed** (i.e. y_{out} from the second state-space equation $y_{out} = C_{out} \cdot x + D_{out} \cdot u$).

Elements of Arg3 INTEGER array (NI >= 1):

1	To DLL	Nout
---	--------	------

Elements of Arg4 CHAR*1 array:

1 to NC(max)	From DLL	"Variable name:Units+;" for each output variable
--------------	----------	--

+ Units are defined as for external controller output variables: see [Table A.2](#).

Call type 4: Initial conditions (Arg1(2) = 4)

This call executes one step of the initial conditions iteration, and also returns the initial values of the states (assuming everything is in the steady state).

Elements of Arg3 INTEGER array (NI >= Nx+17):

1	To DLL	Nx
2	To DLL	NinInit = No. of variables to DLL = 17
3	To DLL	NoutInit = No. of variables from DLL = 15
4	To DLL	0 = trial call, 1 = final call
5 to 17	-	<i>Reserved for future interface enhancement</i>
18 to Nx+17	From DLL	0 = State disabled; 1 = State enabled

Elements of Arg5 REAL*8 array (NR1 >= NinInit):		
1	To DLL	θ orientation mounting (about x-axis) ¹
2	To DLL	ψ orientation mounting (about y-axis) ¹
3	To DLL	ϕ orientation mounting (about z-axis) ¹
4	To DLL	LSS force x
5	To DLL	LSS force y
6	To DLL	LSS force z
7	To/From DLL	LSS torque θ (rotation x-axis) ²
8	To DLL	LSS torque ψ (rotation y-axis)
9	To DLL	LSS torque ϕ (rotation z-axis)
10	To DLL	HSS torque θ (rotation x-axis)
11	To DLL	Mounting force x
12	To DLL	Mounting force y
13	To DLL	Mounting force z
14	To DLL	Mounting torque θ (rotation x-axis)
15	To DLL	Mounting torque ψ (rotation y-axis)
16	To DLL	Mounting torque ϕ (rotation z-axis)
17	To DLL	Brake applied torque if applicable (Nm) ³
18	To DLL	Loss torque if applicable (Nm) ³
19	To DLL	LSS angular position θ (rad) (rotation x-axis)
20	To DLL	HSS angular velocity θ (rad/s) (rotation x-axis)
Elements of Arg6 REAL*8 array (NR2 >= NoutInit):		
1	From DLL	HSS angular position θ (rad) (rotation x-axis)
2	From DLL	LSS angular velocity θ (rad/s) (rotation z-axis)
3	From DLL	LSS acceleration x ⁴
4	From DLL	LSS acceleration y ⁴
5	From DLL	LSS acceleration z ⁴
6	From DLL	LSS acceleration θ (rad/s ²) (rotation y-axis) ⁴
7	From DLL	LSS acceleration ψ (rad/s ²) (rotation y-axis) ⁴
8	From DLL	LSS acceleration ϕ (rad/s ²) (rotation z-axis) ⁴
9	From DLL	HSS acceleration θ (rad/s ²) (rotation y-axis)
10	From DLL	Mounting acceleration x (global coordinates)
11	From DLL	Mounting acceleration y (global coordinates)
12	From DLL	Mounting acceleration z (global coordinates)
13	From DLL	Mounting acceleration θ (rad/s ²) (rotation x-axis) (global coordinates)
14	From DLL	Mounting acceleration ψ (rad/s ²) (rotation y-axis) (global coordinates)
15	From DLL	Mounting acceleration ϕ (rad/s ²) (rotation z-axis) (global coordinates)
Elements of Arg7 REAL*8 array (NR3 >= Nx+10):		
1 to Nx	From DLL	Initial values of states if Arg3(4) = 1
Nx+1 to Nx+10	To/From DLL	User variables 1 to 10 ⁵
Nx+11 onwards	To/From DLL	Additional user variables

¹ Orientation of the mounting in relation to the tower reference frame. The rotations are carried out in the order, yaw (about z), nod (about y), roll (about x)

² Normally recalculated by the DLL; however in certain situations this cannot be done, in which case the value from **Bladed** can be used. This might be the case if the gearbox rotation is initially stopped by brake friction for example.

³ May be used by the DLL if the **Bladed** shaft brake position is within the scope of the DLL; the applied brake torque is calculated (as a positive number) by **Bladed** but the DLL may substitute a different brake model; similarly with the loss torque, which is calculated with reference to the brake position.

⁴ All LSS Acceleration is relative to the mounting acceleration

⁵ May be used to share information between user-defined DLLs for different turbine components

Call type 5: State derivatives (Arg1(2) = 5)

This call passes the current DLL state values and any required variables derived from **Bladed** states to the DLL, which returns the state derivatives (xdot) calculated by the DLL.

Elements of Arg3 INTEGER array (NI >= Nx+15):		
1	To DLL	Nx
2	To DLL	Nu = No. of input variables to DLL = 15
3 to 15	-	<i>Reserved for future interface enhancement</i>
16 to Nx+15	To DLL	0 = State disabled; 1 = State enabled
Elements of Arg5 REAL*8 array (NR1 >= Nx+10):		
1	To DLL	θ orientation mounting (about x-axis) ¹
2	To DLL	ψ orientation mounting (about y-axis) ¹
3	To DLL	ϕ orientation mounting (about z-axis) ¹
4 to Nx+3	To DLL	State variables
Nx+4 to Nx+13	To/From DLL	User variables 1 to 10
Nx+14 onwards	To/From DLL	Additional user variables
Elements of Arg6 REAL*8 array (NR2 >= Nu):		
1	To DLL	LSS force x
2	To DLL	LSS force y
3	To DLL	LSS force z
4	To DLL	LSS torque θ (rotation x-axis)
5	To DLL	LSS torque ψ (rotation y-axis)
6	To DLL	LSS torque ϕ (rotation z-axis)
7	To DLL	HSS torque θ (rotation x-axis)
8	To DLL	Mounting force x
9	To DLL	Mounting force y
10	To DLL	Mounting force z
11	To DLL	Mounting torque θ (rotation x-axis)
12	To DLL	Mounting torque ψ (rotation y-axis)
13	To DLL	Mounting torque ϕ (rotation z-axis)
14	To DLL	Brake applied torque if applicable (Nm)
15	To DLL	Loss torque if applicable (Nm)
Elements of Arg7 REAL*8 array (NR3 >= Nx+13):		
1	From DLL	LSS acceleration x
2	From DLL	LSS acceleration y
3	From DLL	LSS acceleration z
4	From DLL	LSS acceleration θ (rad/s ²) (rotation x-axis)
5	From DLL	LSS acceleration ψ (rad/s ²) (rotation y-axis)
6	From DLL	LSS acceleration ϕ (rad/s ²) (rotation z-axis)
7	From DLL	HSS acceleration θ (rad/s ²) (rotation x-axis)
8	From DLL	Mounting acceleration x(global coordinates)
9	From DLL	Mounting acceleration y(global coordinates)
10	From DLL	Mounting acceleration z(global coordinates)
11	From DLL	Mounting acceleration θ (rad/s ²) (rotation x-axis) (global coordinates)
12	From DLL	Mounting acceleration ψ (rad/s ²) (rotation y-axis) (global coordinates)
13	From DLL	Mounting acceleration ϕ (rad/s ²) (rotation z-axis) (global coordinates)
14 to Nx+13	From DLL	State derivatives
Nx+14 to Nx+ICRF+14	From DLL	Output referring to AutoICFlag whose approximation to zero has to be found (Could be state derivatives or User Defined variables)

¹ Orientation of the mounting in relation to the tower reference frame. The rotations are carried out in the order, yaw (about z), nod (about y), roll (about x)

Call type 6: Variables required by *Bladed* (Arg1(2) = 6)

This call passes the current DLL state values and any required variables derived from *Bladed* states to the DLL, which returns the values of variables y ($y = Cx + Du_{in}$) which are required by *Bladed*.

Elements of Arg3 INTEGER array (NI >= Nx+16):		
1	To DLL	Nx
2	To DLL	Nin = 2
3	To DLL	Ny = no. of variables required = 2
4 to 16	To DLL	<i>Reserved for future interface enhancement</i>
17 to Nx+16	To DLL	0 = State disabled; 1 = State enabled
Elements of Arg5 REAL*8 array (NR1 >= Nx+10):		
1	To DLL	θ orientation mounting (about x-axis)
2	To DLL	ψ orientation mounting (about y-axis)
3	To DLL	ϕ orientation mounting (about z-axis)
4 to Nx+3	To DLL	State variables
Nx+4 to Nx+13	To/From DLL	User variables 1 to 10
Nx+14 onwards	To/From DLL	Additional user variables
Elements of Arg6 REAL*8 array (NR2 >= Nin):		
1	To DLL	Brake applied torque if applicable (Nm)
2	To DLL	Loss torque if applicable (Nm)
Elements of Arg7 REAL*8 array (NR3 >= Ny):		
1	From DLL	High speed (output) shaft angular position (rad)
2	From DLL	High speed (output) shaft angular velocity (rad/s)

Call type 7: Output variables (Arg1(2) = 7)

This call passes the current DLL state values and any required variables derived from *Bladed* states to the DLL, which returns the values of the output variables (y_{out}) which will be presented in the simulation time-history output.

Elements of Arg3 INTEGER array (NI >= Nx+16):		
1	To DLL	Nx
2	To DLL	Nu = 15
3	From DLL	Nout
4 to 16	To DLL	<i>Reserved for future interface enhancement</i>
17 to Nx+16	To DLL	0 = State disabled; 1 = State enabled
Elements of Arg5 REAL*8 array (NR1 >= Nx+10):		
1	To DLL	θ orientation mounting (about x-axis)
2	To DLL	ψ orientation mounting (about y-axis)
3	To DLL	ϕ orientation mounting (about z-axis)
4 to Nx+3	To DLL	State variables
Nx+4 to Nx+13	To/From DLL	User variables 1 to 10
Nx+14 onwards	To/From DLL	Additional user variables
Elements of Arg6 REAL*8 array (NR2 >= Nu):		
1 to 3	To DLL	LSS forces x, y, z
4 to 6	To DLL	LSS torques θ, ψ, ϕ
7	To DLL	HSS torque θ
8 to 10	To DLL	Mounting forces x, y, z
11 to 13	To DLL	Mounting torques θ, ψ, ϕ
14	To DLL	Brake applied torque if applicable (Nm)
15	To DLL	Loss torque if applicable (Nm)

Elements of Arg7 REAL*8 array (NR3 >= Nout):		
1 to Nout	From DLL	Output variables

Call type 8: Discontinuity check (Arg1(2) = 8)

This call asks the DLL whether a discontinuity has been passed; if so, the DLL has the option of specifying a particular step reduction. These calls may be ignored by the DLL if it does not model any discontinuities.

Elements of Arg3 INTEGER array (NI >= Nx+16):		
1	To DLL	Nx
2	To DLL	Nu = 15
3	To DLL	0 means OK, <0 means need to step back
4 to 16	-	<i>Reserved for future interface enhancement</i>
17 to Nx+16	To DLL	0 = State disabled; 1 = State enabled

Elements of Arg5 REAL*8 array (NR1 >= Nx+10):		
1	To DLL	θ orientation mounting (about x-axis)
2	To DLL	ψ orientation mounting (about y-axis)
3	To DLL	ϕ orientation mounting (about z-axis)
4 to Nx+3	To DLL	State variables
Nx+4 to Nx+13	To/From DLL	User variables 1 to 10
Nx+14 onwards	To/From DLL	Additional user variables

Elements of Arg6 REAL*8 array (NR2 >= Nu):		
1 to 3	To DLL	LSS forces x, y, z
4 to 6	To DLL	LSS torques θ, ψ, ϕ
7	To DLL	HSS torque θ
8 to 10	To DLL	Mounting forces x, y, z
11 to 13	To DLL	Mounting torques θ, ψ, ϕ
14	To DLL	Brake applied torque if applicable (Nm)
15	To DLL	Loss torque if applicable (Nm)

Elements of Arg7 REAL*8 array (NR3 >= 1):		
1	From DLL	Suggested time to step back to if Arg3(3) = -2

Call type 9: Completed step (Arg1(2) = 9)

This call tells the DLL that the timestep is complete, so that a discontinuous change to the dynamics can be implemented if required. These calls may be ignored by the DLL if it does not model any discontinuities.

Elements of Arg3 INTEGER array (NI >= Nx+15):		
1	To DLL	Nx
2	To DLL	Nu = 15
3 to 15	-	<i>Reserved for future interface enhancement</i>
16 to Nx+15	To DLL	0 = State disabled; 1 = State enabled

Elements of Arg5 REAL*8 array (NR1 >= Nx+10):		
1	To DLL	θ orientation mounting (about x-axis)
2	To DLL	ψ orientation mounting (about y-axis)
3	To DLL	ϕ orientation mounting (about z-axis)
4 to Nx+3	To DLL	State variables
Nx+4 to Nx+13	To/From DLL	User variables 1 to 10
Nx+14 onwards	To/From DLL	Additional user variables

Elements of Arg6 REAL*8 array (NR2 >= Nu):		
1 to 3	To DLL	LSS forces x, y, z

4 to 6	To DLL	LSS torques θ, ψ, ϕ
7	To DLL	HSS torque θ
8 to 10	To DLL	Mounting forces x, y, z
11 to 13	To DLL	Mounting torques θ, ψ, ϕ
14	To DLL	Brake applied torque if applicable (Nm)
15	To DLL	Loss torque if applicable (Nm)

APPENDIX D GENERATOR DLL INTERFACE SPECIFICATION

Users with a licence for the Advanced Transmission Interface module may run **Bladed** simulations with a user-defined dynamic model of the electrical system provided in a Windows Dynamic Link Library (DLL). This appendix specifies the DLL interface in detail. Simple example DLL source codes in C++ is provided in the Code Samples folder of your **Bladed** installation.

The dynamics in the DLL may be expressed in state-space form:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du_{in} \\ y_{out} &= C_{out}x + D_{out}u\end{aligned}$$

where x is a vector of state variables defined in the DLL, u and u_{in} are vectors of input variables defined below, y is a vector of output variables defined below, and y_{out} is a vector of output variables defined in the DLL. Matrices A , B , C , D , C_{out} and D_{out} represent terms defined in the DLL, although they need not be defined explicitly as matrices, and need not be constant: in other words the electrical dynamics model may be non-linear and time-varying, and discontinuous changes are allowed. The **Bladed** integrator will integrate the states x ; note however that the number of states defined in the DLL may be zero, in which case the DLL may use its own integrator, whether in continuous or discrete time, to integrate any dynamic equations it contains – this is known as co-simulation. If this is done it is important to realise that any dynamic coupling to the **Bladed** equations of motion will be lost, and this should therefore be done only for dynamics which do not couple strongly with the dynamics of the turbine modelled by **Bladed**. It is also important to realise that the variable timestep integrator used by **Bladed** will make DLL calls at varying intervals of simulated time, and that the integrator trial steps, which result in call types 5 and 6 defined below, may step backwards in time as well as forwards, but will never step backwards further than the last completed integrator step which corresponds to call type 9 defined below. It is also possible to use a combination of “**Bladed**” and “co-simulation” states. Using the terms ‘displacement’ and ‘velocity’ as the last word in the state name for physical states that are part of a pair will help in the identification of states for the Campbell diagram.

The DLL may be written in any language which can be compiled to a Windows DLL. The DLL entry point from **Bladed** is a single procedure called DLL_GENER (upper case) with eight arguments and no return value. Either the `__cdecl` or the `__stdcall` calling convention may be used. The procedure definition is as follows:

All variables exchanged between **Bladed** and the DLL are in strict SI units.

Name	DLL_GENER		
Return value	None		
Arguments:	Data type	Array length	Description
Arg1	4-byte integer	11	See below
Arg2	8-byte real ⁺	1	Time (s) since start of simulation
Arg3	4-byte integer	Arg1(3)	See below
Arg4	1-byte char*	Arg1(4)	See below
Arg5	8-byte real ⁺	Arg1(5)	See below
Arg6	8-byte real ⁺	Arg1(6)	See below
Arg7	8-byte real ⁺	Arg1(7)	See below
Arg8	1-byte char*	Arg1(8)	Message returned by DLL

⁺ i.e. double precision float point

*declared as INTEGER*1 in FORTRAN.

All arguments are passed by reference, so for example Arg1 is a pointer to the start of an array of ten 4-byte integers, Arg2 is a pointer to a single 8-byte (double-precision) floating point number, and Arg4 is a pointer to an array of one-byte characters, the number of elements in the array being given by the fourth element of Arg1. The other elements of Arg1 are defined as follows:

Elements of Arg1 INTEGER array		
1	To DLL	0 = DLL interface version number (for future use).
2	To DLL	Call type: 1 to 9.
3	To DLL	NI = Size of Arg3

4	To DLL	NC = Size of Arg4
5	To DLL	NR1 = Size of Arg5
6	To DLL	NR2 = Size of Arg6
7	To DLL	NR3 = Size of Arg7
8	To DLL	NC1 = Size of Arg8
9	From DLL	0 means OK, <0 means Abort
10	To DLL	Grid status: 0 = off, 1 = on.

The procedure is called in different ways by **Bladed**, depending on the Call Type given by element 2 of Arg1. Arg2 always contains the time in seconds since the start of the simulation. Arg3 to Arg7 depend on the value of the Call type, Arg1(2), as described below. Any arguments not mentioned are not used in that call, but must still be present.

If Arg1(9) is returned with a non-zero value an explanatory message (terminated by a null character) should be returned in Arg8.

The number of characters returned in Arg4 and Arg8 must never exceed NC and NC1 respectively.

Call type 1: Initialisation (Arg1(2) = 1)

This call initialises the DLL model (using any constants which might come from the **Bladed** model) and returns the required number of integrator states and output variables.

Elements of Arg3 INTEGER array (NI = 2):		
1	From DLL	Nx = No. of states to integrate
2	From DLL	Nout = No. of variables to output
Elements of Arg4 CHAR*1 array:		
1 to NC(max)	To DLL	Name of Parameter File and Verification File, each terminated by a semi-colon (;). If desired the DLL may open and read the Parameter File, which contains any user-defined data. The Verification File must be opened FOR APPENDING ONLY , and can be used to keep a record of the DLL parameters. This will be the \$VE file for the simulation, so it is important not to overwrite the data which it already contains. If opened, this file must be closed again before the DLL returns.
Elements of Arg5 REAL*8 array (NR1 = 1):		
1	To DLL	Model type: For future use, currently set to 1

Call type 2: State definition (Arg1(2) = 2)

This call returns the names and absolute tolerances of the integrator states required by the DLL.

Elements of Arg3 INTEGER array (NI >= Nx+1):		
1	To DLL	Nx
2 to NX+1	From DLL	Auto-initialisation flag for each state: normally 0, but set to 1 if the state derivative must be zero in the steady state and you wish to include this state in the Bladed initial condition refinement step.
Elements of Arg4 CHAR*1 array:		
1 to NC(max)	From DLL	"State name:Units+;" for each state
Elements of Arg5 REAL*8 array (NR1 >= Nx):		
1 to Nx	From DLL	Absolute tolerances of the integrator states

+ Units are defined as for external controller output variables: see [Table A.2](#).

Call type 3: Output variable definition (Arg1(2) = 3)

This call returns the names and units of variables from the DLL to be output in time history output from **Bladed** (i.e. y_{out} from the second state-space equation $y_{out} = C_{out} \cdot x + D_{out} \cdot u$).

Elements of Arg3 INTEGER array (NI >= 1):		
1	To DLL	Nout
Elements of Arg4 CHAR*1 array:		
1 to NC(max)	From DLL	"Variable name:Units ⁺ ;" for each output variable

⁺ Units are defined as for external controller output variables: see [Table A.2](#).

Call type 4: Initial conditions (Arg1(2) = 4)

This call executes one step of the initial conditions iteration, and also returns the initial values of the states (assuming everything is in the steady state).

Elements of Arg3 INTEGER array (NI >= Nx+4):		
1	To DLL	Nx
2	To DLL	NinInit = No. of variables to DLL = 5
3	To DLL	NoutInit = No. of variables from DLL = 2
4	To DLL	0 = trial call, 1 = final call
5 to Nx+4	From DLL	0 = State disabled; 1 = State enabled
Elements of Arg5 REAL*8 array (NR1 >= NinInit):		
1	To DLL	Generator speed (rad/s)
2	To DLL	Generator angular position (rad)
3	To DLL	Torque demand (Nm)
4	To DLL	Network voltage disturbance (fraction of nominal)
5	To DLL	Network frequency disturbance (fraction of nominal)
Elements of Arg6 REAL*8 array (NR2 >= NoutInit):		
1	From DLL	Air-gap torque (Nm)
2	From DLL	Electrical power output to sensor (W)
Elements of Arg7 REAL*8 array (NR3 >= Nx+10):		
1 to Nx	From DLL	Initial values of states if Arg3(4) = 1
Nx+1 to Nx+10	To/From DLL	User variables 1 to 10***
Nx+11 onwards	To/From DLL	Additional user variables

***May be used to share information between user-defined DLLs for different turbine components

Call type 5: State derivatives (Arg1(2) = 5)

This call passes the current DLL state values and any required variables derived from **Bladed** states to the DLL, which returns the state derivatives (x_{dot}) calculated by the DLL.

Elements of Arg3 INTEGER array (NI ≥ Nx+2):		
1	To DLL	Nx
2	To DLL	Nu = No. of input variables to DLL = 5
3 to Nx+2	To DLL	0 = State disabled; 1 = State enabled
Elements of Arg5 REAL*8 array (NR1 ≥ Nx+10):		
1 to Nx	To DLL	State variables
Nx+1 to Nx+10	To/From DLL	User variables 1 to 10
Nx+11 onwards	To/From DLL	Additional user variables
Elements of Arg6 REAL*8 array (NR2 ≥ Nu):		
1	To DLL	Generator speed (rad/s)
2	To DLL	Generator angular position (rad)
3	To DLL	Torque demand (Nm)
4	To DLL	Network voltage disturbance (fraction of nominal)
5	To DLL	Network frequency disturbance (fraction of nominal)
Elements of Arg7 REAL*8 array (NR3 ≥ Nx):		
1 to Nx	From DLL	State derivatives

Call type 6: Variables required by Bladed (Arg1(2) = 6)

This call passes the current DLL state values and any required variables derived from **Bladed** states to the DLL, which returns the values of variables y (y = Cx + Du_{in}) which are required by **Bladed**.

Elements of Arg3 INTEGER array (NI ≥ Nx+3):		
1	To DLL	Nx
2	To DLL	Nin = 5
3	To DLL	Ny = no. of variables required = 2
4 to Nx+3	To DLL	0 = State disabled; 1 = State enabled
Elements of Arg5 REAL*8 array (NR1 ≥ Nx+10):		
1 to Nx	To DLL	State variables
Nx+1 to Nx+10	To/From DLL	User variables 1 to 10
Nx+11 onwards	To/From DLL	Additional user variables
Elements of Arg6 REAL*8 array (NR2 ≥ Nin):		
1	To DLL	Generator speed (rad/s)
2	To DLL	Generator angular position (rad)
3	To DLL	Torque demand (Nm)
4	To DLL	Network voltage disturbance (fraction of nominal)
5	To DLL	Network frequency disturbance (fraction of nominal)
Elements of Arg7 REAL*8 array (NR3 ≥ Ny):		
1	From DLL	Air-gap torque (Nm)
2	From DLL	Electrical power output to sensor (W)

Call type 7: Output variables (Arg1(2) = 7)

This call passes the current DLL state values and any required variables derived from **Bladed** states to the DLL, which returns the values of the output variables (y_{out}) which will be presented in the simulation time-history output.

Elements of Arg3 INTEGER array (NI >= Nx+3):		
1	To DLL	Nx
2	To DLL	Nu = 5
3	To DLL	Nout
4 to Nx+3	To DLL	0 = State disabled; 1 = State enabled
Elements of Arg5 REAL*8 array (NR1 >= Nx+10):		
1 to Nx	To DLL	State variables
Nx+1 to Nx+10	To/From DLL	User variables 1 to 10
Nx+11 onwards	To/From DLL	Additional user variables
Elements of Arg6 REAL*8 array (NR2 >= Nu):		
1	To DLL	Generator speed (rad/s)
2	To DLL	Generator angular position (rad)
3	To DLL	Torque demand (Nm)
4	To DLL	Network voltage disturbance (fraction of nominal)
5	To DLL	Network frequency disturbance (fraction of nominal)
Elements of Arg7 REAL*8 array (NR3 >= Nout):		
1 to Nout	From DLL	Output variables

Call type 8: Discontinuity check (Arg1(2) = 8)

This call asks the DLL whether a discontinuity has been passed; if so, the DLL has the option of specifying a particular step reduction. These calls may be ignored by the DLL if it does not model any discontinuities.

Elements of Arg3 INTEGER array (NI >= Nx+3):		
1	To DLL	Nx
2	To DLL	Nu = 5
3	From DLL	0 means OK, <0 means need to step back
4 to Nx+3	To DLL	0 = State disabled; 1 = State enabled
Elements of Arg5 REAL*8 array (NR1 >= Nx+10):		
1 to Nx	To DLL	State variables
Nx+1 to Nx+10	To/From DLL	User variables 1 to 10
Nx+11 onwards	To/From DLL	Additional user variables
Elements of Arg6 REAL*8 array (NR2 >= Nu):		
1	To DLL	Generator speed (rad/s)
2	To DLL	Generator angular position (rad)
3	To DLL	Torque demand (Nm)
4	To DLL	Network voltage disturbance (fraction of nominal)
5	To DLL	Network frequency disturbance (fraction of nominal)
Elements of Arg7 REAL*8 array (NR3 >= 1):		
1	From DLL	Suggested time to step back to if Arg3(3) = -2

Call type 9: Completed step (Arg1(2) = 9)

This call tells the DLL that the timestep is complete, so that a discontinuous change to the dynamics can be implemented if required. These calls may be ignored by the DLL if it does not model any discontinuities.

Elements of Arg3 INTEGER array (NI >= Nx+2):		
--	--	--

1	To DLL	Nx
2	To DLL	Nu = 5
3 to Nx+2	To and from DLL	0 = State disabled; 1 = State enabled
Elements of Arg5 REAL*8 array (NR1 >= Nx+10):		
1 to Nx	To/from DLL	State variables
Nx+1 to Nx+10	To/From DLL	User variables 1 to 10
Nx+11 onwards	To/From DLL	Additional user variables
Elements of Arg6 REAL*8 array (NR2 >= Nu):		
1	To DLL	Generator speed (rad/s)
2	To DLL	Generator angular position (rad)
3	To DLL	Torque demand (Nm)
4	To DLL	Network voltage disturbance (fraction of nominal)
5	To DLL	Network frequency disturbance (fraction of nominal)

APPENDIX E PITCH ACTUATOR DLL INTERFACE SPECIFICATION

Users with a licence for the Advanced Pitch actuator interface module may run **Bladed** simulations with user-defined pitch actuator dynamics provided in a Windows Dynamic Link Library (DLL). There are two DLL interfaces available, a continuous time interface and a discrete time interface. As the feedback loops of pitch actuator dynamics work on a very short time-scale, the continuous time interface is recommended for performance reasons. This appendix specifies the DLL interfaces in detail.

For the DLL interfaces, all pitch variables hold the convention of positive from fine to feather, with pitch angle defined to be zero at fine pitch.

E.1 Continuous DLL interface

The pitch actuator dynamics in the DLL are defined in state-space form:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du_{in} \\ y_{out} &= C_{out}x + D_{out}u\end{aligned}$$

where x is a vector of state variables defined in the DLL, u and u_{in} are vectors of input variables defined below, y is a vector of output variables defined below, and y_{out} is a vector of output variables defined in the DLL. Matrices A , B , C , D , C_{out} and D_{out} represent terms defined in the DLL, although they need not be defined explicitly as matrices, and need not be constant: in other words the pitch actuator model may be non-linear and time-varying, and discontinuous changes are allowed. The **Bladed** integrator will integrate the states x ; note however that the number of states defined in the DLL may be zero, in which case the DLL may use its own integrator, whether in continuous or discrete time, to integrate any dynamic equations it contains – this is known as co-simulation. If this is done it is important to realise that any dynamic coupling to the **Bladed** equations of motion will be lost, and this should therefore be done only for dynamics which do not couple strongly with the dynamics of the turbine modelled by **Bladed**. It is also important to realise that the variable timestep integrator used by **Bladed** will make DLL calls at varying intervals of simulated time, and that the integrator trial steps, which result in call types 5 and 6 defined below, may step backwards in time as well as forwards, but will never step backwards further than the last completed integrator step which corresponds to call type 9 defined below. It is also possible to use a combination of “**Bladed**” and “co-simulation” states. Using the terms ‘displacement’ and ‘velocity’ as the last word in the state name for physical states that are part of a pair will help in the identification of states for the Campbell diagram.

The DLL may be written in any language which can be compiled to a Windows DLL. The DLL entry point from **Bladed** is a single procedure called DLL_PITCH (upper case) with eight arguments and no return value. Either the `__cdecl` or the `__stdcall` calling convention may be used. The procedure definition is as follows:

Name	DLL_PITCH		
Return value	0 means OK, <0 means Abort		
Arguments:	Data type	Array length	Description
Arg1	4-byte integer	11	See below
Arg2	8-byte real ⁺	1	Time (s) since start of simulation
Arg3	4-byte integer	Arg1(3)	See below
Arg4	1-byte char	Arg1(4)	See below
Arg5	8-byte real ⁺	Arg1(5)	See below
Arg6	8-byte real ⁺	Arg1(6)	See below
Arg7	8-byte real ⁺	Arg1(7)	See below
Arg8	1-byte char	Arg1(8)	Message returned by DLL

⁺ i.e. double precision float point

All arguments are passed by reference, so for example Arg1 is a pointer to the start of an array of ten 4-byte integers, Arg2 is a pointer to a single 8-byte (double-precision) floating point number, and Arg4 is a pointer to an array of one-byte characters, the number of elements in the array being given by the fourth element of Arg1. The other elements of Arg1 are defined as follows:

Elements of Arg1 INTEGER array		
1	To DLL	DLL interface version number = 4301
2	To DLL	Call type: 1 to 9.
3	To DLL	NI = Size of Arg3
4	To DLL	NC = Size of Arg4
5	To DLL	NR1 = Size of Arg5
6	To DLL	NR2 = Size of Arg6
7	To DLL	NR3 = Size of Arg7
8	To DLL	NC1 = Size of Arg8
9	To DLL	Blade number. So that blades may be modelled individually. The DLL is called for each blade on every call type.

The procedure is called in different ways by **Bladed**, depending on the Call Type given by element 2 of Arg1. Arg2 always contains the time in seconds since the start of the simulation. Arg3 to Arg7 depend on the value of the Call type, Arg1(2), as described below. Any arguments not mentioned are not used in that call, but must still be present.

If the return value is returned with a non-zero value an explanatory message (terminated by a null character) should be returned in Arg8.

The number of characters returned in Arg4 and Arg8 must never exceed NC and NC1 respectively.

Call type 1: Initialisation (Arg1(2) = 1)

This call initialises the DLL model (using any constants which might come from the **Bladed** model) and returns the required number of integrator states and output variables.

Elements of Arg3 INTEGER array (NI = 2):		
1	From DLL	Nx = No. of states to integrate
2	From DLL	Nout = No. of variables to output
3	To DLL	Input type: 0 = position demand, 1 = rate demand.
4	From DLL	Output type: 0 = pitch rate (rad/s), 1 = pitch acceleration (rad/s ²), 2 = actuator torque (Nm).
Elements of Arg4 CHAR*1 array:		
1 to NC(max)	To DLL	Name of Parameter File and Verification File, each terminated by a semi-colon (;). If desired the DLL may open and read

		the Parameter File, which contains any user-defined data. The Verification File must be opened FOR APPENDING ONLY , and can be used to keep a record of the DLL parameters. This will be the \$VE file for the simulation, so it is important not to overwrite the data which it already contains. If opened, this file must be closed again before the DLL returns.
--	--	--

Call type 2: State definition (Arg1(2) = 2)

This call returns the names and absolute tolerances of the integrator states required by the DLL.

Elements of Arg3 INTEGER array (NI >= Nx+1):		
1	To DLL	Nx
2 to NX+1	From DLL	Auto-initialisation flag for each state: normally 0, but set to 1 if the state derivative must be zero in the steady state and you wish to include this state in the Bladed initial condition refinement step.
Elements of Arg4 CHAR*1 array:		
1 to NC(max)	From DLL	"State name:Units+;" for each state
Elements of Arg5 REAL*8 array (NR1 >= Nx):		
1 to Nx	From DLL	Absolute tolerances of the integrator states

+ Units are defined as for external controller output variables: see [A.5 Sending logging output to](#)

Call type 3: Output variable definition (Arg1(2) = 3)

This call returns the names and units of variables from the DLL to be output in time history output from **Bladed** (i.e. y_{out} from the second state-space equation $y_{out} = C_{out} \cdot X + D_{out} \cdot u$).

Elements of Arg3 INTEGER array (NI >= 1):		
1	To DLL	Nout
Elements of Arg4 CHAR*1 array:		
1 to NC(max)	From DLL	"Variable name:Units+;" for each output variable

+ Units are defined as for external controller output variables: see [A.5 Sending logging output to](#)

Call type 4: Initial conditions (Arg1(2) = 4)

This call executes one step of the initial conditions iteration, and also returns the initial values of the states (assuming everything is in the steady state).

Elements of Arg3 INTEGER array (NI >= Nx+4):		
1	To DLL	Nx
2	To DLL	NinInit = No. of variables to DLL = 7
3	To DLL	NoutInit = No. of variables from DLL = 1
4	To DLL	0 = trial call, 1 = final call
5 to Nx+4	From DLL	0 = State disabled; 1 = State enabled
Elements of Arg5 REAL*8 array (NR1 >= NinInit):		
1	To DLL	Pitch position/velocity demand (rad)/(rad/s)*
2	To DLL	Pitch angular displacement (rad)
3	To DLL	Pitch angular velocity (rad/s)
4	To DLL	Pitch Bearing F_x (N)
5	To DLL	Pitch Bearing F_y (N)
6	To DLL	Pitch Bearing F_z (N)

7	To DLL	Pitch Bearing M_x (Nm)
8	To DLL	Pitch Bearing M_y (Nm)
9	To DLL	Pitch Bearing M_z (Nm)
10	To DLL	Pitching inertia (kgm^2)
11	To DLL	Pitching friction (Nm)
12	To DLL	Pitching stiction (Nm)
Elements of Arg6 REAL*8 array (NR2 >= NoutInit):		
1	From DLL	Pitch output variable according to output type
Elements of Arg7 REAL*8 array (NR3 >= Nx+10):		
1 to Nx	From DLL	Initial values of states if Arg3(4) = 1
Nx+1 to Nx+10	To/From DLL	User variables 1 to 10**

* Depending on whether control system implements a position or rate demand

**May be used to share information between user-defined DLLs for different turbine components

Call type 5: State derivatives (Arg1(2) = 5)

This call passes the current DLL state values and any required variables derived from **Bladed** states to the DLL, which returns the state derivatives (xdot) calculated by the DLL.

Elements of Arg3 INTEGER array (NI >= Nx+2):		
1	To DLL	Nx
2	To DLL	Nu = No. of input variables to DLL = 7
3 to Nx+2	To DLL	0 = State disabled; 1 = State enabled
Elements of Arg5 REAL*8 array (NR1 >= Nx+10):		
1 to Nx	To DLL	State variables
Nx+1 to Nx+10	To/From DLL	User variables 1 to 10
Elements of Arg6 REAL*8 array (NR2 >= Nu):		
1	To DLL	Pitch position/velocity demand (rad)/(rad/s)*
2	To DLL	Pitch angular displacement (rad)
3	To DLL	Pitch angular velocity (rad/s)
4	To DLL	Pitch Bearing F_x (N)
5	To DLL	Pitch Bearing F_y (N)
6	To DLL	Pitch Bearing F_z (N)
7	To DLL	Pitch Bearing M_x (Nm)
8	To DLL	Pitch Bearing M_y (Nm)
9	To DLL	Pitch Bearing M_z (Nm)
10	To DLL	Pitching inertia (kgm^2)
11	To DLL	Pitching friction (Nm)
12	To DLL	Pitching stiction (Nm)
Elements of Arg7 REAL*8 array (NR3 >= Nx):		
1 to Nx	From DLL	State derivatives

* Depending on whether control system implements a position or rate demand

Call type 6: Variables required by Bladed (Arg1(2) = 6)

This call passes the current DLL state values and any required variables derived from **Bladed** states to the DLL, which returns the values of variables y ($y = Cx + Du_{in}$) which are required by **Bladed**.

Elements of Arg3 INTEGER array (NI >= Nx+3):		
--	--	--

1	To DLL	Nx
2	To DLL	Nin = 7
3	To DLL	Ny = no. of variables required = 1
4 to Nx+3	To DLL	0 = State disabled; 1 = State enabled
Elements of Arg5 REAL*8 array (NR1 >= Nx+10):		
1 to Nx	To DLL	State variables
Nx+1 to Nx+10	To/From DLL	User variables 1 to 10
Elements of Arg6 REAL*8 array (NR2 >= Nin):		
1	To DLL	Pitch position/velocity demand (rad)/(rad/s)*
2	To DLL	Pitch angular displacement (rad)
3	To DLL	Pitch angular velocity (rad/s)
4	To DLL	Pitch Bearing F _x (N)
5	To DLL	Pitch Bearing F _y (N)
6	To DLL	Pitch Bearing F _z (N)
7	To DLL	Pitch Bearing M _x (Nm)
8	To DLL	Pitch Bearing M _y (Nm)
9	To DLL	Pitch Bearing M _z (Nm)
10	To DLL	Pitching inertia (kgm ²)
11	To DLL	Pitching friction (Nm)
12	To DLL	Pitching stiction (Nm)
Elements of Arg7 REAL*8 array (NR3 >= Ny):		
1	From DLL	Pitch output variable according to output type

* Depending on whether control system implements a position or rate demand

Call type 7: Output variables (Arg1(2) = 7)

This call passes the current DLL state values and any required variables derived from **Bladed** states to the DLL, which returns the values of the output variables (y_{out}) which will be presented in the simulation time-history output.

Elements of Arg3 INTEGER array (NI >= Nx+3):		
1	To DLL	Nx
2	To DLL	Nu = 7
3	To DLL	Nout
4 to Nx+3	To DLL	0 = State disabled; 1 = State enabled
Elements of Arg5 REAL*8 array (NR1 >= Nx+10):		
1 to Nx	To DLL	State variables
Nx+1 to Nx+10	To/From DLL	User variables 1 to 10
Elements of Arg6 REAL*8 array (NR2 >= Nu):		
1	To DLL	Pitch position/velocity demand (rad)/(rad/s)*
2	To DLL	Pitch angular displacement (rad)
3	To DLL	Pitch angular velocity (rad/s)
4	To DLL	Pitch Bearing F _x (N)
5	To DLL	Pitch Bearing F _y (N)
6	To DLL	Pitch Bearing F _z (N)

7	To DLL	Pitch Bearing M_x (Nm)
8	To DLL	Pitch Bearing M_y (Nm)
9	To DLL	Pitch Bearing M_z (Nm)
10	To DLL	Pitching inertia (kgm ²)
11	To DLL	Pitching friction (Nm)
12	To DLL	Pitching stiction (Nm)
Elements of Arg7 REAL*8 array (NR3 >= Nout):		
1 to Nout	From DLL	Output variables

* Depending on whether control system implements a position or rate demand

Call type 8: Discontinuity check (Arg1(2) = 8)

This call asks the DLL whether a discontinuity has been passed; if so, the DLL has the option of specifying a particular step reduction. These calls may be ignored by the DLL if it does not model any discontinuities.

Elements of Arg3 INTEGER array (NI >= Nx+4):		
1	To DLL	N_x
2	To DLL	$N_u = 7$
3	From DLL	0 means OK, <0 means need to step back
4 to N_x+3	To DLL	0 = State disabled; 1 = State enabled

Elements of Arg5 REAL*8 array (NR1 >= Nx+10):		
1 to N_x	To DLL	State variables
N_x+1 to N_x+10	To/From DLL	User variables 1 to 10

Elements of Arg6 REAL*8 array (NR2 >= Nu):		
1	To DLL	Pitch position/velocity demand (rad)/(rad/s)*
2	To DLL	Pitch angular displacement (rad)
3	To DLL	Pitch angular velocity (rad/s)
4	To DLL	Pitch Bearing F_x (N)
5	To DLL	Pitch Bearing F_y (N)
6	To DLL	Pitch Bearing F_z (N)
7	To DLL	Pitch Bearing M_x (Nm)
8	To DLL	Pitch Bearing M_y (Nm)
9	To DLL	Pitch Bearing M_z (Nm)
10	To DLL	Pitching inertia (kgm ²)
11	To DLL	Pitching friction (Nm)
12	To DLL	Pitching stiction (Nm)

Elements of Arg7 REAL*8 array (NR3 >= 1):		
1	From DLL	Suggested time to step back to if Arg3(3) = -2

* Depending on whether control system implements a position or rate demand

Call type 9: Completed step (Arg1(2) = 9)

This call tells the DLL that the timestep is complete, so that a discontinuous change to the dynamics can be implemented if required. These calls may be ignored by the DLL if it does not model any discontinuities.

Elements of Arg3 INTEGER array (NI >= Nx+2):		
1	To DLL	N_x
2	To DLL	$N_u = 7$

3	To DLL	Blade failure type: 0 = no failure, 1 = blade is seized, 2 = blade runaway, 3 = blade is free/constrained, 4 = safety system activated*
4	From DLL	Output type: 0 = pitch rate, 1 = pitch acceleration, 2 = actuator torque.
5 to Nx+4	To/From DLL	0 = State disabled; 1 = State enabled
Elements of Arg5 REAL*8 array (NR1 >= Nx+10):		
1 to Nx	To/From DLL	State variables
Nx+1 to Nx+10	To/From DLL	User variables 1 to 10
Elements of Arg6 REAL*8 array (NR2 >= Nu):		
1	To DLL	Pitch position/velocity demand (rad)/(rad/s)*
2	To DLL	Pitch angular displacement (rad)
3	To DLL	Pitch angular velocity (rad/s)
4	To DLL	Pitch Bearing F _x (N)
5	To DLL	Pitch Bearing F _y (N)
6	To DLL	Pitch Bearing F _z (N)
7	To DLL	Pitch Bearing M _x (Nm)
8	To DLL	Pitch Bearing M _y (Nm)
9	To DLL	Pitch Bearing M _z (Nm)
10	To DLL	Pitching inertia (kgm ²)
11	To DLL	Pitching friction (Nm)
12	To DLL	Pitching stiction (Nm)

* Refer to table below about DLL actions for different failures

** Depending on whether control system implements a position or rate demand

DLL Responsibility on Blade Failures		
Index	Failure Type	DLL responsibility
0	No failure	DLL has normal responsibilities.
1	Blade seized	Inputs to DLL are not modified. DLL is responsible for modelling failure.
2	Blade runaway	Input position/rate demand is changed by Bladed .
3	Blade free/constrained	Bladed handles failure. DLL is ignored.
4	Safety System (rate demand)	Input position/rate demand is changed by Bladed .
4	Safety System (torque)	Bladed handles failure. DLL is ignored.

E.2 Discrete DLL interface

The DLL may be written in any language which can be compiled to a Windows DLL. The DLL entry point from **Bladed** is a single procedure called DLL_PITCH (upper case) with nine arguments and no return value. Either the `__cdecl` or the `__stdcall` calling convention may be used. The procedure definition is as follows:

Name	DLL_PITCH		
	Data type	Array length	Description
Return value	4-byte integer	-	Fail Flag: 0 = OK, <0 = Abort
Arguments:			
Arg1	4-byte integer	7	See below
Arg2	8-byte real ⁺	1	DLL time step
Arg3	8-byte real ⁺	12	Input data: See below
Arg4	1-byte char	Null terminated	File data/ Message
Arg5	8-byte real ⁺	1	Output data: See below
Arg6	4-byte integer	1	Fault state of each blade
Arg7	8-byte real ⁺	Arg1(5)	Output variables
Arg8	1-byte char	Arg1(6)	Output names

⁺ i.e. double precision float point

All arguments are passed by reference, so for example Arg1 is a pointer to the start of an array of eight 4-byte integers, Arg2 is a pointer to a single 8-byte (double-precision) floating point number. Character arguments are terminated by a null character (as C strings).

Elements of Arg1 INTEGER array		
1	To DLL	Flag number: 0 = first call, 1 = normal call, -1 = final call
2	To DLL	Pitch actuator type: 1 = Position, 2 = Rate
3	To DLL	Blade Number
4	From DLL	Number of output variables (constant)
5	From DLL	Length of output names (constant)
6	To DLL	Version number = 4401
7	To/From DLL	User variable (passed to and from external controller)

Elements of Arg3 REAL array		
1	To DLL	Pitch position/velocity demand (rad)/(rad/s)*
2	To DLL	Pitch angular displacement (rad)
3	To DLL	Pitch angular velocity (rad/s)
4	To DLL	Pitch Bearing F _x (N)
5	To DLL	Pitch Bearing F _y (N)
6	To DLL	Pitch Bearing F _z (N)
7	To DLL	Pitch Bearing M _x (Nm)
8	To DLL	Pitch Bearing M _y (Nm)
9	To DLL	Pitch Bearing M _z (Nm)
10	To DLL	Pitching inertia (kgm ²)
11	To DLL	Pitching friction (Nm)
12	To DLL	Pitching stiction (Nm)

Arg4 CHARACTER array	
To DLL	"Input file; output path; run name"
From DLL	Message returned by DLL (if return value < 0)

Elements of Arg5 REAL array		
1	From DLL	Actuator torque (Blade side)

Arg6 – Faults	
0	Normal operation
1	Seized
2	Runaway (Constant Rate)
3	Limp
4	Safety system activated

APPENDIX F TURBULENCE FILE FORMAT

The file format for turbulence files is described below.

Information in *italics* is only for information, and not actually used in simulations.

Record 1 (2-byte integer) with value -99 (indicates the current format).

Record 2 (2-byte integer):

- 1 = 1-component von Karman
- 2 = 1-component Kaimal
- 3 = 3-component von Karman
- 4 = improved von Karman
- 5 = IEC-2 Kaimal
- 6 = (not supported)
- 7 = General Kaimal
- 8 = Mann model

Subsequent records in the header are 4 byte real or integer values as follows:

The next group of records is only present if Record 2 \geq 7:

- (int) Number of bytes in the header (N_{header})
- (int) Number of turbulence components (1, 2 or 3)

The next group of records is only present in the case of the Improved von Karman model:

- (int) Number of turbulence components (1, 2 or 3)
- (real) *Latitude(in degrees)*
- (real) *Roughness length(m)*
- (real) *Reference height(m)*
- (real) *Longitudinal turbulence intensity(%)*
- (real) *Lateral turbulence intensity(%)*
- (real) *Vertical turbulence intensity(%)*

The next group is always present:

- (real) Grid point spacing in vertical direction, in m
- (real) Grid point spacing in lateral direction, in m
- (real) Grid point spacing in longitudinal direction, in m
- (int) Half the number of points in the alongwind direction
- (real) *The mean wind speed in m/s*
- (real) *The vertical length scale of the longitudinal component (zL_u) in m*
- (real) *The lateral length scale of the longitudinal component (yL_u) in m*
- (real) *The longitudinal length scale of the longitudinal component (xL_u) in m*
- (real) The maximum frequency in Hz.
- (int) *The random number seed*
- (int) The number of grid points vertically
- (int) The number of grid points laterally

The next group of records is only present if there are three components of turbulence:

- (real) *The vertical length scale of the lateral component (zL_v) in m*
- (real) *The lateral length scale of the lateral component (yL_v) in m*

- (real) *The longitudinal length scale of the lateral component (xL_v) in m*
- (real) *The vertical length scale of the vertical component (zL_w) in m*
- (real) *The lateral length scale of the vertical component (yL_w) in m*
- (real) *The longitudinal length scale of the vertical component (xL_w) in m*

The next group of records is only present if Record 2 = 7 (General Kaimal model):

- (real) *Coherence decay constant*
- (real) *Coherence scale parameter in m*

The next group of records is only present if Record 2 = 8 (Mann model):

- (real) *Shear parameter (gamma)*
- (real) *Scale length (L) in m*
- (real) *Ratio of lateral to longitudinal turbulence intensity*
- (real) *Ratio of vertical to longitudinal turbulence intensity*
- (real) *Maximum lateral/vertical wavelength in m*
- (real) *Reserved*
- (int) *Reserved*
- (int) *No. of FFT points*
- (int) *Reserved*
- (real) *Reserved*
- (real) *Reserved*
- (int) *Reserved*
- (int) *Reserved*
- (int) *Reserved*
- (real) *Reserved*
- (real) *Reserved*

Then the actual data starts (at byte $N_{\text{header}}+1$ if defined). These records are TWO-BYTE INTEGERS x 1000, i.e. the integer values should be divided by 1000. The result will be: Normalised, zero mean, unit standard deviation wind speed deviations*. One entry per grid point, each entry being one record if just the longitudinal component, or 3 records if all three components (longitudinal, lateral, vertical in that order).

Start in the bottom right corner of the grid, work along the row from right to left, then next row, etc. Repeat for each 'plane' in the alongwind direction.

The file must have a .wnd extension.

* Mann model: although the wind speed deviations are nominally zero-mean and unit-standard-deviation, this is not quite true for the Mann model since the time histories at any one grid point do not necessarily all have the chosen mean wind speed and turbulence intensity: there is a certain amount of natural variation between grid points, especially for short time histories.



REFERENCES

- [1] Germanischer Lloyd, Rules and Regulations IV, Non-marine technology, Part 1 – Wind Energy, Section 2.1993.
- [2] MATLAB – the Language of Technical Computing, www.mathworks.com.
- [3] “WindFarmer Theory Manual”, Garrad Hassan & Partners Ltd, 2002.
- [4] “WindFarmer User Manual”, Garrad Hassan & Partners Ltd, 2002.
- [5] IEC 1400-1, Turbine generator systems - Part 1: Safety requirements, Second edition, 1999.
- [6] Geometric stiffening options in Bladed (Bladed Portal document)
- [7] Bladed “Project Info” Features, Doc No 110052-UKBR-T-29-E, 11/08/2017.
www.dnvgl.com/services/bladed-3775.
- [8] Bladed 4.8 aerodynamics verification (Bladed Portal Document)
- [9] Guideline for the Certification of Wind Turbines, Germanischer Lloyd Industrial Services GmbH, 2010

LIST OF FIGURES

Figure 3-1: Diagram showing the Chord, the position of the Neutral axis relative to the blade root Z-axis and the leading edge, and the aerodynamic twist (seen from above). Note that this figure shows a Neutral axis with negative x and y co-ordinates with respect to the blade root co-ordinate system.....	16
Figure 3-2: Diagram showing the position of the Centre of mass relative to the leading edge, and the mass axis orientation	17
Figure 3-3: Diagram showing the position of point masses relative to the leading edge	18
Figure 3-4: Diagram showing the orientation of the principal axes and the location of the shear centre relative to the leading edge.....	20
Figure 4-1: Hub and blade coordinate system summary (view from the side of the turbine)	27
Figure 4-2: Hub and blade coordinate system summary (view from in front of the turbine)	28
Figure 4-3: Sealed and unsealed structural elements	33
Figure 4-4: Distributed springs vs. equivalent point springs	36
Figure 4-5: Spring conversion method.....	36
Figure 4-6: The mooring line layout, illustrating the mooring line coordinate system, which has an origin at the line anchor point.	40
Figure 4-7: Diagram showing positive directions for centre of mass position relative to the top of the tower, showing wind direction during normal running.	58
Figure 5-1: <i>A simple lag, where the transducer response tends towards the inputted simulated value.</i> .	68
Figure 5-2: <i>A lag with damping, where the transducer response can overshoot the inputted simulated value.</i>	68
Figure 5-3: Visualisation of the LIDAR axis defined in the mounting component coordinate system	76
Figure 5-4: LIDAR beam direction defined relative to the LIDAR axis coordinate system	77
Figure 5-5: Visualisation of the angles Y and Z returned by external controller API calls.	79
Figure 6-1: Propagation of a wake profile from upstream turbine to modelled turbine.....	82
Figure 6-2: Default nacelle orientation.....	86
Figure 7-1: Single-part blade model	97
Figure 7-2: Two-part blade model	98
Figure 7-3: Blade principal axes coordinate system	117
Figure 7-4: Relationship between blade "local element axes" and "principal axes coordinates"	118
Figure 7-5: Blade root axes coordinate system	119
Figure 7-6: Blade Aerodynamic axes coordinate system.....	120
Figure 7-7: Position of the user defined output axis.....	121
Figure 7-8: Co-ordinate system for stationary hub loads.....	123
Figure 7-9: Co-ordinate system for rotating hub loads.....	124
Figure 7-10: Blade and rotor coordinate systems for a clockwise upstream rotor	125
Figure 7-11: Blade and rotor coordinate systems for an anticlockwise upstream rotor	126
Figure 7-12: Blade and rotor coordinate systems for a clockwise downstream rotor	127
Figure 7-13: Blade and rotor coordinate systems for an anticlockwise downstream rotor	128
Figure 7-14: Co-ordinate system for tower loads (monopile only) and deflections	130
Figure 7-15: Multi-member tower coordinate output convention	131
Figure 7-16: Pitch system sign convention	132
Figure 7-17: Geometry of rotor including rotor and blade coordinate system	134
Figure 7-18: Aerofoil section with velocities and aerodynamic loads. The blade is pointing out of the paper and moves towards the right, while the flow is coming from below	135
Figure 8-1: Multiple processing: Channel combination	147
Figure 11.1: WindFarmer data entry	164
Figure 11.2: Turbulence intensity input to <i>WindFarmer</i>	165
Figure 11.3: Export channel selection in <i>WindFarmer</i>	166
Figure 11.4: Turbine selection in <i>WindFarmer</i>	166
Figure 11.5: Example turbulence factor file. Entries in bold must not be changed.	169



LIST OF TABLES

Table 4.1: Summary of hydrodynamic and water mass assumptions	34
Table 7.1: Blade loads output coordinate systems	122
Table 7.2: HydroAerodynamic outputs in Tidal Bladed	136



ABOUT DNV GL

Driven by our purpose of safeguarding life, property and the environment, DNV GL enables organizations to advance the safety and sustainability of their business. We provide classification and technical assurance along with software and independent expert advisory services to the maritime, oil and gas, and energy industries. We also provide certification services to customers across a wide range of industries. Operating in more than 100 countries, our 16,000 professionals are dedicated to helping our customers make the world safer, smarter and greener.