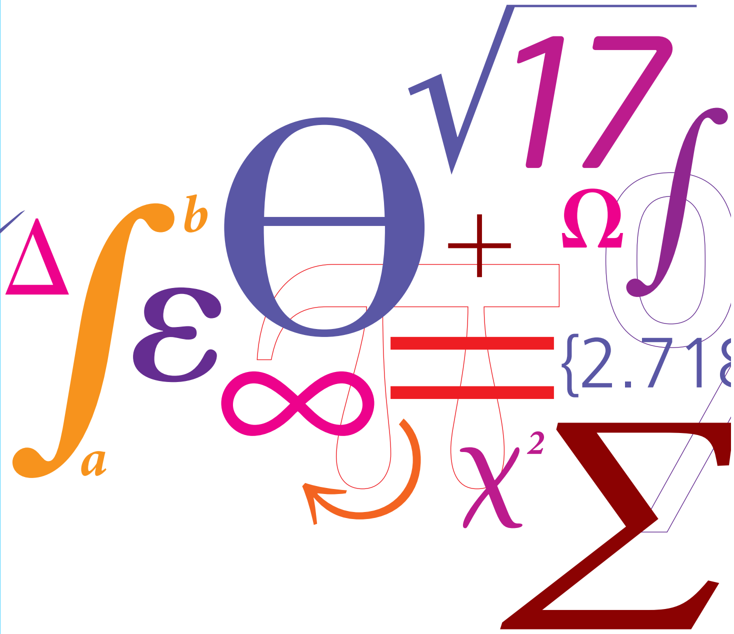
HAWCStab2 2.16: User Manual

DTU Wind

$$P = \frac{1}{2}\rho A v^3 C_p$$



Morten Hartvig Hansen, Lars Christian Henriksen, Carlo Tibaldi, Leonardo Bergami, David Verelst, Georg Pirrung, Riccardo Riva, Fanzhong Meng, Jennifer Rinker

June 2024

DTU Wind Energy
Department of Wind Energy

Authors: Morten Hartvig Hansen, Lars Christian Henriksen, Carlo Tibaldi, Leonardo Bergami, David Verelst, Georg Pirrung, Riccardo Riva, Fanzhong Meng, Jennifer Rinker
Title: HAWCStab2 2.16: User Manual
Institute: Department of Wind and Energy Systems

Summary:
This report is a user manual for the code HAWCStab2. HAWCStab2 is an implementation of an analytical linearization of a nonlinear finite beam element model. The beam model is coupled with an unsteady blade element momentum model of the blade aerodynamics. The aerodynamic model includes shed vorticity, dynamic stall, and dynamic inflow. The code allows for steady-states computations and open-loop and closed-loop modal analysis.

Publication Date: June 2024

HAWCStab2 version:
2.16

E-mail:
hawcstab2@vindenergi.dtu.dk

Web-page:
www.hawcstab2.vindenergi.dtu.dk

Address:
Technical University of
Denmark
DTU Wind Energy
Frederiksborgvej 399
4000 Roskilde
Denmark

Preface

This report is the user manual of HAWCStab2. HAWCStab2 was originally developed by Morten Hartvig Hansen. HAWCStab2 is a frequency based aeroservoelastic code for steady states computation and stability analysis of wind turbines. The code, to some extent, reads the same input files as HAWC2. HAWCStab2 is available in three versions: HAWCStab2, which is graphical user interface based program, HAWC2S, which is a command line based program suitable for e.g. optimization, and HS2pid, which is another command line program, is available with reduced functionality. HS2pid is only able to calculate tuning parameters for the Basic DTU Wind Energy Controller assuming torsionally stiff blades. HAWCStab2 is, so far, only able to handle 3 bladed wind turbines.

Contents

1. Introduction

HAWCStab2 is a tool developed at the Department of Wind and Energy Systems of the Technical University of Denmark. HAWCStab2 is an improved version of HAWCStab [1] with a different kinematic. The model is an analytical linearization of a nonlinear finite beam element model. The beam model is coupled with an unsteady blade element momentum model of the blade aerodynamic. The aerodynamic model includes shed vorticity, dynamic stall, and dynamic inflow. Hansen [2] gives a detailed description of the model. A validation and analysis of the open-loop performances are provided by Sønderby and Hansen [3]. An analysis in closed-loop is shown by Tibaldi et al. [7].

HAWC2S is the command-line version of HAWCStab2 and is run in a similar manner to HAWC2.

In Chapter 2 the basic structure of the htc file is explained. In Chapter 3 the output files are explained. In Chapter 4 a few examples on how to use the program are shown.

The examples shown in this document are based on:

- HAWC2 (version 13.1)
- HAWCStab2 (version 2.16)
- DTU 10MW RWT (version 2.0)

2. Input file and commands

The input to HAWCStab2 (and HAWC2S) is an htc-file, which is also used for HAWC2. The file used by HAWCStab2 has the normal HAWC2 specific commands as well as some HAWCStab2 specific commands. Extra HAWC2S commands must be included in the `hawcstab2` block if running with HAWC2S.

2.1. HAWC2 commands

The following subsections give a short descriptions of the HAWC2 input required by HAWCStab2. The HAWC2 user manual [4] available at www.hawc2.dk should be consulted for a detailed description of the commands.

The following sections describe the HAWC2 blocks of the htc file are used by HAWCStab2. Other blocks such as e.g. `simulation`, `aerodrag`, `force`, `hydro`, `soil` and `outputs` are not used by HAWCStab2.

2.1.1. Block `new_htc_structure`

The `new_htc_structure` block defines the structural setup of the wind turbine. Herein, it defines the various main bodies e.g. *tower*, *towertop*, *shaft*, *hub* and *blade* in the `main_body` sub block. The orientation of the main bodies is then defined in then `orientation` sub block. The interconnection of the main bodies is defined in the `constraints` sub block.

```
begin new_htc_structure;
  begin main_body;
  ...
  end main_body;
;
  begin orientation;
  ...
  end orientation;
;
  begin constraint;
  ...
  end constraint;
end new_htc_structure;
```

Main bodies The `main_body` blocks define all the beams used in the model. They may use the standard Timoshenko beam element, or the one with the fully populated matrix (FPM).

Constraints The bottom of the first main body defined in the `ground_fixed_substructure` (see Section 2.2.1) is fixed to the ground. For example, a tower clamped to the ground would then be written as

```
begin fix0;
  body tower;
end fix0;
```

The remaining constraints between the main bodies will follow the HAWC2 commands, with the exception of the shaft bearing.

Two different types of bearing constraints are basically available in HAWCStab2: `bearing1` and `bearing2`. The first type of bearing allows free rotation about one axis. This bearing is normally used for the shaft. A `bearing2` allows for a rotation about one axis where the angle is set from an input to the system. This type of bearing is normally used for the pitch bearing of pitch regulated wind turbines.

Note that if the user specifies a `fix1`, `bearing2` or `bearing3` for the constraint between the last main body of the HAWCStab2 substructures `ground_fixed_substructure` and `rotating_axissym_substructure`, then this bearing constraint will be treated as a constant speed bearing (the `omegas` input is ignored in case of a the `bearing3` command) and there is no generator rotation degree of freedom.

2.1.2. Block wind

The `wind` block contains information about `density` of air, which is only parameter used by HAWCStab2.

```
begin wind;
  density 1.225 ;
end wind;
```

2.1.3. Block aero

The `aero` block links the blades, to form a rotor. It contains information about aerodynamic properties for the blade such drag and lift coefficients. Furthermore, `induction_method` and `tiploss_method` are used by HAWCStab2.

```
begin aero ;
  nblades  3;
```

```
  hub_vec shaft -3 ;
  link 1 mbdy_c2_def blade1;
  link 2 mbdy_c2_def blade2;
  link 3 mbdy_c2_def blade3;
  ae_filename        ./data/DTU_10MW_RWT_ae.dat ;
  pc_filename        ./data/DTU_10MW_RWT_pc.dat ;
  induction_method   1 ;     0=none, 1=normal BEM dynamic induction
  aerocalc_method    1 ;     0=none, 1=normal
  aerosections       50 ;
  ae_sets            1 1 1;
  tiploss_method     1 ;     0=none, 1=prandtl
  dynstall_method    2 ;     0=none, 1=stig oye, 2=mhh method, 3=ATEFlap
end aero ;
```

There must be exactly one `aero` block.

2.2. HAWCStab2 commands

HAWCStab2 needs a specific block called `hawcstab2`. Within this block different commands are specified, which can be divided into the following inputs:

- structural setup
- damping
- aerodynamic options
- operational data
- controller tuning
- controller input/output
- HAWC2s specific commands
- advanced options

All of these inputs are explained in the following sections.

2.2.1. Structural setup

The structural setup is specified through three different blocks where the bodies are listed:

- `ground_fixed_substructure`: main bodies that are fixed with respect to the ground, e.g. tower, and tower top;
- `rotating_axissym_substructure`: rotating main bodies that are not part of the rotor, e.g. shaft. These bodies have to be axis-symmetric;
- `rotating_threebladed_substructure`: rotating main bodies that are part of the rotor, e.g. hub and blades. Since HAWCStab2 assumes 3 bladed with isotropic rotor, only the first blade and hub bodies need to be specified, the others will be included automatically.

A second-order model of a pitch actuator can also be included in the wind turbine model. The model is included adding the line

```
second_order_actuator pitch1 100.0 0.7 ;
```

in the block `rotating_threebladed_substructure`. The first number in the command indicates the frequency of the second-order model, the second its damping ratio.

All the aerodynamic forces are assumed to be applied on the last main body in the block `rotating_threebladed_substructure`.

The format of these commands is:

```
begin hawcstab2 ;
  begin ground_fixed_substructure ;
    main_body tower ;
    main_body towertop ;
  end ground_fixed_substructure ;
  begin rotating_axissym_substructure ;
    main_body shaft ;
  end rotating_axissym_substructure ;
  begin rotating_threebladed_substructure ;
    main_body hub1 ;
    main_body blade1 ;
    second_order_actuator pitch1 100.0 0.7 ;
  end rotating_threebladed_substructure ;
end hawcstab2 ;
```

2.2.2. Damping

If `log_decrements` is present in the block of either `ground_fixed_substructure`, `rotating_axissym_substructure` or `rotating_threebladed_substructure` then the HAWC2 specific damping commands will be overwritten by a spectral damping model will be used to calculate the damping properties. If for example the following command is present in the `rotating_threebladed_substructure`:

```
log_decrements 1.0 1.2 1.5 2.0 ;
```

then the first four modes of the unloaded blade are structurally damped 1.0%, 1.2%, 1.5%, and 2.0%. The logarithmic decrements of higher order modes will be increased relatively with the factor 1.1 until a hard-coded maximum of 70%.

If `log_decrements` is not used then the Rayleigh type damping model of HAWC2 will be used. The damping properties will be calculated for the unloaded, standstill wind turbine. It is strongly recommended only to use stiffness proportional terms. If mass proportional terms are used, the damping for HAWC2 and HAWCStab2 will not be the same. Consult Hansen [5] for more information about the mixed mass/stiffness damping model.

2.2.3. Aerodynamic options

Two unsteady aerodynamics models are included in HAWCStab2: dynamic stall (including unsteady airfoi aerodynamics in attached flow, [8]) and dynamic inflow [9]. The following options can be set in the GUI in the `Lock DOFs` dialog or in the HAWC2s command `degrees_of_freedom`, see Section 2.2.11

Dynamic stall

- **Unsteady airfoil aerodynamics (default)** An effective angle of attack lags behind the unsteady angle of attack (Theodorsen effect in attached flow) and there is a time lag on the separation point position, creating dynamic stall loops. The modelling of these effects needs 4 states per aerodynamic section per blade.
- **Quasi-steady airfoil aerodynamics** The angle of attack and separation point position are always at their quasi steady value and the lift, drag and moment coefficients follow directly from the airfoil polars. No aerodynamic states are needed.

Dynamic inflow

- **Frozen wake (default)** The induced velocities remain at the steady state value when the rotor forces change. No aerodynamic states are needed.
- **Quasi steady inflow** The induced velocities change immediately to the steady state values, such that the wake is always in equilibrium. This setting is quite academic because the inflow reacts very slowly in reality. No aerodynamic states are needed.
- **Dynamic inflow** The induced velocities react slowly to changes in the forces on the rotor disc. This is the most realistic setting, applying two first order filters per aerodynamic section. The aeroelastic model grows by two aerodynamic states per aerodynamic section per blade. The time constants depend on the operating point (mainly the wind speed) and the radial position of the respective section on the blade.

Both dynamic inflow and dynamic stall are implemented in Coleman coordinates (collective, cosine, sine).

To obtain accurate aerodynamic damping the dynamic stall model should be active (which is the default). The resulting phase lag and diminished amplitude of the aerodynamic forces typically reduces the absolute value of the aerodynamic damping and leads to less conservative estimations of the aeroelastic stability limit in attached flow. The dynamic inflow model, on the other hand, operates on a much slower time scale and is mainly important for low frequency modes, such as the fore-aft motion of a floating turbine or the slow controller action. For blade stability the influence of dynamic inflow is typically small, but activating dynamic inflow nevertheless leads to the most accurate results.

2.2.4. Operational data

The `operational_data` block is optional. It is used to set the default values of the parameters in the dialogue window to compute the operational data points and to set the values when running with HAWC2S. The parameters of this block are:

- `windspeed` followed by either 3 or 4 arguments:

1) V_{min} minimum wind speed
2) V_{max} maximum wind speed
3) number of wind speeds between min and max wind speed

When using rotor speed curtailment for wind speeds above maximum wind speed until the storm wind speed, there is one additional argument, and the 3rd argument changes context:

3) V_{storm} storm wind speed
4) number of wind speeds between min and storm wind speed

- `genspeed` followed by the minimum rotational speed and the maximum rotational speed in rpm. If the user gives the rotational speed on the high speed shaft, the correct gearbox ratio must be given in the following parameter `gearratio`. If instead the user provides the rotational speed on the low speed shaft, or it is a direct drive wind turbine, the gearbox ratio must be 1 in the following parameter `gearratio`.
- `gearratio` and the gearbox ratio.
- `minpitch` and the minimum pitch angle in degree.
- `opt_lambda` and the value of the tip-speed-ratio for the variable speed region.
- `maxpow` and the value of the aerodynamic rated power in kW.
- `prvs_turbine` and an integer to indicate the type of pitch regulation. 0 for fixed pitch and 1 for variable pitch.
- (optional) `include_torsiondeform` and an integer to indicate if blade flap/edge/torsion deformations should be included in the computation. 0 (default) for no blade deformations and 1 for with blade deformations.
- (optional) `operational_data_file_wind` and an integer to indicate if the optimal pitch angle and rotor speed should be computed at the wind speeds specified in the operational data file. 0 (default) for using the equidistant wind speed distribution specified by `windspeed` above and 1 for using the windspeeds as defined in the operational data file.
- (optional) `set_torque_limit` and an integer to indicate whether torque limits should be applied during calculation of optimal pitch angle. 0 (default) sets torque limit and 1 is no torque limits. See more details in Sec. 2.2.5.

If the operational data points have been precomputed or the user wants to enter them manually, it is possible to specify them through a file. The file is specified by the following command:

```
operational_data_filename ./operational_data_filename.opt ;
```

When given as input, the file requires three columns: one for the wind speed, one for the pitch angle and one for the rotor speed. The number of data points included in the file needs to be specified in the first row of the file. HAWCStab2 modifies this file, by adding two extra columns containing the aerodynamic power and thrust. *These last two columns are not needed as inputs*, because HAWCStab2 is used to compute them.

If the operational data file is only used to specify the wind speeds at which the optimum pitch angle and rotor speed should be computed, the second and third columns need to be present (pitch and rotor speed), but they can be filled with zeros as dummy values.

```
22 Wsp [m/s] Pitch [deg] Rot.speed [rpm] Aero power [kW] Aero thrust [kN]
         4.0    2.889748         6.000000       287.319260        224.286816
         5.0    2.115800         6.000000       805.573745        352.209828
         6.0    1.109058         6.000000      1543.002742        500.388658
         7.0    0.000048         6.000000      2525.245528        658.232557
         8.0    0.000055         6.424607      3770.277010        816.795864
         9.0    0.000019         7.226938      5374.530562       1034.430140
        10.0    0.000056         8.031337      7378.855371       1277.059791
        11.0    0.000048         8.839966      9826.489718       1544.121977
        12.0    4.807932         9.600000     10636.875545       1262.557036
        13.0    7.388350         9.600000     10640.312112       1080.883000
        14.0    9.289680         9.600000     10634.865878        970.253169
        15.0   10.887191         9.600000     10652.538640        892.739336
        16.0   12.346992         9.600000     10618.567809        828.619150
        17.0   13.672693         9.600000     10631.933899        780.318935
        18.0   14.926127         9.600000     10646.833268        740.590975
        19.0   16.120324         9.600000     10640.981055        706.090035
        20.0   17.268079         9.600000     10646.834596        677.263923
        21.0   18.374175         9.600000     10632.078861        651.469908
        22.0   19.443877         9.600000     10648.917163        630.689690
        23.0   20.484638         9.600000     10622.019335        610.636055
        24.0   21.496839         9.600000     10628.972396        594.608862
        25.0   22.485124         9.600000     10638.692060        580.889756
```

Figure 2.1.: Example of the operational data file file containing information about the operational points for selected wind speeds.

2.2.5. Compute optimal operating points and storm curtailment

The coomand `compute_optimal_pitch_angle use_operational_data;` will calculate the optimal pitch and rotor speed curves based on the values given in the `operational_data` block. All necessary inputs for the htc code block `operational_data` are described under the previous section “Operational data”.

The algorithm is described as follows:

- Variable pitch and rotor speed:
  - Set rotor speed Ω so it tracks the optimal tip speed ratio λ_{opt} until the maximum rotor speed is reached.
  - Find the appropriate pitch angle in order not to exceed the rated power. In this process the following is minimized: $\sqrt{(P_{ref} - P)^2}$, and where the reference power is defined as: $P_{ref} = P_{max}\Omega/\Omega_{max}$ if `set_torque_limits` is set to 0 (default value). If `set_torque_limits` is set to 1 (no limits), the reference power is defined as $P_{ref} = P_{max}$.
- Fixed pitch and variable rotor speed:
  - Pitch angle is set to `minpitch`

- Find the rotor speed for which $\sqrt{(P_{ref} - P)^2}$ is minimized, and where $P_{ref} = P_{max}\Omega/\Omega_{max}$.

In the storm curtailment region, $V_{max} < V < V_{storm}$, the rotor speed is linearly decreased from maximum to minimum rotor speed.

2.2.6. Controller tuning

This section contains two main commands. A command to set the parameters to automatically compute the tuning of the controller and a command to manually specify the controller tuning.

The `controller_tuning` block is optional, see 4.1.4 for an example. It is used to set the default values of the parameters in the dialog window to tune the controller and to set the values when running with HAWC2S.

When tuning the controller, HAWCStab2 will attempt to identify the control regions but may not identify them correctly, causing the tuning to fail. Sec. 2.2.7 explains how to manually specify them.

The parameters of this block are:

- `partial_load`, the frequency [Hz], and damping ratio [–] of the regulator mode. These vales are used for the pole placement of the PI controller on the generator torque in partial load region.
- `full_load`, the frequency [Hz], and damping ratio [–] of the regulator mode. These vales are used for the pole placement of the PI controller on the pitch in full load region.
- `gain_scheduling` and an integer to specify the type of gain scheduling. `1` for linear and `2` for quadratic.
- `constant_power` and an integer to specify if the regulator strategy is constant torque `0` or constant power `1`.
- `rotorspeed_gs` and an integer to specify if the gain scheduling should contain also a term due to the aerodynamic damping `0` or `1`.
- `regions` and four integers to specify the operational points at which there is a transition in the controller operational regions. This command is optional and overwrites the build-in function that identifies the operational regions. See the following Section 2.2.7 for additional details.

Two different controllers can be added to the model through the following commands:

- `basic_dtu_we_controller` (# 1)
- `pi_pitch_controller` (# 2)

The first controller is a simplified linearization of the Basic DTU Wind Energy controller, so it includes sub-controllers to handle the different operational regions. The second controller is only meant for the full load region and it is a basic PI pitch controller. Both commands require several tuning parameters. The parameters are described in Table 2.1.

# 1	# 2	Parameter	Unit	Description
	1	`P_rated`	kW	Rated power.
	2	`Omega_rated`	rad/s	Rated rotor rotational speed.
1		`Kp_partial`		Prop. gain of partial load PI torque controller.
2		`Ki_partial`		Int. gain of partial load PI torque controller.
3		`Kopt_partial`		K-omega control parameter.
4	3	`Kp_full`		Prop. gain of full load PI pitch controller.
5	4	`Ki_full`		Int. gain of full load PI pitch controller.
6	5	`K1_theta`		Gain scheduling parameter of the full load PI gains w.r.t. pitch angle.
7	6	`K2_theta`		Gain scheduling parameter of the full load PI gains w.r.t. pitch angle.
8	7	`omega_filt`		Natural frequency of second order speed filter.
9	8	`csi_filt`		Damping ratio of second order speed filter.
10		`DT_freq`		Frequency of a band-stop filter to remove the drivetrain frequencies.
11	9	`type`		Full load generator control type: 1 constant power, 0 constant torque.
12	10	`K0_omega`		(Optional) Gain scheduling of the full load PI gains w.r.t. rotor speed.
13	11	`K1_omega`		(Optional) Gain scheduling of the full load PI gains w.r.t. rotor speed.
14	12	`K2_omega`		(Optional) Gain scheduling of the full load PI gains w.r.t. rotor speed.

Table 2.1.: Parameters for build-in controller commands.

2.2.7. Controller regions

The different control regions, the commands to choose them in *htc* input file, and the effects of choosing the respective region on the rotor speed and pitch control are shown in Table 2.2. The four integers in the htc command determine at which operating point the controller should change to the next region. If the `regions` command in the table is used, the respective region will be selected for a single operating point.

As an example for 12 operating points between 4 and 26 m/s wind speed the command to run all regions in one computation could be `regions 2 3 5 13 .` Then HAWCStab2 uses fixed speed for operating point 1 (4 m/s), variable speed for operating point 2 (6 m/s), changes to fixed speed at operating point 3 (8 m/s), starts pitching at operating point 5 (12 m/s) and doesn't use storm control.

Region name	Htc command	Rotor speed	Pitch
Region 1	`regions 2 2 2 2`	fixed	fixed
Region 2	`regions 1 2 2 2`	variable	fixed
Region 2.5	`regions 1 1 2 2`	fixed	fixed
Region 3	`regions 1 1 1 2`	fixed	variable
Region 4	`regions 1 1 1 1`	storm control	

Table 2.2.: Brief description of the control regions available in HS2. The htc commands shown here will select the respective region for a computation on a single operating point.

2.2.8. Controller input/output

This section specifies the input and outputs to the wind turbine models. These are used to compute the input and output matrices, and in turn the transfer function. The inputs and outputs are specified following the outputs convention of HAWC2.

The controller block can contain the following commands:

- `operational_data_filename`
- `begin input;` closed by `end;`
- `begin output;` closed by `end;`

Command operational_data_filename

The syntax is: `operational_data_filename` *file_name*`;`. The argument *file_name* is the path to the file containing the operational data for control purposes. This file is different from the one used outside of the controller block.

2.2.9. Block input

The model inputs are specified in the `input` block. This block can contain:

- `constraint bearing1` *bearing_name multi_blade_name*`;`
- `constraint bearing2` *bearing_name multi_blade_name*`;`
- `mbdy force_ext` *mbody_name node_number axis coordinate_system multi_blade_name*`;`
- `mbdy moment_ext` *mbody_name node_number axis coordinate_system multi_blade_name*`;`
- `aero beta` *multi_blade_name flap_number*`;`

The argument *multi_blade_name* can be:

- `no` (default)
- `collective`
- `cosine`
- `sine`

The argument *coordinate_system* can be

- `local`
- *mbody_name*

2.2.10. Block output

The model outputs are specified in the `output` block. This block can contain:

- `constraint bearing1` *bearing_name unit_code* [option 1 or 2]`;`
- `constraint bearing2` *bearing_name unit_code* [option 1 or 2]`;`
- `mbdy forcevec` *mbody_name element_number node_number coordinate_system* [option 1 or 2]`;`
- `mbdy momentvec` *mbody_name element_number node_number coordinate_system* [option 1 or 2]`;`
- `mbdy state` *state_type mbody_name element_number location coordinate_system* [option 1 or 2]`;`
- `mbdy state_rot eulerang_xyz` *mbody_name element_number location coordinate_system* [option 1 or 2]`;`

Options 1 and 2 are:

1. *multi_blade_name*
2. `only` *axis multi_blade_name*. The argument *axis* can be `1`, `2` or `3`.

The argument *unit_code* specifies the unit of the output for angles and angular speeds. It can be:

1. rad and rad/s
2. deg and rpm
3. deg and rad/s
4. deg and rad/s

5. deg and deg/s

The argument *multi_blade_name* can be:

- `no` (default)
- `collective`
- `cosine`
- `sine`

The argument *coordinate_system* can be

- `local`
- *mbody_name*
- `global` (only available for `mbdy state` and `mbdy state_rot eulerang_xyz`)

The argument *state_type* can be:

- `pos` for position
- `vel` for velocity
- `acc` for acceleration

The argument *location* goes from 0.0 for node 1 to 1.0 for node 2.

An example of controller block is the following.

```
begin controller ;
    begin input ;
        constraint bearing1 shaft_rot ; 0
        constraint bearing2 pitch1 collective ; 1
    end input ;
    begin output ;
        constraint bearing1 shaft_rot 1 only 2 ; 0
        constraint bearing2 pitch1 1 only 1 collective ; 1
        mbdy momentvec tower 1 2 tower ; 2 3 4
        mbdy momentvec blade1 1 2 blade1 collective ; 5 6 7
        mbdy momentvec blade1 1 2 blade1 cosine ; 8 9 10
        mbdy momentvec blade1 1 2 blade1 sine ; 11 12 13
    end output ;
end controller ;
```

2.2.11. HAWC2S commands

HAWC2S is the command line version of HAWCStab2. The input file for HAWC2S must contain all the parts used in the input file for HAWCStab2.

When using HAWC2S, the commands, that are selected through the GUI interface in HAWCStab2, must be included in the htc file as command lines. These are then executed as a workflow. The commands have to be inserted in the `hawcstab2` section. Several commands require the `compute_steady_states` command to be run beforehand. These commands are marked with a "§" symbol.

For some commands an output file is generated and the file name is derived from the used `htc` input file, and which is represented here conceptually as: `input_htc_file.ext`. The new output is then converted to the following format: `input_htc_file_APPENDIX.NEW_EXT`, where `_APPENDIX` is used for some of the output commands but not all (see below for more details), and depends on the analysis executed prior to save command. The file extension is replaced with an appropriate alternative `NEW_EXT` for each of the different commands.

The commands available are:

- `output_folder`
  Parameters:
  - a string to specify the folder where all output files will be created. Defaults to the current working directory.
- `compute_optimal_pitch_angle use_operational_data`
  will compute and save to a file the operational data points according to the parameters inserted in the `operational_data` block. The results are saved to file `input_htc_file.opt`. See Sec. 2.2.5.
- `compute_structural_modal_analysis`
  Parameters:
  - `bladeonly` or `nobladeonly` to specify if the analysis is for the blade only or for the whole wind turbine.
  - an integer to specify the number of modes.

  Results saved to file (see section 3.6):
  - `input_htc_file_Blade_struc.cmb` for a structural blade-only analysis.
  - `input_htc_file_struc.cmb` for a structural system analysis
- `compute_steady_states` - (Sec. 4.1.2)
  to compute the steady states from given operational points. The command needs four parameters
  - `bladedeform` or `nobladedeform` to specify if blade deformations needs to be included in the computations.
  - `tipcorrect` or `notipcorrect` to specify if tip correction needs to be included in the computations.
  - `induction` or `noinduction` to specify if induction needs to be included in the computations.
  - `gradients` or `nogradients` to specify if gradients needs to be computed. The gradients are then printed in the `.pwr` file.
- `compute_stability_analysis`§ - (Sec. 4.1.3)
  Parameters:
  - `bladeonly` or `windturbine`
  - an integer to specify the number of modes.

  Results saved to file (see section 3.6):
  - `input_htc_file_Blade.cmb` for a blade-only aeroelastic analysis.
  - `input_htc_file.cmb` for a system aeroelastic analysis.
- `compute_aeroservoelastic`§

Parameters:
  - an integer to specify the number of modes.

Results saved to file (see section 3.6):
  - input_htc_file_Servo.cmb
- save_ol_matrices§ - Writes out the open-loop A,B,C,D matrices to text files.
- save_ol_matrices_full§ - Writes out the M,D,K matrices to text files.
- save_ol_matrices_all§ - Writes out both A,B,C,D and M,D,K matrices to text files.
- save_cl_matrices_all§ - Writes out the closed-loop A, B, Bv, C, D, Dv, E, F, Fv matrices to text files.
  This command needs the block controller. Beside the specified closed-loop aero-servo-elastic matrices, additional matrices can be saved by specifying *one* of the following additional arguments:
  - ctrl_out Saves also the controller matrices, Ac, Bc, Cc, Dc.
  - vloc_out Saves also the local wind matrices, Bv loc, Dv loc, Fv loc
  - ctrl_vloc_out Saves both the controller matrices and the local wind ones.
- compute_controller_input§ (Sec. 4.1.4)
  This command needs the block controller_tuning. Optional parameter:
  - outputfile.txt, defaults to input_htc_file_ctrl_tuning.txt.
- save_power§
  Results saved to file: input_htc_file.pwr.
- save_induction§
  For each operating point three files are saved, and each operating point contains the used wind speed as a reference WSP=int(windspeed*1000):
  - input_htc_file_uWSP.ind
  - input_htc_file_fext_uWSP.ind
  - input_htc_file_defl_uWSP.ind
- degrees_of_freedom
  Lock different degrees of freedom and select the inflow model. Parameters:
  - true or false (default) to specify if the ground fixed substructure is rigid
  - true or false (default) to specify if the rotating axial symmetric substructure is rigid
  - true or false (default) to specify if the rotating three bladed substructure is rigid
  - true or false (default) to set quasi-steady aerodynamic
  - frozen (default), quasi or dynamic to indicate the desired type of inflow.
- save_beam_data
- save_blade_geometry
- save_aero_point_data
- save_profile_coeffs
- save_modal_amplitude
  Save modal amplitudes and phases to file: input_htc_file_APPENDIX.amp as follows:
  - input_htc_file_Blade_struc.amp for a structural blade-only analysis.
  - input_htc_file_struc.amp for a structural system analysis
  - input_htc_file_Blade.amp for a blade-only aeroelastic analysis.
  - input_htc_file.amp for a system aeroelastic analysis.
  - input_htc_file_Servo.amp for an aero-servo-elastic analysis

  In the case of a blade-only analysis, for each operating point the program saves the mode shapes of the whole blade. Instead, for a system analysis, for each operating point the program saves only a summary of the complete mode shapes matrix. For the ground_fixed_substructure the program looks for a body named tower and, if it does not find any body with this name, it selects the last body of this substructure (normally the tower top). It then picks the last node for this body, i.e. the top of the tower. For the rotating_axissym_substructure, the program selects the last node of the last body, i.e. the end of the shaft towards the rotor center. Lastly, the program selects the three blade tips, by picking the last node of the last body in the rotating_threebladed_substructure. The description of the output is provided in section 3.7.

- save_modal_binary
  Save modal results in binary format to file: input_htc_file_APPENDIX.NEW_EXT as follows:
  - input_htc_file_Blade_struc_Modal.hmd for a structural blade-only analysis.
  - input_htc_file_struc_Modal.hmd for a structural system analysis
  - input_htc_file_Blade_Modal.hsd for a blade-only aeroelastic analysis.
  - input_htc_file_Modal.hsd for a system aeroelastic analysis.
  - input_htc_file_Servo_Modal.hsd for an aero-servo-elastic analysis
- save_eigenvalues
  Save eigenvalues to file: input_htc_file_APPENDIX.dat as follows:
  - input_htc_file_Blade_struc.dat for a structural blade-only analysis.
  - input_htc_file_struc.dat for a structural system analysis
  - input_htc_file_Blade.dat for a blade-only aeroelastic analysis.
  - input_htc_file.dat for a system aeroelastic analysis.
  - input_htc_file_Servo.dat for an aero-servo-elastic analysis

2.2.12. Advanced options

Advanced options commands can be entered as HAWC2S commands but will be executed also with HAWCStab2.

- verbose
  This command prints additional information in the log files.
- steady_state_convergence_limits
  Modify the convergence criterion for the computation of the operational points and steady states. The command is followed by a sequence of nine parameters. The parameters and their default values when the command is not issued are:
  1. Absolute tolerance on the 2-norm of the change of induction factors in each aerodynamic section, default=1e-6
  2. Maximum number of BEM iterations in a single aerodynamic section, default=10000
  3. Relaxation factor of the BEM iterations (low number is stable but slower), default=0.02
  4. Relative tolerance on the force differences in the inner and outer iteration loops (see Figure 2.2), default=1e-5
  5. Maximum number of iterations in either the inner or outer iteration loop, default=500
  6. Relaxation factor of the increment of the blade deformation, default=0
  7. Maximum variation of operating point characteristic in compute optimal operation data (e.g. pitch angle above rated), default=10.0. The variation is given with respect to the previously computed value, or to zero for the first point; hence this value should be increased when computing operational data at a single operating point above rated.

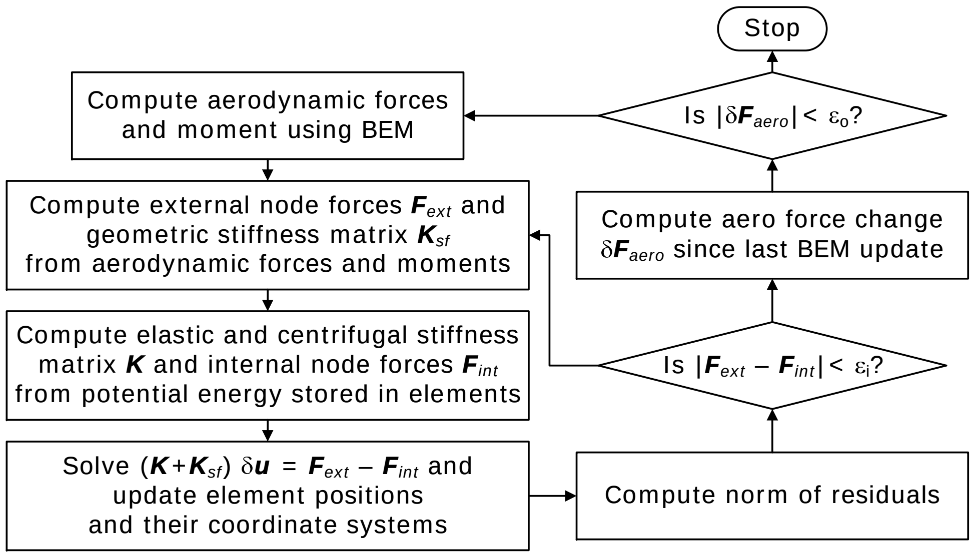


Figure 2.2.: Diagram representing the iterative process to obtain the nonlinear steady-states solution.

8. Maximum variation of operating point characteristic between stiff computations and computations with blade deformation in compute optimal operation data. The default value `5.0` can be decreased for wind turbines that are not very flexible.
9. Absolute tolerance on pitch angle for the optimal operational point computations, default=`1e-9`.

Example:
`steady_state_convergence_limits 1e-7 1e4 0.02 1e-6 1e3 0 10.0 5.0 1e-9`
Figure 2.2 shows a representation of the iterative process to obtain the nonlinear steady-states solution. Two loops can be identified: an outer loop where the aerodynamic forces are calculated with the BEM and an inner loop where the deflections are computed for fixed forces. Both loops these use the aeroelastic convergence parameters (4, 5, and 6). Their absolute tolerances are computed from the relative tolerance (denoted ϵ_{rel}) as $\epsilon_i = \epsilon_o = 150\epsilon_{rel}S_o$, where S_o is the blade curve length.

- `print_full_precision`
  Save operational points file with extended precision.
- `factor_eigenvalue_distance`
  Followed by one integer that will be used as the penalty factor for the eigenvalue distance in the mode sorting routine, see Section 2.3.
- `phi_per_node`
  Followed by one integer allows to control the amount of model reduction. By setting it to 6, no reduction is performed. The default is 3.

2.2.13. Encrypted models support

HAWCStab2 is able to read models where the structure and/or the aerodynamics are encrypted. In this case, some output commands are disabled.

When only the aerodynamic layout or profile coefficients are encrypted, the following commands are disabled:

- `save_aero_point_data`
- `save_profile_coeffs`
- `save_induction`

When at least one Timoschenko input file is encrypted, the following commands are disabled:

- `save_beam_data`
- `save_blade_geometry`

The following commands are disabled in both cases.

- `save_ol_matrices`
- `save_ol_matrices_full`
- `save_ol_matrices_aero`
- `save_ol_matrices_all`
- `save_cl_matrices_all`
- `save_modal_binary`
- `save_modal_amplitude`
- `save_eigenvectors`

The HAWCStab2 GUI is allowed to run, but with the corresponding menu items disabled.

2.3. Automatic mode sorting

Since version 2.14, HAWCStab2 and HAWC2S have an automatic mode sorting algorithm. The modes are the eigensolutions most similar to the structural modes. Thus, the open- or closed-loop aeroelastic modes sorted out are only the modes that have a mode shape similar to the structural modes. This algorithm removes eigensolutions dominated by state variables of the unsteady aerodynamic models or the controller equations. Note that the user can change the number of modes to be plotted and saved under the menu **Plot**.

The mode sorting works by comparing mode shapes using a modal assurance criterion (MAC). The modal assurance criterion is 1 for identical mode shapes and 0 for mode shapes with no similarity. The mode sorting works differently for the first or a following operating point:

First operating point The aeroelastic modes at the first operating point are compared to structural modes of the turbine at the first operating point. A penalty on the MAC is applied based on the frequency difference $\omega_i - \omega_j$ between the structural and aeroelastic mode: $\mathrm{MAC}^* = \mathrm{MAC} e^{-\gamma|\omega_i - \omega_j|}$, where γ is a user-defined parameter.

N'th operating point The aeroelastic modes at the n'th operating point are compared to the aeroelastic modes at the (n-1)'th operating point. A penalty is applied based on the eigenvalue difference $\lambda_i - \lambda_j$ (both real and imaginary part / frequency and damping) between the aeroelastic modes: $\mathrm{MAC}^* = \mathrm{MAC}e^{-\gamma|\lambda_i - \lambda_j|}$.

If the sorting does not perform as desired, this penalty can be adjusted by changing the value γ with the command `factor_eigenvalue_distance` followed by an integer. The default value is 6, which works well for the DTU 10 MW reference turbine. If the factor is 0, the modes are sorted only based on mode shapes and with increasing number the difference in frequencies or eigenvalues becomes increasingly important.

3. Output files

This chapter describes some of the output files that can be generated with HAWCStab2. When saving result files, extensions must be added in the file name. It is an advantage to use the extensions suggested in the dialog window because the already existing files of similar format are then filtered out.

3.1. Aeroelastic model properties in the logfile

Most logfile outputs are self-explanatory, such as for example *relative orientation input commands read with succes*. But the logfile also includes a list with information about the size of the aeroelastic problem. A short description of the listed parameters is shown in Table 3.1. They include some parameters that are intended to prepare for a future HAWCStab2 version for n-bladed rotors but default to a three bladed rotor in the current version. At the moment no further development effort in the direction of n-bladed rotors is planned due to the inherent modeling difficulties of this task.

Name	Description
nblades	# of blades
nblades3	# of blades (3 bladed rotor)
ndofs	# of structural degrees of freedom (DOF)
ndofs3	# of structural DOF (3 bladed rotor)
ndofs_n	# of structural DOF (Actual blade number)
nfree_bearings	# of free bearings
nbearings	# of bearings
nbearings_n	# of bearings (actual blade number)
naerostates_ds_bld	# of dynamic stall states per blade (equals 4*naerosections)
naerostates_ds	# of dynamic stall states for a 3 bladed rotor
naerostates_ds_n	# of dynamic stall states (actual blade number)
naeroforces_bld	# of aerodynamic force components (fx,fy,m) per blade (equals 3*naerosections)
naeroforces	# of aerodynamic force components (fx,fy,m) for a 3 bladed rotor
naeroforces_n	# of aerodynamic force components (fx,fy,m) (Actual blade number)
naerosections	# of aerodynamic sections per blade. In comparison to HAWC2 the sections at the very blade root and tip are ignored. Therefore this is the number of naerosections(htc-file) minus 2.
nphis	# of DOF of the reduced structural model
nphis3	# of DOF of the reduced structural model for a 3 bladed rotor
nphis_n	# of DOF of the reduced structural model (Actual blade number)
phi_per_node	# of reduced DOF per structural node
ndim_y	# of outputs
ndim_u	# of inputs

Table 3.1.: Description of the aeroelastic model properties in the log file.

3.2. Operational data in .opt

The operational data file is an input/output file. To perform steady-states computations a set of operational points is required, and these are passed to HAWCStab2 with an .opt file. This file can be generated by HAWCStab2 with the commands `Optimal operational data` plus `Save optimal power data`. When saved, the file contains five columns, each column correspond to:

1. Wind speed [m/s]
2. Pitch angle [deg]
3. Rotor speed [rpm]
4. Aerodynamic power [kW]
5. Aerodynamic thrust [kN]

When the file is used as input, it can be arbitrary modified by the user, i.e., any operational point can be given as an input. Because an operational point is defined uniquely by wind speed, pitch angle, and rotor speed only the first three columns need to be present in the file, all the other columns are not read. Special attention need to be paid to the first line of the file because it contains a number. This number needs to be equivalent to the number of operational points included in the file.

3.3. Controller tuning parameters

The controller tuning output for the Basic DTU Wind Energy is saved into text file that has the following form:

```
PI generator torque controller in region 1
K =   0.861734E+07 [Nm/(rad/s)^2]
PI generator torque controller in region 2
I =   0.161031E+09 [kg*m^2]
Kp =   0.708251E+08 [Nm/(rad/s)]
Ki =   0.158931E+08 [Nm/rad]
PI pitch angle controller in region 3 (constant speed, constant torque)
Kp =   0.125213E+01 [rad/(rad/s)]
Ki =   0.337174E+00 [rad/rad]
K1 =       11.32035 [deg], K2=464.52578 [deg^2] (dq/dtheta=-1184.66330 kNm/deg)
Additional terms due to the Aerodynamic damping
Kp2 =  -0.142837E-01 [rad/(rad/s)]
Ko1 =        1.28484 [deg], Ko2=6.51226 [deg^2] (dq/domega=-969.52042 kNm/(rad/s))
********************************************
Aerodynamic gains:
********************************************
 (1) theta [deg] (2) dq/dtheta [kNm/deg] (3) fit [kNm/deg]
 (4) dq/domega [kNm/(rad/s)] (5) fit [kNm/(rad/s)]
       0.00000   -1166.90939   -1184.66324    -410.28617    -969.52037
       4.10000   -1667.04713   -1656.59407   -6942.70178   -6565.93013
```

```
 8.62000   -2291.02335   -2276.23342  -18931.40061  -18536.20107
11.74000   -2757.48062   -2764.73921  -30204.98249  -30347.59595
14.38000   -3213.72579   -3216.87059  -42425.17141  -42605.72279
15.59000   -3428.58517   -3435.97787  -48638.86924  -48917.57026
17.88000   -3858.51374   -3871.09235  -61710.31343  -62056.37484
20.03000   -4295.32311   -4303.95077  -75704.27987  -75813.10504
22.05000   -4758.62584   -4732.11824  -90638.23374  -89992.07594
```

Note that some lines in the above example are truncated or broken over multiple lines to fit the text width of this page.

These tuning parameters can be used directly for the Basic DTU Wind Energy controller with HAWC2 as is shown in table 3.2.

Table 3.2.: Controller tuning parameters for the Basic DTU Wind Energy controller.

HAWC2	Var	Units	Reg	Description
`constant 11`	`K`	kNm/(rad/s)^2	1	Optimal C_P tracking `K` factor
`constant 12`	`Kp`	Nm/(rad/s)	2	Proportional gain of torque controller
`constant 13`	`Ki`	Nm/rad	2	Integral gain of torque controller
`constant 16`	`Kp`	rad/(rad/s)	3	Proportional gain of pitch controller
`constant 17`	`Ki`	rad/rad	3	Integral gain of pitch controller
`constant 21`	`K1`	deg	3	Coefficient of linear term in aerodynamic gain scheduling
`constant 22`	`K2`	deg^2	3	Coefficient of quadratic term in aerodynamic gain scheduling

3.4. Performance data in .pwr

This file is generated with the command `save_power`. In the file each row corresponds to an operational point and each column refers to a parameter or computed result, as reported in table 3.3.

The derivatives marked by * are only saved if the option to compute the aerodynamic gradients is selected in the dialog window of the computation of the steady states. The gradients are either assuming an instantly updated wake corresponding the gradients on the C_P and C_T surfaces, or assuming frozen wake where the induced velocities are kept constant.

Table 3.3.: Description of the .pwr file.

#	Name	Description
1	`V`	Wind speed [m/s]
2	`P`	Aerodynamic power [kW]
3	`T`	Aerodynamic thrust [KN]
4	`Cp`	Power coefficient [-]
5	`Ct`	Thrust coefficient [-]
6	`Pitch Q`	Pitch torque [kNm]
7	`Flap M`	Hub root out-of-plane bending moment [kNm]
8	`Edge M`	Hub root in-plane bending moment [kNm]
9	`Pitch`	Pitch angle [deg]
10	`Speed`	Rotor speed [rpm]
11	`Tip x`	In-plane tip position relative to the rotor center. By looking upwind, it is positive towards the left. [m]
12	`Tip y`	Out-of-plane tip position relative to the rotor center, positive downwind. [m]
13	`Tip z`	Radial tip position relative to the rotor center, positive towards the blade tip. [m]
14	`J_rot`	Rotor inertia [kg m^2]
15	`J_DT`	Inertia of entire drivetrain including rotor [kg m^2]
16*	`dQ/dt`	Wake updated: Aero. torque gain of pitch angle change [kNm/deg]
17*	`dQ/dV`	Wake updated: Aero. torque gain of wind speed change [kNs]
18*	`dQ/dO`	Wake updated: Aero. torque gain of rotor speed change [kNm/rpm]
19*	`dT/dt`	Wake updated: Aero. thrust gain of pitch angle change [kN/deg]
20*	`dT/dV`	Wake updated: Aero. thrust gain of wind speed change [kNs/m]
21*	`dT/dO`	Wake updated: Aero. thrust gain of rotor speed change [kN/rpm]
22*	`dQ/dt`	Frozen wake: Aero. torque gain of pitch angle change [kNm/deg]
23*	`dQ/dV`	Frozen wake: Aero. torque gain of wind speed change [kNs]
24*	`dQ/dO`	Frozen wake: Aero. torque gain of rotor speed change [kNm/rpm]
25*	`dT/dt`	Frozen wake: Aero. thrust gain of pitch angle change [kN/deg]
26*	`dT/dV`	Frozen wake: Aero. thrust gain of wind speed change [kNs/m]
27*	`dT/dO`	Frozen wake: Aero. thrust gain of rotor speed change [kN/rpm]
28	`Tors.`	Projection of the total tip rotation on the pitch axis [rad], positive to stall
29	`Torque`	Aerodynamic torque [kNm]

3.5. Spanwise steady-state results in .ind

This file is generated with the command `save_induction`. One file for each operational point is saved. The files contain a matrix where each row corresponds to a spanwise aerodynamic station on the last main body of the `rotating_threebladed_substructure`. The columns are specified in table 3.4.

Additional files with `_fext_uXXX` inserted in the file names are also saved with the extension .ind. These files contain the spanwise distributions of the external (aerodynamic) forces and moments on the last main body of the `rotating_threebladed_substructure`. The columns are specified in the table 3.5. Internal forces and moments, such as the inertial loading, are not included in this output.

Additional files with `_defl_uXXX` inserted in the file names are also saved with the extension .ind. These files contain the spanwise distributions of the nodal positions and deformations of the elements on the last main body of the `rotating_threebladed_substructure`. The columns are specified in the table 3.6.

Table 3.4.: Description of the .ind file.

#	Name	Description
1	s [m]	Curvilinear coordinate
2	A [-]	Axial induction factor
3	AP [-]	Tangential induction factor
4	PHI0 [rad]	Inflow angle in rotor plane coordinates
5	ALPHA0 [rad]	Angle of attack
6	U0 [m/s]	Relative wind speed
7	FX0 [N/m]	Aerodynamic force in rotor plane coordinates (in-plane)
8	FY0 [N/m]	Aerodynamic force in rotor plane coordinates (out-of-plane)
9	M0 [Nm/m]	Aerodynamic moment in rotor plane coordinates
10	UX0 [m]	In-plane deflection of aero. center relative to rotor center
11	UY0 [m]	Out-of-plane deflection of aero. center relative to rotor center
12	UZ0 [m]	Radial deflection of aero. center relative to rotor center
13	Twist [rad]	Static chord twist including pitch, positive to feather
14	X_AC0 [m]	In-plane position of aero. center relative to rotor center
15	Y_AC0 [m]	Out-of-plane position of aero. center relative to rotor center
16	Z_AC0 [m]	Radial position of aero. center relative to rotor center
17	CL0 [-]	Lift coefficient, interpolated from airfoil polar at ALPHA0
18	CD0 [-]	Drag coefficient, interpolated from airfoil polar at ALPHA0
19	CM0 [-]	Moment coefficient, interpolated from airfoil polar at ALPHA0
20	CLp0 [1/rad]	Slope of lift coefficient
21	CDp0 [1/rad]	Slope of drag coefficient
22	CMp0 [1/rad]	Slope of moment coefficient
23	F0 [-]	Steady value of the separation function
24	F'[1/rad]	Slope of the separation function
25	CL_FS0 [-]	Lift coefficient of the fully separated lift curve
26	CLFS'[1/rad]	Slope of the fully separated lift coefficient
27	V_a [m/s]	Axial induced velocity
28	V_t [m/s]	Tangential induced velocity
29	Tors. [rad]	Projection of the total rotation on the pitch axis, positive to stall
20	vx [m/s]	Relative inflow in chord reference system (chordwise)
31	vy [m/s]	Relative inflow in chord reference system (normal)
32	chord [m]	Chord
33	CT [-]	Thrust coefficient
34	CP [-]	Power coefficient
35	angle [rad]	Angle describing together with columns 36–38 the complete rotation of the chord coordinate system from the undeformed blade
36	v1 [-]	In-plane vector comp. related to the rotation angle in column 35
37	v2 [-]	Out-of-plane vector comp. related to the rot. angle in column 35
38	v3 [-]	Radial vector comp. related to the rotation angle in column 35
39	CL_dyn0 [-]	Lift coefficient, including torsion rate and added mass terms
40	CD_dyn0 [-]	Drag coefficient, including torsion rate and added mass terms
41	CM_dyn0 [-]	Moment coefficient, including torsion rate and added mass terms

Table 3.5.: Description of the _fext_uXXX.ind file.

#	Name	Description
1	s [m]	Curvilinear coordinate
2	Node [-]	Node number (node 1 is the blade flange)
3	Fx_e [N]	Edgewise force in element coordinates
4	Fy_e [N]	Flapwise force in element coordinates
5	Fz_e [N]	Spanwise force in element coordinates
6	Mx_e [Nm]	Flapwise moment in element coordinates
7	My_e [Nm]	Edgewise moment in element coordinates
8	Mz_e [Nm]	Torsional moment in element coordinates
9	Fx_r [N]	Edgewise force in rotor coordinates
10	Fy_r [N]	Flapwise force in rotor coordinates
11	Fz_r [N]	Spanwise force in rotor coordinates
12	Mx_r [Nm]	Flapwise moment in rotor coordinates
13	My_r [Nm]	Edgewise moment in rotor coordinates
14	Mz_r [Nm]	Torsional moment in rotor coordinates

Table 3.6.: Description of the _defl_uXXX.ind file.

#	Name	Description
1	s [m]	Curvilinear coordinate
2	Element no [-]	Element number
3	pos_xR [m]	In-plane position of element origo relative to the blade flange (origo)
4	pos_yR [m]	Out-of-plane position of element origo relative to the blade flange (origo)
5	pos_zR [m]	Radial position of element origo relative to the blade flange (origo)
6	Elem angle [rad]	Angle describing together with columns 7–9 the complete rotation of the element coordinate system from the undeformed blade
7	Elem v_1 [-]	In-plane vector comp. related to the rotation angle in column 6
8	Elem v_2 [-]	Out-of-plane vector comp. related to the rot. angle in column 6
9	Elem v_3 [-]	Radial vector comp. related to the rotation angle in column 6
10	Node 1 angle [rad]	Angle describing together with columns 11–13 the complete rotation of the first element node relative to the element coordinate system
11	Node 1 v_1 [-]	In-plane vector comp. related to the rotation angle in column 10
12	Node 1 v_2 [-]	Out-of-plane vector comp. related to the rot. angle in column 10
13	Node 1 v_3 [-]	Radial vector comp. related to the rotation angle in column 10
14	Node 2 angle [rad]	Angle describing together with columns 15–17 the complete rotation of the second element node relative to the element coordinate system
15	Node 2 v_1 [-]	In-plane vector comp. related to the rotation angle in column 14
16	Node 2 v_2 [-]	Out-of-plane vector comp. related to the rot. angle in column 14
17	Node 2 v_3 [-]	Radial vector comp. related to the rotation angle in column 14
18	Elongation [m]	Elongation of the element

3.6. Frequencies and damping ratios in .cmb

The files with these extensions contain results from eigenvalues analysis. Depending on what the user selects these results can be from structural, aeroelastic, open-loop and closed-loop analyses. In the file each line refers to an operational point.

For the structural eigenanalysis there are $1 + 2N$ columns, where N refers to the number of modes. The first column refers to the rotor speed, the following N columns refer to the damped frequencies and the last N columns refer to the respective damping ratios. For the aeroelastic analysis, the first column contains instead the wind speed.

In case of an aero- or aeroservo-elastic analysis (for both blade only and turbine), an additional N columns are added (after the $1 + 2N$ columns for frequencies and damping) referring to the real part of the eigenvalues. As a result, the output file will now contain $1 + 3N$ columns.

By indicating with λ an eigenvalue, the damped frequency is $\mathrm{Im}(\lambda)$ and the damping ratio is $-\mathrm{Re}(\lambda)/|\lambda|$.

3.7. Mode shapes in .amp

The file contains modal amplitudes and phases, either of the blade or of the complete turbine. In the latter case, only a selection of degrees of freedom is written.

3.7.1. Blade-only analysis

In this case the file has three header lines, followed by m matrices separated by blank lines, where m is the number of values of the scheduling variable. The header lines are:

1. Mode to which each column of the file is associated. It goes from 1 to N, where N is the number of computed modes.
2. Number of each column. For the original beam model, it goes from 1 to $2 + 3 \cdot 2 \cdot N$, while for the FPM one is goes till $2 + 6 \cdot 2 \cdot N$
3. Description of each column.

The first column contains the scheduling variable. It can be either the rotor speed, in rad/s, or the wind speed, in m/s. The second column contains the radial station of each node. The subsequent columns contain the amplitude and phase of the degrees of freedom for each mode. The components for the original beam model are listed in table 3.7, while the ones for the FPM model are listed in table 3.8.

3.7.2. Full system analysis

In this case the file contains only one matrix, preceded by five header lines with the following content:

Table 3.7.: Description of the columns in the .amp file, for the blade-only analysis, with the original beam model.

Name	Description
u_x bld [m], phase_x [deg]	Edgewise component, parallel to the pitched chord at a radial station with zero twist and zero torsional steady state deflection. It is positive towards the leading edge.
u_y bld [m], phase_y [deg]	Flapwise component, perpendicular to the pitched chord at a radial station with zero twist and zero torsional steady state deflection. It is positive towards the suction side, i.e. downwind.
theta [rad], phase_t [deg]	Torsional component. It is positive nose up, i.e. towards positive stall.

Table 3.8.: Description of the columns in the .amp file, for the blade-only analysis, with the FPM beam model.

Name	Description
u_x bld [m], phase_x [deg]	Edgewise component, parallel to the pitched chord at a radial station with zero twist and zero torsional steady state deflection. It is positive towards the leading edge.
u_y bld [m], phase_y [deg]	Flapwise component, perpendicular to the pitched chord at a radial station with zero twist and zero torsional steady state deflection. It is positive towards the suction side, i.e. downwind.
u_z bld [m], phase_z [deg]	Axial component. It is positive towards the tip.
theta_x [rad], phase_tx [deg]	Rotation around the x axis.
theta_y [rad], phase_ty [deg]	Rotation around the y axis.
theta_z [rad], phase_tz [deg]	Rotation around the z axis (torsional component). It is positive nose up, i.e. towards positive stall.

1. Info about the file.
2. The file contains the mode shapes for the last node of the indicated bodies.
3. Mode to which each column of the file is associated. It goes from 1 to N, where N is the number of computed modes.
4. Number of each column, it normally goes from 1 to $1 + 15 \cdot 2 \cdot N$. But, if the model include super elements, it goes till $1 + 18 \cdot 2 \cdot N$.
5. Description of each column.

The first column contains the scheduling variable. It can be either the rotor speed, in rad/s, or the wind speed, in m/s. For each mode, the subsequent columns contain the amplitude and phase of the indicated degrees of freedom. The mode shapes of the first bodies (super elements, tower and shaft) are written in physical coordinates (x, y, yaw/torsion), while the ones of the blades are written as symmetric, backward and forward components of edgewise, flapwise and torsion. The last columns are written only if super elements are present. The description of the columns is provided in table 3.9.

Table 3.9.: Description of the columns in the .amp file, for the full system analysis.

Name	Description
TWR x [m], phase [deg]	Side-Side component of the tower top, positive towards right when looking upwind.
TWR y [m], phase [deg]	Fore-Aft component of the tower top, positive downwind.
TWR yaw [rad], phase [deg]	Yaw component of the tower top, positive clockwise when looking down.
SFT x [m], phase [deg]	Horizontal component of the shaft tip, positive towards right when looking upwind.
SFT y [m], phase [deg]	y component of the shaft tip (vertical with zero tilt angle), positive downwards.
SFT tor [rad], phase [deg]	Torsional component of the shaft tip, positive clockwise when looking upwind.
Sym edge [m], phase [deg]	Symmetric edgewise component of the blade tip, parallel to the pitched chord at a radial station with zero twist and zero torsional steady state deflection (pitched blade root coordinate system). Positive towards the leading edge.
BW edge [m], phase [deg]	Backward edgewise component of the blade tip, parallel to the pitched chord at a radial station with zero twist and zero torsional steady state deflection (pitched blade root coordinate system). Positive towards the leading edge.
FW edge [m], phase [deg]	Forward edgewise component of the blade tip, parallel to the pitched chord at a radial station with zero twist and zero torsional steady state deflection (pitched blade root coordinate system). Positive towards the leading edge.
Sym flap [m], phase [deg]	Symmetric flapwise component of the blade tip, perpendicular to the pitched chord at a radial station with zero twist and zero torsional steady state deflection (pitched blade root coordinate system). Positive towards the suction side, i.e. downwind.
BW flap [m], phase [deg]	Backward flapwise component of the blade tip, perpendicular to the pitched chord at a radial station with zero twist and zero torsional steady state deflection (pitched blade root coordinate system). Positive towards the suction side, i.e. downwind.
FW flap [m], phase [deg]	Forward flapwise component of the blade tip, perpendicular to the pitched chord at a radial station with zero twist and zero torsional steady state deflection (pitched blade root coordinate system). Positive towards the suction side, i.e. downwind.
Sym tors [rad], phase [deg]	Symmetric torsional component of the blade tip. Positive nose up, i.e. towards positive stall.
BW tors [rad], phase [deg]	Backward torsional component of the blade tip. Positive nose up, i.e. towards positive stall.
FW tors [rad], phase [deg]	Forward torsional component of the blade tip. Positive nose up, i.e. towards positive stall.

3.8. System matrices

When saving the system matrices the following files are generated, depending on the command selected.

3.8.1. Structural matrices

- `tm_mat` Structural mass matrix.
- `tc_mat` Damping matrix.
- `tk_mat` Stiffness matrix.
- `phi_mat` Transformation matrix to reduce the system.
- `vtmtotv` Reduced structural mass matrix.
- `vtctotv` Reduced structural damping matrix.
- `vtktotv` Reduced structural stiffness matrix.

3.8.2. Open-loop matrices

Corresponding to the state and output equation:

$$\begin{aligned} \dot{x} &= Ax + B_u u + B_v v \\ y &= Cx + D_u u + D_v v \end{aligned} \tag{3.1}$$

- `amat` Open-loop A matrix.
- `bmat` Open-loop B_u matrix, with given input channels (possibly for controller).
- `bvmat` Open-loop B_v matrix, input from uniform wind in three components, collective, cosine, and sine.
- `bvmat_loc_v` Open-loop B_v matrix, input from wind at each aerodynamic section along the blade in three components, collective, cosine, and sine.
- `cmat` Open-loop C matrix.
- `dmat` Open-loop D_u matrix, with given input channels (possibly for controller).
- `dvmat` Open-loop D_v matrix, input from uniform wind.
- `dvmat_loc_v` Open-loop D_v matrix, input from wind at each aerodynamic section along the blade in three components, collective, cosine, and sine.
- `gmat` Static gain matrix, based on given input channels $G = -CA^{-1}B_u + D_u$.
- `gvmat` Static gain matrix, based on uniform wind in three components $G = -CA^{-1}B_v + D_v$.
- `gvmat_loc_v` Static gain matrix, based on uniform wind each aerodynamic section $G = -CA^{-1}B_v + D_v$.

3.8.3. Closed-loop matrices

Corresponding to the state and output equation:

$$\begin{aligned} \dot{x} &= Ax + B_u u_{\text{pert}} + B_v v \\ y &= Cx + D_u u_{\text{pert}} + D_v v \\ z_{\text{all}} &= Ex + F_u u_{\text{pert}} + F_v v \end{aligned} \tag{3.2}$$

with $y = \left[z_{\mathrm{ctrl}}^T, u_{\mathrm{ctrl}}^T\right]^T$.

- `amat_ase` Closed-loop A matrix.
- `bmat_ase` Closed-loop B_u matrix, from perturbation on the input signals, for all the inputs specified, either used by the controller or not.
- `bvmat_ase` Closed-loop B_v matrix, input from uniform wind in three components, collective, cosine, and sine.
- `bvmat_loc_v_ase` Closed-loop B_v matrix, input from wind at each aerodynamic section along the blade in three components, collective, cosine, and sine.
- `cmat_ase` Closed-loop C matrix. From aero-servo-elastic states to y output, which includes the outputs used by the controller, and the input signals returned by the closed-loop controller u_{ctrl}.
- `dmat_ase` Closed-loop D_u matrix, input from the controller.
- `dvmat_ase` Closed-loop D_v matrix, input from uniform wind.
- `dvmat_loc_v_ase` Closed-loop D_v matrix, input from wind at each aerodynamic section along the blade in three components, collective, cosine, and sine.
- `emat_ase` Closed-loop E matrix. From aero-servo-elastic states to all outputs.
- `fmat_ase` Closed-loop F_u matrix. Direct term from perturbation input to all outputs.
- `fvmat_ase` Closed-loop F_v matrix. Direct term from wind input to all outputs. Wind in three components: collective, cosine, and sine.
- `fvmat_loc_v_ase` Closed-loop F_v matrix. Direct term from local wind input to all outputs. Wind input for each aerodynamic section along the blade, in the three components: collective, cosine, and sine.

4. Examples

In this chapter a few examples on how to use the program are shown.

4.1. Examples with the GUI: HAWCStab2.exe

In this section a small example on how to use HS2 is shown.

Assuming that no prior calculations are performed, the first thing to calculate is operational points for different wind speeds. When opening the desired htc file under

`File->Open HAWC2 model file...`

HS2 will produce an error because the *operational_data_filename* file does not exist. This should be ignored by pressing ok on the error dialog box.

4.1.1. Calculating operational points

Normal operation

The first step is to create the *operational_data_filename*. This is done under

`Computation->Optimal operational data`

A dialogue box will appear where the user is required to fill various information. If the htc file contains the following

```
begin operational_data ;
  windspeed 4.0 25.0 22 ; cut-in [m/s], cut-out [m/s], points [-]
  genspeed 300.0 480.0 ; gen. speed. min. [rpm], gen. speed. max. [rpm]
  gearratio 50.0 ; [-]
  minpitch 0.0 ; [deg.]
  opt_lambda 7.5 ; [-]
  maxpow 10638.3 ; [kW]
  prvs_turbine 1 ; [-] 0 Fixed pitch, 1 PRVS, 2 SRVS
  include_torsiondeform 1 ; [-]
end operational_data ;
```

then the default values in the dialogue box are replaced by the values given by the htc file.

Once the computations have been performed the user should save the computed data. This is done under

```
File->Save optimal power data
```

The saved data file should be named to match the file name specified by *operational_data_filename*.

The Gnuplot code found in Listings A.1 has been used to generate Fig. 4.1.

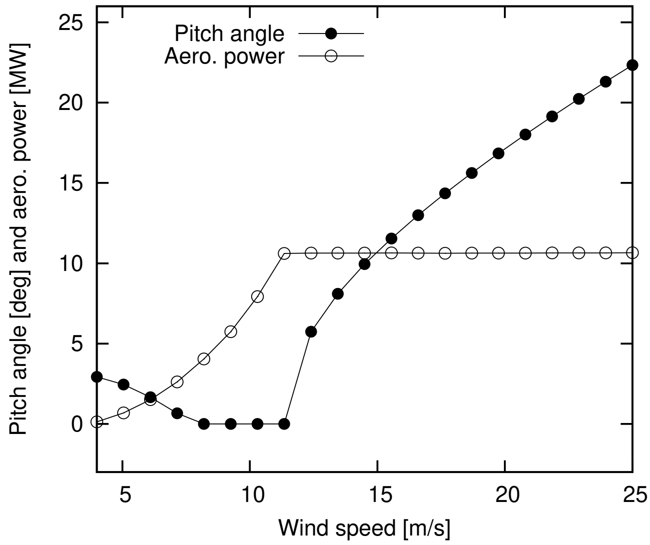


Figure 4.1.: Steady state power and pitch angle values.

Run away operation to identify flutter speed

A run away situation at fixed fine (zero) pitch is one way to determine the critical flutter speed of a wind turbine, [10]. In this situation no generator torque is applied and the rotor is free to rotate. With increasing rotor speed, the angle of attack along the blade is decreasing for a fixed wind speed until a terminal rotor speed is reached where the aerodynamic power is zero. To achieve a higher terminal rotor speed, the wind speed has to be increased. The advantage of this approach in comparison to spinning up the rotor in zero wind is that the generator torque is zero. Thus no unrealistically large force has to be applied on the turbine and the edgewise deflections are typically small. Large edgewise deflections would change the flap- torsion coupling and lead to a different and unrealistic critical rotor speed.

For run away stability analysis in HAWCStab2, the operating points can be calculated for pitch angles of 0 degrees and no generator torque, i.e. max power equal to 0 kW. Furthermore, fixed pitch is selected in the dialog box. The wind speed range being examined is typically from e.g. 6 to 12 m/s with e.g. 13 points.

```
begin operational_data ;
  windspeed 6.0 12.0 13 ; cut-in [m/s], cut-out [m/s], points [-]
```

```
    genspeed 300.0 480.0 ; gen. speed. min. [rpm], gen. speed. max. [rpm]
    gearratio 50.0 ; [-]
    minpitch 0.0 ; [deg.]
    opt_lambda 7.5 ; [-]
    maxpow 0.0 ; [kW]
    prvs_turbine 0 ; [-] 0 Fixed pitch, 1 PRVS, 2 SRVS
    include_torsiondeform 1 ; [-]
    operational_data_file_wind 0 ; [-]
  end operational_data ;
```

Once these operating points have been found an aeroelastic stability analysis can be performed for the specific operating conditions. This stability analysis will show at which critical rotor speed a mode becomes negatively damped. An advantage of performing this analysis in frequency domain is that a critical flutter speed (where the damping can become strongly negative) can be found even if there is for example an edgewise mode with a slightly negative damping at a rotor speed closer to rated. This slightly negatively damped mode would cause vibrations to slowly build up in a time domain analysis that might obscure an actual flutter mode at higher rpm.

4.1.2. Calculating steady state and induction

First ensure that the steps found in Sec. 4.1.1 have been performed. Then

```
Compute->Steady state and induction
```

should be chosen. Afterwards further analysis can be performed.

Using

```
File->Save power...
```

to produce *def.pwr* provides steady state value for power, pitch angle, blade tip deflections etc. The Gnuplot code found in Listings A.2 has been used to generate Fig. 4.2, where flapwise and edgewise tip deflections are shown.

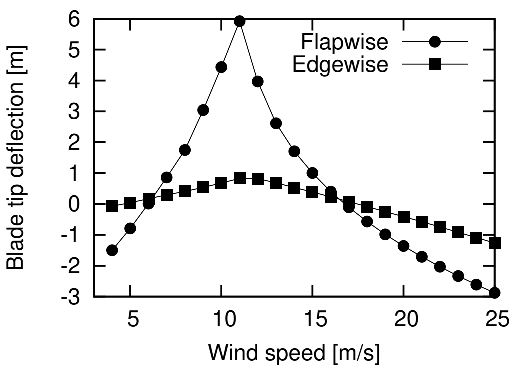


Figure 4.2.: Steady state blade tip deflections.

Using

`File->Save steady state...`

to produce multiple files *opt_u*.ind*, preferably in a dedicated folder, for various wind speeds provide an extended number of steady state values. The Gnuplot code found in Listings A.3 has been used to generate Fig. 4.3, where the torsion of the blade along the blade span for various wind speed is seen. Steady state pitch values has been added to the total torsion of the blade to get the shown plots.

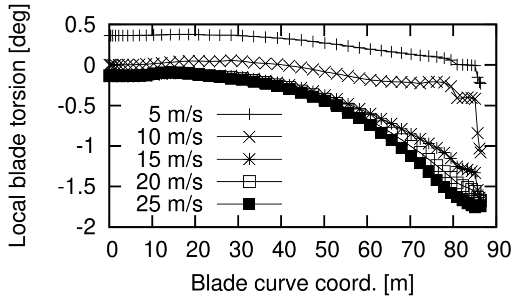


Figure 4.3.: Steady state blade torsion.

4.1.3. Performing open-loop aeroelastic modal analysis

First, ensure that the steps found in Sec. 4.1.2 have been performed.

Selecting

`Compute->Structural modal analysis->Entire turbine`

will compute the structural modes. This calculation is required to perform the

`Compute->Aeroelastic modal analysis->Entire turbine`

The sort the modes, the following values was used: (0.01,0.30,0.50Hz,0.1,8,sort after mode shapes)

Results obtained from the analysis can be saved under

`File->Save modal amplitudes`

as e.g. *turbine_ae.cmb*.

The Gnuplot code found in Listings A.4 and A.5 has been used to generate Fig. 4.4.

4.1.4. Tuning of PI controller

Selecting

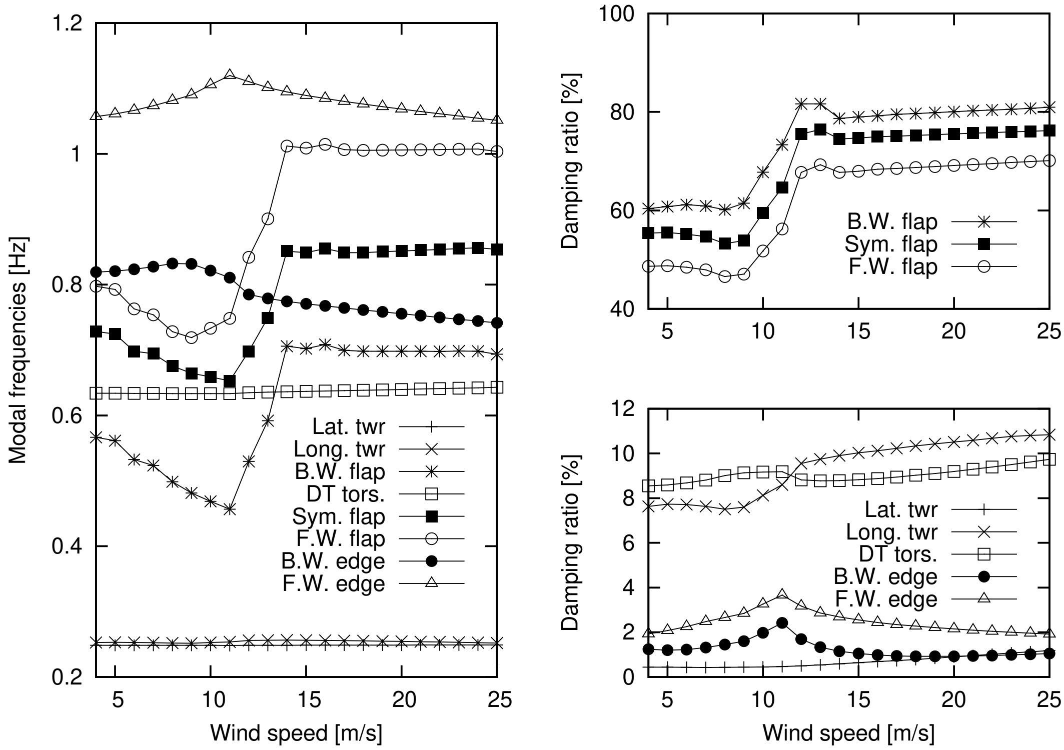


Figure 4.4.: Open-loop modal frequencies and damping ratios.

```
Compute->Tune pitch controller by DTU Wind Energy
```

A dialogue box will appear where the user is required to fill various information. If the htc file contains the following

```
begin controller_tuning ;
  partial_load 0.05 0.7; fn [hz], zeta [-]
  full_load 0.06 0.7 ; fn [hz], zeta [-]
  gain_scheduling 1 ; 1 linear, 2 quadratic
  constant_power 1 ; 0 constant torque, 1 constant power
end controller_tuning ;
```

then the default values in the dialogue box are replaced by the values given by the htc file.

The computations produces *controller_input.txt*, which can be used with the Basic DTU Wind Energy controller [6].

4.1.5. Performing closed-loop aeroelastic modal analysis

To perform a closed-loop analysis several approaches can be used. The first approach is to use one of the two built-in hard coded PI controllers

- basic_dtu_we_controller (# 1)
- pi_pitch_controller (# 2)

where the first can handle full range operation and the second can only handle above rated operation. A description of the parameters is found in Table 2.1, the first two columns in table describe the parameter number for the two controller commands, respectively. Many of the parameters can be calculated by the controller tuning described in Sec. 4.1.4. An example:

```
 pi_pitch_controller 5200 1.2671 0.771100 0.319309 102.68665 754.18745
... 0.6 0.7   1;


 basic_dtu_we_controller 0.19297E+08 0.43304E+07 0.21E+07 1.36516
 ...0.669945 11.63317 553.75769 0.6 0.7 1.622 0 ;
```

Furthermore, the following should be included

```
begin controller ;
  begin input ;
    constraint bearing1 shaft_rot ;
    constraint bearing2 pitch1 collective ;
    constraint bearing2 pitch1 cosine ;
    constraint bearing2 pitch1 sine ;
  end input ;
  begin output ;
    constraint bearing1 shaft_rot 1 only 2 ; 1
    constraint bearing2 pitch1 1 only 1 collective ; 2
    constraint bearing2 pitch1 1 only 1 cosine ; 3
    constraint bearing2 pitch1 1 only 1 sine ; 4
  end output ;
end controller ;
```

The inputs are defining how the wind turbine is controlled. The outputs are defining which sensors the controller is using. The cosine and sine pitch actuators/sensors can be used by an individual pitch controller in the Coleman coordinates.

Additional outputs can be added to the output vector. Those will not be used to close the loop with the controller but they can be used to examine e.g. their transfer functions.

4.2. Examples with the command line program: HS2pid.exe

This program is free but has reduced functionality. Its sole purpose is to provide tuning parameters for a PI controller for the wind turbine. The program is hard coded with blade torsion disabled. If blade torsion is to be included in the analysis HAWC2S.exe is to be used instead.

Procedure for using HS2pid.exe to tune the Basic DTU Wind Energy controller [6].

- The operational parameters should be added to the htc file (sec. 4.1.1).
- The controller data parameters should be added to the htc file (sec. 4.1.4).
- Execute "HS2pid.exe *xxx*.htc" in a MS-DOS command prompt.
- Use the calculated values from *controller_input.txt* to tune the controller in the htc file.

The closed loop frequencies should be below the first tower mode. Thus for a floating wind turbine, very low frequencies has to be selected.

5. Keyboard shortcuts

Shortcut	Action
Shift + x	Rotation about tilt axis
Shift + y	Rotation about yaw axis
Shift + s	Zoom out
Shift + w	Zoom in
Arrow up	Move turbine up (only in turbine view)
Arrow down	Move turbine down (only in turbine view)
Shift + h	Recenter the view
Shift + b	Toggle between blade turbine views
Shift + v	Transparent view
Shift + n	Toggle drawing of nacelle
Shift + a	Decrease amplitude of modal vibration
Shift + q	Increase amplitude of modal vibration
Shift + f	Animate forces due to vibration
Shift + k	Increase speed of modal vibration
Shift + i	Decrease speed of modal vibration
Shift + c	Draw aerodyn. choord. sys.
Shift + e	Draw struct. choord. sys.

Appendix A.

GnuPlot files

Listing A.1: Gnuplot commands used to power and pitch figure.

```
reset
set term post eps soli mono 12
set out 'power_and_pitch.eps
set key at 15,25
set size 0.5,0.6
set xr [4:25]
set yr [-3:26]
set format y '%3g'
set xlabel 'Wind speed [m/s]'
set ylabel 'Pitch angle [deg] and aero. power [MW]'
plot 'operational_data.opt' us 1:2 t 'Pitch angle' w lp pt 7 lt 7, \
     'operational_data.opt' us 1:($4*1e-3) t 'Aero. power' w lp pt 6 lt 7
set term wxt
set out
```

Listing A.2: Gnuplot commands used to deflection figure.

```
reset
set term post eps soli mono 12
set out 'deflec.eps
set key right
set size 0.4,0.4
set xr [3:25]
#set yr [0:11]
set format y '%3g'
set xlabel 'Wind speed [m/s]'
set ylabel 'Blade tip deflection [m]'
plot 'def.pwr' us 1:($12+3.766) t 'Flapwise' w lp pt 7 lt 7, \
     'def.pwr' us 1:($11) t 'Edgewise' w lp pt 5 lt 7
set term wxt
set out
```

Listing A.3: Gnuplot commands used to power and pitch figure.

```
reset
set term post eps soli mono 12
set out 'torsion.eps'
set key left bottom
set size 0.4,0.32
set xr [0:90]
#set yr [-1.5:2]
set format y '%3g'
set xlabel 'Blade curve coord. [m]'
```

```
set ylabel 'Local blade torsion [deg]'
plot 'opt_u5000.ind' us 1:($29*180/pi+0.2477776000E+01) t '5 m/s' w lp pt 1 lt 7, \
     'opt_u10000.ind' us 1:($29*180/pi+0.4800000000E-04) t '10 m/s' w lp pt 2 lt 7,
         \
     'opt_u15000.ind' us 1:($29*180/pi+0.1074508600E+02) t '15 m/s' w lp pt 3 lt 7,
         \
     'opt_u20000.ind' us 1:($29*180/pi+0.1713331300E+02) t '20 m/s' w lp pt 4 lt 7,
         \
     'opt_u25000.ind' us 1:($29*180/pi+0.2234911500E+02) t '25 m/s' w lp pt 5 lt 7
set term wxt
set out
```

Listing A.4: Gnuplot commands used to generate modal frequencies figure.

```
reset
set term post eps soli mono 12
set out 'turbine_frq.eps'
set key at 24,0.6
set size 0.4,0.8
set xr [4:25]
set yr [0.2:1.2]
set format y '%3g'
set xlabel 'Wind speed [m/s]'
set ylabel 'Modal frequencies [Hz]'
plot 'turbine_ae.cmb' us 1:2 t 'Lat. twr' w lp pt 1 lt 7, \
     'turbine_ae.cmb' us 1:3 t 'Long. twr' w lp pt 2 lt 7, \
   'turbine_ae.cmb' us 1:4 t 'B.W. flap' w lp pt 3 lt 7, \
     'turbine_ae.cmb' us 1:5 t 'DT tors.' w lp pt 4 lt 7, \
     'turbine_ae.cmb' us 1:6 t 'Sym. flap' w lp pt 5 lt 7, \
     'turbine_ae.cmb' us 1:7 t 'F.W. flap' w lp pt 6 lt 7, \
     'turbine_ae.cmb' us 1:8 t 'B.W. edge' w lp pt 7 lt 7, \
     'turbine_ae.cmb' us 1:9 t 'F.W. edge' w lp pt 8 lt 7
set term wxt
set out
```

Listing A.5: Gnuplot commands used to generate modal damping ratios figure.

```
reset
set term post eps soli mono 12
set out 'turbine_dmp.eps'
set xr [4:25]
set multiplot
set size 0.4,0.39
set orig 0,0.42
set format x '%g'
set format y '%3.0f'
set xlabel ''
set yr [40:100]
set ytics 40,20,100
set ylabel 'Damping ratio [%]'
set key at 24,60
plot 'turbine_ae.cmb' us 1:12 t 'B.W. flap' w lp pt 3 lt 7, \
     'turbine_ae.cmb' us 1:14 t 'Sym. flap' w lp pt 5 lt 7, \
     'turbine_ae.cmb' us 1:15 t 'F.W. flap' w lp pt 6 lt 7
set size 0.4,0.39
set orig 0,0
set format x '%g'
```

```
set format y '%3.0f'
set xlabel 'Wind speed [m/s]'
set ylabel 'Damping ratio [%]'
set key at 24,8
set ytics 0,2,12
set yr [0:12]
plot 'turbine_ae.cmb' us 1:10 t 'Lat. twr' w lp pt 1 lt 7, \
     'turbine_ae.cmb' us 1:11 t 'Long. twr' w lp pt 2 lt 7, \
     'turbine_ae.cmb' us 1:13 t 'DT tors.' w lp pt 4 lt 7, \
     'turbine_ae.cmb' us 1:16 t 'B.W. edge' w lp pt 7 lt 7, \
     'turbine_ae.cmb' us 1:17 t 'F.W. edge' w lp pt 8 lt 7
unset multiplot
set term wxt
set out
```

Bibliography

[1] Hansen MH. Aeroelastic stability analysis of wind turbines using an eigenvalue approach. *Wind Energy* 2004; **7**(2):133–143, doi:10.1002/we.116.

[2] Hansen MH. Aeroelastic properties of backward swept blades. *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics, 2011, doi:10.2514/6.2011-260.

[3] Sønderby I, Hansen MH. Open-loop frequency response analysis of a wind turbine using a high-order linear aeroelastic model. *Wind Energy* 2014; **17**: 1147–1167, doi:10.1002/we.1624.

[4] Larsen TJ, Hansen MA. How 2 HAWC2, the user's manual. *Technical Report Risø-R-1597(ver. 3-1)(EN)*, Risø National Laboratory, 2007. www.hawc2.dk

[5] Hansen MH. Anisotropic damping of Timoshenko beam elements. *Technical Report Risø–R–1267(EN)*, Risø National Laboratory, Denmark, 2001.

[6] Hansen MH, Henriksen LC. Basic DTU Wind Energy controller. *Technical Report E-0028*, DTU Wind Energy, 2013.

[7] Tibaldi C, Henriksen LC, Hansen MH, Bak C. Effects of gain-scheduling methods in a classical wind turbine controller on wind turbine aero-servo-elastic modes and loads. *32nd ASME Wind Energy Symposium*. American Institute of Aeronautics and Astronautics, 2014, doi:10.2514/6.2014-0873.

[8] Hansen MH, Gaunaa M and Madsen HAa, A Beddoes-Leishman type dynamic stall model in state-space and indicial formulations, *Risø-R-1354*, 2004

[9] Sørensen NN and Madsen MAa, Modelling of transient wind turbine loads during pitch motion, *Proceedings European Wind Energy Conference and Exhibition*, 2006

[10] Pirrung G.R., Madsen H.Aa. and Kim T., The influence of trailed vorticity on flutter speed estimations, *Proceedings of the Science of Making Torque from Wind*, 2014

DTU Wind
Department of Wind and Energy Systems
Technical University of Denmark

RisøCampus Building 118
Frederiksborgvej 399
DK-4000 Roskilde
https://wind.dtu.dk/