

# HAWCStab2 User Manual

Department of  
Wind Energy  
Report 2015

Lars Christian Henriksen, Carlo Tibaldi, and Leonardo Bergami

DTU Wind Energy

September 2015



**Author(s):** Lars Christian Henriksen, Carlo Tibaldi, and Leonardo Bergami  
**Title:** HAWCStab2: User Manual  
**Institute:** Department of Wind Energy

**Summary (max. 2000 char.):**

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 aerodynamic. 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:**  
September 2015

**HAWCStab2 version:**  
2.11

**E-mail:**  
[hawcstab2@vindenergi.dtu.dk](mailto:hawcstab2@vindenergi.dtu.dk)

**Web-page:**  
[www.hawcstab2.vindenergi.dtu.dk](http://www.hawcstab2.vindenergi.dtu.dk)

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

# 1. 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

<b>1. Preface</b>	<b>4</b>
<b>2. Introduction</b>	<b>6</b>
<b>3. Input file and commands</b>	<b>7</b>
3.1. HAWC2 commands . . . . .	7
3.1.1. new_htc_structure . . . . .	7
3.1.2. wind . . . . .	8
3.1.3. aero . . . . .	8
3.2. HAWCStab2 commands . . . . .	9
3.3. HAWC2S commands . . . . .	13
3.4. Advanced options . . . . .	15
3.5. Functionalities description . . . . .	16
3.5.1. Damping . . . . .	16
3.5.2. Computation of operational data . . . . .	16
3.5.3. Mode sorting . . . . .	17
<b>4. Output files</b>	<b>19</b>
4.1. .pwr . . . . .	19
4.2. .ind . . . . .	20
4.3. .cmb . . . . .	20
4.4. System matrices . . . . .	21
4.4.1. Structural matrices . . . . .	21
4.4.2. Open-loop matrices . . . . .	21
4.4.3. Closed-loop matrices . . . . .	22
<b>5. Examples</b>	<b>23</b>
5.1. Examples with the GUI: HAWCStab2.exe . . . . .	23
5.1.1. Calculating operational points . . . . .	23
5.1.2. Calculating steady state and induction . . . . .	25
5.1.3. Performing open-loop aeroelastic modal analysis . . . . .	26
5.1.4. Tuning of PI controller . . . . .	27
5.1.5. Performing closed-loop aeroelastic modal analysis . . . . .	27
5.2. Examples with the command line program: HS2pid.exe . . . . .	29
<b>6. Keyboard shortcuts</b>	<b>30</b>
<b>A. GnuPlot files</b>	<b>31</b>

## 2. Introduction

HAWCStab2 is a tool developed at the Department of Wind Energy 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].

In Chapter 3 the basic structure of the htc file is explained. In Chapter 4 the output files are explained. In Chapter 5 a few examples on how to use the program are shown.

The examples shown in this document are based on:

- HAWC2 (version 11.8)
- HAWCStab2 (version 2.11)
- DTU 10MW RWT (version 2.0)

## 3. Input file and commands

The input to HAWCStab2 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.

### 3.1. HAWC2 commands

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

#### 3.1.1. `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 main 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;
```

**Bearings** Two different types of bearings are 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. **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.

To perform an investigation where the rotor speed is kept constant, i.e. a linear model where the rotor speed is not a degree of freedom, the shaft bearing commands in the constraint sub block need to be removed.

### 3.1.2. wind

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

```
begin wind ;
  density          1.225 ;
  wsp              11   ;
  tint             0.20145454545454545   ;
  horizontal_input 1    ;           0=false, 1=true
  windfield_rotations 0 0.0 0.0 ;   yaw, tilt, rotation
  center_pos0      0.0 0.0 -119.00 ; hub height
  shear_format     3 0.2 ;
  turb_format      0    ;   0=none, 1=mann,2=flex
  tower_shadow_method 0    ; 0=none, 1=potential flow, 2=jet
end wind;
```

### 3.1.3. aero

The **aero** block 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
  aerocalc_method 1 ;      0=ingen aerodynamic, 1=med aerodynamic
  arosections      50 ;
  ae_sets          1 1 1;
  tiploss_method  1 ;      0=none, 1=prandtl
```

```

    dynstall_method    2 ;      0=none, 1=stig oye method,2=mhh method
end aero ;

```

Other blocks such as e.g. **simulation**, **aerodrag**,**force**, **hydro**, **soil**, **dll** and outputs are not used by HAWCStab2.

## 3.2. HAWCStab2 commands

HAWCStab2 also uses a HAWCStab2 specific block **hawcstab2** wherein the main bodies in the

### **ground\_fixed\_substructure**

sub block are used for the tower. The main bodies in the

### **rotating\_axissym\_substructure**

sub block are used as the shaft. The main bodies in the

### **rotating\_threebladed\_substructure**

sub block are used for the blade. All the aerodynamic forces are assumed to be applied on the last main body in the block **rotating\_threebladed\_substructure**.

```

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 ;
  operational_data_filename ./operational_data.opt ;
  begin operational_data ;
    ...
  end operational_data ;
  begin controller_tuning ;
    ...
  end controller_tuning ;
  begin controller ;
    ...
  end controller ;
end hawcstab2 ;

```

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.

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

- **windspeed**, the minimum wind speed value, the maximum wind speed value, and the number of wind speed between the minimum and maximum.
- **genspeed**, the minimum rotational speed, and the maximum rotational speed in rpm.
- **gearratio** and the gear 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.
- **include\_torsiondeform** and an integer to indicate if blade deformations should be included in the computation. 0 for no deformations and 1 for with deformations.

The **controller\_tuning** block is optional, see 5.1.4 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. 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.

In HAWCStab2 there are two commands to add to the model two different build-in controllers.

- `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 3.1.

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

Table 3.1.: Parameters for build-in controller commands.

The `operational_data_filename` file contains information about the operational points for selected wind speeds.

22 Wind speed [m/s]	Pitch [deg]	Rot. speed [rpm]	Aero power [kW]	Aero thrust [kN]
4.000000	2.889748	6.000000	287.319260	224.286816
5.000000	2.115800	6.000000	805.573745	352.209828
6.000000	1.109058	6.000000	1543.002742	500.388658
7.000000	0.000048	6.000000	2525.245528	658.232557
8.000000	0.000055	6.424607	3770.277010	816.795864
9.000000	0.000019	7.226938	5374.530562	1034.430140
10.000000	0.000056	8.031337	7378.855371	1277.059791
11.000000	0.000048	8.839966	9826.489718	1544.121977
12.000000	4.807932	9.600000	10636.875545	1262.557036
13.000000	7.388350	9.600000	10640.312112	1080.883000
14.000000	9.289680	9.600000	10634.865878	970.253169
15.000000	10.887191	9.600000	10652.538640	892.739336
16.000000	12.346992	9.600000	10618.567809	828.619150
17.000000	13.672693	9.600000	10631.933899	780.318935
18.000000	14.926127	9.600000	10646.833268	740.590975
19.000000	16.120324	9.600000	10640.981055	706.090035
20.000000	17.268079	9.600000	10646.834596	677.263923
21.000000	18.374175	9.600000	10632.078861	651.469908
22.000000	19.443877	9.600000	10648.917163	630.689690
23.000000	20.484638	9.600000	10622.019335	610.636055
24.000000	21.496839	9.600000	10628.972396	594.608862
25.000000	22.485124	9.600000	10638.692060	580.889756

### 3.3. 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 has to be inserted in the `hawcstab2` section.

The commands available are:

- `compute_optimal_pitch_angle use_operational_data` to compute the operational data points according to the parameters inserted in the `operational_data` block.
- `compute_steady_states` - (Sec. 5.1.2)  
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`<sup>§</sup> - (Sec. 5.1.3)  
Parameters:
  - `nomatrixwriteout` to save system matrices computed during the analysis.
  - `noeigenvaluewriteout`
  - an integer to specify the number of modes.
  - a float to specify the maximum damping ratio.
  - a float to specify the minimum frequency
  - a float to specify the zero pole threshold
  - a float to specify the aerodynamic and deflection ratio
  - `frequencysorting`
  - `bladeonly` or `windturbine`\*
- `compute_aeroservoelastic`<sup>§</sup>  
Parameters:
  - `nomatrixwriteout` to save system matrices computed during the analysis.
  - `noeigenvaluewriteout`

- an integer to specify the number of modes.
- a float to specify the maximum damping ratio.
- a float to specify the minimum frequency
- a float to specify the zero pole threshold
- a float to specify the aerodynamic and deflection ratio
- `frequencysorting`
- `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.
- `save_ol_matrices`<sup>§</sup> - Writes out the open-loop A,B,C,D matrices to text files.
- `save_ol_matrices_full`<sup>§</sup> - Writes out the M,D,K matrices to text files.
- `save_ol_matrices_all`<sup>§</sup> - Writes out both A,B,C,D and M,D,K matrices to text files.
- `save_cl_matrices_all`<sup>§</sup> - 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`<sup>§</sup> (Sec. 5.1.4)  
This command needs the block `controller`.
- `save_power`<sup>§</sup>
- `save_induction`<sup>§</sup>
- `degrees_of_freedom`  
Lock different degrees of freedom and select the inflow model. Parameters:
  - `true` or `false` to specify if the ground fixed substructure is rigid
  - `true` or `false` to specify if the rotating axial symmetric substructure is rigid
  - `true` or `false` to specify if the rotating three bladed substructure is rigid
  - `true` or `false` to set quasi-steady aerodynamic
  - `frozen`, `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 amplitude. Parameters:
  - `aeroelastic` or `structural`
- `save_modal_binary` Save modal results in binary file. Parameters:
  - `aeroelastic` or `structural`

The commands with the symbol § require the commands `compute_steady_states` to be executed first. The commands with the symbol \* are optional.

### 3.4. 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. bem tolerance `1e-7`
  2. bem max iteration `1e4`
  3. bem relaxation `0.02`
  4. aeroelastic tolerance `1e-6`
  5. aeroelastic max iteration `1e3`
  6. aeroelastic relaxation `0`
  7. maximum variation of operating point characteristic in compute optimal operation data (e.g. pitch angle above rated) `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.
  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. tolerance on aerodynamic power for optimal operational point computations  
1e-9.

Example:

```
steady_state_convergence_limits 1e-7 1e4 0.02 1e-6 1e3 0 10.0 5.0 1e-9
```

Figure 3.1 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).

- `print_full_precision`  
Save operational points file with extended precision.

## 3.5. Functionalities description

In this section the main functionalities and modeling options of HAWCStab2 are described.

### 3.5.1. Damping

If `log_decrements` is present in either:

`ground_fixed_substructure`,

`rotating_axissym_substructure` or

`rotating_threebladed_substructure`

then the HAWC2 specific damping commands will be overwritten by this command and a spectral damping model will be used to calculate the damping properties.

Otherwise, a damping model similar to HAWC2 (Rayleigh) will be used. The damping properties will be calculated for the unloaded, standstill wind turbine. N.B. use only 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.

### 3.5.2. Computation of operational data

The computation of the operational data points for a pitch-regulated turbine occurs as follow:

1. The rotor rotational speed is computed according to the user-defined tip-speed-ratio.
2. The user-defined lower and upper rotational speed limits are enforced to the previously computed rotational speeds.

3. The pitch angle at each operational point is computed to maximize the aerodynamic torque or to guarantee the user-defined rated power value.

### 3.5.3. Mode sorting

When performing modal analysis a set of parameters need to be specified by the user to extract the desired modes. Nine different inputs and options are available:

1. **Minimum structural content.** Accept only modes with a norm of the structural component in the eigenvector that is higher than the value specified.
2. **Accept modes with frequency >.** Accept modes with frequency higher than the value specified. This condition is linked to condition #4 with an "or" statement.
3. **Omit modes with frequency <.** Discharge all the modes with frequency lower than the value specified.
4. **Accept modes with damping ratio <.** Accept modes with damping lower than the value specified. This condition is linked to condition #2 with an "or" statement.
5. **Omit modes with damping ratio >.** Discharge all the modes with damping higher than the value specified.
6. **1P freq. exclusion band ratio.** Exclude modes with frequency within a band around the 1P frequency.
7. **Zero pole exclusion threshold.**
8. **Number of modes.** Number of modes to be plotted;
9. **Sort.** Connecting modes at the different operational points according to the natural frequency value, i.e. increasing value, or according to the mode shape, i.e. modes with similar shape will be connected.

A representation of the conditions is shown below:

```
if( (eigenvect_norm > min_struct) and (abs(1-freq/rot_freq) >= 1P_excl_band))
  if( damp < omit_damp_ratio) and ( freq > omit_freq) )
    if( Imag(eigenval) >= zero_pole_excl)
      if( (damp < accept_damp_ratio) or (freq > accept_freq) )
        Mode accepted
      endif
    endif
  endif
else
  if( (Imag(eigenval) >= zero_pole_excl) and (zero_pole_excl != 0))
    Mode accepted
  endif
endif
endif
```

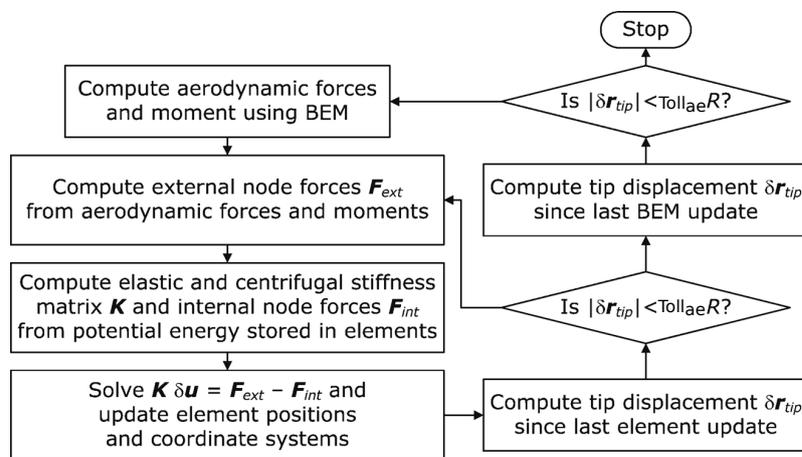


Figure 3.1.: Diagram representing the iterative process to obtain the nonlinear steady-states solution.

## 4. Output files

This chapter lists all the output files that can be generated with HAWCStab2 and it explains their content.

Note that when saving the results files, the extensions suggested in the dialog window have to be added in the file name.

### 4.1. .pwr

This file is generated with the command `save power`. The file contains a matrix where each row corresponds to an operational point and each column to a parameter. The parameters in order are:

#	Name	Description	#	Name	Description
1	V	Wind speed [m/s]	18	dQ/dO	Steady: aero. torque w.r.t. rotor speed [kNm/rpm]
2	P	Aerodynamic power [kW]	19	dT/dt	Steady: aero. thrust w.r.t. pitch angle [kN/deg]
3	T	Aerodynamic thrust [kN]	20	dT/dV	Steady: aero. thrust w.r.t. wind speed [kNs/m]
4	Cp	Power coefficient [-]	21	dT/dO	Steady: aero. thrust w.r.t. rotor speed [kN/rpm]
5	Ct	Thrust coefficient [-]	22	dQ/dt	Quasi-steady: aero. torque w.r.t. pitch angle [kNm/deg]
6	Pitch Q	Pitch torque [kNm]	23	dQ/dV	Quasi-steady: aero. torque w.r.t. wind speed [kNs]
7	Flap M	Hub root out-of-plane bending moment [kNm]	24	dQ/dO	Quasi-steady: aero. torque w.r.t. rotor speed [kNm/rpm]
8	Edge M	Hub root in-plane bending moment [kNm]	25	dT/dt	Quasi-steady: aero. thrust w.r.t. pitch angle [kN/deg]
9	Pitch	Pitch angle [deg]	26	dT/dV	Quasi-steady: aero. thrust w.r.t. wind speed [kNs/m]
10	Speed	Rotor speed [rpm]	27	dT/dO	Quasi-steady: aero. thrust w.r.t. rotor speed [kN/rpm]
11	Tip x	Tip position w.r.t hub [m]			
12	Tip y	Tip position w.r.t hub [m]			
13	Tip z	Tip position w.r.t hub [m]			
14	J_rot	Rotor inertia [kg m <sup>2</sup> ]			
15	J_DT	Drive train inertia [kg m <sup>2</sup> ]			
16	dQ/dt	Steady: aero. torque w.r.t. pitch angle [kNm/deg]			
17	dQ/dV	Steady: aero. torque w.r.t. wind speed [kNs]			

The derivatives are printed only if the option to compute the gradients is selected in the dialog window of the computation of the steady-states.

## 4.2. .ind

This file is generated with the command `save steady state`. One file for each operational point is saved. The files contain a matrix where each row corresponds to a spanwise aerodynamic station and each column to a parameter. The parameters in order are:

#	Name	Description
1	s [m]	Curvilinear choordinate
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]	Force in rotor plane coordinates (lateral)
8	FY0 [N/m]	Force in rotor plane coordinates (axial)
9	M0 [Nm/m]	Moment in rotor plane coordinates
10	UX0 [m]	Change in position of the aero. center due to deformation
11	UY0 [m]	Change in position of the aero. center due to deformation
12	UZ0 [m]	Change in position of the aero. center due to deformation
13	Twist [rad]	Twist
14	X_AC0 [m]	Position of aero. center
15	Y_AC0 [m]	Position of aero. center
16	Z_AC0 [m]	Position of aero. center
17	CL0 [-]	Lift coefficient
18	CD0 [-]	Drag coefficient
19	CM0 [-]	Moment coefficient
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 full separated curve
26	CLFS'[1/rad]	Slope of the full separated lift coefficient
27	V_a [m/s]	Axial induced velocity
28	V_t [m/s]	Tangential induced velocity
29	Tors. [rad]	Torsional deformation
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

## 4.3. .cmb

The files with these extensions can contain results form eigenvalues analysis. Depending on what the user selects these results can be from structural analysis, open-loop

analysis, and closed-loop analysis. The file contain a matrix where each line is an operational point. The first column contain the wind speed values, a second set of column contain the values of the damped frequencies of the modes and the second set of columns the respective damping. Hence if the user selects to compute 15 modes, with the command `aeroelastic modal analysis`, and then saves the file, with the command `save frequencies and damping`, the file will contain 31 columns.

## 4.4. System matrices

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

### 4.4.1. 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.

### 4.4.2. Open-loop matrices

Corresponding to the state and output equation:

$$\begin{cases} \dot{x} &= Ax - Bu + B_V w \\ y &= Cx + Du + D_V w \end{cases} \quad (4.1)$$

- `amat` Open-loop A matrix.
- `bmat` Open-loop B matrix, input from the controller.
- `cmat` Open-loop C matrix.
- `dmat` Open-loop D matrix, input from the controller.
- `bvmat` Open-loop B matrix, input from uniform wind in three components, collective, cosine, and sine.
- `dvmat` Open-loop D matrix, input from uniform wind.
- `bvmat_loc_v` Open-loop B matrix, input from wind at each aerodynamic section along the blade in three components, collective, cosine, and sine.

- `dvmat_loc_v` Open-loop D matrix, input from wind at each aerodynamic section along the blade in three components, collective, cosine, and sine.
- `gmat`
- `gvmat`
- `gvmat_loc_v`

#### 4.4.3. Closed-loop matrices

Corresponding to the state and output equation:

$$\begin{cases} \dot{x} & = Ax - Bu_{\text{pert}} + B_V w \\ y = [z_{\text{ctrl}}^T, u_{\text{ctrl}}^T]^T & = Cx + Du_{\text{pert}} + D_V w \\ z_{\text{all}} & = Ex + Fu_{\text{pert}} + F_V w \end{cases} \quad (4.2)$$

- `amat_ase` Closed-loop A matrix.
- `bmat_ase` Closed-loop B matrix, from perturbation on the input signals, for all the inputs specified, either used by the controller or not.
- `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  $u_{\text{ctrl}}$ .
- `dmat_ase` Closed-loop D matrix, input from the controller.
- `bvmat_ase` Closed-loop B matrix, input from uniform wind in three components, collective, cosine, and sine.
- `dvmat_ase` Closed-loop D matrix, input from uniform wind.
- `bvmat_loc_v_ase` Closed-loop B matrix, input from wind at each aerodynamic section along the blade in three components, collective, cosine, and sine.
- `dvmat_loc_v_ase` Closed-loop D 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 matrix. Direct term from perturbation input to all outputs.
- `fvmat_ase` Closed-loop Fv matrix. Direct term from wind input to all outputs. Wind in three components: collective, cosine, and sine.
- `fvmat_loc_v_ase` Closed-loop Fv 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.

## 5. Examples

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

### 5.1. 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.

#### 5.1.1. 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. 5.1.

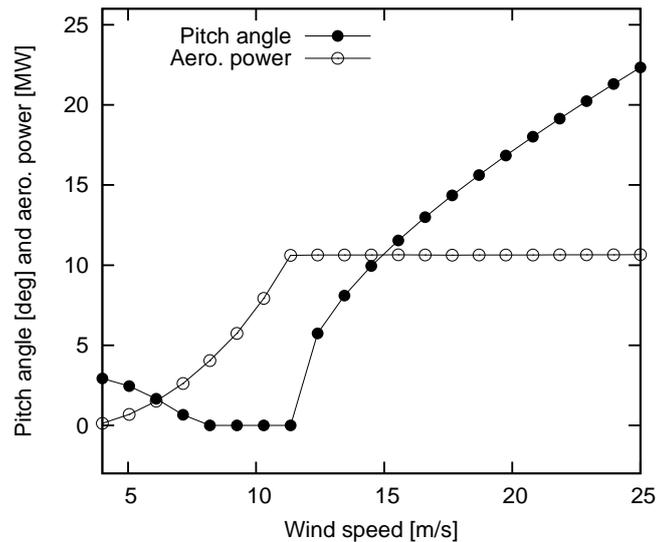


Figure 5.1.: Steady state power and pitch angle values.

### Run away operation

For run away stability analysis the operating points are typically 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. 8 to 12 m/s with e.g. 9 points. Once these operating points have been found a stability analysis can be performed for the specific operating conditions.

```
begin operational_data ;
  windspeed 8.0 12.0 9 ; 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 ; [-]
end operational_data ;
```

### 5.1.2. Calculating steady state and induction

First ensure that the steps found in Sec. 5.1.1 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. 5.2, where flapwise and edgewise tip deflections are shown.

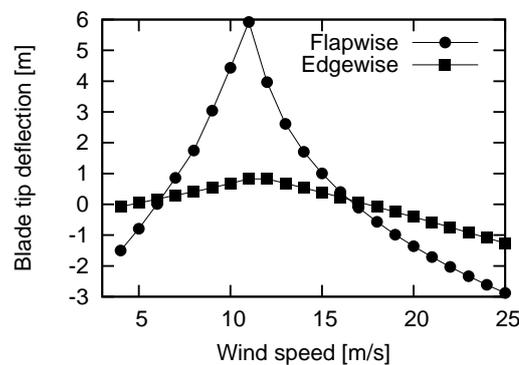


Figure 5.2.: Steady state blade tip deflections.

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. 5.3, 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.

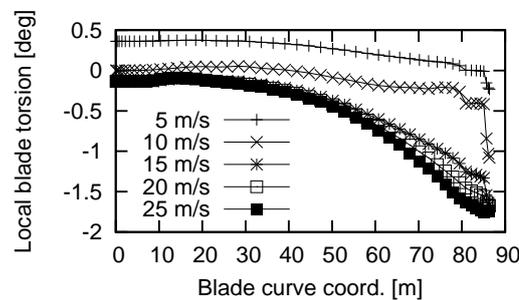


Figure 5.3.: Steady state blade torsion.

### 5.1.3. Performing open-loop aeroelastic modal analysis

First, ensure that the steps found in Sec. 5.1.2 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. 5.4.

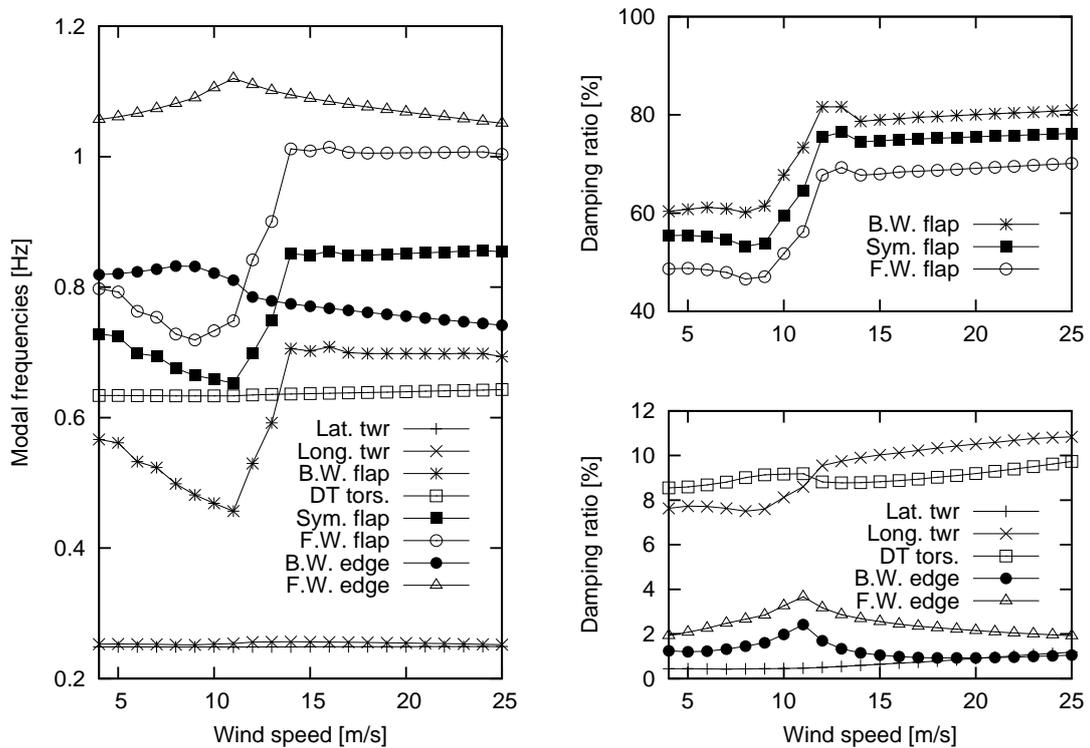


Figure 5.4.: Open-loop modal frequencies and damping ratios.

### 5.1.4. Tuning of PI controller

Selecting

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].

### 5.1.5. 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 3.1, 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. 5.1.4. 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.

## 5.2. 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. 5.1.1).
- The controller data parameters should be added to the htc file (sec. 5.1.4).
- 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.

## 6. Keyboard shortcuts

Keys	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	Decrease speed of modal vibration
Shift + i	Increase 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] Sørensen 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.