View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Online Research Database In Technology

Technical University of Denmark

How 2 HAWC2, the user's manual

Larsen, Torben J.; Hansen, Anders Melchior

Publication date:
2007

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Larsen, T. J., & Hansen, A. M. (2007). How 2 HAWC2, the user's manual. Risø National Laboratory. (Denmark. Forskningscenter Risoe. Risoe-R; No. 1597(ver. 3-1)(EN)).

DTU Library
Technical Information Center of Denmark

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

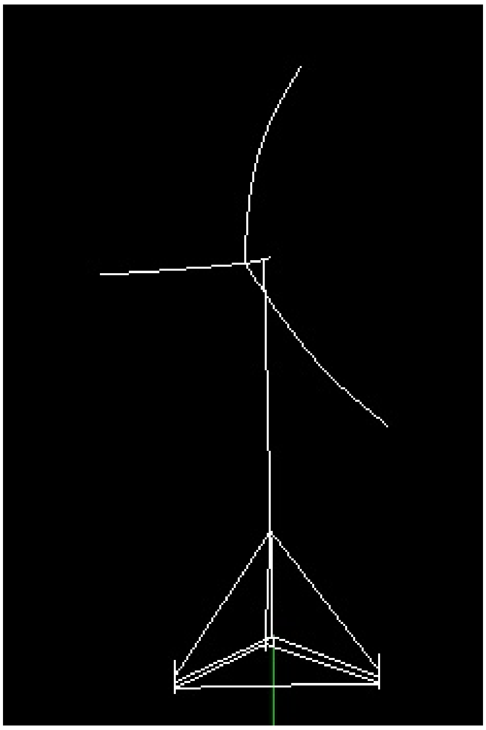
- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

How 2 HAWC2, the user's manual

Torben Juul Larsen, Anders Melchior Hansen

Risø-R-1597(ver. 3-1)(EN)



Risø National Laboratory
Technical University of Denmark
Roskilde, Denmark
December 2007

Author: Torben Juul Larsen, Anders Melchior Hansen
Title: How 2 HAWC2, the user's manual
Department: Wind Energy Department

Risø-R-1597(ver. 3-1)(EN)
December 2007

Abstract (max. 2000 char.):

The report contains the user's manual for the aeroleastic code HAWC2. The code is intended for calculating wind turbine response in time domain and has a structural formulation based on multi-body dynamics. The aerodynamic part of the code is based on the blade element momentum theory, but extended from the classic approach to handle dynamic inflow, dynamic stall, skew inflow, shear effects on the induction and effects from large deflections. It has been developed within the years 2003-2006 at the aeroelastic design research programme at Risoe, National laboratory Denmark.

This manual is updated for HAWC2 version 6.4

ISSN 0106-2840
ISBN 978-87-550-3583-6

Contract no.:

Groups own reg. no.:
1110400-1

Sponsorship:

Cover :

Pages:
Tables:
References:

Information Service Department
Risø National Laboratory
Technical University of Denmark
P.O.Box 49
DK-4000 Roskilde
Denmark
Telephone +45 46774004
bibl@risoe.dk
Fax +45 46774013

Content

Preface

The HAWC2 code is a code intended for calculating wind turbine response in time domain. It has been developed within the years 2003-2006 at the aeroelastic design research programme at Risoe, National laboratory Denmark.

The structural part of the code is based on a multibody formulation where each body is an assembly of timoshenko beam elements. The formulation is general which means that quite complex structures can be handled and arbitrary large rotations of the bodies can be handled. The turbine is modeled by an assembly of bodies connected with constraint equations, where a constraint could be a rigid coupling, a bearing, a prescribed fixed bearing angle etc. The aerodynamic part of the code is based on the blade element momentum theory, but extended from the classic approach to handle dynamic inflow, dynamic stall, skew inflow, shear effects on the induction and effects from large deflections. Several turbulence formats can be used. Control of the turbine is performed through one or more DLL's (Dynamic Link Library). The format for these DLL's is also very general, which means that any possible output sensor normally used for data file output can also be used as a sensor to the DLL. This allows the same DLL format to be used whether a control of a bearing angle, an external force or moment is placed on the structure.

The code has internally at Risoe been tested against the older validated code HAWC. Further on a detailed verification is at moment performed in the IEA annex 23 research project.

During the programming of the code a lot of focus has been put in the input checking so hopefully meaningful error messages are written to the screen in case of lacking or obvious erroneous inputs. However since the code is new and still constantly improved we appreciate feedback from the users – both good and bad critics are welcome.

The manual is also constantly updated and improved, but should at the moment cover the description of available input commands.

Acknowledgements

The code has been developed primarily by internal funds from Risø National Laboratory – Technical University of Denmark, but the research that forms the basis of the code is mainly done under contract with the Danish Energy Authority.

The structural formulation of the model is written by Anders M. Hansen as well as the solver and the linking between external loads and structure. The aerodynamic module is written by Helge A. Madsen and Torben J. Larsen. Three different stall models are implemented where the S.Ø. (Stig Øye) model is implemented by Torben J. Larsen, the mhh Beddoes model is written by Morten Hansen and Mac Gaunaa and the mhhmacg model used for trailing edge flaps is written by Mac Gaunaa and Peter Bjørn Andersen. The wind and turbulence module as well as the hydrodynamic, soil and DLL modules are written by Torben J. Larsen. The dynamic wake meandering module is written by Helge A. Madsen, Gunner Larsen and Torben J. Larsen.

General input layout

The HAWC 2 input format is written in a form that forces the user to write the input commands in a structured way so aerodynamic commands are kept together, structural commands the same etc.

The commands are divided into command blocks using the begin-end syntax. Each line has to be ended with a semi colon “;” which gives the possibility for writing comments and the end of each line after the semi colon. All command lines can be written with capital or small letters, but inside the code all lines are transformed into small letters. This could have importance if something case sensitive is written (e.g. the name of a subroutine within a DLL).

```
begin simulation;
  time_stop    100.0 ;
  solvertype   1 ;    (newmark)
;
  begin newmark;
    beta      0.27;
    gamma     0.51;
    deltat    0.02;
    bdynamic  1.0 ;
  end newmark;
end simulation;
```

In the next chapters the input commands are explaned for every part of the code. The notation is main command for a begin-end command block that is not a sub part of another begin-end block, and sub command block for a begin-end block that is included within another block. In the above written example “simulation” is a main command block and “newmark” is a sub command block.

Continue_in_file option

A feature from version 6.0 and newer is the possibility of continuing reading of the main input file into another. The command word **continue_in_file** followed by a file name causes the program to open the new file and continue reading of input until the command word **exit**. When **exit** is read the reading will continue in the previous file. An infinite number of file levels can be used.

Command name	Explanation
continue_in_file	1. File name (and path) to sublevel input file
exit	End of input file. Input reading is continued in higher level input file.

HAWC2 version handling

The HAWC2 code is still frequently updated and version handling is therefore of utmost importance to ensure quality control. For every new released version of the code a new version number is hard coded in the source. This number can be found by executing the HAWC2.exe file without any parameters. The version number is echoed to screen. The same version number is also written to every result file no matter whether ASCII or binary format is chosen. Hereby it is possible to reproduce all results at later stage and to dig in the source code for at previous version if special problems occur.

All information covering the different code versions has been made. These data are listed on the next pages.

```
!       Version information:
!       Version name                     ! Date          Resp            Info
!-----------------------------------------------------------------------------------------------------------------------------------------------------------
!       global%version='HAWC2MB 1.0'     ! 20.04.2006    TJUL            Version system started. Changes in so_dyn_stall model performed.
!                                        ! 24.04.2006    TJUL/ANMH       Bearing3 in topology - slight modification still needed, but now mhha needs a version
!       global%version='HAWC2MB 1.1'     ! 25.04.2006    TJUL            mhha laptop in MAC check, integer overflow negletec in compiler settings
!       global%version='HAWC2MB 1.1work' ! 26.04.2006    TJUL            tjul stationairy pc in MAC check
!                                                                        New check regarding thicknesses in aeodynamic files
!                                        ! 28.04.2006    TJUL            ktho stationairy pc in MAC check
!       global%version='HAWC2MB 1.2'     ! 28.04.2006    TJUL            Radius non-dim in structural _st input data and aerodynamic _ae data
!       global%version='HAWC2MB 1.3'     ! 01.05.2006    TJUL            Extra check in structural files reading procedures
!                                                                        Tab characters can now be used in htc files and other input files
!                                                                        Check that c2_def structure length larger than eps
!       global%version='HAWC2MB 1.4'     ! 02.05.2006    TJUL            New check in hawc_file output that time_stop>time_start
!                                                                        Topologi_timoschenko.f90 updated related to changes in version 1.3
!       global%version='HAWC2MB 1.5'     ! 03.05.2006    TJUL            ktho laptop in MAC check
!                                                                        Get_state_rot function in body.f90
!                                                                        New mbdy state_rot output command in topologi_mainbody_output
!                                                                        Rotation velocity and acceleration in aerodynamic blade section variables
!                                                        MACQ/TJUL       Dynamic_stall_mhh included
!       global%version='HAWC2MB 1.6'     ! 04.05.2006    TJUL            Extension of bladelink criteria for execution stop
!       global%version='HAWC2MB 1.7'     ! 09.05.2006    TJUL            New error message in windturb_mann.f90
!                                                                        New error messages regarding matrix not definite problems
!                                                                        New MAC checks (Niels Kjølstad + students)
!       global%version='HAWC2MB 1.8'     ! 09.05.2006    TJUL            New MAC check
!       global%version='HAWC2MB 1.9'     ! 16.05.2006    TJUL            New MAC check
!       global%version='HAWC2MB 2.0'     ! 18.05.2006    TJUL            New MAC check
!       global%version='HAWC2MB 2.1'     ! 19.05.2006    TJUL            Error messages corrected in mbdy state_rot command
!                                                        MHHA/TJUL       New MAC check procedure (loop over all adresses instead of only one)
!       global%version='HAWC2MB 2.2'     ! 22.05.2006    TJUL            New ignore function in body actions
!                                        ! 30.05.2006    TJUL            Old MAC check procedure reimplemented since troubles occured with the new version
!       global%version='HAWC2MB 2.3'     ! 30.05.2006    TJUL            Replacement of procedure that calculates euler parameters based on transformation matrix
!                                                                        (only important for cases with eulerp output used)
!       global%version='HAWC2MB 2.4'     ! 31.05.2006    TJUL            General cleanup in multibodyproto.f90 file (simple generator model excluded, now
!                                                                        tmp_gen_speed output command is excluded)
!                                        ! 01.06.2006    TJUL            New MAC checks
!       global%version='HAWC2MB 2.5'     ! 04.06.2006    TJUL            External Licence manager DLL used. Avoids new versions of the HAWC2 code to be build at
!                                                                        every new MAC number
!                                        !                               and and also works when the computer is not connected to a LAN
!       global%version='HAWC2MB 2.6'     ! 13.06.2006    TJUL            Newmark variables reorganized
!                                        !                               Hydrodynamic loads cut-in at 2secs, as for the aero loads. To reduce initial transients
!                                        !                               New acceptance criteria from License manager
!                                        !                               New input check in topologi_mainbody
!                                        !                               Order of radius of gyration input shifted for the new_htc_structure input. Now: 1st
!                                                                        column (Rix) is the one affected if mass center position changes on the chord line
```

```
!       global%version='HAWC2MB 2.7'      ! 23.06.2006     ANMH           Normalisation of vectors in utils funtions get_two_plane_vectors. Used for better
!                                                                         accuracy in bearing1 and bearing2 definitions
!       global%version='HAWC2MB 2.8'      ! 17.07.2006     TJUL/FRBA      Correction of bug in get_ae_data procedure in aeroload_calcforces unit. Profile sets
!                                                                         higher than one is now also usable.
!       global%version='HAWC2MB 2.9'      ! 17.07.2006     TJUL           Gravity loads cut-in at 0.5secs, same method as for the aero loads. To reduce initial
!                                                                         transients
!                                                                         Harmonic2 function in general output (time limitid harmonic function)
!       global%version='HAWC2MB 3.0'      ! 24.07.2006     TJUL           topologi_mainbody_actions module added. New features to the actions list.
!       global%version='HAWC2MB 3.1'      ! 26.07.2006     TJUL           Mann turbulence is reused if simulation time is longer than included in turbulence box
!       global%version='HAWC2MB 3.2'      ! 28.07.2006     TJUL           Correction of bug in aerodynamic moment integration procedure (only related to
!                                                                         aerodynamic file output)
                                          ! 31.07.2006     TJUL           Change of error message criteria regarding alowable number of bodies within a mainbody
!                                                                         (<n elements)
                                          ! 01.08.2006     TJUL           Correction of bug in dynstall_mhh model so no division by zero occurs when a zerolift
!                                                                         profile is used.
                                          ! 01.08.2006     ANMH           Correction of bug related to torsion of blade in the blade linker
!       global%version='HAWC2MB 3.3'      ! 04.08.2006     TJUL           Check applied on exp expressions in dynamic stall mhh model to avoid underflow errors
!       global%version='HAWC2MB 3.4'      ! 09.08.2006     TJUL           New check applied in mann turbulence unit to avoid array out of bounds during bizar
!                                                                         startup transients
!                                         ! 11.08.2006     TJUL           Correction of exp check in dynstall_mhh model just created in version 3.2
!       global%version='HAWC2MB 3.5'      ! 28.08.2006     TJUL           Generator_rotation sensor setup for old_htc_structure format - replaces the older
!                                                                         tmp_gen_speed sensor. Updates in hawcstructure.f90 and body_output.f90
!                                         !                               New error message in body_output
!                                         !                               Improvement of general command reader in genout_tools in order to accept tabulator
!                                                                         spacings
                                          !                               General shine up of aerodynamic calculations regarding induction and tiploss –
!                                                                         calculations rechecked against IEA rev 3 calculations
                                          !                               Number of radial point in the induction calculation is default set to the name number as
!                                                                         number of aero sections. Previous default of 30 stations
                                          !                               Linear interpolation in aeroload_tools updated so no division by zero occurs when x0=x1,
!                                                                         used in cases where extrapolation is not wanted
                                          ! 29.08.2006    ANMH/TJUL       Fix1 constraints updated in topologi_constraints_fix1.f90 and hawcstructure.f90. Ensures
!                                                                         e.g. that constraint properties are identical for blades. Ensures that blades performs
!                                                                         identically.
!       global%version='HAWC2MB 3.6'      ! 14.09.2006     TJUL           New acceptance criteria from license manager
!       global%version='HAWC2MB 3.7'      ! 15.09.2006     TJUL           New general load linker that replaces bladelink.f90 and wavelink.f90
                                          !                ANMH/TJUL      Pitchsensors (bearing sensor) updated during iterations too. Especially important for DLL
!                                                                         controllers
!       global%version='HAWC2MB 3.8'      ! 06.10.2006     TJUL           Correction of bug related to aero int_force and int_moment sensors
                                          !                               Correction of bug in DLL actions. On nodes different from nr. 1, in- and external forces
!                                                                         and moments were placed on the node 1 number lower.
                                          !                TJUL           Pitch sensor modified. Now pitch velocity is clculated based on numerical differentiation
!                                                                         of calculated angle. Should be less sensitive to solver inaccuracies.
                                          !                TJUL           In output of bearing sensor new options are added. (-180:180 deg output etc.)
                                          !                ANMH           Correction of bug in loadlinker. It turned out that loadfunction were only correct if an
```

```
!                                                                              even number of calculation points were used (aero or hydro). Now OK also for odd numbers
!       global%version='HAWC2MB 3.9'      ! 06.10.2006     TJUL                Soil spring module added (soil stuff from hydro module removed)
!       global%version='HAWC2MB 4.0'      ! 02.11.2006     TJUL                Extra output commands in aero output_at
!       global%version='HAWC2MB 4.1'      ! 02.11.2006     ANMH/TJUL           Replacement of added stiffness method for soil springs. Much better and faster than
!                                                                              previous. Still not perfect.
                                          ! 10.11.2006     ANMH/TJUL           Update of bearing3. Now it is general.
                                          ! 10.11.2006     TJUL                Output variables rearranged. Only command included in bearing outputs
                                          ! 10.11.2006     TJUL                Topologi input modified so many bases are allowable.
                                          ! 15.11.2006     ANMH/TJUL           Files synchronized with Anders. Slight update in dll_calls,dll_types and windturb_mann.
!       global%version='HAWC2MB 4.2'      ! 16.11.2006     TJUL                Rearrangement of output/action sensor allocation. Reduces .exe size from 23MB to 2.3MB
                                          !                TJUL/HAMA           Correction of sensors induc and windspeed in output_at aero. They were previously in a
!                                                                              wrong coordinate system when written to output_at.
!       global%version='HAWC2MB 4.3'      ! 17.11.2006     TJUL                New fix3 constraint. Locks a node to ground in a given rotation direction.
                                          ! 23.11.2006     ANMH                Update of loadlinker with respect to procedures for numerical update of stiffness,
!                                                                              damping and mass terms. Improves solutions of soil spring systems significant.
                                          ! 27.11.2006     TJUL                Same procedure used for the hydrodynamic part => faster convergence.
!       global%version='HAWC2MB 4.4'      ! 27.11.2006     TJUL                Bug fixed related to input for action sensor: mbdy moment_int
                                          ! 04.12.2006     ANMH                index+7 -> index+6 in body_get_state_rot subroutine. Affects orientation of all local
!                                                                              load elements in load linker.
!       global%version='HAWC2MB 4.5'      ! 06.12.2006     TJUL                New sensors in aero module.
                                          !                                    Out of bounds bug in aero output_at corrected
                                          ! 07.12.2006     ANMH                Files used in topologi_tools are closed after use.
                                          ! 12.12.2006     TJUL                In make output command, outputs are bypassed if global time > output stoptime
                                          ! 13.12.2006     TJUL                In mann turbulence a new command (dont_scale) is made.
                                          ! 21.12.2006     TJUL                Update of HAWC_mann module. Important only if turbulence outside box is used.
                                          ! 04.01.2007     TJUL                omega vector for aerodynamic module in rotor reference coordinates. In aero files only
!                                                                              rotation speed around y-axis is used. Eliminates influence from e.g. pitch velocity
                                          ! 04.01.2007     MHHA/TJUL           New optional relaxation parameter for solver. Extra command in simulation input.
!       global%version='HAWC2MB 4.6'      ! 12.01.2007     ANMH                Change of sign in forcedll.f90. Important only if an external force dll as coupled
!                                                                              springs are used. Not important for hawc_dll
!       global%version='HAWC2MB 4.7'      ! 06.02.2007     ANMH/TJUL           Update of code structure, multibodyproto split into several subroutines
                                          !                                    New logical variables related to simulation_input
                                          !                                    New state_at in mbdy output
!       global%version='HAWC2MB 4.8'      ! 08.02.2007     TJUL                mbdy actions force/moment commands updated with sign possibility on force component
!       global%version='HAWC2MB 4.9'      ! 12.02.2007     TJUL                new error message in turbulence input reader
                                          ! 19.02.2007     TJUL                New potential flow tower shadow model where source is linked to tower motion
        global%version='HAWC2MB 5.0'      ! 26.02.2007     TJUL                New mbdy state_rot output option: orientation in euler angles defined through the
                                                                               rotation order xyz
!       global%version='HAWC2MB 5.1'      ! 27.02.2007     TJUL                Correction of method used to calculate mbdy state_rot rotation in general
!                                                                              New mbdy state_rot output option: orientation in euler angles defined through the !
                                                                               rotation order yxz
!                                                                              Small adjustements in DLL_output to avoid array out of bounds when long mbdy names are !
                                                                               used
!       global%version='HAWC2MB 5.2'      ! 02.03.2007     ANMH/TJUL           Bug fixed related to continue_on_no_convergence criteria
!                                                          TJUL                HAWC2MB version echoed to screen before input is read.
```

```
!                                                          TJUL        New licence manager compiler option
!    global%version='HAWC2MB 5.3'       ! 13.03.2007       ANMH/TJUL   Bug fixed related to bearing3. Somehow the coupling nodes was not defined since version
!                                                                      4.0 It affects the transfer of loads from bearing3 and further dow the tower.
!    global%version='HAWC2MB 5.4'       ! 21.03.2007       TJUL/ANMH   Eigenfrequency analysis feature added. Performs analysis on every individual body
!                                       !                  TJUL        Some pointer nullify's are changed to deallocate(pointer).
!    global%version='HAWC2MB 5.4'       ! 21.03.2007       TJUL/ANMH   Eigenfrequency analysis feature added. Performs analysis on every individual body
!                                       !                  TJUL        Some pointer nullify's are changed to deallocate(pointer).
!    global%version='HAWC2MB 5.5'       ! 29.03.2007       TJUL        Small change in constraint bearing2 action input. Now only 4 parameters nessecairy as was
!                                                                      allways the idea.
!    global%version='HAWC2MB 5.6'       ! 10.04.2007       TJUL        Bug fix related to number of output sensors in DLL output
!                                       !                  ANMH        Change in external force module force_dll.f90. Update sequence of affected body changed.
!                                       !                  TJUL        body_update_T is called in the end of post_init in order to allow for added stiffness,
                                                                       damping etc. by the rest of the initialization subroutines.
!    global%version='HAWC2MB 5.7'       ! 16.04.2007       TJUL        Small update of continue on no convergence
!    global%version='HAWC2MB 5.8'       ! 18.04.2007       TJUL        Mann turbulence files is closed after every buffer read. To allow several simulations
                                                                       acces to the same turbulence files.
!                                                                      New initial buffer read so out of x-bounds errors are avoided. Uses periodicity of
                                                                       turbulence boxes. In principal this allows for infinitely large simulations.
!    global%version='HAWC2MB 5.9'       ! 23.04.2007       TJUL        Opening of mann turbulence boxed with loops and waits so several simulations can acces
                                                                       the same turbulence.
                                        ! 26.04.2007       TJUL        Only option in mbdy output, wind output, hydro output
                                                                       SO dynamic stall input parameters put in as default. No need for parameter input if not
                                                                       changed.
                                        ! 23.05.2007       TJUL        Check that turbulence scale_time_start is less the total simulation length
!                                                                      Correction of bug related to "only" option for output for main_body, wind and hydro
                                                                        output commands
     global%version='HAWC2MB 6.0'       ! 01.06.2007       TJUL        New error check that animation can be written to. Error message if not.
                                        ! 08.06.2007       TJUL        New possibility of continuing read in masterfile in a new file with the
                                                                       command:'continue_in_file'. Infinite number of level can be made. Filename also written
                                                                       to logfile when line number is written.
                                        ! 15.06.2007       TJUL        Logfile_name command option in simulation_input. Enables file written logfiles. Error
                                                                       messages more clear with *** ERROR *** as key word
                                        ! 08.08.2007       TJUL        Aerodynamic drag forces on structures enables with the new module aerodrag.
!    global%version='HAWC2MB 6.1'       ! 03.09.2007       TJUL        Corrections made in continue_in_file option. End of file check removed -
                                                                       replaced with exit command.
                                        ! 03.09.2007       ANMH        New unitnumber used when turbulence files are reopened. To avoid unit mismatch especiall
                                        ! 05.09.2007       TJUL        Bug fixed in hydroload module. Only important when more than one hydro element are used.
                                        ! 06.09.2007       TJUL        Bug fixed in hydroload module. Important if hydroelements have different coo than global.
                                        ! 07.09.2007       TJUL        Bug fixed in hydroload module. Important if relative z_distances has been used as hydro
                                                                       element input
     global%version='HAWC2MB 6.2'       ! 20.09.2007       PBJA/TJUL   Dynamic stall module that combines the mhh Beddoes stall model with the MACflap model.
                                                                       Coded by PBJA, implemented by TJUL.
     global%version='HAWC2MB 6.3'       ! 10.10.2007       TJUL        New general output command "general stairs" for a series of step functions.
     global%version='HAWC2MB 6.4'       ! 29.10.2007       TJUL        Some files synchronized with HAWC2aero regarding !IFDEF compiler directives
                                        ! 12.11.2007       TJUL        Torque and power output sensor in aero module modified to give correct results also with
```

		use of hub extenders
! 27.11.2007	TJUL	Wake meandering model implemented, rearrangement of aero files to avoid compiler linker (circulation) errors
! 29.11.2007	ANMH	Eigenvalue solver for complete turbine at standstill, initialisation of aerodrag element number!

Coordinate systems

The global coordinate system is located with the z-axis pointing vertical downwards. The x and y axes are horizontal to the side.
When wind is submitted, the default direction is along the global y-axes. Within the wind system meteorological u,v,w coordinates are used, where u is the mean wind speed direction, v is horizontal and w vertical upwards. When x,y,z notation is used within the wind coo. this refers directly to the u,v,w definition.
Every substructure and body (normally the same) is equipped with its own coordinate system with origo in node1 of this structure. The structure can be arbitrarily defined regarding orientation within this coordinate system. Within a body a number of structural elements are present. The orientation of coordinate systems for these elements are chosen automatically by the program. The local z axis is from node 1 to 2 on the element.
The coordinate system for the blade structures must be defined with the z axis pointing from the blade root and outwards, x axis in the tangential direction of rotation and y axis from the pressure side towards the suction side of the blade profiles. This is in order to make the linkage between aerodynamics and structure function.

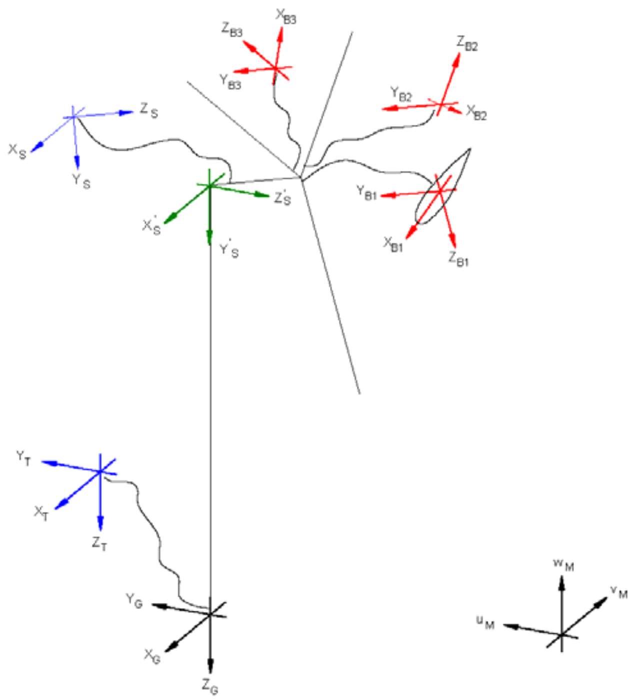


Figure 1. Illustration of coordinate system as result of user input from example in section Example of main input file at page 67. There are two coordinate systems in ***black*** *which are the default coordinate systems of gloabl reference and default wind direction. The* ***blue*** *coordinate systems are main body coordinate systems attached to node 1 of the substructure, the orientation of these are fully determined by the user. The* ***red*** *coordinate systems are also defined by the user, but in order to make the linkage between aerodynamic forces and structure work these have to have the z from root to tip, x in chordwise direction and y towards the suction side. The* ***green*** *coordinate system is just included to illustrate an intermediate coordinate rotation of 90° of the tower coo. before the final tilt angle rotation.*

Simulation

Main command block - Simulation

This block shall be present when time simulations are requested – always.

Obl.	Command name	Explanation
*	time_stop	1. Simulation length [s]
*	solvertype	1. Choice of available solver method (1=newmark)
	solver_relax	1. Relaxation parameter on increment within a timestep. Can be used to make difficult simulation run through solver when parameter is decreased, however on the cost of simulation speed. Default=1.0
	on_no_convergence	Parameter that informs solver of what to do if convergence is not obtained in a time step. 1. 'stop': simulation stops – default. 'continue': simulation continues, error message is written.
	convergence_limits	Convergence limits that must be obtained at every time step. 1. epsresq, residual on internal-external forces, default=10.0 2. epsresd, residual on increment, default=1.0 3. epsresg, residual on constraint equations, default=0.7
	max_iterations	1. Number of maximum iterations within a time step.
	animation	Included if animation file is requested 1. Animation file name incl. relative path. E.g. ./animation/animation1.dat
	logfile	Included if a logfile is requested internally from the htc command file. 1. Logfile name incl. relative path. E.g. ./logfiles/log1.txt

Sub command block – newmark

This block shall be present when the solvertype is set to the newmark method.

Obl.	Command name	Explanation
	beta	1. beta value (default=0.27)
	gamma	1. gamma value (default=0.51)
*	deltat	1. time increment [s]
	bdynamic	1. Bodies assumed rigid or flexible (1=flexible - default)

Structural input

Main command block – new_htc_structure

Obl.	Command name	Explanation
	beam_output_file_name	1. Filename incl. relative path to file where the beam data are listed (output) (example ./info/beam.dat)
	body_output_file_name	1. Filename incl. relative path to file where the body data are listed (output) (example ./info/body.dat)
	body_eigenanalysis_file_name	1. Filename incl. relative path to file where the results of an eigenanalysis are written. (output) (example ./info/eigenfreq.dat)
	constraint_output_file_name	1. Filename incl. relative path to file where the constraint data are listed (output) (example ./info/constraint.dat)

Sub command block – main_body

This block can be repeated as many times as needed. For every block a new body is added to the structure. A main body is a collection of normal bodies which are grouped together for bookkeeping purposes related to input output. When a main body consist of several bodies the spacing the name of each body inherits the name of the master body and is given an additional name of '_#', where # is the body number. An example could be a main body called 'blade1' which consist of two bodies. These are then called 'blade1_1' and blade1_2' internally in the code. The internal names are only important if (output) commands are used that refers to the specific body name and not the main body name.

Obl.	Command name	Explanation
*	name	1. Main_body identification name (must be unique)
*	type	1. Element type used (options are: timoschenko)
*	nbodies	1. Number of bodies the main_body is divided into (especially used for blades when large deformation effetcs needs attention). Equal number of elements on each body, eventually extra elements are placed on the first body.
*	node_distribution	1. Distribution method of nodes and elements. Options are: • "uniform" nnodes. Where uniform ensures equal element length and nnodes are the node numbers. • "c2_def", which ensures a node a every station defined with the sub command block c2_def.
*	damping	Rayleigh damping parameters containing factors that are multiplied to the mass and stiffness matrix respectfully. 1. M_x 2. M_y 3. M_z 4. K_x 5. K_y 6. K_z
	copy_main_body	Command that can be used if properties from a previously defined body shall be copied. The name command still have to be present, all other data are overwritten. 1. Main_body identification name of main_body that is copied.
	gravity	1. Specification of gravity (directed towards z_G). NB! this gravity command only affects the present main body. Default=9.81 [m/s^2]

Sub sub command block – timoschenko_input

Block containing information about location of the file containing distributed beam property data and the data set requested.

Obl.	Command name	Explanation
*	filename	1. Filename incl. relative path to file where the distributed beam input data are listed (example ./data/hawc2_st.dat)
*	set	1. Set number 2. Sub set number

Sub sub command block – c2_def

In this command block the definition of the centerline of the main_body is described (position of the half chord, when the main_body is a blade). The input data given with the sec commands below is used to define a continous differentiable line in space using akima spline functions. This centerline is used as basis for local coordinate system definitions for sections along the structure. If a straight line is requested a minimum of three points of this line must be present.

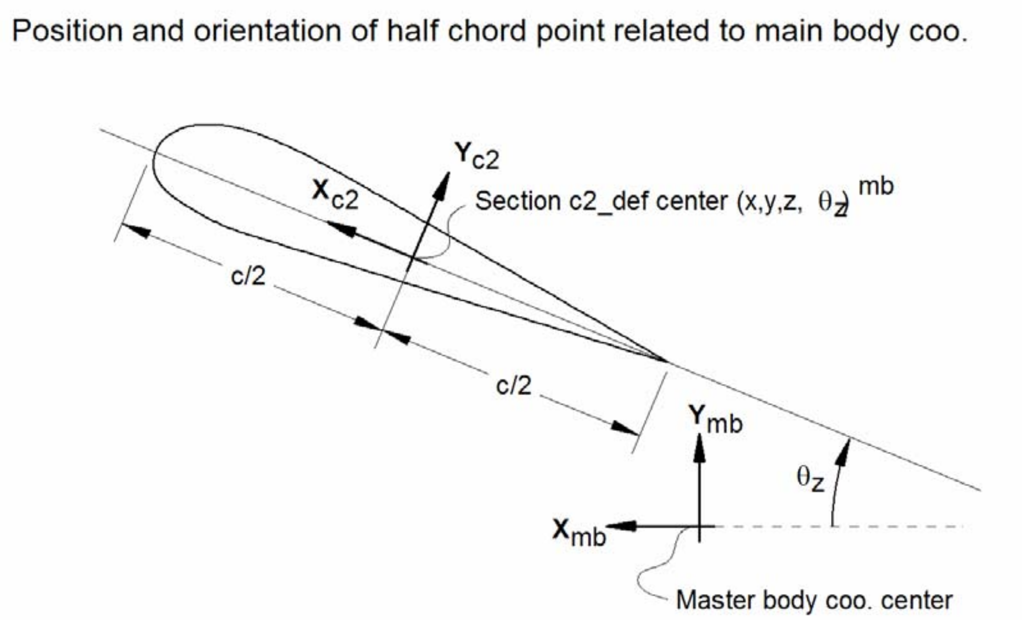


Figure 2: Illustration of c2_def coordinate system related to main body coordinates.

Obl.	Command name	Explanation
*	nsec	Must be the present before a "sec" command. 1. Number of section commands given below
*	sec	Command that must be repeated "nsec" times. Minimum 4 times. 1. Number 2. x-pos [m] 3. y-pos [m] 4. z-pos [m] 5. θ_z [deg]. Angle between local x-axis and main_body x-axis in the main_body x-y coordinate plane. For a straight blade this angle is the aerodynamic twist. Note that the sign is positive around the z-axis, which is opposite to traditional notation for etc. a pitch angle.

Format definition of file including distributed beam properties

The format of this file which in the old HAWC code was known as the hawc_st file is changed slightly for the HAWC2 new_htc_structure format.
In the file (which is a text file) two different datasets exist. There is a main set and a sub set. The main set is located after a "#" sign followed by the main set number. Within a main there can be as many subsets as desired. They are located after a "$" sign followed by the local set number. The next sign of the local set number is the number of lines in the following rows that belong to this sub set.

The content of the columns in a data row is specified in the table below. In general all centers are given according to the $C_{1/2}$ center location and all other are related to the principal bending axes.

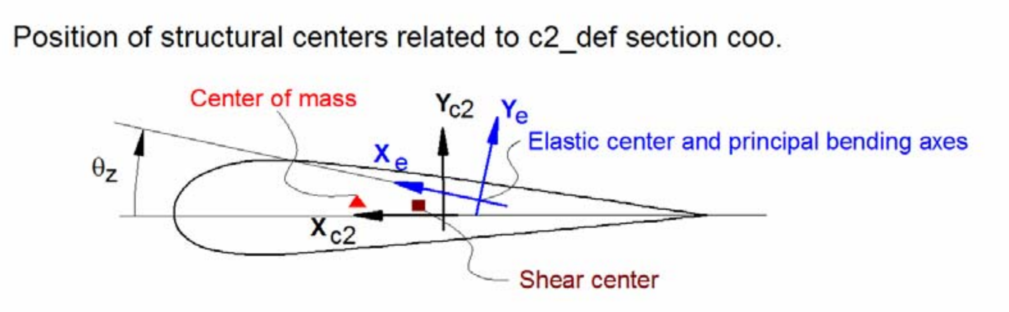


Figure 3: Illustration of structural properties that in the input files are related to the c2 coordinate system

Table 1 Structural data

Column	Parameter
1	**r, curved length distance from main_body node 1 [m]**
2	**m, mass per unit length [kg/m]**
3	**x_m, x_{c2}-coordinate from $C_{1/2}$ to mass center [m]**
4	**y_m, y_{c2}-coordinate from $C_{1/2}$ to mass center [m]**
5	**r_{ix}, radius of inertia related to elastic center projected on the x_e axis, corresponding to rotation about principal bending y_e axis [m]**
6	**r_{iy}, radius of inertia related to elastic center projected on the y_e axis, corresponding to rotation about principal bending x_e axis [m]**
7	**x_s, x_{c2}-coordinate from $C_{1/2}$ to shear center [m]**
8	**y_s, y_{c2}-coordinate from $C_{1/2}$ to shear center [m]**
9	**E, modulus of elasticity [N/m^2]**
10	**G, shear modulus of elasticity [N/m^2]**
11	**I_x, area moment of inertia with respect to principal bending x_e axis [N/m^4]**
12	**I_y, area moment of inertia with respect to principal bending y_e axis [N/m^4]**
13	**K, torsional stiffness constant with respect to z_e axis at the shear center [m^4/rad]. For a circular section only this is identical to the polar moment of inertia.**
14	**k_x shear factor for force in principal bending x_e direction [-]**
15	**k_y, shear factor for force in principal bending y_e direction [-]**
16	**A, cross sectional area [m^2]**
17	**θ_s, structural pitch about z_{c2} axis. This is the angle between the x_{c2}-axis defined with the c2_def command and the 1st main principal bending axis x_e.**
18	**x_e, x_{c2}-coordinate from $C_{1/2}$ to center of elasticity [m]**
19	**y_e, y_{c2}-coordinate from $C_{1/2}$ to center of elasticity [m]**

An example of an inputfile can be seen on the next page. The most important features to be aware of are colored with red.

```
1 main data sets available
-----------------------------------
Here is space for comments etc
.
.
.
-----------------------------------
#1 Main data set number 1 – an example of a shaft structure
-----------------------------------
More comments space
r          m      x_cg   y_cg ri_x   ri_y   x_sh   y_sh   E          G          I_x        I_y        K        k_x    k_y    A      theta_s x_e    y_e
[m]        [kg/m] [m]    [m]  [m]    [m]    [m]    [m]    [N/m^2]    [N/m^2]    [N/m^4]    [N/m^4]    [N/m^4]  [-]    [-]    [m^2]  [deg]   [m]    [m]
$1 10 Sub set number 1 with 10 data rows
0.00       100    0      0    224.18 224.18 0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
0.10       100    0      0    224.18 224.18 0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
0.1001     1      0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
1.00       1      0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
1.90       1      0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
2.00       1      0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
3.00       1      0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
3.20       1      0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
4.00       1      0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
5.0191     1      0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
-----------------------------------
More comments space

r          m      x_cg   y_cg ri_x   ri_y   x_sh   y_sh   E          G          I_x        I_y        K        k_x    k_y    A      theta_s x_e    y_e
[m]        [kg/m] [m]    [m]  [m]    [m]    [m]    [m]    [N/m^2]    [N/m^2]    [N/m^4]    [N/m^4]    [N/m^4]  [-]    [-]    [m^2]  [deg]   [m]    [m]
$2 10 As dataset 1, but stiff
0.00       100    0      0    224.18 224.18 0      0      2.10E+16   8.10E+15   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
0.10       100    0      0    224.18 224.18 0      0      2.10E+16   8.10E+15   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
0.1001     1      0      0    0.2    0.2    0      0      2.10E+16   8.10E+15   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
1.00       1      0      0    0.2    0.2    0      0      2.10E+16   8.10E+15   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
1.90       1      0      0    0.2    0.2    0      0      2.10E+16   8.10E+15   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
2.00       1      0      0    0.2    0.2    0      0      2.10E+16   8.10E+15   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
3.00       1      0      0    0.2    0.2    0      0      2.10E+16   8.10E+15   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
3.20       1      0      0    0.2    0.2    0      0      2.10E+16   8.10E+15   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
4.00       1      0      0    0.2    0.2    0      0      2.10E+16   8.10E+15   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
5.0191     1      0      0    0.2    0.2    0      0      2.10E+16   8.10E+15   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
-----------------------------------
More comments space

r          m      x_cg   y_cg ri_x   ri_y   x_sh   y_sh   E          G          I_x        I_y        K        k_x    k_y    A      theta_s x_e    y_e
[m]        [kg/m] [m]    [m]  [m]    [m]    [m]    [m]    [N/m^2]    [N/m^2]    [N/m^4]    [N/m^4]    [N/m^4]  [-]    [-]    [m^2]  [deg]   [m]    [m]
$3 10 as data set 1 but changed mass properties
0.00       1000   0      0    2.2418 2.2418 0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
0.10       1000   0      0    2.2418 2.2418 0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
0.1001     1      0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
1.00       1      0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
1.90       1      0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
2.00       1      0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
3.00       1      0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
3.20       1      0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
4.00       1      0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
5.0191     1      0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0       0.0    0.0
```

Sub command - orientation

In this command block the orientation (regarding position and rotation) of every main_body are specified.

Sub sub command - base

The orientation of a main_body to which all other bodies are linked – directly or indirectly.

Obl.	Command name	Explanation
*	body	1. Main_body name that is declared to be the base of all bodies (normally the tower or foundation)
*	inipos	Initial position in global coordinates. 1. x-pos [m] 2. y-pos [m] 3. z-pos [m]
♣	body_eulerang	Command that can be repeated as many times as needed. All following rotation are given as a sequence of euler angle rotations. All angle can be filled in (rotation order x,y,z), but it is recommended only to give a value different from zero on one of the angles and reuse the command if several rotations are needed. 1. θ_x [deg] 2. θ_y [deg] 3. θ_z [deg]
♣	body_eulerpar	The rotation is given as euler parameters (quaternions) directly (global coo). 1. r_0 2. $\mathbf{r}_1$ 3. $\mathbf{r}_2$ 4. $\mathbf{r}_3$
♣	body_axisangle	Command that can be repeated as many times as needed. A version of the euler parameters where the input is a rotation vector and the rotation angle of this vector. 1. x-value 2. y-value 3. z-value 4. angle [deg]

♣ One of these commands must be present.

Sub sub command - relative

This command block can be repeated as many times as needed. However the orientation of every main_body should be described.

Obl.	Command name	Explanation
*	body1	1. Main_body name to which the next main_body is attached. 2. Node number of body1 that is used for connection. (“last” can be specified which ensures that the last node on the main_body is used). “last” must be used for now – first or internal nodes doesn’t work for now...!
*	body2	1. Main_body name of the main_body that is positioned in space by the relative command. 2. Node number of body2 that is used for connection. (“last” can be specified which ensures that the last node on the main_body is used). “1” must be used for now – last or internal nodes don’t work for now...!
♣	body2_eulerang	Command that can be repeated as many times as needed. All following rotation are given as a sequence of euler angle rotations. All angle can be filled in (rotation order x,y,z), but it is recommended only to give a value different from zero on one of the angles and reuse the command if several rotations are needed. Until a rotation command is specified body2 has same coo. as body1. Rotations are performed in the present body2 coo. system. 1. θ_x [deg] 2. θ_y [deg] 3. θ_z [deg]
♣	body_eulerpar	The rotation is given as euler parameters (quaternions) directly (global coo). 1. r_0 2. $\mathbf{r}_1$ 3. $\mathbf{r}_2$ 4. $\mathbf{r}_3$
♣	body_axisangle	Command that can be repeated as many times as needed. A version of the euler parameters where the input is a rotation vector and the rotation angle of this vector. Until a rotation command is specified body2 has same coo. as body1. Rotations are performed in the present body2 coo. system. 1. x-value 2. y-value 3. z-value 4. angle [deg]
	body2_ini_rotvec_d1	Initial rotation velocity of main body and all subsequent attached bodies. A rotation vector is set up and the size of vector (the rotational speed) is given. The coordinate system used is body2 coo. 1. x-value 2. y-value 3. z-value 4. Vector size (rotational speed [rad/s])

Sub command - constraint

In this block constraints between the main_bodies and to the global coordinate system are defined.

Sub sub command – fix0

This constraint fix node number 1 of a given main_body to ground.

Obl.	Command name	Explanation
*	body	Name of main body that is fixed to ground at node 1

Sub sub command – fix1

This constraint fix a given node on one main_body to another main_body's node.

Obl.	Command name	Explanation
*	body1	1. Main_body name to which the next main_body is fixed. 2. Node number of body1 that is used for the constraint. ("last" can be specified which ensures that the last node on the main_body is used). "last" must be used for now – first or internal nodes doesn't work for now...!
*	body2	1. Main_body name of the main_body that is fixed to body1. 2. Node number of body2 that is used for the constraint. ("last" can be specified which ensures that the last node on the main_body is used). "1" must be used for now – last or internal nodes doesn't work for now...!

Sub sub command – fix2

This constraint fix a node 1 on a main_body to ground in x,y,z direction. The direction that is free or fixed is optional.

Obl.	Command name	Explanation
*	body	1. Main_body name to which node 1 is fixed.
*	dof	Direction in global coo that is fixed in translation 1. x-direction (0=free, 1=fixed) 2. y-direction (0=free, 1=fixed) 3. z-direction (0=free, 1=fixed)

Sub sub command – fix3

This constraint fix a node to ground in tx,ty,tz rotation direction. The rotation direction that is free or fixed is optional.

Obl.	Command name	Explanation
*	body	1. Main_body name to which node 1 is fixed. 2. Node number
*	dof	Direction in global coo that is fixed in rotation 1. tx-rot.direction (0=free, 1=fixed) 2. ty-rot.direction (0=free, 1=fixed) 3. tz-rot.direction (0=free, 1=fixed)

Sub sub command – bearing1

Constraint with properties as a bearing without friction. A sensor with same identification name as the constraint is set up for output purpose.

Obl.	Command name	Explanation
*	name	1. Identification name
*	body1	1. Main_body name to which the next main_body is fixed with bearing1 properties. 2. Node number of body1 that is used for the constraint. (“last” can be specified which ensures that the last node on the main_body is used). “last” must be used for now – first or internal nodes doesn’t work for now...!
*	body2	1. Main_body name of the main_body that is fixed to body1 with bearing1 properties. 2. Node number of body2 that is used for the constraint. (“last” can be specified which ensures that the last node on the main_body is used). “1” must be used for now – last or internal nodes doesn’t work for now...!
*	bearing_vector	Vector to which the free rotation is possible. The direction of this vector also defines the coo to which the output angle is defined. 1. Coo. system used for vector definition (0=global,1=body1,2=body2) 2. x-axis 3. y-axis 4. z-axis

Sub sub command – bearing2

This constraint allows a rotation where the angle is directly specified by an external dll action command.

Obl.	Command name	Explanation
*	name	1. Identification name
*	body1	1. Main_body name to which the next main_body is fixed with bearing2 properties. 2. Node number of body1 that is used for the constraint. (“last” can be specified which ensures that the last node on the main_body is used). “last” must be used for now – first or internal nodes doesn’t work for now...!
*	body2	1. Main_body name of the main_body that is fixed to body1 with bearing1 properties. 2. Node number of body2 that is used for the constraint. (“last” can be specified which ensures that the last node on the main_body is used). “1” must be used for now – last or internal nodes doesn’t work for now...!
*	bearing_vector	Vector to which the rotation occur. The direction of this vector also defines the coo to which the output angle is defined. 1. Coo. system used for vector definition (0=global,1=body1,2=body2) 2. x-axis 3. y-axis 4. z-axis

Sub sub command – bearing3

This constraint allows a rotation where the angle velocity is kept constant throughout the simulation.

Obl.	Command name	Explanation
*	name	1. Identification name
*	body1	1. Main_body name to which the next main_body is fixed with bearing3 properties. 2. Node number of body1 that is used for the constraint. ("last" can be specified which ensures that the last node on the main_body is used). "last" must be used for now – first or internal nodes doesn't work for now...!
*	body2	1. Main_body name of the main_body that is fixed to body1 with bearing3 properties. 2. Node number of body2 that is used for the constraint. ("last" can be specified which ensures that the last node on the main_body is used). "1" must be used for now – last or internal nodes doesn't work for now...!
*	bearing_vector	Vector to which the rotation occur. The direction of this vector also defines the coo to which the output angle is defined. 1. Coo. system used for vector definition (0=global,1=body1,2=body2) 2. x-axis 3. y-axis 4. z-axis
*	omegas	1. Rotational speed [rad/sec]

Main command block – old_htc_structure

The old_htc_structure format is taken from the old HAWC code, which is also the format used in the code HAWCStab.

One of the main differences of this format compared to the new_htc_format is that the number of bodies and their properties are more restricted in the old format. In this format there is always one body named 'tower', one body named 'shaft' and three blades. The naming procedure for blade 1 is 'blade1' if the blade only consist of one body, or 'blade1body4' if the blade consist of several bodies. In this case body 4 of blade 1 was addressed.

The command options are listed in the table below.

Obl.	Command name	Explanation
*	htcfile	1. File name where old_htc_structure input is located. E.g. ./data/old_htc_structure.htc
*	nbbl	1. Number of bodies a blade is divided into.
*	omegas	1. Initial rotation velocity of the shaft. [rad/s]
*	fixed_speed	1. Specification whether the rotational speed is kept constant the constraint acts as a bearing. 'yes'=constant, 'no'=bearing
*	fixed_pitch	1. Specification whether the pitch angle is kept constant (or controlled with direct specification of the angle), or the constraint acts as a bearing. 'yes'=fixed, 'no'=bearing
*	free_tower_suspension	1. Specification whether the first node on the tower is free in translation and rotation or fully fixed. 'yes'=free, 'no'=fixed

Description of the old htc file format

In this file all commands must be given with CAPITAL letters.

Command 1	Command 2	Explanation
DEFINE_STRING	STRUCTURE_FILE_EXT	1. File extension ex1 for identification of input file with element structural data. The file has to be located in the ./data/ subdirectory.located.
PARAMETERS	TURBINE	1. Number of blades. 2. Yawing DOF. 1=yes and a real DOF, 2='fixed'. Default=2. 3. Shaft rotation DOF. 1=yes and a real DOF (used with normal generator model), 2='fixed', 3=Prescribed, 4='variable speed controlled through regulation 4. Teeter DOF. 1=yes and a real DOF. 2=fixed
PARAMETERS	TILT	1. Tilt angle [deg]
ROTOR	BASIC_LAYOUT	1. Number of blades. Default=3 2. Blade flange node number. 3. Cone angle for blade 1 [deg] 4. Cone angle for blade 2 [deg] 5. Cone angle for blade 3 [deg] 6. δ3 angle for blade 1 [deg] 7. δ3 angle for blade 2 [deg] 8. δ3 angle for blade 3 [deg] 9. Pitch angle (pos sign acc. to z-dir.) for blade 1. [deg] 10. Pitch angle (pos sign acc. to z-dir.) for blade 2. [deg]

		11. Pitch angle (pos sign acc. to z-dir.) for blade 3. [deg] 12. Collective pitch angle for blade 1-3. [deg]
DAMPING	BLADE	Rayleigh damping proportionality factors for the mass matrix: 1. Bending about x-axis 2. Bending about y-axis 3. Bending about z-axis Rayleigh damping proportionality factors for the stiffness matrix: 4. Bending about x-axis 5. Bending about y-axis 6. Bending about z-axis
DAMPING	SHAFT	Rayleigh damping proportionality factors for the mass matrix: 1. Bending about x-axis 2. Bending about y-axis 3. Bending about z-axis Rayleigh damping proportionality factors for the stiffness matrix: 4. Bending about x-axis 5. Bending about y-axis 6. Bending about z-axis
DAMPING	TOWER	Rayleigh damping proportionality factors for the mass matrix: 1. Bending about x-axis 2. Bending about y-axis 3. Bending about z-axis Rayleigh damping proportionality factors for the stiffness matrix: 4. Bending about x-axis 5. Bending about y-axis 6. Bending about z-axis
NODE	BLADE	Command that must be repeated for every node on the blade. Node numbers must increase continuously. The nodes should be placed in the center of elasticity of the blade. 1. Blade number 2. Node number 3. x-coordinate [m] 4. y-coordinate [m] 5. z-coordinate [m]
NODE	NACELLE	Command that must be repeated for every node on the shaft. Node numbers must increase continuously. 1. Node number 2. x-coordinate [m] 3. y-coordinate [m] 4. z-coordinate [m]
NODE	TOWER	Command that must be repeated for every node on the tower. Node numbers must increase continuously. 1. Node number 2. x-coordinate [m] 3. y-coordinate [m] 4. z-coordinate [m]
TYPES	BLADE_STRUCTURE_3	Specification of actual structural blade data sets for a 3-bladed turbine. The data set must be available in the HAWC_st.ex1 file. 1. Set number for blade 1 2. Set number for blade 2 3. Set number for blade 3

TYPES	SHAFT_STRUCTURE	Specification of actual structural shaft data sets. The data set must be available in the HAWC_st.ex1 file. 1. Set number
TYPES	TOWER_STRUCTURE	Specification of actual structural tower data sets. The data set must be available in the HAWC_st.ex1 file. 1. Set number
TYPES	USE_CALCULATED_BEAMS	No parameters.
STOP		End of file. No parameters.

Description of the data format in the hawc_st.ex1 file

This file contains information of the distributed element properties that is being discretized by to HAWC2 code for each beam element.

The general layout of the file is specified in the table below.

Line 1	Three numbers are red. They declare whether data is included for respectively BLADE, SHAFT, TOWER. They shall always be set to 1 because of all substructures are needed for the model.		
Line 2,3,...	The file is divided into 3 sections. One for the blade, nacelle and tower.		
In a section	Line 1 Line 2,3,...	One number is read. This is the number of data sets in the section. The file is divided into the number of sets.	
	In a set	Line 1	Two numbers are read. First number is the data number, second is the number of lines N included in the data set.
		Lines 3 to N+2	The data information is read. The columns of the data set contain cross sectional properties of the respective substructures as function of distance from substructures as function of distance from substructure base node as shown in table below.

The content of the colums in a data row is specified in the table below.

Table 2 Structural data

Column	Parameter
1	r, distance from main_body node 1 along z-coordinate [m]
2	m, mass per unit length [kg/m]
3	x_m, x_l-coordinate from $C_{1/2}$ to mass center [m]
4	y_m, y_l-coordinate from $C_{1/2}$ to mass center [m]
5	r_{ix}, radius of inertia related to elastic center, corresponding to rotation about y_l axis [m]
6	r_{iy}, radius of inertia related to elastic center, corresponding to rotation about x_l axis [m]
7	x_s, x_l-coordinate from $C_{1/2}$ to shear center [m]
8	y_s, y_l-coordinate from $C_{1/2}$ to shear center [m]
9	E, modulus of elasticity [N/m^2]
10	G, shear modulus of elasticity [N/m^2]
11	I_x, area moment of inertia with respect to x_l axis [N/m^4]
12	I_y, area moment of inertia with respect to y_l axis [N/m^4]
13	K, torsional stiffness constant with respect to z_l axis [N/m^4]. For a circular section only this is identical to the polar moment of inertia.
14	k_x shear factor for force in x_l direction [-]
15	k_y, shear factor for force in y_l direction [-]
16	A, cross sectional area [m^2]
17	θ_s, structural pitch about z axis. This angle is the sum of structural pitch and aerodynamic pitch. Positive with leading edge towards the wind! [deg]

An example of an inputfile can be seen on the next page. The most important features to be aware of are colored with red. The data in the table does not necessarily correspond to real turbine properties.

```
1 1 1 A title can be written here---
2 Main data set 1 contains 2 sub sets: The blade
1 10 r     m     x_cg   y_cg ri_x   ri_y   x_sh   y_sh   E          G          I_x        I_y        K        k_x    k_y    A      theta_s  x_e    y_e
0.00       100   0      0    224.18 224.18 0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
0.10       100   0      0    224.18 224.18 0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
0.1001     1     0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
1.00       1     0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
1.90       1     0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
2.00       1     0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
3.00       1     0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
3.20       1     0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
4.00       1     0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
5.0191     1     0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
2 10 As dataset 1, but stiff
0.00       100   0      0    224.18 224.18 0      0      2.10E+16   8.10E+15   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
0.10       100   0      0    224.18 224.18 0      0      2.10E+16   8.10E+15   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
0.1001     1     0      0    0.2    0.2    0      0      2.10E+16   8.10E+15   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
1.00       1     0      0    0.2    0.2    0      0      2.10E+16   8.10E+15   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
1.90       1     0      0    0.2    0.2    0      0      2.10E+16   8.10E+15   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
2.00       1     0      0    0.2    0.2    0      0      2.10E+16   8.10E+15   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
3.00       1     0      0    0.2    0.2    0      0      2.10E+16   8.10E+15   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
3.20       1     0      0    0.2    0.2    0      0      2.10E+16   8.10E+15   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
4.00       1     0      0    0.2    0.2    0      0      2.10E+16   8.10E+15   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
5.0191     1     0      0    0.2    0.2    0      0      2.10E+16   8.10E+15   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
1 Main data set 2 contains 1 sub sets: The shaft
1 2
0.00       100   0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
5.0191     100   0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
1 Main data set 3 contains 1 sub sets: The tower
1 2
0.00       100   0      0    0.2    0.2    0      0      2.10E+11   8.10E+10   1.00E+02   1.00E+02   0.05376  0.52   0.52   0.59   0        0.0    0.0
```

DLL control

This block contains the possible Dynamic Link Library formats accessible for the user. The Dll's are mainly used to control the turbine speed and pitch, but since the DLL format is very general other use is possible too e.g. external loading of the turbine.

Main command block – dll

So far only one DLL format is availbale, which is the hawc_dll format listed below.

Sub command block – hawc_dll

The basic thing regarding the HAWC_DLL format is that two one-dimensional arrays are transferred between the HAWC2 core and the DLL procedure. The first contains data going from the HAWC2 core to the DLL and the other contains data going from the DLL to the core.

This block can be repeated as many times as desired. Reference number to DLL is same order as listed, starting with number 1.

Obl.	Command name	Explanation
*	filename	1. Filename incl. relative path of the DLL (example ./DLL/control.dll)
*	dll_subroutine	1. Name of subroutine in DLL that is addressed (remember to specify the name in the DLL with small letters!)
*	arraysizes	1. size of array with outgoing data 2. size of array with ingoing data
	deltat	1. Time between dll calls. Must correspond to the simulation sample frequency or be a multiple of the time step size. If deltat=0.0 or the deltat command line is omitted the HAWC2 code calls the dll subroutine at every time step.

Sub command block - output

In this block the same block the same sensors are available as when data results are written to a file with the main block command **output**. The order of the sensors in the data array is continuously increased as more sensors are added.

Sub command block - actions

In this command block variables inside the HAWC2 code is changed depending of the specifications. An action commands creates a handle to the HAWC2 model to which a variable in the input array from the DLL is linked.

!NB in the command name two separate words are present.

Obl.	Command name	Explanation
	aero beta	The flap angle beta is set for a trailing edge flap section (is the mhhmagf stall model is used). The angle is positive towards the pressure side of the profile. Unit is [deg] 1. Blade number 2. Flap section number
	body force_ext	An external force is placed on the structure. Unit is [N]. 1. body name 2. node number 3. composant (1 = F_x, 2 = F_y, 3 = F_z)
	body moment_ext	An external moment is placed on the structure. Unit is [Nm]. 1. body name 2. node number 3. composant (1 = M_x, 2 = M_y, 3 = M_z)
	body force_int	An external force with a reaction component is placed on the structure. Unit is [N]. 1. body name for action force 2. node number 3. composant (1 = F_x, 2 = F_y, 3 = F_z) 4. body name for reaction force 5. Node number
	body moment_int	An external moment with a reaction component is placed on the structure. Unit is [N]. 1. body name for action moment 2. node number 3. composant (1 = M_x, 2 = M_y, 3 = M_z) 4. body name for reaction moment 5. Node number
	body bearing_angle	A bearing either defined through the new structure format through bearing2 or through the old structure format (spitch1=pitch angle for blade 1, spitch2=pitch angle for blade 2,...). The angle limits are so far [0-90deg]. 1. Bearing name
	mbdy force_ext	An external force is placed on the structure. Unit is [N]. 1. main body name 2. node number on main body 3. composant (1 = F_x, 2 = F_y, 3 = F_z) , if negative number the force is inserted with opposite sign. 4. coordinate system (possible options are: mbdy name,"global","local"). "local" means local element coo on the inner element (on the element indexed 1 lower that the node number). One exception if node number =1 then the element nr. also equals 1.

Obl.	Command name	Explanation
	mbdy moment_ext	An external moment is placed on the structure. Unit is [Nm]. 1. main body name 2. node number on main body 3. composant (1 = M_x, 2 = M_y, 3 = M_z) , if negative number the moment is inserted with opposite sign. 4. coordinate system (possible options are: mbdy name,”global”,”local”). “local” means local element coo on the inner element (on the element indexed 1 lower that the node number). One exception if node number =1 then the element nr. also equals 1.
	mbdy force_int	An internal force with a reaction component is placed on the structure. Unit is [N]. 1. main body name for action force 2. node number on main body 3. composant (1 = F_x, 2 = F_y, 3 = F_z) , if negative number the force is inserted with opposite sign. 4. coordinate system (possible options are: mbdy name,”global”,”local”). “local” means local element coo on the inner element (on the element indexed 1 lower that the node number). One exception if node number =1 then the element nr. also equals 1. 5. main body name for reaction force 6. Node number on this main body
	mbdy moment_int	An internal force with a reaction component is placed on the structure. Unit is [Nm]. 1. main body name for action moment 2. node number on main body 3. composant (1 = M_x, 2 = M_y, 3 = M_z) , if negative number the moment is inserted with opposite sign. 4. coordinate system (possible options are: mbdy name,”global”,”local”). “local” means local element coo on the inner element (on the element indexed 1 lower that the node number). One exception if node number =1 then the element nr. also equals 1. 5. main body name for reaction moment 6. Node number on this main body
	constraint bearing2 angle	The angle of a bearing2 constraint is set. The angle limits are so far [0-90deg]. 1. Bearing name
	body printvar	Variable is just echoed on the screen. No parameters.
	body ignore	1. Number of consecutive array spaces that will be ignored
	mbdy printvar	Variable is just echoed on the screen. No parameters.
	mbdy ignore	1. Number of consecutive array spaces that will be ignored

DLL format example written in FORTRAN 90

```
subroutine test(n1,array1,n2,array2)
implicit none
!DEC$ ATTRIBUTES DLLEXPORT, ALIAS:'test'::test
integer*4           :: n1, &    ! Dummy integer value containing the array size of array1
                       n2       ! Dummy integer value containing the array size of array2
real*4,dimension(10) :: array1  ! fixed-length array, data from HAWC2 to DLL
                                ! - in this case with length 10
real*4,dimension(5)  :: array2  ! fixed-length array, data from DLL to HAWC2
                                ! - in this case with length 5

! Code is written here

end subroutine test
```

DLL format example written in Delphi

```
library test_dll;

type
  array_10 = array[1..10] of single;
  array_5  = array[1..5] of single;

Procedure test(var n1:integer;var array1 : array_10;
               var n2:integer;var array2 : array_5);stdcall;
// n1 is a dummy integer value containing the size of array1
// n2 is a dummy integer value containing the size of array2
begin
  // Code is written here

end;
exports test;

begin
  writeln('The DLL pitchservo.dll is loaded with succes');

  // Initialization of variables can be performed here
end;

end.
```

Wind and turbulence

Main command block -wind

Obl.	Command name	Explanation
*	wsp	1. Mean wind speed in center [m/s]
*	density	1. Density of the wind [kg/m^3]
*	tint	1. Turbulence intensity [-].
*	horizontal_input	Should the above commands be understood as defined in the global coordinate system (with horizontal axes) or the meteorological coordinates system (u,v,w) witch can be tilted etc. 1. (0=meteorological – default, 1=horizontal)
*	center_pos0	Global coordinates for the center start point of the turbulence box, meteorological coordinate system etc. (default should the hub center) 1. x_G [m] 2. y_G [m] 3. z_G [m]
*	windfield_rotations	Orientation of the wind field. The rotations of the field are performed as a series of 3 rotations in the order yaw, tilt and roll. When all angles are zero the flow direction is following the global y direction. 1. Wind yaw angle [deg], positive when the wind comes from the right, seen from the turbine looking towards -y_G. 2. Terrain slope angle [deg], positive when the wind comes from below. 3. Roll of wind field [deg], positive when the wind field is rotated according to the turbulence u-component.
*	shear_format	Definition of the mean wind shear 1. Shear type (0=none, 1=constant, 2=logarithmic, 3=power law, 4=linear) 2. Parameter used together with shear type (case of shear type: 0=dummy, 1=dummy, 2=roughness length, 3=power law exponent, 4=du/dz at center
*	turb_format	1. Turbulence format (0=none, 1=mann, 2=flex)
*	tower_shadow_method	1. Tower shadow model (0=none, 1=potential flow – default, 2=jet model, 3=potential_2 (flow where shadow source is moved and rotated with tower coordinates system)
	scale_time_start	1. Starting time for turbulence scaling [s]. Stop time is determined by simulation length.
	wind_ramp_factor	Command that can be repeated as many times as needed. The wind_ramp_factor is used to calculate a factor that is multiplied to the wind speed vectors. Can be used to make troublefree cut-in situations. Linear interpolation is performed between t_0 and t_{stop}. 1. time start, t_0 2. time stop, t_{stop} 3. factor at t_0 4. factor at t_{stop}

Obl.	Command name	Explanation
	wind_ramp_abs	Command that can be repeated as many times as needed. The wind_ramp_abs is used to calculate a wind speed that is added to the wind speed u-composant. Can be used to make wind steps etc. Linear interpolation is performed between t_0 and t_{stop}. 1. time start, t_0 2. time stop, t_{stop} 3. wind speed at t_0 4. wind speed at t_{stop}
	user_defined_shear	1. Filename incl. relative path to file containing user defined shear (example ./data/hawc2_shear.dat)
	iec_gust	Gust generator according to IEC 61400-1 1. gust type 'eog' = extreme operating gust 'edc' = extreme direction change 'ecg' = extreme coherent gust 'ecd' = extreme coherent gust with dir. change 'ews' = extreme wind shear 2. gust amplitude [m/s] 3. gust angle [deg] 4. time start, t_0 [m/s] 5. duration [m/s]

Sub command block - mann

Block that must be included if the mann turbulence format is chosen. Normal practice is to use all three turbulence components (u,v,w) but only the specified components are used

Obl.	Command name	Explanation
	filename_u	1. Filename incl. relative path to file containing mann turbulence u-composant (example ./turb/mann-u.bin)
	filename_v	1. Filename incl. relative path to file containing mann turbulence v-composant (example ./turb/mann-v.bin)
	filename_w	1. Filename incl. relative path to file containing mann turbulence w-composant (example ./turb/mann-w.bin)
*	box_dim_u	1. Number of grid points i u-direction 2. Length between grid points in u-direction
*	box_dim_v	1. Number of grid points i v-direction 2. Length between grid points in v-direction
*	box_dim_w	1. Number of grid points i w-direction 2. Length between grid points in w-direction
*	std_scaling	Ratio between standard deveation for specified composant related to turbulence intensity input specified in main wind command block. 1. Ratio to u-direction (default=1.0) 2. Ratio to v-direction (default=0.7) 3. Ratio to w-direction (default=0.5)
	dont_scale	1. (0=scaling according to specified inputs – default, 1=raw turbulence field used without any scaling)

Sub command block - flex

Block that must be included if the mann turbulence format is chosen.

Obl.	Command name	Explanation
*	filename_u	1. Filename incl. relative path to file containing flex turbulence u-composant (example ./turb/flex-u.int)
*	filename_v	1. Filename incl. relative path to file containing flex turbulence v-composant (example ./turb/flex-v.int)
*	filename_w	1. Filename incl. relative path to file containing flex turbulence w-composant (example ./turb/flex-w.int)
*	std_scaling	Ratio between standard deveation for specified composant related to turbulence intensity input specified in main wind command block. 1. Ratio to u-direction (default=1.0) 2. Ratio to v-direction (default=0.7) 3. Ratio to w-direction (default=0.5)

File description of user defined shear

In this file a user defined shear used instead, or in combination with one of the default shear types (logarithmic, exponential...). When the user defined shear is used the name and location of the datafile must be specified with the *wind – user_defined_shear* command. This command specifies the location of the file and activates the user defined shear. If this shear is replacing the original default shear the command *wind – shear_format* must be set to zero!

Only one shear can be present in a single file. The shear describes the mean wind profile of the u, v and w component of a vertical cross section at the rotor. The wind speeds are normalized with the mean wind speed defined with the command *wind – wsp.*

Line number	Description
1	Headline (not used by HAWC2)
2	Information of shear v-component. #1 is the number of columns, NC #2 is the number of rows, NR
3	Headline (not used by HAWC2)
4..+NR	Wind speed in v-direction, normalized with u-mean. # NC columns
+1	Headline (not used by HAWC2)
+1..+NR	Wind speed in v-direction, normalized with u-mean. # NC columns.
+1	Headline (not used by HAWC2)
+1..+NR	Wind speed in v-direction, normalized with u-mean. # NC columns
+1	Headline (not used by HAWC2)
+1..+NC	Horizontal position of grid points (meteorological coo)
+1	Headline (not used by HAWC2)
+1..+NR	Vertical position of grid points (meteorological coo)

Example of user defined shear file

```
# User defined shear file
3 5  # nr_v, nr_w     array sizes
# shear_v component, normalized with U_mean
0.0 0.0 0.0
0.0 0.0 0.0
0.0 0.0 0.0
0.0 0.0 0.0
0.0 0.0 0.0
# shear_u component
1.0 1.0 1.0
1.0 1.0 1.0
1.0 1.0 1.0
1.0 1.0 1.0
1.0 1.0 1.0
# shear_w component
0.0 0.0 0.0
0.0 0.0 0.0
0.0 0.0 0.0
0.0 0.0 0.0
0.0 0.0 0.0
# v coordinates
-50.0
0.0
50.0
# w coordinates
0.0
20.0
60.0
100.0
200.0
```

Sub command block - wakes

Block that must be included if the Dynamic Wake Meandering model is used to model the wind flow from one or more upstream turbines.
In order to make the model function two Mann turbulence boxed must be used. One for the meandering turbulence – which is a box containing atmospheric turbulence, but generated with a course resolution in the v,w plane (grid size of 1 rotor diameter). It is important that the turbulence vectors at the individual grid points represent a mean value covering a grid cube. It is also important that the total size of the box is large enough to cover the different wake sources including their meandering path. The resolution in the u-direction should be as fine a possible. The other turbulence box that is needed is a box representing the micro scale turbulence from the wake of the upstream turbine. The resolution of this box should be fine (e.g. 128x128 points) in the v,w plane which should cover 1 rotor diameter. The resolution in the u direction should also be fine, but a short length of the box (e.g. min. 2.5Diameter) is OK.
The two turbulence boxed are included by the following sub commands

```
begin mann_meanderturb;
       (parameters are identical to the normal Mann turbulence box, see
       above)
end mann_meanderturb;

begin mann_microturb;
       (parameters are identical to the normal Mann turbulence box, see
       above)
end mann_microturb;
```

The rest of the wake commands are given in the following table.

Obl.	Command name	Explanation
*	nsource	1. Number of wake sources. If 0 is used the wake module is by-passed (no source positions can be given in this case).
*	source_pos	Command that must be repeated *nsource* times. This gives the position of the wake source in global coordinates. Wake source position can also be given for down stream turbines. These downstream turbines are however not used in the simulations. 1. x-pos [m] 2. y-pos [m] 3. z-pos [m]
*	op_data	Operational conditions for the wake sources. 1. Rotational speed [rad/s] 2. Collective pitch angle [deg]. Defined positive according to the blade root coo, with z-axis from root towards tip.
	ble_parameters	Parameters used for the BLE model used for developing the wake deficit due to turbulent mixing. 1. k_1 [-], default=0.001 2. k_2 [-], default=0.002 3. clean-up parameter (0=intermediate files are kept, 1=intermediate files are deleted), default=1
	microturb_factors	Paraemters used for scaling the added wake turbulence according to the deficit depth and depth derivative. 1. k_{m1} [-], factor on deficit depth, default=0.60 2. k_{m2} [-], factor on depth derivative, default=0.25
	tint_meander	Turbulence intensity of the meander turbulence box. If this command is not used then the default turbulence intensity from the general wind commands is used (normal use) 1. Turbulence intensity [-]

Sub command block – tower_shadow_potential

Block that must be included if the potential flow tower shadow model is chosen.

Obl.	Command name	Explanation
*	tower_offset	The tower shadow has its source at the global coordinate z axis. The offset is the base point for section 1 1. Offset value (default=0.0)
*	nsec	Command that needs to present before the radius commands. 1. Number of datasets specified be the radius command.
*	radius	Command that needs to be listed nsec times. 1. z coordinate [m] 2. Tower radius at z coordinate [m]

Sub command block – tower_shadow_jet

Block that must be included if the model based on the boundary layer equations for a jet is chosen. This model is especially suited for downwind simulations.

Obl.	Command name	Explanation
*	tower_offset	The tower shadow has its source at the global coordinate z axis. The offset is the base point for section 1 1. Offset value (default=0.0)
*	nsec	Command that needs to present before the radius commands. 1. Number of datasets specified be the radius command.
*	radius	Command that needs to be listed nsec times. 1. z coordinate [m] 2. Tower radius at z coordinate [m] 3. C_d drag coefficient of tower section (normally 1.0 for circular section, but this depends heavily on the reynold number)

Sub command block – tower_shadow_potential_2

Block that must be included if the tower shadow method 3 is chosen. This potential model is principally similar to the potential flow model described previously but differs in the way that the shadow source is moved and rotated in space as the tower coordinate system is moving and rotating.

The coordinate the shadow method is linked to is specified by the user, e.g. the mbdy coordinate from the tower main body. To make sure that the tower source model is always linked in the same way as the tower (could be tricky since the tower is fully free to be specified along the x,y or z axis or a combination) the base coordinate system for the shadow model is identical to the coordinates system obtained by the local element coordinates, where the z axis is always pointing from node 1 towards node 2. This is the reason that the tower radius input has to specified with positive z-values, see below.

Obl.	Command name	Explanation
*	tower_mbdy_link	Name of the main body to which the shadow source is linked. 1. mbdy name
*	nsec	Command that needs to present before the radius commands. 1. Number of datasets specified by the radius command.
*	radius	Command that needs to be listed nsec times. 1. z coordinate [m] (allways positive!) 2. Tower radius at z coordinate [m]

Aerodynamics

Main command block - aero

This module set up parameters for the aerodynamic specification of the rotor. It is also possible to submit aerodynamic forces to other structures as example the tower or nacelle, but see chapter (Aerodrag) regarding this.

Obl.	Command name	Explanation
*	nblades	Must be the first line in aero commands! 1. Number of blades
*	hub_vec	Link to main-body vector that points downwind from the rotor under normal conditions. For a upsteam turbine this corresponds to the direction from hub towards tower perpendicular to the rotor. 1. mbdy name or 'old_input' if old_htc_structure format is applied. 2. mbdy coo. component (1=x, 2=y, 3=z). If negative the opposite direction used. Not used together with old_htc_structure input (specify a dummy number).
*	link	Linker between structural blades and aerodynamic blades. There must be same number of link commands as nblades! 1. blade number 2. link chooser – options are • mbdy_c2_def (used with new structure format) • blade_c2_def (used with old structure format, see description below in this chapter) 3. mbdy name (with new structure format), not used to anything with old structure format.
*	ae_filename	1. Filename incl. relative path to file containing aerodynamic layout data (example ./data/hawc2_ae.dat)
*	pc_filename	1. Filename incl. relative path to file containing profile coefficients (example ./data/hawc2_pc.dat)
*	induction_method	1. Choice between which induction method that shall be used (0=none, 1=normal dynamic induction)
*	aerocalc_method	1. Choice between which aerodynamic load calculation method that shall be used. (0=none, 1=normal)
*	aerosections	Number of aerodynamic calculation points at a blade. The distribution is performed automatically using a cosinus transformation which gives closest spacing at root and tip. 1. Number of points at each blade.
*	ae_sets	Set number from ae_filename that is linked to blade 1,2,...,nblades 1. set for blade number 1 2. set for blade number 2 . . . nblades. set for blade number nblades
*	tiploss_method	1. Choice between which tip-loss model that shall be used (0=none, 1=prandtl (default))
*	dynstall_method	1. Choice between which dynamic stall model that shall be used (0=none, 1=Stig Øye method, 2=MHH Beddoes method)

Sub command block – dynstall_so

Block that may be included if the Stig Øye dynamic stall method is chosen. If not included defaults parameters are automatically used.

Obl.	Command name	Explanation
*	dclda	1. Linear slope coefficient for unseparated flow (default=6.28)
*	dcldas	1. Linear slope coefficient for fully separated flow (default=3.14)
*	alfs	1. Angle of attack [deg] where profile flow is fully separated. (default=40)
*	alrund	1. Factor used to generate synthetic separated flow Cl values (default=40)
*	taufak	1. Time constant factor in first order filter for F function (default=10.0). Internally used as tau=taufak*chord*vrel

Sub command block – dynstall_mhh

Block that may be included if the MHH Beddoes dynamic stall method is chosen. If not included defaults parameters are automatically used.

Obl.	Command name	Explanation
	a1	1. Coefficients of the exponential potential flow step response approximation: Phi(s)=1-A1*exp(-b1*s)-A2*exp(-b2*s). (default= 0.165)
	a2	1. Coefficients of the exponential potential flow step response approximation: Phi(s)=1-A1*exp(-b1*s)-A2*exp(-b2*s). (default= 0.335)
	b1	1. Coefficients of the exponential potential flow step response approximation: Phi(s)=1-A1*exp(-b1*s)-A2*exp(-b2*s). (default= 0.0455)
	b2	1. Coefficients of the exponential potential flow step response approximation: Phi(s)=1-A1*exp(-b1*s)-A2*exp(-b2*s). (default= b2=0.300)
	update	Choice between update methods: 1. 1 (default)=>update aerodynamics all iterations all timesteps; 0=>only update aerodynamics first iteration each new timestep
	taupre	1. Non-dimensional time-lag parameters modeling pressure time-lag. Default value =1.5
	taubly	1. Non-dimensional time-lag parameters modeling boundary layer time-lag. Default value=6.0
	only_potential_model	1. 0(default)=>run full MHH beddoes model; 1=>Potential flow model dynamics superposed to steady force coefficients;

Sub command block – dynstall_mhhmagf

Block that may be included if the MHHMAGF Beddoes dynamic stall method is chosen. The stall model is the mhhbeddoes model expanded with dynamic effects of trailing edge flap motion. If not included defaults parameters are automatically used.

Obl.	Command name	Explanation
*	flap	Command that must be repeated for each flap section defined 1. Non-dim radius r/R of flap section start, from blade root. 2. Non-dim radius r/R of flap section end, from blade root. 3. Filename incl. relative path to file containing α-delta C_l static profile coefficients. Fileformat is ASCII – two colums. The delta C_l marks the delta for one degree positive flap angle at various angles of attack.
	ais	Coefficients of the exponential potential flow step response approximation: 1. A1 (default= 0.0821) 2. A2 (default=0.1429) 3. A3 (default=0.3939) Default coefficients is derived for the Risø-B1-18 profile
	bis	Coefficients of the exponential potential flow step response approximation: 1. B1 2. B2 3. B3
	ti1	Camberline coefficients 1. TI1_a (default=0.01095889075152) 2. TI1_b (default=-0.00097224060418)
	ti2	Camberline coefficients 1. TI2_a (default=-0.00105409494045) 2. TI2_b (default=-0.00000964520546) 3. TI2_c (default=0.00011409945431) 4. TI2_d (default=-0.00000096469297)
	ti3	Camberline coefficients 1. TI3_a (default=-0.01823405820608) 2. TI3_b (default=-0.00043120871058) 3. TI3_c (default=-0.00042628415385) 4. TI3_d (default=-0.00004009691664)
	ti4	Camberline coefficients 1. TI4_a (default=-0.04288996443976) 2. TI4_b (default=-0.00090740475877)
	ti5	Camberline coefficients 1. TI5_a (default=-0.17732761097554) 2. TI5_b (default=0.00050518296019)
	ti6	Camberline coefficients 1. TI6_a (default=0.15479786740119) 2. TI6_b (default=0.00144695790428)
	ti7	Camberline coefficients 1. TI7_a (default=-0.20821609838649) 2. TI7_b (default=-0.01746039372665)
	ti8	Camberline coefficients 1. TI8_a (default=0.01329688411426) 2. TI8_b (default=0.00088185679919) 3. TI8_c (default=0.00737988450743) 4. TI8_d (default=0.00054605607792)
	ti9	Camberline coefficients 1. TI9_a (default=-0.02960508187800) 2. TI9_b (default=0.00001446048395) 3. TI9_c (default=-0.00211611339069) 4. TI9_d (default=0.00001171165409)

Obl.	Command name	Explanation
	hdydx	1. Camberline coef. (default=-0.07106384522900)
	hy	1. Camberline coef. (default=-0.00199404265933)
	fdydxle	1. Camberline coef. (default=0.00619732559359)
	gdydxle	1. Camberline coef. (default=0.00288436419056)
	gyle	1. Camberline coef. (default=0.00006407600471)
	update	Choice between update methods: 1. 1 (default)=>update aerodynamics all iterations all timesteps; 0=>only update aerodynamics first iteration each new timestep
	taupre	1. Non-dimensional time-lag parameters modeling pressure time-lag. Default value =1.5
	taubly	1. Non-dimensional time-lag parameters modeling boundary layer time-lag. Default value=6.0

Camberline coefficients used to specify the dynamics of the flap. These coefficients are given by the Gaunaa model. Default vales used are for the Risø B1-18 profile with a 10% chord length flap mounted.

Data format for the aerodynamic layout

The format of this file which in the old HAWC code was known as the hawc_ae file is changed slightly for the HAWC2 input format. The position of the aerodynamic center is no longer an input value, since the definition is that the center is located in $C_{1/4}$ with calculated velocities in $C_{3/4}$.

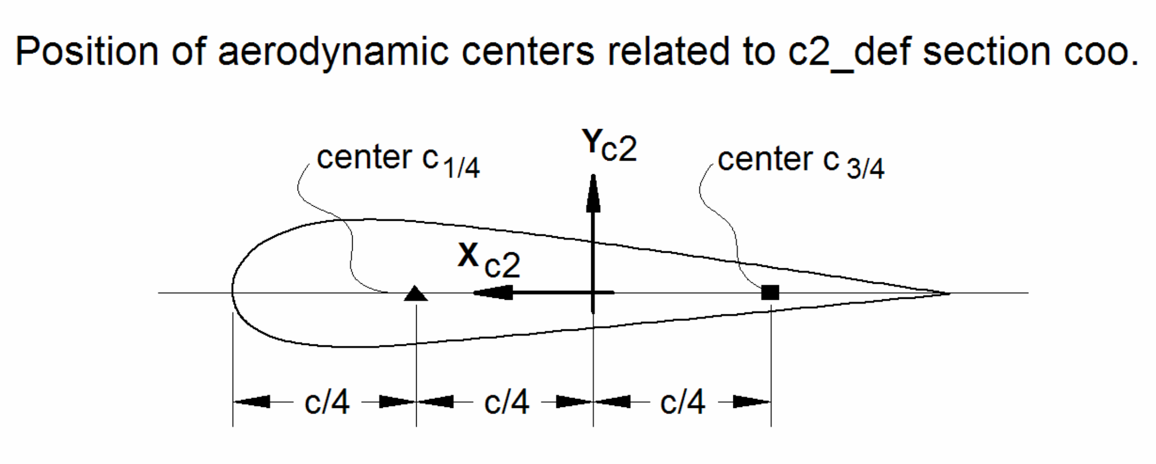


Figure 4: Illustration of aerodynamic centers c1/4 and c3/4

The format of the file is specified in the following two tables

Line number	Description
1	#1: Nset, Number of datasets present in the file. The format of ecah data set can be read below. The datasets are repated without blank lines etc.
2	#1: Set number. #2: Nrows, Number of data rows for this set
3..2+Nrows	Data row according to Table 4

Table 3: Format of main data structure for the aerodynamic blade layout file

The content of the colums in a data row is specified in the table below.

Column	Parameter
1	r, distance from main_body node 1 along z-coordinate [m]
2	chord length [m]
3	thickness ratio between profile height and chord [%]
4	Profile coefficient set number

Table 4 Format of the data rows for the aerodynamic blade layout file

Data format for the profile coefficients file

The format of this file which in the old HAWC code was known as the hawc_pc file has not been changed for the HAWC2 code.

The format of the file is specified in the following two tables

Line number	Description
1	#1: Nset, Number of datasets present in the file. The format of ecah data set can be read below. The datasets are repated without blank lines etc.
2	#1: Nprofiles. Number of profiles included in the data set.
3	#1: Set number. #2: Nrows. #3: Thickness in percent of chord length
4..3+Nrows	Data row according to Table ??

Table 5: Format of main data structure for the profile coefficients file

The content of the colums in a data row is specified in table below.

Column	Parameter
1	α, angle of attack [deg]. Starting with -180.0, ending with +180.0
2	C_l lift coefficient [-]
3	C_d drag coefficient [-]
4	C_m moment coefficient [-]

Table 6 Format of the data rows for the profile coefficients file

Main command block – blade_c2_def (for use with old_htc_structure format)

In this command block the definition of the centerline of the main_body is described (position of the half chord). This command shall be used as a main command even though it is only used together with the aerodynamic module. The reason for this is that it used to submit information that is usually given in the new_htc_structure format, which is also a main command block The input data given with the sec commands below is used to define a continous differentiable line in space using akima spline functions. This centerline is used as basis for local coordinate system definitions for sections along the structure. If a straight line is requested a minimum of three points of this line must be present.

Obl.	Command name	Explanation
*	nsec	Must be the present before a “sec” command. 1. Number of section commands given below
*	sec	Command that must be repeated “nsec” times 1. Number 2. x-pos [m] 3. y-pos [m] 4. z-pos [m] 5. θ_z [deg]. Angle between local x-axis and main_body x-axis in the main_body x-y coordinate plane. For a straight blade this angle is the aerodynamic twist. Note that the sign is positive around the z-axis, which is opposite to traditional notation for etc. a pitch angle.

Aerodrag (for tower and nacelle drag)

Main command aerodrag

With this module it is possible to apply aerodynamic drag forces at a given number of structures.

Subcommand aerodrag_element

Command block that can be repeated as many times as needed. In this command block aerodynamic drag calculation points are set up for a given main body.

Obl.	Command name	Explanation
*	body_name mbdy_name	1. Main_body name to which the hydrodynamic calculation points are linked.
*	aerodrag_sections	1. Distribution method: (“uniform” only possibility) 2. Number of calculation points (min. 2).
	nsec	This command must be present before the sec commands 1. Number of sections given below
	sec	This command must be repeated nsec times 1. Distance along the main_body c2_def line. Positive directed from node 1 to node “last”. Internally this distance is normalized with the distance for the last section so calculation points are ensured in the endpoints of the structure. Let the distance of the last point be 1.0 or same length as the main_body to avoid confusion. 2. C_d drag coefficient (default=1.0) 3. Width of structure (diameter)
	update_states	Logical parameter that determines whethe the movement of the structure is included or not. 1. parameter (1=states are updated (default), 0=not updated)

*) Input commands that must be present

Hydrodynamics

Main command block - hydro

In this command block hydrodynamic forces calculated using Morisons formula is set up.

Sub command block – water_properties

Obl.	Command name	Explanation
*	gravity	1. Gravity acceleration (used for calcultion of buoyancy forces). Default = 9.81 m/s^2
*	mudlevel	1. Mud level [m] in global z coordinates.
*	mwl	1. Mean water level [m] in global z coordinates.
*	rho	1. Density of the water [kg/m^3]. Default=1027
	water_kinematics_dll	1. Filename incl. relative path to file containing water kinematics dll (example ./hydro/water_kin.dll)

Sub sub command block – hydro_element

Command block that can be repeated as many times as needed. This command block set up hydrodynamic calculation points and link them to a main_body.

Obl.	Command name	Explanation
*	body_name mbdy_name	1. Main_body name to which the hydrodynamic calculation points are linked.
*	hydrosections	1. Distribution method of hydrodynamic calculation points. Options are: • "uniform" nnodes. Where uniform ensures equal distance of the calculation points. nnodes are number of calculation points.
*	nsec	This command must be present before the sec commands 1. Number of sections given below
*	sec	This command must be repeated nsec times 1. distance along the main_body c2_def line. Positive directed from node 1 to node "last". 2. C_m inertia coefficient (default=1.0) 3. C_d drag coefficient (default=1.0) 4. Cross sectional area 5. Cross sectional area to which C_m is related. (default=area for circular sections)
	buoyancy	1. Specification whether buoyancy forces are included or not. 0=off (default), 1=on
	update_states	1. Specification whether the hydrodynamic sections are updated in time with respect to pos,vel,acc and orientations, or simply considered to remain fixed. 0=not updated, 1=updated (default)

Description of the water_kinematics_dll format.

```
subroutine init(inputfile)
implicit none
character*(*) :: inputfile
!DEC$ ATTRIBUTES DLLEXPORT, ALIAS:'init'::init

end subroutine init

!---------------------------------------------------------------------------
subroutine set_new_time(time)
implicit none
!DEC$ ATTRIBUTES DLLEXPORT, ALIAS:'set_new_time'::set_new_time
real*8                :: time

end subroutine set_new_time

!---------------------------------------------------------------------------
subroutine get_sea_elevation(pos_xy,elevation)
implicit none
!DEC$ ATTRIBUTES DLLEXPORT, ALIAS:'get_sea_elevation'::get_sea_elevation
real*8,dimension(2) :: pos_xy         ! only present for future use together
with more complex wave fields
real*8            :: elevation

end subroutine get_sea_elevation

!---------------------------------------------------------------------------
subroutine get_kinematics(pos,vel,acc)
implicit none
!DEC$ ATTRIBUTES DLLEXPORT, ALIAS:'get_kinematics'::get_kinematics
real*8,dimension(3)            :: pos, &
                                  vel, &
                                  acc

end subroutine get_kinematics
```

Soil module

Main command block - soil

In this command block soil spring/damper forces can be attached to a main body. The formulation is performed so it can be used for other external distributed spring/damper systems than soil.

Sub command block – soil_element

Command block that can be repeated as many times as needed. In this command block the distributed soil spring/damper system is set up for a given main body.

Obl.	Command name	Explanation
*	body_name	1. Main_body name to which the hydrodynamic calculation points are linked.
*	datafile	1. Filename incl. relative path to file containing soil spring properties (example ./soil/soildata.dat)
*	soilsections	1. Distribution method: ("uniform" only possibility) 2. Number of section (min. 2).
	damping_k_factor	1. Rayleigh kind of damping. Factor the linear stiffness coefficients are multiplied with to obtain the damping coefficients. When the factor is 1.0 the vibration is critically damped for the rigid mainbody connected to the spring and dampers.
♣	set	1. Set number in datafile that is used.

*) Input commands that must be present

♣) Command can be repeated as many times as desired.

Data format of the soil spring datafile

In the file (which is a text file) different distributed springs can be defined. Each set is located after the "#" sign followed by the set number. Within a set the following data needs to be present.

line 1	"spring type"	(can be "axial", "lateral" or "rotation_z")
line 2:	"nrow ndefl"	(nrow is number of rows, ndefl is number of deflections (colums)
line 3..3+nrow	"z_global F(1) F(2),..., F(ndefl)"	First colum is the spring location (global z coordinate). The following colums are Force/length at the different deflection stations. First deflection must be zero. The forces are assumed symmetrical around the zero deflection.

An example is given below:

```
This is a nonlinear soil spring demonstration file
#1
lateral               (axial/lateral)
5 4                    nrow ndefl
       0.0     0.1     0.2     1.0    x1 x2 x3 .....   [m]
0.0    0       15      20      500    Z_G  F_1 F_2 F_3 .... F_ndefl [kN/m]
10.0   0       15      20      500
20.0   0       15      20      500
30.0   0       15      20      500
40.0   0       15      20      500
#2
axial                 (axial/lateral)
5 4                    nrow ndefl
       0.0     0.1     0.2     1.0    x1 x2 x3 .....   [m]
0.0    0       150     200     5000   Z_G  F_1 F_2 F_3 .... F_ndefl [kN/m]
10.0   0       150     200     5000
20.0   0       150     200     5000
30.0   0       150     200     5000
40.0   0       150     200     5000
#3
rotation_z            (axial/lateral/rotation_z)
5 4                    nrow ndefl
       0.0     0.1     0.2     1.0    x1 x2 x3 .....   [rad]
0.0    0       150     200     5000   Z_G M_1 M_2 M_3 .... M_ndefl [kNm/m]
10.0   0       150     200     5000
20.0   0       150     200     5000
30.0   0       150     200     5000
40.0   0       150     200     5000
```

Output

This command **output** can either be a main command block or a sub command block within the hawc_dll command block. In the tables below two special columns are introduced. One is *only option* and the other *label option*. When the check mark is 'yes' in *only option* it is possible to use only one of the fields if mre than one sensor was defined through the command. The sensor that is used is determined by the number following the *only* command word, see example below.

```
constraint bearing1 shaft_rot 2 only 2;
```

If the *only* command (and the following number) was omitted two sensors was defined; one for the angle and one for the velocity. With the *only* command only the velocity sensor is used in the output since the following number is 2.

With the label option it is possible to make a user defined label of the sensor which is written in the sensor list file. The label command is the # symbol. Everything after the # symbol is used as a label. An example of this could be

```
dll inpvec 1 1 # This is a dummy label ;
```

Commands used with results file writing

When the output command is used for output files (the most normal purpose) some information regarding file name and format needs to be give

Obl	Command	Explanation
*	filename	1. Filename incl. relative path to outputfile without extension (example ./res/output)
	data_format	ASCII or compressed binary output can be chosen. Default is the ASCII format if nothing is specified. 1. format ('hawc_ascii'=ASCII format, 'hawc_binary'=compressed binary format
	buffer	Buffer size in terms of time steps. When the buffer is full the data are written to data file. Only used togeter with the ASCII format. 1. buffer size
	time	Time start t_0 and stop t_1 for output is defined. Defult is the entire simulation length if nothin is specified. 1. t_0 2. t_1

File format of HAWC_ASCII files

Results are written to an ascii formatted data file with the name assigned to the filename variable (eg. filename ./res/resfil). The data file will have the extension .dat as a standard. The description of the sensors in the data file is given in another textfile with same filename as the data file but the extension .sel. An example could be: ./res/resfil.dat and ./res/resfil.sel.

In the .sel-file, line numer 9 specifies the following parameters: Number of scans, Number of sensors, Duration of output file, Data format (ASCII/BINARY). Example:

```
10 96 20.000 ASCII
```

From line number 13 and onwards, the sensors are specified with the following information:
Sensor number, Variable description, unit, Long description. Example:

```
5      bea1 angle_speed               rad/s      pitch1 angle speed
```

Full example of the .sel file:

```
________________________________________________________________________________
  Version ID : HAWC2MB 4.3w
                                                          Time : 14:23:28
                                                          Date : 22:11.2006
________________________________________________________________________________
  Result file : ./res2_rev0/case41c_nohydro.dat
________________________________________________________________________________
   Scans    Channels    Time [sec]      Format
     4500      199         90.000       ASCII

  Channel   Variable Description

     1      Time                           s          Time
     2      bea1 angle                     deg        shaft_rot angle
     3      bea1 angle_speed               rpm        shaft_rot angle speed
     4      bea1 angle                     deg        pitch1 angle
     5      bea1 angle_speed               rad/s      pitch1 angle speed
     6      bea1 angle                     deg        pitch2 angle
     7      bea1 angle_speed               rad/s      pitch2 angle speed
     8      bea1 angle                     deg        pitch3 angle
     9      bea1 angle_speed               rad/s      pitch3 angle speed
________________________________________________________________________________
```

File format of HAWC_BINARY files

In this file format results are written to a binary unformatted data file with the name assigned to the filename variable (eg. filename ./res/resfil). The data file will have the extension .dat as a standard. The description of the sensors in the data file is given in another textfile with same filename as the data file but the extension .sel. An example could be: ./res/resfil.dat and ./res/resfil.sel.

The data are scaled to standard 2-byte integers, with a range of 32000 using a scalefactor. The scalefactor is determined for each output sensor

$$s = \frac{MAX(abs(\max), abs(\min))}{32000}$$

where *max* and *min* are the largest and lowest number in the original data for the sensor. These scale factors are written in the end of the accompanying .sel file.

When converting a binary number to the actual number its just a matter of multiplying the binary numbers of a sensor with the corresponding scalefactor.

In the accompanying text file, which has the extension .sel-file, information of the content in the datafile is stored. In line number 9 the following parameters are specified: Number of scans, Number of sensors, Duration of output file, Data format (ASCII/BINARY). Example:

```
10 96 20.000 ASCII
```

From line number 13 and onwards, the sensors are specified with the following information:
Sensor number, Variable description, unit, Long description. Example:

```
5      bea1 angle_speed              rad/s      pitch1 angle speed
```

From line number 9+nsensors+5 and upwards the scalefactors are written.

Full example of the .sel file:

```
________________________________________________________________________________
  Version ID : HAWC2MB 4.3
                                                           Time : 14:23:28
                                                           Date : 22:11.2006
________________________________________________________________________________
  Result file : ./res2_rev0/case41c_nohydro.dat
________________________________________________________________________________
   Scans    Channels    Time [sec]      Format
     4500       9          90.000       ASCII

  Channel   Variable Description

     1      Time                           s          Time
     2      bea1 angle                     deg        shaft_rot angle
     3      bea1 angle_speed               rpm        shaft_rot angle speed
     4      bea1 angle                     deg        pitch1 angle
     5      bea1 angle_speed               rad/s      pitch1 angle speed
     6      bea1 angle                     deg        pitch2 angle
     7      bea1 angle_speed               rad/s      pitch2 angle speed
     8      bea1 angle                     deg        pitch3 angle
     9      bea1 angle_speed               rad/s      pitch3 angle speed
________________________________________________________________________________
Scale factors:
  1.56250E-04
  5.61731E-03
  4.41991E-04
  1.00000E+00
  1.00000E+00
  1.00000E+00
  1.00000E+00
  1.00000E+00
  1.00000E+00
```

An important thing to notice is that in the binary data file all sensors are stored sequentially, i.e. all data for sensor 1, all data for sensor 2, etc. This way of storing the data makes later reading of a sensor extra fast since all data for a sensor can be read without reading any data for the other sensor.

mbdy (main body related commands)

Command 1	Command 2	Explanation	Only option	Label option
mbdy	forcevec	F_x, F_y, F_z shear force vector defined to output. 1. Main_body name 2. Element number 3. Node number on element 4. Main_body name of which coordinate system is used for output. "global" and "local" can also be used. Local is around local beam main bending directions.	yes	yes
mbdy	momentvec	M_x, M_y, M_z moment vector defined to output. 1. Main_body name 2. Element number 3. Node number on element 4. Main_body name of which coordinate system is used for output. "global" and "local" can also be used. Local is around local beam main bending directions.	yes	yes
mbdy	state	Vector with 3 components of either position, velocity or acceleration of a point on an element defined to output. 1. State: 'pos', 'vel' or 'acc' 2. Main_body name 3. Element number 4. Relative distance from node 1 to node 2 on element 5. Main_body name of which coordinate system is used for output. "global" can also be used.	yes	yes
mbdy	state_at	Vector with 3 components of either position, velocity or acceleration of a point on an element defined to output. The point is offset from the element z axis by an x and y distance. 1. State: 'pos', 'vel' or 'acc' 2. Main_body name 3. Element number 4. Relative distance from node 1 to node 2 on element 5. Main_body name of which coordinate system is used for output. "global" can also be used. 6. x-coordinate offset [m] 7. y-coordinate offset [m]	yes	yes

Command 1	Command 2	Explanation	Only option	Label option
mbdy	state_rot	Vector with components of either axis and angle (angle [rad], r_1,r_2,r_3), euler parameters (quaternions r_0,r_1,r_2,r_3), euler angles, rotation velocity (ω -vector) or rotation acceleration ($\dot{\omega}$ -vector) of a point on an element defined to output. For the sensor eulerang_xyx a set of euler angles are created based on the orientation matrix. Be aware that the method used is only valid for rotations in the intervals (θ_x ±180°, θ_y ±90°, θ_x ±180°) 1. State : ‘axisangle’, ‘eulerp’, ’eulerang_xyz’, ‘omega’ or ‘emegadot’ 2. Main_body name 3. Element number 4. Relative distance from node 1 to node 2 on element 5. Main_body name of which coordinate system is used for output. “global” can also be used.	yes	yes

Constraint (constraint related commands)

bearing1

Command 1	Command 2	Explanation	Only option	Label option
constraint	bearing1	Bearing angle and angle velocity defined to output 1. bearing1 name 2. unit of output (1:angle [unit=rad, range -π:π], vel [rad/s]; 2:angle [unit=deg, range 0:360], vel [rpm]; 3:angle [unit=deg, range 0:360], vel [rad/s]); 4:angle [unit=deg, range -180:180], vel [rad/s]; 5:angle [unit=deg, range -180:180], vel [deg/s])	Yes	No

bearing2

Command 1	Command 2	Explanation	Only option	Label option
constraint	bearing2	Bearing angle and angle velocity defined to output 1. bearing1 name 2. unit of output (1:angle [unit=rad, range -π:π], vel [rad/s]; 2:angle [unit=deg, range 0:360], vel [rpm]; 3:angle [unit=deg, range 0:360], vel [rad/s]); 4:angle [unit=deg, range -180:180], vel [rad/s]; 5:angle [unit=deg, range -180:180], vel [deg/s])	Yes	No

bearing3

Command 1	Command 2	Explanation	Only option	Label option
constraint	bearing3	Bearing angle and angle velocity defined to output 1. bearing1 name 2. unit of output (1:angle [unit=rad, range $-\pi:\pi$], vel [rad/s]; 2:angle [unit=deg, range 0:360], vel [rpm]; 3:angle [unit=deg, range 0:360], vel [rad/s]); 4:angle [unit=deg, range -180:180], vel [rad/s]; 5:angle [unit=deg, range -180:180], vel [deg/s])	Yes	No

body (old body related commands)

Command 1	Command 2	Explanation	Label option
body	forcevec	F_x, F_y, F_z shear force vector defined to output. Unit [kN] 1. body number 2. Element number 3. Node number on element 4. coordinate system (1=body, 2=global, 3=element)	No
body	momentvec	M_x, M_y, M_z moment vector defined to output. Unit [kNm] 1. body number 2. Element number 3. Node number on element 4. coordinate system (1=body, 2=global, 3=element)	No
body	node_defl	x,y,z deflection vector (within a body) defined to output. Unit [m] 1. body number 2. Element number 3. Node number on element 4. coordinate system (1=body, 2=global, 3=element)	No
body	node_rot	θ_x, θ_y, θ_z, rotations (within a body) define to output. Unit [rad] 1. body number 2. Element number 3. Node number on element 4. coordinate system (1=body, 2=global, 3=element)	No
body	pitchangle	Pitchangle of pitch bearing defined with the old_htc_structure is defined to output. 1. Unit (1=[rad], 2=[deg] 2. Pitch bearing number	No
body	pitchspeed	Pitch velocity of pitch bearing defined with the old_htc_structure is defined to output. 1. Unit (1=[rad/s], 2=[deg/s] 2. Pitch bearing number	No
body	node_state	State vector (position, velocity or accelertion) of a given on an element is defined to output. 1. state ("pos"=position, "vel"=velocity, "acc"=acceleration) 2. body name 3. element number 4. z_{rel} (distance between node 1 and 2 divided by element length) 5. coordinate system (1=global)	No

aero (aerodynamic related commands)

Command 1	Command 2	Explanation	Label option
aero	time	Simulation time to output. No parameters.	No
aero	azimuth	Azimuth angle of selected blade. Zero is vertical downwards. Positive clockwise around blade root y-axis. Unit [deg] 1. Blade number	No
aero	omega	Rotational speed of rotor. Unit [rad/s]	No
aero	vrel	Relative velocity in x-y local aerodynamic plane. Unit [m/s] 1. Blade number 2. Radius [m] (nearest inner calculation point is used)	No
aero	alfa	Angle of attack in x-y local aerodynamic plane. Unit [deg] 1. Blade number 2. Radius [m] (nearest inner calculation point is used)	No
aero	beta	Flap deflection angle in x-y local aerodynamic plane. Unit [deg] 3. Blade number 4. Flap number as specified in the dynstall_mhhmagf section starting with 1	No
aero	cl	Instantaneous lift coefficient. Unit [-] 1. Blade number 2. Radius [m] (nearest inner calculation point is used)	No
aero	cd	Instantaneous drag coefficient. Unit [-] 1. Blade number 2. Radius [m] (nearest inner calculation point is used)	No
aero	cm	Instantaneous moment coefficient. Unit [-] 1. Blade number 2. Radius [m] (nearest inner calculation point is used)	No
aero	lift	Lift force at calculation point. Unit [kN/m] 1. Blade number 2. Radius [m] (nearest inner calculation point is used)	No
aero	drag	Drag force at calculation point. Unit [kN/m] 1. Blade number 2. Radius [m] (nearest inner calculation point is used)	No
aero	moment	Aerodynamic moment at calculation point. Unit [kNm/m] 1. Blade number 2. Radius [m] (nearest inner calculation point is used)	No
aero	secforce	Aerodynamic force at calculation point. Local aero coo. Unit [kN/m] 1. Blade number 2. Dof number (1=F_x, 2=F_y, 3=F_z) 3. Radius [m] (nearest inner calculation point is used)	No

Command 1	Command 2	Explanation	Label option
aero	secmoment	Aerodynamic moment at calculation point. Local aero coo. Unit [kN/m] 1. Blade number 2. Dof number (1=M_x, 2=M_y, 3=M_z) 3. Radius [m] (nearest inner calculation point is used)	No
aero	int_force	Integrated aerodynamic forces from tip to calculational point. NB the integration is performed around the $C_{3/4}$ location. Unit [kN] 1. Coordinates system (1=local aero coo, 2=blade ref. system, 3=global, 4=rotor polar) 2. Blade number 3. Dof number (1=M_x, 2=M_y, 3=M_z) 4. Radius [m] (nearest inner calculation point is used)	No
aero	int_moment	Integrated aerodynamic moment from tip to calculational point. NB the integration is performed around the $C_{3/4}$ location. Unit [kN] 1. Coordinates system (1=local aero coo, 2=blade ref. system, 3=global, 4=rotor polar) 2. Blade number 3. Dof number (1=M_x, 2=M_y, 3=M_z) 4. Radius [m] (nearest inner calculation point is used)	No
aero	torque	Integrated aerodynamic forces of all blades to rotor torsion. Unit [kNm]. No parameters	No
aero	thrust	Integrated aerodynamic forces of all blades to rotor thrust. Unit [kN]. No parameters	No
aero	position	Position of calculation point. Unit [m]. 1. Coordinates system (1=local aero coo, 2=blade ref. system, 3=global, 4=rotor polar) 2. Blade number 3. Dof number (1=M_x, 2=M_y, 3=M_z) 4. Radius [m] (nearest inner calculation point is used)	No
aero	rotation	Orientation of calculation point. Unit [deg]. 1. Coordinates system (1=blade_ref. coo, 2=rotor polar coo.) 2. Blade number 3. Dof number (1=θ_x, 2=θ_y, 3=θ_z) 4. Radius [m] (nearest inner calculation point is used)	No
aero	velocity	Velocity of calculation point. Unit [m/s]. 1. Coordinates system (1=local aero coo, 2=blade ref. system, 3=global, 4=rotor polar) 2. Blade number 3. Dof number (1= V_x, 2=V_y, 3=V_z) 4. Radius [m] (nearest inner calculation point is used)	No

Command 1	Command 2	Explanation	Label option
aero	acceleration	Acceleration of calculation point. Unit [m/s^2]. 1. Coordinates system (1=local aero coo, 2=blade ref. system, 3=global, 4=rotor polar) 2. Blade number 3. Dof number (1= V_x, 2=V_y, 3=V_z) 4. Radius [m] (nearest inner calculation point is used)	No
aero	windspeed	Free wind speed seen from the blade. Unit [m/s] 1. Coordinates system (1=local aero coo, 2=blade ref. system, 3=global, 4=rotor polar) 2. Blade number 3. Dof number (1= V_x, 2=V_y, 3=V_z) 4. Radius [m] (nearest inner calculation point is used)	No
aero	induc	Local induced velocity at calculation point. Unit [m/s] 1. Coordinates system (1=local aero coo, 2=blade ref. system, 3=global, 4=rotor polar) 2. Blade number 3. Dof number (1= V_x, 2=V_y, 3=V_z) 4. Radius [m] (nearest inner calculation point is used)	No
aero	induc_sector_ct	Thrust coefficient at a position on the rotor. Unit [-] 1. Radius [m/s] 2. Azimuth angle (zero downwards) [deg]	No
aero	induc_sector_cq	Torque coefficient at a position on the rotor. Unit [-] 1. Radius [m/s] 2. Azimuth angle (zero downwards) [deg]	No
aero	induc_sector_a	Axial induction coefficient at a position on the rotor. Unit [-] 1. Radius [m/s] 2. Azimuth angle (zero downwards) [deg]	No
aero	induc_sector_am	Tangential induction coefficient at a position on the rotor. Unit [-] 1. Radius [m/s] 2. Azimuth angle (zero downwards) [deg]	No
aero	induc_a_norm	Axial velocity used in normalization expression of rotor thrust coefficients. The average axial wind velocity incl. induction. Unit [m/s]. No parameters.	No
aero	induc_am_norm	Tangential velocity used in normalization expression of torque coefficient. Average tangential velocity at a given radius. Unit [m/s]. 1. Radius [m]	No

Command 1	Command 2	Explanation	Label option
aero	inflow_angle	Angle of attack + rotation angle of profile related to polar coordinates (not pitching). Unit [deg] 1. Blade number 2. Radius [m] (nearest inner calculation point is used)	No
aero	dcldalfa	Gradient $dCl/d\alpha$. Unit [deg^{-1}] 1. Blade number 2. Radius [m] (nearest inner calculation point is used)	No
aero	dcddalfa	Gradient $dCd/d\alpha$. Unit [deg^{-1}] 1. Blade number 2. Radius [m] (nearest inner calculation point is used)	No

wind (wind related commands)

Command 1	Command 2	Explanation	Only option	Label option
wind	free_wind	Wind vector V_x, V_y, V_z, (wind as if the turbine didn't exist). 1. Coordinate system (1=global, 2=non rotating rotor coordinates (x always horizontal, y always out-of-plane)) 2. x-pos (global coo) 3. y-pos (global coo) 4. z-pos (global coo)	Yes	No
wind	free_wind_hor	Horizontal wind component velocity [m/s] and direction [deg] defined to output. Dir=0 when wind equals y-dir. 1. Coordinate system (1=global, 2=non rotating rotor coordinates (x always horizontal, y always out-of-plane)) 2. x-pos (global coo) 3. y-pos (global coo) 4. z-pos (global coo)	Yes	No

wind_wake (wind wake related commands)

Command 1	Command 2	Explanation	Only option	Label option
wind_wake	wake_pos	Position of the wake deficit center after the meandering proces to the downstream end position. x,y and z position is written in meteorological coordinates $(x,y,z)_M=(u,v,w)$ with origo in the position defined with center_pos0 in the general wind commands. 1. wake source number	Yes	No

dll (DLL related commands)

Command 1	Command 2	Explanation	Label option
dll	inpvec	Value from DLL input vector is defined to output 1. DLL number 2. array index number	yes
dll	outvec	Value from DLL output vector is defined to output 1. DLL number 2. array index number	yes

hydro (hydrodynamic related commands)

Command 1	Command 2	Explanation	Only option	Label option
hydro	water_surface	Water surface level at a given horizontal location is defined to output (global coordinates). Unit [m] 1. x-pos 2. y-pos	No	No
hydro	water_vel_acc	Water velocity V_x, V_y, V_z, and acceleration A_x, A_y, A_z vectors defined to output. Unit [m/s] and [m/s^2]. 1. x-pos 2. y-pos 3. z-pos	Yes	No
hydro	fm	Inertia force F_x, F_y, F_z contribution from Morisons formula in a given calculation point. Unit [kN] 1. hydro element number 2. sec number 3. coordinate system (1=global)	Yes	No
hydro	fd	Drag force F_x, F_y, F_z contribution from Morisons formula in a given calculation point. Unit [kN] 1. hydro element number 2. sec number 3. coordinate system (1=global)	Yes	No

general (general output commands)

Command 1	Command 2	Explanation	Label option
general	constant	A constant value is send to output 1. constant value	No
general	step	A step function is created. This function changes from f_0 to f_1 at time t_0. 1. t_0 [sec] 2. f_0 3. f_1	No
general	stairs	A series of steps are created (a staircase) 1. f_0 start value 2. Stepsize 3. Step duration [s] 4. Number of steps	No
general	time	The time is send to output. No parameters	No
general	deltat	The time increment is send to output. No parameters	No
general	harmonic	A harmonic function is send to output $F(t) = A\sin(2\pi f_0 t) + k$ 1. A 2. f_0 3. k	No

Command 1	Command 2	Explanation	Label option
general	harmonic2	A harmonic function is send to output $$F(t)=\begin{cases}0 & t<t_0\\ A\sin(2\pi f_0(t-t_0))+k & t_0\le t\le t_1\\ 0 & t>t_1\end{cases}$$ 1. A 2. f_0 3. k 4. t_0 5. t_1	No
general	stairs	A series of steps resulting in a staircase signal is created. 1. f_0 start value of function 2. t_0 time for first step change [s] 3. Step size 4. Step duration [s] 5. Number of steps	No

Output_at_time (output at a given time)

This command is especially usefull if a snapshot of loads or other properties are required at a specific time. This is mostly used for writing calculated aerodynamic properties as function of blade location. The command block can be repeated as many times as needed (e.g. if outputs at more than one time is needed)

The command must be written with the following syntax
`output_at_time` *`keyword time`*
where *`keyword`* is a command listed in the subsections below. Sofar only the command `aero` is present. The last command word *`time`* is the time in seconds from simulation start to which the output are written.

aero (aerodynamic output commands)

The first line in the output_at block must be the information regarding which file the outputs are written (the filename command listed in the table below)

Command 1	Explanation	Label option
filename	Filename incl. relative path to output file (example ./output/output_at.dat). 1. filename	No
alfa	Angle of attack [deg]. 1. Blade number	No
vrel	Relative velocity [m/s] 1. Blade number	No
cl	Lift coefficient [-] 1. Blade number	No
cd	Drag coefficient [-] 1. Blade number	No
cm	Moment coefficient [-] 1. Blade number	No
lift	Lift force L [N] 1. Blade number	No
drag	Drag force D [N] 1. Blade number	No

Command 1	Explanation	Label option
moment	Moment force M [Nm] 1. Blade number	No
secforce	Aerodynamic forces [N] 1. Blade number 2. DOF number (1=x,2=y,3=z) 3. Coordinate system (1=aero, 2=blade, 3=global, 4=rotor polar)	No
secmoment	Aerodynamic moments [Nm] 1. Blade number 2. DOF number (1=x,2=y,3=z) 3. Coordinate system (1=aero, 2=blade, 3=global, 4=rotor polar)	No
int_force	Aerodynamic forces integrated from tip to given radius [N] 1. Blade number 2. DOF number (1=x,2=y,3=z) 3. Coordinate system (1=aero, 2=blade, 3=global, 4=rotor polar)	No
int_moment	Aerodynamic moment integrated from tip to given radius [N] 1. Blade number 2. DOF number (1=x,2=y,3=z) 3. Coordinate system (1=aero, 2=blade, 3=global, 4=rotor polar)	No
inipos	Initial position of sections in blade coo [m] 1. Blade number 2. DOF number (1=x,2=y,3=z)	No
position	Actual position of section [m] 1. Blade number 2. DOF number (1=x,2=y,3=z) 3. Coordinate system (1=aero, 2=blade, 3=global, 4=rotor polar)	No
velocity	Actual velocity of section [m/s] 1. Blade number 2. DOF number (1=x,2=y,3=z) 3. Coordinate system (1=aero, 2=blade, 3=global, 4=rotor polar)	No
acceleration	Actual acceleration of section [m/s] 1. Blade number 2. DOF number (1=x,2=y,3=z) 3. Coordinate system (1=aero, 2=blade, 3=global, 4=rotor polar)	No
ct_local	Local thrust coefficient [-]. Calculated based on the expression $C_t = \frac{V_r^2 F_{axial} c B}{2\pi r V_{\inf}}$ 1. Blade number	No
cq_local	Local thrust coefficient [-]. Calculated based on the expression $C_q = \frac{V_r^2 F_{\tan} c B}{2\pi r V_{\inf}}$ 1. Blade number	No
chord	Chord length [m] 1. Blade number	No
induc	Induced velocity [m/s] 1. Blade number 2. DOF number (1=x,2=y,3=z) 3. Coordinate system (1=aero, 2=blade, 3=global, 4=rotor polar)	No

Command 1	Explanation	Label option
windspeed	Free windspeed (without induction but incl. tower shadow effects if used) [m/s] 1. Blade number 2. DOF number (1=x,2=y,3=z) 3. Coordinate system (1=aero, 2=blade, 3=global, 4=rotor polar)	No
inflow_angle	Angle of attack + rotation angle of profile related to polar coordinates (not pitching). Unit [deg] 1. Blade number	No
dcldalfa	Gradient $dCl/d\alpha$. Unit [deg^{-1}] 1. Blade number	No
dcddalfa	Gradient $dCd/d\alpha$. Unit [deg^{-1}] 1. Blade number	No

Example of main input file

```
; Fictitious 2MW Turbine for wake simulationes
;
begin Simulation;
  time_stop    625.00 ;
  solvertype   1 ;    (newmark)
  animation    ./anim/anim_2MW_step.dat;
;
  begin newmark;
    beta      0.27;
    gamma     0.51;
    deltat    0.02;
    bdynamic  1.0 ;
  end newmark;
end simulation;
;----------------------------------------------------------
begin new_htc_structure;
  beam_output_file_name   ./info/2MW_beam.txt ;
  body_output_file_name   ./info/2MW_body.txt ;
  body_eigenanalysis_file_name ./info/body_eigen.dat ;
;
  begin main_body;          tower
    name        tower ;
    type        timoschenko ;
    nbodies     1 ;
    node_distribution     uniform 10 ;
    damping   0.02  0.02  0.02   0.0022  0.0022 0.0007 ;
    begin timoschenko_input;
      filename ./data/hawc_st.001 ;
      set 3 1;  set subset
    end timoschenko_input;
    begin c2_def;
          nsec      10        ;
          sec  1   0   0   0.000 0.00; Ground BC element start
          sec  2  0   0  -0.050 0.00; Ground BC element end
          sec  3  0   0  -3.000 0.00; Foundation top
          sec  4  0   0  -3.875 0.00; Lower flange
          sec  5  0   0 -13.020 0.00;
          sec  6  0   0 -25.000 0.00; Mid flange
          sec  7  0   0 -37.040 0.00;
          sec  8  0   0 -49.000 0.00;
          sec  9  0   0 -58.290 0.00; Nacelle element start
          sec 10  0   0 -59.890 0.00; Tower top
   end c2_def;
 end main_body;
;----------------------------------------------------------
  begin main_body;
    name        shaft ;
    type        timoschenko ;
    nbodies     1 ;
    node_distribution     c2_def ;
    damping   0.03  0.03  0.03   0.005   0.005  0.005 ;
    begin timoschenko_input;
           filename ./data/hawc_st.001 ;
      set 2 1 ;
    end timoschenko_input;
    begin c2_def;
          nsec      5         ;
          sec 1    0.0        0.0    0.000   0.0 ; Tower top
          sec 2   0.0         0.0    0.500   0.0 ; Gearbox
          sec 3   0.0         0.0    1.840   0.0 ; Main bearing
          sec 4   0.0         0.0    2.582   0.0 ; Hub start
          sec 5   0.0         0.0    4.030   0.0 ; Rotor center
    end c2_def;
  end main_body;
;----------------------------------------------------------
  begin main_body;
    name        blade1 ;
    type        timoschenko ;
    nbodies     4 ;
    node_distribution     c2_def ;
    damping   0.028 0.042 0.009  0.00023 0.0002 0.0002 ;
    begin timoschenko_input;
      filename ./data/hawc_st.001 ;
      set 1 1 ;                set subset
    end timoschenko_input;
    begin c2_def;
          nsec      15        ;
          sec       1  0.000  0.000     0.000     0.000 ;
          sec       2  0.000  0.000     1.031     0.000 ;
          sec       3  0.000  0.000     1.240     0.000 ;
          sec       4  0.000  0.000     3.08      -2.00 ;
          sec       5  0.000  0.000     5.240     -6.690 ;
          sec       6  0.000  0.000     9.240     -9.110 ;
          sec       7  0.000  0.000     13.240    -9.390 ;
          sec       8  0.000  0.000     17.240    -5.450 ;
          sec       9  0.000  0.000     20.440    -3.840 ;
          sec       10  0.000           0.000     24.060    -2.860 ;
          sec       11 0.000            0.000    29.240     -1.280 ;
          sec       12 0.000            0.000    35.000     -0.230 ;
          sec       13 0.000            0.000    37.240     -0.030 ;
          sec       14 0.000            0.000    39.240     -0.930 ;
          sec       15 0.000            0.000    40.040     -6.130 ;
    end c2_def;
  end main_body;
;----------------------------------------------------------
```

```
  begin main_body;
    name           blade2 ;
    copy_main_body blade1;
  end main_body;
;-----------------------------------------------------------
  begin main_body;
    name           blade3 ;
    copy_main_body blade1 ;
  end main_body;
;-----------------------------------------------------------
;-----------------------------------------------------------
  begin orientation;
    begin base;
      body   tower;
      inipos        0.0 0.0 0.0 ;         initial position of node 1
    end base;
;-----------------------------------------------------------
    begin relative;
      body1  tower last;         only last is valid!
      body2  shaft 1;
      body2_eulerang 90.0 0.0 0.0;    horizontal position
      body2_eulerang 5.0 0.0 0.0;     5 degrees tilt
      body2_ini_rotvec_d1 0.0 0.0 -1.0 1.3 ; body ini. rot. vel. x,y,z,angle vel.[rad/s]  (body 2
coo.)
    end relative;
;-----------------------------------------------------------
    begin relative;
      body1  shaft last;         only last is valid!
      body2  blade1 1;
      body2_eulerang -90.0 0.0 0.0;    blade 1 downwards
    end relative;
;-----------------------------------------------------------
    begin relative;
      body1  shaft last;         only last is valid!
      body2  blade2 1;
      body2_eulerang 0.0 0.0 120.0;    Blade passage nr 2
      body2_eulerang -90.0 0.0 0.0;
    end relative;
;-----------------------------------------------------------
    begin relative;
      body1  shaft last;         only last is valid!
      body2  blade3 1;
      body2_eulerang 0.0 0.0 -120.0;    Blade passage nr 3
      body2_eulerang -90.0 0.0 0.0;
    end relative;
  end orientation;
;-----------------------------------------------------------
;-----------------------------------------------------------
  begin constraint;
    begin fix0;  fixed to ground in translation and rotation of node 1
      body tower;
    end fix0;
;-----------------------------------------------------------
    begin bearing1;                       free bearing
      name  shaft_rot ;
      body1 tower last;
      body2 shaft 1;
      bearing_vector 2 0.0 0.0 -1.0;        x=coo (0=global,1=body1,2=body2) vector in body2 coo.
    end bearing1;
;-----------------------------------------------------------
;    begin bearing3;                       Prescribed rotation speed
;      name  shaft_rot ;
;      body1 tower last;
;      body2 shaft 1;
;      bearing_vector 2 0.0 0.0 -1.0;        x=coo (0=global,1=body1,2=body2) vector in body2 coo.
;      omegas 1.236 ;
;    end bearing3;
;-----------------------------------------------------------
    begin bearing2;                       forced bearing
      name pitch1;
      body1 shaft last;
      body2 blade1 1;
      bearing_vector 2 0.0 0.0 -1.0;        x=coo (0=global,1=body1,2=body2) vector in body2 coo.
    end bearing2;
;-----------------------------------------------------------
    begin bearing2;                       forced bearing
      name pitch2;
      body1 shaft last;
      body2 blade2 1;
      bearing_vector 2 0.0 0.0 -1.0;        x=coo (0=global,1=body1,2=body2) vector in body2 coo.
    end bearing2;
;-----------------------------------------------------------
    begin bearing2;                       forced bearing
      name pitch3;
      body1 shaft last;
      body2 blade3 1;
      bearing_vector 2 0.0 0.0 -1.0;        x=coo (0=global,1=body1,2=body2) vector in body2 coo.
    end bearing2;
  end constraint;
;
end new_htc_structure;
;-----------------------------------------------------------
;-----------------------------------------------------------
begin wind ;
  density                1.25 ;
  wsp                    5.75  ;
  horizontal_input       1     ;
  windfield_rotations    8.0  0.0  0.0 ;
  center_pos0            0.0 0.0 -59.89 ;  hub_height
  shear_format           1 0.1  ;
  turb_format            1 ;                    (0=no turbulence, 1:mann, 2:flex)
  tower_shadow_method    1     ;
  tint                   0.03 ;
```

```
;-------------------------------------------------------
  begin wakes;
    nsource 1;
    source_pos                0.0 -280.0 -59.89;  3.5 D
    ble_parameters  0.001 0.0012 0 ; k1 k2  delete file
    op_data         1.3 0.0 ; rad/sec, pitch [grader] opstrøms;
;-------------------------------------------------------
    begin mann_meanderturb ;
      filename_v    .\wake-meander\meander_8_6v.bin ;
      filename_w    .\wake-meander\meander_8_6w.bin ;
      box_dim_u   8192 0.732421875 ;
      box_dim_v     32 80 ;
      box_dim_w     32 80 ;
      std_scaling   1.0 0.8 0.5 ;
    end mann_meanderturb;
;-------------------------------------------------------
    begin mann_microturb ;
      filename_u    .\wake-turbulence\wake-l08_6u.bin ;    wake-turbulence
      filename_v    .\wake-turbulence\wake-l08_6v.bin ;
      filename_w    .\wake-turbulence\wake-l08_6w.bin ;
      box_dim_u   128 1.5625 ;
      box_dim_v     128 0.78125 ;
      box_dim_w     128  0.78125 ;
      std_scaling   1.0 1.0 1.0 ;
    end mann_microturb;
  end wakes;
;-------------------------------------------------------
  begin mann;
    filename_u    .\turb\80m_8ms_8u.bin ;
    filename_v    .\turb\80m_8ms_8v.bin ;
    filename_w    .\turb\80m_8ms_8w.bin ;
    box_dim_u   8192 0.732421875 ;
    box_dim_v     32 2.5625 ;
    box_dim_w     32 2.5625 ;
    std_scaling   1.0 0.8 0.5 ;
  end mann;
;-------------------------------------------------------
  begin tower_shadow_potential;
    tower_offset 0.0 ;
    nsec  2;
    radius      0.0 2.1 ;
    radius    -80.0 1.25 ;
  end tower_shadow_potential;
end wind;
;-------------------------------------------------------
;-------------------------------------------------------
begin aero ;
  nblades  3;
  hub_vec shaft -3 ;          vector from hub (normal to rotor plane) directed towards tower top
  link 1 mbdy_c2_def blade1;
  link 2 mbdy_c2_def blade2;
  link 3 mbdy_c2_def blade3;
  ae_filename        ./data/hawc_ae.002 ;
  pc_filename        ./data/hawc_pc.388 ;
  induction_method   1 ;     0=none, 1=normal
  aerocalc_method    1 ;     0=ingen aerodynamic, 1=med aerodynamic
  aerosections       30 ;
  ae_sets            1 1 1;
  tiploss_method     1 ;     0=none, 1=normal
  dynstall_method    2 ;     0=none, 1=stig øye method,2=mhh method
end aero ;
;-------------------------------------------------------
;-------------------------------------------------------
begin dll;
  begin hawc_dll;
    filename  ./control/basic_3ba_ct5.dll ;
    dll_subroutine regulation ;
    arraysizes  15 15 ;
    deltat  0.02;
    begin output;
      general time ;                                                1
      constraint bearing1 shaft_rot 1 only 2;                       2
      constraint bearing2 pitch1 1 only 1;  angle written to dll    3
      constraint bearing2 pitch2 1 only 1;  angle written to dll    4
      constraint bearing2 pitch3 1 only 1;  angle written to dll    5
      wind free_wind 1 0.0 0.0 -120.0;                              6,7,8
      general constant 1.44   ; Kp  pitch                           9
      general constant 0.47   ; Ki   pitch                          10
      general constant 0.00   ; Kd   pitch                          11
      general constant 4.30e6 ; Kp   torque                         12
      general constant 9.66e5 ; Ki   torque                         13
      general constant 0.0    ; Kd   torque                         14
    end output;
  end hawc_dll;
;-------------------------------------------------------
  begin hawc_dll;
    filename  ./control/basic_3ba_ct5.dll ;
    dll_subroutine generator ;
    arraysizes  15 15 ;
    deltat    0.02 ;
    begin output;
      general time ;
      dll inpvec 1 1; input til h2, dll no 1, plads no 1
      general constant 0.0;
      constraint bearing1 shaft_rot 1 only 2;
    end output;
;
    begin actions;
      body moment_int shaft 1 3 tower 10 ;  generator moment between shaft n1 My and tower top
    end actions;
  end hawc_dll;
;-------------------------------------------------------
  begin hawc_dll;
```

```
    filename  ./control/basic_3ba_ct5.dll ;
    dll_subroutine pitchservo ;
    arraysizes  15 15 ;
    deltat    0.02 ;
    begin output;
      general time ;
;      general step 5.0 0.0 0.02 ;
      dll inpvec 1 2;
      dll inpvec 1 3;
      dll inpvec 1 4;
      constraint bearing2 pitch1 1 only 1 ;  angle written to dll         5
      constraint bearing2 pitch2 1 only 1 ;  angle written to dll         6
      constraint bearing2 pitch3 1 only 1 ;  angle written to dll         7
    end output;
;
    begin actions;
      body bearing_angle pitch1;
      body bearing_angle pitch2;
      body bearing_angle pitch3;
    end actions;
  end hawc_dll;
end dll;
;----------------------------------------------------------
;----------------------------------------------------------
begin output;
  filename ./res/2MW-wake ;
  data_format hawc_ascii;
  buffer 1 ;
;
  general time;
  aero azimuth 1;
  aero omega ;
  aero thrust ;
  aero power;
  wind free_wind 1 -80.0 0.0 -60.0;
  wind free_wind 1 -60.0 0.0 -60.0;
  wind free_wind 1 -40.0 0.0 -60.0;
  wind free_wind 1 -20.0 0.0 -60.0;
  wind free_wind 1 0.0 0.0 -60.0;
  wind free_wind 1 20.0 0.0 -60.0;
  wind free_wind 1 40.0 0.0 -60.0;
  wind free_wind 1 60.0 0.0 -60.0;
  wind free_wind 1 80.0 0.0 -60.0;
  aero alfa 1 10.0 ;
  aero alfa 1 20.0 ;
  aero alfa 1 24.0 ;
  aero alfa 1 30.0 ;
  aero alfa 1 39.0 ;
  aero alfa 2 24.0 ;
  aero alfa 3 24.0 ;
  aero vrel 1 23.0 ;
  aero vrel 1 23.5 ;
  aero vrel 1 24.0 ;
  aero induc 4 1 2 39;
  aero induc 4 1 2 24;
  aero secforce 1 2 5;
  aero secforce 1 2 10;
  aero secforce 1 2 15;
  aero secforce 1 2 24;
  aero secforce 1 2 39;
  aero windspeed 4 1 2 39;
  wind_wake wake_pos 1 ;
  mbdy momentvec tower 1 1 tower  # Tower bottom;
  mbdy forcevec  tower 1 1 tower  # Tower botttom;
  mbdy momentvec tower 9 2 tower  # Tower top (yaw bearing);
  mbdy forcevec  tower 9 2 tower  # Tower top (yaw bearing);
  mbdy momentvec shaft 3 1 shaft  # Shaft (1st main bearing);
  mbdy forcevec  shaft 3 1 shaft  # Shaft (1st main bearing);
  mbdy momentvec blade1 1 1 blade1   # Blade1 (root);
  mbdy momentvec blade1 4 1 blade1   # Blade1 (SG pos 3.08);
  mbdy forcevec  blade1 1 1 blade1   # Blade1 (root);
  mbdy momentvec blade1 12 1 blade1  # Blade1 (rad 50%);
  mbdy momentvec blade3 1 1 blade3   # Blade3 (root);
  constraint bearing2 pitch1 5;
  constraint bearing2 pitch2 5;
  constraint bearing2 pitch3 5;
  constraint bearing1 shaft_rot 2;
  mbdy state pos tower 1 0.0 global     # Position tower bottom;
  mbdy state pos tower 9 1.0 global     # Position tower top;
  mbdy state pos blade1  14 1.0 blade1  # blade 1 tip pos ;
  mbdy state pos blade2  14 1.0 blade2  # blade 2 tip pos ;
  mbdy state pos blade3  14 1.0 blade3  # blade 3 tip pos ;
  mbdy state vel tower 9 1.0 global     # Velocoty tower top;
  mbdy state acc tower 9 1.0 global     # Acceleration tower top;
  DLL inpvec 1 1                        # Ref. power [w];
  DLL inpvec 2 1                        # Generator torque LSS [Nm];
end output;
;
exit ;
```

Risø's research is aimed at solving concrete problems in the society.

Research targets are set through continuous dialogue with business, the political system and researchers.

The effects of our research are sustainable energy supply and new technology for the health sector.

www.risoe.dk