



HAL
open science

Simulation and control of FOWT using OpenFAST/Matlab co-simulation

Seydali Ferahtia, Mourad Aït-Ahmed, Azeddine Houari, Mohamed
Machmoum

► **To cite this version:**

Seydali Ferahtia, Mourad Aït-Ahmed, Azeddine Houari, Mohamed Machmoum. Simulation and control of FOWT using OpenFAST/Matlab co-simulation. IREENA - Institut de Recherche sur l'Énergie Electrique de Nantes-Atlantique (IREENA) EA 4642; Nantes Université. 2024. hal-04858231

HAL Id: hal-04858231

<https://hal.science/hal-04858231v1>

Submitted on 29 Dec 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Simulation and control of FOWT using OpenFAST/Matlab co-simulation

User Guide

Ferahtia Seydali

Mourad Ait-Ahmed

Azeddine Houari

Machmoum Mohammed

2024



Table of Contents

Table of Contents	ii
Introduction.....	1
1- Overview	1
2- OpenFAST.....	3
2-1- Structuring multi-physical models of floating wind turbines under the Open-Fast software interface.....	3
2-2- Configuring OpenFast/MATLAB-Simulink co-simulation	6
2-3- First simulation.....	6
3- Outputs	9
FOWT Modeling – Basics	11
1-WT modeling.....	11
1-1- Aerodynamic model:	11
1-2- Mechanical model (drivetrain):	13
1-3- Model linearization:	14
2- Control designing	15
2-1- Control objectives	15
2-3- Region II controller (MPPT):	16
2-4- Region II controller	17
References	19

Introduction

1- Overview

Recently, wind energy costs have decreased significantly due to larger rotor sizes, especially for floating offshore wind turbines (FOWTs). FOWTs are installed in broad ocean regions, where floating platforms moor them to the seabed, enabling them to function at more robust and steady wind speeds with larger blades. These WT minimize visual impact, reduce carbon emissions, and fight climate change while powering coastal towns. However, these giant turbines introduce challenges, particularly regarding platform stability and power quality. Increased blade length leads to an increase in their deflections and significant deviations in the swept area from the rotor plane. Additionally, flexible blades may increase aerodynamic unsteadiness and complexity, necessitating advanced dynamic models to accommodate these problems. Moreover, the tower's flexibility and floating platform motions can induce extra motions that can affect the performance. Therefore, establishing a suitable model to reproduce the FOWT behavior and to design robust controllers is mandatory. As illustrated in Figure 1, the FOWT model includes complex and highly nonlinear sub-models, including:

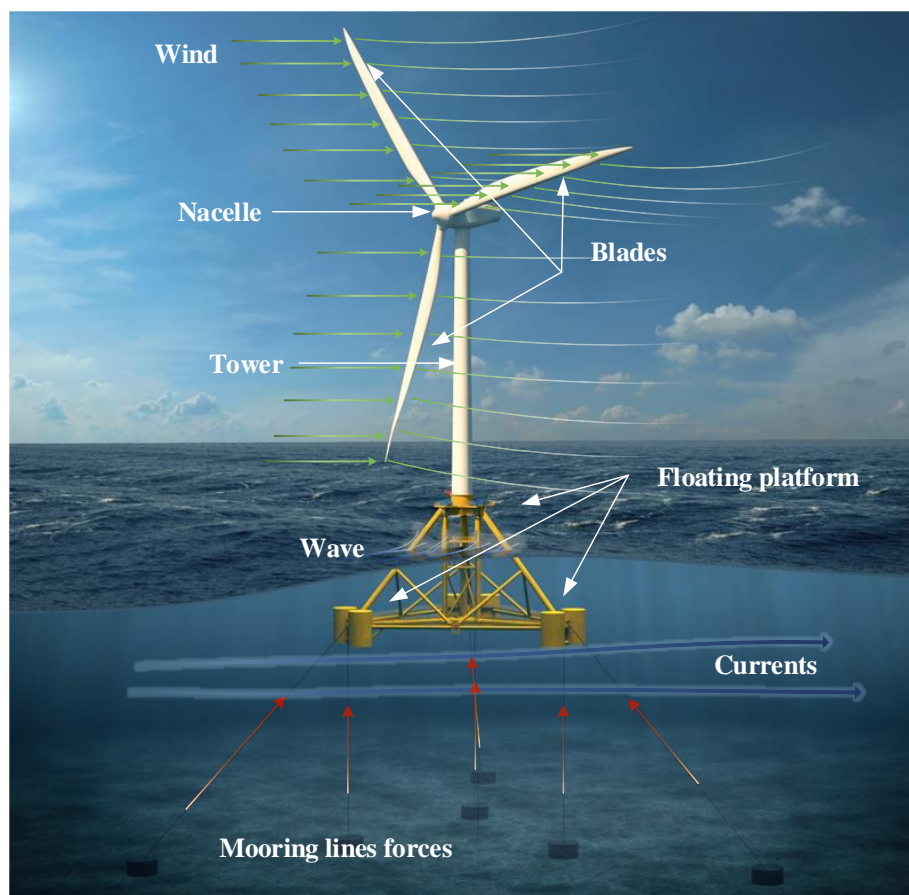


Figure 1 – FOWT representation

-
- **Aerodynamic effects:** Designing and analyzing the air flows around the blades and their shapes is crucial to maximizing energy capture while minimizing drag and turbulence. These models help predict performance under varying wind conditions.
 - **Hydrodynamic effects:** Precise simulation of the interaction between the floating platform and the hydrodynamics (waves and currents) is essential since the floating platform must adapt to diverse sea environments.
 - **Structural Dynamics:** The turbine and platform must be developed to resist various environmental loads and to guarantee durability and stability.
 - **Mooring systems:** This model is required to understand how the seabed interacts with the floating structure and how this affects stability and performance.
 - **Control Systems:** Developing effective control strategies to maintain stability and optimize turbine performance and lifespan in the face of changing environmental conditions is challenging. This encompasses coordinating the turbine and platform blade orientation.

The National Renewable Energy Laboratory (NREL) has developed the OpenFAST software to simulate the dynamic behavior of FOWT models accurately while considering the above-mentioned parts. OpenFAST is a multi-physics tool that simulates wind turbines' dynamic response. It combines computer modules for structural dynamics, control and electrical system dynamics, and hydrodynamics. This lets you run coupled nonlinear aero-hydro-servo-elastic simulations in the time domain. It can analyze wind turbine configurations, including two—or three-blade horizontal-axis rotors, pitch regulation, hubs, and towers.

2- OpenFAST

OpenFAST is an open-source wind turbine simulation tool established to reproduce wind turbines based on aero-hydro-servo-elastic engineering models. OpenFAST can provide a solid, high-fidelity software engineering platform [1]. This tool was created based on the FAST (fatigue, aerodynamic, structure, and turbulence) simulation tool published by the NREL.

2-1- Structuring multi-physical models of floating wind turbines under the Open-Fast software interface

The FOWT model OpenFAST provides can be divided into three levels: external conditions, applied loads, and structural dynamics, as illustrated in Figure 2. The main files of the OpenFAST are:

- **OpenFAST main file (.fst):** this file loads the simulation parameters and the required sub-codes. It includes:
 - Simulation control: provide the simulation parameters, such as the sampling time (DT) and the simulation time (TMax).
 - Feature switches and flags: select the source code for each part.
 - Environmental conditions: initialize the environmental conditions.
 - Input files: this part loads the sub-code for each part.
 - Output: enable or disable the creation of the simulation results (.sum, .out, and .outb).
 - Linearization: enable or disable the linearization function (only with constant wind speed without waves).
 - Visualization: enable or disable the creation of VTK data that is used to make an animation of the FOWT (external software is required, '*Paraview*').

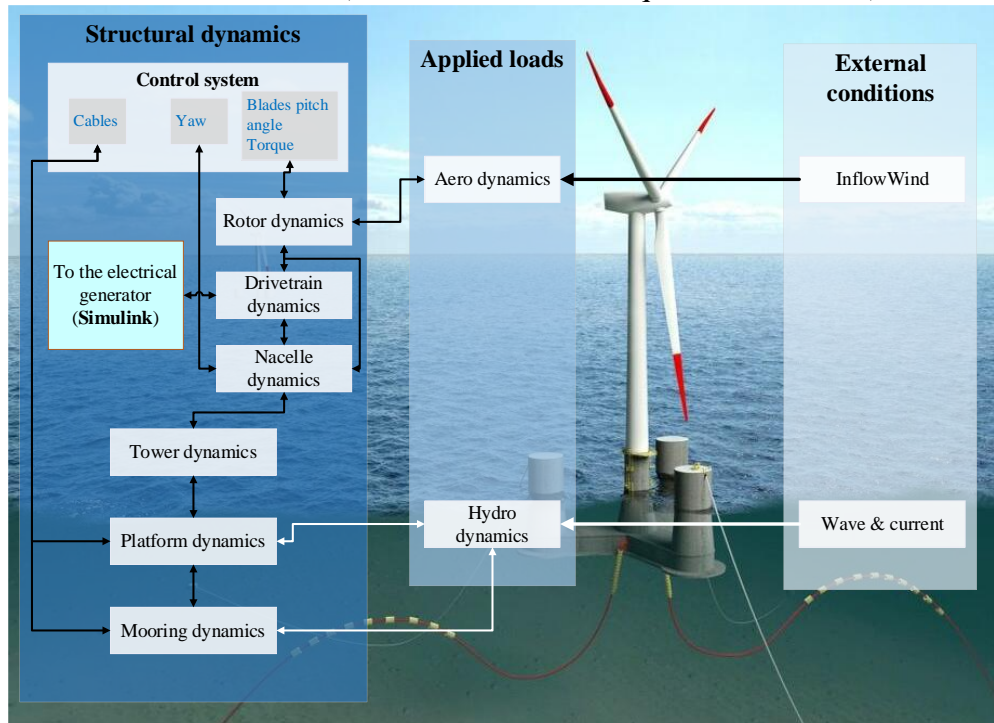


Figure 2 – Schematic of the main parts of OpenFAST

-
- **Structural parameters file (xx_ElastoDyn.dat):** the main parameters of the WT are included in this file, such as:
 - Degrees of freedom (DOF): enable or disable any DOF.
 - Initial conditions: speed, positions, deflections ...
 - Turbine configuration: blades, tower, drivetrain, and mass and inertia of the different parts of the structure....
 - Mass and inertia: shafts, platform, nacelle masses, and inertia, blades' masses.
 - Blade: blades geometry files (to be loaded from external files).
 - Rotor-teeter: for 2-blades WTs.
 - Drivetrain: gearbox parameters.
 - Furling: if there are additional properties for the furling turbine.
 - Tower: tower geometry files (to be loaded from external files).
 - Outputs: the generated outputs by this file are mentioned in this section.
 - **External wind profile file (xx_InflowWind.dat):** this file includes a switching flag for the wind type 'WindType', such as:
 - Steady (1): the wind speed is constant (= HWindSpeed).
 - Uniform (2): the wind speed is variable based on the designed profile loaded from an external file (.wnd or .hh). These files can be modified by the used using the *notebook* application.
 - Binary TurbSim (3): the wind speed is variable based on the profile loaded from another file (.bts). These files are generated using *TurbSim* software based on several meteorological parameters.
 - **Aerodynamics file (xx_AeroDyn15.dat):** this file includes the aerodynamic power calculation models that include the Blade-Element/Momentum theory (BEM), Dynamic Blade-Element/Momentum theory (DBEM), and a Free Vortex Wake 'FVW' called cOnvecting LAgrangian Filaments theory (OLAF). This also requires the airfoil information for each blade. The blades' geometry (distributed blade properties) are provided in xx_Blade.dat, while the airfoil information (angle of attack, pitching-moment coefficient, lift, and drag coefficients) are provided in xx_CylinderX.dat and xx_DUXX_A17dat files.
 - **Waves profile and hydrodynamics file (xx_HydroDyn.dat):** this file includes a switching flag for defining the kinematic model of the incident wave and current modes as well as parameters of the applied loads on the floating platform. For FOWTs, the effect of the currents is minor, so the main significant parameter in this file is the wave mode 'WaveMod'; the main modes are:
 - None (0): non-waves, still water.
 - Regular (1): periodic waves.
 - JONSWAP/Pierson-Moskowitz spectrum (2): irregular profile based on JONSWAP equation.
 - **Control file (xx_ServoDyn.dat):** This module enables the control of various actuators, such as the blade's pitch angle and the generator's torque. The following flags should be adjusted:
 - PCMode: pitch control model = 5 for co-simulation with MATLAB/Simulink.
 - VSContrl: variable-speed control mode = 5 for the co-simulation MATLAB/Simulink.

Mooring lines file (xx_FEAMooring.dat/xx_Mooring.dat/xx_MAP.dat): This module provides the various properties of the mooring lines attached to the rig, such as line type, material, diameter, and length. It is chosen based on the platform type. Each file includes some parameters and properties for the interaction soil structure.

The hierarchy of the codes is illustrated in Figure 3.

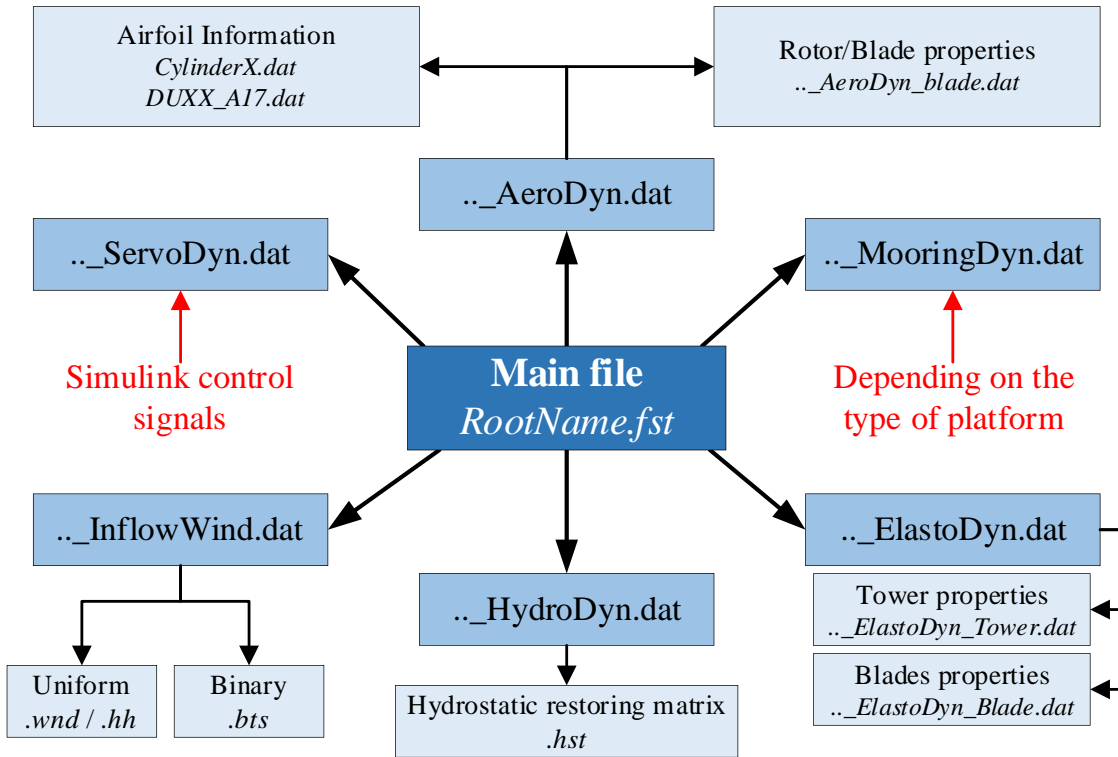


Figure 3 – Codes hierarchy in OpenFAST

On the other hand, the OpenFAST has the following outputs:

Table 1 - List of output files

Generated file	Extension	Format	Description
Summary	.sum	Txte	Report files generated for each part. These reports include a summary of the simulation.
Time series	.out	Txte	Output matrix of all the numerical outputs mentioned at the end of each part.
	.outb	Binary	Time-marching tabular output in a binary format.
Linearization	.lin	Txte	Linearized output matrix.
Visualization	.vtp	VTK XML (text)	Polygonal data file for visualization and animation.

Note: If a certain module remains inactive during a simulation, executions of that module's subroutines are excluded, and the module's inputs/outputs are neither established nor used.

2-2- Configuring OpenFast/MATLAB-Simulink co-simulation

Step 01- Download OpenFAST: OpenFAST is open-access software that can be downloaded freely from GitHub: <https://github.com/OpenFAST/openfast>.

Step 02- Download the build folder: you can find it at the following link: <https://github.com/sido28/OpenFAST-build-folder-.git>.

Step 03- Add build folder to MATLAB paths: After downloading this folder:

- 1- Copy and paste it into the main Folder of OpenFAST,
- 2- Open Matlab and select the Set Path option,
- 3- Select Add Folder.
- 4- Open the repository of the bin folder included in the build folder, as illustrated in Figure 4.
- 5- Finally, choose this folder and save the changes.

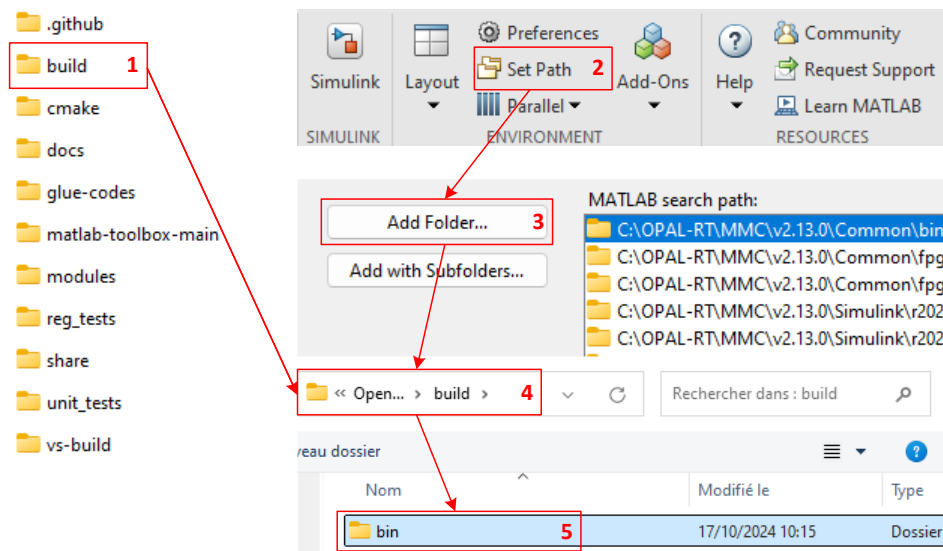


Figure 4 – Coupling OpenFAST with Matlab

2-3- First simulation

After coupling OpenFAST with Matlab, a first simulation should be performed to check the communication between the two software. Two example models (OpenLoop and Test01_SIG), included in the OpenFAST folder (go to ...OpenFAST\glue-codes\simulink\examples), can be used for this purpose. Each example has a Simulink model (.mdl) and a script file (.m).

- **OpenLoop :**

Open OpenLoop.mdl:

- As illustrated in Figure 5, the main block Simulink file 'FAST Nonlinear Wind Turbine' has seven inputs and one output, 'OutData'. Here is the S-Function responsible for communicating between Simulink and OpenFAST. It requires the name (FAST_InputFileName) and the simulation time (TMax).

Open Run_OpenLoop.m:

- As illustrated in Figure 6, the parameters FAST_InputFileName and the TMax should be defined in the associated script file (.m).

- FAST_InputFileName should have the same name as the main file (.fst) (if this file is not included in the current Matlab operating repository, the full path should be written before the name, as illustrated in this example).
- Running this simulation helps check and understand the co-simulation function.

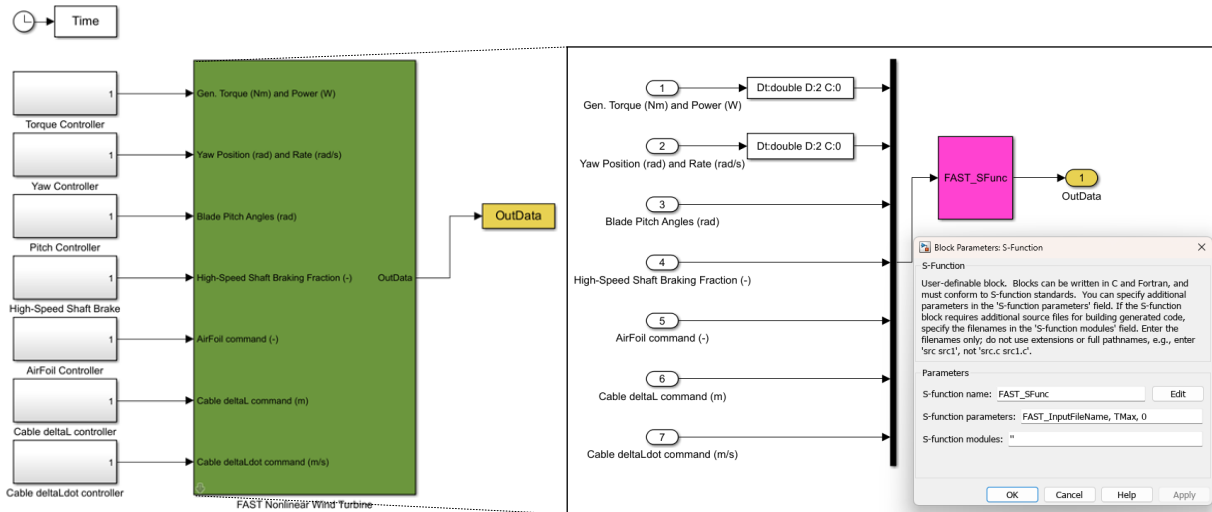


Figure 5 – OpenLoop Simulink blocks

```

Run_OpenLoop.m
1 % make sure the OpenFAST directory where the FAST_SFunc.mex* file is located
2 % is in the MATLAB path (also make sure any other OpenFAST library files that
3 % are needed are on the MATLAB path)
4 % (relative path names are not recommended in addpath()):
5 % addpath(' ../../build/bin'); % install location for Windows Visual Studio builds
6 % addpath(genpath(' ../../install')); % cmake default install location
7
8 % these variables are defined in the OpenLoop model's FAST_SFunc block:
9 FAST_InputFileName = ' ../../reg_tests/r-test/glue-codes/openfast/AOC_WSt/AOC_WSt.fst';
10 TMax = 60; % seconds
11
12 sim('OpenLoop.mdl',[0,TMax]);

```

Figure 6 – OpenLoop Matlab script

• **Test01_SIG :**

In this case, a simplified model of a simple induction generator (SIG) is associated with the model to assess the performance in the presence of a load coupled with the high-speed shaft (generator). The model and the conversion chain are.

Open Test01_SIG.mdl:

- Figure 7 illustrates the coupling of the WT with a SIG. The first input is a double signal that includes the generator's torque and power.
- The torque can be controlled to achieve maximum power at low wind speeds.
- Additionally, the machine model uses the feedback signal (low-speed shaft rotational speed, or 'rpm') as an input. The value is extracted from the OutData matrix using the Fnc block with the following expression: `u(strmatch('LSSGagVxa',OutList))`, where LSSGagVxa represents the generator speed.

Open Run_Test01_SIG.m:

- This file contains the IG machine parameters (.m), in addition to the OpenFAST parameters, FAST_InputFileName, and the TMax.
- Running this simulation helps check and understand the closed-loop simulation.

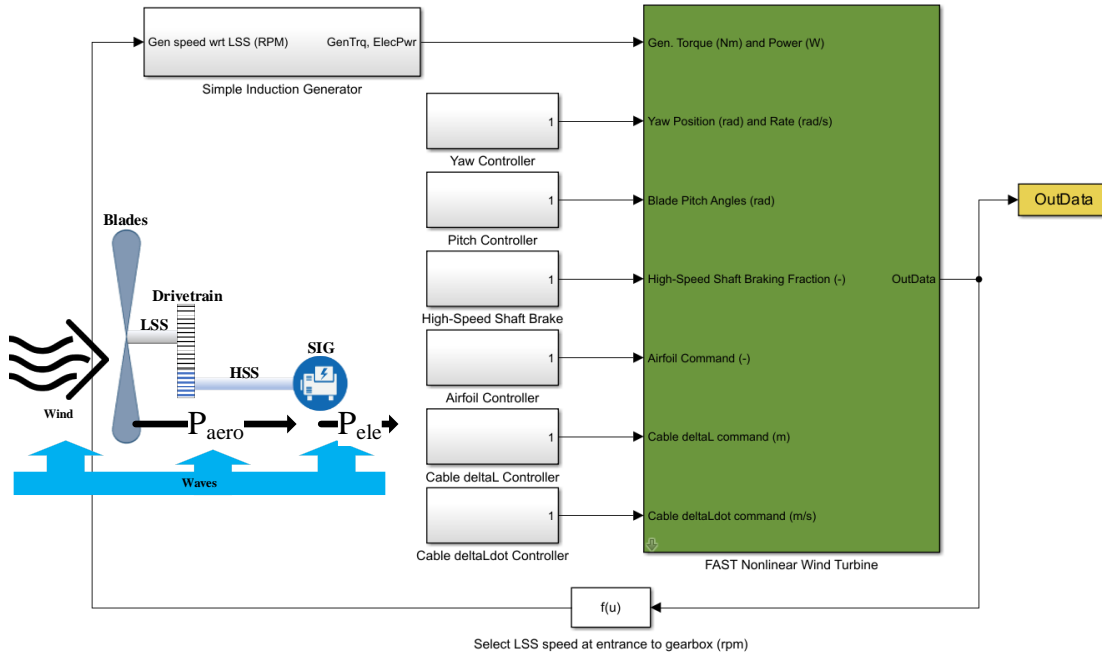


Figure 7 – Test01_SIG Simulink script

AOC_YFix_WSt.AD	12/04/2023 09:12	Fichier SUM	5 Ko
AOC_YFix_WSt.ED	12/04/2023 09:12	Fichier SUM	8 Ko
AOC_YFix_WSt Main file (.fst)	12/04/2023 09:12	Fichier FST	8 Ko
AOC_YFix_WSt	12/04/2023 09:12	Document texte	3 Ko
AOC_YFix_WSt Outputs file (.out)	12/04/2023 09:12	Fichier OUT	131 Ko
AOC_YFix_WSt	12/04/2023 09:12	Fichier OUTB	95 Ko
AOC_YFix_WSt.SrvD	12/04/2023 09:12	Fichier SUM	4 Ko
AOC_YFix_WSt Summary file (.sum)	12/04/2023 09:12	Fichier SUM	6 Ko
AOC_YFix_WSt.UA	12/04/2023 09:12	Fichier SUM	6 Ko
AOC_YFix_WSt.AD15.ipt	12/04/2023 09:12	Fichier IPT	10 Ko
AOC_YFix_WSt.ElastoDyn	12/04/2023 09:12	Fichier DAT	13 Ko
AOC_YFix_WSt.InflowWind	12/04/2023 09:12	Fichier DAT	6 Ko

Figure 8 – Main and summary files

Note: You can check the outputs list generated. Open the folder that includes the main file (.fst); after the simulation, .sum, .out, and .outb files are generated. There is a .sum file with the same name as the main file. For example, in the Test01_SIG case, the main file has the name AWT_YFix_WSt.fst, so the summary file that includes the list of all the outputs has the name AWT_YFix_WSt.sum, while the output results are provided in file with a similar name but its extension is .out. See Figure 8.

3- Outputs

As previously mentioned, each module of the OpenFAST sub-codes provides a list of outputs, depending on the WT type and characteristics. However, it is possible to delete or add more outputs that are not mentioned in the module output list.

1. Download the full list of the OpenFAST outputs ([link](#)).
2. The downloaded file is an Excel file with several sheets. Specify the module that should generate your output by choosing the corresponding sheet.
3. Assuming that I want to add the nacelle fore-aft acceleration to the list of the outputs:
 - a. Select ElastDyn sheet,
 - b. Go to Nacelle IMU Motions section,
 - c. As illustrated in Figure 9, the corresponding value is the translational acceleration of the nacelle inertial measurement unit along the x-axis (NcIMUTAxS). Copy the name of this value.
 - d. Go to the ElastoDyn file associated with the model that you are working on and paste it (NcIMUTAxS) to the output list at the end of this code, as illustrated in Figure 10.

1	A	B	C	D	E	F	
Category	Name	Other Name(s)	Description	Convention	Units	Inv	
	HSShftA	GenAccel	Angular acceleration of the high-speed shaft and generator	Same sign as LSSGagAxa / LSSGagAxs / LSSGagA	(deg/s^2)		
303	Nacelle IMU Motions						
304	NcIMUTVxs		Nacelle inertial measurement unit translational velocity (absolute)	Directed along the xs-axis	(m/s)		
305	NcIMUTVys		Nacelle inertial measurement unit translational velocity (absolute)	Directed along the ys-axis	(m/s)		
306	NcIMUTVzs		Nacelle inertial measurement unit translational velocity (absolute)	Directed along the zs-axis	(m/s)		
307	NcIMUTAxS		Nacelle inertial measurement unit translational acceleration (absolute)	Directed along the xs-axis	(m/s^2)		
308	NcIMUTAyS		Nacelle inertial measurement unit translational acceleration (absolute)	Directed along the ys-axis	(m/s^2)		
309	NcIMUTAzS		Nacelle inertial measurement unit translational acceleration (absolute)	Directed along the zs-axis	(m/s^2)		
310	NcIMURVxs		Nacelle inertial measurement unit angular (rotational) velocity (absolute)	About the xs-axis	(deg/s)		
311	NcIMURVys		Nacelle inertial measurement unit angular (rotational) velocity (absolute)	About the ys-axis	(deg/s)		
312	NcIMURVzs		Nacelle inertial measurement unit angular (rotational) velocity (absolute)	About the zs-axis	(deg/s)		
313	NcIMURAxS		Nacelle inertial measurement unit angular (rotational) acceleration (absolute)	About the xs-axis	(deg/s^2)		
314	NcIMURAyS		Nacelle inertial measurement unit angular (rotational) acceleration (absolute)	About the ys-axis	(deg/s^2)		
315	NcIMURAzS		Nacelle inertial measurement unit angular (rotational) acceleration (absolute)	About the zs-axis	(deg/s^2)		
316	Rotor-Furl Motions						
317	RotFurlP	RotFurl	Rotor-furl angle (position)	About the rotor-furl axis	(deg)		
318	RotFurlV		Rotor-furl angular velocity	About the rotor-furl axis	(deg/s)		
319	RotFurlA		Rotor-furl angular acceleration	About the rotor-furl axis	(deg/s^2)		
320	Tail-Furl Motions						
321	TailFurlP	TailFurl	Tail-furl angle (position)	About the tail-furl axis	(deg)		
322	TailFurlV		Tail-furl angular velocity	About the tail-furl axis	(deg/s)		
323	TailFurlA		Tail-furl angular acceleration	About the tail-furl axis	(deg/s^2)		
324	Nacelle Yaw Motions						
325	YawPzn	YawPzp, NacYawP, NacYawz, YawPos	Nacelle yaw angle (position)	About the zn- and zp-axes	(deg)		
326							

Figure 9 – Selecting a new output from the output list.

```

"Baseline_Tower.dat" TwrFile "Name of file containing tower properties (quoted string)
-----
OUTPUT
-----
True SumPrint - Print summary data to "<RootName>.sum" (flag)
1 OutFile - Switch to determine where output will be placed: {1: in module output file only; 2: in glue code output file only; 3: both} (currently unused)
True TabDelim - Use tab delimiters in text tabular output files? (flag) (currently unused)
"ES10_3E2" OutFmt - Format used for text tabular output (except time). Resulting field should be 10 characters. (quoted string) (currently unused)
0 TStart - Time to begin tabular output (s) (currently unused)
10 DecFact - Decimation factor for tabular output {1: output every time step} (-) (currently unused)
4, 7 NTwGages - Number of tower nodes that have strain gages for output [0 to 9] (-)
0 TwrGagM - List of tower nodes that have strain gages [1 to TwrNodes] (-) [unused if NTwGages=0]
0 NB1Gages - Number of blade nodes that have strain gages for output [0 to 9] (-)
0 BldGagM - List of blade nodes that have strain gages [1 to BldNodes] (-) [unused if NB1Gages=0]
OutList - The next line(s) contains a list of output parameters. See OutListParameters.xlsx for a listing of available output channels, (-)
"RotSpeed" - Rotor speed
"BldPitch1" - Blade 1 pitch angle
"OoPDefl1" - OoP blade 1,3 tip deflections
"OoPDefl3" - OoP blade 1,3 tip deflections
"TipALxb1" - Blade 1 flapwise tip acceleration
"RootMxb1" - Blade 1 root moments
"RootMyb1" - Blade 1 root moments
"RootMzb1" - Blade 1 root moments
"RotThrust" - Rotor thrust and rotating LSS shear forces
"LSShftFya" - Rotor thrust and rotating LSS shear forces
"LSShftFza" - Rotor thrust and rotating LSS shear forces
"LSSTipMya" - Rotating LSS bending moments at the shaft tip
"LSSTipMza" - Rotating LSS bending moments at the shaft tip
"RotTorg" - Rotor (mechanical) torque & power and generator (electrical) power
"RotPwr" - Rotor (mechanical) torque & power and generator (electrical) power
"TTDspFA" - FA and SS tower-top deflections
"TTDspSS" - FA and SS tower-top deflections
"TwHt1MLxt" - Tower, Gage-1 (Node 4) roll, pitch, and yaw moments
"TwHt1MLyt" - Tower, Gage-1 (Node 4) roll, pitch, and yaw moments
"TwHt1MLzt" - Tower, Gage-1 (Node 4) roll, pitch, and yaw moments
"TwHt" - local tower forces
"TwHt" - local tower forces
"TwHt" - local tower forces
"McIMUTAx"
END of input file (the word "END" must appear in the first 3 columns of this last OutList line)

```

Figure 10 – Adding a new output to the output list.

Note: Some outputs have more than one name.

Note: The updated version of OpenFAST has removed some of the output names. If you are unable to find the output name, it is recommended that you check the version.

FOWT Modeling – Basics

1-WT modeling

1-1- Aerodynamic model:

The turbine blades capture the aerodynamic power (kinetic of wind) and convert it to mechanical power by rotating the low-speed shaft (LSS). The captured aerodynamic power (P_{aero}) can be expressed as follows [2]:

$$P_{aero} = \frac{C_p(\beta, \lambda)}{2} \rho \pi R^2 V^3 = \frac{C_p(\beta, \lambda)}{2} \rho S V^3 \quad (1)$$

where V is the wind speed, C_p is the power coefficient, β inclination angle, λ tip speed ratio, R is the radius, S is the surface swept by the blades, and ρ ($=1.225$) is the air density. The aerodynamic torque (T_{aero}) can be expressed as

$$T_{aero} = \frac{P_{aero}}{\omega_{LSS}} = \frac{C_p(\beta, \lambda)}{2\lambda} \rho \pi R^3 V^2 \quad (2)$$

$$\lambda = \frac{R\omega_{LSS}}{V} \quad (3)$$

where ω_{LSS} is the low-speed shaft (LSS) rotation speed, the C_p function is nonlinear and has several forms [2]. The most common one is

$$C_p(\beta, \lambda) = c_1 \left(\frac{c_2}{\xi} - c_3\beta - c_4 \right) e^{\frac{-c_5}{\xi}} + c_6\lambda \quad (4)$$

$$\frac{1}{\xi} = \frac{1}{\lambda + 0.08\beta} - \frac{0.035}{\beta^3 + 1}$$

where $c_1= 0.5176$, $c_2= 116$, $c_3= 0.4$, $c_4= 5$, $c_5= 21$, $c_6= 0.0068$. Figure 9 presents the C_p curves as a function of the tip speed ratio (λ).

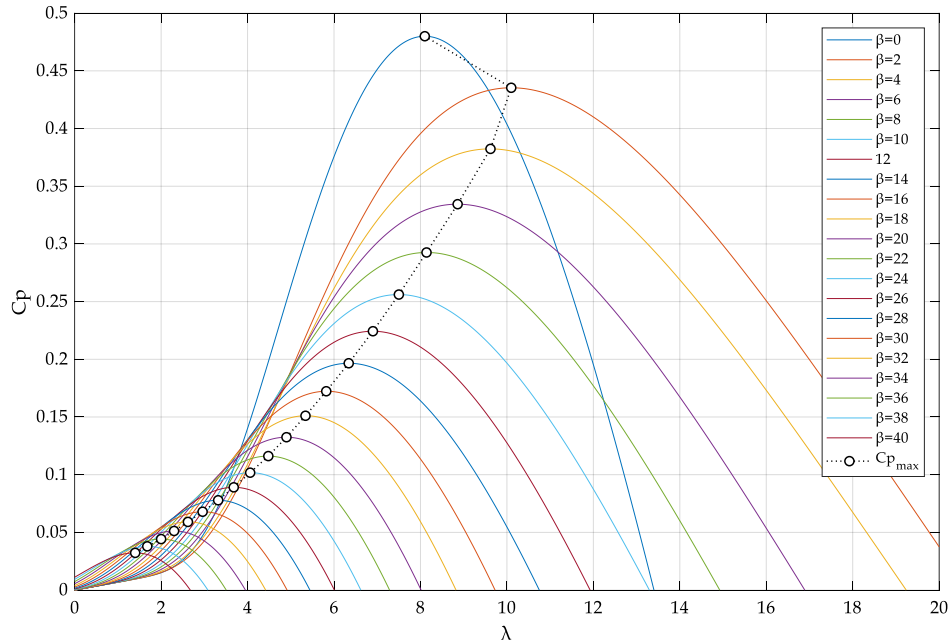


Figure 9 – C_p curves as a function of the tip speed ratio (λ)

Defining the parameters in FAST:

Parameters	In FAST	Location	
R	TipRad	ElastoDyn	- Turbine configuration
ρ	AirDens	Main file (.fst)	- Environmental conditions

However, the real harvested power is lower than the theoretical due to several structural and environmental factors. From Figure 10, the FOWT is installed on a floating platform that may have extra motions. In addition, it has a significant tower length, which can increase the twist angle of the tower deflection. These factors can induce significant nacelle motions that can change the hub's rotation speed. Therefore, nacelle fore-aft motions should be included in Equation (1) in addition to the wind speed. The OpenFAST provides these motions in the output list, section nacelle inertial measurement unit (NcIMU) motions in the ElastoDyn file, such as:

- NcIMURVxs: rotational velocity according to the x-axis (for-aft) in deg/s.
- NcIMUTVxs: fore-aft translational velocity in m/s.
- NcIMUTAxS: fore-aft translational acceleration in m^2/s .

Moreover, one essential factor is the blades' flexibility. Figure 11 illustrates how large blade deflections can significantly reduce the swept area (S) compared to rigid blades' swept area (S_0). OpenFAST includes higher-fidelity aerodynamic models, such as BEM, dynamic BEM (DBEM), and OLAF, to accommodate this flexibility. To change this model in OpenFAST, change the flag 'WakeMod' in AeroDynV15 (0=none, 1=BEM, 2=DBEM, 3=OLAF).

Note: WakeMod cannot be 2 or 3 when linearizing.

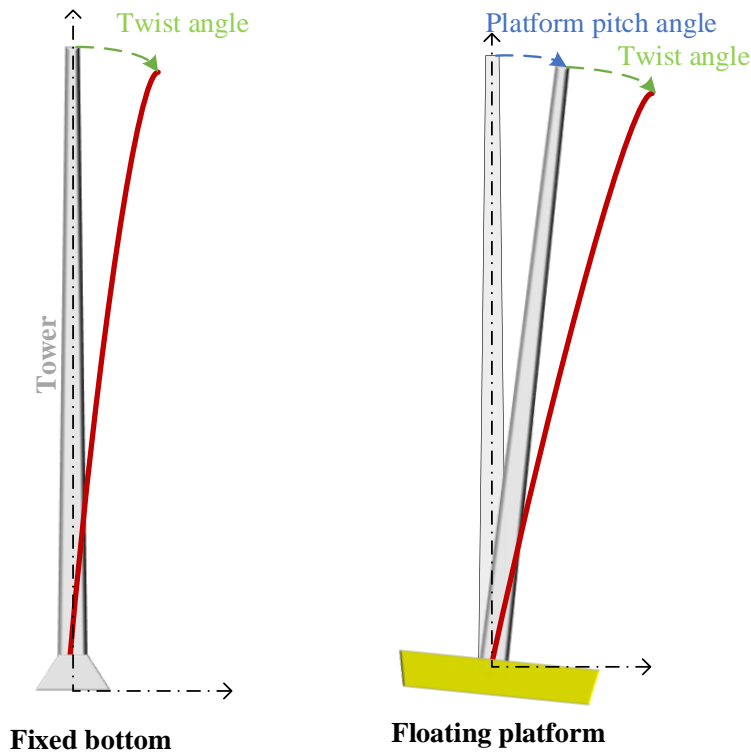


Figure 10 – nacelle fore-aft displacement as a function of the tower deflections: fixed bottom vs floating platform

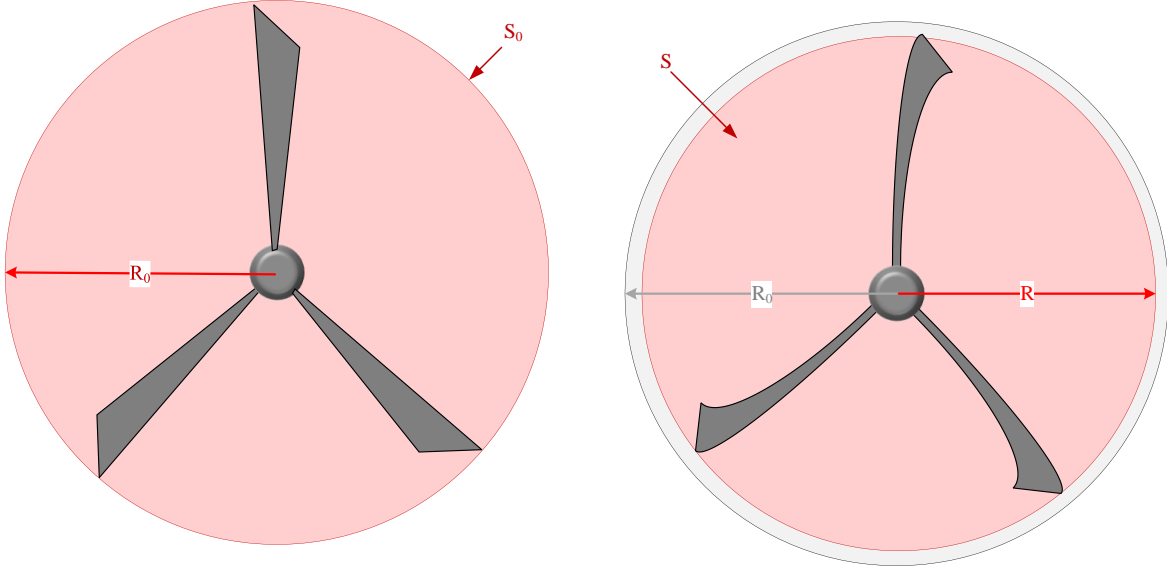


Figure 11 – Swept area (S_0) using rigid blades and swept area (S) using flexible blades

1-2- Mechanical model (drivetrain):

The two-mass drivetrain mechanical model of the wind turbine is modeled based on the gearbox mass and the wind wheel mass [3], as explained in Figure 12.

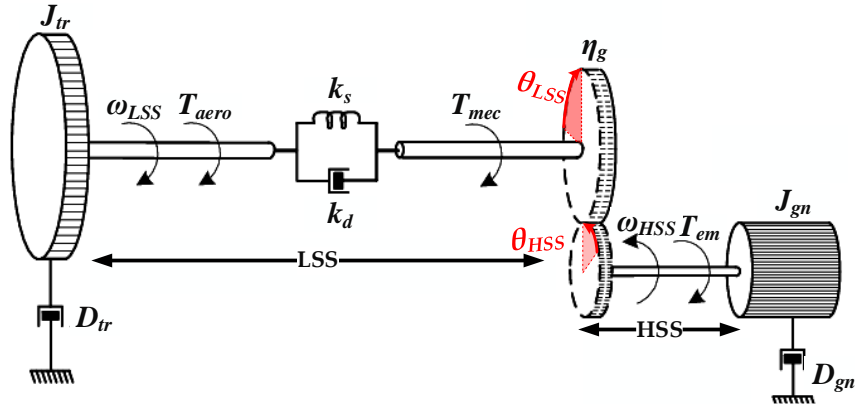


Figure 12 – two masses gearbox system

The mathematical equations of a two-mass drive train model are given based on the second Newton's law. The low-speed shaft dynamics can be expressed as follows:

$$J_{tr} \frac{d\omega_{LSS}}{dt} = T_{aero} - T_{mec} - \omega_{LSS} D_{tr} \quad (5)$$

where T_{aero} is aerodynamic torque, J_{tr} is the turbine shaft inertia, D_{tr} is the turbine damping coefficient, and T_{mec} is mechanical torque, which can be calculated as follows:

$$T_{mec} = k_s (\underbrace{\theta_{LSS} - \theta_{HSS}}_{\theta_r}) + k_d (\underbrace{\omega_{LSS} - \omega_{HSS}}_{\dot{\theta}_r}) = k_s \theta_r + k_d \dot{\theta}_r \quad (6)$$

where k_s is drivetrain stiffness, k_d is the damping factor, and θ_r is the angle between the turbine rotor and the generator rotor (torsion angle of the mass).

On the other hand, the HSS dynamics can be expressed as follows:

$$J_{gn} \frac{d\omega_{HSS}}{dt} = \frac{T_{mec}}{\eta_g} - T_{em} - \omega_{HSS} D_{gn} \quad (7)$$

where T_{em} is electromagnetic torque, ω_{HSS} is the speed of the HSS, J_{gr} is the generator side shaft inertia, D_{gn} is the generator self-damping coefficient, and η_g is the gear ratio. Using Laplace transformation

$$\omega_{LSS} = (T_{aero} - T_{mec}) \frac{1}{sJ_{tr} + D_{tr}} \quad (8)$$

Neglecting all the damping and stiffness factors, Equation (7) can be rewritten as

$$\begin{aligned} J_{gn} \frac{d\omega_{HSS}}{dt} &= \frac{1}{\eta_g} \left(T_{aero} - J_{tr} \frac{d\omega_{LSS}}{dt} \right) - T_{em} \\ &= \frac{1}{\eta_g} \left(T_{aero} - J_{tr} \frac{1}{\eta_g} \frac{d\omega_{HSS}}{dt} \right) - T_{em} \\ &= \frac{T_{aero}}{\eta_g} - \frac{J_{tr}}{\eta_g^2} \frac{d\omega_{HSS}}{dt} - T_{em} \end{aligned} \quad (9)$$

After simplifications

$$\left(J_{gr} + \frac{J_{tr}}{\eta_g^2} \right) \frac{d\omega_{HSS}}{dt} = \frac{T_{aero}}{\eta_g} - T_{em} \quad (10)$$

For the low-speed shaft, Eq(10) can be provided as follows :

$$\left(J_{gr} + \frac{J_{tr}}{\eta_g^2} \right) \eta_g \frac{d\omega_{LSS}}{dt} = \frac{T_{aero}}{\eta_g} - T_{em} \quad (11)$$

$$\underbrace{\left(\eta_g^2 J_{gr} + J_{tr} \right)}_{J_{mec}} \frac{d\omega_{LSS}}{dt} = T_{aero} - \eta_g T_{em} \quad (12)$$

Defining the parameters in FAST:

Parameters	In FAST	Location
J_{tr}	HubIner	Elastodyn- Mass and inertia
J_{gr}	GenIner	Elastodyn- Mass and inertia
η_g	GBRatio	Elastodyn- Drivetrain
k_s	DTTorSpr	Elastodyn- Drivetrain
k_d	DTTorDmp	Elastodyn- Drivetrain

1-3- Model linearization:

The linearized aerodynamic torque at the linearization points (lp) is provided as follows:

$$\Delta T_{aero} = \left. \frac{\partial T_{aero}}{\partial \omega_{HSS}} \right|_{\omega_{g,lp}} \Delta \omega_{HSS} + \left. \frac{\partial T_{aero}}{\partial \beta} \right|_{\beta_{lp}} \Delta \beta + \left. \frac{\partial T_{aero}}{\partial V} \right|_{V_{lp}} \Delta V \quad (13)$$

The linear model of the generator speed based on Equation (13) is:

$$\Delta \dot{\omega}_{HSS} = \frac{\eta_g}{J_{mec}} (\Delta T_{aero} - \eta_g \Delta T_{em}) \quad (14)$$

$$\Delta \dot{\omega}_{HSS} = A_{\omega_{HSS}} \Delta \omega_{HSS} + B_{\beta} \Delta \beta + B_V \Delta V + B_{T_{em}} \Delta T_{em} \quad (15)$$

$$\Delta \omega_{HSS} (s + A_{\omega_{HSS}}) = B_{\beta} \Delta \beta + B_V \Delta V + B_{T_{em}} \Delta T_{em} \quad (16)$$

2- Control designing

2-1- Control objectives

As illustrated in Figure 12, the main objectives of controlling the wind turbine are maximizing the power production when the wind speed is above the cut-in value (in R2) and maintaining it at the rated value when the wind speed is above the rated value (in R3). This can be achieved by controlling the C_p function. The control system maximized it in R2 and reduced it to reduce the aerodynamic torque in R3 value by adjusting the pitch angle and the generator torque. If the wind speed didn't exceed the cut-on value but was below the rated value (in R1), the wind blades turned at a very low speed, producing no power. Hence, the wind turbine is turned off. If the wind speed exceeds the cut-off value, the control should stop the rotation speed at this (in R4).

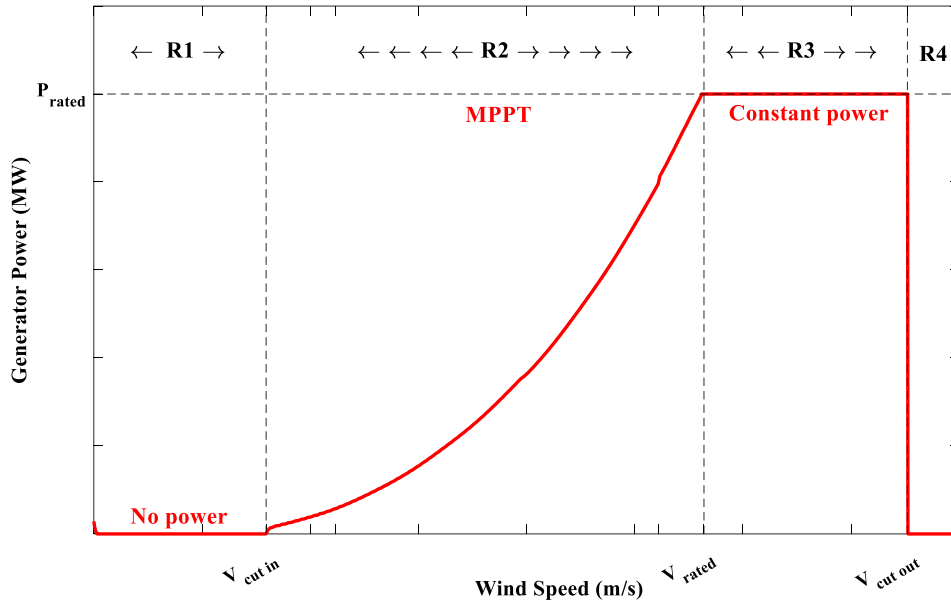


Figure 12 – Wind power as a function of the wind speed

Assumptions:

- In region 3, the generator torque is constant, so $\Delta T_{em} = 0$.
- In region 2, the blade pitch angles are assumed to be constant, so $\Delta \beta = 0$.
- When linearizing at constant wind speed, the disturbance ΔV is zero, and B_V is set to 0. Therefore, the parameters of Equation (16) are calculated at constant wind speed (V_{lp}) as follows:

$$A_{\omega_{HSS}} (V_{lp}) = \frac{\eta_g}{J_{mec}} \frac{\partial T_{aero}}{\partial \lambda} \frac{\partial \lambda}{\partial \omega_{HSS}} \quad (17)$$

$$\frac{\partial T_{aero}}{\partial \lambda} = \frac{1}{2\lambda_{lp}^2} \rho SRV_{lp}^2 \left(\left. \frac{\partial Cp}{\partial \lambda} \right|_{\lambda_{lp}} \Delta \lambda - Cp(\lambda_{lp}) \right) \quad (18)$$

$$\frac{\partial \lambda}{\partial \omega_{HSS}} = \frac{R}{V_{lp}} \quad (19)$$

$$B_{\beta}(V_{lp}) = \frac{\eta_g}{J_{mec}} \frac{\partial T_{aero}}{\partial \beta} \quad (20)$$

$$B_{Tem}(V_{lp}) = -\frac{\eta_g}{J_{mec}} \eta_g \quad (21)$$

Based on [4], the power coefficient Cp can be calculated at any operating point as follows:

$$Cp(\lambda_{lp}) = Cp_{rated} \left(\frac{\lambda_{lp}}{\lambda_{rated}} \right)^3 \quad (22)$$

2-3- Region II controller (MPPT):

The control objective in this region is to maximize the generated power by controlling the generator's torque. There are many strategies in the literature, such as optimal torque control (OTC), tip speed ratio control (TSRC), fuzzy logic, and reinforcement learning.

2-3-1- OTC:

Based on the Cp curves Eq(4) and **Fig.4**, the WT can achieve its MPP using an MPPT strategy, such as the optimal torque control strategy (OTC) [5]. The max Cp value (Cp_{max}) and the corresponding lambda (λ_{opt}) value can be detected. λ_{opt} and Cp_{max} are obtained from the Cp curves. Using Eq(2), the wind speed corresponding to the max power can be estimated as follows:

$$V_{est} = \frac{R\omega_{LSS}}{\lambda_{opt}} \quad (23)$$

Therefore, the optimal torque that maximizes the power can be calculated based on Equation (2-3) as follows:

$$T_{aero}^{opt} = \frac{Cp_{max}}{2\lambda_{opt}} \rho \pi R^3 V^2 \quad (24)$$

Hence, the optimal generator torque value can be calculated as follows:

$$\begin{aligned} T_{em}^{opt} &= \frac{T_{aero}^{opt}}{\eta_g} = \frac{Cp_{max}}{2\eta_g \lambda_{opt}^3} \rho \pi R^5 \omega_{LSS}^2 \\ &= k_{mppt} \omega_{LSS}^2 \end{aligned} \quad (25)$$

2-3-2- TSR tracking PI controller:

Modeling errors may significantly affect the reliability of the traditional OTC method, as modern rotor flexibility varies with wind speeds. Therefore, the TSR tracking controller has been adopted. This controller can hold the TSR at its optimal value based on the wind speed (measured or

estimated value). A PI controller is used to minimize the error between the measured and reference generator speed values, where the reference speed is calculated as follows:

$$\omega_{ref} = \eta_g \frac{\lambda_{opt} V}{R} \leq \omega_{rated} \quad (26)$$

The closed-loop of Equation (16) with a PI controller in the below-rated operation region can be presented as follows:

$$H_{Tem}(s) = \frac{B_{Tem} (k_{pTSR}s + k_{iTSR})}{s^2 + s(B_{Tem}k_{pTSR} - A_{\omega_{HSS}}) + B_{Tem}k_{iTSR}} \quad (27)$$

The PI controller parameters can be calculated as follows:

$$k_{pTSR} = \frac{1}{B_{Tem}} (2\zeta_{TSR} \omega_{nTSR} + A_{\omega_{HSS}}) = -\frac{J_{mec}}{\eta_g^2} (2\zeta_{TSR} \omega_{nTSR} + A_{\omega_{HSS}}) \quad (28)$$

$$k_{iTSR} = \frac{\omega_{nTSR}^2}{B_{Tem}} = -\frac{J_{mec}}{\eta_g^2} \omega_{nTSR}^2 \quad (29)$$

where ω_{nTSR} represents the natural frequency controller (=0.15 rad/s for 5MW NREL FOWT), and ζ_{TSR} represents its damping factor (=0.7 for 5MW FOWT). Often, they are tuned for $V=V_{rated}$.

2-4- Region II controller

2-4-1- Torque controller

It is possible to control the torque in this region by one of two methods: constant torque control, Equation (30), or constant power control, Equation (31):

$$T_{em} = \frac{P_{rated}}{\omega_{rated}} \quad (30)$$

$$T_{em}(t) = \frac{P_{rated}}{\omega_{HSS}(t)} \quad (31)$$

In constant torque control, variations in power output are directly linked to variations in generator speed. If constant power control is used, there are still some changes in power output because the generator speed feedback is low-pass filtered, but the power is more consistent. Jonkman [6] asserts that employing a constant generator torque controller can enhance the stability of the floating platforms.

2-4-2- Pitch controller (GSPI baseline controller):

In this region, the generator speed and power are regulated around their rated values, so Equation (14) can be rewritten as:

$$\frac{d}{dt}(\omega_{rated} + \Delta\omega_{HSS}) = \frac{\eta_g}{J_{mec}} (T_{aero} - \eta_g T_{em}) \quad (32)$$

Where the aerodynamic and generator torques are simplified using first-order Taylor series expansion as follows:

$$T_{aero} = \eta_g \frac{P_{aero}}{\omega_{HSS}} = \eta_g \left(\frac{P_{aero}}{\omega_{rated}} + \frac{1}{\omega_{rated}} \left(\frac{\partial P_{aero}}{\partial \beta} \right) \Delta \beta \right) \quad (33)$$

$$T_{em} = \frac{P_{nom}}{\omega_{HSS}} = \frac{P_{nom}}{\omega_{rated}} - \frac{P_{nom}}{\eta_g \omega_{rated}^2} \Delta \omega_{HSS} \quad (34)$$

Replacing in Equation (32):

$$\begin{aligned} \frac{d}{dt} \Delta \omega_{HSS} &= \frac{\eta_g}{J_{mec}} \left(\eta_g \left(\frac{P_{aero}}{\omega_{rated}} + \frac{1}{\omega_{rated}} \left(\frac{\partial P_{aero}}{\partial \beta} \right) \Delta \beta \right) - \eta_g \left(\frac{P_{nom}}{\omega_{rated}} - \frac{P_{nom}}{\eta_g \omega_{rated}^2} \Delta \omega_{HSS} \right) \right) \\ &= \frac{\eta_g^2}{J_{mec} \omega_{rated}} \left(\left(\frac{\partial P_{aero}}{\partial \beta} \right) \Delta \beta + \frac{P_{nom}}{\eta_g \omega_{rated}} \Delta \omega_{HSS} \right) \end{aligned} \quad (35)$$

On the other hand, $\Delta \beta$ is generated by a PI controller. Hence, Equation (35) can be rewritten as:

$$\begin{aligned} \frac{d}{dt} \Delta \omega_{HSS} &= \frac{\eta_g^2}{J_{mec} \omega_{rated}} \left(\frac{\partial P_{aero}}{\partial \beta} \left(k_{pGSPI} \Delta \omega_{HSS} + k_{iGSPI} \int \Delta \omega_{HSS} dt \right) + \frac{P_{nom}}{\eta_g \omega_{rated}} \Delta \omega_{HSS} \right) \\ &= \frac{\eta_g^2}{J_{mec} \omega_{rated}} \left(k_{pGSPI} \frac{\partial P_{aero}}{\partial \beta} \Delta \omega_{HSS} + k_{iGSPI} \frac{\partial P_{aero}}{\partial \beta} \int \Delta \omega_{HSS} dt + \frac{P_{nom}}{\eta_g \omega_{rated}} \Delta \omega_{HSS} \right) \end{aligned} \quad (36)$$

Based on the recommendation in [7], using Laplace transformation and by setting $de/dt = \Delta \omega_{LSS}$

$$e \left(s^2 + \frac{k_{pGSPI} \eta_g^2 \left(-\frac{\partial P_{aero}}{\partial \beta} \right)}{J_{mec} \omega_{rated}} s + \frac{k_{iGSPI} \eta_g^2 \left(-\frac{\partial P_{aero}}{\partial \beta} \right)}{J_{mec} \omega_{rated}} \right) = e \left(s^2 + \frac{\zeta_{GSPI}}{\omega_{nGSPI}} s + \frac{1}{\omega_{nGSPI}^2} \right) = 0 \quad (37)$$

Therefore, the control gains are calculated as follows:

$$k_{pGSPI} = 2\zeta_{GSPI} \omega_{nGSPI} \frac{J_{mec} \omega_{rated}}{\eta_g^2 \left(-\frac{\partial P_{aero}}{\partial \beta} \right)} = 2\zeta_{GSPI} \omega_{nGSPI} \frac{J_{mec}}{\eta_g \left(-\frac{\partial T_{aero}}{\partial \beta} \right)} = \frac{-2\zeta_{GSPI} \omega_{nGSPI}}{B_\beta} \quad (38)$$

$$k_{iGSPI} = \omega_n^2 \frac{J_{mec} \omega_{rated}}{\eta_g^2 \left(-\frac{\partial P_{aero}}{\partial \beta} \right)} = \omega_n^2 \frac{J_{mec}}{\eta_g \left(-\frac{\partial T_{aero}}{\partial \beta} \right)} = -\frac{\omega_{nGSPI}^2}{B_\beta} \quad (39)$$

According to [8], the PI gains change linearly with the change in the pitch angle. Therefore, the control law, including the gain-scheduling correction factor, can be presented as follows:

$$\Delta \beta = \alpha(\beta) \left(k_{pGSPI} e(t) + k_{iGSPI} \int e(t) dt \right) \quad (40)$$

$$\alpha(\beta) = \frac{1}{1 + \beta / \beta_k} \quad (41)$$

where β_k symbolizes the pitch angle at which the pitch angle sensitivity value doubles from that at the rated operating point.

References

- [1] J. M. Jonkman and M. L. Buhl, "FAST Users Guide NREL," Golden, CO (United States), 2005.
- [2] I. Guenoune, F. Plestan, A. Chermitti, and C. Evangelista, "Modeling and robust control of a twin wind turbines structure," *Control Eng. Pract.*, vol. 69, pp. 23–35, Dec. 2017, doi: 10.1016/j.conengprac.2017.08.009.
- [3] S. Hu and B. Song, "Optimal Control Strategies of Wind Turbines for Load Reduction," in *Modeling and Modern Control of Wind Power*, 2017, pp. 63–83. doi: 10.1002/9781119236382.ch4.
- [4] C. L. Bottasso and A. Croce, "Power curve tracking with tip speed constraint using LQR regulators," *Sci. Rep. DIA-SR*, no. December, pp. 1–19, 2009, [Online]. Available: http://www.aero.polimi.it/bottasso/downloads/CLBottasso_ACroce_2009c.pdf
- [5] M. Yin, W. Li, C. Y. Chung, L. Zhou, Z. Chen, and Y. Zou, "Optimal torque control based on effective tracking range for maximum power point tracking of wind turbines under varying wind conditions," *IET Renew. Power Gener.*, vol. 11, no. 4, pp. 501–510, Mar. 2017, doi: 10.1049/iet-rpg.2016.0635.
- [6] J. Jonkman, "Definition of the Floating System for Phase IV of OC3," *Contract*, vol. 1, no. May, p. 31, 2010, [Online]. Available: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Definition+of+the+Floating+System+for+Phase+IV+of+OC3#0>
- [7] M. H. Hansen, A. Hansen, T. J. Larsen, S. Øye, P. Sørensen, and P. Fuglsang, *Control design for a pitch-regulated, variable speed wind turbine*, vol. 1500, no. January. Roskilde, Denmark, 2005. [Online]. Available: https://backend.orbit.dtu.dk/ws/portalfiles/portal/7710881/ris_r_1500.pdf
- [8] J. M. Jonkman, "Dynamics modeling and loads analysis of an offshore floating wind turbine," *Natl. Renew. Energy Lab. NREL/TP-500-41958*, vol. 68, no. November, p. 233, Dec. 2007, doi: 10.2172/921803.

Appendix B: Glossary

Instructions: Provide clear and concise definitions for terms used in this document that may be unfamiliar to readers of the document. Terms are to be listed in alphabetical order.

Table 3 - Glossary

Term	Acronym	Definition
<Term>	<Acronym>	<Definition>
<Term>	<Acronym>	<Definition>
<Term>	<Acronym>	<Definition>

Appendix C: Referenced Documents

Instructions: Summarize the relationship of this document to other relevant documents. Provide identifying information for all documents used to arrive at and/or referenced within this document (e.g., related and/or companion documents, prerequisite documents, relevant technical documentation, etc.).

Table 4 - Referenced Documents

Document Name	Document Location and/or URL	Issuance Date
<Document Name>	<Document Location and/or URL>	<MM/DD/YYYY>
<Document Name>	<Document Location and/or URL>	<MM/DD/YYYY>
<Document Name>	<Document Location and/or URL>	<MM/DD/YYYY>

Appendix D: Approvals

The undersigned acknowledge that they have reviewed the User Manual and agree with the information presented within this document. Changes to this User Manual will be coordinated with, and approved by, the undersigned, or their designated representatives.

Instructions: List the individuals whose signatures are desired. Examples of such individuals are Business Owner, Project Manager (if identified), and any appropriate stakeholders. Add additional lines for signature as necessary.

Table 5 - Approvals

Document Approved By	Date Approved
----- Name: <Name>, <Job Title> - <Company>	----- Date
----- Name: <Name>, <Job Title> - <Company>	----- Date
----- Name: <Name>, <Job Title> - <Company>	----- Date
----- Name: <Name>, <Job Title> - <Company>	----- Date

Appendix E: Notes to the Author/Template Instructions

This document is a template for creating a User Manual for a given investment or project. The final document should be delivered in an electronically searchable format. The User Manual should stand on its own with all elements explained and acronyms spelled out for reader/reviewers, including reviewers outside CMS who may not be familiar with CMS projects and investments.

This template was designed based on best practices and information to support CMS governance and IT processes. Use of this template is not mandatory, rather programs are encouraged to adapt this template to their needs by adding or removing sections as appropriate. Programs are also encouraged to leverage these templates as the basis for web-based system development artifacts.

This template includes instructions, boilerplate text, and fields. The author should note that:

- Each section provides instructions or describes the intent, assumptions, and context for content included in that section. Instructional text appears in blue italicized font throughout this template.*
- Instructional text in each section should be replaced with information specific to the particular investment.*
- Some text and tables are provided as boilerplate examples of wording and formats that may be used or modified as appropriate.*

When using this template, follow these steps:

- 1. Table captions and descriptions are to be placed left-aligned, above the table.*
- 2. Modify any boilerplate text, as appropriate, to your specific project.*
- 3. All documents must be compliant with Section 508 requirements.*
- 4. Figure captions and descriptions are to be placed left-aligned, below the figure. All figures must have an associated tag providing appropriate alternative text for Section 508 compliance.*
- 5. Delete this "Notes to the Author/Template Instructions" page and all instructions to the author before finalizing the initial draft of the document.*
- 6.*