

ARPA-E ATLAS COMPETITION

MODELING, CONTROL, AND SIMULATION DOCUMENT

This document includes the guidelines to implement the controllers and run the simulation for the ARPA-E ATLAS Competition. Participants are urged to download the simulation package from the competition website and extract the various files and folders. Extracting the contents of the download package will produce the various folders referred to in the discussions below. Section 6 describes this process and the folders in more detail.

Questions regarding the rules, privacy policy, or other aspects of the ATLAS Competition may be directed to atlascompetition@nrel.gov.

1. Goals

The goal of the ATLAS Competition is to design advanced blade pitch controllers that result in improved performance compared to the standard baseline controllers in use today. This is particularly critical for reducing levelized cost of energy (LCOE) in large onshore and offshore wind turbine systems.

In this competition, participants are asked to design a blade pitch controller with collective pitch control (CPC) and individual pitch control (IPC).

The system for the focus of this competition is the NREL 5 MW baseline turbine. For this competition, there are two Challenges. In the *Onshore Challenge*, participants will design an improved blade pitch controller that governs normal operation in above rated conditions (excludes start-ups, shut-downs, and fault cases) for the NREL 5 MW land-based machine [1]. The *Offshore Challenge* will focus on an improved blade pitch controller for normal operation of an offshore floating version of this machine. Specifically, the NREL 5 MW baseline turbine model is attached to the OC3 Hywind Spar floating platform [2].

For general overviews of wind turbine controls technology, please see [3, 4, 5, and 6].

2. NREL 5 MW Baseline Turbine Description

The NREL 5 MW baseline land-based turbine model represents a typical 3-bladed variable speed upwind machine. Please see [1] for a full turbine description along with numerical values of all model input parameters for this machine.

2.1. Onshore Challenge

The NREL 5 MW baseline controller in this model is composed of two parts for normal operation: the generator torque controller and the blade pitch controller. The generator torque controller optimizes energy capture in below rated wind-speeds (Region 2) using a typical $k\Omega^2$ torque control law, where k is a constant, and Ω is measured generator or rotor speed [1]. Generator torque is held constant at the rated torque value in above rated wind speeds (Region 3). A linear transition region is added to allow smooth transition between Regions 2 and 3 (Region 2-1/2).



Advanced Research Projects Agency • ENERGY

The rotor collective pitch controller (all three blades are pitched identically) regulates turbine speed to a required set-point in above rated wind-speeds (Region 3). In Region 2 blade collective pitch is held constant at a minimum pitch angle.

A thorough description of the modeling for the NREL 5 MW turbine baseline generator torque and rotor collective pitch controller can be found in [1]. Table 7-2 (page 27) of [1] summarizes numerical values of key turbine properties and control parameters. Design of the yaw and high-speed shaft brake controllers is not part of this competition and will not be described.

2.2. Offshore Challenge

The NREL 5 MW baseline turbine attached to the OC3 Hywind floating spar platform is fully described in [2]. The turbine properties for the blades, drive-train and nacelle are identical to the *Onshore Challenge* turbine properties. The *Offshore Challenge* turbine's tower properties are different than the *Onshore Challenge* tower properties. The turbine, platform, and mooring system properties are described in detail in [2].

The NREL 5 MW baseline control system for the OC3-Hywind floating spar platform is nearly identical to the baseline control system for the land-based version of the NREL 5 MW. The main change is that the land-based blade collective pitch controller gains (Proportional-Integral [PI] gains) have been de-tuned in order to prevent controller induced excitation of the platform's pitching degree of freedom (DOF). Otherwise, the baseline control systems for the two configurations are identical. Please see [2] for details. A goal of the competition is to improve the controls by designing improved blade pitch controllers.

3. Modeling and Simulation Software

For modeling and simulation of the NREL 5 MW turbine, participants must use the OpenFAST wind turbine simulator, developed by NREL [7], in conjunction with Simulink [8]. OpenFAST is an open-source wind turbine simulation tool established in 2017, with the FAST v8 code [9] as its starting point. Please see [https://openfast.readthedocs.io/en/master/source/user/fast_to_openfast.html] for transitioning from FAST Version 8 [9] to OpenFAST [7]. The appropriate OpenFAST executables and Simulink functions are supplied to the participants in the package download, eliminating the need for participants to compile OpenFAST on their own.

3.1. OpenFAST input and output files

For a description of the FAST version 8 (FAST v8) input and output files, please see [9] page 17. OpenFAST has an identical input and output file format to FAST v8. Table 1 lists the input files required by OpenFAST and its modules for this competition. The naming conventions will be described later when the load simulation cases and demo are discussed (Sections 5 and 6). In general, the symbol "x" in front of each file extension will be populated with a name that indicates the particular load case being simulated. All of the required input files for each module and for each load case will be automatically generated from a template file at the beginning of the load suite simulation, to be described in Section



6. The template file is supplied in the download. The participants should not need to configure their own input files.

Table 1: OpenFAST Input Files

File Name	Description	Challenge
x_.fst	Primary OpenFAST input file	Onshore and Offshore
x_IW.dat	Contains inflow wind input parameters	Onshore and Offshore
x_AD15_.dat	Contains aerodynamic input parameters (AeroDyn version 15)	Onshore and Offshore
x_SD.dat	Contains control and electrical-drive input parameters	Onshore and Offshore
x_HD.dat	Contains hydrodynamic input parameters (not used for land-based case)	Offshore
x_MAP.dat	Contains mooring system input parameters	Offshore
x_ED_Blade.dat	Contains blade structural properties for ElastoDyn	Onshore and Offshore
x_ED_Tower.dat	Contains tower structural properties for ElastoDyn	Onshore and Offshore
x_AD15_blade.dat	Contains distributed aerodynamic blasé properties for AeroDyn15	Onshore and Offshore

Participants are free to modify the input files in this competition download for their own controller testing purposes. However, they should perform final controller simulation testing using the unmodified versions of these files as supplied in the download. The participants’ controllers will be evaluated using these original input files (see ATLAS Rule R23).

Output files from OpenFAST (using Simulink for simulation) have the form “_SFunc.out” (or “_SFunc.outb” for binary output files). The underscore before the file extension will hold a name indicating the particular load case being simulated. The contents of this output file consist of the output channels contained in an array named “OutList.” This array is set within OpenFAST and depends on the sensor signals entered by name into the OUTPUT sections of the different OpenFAST module input files. Participants can select the sensors used for the IPC system from a list of available signals. This will be described further in Section 4.1 below.

3.2. OpenFAST Executables

A number of executable files required to run OpenFAST will be provided to the participants (see Section 7). These executable files assume OpenFAST has been compiled for the Windows 64-bit operating system. Executable files to run OpenFAST on other operating systems are not supplied. These Dynamic Link Library (DLL) files (and a “mex” compiled OpenFAST S-function) appear in the download folder named “_Functions/OpenFASTLibraries” for both challenges.



3.3. Simulating Controllers in OpenFAST

OpenFAST [7] provides different options for incorporating controllers for closed-loop simulation [7, 9, 10]. This competition will exclusively use Simulink with OpenFAST for closed-loop controller simulation testing.

Simulink has the ability to incorporate an OpenFAST turbine model. The OpenFAST routines have been linked with a standard Matlab gateway subroutine in order to represent the OpenFAST equations of motion as an S-Function in Simulink [9, 10]. For this competition, the compiled S-Function will be in the form of a “mex” function named “FAST_SFunc.mexw64” (for 64-bit). This function is supplied to the participants in the download package. The participants should not have to compile their own versions of this function.

Participants should make exclusive use of OpenFAST linked with Simulink for controller implementation and simulation testing. The Simulink models can include m- or p-files that contain Matlab programming that provides the required controller logic.

4. Wind Turbine Controller Implementation Guidelines: NREL 5MW Baseline Turbine Controller Simulink Model

The NREL 5 MW baseline controller Simulink model is identical for both Onshore and Offshore Challenges. The only change is that the *Onshore Challenge* baseline blade pitch controller gains have been de-tuned for the *Offshore Challenge* (the numerical values for these gains are included in an m-file to be described below).

Figure 1 shows the overall layout of the NREL 5MW baseline turbine controller Simulink model. Sensor signals from the “OpenFAST Turbine Model” (“y”) are used as inputs to the “NREL Baseline Generator Torque Control” and “NREL Baseline Blade Pitch Control” blocks. The outputs of the controllers (“M_g_c” for generator torque and “theta_c” for blade pitch) are stacked into the vector signal “u” for input to the “OpenFAST Turbine Model” block. The goal of this controller is to optimize energy capture in below rated wind speeds and to regulate turbine speed to 12.1 rpm (rated speed). If the controller results in the rotor’s speed reaching or exceeding 15.73 rpm (this includes an overspeed of 30%) the cost function used to judge controller performance is given a value of 1000, resulting in controller disqualification. Please see Section 5 and Table 7, which describes this constraint and others used to evaluate controller performance.

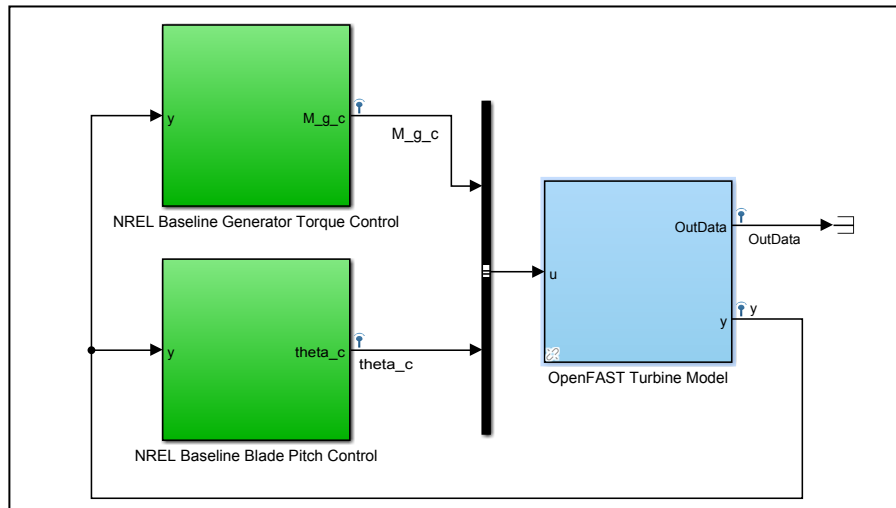


Figure 1: NREL 5MW OpenFAST Turbine Model controlled by the Baseline Generator Torque and Baseline Blade Pitch Controllers.

4.1. OpenFAST Turbine Model Block

The contents of the “OpenFAST Turbine Model” block is shown in Figure 2. For clarity, Figure 2 (top) shows details to the left of the “OpenFAST S-Function” block, and Figure 2 (bottom) shows details to the right of this block. Figure 2 (top) shows the generator torque (“M_g_c”), and the blade pitch commands (“theta_c”) extracted from the input vector “u.” They are passed to the “Generator Torque (limits) and Power” block and to the blade pitch actuator blocks.

The input vector “theta_c”, which has three components, is separated (de-muxed) into individual blade pitch components. Each blade’s pitch angle command is passed to a pitch actuator. The pitch actuators for all three blades are identical since we assume identical blades in the CPC case.

The nacelle “Yaw Position” and “Yaw rate” inputs to the FAST S-Function are zero, representing fixed yaw. The high-speed shaft brake torque (“HSS Brake Torque”) is also set to zero.

Figure 2 (bottom) highlights signal extraction from OpenFAST. The OpenFAST “S-Function” outputs a variable array named “OutData” that contains the signals defined in the input files for the various OpenFAST modules (in the OUTPUT sections). Some of these signals can be selected for the IPC case. A corresponding array named “OutList” is set-up within OpenFAST and is made available to the Matlab workspace during Simulink simulation. This allows the individual sensor signals to be accessed in Simulink.

An individual sensor signal is extracted from “OutData” by passing it through a block that contains a string match statement. Please see Figure 2 (bottom) as an example. The particular signal name being extracted must be contained in the “OutData” array or an error message will occur.

An extensive set of sensor signals are already provided in this competition through the input file definitions. Participants should not have to add additional signals to this list. The complete list that is

made available can be determined by looking in the input files, going to the section under the OUTPUT headers, and finding the different signal definitions included in these files. The signal names are grouped according to the particular module input file they are associated with. The signal names associated with different OpenFAST modules can be found the Excel sheet “OutListParameters.xls” provided in the file download.

Details of the “Generator Torque (limits) and Power” block are shown Figure 3 (top). The generator power is calculated based on the commanded generator torque, measured rotor speed and the generator efficiency (“Parameter.Generator.eta_el”). A one-time-step delay is applied to the generator torque and rotor speed signals in order to prevent issues with algebraic loops in Simulink. A generator torque rate limiter (“Parameter.Generator.M_g_dot_max”) is applied to the delayed generator torque input. Numerical values for these parameters are set in the m-file named “fSetControllerParametersOnshore.m” for the *Onshore Challenge* and “fSetControllerParametersOffshore.m” for the *Offshore Challenge*, located in the “_Controller” folder in the download package.

One of the blade pitch actuator blocks is depicted in Figure 3 (bottom). This block places angle and rate limits on the commanded blade pitch and also models the actuator dynamics (pitch motor) by a simple 2nd order linear model. This model can be expressed in the Laplace domain as:

$$y(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} u(s) \quad (1)$$

where $u(s)$ is the commanded pitch angle (output from the pitch controller block), and $y(s)$ the blade pitch angle after passing through the actuator model. The parameter, ζ , (“Parameter.PitchActuator.xi”) is the damping factor, and ω_n (“Parameter.PitchActuator.omega”) is the natural frequency (rad/s) of this 2nd order system.

The pitch actuator can be switched based on a parameter that is assigned the values 0, 1, or 2 depending on the value of a parameter (“Parameter.PitchActuator.Mode”). The value 0 denotes no pitch actuation, value 1 switches to a delay, and value 2 switches to the 2nd order pitch actuator model (Eq. (1)). Pitch angle (“Parameter.Pitch.Actuator.theta_max(min)”) and rate (“Parameter.Pitch.Actuator.theta_dot_max(min)”) limits are applied to the pitch commands in the block “P2.” Numerical values for all of these parameters are defined in the m-file named “fSetSimulinkParameters.m” located in the “_Functions” folder in the file download and extraction (to be described in Section 6). Numerical values in this file must not be changed by the participants.

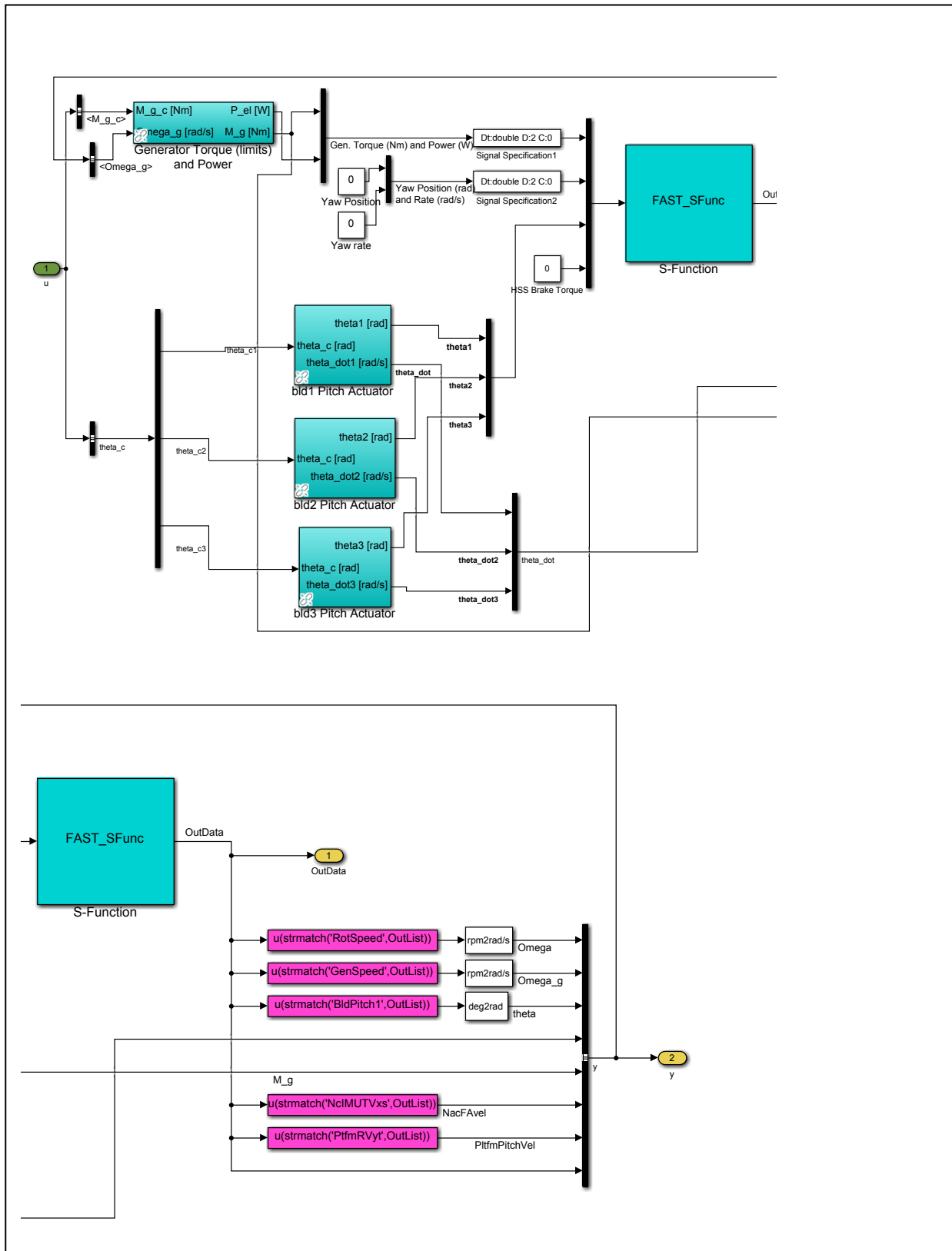


Figure 2: (top) Inside the turbine model block: generator torque limits and blade pitch actuator; (bottom) Inside the turbine model block: sensor signals from OpenFAST for input to the controllers

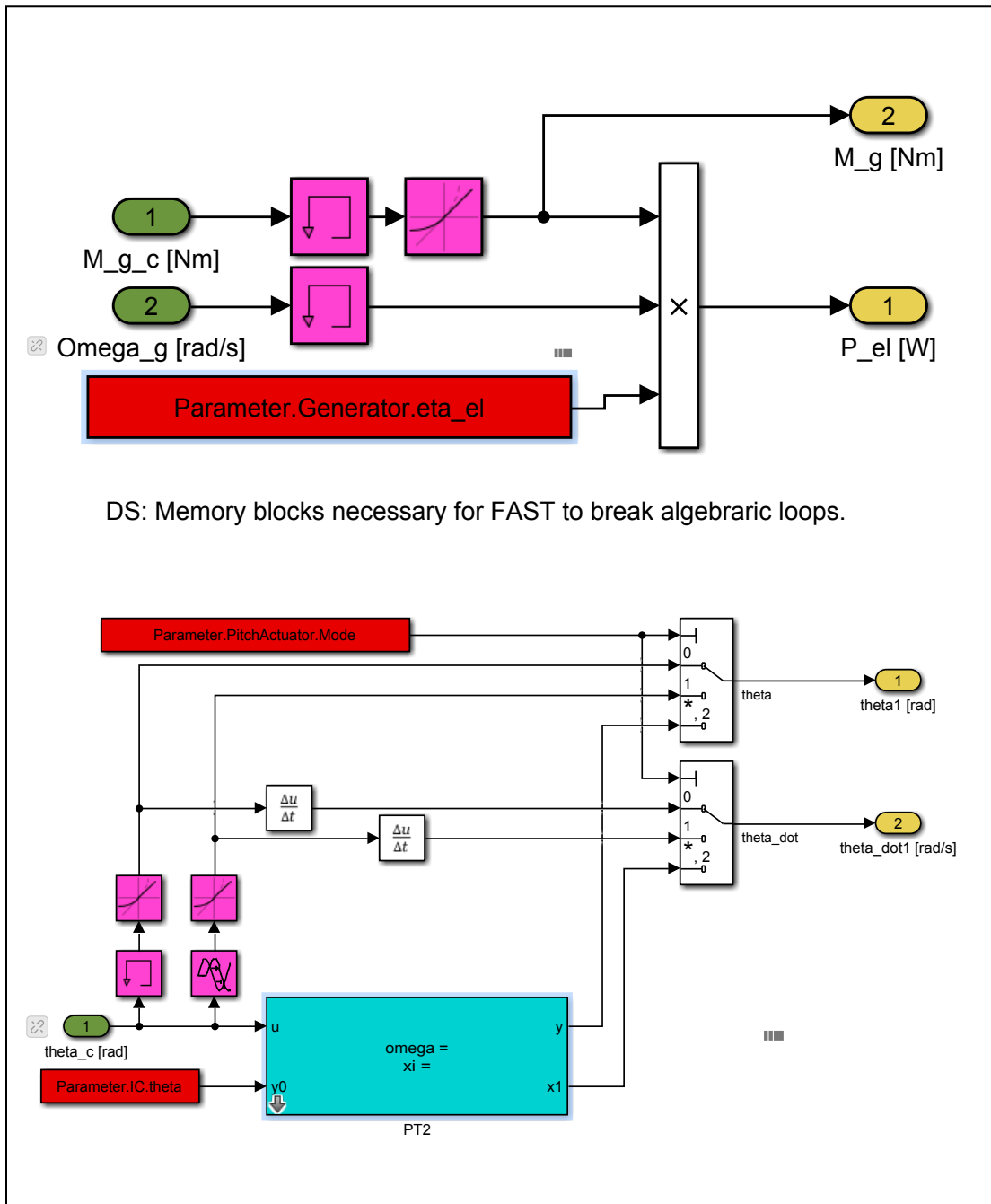


Figure 3: (top) Inside the “Generator Torque (limits) and Power” block; (bottom) Inside one of the blade pitch actuator blocks

4.2. Baseline Generator Torque Controller

The NREL Baseline Generator Torque Control box contains the elements shown in Figure 4. As shown in Figure 4 (top), the generator speed signal (“Omega_g”) and blade pitch angle (“theta”) are extracted from the sensor signal “y.” The blade pitch angle is used in the generator torque algorithm to decide if the turbine is operating in Region 3.

A low-pass filter (“FilterGenSpeed.Omega_g”) is applied to the generator speed signal before it is input to the generator torque controller. This simple 1st order filter can be described by:

$$y(s) = \frac{1}{Ts+1}u(s) \tag{2}$$

where $u(s)$ is the input generator speed signal, and $y(s)$ is the filtered generator speed signal. The time-constant T (“Parameter.Filter.FilterGenSpeed.T63”) is assigned a numerical value in the m-file “fSetSimulinkParameters.m” located in the “_Functions” folder produced during download and file extraction (Section 6). Again, these parameters must not be changed by the participants.

Figure 4 (bottom) shows the contents inside of the filter block. An additional parameter (“Parameter.Filter.FilterGenSpeed.Omega_g.Enable”) indicates whether the filter is enabled or not. This parameter is also set in the m-file discussed above.

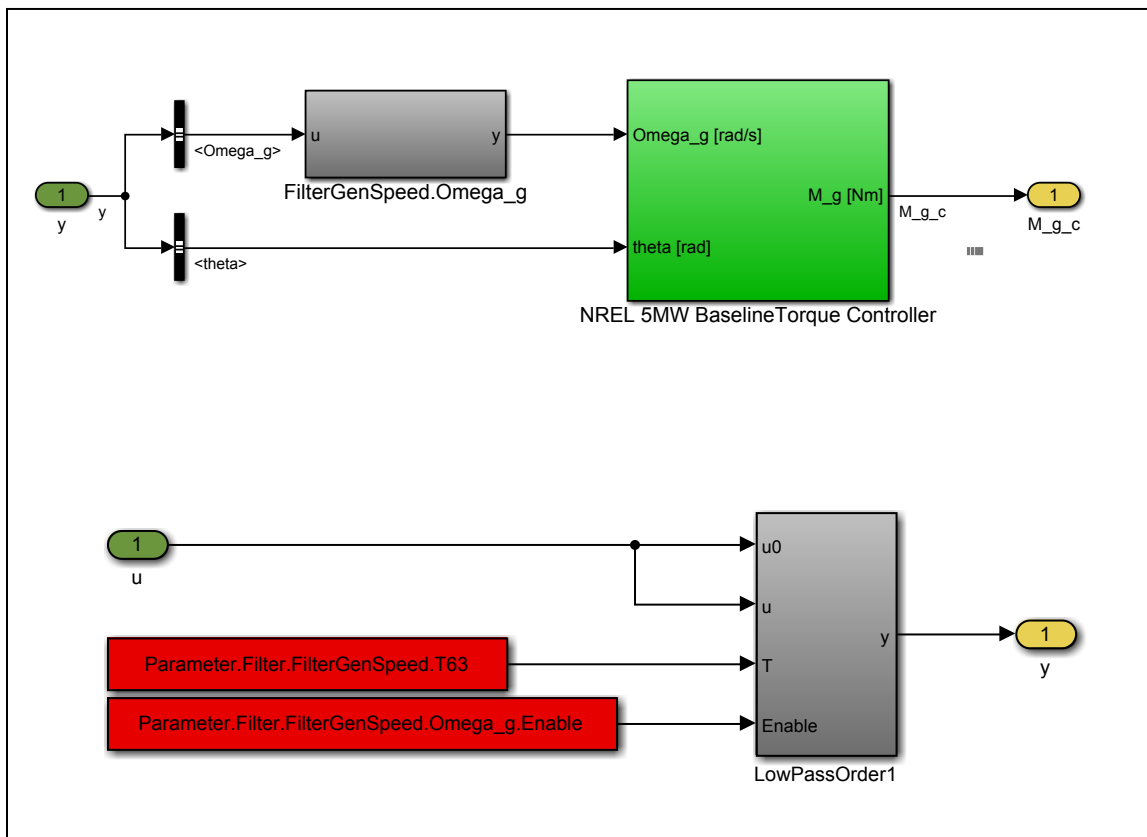


Figure 4: (top) Inside the “Baseline Generator Torque Control” Block; (bottom) Inside the “FilterGenSpeed.Omega_g” Block

Figure 5 shows the contents of the “NREL 5MW Baseline Torque Controller” block of Figure 4 (top). Several parameters are input to this block which are used in determining the turbine’s current operating region. These parameters are defined in “fSetSimulinkParameters.m” (not to be changed) located in the

“_Functions” folder (see Section 6). Generator torque is calculated inside this block, based on filtered generator speed input. The “Baseline Torque Controller” block contains Matlab code that calculates the generator torque depending on the turbine’s current operating region and generator speed. The user can examine the contents of this block by opening it in Simulink. The output of this block is the commanded generator torque signal “M_g.”

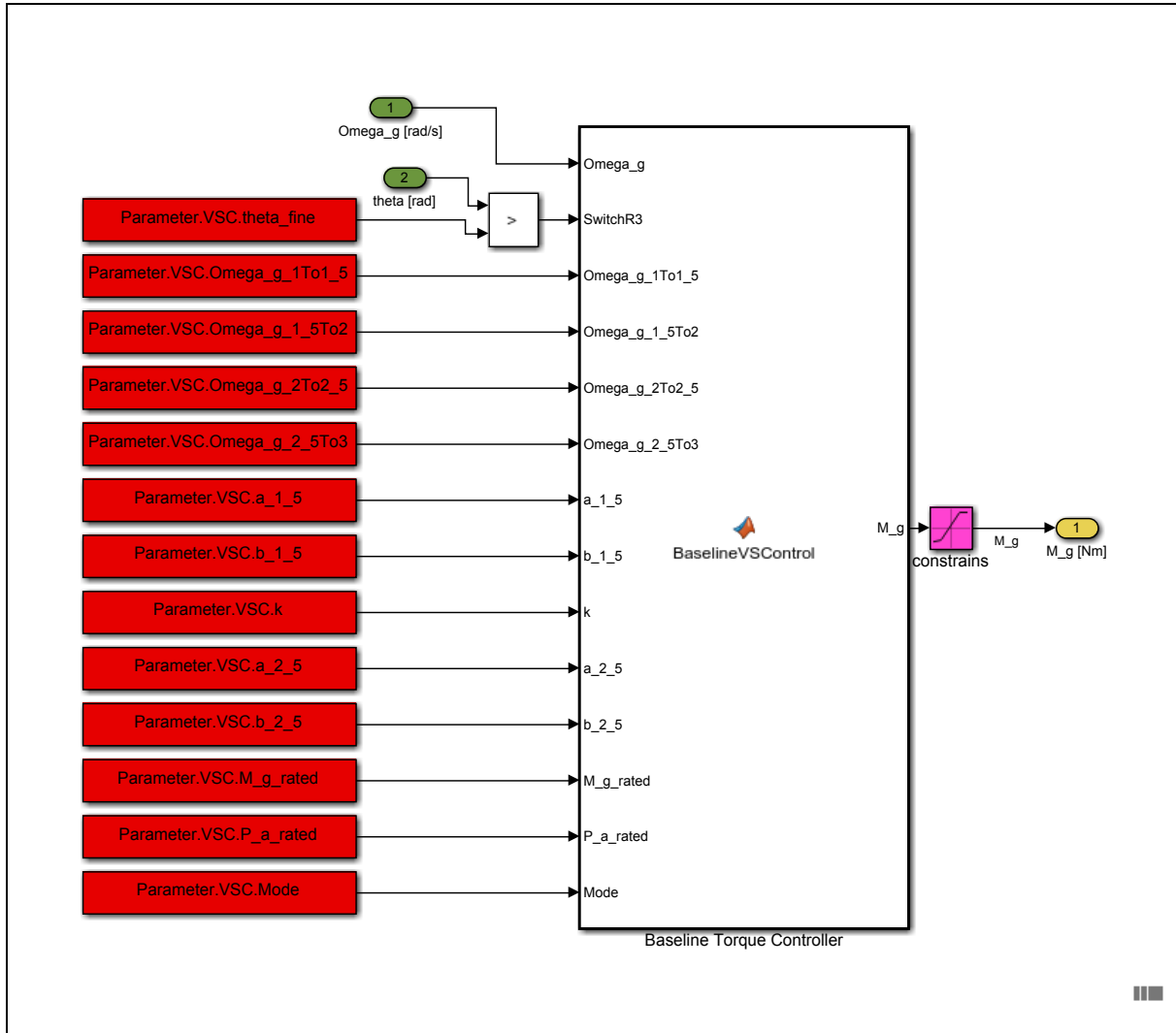


Figure 5: Inside the “NREL 5MW Baseline Torque Controller Block”

4.3. Rotor Collective Pitch Controller (CPC)

The inside of the “NREL Baseline Blade Pitch Control” block is shown in Figure 6 (top). This controller commands collective blade pitch. The generator speed is again filtered by the same low-pass filter (Eq. (2)) used in the generator torque controller. In addition, the measured blade pitch angle is used to gain

schedule the controller. For a discussion of the equations used to model this controller including the method of implementing gain scheduling, please see [1] pages 20-26.

Figure 6 (middle) shows the “NREL 5MW Baseline Pitch Controller” block. Additional control parameters are defined in the m-file “fSetControllerParametersOnshore.m” (*Onshore Challenge*) or “fSetControllerParametersOffshore.m” (*Offshore Challenge*) located in the “_Controller” folder (Section 6). The parameters are associated with the Proportional-Integral (PI) controller gains (“Parameter.CPC.kp” and “Parameter.CPC.Ti”), the upper pitch angle limit (“Parameter.CPC.theta_max”), lower pitch angle limit (“Parameter.CPC.theta_min”), the pitch angle reference value for gain scheduling (“Parameter.CPC.theta_K”), and rated generator speed (“Parameter.CPC.Omega_g_rated”).

The inside of the “PI Anti-Windup with rate feedforward” block is shown in Figure 6 (bottom). The generator speed error (difference between current generator speed and rated generator speed) is multiplied by the controller’s proportional gain, K_p . A signal that accounts for anti-windup (see below) is added to this signal and the resulting sum is divided by the gain T_i . Here, $T_i = \frac{K_p}{K_i}$, where K_i is the integral gain. This signal is then passed to the integrator block to form the controller’s integrated speed error term (the “u_dot_FF” term is zero, so no derivative term). The proportional error term is then added to this signal. This signal then passes through the pitch angle limiter block, which applies minimum and maximum pitch angle limits (“u_min” and “u_max”) to this signal.

The anti-windup feedback signal consists of the error term formed by subtracting the unlimited pitch angle from the limited pitch angle command. This error term is fed back and added to the proportional error term which is then input to the integrator. For more information on integrator anti-windup methods, please see [4]. The final pitch angle command is formed by combining the pitch angle from each blade into a stacked (muxed) vector having three components. Since we are commanding collective pitch, these signals are all equal.

Please note that the PI controller gains have been detuned for the *Offshore Challenge* case. For more information on these gain selections, please see [2]. The controller gains are given numerical values in an m-file to be described in Section 5. There are two set of these gains, one for the *Onshore Challenge* and one for the *Offshore Challenge*. Participants should not have to modify these gains as they are already set in the appropriate m-file. Since the *Onshore Challenge* m-files are separate from the *Offshore Challenge* m-files, the appropriate controller gains are already set correctly in the download.

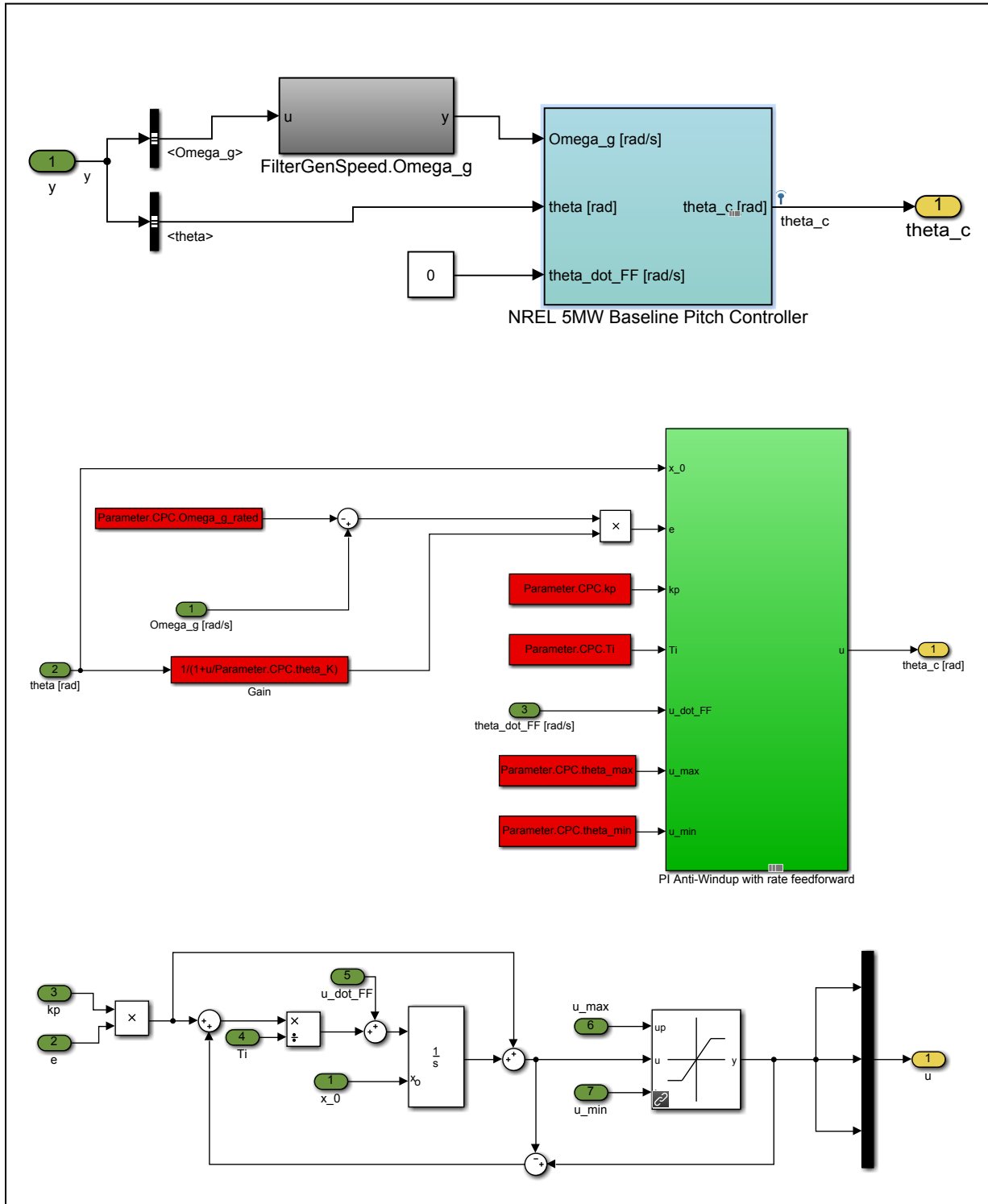


Figure 6: (top) Inside the “NREL Baseline Generator Pitch Control” block; (middle) Inside the “NREL 5MW Baseline Pitch Controller” block; (bottom) Inside the “PI Anti-Windup with rate feedforward” block



4.4. Participant Controller Implementation Guidelines

Participants should use the template described below (the Matlab Simulink model named “NREL5MW_Example_IPC.mdl” which is located in the “_Controller” folder) for incorporating their own controllers. Participants are free to retain the “NREL 5MW Baseline Pitch Controller” and design their own pitch controller that adds pitch increments to the baseline (IPC). Alternatively, participants can eliminate the “NREL 5MW Baseline Pitch Controller” (CPC with PI and Anti-Windup) completely and substitute their own pitch controller.

The objective of the ATLAS Competition is to design and implement blade pitch controllers that improve control performance relative to the NREL Baseline Pitch Controller. These controllers must decrease loads in the turbine components and increase energy capture, while minimizing blade pitch actuator usage compared to the baseline. Individual blade pitch control has been shown to be an important improvement to typical turbine controllers [3, 11, 12]. They assist in reducing blade and component loads due to asymmetric wind distributions across the rotor disk.

Participants’ total pitch controllers must contain an individual pitch control (IPC) and a collective pitch control (CPC). A possible control architecture is to retain the “NREL 5MW Baseline Pitch Controller” with the participants designing only their IPC to add incremental pitch commands to the NREL baseline. Alternatively, participants are free to design their controllers in any way they choose as long as the controllers contain a CPC component and an IPC component.

Figure 7 shows one possible participant Simulink model, with the “Participant’s New Blade Pitch Control (CPC + IPC)” replacing the NREL baseline blade pitch controller. This blade pitch controller has both CPC and IPC components. This Simulink model named NREL5MW_Example_IPC.mdl is included in the “_Controller” folder. The pitch controller block consists of just a shell. It commands just a constant pitch angle value. The participants will be responsible for designing the real “guts” of this controller. There are many different approaches for designing both CPC and IPC controllers. Some example approaches for IPC can be found in [3,12].

The competition participants are free to choose their own control architectures and approaches for the CPC and IPC controller components. They are also free to choose other sensors, provided that they are a subset of the sensors already provided (these sensors should be contained in the “OutList” array that is set-up by OperFAST through the module input files supplied) in this competition download. The participants should not add sensors that are not already in these input files.

Participants should incorporate controller parameters and define their numerical values in the m-file “fSetControllerParametersOnshore.m” (*Onshore Challenge*) or “fSetControllerParametersOffshore.m” (*Offshore Challenge*) provided in the “_Controller” folder. This folder is to be described in Section 6.

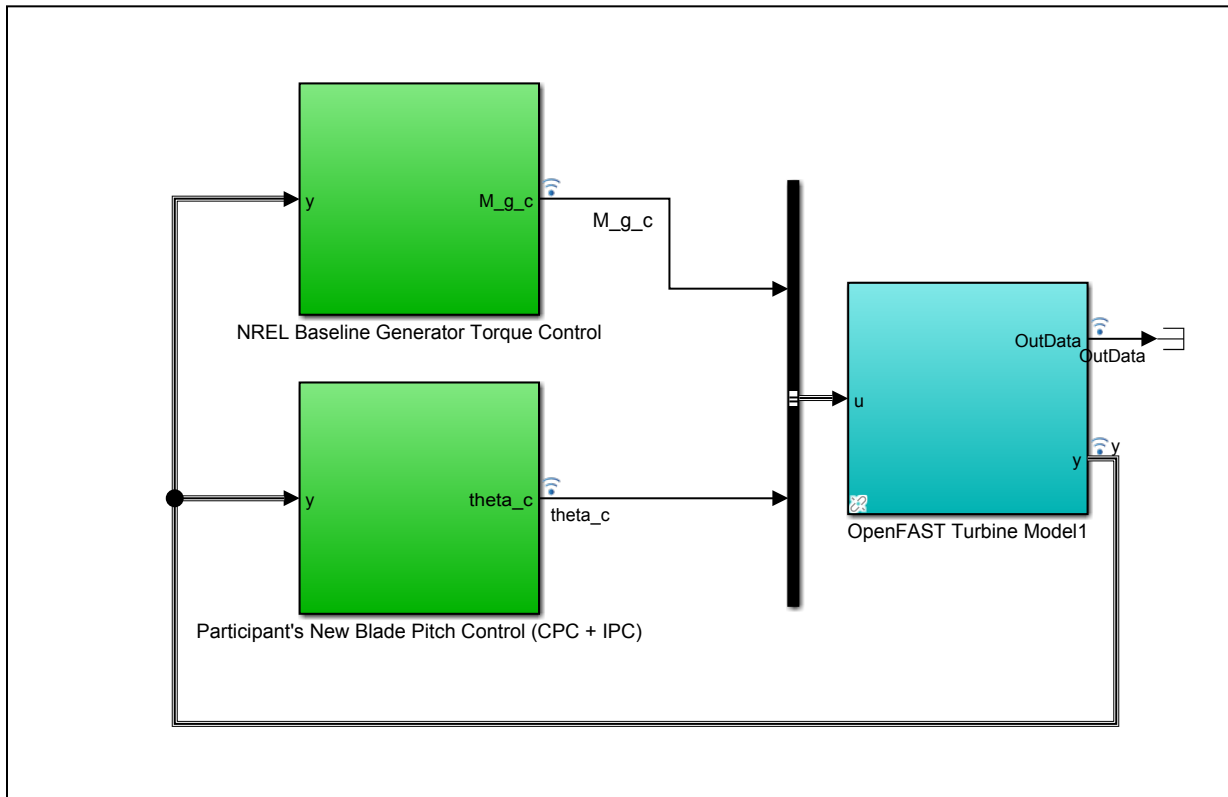


Figure 7: A possible participant control architecture with the participant blade pitch controller replacing the NREL baseline pitch controller. The “NREL Baseline Generator Torque Control” block is not replaced or modified.

For simulation testing of controllers and for scoring the controller performance, a number of load cases will be simulated as described below.

5. Competition Load Case Suite, Performance Metrics, and Controller Cost Function

5.1. Load Case Definitions

A set of 12 load cases are retained for this contest. The load cases are inspired by the design load cases (DLCs) presented in the standards [13]. The cases are restricted to wind speeds above the rated wind speed of 11.4 m/s since the competition focuses on the pitch-controlled region. The number of simulations is significantly reduced compared to the ones required by the standards in order to achieve computational times of about one hour for a sequential evaluation of all the loads cases with Simulink on one machine, while still providing an adequate survey of load cases and conditions.

The load cases are provided in Table 2. Each line of Table 2 represents a simulation and each column provides the conditions used for this simulation. The column “WS” is the mean wind speed of the simulation. The “Yaw error” represents the error of alignment of the turbine compared to the mean wind direction. The wind and wave seeds are used to initialize a random number generator, which is then used to generate different turbulent wind and wave times series. The “Pitch offset” column



Advanced Research Projects Agency • ENERGY

indicates whether the blades were mounted with a pitch error. When this variable is true, blade 1 is mounted correctly, blade 2 is mounted with -1deg of pitch offset, and blade 3 with +1 deg of pitch offset. The column “Wind” indicates the type of wind conditions: Turbulent (Turb), Extreme Coherent gust with Direction Change (ECD), Extreme Wind Shear (EWS), Extreme Operating Gust (EOG), Step. The deterministic wind ECD EWS and EOG and defined in the standards. The step of wind speed corresponds to a step from 15 to 13 m/s. The column “Dirty aero” indicates whether the blade is clean or dirty. In the latter case, the airfoil performances are reduced. This is effectively done in the simulation by using a different set of airfoil coefficients. The column “Time” is the simulation length. The column “CaseName” is used to name the input and output files of the simulations.



Table 2: Design Load Cases

DLC	WS (m/s)	Yaw Error (deg)	Wind Seed	Wave Seed	Pitch Offset	Wind	Dirty aero	Time (s)	Case name
1.2	13.4	-10	2	3	Yes	Turb	No	600	'DLC120_ws13_yeNEG_s2_r3_PIT'
1.2	13.4	0	1	1	No	Turb	No	600	'DLC120_ws13_ye000_s1_r1'
1.2	19.4	-10	3	2	No	Turb	No	600	'DLC120_ws19_yeNEG_s3_r2'
1.2	19.4	0	2	1	Yes	Turb	No	600	'DLC120_ws19_ye000_s2_r1_PIT'
1.2	23.4	0	3	3	No	Turb	No	600	'DLC120_ws23_ye000_s3_r3'
1.4	13.4	0	0	1	No	ECD	No	200	'DLC140_ws13_ye000_s0_r1_ECD'
1.5	13.4	0	0	1	No	EWS	No	200	'DLC150_ws13_ye000_s0_r1_EWS'
2.3	13.4	0	0	1	No	EOG	No	200	'DLC230_ws13_ye000_s0_r1_EOG'
1.21	13.4	0	1	1	No	Turb	Yes	600	'DLC121_ws13_ye000_s1_r1_DRT'
1.21	19.4	0	2	2	No	Turb	Yes	600	'DLC121_ws19_ye000_s2_r2_DRT'
1.21	23.4	0	3	3	No	Turb	Yes	600	'DLC121_ws23_ye000_s3_r3_DRT'
1.22	15.4	0	0	1	No	Step	No	1200	'DLC122_ws15_ye000_s0_r1_STP'

The load cases are gathered into groups referred to as DLC (Design Load Cases). Most of these DLC groups are presented in the IEC-61400-1 standards [13]. The definitions used in this competition are as follows:

- DLC 1.2 represents normal operating conditions. Two yaw errors are used for 13.4 and 18.4m/s, and one yaw error used at 23.4m/s.
- DLC 1.4 and 1.5 correspond to extreme direction change (ECD) and extreme wind shear (EWS), as defined in the standards.
- DLC 1.21 is created for this competition. It is similar to DLC 1.2, but the airfoils are soiled and do not perform as well. This is introduced to evaluate the robustness of the controller in conditions outside of ideal conditions.
- DLC 2.3 is adapted for this competition. An extreme operating gust (EOG) occurs during the turbine operation.
- DLC 1.22 is created for this competition. It contains a step of wind speed from 15 to 13 m/s. This sudden change of wind speed is known to excite the pitch motion of the floating platform. It is hence used to ensure the controller does not make the system unstable in that situation.

Wind time series are generated using NREL’s TurbSim software [14] and are in accordance with IEC 61400-1. For this study, wind time series have an IEC Class B turbulence level, $I_{ref}=0.14$. Where I_{ref} is the mean turbulence intensity at 15 m/s. Wave time series are created using NREL’s HydroDyn software. For this study, the significant wave heights and wave periods are given in Table 3. This sea state data matches the OC3-Hywind turbine reported by Jonkman [16]. The site for these data corresponds to a location off the coast of Scotland where the water depth is 160 m. This site has rather severe wave conditions with a 50y significant wave height above 13m. For its characteristics, and since wave heights are not depth-limited at such depths, it was selected by Jonkman for the 320 m water depth conditions of the OC3-Hywind.

Table 3: Sea state variables for the load cases

Wind speed (m/s)	Significant wave height (m)	Wave period (s)
13.4	2.88	12.26
15.4	3.38	13.38
19.4	4.39	14.27
23.4	5.43	15.68

5.2. Simulation of Additional Load Cases

As mentioned in the previous section, the set of load cases selected is set to a minimum in order to reduce the total run-time of the simulations. However, additional wind and wave seeds shall be run for each load case during the evaluation in order to reduce uncertainty in loads associated with using a

limited number of wind and wave seeds. Participants are encouraged to generate and run more load cases if they want to reduce the uncertainty of their results. Simulating with additional seeds is not required for submission but is reserved for advanced users and those with additional resources.

To generate more load cases, find the script “Main_GenerateCases.m” located in the folder “_Inputs”. The script will generate cases based on the file “_Inputs_inputs\CasesExtended.csv” which correspond to the cases given in Table 2. This file can be edited directly to add more wave seeds or wind seeds. The script “Main_GenerateCases.m” will use this .csv file. Running the script will generate all the input files for OpenFAST, including the waves and wind files. By default, the script generates the cases in a folder called “LoadCases_Offshore” or “LoadCases_Onshore”. The user can run these new cases by adapting the script “Main.m” to point to the proper case file and the proper load case folder.

5.3. Controller Performance Metrics and Scoring

5.3.1. Background

According to the evaluation rules (R19 to R24) of the competition, the submissions of both Challenges (Onshore and Offshore) will be automatically scored with a cost function (CF) that calculates the mechanical fatigue and ultimate loads on blades, tower and other main turbine components, as well as the annual energy production. The script that calculates the cost function is provided with the user package and need not be implemented by the participants.

The cost function returns a scalar value that should be minimized (i.e. a lower cost equates to a better design). Each controller submission will be evaluated using the same cost function and will be compared to a baseline. The winning submission will be the one that most reduces that CF across a set of input cases that include wind cases in the *Onshore Challenge* and also wave cases in the *Offshore Challenge* (Set B).

The goal is for a submission to score lower than the NREL 5MW baseline controller (lower scores indicate better control performance relative to the baseline). The cost function is normalized so that the NREL 5MW baseline controller gives a performance metric of one (=1) in both Challenges. Only the participants that score lower than this NREL 5MW baseline controller (<1) will be eligible for the awards described in Rules R4 and R7.

This section presents the cost function used in this contest and the rationale behind it. A Matlab function is provided to compute the cost function automatically using the proper weights and scaling so that the participants do not need to be concerned about the implementation. For additional information, see Sections 5.3.2 and 5.3.3.

5.3.2. Cost Function Definition

The cost function is defined according to the following principles:

- It represents the economic cost of the main components, i.e., a function of CapEx.

- The components ($i = 1$ to 4) for the *Onshore Challenge* are: 1. Rotor (blades), 2. Hub (bearing and pitch systems), 3. Nacelle (drive-train, gearbox, generator, yaw), and 4. Tower.
- The components ($i = 1$ to 5) for the *Offshore Challenge* are the same as for the *Onshore Challenge* but with the addition of 5. Floating Platform.
- Each component (i) has a weight factor $[\alpha_i]$ related to the relative economic cost of the component, with $\sum \alpha_i = 1$, and with $i = 1$ to 4 in the *Onshore Challenge* and $i = 1$ to 5 in the *Offshore Challenge*. The weights are provided in Table 4. The values of the weights (α_i) are based on the report: [ATLANTIS RFI/ARPA-E](#), December 2018.
- The cost function is a function of the mechanical loads (fatigue and ultimate load) in both challenges but also includes the movements of the floating platform in the *Offshore Challenge*.
- Each component (i) is represented by a function (f_i). This function is a linear combination of ratios (R_{ik}) at several frequencies of interest (ω_k). The ratios are defined as the amplitudes of the Fast Fourier Transform (FFT) of the temporal simulation of mechanical loads (or movements in case of the floating platform), being the numerator of the FFT amplitude with the participant controller (" $L_{p_i_k}$ ") and the denominator the FFT amplitude with the baseline controller (" $L_{ref_i_k}$ ") at the frequency ω_k , and for each component i .
- The set of all frequencies considered are: $[\omega_k] = [\omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6, \omega_7, \omega_8, \omega_9, \omega_{10}]$. Table 5 gives a description of these frequencies and shows their values.

Table 4: Weight factors for the Onshore and Offshore Challenges

Onshore	i = 1 to 4	
Rotor (blades)	α_1	0.31
Hub (with bearings and pitch systems)	α_2	0.03
Nacelle (with drive train, gearbox, generator, yaw, etc)	α_3	0.34
Tower and foundation	α_4	0.32

Offshore	i = 1 to 5	
Rotor (blades)	α_1	0.11
Hub (with bearings and pitch systems)	α_2	0.02
Nacelle (with drive train, gearbox, generator, yaw, etc)	α_3	0.11
Tower	α_4	0.11
Floating platform, mooring system and anchor system	α_5	0.65

Table 5: Key natural frequencies considered in the cost function and scale coefficients

Frequency	Description	Natural Frequency (Hz.)	Scale coefficient n_k
ω_1	floating platform pitch	0.036	0.375
ω_2	Once-per-revolution [1P]	0.20	1.0
ω_3	tower_fore-aft_1st_mode	0.38-onshore; 0.50-offshore	1.0
ω_4	2P	0.40	1.0
ω_5	3P	0.60	1.0
ω_6	blade edgewise regressive	0.89-onshore; 0.9489-offshore	0.7
ω_7	blade edgewise collective	1.10-onshore; 1.10-offshore	0.5
ω_8	blade edgewise progressive	1.3130-onshore; 1.3130-offshore	0.4
ω_9	drive-train torsion	1.6970-onshore; 1.7271-offshore	0.2
ω_{10}	k_ultimate load	n/a	$0.25 \sum_{k \in K_i \setminus \{10\}} n_k$

The frequencies included in the cost function for each component are in general a subset of the complete set of frequencies listed above. Table 6 lists these components and the frequencies included in their cost function. For a given component i , the subset of the frequency index that are retained is noted K_i . For example, for the rotor, $K_1 = [2,4,7,10]$.

Table 6: Frequencies in the cost function by component. "x" indicates the frequency is included in the cost calculation of that component

Component	Signal	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6	ω_7	ω_8	ω_9	ω_{10}
Rotor (i=1)	Bld Root My (flap)		x		x			x			x
Hub (i=2)	Bld Root Mz (torsion)		x		x			x			x
Nacelle (i=3)	LSS torque					x				x	x
Tower (i=4)	Tower Bottom My			x		x	x		x		x
Platform (i=5)	Platform pitch motion	x		x							x

The cost function has been normalized, so that a value of “1” corresponds to the baseline controller performance.

The baseline controller is denoted as “ref”, and the participant controller as “p”.

The cost function also penalizes the reduction of AEP and rewards the increase of AEP: $\left(\frac{AEP_{ref}}{AEP_p}\right)$

The expression of the cost function is:

$$CF_p = \left(\frac{AEP_{ref}}{AEP_p}\right) \sum_{i=1}^{n \text{ components}} (\alpha_i f_i) \quad (3)$$

with,

$$f_i = \frac{\sum_{k \in K_i} (R_{ik} n_k)}{\sum_{k \in K_i} n_k} \quad (4)$$

$$R_{ik} = \frac{L_{p_{i_k}}}{L_{ref_{i_k}}} \quad (5)$$

K_i is the set frequency index that are retained for a given component, according to Table 6. The scaling coefficients n_k are provided in Table 5. They are used to give extra focus on some frequencies. The values $L_{p_{i_k}}$ and $L_{ref_{i_k}}$ are metrics values computed based on the results from the participant controller and from the reference baseline controller respectively. The metrics are either computed from the FFT of a signal (mechanical loads or motion) or the maximum amplitude of the signal. Their definition as function of i and k are given below:

- For $i = 1$ to 4 and $k = 1$ to 9, $L_{p_{i_k}}$ is the *FFT amplitude* at the frequency ω_k of the *mechanical loads* supported by the component i as obtained with the participant controller.
- For $i = 1$ to 4 and $k = 10$, $L_{p_{i_{10}}}$ is the *maximum amplitude* of the mechanical load (ultimate load) supported by the component i ($i = 1$ to 4) as obtained with the participant controller.
- For $i = 5$, and $k = 1$ to 9 (in practice $k = 1, 3$), $L_{p_{5_k}}$ is the *FFT amplitude* at the frequency ω_k of the *movements* of the floating platform as obtained with the participant controller.
- For $i = 5$ and $k = 10$, $L_{p_{5_{10}}}$ is the *maximum amplitude* of the movement (ultimate movement) of the floating platform as obtained with the participant controller

Similar definitions are used to define $L_{ref_{i_k}}$ based on the results obtained with the reference controller.

In addition to the metrics described above, a set of constraints are introduced in the cost function. If any of the constraints are exceeded, the cost function will evaluate to a value above 1000 and hence disqualify the controller. The constraints are listed in Table 7. Their calculation is automatically included in the script provided.

In Table 7, the tower clearance is the minimum distance allowed between the tower and the deflected blade. The value is taken as 30% of the clearance when the blade is undeflected. The maximum tower top acceleration is taken as 30% of one-g, with the gravitational acceleration g taken as 9.813 m/s^2 . The maximum overspeed allowed for the rotor was taken as 30% of the rated rotational speed (12.1 rpm). The maximum pitching rate was taken as 10 deg/s. The rotational speed and pitch rate constraints have been relaxed compared to the values that would be expected for a real wind turbine design to ease the task of the participants.

Table 7: Cost function constraints (maximum value allowed)

Signal	Constraint value
Tower clearance	4 m
Tower top acceleration	3.3 m/s^2
Rotational speed of the rotor	15.73 rpm
Blade 1 pitch speed	10 deg/s
Blade 2 pitch speed	10 deg/s
Blade3 pitch speed	10 deg/s

5.3.3. Matlab Function to Evaluate the Cost Function

The cost function is evaluated using the Matlab function “fCostFunctionFolders”, which takes as main input a list of folders. The first folder is the one where the baseline results are found (“_BaselineResults”), and the second one will be the one where the contestants results are located. More folders can be provided if several designs are to be compared. This function loops through the folders, compute some basic statistics for each load cases (fComputeOutStats.m), then evaluate the metric values (fEvaluateMetrics.m) and then computes the cost function (fCostFunction.m). The result main result is a vector of cost function values, one for each of the folders given as input to the function “fCostFunctionFolders”. Additionally, the contribution from each component and metrics are also returned by this function. The participants are not allowed to modify any of these functions.

6. Package Download, File Extraction, and Simulation Demonstration

A complete package of files for modeling and simulating the NREL 5 MW turbine (for either *Onshore* or *Offshore Challenges*) in Simulink will be provided to contest participants. Participants will be provided with a single Matlab script (“Main.m”) that does the following: initializes the Simlink simulation, simulates each of the supplied load cases, calculates controller performance metrics, and calculates the final controller scalar cost function value for scoring. A short demonstration of this process is provided below.



Advanced Research Projects Agency • ENERGY

Competition participants will download the zip file named “ATLAS-On(Off)shore-Challenge.zip” from the competition website. For the On(Off)shore Challenge, this zip file contains only the folders and files pertinent to the On(Off)shore Challenge. Extracting the contents of this file will produce a folder named “ATLAS-On(Off)shore-Challenge” that contains several subfolders plus an m-file named “Main.m.” Figure 8 shows a screenshot of the folder structure of the download. These sub-folders are named “_BaselineResults”, “_Controller”, “_Functions”, and “_Inputs.” The folder named “_Outputs” is created during the simulation and hold the simulation output results. The Matlab script file “Main.m” runs the entire process of setting up and initializing the Simulink simulation, performing post-processing of results, and finally calculating turbine performance metrics and controller cost function (see Section 5 above).

The folder “_BaselineResults” contains output files for the various load cases from simulations using the NREL Baseline Controller Simulink model. The contents of this folder are already populated with the baseline controller results, which have already been simulated for use by the participants in judging their own controller results. The contents of this folder should not be modified.

The folder “_Controller” contains the Simulink models for the NREL Baseline controller (NREL5MW_Baseline.mdl) and the example controller in which the participants replace the NREL baseline pitch controller with their own controller (NREL5MW_Example_IPC.mdl). The IPC+CPC controller highlighted in Figure 7 corresponds to this model (NREL5MW_Example_IPC.mdl). In addition, the m-file that contains the controller parameters is located here (“fSetControllerParametersOn(Off)shore.m”). The participants should add their own controller parameters in this file, rather than creating a new one. The contents of this folder may be modified. This is the folder that needs to be uploaded by the participants.

The “_Functions” folder contains all of the files used in the script that performs post-processing, calculates key turbine metrics, and finally calculates the controller performance metric (cost function). It also contains the sub-folder named “OpenFASTLibraries” that contain the DLLs and mex-function needed to run Simulink. These executable files correspond to OpenFAST compiled for Windows 64-bit operating systems. The contents of this folder should not be modified.

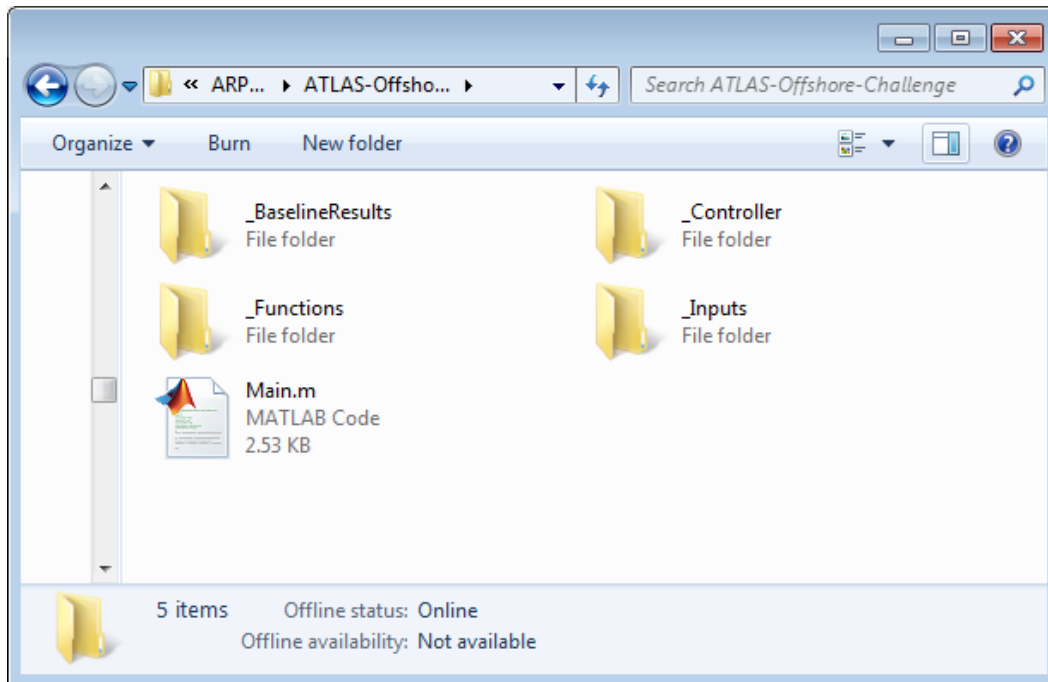


Figure 8: Folder structure of the download

The “_Inputs” folder contains all of the necessary OpenFAST and OpenFAST module input files needed for a particular challenge. These input files are separated out in terms of load case name and the particular module they represent (see Table 1). The contents folder should not be modified.

The folder named “_Outputs” collects the output files for the simulated load cases. This depends on the particular challenge selected for simulation as will be described below. The output files are named “x.SFunc.outb”, where the load case name is substituted for the symbol “x” in the file name. The outputs files can be visualized using the tool pyDatview [15] or using the matlab function “_Functions\ReadFASTBinary.m”. The folder “_Outputs” is automatically updated with the participant output files.

To begin the simulation, metrics calculation, and controller scoring process, the participants should open Matlab and point it to the folder named “ATLAS-On(Off)shore-Challenge.” This folder contains the Matlab script named “Main.m “ that runs the entire process. To begin the process, the user types in “Main” at the Matlab command line. This initiates simulation of the full set of load cases for a particular challenge as well as the performance and cost function calculations.

After invoking this command, Matlab will begin the load suite simulation. An example Matlab screenshot is shown below in Figure 9.


```

Command Window
New to MATLAB? See resources for Getting Started.

-----
>>> 1/12 - Simulating: _Inputs/LoadCases//DLC120_ws13_ye000_s1_r1.fst
-----
Simulation params: T=600 - pitch_0=7.2

*****
OpenFAST

Copyright (C) National Renewable Energy Laboratory
Copyright (C) Envision Energy USA LTD

This program is licensed under Apache License Version 2.0 and comes with ABSOLUTELY NO WARRANTY.
See the "LICENSE" file distributed with this software for details.
*****

OpenFAST-108121b2-dirty
Compile Info:
- Architecture: 64 bit
- Precision: single
- Date: Nov 01 2018
- Time: 15:29:23
Execution Info:
- Date: 01/10/2019
- Time: 16:13:20-0700

OpenFAST input file heading:
FAST Certification Test #24: NREL 5.0 MW Baseline Wind Turbine with OCS Hywind Configuration,
for use in offshore analysis

Running ElastoDyn.
Running AeroDyn.
Running AirfoilInfo.
Running BEM.
    
```

Figure 9: Matlab simulation initiation

The full set of 12 load cases will be simulated. At the end of the simulation, the controller performance metrics and cost function will be calculated. The screenshot below (Figure 10) shows plots produced by Matlab at the conclusion of the simulation. The first time that the users will run the Main from the user package, the cost function returned should be 1 for the baseline controller and 1 for the participant controller, since by default the participant controller is the same as the baseline. In the example below, the participant controller was modified and the cost function was evaluated to 0.9.

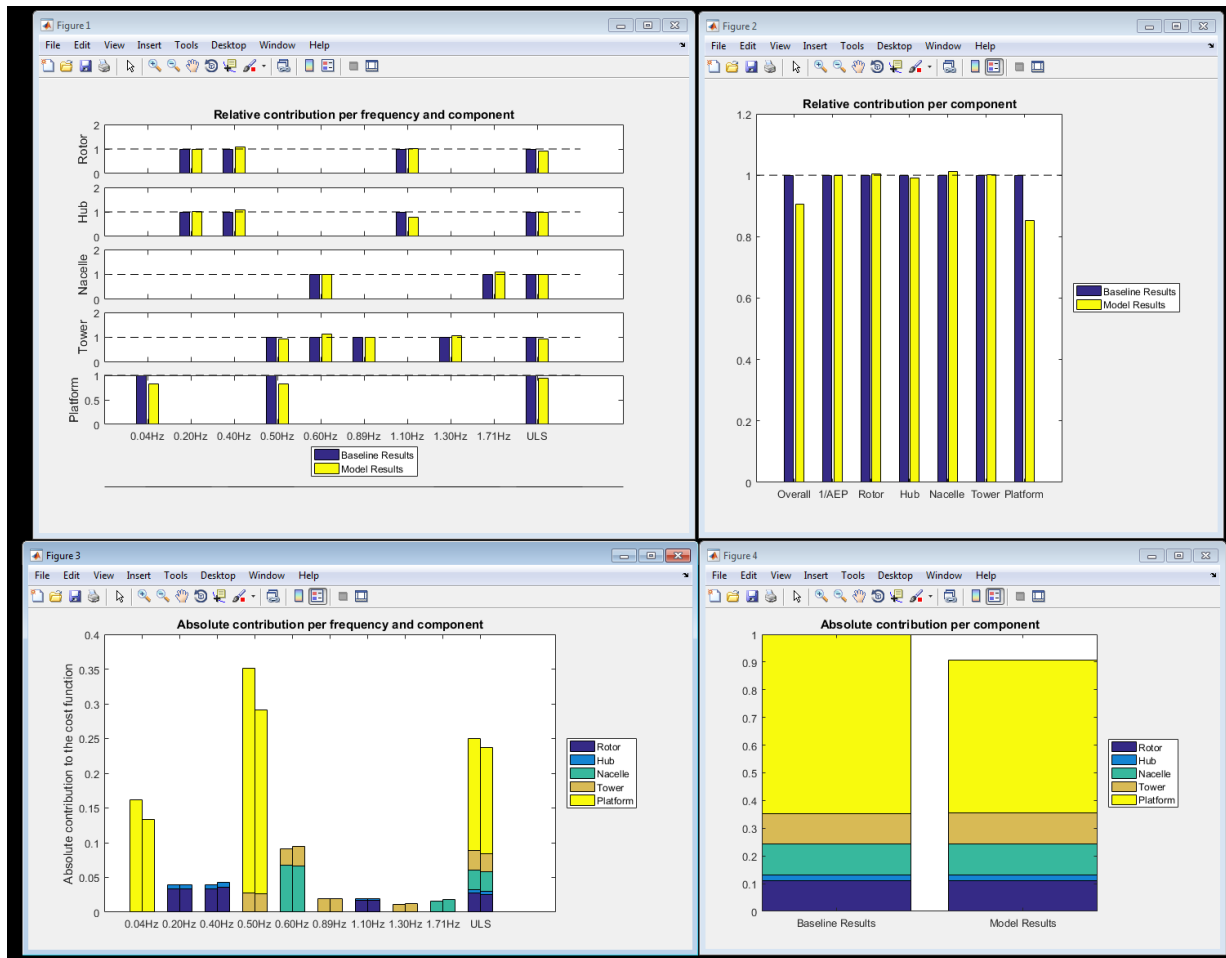


Figure 10: Matlab output plots

The different output plots are described below:

- Figure 10-1: shows the relative contribution to the cost function by each frequency and component. This corresponds to the values R_{ik} of the cost function. The values for the baseline controller and the participant controller are shown side by side. A value above one indicates that the controller performs worse than the baseline controller for this given frequency and component. The values of the baseline controller evaluate to 1 from the definition of the variable R_{ik} . The last bar labelled “ULS”, corresponds to ω_{10} , the maximum value of the signal considered.
- Figure 10-2: shows the relative contribution of each turbine component (Annual Energy Production [AEP], rotor, hub, nacelle, tower, and platform) to the cost function. The label “Overall” represents the total cost function. The label “1/AEP” is the ratio $\frac{AEP_{ref}}{AEP_p}$. For each component, the quantity plotted is the ratio of $\alpha_i f_i$ evaluated with the participant’s controller

divided by the same quantity evaluated with the baseline controller. The baseline results (blue bars) of each component are always a value of 1.

- Figure 10-3: shows the absolute contribution per frequency and component. This corresponds to the quantity $\alpha_i \frac{R_{ik}n_k}{\sum_{k \in K_i} n_k}$. The sum of all these quantities gives the total cost function. The figure shows for each frequency the results from the baseline and participant controller in two bars side by side. The contribution from each component are stacked with different colors in each bar. The last bar labelled “ULS”, corresponds to ω_{10} . By design, the contribution of the ULS loads is equal to $\frac{1}{4}$ of the total contribution for a given component
- Figure 10-4: shows the absolute contribution of the turbine components to the cost function. For the baseline results the sum of these contributions is 1. The results are presented relative to the Baseline results to show where the contestants’ controller performs better or worse than the baseline controller. A value below 1 indicates that the participant’s controller performs better than the baseline controller. In the example above, the cost function from the contestant result is about 0.90.

For more details of these metrics and cost function, please see Section 5 above.

The location of the various folders should not be changed by the participants. Automation of this process using “Main.m” depends the location of the folders within the main folder “ATLAS-On(Off)shore-Challenge.”.

7. Participants upload

Participants will be requested to submit their controller files to the website. Additional information will be provided on the website concerning the upload. Some guidelines for the upload are presented here. Participants should put all the files necessary to run the simulations with their controller into the folder “_Controller” of the user package. Before the submission, this folder should contain:

- One Simulink model file, which name should be of the form “*.mdl”
- One Matlab function that sets up the parameters of the participants controller, which name should be “fSetControllerParameters*.m”
- Additional files and potential sub-folder required for the participants controller to run.

The folder “_Controller” should not contain:

- More than one model of the form “*.mdl”
- More than one function of the form “fSetControllerParameters*.m”
- Any of the files and folders present in the original-package



Advanced Research Projects Agency • ENERGY

Once the folder “_Controller” satisfies these conditions the folder need to be compressed into a zip-archive and uploaded to the website.

References

1. Jonkman, J. , Butterfield, C., Musial, W., and Scott, G. “Definition of a 5-MW Reference Wind Turbine for Offshore System Development,” Golden, CO: National Renewable Energy Laboratory, NREL/TP-500-38060, 2009. <https://www.nrel.gov/docs/fy09osti/38060.pdf>
2. Jonkman, J. “Definition of the Floating System for Phase IV of OC3.” Golden, CO: National Renewable Energy Laboratory, NREL/TP-500-47535, 2010. <https://www.nrel.gov/docs/fy10osti/47535.pdf>
3. Bossanyi, E., “The Design of Closed-loop Controllers for Wind Turbines.” *Wind Energy*. 2000; 3:149-163 (DOI: 10.1002/we.34).
4. Wright, A. and L. Fingersh, “Advanced Control Design for Wind Turbines. Part I: Control design, implementation, and initial tests. NREL/TP-500-42437. 2008. <https://www.nrel.gov/docs/fy08osti/42437.pdf>
5. Johnson, K., Pao, L., Balas, M., and Fingersh, L. “Control of Variable-Speed Wind Turbines.” *IEEE Control Systems Magazine*, Volume 26, Issue 3, pages 70-81, June 2006.
6. Garcia-Sanz, M., and Houpis, C. (2012). *Wind Energy Systems: Control Engineering Design*. CRC Press, Boca Raton, Florida. ISBN: 978-1-4398-2179-4-CAT# K11158.
7. OpenFAST <https://openfast.readthedocs.io/en/master>
8. Mathworks, The MATLAB/Simulink design tool. <http://www.mathworks.se/products/simulink/>, 2014.
9. Jonkman, B., and Jonkman. README_FAST8.pdf: July 2016. https://wind.nrel.gov/nwtc/docs/README_FAST8.pdf
10. Jonkman, J. M. and Buhl, M. L., “FAST User’s Guide,” Golden, CO: National Renewable Energy Laboratory, NREL/EL-500-38230, 2005. <https://nwtc.nrel.gov/system/files/FAST.pdf>
11. Bir, G. “Multiblade Coordinate Transformation and Its Application to Wind Turbine Analysis.” NREL/CP-500-42553. January 2008. <https://www.nrel.gov/docs/fy08osti/42553.pdf>
12. Wheeler L. and Garcia-Sanz M. (2017). “*Wind turbine collective and individual pitch control using quantitative feedback theory*”. ASME 2017 Dynamic Systems and Control Conference, Tysons Corner, Virginia, USA.



Advanced Research Projects Agency • ENERGY

13. T. WG1, *IEC 61400-1 wind turbines: Design requirements*. 2005.
14. Jonkman, B., and Kilcher, L. "TurbSim User's Guide: Version 1.06.00." (draft) 2012.
<https://nwtc.nrel.gov/system/files/TurbSim.pdf>
15. E. Branlard - pyDatView - <https://github.com/ebranlard/pyDatView/releases>
16. Jonkman, J.M. Dynamics Modeling and Loads Analysis of an Offshore Floating Wind Turbine, Technical Report NREL/TP-500-41958, November 2007.