



Simulation for Wind Turbine Generators—With FAST and MATLAB-Simulink Modules

M. Singh, E. Muljadi, J. Jonkman,
and V. Gevorgian
National Renewable Energy Laboratory

I. Girsang and J. Dhupia
Nanyang Technological University

**NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC**

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Technical Report
NREL/TP-5D00-59195
April 2014

Contract No. DE-AC36-08GO28308

Simulation for Wind Turbine Generators—With FAST and MATLAB-Simulink Modules

M. Singh, E. Muljadi, J. Jonkman,
and V. Gevorgian

National Renewable Energy Laboratory

I. Girsang and J. Dhupia

Nanyang Technological University

Prepared under Task Nos. WE11.0336 and WE14.3F01

**NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC**

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

NOTICE

This report was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or any agency thereof.

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Available electronically at <http://www.osti.gov/scitech>

Available for a processing fee to U.S. Department of Energy and its contractors, in paper, from:

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
phone: 865.576.8401
fax: 865.576.5728
email: <mailto:reports@adonis.osti.gov>

Available for sale to the public, in paper, from:

U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
phone: 800.553.6847
fax: 703.605.6900
email: orders@ntis.fedworld.gov
online ordering: <http://www.ntis.gov/help/ordermethods.aspx>

Cover Photos: (left to right) photo by Pat Corkery, NREL 16416, photo from SunEdison, NREL 17423, photo by Pat Corkery, NREL 16560, photo by Dennis Schroeder, NREL 17613, photo by Dean Armstrong, NREL 17436, photo by Pat Corkery, NREL 17721.



Printed on paper containing at least 50% wastepaper, including 10% post consumer waste.

Acknowledgments

This work was supported by the Energy Research Institute at Nanyang Technological University, Singapore. The authors would like to thank Dr. Khanh Nguyen for fruitful discussions about integrating the National Renewable Energy Laboratory's Fatigue, Aerodynamics, Structures, and Turbulence model with the external drivetrain model, especially as related to Section 6.

List of Acronyms

CAE	computer-aided engineering
DFIG	doubly-fed induction generator
FAST	Fatigue, Aerodynamics, Structures, and Turbulence model
GRC	Gearbox Research Collaborative
HSS	high-speed shaft
IGBT	insulated-gate bipolar transistor
LSS	low-speed shaft
MATLAB	Matrix Laboratory
NREL	National Renewable Energy Laboratory
SDC	stress damper controller
VIDC	virtual inertia and damping control
WRIG	wound-rotor induction generator
WTG	wind turbine generator

Abstract

This report presents the work done to develop generator and gearbox models in the Matrix Laboratory (MATLAB) environment and couple them to the National Renewable Energy Laboratory's Fatigue, Aerodynamics, Structures, and Turbulence (FAST) program. The goal of this project was to interface the superior aerodynamic and mechanical models of FAST to the excellent electrical generator models found in various Simulink libraries and applications. The scope was limited to Type 1, Type 2, and Type 3 generators and fairly basic gear-train models. The final product of this work was a set of coupled FAST and MATLAB drivetrain models. Future work will include models of Type 4 generators and more-advanced gear-train models with increased degrees of freedom. As described in this study, the developed drivetrain model can be used in many ways. First, the model can be simulated under different wind and grid conditions to yield further insight into the drivetrain dynamics in terms of predicting possible resonant excitations. Second, the tool can be used to simulate and understand transient loads and their couplings across the drivetrain components. Third, the model can be used to design the various flexible components of the drivetrain such that transmitted loads on the gearbox can be reduced. Several case studies are presented as examples of the many types of studies that can be performed using this tool.

Table of Contents

1	Introduction.....	1
2	FAST Description	3
3	Interfacing FAST and MATLAB/Simulink.....	5
3.1	Step-by-Step Preparation.....	5
3.2	FAST Files and Data Entry	9
4	Wind Turbine Modeling.....	11
4.1	Type 1 Wind Turbine Model	14
4.1.1	Preexisting FAST Type 1 Turbine Models (Steady-State Model)	14
4.1.2	Dynamic Induction Machine Model	20
4.1.3	Addition of Pitch Controller	28
4.2	Type 2 Wind Turbine Model	34
4.2.1	Rotor Resistance Control Concept.....	35
4.2.2	Implementation.....	36
4.2.3	Type 2 Turbine Model Results	41
4.3	Type 3 Turbine Model—SimPowerSystems	43
4.4	Type 4 Turbine Model—SimPowerSystems	45
5	Simulation of Normal and Abnormal Events.....	48
5.1	Normal and Abnormal Events	48
5.2	Electrical Abnormal Events.....	51
5.2.1	Grid-Related Events	51
5.2.2	Generator and Power Converter–Related Events	51
5.3	Mechanical and Aerodynamic Abnormal Events	52
5.4	Wind Turbine Requirements	52
5.4.1	Grid Interface Requirement	52
5.4.2	Electrical Component Requirement	54
5.4.3	Energy-Harvesting Requirement	54
5.4.4	Mechanical Component Requirement.....	55
5.5	Designing Controls to Mitigate Impacts	55
5.5.1	Mechanical Linkage	56
5.5.2	Electrical Linkage.....	56
6	Case Studies.....	58
6.1	Example Case 1: Grid – Turbine Interaction	58
6.1.1	Type 1 WTG.....	58
6.1.2	Type 3 WTG.....	59
6.2	Example Case 2: Impact on Mechanical Linkages	62
6.2.1	Drivetrain Modeling	63
6.2.2	Model Integration	69
6.3	Example Case 3: Virtual Inertia and Damping Controller	74
6.3.1	Control Formulation	75
7	Conclusion.....	80
8	References	81
9	Bibliography.....	84
10	Appendices	85
10.1	Appendix A: Wind Turbine Drivetrain Modeling in Simscape/ SimDriveline and Interfacing with FAST in Simulink— <i>User’s Guide</i>	85
10.1.1	Introduction	85
10.1.2	Simscape and SimDriveline Basics	85
10.1.3	Torsional Model of the Gear Box	94
10.1.4	Integration with FAST Code	99
10.1.5	Simulation Checklist	104

10.2 Appendix B: Integrating FAST with a Type 3 Wind Turbine Generator Model.....	105
10.2.1 Introduction	105
10.2.2 Bypassing the SimPowerSystems Turbine Model	106
10.2.3 The DFIG Model.....	108
10.2.4 Complementing the DFIG Model	114
10.2.5 Integrating the DFIG Model	120
10.2.6 Running the Simulations	123

List of Figures

Figure 1. Hybrid simulation performed with a detailed aerodynamic and structural model of a wind turbine within FAST, and a detailed electrical and grid model in the MATLAB/Simulink environment	2
Figure 2. (a) Layout of a conventional, downwind, two-bladed turbine and (b) a close-up of its hub.....	3
Figure 3. Simulink model <i>OpenLoop.mdl</i>	6
Figure 4. Inside the wind turbine block of the FAST wind turbine model in Simulink.....	6
Figure 5. Error diagnostic while running <i>Test01_SIG.mdl</i>	8
Figure 6. <i>Test01_SIG.mdl</i> in Simulink.....	8
Figure 7. Modified model with Switch, Constant, and Add blocks.....	9
Figure 8. Fixed-speed wind turbine schematic	12
Figure 9. Variable-slip wind turbine schematic.....	12
Figure 10. DFIG wind turbine schematic	13
Figure 11. Full-converter wind turbine schematic.....	13
Figure 12. Subsystems for a Type 1 turbine model.....	14
Figure 13. Example of an induction machine torque curve	15
Figure 14. Generation of a torque-slip curve from parameters supplied in Table 1	16
Figure 15. Induction machine single-phase equivalent circuit	17
Figure 16. Torque calculation from speed, implemented in Simulink.....	18
Figure 17. Example of a MATLAB scope output during run time	19
Figure 18. Torque, speed, and output power from <i>Test01_SIG.mdl</i> with a step change in the wind speed	20
Figure 19. Induction machine model <i>S1.mdl</i>	21
Figure 20. Modified <i>Test01_SIG.mdl</i> showing the Simple Induction Generator block replaced with the Induction Machine Model block.....	22
Figure 21. Blocks for the subsystem (Gain, Product, Terminator, and Mux)	23
Figure 23. Change in CP curves with change in pitch angle (beta)	29
Figure 24. Pitch control inputs to FAST block and dummy pitch controller	30
Figure 25. Contents of the pitch controller block before and after modification	30
Figure 26. PI controller settings for the pitch controller.....	31
Figure 27. Connections in the main Simulink window	32
Figure 28. Results with pitch controller enabled	32
Figure 29. Pitch controller output	33
Figure 30. Type 1 turbine SimPowerSystems model	34
Figure 31. Power converter for external resistance control in variable-slip turbines	35
Figure 32. Induction machine equivalent circuit with external resistor present.....	36
Figure 33. Example torque-speed curves with different values of external rotor resistance R_{ext} (expressed per unit of internal rotor resistance R_2)	36
Figure 34. Original contents of “Daxis” subsystem	37
Figure 35. Modified contents of “Daxis” subsystem	37
Figure 36. Original contents of “Qaxis” subsystem	38
Figure 37. Modified contents of “Qaxis” subsystem	38
Figure 38. Modified “Induction Machine Model” subsystem with new “RotorResCtrl” subsystem	39
Figure 39. Blocks and connections within the “RotorResCtrl” subsystem	40
Figure 40. Gains and limits for power PI controller in “RotorResCtrl” subsystem	40
Figure 41. Rotor resistance variations.....	41
Figure 42. Pitch angle variations with rotor resistance controller present and enabled	41
Figure 43. Results with pitch and rotor resistance controller present	42
Figure 44. Type 2 wind turbine model using SimPowerSystems	43
Figure 45. Type 3 wind turbine connection diagram.....	44
Figure 46. Type 3 wind turbine model using SimPowerSystems	45

Figure 47. Type 4 wind turbine connection diagram.....	46
Figure 48. Type 4 turbine model using SimPowerSystems.....	46
Figure 49. Type 4 turbine model using SimPowerSystems—detailed view of the electrical topology.....	47
Figure 50. Layout of a typical wind power plant	48
Figure 51. A simplified power system configuration often used in simulating fault ride-through capability of a turbine	51
Figure 52. A simplified diagram showing various linkages and the power flow in a wind power plant ...	55
Figure 53. Simulation results showing the impact of tower shadow on the generator torque and speed for a Type 1 WTG	58
Figure 54. Simulation results showing the impact of tower shadow on the generator thrust load and output power for a Type 1 WTG	59
Figure 55. (a) Physical diagram and (b) power versus speed characteristic of a Type 3 WTG	59
Figure 56. Simulation results showing the impact of a single-phase voltage sag for a Type 3 WTG on (a) high-speed shaft torque, speed, and power and (b) edgewise and flapwise blade moments at the blade root.....	60
Figure 57. Simulation results showing the impact of a three-phase voltage sag on the output power of a Type 3 WTG on high-speed shaft torque, speed, and power, as well as on edgewise and flapwise blade moments at the blade root.....	61
Figure 58. Modular drivetrain configuration of a wind turbine.....	64
Figure 59. Five-mass model of a wind turbine drivetrain in SimDriveline with a fixed-speed induction generator	64
Figure 60. Two-mass model of a wind turbine drivetrain	65
Figure 61. Parallel gear stage and gear mesh stiffness representations	66
Figure 62. Planetary gear set with three planet gears	67
Figure 63. Torsional model of a planetary gear stage with M planet gears	68
Figure 64. Frequency response function of a three-planet planetary gear stage for the gearbox presented in [16].....	68
Figure 65. Proposed schematic of integrating Simscape drivetrain model into the FAST aeroelastic CAE tool	70
Figure 66. Rotor speed response	71
Figure 67. Transient response comparison of the rotor torque comparison	71
Figure 68. Steady-state response comparison of the rotor torque	72
Figure 69. Electromagnetic torque excitations caused by a voltage drop on the grid	72
Figure 70. Transmitted loads onto the gears caused by grid excitation in (a) time domain and (b) frequency domain.....	73
Figure 71. Loads on pinion of second parallel gear stage under various high-speed shaft stiffness values	74
Figure 72. Schematic of SDC	74
Figure 73. VIDC for drivetrain resonance prevention	75
Figure 74. Implementation of the VIDC	76
Figure 75. Sinusoidal wind speed input at the drivetrain eigenfrequency	76
Figure 76. Wind turbine drivetrain response in terms of speed and power	77
Figure 77. Drivetrain angle of twist under resonant excitation.....	78
Figure 78. Power fluctuations under varying frequency excitations	78
Figure A.1. (a) Modular wind turbine drivetrain and (b) multistage representation of the drivetrain.....	88
Figure A.2. Five-inertia representation of a wind turbine drivetrain	89
Figure A.3. SimDriveline model of a wind turbine drivetrain.....	89
Figure A.4. Assigning parallel gear-teeth ratio to the wind turbine drivetrain	90
Figure A.5. Drivetrain model with torque inputs and measurement sensors	90
Figure A.6. Motion sensor to measure the rotational speed of the generator.....	91
Figure A.7. Torque sensor to measure the rotor-opposing torque	91
Figure A.8. Modifications to the Configuration Parameters for the drivetrain simulation	92

Figure A.9. Defining the input point for eigenfrequency analysis.....	93
Figure A.10. Defining the output point for eigenfrequency analysis.....	93
Figure A.11. Frequency response function of the drivetrain model	94
Figure A.12. Parallel gear stage and gear meshing stiffness representations.....	95
Figure A.13. Parallel gear stage representation in SimDriveline.....	95
Figure A.14. Planetary gear set with three planet gears	96
Figure A.15. One meshing set of a planetary gear set	96
Figure A.16. SimDriveline model of a planetary gear stage with three planet gears.....	97
Figure A.17. Frequency response function of a three-planet planetary gear stage	97
Figure A.18. Drivetrain model readily integrated with FAST.....	98
Figure A.19. Torque and speed sensor at the low-speed shaft.....	98
Figure A.20. Two-mass drivetrain model in FAST.....	99
Figure A.21. Blade layout in FAST	100
Figure A.22. Proposed schematic of integrating the drivetrain model with FAST	101
Figure A.23. Implementation of an integrated wind turbine drivetrain model	102
Figure A.24. Original FAST S-Function block as <i>OpenLoop.mdl</i>	102
Figure A.25. FAST Output Manager block.....	102
Figure A.26. Fcn block parameters to isolate the rotor acceleration	103
Figure A.27. Generator torque law for variable-speed variation	104
Figure B.1. A DFIG (a) with a turbine model and (b) without a turbine model	105
Figure B.2. (a) Option to include the turbine model and (b) option to bypass the turbine model	106
Figure B.3. Default power-speed characteristic of a turbine	107
Figure B.4. The original and recommended turbine power-speed characteristic.....	108
Figure B.5. Type 3 wind turbine connection diagram	108
Figure B.6. Select the “Look Under Mask” to reveal the model components	110
Figure B.7. Internal components of the SimPowerSystems DFIG model	110
Figure B.8. Internal component of the Generator & Converter block	111
Figure B.9. Internal component of the Asynchronous Machine block	111
Figure B.10. Internal components of the DFIG Mechanical block.....	112
Figure B.11. The mechanical equation of motion of the generator inertia	112
Figure B.12. Dialog window to disable the DFIG link.....	112
Figure B.13. The modified Mechanical block to take the speed input	113
Figure B.14. Attempting to modify the input label from “Tm” to “w”	113
Figure B.15. Generator model readily integrated with FAST	114
Figure B.16. Modifying the generator parameters	114
Figure B.17. The original converter parameters.....	115
Figure B.18. The original control parameters	116
Figure B.19. Supplying the required line voltage.....	117
Figure B.20. Representing a pure resistor	118
Figure B.21. Selecting the generator torque and output power	119
Figure B.22. Multiplying the rated torque to convert the unit from p.u. to Nm.....	119
Figure B.23. Implementation of an integrated wind turbine generator model	120
Figure B.24. Original FAST S-Function block as <i>OpenLoop.mdl</i>	120
Figure B.25. FAST Output Manager block.....	121
Figure B.26. Fcn block parameters to isolate the generator speed	121
Figure B.27. Changing the Powergui from “Continuous” to “Phasor”.....	122
Figure B.28. Modifying the “Configuration Parameters”	123
Figure B.29. Simulated generator response.....	125

List of Tables

Table 1. Parameters for the Simple Induction Generator Model (VSContrl = 0, GenModel = 1)	16
Table 2. Parameters for the Thevenin-Equivalent Model (VSContrl = 0, GenModel = 2)	17
Table 3. Gain Block Data.....	23
Table 4. Modeling Properties of GRC Wind Turbine	63
Table 5. Parameters of the 750-kW DFIG	63
Table 6. Eigenfrequencies of a Five-Mass Model.....	65
Table 7. Eigenfrequencies of Planetary Gear Stage	67
Table 8. Eigenfrequencies of Drivetrain with Torsional Gearbox Model.....	69
Table A.1. Properties of a Planetary Gear Set.....	96
Table A.2. Comparison of Eigenfrequencies of Planetary Gear Sets	98

1 Introduction

Wind energy deployment has experienced substantial growth in recent decades. In the past, wind turbine generators (WTGs) utilized a very simple wind turbine with stall control and a fixed-speed directly-connected induction generator (Type 1). Affordable power converters, advances in modern control, and the study of aeroelasticity (the interactions between inertial, elastic, and aerodynamic forces that occur when an elastic body is exposed to a fluid flow), and the availability of fast-computing microprocessors enabled wind turbine engineers to design very sophisticated, modern WTGs capable of delivering high-quality output power while at the same time enhancing power system operations.

To perform a holistic design, all aspects of a WTG need to be considered. In this work, we attempted to demonstrate a holistic WTG model by using the National Renewable Energy Laboratory's (NREL's) Fatigue, Aerodynamics, Structures, and Turbulence (FAST) [1] software to simulate the detailed aerodynamics and mechanical aspects of a WTG, and Matrix Laboratory (MATLAB)/Simulink [2] to simulate the electrical generators, converters, collector systems, and grid aspects of a grid-connected WTG. The references [1–9] cover the basic equations used in FAST as well as in drivetrain and generator models in more detail.

Mechanical load and stress in a wind turbine drivetrain are influenced by the torque and non-torque loads applied between the input shaft at the blade side and the output shaft at the generator side. For example, unwanted loads entering the input shaft are caused by wind turbulence, tower shadow effect, uneven loading of the blades, and sudden changes in the wind direction. Examples of unwanted disturbances on the output power (and thus output torque) include transmission line disturbances such as voltage and/or frequency dips, unbalanced voltage, and under- or overvoltage. The differences between input and output torque manifest in the stresses, loads, and losses of the components (e.g., gearbox, shaft, and bearing) in different parts of a WTG. Using FAST in conjunction with MATLAB/Simulink allows us to examine the loading of different components under grid transients and/or wind turbulence and also to design controllers to mitigate the effects of these unusual conditions on a turbine's structure and components. If the torque difference between input and output cannot be influenced, there is very little that can be done to influence the lifetime and operations and maintenance of these components, which eventually affects the cost of the energy during the lifetime of a WTG. Figure 1 illustrates the interfacing of FAST and MATLAB/Simulink.

FAST is very suitable to model and analyze wind turbines. A wind turbine is controlled by the electrical generator and its corresponding control. The impact of the control on its mechanical components can be observed because the modeled details of a wind turbine are represented in FAST. The generator representation in FAST is very simplified, whereas Simulink models for electrical generators are very well represented. However, MATLAB/Simulink models do not represent the aerodynamic and mechanical components of wind turbines very well.

The goal of this project was to interface the superior aerodynamic and mechanical models of FAST to the excellent electrical generator models found in various Simulink libraries and applications. The scope of the work was limited to Type 1, Type 2, and Type 3 generators and fairly basic gear-train models. Future work will include models of Type 4 generators and more-advanced gear-train models with increased degrees of freedom.

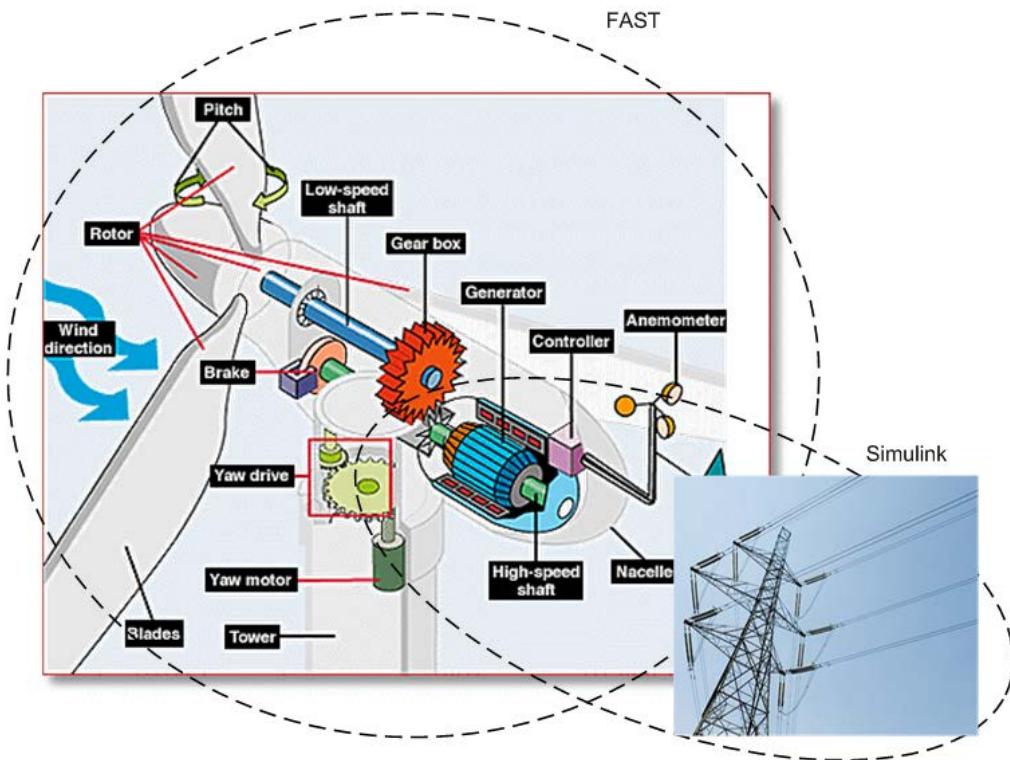


Figure 1. Hybrid simulation performed with a detailed aerodynamic and structural model of a wind turbine within FAST, and a detailed electrical and grid model in the MATLAB/Simulink environment

The goal of this project was also to improve FAST with better electrical models executable through a MATLAB/Simulink interface. To consider all of these models and shorten the development time, we took advantage of the Simulink library. Interfacing and rewriting code from scratch takes a lot of work, but using the available modules in the Simulink library is useful to build various models.

Because a wind turbine is a system comprised of many components with different functions, some of the figures are repeated several times throughout the following chapters and sections. This repetition is intended to enhance readability and prevent the reader from having to return to a previous description for reference and understanding.

2 FAST Description

NREL sponsored the development, verification, and validation of various computer-aided engineering (CAE) tools for the prediction of wind turbine loads and responses. A streamlined CAE tool called FAST was developed through a subcontract between NREL and Oregon State University. FAST is a comprehensive aeroelastic simulator capable of predicting both the extreme and fatigue loads of two- and three-bladed, horizontal-axis wind turbines.

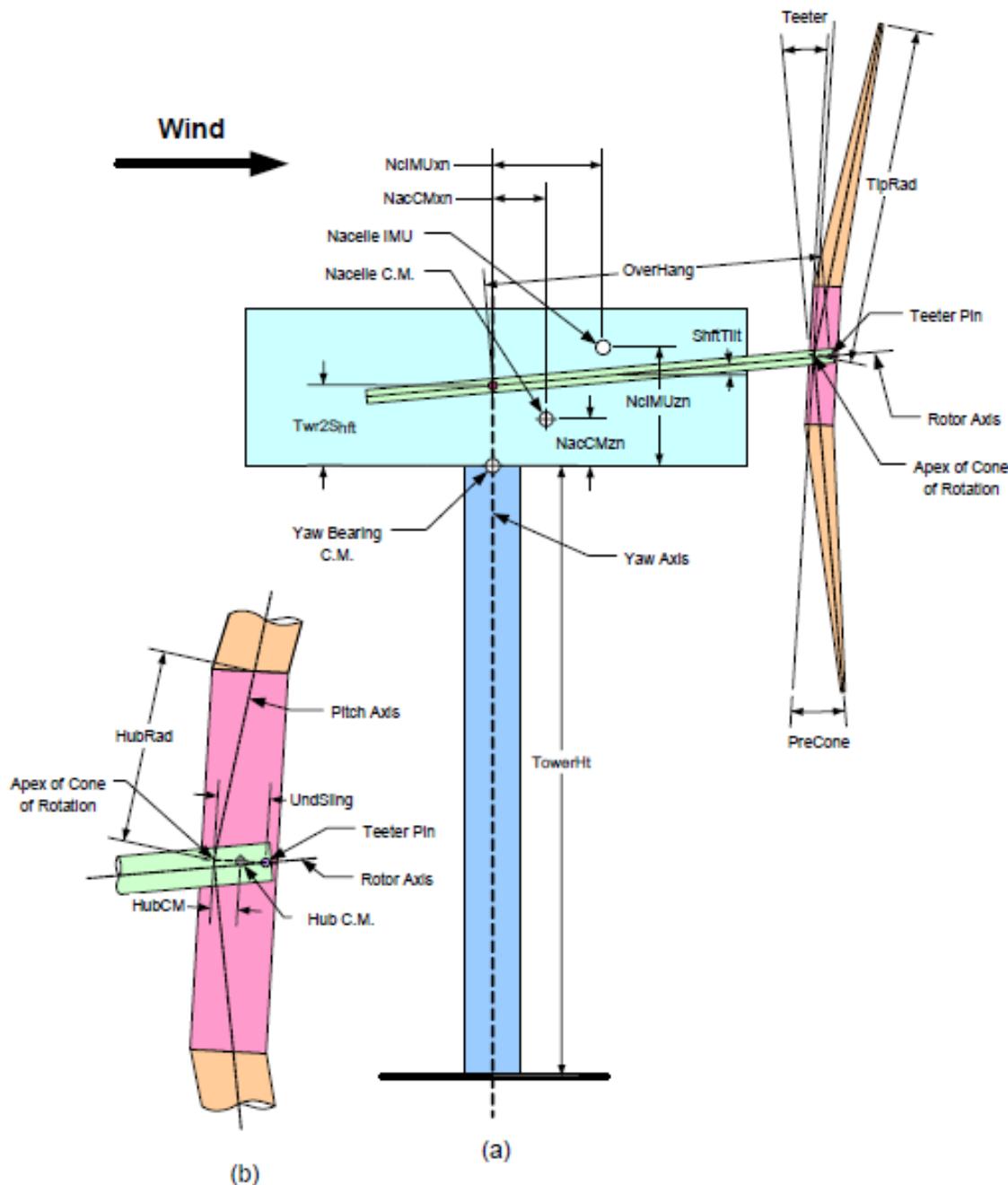


Figure 2. (a) Layout of a conventional, downwind, two-bladed turbine and (b) a close-up of its hub

The FAST code models a wind turbine as a combination of rigid and flexible bodies. For example, two-bladed, teetering-hub turbines are modeled as four rigid bodies and four flexible ones. The rigid bodies are the earth, nacelle, hub, and optional tip brakes (point masses). The flexible bodies include blades, tower, and drive shaft. The model connects these bodies with several degrees of freedom. These include tower bending, blade bending, nacelle yaw, rotor teeter, rotor speed, and drive-shaft torsional flexibility. The flexible tower has two modes each in the fore-and-aft and side-to-side directions. The flexible blades have two flapwise modes and one edgewise mode per blade. These degrees of freedom can be turned on or off individually in the analysis by simply setting a switch in the input data file. Figure 2 shows the layout of a conventional, downwind, two-bladed turbine.

FAST uses Kane's method to set up equations of motion that are solved by numerical integration. The implemented method makes direct use of the generalized coordinates, eliminating the need for separate constraint equations. FAST uses the AeroDyn subroutine package developed by Windward Engineering to generate aerodynamic forces along the blade. FAST is extensively documented in the *FAST User's Guide*. Please refer to it for details on the use of the program; it is continuously updated by Jason Jonkman.

3 Interfacing FAST and MATLAB/Simulink

FAST is an aeroelastic simulation package composed of many FORTRAN subroutines. According to the *FAST User's Guide*, FAST “can model the dynamic response of both two- and three-bladed, conventional, horizontal-axis wind turbines. A wind turbine configuration may optionally include rotor-furling, tail-furling, and tail aerodynamics.”

3.1 Step-by-Step Preparation

As a first step, readers should download the latest version of FAST from <http://wind.nrel.gov/designcodes/simulators/fast/>. It is assumed that the reader has already installed MATLAB/Simulink. (Version 2012b was used to prepare this document.) FAST (version 7.0 was used to prepare this document) includes a MATLAB/Simulink interface, as detailed in the *FAST User's Guide* (pp. 35–37; included in the FAST download archive). Readers should familiarize themselves with the interface and follow the steps presented in the *FAST User's Guide* to set up and run the interface. Readers should also familiarize themselves with the Simulink example models, *OpenLoop.mdl* and *Test01_SIG.mdl*, supplied with FAST. The steps to interface FAST and MATLAB/Simulink are as follows, in brief:

1. Download the FAST self-extracting *.exe* archive file and *Fast User's Guide* from the link above.
2. Extract FAST files to a directory of your choice (e.g., *C:\FAST*).
3. Copy the files *Simsetup.m* and *OpenLoop.mdl* from the *C:\FAST\Simulink\Samples* folder to the *C:\FAST\CertTest* folder. This folder contains the model input *.fst* files from many certification tests on different turbines.
4. Run MATLAB. In the command window's *File* menu, select **Set Path**, select **Add Folder**, and, in the browser window that opens, choose the path in which the Simulink interface files are stored (in our case, *C:\FAST\Simulink*). Save and close the **Set Path** dialog box.
5. Change the current folder in the command window to the *C:\FAST\CertTest* folder to be able to run *Simsetup.m* and *OpenLoop.mdl*.
6. The following commands should be entered at the MATLAB command prompt to clear all variables and close all open files:

```
clear all; fclose('all');
```

7. At the MATLAB command prompt, type:

```
Simsetup
```

to run *Simsetup.m*. Upon running, *Simsetup.m* will call script *Read_FAST_input.m*, which will then prompt the reader to enter the name of the FAST input file to use.

8. For now, the user should enter:

```
Test01.fst.
```

- Open the example Simulink model, *OpenLoop.mdl*, by choosing **Open** from the **File** menu. The Simulink model should appear as it does in Figure 3 below. (The green block in Figure 3 contains the FAST wind turbine blocks shown in Figure 4.)
- In the Simulink model window, click on the **Simulation** menu and then click on **Configuration parameters**. In the Solver Options section, choose fixed-speed as the solver type, and choose the *ode4* solver. Finally, click on the Play (▶) button in the Simulink window to run the simulation.

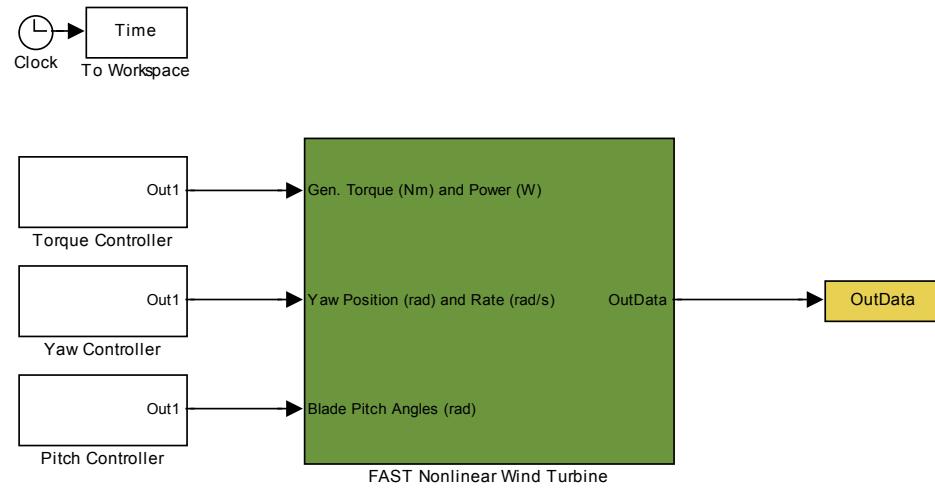


Figure 3. Simulink model *OpenLoop.mdl*

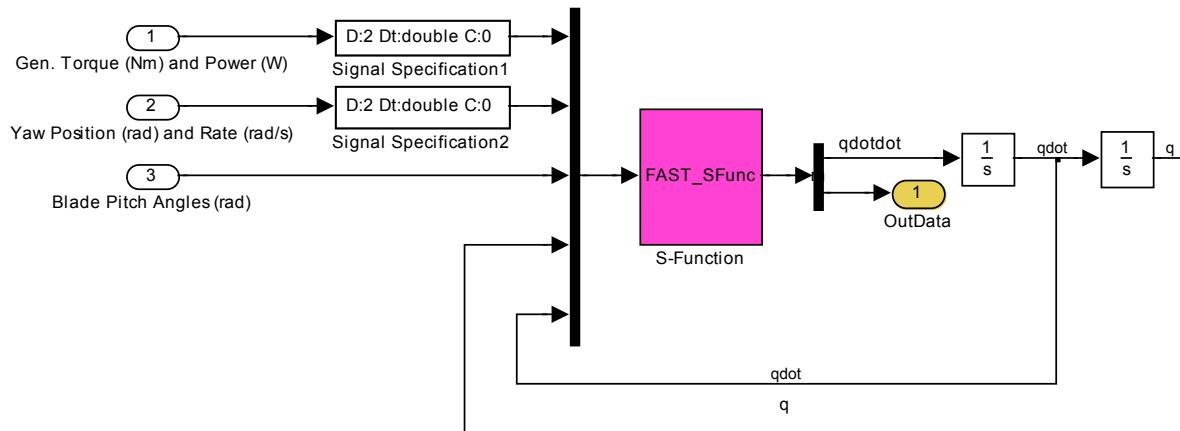


Figure 4. Inside the wind turbine block of the FAST wind turbine model in Simulink

- On the first try, the simulation will not run because the *Test01.fst* file has been set up to use the ADAMS preprocessor, which is not available in MATLAB/Simulink. The reader has to open and modify the *Test01.fst* using a text editor, such as Notepad. Find the following line near the top of the page:

```
3 ADAMSPrep - ADAMS preprocessor mode {1: Run FAST, 2: use FAST as a preprocessor to create an ADAMS model, 3: do both} (switch)
```

and change it to:

```
1 ADAMSPrep - ADAMS preprocessor mode {1: Run FAST, 2: use FAST as a preprocessor to create an ADAMS model, 3: do both} (switch)
```

The reader may now save the file in the same folder under a different name, for example, *Test01A.fst*, and return to Step 6 to repeat the instructions, except load *Test01A.fst* instead of *Test01.fst* when prompted to in Step 8.

12. On the second attempt, the simulation should run smoothly. If the user would like to view the model outputs, scopes can be added. At the MATLAB prompt, type :

```
simulink
```

This will open the Simulink Library Browser window. In this window, Scopes can be found under the **Sinks** directory. Drag and drop a Scope from this browser onto the *OpenLoop.mdl* and attach the input of the scope to signal(s) of choice with a wire element. Repeat Step 6 through Step 10 (with *Test01A.fst*). After the simulation, double-click on the Scope to view the output.

Having become familiar with the *OpenLoop.mdl*, the reader can explore a slightly more advanced model. The file *Test01_SIG.mdl* is also provided inside the FAST/Simulink archive. This model implements an identical induction generator representation to that in the *OpenLoop.mdl* in Simulink rather than in FAST. The reader can find and follow the additional steps provided in the commented lines of *Simsetup.m* to set up and run the model. There are four necessary modifications on both *Test01.fst* and *Simsetup.m*:

1. *Test01.fst*: Change the ADAMSPrep value from 3 to 1. (Step 10 above.)
2. *Test01.fst*: Change the VSContrl value from 0 to 3.
3. *Test01.fst*: Add “LSSGagVxa” to the OutList.
4. *Simsetup.m*: Uncomment the indicated lines for variable initialization.

It should be noted that instructions pertaining to the “OutList” mentioned in *Simsetup.m* refer to the list of outputs provided at the end of *Test01.fst*. To follow these instructions, add the following line on a new line preceding the END statement in *Test01.fst*:

```
"LSSGagVxa" - LSS rotor speed RPM
```

Rename the edited files, for example to *Test01B.fst* and *SimsetupB.m*, respectively. Repeat Step 6 through Step 10, replacing the Simsetup command with SimsetupB in Step 7 and loading *Test01B.fst* instead of *Test01.fst* in Step 8. Again, the simulation may not work on the first try, and the following error may appear:

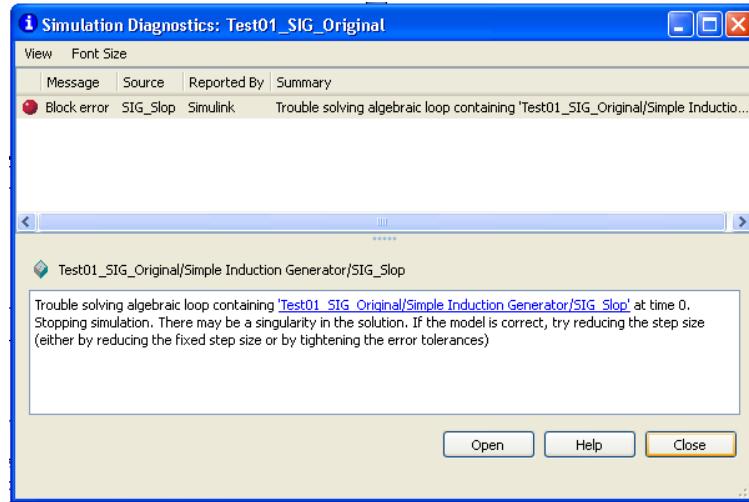


Figure 5. Error diagnostic while running *Test01_SIG.mdl*

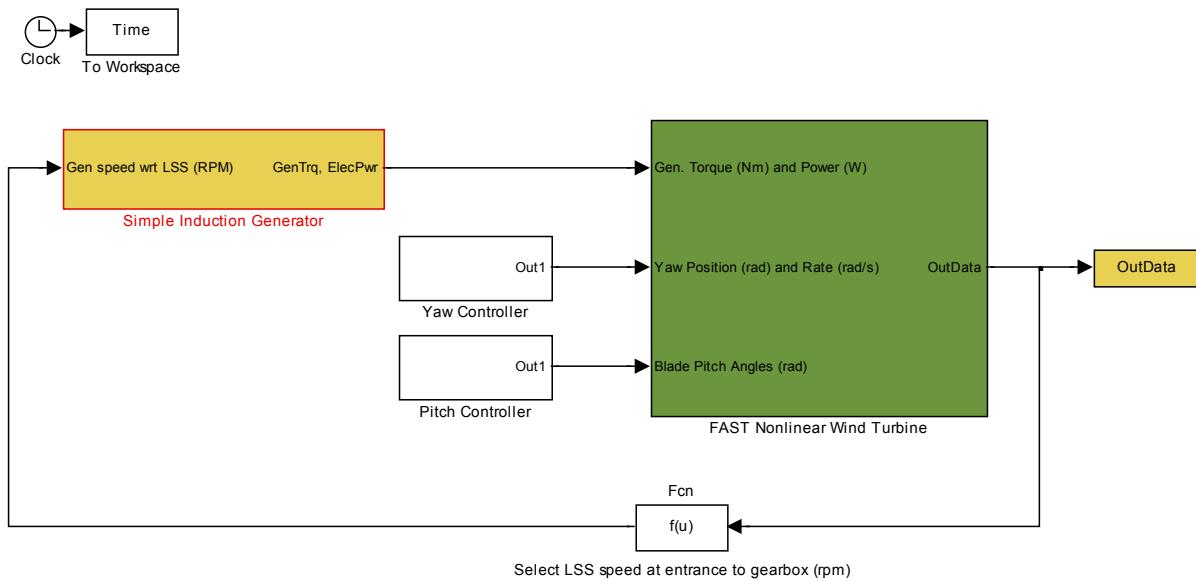


Figure 6. *Test01_SIG.mdl* in Simulink

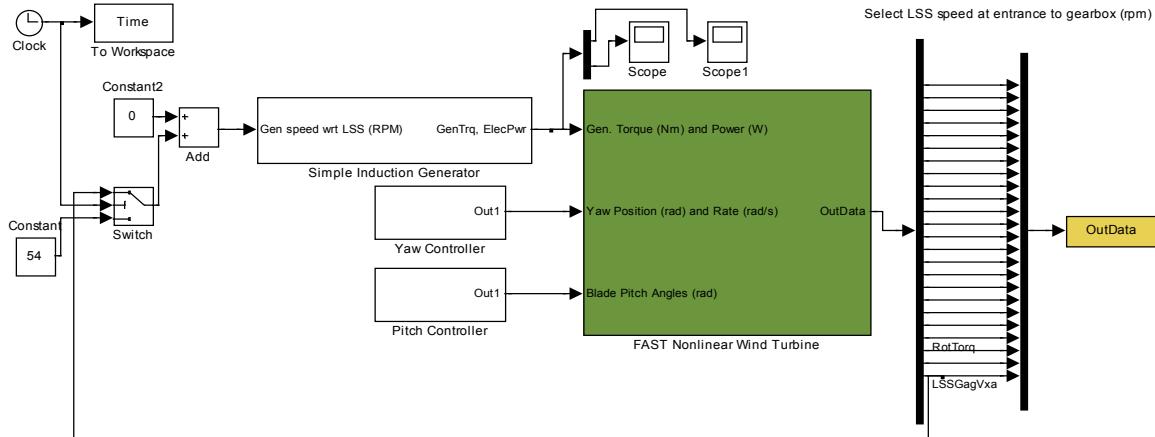


Figure 7. Modified model with Switch, Constant, and Add blocks

This error occurs because the Simple Induction Generator model does not receive any input for the first time step (i.e., no initial condition). Thus, this error can be resolved by modifying the model to initialize an input to the generator model. Consider the *Test01_SIG.mdl* model shown in Figure 6. Drag and drop a Constant and a Switch block from the **Commonly Used Blocks** directory and an Add block from the **Math Operations** directory in the Simulink Library Browser. Using these blocks, modify the model as shown in Figure 7. The initialization and zero-addition ensure that an initial condition is defined to the generator model. The time threshold value for the switch is set to $t=0.1$ s. Also, note the addition of a signal Demux block (from the **Signal Routing** directory). This allows the reader to extract and plot the desired FAST outputs.

3.2 FAST Files and Data Entry

Some text editing is necessary to set up the FAST-MATLAB/Simulink interface. Programs such as Notepad and Wordpad are sufficient for these tasks. The reader should associate the following files:

- *.fst* files: These are FAST input files that contain the turbine main parameters that are to be loaded by the *Simsetup.m* file. Many *.fst* files (*Test01.fst* through *Test17.fst*) are provided in the *C:\FAST\CertTest* folder for different turbines under a variety of operating conditions. Editing these files is necessary to change the turbine data, control methods, simulation conditions, step times, and outputs.
- *.ipt* files: These are aerodynamic data files defined under the Aerodyn section of *.fst* file. These files “call” the blade airfoil and wind files (*.wnd* files).
- *.wnd* files: These files contain the wind profiles: speed, direction, etc.

Editing *.fst* files was discussed above. The only editing employed for *.ipt* files is to change the name of the *.wnd* file that the *.ipt* file calls. These files can be found in the *C:\FAST\CertTest\Wind* folder, which contains multiple *.wnd* files for different turbine types. The *Test01.fst* file models the AWT-27, a two-bladed downwind turbine. The contents of *Shr12_30.wnd*, a wind file associated with *Test01.fst*, are presented below:

```
! Wind file for sheared 18 m/s wind with 30-degree direction.
! Time Wind Wind Vert. Horiz. Vert. LinV Gust
```

!	Speed	Dir	Speed	Shear	Shear	Shear	Speed
0.0	12.0	30.0	0.0	0.0	0.2	0.0	0.0
0.1	12.0	30.0	0.0	0.0	0.2	0.0	0.0
999.9	12.0	30.0	0.0	0.0	0.2	0.0	0.0

For simplicity and future testing of controllers, we recommend editing this file to include a step change in the wind speed from 12 m/s to 15 m/s at time $t=10$ s. For now, all gust and shear components can be removed, and wind direction can be assumed to be perpendicular to the plane of rotation of the turbine. The file can be saved as a new file, *Shr12_30B.wnd*, and the *Test01_AD.ckpt* file can be modified to call the modified file rather than *Shr12_30.wnd*. The file should look as shown below:

```
! Wind with step change at t = 10 s from 12 m/s to 15 m/s.
! Time Wind Wind Vert. Horiz. Vert. LinV Gust
! Speed Dir Speed Shear Shear Shear Speed
0.0 12.0 0.0 0.0 0.0 0.0 0.0 0.0
9.9 12.0 0.0 0.0 0.0 0.0 0.0 0.0
10.0 15.0 0.0 0.0 0.0 0.0 0.0 0.0
999.9 15.0 0.0 0.0 0.0 0.0 0.0 0.0
```

At this stage, the reader should be comfortable working with FAST and MATLAB/Simulink and should be confident about making changes to the model and FAST input files. The reader should consult the *FAST User's Guide* if additional information is required. The next section focuses on creating realistic induction generator models instead of using the ones employed by FAST.

4 Wind Turbine Modeling

Wind turbines are complex electromechanical devices interacting with a changing environment. Modelers of wind turbines typically concentrate on the details of subsystems or aspects of a turbine that they are interested in while using simplistic representations of other subsystems. In particular, aerodynamic modelers of wind turbines tend to oversimplify a turbine's electrical systems; likewise, electrical modelers ignore or oversimplify turbine aerodynamics. These approaches may lead to inaccurate and unrealistic models. For example, torque pulsations caused by the tower shadow effect observed in downwind turbines may impact electrical systems, but most electrical models do not account for this effect. This user's guide is intended for those interested in developing holistic wind turbine models that include detailed aerodynamics and structural, mechanical, and electrical systems using the FAST code developed by NREL interfaced with the popular MATLAB/Simulink platform.

Because FAST's in-built functionality accurately represents wind turbine aerodynamics and structures (see the *FAST User's Guide*), this guide concentrates on modeling electrical systems in MATLAB/Simulink and on how to interface these electrical system models with the FAST code. This guide will be particularly useful for non-electrical engineers looking to evaluate turbine performance with a realistic generator model. It is assumed that the reader is familiar with the MATLAB/Simulink environment and is capable of some simple programming. In the following subsection, classification of wind turbine technology is presented from an electrical engineering point of view.

According to differences in generation technology, wind turbines have been classified into four basic types:

- Type 1: Fixed-speed wind turbines
- Type 2: Variable-slip wind turbines
- Type 3: Doubly-fed induction generator (DFIG) wind turbines
- Type 4: Full-converter wind turbines

Fixed-speed wind turbines (popularly known as the “Danish concept”) are the most basic utility-scale wind turbines in operation. They operate with very little variation in turbine rotor speed and employ squirrel-cage induction machines directly connected to the grid. External reactive power support is necessary to compensate for the reactive power consumed by the induction machine. Because of the limited speed range in which these turbines operate, they are prone to torque spikes that may damage the mechanical subsystems within a turbine and cause transients in the electrical circuitry. These turbines may employ stall regulation, active stall regulation, or blade pitch regulation to regulate power at high wind speeds. Despite being relatively robust and reliable, there are significant disadvantages of this technology, namely that energy capture from the wind is suboptimal and reactive power compensation is required. An example of a popular fixed-speed wind turbine is the NEG Micon NM64/1500 turbine, rated at 1.5 MW. A schematic for a fixed-speed wind turbine is shown in Figure 8.

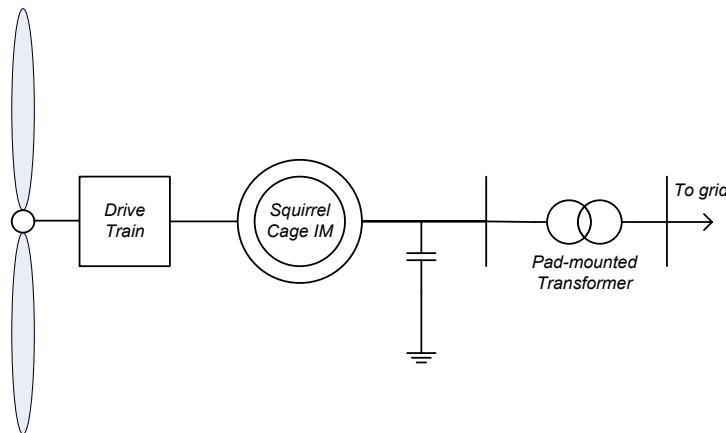


Figure 8. Fixed-speed wind turbine schematic

Variable-speed wind turbines (the broad category into which the other three dominant technologies fall) are designed to operate at a wide range of rotor speeds. These turbines usually employ blade pitching for power regulation. Speed and power controls allow these turbines to extract more energy from a given wind regime than fixed-speed turbines can. Variable-slip turbines employ wound-rotor induction machines that allow access to both the stator and the rotor of the machine. The rotor circuit of the machine is connected to an alternating current (AC)/direct current (DC) converter and a fixed resistance. The converter is switched to control the effective resistance in the rotor circuit of the machine to allow a wide range of operating slip (speed) variation (up to 10%). However, power is lost as heat in the external rotor circuit resistance. A controller may be employed to vary the effective external rotor resistance for optimal power extraction. Reactive power compensation is still required. Vestas OptiSlip turbines, such as the Vestas V66 (1.65 MW), were the most successful turbines to employ this technology. A schematic for this technology is shown in Figure 9.

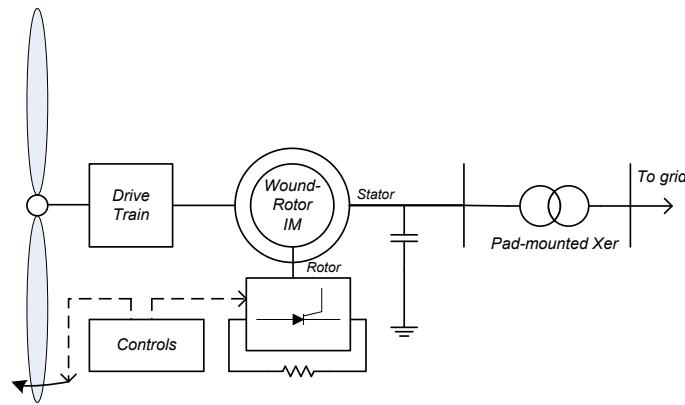


Figure 9. Variable-slip wind turbine schematic

DFIG turbines remedy the problem of power loss in the rotor circuit by employing a back-to-back AC/DC/AC converter in the rotor circuit to recover the slip power. Flux-vector control of rotor currents allows decoupled real and reactive output power as well as maximized wind power extraction and lowered mechanical stresses. Also, these turbines usually employ blade pitching for power regulation. Because the converter handles only the power in the rotor circuit, it does not need to be rated at the machine's full output power. The disadvantages of this technology—

namely, higher cost and complexity—are offset by the ability to extract more energy from a given wind regime than the preceding technologies. The General Electric 1.5-MW turbine is an example of a successful DFIG implementation; more than 15,000 have been installed. A schematic for this technology is shown in Figure 10.

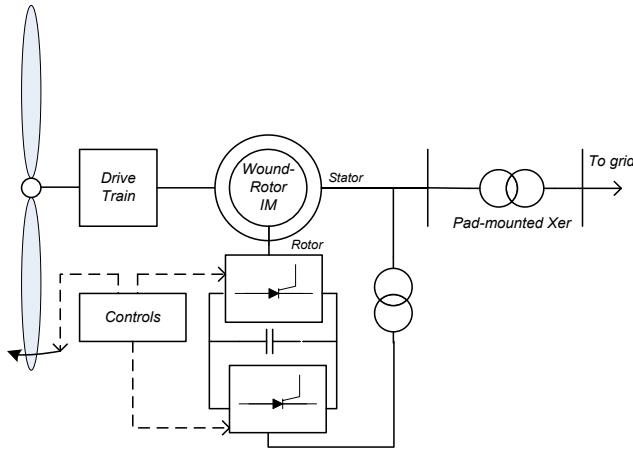


Figure 10. DFIG wind turbine schematic

In full-converter turbines, a back-to-back AC/DC/AC converter is the only power flow path from a wind turbine to the grid. Thus, there is no direct connection to the grid, and the converter has to be rated to handle the entire output power. These turbines usually employ high-pole-count, permanent magnet, synchronous generators to allow low-speed operation, thus allowing the elimination of the gearbox to increase reliability. Nonetheless, using induction generators is also possible. Also, full-converter turbines offer independent real and reactive power control, and they typically employ blade pitching for power regulation. A schematic for this technology is shown in Figure 11. Although these turbines are relatively expensive, the increased reliability and simplicity of the control scheme vis-à-vis DFIG turbines are attractive features, especially in offshore installations where maintenance is costly. Enercon manufactures turbines based on this technology, such as the popular E82 2-MW turbine.

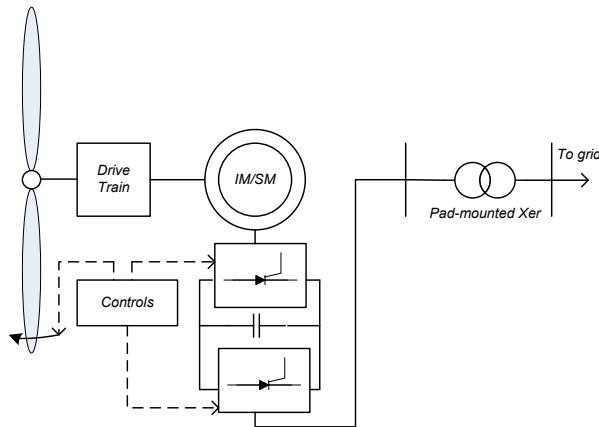


Figure 11. Full-converter wind turbine schematic

Of the four types of turbines, this document focuses on Type 1 and 2 turbines because they show the most coupling between mechanical and electrical systems. The next section describes the modeling of Type 1 turbines.

4.1 Type 1 Wind Turbine Model

Type 1 wind turbines are the least complex utility-scale turbines. They consist of a rotor (blades and hub) coupled to a squirrel-cage induction generator through a gearbox. The gearbox and generator are situated within the nacelle of the turbine at the top of the tower. The stator of the induction generator is connected to the grid through a step-up transformer. A shunt capacitor bank is typically added to provide reactive power support. Electrical controls are typically minimal, though mechanical controls such as yaw control and blade pitch control may be employed. An example model provided in FAST, *Test01_SIG.mdl*, is a Type 1 turbine models. This section covers modifications to *Test01_SIG.mdl* to improve the existing induction generator model, which inadequately represents the generator's dynamics. The following subsections evaluate the deficiencies of the existing models, identify an alternate model, and discuss the implementation of the model in Simulink. It also discusses the development of a blade pitch angle controller to complete the Type 1 model. Yaw control will be addressed in the future.

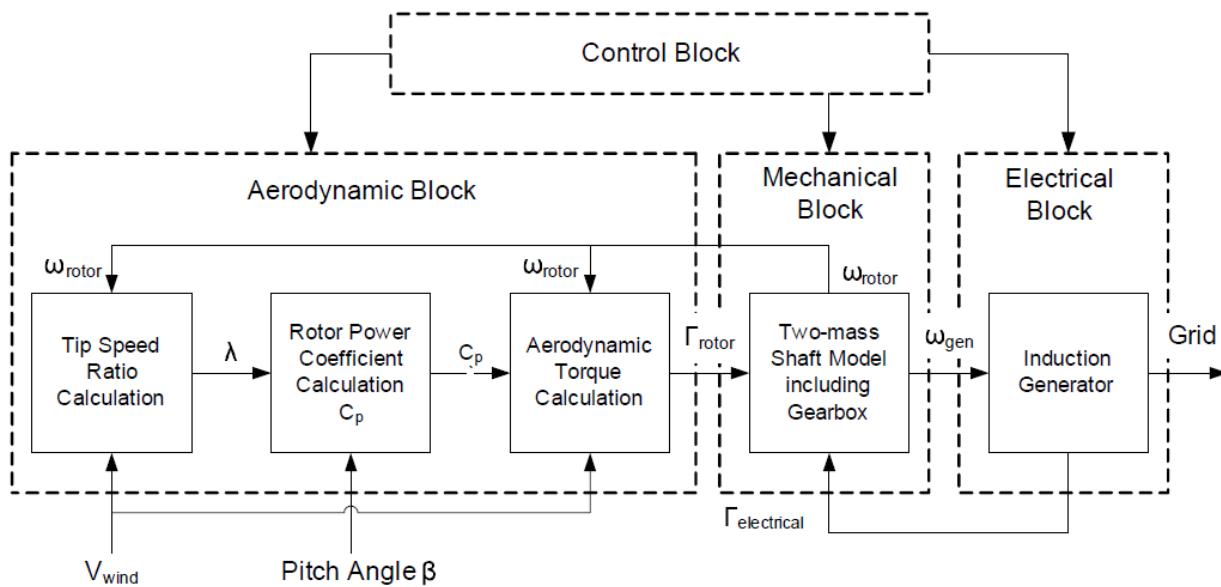


Figure 12. Subsystems for a Type 1 turbine model

4.1.1 Preexisting FAST Type 1 Turbine Models (Steady-State Model)

Type 1 turbines may be represented as a combination of subsystems. The framework shown in Figure 12 is typically used for modeling purposes. For our purposes, FAST performs all the functions of the aerodynamic and mechanical blocks, with some additional functionality not shown in Figure 12. We chose FAST because of its great fidelity to real-world turbine aeroelastic characteristics.

FAST inherently provides induction generator models. Two parameters in the *.fst* input file govern FAST's choice of the generator model: VSContrl and GenModel. The parameter VSContrl determines if torque is actively controlled (i.e., whether a turbine is fixed-speed or variable-speed). If VSContrl is set to 0, the turbine is assumed to be one of fixed speed. FAST

then determines which generator model to use based on parameter GenModel, which may be set to 1, 2, or 3. Setting GenModel to 1 tells FAST to use a simple generator model, represented as an induction machine torque-speed curve, an example of which is shown in Figure 13.

The generating region is a narrow band on the torque-speed curve, with high-torque variations within a small speed range. Type 1 fixed-speed machines typically operate in an even narrower region. The torque-speed curve is calculated by FAST from the four parameters in the Simple Induction Generator section of the *.fst* input file. The parameters explained in Table 1 were reproduced directly from the *FAST User's Guide* (pp. 64). Figure 14 shows the curve based on these parameters, based on a linear approximation employed by FAST. (It should be noted that this curve is a torque-slip curve rather than a torque speed curve. Slip is an analogy of speed; its definition is provided in Table 1.) After FAST outputs a rotor speed value, this value is used as an input to the torque calculation. Because of the linear relationship between torque and slip (and thus speed), the torque can easily be calculated. This torque is fed back into the FAST turbine model. To maintain equilibrium, the generator electrical torque is considered to be equal to and in the opposite direction of the aerodynamic torque from the wind, preventing runaway acceleration of the rotor and resulting in fixed-speed operation. Power can be calculated by multiplying the generator torque, speed, and efficiency factor.

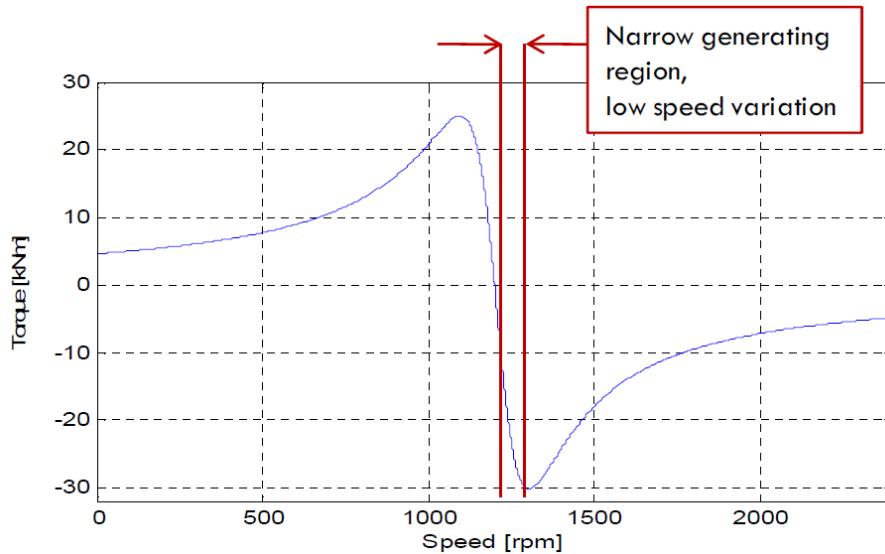


Figure 13. Example of an induction machine torque curve

Table 1. Parameters for the Simple Induction Generator Model (VSContrl = 0, GenModel = 1)

SIG_SIPc	The rated generator slip percentage is the difference between the rated and the synchronous generator speed, divided by the synchronous generator speed, and then converted to a percentage. This value must be greater than 0, but it is ignored if GenModel is not equal to 1 or VSContrl is not equal to 0 (%).
SIG_SySp	This is the synchronous, or zero-torque, generator speed. This value must be greater than 0, but it is ignored if GenModel is not equal to 1 or VSContrl is not equal to 0 (rpm).
SIG_RtTq	This is the torque supplied by the generator when running at rated speed. This value must be greater than 0, but it is ignored if GenModel is not equal to 1 or VSContrl is not equal to 0 (N·m).
SIG_PORT	The pullout ratio is the ratio of the pullout torque and the rated torque. The negative of this value is also used for the startup torque. This value must be greater than or equal to 1, but it is ignored if GenModel is not equal to 1 or VSContrl is not equal to 0.

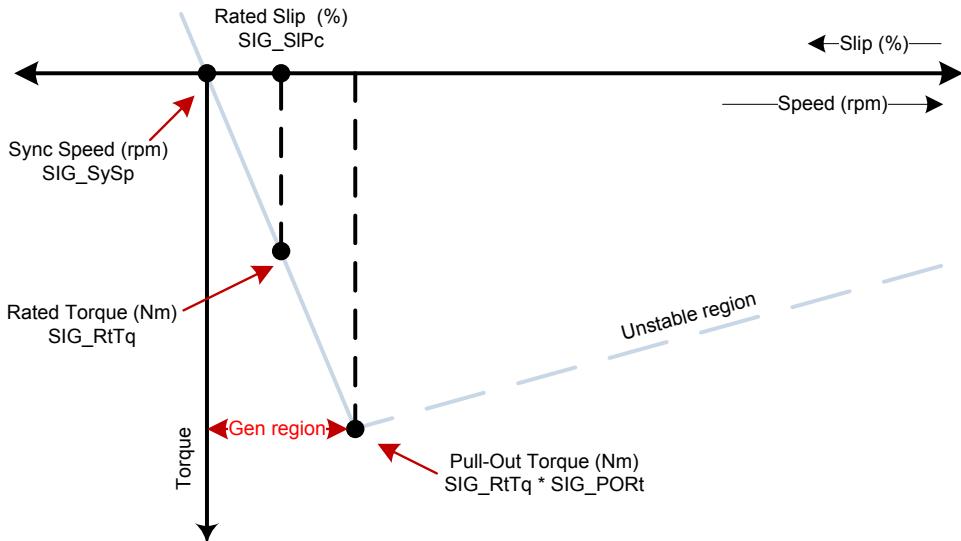


Figure 14. Generation of a torque-slip curve from parameters supplied in Table 1

The example model *Test01.fst* employs the simple induction generator model (*VSContrl* = 0, *GenModel* = 1). The parameter *GenModel* may be changed to 2, in which case a slightly more advanced model based on the induction machine single-phase equivalent circuit (shown in Figure 15) is used. All rotor-side quantities have been referred to the stator side. The circuit is further simplified into a Thevenin-equivalent circuit using these quantities before being solved by FAST. The induction machine parameters can be found in the Thevenin-Equivalent Three-Phase Generator section in the *.fst* input file. Table 2 explains the parameters in the *.fst* file and in the equivalent circuit in Figure 15. For the equations relating torque and speed that emerge from this equivalent circuit model, consult *Electric Machinery Fundamentals* by S. J. Chapman (McGraw-Hill 1985). This induction machine equivalent circuit model, though superior to the linear model discussed previously, is only a steady-state model and does not adequately represent the dynamics of the induction machine.

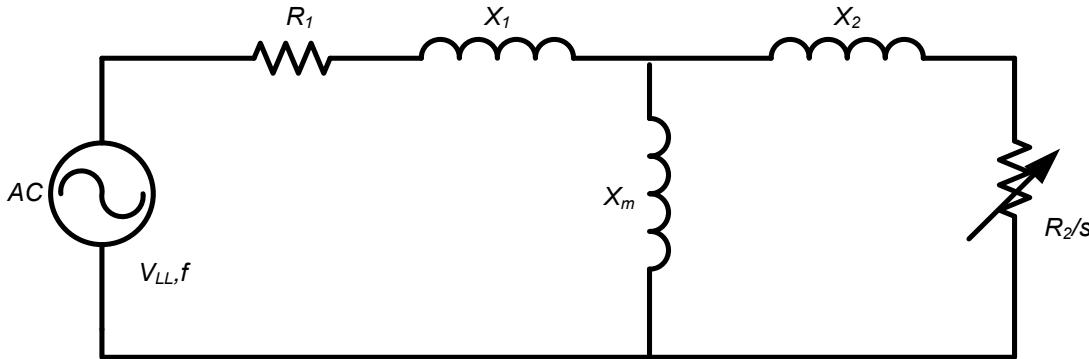


Figure 15. Induction machine single-phase equivalent circuit

Table 2. Parameters for the Thevenin-Equivalent Model (VSContrl = 0, GenModel = 2)

TEC_Freq	f	This is the line frequency of the electrical grid. This value must be greater than 0 and should be 50 (Europe) or 60 (United States), but it is ignored if GenModel is not equal to 2 or VSContrl is not equal to 0.
TEC_NPol		This is the number of poles in the generator. This value must be an even integer greater than 0, but it is ignored if GenModel is not equal to 2 or VSContrl is not equal to 0.
TEC_SRes	R_1	This is the resistance of the generator stator in the complete circuit. This value must be greater than 0, but it is ignored if GenModel is not equal to 2 or VSContrl is not equal to 0 (ohms).
TEC_RRes	R_2	This is the resistance of the generator rotor in the complete circuit. This value must be greater than 0, but it is ignored if GenModel is not equal to 2 or VSContrl is not equal to 0 (ohms).
TEC_VLL	V_{LL}	This is the line-to-line voltage of the generator. This value must be greater than 0 and is often 690 in Europe or 480 or 575 in the United States, but it is ignored if GenModel is not equal to 2 or VSContrl is not equal to 0 (volts).
TEC_SLR	X_1	This is the leakage reactance of the generator stator in the complete circuit. This value must be greater than 0, but it is ignored if GenModel is not equal to 2 or VSContrl is not equal to 0. It is usually a small number and close in value to the stator resistance (ohms).
TEC_RLR	X_2	This is the leakage reactance of the generator rotor in the complete circuit. This value must be greater than 0, but it is ignored if GenModel is not equal to 2 or VSContrl is not equal to 0. It is usually a small number and close in value to the rotor resistance (ohms).
TEC_MR	X_m	This is the magnetizing reactance of the complete generator circuit. This value must be greater than 0, but it is ignored if GenModel is not equal to 2 or VSContrl is not equal to 0. It is usually about 10 to 50 times greater than the leakage reactances (ohms).

Setting GenModel =3 allows the user to write a user-defined generator model in FORTRAN. Nonetheless, we opted to develop generator models using a more visual block diagram representation of Simulink. For this purpose, we could not set VSContrl = 0 because FAST would require input from Simulink. Setting VSContrl = 1 would cause FAST to use an in-built, simple, variable-speed generator model; whereas setting VSContrl = 2 allows users to write their

own variable-speed generator model in FORTRAN. Neither of these options was applicable. Setting VSContrl = 3 allows input from Simulink, which is desired. This setting is to be used to run the example model *Test01_SIG.mdl*. When VSContrl is no longer zero, the GenModel parameter is ignored by FAST and the torque input to the FAST turbine model must come from elsewhere; in our case that was Simulink. Although a nonzero value of VSContrl implied variable-speed operation, we could still model a fixed-speed turbine. Figure 6 shows the model *Test01_SIG.mdl*. The top left shows a subsystem block labeled “Simple Induction Generator.” This block received a speed input from FAST and delivered torque power vectors as output. The model inside this block was implemented the same linear torque calculation as in Figure 15, solved using Simulink blocks instead of FORTRAN. Double-clicking on the subsystem block opened a new window of its internal components, as shown in Figure 15. The low-speed shaft (LSS) speed in rpm was converted to the high-speed shaft (HSS) speed ω at the generator in radians/sec using the gearbox ratio, defined in the *.fst* file. Synchronous speed value SIG_SySp ω_s was then subtracted from the HSS speed. The resulting difference ($\omega - \omega_s$) was multiplied by the torque-speed slope (SIG_Slop), which, from Figure 14, can be written as $\frac{\tau_{rated}}{\omega_{rated} - \omega_s}$. The resulting output torque was $\tau = \tau_{rated} \frac{\omega - \omega_s}{\omega_{rated} - \omega_s}$. This output torque was limited to a maximum value specified by the pullout torque SIG_POrt. The output torque was multiplied by speed ω and efficiency η to give output power (i.e., $P = \eta \tau \omega$). The torque and power were multiplexed into a 2×1 vector as a FAST input because this is the way it must be delivered. Note that there was a minor error in the example file *Test01_SIG.mdl*. The efficiency was specified in percent rather than per unit, hence the output power from the model was 100 times larger than the actual output in watts. Thus, we divided the power results by a factor of 100 before plotting. It appears that FAST does not use the power value, so this anomaly did not affect the simulation results. To run the simulation, please follow the steps prescribed in Section 3 for *Test01_SIG.mdl*, using the modified wind file *Shr12_30B.wnd*, which has a step change in the wind speed.

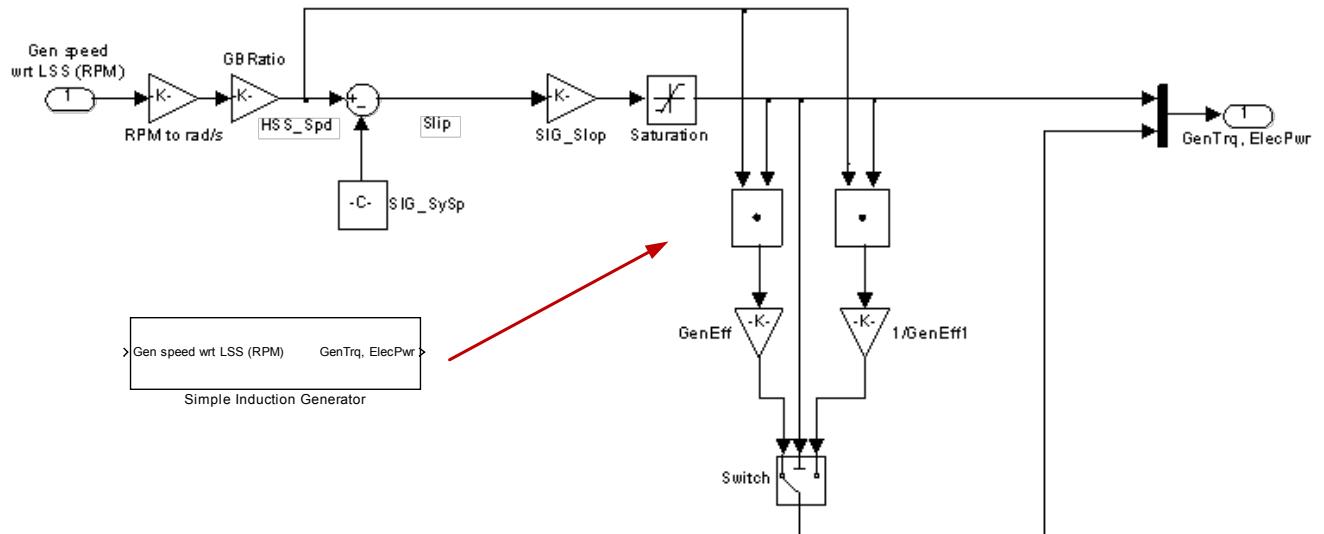


Figure 16. Torque calculation from speed, implemented in Simulink

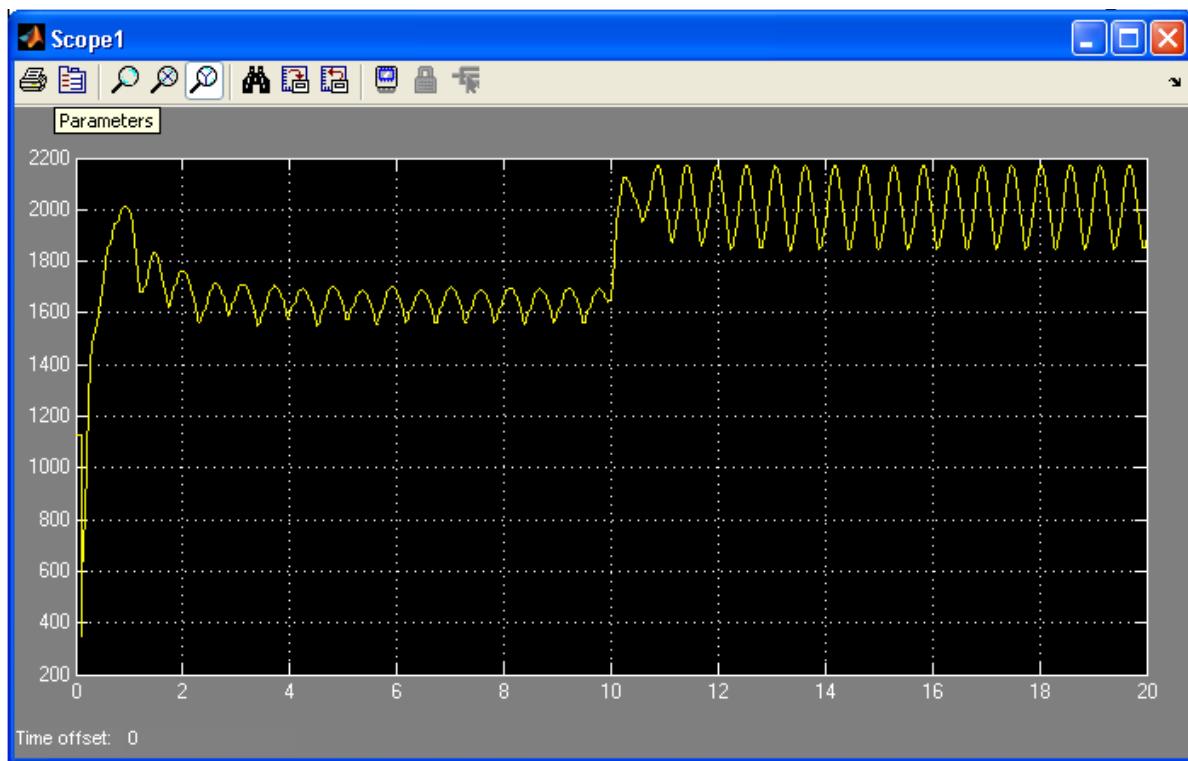


Figure 17. Example of a MATLAB scope output during run time

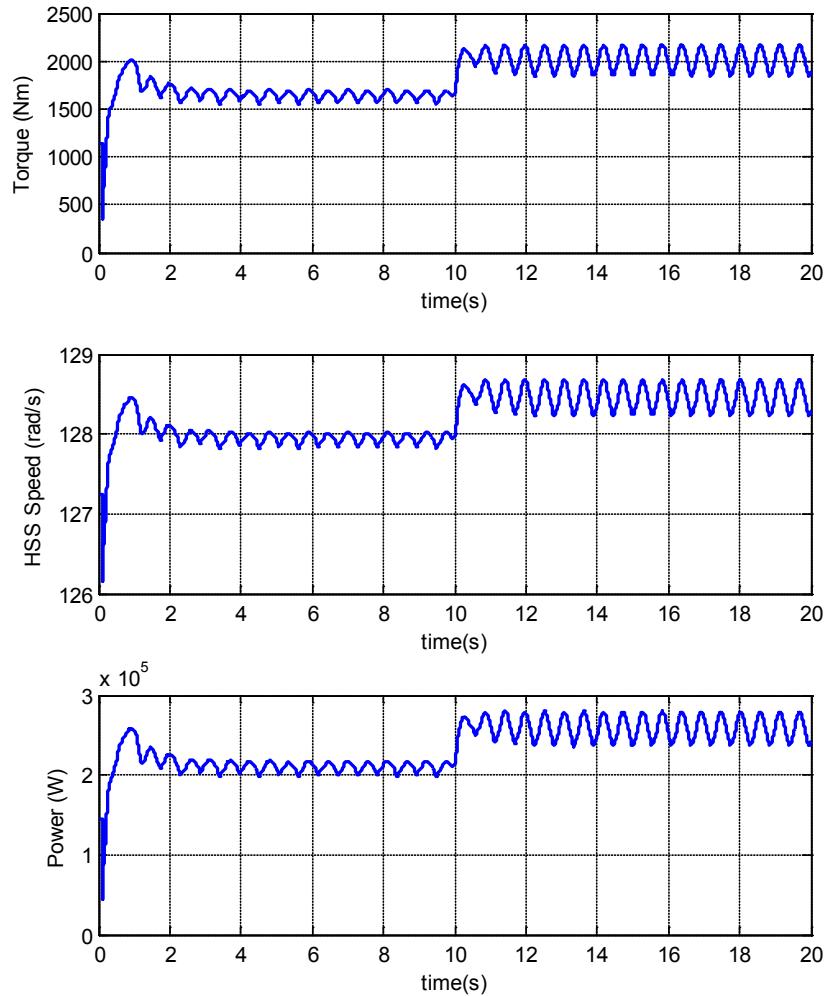


Figure 18. Torque, speed, and output power from *Test01_SIG.mdl* with a step change in the wind speed

The results showed that a significant step change in wind speed, from 12 m/s to 15 m/s, caused a very small (less than 1%) change in the HSS speed (from 128 rad/s to 128.5 rad/s). However, the output torque and power showed large changes. There was no yaw or pitch control in place. The oscillations in the quantities were probably the result of the torque shadow effect, a phenomenon caused by changing aerodynamic torque resulting from the tower interrupting the wind flows in downwind turbines. This model does not have the capability to change the electrical excitation to the generator because it was effectively connected to a perfect fixed-voltage, fault-proof, “infinite” bus. Thus, the electrical faults and their influence on the quantities associated with turbine performance cannot be simulated.

4.1.2 Dynamic Induction Machine Model

Induction machines are highly nonlinear devices. Standard nonlinear dynamic induction machine models are widely used. Many preexisting MATLAB/Simulink induction machine models are available, some of which can be acquired from Mathworks through the SimPowerSystems Simulink toolbox. Other Simulink models are freely available, such as those developed by Chee Mun Ong for his popular book *Dynamic Simulation of Electric Machinery*. These Simulink models are available at http://cobweb.ecn.purdue.edu/~ong/book_projects/c6/. We employed

these models in this document. The reader should download all files in the online repository to the *C:\FAST\CertTest* folder, which is the MATLAB working directory. We recommend that the reader becomes familiar with these models, using Ong's book as a reference. We used the induction motor model in the *S1.mdl* file, shown in Figure 19. This machine model will later be configured as a generator model.

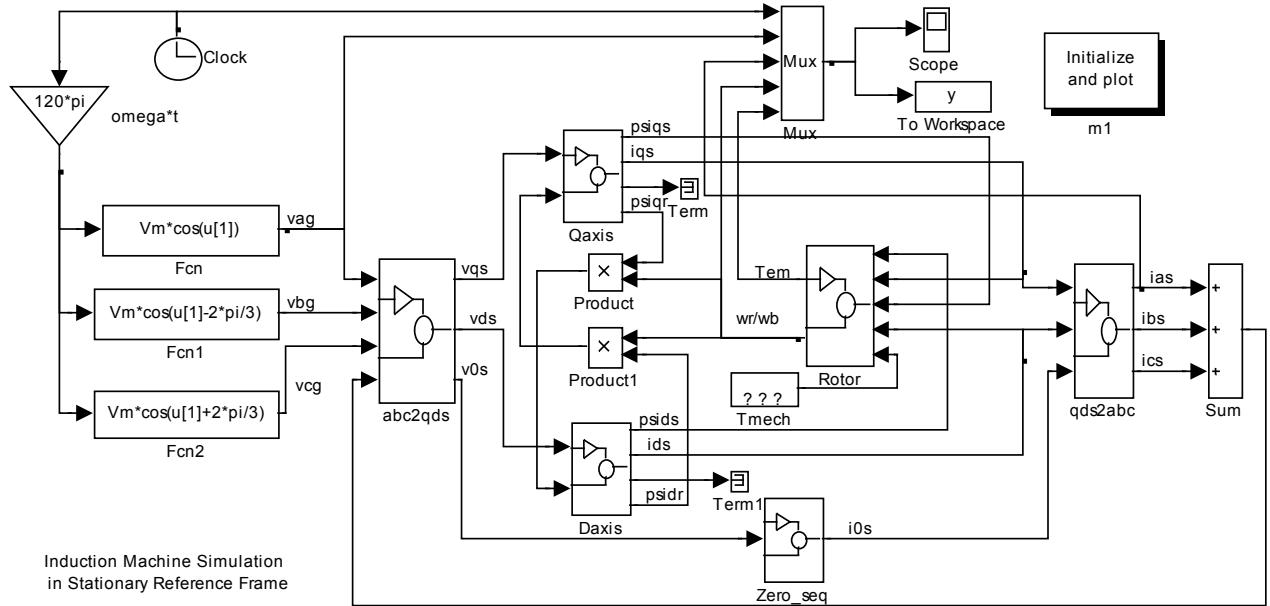


Figure 19. Induction machine model *S1.mdl*

As shown on the left of Figure 19, three voltage signals were generated, which took the form of $V = V_m \cos(\omega t + \theta)$. The signals were time-shifted by 120° from each other, with θ taking the values of 0° , -120° , and $+120^\circ$ for phases *a*, *b*, and *c* respectively. The three-phase voltages were then transformed into two orthogonal vectors (*d*-axis and *q*-axis) and a DC component (0-axis) using the *dq0* transformation, also known as Park's transformation, the details of which can be found in *Analysis of Electric Machinery* by P.C. Krause (McGraw-Hill 1998). This transformation was performed to facilitate easier analysis. Equations for the machine dynamics for the stator and rotor were solved separately. The currents were calculated from the voltages, and the electrical torque was calculated from the current. This model also included a one-mass mechanical model of the motor. Two opposing torques acted on the machine shaft—a mechanical one caused by the driven load and an electrical one caused by the winding currents. The sum of the opposing torques gave the net torque, from which acceleration and speed could be calculated. If constant speed operation is desired in a motor, the torques must be equal and in the opposite direction. Currents were converted back to *abc*-phase currents and summed. They should sum to 0, unless there is an unbalance in the input voltage(s) (i.e., voltage magnitude in one or two phases deviates from the nominal value). It should be noted that the 0-axis voltage and current will present only if there is voltage unbalance. The reader should note the initialization block in the top right corner of Figure 19. Double-clicking this block runs the file *m1.m*, which initializes some variables for the simulation. File *m1.m* in turn calls *p1hp.m*, which loads the induction machine parameters.

Once comfortable with *S1.mdl*, the reader can proceed to interfacing this machine model with FAST. The steps for this interfacing are as follows:

1. Open the Simulink models *Test01_SIG.mdl* and *S1.mdl*. From the **Ports & Subsystems** directory of the Simulink library browser, input a Subsystem block to the *Test01_SIG.mdl*.
2. Double-click on the newly added Subsystems block. The newly opened window will show an input port directly connected to an output port. Delete the connection between the two. Copy input 1 and paste it back in. This will provide the second input port (i.e., input 2). In this example, input 1 is for the clock signal and input 2 is for the speed signal from FAST. Modify the input and output port labels accordingly by double-clicking the labels. Close the subsystem window. In the main Simulink window, double-click the subsystem label and enter a label of choice, for example “Induction Machine Model,” as shown in Figure 20.
3. Delete the Simple Induction Generator block from the *Test01_SIG.mdl*.
4. In the main Simulink window, connect the Induction Machine Model’s input 1 to the Clock signal and input 2 to the speed signal LSSGagVxa from FAST. Connect the Induction Machine Model’s output to the FAST Gen. Torque (Nm) and Power (W) input. The model should appear as shown in Figure 16.

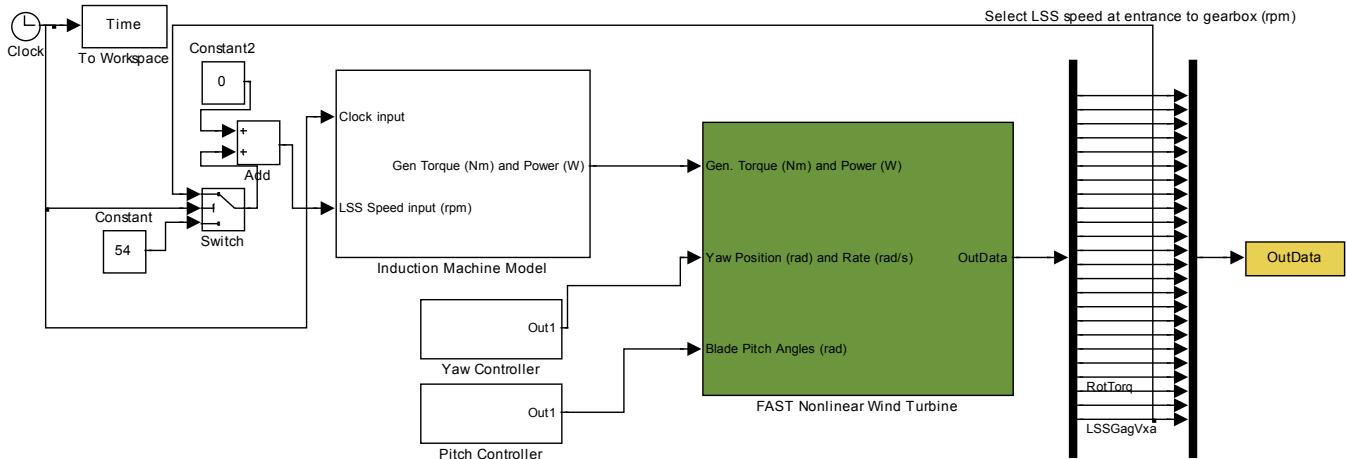


Figure 20. Modified *Test01_SIG.mdl* showing the Simple Induction Generator block replaced with the Induction Machine Model block

Double-click on the subsystem. Copy all the blocks from *S1.mdl* into the Induction Machine Model in *Test01_SIG.mdl*. Inside this subsystem, delete the initialization Clock blocks. Connect the input 1 to the input of the “omega*t” Gain block and to the white block labeled Mux.

5. From the Simulink Library browser, drag and drop into the subsystem four Gain blocks and a Product block from the **Math Operations** directory, a signal Terminator block from the **Sinks** directory, and a Mux block from the **Signal Routing** directory. These blocks are shown in Figure 21.

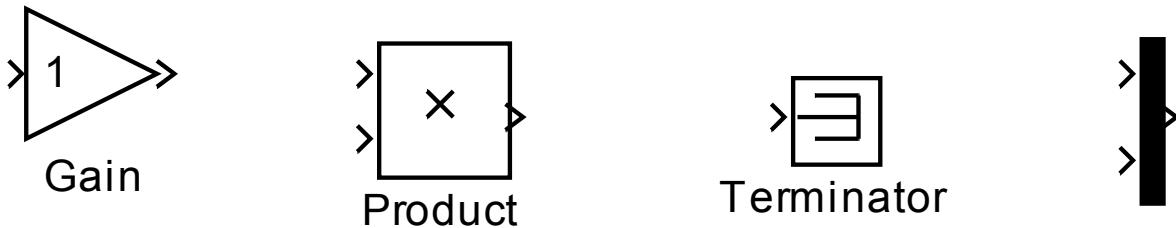


Figure 21. Blocks for the subsystem (Gain, Product, Terminator, and Mux)

- Label the Gain blocks and enter the gain values according to Table 3.

Table 3. Gain Block Data

 torque gain $\times -1 \gg$	Gain = -1 , to change the sign of the electrical output torque of the induction generator. This is required because FAST needs a positive torque input, but the convention for electrical machinery is negative electrical torque when the machine acts as a generator.
 omega*t2 $\times -K- \gg$	Gain = 2*60*pi/3 , to change the per unit speed required by the induction machine model to radians per second.
 GBRatio $\times -K- \gg$	Gain = GBRatio , to change the LSS speed from FAST to HSS speed. This block is used in conjunction with the block below. The parameter GBRatio can be found in the input <i>.fst</i> file.
 RPM to pu speed $\times -K- \gg$	Gain = 3*(pi/30)/(2*pi*60) = 1/1200 to change the HSS speed to per-unit speed required by the induction machine model.

- Modify the induction generator model to use a speed input from FAST. Navigate to the induction machine component labeled *Rotor*. This component performs two functions: it calculates electrical torque from currents and contains a one-mass mechanical model to calculate the machine rotational speed. We bypassed the mechanical model of this component because FAST already has a refined mechanical model. To do this, set the constant $T_{mech} = 0$ because it will not be used, disconnect output speed ω_b/ω_r from the other blocks, and terminate it using the terminator because the speed input from FAST will be used instead. Connect input 2 to the input of “GBRatio” Gain block, and connect the output of this gain block to the input of “RPM to pu speed” Gain block. The output of this block is the per unit speed. This per unit speed signal can be used, instead of ω_b/ω_r , as an input to the blocks that require it. Thus, it can be connected to the blocks from which the ω_b/ω_r signal was connected. The inputs have now been configured.
- Proceed to the outputs. Connect the electrical torque signal T_{em} from the *Rotor* block to the input of the “torque gain” Gain block. This is to invert the torque so that FAST can receive a positive value. The output of this gain block can be connected to the top input

of the newly added Mux block and to one of the inputs of the Product block. Connect the per unit speed output to the input of the “omega*t2” Gain block to get speed in radians per second. Connect the output of this block to the remaining input of the Product block so that the product of torque and speed gives the output power. Connect the output of this block to the lower input of the Mux block. Connect the output of the Mux block to the output port 1 of the subsystem. The output is now configured.

9. Ensure that the model is initialized with the correct data. Open *p1hp.m* and find the following statements:

```
% Parameters of machine used in Projects 1 and 3 of Chapter 6
```

```
Sb = 750; % rating in VA  
Prated = 750; % output power in W  
Vrated = 200; % rated line to line voltage in V  
pf = 0.8;  
Irated = Sb/(sqrt(3)*Vrated*pf); % rated rms current  
P = 4; % number of poles  
frated = 60; % rated frequency in Hz  
wb = 2*pi*frated; % base electrical frequency  
we = wb;  
wbm = 2*wb/P; % base mechanical frequency  
Tb = Sb/wbm; % base torque  
Zb = Vrated*Vrated/Sb; %base impedance in ohms  
Vm = Vrated*sqrt(2/3); % magnitude of phase voltage  
Vb = Vm; % base voltage  
Tfactor = (3*P)/(4*wb); % factor for torque expression  
  
rs = 3.35; % stator wdg resistance in ohms  
xls = 6.94e-3*wb;% stator leakage reactance in ohms  
xplr = xls; % rotor leakage reactance  
xm = 163.73e-3*wb; %stator magnetizing reactance  
rpr = 1.99; % referred rotor wdg resistance in ohms  
xM = 1/(1/xm + 1/xls + 1/xplr);  
J = 0.1; % rotor inertia in kg m2  
H = J*wbm*wbm/(2*Sb); % rotor inertia constant in secs.  
Domega = 0; % rotor damping coefficent
```

Copy this entire section and paste it into *SimsetupB.m*. Change the values to the ones shown below. Note that changing J , H , and $Domega$ is inconsequential because the rotor one-mass model has been bypassed. The data here is for a 500-hp, 4-pole machine from

Analysis of Electric Machinery by P. C. Krause, with slight adaptation to represent the 250-kW, 6-pole machine employed in the AWT-27 turbine:

```
Sb = 277777.777; % rating in VA  
Prated = 250000; % output power in W  
Vrated = 2300; % rated line to line voltage in V  
pf = 0.9;  
Irated = Sb/(sqrt(3)*Vrated*pf); % rated rms current  
P = 6; % number of poles  
frated = 60; % rated frequency in Hz  
wb = 2*pi*frated; % base electrical frequency  
we = wb;  
wbm = 2*wb/P; % base mechanical frequency  
Tb = Sb/wbm; % base torque  
Zb = Vrated*Vrated/Sb; %base impedance in ohms  
Vm = Vrated*sqrt(2/3); % magnitude of phase voltage  
Vb = Vm; % base voltage  
Tfactor = (3*P)/(4*wb); % factor for torque expression  
  
rs = 0.262; % stator wdg resistance in ohms  
% xls = 6.94e-3*wb;% stator leakage reactance in ohms  
xls = 1.206;  
xplr = xls; % rotor leakage reactance  
% xm = 163.73e-3*wb; %stator magnetizing reactance  
xm = 54.02;  
rpr = 0.187; % referred rotor wdg resistance in ohms  
xM = 1/(1/xm + 1/xls + 1/xplr);  
J = 0.1; % rotor inertia in kg m2  
H = J*wbm*wbm/(2*Sb); % rotor inertia constant in secs.  
Domega = 0; % rotor damping coefficient
```

10. Open *m1.m*, find the following lines, and copy them into *SimsetupB.m*. These are the initial values used for the *dq0* transformation:

```
Psiqso = 0; % stator q-axis total flux linkage
Psipqro = 0; % rotor q-axis total flux linkage
Psidso = 0; % stator d-axis total flux linkage
Psipdro = 0; % rotor d-axis total flux linkage
wrbywbo = 0; % pu rotor speed
```

11. The simulation is almost ready to be run. In the Simulink window, change the solver to variable-step, *ode45* (Dormand-Price), because the presented induction machine model will be unstable with the *ode4* solver. Type “*SimsetupB*” at the MATLAB command window, and, when prompted, enter the required *.fst* file; in our case, *Test01B.fst*. Proceed to run the simulation and observe the results. Note that the simulation will be very slow (it ran for more than 30 minutes using our computer) because the variable-step solver chose a very small time step. The reader should see results in agreement with those in Figure 22.

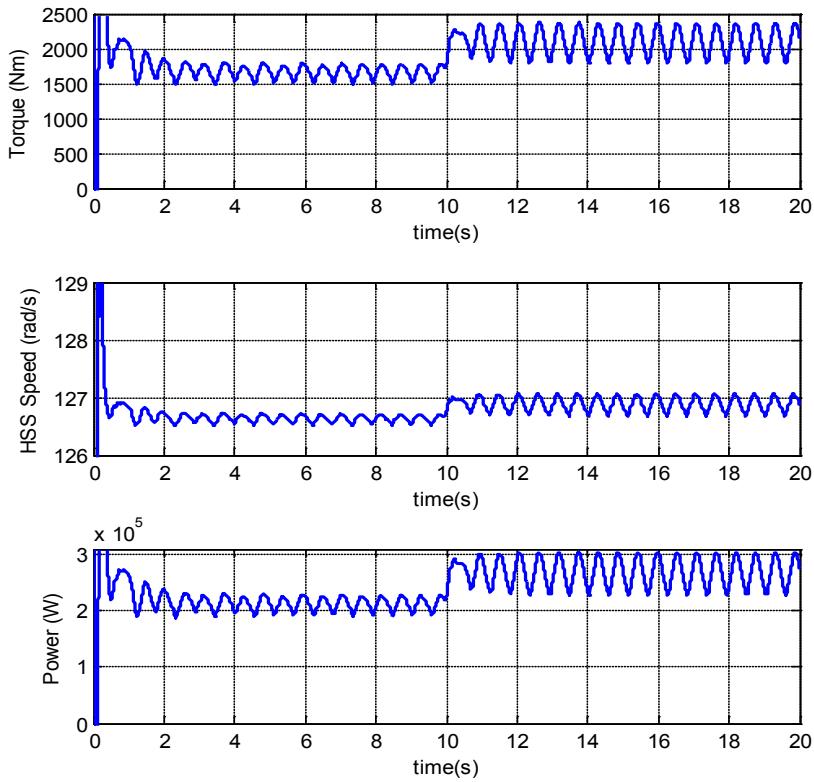


Figure 22. Simulation results using the Dynamic Induction Machine Model

To speed up the simulation, two small changes need to be made. In the *abc2qds* block, disconnect the last input (i.e., the non-voltage input that receives the current sum). Instead, connect a Constant block with value of 0. This approximation is valid under

almost all conditions, except for severe voltage unbalance. This removes an algebraic loop from the simulation, allowing the use of a fixed-step ode4 solver. Also, open the *.fst* input file and change the time-step DT to 0.0005 seconds. Find the following line in *Test01B.fst*:

0.004 DT - Integration time step (s)

and change it to:

0.0005 DT - Integration time step (s)

Upon re-running the simulation, the results should be identical to the ones shown in Figure 18, but the simulation should complete faster (e.g., it took 10 to 15 minutes using our computer).

The Figure 22 results showed some differences from those shown in Figure 18. The torque and power were fairly similar in the two sets, but the speeds differed. This is because the more-complex induction machine model had a different (steeper) torque-slip characteristic than that of the simple linear model employed to obtain the results shown in Figure 18. If the reader closely observes the waveforms at the instant of the wind speed change ($t = 10$ s), additional dynamic behavior from the complex model can be observed in the results.

With the completion of these steps, we successfully interfaced FAST with a dynamic model of an induction generator. We can now observe how the electrical system interacts with the FAST model. In Section 6, we discuss the effects of voltage sags or faults at the machine terminals of this WTG.

4.1.3 Addition of Pitch Controller

This section describes the inclusion of a pitch controller. The rotor power coefficient C_P determines the proportion of available aerodynamic power a turbine can extract. This value depends on the blade pitch angle. Rotating each blade about its longitudinal axis changes the pitch angle, modifying C_P , and thus changing the power extracted from the wind. An example of relation between C_P and tip speed ratio for different pitch angles is shown in Figure 23. Blade pitching can be achieved precisely and quickly with the use of electric servomotor controls, allowing smooth control of the output power. Most wind turbines rated above 0.5 MW employ the pitch control method for power regulation. Another purpose of the pitch controller is to prevent the aerodynamic input power and torque from exceeding the ratings of the electric and mechanical equipment at wind speeds higher than rated wind speed.

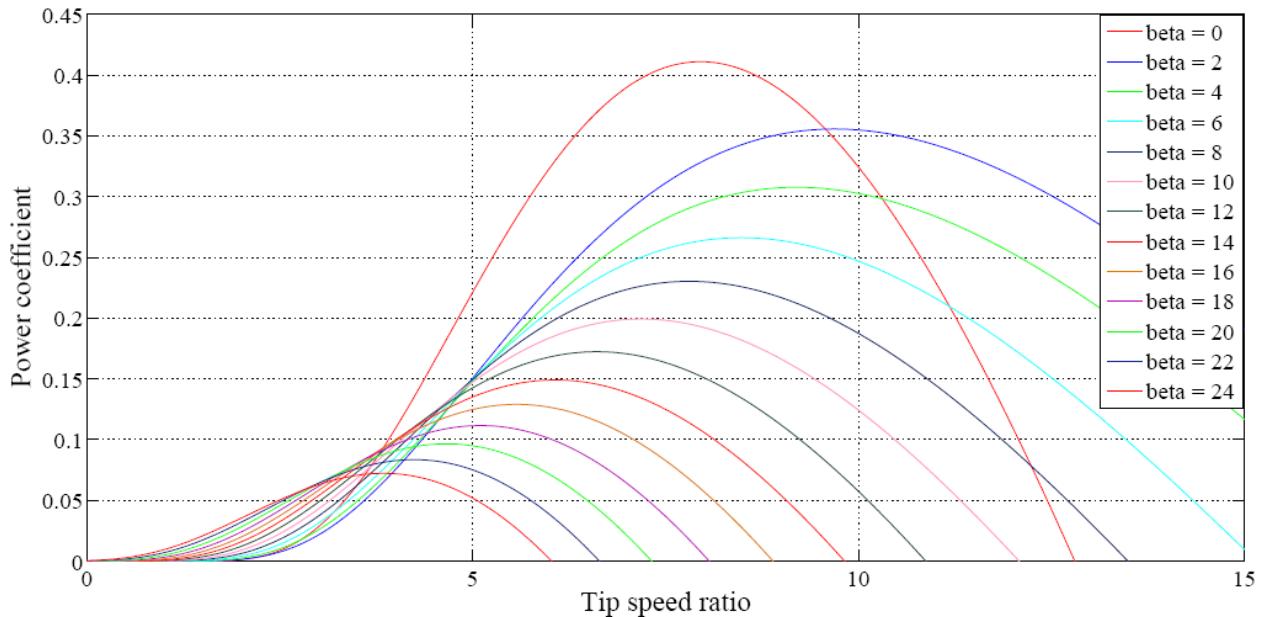


Figure 23. Change in CP curves with change in pitch angle (beta)

There is a nonlinear relationship between the blade pitch angle and rotor power coefficient, and any controller design must take this into account. Blade pitch angle actuators must also be able to contend with dynamic torques acting on the turbine blades while pitching them. We implemented a simple pitch controller in Simulink that uses power and speed inputs to set an appropriate blade pitch angle.

The FAST block in Simulink allows users to develop their own pitch controllers, which provide the pitch angle command to FAST through the specified input port, as shown in Figure 24. An $N \times 1$ vector of pitch angle inputs is required, where N is the number of blades a turbine employs; in the case of AWT27, $N = 2$. With the configuration at this point, the pitch angle command supplied from Simulink is ignored by FAST. To use the command, the user will have to edit the *fst* input file to change the pitch control mode. In *Test01B.fst*, find the following line:

0 PCMode - Pitch control mode {0: none, 1: user-defined from routine PitchCntrl, 2: user-defined from Simulink}
(switch)

Change the PCmode parameter value from 0 to 2:

2 PCMode - Pitch control mode {0: none, 1: user-defined from routine PitchCntrl, 2: user-defined from Simulink}
(switch)

After this change has been made, the FAST Simulink block will begin using the pitch controller's command from Simulink.

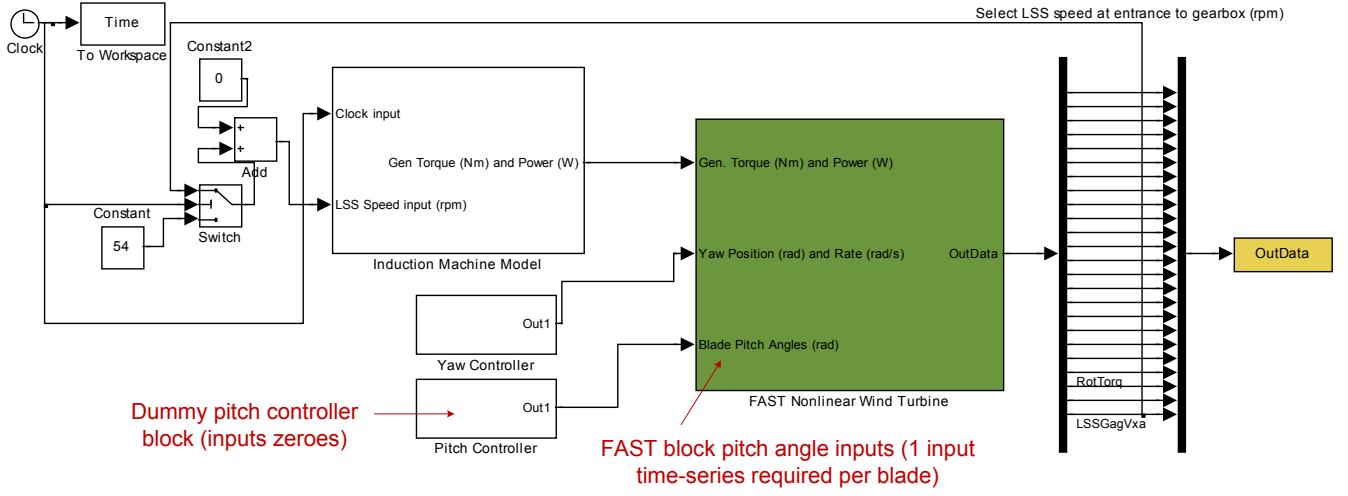
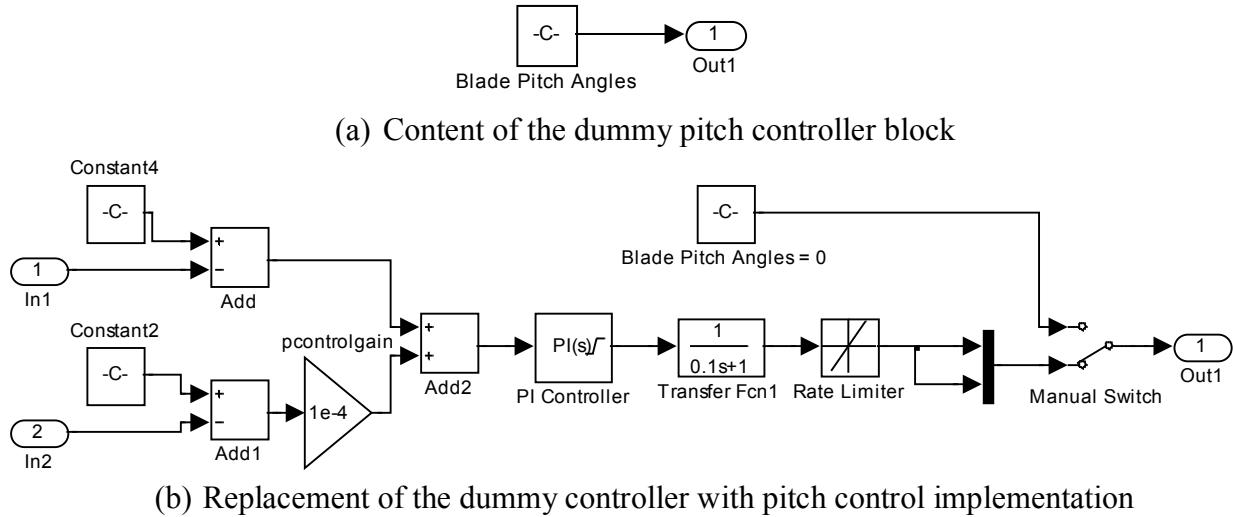


Figure 24. Pitch control inputs to FAST block and dummy pitch controller

Next, double-click the dummy pitch controller block supplied with the model to modify it. The default settings in the dummy pitch controller block result in an input of a series of 0 pitch angle. The following blocks need to be inputted from the Simulink Library Browser and connected, as shown in Figure 25 (b): two Input ports for the power and speed inputs, two Constant for power and speed references, a Gain, three Additions, a PID Controller, a Transfer Function, a Rate Limiter, a Mux block, and a Manual Switch block.



(b) Replacement of the dummy controller with pitch control implementation

Figure 25. Contents of the pitch controller block before and after modification

Set the Constant block associated with input 1 (i.e., the reference speed) to “ $120*\pi*1.01/3$.” Effectively, this value causes the pitch controller to ensure that the slip stays at approximately 1% of the synchronous speed. The Constant block associated with input 2 (i.e., the reference power) should be set to “ 250000 ” to ensure that output power does not significantly exceed 250 kW. The inputs should be subtracted from these reference quantities, as shown in Figure 25 (b). The power error was multiplied by a gain of $1e-4$ to limit its influence on the controller output. The errors were summed, and the sum was inputted to a PI controller.

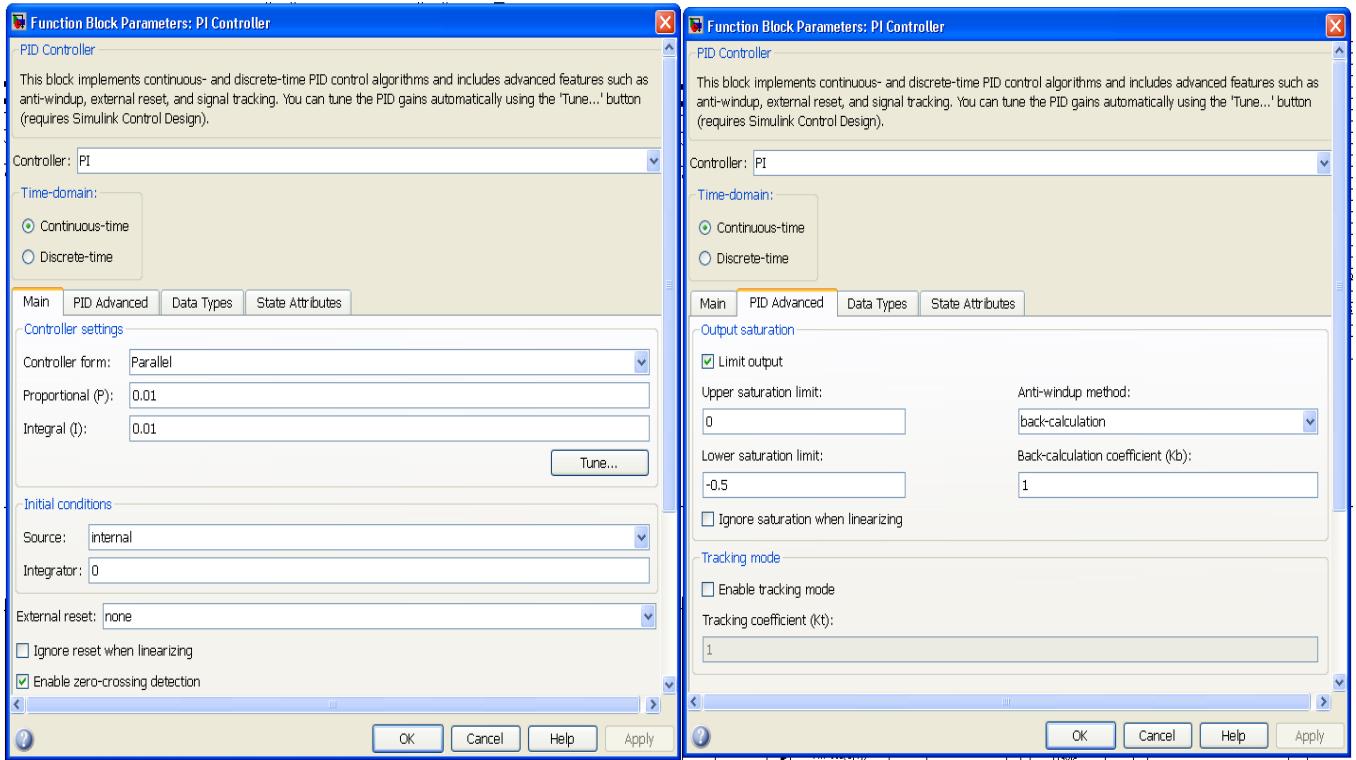


Figure 26. PI controller settings for the pitch controller

Note the default setting of the PID Controller block. The user should change the block to a PI Controller, as shown in the drop-down menu in Figure 26. Gains and limits can also be set as shown in Figure 26. The PI Controller output is smoothed using the Transfer Function block ($\frac{1}{0.1s+1}$) , and fed to a Rate Limiter (for pitch rate limits of +0.1 and -0.1 rad/sec). The rate limiter output is fed to both inputs of a Mux block to generate a 2×1 vector to represent the pitch angle of a two-bladed turbine. Output of the Mux block is fed to one input of the Manual Switch while another input is connected to the preexisting zero Constant block. This way, the user can manually enable or disable the pitch controller by connecting the switch to the Mux or the zero Constant blocks, respectively. In this example, the user should enable the pitch controller.

In the main Simulink window, drag and drop a Gain, a Demux, and a Terminator block. Set the value within the gain block to “ $GBRatio*pi/30$ ”, which will convert the LSS speed in RPM to HSS speed in rad/sec. Connect the input 1 of the Pitch Controller block to the output of this Gain block, and connect the input of the Gain block to the LSS speed signal, as shown in Figure 27. Connect the input 2 of the Pitch Controller block to the lower output of the recently added Demux block. Connect the upper output of the Demux to the terminator; this is a torque signal that is not required. Connect the input of the Demux block to the output of the Induction Machine Model block, carrying the torque and power signals to FAST, as shown in Figure 27. In this example, the simulation run time in *Test01B.fst* (parameter of TMax) should be changed to 40s.

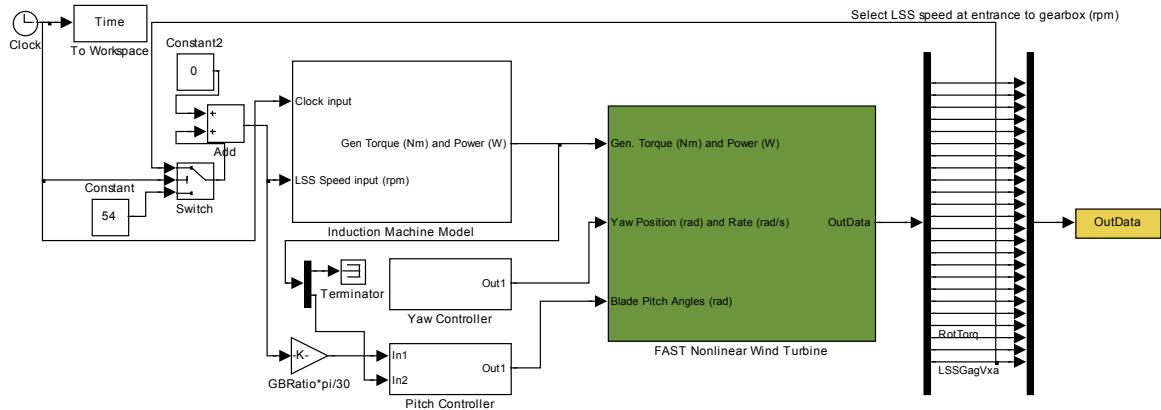


Figure 27. Connections in the main Simulink window

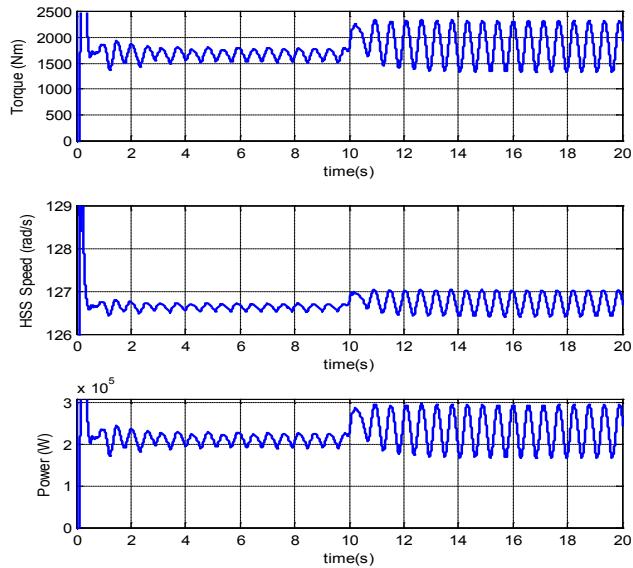


Figure 28. Results with pitch controller enabled

Now, with the pitch controller implemented, run the simulation in a similar fashion, as before. A scope should be connected to the Pitch Controller's output. After the simulation, the results should agree with those shown in Figure 28. Note that there was no jump in HSS speed at $t = 10$ s as there was in the examples shown in Figure 22 or Figure 18. Also, note the output of the pitch controller shown in Figure 29. The pitch angle is shown in degrees. The pitch controller was initially active during startup before reaching a steady-state value of 0. When the wind speed changed at $t = 10$ s, the controller became active and moved to a negative value to spill the excess aerodynamic power and maintain a slip of 1%. It stabilized at an average value of -1.5° , but oscillated because of the tower shadow effect.

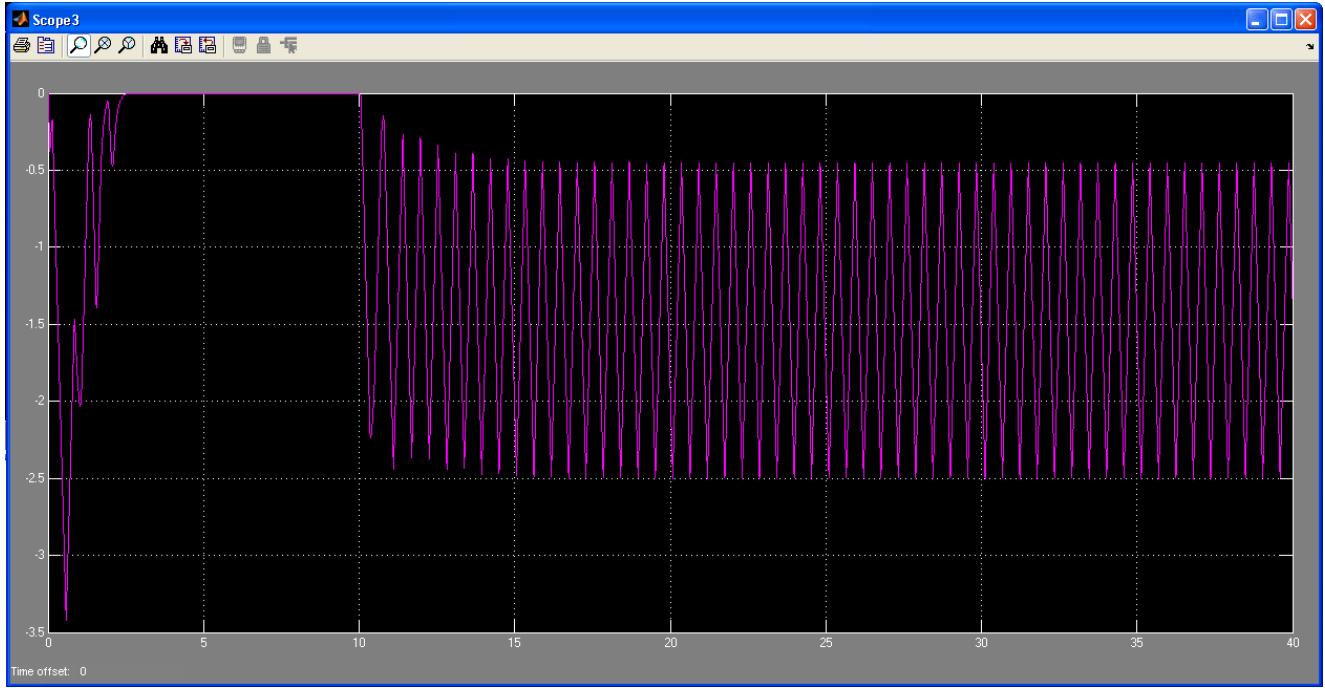


Figure 29. Pitch controller output

Comparing the results shown in Figure 29 to those shown in Figure 22 demonstrates that the speed change and output power change were limited by the Pitch Control action. This verifies that the pitch controller implementation is functional. The user may modify the settings within the Pitch Controller block and observe the effects in the response. The next section describes the modification of the Type 1 turbine model with the addition of a rotor resistance controller to create a Type 2 model.

Dynamic model of Type 1 turbines using the SimPowerSystems toolbox in Simulink have also been developed. Despite having less utility for academic purposes than the aforementioned model, because the machine characteristics are hidden, these models are more useful for engineers because they can be coupled with grid and other power system device models built in SimPowerSystems. The performance of these models is identical to that of the model described above. The Type 1 Turbine SimPowerSystems model is shown in Figure 30.

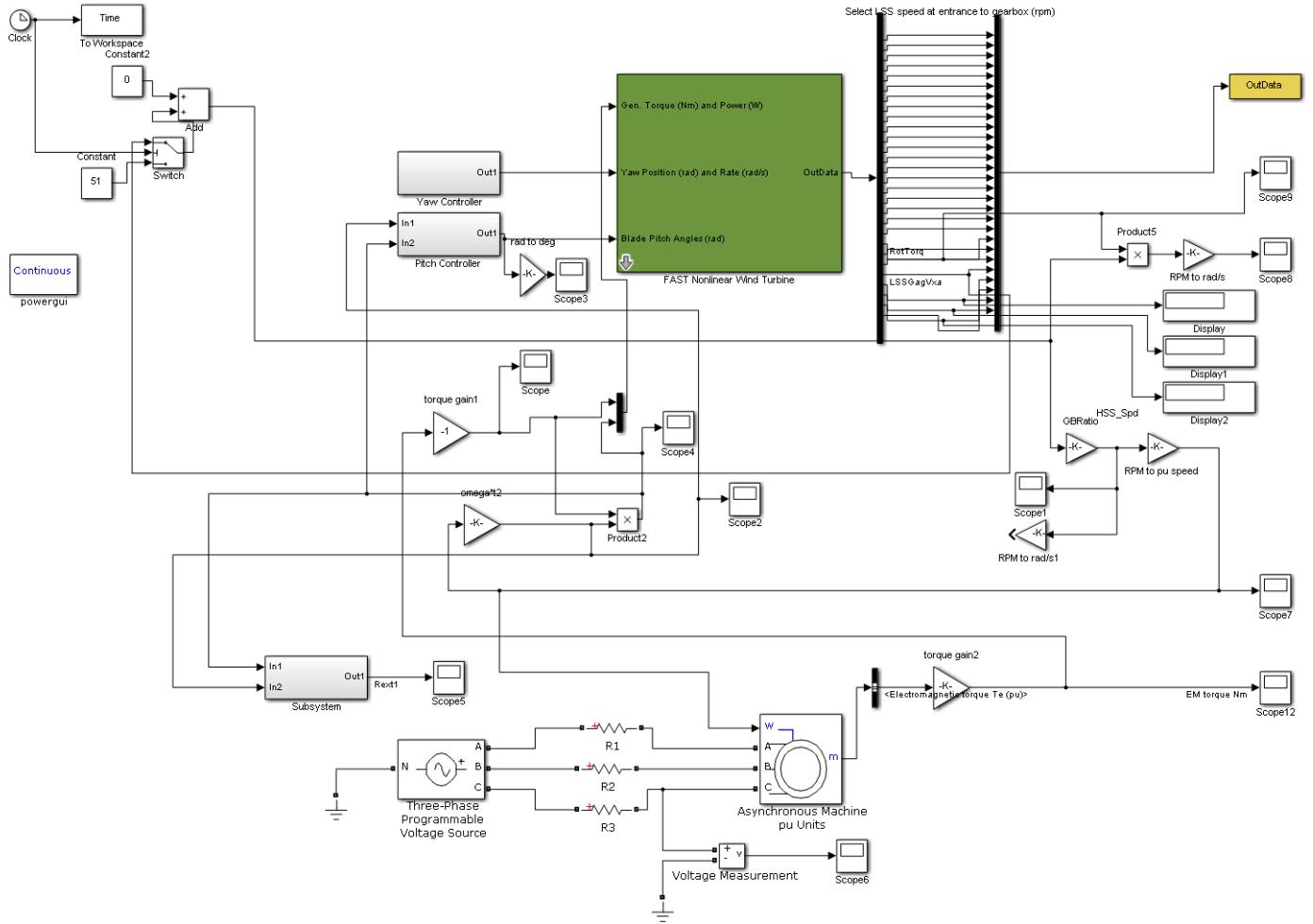


Figure 30. Type 1 turbine SimPowerSystems model

4.2 Type 2 Wind Turbine Model

As previously mentioned, Type 1 fixed-speed turbines, although simple, have disadvantages. The very narrow allowable speed range causes high torques acting on the mechanical and electrical components every time the wind speed changes. Type 2 turbines, also known as variable-slip turbines, seek to rectify the shortcomings of Type 1 turbines. Variable-slip turbines employ wound-rotor induction machines. Brushes and slip rings are used to access the rotor windings. To achieve the desired output power, the torque-slip (and consequently the power-slip) curve of the machine is modified by varying the effective external rotor resistance. The higher the external rotor resistance, the higher the slip.

The use of a mechanically-variable resistor in each phase is impractical, slow, and expensive; therefore, a power electronic AC/DC converter and a static resistor are employed to achieve the same effects. The converter typically consists of a three-phase diode bridge rectifier and an insulated-gate bipolar transistor (IGBT) chopper. The diode bridge rectifier consists of six diodes (three legs with two diodes each) and converts the three-phase AC voltages at the rotor terminals to a DC voltage. The AC/DC conversion is employed because it reduces the number of required IGBTs from three (one for each phase) to one. The IGBT is employed on the DC side of the

diode bridge rectifier. When the IGBT is in the on state, it shorts the rotor circuit, reducing external rotor resistance to near zero. When it is in the off state, the external resistance is not bypassed and forms a part of the rotor circuit. By varying the duty cycle of the IGBT switching, the effective rotor resistance of the machine can be varied. The effective rotor resistance is a value between zero and the fixed value of the external resistor. The higher the duty cycle, the lower the effective external resistance is. A detailed explanation of the effects of external resistance on the torque-slip characteristics of the wound-rotor induction machine, and a controller to change external resistance, are described in the following subsection.

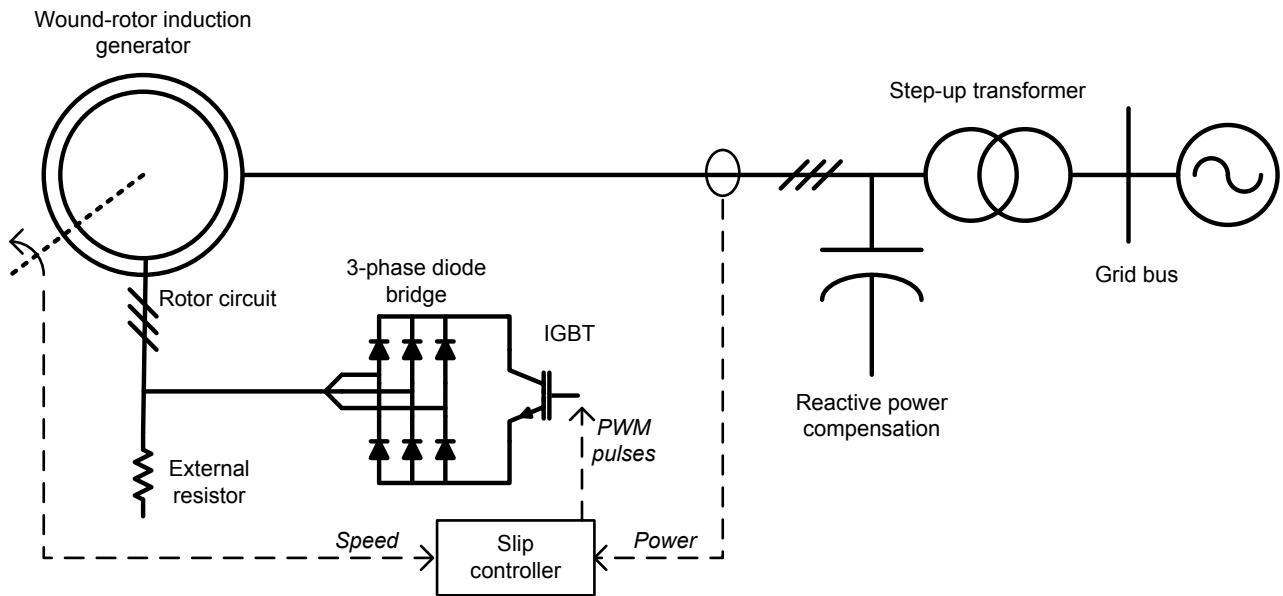


Figure 31. Power converter for external resistance control in variable-slip turbines

The effective external resistance can be varied smoothly to allow the machine to operate at different operating speeds for the same output power. This flexibility reduces the strain on the gearbox and allows maximum power point tracking (i.e., maximizing aerodynamic power extraction from the wind). However, some portion of the extracted energy is lost in the rotor resistances as heat. Also, reactive power compensation is still required in variable-slip turbines.

4.2.1 Rotor Resistance Control Concept

The equivalent circuit of a wound-rotor induction machine with external rotor resistance is shown in Figure 32. The role of the rotor circuitry described in Figure 31 is to achieve variable resistance in the rotor circuit.

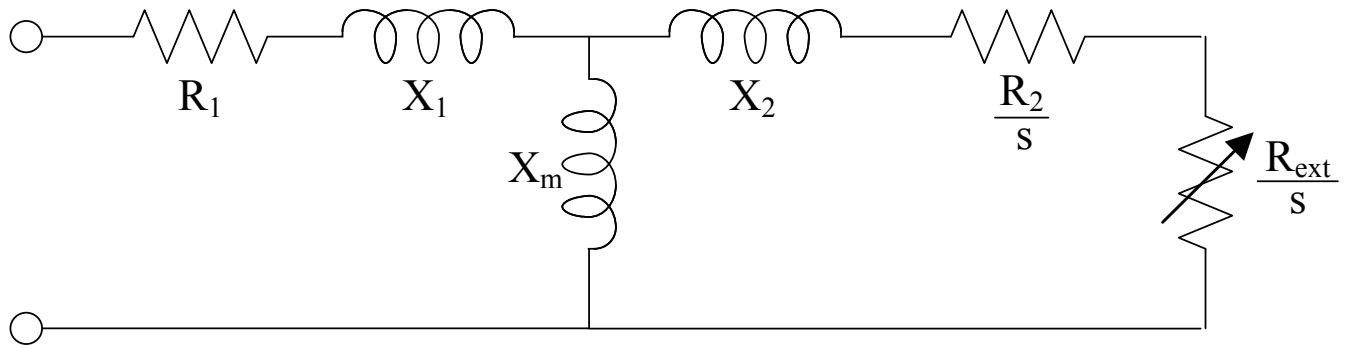


Figure 32. Induction machine equivalent circuit with external resistor present

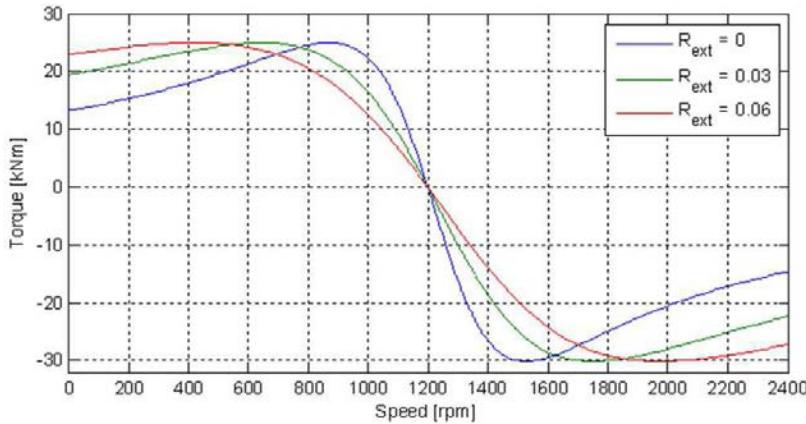


Figure 33. Example torque-speed curves with different values of external rotor resistance R_{ext} (expressed per unit of internal rotor resistance R_2)

A variable resistor is present in each phase because the equivalent circuit represents each phase of a balanced three-phase circuit. A desired torque value can thus be achieved at many different speeds by varying the external rotor resistance, as shown in Figure 33. The model described here lumps the two resistances R_2 and R_{ext} into one combined rotor resistance R_{rotor} . We did not explicitly model the power electronics or resistances, but rather calculated a value of the resistance and input this value into the model.

4.2.2 Implementation

In our implementation, we attempted to deliver constant effective rotor resistance, thus constant torque, within a given range of rotational speed. This may be expressed by the equation $\frac{R_2}{s_{rated}} = \frac{R_2 + R_{ext}}{s_{new}} = \frac{R_{rotor}}{s_{new}}$. The external resistance value was chosen such that, whatever the new value of slip was, the effective rotor circuit resistance remained the same. The user will need to make some modifications to input a rotor resistance value to the induction machine model. These modifications involve replacing all constant rotor resistance values rpr (see *SimsetupB.m*) with variable input. Consider the diagram of the induction machine model shown in Figure 19 (i.e., within the “Induction Machine Model” subsystem). Note the subsystems labeled “Qaxis” and

“Daxis.” Double-click the Daxis subsystem. The contents of the subsystem are shown in Figure 34.

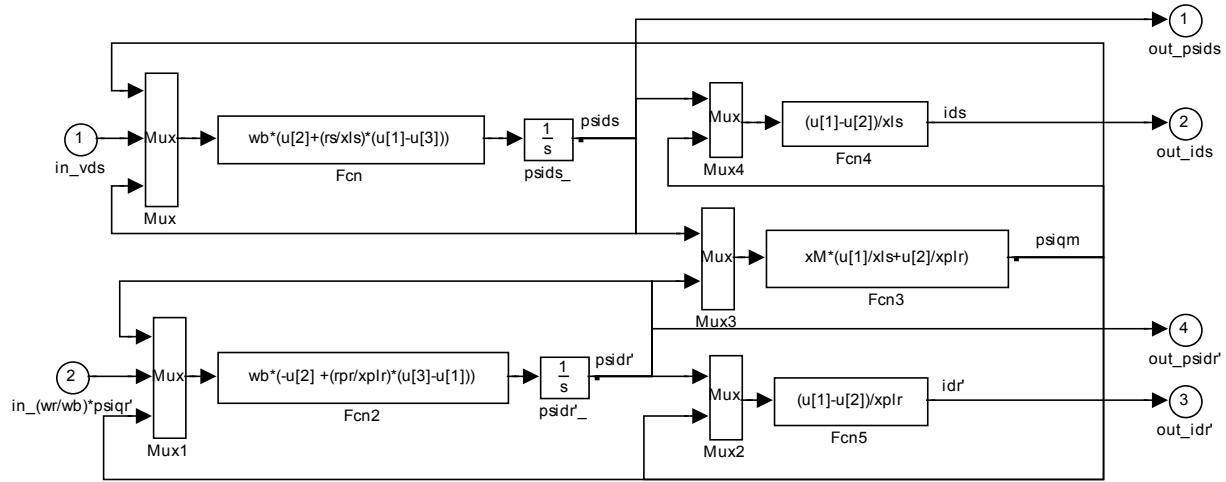


Figure 34. Original contents of “Daxis” subsystem

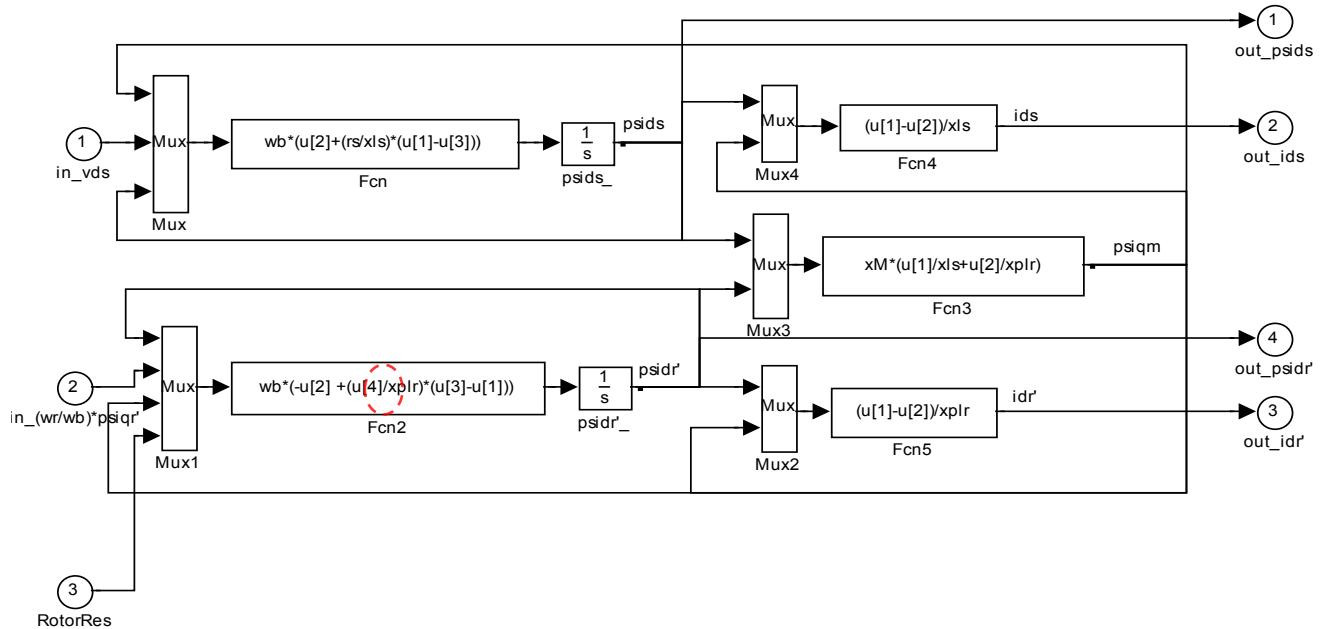


Figure 35. Modified contents of “Daxis” subsystem

Note that the expression of *Fcn2* block, following the *Mux1* block, makes use of the constant *rpr*. Each of the multiplexed signals is represented by *u[1]*, *u[2]*, and *u[3]*. A fourth signal, *u[4]*, needs to be added to replace the *rpr*. To do this, double-click on the *Mux1* block and change the number of inputs from three to four. A fourth input port will appear. Copy the input port 2 and paste it back in. It will create the input port 3. Modify the label of the input port 3 to “RotorRes.” Connect this input port to the fourth input of *Mux1*. Double-click the *Fcn2* block and replace the string “*rpr*” with “*u*” [4]. The modified diagram is shown in Figure 35. An identical process must be followed with the “Qaxis” subsystem, as shown in Figure 36 and Figure 37.

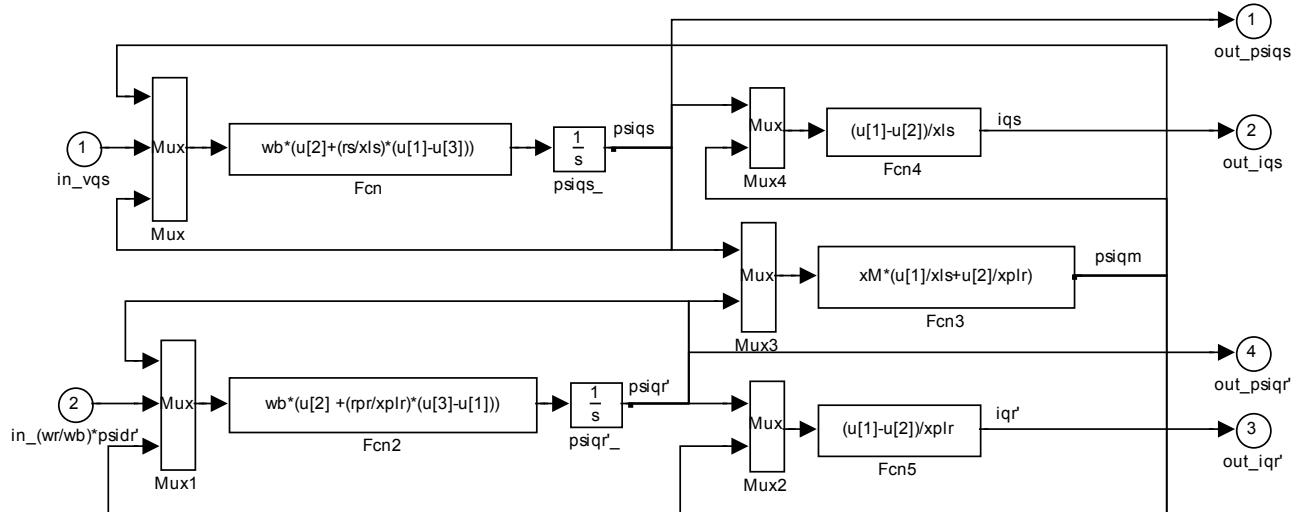


Figure 36. Original contents of “Qaxis” subsystem

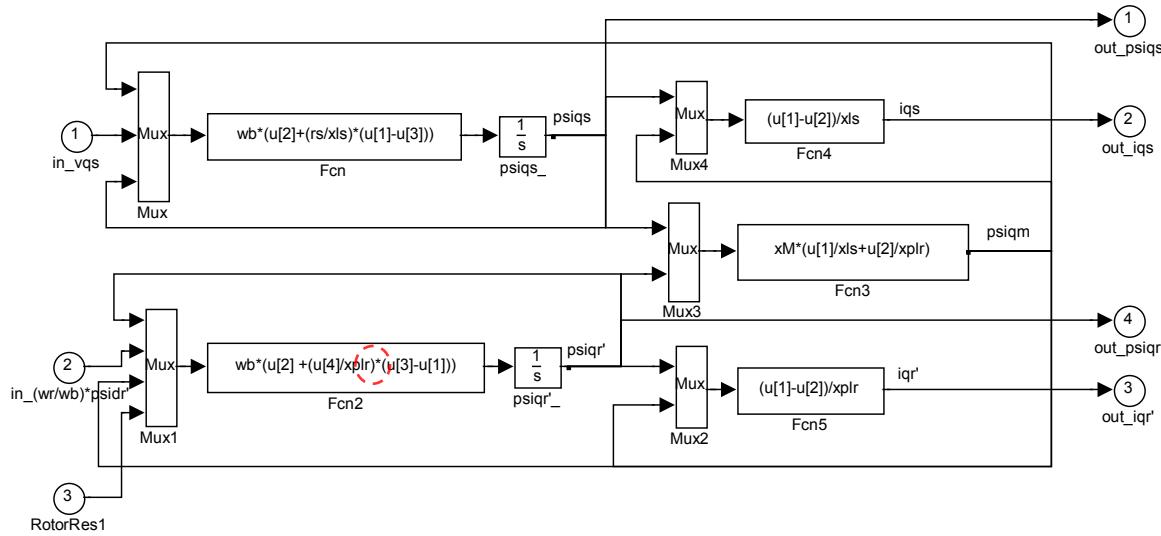


Figure 37. Modified contents of “Qaxis” subsystem

With these modifications, the “Daxis” and “Qaxis” subsystems will each have an additional input port for the R_{rotor} value. A controller needs to be developed to generate this resistance value. Double-click the Induction Machine Model subsystem. From the Simulink Library Browser, drag and drop a new Subsystem block into this subsystem. Label this subsystem “RotorResCtrl.” Double-click the “RotorResCtrl” subsystem. It will contain one input port connected to one output port. Delete the connection between them, copy input port 1, and paste it back in to obtain the input port 2. Label input 1 as “HSS Speed (rpm)” and input 2 as “Power (W).” Label output 1 as “Rrotor.” In the Induction Machine Model subsystem, connect the output of the “RotorResCtrl” subsystem to the free inputs of the “Daxis” and “Qaxis” blocks. Connect the input 2 of the “RotorResCtrl” (i.e., the power) to the output power from the induction machine

and the input 1 (HSS speed in rpm) to the output of the “GBRatio” gain block, as shown in Figure 38.

The content of the “RotorResCtrl” can now be configured, as shown in Figure 39. The user can obtain the required blocks from the Simulink Library Browser. The control technique in this subsystem is to calculate a rotor resistance value using the speed input and adjust it within a small range using the power input. We first converted the speed input from rpm to rad/sec, then used it to calculate slip $s_{new} = \frac{\omega_{new} - \omega_{syn}}{\omega_{syn}}$. This slip value was used to calculate the desired rotor resistance in Ω as $R_{rotor} = R_2 \frac{s_{new}}{s_{rated}}$, then added to the output of the PI Controller based on the power input and reference power comparison. It is important to note the output of the PI controller is limited to 0.1Ω .

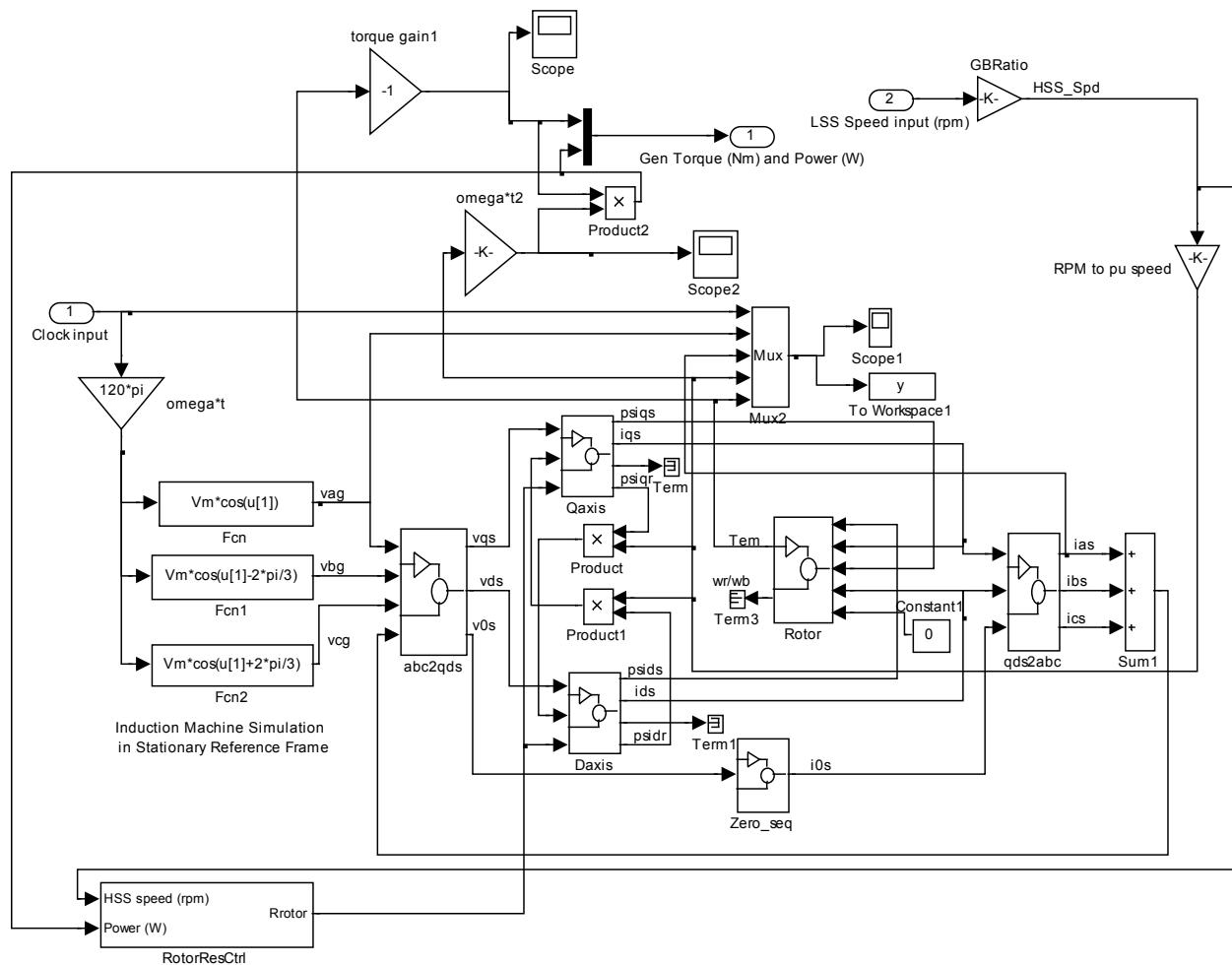


Figure 38. Modified “Induction Machine Model” subsystem with new “RotorResCtrl” subsystem

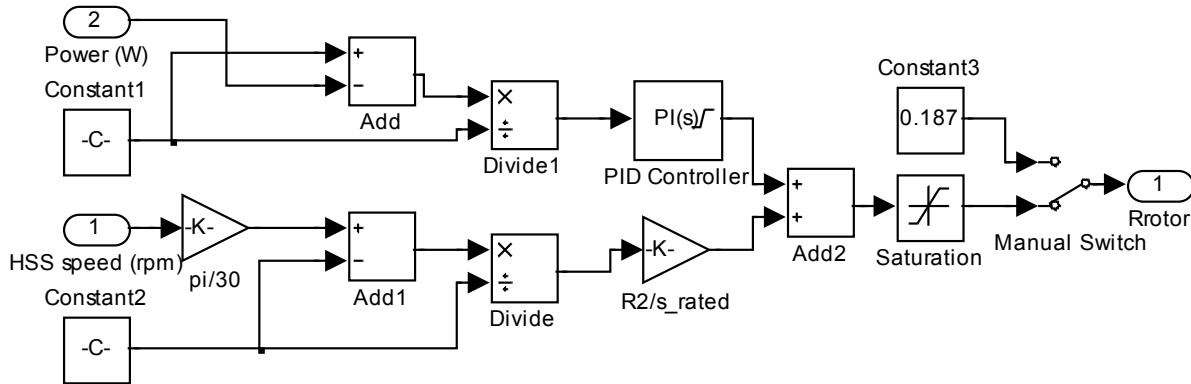


Figure 39. Blocks and connections within the “RotorResCtrl” subsystem

In Figure 39, the value of Constant1 (i.e., the reference power in Watts) was set to “223000”; whereas the value of Constant2 (i.e., the mechanical synchronous speed in rad/sec) was set to “ $2*60*\pi/3$,” because the frequency was 60 Hz and there were three pole-pairs (six poles). The Gain block directly after the speed input was set to “ $\pi/30$ ” to convert HSS speed in rpm to rad/sec. The Gain block “ R_2/s_{rated} ” was set to a value of “ $0.187/0.0095$,” because 0.187Ω was the value of R_2 (i.e., the rpr) and 0.0095 was the rated slip for this machine. Gains and limits for the power PI Controller block are shown in Figure 40. A saturation block was placed to limit the maximum value of Rrotor between rpr (0.187Ω) and 2 Ω . A manual switch was provided to enable or disable the controller. If it is desired to disable the controller, the user can switch the output to a constant value of 0.187Ω , ensuring that the simulation runs as if the controller were not present and rotor resistance were $rpr = 0.187 \Omega$ as before. In this example, the controller should be enabled at this time.

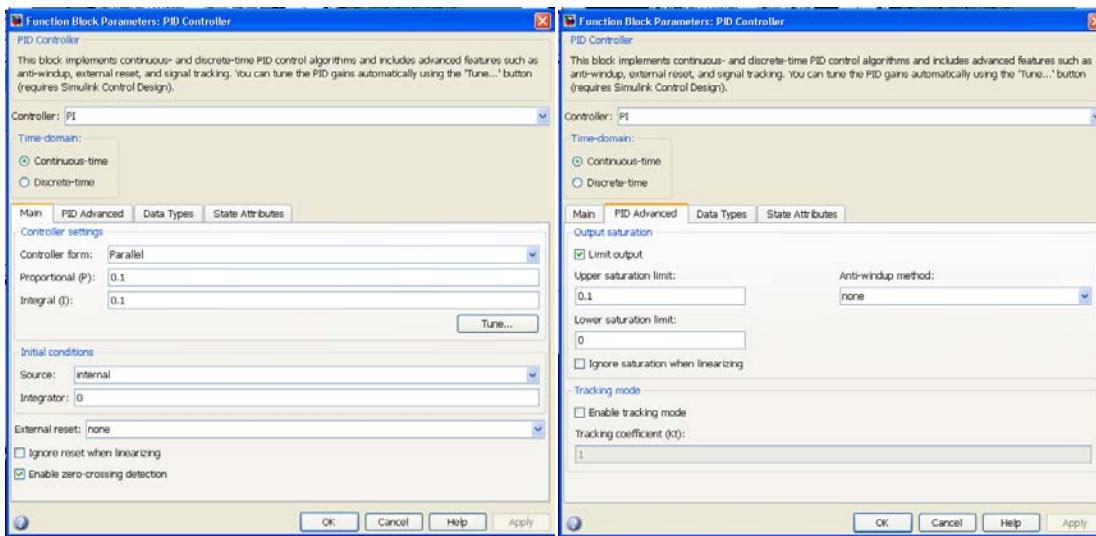


Figure 40. Gains and limits for power PI controller in “RotorResCtrl” subsystem

One more modification is required, this time to the pitch controller block (see Figure 25 (b)). Because a higher slip can be allowed now, we changed the Constant4 (i.e., the reference value associated with the speed input 1) from “ $120*\pi*1.01/3$ ” to “ $120*\pi*1.06/3$,” allowing a speed change of up to 6% before the pitch controller was activated. A scope can be added to monitor

the value of rotor resistance (i.e., the output of the “RotorResCtrl” subsystem). The simulation can now be run.

4.2.3 Type 2 Turbine Model Results

The simulation results are shown in Figure 41–Figure 43. Figure 42 shows the value of rotor resistance in Ω . After the initial transient, the rotor resistance reached steady-state value at the original value of 0.187Ω , until at $t = 10$ s, when the wind speed change occurred. Then, the rotor resistance began to increase, and it eventually stabilized at approximately 0.8Ω . Note the oscillations in the rotor resistance. Again, these were caused by the torque pulsations arising from the tower shadow effect. The rotor resistance controller effectively damped out the torque pulsations, smoothing the torque waveform.

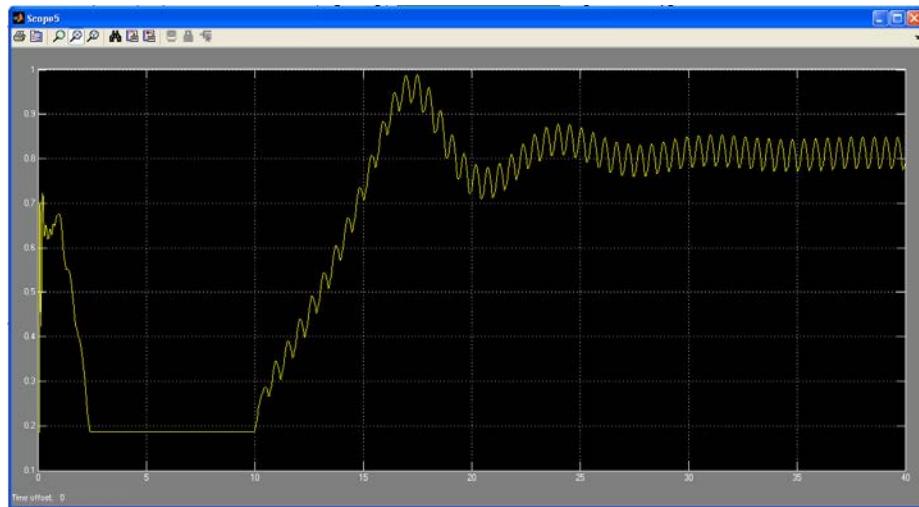


Figure 41. Rotor resistance variations

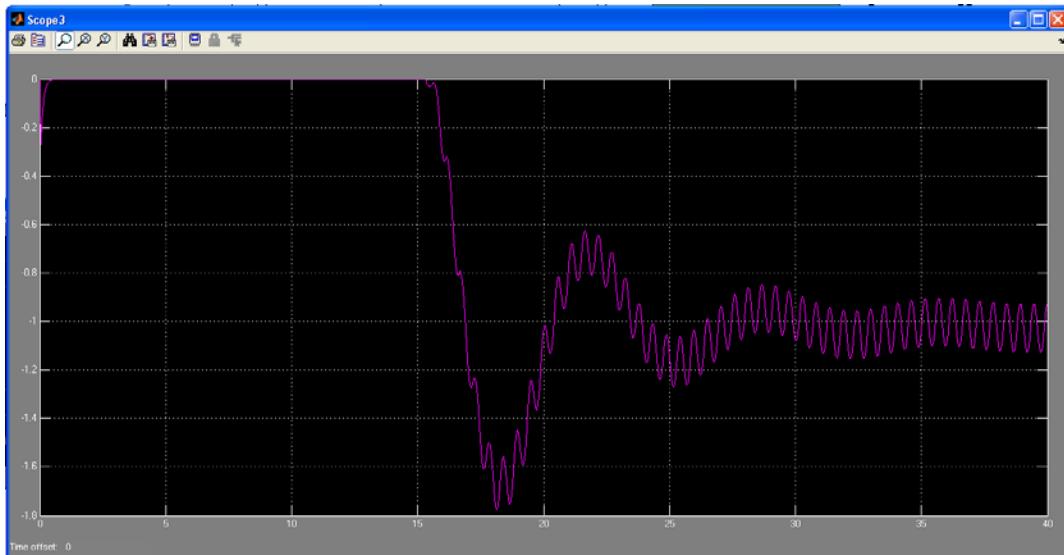


Figure 42. Pitch angle variations with rotor resistance controller present and enabled

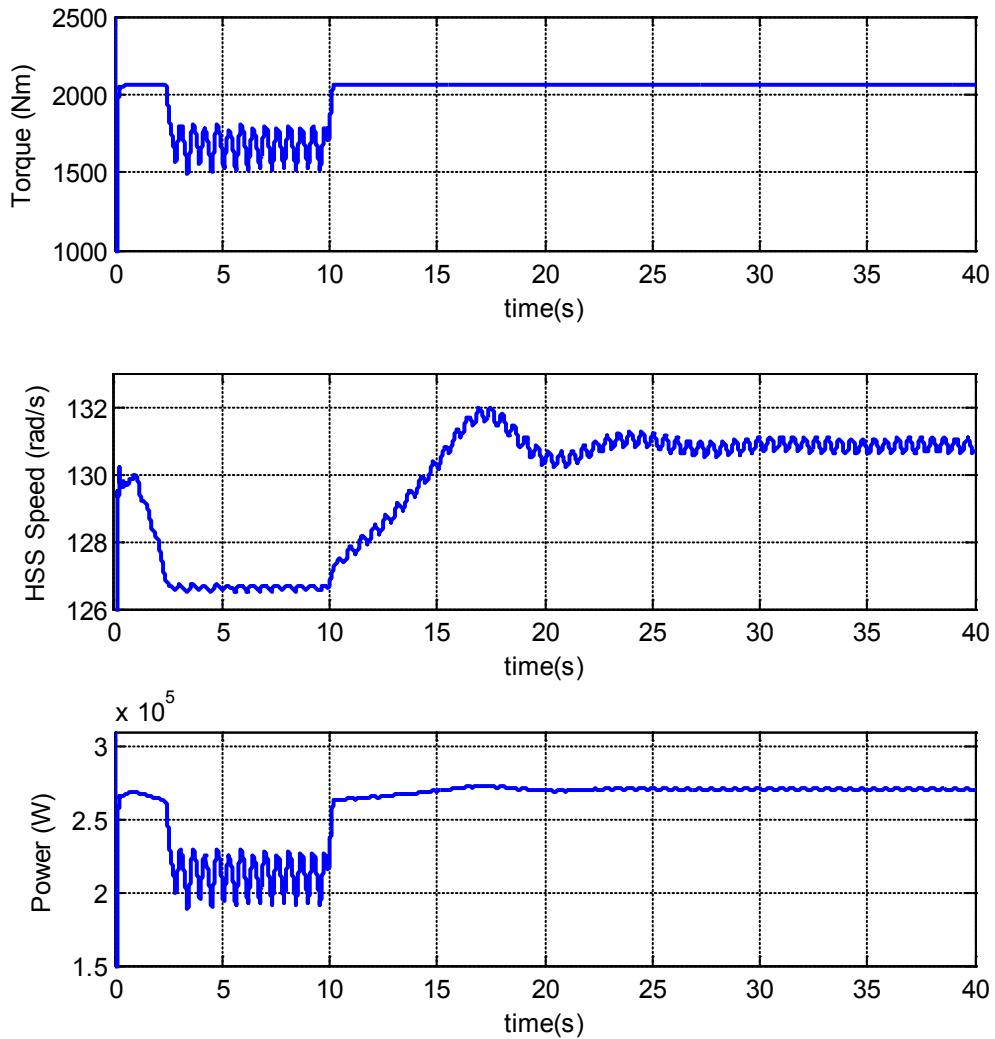


Figure 43. Results with pitch and rotor resistance controller present

Because of the damping effect from the rotor resistance controller on the torque oscillations, the output power oscillations were also damped. Also, the pitch controller output shows that the oscillations were smaller than those shown in Figure 29. The speed variation was larger, with an observed maximum speed variation (i.e., slip) of approximately 6%. Allowing this speed variation by changing the rotor resistance smoothed the torque and power waveforms. These less-oscillatory conditions are much friendlier to the mechanical and electrical components of a turbine. This is one of the primary reasons variable-speed turbines are preferred in the real world. The implemented rotor resistance controller was proven effective to modify the Type 1 to Type 2 turbine model.

Dynamic Type 2 wind turbine models have also been developed using the SimPowerSystems toolbox in Simulink. Despite having less utility for academic purposes than the aforementioned model, because the machine characteristics are hidden, these models are more useful for engineers because they can be coupled with grid and other power system device models built in SimPowerSystems. The performance of these models is identical to that of the model described

above. An example of a Type 2 SimPowerSystems model is shown in Figure 44. In this model, the effective resistance of a single external resistor is modulated using an IGBT that has a duty cycle, which was the control variable. The IGBT was switched using a rotor resistance control command overlaid onto a PWM signal. Models of other external rotor resistance control methods have also been developed.

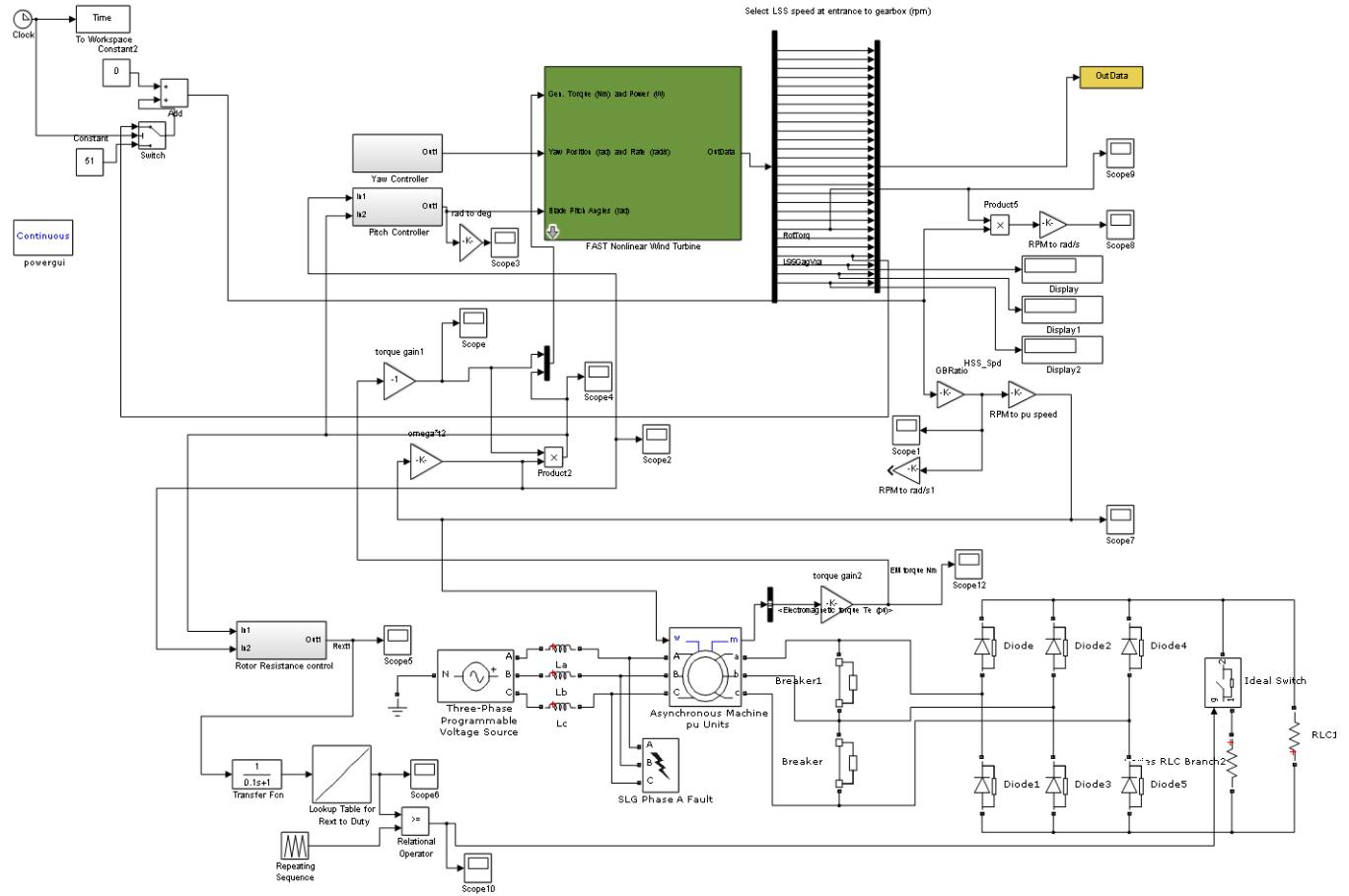


Figure 44. Type 2 wind turbine model using SimPowerSystems

4.3 Type 3 Turbine Model—SimPowerSystems

A Type 3 (DFIG) turbine model has also been developed using the MATLAB SimPowerSystems toolbox. This model is a phasor model [7]; it treats the power system as a balanced three-phase fixed-frequency network in which each phase voltage is identical in magnitude but out of phase by 120 degrees. Phasor simulation replaces the differential equations representing the network with a set of algebraic equations at a fixed frequency. Phasor simulation facilitates transient stability studies of systems with multiple machines. Phasor simulations, also known as positive sequence simulations, cannot be used to study unbalanced events. This model is better adapted to simulate the low-frequency electromechanical oscillations within seconds to minutes. However, they can also be useful for a variety of other studies. Another technique available to simulate a generator is to use a three-phase representation, in which an unbalanced simulation can be performed. The unbalanced conditions may come from the grid unbalanced voltage (faults, dips,

or other transients) or unbalanced grid impedance. The model developed so far does not account for these factors, but will do so in the future.

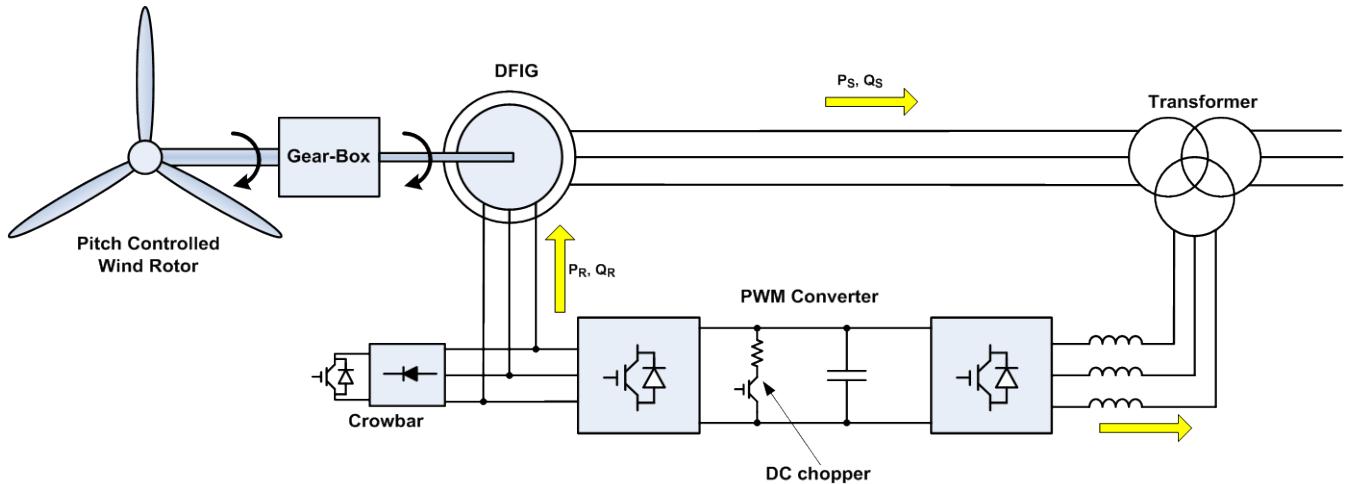


Figure 45. Type 3 wind turbine connection diagram

Type 3 WTGs (as shown in Figure 45) are variable-speed wind turbines with DFIGs. A DFIG is operated in variable-speed mode using a partial-size power converter connected to the rotor winding of the WRIG. The stator winding of the WRIG is connected to the grid at a frequency of 60 Hz. This turbine type is probably the most popular type available in the market, and it has been deployed in large numbers. A WTG is normally operated between 30% slip (i.e., subsynchronous speed) and -30% slip (i.e., supersynchronous speed), and the converter is typically at approximately 30% of rated output power. The power converter performs a back-to-back AC/DC/AC conversion using two pulse-width modulation–switched voltage-source inverters coupled with a DC link. A crowbar circuit is also provided as protection, to allow shorting the rotor circuit, if necessary.

A Type 3 WTG has a torque characteristic that is a quadratic function of the rotational speed. Type 3 WTGs allow maximal extraction of wind power because their output power can be electronically controlled to follow the optimal power curve. The optimal power curve is a cube function of the rotational speed. If the rotor speed exceeds its rated value, the pitch controller must be deployed to limit the rotational speed at its rated speed. If the pitch controller cannot control the aerodynamic power of a wind turbine, a WTG may experience a runaway event. Note that the speed range of a Type 3 WTG is much larger than the speed range of a Type 1 WTG; thus, the kinetic energy stored in the rotating blades and other components within a wind turbine is sufficiently large, and the output of the generator is not impacted as much by the wind fluctuations and turbulence because some of the energy is stored and restored in the kinetic energy of the rotating mass.

MATLAB/SimPowerSystems toolbox provides an example phasor model of a DFIG turbine with highly simplified mechanics. We modified this model and replaced the basic aerodynamic and mechanical aspects with the FAST Simulink block. A top-level view of the model is shown in Figure 46. Considering that the Type 3 WTG is presently the most popular

turbine installed globally, a more detailed description of SimPowerSystems' implementation of a Type 3 WTG is given in Appendix B.

The DFIG (light blue) block model was previously supplied with a torque input, but because FAST handles all the calculations for the two-mass (generator and turbine) shaft model, the generator can be provided directly with a speed input instead. Within the generator block, the two-mass shaft sub-model was bypassed. This generator model does not include a crowbar or DC chopper. A pitch control subsystem not present in the original model was added as well, based on the one designed for the previous turbine models. Some results from this model are provided in Section 6, in which this model was also used to test the efficacy of stress damping controllers.

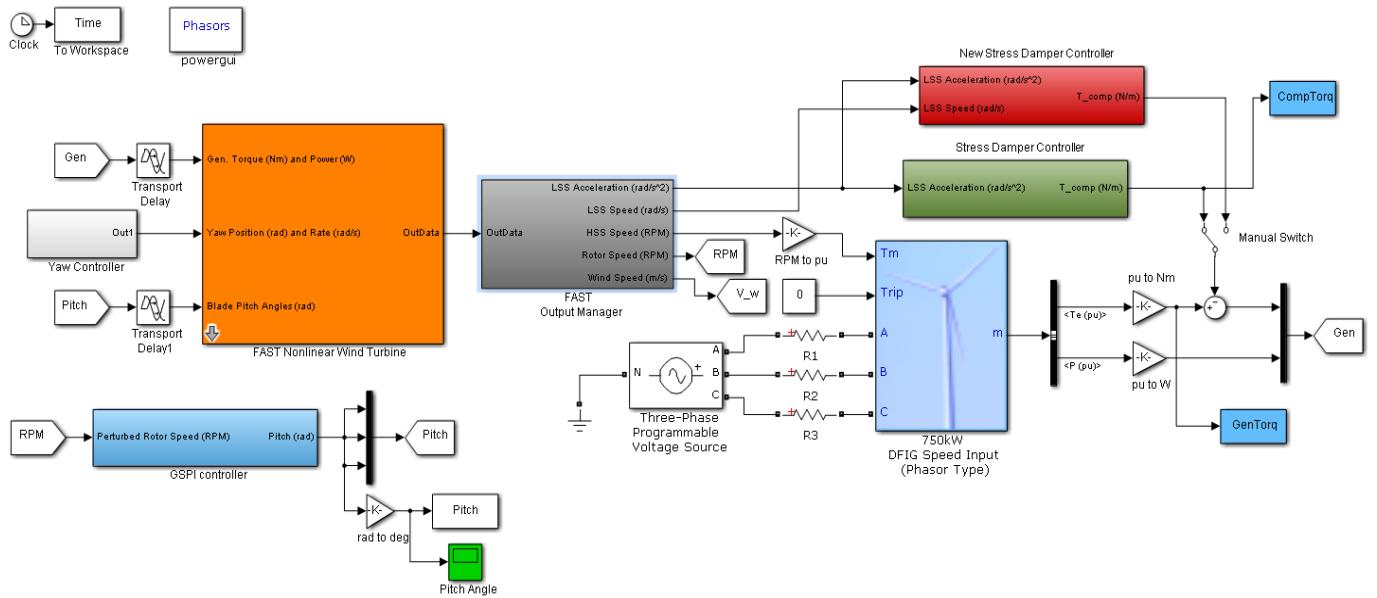


Figure 46. Type 3 wind turbine model using SimPowerSystems

4.4 Type 4 Turbine Model—SimPowerSystems

Type 4 WTGs are variable-speed wind turbines with a generator connected to the grid via the AC-DC-AC power converter. The generator is connected to the machine-side converter, and the grid is connected to the grid-side converter. The generator is operated at variable speed to optimize the aerodynamic operation. The grid-side converter is capable of producing real and reactive power instantaneously and independently. The real output power to the grid is usually adjusted to balance the energy in the power converter by controlling the voltage across the DC link. The reactive output power to the grid is usually controlled either at constant power factor, constant reactive power, or constant voltage. The mechanical connection from the generator to the slow-speed shaft can be accomplished by the gearbox to match the operating speed of the generator to the rotational speed of the turbine rotor. If a direct-drive generator is used, there is no need for a gearbox, as shown in Figure 47. However, direct drive generators must be designed to operate at low speed to match the rotational speed of the turbine rotor. The low-speed generator usually has a large diameter and consists of many poles.

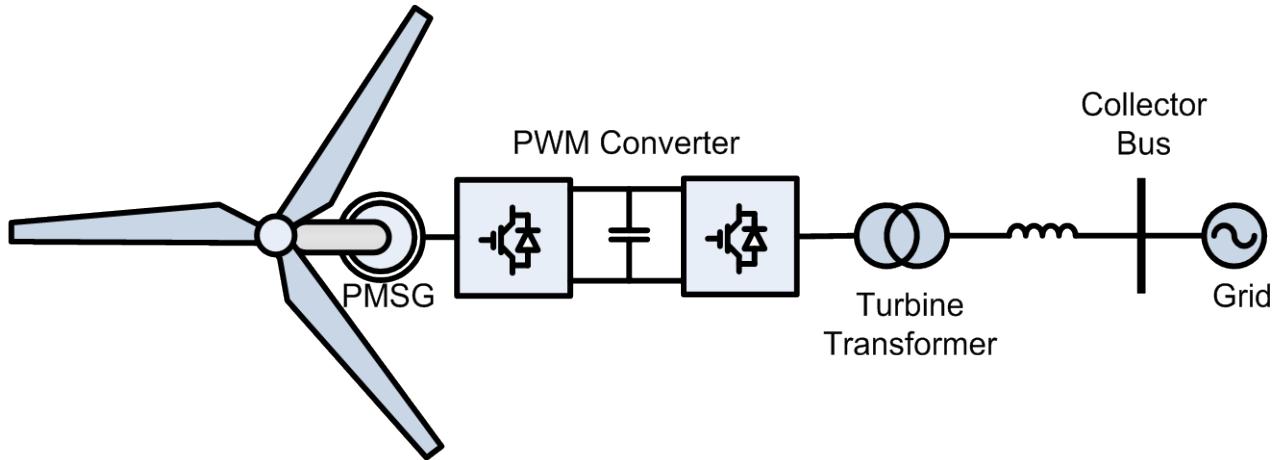


Figure 47. Type 4 wind turbine connection diagram

MATLAB/SimPowerSystems toolbox currently provides an example model for a full-converter turbine, shown in Figure 48. In this model, the aerodynamic, structural, and mechanical aspects of a turbine are modeled in a very approximate manner. Additionally, the electrical topology of this model is nonstandard. Hence, the model may not accurately represent the dynamics of many full-converter turbines in the field because the permanent machine generator's output is directly rectified using a diode bridge, and a DC/DC boost converter is used for maximum power point tracking. Many real-world turbine designs employ a full six-pulse IGBT bridge to provide additional controllability. Additional work is being done to modify this model such that it can represent the dynamics of real-world, full-converter turbines.

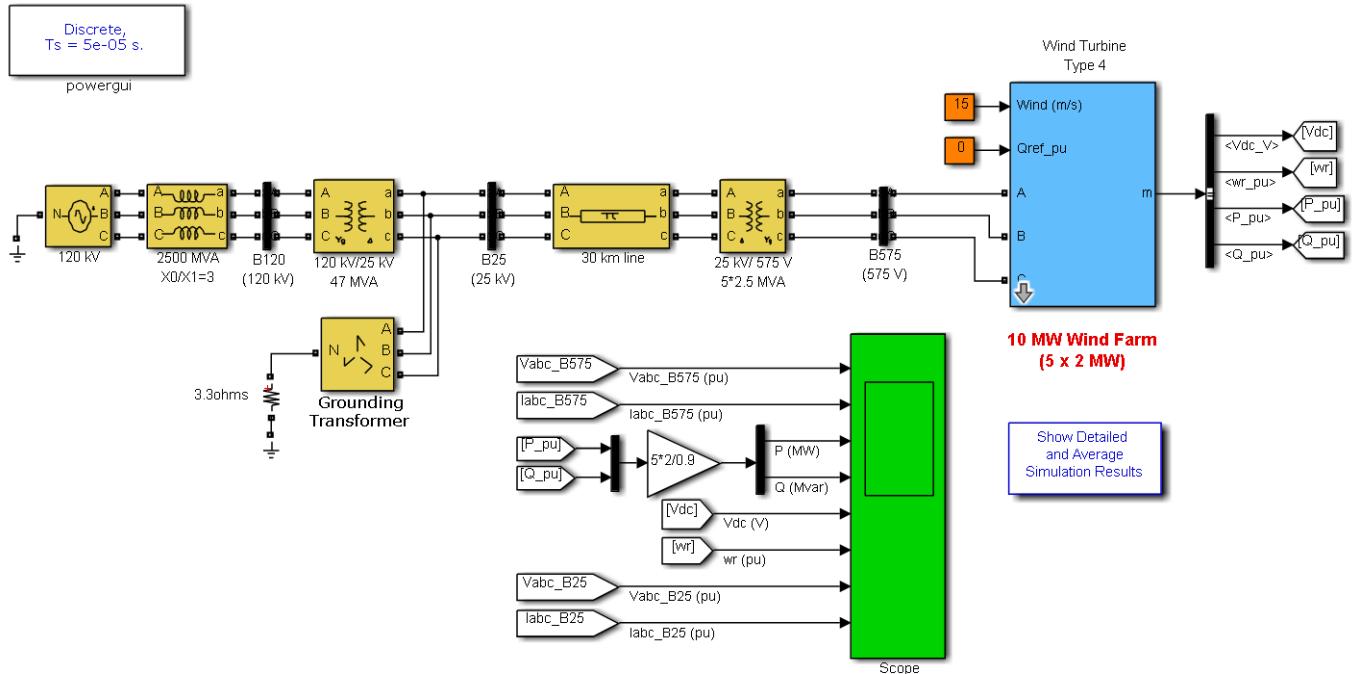


Figure 48. Type 4 turbine model using SimPowerSystems

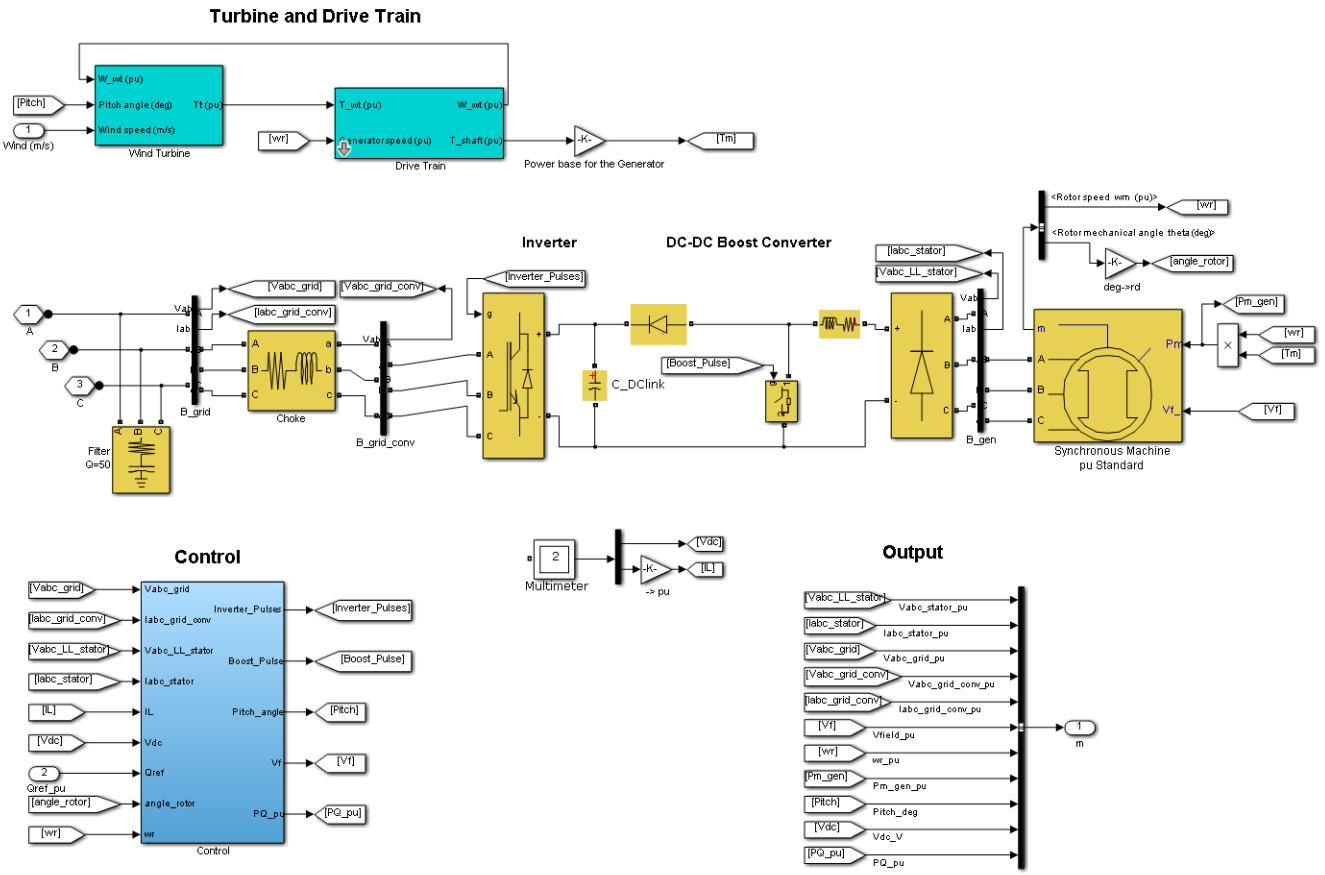


Figure 49. Type 4 turbine model using SimPowerSystems—detailed view of the electrical topology

5 Simulation of Normal and Abnormal Events

5.1 Normal and Abnormal Events

Both normal and abnormal events need to be simulated to study the impact of abnormal events. The former is usually dedicated to establish the baseline, whereas the latter is to simulate the case study of interests depending on the type and scope.

Holistic simulations cover overall wind turbine components, including the pitch controller, blades, shafts, gearbox, generator, and power converter; as well as the overall grid or power system components, including the line impedance, transformer (transformer leakage impedance, magnetic saturation, etc.), and switch gear (circuit breakers, line switches, fault current limiters, capacitor compensations, etc.). The impact of turbines or a wind power plant on the grid can be studied with simulations, together with the impact of the grid on the turbine components. Each topic and scope of investigation depends on the purpose of the study. Wind turbine manufacturers may be interested in studying the impact of transmission events on the turbine components, and transmission operators may be interested in studying the impact of component failures, wind turbulence, or extreme ramp rates on power system stabilities. The examples covered in this report are intended to show some of the applications and capabilities of the program simulations presented in this document.

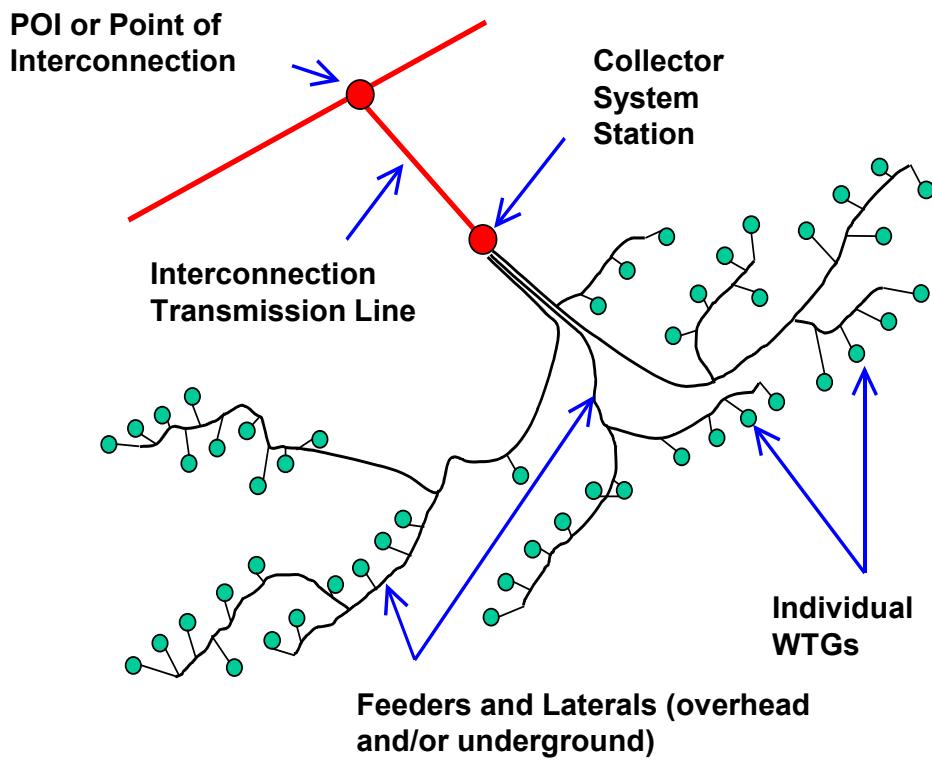


Figure 50. Layout of a typical wind power plant

As shown in Figure 50, a wind power plant consists of many (up to hundreds) wind turbines (1 MW to 5 MW each). Wind power plants are usually located in places with high wind resources and may be far from a load center. Each generator is driven by a turbine when the wind reaches above a certain minimum wind speed (i.e., the cut-in wind speed). The generator philosophy is

generally based on maximizing the energy production (unscheduled operation). Nonetheless, a WTG is controllable, although its controllability is only in one direction—curtailment (i.e., it can only generate less than the available aerodynamic power by a combination of pitch and generator controls). An exceptional case is when the turbine is de-rated, in which case it can be controlled upward as well as downward. A WTG output is also predictable. Wind variability can be estimated based on wind forecasting.

In a conventional power plant, synchronous generators are directly connected to the grid. The electromagnetic flux generated by the stator winding rotates synchronously according to the frequency of the grid. There is a direct correlation between the frequency and voltage of the grid and the mechanical rotor of the generator, which is mechanically and tightly synchronized to the grid. Any oscillation in the electrical power system on the grid is translated directly to the oscillation of the generator rotor, shaft, gearbox, and the prime mover. Thus, a sudden change in the grid will have a direct impact on the mechanical components of the generator and the prime mover.

All four WTG types (i.e., fixed-speed, variable-slip, variable-speed, and full-converter) are nonsynchronous. This is the difference between wind and conventional generators. A WTG has nonsynchronous characteristics. Thus, any electrical events on the transmission lines will have some damping before being transmitted to the mechanical components of the turbines. A wind turbine has a better mechanical compliance and mechanical coupling between the prime mover and the generator. Thus, any power spikes developed in the generator as a result of abnormal events in the transmission line do not have to be translated directly to mechanical stresses. Instead, they may be buffered by a nonsynchronously-rotating WTG, in which case some of the electrical power spikes will be converted to kinetic energy of the generator (and the turbine blades) and the expected damage can be significantly reduced.

Type 3 and Type 4 WTGs operate in variable speed with a flux-oriented controller via power converter. Thus, the rotor does not have to rotate synchronously with the stator flux created by the grid rotating at the grid frequency. Any oscillations on the power system grid frequency may be compensated by the power converter control and thus can be prevented from affecting the mechanical components of a WTG.

From a power system perspective, a wind power plant is usually spread across a very large area to optimize the aerodynamic energy capture. Thus, there are diversities within a wind power plant. A turbine located at one corner may be exposed to a high wind speed, whereas a wind turbine located at another corner may experience low wind speeds. Any fluctuations at each wind turbine can be significantly different one from another. Thus, the power fluctuation at the point of interconnection (where the output of all turbines meet before transmitted to the transmission lines) will be a lot smoother than the output fluctuations at an individual turbine. This smoothing effect is a result of spatial diversity within a wind power plant. Obviously, the smooth output fluctuations will have a milder impact on a power system than if there is no diversity within a wind power plant.

Another diversity found in a wind power plant is the length of cables connecting individual turbines to the point of interconnection. The difference in the cable lengths and the diversity in the wind resource make each wind turbine experience different voltage drops along the cables

(from the point of interconnection to each individual turbine). This is actually a benefit for the power system. As shown in [28], the number of turbines disconnected from the grid during fault events is about 15% of the total turbines in most fault events observed for a one-year period. In addition, because of the diversity in a large wind power plant, the oscillations developed after a transmission fault is cleared may not be synchronized; thus, the impact of individual turbines on power system stability will be smoothed out because the oscillation from one turbine may cancel the oscillations from the other turbines.

Consider a 100-MW synchronous generator representing a conventional power plant. Also consider a wind power plant with 100 turbines of 1 MW each. When there is a severe fault at the transmission line, there may be a loss of 100 MW because of the disconnection of this synchronous generator from a power system. On the other hand, the response from a wind power plant could be very minimal, perhaps a loss of 15 MW. Similarly, if there is a power system oscillation on the power grid, the 100-MW synchronous generator may oscillate with the power system, unless some kind of damping (i.e., power systems stabilizer) is available. In a wind power plant, especially one that has turbines with power electronics to control their generators, the power system oscillations can be mitigated by the fast, flexible WTGs.

In power system analysis, often the worst case scenario is preferred; thus, a wind power plant is often represented by a single large generator. As described in the references [29–31], equivalencing a large wind power plant can be accomplished by considering the line impedances of the collector system. Figure 51 shows a single-line diagram describing the common layout of a wind power plant connected to a power system network. A turbine is electrically connected to a power system through a network of cabling, transformers, and overhead lines. The interaction between a wind turbine and the power system network is very important to the stability of the power system, especially when the size of the wind power generation is very large (e.g., at high penetration levels of wind power generation). The interaction occurs both ways. Any transients occurring at the transmission line will affect the generator and its components. Similarly, any perturbation in the wind speed at a turbine site will be reflected in the utility grid, unless smoothed out by opposing effects from other turbines within the plant.

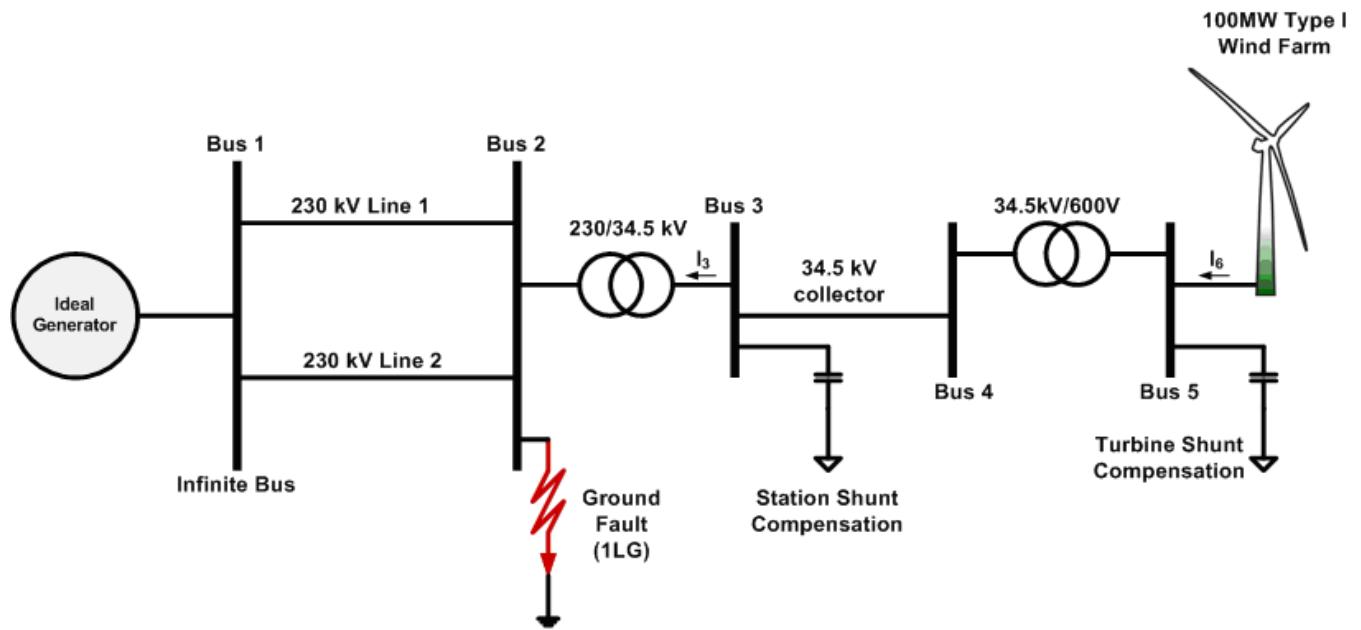


Figure 51. A simplified power system configuration often used in simulating fault ride-through capability of a turbine

5.2 Electrical Abnormal Events

5.2.1 Grid-Related Events

Abnormal events occurring on the grid affect the performance and integrity of wind turbines. Each turbine type has its own advantages and disadvantages when facing such events. Examples of abnormal events related to generators and power converters include the following:

- Balanced voltage events (equal undervoltage or overvoltage in the three phases)
- Unbalanced voltage event (undervoltage or overvoltage in one or two phases)
- Fault transients (three phase-to-ground faults, single or two-phase faults, grounded or floating)
- Voltage dips (direct online start-up of large induction motors, loss of lines or generations)
- Power system oscillations (inter-area, intra-area, sub-synchronous, etc.)
- Switching transients (capacitor switching, load switching, stuck breakers, tap changer transformer)

Although not listed here, an additional example demonstrated in Section 6 may also exacerbate the impact on a WTG for different grid conditions (stiff versus weak, balanced versus unbalanced, undervoltage versus overvoltage, steady versus oscillating frequency), different levels and types of reactive compensation (active versus passive compensation), different types and the winding connections of the transformers, and obviously different types of WTGs.

5.2.2 Generator and Power Converter-Related Events

Abnormal events occurring in the generator and power converter also affect the performance and the integrity of wind turbines. The types of generators, power converters, and control systems

affect a wind turbine operation, and power system stability. Examples of abnormal events related to grids include the following:

- Unbalanced impedance
- Unbalanced phase windings,(e.g., because of inter-turns shorts)
- Fault transients (three phase-to-ground faults, single or two-phase faults, grounded or floating)
- Imbalance between input and output power flowing through the DC bus because of loss of lines
- Power-switching failures and the corresponding DC bus fluctuations
- DC bus protection with dynamic braking, different types of storage, capacitor failures

5.3 Mechanical and Aerodynamic Abnormal Events

Abnormal events developed because of the wind resource, mechanical vibrations of the turbine blades or other components, and turbine controls may impact the grid and affect the performance of wind turbines. Examples of abnormal events related to aerodynamic and mechanical components include the following:

- Blade pitch actuator/control sluggishness and unbalanced pitch control response
- Runaway conditions resulting from failure of pitch actuator/control or brake mechanism
- Uncontrollable ramping, a sudden loss of wind, and other extreme aerodynamic input perturbations
- Severe wind turbulence

5.4 Wind Turbine Requirements

5.4.1 Grid Interface Requirement

In the early development of wind power, the level of wind power penetration into the grid is very low. For an abnormal condition on the grid (under- or overvoltage, frequency dip, etc.), a wind turbine is allowed to be disconnected from the grid to ensure that a wind turbine will not be harmed by the abnormal grid condition. Early standards for grid interface requirements were covered in the Institute of Electrical and Electronics Engineers 1547, applicable for generations less than the 20-MW power rating.

5.4.1.1 Voltage-Related Requirement

As wind power plants and the level of wind power penetration increases, the generated output power is considered significant to the overall generation pools. As such, the transmission operator requires that a wind turbine stays connected under general disturbance. This requirement is reflected in the Federal Energy Regulatory Commission Order 661 and 661A, also known as low-voltage ride-through and fault ride-through capability. This requirement covers both the voltage and frequency envelope that requires a wind turbine to stay connected to the grid. Beyond or outside this envelope, the turbine is allowed to be disconnected from the grid.

5.4.1.2 Frequency-Related Requirement

Recent trends indicate that there are more Type 3 and Type 4 turbines being installed in wind power plants. Type 3 and Type 4 turbines are known to have very flexible control capabilities and are operated in variable speeds; thus, the aerodynamic efficiency and energy yield is higher than that of constant-speed turbines. The output power of Type 3 and Type 4 wind turbines is controlled by the power converter, and it is usually controlled to follow a cubic function of the rotational speed. Thus, unlike the operation of grid-connected conventional synchronous generators, variable-speed wind turbines are not affected by sudden frequency changes.

In an interconnected power system, all the synchronous generators connected to the grid are synchronized to the grid frequency. The grid frequency is affected by the balance between the generation and the demand. If the level of generation is higher than the power demand, the grid frequency rises. Similarly, if the level of generation is lower than the total loads, the grid frequency drops. The rate of change of the frequency is affected by the total inertia of the rotating mass connected to the grid. The higher the inertia available in the grid and the lower the difference between the generation and load, the smaller the rate of change of frequency will be. As a result, the slower the change of the grid frequency will be. The grid frequency is controlled to be as constant as possible via the automatic generation control and governor control, thus balancing load and generation all the time.

5.4.1.3 Inertial Response Requirement

As wind power penetration increases, and conventional power plants are retired from the generation pools, the total inertia in the grid will be reduced. At higher levels of wind power penetration, there is concern that the total inertia will be too small to support the frequency stability in a power system. Transmission system operators in many regions, including the Electric Reliability Council of Texas and Hydro Quebec are starting to require the additional capability of wind turbines to provide inertial response, thus helping to impede the change of frequency in the grid [32].

5.4.1.4 High-Wind Ride-Through Requirement

In the future, some transmission system operators may require that a wind turbine stays connected to the grid during high wind speeds to ensure that there is no sudden drop in generation when the wind speed increases above the rated wind speeds. Obviously, wind turbine manufacturers design and decide the cut-off wind speed to ensure safe operation of normal wind turbines. However, in anticipation of future requirements, in which transmission system operators mandate operation above the rated wind speed, wind turbine manufacturers can include this requirement in the design and provide wind turbine customers with the option of high-wind ride-through capability.

5.4.1.5 Impact of Grid Interface Requirement

Although all of the aforementioned grid requirements can technically be provided by turbine manufacturers, the impacts (thermal, electrical, magnetic, structural stresses, and strains) on the mechanical and electrical turbine components, and on its integrity and lifetime, are yet to be fully understood and will be known further as the industry gathers years of experience. The FAST-Simulink combination is the perfect tool to investigate these impacts and enhance the design and

fortification of turbine components. The electrical aspects can utilize several modules available in Simulink, such as SimPowerSystem.

5.4.2 Electrical Component Requirement

Electrical component requirements are mostly on voltage and current limits. The voltage limit is related to the level of dielectric and insulation necessary to withstand the electrical field imposed on them. The voltage blocking capability of a component is specified in the data sheet, and the component must be protected from operating beyond the allowable voltage range. The current limit is usually related to the amount of current passing through the device without generating so much heat that it will degrade the dielectric and insulation of the components. The electrical components that form the linkages to convert and transfer mechanical energy into electrical energy to customers must be carefully designed to bear the loads and stresses of the process.

The rise of temperature above a critical point (specified in the data sheet) can be very damaging (irreversible degradation) to the electrical insulation and magnetic characteristics. The requirements for electric machines (rotating machineries, transformers, inductors, etc.) are usually easier to maintain because the technology, the size of the mass to store and conduct thermal losses to the ambient air, the auxiliary efforts to dissipate the heats, and the filtering/screen of the dust are very well established. Also, electric machines can better tolerate overloads (overcurrent) and overvoltage conditions. However, the power electronic components (IGBT, diodes, etc.) are very sensitive to the temperature because the electronic components are based on p-n junction. The bottleneck in electrical components is mostly dictated by the power electronic design ratings (voltage and current).

Because modern wind turbines must provide a good grid interface, the impact of providing fault ride-through capability and providing other ancillary services must be investigated to ensure that these requirements will not shorten the lifespan of the electrical components of a turbine and to better understand how grid interface requirements will drive the future design of wind turbines. Because requirements differ from region to region, it is probable that the same turbine types will be built at different enhancements to keep the costs of turbines as affordable as possible.

5.4.3 Energy-Harvesting Requirement

The main purpose of wind generation is to harvest as much energy as possible as soon as the wind speed available increases above the cut-in wind speed. Maximum power point tracking is generally implemented indirectly through passive mapping of output power commanded to the power converter to the rotational speed of turbine rotor. Because the grid interface requirement affects the reliability of a power system, and electrical disturbances usually last for a very short time, the grid interface controller takes precedence over the maximum power point tracking operation controller.

As wind power penetration levels increase, there will be times when the output of a wind power plant must be reduced to maintain the reliability of a power system. This is called curtailment, and it is a common practice when the available transmission capacity of the transmission lines is exceeded. Curtailment is also needed when the output power of a conventional generator falls short of its minimum because the wind power is high but the load connected to the grid is low. This condition often occurs at nights. Curtailment might also be profitable when the cost of energy to operate as spinning reserves is sufficiently higher than generating the output power at

normal operation. The spinning reserve operations of WTGs have been discussed and published in several papers [33].

Because curtailment as a spinning reserve is not currently common practice, the impact of this operation on the stresses and strains on mechanical and electrical components of a wind turbine needs to be investigated.

5.4.4 Mechanical Component Requirement

Mechanical components of wind turbines are the main path to transfer wind energy into electrical energy. The mechanical link between the turbine rotor and generator are mostly the blades, low-speed shaft, gearbox, yaw drives, and the generator high-speed shaft. The mechanical linkages are very rigid, and the conversion of mechanical energy into electrical energy occurs via electromagnetic conversion at the air gap of the generator.

All the aforementioned requirements may impact the mechanical components linked together to convert aerodynamic input power from the wind into mechanical power into electrical output power. The tools presented in this report will be able to simulate the impact on mechanical components. Most of the mechanical components are simulated in FAST; thus, the output data representing the stresses and strains on each of the sub-components modeled in FAST can be exported, plotted, and compared to the base case. An additional detailed gearbox model built in Simulink can be readily assembled to replace the simple model available in FAST. This model is explained in Appendix A.

5.5 Designing Controls to Mitigate Impacts

The overall energy flow diagram of wind power generation is illustrated in Figure 52. The input energy is the kinetic energy stored in the wind. The wind drives the mechanical linkage that converts wind energy into mechanical energy, and the electrical linkage converts mechanical energy into useful electrical power from a wind power plant to the energy consumers via transmission and distribution lines.

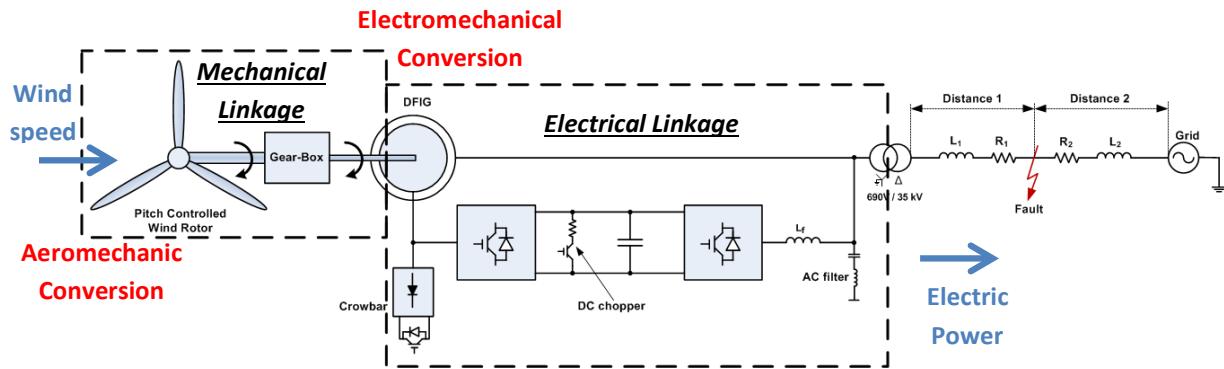


Figure 52. A simplified diagram showing various linkages and the power flow in a wind power plant

5.5.1 Mechanical Linkage

As shown in Figure 52, the mechanical linkage consists of different components, as discussed previously (shaft, blades, gearbox etc.). The input is an aerodynamic input, and the output is the electrical output. The source of mechanical stresses (mechanical loads) are diverse. Only a few sources relevant to this project are discussed in this report.

5.5.1.1 Aerodynamic Input

The aerodynamic input consists of the average wind speed and additional turbulent wind speed. The turbulence is the higher frequency components, caused by wind obstruction (ridges, trees, other turbines) that creates mini swirls of wind imposed on individual blades. In addition, one turbine may experience different wind turbulence from other turbines within a wind power plant. Generally speaking, wind turbulence may excite the mechanical components within the mechanical linkage and may be detrimental to the mechanical component when it hits its frequency modes.

5.5.1.2 Mechanical Output

The mechanical output is the torque that drives the generator. In Type 1 and Type 2 turbines, this torque is a function of the rotational speed. In Type 3 and Type 4 turbines, this torque is controllable by the power converter. The mechanical output converted from the aerodynamic power is affected by the aerodynamic and mechanical controls.

5.5.1.3 Inherent Short-Term Storage and Damping

The inherent short-term storage that exists within the mechanical linkage includes the blade aeroelasticity, the kinetic energy in and out of the rotating mass (shaft, gearbox, generator, blades), and potential energy within the shaft and gearbox stiffness. Short-term storage also includes the inherent damping in the blade-air interaction, the gear-to-gear in the oil bath of the gearbox, the windage from the air-cooled generator, and/or liquid friction losses in the water- or oil-cooled generator. All of these actually provide some kind of buffer to smooth out the energy spikes presented to the turbines by the presence of turbulence or other sources.

5.5.1.4 Aerodynamic and Mechanical Control

The aerodynamic control of a wind turbine is provided by stalling a wind turbine (self-limiting aerodynamic control). However, the stalled-control wind turbine has not been adapted in the newer turbines. Most of the new turbines use pitch control to adjust the aerodynamic input. Another type of control used in a wind turbine affecting the aerodynamic input is the yaw drive. Mechanical brakes (electromagnetic or hydraulic) are also common in WTGs to avoid a possible runaway event when a wind turbine loses its connection to the grid, and to stop the turbine during parking or repair/maintenance.

5.5.2 Electrical Linkage

As shown in Figure 52, the electrical linkage consists of different components (generator, power converter, capacitor, inductor, etc.). The input is a mechanical torque at the generator shaft input, and the output is the electrical power. The sources of electrical stresses (voltage and current) in a wind turbine are diverse. Only a few sources of electrical stresses relevant to power generation are discussed in this report.

5.5.2.1 Grid Side-Transmission Lines

Many events may occur at the grid as results of natural causes (lightning; short circuits caused by falling trees or animals; shorted, sagging lines caused by high winds, etc.) or man-made events (capacitor switching, loss of lines during fault clearing, loss of generators, loss of loads, etc.). These events may create overload currents, over- and under-voltages, or normal/unbalanced voltages. Most severe events in transmission lines can be quickly removed by activating the circuit breakers to minimize the affected lines and/or customers. However, before being cleared, the abnormal event may be severe enough that it creates irreversible damage on the turbine components (the gearbox, generator, power converters, etc.), especially if the event creates torque or voltage spikes. Other, less-severe events, such as unbalanced voltage, may go unnoticed for a longer duration than acceptable because they are undetected by the sensors and relay protection is not triggered. These events may not cause instant fatal effects; however, if left uncorrected, the torque pulsations and unequal heating in the generator's stator windings may lead to catastrophic failures.

5.5.2.2 Point of Interconnection—Substation

The point of interconnection of a transformer is often chosen to be the high side of the substation transformer, where the metering is installed and the revenue is accounted. At the substation, protections (circuit breakers, transient voltage suppressors, fuses, etc.) are in place to keep the transformer, switchgear, and generators from experiencing abnormal conditions (short-circuit currents, over-voltages, etc.). If the various protections installed at the substation malfunction (e.g., because of aging, improper maintenance, damage, or the wrong coordination of relay protection), the abnormal events can be transmitted to the generators and power converters connected to a power line.

5.5.2.3 Generators and Power Converters

Generators and power converters are connected to the grid. In Type 1, 2, and 3 WTGs, the generator stator windings are connected to a power grid; hence, any voltages and frequency disturbances may affect wind turbine components instantly and directly. In Type 4 WTGs, a power converter connects the grid to the stator winding of the generator; thus, there is less direct impact of line disturbance on a WTG.

5.5.2.4 Power Converter

The power converters for Type 3 and Type 4 WTGs are connected to the grid; thus, anything that occurs on the grid can be passed to a generator directly (for Type 1, 2, and 3 WTGs) or indirectly (for Type 4 WTGs). The power converter and generators are very susceptible to abnormal events occurring on transmission lines. Subsequently, because of abnormal events on the grid, the generated torque may impact the wind turbine mechanical components. Further, the extreme ramping rates and turbulence coming to the wind power plant may affect the power system to which it is connected.

6 Case Studies

6.1 Example Case 1: Grid – Turbine Interaction

6.1.1 Type 1 WTG

Using FAST and MATLAB/Simulink, a model of the AOC AWT-27 turbine [19], a Type 1 WTG, was developed. A brief subset of simulation results using this model is presented here. Because Type 1 WTGs are infrequently installed today, the bulk of this section is devoted to modeling Type 3 WTG behavior. Nonetheless, modeling Type 1 turbines may be useful in academic settings because these models illustrate many basic principles of wind turbines as well as show how far the technology has progressed and how many initial problems faced by these turbines have been resolved in more-modern designs.

Figure 53 shows that under steady wind conditions, the generator torque and speed of this WTG were affected by the tower shadow effect causing aerodynamic torque reduction every time one of the blades passed through the wake of the tower, thus creating torque pulsation. AWT-27 has two blades; thus, this phenomenon occurred two times per full rotational angle of the rotor. Figure 54 shows that the thrust load and the output power of this WTG were affected by the tower shadow. Note that in a real wind power plant, the wind speed contains wind turbulence. The pulsations shown in Figure 53 and Figure 54 would have additional turbulence components on top of the pulsations because of the tower shadow.

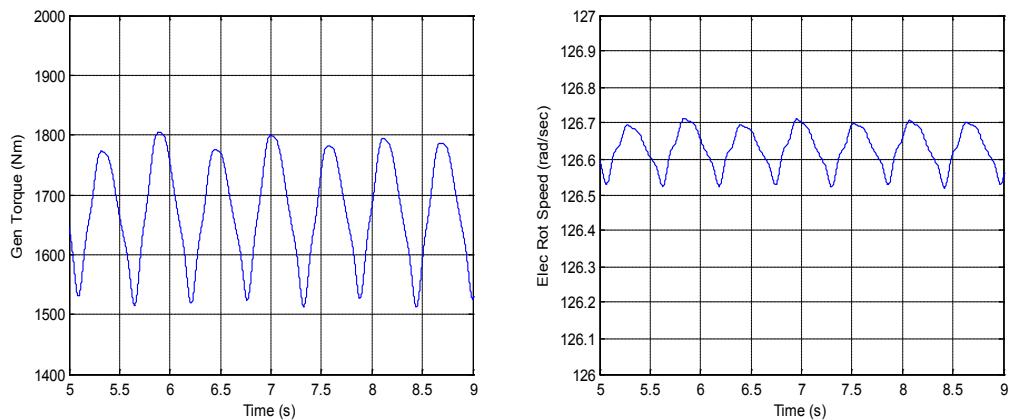


Figure 53. Simulation results showing the impact of tower shadow on the generator torque and speed for a Type 1 WTG

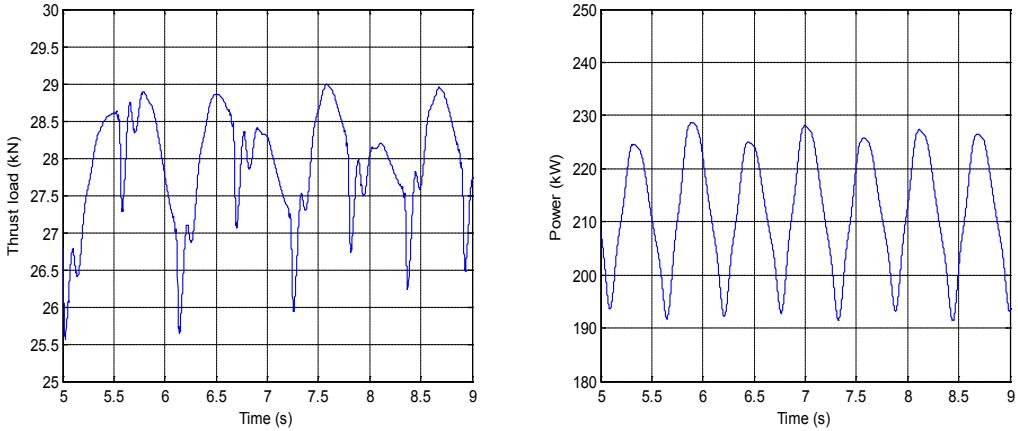


Figure 54. Simulation results showing the impact of tower shadow on the generator thrust load and output power for a Type 1 WTG

6.1.2 Type 3 WTG

Using FAST and MATLAB/Simulink, a model of the GRC 750-kW turbine [4], a Type 3 WTG, was developed. Parameters of this turbine are discussed in detail in the following case study. The model was used to demonstrate that turbines connected in stiff grids will experience slightly different transient behavior than turbines connected in weaker grids. These demonstrations may have a value in determining turbine operations and maintenance schedules or turbine life. In the simulations, at $t = 10$ s, a voltage sag occurred. Both single-phase unbalanced and three-phase balanced faults were modeled in this example. The sag dropped the grid voltage from 1 p.u. to 0.1 p.u. The sag persisted for nine cycles (150 ms) and then cleared. The simulations were carried out for both stiff and weak grids, and the results were plotted together. Note that the weak grid was simulated by doubling the line impedance of the grid. The wind speed was held steady at 12 m/s, below the rated speed, so pitch control was inactive.

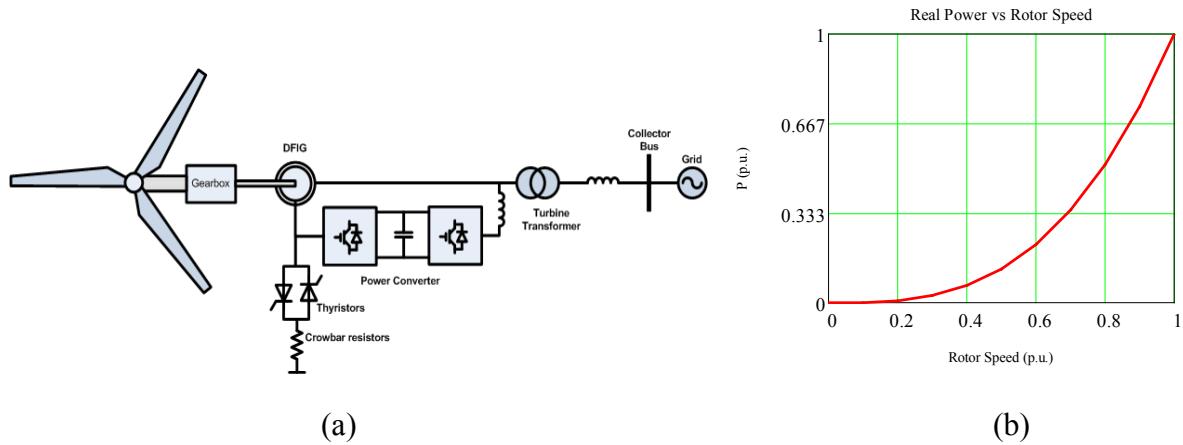


Figure 55. (a) Physical diagram and (b) power versus speed characteristic of a Type 3 WTG

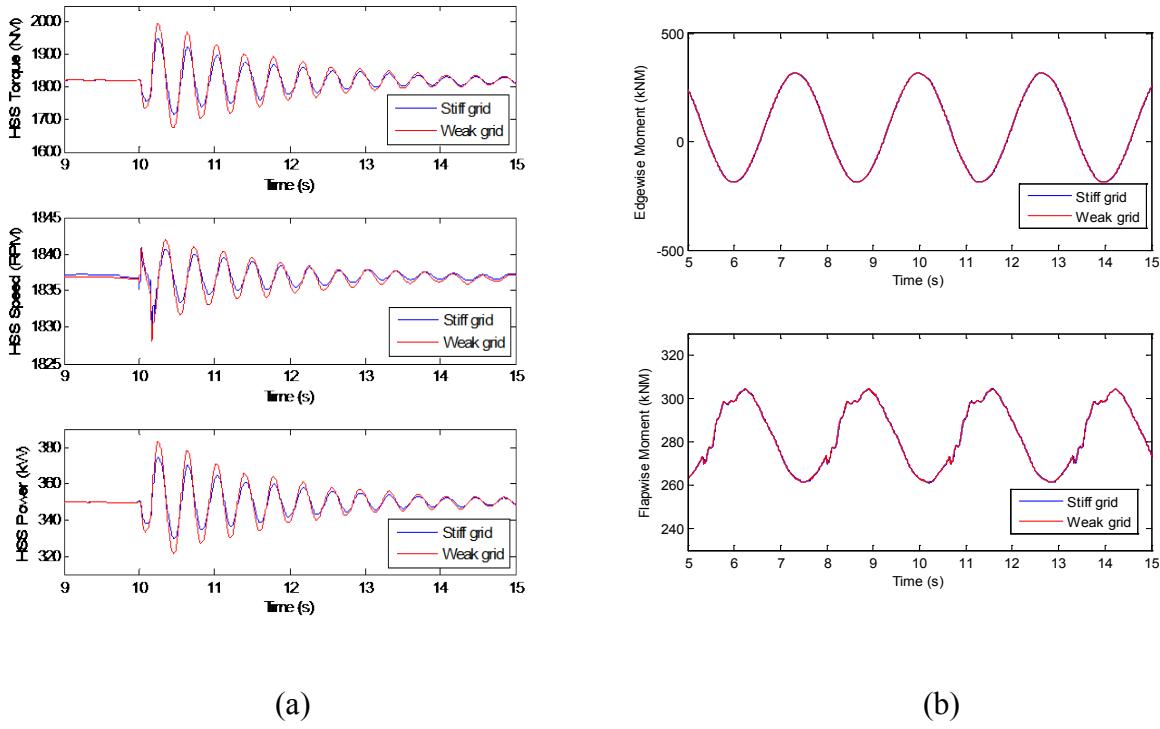
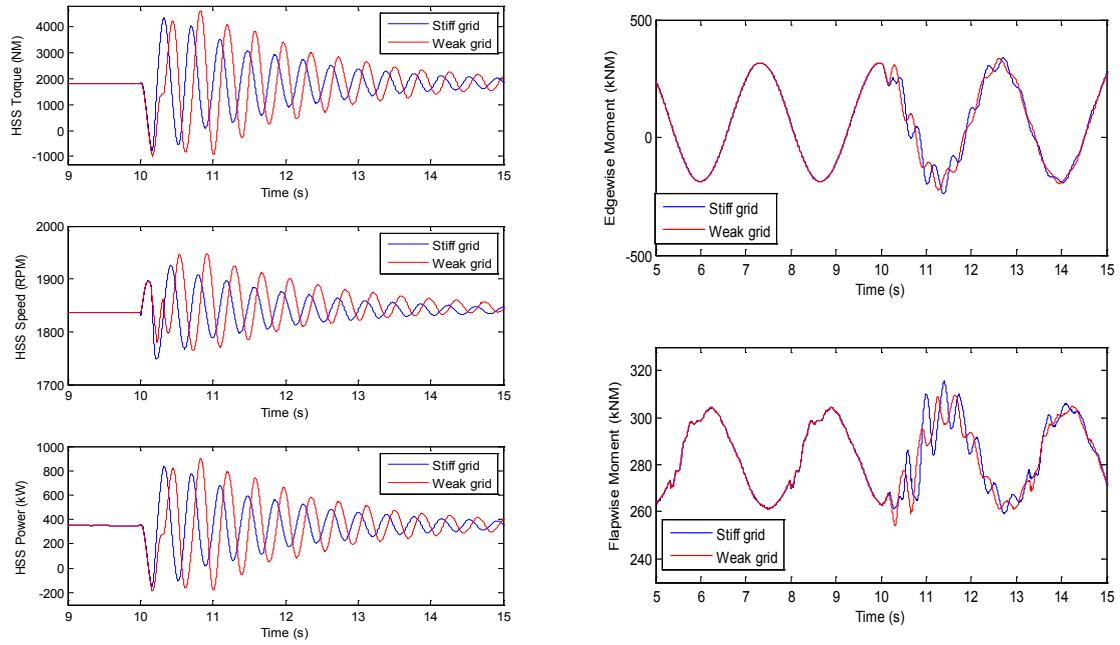


Figure 56. Simulation results showing the impact of a single-phase voltage sag for a Type 3 WTG on (a) high-speed shaft torque, speed, and power and (b) edgewise and flapwise blade moments at the blade root

The physical diagram and the power-speed characteristic of a Type 3 WTG are shown in Figure 55. Figure 56 (a) and Figure 56 (b) show some results from a simulation of a single-phase sag. Figure 56 (a) shows torque, speed, and power at the high-speed shaft. The sag caused an approximately 2.5-Hz oscillation to occur that persisted long after the fault cleared, indicating that some mechanical oscillation mode within the drivetrain was likely excited. The plots also show that the amplitude of the oscillation was greater for the weak grid case, potentially leading to more damage to the high-speed shaft and drivetrain during the life of a turbine. Although Figure 56 (a) is informative, Figure 56 (b) shows the true potential of the FAST and MATLAB/Simulink coupling. Figure 56 (b) shows the edgewise and flapwise blade moments at the blade root during the single-phase sag. As shown, the sag produced no noticeable difference between the turbines connected to the weak or stiff grid. It is also shown that the strength of the grid connection had no impact on these blade moments. These insights would be difficult to obtain without the coupled FAST and MATLAB/Simulink model.



(a) HSS torque, speed and power (b) Edgewise and flapwise blade moments at the blade root

Figure 57. Simulation results showing the impact of a three-phase voltage sag on the output power of a Type 3 WTG on high-speed shaft torque, speed, and power, as well as on edgewise and flapwise blade moments at the blade root

Figure 57 shows results for the three-phase sag. In this case, the transient was much more severe. The torque swing was near an order of magnitude greater than in the single-phase case (compare Figure 57 (a) to Figure 56 (a)). There were similar swings in the speed and torque, indicating the severity of the event. It is also shown that the level of grid stiffness affected the response both in frequency and damping. Note that this was a worst-case scenario: in this case, the crowbar was inactive; thus, the crowbar did not operate to limit the rotor currents and the turbine was exposed to the full intensity of the event. There was also a noticeable difference between the weak and stiff grid cases: a phase shift occurred in the weak grid case for which the cause was unknown. Again, the oscillations had higher amplitude in the weak grid case. Figure 57 (b) shows that in the three-phase fault case, in contrast to the single-phase case shown in Figure 56 (b), the severity of the three-phase fault actually led to noticeable oscillations in the edgewise and flapwise moments at the blade root. It is shown in Figure 57 (b) that the system with a stiff grid responded more favorably than the one connected to a weak grid. Although the models display the ability to provide valuable insights into turbine mechanical and electrical coupled transients, the reliability of the results cannot be confirmed yet. In further work, efforts will be made to validate results obtained from coupled FAST and MATLAB/Simulink models using real field measurements. Potential methods to mitigate stresses using advanced controls will also be studied.

This demonstration of the ability of the coupled FAST and MATLAB/Simulink models to provide insights about the effects of strong and weak grid connections is one example of the potential of this coupling. This research will also study the effects of electrical faults on mechanical components (gearbox thrust loading, tower dynamics etc.); the effects of mechanical

oscillations on output power/current/voltage; the effects of voltage unbalance; the benefits of soft-starting and other power electronic-based damping mechanisms; the converter-control effects on mechanical systems (and other controller interactions); and the effects of wind gusts and turbulence on electrical systems.

6.2 Example Case 2: Impact on Mechanical Linkages

Because a wind turbine drivetrain (i.e., the mechanical linkages) consists of components that directly convert rotational kinetic energy from the wind to electrical energy, ensuring the reliability of drivetrain designs is critical to preventing wind turbine downtime. Because of the steadily increasing size of wind turbines, larger forces and torques bring up the influence of the gearbox and other drivetrain flexibilities in the overall turbine dynamic response [14–15], often leading to failure in the drivetrain components. Failure in drivetrain components is currently listed among the more problematic failures during the operational lifetime of a wind turbine. In particular, gearbox-related failures are responsible for more than 20% of the downtime of wind turbines. Although the expected lifetime of gearboxes is usually advertised as 20 years, in practice gearboxes usually need to be replaced every 6 years to 8 years [16–17].

This case study demonstrates the versatility and usefulness of the coupled FAST and MATLAB/Simulink model in enhancing the fidelity of the wind turbine model by incorporating various libraries Simulink has offered. Further insights into wind turbine drivetrain dynamics will be helpful to understand the global dynamic response of a wind turbine as well as to design and preserve its internal drivetrain components. Thus far, however the drivetrain model of FAST is reduced to two degrees of freedom, resulting in restricted detail in describing its complex dynamic behavior. Although researchers have developed dynamic models for wind turbine drivetrains with various levels of fidelity [18–25], these studies do not provide direct insights on the dynamic interactions between the drivetrain and other components of the entire wind turbine. The most recent study in [25] takes a decoupled approach in which the global turbine response was first simulated using an aeroelastic CAE tool. After the simulation, the resulting loads and motions of the rotor as well as the nacelle were used as inputs to a high-fidelity model of the drivetrain to simulate its internal dynamic behavior. Thus, this approach fails to capture the influence of the drivetrain dynamics onto the overall turbine response.

This case study aims to investigate the global dynamic response of a wind turbine drivetrain by integrating a dynamic model of the drivetrain. The drivetrain model was built using SimDriveline, which is a part of Simscape in Simulink [7]. The step-by-step guide to build this drivetrain model is in Appendix A. This tool extends the coupled FAST and MATLAB/Simulink model, which in the future can help in the design and verification of active control strategies to mitigate the drivetrain loads.

The turbine modeled in this study was based on the GRC turbine [11] at NREL. Table 4 summarizes the important properties of this turbine.

Table 4. Modeling Properties of GRC Wind Turbine

Configuration, Rating	3 Blades, 750 kW
Control	Variable Speed, Collective Pitch
Gearbox, Overall Ratio	3 Stages, 81.49
Rotor, Hub Diameter	24.1, 0.6 m
Hub Height	54.8 m
Rated Rotor Speed	22.1 m/s
Maximum Rotor C_p	0.43

Table 5. Parameters of the 750-kW DFIG

Generator	
Line-Line Voltage (RMS)	690 V
Frequency, No. of Pole Pairs	60 Hz, 2
Stator Resistance, Leakage Inductance (p.u.)	0.016, 0.06
Rotor Resistance, Leakage Inductance, both referred to Stator (p.u.)	0.016, 0.06
Magnetizing Inductance (p.u.)	2.56
Inertia Constant (s)	2
Converter	
Converter Maximum Power (p.u.)	0.5
Grid Side Coupling Inductance, Reactance (p.u.)	0.15, 0.0015
Nominal DC Bus Voltage	1200 V
DC Bus Capacitor	0.1 F

The GRC wind turbine originally employed two fixed-speed WTGs. To model variable-speed operation, a WTG was modeled using a DFIG, shown in Figure 55. Because the drivetrain had much slower dynamics than the mechanical drivetrain, an average model of the AC-DC-AC converter was used in this study, in which the power electronic devices were replaced by controlled current sources. Details of this average model are discussed in Appendix B. The induction generator model was based on a commercial wind turbine of the same rating available in the market. Table 5 summarizes the key parameters of the generator and the converter, which were used to build the variable-speed WTG model in the SimPowerSystems environment.

For wind speeds below 12.5 m/s, which is the rated wind speed of this turbine, the output power of this WTG was controlled to track the maximum power coefficient (C_{Pmax}), while maintaining constant pitch angle at its optimum (-3.5° for this turbine). In variable-speed operation, the rotor speed ω_{rot} is proportional to the wind speed V_w :

$$\omega_{rot} = \frac{\lambda_{opt} V_w}{R_{rot}}$$

where λ_{opt} is the desired optimum tip-speed ratio.

6.2.1 Drivetrain Modeling

6.2.1.1 Five-Mass Model

This study focused on a commonly used modular drivetrain configuration in operating turbines [5]. Figure 58 shows the building blocks of the configuration. In this turbine, the multistage gearbox consisted of a planetary (epicyclic) gear set and two parallel gear sets, with two intermediate shafts. Figure 59 shows the five-mass model of a wind turbine drivetrain with fixed-

speed generator developed in the Simscape/SimDriveline environment. A fixed-speed generator has an electrical torsional stiffness between the air gap magnetic field and the generator rotor. This stiffness behaves as a spring to the inertial reference frame of the drivetrain, which provides restoring torque to the rest of the drivetrain. Such stiffness arises because of tight allowable speed variation in the fixed-speed turbine. Effects of this stiffness are prominent in the transient response of the generator (e.g., during generator start-up or grid-fault events). This stiffness value can be obtained through experiments [13]. In a variable-speed generator, this restoring effect does not exist, and the drivetrain model shown in Figure 59 had a free boundary condition on the other side of the generator.

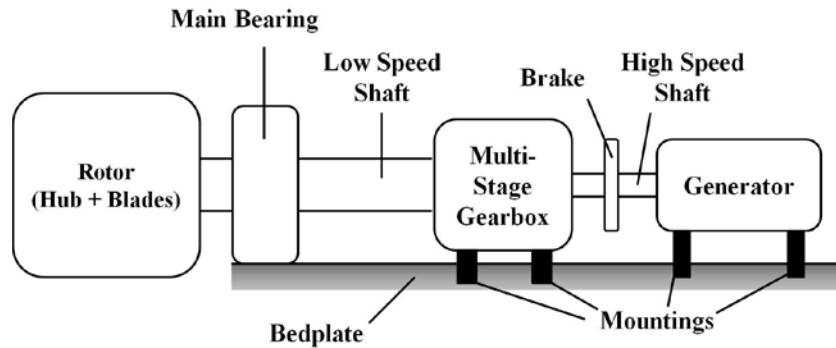


Figure 58. Modular drivetrain configuration of a wind turbine

Frequency analysis on this five-mass model revealed eigenfrequencies of the drivetrain as summarized in Table 6. These frequencies imply several possible resonant excitations that will lead to amplified loads in the drivetrain.

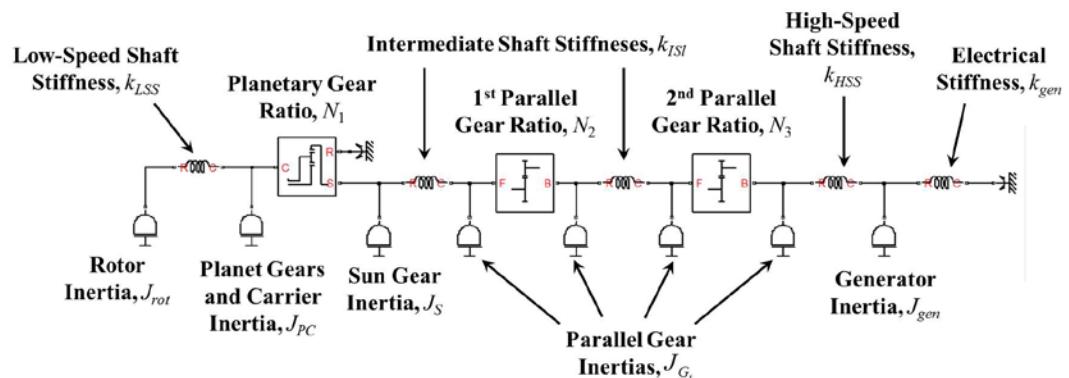


Figure 59. Five-mass model of a wind turbine drivetrain in SimDriveline with a fixed-speed induction generator

Table 6. Eigenfrequencies of a Five-Mass Model

Drivetrain with Fixed-Speed Generator		
Mode	Description	Frequency (Hz)
1	Low-Speed Shaft Mode	0.86
2	Generator Start-Up Mode	6.05
3	High-Speed Shaft Mode	312
4	1 st Gearbox Mode	402
5	2 nd Gearbox Mode	1960
Drivetrain with Variable-Speed Generator		
Mode	Description	Frequency (Hz)
1	Generator Static Model	0
2	Low-Speed Shaft Mode	2.52
3	High-Speed Shaft Mode	312
4	1 st Gearbox Mode	402
5	2 nd Gearbox Mode	1960

6.2.1.2 Two-Mass Model

Figure 60 illustrates the configuration of the two-mass model commonly used to model the dynamics of drivetrains in FAST. Inputs into the model were the five parameters: J_{rot} , k_{eff} , c_{eff} , N , and J_{eff}/N^2 . The generator electrical torsional stiffness is generally not required in such codes because this stiffness is inherent to the generator model used for the analysis. Parameters of the two-mass model can be derived from the five-mass model as follows:

$$J_{eff} = J_{PC} + N_1^2(J_S + J_{G_1} + N_2^2(J_{G_2} + J_{G_3} + N_3^2(J_{G_4} + J_{gen})))$$

$$\frac{1}{k_{eff}} = \frac{1}{k_{LSS}} + \frac{1}{N_1^2 k_1} + \frac{1}{(N_1 N_2)^2 k_2} + \frac{1}{(N_1 N_2 N_3)^2 k_{HSS}}$$

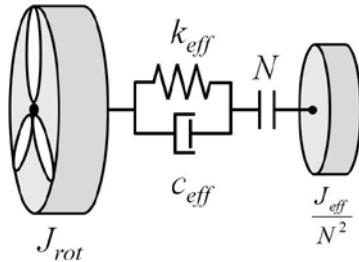


Figure 60. Two-mass model of a wind turbine drivetrain

The effective drivetrain torsional damping, c_{eff} , can be determined experimentally through several braking events [8]. The nonzero eigenfrequency of this two-mass model was calculated as:

$$f_n = \frac{1}{2\pi} \sqrt{k_{eff} \left(\frac{1}{J_{rot}} + \frac{1}{J_{eff}} \right)}$$

The resulting eigenfrequency using the two-mass model for the GRC drivetrain was 2.96 Hz, which was quite different from the first nonzero eigenfrequency of the five-mass model that was 2.52 Hz. This first eigenfrequency is of importance because it stores the most torsional energy. As exhibited in the following section, the two-mass model has limitations in providing insights on possible resonant excitations of the drivetrain as well as in analyzing the loads experienced by different components of the drivetrain.

6.2.1.3 Pure Torsional Model of Gearbox

In the two previously described drivetrain models, the meshing gear was modeled as an ideal static gain for torque and speed transmission. In reality, the gear transmission error, which is defined as the difference between the actual and ideal angular positions of the rotating gear mainly because of the gear elastic deformation, contributes to the dynamics of the pair meshing gear. This phenomenon contributes to the definition of gear meshing stiffness. For this study, a purely torsional model of the gearbox with constant meshing stiffness was built in the Simscape/SimDriveline environment.

6.2.1.3.1 Parallel Gear Stage

Figure 61 (a) shows a parallel gear set, which is a torque reducer, commonly employed in wind turbine drivetrains. Figure 61 (b) represents its flexible equivalent, in which the meshing stiffness acts on the line of action of the meshing gears. This meshing stiffness k_{mesh} , with respect to the input gear, can be represented as [4]:

$$k_{mesh} = k_{gear} (r_{b1} \cos \beta)^2$$

where the gear-tooth stiffness k_{gear} can be determined according to standards [26–27].

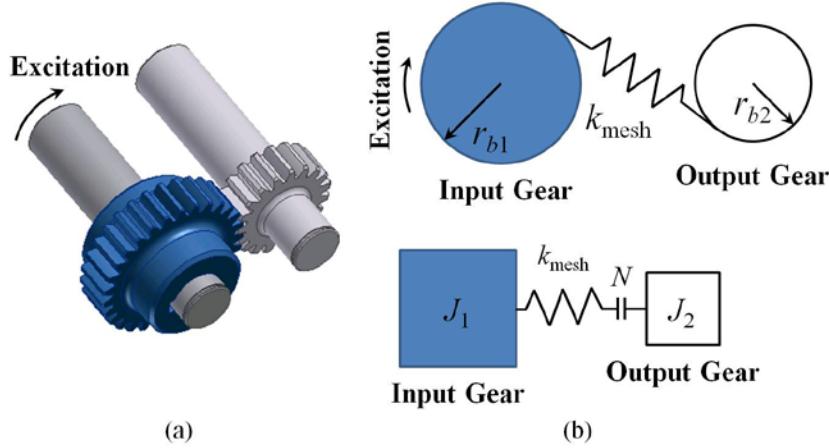


Figure 61. Parallel gear stage and gear mesh stiffness representations

6.2.1.3.2 Planetary Gear Stage

Figure 62 shows a planetary gear set with three planet gears and similar configuration to the one installed in the GRC turbine. The rotational input was from the carrier of the planetary gear stage, which provides rotational motion through the planet gears, and finally to the sun gear. The ring gear was modeled to have flexible coupling with the gear housing. Flexibility between the meshing planet and ring gears, as well as between the meshing planet and sun gears, can be modeled similar to that of a parallel gear set, as shown in Figure 61 (b).

Figure 63 shows the torsional model of a planetary gear set built in the Simscape/SimDriveline environment. This model can be adapted for any M equispaced planet gear set. To validate this model, comparisons with published frequency analysis [4] of planetary gear sets to various numbers of planet gears were performed. The results of the comparisons showed good agreement and are summarized in Table 7.

Table 7. Eigenfrequencies of Planetary Gear Stage

Three Planet Gears		
Mode	SimDriveline Model	Peeters [11]
1	2.273 kHz	2.217 kHz
2	6.340 kHz	6.159 kHz
3	11.296 kHz	11.205 kHz
Four Planet Gears		
1	2.207 kHz	2.138 kHz
2	6.911 kHz	6.688 kHz
3	12.699 kHz	12.577 kHz
Five Planet Gears		
1	2.153 kHz	2.059 kHz
2	7.403 kHz	7.105 kHz
3	13.980 kHz	13.810 kHz

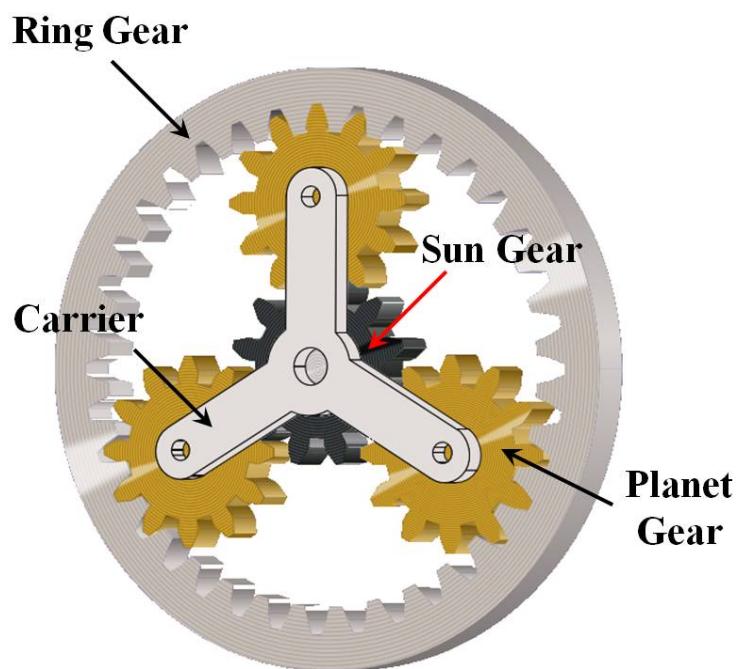


Figure 62. Planetary gear set with three planet gears

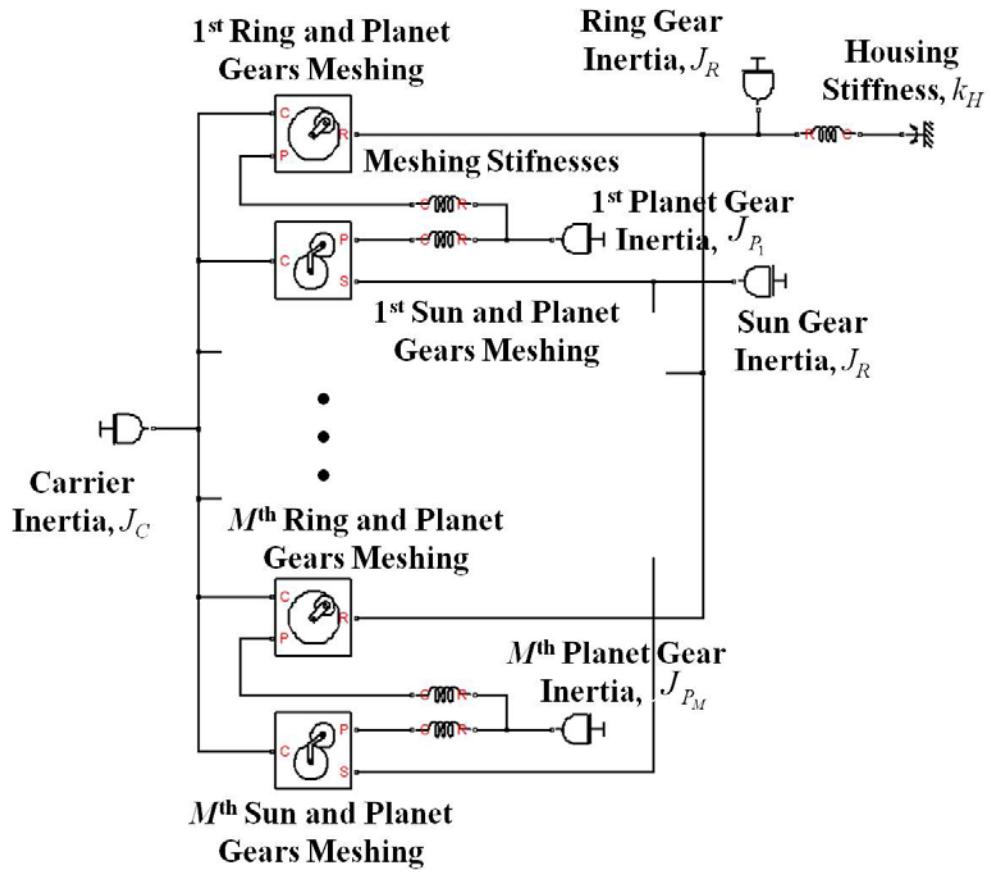


Figure 63. Torsional model of a planetary gear stage with M planet gears

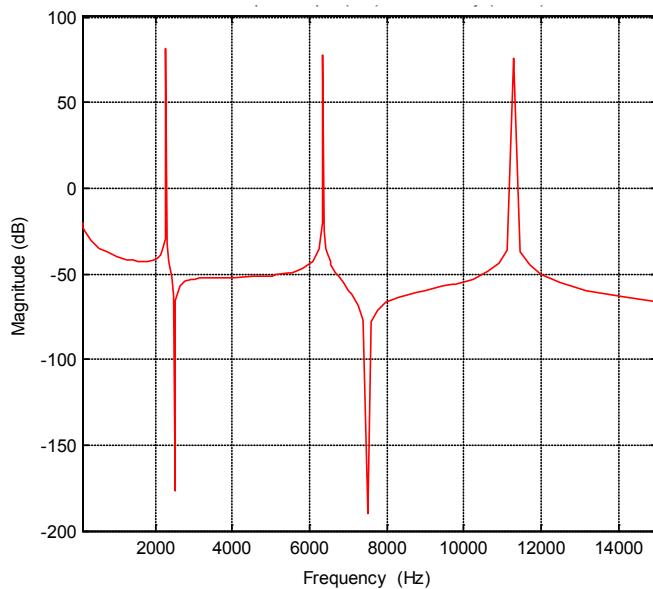


Figure 64. Frequency response function of a three-planet planetary gear stage for the gearbox presented in [16]

Figure 64 shows the frequency response function of the planetary gear set in Table 7 that has three planet gears. The frequency response function was produced by inputting torque at the carrier and taking the rotational speed of the sun gear as the output. It is important to note that the planetary gear sets used for the analysis in Table 7 and Figure 64 were different from the planetary gear set used in the GRC wind turbine drivetrain.

Table 8 summarizes the frequency analysis results of the GRC wind turbine drivetrain using the torsional model of the gearbox and variable-speed generator model. Exciting any of these eigenfrequencies will lead to amplified loads in the drivetrain.

Table 8. Eigenfrequencies of Drivetrain with Torsional Gearbox Model

Mode	Frequency (Hz)
1	0
2	2.44
3	154
4	307
5	353
6	748
7	1020
8	1530

6.2.2 Model Integration

Figure 65 illustrates the proposed strategy to integrate the described drivetrain models into the two-mass model inherent inside the FAST. For simplicity, the flexible modes of the other turbine components modeled inside FAST, such as those of the blades and tower, were not depicted in the schematic diagram in Figure 65.

In FAST, the two-mass drivetrain model was reduced to a single-mass model consisting of solely the rotor and the rigid shaft (as shown in the bottom part of Figure 65). This was done by deactivating the flexibility of the drivetrain (simulating rigid transmission) and setting the gear ratio and the generator inertia to unity and zero, respectively. The rotor equation of motion can be expressed as:

$$J_{rot} \alpha_{rot} = Q_{aero} - Q_{opp}$$

FAST internally calculates the input aerodynamic torque Q_{aero} from the defined wind profile but does not provide this torque as an output. However, because the rotor acceleration α_{rot} is an available FAST output, the aerodynamic torque Q_{aero} could be reconstructed using the rotor equation of motion as one of the inputs to the external drivetrain model. In this process, the rotor inertia J_{rot} was assumed constant and replicated in the external drivetrain model. The rotor inertia was connected to the flexible low-speed shaft, the purely torsional gearbox model, the high-speed shaft, and the generator inertia. The electrical machine and grid model took the generator speed and provided the generator electromagnetic torque to the drivetrain.

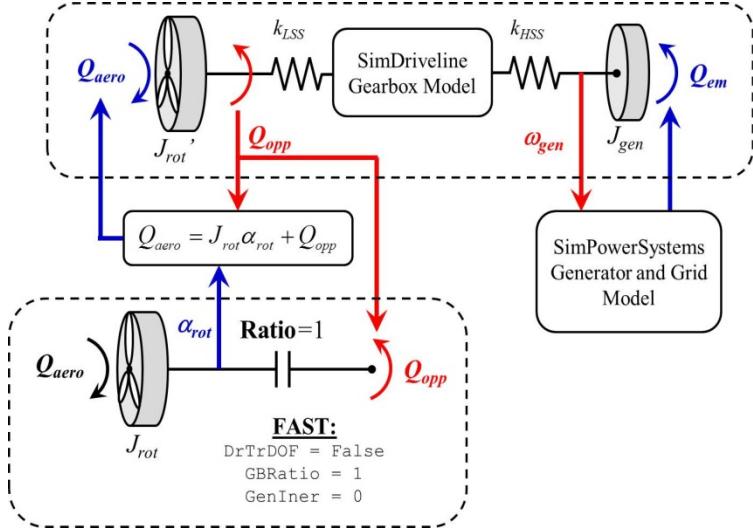


Figure 65. Proposed schematic of integrating Simscape drivetrain model into the FAST aeroelastic CAE tool

The rotor-opposing torque Q_{opp} was required as an input to the FAST drivetrain model as well as to calculate the aerodynamic torque Q_{aero} . In SimDriveline, this rotor-opposing torque could be retrieved by utilizing the torque sensor element behind the built rotor body. In general, torque, velocity, and angular position sensor elements can be placed flexibly within the Simscape drivetrain model to monitor the response of the drivetrain under various load conditions.

The remainder of this section presents simulation results showing the effectiveness of the torsional model of the gearbox under different transient load cases. Simulations using the FAST wind turbine aeroelastic code were conducted in the Simulink environment. In the simulation, all available wind turbine flexible modes in FAST—including that of the blades, tower, and drivetrain—were activated. No damping was defined within the drivetrain model to highlight the transient response of the drivetrain. Aerodynamic damping computed within FAST was the only source of damping that stabilized the overall drivetrain response. The results were compared with those using an undamped two-mass model of the drivetrain inherent in FAST.

6.2.2.1 Transient Response Caused by Wind Excitation

This simulation was performed under a constant wind speed of 9 m/s (below the rated wind speed), and the turbine speed was initialized to be 22.1 RPM. The start of simulation effectively imparted a large step input to the system that could excite all of the drivetrain modes, especially during the transient period. Figure 66 shows the turbine rotor speed using the integrated drivetrain model as well as the inherent FAST two-mass model. The rotor speed steadily increased to reach the optimal tip-speed ratio. Both models were in good agreement in the speed response of the turbine.

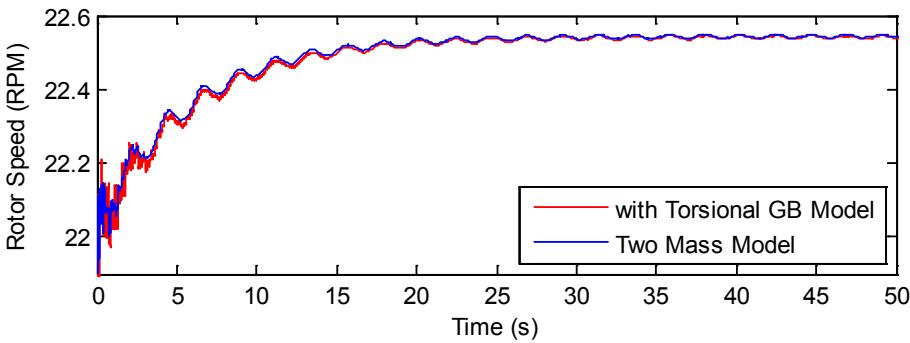


Figure 66. Rotor speed response

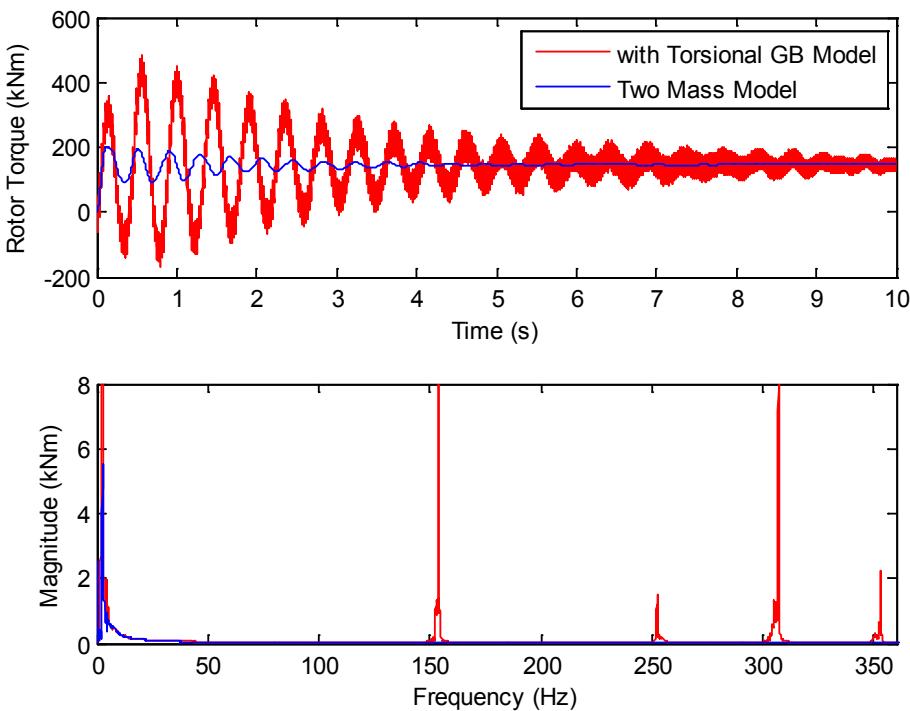


Figure 67. Transient response comparison of the rotor torque comparison

Figure 67 highlights the distinction between the two models in predicting the transient load response, and Figure 68 shows the corresponding steady-state load response of the drivetrain. Some frequency components came from other parts of a turbine's structure. The 0.4-Hz frequency came from the tower fore aft mode; whereas, the 1.1-Hz frequency and its harmonics came from the blade pass frequency (3P). The blade pass frequency is the signature of the tower shadow effect. The response of the drivetrain model with purely torsional gearbox was particularly high—twice the blade-pass frequency (i.e., 6P) of 2.25 Hz—because it was quite close to the estimated first eigenfrequency of 2.44 Hz. Thus, it predicted amplification of load caused by the resonance at the wind speed of 9 m/s. On the other hand, the two-mass drivetrain model estimated an eigenfrequency of 2.96 Hz, which was at some distance from the harmonics of the blade pass frequency, and hence did not predict any resonance.

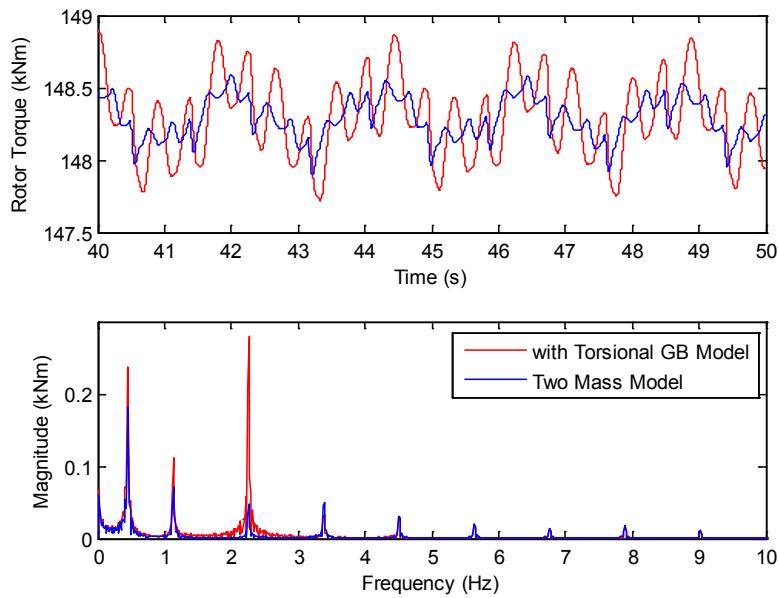


Figure 68. Steady-state response comparison of the rotor torque

6.2.2.2 Transient Response Caused by Grid Excitation

Another transient load can arise because of excitations from the grid. One example of grid excitation was simulated to predict loads on the gearbox (refer to Figure 69). A voltage drop for 0.15 second, from 100% to 90% and back to 100% of the nominal root mean square voltage, was simulated after the turbine reached steady state.

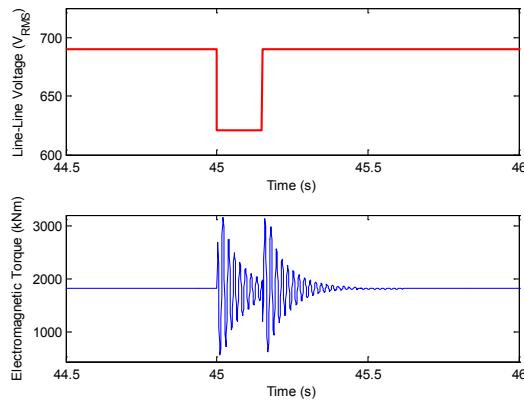


Figure 69. Electromagnetic torque excitations caused by a voltage drop on the grid

As shown in Figure 70, this voltage drop resulted in harmonic torque excitations onto the drivetrain with frequencies of 50.78 Hz and 56.15 Hz. These frequencies were inherent to the generator characteristic. It is important to note that the frequency component of this torque excitation may cause resonances if the frequency matches any of the drivetrain eigenfrequencies. However, these resonances cannot be predicted using the standard two-mass model because the two-mass model can only predict the lowest eigenfrequency of the drivetrain.

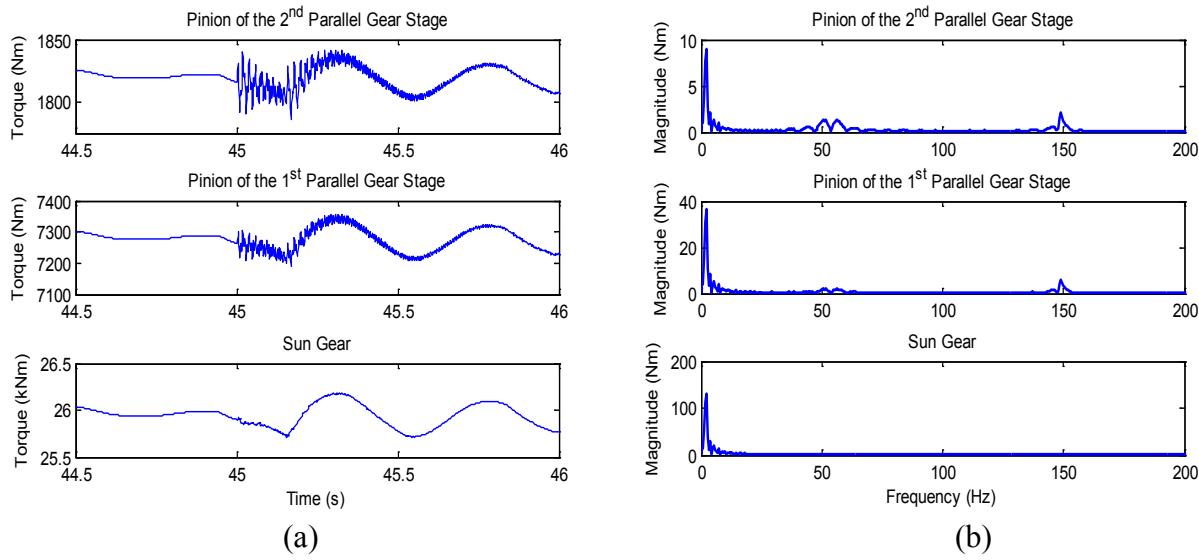


Figure 70. Transmitted loads onto the gears caused by grid excitation in (a) time domain and (b) frequency domain

Figure 70 illustrates how this load was transmitted to each stage of the multistage gearbox. The torques acting on the high-speed side of each gear stage (the pinion of each parallel gear stage and the sun gear of the planetary gear stage) are shown in both the time and frequency domains. Because of grid excitations, the pinion that was directly connected with the generator through the high-speed shaft experienced the largest proportion of high-frequency loads. Therefore, it was most prone to failures caused by fatigue in the event of grid disturbances. Sudden increases in the generator electromagnetic torque excited the two lowest and most dominant modes of the drivetrain (i.e., 2.44 Hz and 154 Hz). This torque also excited the system at its excitation frequencies of 50.78 Hz and 56.15 Hz, but with less dominant effect than at the eigenfrequencies. The transmitted loads on the gear were reduced as the gear got farther from the source of excitation, but the most dominant drivetrain eigenfrequency of 2.44 Hz prevailed during the transient regime.

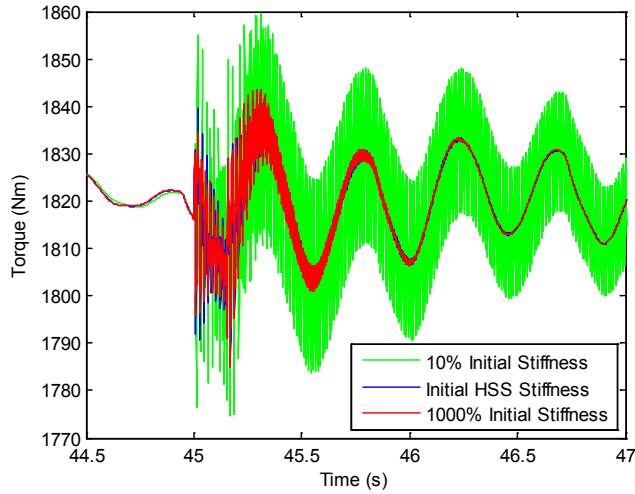


Figure 71. Loads on pinion of second parallel gear stage under various high-speed shaft stiffness values

The developed high-fidelity model can also be used to design drivetrain components to preserve or extend the life of the gearbox. The stiffness of the high-speed shaft, which connects the generator and the gearbox, was varied to investigate its influence on the load transmitted to the gearbox. In practice, this stiffness can be varied by altering the size of the shaft. In simulation analysis, the stiffness was varied to 0.1 and 10 times of the nominal value. The transmitted loads on the pinion of the second gear stage were evaluated and are shown in Figure 71. A lower stiffness appeared to transmit more-severe loads onto the pinion; whereas a higher stiffness slightly reduced the loads at the start. A higher stiffness can be achieved by using a shorter and/or larger diameter shaft. The developed model can help choose the optimal stiffness value to meet design and cost specifications as well as to maintain a certain range of transmitted loads onto the gear.

6.3 Example Case 3: Virtual Inertia and Damping Controller

The flexibilities of the drivetrain infer the possibility of unwanted resonant excitations of a wind turbine drivetrain. Resonance will incur additional stresses and in turn shorten the lifetime of the components. A protective scheme, such as the stress damper controller (SDC), has been designed to attenuate unwanted resonant loads in the variable-speed drivetrain. In Figure 72, α_{rot} represents the angular acceleration of the low-speed shaft, $\tilde{\alpha}_{rot}$ represents the filtered acceleration signal, and Q_{comp} represents the compensating torque. The SDC employs a band pass filter to generate compensating torque that will oppose the incoming loads that have frequency components close to the drivetrain eigenfrequency [13]. In this case study, a novel control design to prevent the drivetrain resonance was proposed based on added virtual inertia and damping control (VIDC).

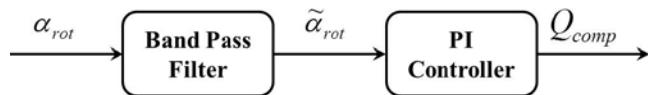


Figure 72. Schematic of SDC

6.3.1 Control Formulation

The system total inertia affects the eigenfrequency of the drivetrain system. And equation of motion of the two-mass model with respect to the low-speed side can be expressed as:

$$(J_{rot} + J_{gen})\ddot{\alpha}_{rot} = Q_{aero} - NQ_{em}$$

where Q_{aero} represents the aerodynamic torque and Q_{gen} represents the electromagnetic torque from the generator.

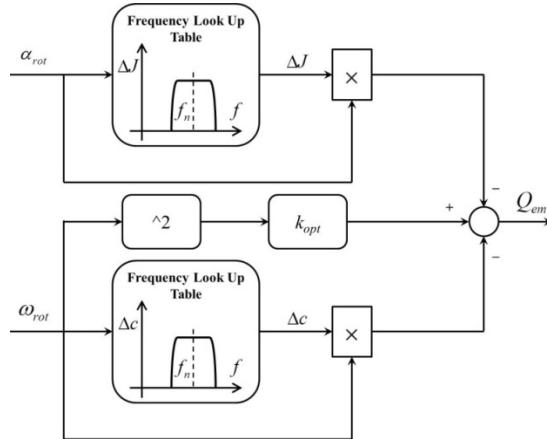


Figure 73. VIDC for drivetrain resonance prevention

By setting the generator electromagnetic torque for below-rated operation (i.e., variable-speed mode) as:

$$Q_{em} = k_{opt}\omega_{rot}^2 + Q_{comp}$$

where

$$Q_{comp} = -\Delta J \alpha_{rot} + (-\Delta c \omega_{rot})$$

the compensating torque injects “virtual” inertia and damping the drivetrain when it passes through its inherent resonant region. This strategy effectively shifts the effective eigenfrequency of the system and thus avoids the resonance. Otherwise, beyond the characteristic resonant region, there is no compensating torque required. In other words, the maximum C_P operation, which is implemented as the $k_{opt} \times \omega_{rot}^2$ in the DFIG (Type 3) WTG, is kept intact during normal operation. The idea of strategy is illustrated in Figure 73.

Figure 74 shows the proposed implementation of the above-mentioned compensating torque of VIDC. The proposed implementation tackles the efficiency issues encountered while trying to design an online fast Fourier transform for the frequency look-up table shown in Figure 73. The band-pass (or peak) filter was tuned to isolate the frequency contents of interest around the drivetrain eigenfrequency. Logic was implemented to check the magnitude of the filter output and decide that the system was under resonant if the magnitude was beyond a certain threshold. Based on how much it exceeded the defined threshold, the logic decided the amount of virtual parameters injected to the drivetrain. In this study, hysteresis logic was implemented to decide whether to add zero, medium, or high virtual inertia and damping.

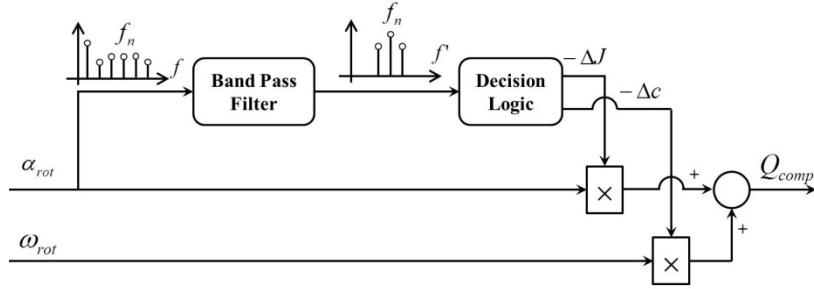


Figure 74. Implementation of the VIDC

Simulation using the FAST wind turbine aeroelastic code in the Simulink environment was conducted to evaluate the performance of the proposed control strategy (refer to Figure 46). In the simulation, all available wind turbine flexible modes in FAST—including that of the blades, tower, and drivetrain—were activated to emulate the real physical conditions of the 750-kW NREL GRC turbine. Inside the controller, the virtual inertia took values of 0.03 % and 0.1 % of original total inertia for the medium and high values, respectively. Because the drivetrain of this turbine had relatively large damping, the virtual damping was designed to be only 1 % (in terms of value) of the implemented virtual inertia to avoid a sluggish (overly damped) response of the system.

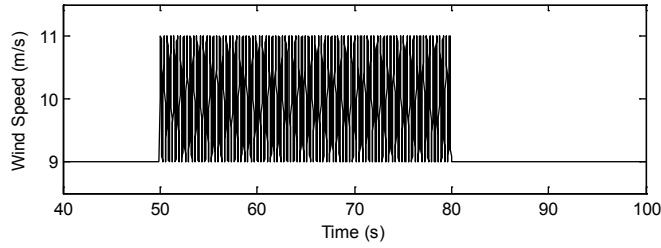


Figure 75. Sinusoidal wind speed input at the drivetrain eigenfrequency

To assess effectiveness of the proposed controller in the event of resonant excitation, simulations were performed using sinusoidal wind speed input. Its performance was compared with that using the SDC previously designed for this turbine [13]. First, the simulation was performed under sinusoidal wind speed input with frequency that matched the drivetrain eigenfrequency. Initialization for 50 seconds with constant wind speed input of 9 m/s was performed to allow the turbine to reach steady-state condition, before the mean wind speed was increased to 10 m/s with amplitude of 1 m/s, as shown in Figure 75.

Figure 77 shows a wind turbine response in terms of rotor speed and high-speed shaft power, respectively. The blue graph represents the response without any resonance protection, the red graph represents the response with the SDC, and the green one represents the response with the proposed VIDC. As expected from below-rated operation of variable-speed turbine, the rotor speed varied to optimally capture the available wind power. However, without any virtual inertia implemented, the output power showed large fluctuations as a result of the amplified load transmitted through the drivetrain. Although resonance delivers higher amplitude of power, these huge variations in power are not good for the grid and hence need to be compensated.

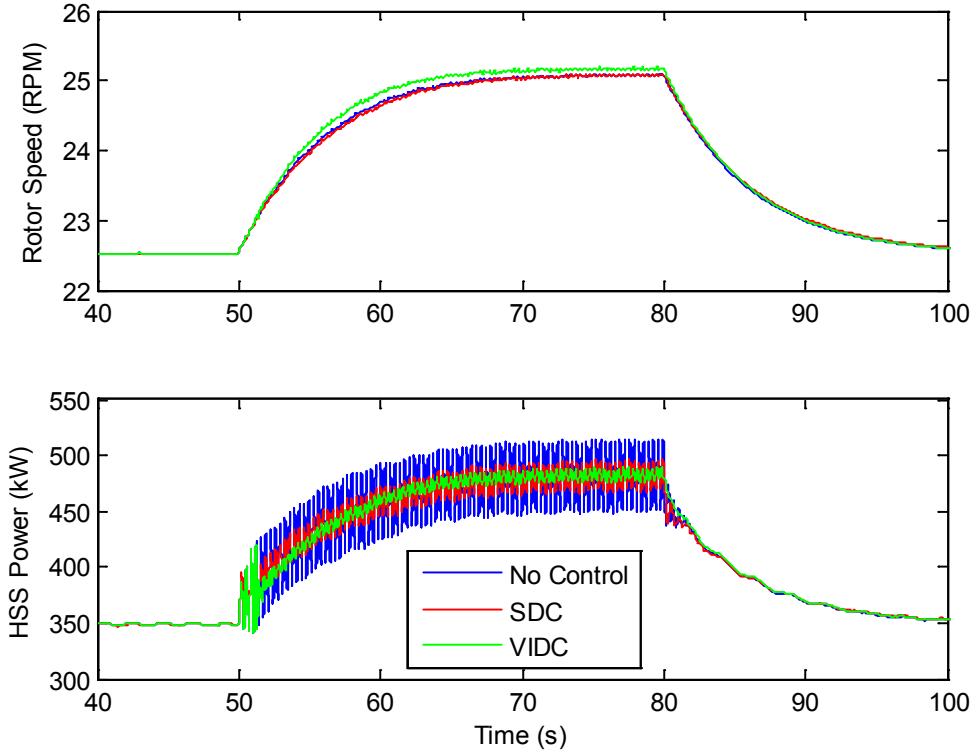


Figure 76. Wind turbine drivetrain response in terms of speed and power

Likewise, this resonance contributes to the significant load experienced by the drivetrain. Figure 77 shows the drivetrain angle of twist under this harmonic excitation. For a two-mass model, the angle of twist is defined as the difference between the angular position of one inertia body (in this case, the turbine rotor) and another (in this case, the generator), while taking into account the scaling effect of the gear ratio. Variations in the twist angle represent the amount of fatigue load experienced by the drivetrain, which had a similar profile with variations observed in the mechanical power, as shown in Figure 77.

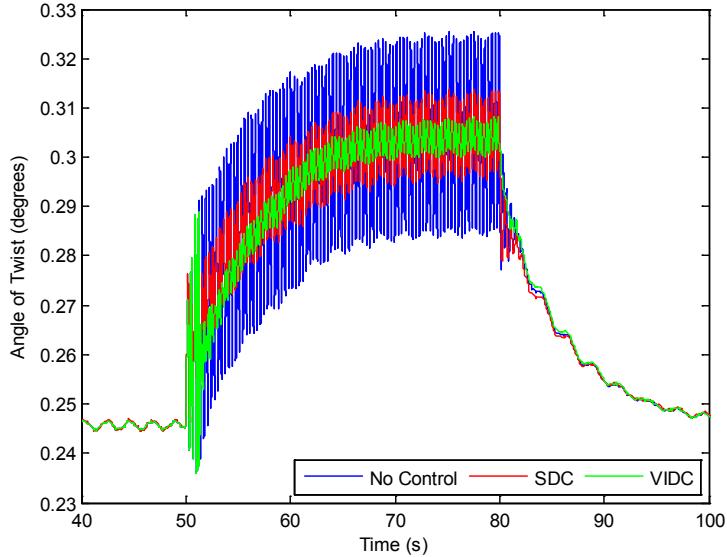


Figure 77. Drivetrain angle of twist under resonant excitation

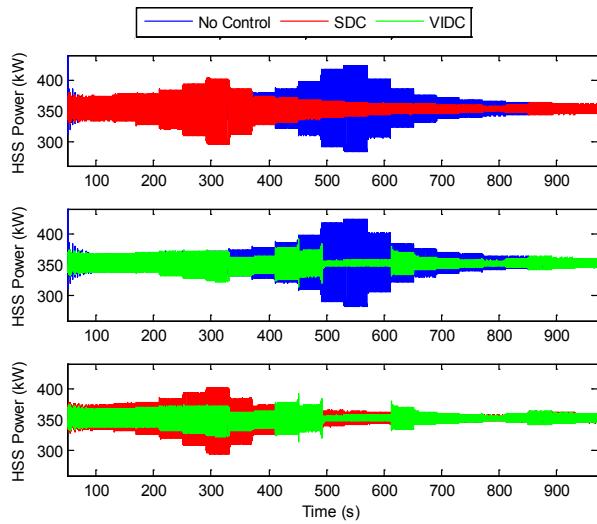


Figure 78. Power fluctuations under varying frequency excitations

Both SDC and VIDC are capable of attenuating the undesired fluctuations in both power and mechanical load. Although the SDC mechanism works by opposing the resonant response at the system eigenfrequency, VIDC has noticeable performance in resonance protection, which stems from shifting the effective eigenfrequency, thus preventing the system resonant response at its inherent eigenfrequency.

To further investigate the control performance under wider frequency range, a simulation was performed using wind speed input with varying frequency. This time, the sinusoidal wind speed input had an average wind speed of 9 m/s and an amplitude of 1 m/s. Again, an initialization for 50 seconds with a constant wind speed input of 9 m/s was performed. The frequency of excitation steadily increased from 0.6 Hz to 5 Hz with increments of 0.2 Hz, and each frequency excitation lasted for 40 seconds.

Figure 78 shows the power fluctuations of the drivetrain, which had a similar trend to the angle of twist. Without any controller, the peak response occurred at 3 Hz excitation, which was very close to the drivetrain eigenfrequency of 2.96 Hz. It occurred from 530 seconds to 570 seconds. With SDC, this peak response was reduced but with the side effect of peak response at a different frequency location. On the other hand, the proposed VIDC specifically eliminated resonant response around the eigenfrequency region and thus did not amplify the system response at other frequency locations.

7 Conclusion

In the early development, wind power generation was not considered to be a major player in the electric generation world. As the technology matures, the wind industry is making a lot of progresses that leads to better generators with excellent grid interfaces. The utility industry considers wind power generation an important type of generation, and the requirements for WTGs are becoming more demanding. As the level of wind power penetration increases, wind power plants will be required to have fault ride-through capability to ensure that the wind generation will not be disconnected from the grid during minor disturbances. Other requirements that will soon be enforced by transmission system operators include governor response capability, spinning reserves capability, and inertial response. With high demand on wind turbine manufacturers to support power system reliability, wind turbine manufacturers need to understand the impact of the grid disturbances and the impact of new requirements on the design of WTGs. Similarly, a power system planner needs to understand the limitation of WTGs to provide these ancillary services to support the grid. This project was inspired by these needs, and we attempted to combine the tools used by turbine manufacturers (FAST) and the tools used by power system planners (MATLAB/Simulink) to conduct a holistic approach to the design and analysis of wind turbines, both in terms of how turbines impact the grid and how the grid impacts the turbines.

Ongoing and future work includes verification and validation on the 2.5-MW or 5-MW dynamometer testing at NREL. We are also going to extend the Simulink model for Type 3 and 4 WTGs. As even-higher fidelity models are developed to give more insight into the dynamics of the drivetrain, further control strategies will be explored to attenuate unwanted loads and vibrations on turbines.

8 References

1. Jonkman, J.M.; Buhl, Jr., M.L. *FAST User's Guide*. NREL/TP-500-38230. Golden, CO: National Renewable Energy Laboratory, August 2005. Accessed November 2013: www.nrel.gov/docs/fy06osti/38230.pdf.
2. Bossanyi, E.A. *GH Bladed Version 3.51 User Manual*. 282/BR/010. Garrad Hassan and Partners Limited, June 2003. Accessed November 2013: http://ocw.tudelft.nl/fileadmin/ocw/courses/OffshoreWindFarmEnergy/res00099/User_Manual.pdf.
3. Larsen, T.J.; Hansen, A.M. *How 2 HAWC2: The User's Manual*. Risø-R-1597 (ver. 3-1). Roskilde, Denmark: Technical University of Denmark, December 2007. Accessed November 2013: www.risoe.dk/rispubl/reports/ris-r-1597.pdf.
4. Peeters, J. *Simulation of Dynamic Drivetrain Loads in a Wind Turbine*. Ph.D. Thesis. Leuven, Belgium: Department of Mechanical Engineering, Katholieke Universiteit Leuven, June 2006.
5. Oyague, F. *Gearbox Modeling and Load Simulation of a Baseline 750-kw Wind Turbine Using State-of-the-Art Simulation Codes*. NREL/TP-500-41160. Golden, CO: National Renewable Energy Laboratory, February 2009. Accessed November 2013: www.nrel.gov/docs/fy09osti/41160.pdf.
6. Helsen, J.; Vanhollebeke, F.; Coninck, F.D.; Vandepitte, D.; Desmet, W. "Insights in Wind Turbine Drivetrain Dynamics Gathered by Validating Advanced Models on a Newly Developed 13.2-MW Dynamically Controlled Test-Rig." *Mechatronics* (21), 2011; pp. 737–752.
7. The MathWorks, Inc. *Simscape User's Guide*. March 2012. Accessed November 2013: www.mathworks.cn/help/pdf_doc/physmod/simscape/simscape_ug.pdf.
8. Slootweg, J.G. *Wind Power: Modeling and Impact on Power System Dynamics*. Ph.D. thesis. 2003.
9. Heier, S. *Grid Integration of Wind Energy Conversion Systems*. New York: Wiley, 1998.
10. Buhl, Jr. M.L.; Wright, A.D.; Pierce, K.G. "Wind Turbine Design Codes: A Comparison of the Structural Response." 2000 American Society of Mechanical Engineers Wind Energy Symposium/38th American Institute of Aeronautics and Astronautics Aerospace Sciences Meeting and Exhibit Proceedings; January 2001, Reno, Nevada. AIAA-2000-0022; pp. 12–22.
11. Oyague, F. "Gearbox Reliability Collaborative Description and Loading." NREL/TP-5000-47773. Golden, CO: National Renewable Energy Laboratory, November 2011. Accessed November 2013: www.nrel.gov/docs/fy12osti/47773.pdf.

12. Wright, A.D.; Fingersh, L.J. *Advanced Control Design for Wind Turbines—Part I: Control Design, Implementation, and Initial Tests*. NREL/TP-500-42437. Golden, CO: National Renewable Energy Laboratory, March 2008. Accessed November 2013: www.nrel.gov/docs/fy08osti/42437.pdf.
13. Mandic, G.; Ghotbi, E.; Nasiri, A.; Oyague, F.; Muljadi, E. “Mechanical Stress Reduction in Variable-Speed Wind Turbine Drivetrains.” *2011 IEEE Energy Conversion Congress and Exposition Proceedings*; pp. 306–312.
14. Helsen, J.; Heirman, G.; Vandepitte, D.; Desmet, W. “The Influence of Flexibility Within Multibody Modelling of Multimegawatt Wind Turbine Gearboxes.” *International Conference on Noise and Vibration Engineering Proceedings*; Leuven, Belgium, 2008.
15. Helsen, J.; Vanhollebeke, F.; Vandepitte, D.; Desmet, W. “Optimized Inclusion of Flexibility in Wind Turbine Gearbox Multibody Model in View of Model Updating on Dynamic Test Rig.” *The 1st Joint International Conference on Multibody System Dynamics Proceedings*; Lappeenranta, Finland, 2011.
16. Asmus, P; Seitzler, M. “The Wind Energy Operation and Maintenance Report.” *Wind Energy Update*, February 2010.
17. Villaa, L.F.; Reñonesa, A.; Perána, J.R.; de Miguelb, L.J. “Statistical Fault Diagnosis Based on Vibration Analysis for Gear Test-Bench Under Nonstationary Conditions of Speed and Load.” *Mechanical Systems and Signal Processing* (29), 2012; pp. 436–446.
18. Oyague, F.; Butterfield, C.P.; Sheng, S. “Gearbox Reliability Collaborative Analysis Round Robin.” *American Wind Energy Association WINDPOWER 2009 Conference and Exhibition Proceedings*; Chicago, Illinois. Accessed November 2013: www.nrel.gov/docs/fy09osti/45325.pdf.
19. Oyague, F.; Gorman, D.; Sheng, S. “NREL Gearbox Reliability Collaborative Experiment Data Overview and Analysis.” *American Wind Energy Association WINDPOWER 2010 Conference and Exhibition Proceedings*; Dallas, Texas. Accessed November 2013: www.nrel.gov/docs/fy10osti/48232.pdf.
20. Helsen, J.; Vanhollebeke, F.; Marrant, B.; Vandepitte, D.; Desmet, W. “Multibody Modelling of Varying Complexity for Modal Behaviour Analysis of Wind Turbine Gearboxes.” *Renewable Energy* (36), 2011; pp. 3,098–3,113.
21. Guo, Y.; Keller, J.; Parker, R. “Dynamic Analysis of Wind Turbine Planetary Gears Using an Extended Harmonic Balance Approach.” *Conference on Noise and Vibration Engineering Proceedings*; September 17–19, 2012, Leuven, Belgium.
22. Guo, Y.; Parker, R.G. “Dynamic Modeling and Analysis of a Spur Planetary Gear Involving Tooth Wedging and Bearing Clearance Nonlinearity.” *European Journal of Mechanics A/Solids* (29), 2010; pp. 1,022–1,033.

23. Todorov, M.; Dovbrev, I.; Massouh, F. "Analysis of Torsional Oscillation of the Drivetrain in Horizontal Axis Wind Turbine." *Electromotion-2009-EPE Chapter "Electric Drives" Joint Symposium Proceedings*; July 1–3, 2009.
24. Xing, Y.; Moan, T. "Wind Turbine Gearbox Planet Carrier Modelling and Analysis in a Multibody Setting." *Wind Energy* (preprint), 2012.
25. Xing, Y.; Karimirad, M.; Moan, T. "Modelling and Analysis of Floating Spar-Type Wind Turbine Drivetrain." *Wind Energy* (preprint), 2013.
26. Deutsches Institut für Normung. *Calculation of Load Capacity of Cylindrical Gears*. DIN 3990. 1987.
27. International Organization for Standardization. ISO 6336-1. *Calculation of Load Capacity of Spur and Helical Gears*. Second edition 2006-09-01. Corrected version 2007-04-01.
28. Muljadi, E.; Mills, Z.; Foster, R.; Conto, J.; Ellis, A. "Fault Analysis at a Wind Power Plant for One Year of Observation," presented at the 2008 IEEE Power Engineering Society General Meeting, 20-24 July 2008, Pittsburgh, Pennsylvania
29. Brochu, J.; Gagnon, R.; Larose, C. "Generic Equivalent Collector System Parameters for Large Wind Power Plant." *IEEE Transactions on Energy Conversion*. (26:2), June 2011; pp.542–49.
30. Muljadi, E.; Butterfield, C.P.; Ellis, A; Mechenbier, J.; Hocheimer, J.; Young, R.; Miller, N.; Delmerico, R.; Zavadil, R.; Smith, J.C. "Equivalencing the Collector System of a Large Wind Power Plant." Prepared for the IEEE Power Engineering Society, General Meeting, June 12–16, 2006, Montreal, Quebec.
31. Muljadi, E.; Pasupulati, S.; Ellis, A.; Kosterov, D." Method of Equivalencing for a Large Wind Power Plant with Multiple Turbine Representation." Prepared for the IEEE Power Engineering Society, General Meeting, July 20–24, 2008, Pittsburgh, Pennsylvania.
32. Muljadi, E.; Gevorgian, V.; Singh, M.; Santoso, S. "Understanding Inertial and Frequency Response of Wind Power Plants." Prepared for the IEEE Symposium on Power Electronics and Machines in Wind Applications, July 16–18, 2012, Denver, Colorado.
33. Muljadi, E.; Singh, M.; Gevorgian, V. "Fixed-Speed and Variable-Slip Wind Turbines Providing Spinning Reserves to the Grid." Prepared for the IEEE Power and Energy Society General Meeting, July 21–25, 2013, Vancouver, British Columbia.
34. Singh, M.; Vyas, M.; Santoso, S. "Using Generic Wind Turbine Models to Compare Inertial Response of Wind Turbine Technologies." *IEEE Power and Energy Society General Meeting Proceedings*; July 2010; pp. 1–7.

9 Bibliography

Jonkman, J.; Buhl, M., Jr. (2005). *FAST User's Guide*. NREL/TP-500-38230. Golden, Colorado: National Renewable Energy Laboratory.

Krause, P.C. (1986). *Analysis of Electric Machinery*. New York: McGraw-Hill.

Ong, C.M. (1998). *Dynamic Simulation of Electric Machinery*. Upper Saddle River, New Jersey: Prentice-Hall.

10 Appendices

10.1 Appendix A: Wind Turbine Drivetrain Modeling in Simscape/SimDriveline and Interfacing with FAST in Simulink—User’s Guide

10.1.1 Introduction

This guide focuses on enhancing the capability of the FAST code through the integration of a dynamic model of a drivetrain built using Simscape/SimDriveline simulated in the MATLAB/Simulink environment. The developed model can be used to design drivetrain components and predict their dynamic responses.

To build the model, it is required that the user has the SimScape and SimDriveline Library installed with the MATLAB/Simulink package. To maintain the focus of this guide, it is assumed that the user has the FAST code installed and is familiar with running FAST simulations in the Simulink environment using the provided Simulink interface blocks. The latest version of the FAST code as well as the *FAST User’s Guide* can be downloaded from <http://wind.nrel.gov/designcodes/simulators/fast/>. Simulations in this guide were run using FAST v7.01.00a-bjj (updated on February 22, 2012) in MATLAB version 7.14 (R2012a), Simulink version 7.9, and SimDriveline version 2.2.

10.1.2 Simscape and SimDriveline Basics

Simscape provides an environment within Simulink for modeling and simulating the physical systems spanning the mechanical, electrical, hydraulic, and other physical domains. It provides the fundamental building blocks from these domains that can be assembled into models of physical components. Simscape components use physical connections to represent the bidirectional flow of power, which is different from the conventional, unidirectional flow of signals in the Simulink environment. Because of this difference in connection, it is important to note that the Simscape blocks or components have different input and output ports from those of the conventional Simulink blocks. Therefore, it is impossible to directly connect the Simscape blocks with the Simulink ones without the converter block. In Simscape, □ port represents the physical connections (either mechanical, electrical, hydraulic, or other forms of physical connections) and ▷ port represents the unidirectional input or output physical signal.

SimDriveline is part of Simscape, which provides the component libraries for modeling and simulating one-dimensional mechanical systems. It includes the models of rotational and translational components, such as gears, lead screws, and clutches. Components in this library are generally used to model the transmission of mechanical power in helicopter drivetrains, industrial machinery, vehicle power trains, and other applications. Full details about this library, including examples and tutorial videos, are available at www.mathworks.com/products/simdrive/.

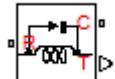
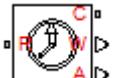
10.1.2.1 Basic Building Blocks

In this guide, the drivetrain was modeled considering only the torsional degree of freedom of each inertial body. Several Simscape components that were extensively used in building the model are listed below and briefly described.

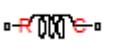
The following component can be found at the Simulink library under *Simscape > Foundation Library > Mechanical > Mechanical Sources*:

Name	Icon	Description
Ideal Torque Source		It generates torque difference between R and C that is proportional to the input signal S .

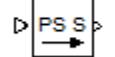
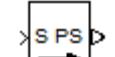
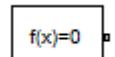
These components can be found at the Simulink library under *Simscape > Foundation Library > Mechanical > Mechanical Sensors*:

Name	Icon	Description
Ideal Torque Sensor		It converts an across variable measured between R and C into output signal T proportional to the torque.
Ideal Rotational Motion Sensor		It converts an across variable measured between R and C into output signal W proportional to the angular velocity, or output signal A proportional to the angular position.

These components can be found at the Simulink library under *Simscape > Foundation Library > Mechanical > Rotational Elements*:

Name	Icon	Description
Inertia		It represents the inertia body with user-defined rotational inertia and initial velocity.
Rotational Spring		It represents a rotational linear spring with user-defined rotational stiffness and initial deformation (to define precompressed or pretensioned spring).
Mechanical Rotational Reference		It represents rigid connection to the reference frame (ground).

These components can be found at the Simulink library under *Simscape > Utilities*:

Name	Icon	Description
PS-Simulink Converter		It converts a Simscape physical input signal into a Simulink output signal.
Simulink-PS (Physical Signal) Converter		It converts a Simulink input signal into a Simscape physical signal.
Solver Configuration		It specifies the solver parameters that the model needs before simulation starts. Each topologically distinct Simscape block diagram

		requires exactly one Solver Configuration block.
--	--	--

These components can be found at the Simulink library under *Simscape > SimDriveline > Gears*:

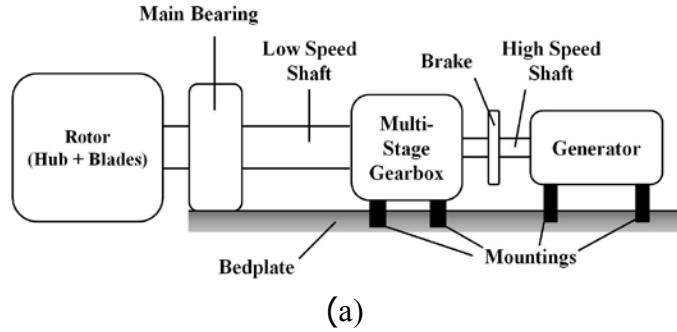
Name	Icon	Description
Simple Gear		It represents a gear stage that constrains two connected driveline axes, corotating with the user-defined fixed-gear ratio.
Planetary Gear		It represents a set of carrier, ring, planet, and sun-gear wheels.

To build higher-fidelity models of the planetary gear stage, the aforementioned Planetary Gear block was not sufficient. Thus, the components listed below, which are the basic elements of a planetary gear stage, were used instead. These components can be found at the Simulink library under *Simscape > SimDriveline > Gears > Planetary Subcomponents*:

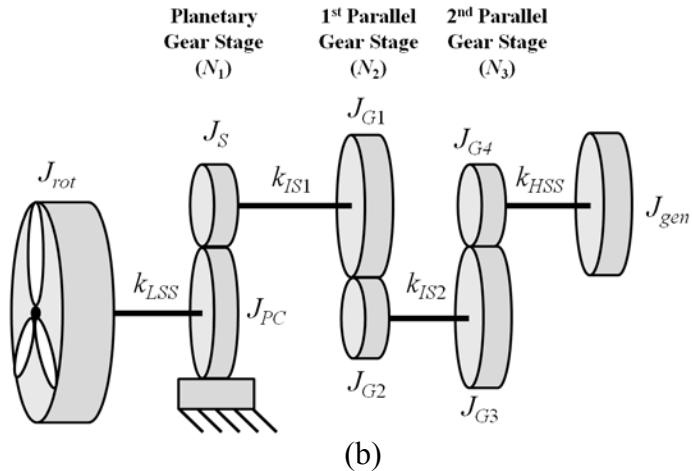
Name	Icon	Description
Ring-Planet		It represents a set of carrier, planet, and ring gears. The planet is connected to and rotates with respect to the carrier. The planet and ring corotate with the user-defined fixed gear ratio.
Sun-Planet		It represents a set of carrier, planet, and sun gears. The planet is connected to and rotates with respect to the carrier. The planet and sun corotate with the user-defined fixed gear ratio and in the same direction with respect to the carrier.

10.1.2.2 Example of Drivetrain Model

Figure A.1 (a) shows a schematic of a wind turbine drivetrain, which is called the modular drivetrain. The multistage gearbox consists of a low-speed planetary gear stage and two high-speed parallel gear sets, with two intermediate shafts. Figure A.1 (b) illustrates the multistage representation of this drivetrain. It has one planetary gear stage and two parallel gear stages.



(a)



(b)

Figure A.1. (a) Modular wind turbine drivetrain and (b) multistage representation of the drivetrain

Each shaft was modeled as an ideal massless linear torsional spring. The eight inertial bodies considered in this model were:

- Rotor (hub and blades) inertia, J_{rot} . The blades are considered rigid to result in constant rotor inertia.
- Planet gears and carrier inertia, J_{PC} . All of the planet gears and the carrier are considered one inertial body corotating at the rotational axis of the carrier shaft.
- Sun gear inertia, J_S .
- Parallel gears inertia, J_{Gi} .

The ring gear was assumed to be rigidly connected to the gearbox housing; therefore, its inertia was not considered and the ring gear modeled was an inertial frame.

Despite having eight rotational bodies, this model had five *effective* inertias connected through four linear springs, as shown in Figure A.2. The five effective inertias, with respect to the low-speed side of the multistage gearbox, were calculated as follows:

$$J_1 = J_{rot}$$

$$J_2 = J_{PC} + N_1^2 J_S$$

$$J_3 = N_1^2 J_{G1} + (N_1 N_2)^2 J_{G2}$$

$$J_4 = (N_1 N_2)^2 J_{G3} + (N_1 N_2 N_3)^2 J_{G4}$$

$$J_5 = (N_1 N_2 N_3)^2 J_{gen}$$

N_1 is the gear ratio of the planetary gear stage; whereas N_2 and N_3 are the gear ratios of the first and second parallel gear stages, respectively. On the other hand, the stiffness, with respect to the low-speed side of the multistage gearbox, was calculated as follows:

$$k_1 = k_{LSS}$$

$$k_2 = N_1^2 k_{IS1}$$

$$k_3 = (N_1 N_2)^2 k_{IS2}$$

$$k_4 = (N_1 N_2 N_3)^2 k_{HSS}$$

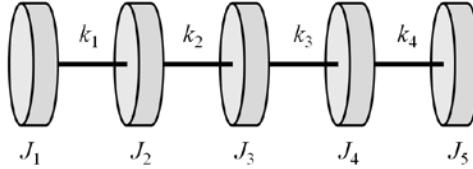


Figure A.2. Five-inertia representation of a wind turbine drivetrain

LSS represents the low-speed shaft. IS1 and IS2 stand for the first and second intermediate shafts, respectively. HSS represents the high-speed shaft.

Figure A.3 shows the SimDriveline model built in the Simulink environment for the drivetrain depicted in Figure A.1 (b) using the components listed before. It is very important to note that the Simple Gear block of SimDriveline assumes a speed reducer configuration, which is commonly used in automotive transmissions. However, in a wind turbine drivetrain, the gear stage is used to step up the speed. This issue can be tackled by flipping the Simple Gear block left-right (by pressing Ctrl+I) to result in the Simple Gear block orientation shown in Figure A.3. Therefore, the gear-teeth ratio can be kept as N_i , where $N_i > 1$. Alternatively, the original orientation of the Simple Gear block can be maintained (as shown in the component list table), but the gear-teeth ratio needs to be defined as $1/N_i$ (as illustrated in Figure A.4). In both cases, it is important to keep the output shaft rotating in the opposite direction of the input shaft.

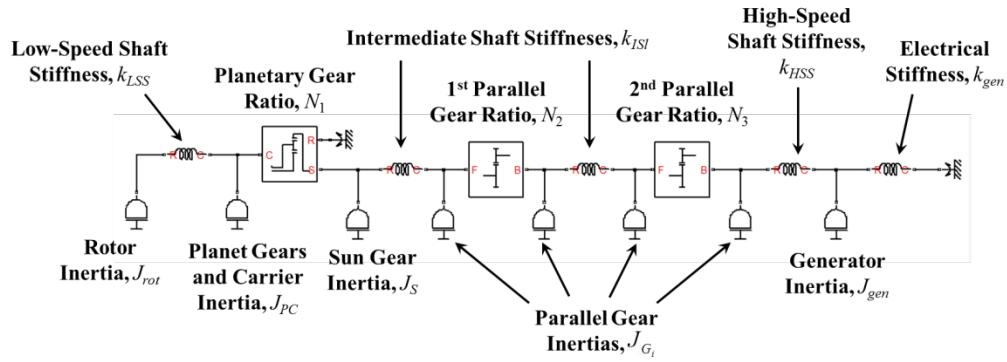


Figure A.3. SimDriveline model of a wind turbine drivetrain

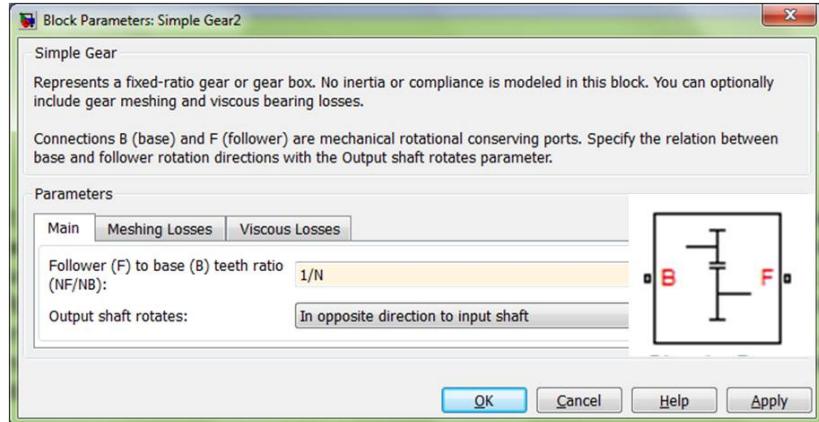


Figure A.4. Assigning parallel gear-teeth ratio to the wind turbine drivetrain

10.1.2.3 Actuators and Sensors

In a wind turbine drivetrain, torques come from both free ends: aerodynamic torque from the rotor side and electromagnetic torque from the generator side. The torque excitation can be injected into the model using the Ideal Torque Source block, as shown in Figure A.5. For illustration purposes, the torque input commands are represented by the Step block, although, in general, this input can be any Simulink signal. In each Ideal Torque Source block, the reference port **C** is connected to the inertial frame. In the Simulink-PS Converter block, the user must remember to define the proper unit for the input signal.

The drivetrain components shown in Figure A.3, located between the rotor and generator inertias, comprise the Drivetrain block shown in Figure A.5.

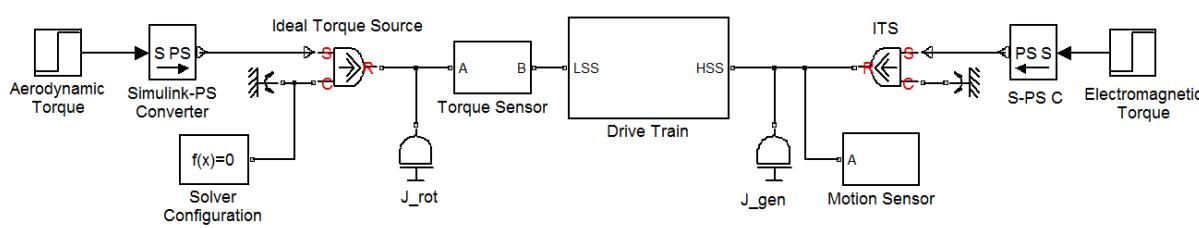


Figure A.5. Drivetrain model with torque inputs and measurement sensors

Measurements can be taken at any location within the model to monitor the motion and load, using the Ideal Rotational Motion Sensor and Ideal Torque Sensor, respectively. One important measurement is the rotational speed of the generator. This measurement serves as the input to the electrical generator model to determine the generator electromagnetic torque.

Figure A.6 shows the internal component of the Motion Sensor block of Figure A.5, which is used to measure the generator speed. As the absolute angular speed (i.e., with respect to the fixed rotational frame) is measured, the sensor has only one physical connection port to the drivetrain model, whereas the reference port **C** is connected to the ground.

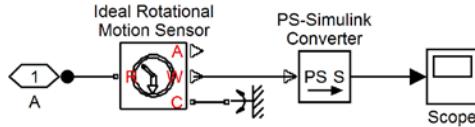


Figure A.6. Motion sensor to measure the rotational speed of the generator

As described later in this guide, another important measurement is the rotor restoring/opposing torque. Figure A.7 shows the internal component of the Torque Sensor block of Figure A.5. In SimDriveline, the torque is measured by connecting the sensor ports along the physical connections, similar to the concept of current measurement in the electrical circuit. The PS-Simulink Converter block is connected to the output of each sensor. The output of this converter block can be connected to any compatible Simulink block.

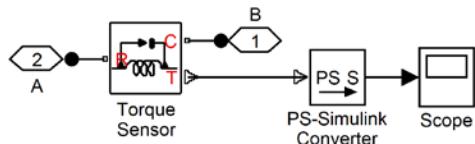


Figure A.7. Torque sensor to measure the rotor-opposing torque

In every converter block, the unit of the input or output signal is required. Users can define any compatible units to match the simulation criteria/preference. The Converter block will perform an internal conversion to yield the defined unit.

10.1.2.4 Solver Configuration

Users must remember to connect exactly one Solver Configuration block at any location within the drivetrain model. The location shown in Figure A.5 is an example. Settings for this *local* Solver Configuration block can be left as its default. The default global solver for Simulink is ode45. However, for physical models, MathWorks recommends implicit *global* solvers, such as ode14x, ode23t, or ode15s. We recommend ode15s to simulate pure SimDriveline models because of its accuracy and ode23tb to simulate the integrated drivetrain models (if the SimDriveline model is later combined with general Simulink and SimPowerSystems blocks) because of its much higher computational efficiency.

The solver setting can be modified through the Configuration Parameters window (from the Simulink tool bar: *Simulation > Configuration Parameters* or by pressing Ctrl+E). Under the Solver Options, the user can set the Type to the Variable-step and the Solver to ode15s (stiff/NDF) or ode23tb (stiff/TR-BDF2). To avoid numerical instability, it is advised to change the Maximum Order, under Solver Options, from 5 to 2. Figure A.8 highlights the necessary modifications to the Configuration Parameters.

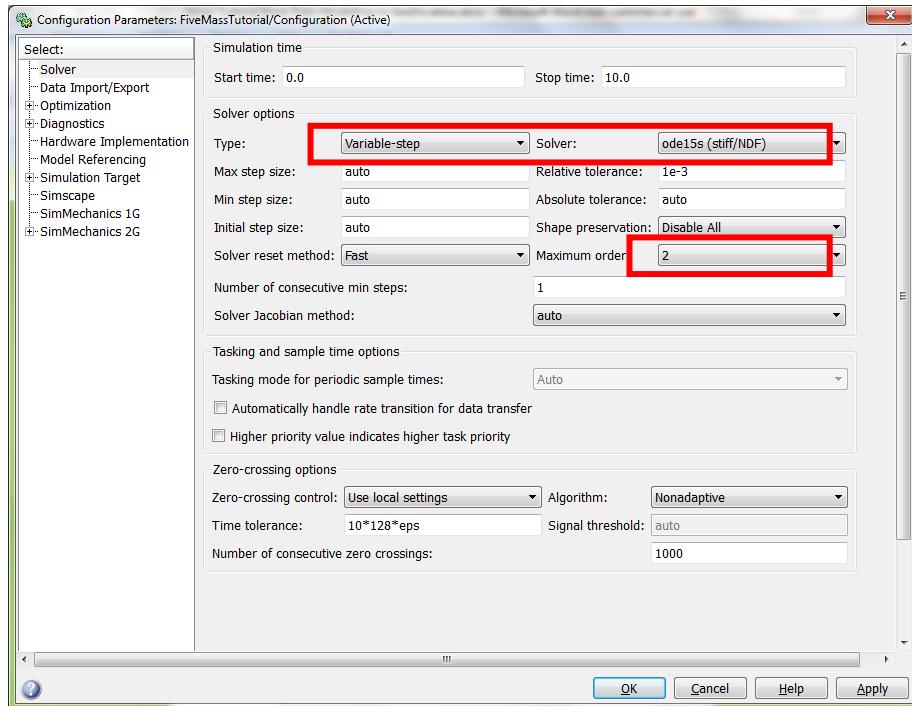


Figure A.8. Modifications to the Configuration Parameters for the drivetrain simulation

10.1.2.5 *Eigenfrequency Analysis*

One important aspect of dynamic model analysis is to find the eigenfrequencies of the drivetrain. This analysis can be performed using the Control System Toolbox. This toolbox normally comes in the basic MATLAB/Simulink package, so most users should find this toolbox already installed.

The first step in this analysis is to determine the input and output of the system. In a drivetrain system, the aerodynamic and electromagnetic torques are the inputs, whereas the angular position of an inertial body is normally defined as the output for vibration analysis. It is important to heed the measurement units of the PS-Simulink and Simulink-PS Converter blocks.

As an illustration, the aerodynamic torque is defined as the input. This is done by right-clicking on the signal line connecting the Step Aerodynamic Torque block to the Simulink-PS Converter blocks (as highlighted in Figure A.9), selecting the Linearization Points, and choosing the Input Point.

The generator angular position is defined as the output. This is done by first setting the motion sensor to measure the angular position, as shown in Figure A.10. Similar to defining the input point, the output of the system can be defined by right-clicking on the signal line connecting the PS-Simulink Converter to the Scope (as highlighted in Figure A.10), selecting the Linearization Points, and choosing the Output Point.

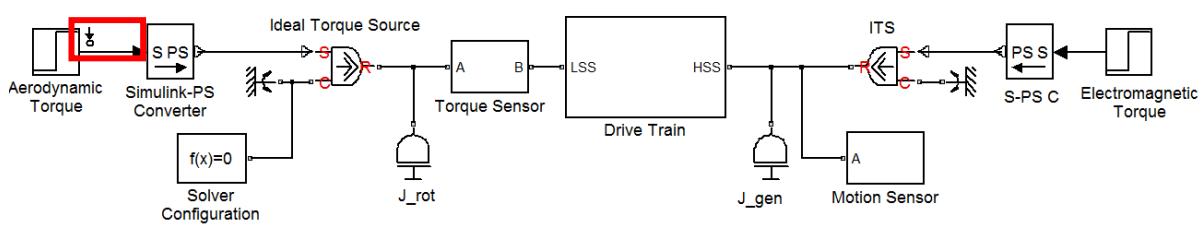


Figure A.9. Defining the input point for eigenfrequency analysis

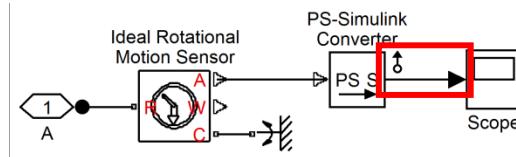


Figure A.10. Defining the output point for eigenfrequency analysis

After setting up the input and output points inside the drivetrain model, the user can run the following code lines to perform the linear frequency analysis:

```

mdl='ModelName'; % set to file name of the simulink model, omit the .mdl
io=getlinio(mdl); % to get input-output signals of the 'mdl' model
op=operspec(mdl);
op=findop(mdl,op); % to calculate model operating point

lin=linearize(mdl,op,io); % to compute state space model of linearized
                           system

figure
P=bodeoptions; % for bode plotting preference
P.FreqScale ='linear';
P.Grid='on';
P.FreqUnits='Hz';
P.PhaseVisible='off';
h=bodeplot(lin,P);

```

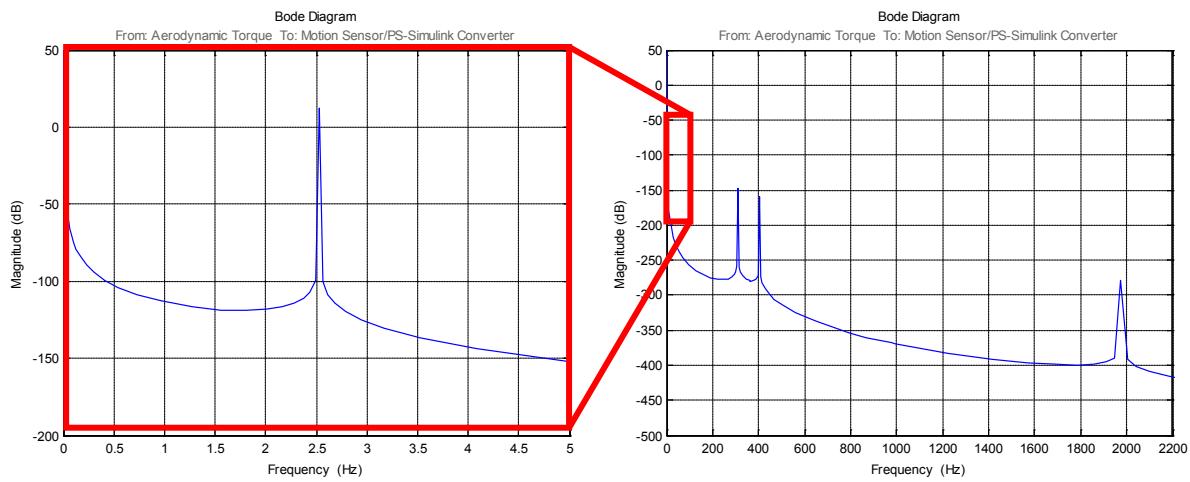


Figure A.11. Frequency response function of the drivetrain model

In general, first the codes find the equilibrium operating points of the model, based on its initial conditions. Then the codes perform linearization based on these operating conditions and the input-output configuration of the model. The result of this linearization, in the form of linear state space representation, can be used to perform the eigenfrequency analysis. In this example, it was done by plotting the Bode plot of the system. It is important to note that because the drivetrain model is already linear, the operating point will not affect the analysis result.

Figure A.11 shows the frequency response function (in Bode plot) of the drivetrain model. Peaks of the frequency indicate the global eigenfrequencies of the system. There are five peaks because the model has effectively five inertial bodies, as shown in Figure A.2. They are 0 Hz, 2.52 Hz, 312 Hz, 402 Hz, and 1.96 kHz. The user can define the Electromagnetic Torque as the model input and take the angular position of any inertial body (e.g., the first parallel gear) as the output. The resulted peak frequencies (i.e., the eigenfrequencies) will be the same.

10.1.3 Torsional Model of the Gear Box

In the above elementary drivetrain models, the meshing gear was modeled as an ideal static gain for torque and speed transmission. In reality, the gear transmission error, which is defined as the difference between the actual and ideal angular position of the rotating gear, contributes to the dynamics of the meshing gears. This transmission error is mainly a result of the gear-tooth elastic deformation. This phenomenon contributes to the definition of gear meshing stiffness.

This guide focuses on building a purely torsional model of the gear stages with constant meshing stiffness. Model development on both the planetary and parallel gear stages is discussed.

10.1.3.1 Parallel Gear

Figure A.12 (a) shows a parallel torque-reducing (i.e., speed-increasing) gear set, which is commonly employed in wind turbine drivetrains. Figure A.12 (b) represents its flexible equivalent, in which the meshing stiffness acts on the line of action of the meshing gears. The

constant meshing stiffness k_{mesh} can be represented as the function of geometric and material properties of the input gear:

$$k_{mesh} = k_{gear} (r_{b1} \cos \beta)^2$$

where r_{b1} is the base circle radius of the input gear and β is the helix angle of the gears. The gear-tooth stiffness k_{gear} can be determined according to the international standards, such as the DIN 3990 [26] and ISO 6336 [27].

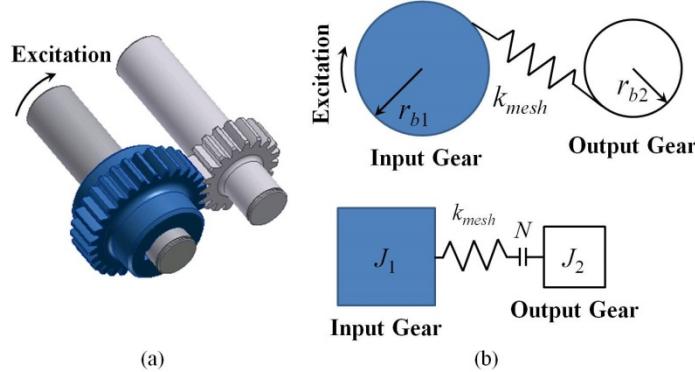


Figure A.12. Parallel gear stage and gear meshing stiffness representations

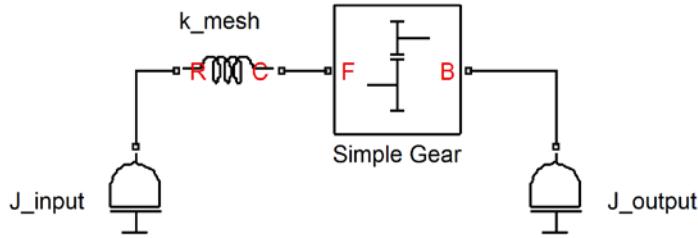


Figure A.13. Parallel gear stage representation in SimDriveline

Figure A.13 shows the parallel gear set model of Figure A.12 (b) in SimDriveline. As highlighted before, the Simple Gear block has been flipped to represent the torque reducing configuration.

10.1.3.2 Planetary Gear

Figure A.14 shows a planetary gear set with three planet gears. The rotational input is from the carrier, which then transfers the rotational motion from the planet gears to the sun gear. In typical wind turbine drivetrain, the ring gear is stationary. The ring gear can be modeled to have flexible coupling with the rigid gear housing. The meshing stiffness between the planet and ring gears as well as between the planet and sun gears can be modeled similarly to that of a parallel gear set, with the planet gear acting as the input gear.

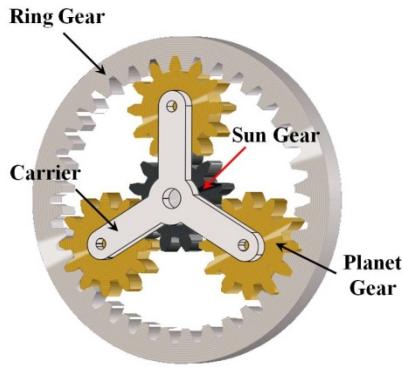


Figure A.14. Planetary gear set with three planet gears

A planet gear meshing with a sun gear and a ring gear in the SimDriveline comprises a meshing set, as shown in Figure A.15. This model can be adapted for a three-planet gear set (as shown in Figure A.16), or any M equispaced planet gear set, by duplicating the meshing sets.

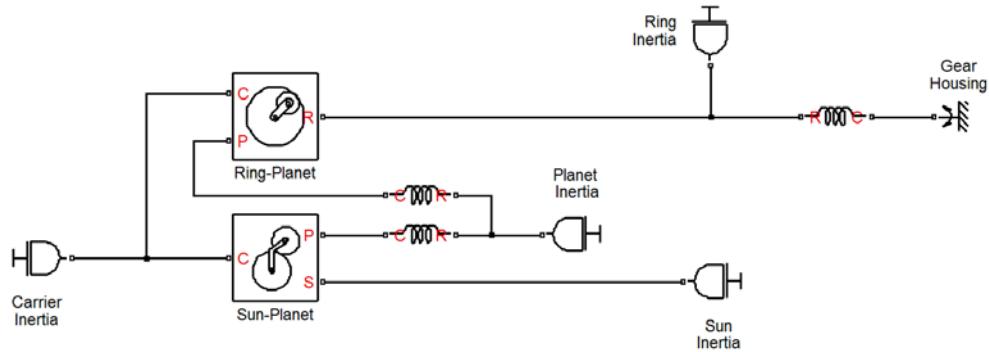


Figure A.15. One meshing set of a planetary gear set

Frequency analysis can be performed on the planetary gear set to validate the model. Properties of the planetary gear set used in [4], listed in Table A.1, can be used as an example.

Table A.1. Properties of a Planetary Gear Set

	Sun	Planet	Carrier	Ring
Inertia ($\times 10^{-3}$)	0.58 kg m^2	1.53 kg m^2	49.2 kg m^2	56.7 kg m^2
Base Radius	38.7 mm	50.2 mm	96.9 mm	137.5 mm
Gear Stiffness	$k_{gear} = 5 \times 10^8 \text{ N/m}$			
Torsional Stiffness	$k_{housing} = 19 \times 10^6 \text{ Nm/rad}$			
Helix Angle	$\beta = 0^\circ$			

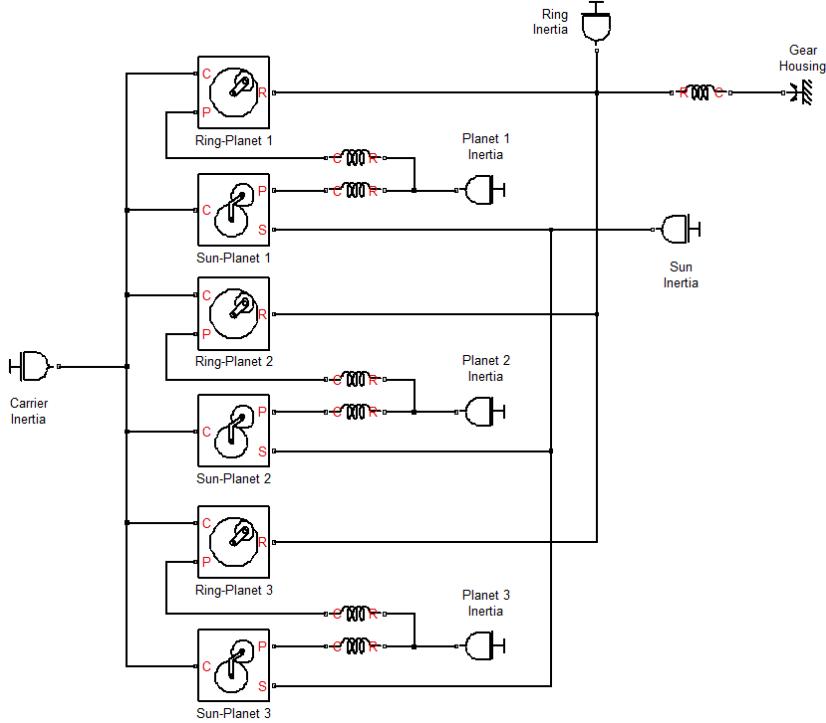


Figure A.16. SimDriveline model of a planetary gear stage with three planet gears

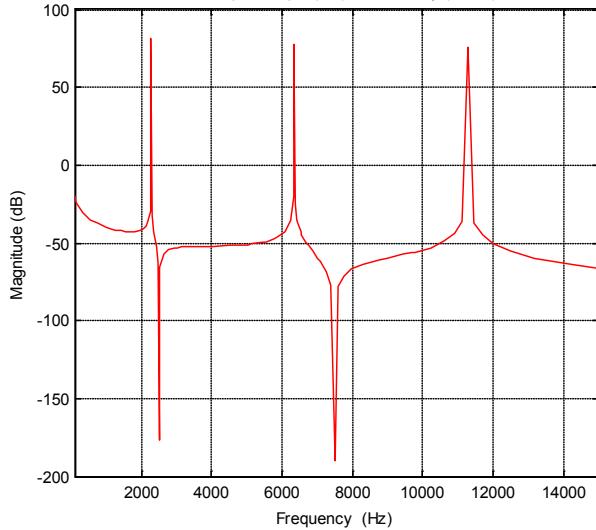


Figure A.17. Frequency response function of a three-planet planetary gear stage

The steps to perform eigenfrequency analysis on any SimDriveline models have been discussed previously. In [4], the frequency response function was produced by inputting the torque at the carrier and taking the rotational speed of the sun gear as the output. Figure A.17 shows the frequency response function of the planetary gear set in Table A.1 that has three planet gears. Table A.2 show good agreements between the eigenfrequencies of the SimDriveline model and the published ones with many different planet gears.

Table A.2. Comparison of Eigenfrequencies of Planetary Gear Sets

3 Planet Gears		
Mode	SimDriveline Model	Peeters [4]
1	2.273 kHz	2.217 kHz
2	6.340 kHz	6.159 kHz
3	11.296 kHz	11.205 kHz
4 Planet Gears		
Mode	SimDriveline Model	Peeters [4]
1	2.207 kHz	2.138 kHz
2	6.911 kHz	6.688 kHz
3	12.699 kHz	12.577 kHz
5 Planet Gears		
Mode	SimDriveline Model	Peeters [4]
1	2.153 kHz	2.059 kHz
2	7.403 kHz	7.105 kHz
3	13.980 kHz	13.810 kHz

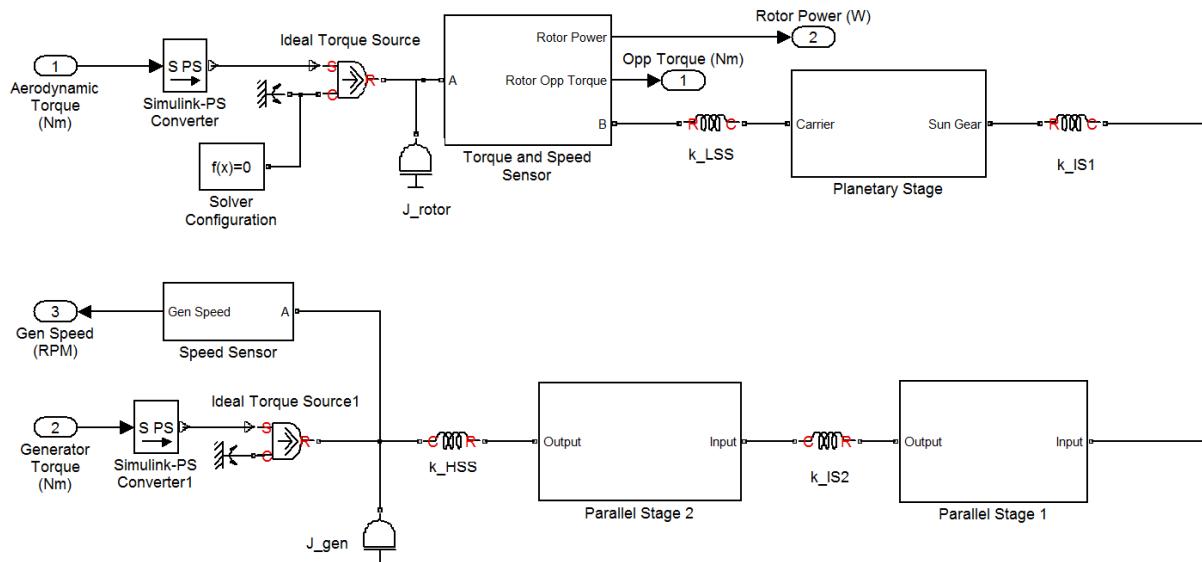


Figure A.18. Drivetrain model readily integrated with FAST

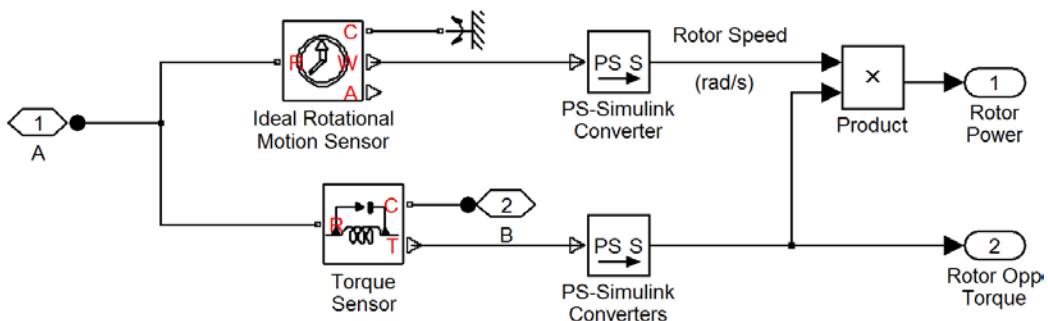


Figure A.19. Torque and speed sensor at the low-speed shaft

10.1.4 Integration with FAST Code

Figure A.18 shows an example of the overall drivetrain model with one planetary and two parallel gear stages. This model is ready to be integrated with the FAST aeroelastic code. The rotor inertia is calculated by assuming rigid blades of a turbine. The Planetary Stage block consists of the configuration shown in Figure A.15; whereas each of the Parallel Stage block consists of the configuration shown in Figure A.13. The Speed Sensor block consists of the configuration shown in Figure A.6. It measures the rotational speed of the generator, which is the input for the electric generator mode.

The Torque and Speed Sensor block consists of the configuration shown in Figure A.19. It measures the rotor-opposing torque and the rotational speed of the low-speed shaft. The product of both measurements is the mechanical power transmitted through the shaft. Both torque and power are the required inputs for FAST. The drivetrain model takes two input torques: the aerodynamic torque and the electromagnetic torque of the generator.

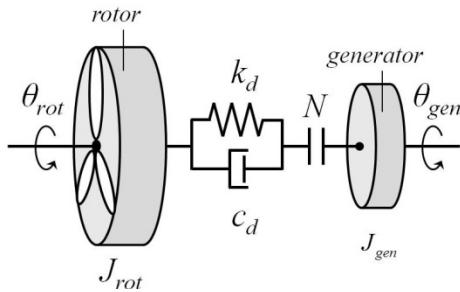


Figure A.20. Two-mass drivetrain model in FAST

10.1.4.1 FAST Inherent Drivetrain Model

Figure A.20 illustrates the configuration of the two-mass model. This model is inherently in FAST to represent wind turbine drivetrain dynamics. Inputs into the model are the five parameters: J_{rot} , k_d , c_d , N , and J_{gen} . Parameters of the two-mass model can be derived from the parameters of Figure A.2 as:

$$N = N_1 N_2 N_3$$

$$J_{gen} = \frac{J_2 + J_3 + J_4 + J_5}{N^2}$$

$$\frac{1}{k_d} = \frac{1}{k_1} + \frac{1}{k_2} + \frac{1}{k_3} + \frac{1}{k_4}$$

In the FAST input files .fst, those three parameters are GBRatio, GenIner, and DTTorSpr, respectively. The effective drivetrain torsional damping, c_d , can be determined experimentally through several braking events [5]. In FAST, this parameter is named DTTorDmp. The constant inertia of the hub (HubIner in FAST) and the varying inertia as a result of the flexible blades comprise the rotor inertia. The rotor inertia can be estimated by assuming rigid blades, which can be expressed as:

$$J_{rot} = J_{hub} + \sum_{i=1}^L m_i r_i^2$$

In FAST, the flexible blades are represented as a combination of the blade elements, as shown in Figure A.21. Each blade element is assumed to have a lumped mass (node) of m_i located at a distance of r_i from the rotor axis of rotation. Rigid blades mean that the distance of each node is fixed; whereas flexible blades allow independent motion of each node. The mass and distance of the nodes are available in the AeroDyn input .ipt file of the FAST input .fst file.

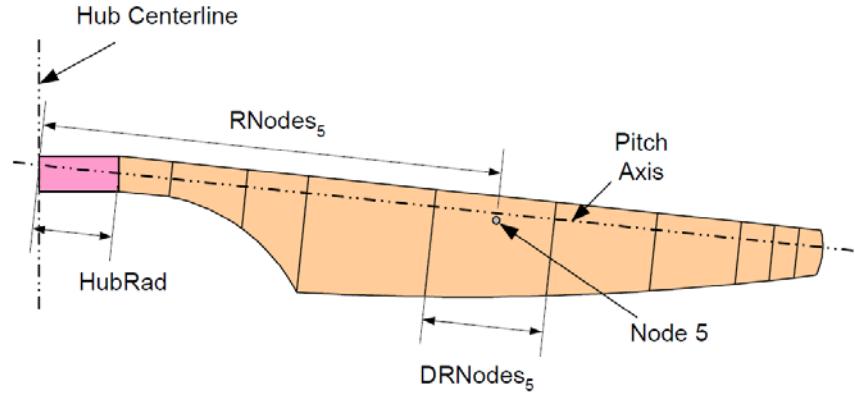


Figure A.21. Blade layout in FAST

10.1.4.2 FAST-Enhanced Drivetrain Model

Figure A.22 illustrates the proposed strategy to integrate the described drivetrain models into the two-mass model inherent inside the FAST CAE tool. For simplicity, the flexible modes of the other turbine components modeled inside FAST, such as those of the blades and tower, are not depicted.

In FAST, the two-mass drivetrain model is reduced to a single-mass model consisting of solely the rotor and the rigid shaft. The flexibility of the drivetrain is to be deactivated to simulate rigid transmission. This is done by first setting the DrTrDOF in the FAST input .fst file to FALSE. Second, set the gear ratio GBRatio to 1. Finally, set the generator inertia to 0. The equation of motion of the rotor can be expressed as:

$$J_{rot} \alpha_{rot} = Q_{aero} - Q_{opp}$$

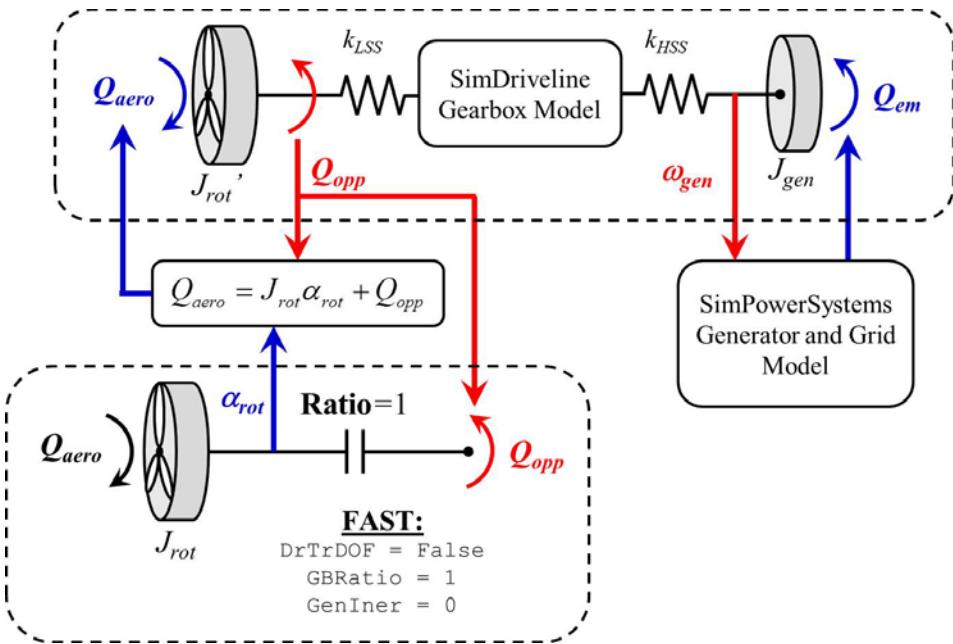


Figure A.22. Proposed schematic of integrating the drivetrain model with FAST

FAST internally calculates the input aerodynamic torque Q_{aero} from the defined wind profile. The current version of FAST does not provide the aerodynamic torque as an output, but it provides the rotor acceleration α_{rot} . Therefore, the aerodynamic torque Q_{aero} can be reconstructed using the equation of motion of the rotor as one of the inputs to the SimDriveline drivetrain model. In this process, the rotor inertia J_{rot} is assumed constant. This rotor inertia is replicated in the drivetrain model, as shown in Figure A.18 and Figure A.22, and is connected to the flexible low-speed shaft, the purely torsional gearbox model, the high-speed shaft, and the generator inertia.

As shown in Figure A.18 and Figure A.22, the drivetrain model provides the generator speed, which is taken as the input by the electrical machine and the grid model. This electrical model will then provide the generator electromagnetic torque to the drivetrain. The drivetrain model also provides the rotor-opposing torque Q_{opp} , which is required as an input to FAST as well as to calculate the aerodynamic torque Q_{aero} .

10.1.4.3 Building the Integrated Simulink Block

Figure A.23 shows the integrated wind turbine model in the Simulink environment. The SimDriveline model shown in Figure A.18 is in the red block.

The FAST S-Function block is in the orange block. This block is readily available within the downloaded FAST package as the *OpenLoop.mdl* shown in Figure A.24. The Pitch and Yaw Controller blocks are not affected by this integration. The Torque Controller, which gives the torque and power inputs to FAST, is replaced by the drivetrain model.

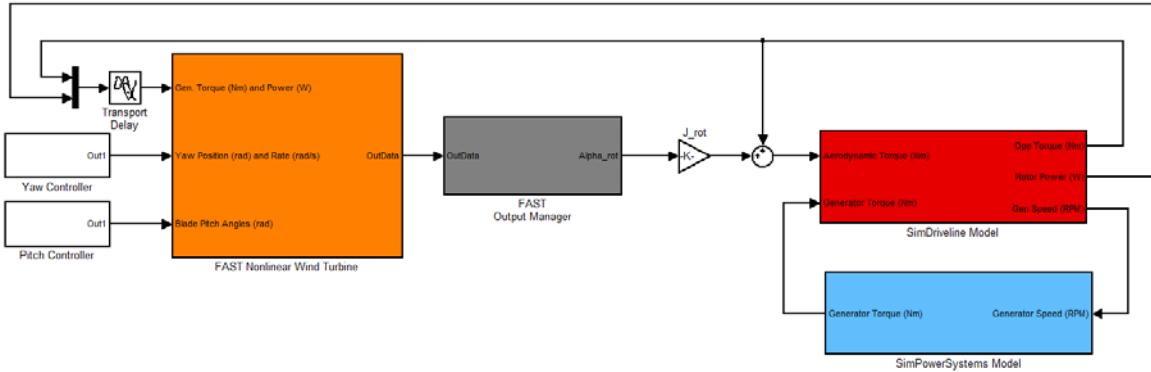


Figure A.23. Implementation of an integrated wind turbine drivetrain model

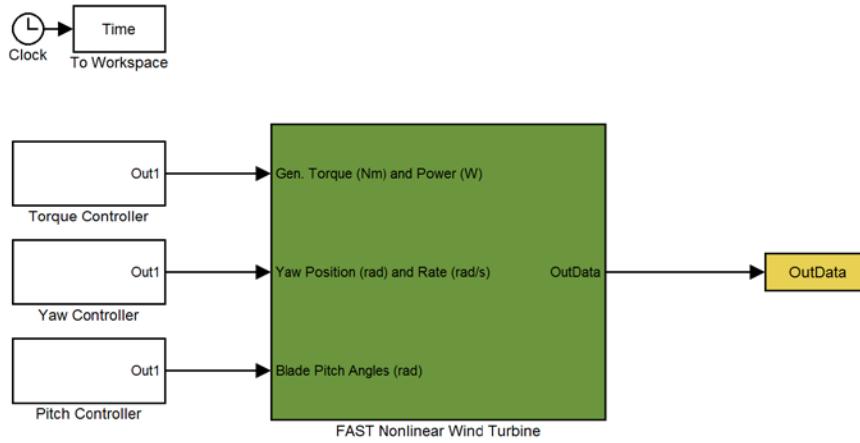


Figure A.24. Original FAST S-Function block as *OpenLoop.mdl*

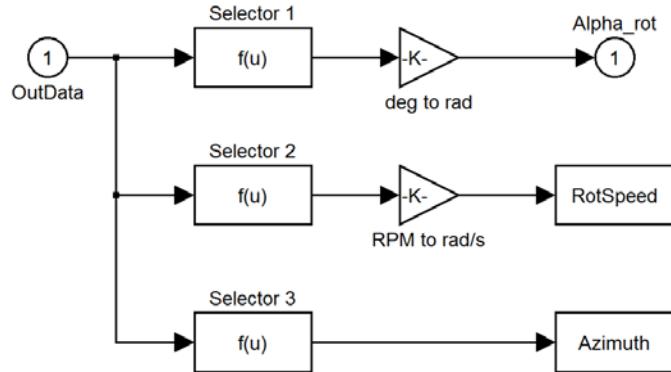


Figure A.25. FAST Output Manager block

The grey FAST Output Manager block contains parallel Fcn blocks, as shown in Figure A.25. This Fcn block can be found in the Simulink library under *Simulink > User-Defined Function*:

Name	Icon	Description
Fcn		It applies the specified mathematical expression to its input.

Each Fcn block can be used to isolate a variable of interest from the arrays of outputs. The FAST outputs are defined at the end of the FAST input .fst file, under the section OutList. As discussed before, the rotor acceleration is the required output from FAST. Therefore, the rotor acceleration must be listed in the FAST input .fst file. According to the *FAST User's Guide* (pp. 102), the LSSTipAxa corresponds to the measurement of the rotor acceleration in FAST. Other equivalent names of the parameters are LSSTipAxs, LSSTipA, and RotAccel. This measurement can be isolated by setting the Fcn block as:

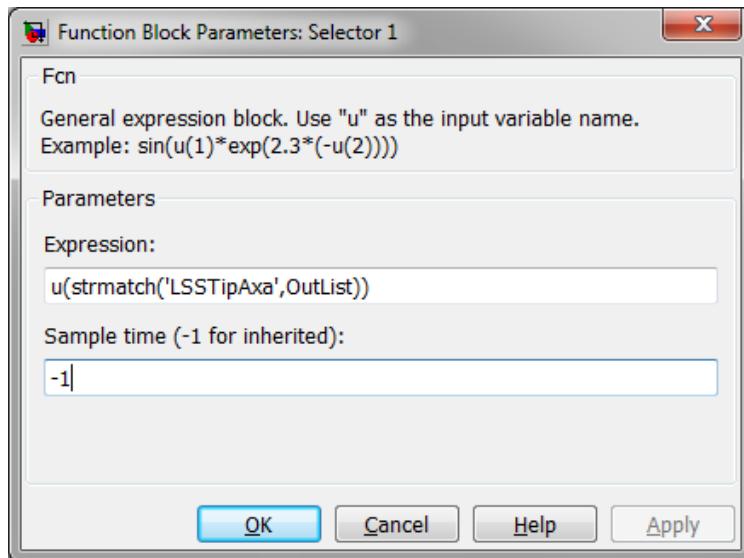


Figure A.26. Fcn block parameters to isolate the rotor acceleration

According to the guide, the FAST output for the rotor acceleration is in deg/s². In Figure A.25, a Gain block is used to perform the conversion from degrees to the SI unit of radian, by the multiplication of $\pi/180$. Figure A.25 also shows other Fcn blocks, each of which is used to isolate one variable of interest (e.g., the rotor speed). To do so, the LSSTipAxa can be replaced by another proper parameter name (e.g., LSSTipVxa).

The rotor acceleration in the SI unit is multiplied by the rotor inertia and then added by the rotor-opposing torque. The resulting aerodynamic torque is then fed into the drivetrain model.

Figure A.23 shows a Transport Delay block in the FAST input. The delay is necessary to break the algebraic loop. The user can define a very small time delay (e.g., 5 ms) to prevent any significant changes in the overall dynamics of the system. This block can be found from *Simulink > Continuous*:

Name	Icon	Description
Transport Delay	> 	It delays the input by a specified amount of time.

10.1.4.4 Integration With Generator and Grid Model

A high-fidelity model of a generator and grid is essential to monitor the internal loads of the drivetrain during grid events. The generator model takes the rotational speed as the input and gives the generator electromagnetic torque as the output. Therefore, the model can easily be adapted to the configuration of Figure A.23, in which the blue block contains the generator model.

As an illustration, an ideal generator torque law can be adopted:

$$Q_{em} = k\omega_{gen}^2$$

This torque law simulates the variable-speed operation of a wind turbine for maximum power tracking. This is applied mostly in Type 3 WTGs in the form of doubly-fed induction generators. This torque law can be implemented in Simulink using the Fcn Block. Figure A.27 shows the implementation of the torque law, in which u represents the input that is the generator speed. The constant k is dependent on the turbine rotor design and not on the generator model.

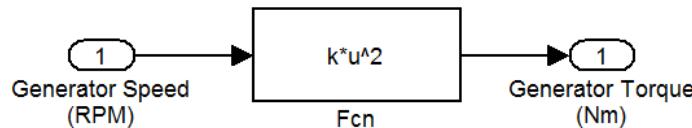


Figure A.27. Generator torque law for variable-speed variation

10.1.5 Simulation Checklist

It is assumed that the user is familiar with running FAST in the Simulink environment. This section summarizes the necessary modifications to run the simulation:

1. Build a SimDriveline model, as shown in Figure A.18, with well-defined drivetrain parameters. Listed below are the necessary elements of the model:
 - A Solver Configuration block is attached
 - An Ideal Torque Source block, with appropriate Simulink–Physical Signal (PS) Converter, on either side of the drivetrain model. One is to represent the aerodynamic torque on the rotor, and the other is to represent the generator electromagnetic torque.
 - Torque and Speed Sensor blocks on the low-speed shaft to provide the inputs to FAST
 - Speed Sensor block on the generator inertia to provide the input to the generator electric model
2. Make the required changes to the FAST input .fst file:
 - DrTrDOF = False
 - GBRatio = 1
 - GenIner = 0

- LSSTipAxa, or the equivalent names, in the OutList
3. Build a generator model (e.g., Type 3 WTG). A much simpler model, such as that shown Figure A.27, can be sufficient to represent the variable-speed operation or simply a constant generator torque to represent the above-rated operation of a turbine.
 4. Integrate FAST, drivetrain, and generator models, as shown in Figure A.23.
 5. Choose a solver, as shown in Figure A.8. Using the variable time step solver of oder23tb (stiff/TR-BDF2) is recommended.
 6. Run the simulation, as illustrated in the *FAST User's Guide*, by running the Simsetup.m file and giving the corresponding FAST input .fst file.

10.2 Appendix B: Integrating FAST with a Type 3 Wind Turbine Generator Model

10.2.1 Introduction

Type 3 wind turbine generators (WTGs) are the widely-used doubly-fed induction generators (DFIGs). A good model of a WTG has been developed and made available in the SimPowerSystems Application Library. This section is based on the MATLAB version 7.14 (R2012a), Simulink version 7.9, and SimPowerSystems version 5.6.

A Type 3 WTG model can be found in the MATLAB/Simulink library under *Simscape > SimPowerSystems > Application Libraries > Renewable Energy Library > Wind Generation*. The default icon of the model is shown in Figure B.1 (a).

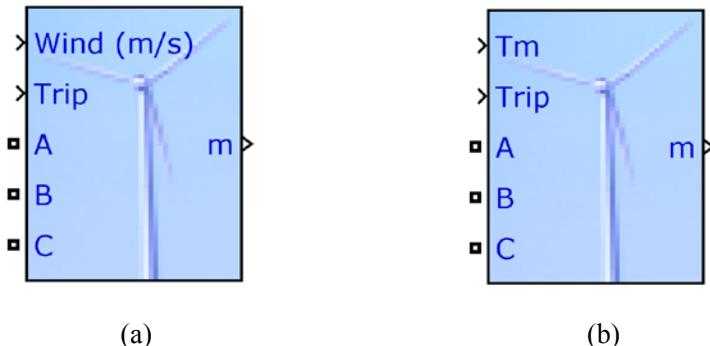


Figure B.1. A DFIG (a) with a turbine model and (b) without a turbine model

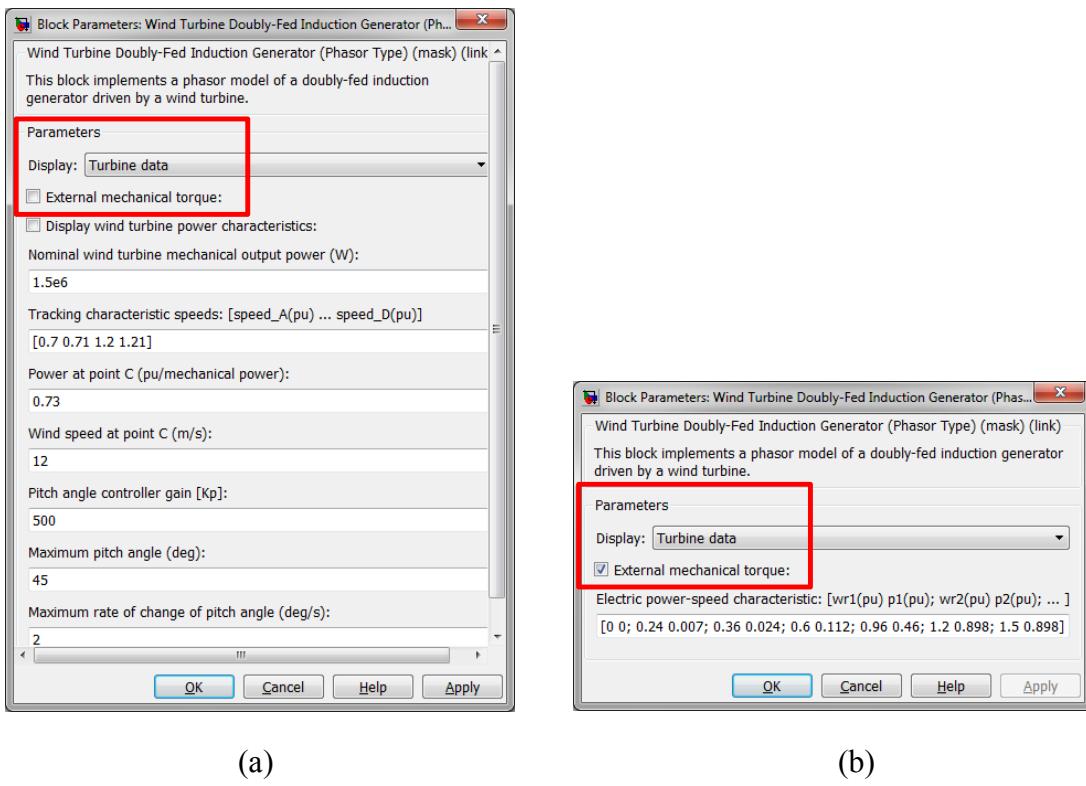


Figure B.2. (a) Option to include the turbine model and (b) option to bypass the turbine model

10.2.2 Bypassing the SimPowerSystems Turbine Model

The SimPowerSystems DFIG model has two options. The first option models the DFIG coupled with a simple wind turbine model; thus, it takes a wind speed profile (in m/s) as the input. The second option bypasses the wind turbine model; thus, it models only the DFIG, which takes the shaft torque as the input.

Choosing either of the two options can be done by changing the setting of the “Turbine Data.” This “Turbine Data” is under the Parameters “Display” list. Figure B.2 (a) is shown by double-clicking the icon shown in Figure B.1 (a), and Figure B.2 (b) is shown by double-clicking the icon shown in Figure B.1 (b). Ticking the “External mechanical torque,” as shown in Figure B.2 (b), bypasses the wind turbine model. Because this report uses FAST as the wind turbine model, this generator model must take the configuration of Figure B.1 (b) and Figure B.2 (b).

Because the wind turbine model of the SimPowerSystems is bypassed, the power-speed characteristic of is needed. This characteristic must be provided in terms of the maximum power points in the power-speed plot. The default turbine characteristic is shown in Figure B.3. This plot can be shown by clicking “Display wind turbine power characteristics,” as shown in Figure B.2 (a).

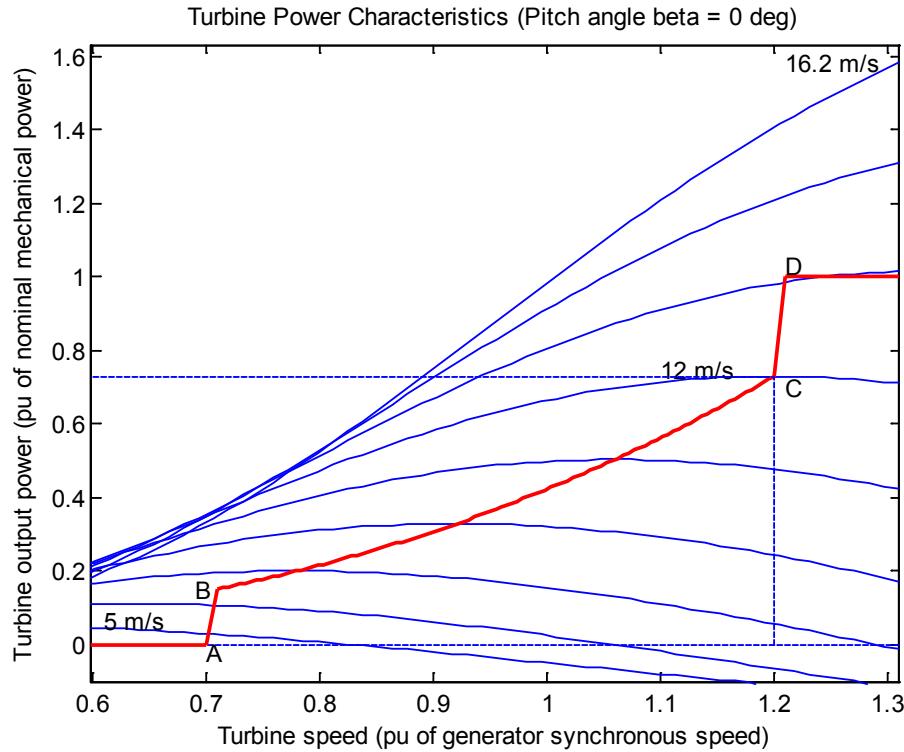


Figure B.3. Default power-speed characteristic of a turbine

The blue plots show turbine output power versus turbine speed at different wind speeds. The red plot connects the maximum points of the blue plots. If a turbine's speed is too low (less than 0.7 p.u.), no power is produced because the available power is insufficient to overcome the losses. If a turbine's speed is too high (more than 1.2 p.u.), the output power is limited at its rated value (1 p.u.).

The red plot is the necessary information for the “Electric power-speed characteristic” field shown in Figure B.2 (b). The default content is [0 0; 0.24 0.007; 0.36 0.024; 0.6 0.112; 0.96 0.46; 1.2 0.898; 1.5 0.898], which is different from that shown in Figure B.3. This default characteristic is shown as the blue plot in Figure B.4. To retain the power-speed characteristic of Figure B.3, these data points are recommended instead: [0 0; 0.7 0; 0.71 0.15; 0.78 0.2; 0.98 0.4; 1.12 0.6; 1.19 0.74; 1.20 1; 1.50 1], which are shown as the red plot in Figure B.4.

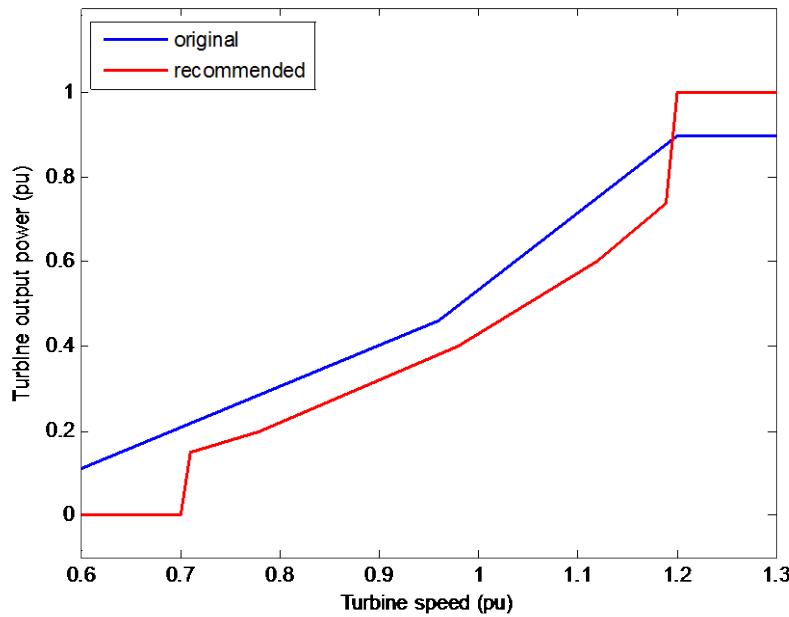


Figure B.4. The original and recommended turbine power-speed characteristic

10.2.3 The DFIG Model

It is important to note that this WTG model is a phasor model, which treats the power system as a balanced three-phase fixed-frequency network. In this system, each phase voltage is identical in magnitude, but out of phase by 120° . Phasor simulation replaces the differential equations representing the electrical network with a set of algebraic equations at a fixed frequency. Phasor simulation facilitates the transient stability analysis of systems with multiple machines. Therefore, this simulation, also known as the positive sequence simulation, cannot be used to study unbalance scenarios.

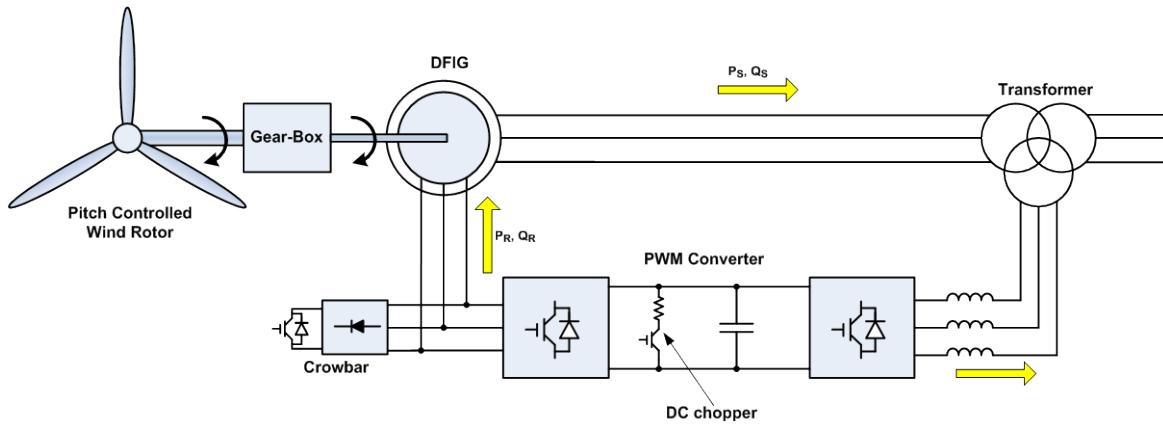


Figure B.5. Type 3 wind turbine connection diagram

This WTG model is better used to simulate the low-frequency electromechanical oscillations over the scale of seconds to minutes. Another available technique to model the DFIG is to use a three-phase representation so that the unbalanced conditions can be simulated. These unbalanced

conditions may come from the grid unbalanced voltage (faults, dips, or other transient events) or unbalanced grid impedance. The model developed so far does not account for these factors.

Figure B.5 shows the connection diagram of the DFIG. The DFIG operates in variable-speed mode using a partial-size power converter connected to the rotor winding of the wound-rotor induction generator (WRIG). The stator winding of the WRIG is connected to the grid at a frequency of 60 Hz. This type of generator is probably the most popular one available in the market and has been deployed in many wind turbines. This WTG is normally operated between 30% slip (subsynchronous speed) and -30% slip (supersynchronous speed), and the converter is typically at about 30% of rated output power. The power converter performs a back-to-back AC-DC-AC conversion using two pulse-width modulation-switched voltage-source inverters coupled with a DC link. A crowbar circuit is also provided as protection, to allow shorting the rotor circuit if necessary. In variable- speed mode, the torque characteristic of the DFIG is a quadratic function of its rotational speed. This WTG allows maximal extraction of wind power because its output power is electronically controlled to follow the optimal power curve, which is a cube function of the generator rotational speed.

If a turbine's rotor speed exceeds its rated value, the turbine pitch controller must be deployed to limit the rotational speed at its rated speed. If the pitch controller could not control the aerodynamic power of a wind turbine, a WTG may experience a runaway event. Note that the speed range of a Type 3 WTG is much larger than the speed range of a Type 1 WTG. Thus, a sufficiently large amount of kinetic energy can be stored and restored in the rotating blades and other mechanical components of a wind turbine. This characteristic makes the output of the generator not as much affected by the wind fluctuations and turbulence.

By bypassing the wind turbine model, the SimPowerSystems DFIG model now takes a torque input. This torque input normally gets transmitted from the high-speed shaft of the gearbox. However, similar to the previous model integrations, the generator model must take the generator (high-speed shaft) rotational speed from FAST and deliver the generator electromagnetic torque and output power to FAST. The following paragraphs describe the modifications of the SimPowerSystems DFIG model to take the generator speed measurement and give the electromagnetic torque.

The internal components of the SimPowerSystems DFIG model can be revealed by right-clicking the DFIG icons shown in Figure B.1Figure B.1 and selecting the “Look Under Mask,” as shown in Figure B.4.



Figure B.6. Select the “Look Under Mask” to reveal the model components

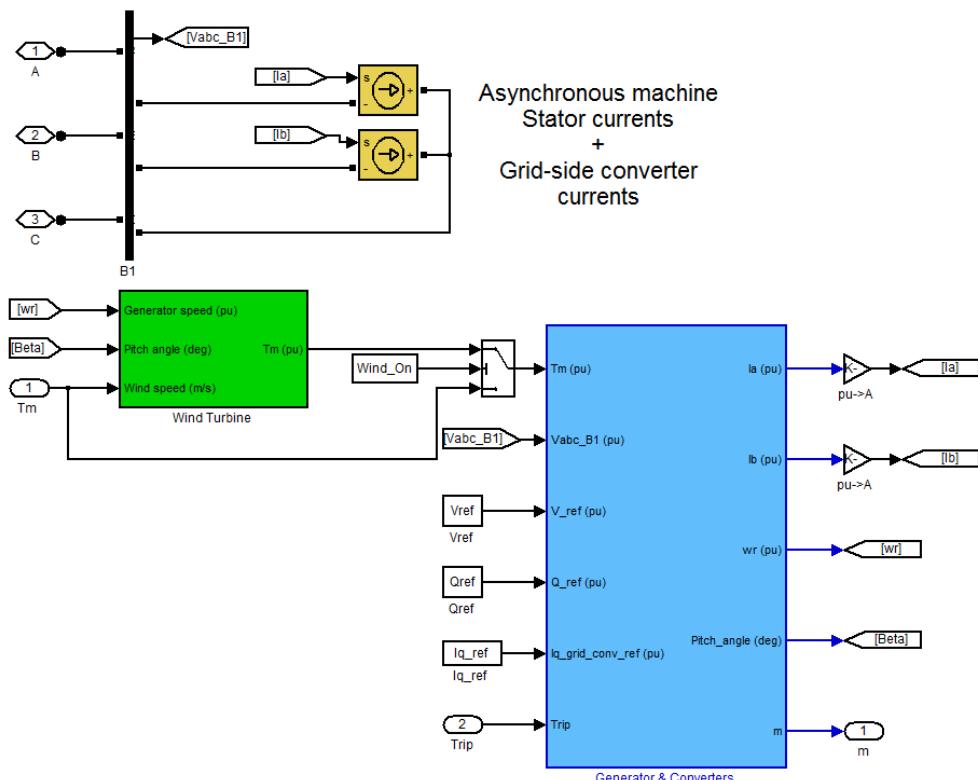


Figure B.7. Internal components of the SimPowerSystems DFIG model

Figure B.7 shows the internal components of the DFIG model. The Currents yellow blocks represent the simplified models of the power electronics. The green block represents the wind turbine model. The value of the Wind_On block is 0 when the user chooses to bypass the wind turbine model. As the logic switch senses a low value of its second input (i.e., the “Wind_On”), the first input of the switch (i.e., the output of the wind turbine model) is ignored. Instead, the third input (i.e., our generator input of interest, “ T_m (pu)”) is directly transmitted into the blue Generator & Converters block.

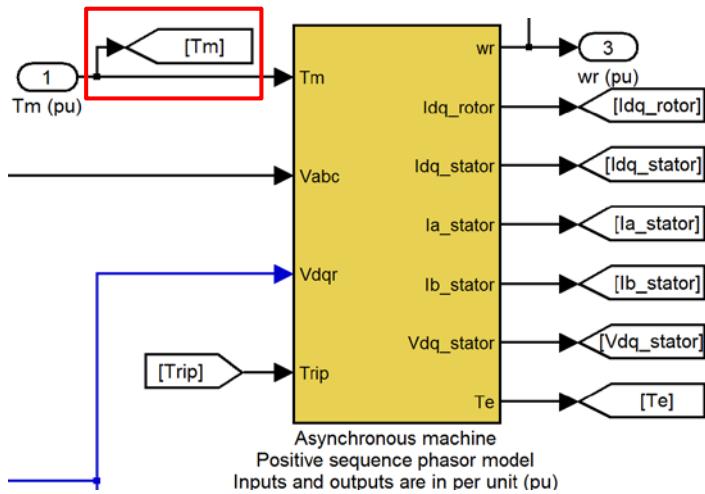


Figure B.8. Internal component of the Generator & Converter block

Double-clicking the blue Generator & Converters block shows its internal components. Again, we would like to track the “ T_m (pu),” which now goes into the yellow Asynchronous Machine block, shown in Figure B.8.

To reveal the internal components, the user can right-click on the yellow block and choose the “Look Under Mask.” The “ T_m (pu)” now comes into the Mechanical block, as highlighted in Figure B.9.

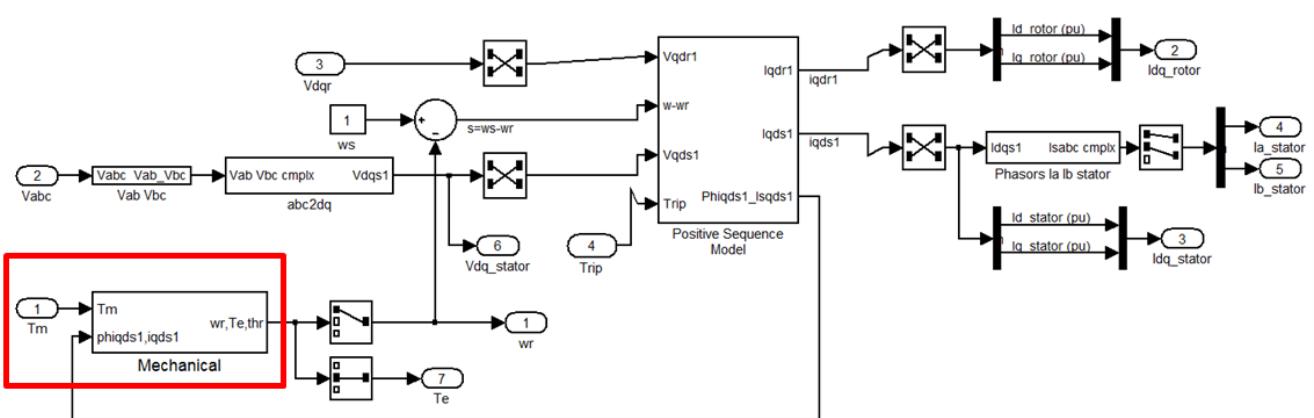


Figure B.9. Internal component of the Asynchronous Machine block

To reveal the internal components, the user can double-click the Mechanical block. The “ T_m ” is being used in the mechanical equation of motion of the generator inertia. The equation of motion is highlighted in Figure B.10 and enlarged in Figure B.11.

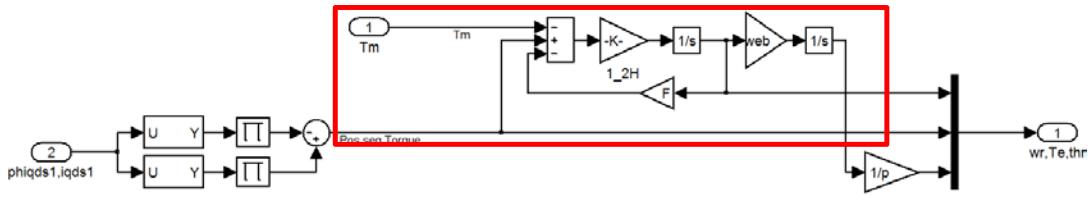


Figure B.10. Internal components of the DFIG Mechanical block

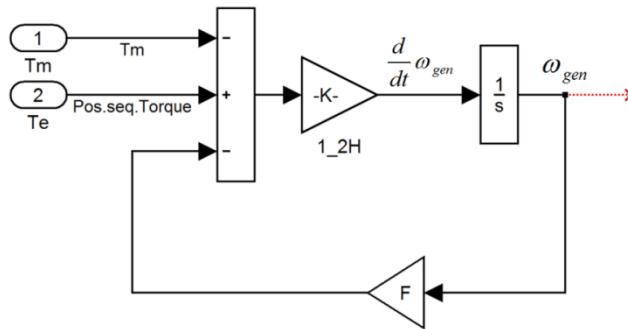


Figure B.11. The mechanical equation of motion of the generator inertia

As shown in Figure B.11, the equation of motion is expressed as:

$$\frac{d}{dt} \omega_{gen} = \frac{1}{2H} (T_e - F\omega_{gen} - T_m)$$

Therefore, if the generator model is to directly take the speed input, the blocks before the ω_{gen} must be deleted. While trying to delete the blocks, the window shown in Figure B.12 will appear:

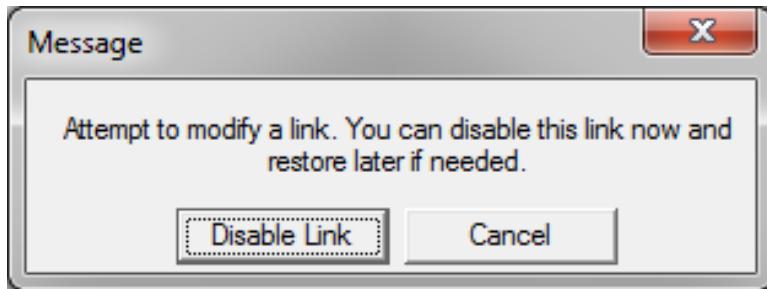


Figure B.12. Dialog window to disable the DFIG link

This warning appears because the DFIG model has been provided as a linked file, which can be called through the MATLAB command window. The user should click “Disable Link.” After deleting the unnecessary blocks, the internal configuration of the Mechanical block should appear, as shown in Figure B.13. The reader can also alter the signal and input port labels from “ T_m ” to “ w ” (to represent ω) and subsequently replace all the “ T_m ” in the input ports from Figure B.7 through Figure B.10 to “ w .” We recommend not altering the signal router, which is highlighted in Figure B.8, to avoid other unnecessary modifications. It is safe to leave it as a

dummy signal routing because the [Tm] signal routing is used only for the data acquisition of the model, not for any calculations.

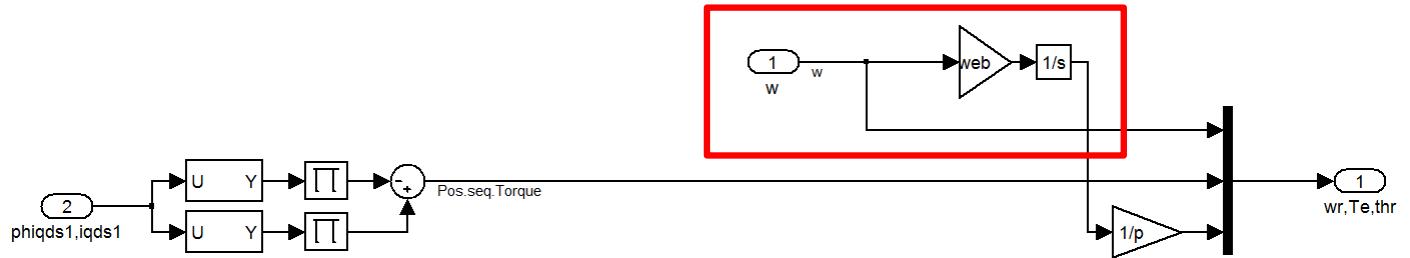


Figure B.13. The modified Mechanical block to take the speed input

Normally, modifying the input label of Figure B.1 (b) can be done by right-clicking the icon and choosing “Edit Mask.” The window shown in Figure B.14 will appear. The user can try to change the highlighted port label from “Tm” to “w,” then click “Apply” and “OK.” However, for some reason, the change does not get applied and the label “Tm” prevails. Therefore, although the label still reads “Tm,” the model in fact now takes the speed input.

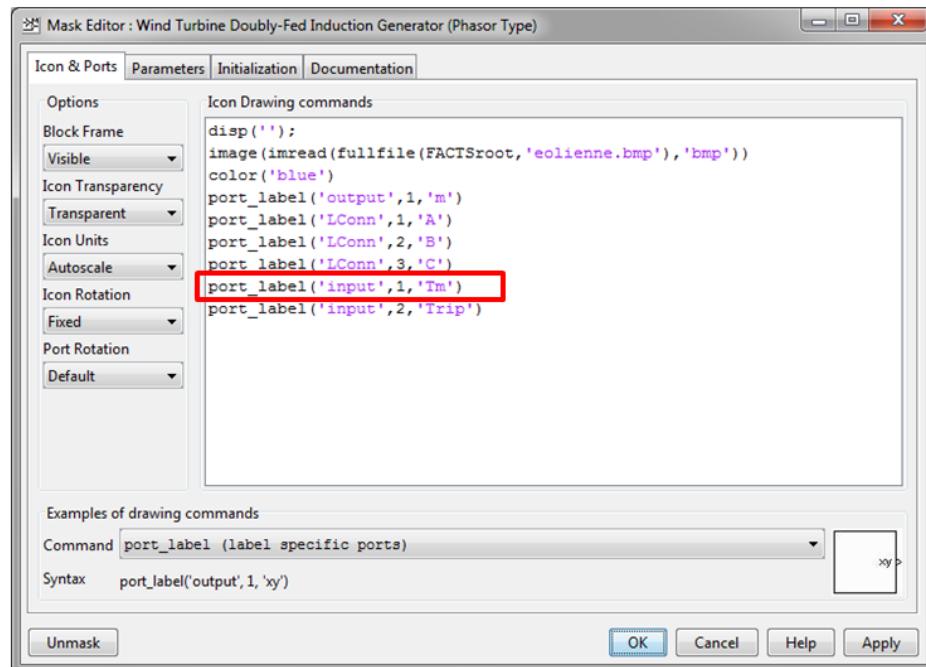


Figure B.14. Attempting to modify the input label from “Tm” to “w”

10.2.4 Complementing the DFIG Model

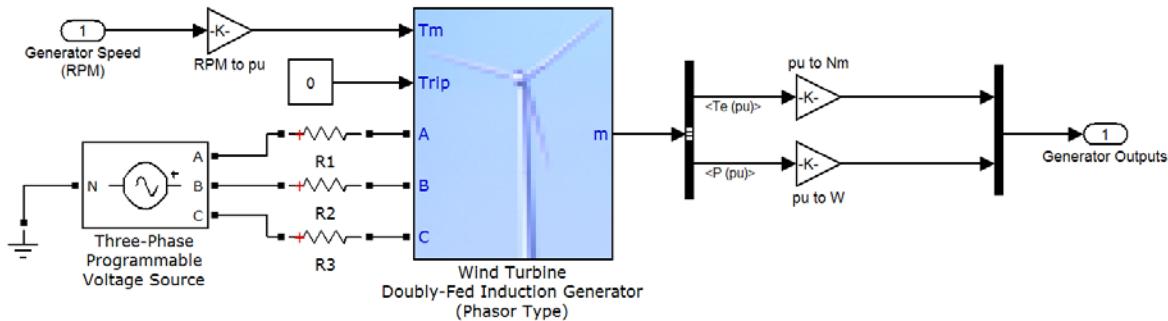


Figure B.15. Generator model readily integrated with FAST

Figure B.15 shows the overall generator model that is ready to be integrated with the FAST aeroelastic code. It shows several additional blocks that are necessary to complement the model. The input and output of this model are described in the following paragraphs, along with the descriptions of the complementary blocks.

10.2.4.1 The Speed Input Into the Generator Model

The generator speed input comes from FAST, with RPM as its default unit. Because this generator model works in the per unit (p.u.) system, the actual speed must be divided by the generator synchronous speed. The parameters of the generator can be accessed through the “Generator Data” under the Parameters “Display” list, shown in Figure B.16. In this study, the generator nominal power has been set to 1.5 MW.

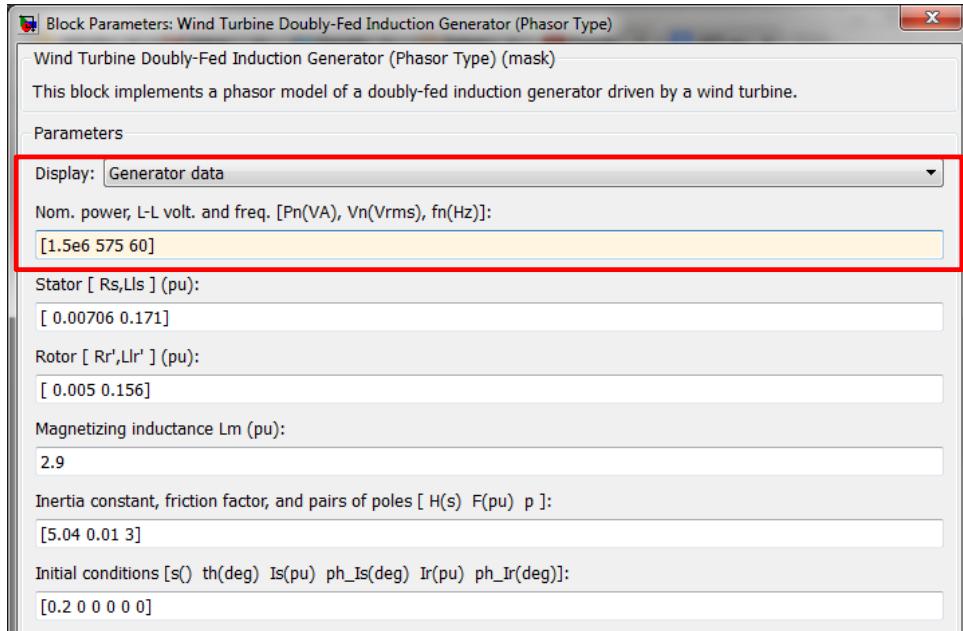


Figure B.16. Modifying the generator parameters

Similarly, the parameters of the converter can be accessed through the “Converter Data” under the Parameters “Display” list. The original parameters are maintained, as shown in Figure B.17.

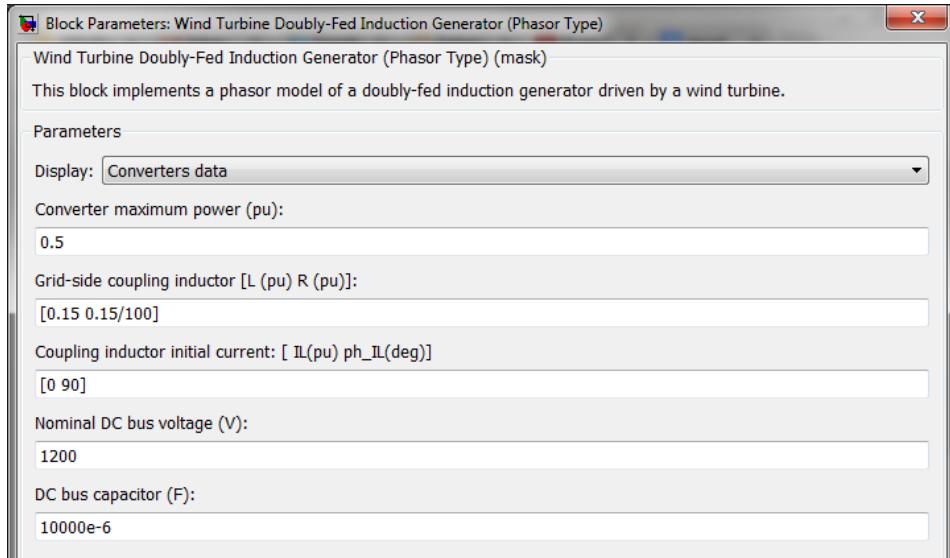


Figure B.17. The original converter parameters

Last, the original controller parameters are also maintained. Figure B.18 shows the parameters as a reference.

The synchronous speed of the generator is calculated as:

$$\omega_{sync} = \frac{120f}{p} = \frac{120(60)}{(3)(2)} = 1200 \text{ RPM}$$

Therefore, the generator speed must be multiplied by 1/1200 to result in p.u. The multiplication can be done using the Gain block. This block can be found at the Simulink library under *Simulink > Commonly Used Blocks*:

Name	Icon	Description
Gain		It multiplies the input by a constant value.

10.2.4.2 The “Trip” Input Into the Generator Model

The trip input defines whether the generator is connected to the grid. High value (e.g., 1) means that the trip occurs and the generator is disconnected (i.e., no output power), otherwise (e.g., 0 input) the generator produces power. To provide this value, a Constant block can be used. This block can be found at the Simulink library under *Simulink > Commonly Used Blocks*:

Name	Icon	Description
Constant		It generates constant value.

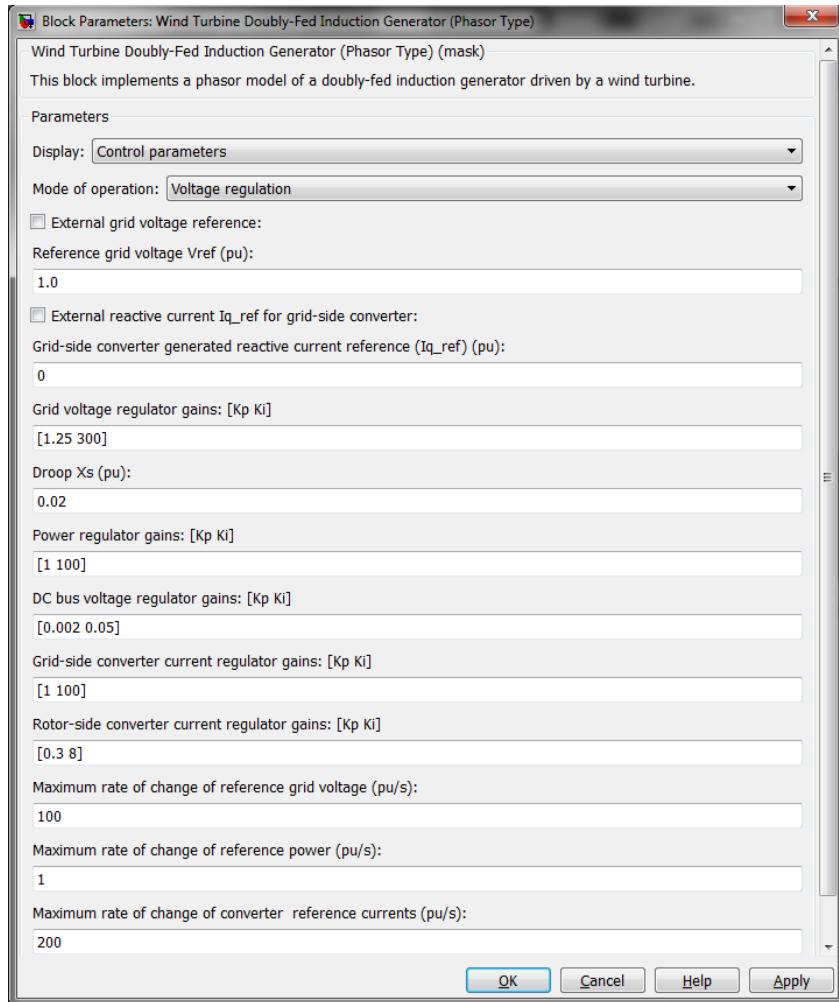


Figure B.18. The original control parameters

10.2.4.3 The Voltage Input Into the Generator Model

Three-phase voltage source (A, B, and C) must be connected to the generator model. The following blocks are used to build the voltage input. These components can be found at the Simulink library under *Simscape > SimPowerSystems > Electrical Sources*:

Name	Icon	Description
Three-Phase Programmable Voltage Source		It implements three-phase voltage source with programmable time variation of amplitude, phase, frequency, and harmonics.

The user must set the voltage source to deliver the required line-to-line voltage of the generator, as shown in Figure B.19. As highlighted in Figure B.16, the required voltage is 575 V.

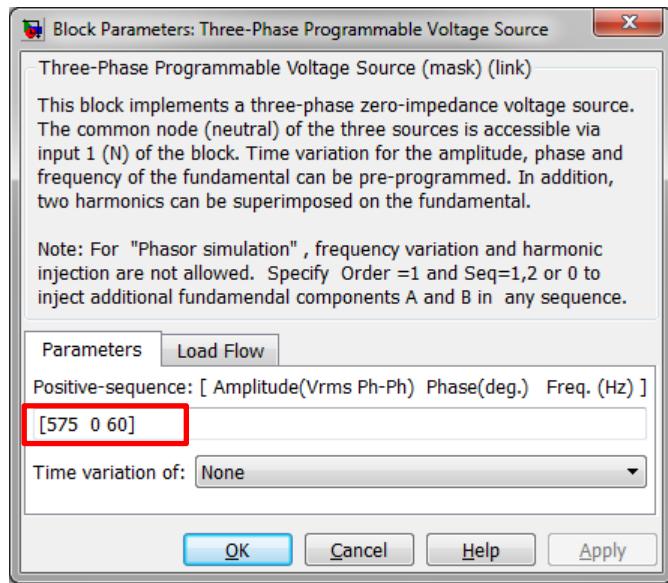


Figure B.19. Supplying the required line voltage

These components can be found at the Simulink library under *Simscape > SimPowerSystems > Elements*:

Name	Icon	Description
Series RLC Branch		It implements a single resistor, inductor, or capacitor, or a series combination of these.
Ground		It provides connection to the ground (zero voltage).

A small resistance normally presents in a realistic voltage source. The user can set the mode of the “Series RLC Branch” to represent only the resistance of the voltage source, as shown in Figure B.20.

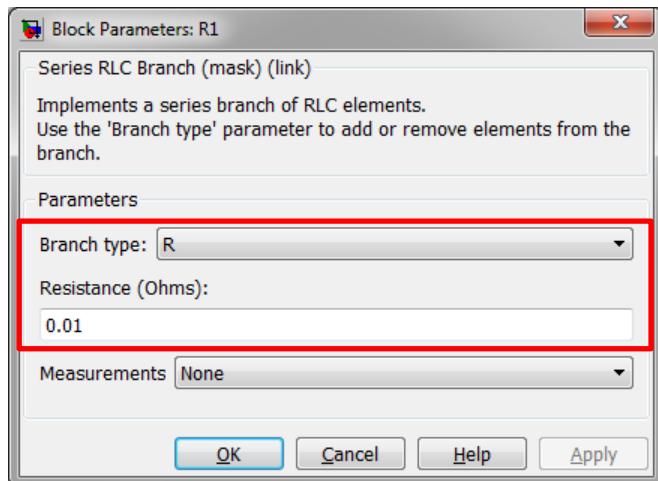


Figure B.20. Representing a pure resistor

10.2.4.4 Extracting the Generator Outputs

The DFIG generator model delivers a series of outputs in p.u. The ones needed by FAST are the generator electromagnetic torque and the output power. To isolate any desired output, a “Bus Selector” can be used. This component can be found at the Simulink library under *Simulink > Commonly Used Blocks*:

Name	Icon	Description
Bus Selector		It outputs a specified subset of the elements of the bus at its input.

To select the torque and output power, the “Bus Selector” must first be connected to the DFIG model through the “m” port. Afterward, double-clicking the “Bus Selector” shows the list of the available outputs on the left column. Selecting one output at a time can be done by first choosing “Te (pu)” and then clicking the “Select >>” button. Similarly, another subsequent variable (i.e., the “P (pu)”) can be chosen to obtain the results shown in Figure B.15. It is important to note that the position of the torque signal *must* be above the power signal.

Because the outputs are in p.u., they must be multiplied by the corresponding rated values to result in the SI units. The rated torque is calculated as:

$$T_{rated} = -\frac{P_{rated}}{\omega_{sync}} = -\frac{1.5 \times 10^6 \text{ W}}{(1200)\left(\frac{2\pi}{60}\right) \text{ rad/s}} \approx -11936.62 \text{ Nm}$$

The negative value is because it is now a generator model; whereas the positive value is for a motor model. This rated torque can be implemented in a Gain block, as shown in Figure B.22. The rated generator power is 1.5 MW, which is also implemented in a Gain block.

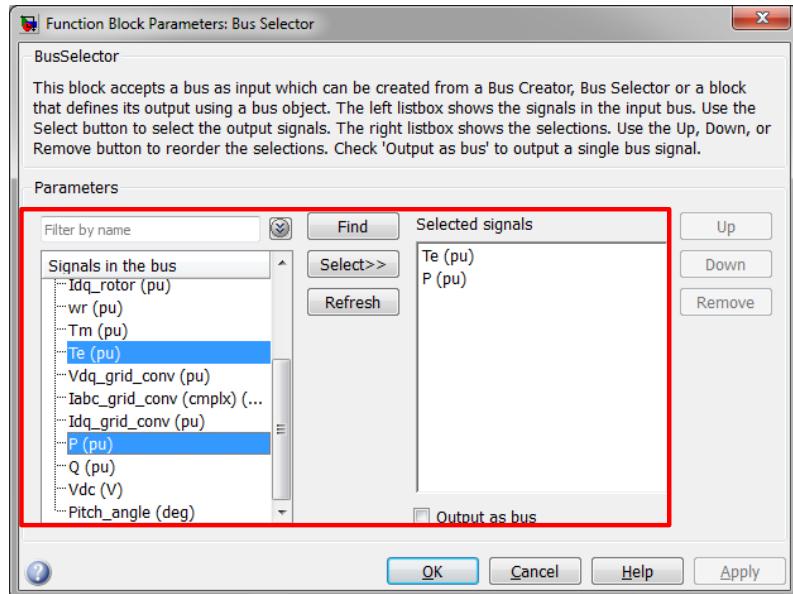


Figure B.21. Selecting the generator torque and output power

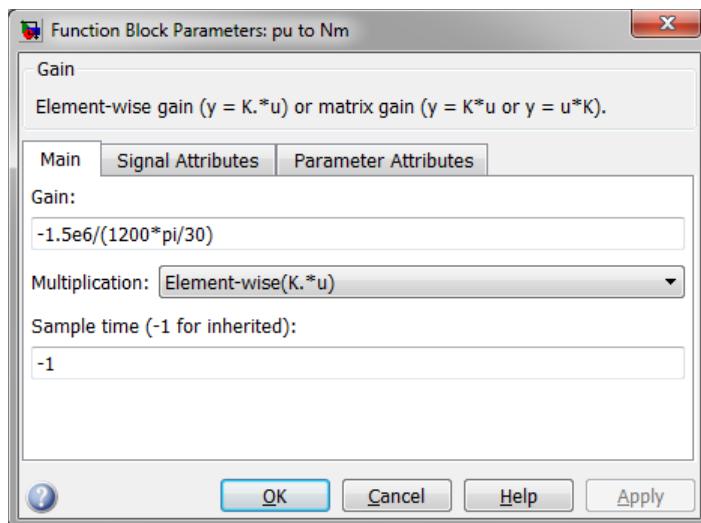


Figure B.22. Multiplying the rated torque to convert the unit from p.u. to Nm

Finally, the torque and power are combined into a vector by using a Mux. This component can be found at the Simulink library under *Simulink > Commonly Used Blocks*:

Name	Icon	Description
Mux		It combines its inputs into a single vector output.

10.2.5 Integrating the DFIG Model

Figure B.23 shows the integrated wind turbine and generator model in the Simulink environment. The SimPowerSystems model shown in Figure B.15 is in the blue block. The FAST S-Function block is in the orange block. This block is readily available within the downloaded FAST package as the *OpenLoop.mdl* shown in Figure B.14. The Pitch and Yaw Controllers block are not affected by this integration. The Torque Controller, which gives the torque and power inputs to FAST, is now replaced by the generator model.

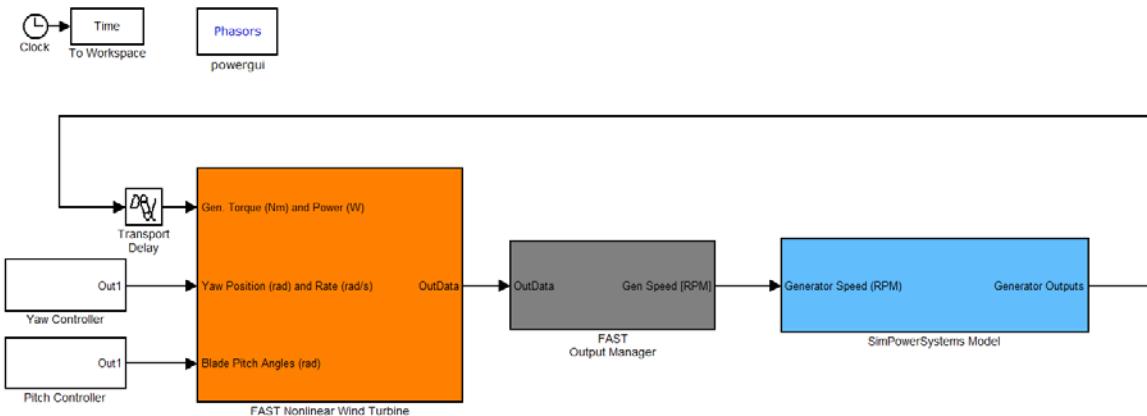


Figure B.23. Implementation of an integrated wind turbine generator model

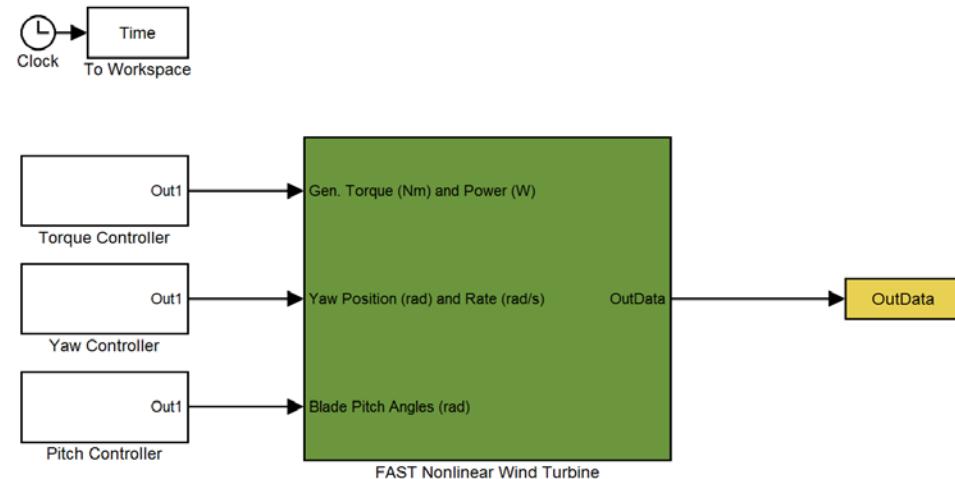


Figure B.24. Original FAST S-Function block as *OpenLoop.mdl*

The grey FAST Output Manager Block contains parallel Fcn blocks, as shown in Figure B.15. This Fcn block can be found at the Simulink library under *Simulink > User-Defined Function*:

Name	Icon	Description
Fcn		It applies the specified mathematical expression to its input.

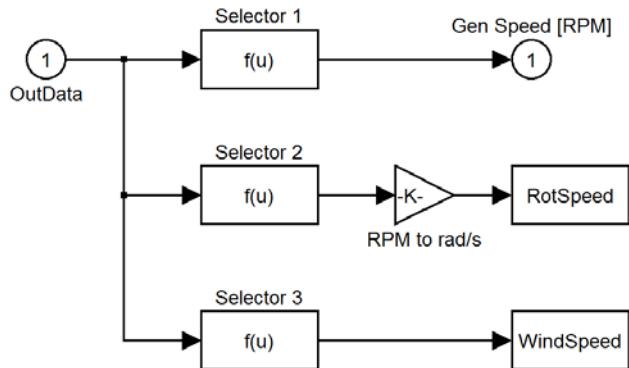


Figure B.25. FAST Output Manager block

Each Fcn block can be used to isolate a variable of interest from the array of FAST outputs. These outputs are defined at the end of the FAST input .fst file, under the section `OutList`. Similar to other generator models in this report, the generator speed is the required output from FAST. Therefore, the generator speed must be listed in the FAST input .fst file. According to the *FAST User's Guide*, pp. 102, the “`GenSpeed`” corresponds to the measurement of the generator speed. Another equivalent name of the parameter is the “`HSShftV`.” This measurement can be isolated by setting the Fcn block, as shown in Figure B.26.

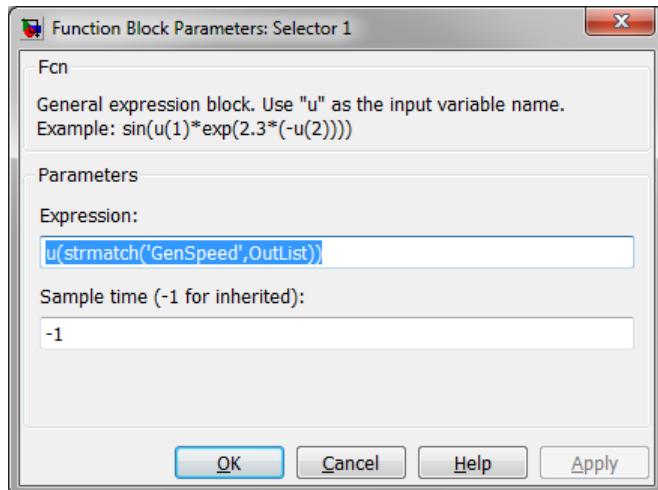


Figure B.26. Fcn block parameters to isolate the generator speed

Figure B.23 also shows other Fcn blocks, each of which is used to isolate one variable of interest (e.g., the rotor speed). To do so, the “`GenSpeed`” can be replaced by another proper parameter name (e.g., “`RotSpeed`”).

Instead of the Switch, Figure B.23 shows a Transport Delay block near the FAST input. This delay serves as an alternative to break the algebraic loop. The user can define a very small time delay (e.g., 5 ms) to prevent any significant changes in the overall dynamics of the system. This block can be found at the Simulink library under *Simulink > Continuous*:

Name	Icon	Description
Transport Delay		It delays the input by a specified amount of time.

Moreover, a Powergui block, shown in the top of Figure B.23, is a must in every simulation using SimPowerSystems models. This block can be found at the Simulink library under *Simscape > SimPowerSystems*:

Name	Icon	Description
Powergui		It serves as the environment block for SimPowerSystems models.

It is important to note that the “Powergui” block is to be placed at the top level of the Simulink model diagram for optimal performance and a maximum of one Powergui block is allowed per model (i.e., for each .mdl file). The default setting of this block is Continuous. As described in the early discussion of this DFIG model, the phasor setting is recommended. As illustrated in Figure B.27, changing its setting can be done by double-clicking the block, clicking “Configure parameters,” and choosing “Phasor” from the “Simulation Type.”

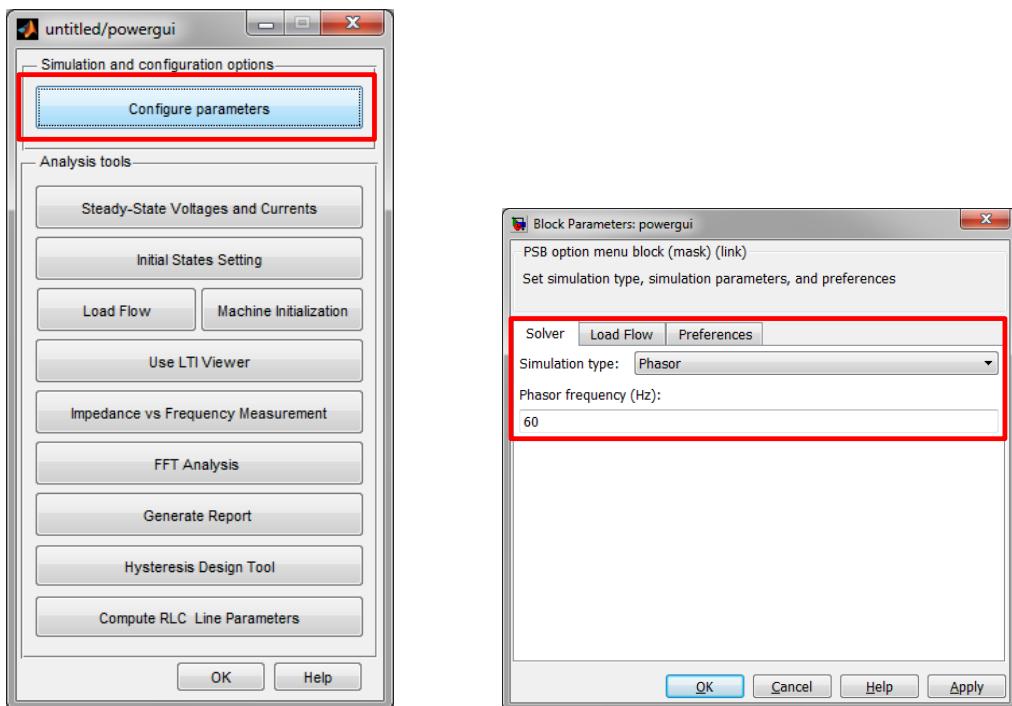


Figure B.27. Changing the Powergui from “Continuous” to “Phasor”

For physical models, such as those of the SimPowerSystems, MathWorks recommends implicit global solvers, such as the ode14x, ode23t, or ode15s. The solver setting can be modified

through the “Configuration Parameters” window (from the Simulink tool bar: Simulation > Configuration Parameters or by pressing Ctrl+E). Under the Solver Options, the user can set the Type to Variable-step and the Solver to ode23tb (stiff/TR-BDF2). Figure B.28 highlights the necessary modifications to the “Configuration Parameters.”

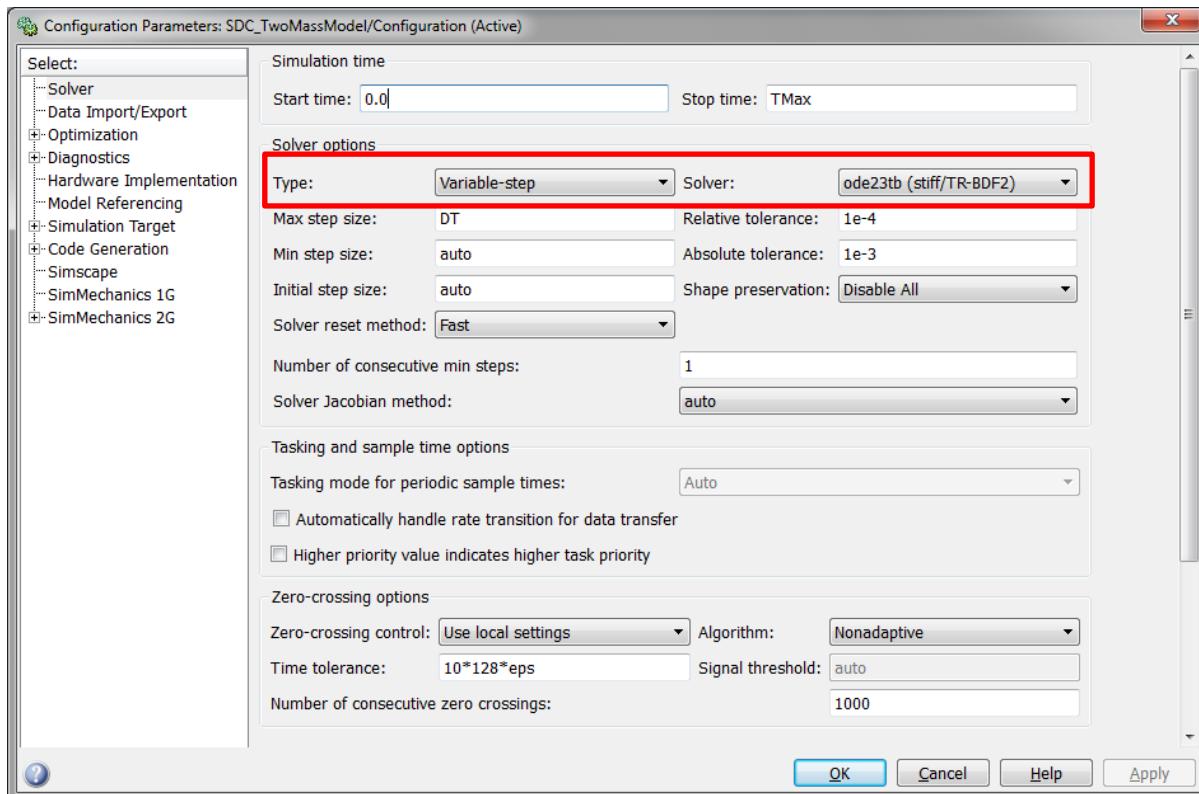


Figure B.28. Modifying the “Configuration Parameters”

10.2.6 Running the Simulations

Test12.fst is provided in the FAST downloaded package as a part of the certification test files to represent the WindPACT 1.5 MW Baseline Turbine. This test file is chosen as an illustration in this report because it represents a turbine with the same rating as the default SimPowerSystems DFIG model.

Several changes are required in the Test12.fst; whereas some are recommended to fit the simulation purposes. These changes are as summarized below:

1	ADAMSPrep	- ADAMS preprocessor mode [REQUIRED]
100	TMax	- Total run time (s) [RECOMMENDED]
0.0005	DT	- Integration time step (s) [REC.]
2	PCMode	- Pitch control mode [REQ.]

```

0          TPCOn      - Time to enable active pitch control
            (s) [REQ.]
3          VSControl - Variable-speed control mode [REQ.]
11.0       RotSpeed   - Initial or fixed rotor speed [REQ.]
----- AERODYN -----
"Test12_ASPS.ipt"    ADFile [REC.]
OutList   - The next line(s) contains a list of outputs
"GenSpeed" - Generator speed [REQ.], and other desired outputs

```

After implementing the changes, it is recommended to save the file as a new file, e.g., `Test12_SPS.fst`. Some changes are also recommended to the original Aerodyn (AD) file, the `Test12_AD.ipt`, to represent a much simpler wind profile:

```
"Wind\onestep.wnd"      WindFile
```

After implementing the changes, it is also recommended to save the AD files file as a new file, e.g., `Test12_ASPS.ipt`. The `onestep.wnd` wind file represents a step wind speed input at 50 seconds, which is in the form of:

```

! Wind file for sheared 18 m/s wind with 30 degree direction.
! Time     Wind     Wind     Vert.    Horiz.    Vert.    LinV    Gust
! Speed    Dir      Speed    Shear    Shear     Shear
0.0      7.0      0.0      0.0      0.0      0.0      0.0      0.0
0.1      7.0      0.0      0.0      0.0      0.0      0.0      0.0
49.9     7.0      0.0      0.0      0.0      0.0      0.0      0.0
50.0     8.0      0.0      0.0      0.0      0.0      0.0      0.0
999.9    8.0      0.0      0.0      0.0      0.0      0.0      0.0

```

The simulation can be set up by typing `Simsetup` on the MATLAB command window, and then by typing in the FAST input file name, which in this report is the `Test12_SPS.fst`. The simulation can then be run. Some warning messages regarding algebraic loops will appear on the MATLAB command window. Those warnings can be ignored because they come from the wind turbine model of the SimPowerSystems DFIG model (i.e., the green block shown in Figure B.7).

The generator responses—in terms of speed, torque, and power—are shown in Figure B.29. Compared to the previous types of generator, these responses show much less fluctuation because of the ability of the Type 3 to store and restore the kinetic energy from the wind. When the wind speed changes, the large inertia of this megawatt-rating turbine contributes to the slow response of the generator (approximately 20 seconds) to reach the optimal rotational speed for maximum power capture.

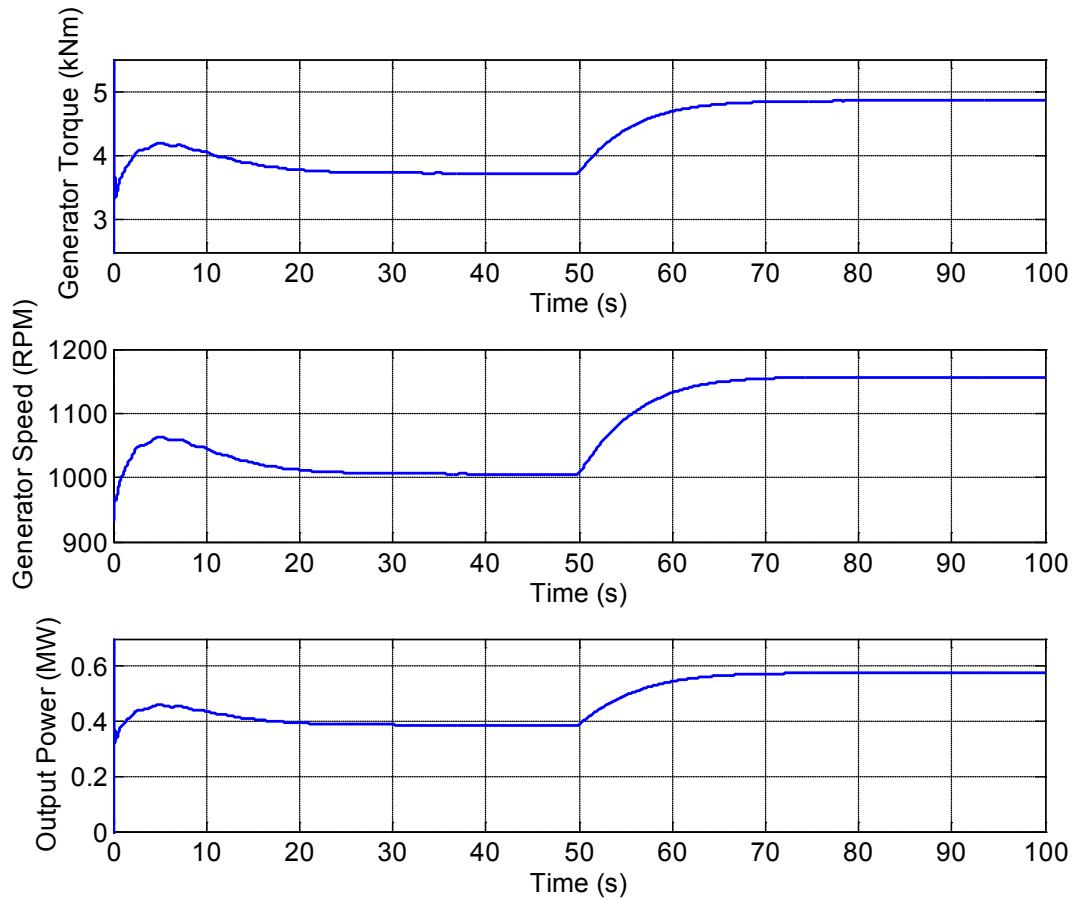


Figure B.29. Simulated generator response