CART3\_FAC.slx

# Introduction

The *CART3\_FAC.slx* Simulink model contains a wind turbine equipped with a fault adaptive controller (FAC). This model was used in the paper *Fault Adaptive Wind Turbine Controller*.The model checks various sensor readings and commands the wind turbine based on these readings. The paper details the controller logic and operation. This readme will instruct the user how to replicate the simulations in the paper and run new simulations.

# Model Basics

The Simulink model uses FAST to model the CART3 wind turbine operation, using a FAST s-function. This existing Simulink implementation of FAST will not be detailed here. A fault control block was integrated into the model to add FAC functionality.

# Acronyms

Warning types:

S: stop

RP: reduce power

NA: no action

Shutdowns:

NS: normal stop

OLNS: open loop normal stop

ES: emergency stop

Modes:

NO: normal operation

WNA: warning - no action

WRP: warning - will reduce power

WS: warning - will stop

NS: normal stop

OLNS: open loop normal stop

ES: emergency stop

Timers:

TRP: (decreasing) time until power change

TS: (increasing) time under S warning

TNA: (increasing) time under NA warning

# Running CART3\_FAC.slx

The model should run as-is, following these instructions:

1. *CART3\_FAC\_paper* and subfolders must be on the MATLAB path
2. *CART3\_FAC\_paper* must be the current folder in MATLAB
3. Parameters are initialized in the .m file called in the Simulink model in the tab: *Modeling/model settings/model properties/Callbacks/InitFcn*. Update to point to *CARTinitFcn.m*

# Replicating Paper Cases

Data from the three cases are stored in .mat files in the main folder, and figures are stored in the *Scope* subfolder. To replicate cases:

1. Find the faults plotted in the case.
2. Insert the appropriate data from *fault\_timeseries\_bank.xlsx* into *fault\_timeseries.xlsx,* from which sensor data is input into the FAC. Zero the other data.
3. Uncomment the corresponding scopes to see the signals in scope.
4. Use *make\_plots.m* to replicate plots. This file plots data in the timeseries *plots*, which is output to the MATLAB workspace with every run.

# Running New Cases

## Changing Wind Turbine Parameters

The FAST input files are contained in the *CART dependencies* subfolder and can be changed to model different wind turbines. FAST v8. was used to build the model. See FAST documentation at <https://www.nrel.gov/wind/nwtc/fastv8.html> for instructions on building/modifying FAST input files, including the inflow wind field.

## Checking for New Faults

To check for a new fault, the controller requires:

1. The initialization of the fault in the ctrl, data, and faulty structures, in *CARTinitFcn.m*
   1. Data: initialize to zero
   2. Ctrl: initialize state to zero
   3. Faulty: specify the check function name, warning type, and shutdown mode.
2. Adding the corresponding new sensor input sheet into *fault\_timeseries.xlsx*
3. Connecting the new sensor input into the data bus in the *Fault Controller* subsystem
4. Adding a fault-check function, whose name you specify in the *faulty* structure in *CARTinitFcn.m*. Use the existing functions as templates. Check functions, defined in helper functions folder, contain hardcoded limits for fault and warning modes. These can be modified as the user desires. Existing functions are:
   1. freqSensorWarnFun.m
   2. torqueSensorWarnFun.m
   3. genTempWarnFun.m
   4. axWarnFun.m
   5. powerWarnFun.m

The Nacelle x-acceleration sensor has been implemented but commented out. Uncomment the appropriate ax lines (29, 47, 60-62) in *CARTinitFcn.m,* and uncomment and connect the ‘from spreadsheet’ block in the *Fault Controller* subsystem to implement the ax check. Change values in the *ax* sheet of *fault\_timeseries.xlsx* to cause a fault/warning. New faults can be added in the same fashion.

