#### I. Before Linearization:

For the operating point in which you are interested, get steady state values for all initial conditions you may be interested in (the initial conditions of interest will be in ElastoDyn.dat).

The initial conditions that need to be updated are:

- Blade Pitch BIPitch
- Rotor Speed RotSpeed
- Platform Roll, Pitch, Yaw, Surge, Sway, Heave

Update the initial conditions in ElastoDyn to be as accurate as possible; doing so will make your linearization run faster as well as help to avoid solver problems in FAST during its initial transients. With the most accurate initial conditions, determine how long until your signals reach a steady state. The linearization process runs OpenFAST so you want to make sure that all initial transients have ended and the system is operating in steady-state for your operating point! So if even with your initial conditions updated, and it still takes 500 seconds for your simulation to reach steady state, then your starting linearization time should be 500 seconds.

#### II. To Linearize:

# A. ServoDyn.dat

PC Mode = 0 (uses a fine pitch value of the initial blade pitch value set in ElastoDyn)

simple vs VS Contrl = 1 (uses simple variable-speed torque control)

In ServoDyn, under the SIMPLE VARIABLE-SPEED TORQUE CONTROL section update the variables for the torque control, they may be initialized to 9999.9 for or set to values from now, but an external controller (such as rosco or Simulink) is not used in the simulation, controller so this torque controller will be used for linearization (seen below).

Note: these are the values for the current USFLOWT 10MW turbine

SIMPLE VARIABLE-SPEED TORQ	UE CONTROL
0.0001	VS_RtGnSp
207234.9879000	VS_RtTq
0.0001	VS_Rgn2K
0.0001	VS S1Pc

If using a rosco or simulink controller, you can find the value for the rated torque (VS RtTq) in your controller parameters (the variable names should be similar for whatever type of controller you are using).

Note: When VS RtGnSp, VS Rgn2K, VS SIPc are very small numbers, the torque will be held constant at the value set for VS RtTq. If linearizing in Region 2, then VS RtTq should be set to the steady state torque value at the operating point of interest.

# B. ElastoDyn.dat

To simplify the verification process, we recommend starting with a simple 1 DOF linearization with only the GenDOF set to True. After this has been verified, additional DOF's can be enabled. The DOF's determine which states are included in the linearization output. Update initial conditions to be steady state values at the operating point you are looking at.

C. HydroDyn.dat

WaveMod= 0 (still water)

i.e. no disturbances

D. InflowWind.dat

 $\underline{\text{WindType}} = 1 \text{ (steady wind)}$ 

Change HWindSpeed to the wind speed corresponding to the operating point for your

### linearization.

### E. Model.fst

Note: the line under NLinTimes is cut off, but there are 36 different times there.

LINEA	ARIZATION	
True	Linearize	L
False	CalcSteady	
3	TrimCase	
0.1	TrimTol	
0.01	TrimGain	
0	Twr_Kdmp	
0	Bld_Kdmp	
36	NLinTimes	
500 500.1	.73520735728 500.347	704147
1 LinInputs		
1	LinOutputs	
False	LinOutJac	
False	LinOutMod	

NLinTimes = 12 omega =

To get <u>LinTimes</u> find how long it takes for your rotor to turn a full rotation (use 10 degrees if you want to linearize at 36 rotor positions, 20 degrees for 18 rotor positions, etc.), (based off the rotor speed at your selected operating point); we will call this delta\_r. Suppose your starting time is t0 = 500. To get all the linearization times you will do  $t(i)=t0+i*delta_r$ , where  $i = \{0, 1, ..., NLinTimes-1\}$  NREL recommends linearizing at 36 azimuth positions separated by 10 deg, so you will have 36 values defined for LinTimes.

Make sure TMax is a few seconds longer than the maximum value in your list of LinTimes.

Run the simulation as you would normally (either through the command line or simulink). You just have to run it once, and you should have 36 linearization output files. (Model.1.lin, Model.2.lin, ..., Model.36.lin)

Each of these linearization files contains the <u>inputs</u>, <u>outputs</u>, <u>states</u>, <u>and A, B, C and D matrices</u> corresponding to the specific azimuth position.

## III. After Linearization:

Note: Since the linearization processes returns 36 state-space models, <u>post-processing</u> is required to find an azimuth-averaged linearized system. NREL provides the 'runMBC' tool for this post-processing.

- A. Download matlab-toolbox from OpenFAST github
- B. Use runmbc.m
- C. Modify the paths and number of linearization files as necessary (see figure below)

```
%addpath(genpath('c:\Users\njohnsol\Documents\matlab-toolbox-Manu\matlab-toolbox\')); % update this path addpath(genpath('C:\Users\eleny\code\ROSCO_toolbox-develop\Matlab_Toolbox')); addpath(genpath('C:\Users\eleny\LINEARIZATION\OpenFAST-Dev2.8a_Models_Dist\13m-Cans\SubDyn-MAP\Hs1.13-WT11.4\20msLIN')); % addpath(genpath('C:\Users\eleny\controller_verf')); addpath(genpath('C:\Users\eleny\LINEARIZATION\matlab-toolbox-master')); LinTimes = 36; % number of lin files you have newFSTName = 'Model.fst'; % change this accordingly % newFSTName = 'NREL5MW_DAC.fst'; FileNames = strcat( strrep(newFSTName, '.fst','.'), strrep( cellstr(num2str( (1:LinTimes)')), ' ', ''), '.lin'); fx_mbc3(FileNames)
```

The runMBC gives you the average across the NLinTimes azimuth positions.. It will have AAvg, BAvg, CAvg, DAvg, etc.

## D. Selecting States

To select specific states (all but the Craig-Bampton modes and generator position) look at the <u>DescStates</u> cell of the runMBC output struct. The DescStates is the order of the states. So, to select the states you want, choose those indices that correspond to the states described in the DescStates cell array.

Note: the generator position state results in an unstable eigenvalue at zero for the AvgA matrix based on the torque-to-position relationship and that it is common practice to eliminate this state since generator position is not usually used in control.

Below is a figure of the output from runMBC.m. DescStates has the order of the states, because they are in a different order than what is in the linearization file.

```
DescStates: {56×1 cell}

ndof2: 28

ndof1: 0

RotSpeed_rpm: 9.5828

WindSpeed: 20

performedTransformation: 0

A: [56×56×8 double]

B: [56×127×8 double]

C: [896×56×8 double]

D: [896×127×8 double]

AvgA: [56×56 double]

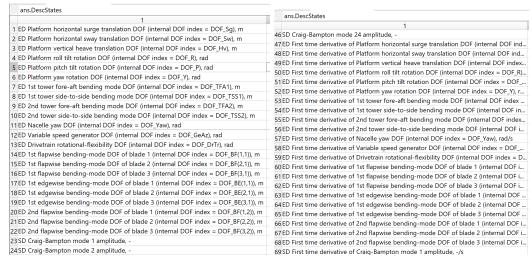
eigSol: [1×1 struct]

AvgB: [56×127 double]

AvgC: [896×56 double]

AvgC: [896×56 double]
```

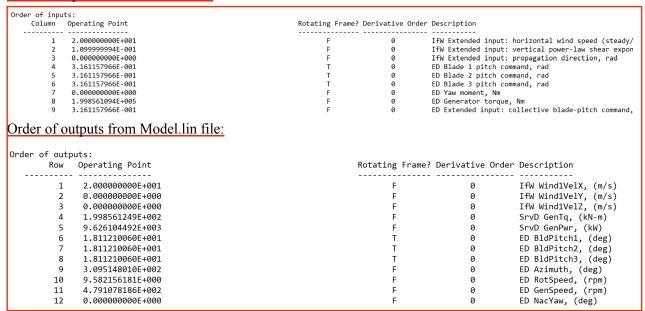
Inside DescStates you will see this (this example has all the DoF turned on, so the maximum number of states). For this example you would choose the states (1:11, 12:22, 47:68).



## E. Selecting Inputs and Outputs

The outputs and inputs of interest are in the same order as they are in the .lin files, so use those corresponding indices to choose your input (should be collective blade pitch, generator torque, wind and wave, and maybe cable tension in the future) and output variables.

Order of inputs from Model.lin file:



F. Putting it all together to get your final state space matrices

Pick the indices of the inputs, states, and outputs to extract the rows and columns of interest from AvgA, AvgB, AvgC, AvgD.

```
A=AAvg(state_idx,state_idx);
B=BAvg(state_idx,input_idx);
C=CAvg(output_idx,state_idx);
D=DAvg(output_idx,input_idx);
```

#### IIII. Verification of the Linearization

Compare to nonlinear model using Simulink. However, this work is pending.

- Step 1. Update ElastoDyn initial conditions for rotor speed and platform motion from average of post-mbc outputs.
- Step 2. Make sure inflow wind has the correct steady wind speed.
- Step 3. Get the avg GenTq and BldPitch from post-run mbc input values. Use these as inputs to the nonlinear model. Make sure in ServoDyn that PCMode and VSCntrl both equal 4.
- Step 4. Run nonlinear model.
- Step 5. Run motion adjusted wind speed calculation code, save this as a .mat file. ?
- Step 6. Use the same GenTq and BldPitch input into the linear model. Put the motion adjusted wind speed as an input to the linear model.
- Step 7. Run linear model.

