Contents

Homework Wind 3

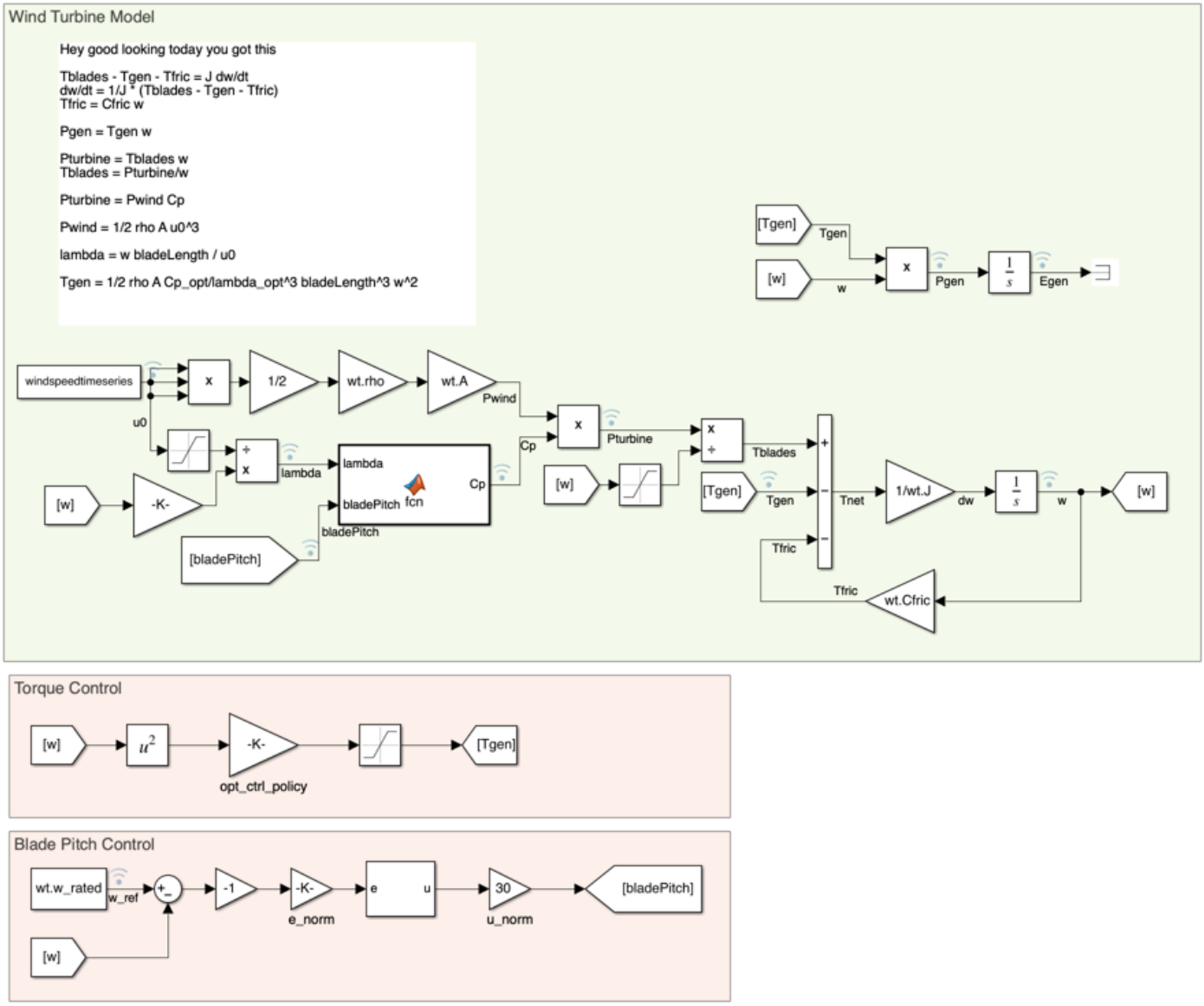
Yen-Chun Chen 934559635

```
% "Publish" this file when have completed items #1 through #3. I started #1
% and #2 for you.
%
% Tip: When using the "sim" command to programmatically call Simulink in
% conjunction with "publish", you'll sometimes get a bunch of warnings
% output to the published file.  You can suppress most of these by putting
% "warning off" in your script before the sim command.  The algebraic loop
% warning can be turned off in the "Model Settings" -> "Diagnostics" dialog
% in under the Simulink "Modeling" tab.
```

Run initialization and simulation

```
microgrid_y24f_step3_init
open_system("microgrid_y24f_step3")       % Includes image of block diagram in publish
simresults = sim("microgrid_y24f_step3");  % Run simulation
log = simresults.logsout;
```

```
Good job!  The init file ran succesfully, hopefully the simulation does too.
```

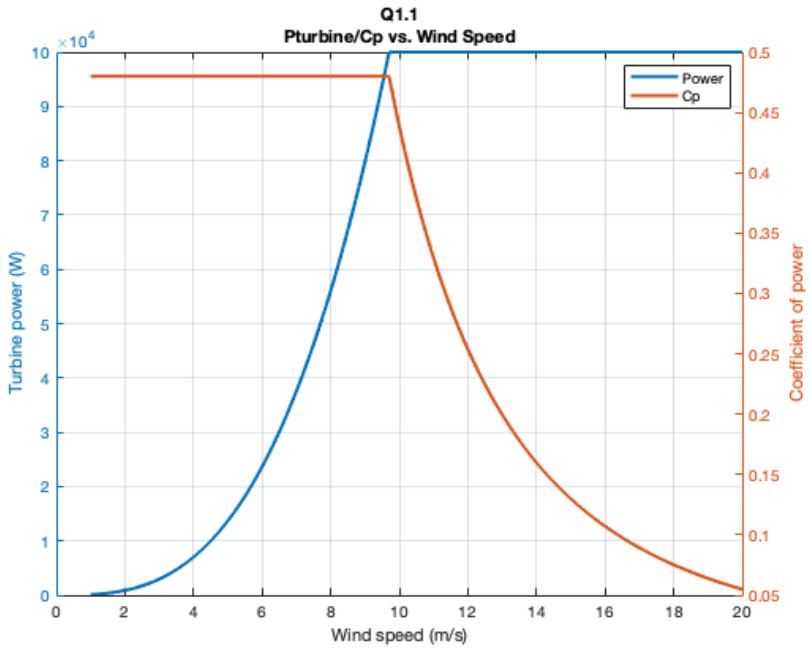


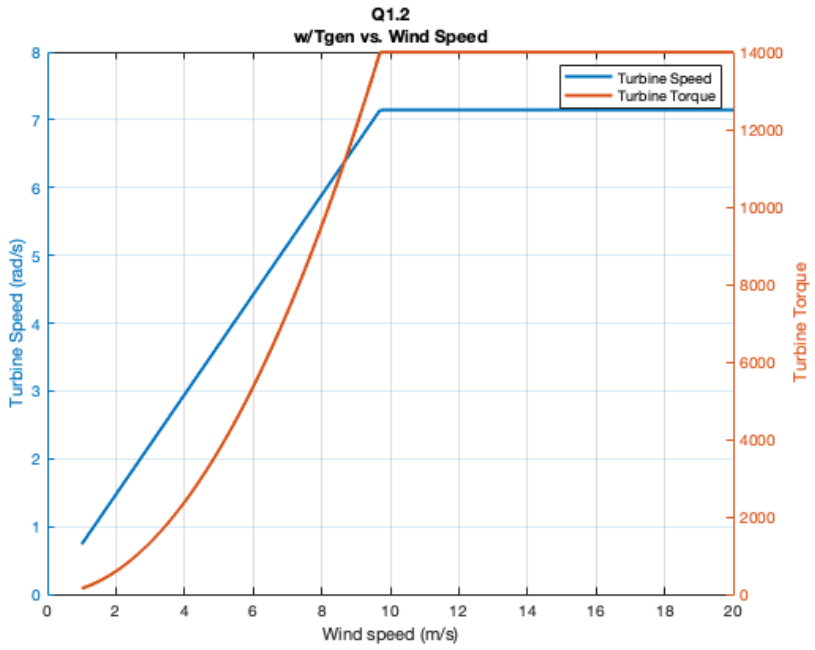
Question 1

Make 2 plots https://www.mathworks.com/help/matlab/creating_plots/plotting-with-two-y-axes.html For this question 1, don't use the dynamic simulation output, but instead create the plots assuming steady state. Assume no region 1 or region 4. Recall that in region 2 the tip speed ratio is fixed at optimal, and the coefficient of power is optimal. In region 3, the turbine speed and torque are constrained to their rated values.

```
wt.Pgen_rated = 100e3;
wt.bladeLength = 11;
wt.A = pi*wt.bladeLength^2;

% From Cp(lambda) plot
wt.lambda_opt = 8.1;
wt.Cp_opt = 0.48;

wt.u_rated = (wt.Pgen_rated/(wt.Cp_opt*0.5*1.2*wt.A))^(1/3);    % P_rated = Cp_max*0.5*A*bladeLength*u_rated^3
wt.w_rated = wt.lambda_opt*wt.u_rated/wt.bladeLength;
wt.Tgen_rated = wt.Pgen_rated/wt.w_rated;

u0 = [1:0.1:20];
Pwind = 1/2*1.2*wt.A*u0.^3;

Pturbine(u0<wt.u_rated) = Pwind(u0<wt.u_rated)*wt.Cp_opt;    % Region 2
Pturbine(u0>=wt.u_rated) = wt.Pgen_rated;                    % Region 3
Cp(u0<wt.u_rated) = wt.Cp_opt;
Cp(u0>=wt.u_rated) = Pturbine(u0>=wt.u_rated)./Pwind(u0>=wt.u_rated);

% The code block above uses a MATLAB technique called "logical indexing"
% It allows you to modify parts of a vector very compactly
% Example: A = [4 7 8], A(boolean([1 0 1])) = [4 8]

% - Turbine power vs wind speed on the left y axis and Cp vs wind speed on
% the right y axis.
```

```
figure
yyaxis left                     % Turbine power on the left
plot(u0,Pturbine,"LineWidth",2)
ylabel("Turbine power (W)")

yyaxis right                    % Coefficient of power on the right
plot(u0,Cp,"LineWidth",2)
ylabel("Coefficient of power")
legend({"Power","Cp"})
xlabel("Wind speed (m/s)")
title({'Q1.1','Pturbine/Cp vs. Wind Speed'})
grid on

w(u0<wt.u_rated) = wt.lambda_opt*u0(u0<wt.u_rated)/wt.bladeLength;
w(u0>=wt.u_rated) = wt.w_rated;
Tgen(u0<wt.u_rated) = Pwind(u0<wt.u_rated)*wt.Cp_opt ./ w(u0<wt.u_rated);
Tgen(u0>=wt.u_rated) = wt.Tgen_rated;

% SECOND FIGURE HERE
% - Turbine speed vs wind speed on the left y axis and turbine torque vs
% wind speed on the right y axis
figure
yyaxis left                     % Turbine speed on the left
plot(u0,w,"LineWidth",2)
ylabel("Turbine Speed (rad/s)")

yyaxis right                    % Turbine torque on the right
plot(u0,Tgen,"LineWidth",2)
ylabel("Turbine Torque")
legend({"Turbine Speed","Turbine Torque"})
xlabel("Wind speed (m/s)")
title({'Q1.2','w/Tgen vs. Wind Speed'})
grid on
```



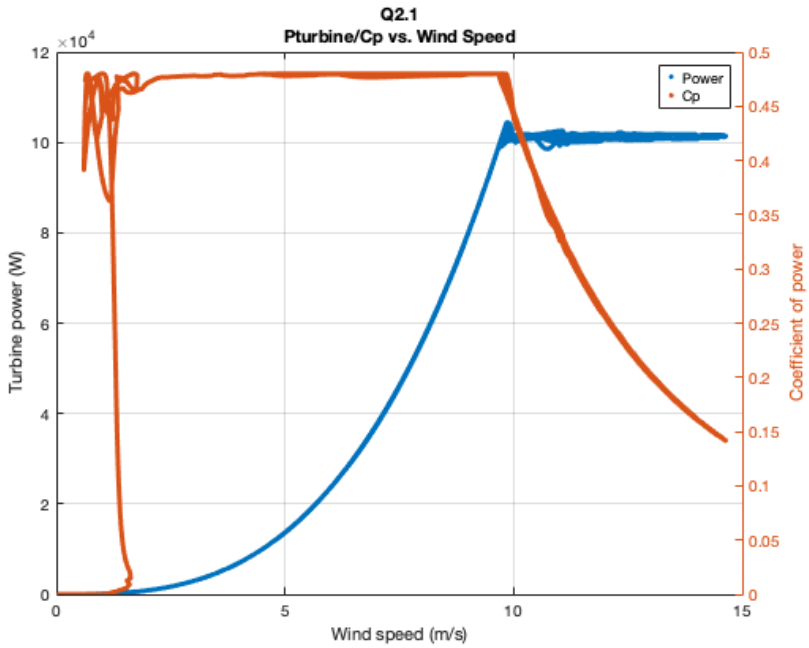


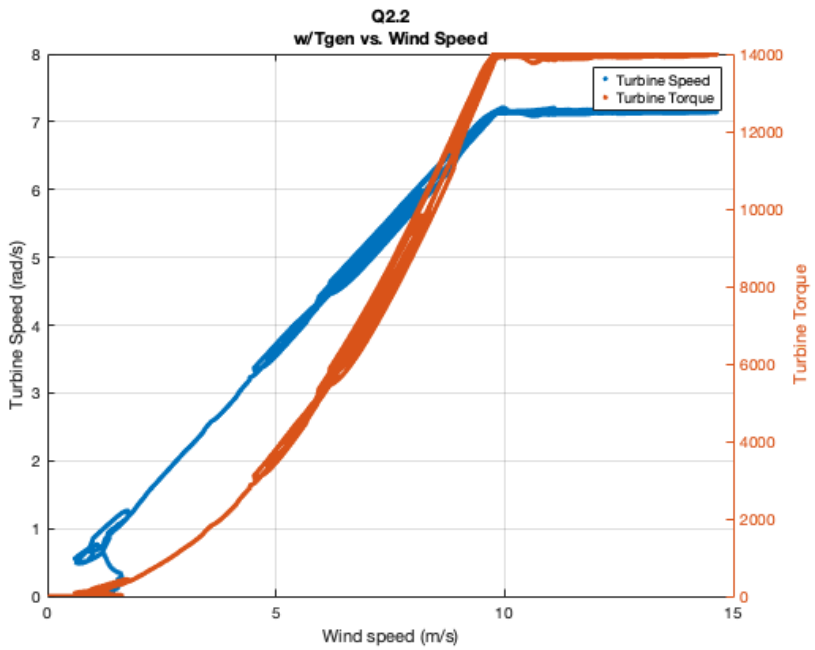
Part 2

Using the simulation, recreate the two plots from part 1 using scatter plots of the recorded power, coefficient of power, torque, and speed all vs wind speed. Comment on if the results match. Be mindful of different yaxis scales.

```
figure
plot(log.getElement("u0").Values.Data,log.getElement("Pturbine").Values.Data,".")
ylabel("Turbine power (W)")
yyaxis right
plot(log.getElement("u0").Values.Data,log.getElement("Cp").Values.Data,".")
ylabel("Coefficient of power")
legend({"Power","Cp"})
xlabel("Wind speed (m/s)")
title({'Q2.1','Pturbine/Cp vs. Wind Speed'})
grid on

% SECOND FIGURE HERE
figure
plot(log.getElement("u0").Values.Data,log.getElement("w").Values.Data,".")
ylabel("Turbine Speed (rad/s)")
yyaxis right
plot(log.getElement("u0").Values.Data,log.getElement("Tgen").Values.Data,".")
ylabel("Turbine Torque")
legend({"Turbine Speed","Turbine Torque"})
xlabel("Wind speed (m/s)")
```

```
title({'Q2.2','w/Tgen vs. Wind Speed'})
grid on

% COMMENT ON IF THE FIGURES FOR PARTS 1 AND 2 MATCH
% In figure Q1-1 and Q2-1, both show similar trends for region 2 and region 3.
% In region 2, the turbine power increases with wind speed up to a certain
% point to keep the optimal lambda and Cp. The turbine power keeps
% increasing until it reaches the rated power and then stabilizes it, which
% is region 3.
% There are some fluctuations when wind speed is low. The reason might be
% the slow response or lag in the turbine adaption to the fast changing
% wind speeds since Cp = Pturbine/Pwind. e.g. When the when wind speed is
% low, and Pwind is low but Pturbine might already be high and the rotating
% turnbine cannot slow down immediately.

% In figure Q1-2 and Q2-2, both also show similar trends for region 2 and region 3.
% Also, there are some fluctuations since the w and Tgen cannot response
% immediately to the fast-changing wind speeds.
```





Part 3

Using the simulation, integrate Pgen to get the energy over the simulation time. Then calculate the capacity factor. There are at least two ways to do this:

```
% 1) Within the simulation, simply use an integrator block to integrate power, and log that signal.
% To calculate average power, just divide that energy by the simulation time, which could be
```

```
% done within the simulation, or in post-processing;
fprintf("Method 1:\n");
Egen = log.getElement("Egen").Values.Data(end)
Pgen_average = Egen/simu.endTime
CF = Pgen_average/wt.Pgen_rated

% 2) When the simulation is complete, access the saved power signal and use the "trapz" command to
% integrate: trapz(time,data).
% Again, divide by the total time to get the average power.
% Note that we are calculating the capacity factor over 1 day.
% Thus, depending on the wind speed, you may get a capacity factor
% well above or below the yearly average, if that particular day is windier
% or calmer than average.

fprintf("\nMethod 2:\n");
Egen_trapz = trapz(log.getElement("Pgen").Values.Time,log.getElement("Pgen").Values.Data)
%Egen_trapz = trapz(log.getElement("Pgen").Values.Data) * simu.maxStepSize;
CF_trapz = Egen_trapz/(wt.Pgen_rated*simu.endTime)

fprintf("\nCapacity Factor = %.2f%%", CF*100);
```

```
Method 1:
Egen =
   5.2560e+09
Pgen_average =
   6.0833e+04
CF =
    0.6083

Method 2:
Egen_trapz =
   5.2560e+09
CF_trapz =
    0.6083

Capacity Factor = 60.83%
```

Published with MATLAB® R2024b