Table of Contents


```
% MEAM 5480 - Homework 4
% Analysis of offshore wind farm data
% Sample code by Nathan Wei
% Revised by Christopher Bianco
```

Inputs

```
basedir = pwd();
file_latlon = fullfile(basedir, 'Displaced_Farm_LatLon.csv');
file_data = fullfile(basedir, 'wind_data.csv');
rho = 1.2; % air density (guess)
D = 154; % m

tInd_plot = 660; % index of time step to plot (1 to 2016) ***
```

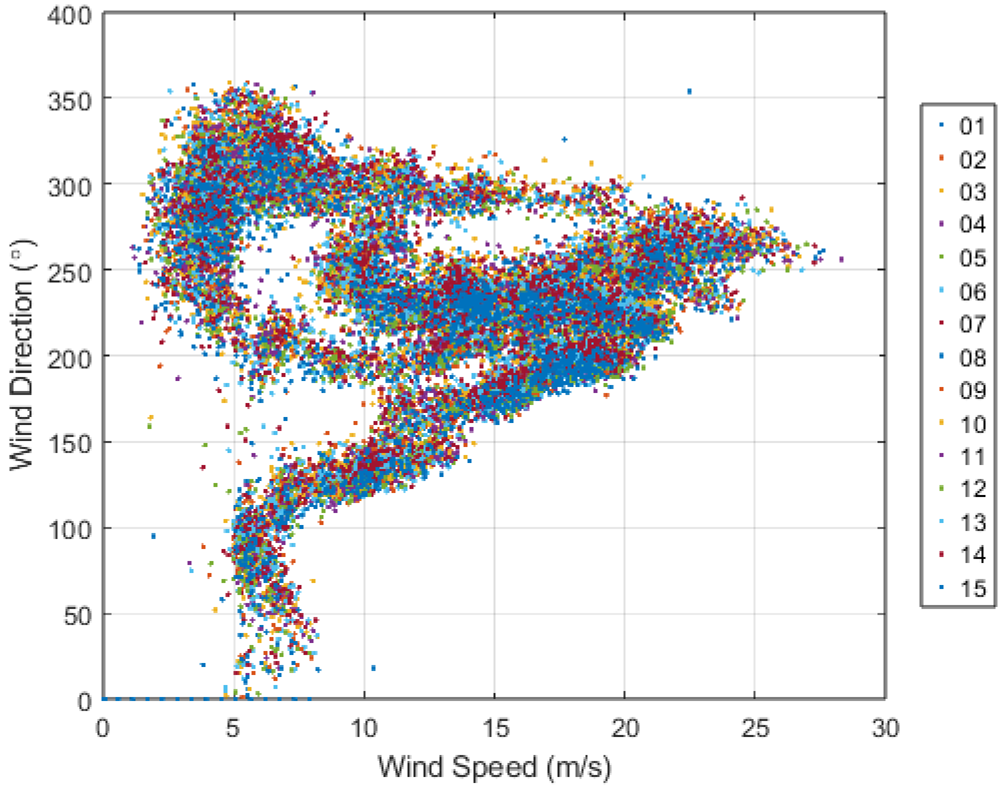
Import data

```
latlon = readtable(file_latlon);
names = table2array(latlon(:,1)); % turbine names
for ii = 1 : length(names)
    names{ii} = names{ii}(4:end); % shorten names to only numbers
end
latlon = table2array(latlon(:,2:3)); % 1st column = latitude, 2nd column =
longitude
lat = (latlon(:,1) - (latlon(1,1))) * 111; % approximate conversion to km
lon = (latlon(:,2) - (latlon(1,2))) * 111; % with shift so first turbine is
at (0, 0)

warning('off','all'); % suppress warning about table variable names
opts = detectImportOptions(file_data);
opts.VariableOptions(2).FillValue = 'NaN'; % fill missing data as NaNs
```

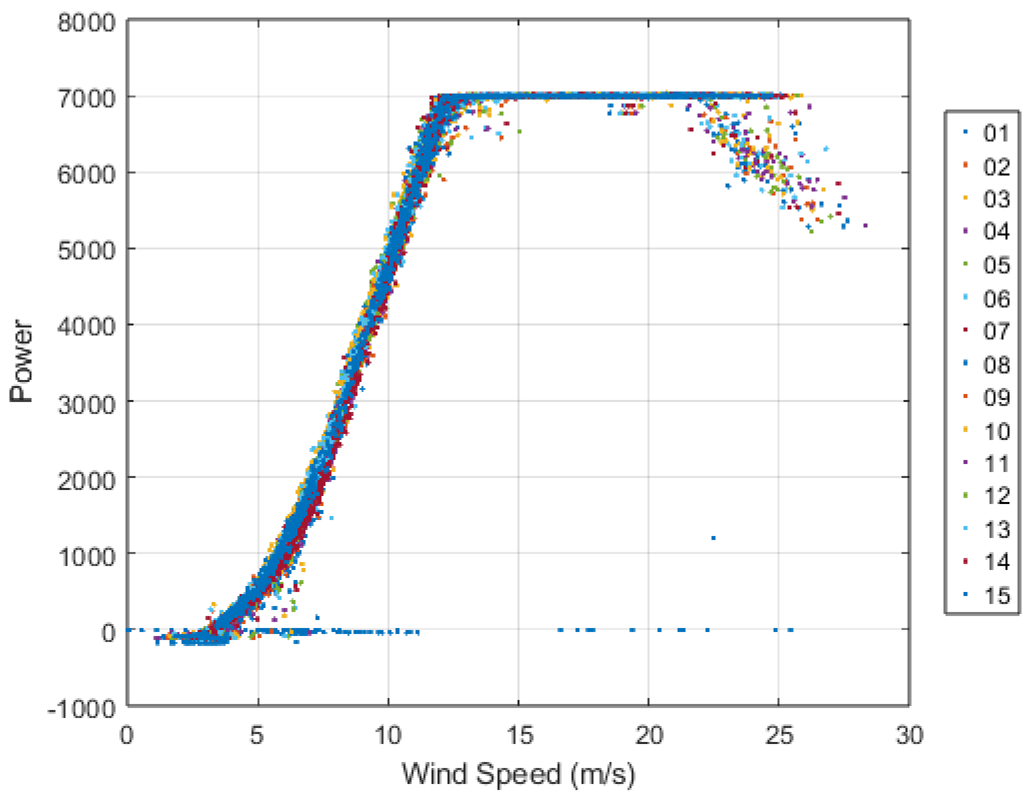
```
data = readtable(file_data,'delimiter',',');
warning('on','all');
time = datetime(table2array(data(:,1)), 'InputFormat', 'yyyy-MM-dd
HH:mm:ss+00:00');
% Each of these matrices has one row per 5-minute interval (2016 total)
% and one column per wind turbine (15 total)
wind_dir = table2array(data(:,2:5:end)); % wind direction (deg)
wind_speed = table2array(data(:,3:5:end)); % wind speed (m/s)
rot = table2array(data(:,4:5:end)); % rotor speed (RPM)
power = table2array(data(:,5:5:end)); % power (kW)
yaw = table2array(data(:,6:5:end)); % yaw angle (deg)
```

Compute power curves

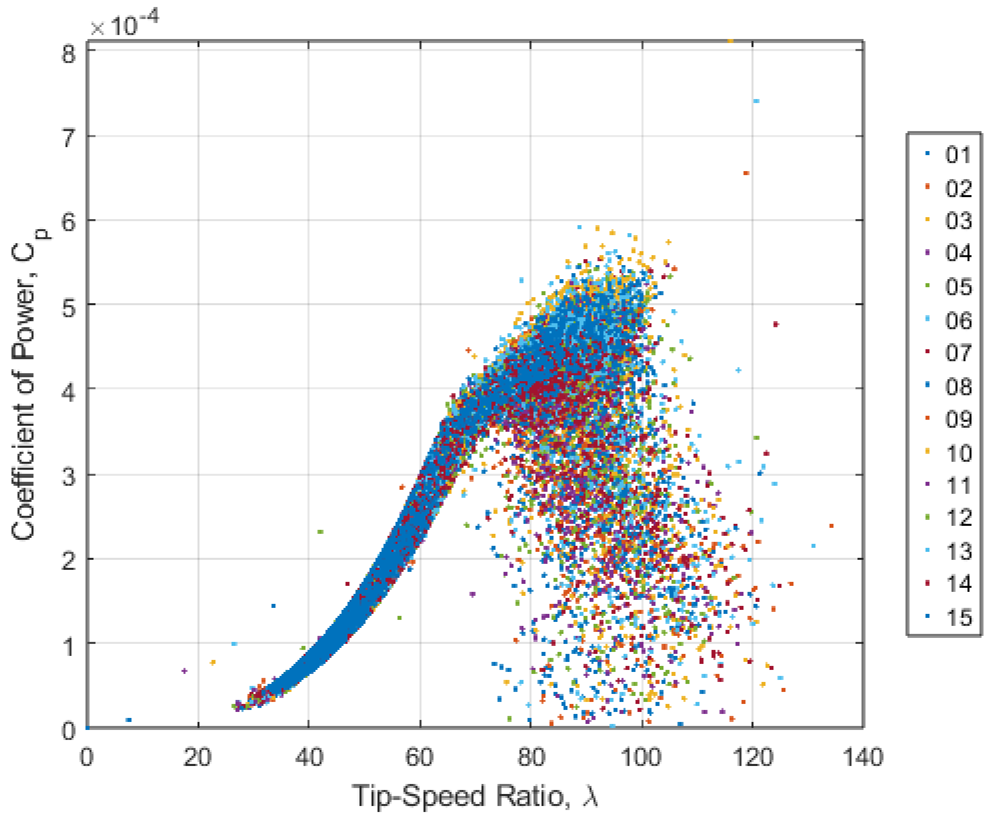
```
% Example plotting code for the above variables
figure;
plot(wind_speed, wind_dir, '.');
xlabel('Wind Speed (m/s)');
ylabel('Wind Direction (\circ)');
legend(names, 'location', 'eastoutside');
grid on;

figure;
plot(wind_speed, power, '.');
xlabel('Wind Speed (m/s)');
ylabel('Power');
legend(names, 'location', 'eastoutside');
grid on;

Cp = power ./(0.5*rho*pi*(77^2).*wind_speed.^3); % ***
TSR = rot .* 77./wind_speed; %***
Cp(Cp<0) = NaN;
figure;
plot(TSR, Cp, '.');
ylim([0, max(Cp(:))]);
xlabel('Tip-Speed Ratio, \lambda');
ylabel('Coefficient of Power, C_p');
legend(names, 'location', 'eastoutside');
grid on;
```



Wind Direction (°)
Wind Speed (m/s)
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15



Power
Wind Speed (m/s)
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15



Part a short answer

```
%From the second plot above, we can see that the cut in wind speed is
%around 4 m/. The rated power is 7000 kW, the rated wind speed is around 15
%m/s, and the cut out wind speed starts around 22 m/s.
```

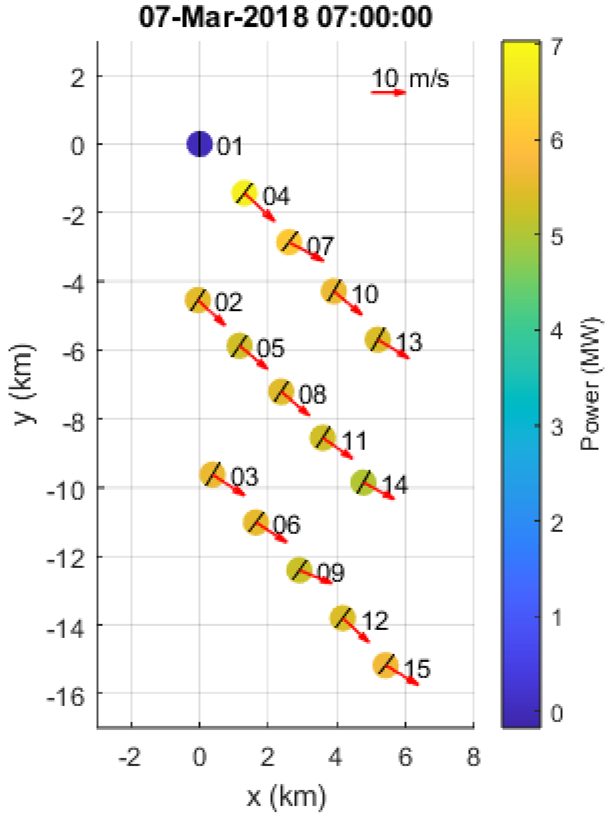
Part b short answer

See plot above. There is so much scatter at high tip speed ratios because that's when we start to enter the cut out wind speeds, so the turbines have to be controlled to reduce their power. This can be done in a variety of ways, and means the power doesn't follow the typical curve because the turbine is being made to perform unoptimally.

Plot wind vectors and wind-farm power generation

```
scaleArrowXY = [5; 1.5]; % lat/lon location for scale arrow
arrowScale = 10; % m/s per km of arrow length on plot
turbineScale = 5; % multiplier for turbine diameter shown on plot
% (e.g. 10 yields a 1 km long arrow for a 10 m/s wind speed)

% Wind direction is where the wind is coming FROM (hence negative sign)
u = -wind_speed .* cos(deg2rad(wind_dir));
v = -wind_speed .* sin(deg2rad(wind_dir));
```

```
% Calculate turbine orientations (draw lines 90 deg from yaw angle)
WTGx1 = lat' + (D/2e3) * sin(deg2rad(yaw))*turbineScale;
WTGy1 = lon' - (D/2e3) * cos(deg2rad(yaw))*turbineScale;
WTGx2 = lat' - (D/2e3) * sin(deg2rad(yaw))*turbineScale;
WTGy2 = lon' + (D/2e3) * cos(deg2rad(yaw))*turbineScale;
% Set x and y limits based on longest expected wind vectors
xlims = [floor(min(lat) + min(u(:))/10); ceil(max(lat) + max(u(:))/
arrowScale)];
ylims = [floor(min(lon) + min(v(:))/10); ceil(max(lon) + max(v(:))/
arrowScale)];

% Plot single frame
figure;
scatter(lat, lon, 100, power(tInd_plot,:)/1e3, 'filled', 'o');
colormap('parula');
cb = colorbar;
cb.Label.String = 'Power (MW)';
clim([min(power(:)), max(power(:))]/1e3);
hold on;
text(lat+0.5, lon, names);
for ii = 1 : length(lat)
    plot([WTGx1(tInd_plot,ii); WTGx2(tInd_plot,ii)], ...
         [WTGy1(tInd_plot,ii); WTGy2(tInd_plot,ii)], ...
         'k', 'LineWidth', 0.5);
end
quiver([lat; scaleArrowXY(1)], [lon; scaleArrowXY(2)], ...
    [u(tInd_plot,:), 10]'/arrowScale, [v(tInd_plot,:), 0]'/arrowScale,
'off', ...
    'r', 'linewidth', 1);
text(scaleArrowXY(1), scaleArrowXY(2) + 0.5, '10 m/s');
grid on;
axis equal;
xlim(xlims);
ylim(ylims);
xlabel('x (km)');
ylabel('y (km)');
title(string(time(tInd_plot)));
```



Part c short answer

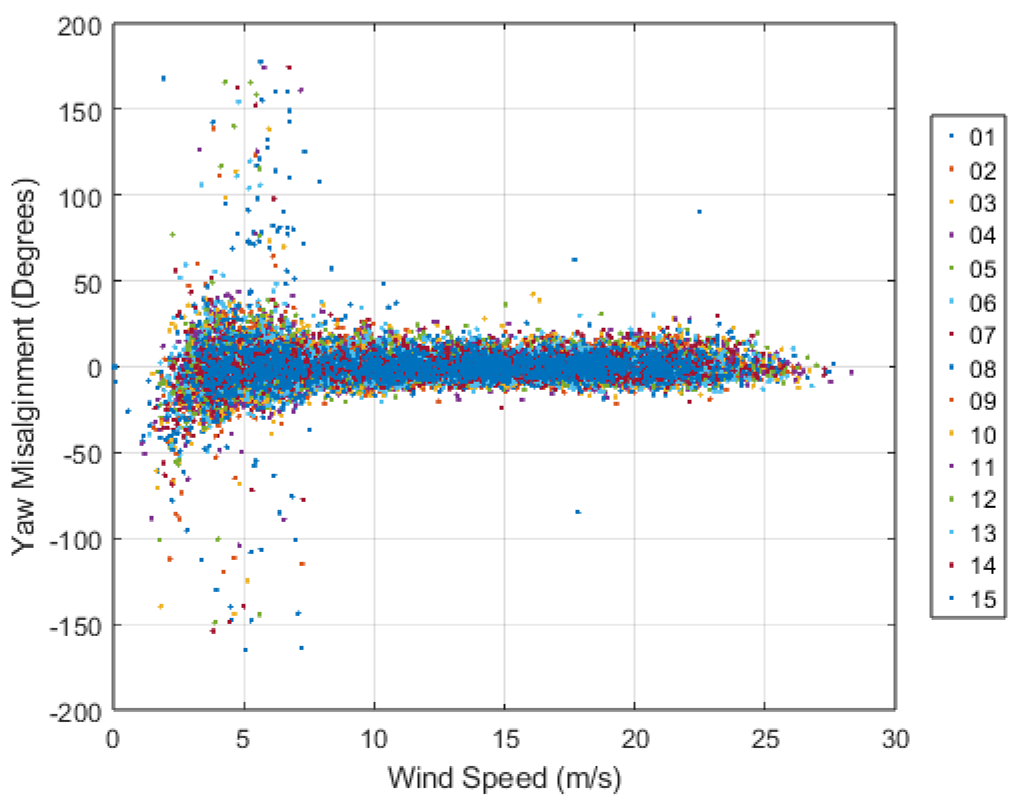
```
%Based on the video, wind speeds are the highest in the very early morning
%(ie 3am). THis is due to low level jets that occur during this time. These
%jets can have flow speeds grater than the geostrophic winds due to the
%stably stratified nature of the ABL at night.
```

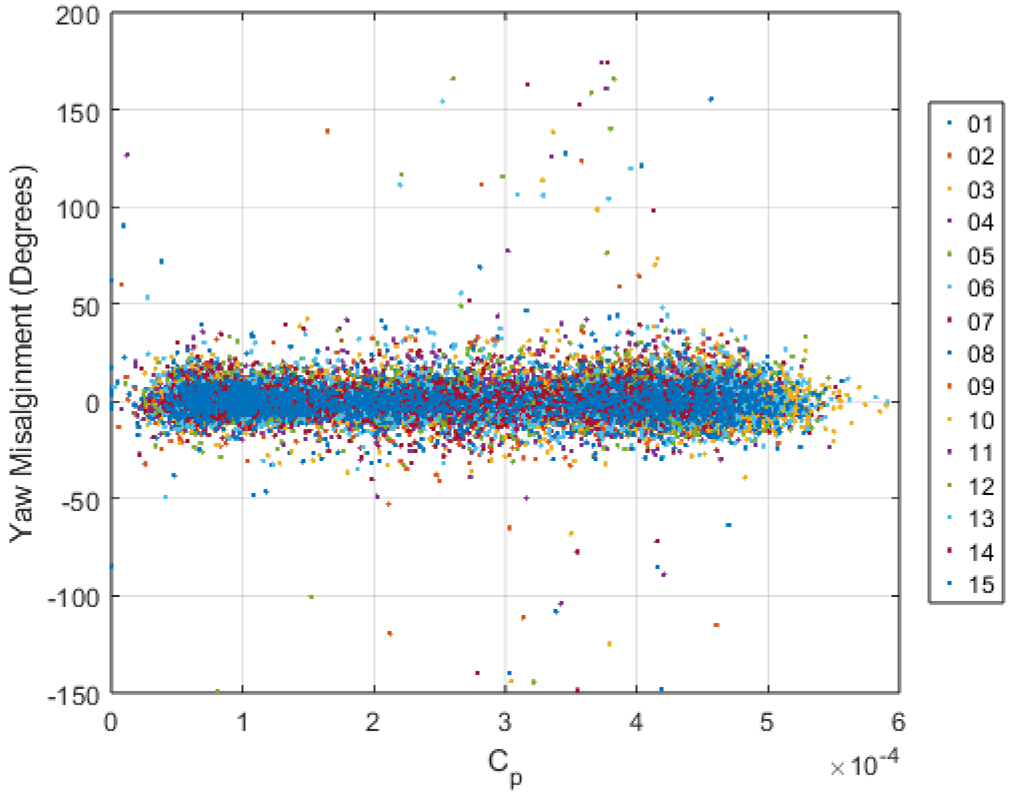
Study yaw misalignment

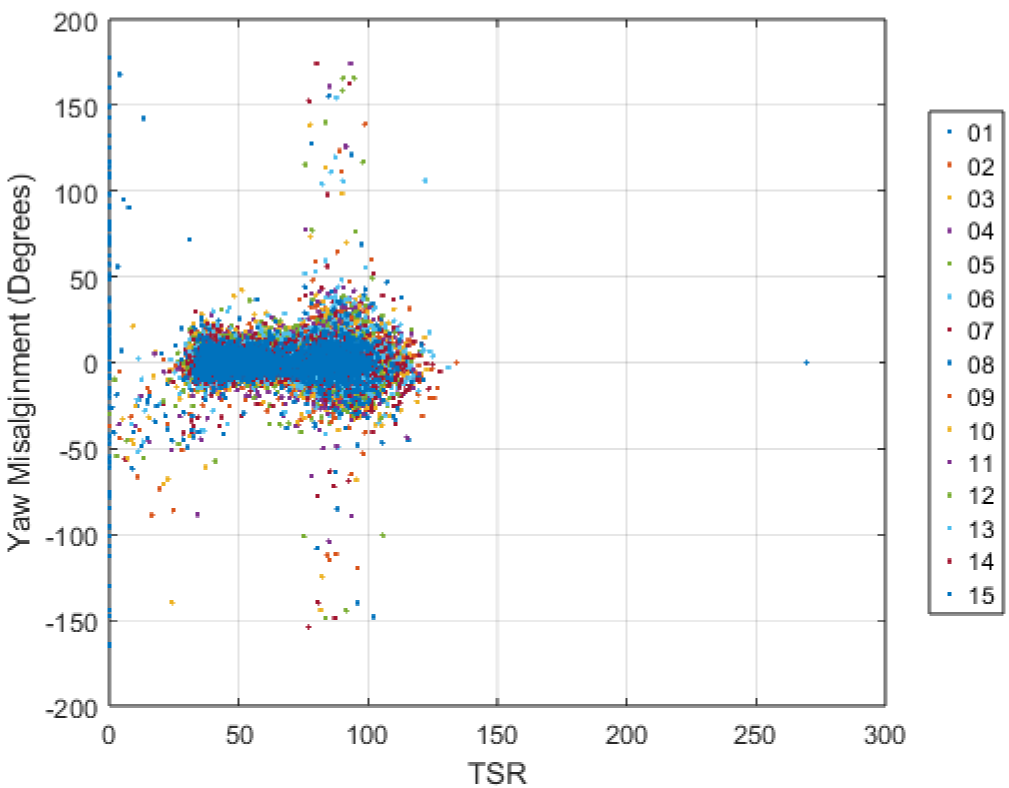
```
yaw_misalign = rad2deg(angdiff(deg2rad(yaw), deg2rad(wind_dir))) ; % ***
% ^ note that angdiff handles logic of subtracting angles on a circle
fprintf('Average magnitude of yaw misalignment: %.2f degrees.\n', ...
    mean(abs(yaw_misalign(:)), 'omitnan'));

% Make various plots here to see what affects yaw misalignment ***
figure;
plot(wind_speed, yaw_misalign, '.');
xlabel('Wind Speed (m/s)');
ylabel('Yaw Misalginment (Degrees)');
legend(names, 'location', 'eastoutside');
grid on;

figure;
plot(Cp, yaw_misalign, '.');
xlabel('C_p');
```

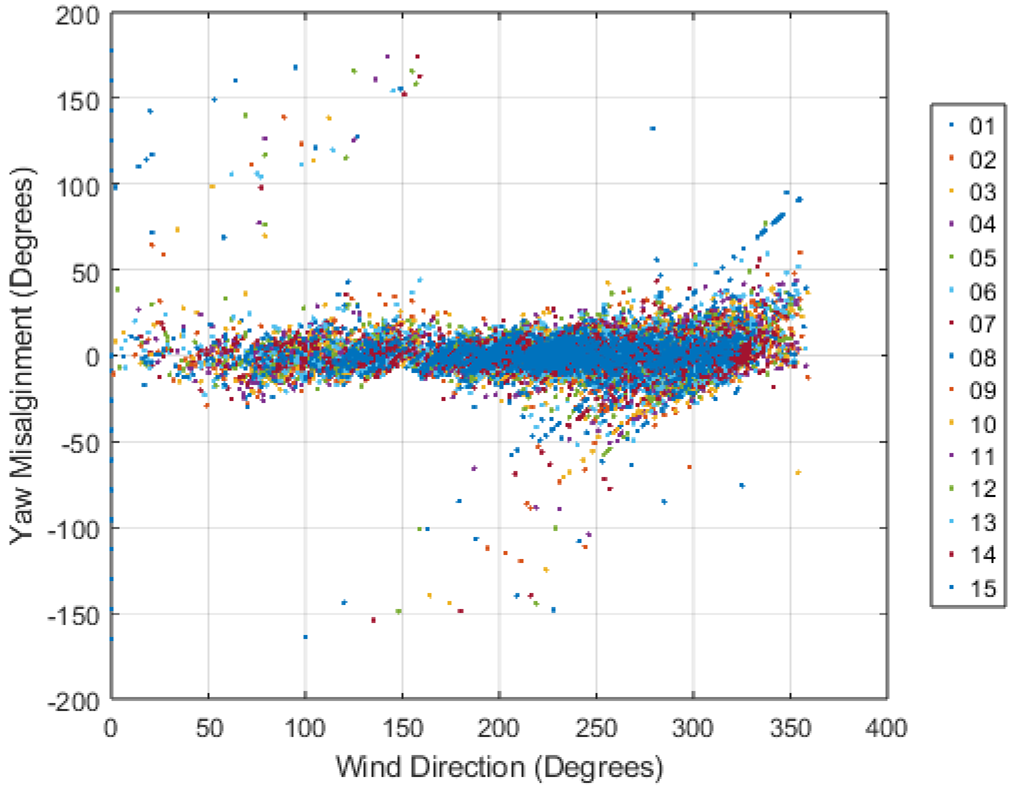
```
ylabel('Yaw Misalginment (Degrees)');
legend(names, 'location', 'eastoutside');
grid on;

figure;
plot(TSR, yaw_misalign, '.');
xlabel('TSR');
ylabel('Yaw Misalginment (Degrees)');
legend(names, 'location', 'eastoutside');
grid on;

figure;
plot(wind_dir, yaw_misalign, '.');
xlabel('Wind Direction (Degrees)');
ylabel('Yaw Misalginment (Degrees)');
legend(names, 'location', 'eastoutside');
grid on;
```

Average magnitude of yaw misalignment: 5.70 degrees.




Yaw Misalginment (Degrees)
C_p
×10^-4
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15


Yaw Misalginment (Degrees)
TSR
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15



Part d short answer

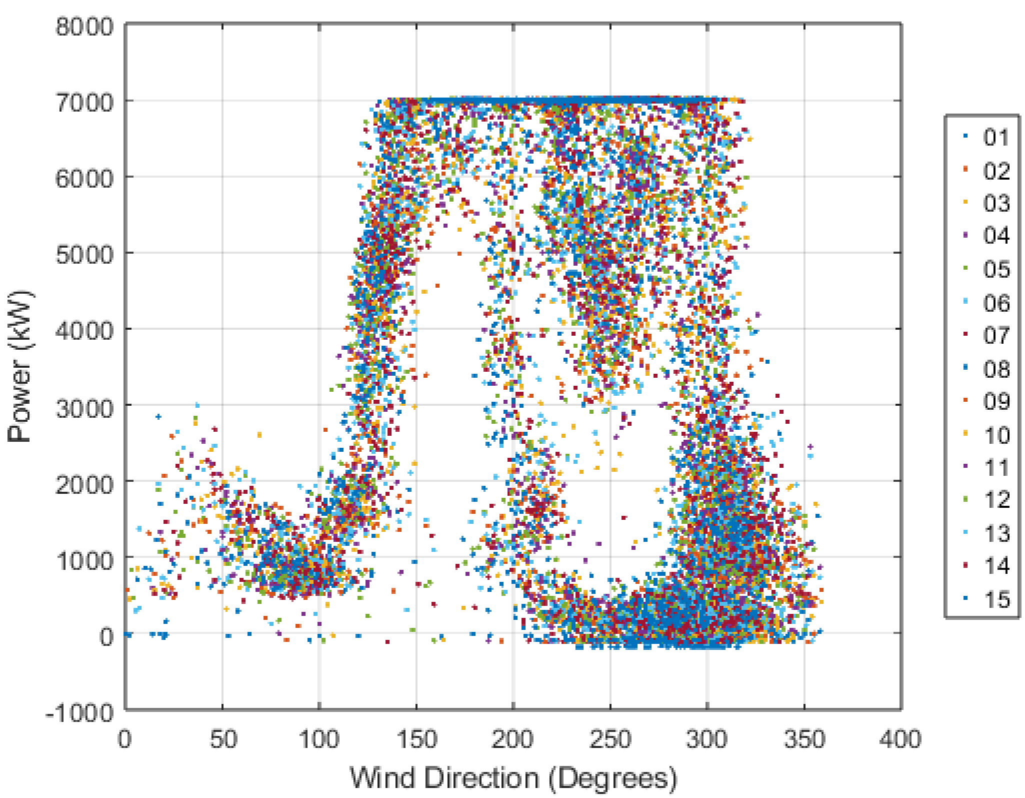
```
%Yaw misalignment seems to be the worst at low wind speeds. This is likely
%because the anemometers on the wind turbinbes have lots of noise that
%these low wind speeds and thus the controller has a harder time
%distingushing the mean wind direction from small, low speed fluctuations.
%See figure 5 above, which shows large nonzero yaw misalginnment at low
%wind speeds that converges after roughly 7 m/s.
```

Study wake losses

```
figure;
plot(wind_dir, power, '.');
xlabel('Wind Direction (Degrees)');
ylabel('Power (kW)');
legend(names, 'location', 'eastoutside');
grid on;

binWidth = 5; % degrees for binning wind direction
total_power_time = nansum(power, 2); %kW per time step
total_power = mean(total_power_time); % kW
mean_wind_dir = mean(wind_dir, 2, 'omitnan');


edges = 0:binWidth:360; %Make bins
```

```
centers = edges(1:end-1) + binWidth/2; %Make bins
mean_power_per_bin = nan(size(centers));
count_per_bin = zeros(size(centers));

for b = 1:length(centers)
    % find times where wind direction (mean_wind_dir) falls within the bin
    d = wrapTo180(mean_wind_dir - centers(b));  % circular difference in
degrees
    inBin = abs(d) <= binWidth/2;
    mean_power_per_bin(b) = mean(total_power_time(inBin), 'omitnan');
    count_per_bin(b) = sum(inBin);
end

% 3) Find worst-case direction (bin center with smallest mean total power,
require some samples)
validBins = count_per_bin >= 5; % require at least 5 samples in bin to be
credible
[~, idx_min] = min(mean_power_per_bin(validBins));
validCenters = centers(validBins);
phi_wc = validCenters(idx_min);
fprintf('Identified worst-case farm wind direction phi_wc = %.1f deg (bin
width %.1f deg).\n', phi_wc, binWidth);
```

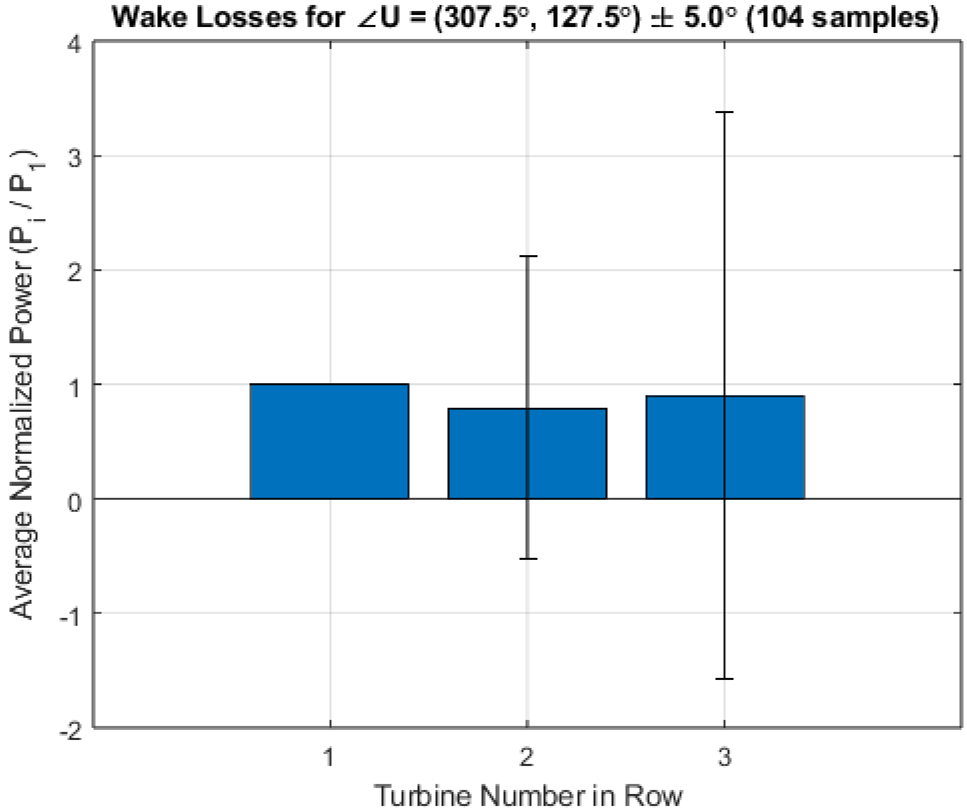
Identified worst-case farm wind direction phi_wc = 307.5 deg (bin width 5.0 deg).



Conditional average

Input parameters*

```
wind_dir_worstcase = 307.5; % deg ***
worstcase_plusminus = 5; % deg (plus/minus about worst-case direction) ***
WTG_select = [2,3,4]; % array of indices, in order from upwind to downwind
***

% Conditional sampling (CS) and averaging for worst-case wind direction(s)
mean_wind_dir = mean(wind_dir, 2, 'omitnan');
tInds_CS0 = find(abs(angdiff(deg2rad(mean_wind_dir), ...
    deg2rad(wind_dir_worstcase*ones(size(time))))) ...
    <= deg2rad(worstcase_plusminus)); % finds time indices that satisfy our
search bounds
tInds_CS180 = find(abs(angdiff(deg2rad(mean_wind_dir), ...
    deg2rad(wind_dir_worstcase*ones(size(time)) + 180)))...
    <= deg2rad(worstcase_plusminus)); % case where wind is in the opposite
direction
power_CS = [power(tInds_CS0, WTG_select); ...
    fliplr(power(tInds_CS180, WTG_select))]; % flip order of turbines for
flipped wind direction
power_CS_norm = power_CS ./ power_CS(:,1); % normalize by first turbine in
row
mean_power_CS_norm = mean(power_CS_norm, 'omitnan');
stddev_power_CS_norm = std(power_CS_norm, [], 'omitnan');

% Plot results
figure;
WTG_number = 1 : length(WTG_select);
bar(WTG_number, mean_power_CS_norm);
hold on;
errbar = errorbar(WTG_number, mean_power_CS_norm, stddev_power_CS_norm, ...
    stddev_power_CS_norm, 'k', 'LineStyle', 'none');
title(sprintf('Wake Losses for %s = (%.1f%s, %.1f%s) %s %.1f%s (%d
samples)', ...
    '\angleU', wind_dir_worstcase, '\circ', wrapTo360(wind_dir_worstcase +
180), ...
    '\circ', '\pm', worstcase_plusminus, '\circ', size(power_CS_norm, 1)));
xlabel('Turbine Number in Row');
ylabel('Average Normalized Power (P_{i} / P_1)');
grid on;
```



Part e short answer

The worst case wind direction is 307.5 +/- 5 degrees. Note that this makes intuitive sense, as this is when the wind lines up with the turbine rows. The total power produced in the time is around 70 MW. As can be seen from the conditional averaging, the wake losses are significant. In the row

```
%formed by turbines 2, 3, and 4, for this wind direction, turbine 3 is
%operating at roughly 80% of 2, and 4 is at about 90%. This corresponds to
%losses on the order of megawatts, which is a lot of power. For reference,
%1 MW can power a home for about a month. Thus, these wake losses are very
%significant.
```

Make video over all frames -- uncomment by removing %{ ... %} %{ vidSpeedFactor = 2; % number of hours per second of video time

```
fig = figure;
videoObject = VideoWriter(fullfile(basedir, 'windfarm'), 'MPEG-4');
videoObject.FrameRate = 12*vidSpeedFactor;
open(videoObject);

% Make frames and record to video
for tInd = 1 : length(time)
```

```
    figure(fig);
    clf;
    scatter(lat, lon, 100, power(tInd,:)/1e3, 'filled', 'o');
    colormap('parula');
    cb = colorbar;
    cb.Label.String = 'Power (MW)';
    clim([min(power(:)), max(power(:))]/1e3);
    hold on;
    text(lat+0.5, lon, names);
    for ii = 1 : length(lat)
        plot([WTGx1(tInd,ii); WTGx2(tInd,ii)], [WTGy1(tInd,ii);
WTGy2(tInd,ii)], ...
            'k', 'LineWidth', 0.5);
    end
    quiver([lat; scaleArrowXY(1)], [lon; scaleArrowXY(2)], ...
        [u(tInd,:), 10]'/arrowScale, [v(tInd,:), 0]'/arrowScale, 'off', ...
        'r', 'linewidth', 1);
    text(scaleArrowXY(1), scaleArrowXY(2) + 0.5, '10 m/s');
    grid on;
    axis equal;
    xlim(xlims);
    ylim(ylims);
    xlabel('x (km)');
    ylabel('y (km)');
    title(string(time(tInd)));
    hold off;
    writeVideo(videoObject, getframe(fig));
end
close(videoObject);
close(fig);
%}
```

Published with MATLAB® R2024b