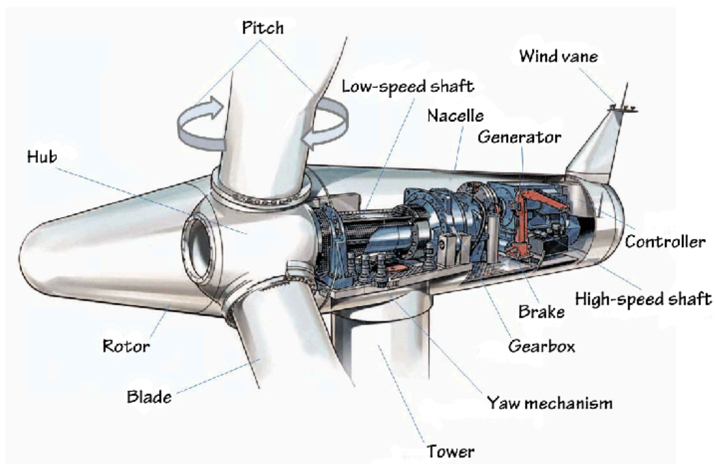
ME 2016 A- Computing Techniques
Fall 2018

Computer Project 1
Due Tuesday, October 23 at 2:55 pm (5 minutes before class time)

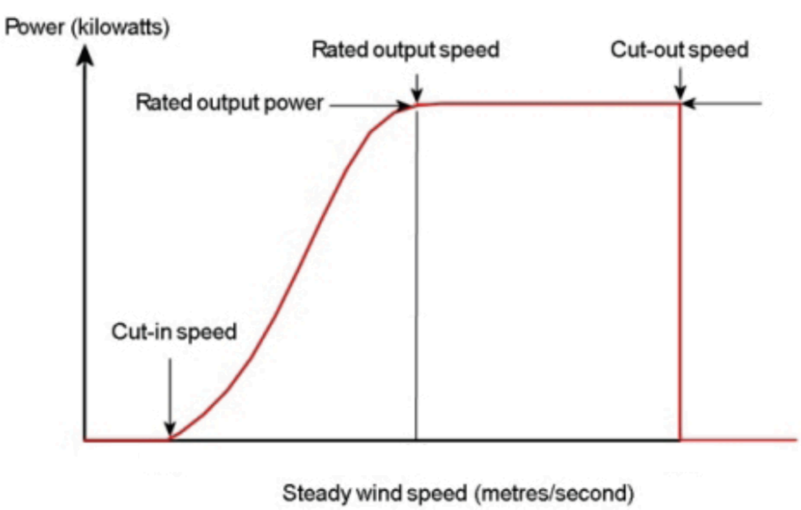
Least-Squares Regression

You have just graduated from Tech and have been hired by the Wind Energy division of Georgia Power. You will be part of a study to determine the viability of generating electrical power from the wind off the Georgia coast, and your team will participate in the design of test turbines. Since you are new to the field, your boss has given you some basic data to analyze and a simple experimental test to perform so that you can become familiar with the fundamentals.



<u>A. Background information on wind power</u>:

The graph below represents the typical power output of a modern wind turbine as a function of wind speed.



The *cut-in speed* is the wind speed at which the blades start to rotate and therefore at which the turbine starts to generate power; below that speed, the wind is not strong enough to make the blades rotate (i.e., it does not exert sufficient torque). Once the cut-in speed is reached, the power output increases quickly with increasing wind speeds. However, beyond a certain speed, the increase rate starts to diminish (there is an *inflexion point* in the curve; if you are not familiar with this term, please look it up) until the so-called *rated output speed* is reached. At that speed, the turbine outputs the maximum power that its electrical generator can produce; this maximum value is called the *rated output power*. Beyond the rated output wind speed, the angle of the turbine blades is adjusted so as to keep their rotational speed constant (the turbine keeps on producing its maximum rated output power) until the wind speed reaches a level that presents risks of structural damage to the rotor; this is the *cut-out speed*, where the rotor is stopped.

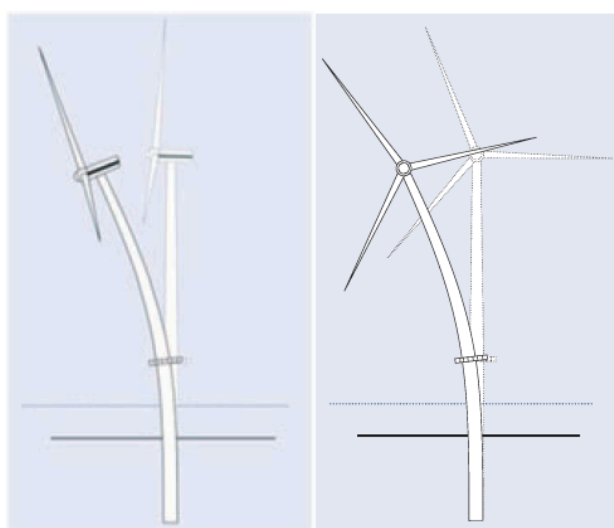
One important characteristic of a wind turbine is its power coefficient, C_p, which relates the power output P (W) to the wind speed v (m.s^{-1}) according to the following equation:

$$P = \frac{\pi}{8} C_p \rho d^2 v^3 \tag{1}$$

where ρ is the air density (1.225 kg.m^{-3}) and d is the rotor diameter. This equation only applies to the turbine regime described by the portion of the curve located between the cut-in speed and the inflexion point. The theory on wind turbine efficiency predicts that the maximum theoretical amount of power that can be extracted from the wind would correspond to a power coefficient of 0.59; in practice, C_p is always lower than this maximum value.

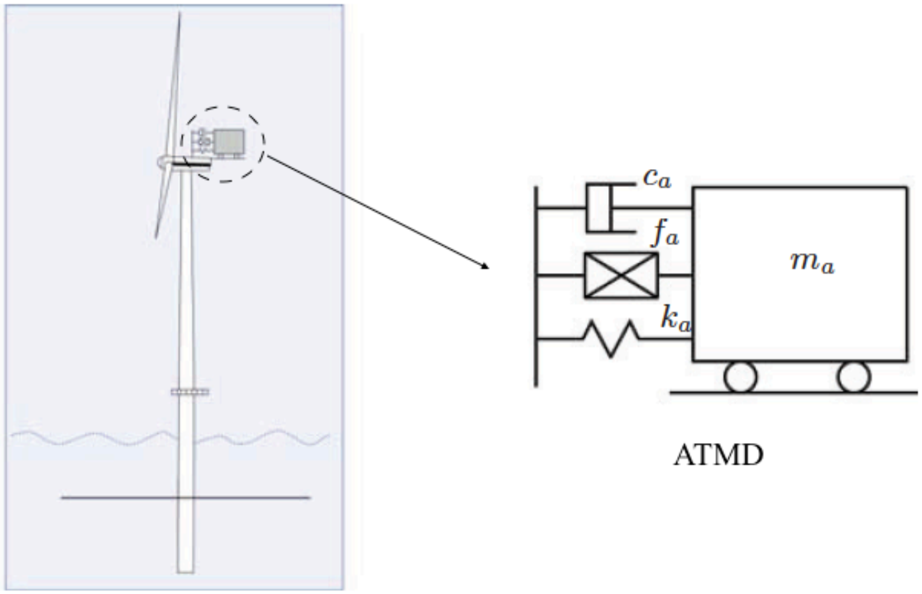
B. Background information on wind turbine vibration damping[1]:

All turbines experience unwanted vibrations, primarily due to fluctuations in the wind flow. In addition to these fluctuations, offshore turbines are also subjected to wave loading, and the damping of the resulting vibrations is especially critical to reduce fatigue damage. The vibrations primarily consist of a combination of the two lowest tower modes, the so-called fore-aft and side-side modes, illustrated below.



[1] Reference and source for illustrations: M. L. Brodersen, Damping of Wind Turbine Tower Vibrations, Ph. D Thesis, *Technical University of Denmark*, Dec. 2015

One way to reduce the vibration amplitude is to attach an Active Tuned Mass Damper (ATMD) to the nacelle, as illustrated schematically in the figure below.

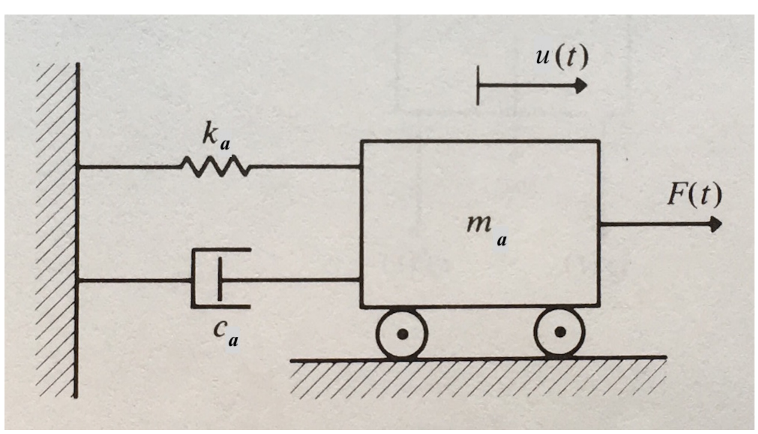


The ATMD is comprised of a dashpot with a coefficient of viscous damping c_a, an actuator characterized by a force f_a, a spring with a spring constant k_a and a mass m_a. For maximum effectiveness, the damper parameters need to be adjusted so that its natural mode of vibration match that of the turbine. Georgia Power has acquired a prototype damper, and you are tasked with the experiment to evaluate two of its crucial characteristics: the natural angular frequency ω_n (units: rad.s^{-1}) and the nondimensional viscous damping factor ζ (whose value is always comprised between 0 and 1), related to the other parameters according to:

$$\omega_n = \sqrt{\frac{k_a}{m_a}} \tag{2}$$

$$\zeta = \frac{c_a}{2m_a\omega_n} \tag{3}$$

Your experimental setup to test the vibration characteristics of the ATMD is shown below.



You have disconnected the actuator ($f_a = 0$) and you apply a harmonic force $F(t)$ at various frequencies: $F(t)$ is proportional to $A\cos(\omega t)$, where A is a known constant, ω is the (varying) angular frequency and t is time, and you measure the resulting displacement $u(t)$, whose amplitude U is a function of the applied frequency ω. The theory predicts that the ratio $|U(\omega)/A| = G(\omega)$, known as the *magnification factor*, is given by

$$G(\omega) = \left|\frac{U(\omega)}{A}\right| = \frac{1}{\left\{\left[1-\left(\frac{\omega}{\omega_n}\right)^2\right]^2 + \left(2\zeta\frac{\omega}{\omega_n}\right)^2\right\}^{1/2}} \tag{4}$$

Description of the experimental data:

Part A: the power.mat file contains measured ouput power P (W) versus wind speed v (m.s^{-1}) data for a turbine with a 123 m rotor diameter.

Part B: the vibration.mat file contains the measured magnification factor G (nondimensional) as a function of the applied frequency ω (rad.s^{-1}).

MATLAB programming:

First, please note the following:

- in MATLAB, `log` is the natural logarithm, while `log10` is the base 10 logarithm
- for all of your plots, the **experimental data** should be plotted as **individual points** not connected by lines (such as large, filled circles), and the **curve fits** should be plotted as **smooth, continuous lines** (use a least 100 points over the given data range).
- in all of your functions below, the `plot_logical` input argument is a logical ("true" or "false") to determine if a plot should be generated: if `plot_logical = true`, the function should **plot the data as well as the fit on the same plot**; if it is omitted in the function call, use `plot_logical = false` as default (the Matlab command `nargin` is useful for this).

1) write a completely generic function to fit a straight line to a data set (x,y) with the following format:

```
function [a0,a1,r] = linear_regression(x,y,plot_logical)
```

where `a0`, `a1` and `r` are the intercept, the slope and the correlation coefficient of the fit, respectively. This function should implement the linear regression technique studied in class and use only vector operations to compute the coefficients a0 and a1 from equations (3) and (4) in my notes (loops are not allowed).

2) write a completely generic function to fit a power law to a data set (x,y) with the following format:

```
function [a,p,r] = power_law(x,y,plot_logical)
```

where `r` is the correlation coefficient of the fit and the power law is defined by $y = ax^p$. This function should call your `linear_regression` function, but the linearized data should never be plotted.

3) write a completely generic function to fit a polynomial to a data set (x,y) with the following format:

```
function [A,r] = polynomial_regression(x,y,m,plot_logical)
```

where `A`, `r` and `m` are the vector of polynomial coefficients, the correlation coefficient, and the order of the polynomial, respectively. This function should calculate the vector of polynomial coefficients (A) by solving the normal equations written in matrix form [M](A) = (V) for a given order m. This is a generalization of equations (5) in my notes, which are the normal equations for quadratic interpolation (case $m = 2$). For a given m, the function should generate the matrix [M] and the vector (V) (you are allowed a maximum of 2 FOR loops for their creation), and solve for (A) using the left matrix divide command (look it up if you are not familiar with it).

Data analysis and report

Part A:

4) Using the provided file, identify which portion of the power vs. wind speed data set is best described by equation (1). To do that, you will have to apply your `power_law` function to various portions of the data and decide on the portion that gives you the best fit. In your report, specify the portion (data range) you identified, and explain the criteria that you used to decide on which part of the data was best modeled by the equation.

5) Write a script called `CP1` that does the following:

- Defines the parameter values used in equation (1)
- Loads the power.mat file
- Plots the entire data set (using circles), with P on the y-axis in a Figure window numbered 1.
- Calls your `power_law` function to do a curve fit over the portion of the data identified in 4) and plots the resulting curve on the plot showing the entire data set (in Figure 1).
- Prints the coefficients a and p corresponding to the fit, as well as the correlation coefficient, in the command window in a nicely formatted way.
- Calculates and prints the power coefficient C_p in the command window in a nicely formatted way.

In your report, include the plot from Figure 1, the formula used to compute the power coefficient and the resulting value of C_p.

6) Now that you have extracted the power coefficient from the portion of the curve that is modeled by a known equation, you are curious to see if you can find a curve fit with your `polynomial_regression` function that would capture the behavior of the entire data set. To do that, you add the following parts to your `CP1` script:

- A polynomial fit of order 3 to the entire data set. Plot the entire data set and the resulting curve in a Figure window numbered 2 and print the corresponding correlation coefficient in the command window in a nicely formatted way.
- Repeat the above for polynomials of order 5 and 7 (Matlab may give you a warning: ignore it). Use Figure 2 to plot each successive polynomial fits, but erase the previous one (i.e., Figure 2 should first have the 3rd order fit, then that fit is erased and Figure 2 shows the 5th order one, etc…). Use the `pause` command in between each polynomial fit in `CP1`, so that the user can examine each polynomial fit appearing in Figure 2 and move one to the next fit by striking any key (do not specify how long the pause should be).

In your report, include all 3 plots from Figure 2 as well as the correlation coefficients and briefly comment on the quality and the usefulness of each curve fit. Conclude on which one is the best, and why.

Part B:

7) Operate a change of variables on equation (4) (original variables (ω,G) --> new variables (X,Y)) so that you can use one of the 3 Matlab functions that you have written for this project to compute the unknowns ω_n and ζ. This operation is similar to (but more slightly more complex than) the linearization procedures that I presented in class.
Hint: if the function you have to use after the change of variables is `polynomial_regression`, do not try to fit a polynomial of order higher than 3. In other words, make sure your change of variables results in a polynomial with an order less than or equal to 3.
In your report, show the math steps that you performed to do the change of variables, clearly define the new variables X and Y and all the new unknown parameters that you will have to compute in order to obtain ω_n and ζ. Show the formulas that you will need to compute ω_n and ζ.

8) Still in `CP1`:

- Load the vibration.mat file
- Plot the entire discrete data set (using circles), with G on the y-axis in a Figure window numbered 3.
- Calculate your new (X,Y) data set and call one of your 3 functions to do a curve fit on it (without plotting that new data or its curve fit).
- Calculate ω_n and ζ from the results of your curve fit and print their values in the command window in a nicely formatted way.
- Plot the curve fit of the original (ω,G) data set in Figure 3 (with the discrete data still displayed in the figure)

In your report, give the values that you obtained for ω_n and ζ, include the plot of the data and the fit from Figure 3 and comment on the quality of the fit.

Make sure you follow the M-file formatting instructions and the report guidelines.

Instructions for Computer Project 1 submission:

Include all 4 m-files and your **PDF-formatted report** in a single zipped folder named **CP1_LASTNAME_FIRSTNAME** and upload to Canvas by the due date and time above. Please use the **.zip** format only to avoid problems with opening your folder zipped in other formats.

Please note the following, valid for all MATLAB assignments:

1. **Refrain from using the commands `clear`, `clc` and `close` (or any of their variations) in any of your m-files, as they interfere with the grading process.**

2. **When your code is run, it should produce only what is asked for in the assignment. Please make sure to suppress any unnecessary intermediate result in any form: no other figure, curve or value printed in the command window should appear.**

   Specifically, when CP1.m is run, we should only see, in that order:

   - **Figure 1 with data and power law fit**
   - **Values asked for in the command window**

- **Figure 2 with data points and 3rd order fit only**
- **Correlation coefficient of 3rd order fit in command window**

- **Figure 2 with data points and 5th order fit only**
- **Correlation coefficient of 5th order fit in command window**

- **Figure 2 with data points and 7th order fit only**
- **Correlation coefficient of 7th order fit in command window**

- **Figure 3 with data and fit**
- **Values asked for in the command window**

You will lose points if you ignore these instructions.

Please note that Canvas will only accept a file with a ZIP extension: all of your M-files and the PDF report must be in that zipped folder. If we cannot open your files because you failed to follow these instructions, you will get a zero for this assignment.

You can upload your folder as often as you like, in case you find an error in your work. Older files will be kept in Canvas, but **only the last submitted folder will be graded**. It is your responsibility for this (and all other) electronic assignment to make sure we are grading the correct folder.

Finally, please keep in mind the policies about late assignments and collaboration outlined in the syllabus, and copied here for your convenience:

- Late assignments: homework and projects will be accepted up to 24 hours after the deadline with a 50% penalty. This policy will be strictly enforced and no submission will be accepted after 24 hours. Hard copies of homework will be due at the end of class. ***It is your responsibility to make sure that you have successfully uploaded and submitted all the required files in the required format to Canvas on time***: please double-check to avoid having to re-submit your work after the due date. The only acceptable proof that you have submitted your work on time is the time stamp of your Canvas submission. ***Files will not be accepted by email even if they show a "last modified date" that is before the due date.***

- Collaboration: students may *discuss* their assignments with each other, but homework and projects must be *completed individually* by each student. ***You must turn in your own work***. Copying someone else's work and submitting it as your own will not be tolerated. In particular, ***for Matlab assignments***, this policy means that students can discuss aspects such as the general approach to solve a problem or the syntax of a specific command, but ***they should not look at each other's codes***. If it is suspected that this has occurred, you will be reported to the Dean of Students for an honor code violation.