

Forced response and save to file

To do the exercises in this notebook, please download the wind file and save it as

`wind_12_ms_TI_0.1.txt`: [download file](#). You will also need the results file from the loading and plotting exercises.

It is now time to add the wind to our Turbie simulations. We modify our dynamical system so it now has forcing:

$$[M]\ddot{\bar{x}}(t) + [C]\dot{\bar{x}}(t) + [K]\bar{x}(t) = \bar{F}(t).$$

With our state vector equal to $\bar{y}(t) = [\bar{x}(t), \dot{\bar{x}}(t)]^T$, just as before, the state-space form is

$$\bar{y}'(t) = \begin{bmatrix} [0]_{N \times N} & [I]_{N \times N} \\ -[M]^{-1}[K] & -[M]^{-1}[C] \end{bmatrix} \bar{y}(t) + \begin{bmatrix} 0 \\ [M]^{-1}\bar{F}(t) \end{bmatrix}$$

where N is the number of degrees of freedom. For Turbie, $N = 2$. We can simplify this expression as

$$\bar{y}'(t) = [A]\bar{y}(t) + [B(t)]$$

Remember from the definition of Turbie that

$$f_{aero}(t) = \frac{1}{2} \rho C_T A [u(t) - \dot{x}_1(t)] |u(t) - \dot{x}_1|,$$

where C_T is constant for a given simulation and calculated from a look-up table using the mean wind speed of a simulation, $U = \overline{u(t)}$. The aerodynamic forcing occurs only on the blades, so the forcing vector for our 2-DOF system is

$$\bar{F}(t) = \begin{bmatrix} f_{aero}(t) \\ 0 \end{bmatrix}.$$

IMPORTANT!!!* The output of your `dydt` function must be a 1-dimensional array of length 4. In other words, if you test your `dydt` function and look at the shape of the output, you should get a shape of `(4,)`. **DO NOT** initialize your force vector as a 4x1 array, or your output will be the wrong size due to something called “broadcasting”.

Exercises for the reader

1. Create a function that takes as input the path to a wind file and returns the time array and the wind array. Try it on the wind file. What is the wind speed at $t = 1$ s?

2. Create a function that takes as input a wind array and returns the value for C_T to be used in a simulation. Test it on the wind file.
3. Explain how we can get the value of $\dot{x}_1(t)$ from vector $y(t)$?
4. Update your `dydt` function from the homogeneous response to include the aerodynamic forcing as defined above. What is the value of your new `dydt` function when $t = 1$ and $y = [1, 2, 3, 4]$? What is the shape of the output?
5. Use your numerical integrator, just like in the homogeneous response, to simulate Turbie's response with the following conditions.
 - Time vector spans from 0 to 660 in spaces of 0.01 s.
 - Turbie initial conditions are `y = [0, 0, 0,]`.
 - Use the wind time series in the response file you used for the "Loading and Plotting" and "PSD" exercises.
6. Compare the tower and blade deflection in the results file versus the deflections you just simulated. Do they match?
 - **Remember** the blade deflection in the file is *relative* deflection with respect to the tower!!!
7. Save the results to a text file in a format similar to the results file we give you.
 - **Python hint:** Use `np.savetxt` with `delimiter='\t'` and `fmt='%.3f'` to get a nice file. You should also pass in a header string that explains the columns.