

Comp3402 Wind Farms

C.B.Price December 2021

Purpose	(i) Study of the wake behind a single turbine (ii) Wake interactions for multiple turbines (iii) Optimization of turbine yaw, (iv) Optimization of turbine placement.
Files Required	Floris Python Code from the website.
ILO Contribution	LO 4
Send to Me	nix
Homework	Read chapter 4

Investigations

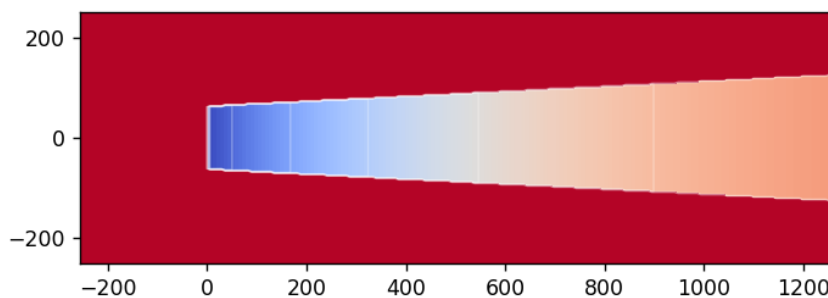
1 Installing and preparing to run.

- Unzip the **Floris_Release.zip** folder from the website to your desktop.
- Open a **command prompt** (type "command prompt" into the start menu). Change the directory to your unzipped folder (copy the folder path, then type **cd** at command then paste the path). You should see a load of **.py** programs and the batch file **setup.bat**.
- Double left click on **setup.bat**. This will prepare the software. It will (i) disable the **anaconda** default, install python and install the **Floris** application.
- Type the following to test things out **python single_wake.py**. You should be looking at the figure shown in the next task.

2 Investigate the single wake

- At command prompt type **python single_wake.py** which shows the wake around a single turbine placed at (0,0). You can see details about the model in the file **jensen.json** in the **jsons** folder.

At the bottom right of the window, your mouse location in the field is shown together with the wind speed in meters/second at that point (see green box).



x=-62. y=-17.
[8.00]

- Make a sweep from left to right through the centre of the wake ($y = 0$) and observe the change in wind speed. Theory predicts there will be a sudden drop as you cross the turbine disc, then the speed will recover (increase).

(c) You may want to run the Octave script **Calc_Jensen_single.m** which will plot the theoretical wind speed curve, and you can even collect some data and add it to the script as indicated.

(d) From two measurements, calculate the velocity deficit (also known as the ‘axial induction factor’) for the Jensen model

$$a = \frac{1}{2} \left(1 - \frac{v_R}{u} \right)$$

where u is the free velocity to the left of the turbine and v_R is the wind velocity *just to the right* of the turbine disc. Check your value with the value calculated in the above Octave file.

(e) Make a note of the power produced by this single turbine.

3 Two Turbines: Study1 - Axial Placement

Here we shall have a look at the powers produced by 2 turbines when we change the spacing between the turbines. The python script is **two_turbines.py** and the parameters for the system are in the json script **Jensen_2Inline.json** in the /jsons folder.

(a) Run the default json. The turbines have a separation of 1200m (1.2 km) and the wind direction is 270 degrees (from the West). Note down the powers of each turbine and the total power produced. Take a snip of the field so you can make a comparison soon.

(b) Edit the json script to place the second turbine at a separation of 2000m from the first. Again, note down the powers and take a snip of the field.

(c) Can you explain why the power has changed? Compare the two fields!

(d) You know that power is proportional to wind-speed cubed. So measure the speeds in front of both turbines, calculate the ratio and raise this to the power of three (e.g., in Octave type $(8/6)^3$ where I made up the speeds 8 and 6). This should be the same as the ratio of the powers Floris has given us for the turbines.

(e) If you like, make a mini-investigation changing the turbine separation, say in a range 750 to 2000m. Use Octave to plot a graph of power and separation. See for what separation change you get the largest improvement in power.

(f) Tie all of this up and make some conclusion about how to place wind turbines. Maximizing power is just one factor; placing turbines further apart will be more expensive – why?

4 Two Turbines: Study2 – “Wake Steering” - Changing the lead turbine yaw

Here the idea is that if you rotate the lead turbine so it is facing the wind not head-on, then the following turbine will experience less (or none) of the lead turbine’s wake. So, we expect an increase in power. Let’s investigate. The python script is **yaw2.py** which reads data from the json **example_input.json** though you will only need to change the **.py** script, here

```
49 # Specify yaw angles
50 yaw_angles = [20,0]
51
```

(a) Before you investigate think how the **total** power may change as you yaw the lead turbine. Sure, the second turbine may well produce more power, but what about the first.

(b) Conduct an investigation to see the effect of **yaw** on individual and total powers. Use Octave to make a plot of powers against yaw angle.

(c) Can you explain what you see in Simple English?

5 Two Turbines: Study3 – “Wake Steering” - Optimizing the Yaw

(a) Run the python script **yaw2_optimal.py** which has been written to find the optimal yaws of the two turbines. (Better check the separations are identical). See if the angles agree with your manual investigations.

(b) If you like, run the optimization for more turbines. The script **yaw3_optimal.py** is set up for three. You could easily extend this by adding to these lines of code. Note how the placing is specified as a multiple of blade diameter **D** which makes the layout scale with turbine size.

```
# Set turbine locations to 3 turbines in a row
D = fi.floris.farm.turbines[0].rotor_diameter
layout_x = [0, 7 * D, 14 * D]
layout_y = [0, 0, 0]
```

6 A column of 6 Turbines

The idea of this investigation is to move to a grid of wind turbines. The python script **columns.py** read the json **Jensen_6Inline.json** which is configured to have

- 6 turbines
- Separation of 1km
- Wind direction 270 degrees (from the west)

You are invited to perform one or more investigations on this system. Possible ideas are

- Stagger the turbines, moving alternative turbines the same amount in the y-direction to reduce wake overlap. How does total power depend on the amount of overlap
 - Change the wind direction and see what happens.
-

7 Optimization of a Wind Farm (2D)

Here you are invited to study optimization of a Wind Farm layout using Floris. A thorough study will take a lot of time, so you might like to have a brief look now, and if you are interested take this forward during the ‘Design-Build-Test’ Unit of work. Initial locations of the turbines will be moved iteratively to maximize the wind farm’s AEP (‘Annual Energy Production’)

Two scenarios have been created for you

- **optimize_layout.py** starts with three turbines given initial locations within a specified boundary. Things you could change in the **.py** script are (i) changing the boundary, (ii) changing the number of turbines.
- **optimize_layout_Knockcronal.py** which is based on the proposed *Statkraft* site in Scotland. Let's see if we can change their design to gather more energy. If you want to make changes, then this could be the shape and size of the boundary.

<https://projects.statkraft.co.uk/knockcronal>

<https://projects.statkraft.co.uk/knockcronal/widgets/38481/documents>

Thanks for support from the folk at