# EXERCISE SET 3: REAL-TIME WIND FARM INTERACTION WITH DEMAND AND STORAGE

JULIAN HAIMOVICH, JIEHUA CHEN AND MATT ZEBIAK
ASSIGNMENTS WEBSITE

The purpose of this problem set is to introduce the economics of wind farm planning via the design of a simple wind farm model that interacts with real-time demand and storage. You will analyze how different turbine organizations and numbers of turbines affect energy usage and costs for various systems. The turbines you will be using for this estimation are the same 1.5 MW turbines used in problem set 2, (height 80m, rotor diameter 67.2m) so the functions you previously developed for power $P(W)$ (power_func) and the power coefficient $C_p$ (p_coeff) will be useful. For your convenience, the given wind data for this problem set has been aggregated hourly, and corrected for height. There is no need to use the wind speed correction function, nor is there any need to aggregate the data as you did in exercise set 2. Refer to NY_wind for New York's wind speed data, 20loc_wind for the wind speed recorded at 20 locations, and nydemand.xls for New York State hourly demand data in Megawatts.

For this exercise set, you will be using pre-written functions that call functions you will be creating, therefore it is essential that you read the following descriptions of these functions carefully.

- Fig_cost_func is a function designed to output the costs and figures associated with different numbers of wind turbines. Note that it can be used for the single location, 20 location unoptimized, and 20 location optimized systems. The code is well-commented, so you will be able to see exactly what data each variable is associated with. The line used to call the function is as follows:

  ```
  [s_max, total_cost, t_g_per, s_i_per, n_s_per, t_per] =
  fig_cost_func(NY_power_MW, 20000,10000,50000);
  ```

  Note that NY_power_MW is the power generated by a single turbine experiencing wind speeds given by the NY_wind file. The next 3 numbers in the input represent a range of turbines in the form of initial:step:end. For this sample input, figures and costs will be calculated for 20000, 30000, 40000, and 50000 turbines.

- You will be responsible for designing the battery_func function that fig_cost_func calls on line 13. The call line is shown below:

  ```
  [storage_array, storage_input, storage_output, not_supplied] =
  battery_func(total_power, demand_data);
  ```

  Storage_array returns the hourly storage size, storage_input returns the total amount of wind turbine energy that went into storage during the year, storage_output returns the total amount of wind energy that was output by storage during the year, and not_supplied contains the total amount of energy demand that could not be supplied by wind turbine/storage energy and therefore had to come from a gas generator.

*Date*: June 28, 2012.

Battery_func should receive the hourly power created by the system, which is generated in line 10 of the fig_cost_func (note this is different from the power input into fig_cost_func because it is scaled by the number of turbines). Demand_data is an array containing the hourly electricity demand for NYC. When designing the battery algorithm, there are three situations that you must consider:

(1) Positive deficit (surplus) that can be placed into the battery.
(2) Negative deficit that the battery can supply.
(3) Negative deficit that the battery cannot completely supply, and must be supplied by a separate generator. (Note that in this situation, the battery is emptied of its remaining power).

For instance, if the deficit for hour 15 is positive (e.g. there is a surplus of energy), that extra energy should be placed in the battery. If, however, the deficit for hour 16 is negative, energy can be drawn from the battery to meet the difference. If the amount in storage cannot meet the difference between the wind turbine energy and the electricity demand for a given hour, then it will be not_supplied (supplied by a generator).

It is critical that you consider how to maintain proper accounting of what is going in and out of storage by thinking how each of those 3 situations affect the storage input/output. Note that storage_array is simply an array containing the storage size per hour.

- Convecopt.m is an optimization function that is included in the data package and will be used in the last part of this problem set. Note: In addition to copying all the necessary data files (NY_wind, 20loc_wind, nydemand) and pre-written functions (fig_cost_func, convecopt) into the MATLAB directory, you must also copy the cvx folder, open it in MATLAB, and run the script cvx_setup.m before convecopt.m can be used. Once the installation has finished, be sure to return to your MATLAB directory before beginning.

- The last script you will need to write for this exercise is HW3_main, which handles basic data input and calls other functions. It should be separated into 3 parts, that each correspond to problems 2, 3, and 4, respectively. This script will be responsible for passing data to the fig_cost_function, so the necessary power calculations will be done here.

(1) Plot a histogram of New York State's hourly electricity demand data. Label the axes and title the plot.
(2) Consider NY state's wind. Using HW3_main, generate an hourly power output based on your previously written function power_func for a single turbine (do not scale the result). Next, call the fig_cost_func as described above using a range of turbines from 20000:1000:50000. Plot the maximum battery capacity v. number of turbines. Next, generate annual figures for the performance of the turbine systems (in MWh) for a range of turbines from 20000:10000:50000 that include:
    - The maximum battery capacity necessary to contain all of the extra energy produced by the wind turbine system.
    - The percent of the total annual wind energy production that went directly from the turbine to the grid. (Hint: Find the total annual energy production by integrating the hourly estimations - use the trapz function - then find how much energy went directly to the grid by taking the difference between the total production and the battery input.)

- The percent of the total annual wind energy production that went into the battery.
- The percent of the total annual wind energy production that was left in the battery.
- The percent of the total energy demand that was supplied by a generator.
- The total cost of the system.

(3) Repeat part 2 (except for the battery v. turbine plot) for a turbine system comprised of 20 locations (file: 20loc_wind.xls), including every location in the data set. Again, put the results for 50,000, 40,000, 30,000, and 20,000 turbines on the same table. (Note that when using the fig_cost_func you should divide the number of turbines by 20 to find how many should be at a given location.)

(4) Repeat part 2 (except for the battery v. turbine plot) using all 20 locations again, but this time using an optimized distribution of wind turbines. Mathematically, the optimization problem can be stated as follows. There are $n$ candidates for windmill sites. Let $0 \leq x_j \leq 1$ be the proportion of budget allocated (or equivalently, total windmills) at the $j$th location, and let $A_{ij}$ be the wind power of the $j$th location at time $i$, where $i = 1, \ldots, m$ and $m = 8760$, the total number of hours in a year. For simplicity, we set the total budget to be 1, i.e. $\sum_{j=1}^{n} x_j = 1$.

Let $y_i = \sum_j A_{ij} x_j$ denote the sum of wind power at time $i$. Then the optimization problem becomes

$$\begin{aligned} \text{maximize} \quad & f(x_1, \ldots, x_n) = \sum_i y_i - \frac{\lambda}{m} \sum_i \left( y_i - \frac{1}{m} \sum_i y_i \right)^2 \\ \text{subject to} \quad & y_i > \tau, \quad i = 1, \ldots, m \\ & 0 \leq x_j \leq 1, \quad j = 1, \ldots, n \\ & \sum_j g(x_j) = B. \end{aligned}$$

We want to find $x_1, \ldots, x_n$ that maximizes $f(x_1, \ldots, x_n)$ and satisfies all the constraints. $g(x_j)$ is a known cost function for setting up a windmill at location $j$, and $B$ is the fiscal budget constraint. The second part of $f(x_1, \ldots, x_n)$ is the penalty term for the variance of wind power over time, where $\lambda$ controls the weight of the penalty term in the objective function. Optimization can be done as follows:

- Make sure that you have installed cvx by running the cvx_setup script. Note that you must be in the cvx folder to run the cvx_setup script, but the convecopt function does not necessarily need to be in that directory to be used. It should, in fact, be in your MATLAB working directory.
- Run the convecopt function for the wind power generated by a total of 20,000 turbines distributed optimally over the 20 locations. The syntax for the input is convecopt(powermatrix, $\lambda$, $y\_min$). The powermatrix term will be an array of size 8760 x 20 that contains the wind power generated at each of the 20 locations (Hint: you probably found this matrix at some step in part 3). Set $y\_min$ to 0 and run a for loop that varies $\lambda$ between 5 and 9. (Hint: the for loop should be located in HW3_main).
  - (a) Compare the maximum battery capacity needed, the percent of the total annual wind energy production that went to the grid, to the battery, remained in the battery, and was supplied by a generator. Also compare the standard deviation and total cost of each system, as well as the capacity of each (note that the number of turbines in use may change with $\lambda$). What can

you infer about how lambda changes the output of the optimized wind farms?

Put the usage and cost calculation for the optimization given by $\lambda = 9$ in the table alongside the figures for a single location and the non-optimized 20 locations for 20,000 turbines. Be sure to include the capacity for each system.

*E-mail address*: jc3288@columbia.edu, jsh2146@columbia.edu and msz2112@columbia.edu