Version 1.0 20 March 2023

EH2745 Computer Applications in Power Systems

Assignment I

Overview

The purpose of Assignment I is to let you combine the software techniques we have studied during the first part of the course, and using these create an embryo of Energy Management System. The assignment combines Python programming, CIM-XML modelling and parsing and finally model building using Pandapower.

This project assignment shall be solved individually, you can collaborate with friends and you can re-use code that you find online. But the code you hand in must be written by yourself, plagiarism is not OK. If you use code from others, you need to explain what the code can do in the screencast you hand in with your code.

Assignment

The assignment involves developing a Python application that fulfills at least the following requirements:

1. Should be able to read the CIM-XML files (EQ and SSH) of the **CIM-XML system** linked on the canvas assignment page.
2. Implement a graph traversal algorithm to identify the grid's topology and create a internal datastructure that represents a model of the grid. You are free to choose how to model the grid in this internal structure.
3. Gather data about the system from the internal datastrucutre that you create according to what pandapower requires to set up a network. This network should describe the properties of the conducting equipment in the grid.
4. Instantiate a pandapower network using the grid topology from #2 and grid parameters identified from #3 above.
5. Plot the grid structure using the pandapower tools.

Plotting can be done with the methods found in pandapower's plotting module, but you are free to use another approach. You are encouraged to look at their website and their documentation to find out how Pandapower works.

Website: https://www.pandapower.org/

Getting Started guide: https://www.pandapower.org/start/

API documentation: https://pandapower.readthedocs.io/en/v2.2.2/

Submission of solutions

The source code and screencast in which you present your solution shall be handed in no later than **the date published on Canvas**. You need to hand in two things:

1. A ZIP file of the GitHub repository containing your software. This code must be runnable, i.e. when the repository is unzipped and installed on another machine, no additional software installation shall be necessary. The GitHub repository shall contain a README file describing the included files.

2. A 10 minute long screencast/video where you explain how your code is structured, which functions you have used and provide a step by step guide through the program code.

Grading of assignment

The assignment is graded in three steps.

The grade Pass (**5 course points**) is awarded to a group that submits a Python application that fulfills the above requirements above. The equipment from the EQ and SSH files that must be at least included for the grade Pass are loads, generators, lines, transformers, buses.

For higher grades, **up to 7 bonus points** are awarded to a group based on the, the quality of the solution above the basic requirements. Below follows a number of examples that describe a high quality project.

Finally, **up to 3 grade points** can be awarded to students who during an individual oral presentation discussing the code and able to answer questions on implementation details. These individual slots are booked separately at course end, and are available to students who score >5 bonuspoints.

Examples of a high quality project follows below:

Good code style
Use the object oriented programming paradigm, separate code into modules and classes.

Consistent naming conventions.

Comments to supplement code.

Try to think logically and put code into function blocks.

Conforms to PEP8.

Implement a GUI
Add a GUI to your program that can be used for instance to:

- Select the xml-files to work with
- Select nodes to display information text about.
- Display information about the algorithm that is running.
- Allow user interaction to step through the algorithm
- Present statistics about the analysed grid

Additional features in the grid
Add breakers to the system and can handle opening and closing them.

Find something interesting in pandapower to do with the model and report the results.

Additional features in the algorithm
That the algorithm an handle other files than the basic example for the assignment. We have supplied several grid models. Try making the topology parser work on other models than the base case.

Late or missing hand-ins
Submissions after the deadline, or re-submissions after receiving comments on an initial submission before the deadline **can only be awarded 5 course points** in total.

References and plagiarism

Please note that when solving the assignments co-operation between students is allowed and even encouraged. However, you as an individual are responsible for the content of your own program and plagiarism will result in an immediate failing of the assignment in addition to a written report to KTH's central disciplinary committee. This means that all students should **write**

their own programs. You are not allowed to use source code from other groups and you are not allowed to copy source code from the internet.

You are however allowed to use code repositories, but your ability to explain this code as shown in your screencasts and possible individual presentations will be part of the assessment. This means that if you use a code library and cannot explain what the code does, you will not be graded as having passed the assignment, alternately you will not be awarded course points.