Generating the TESP R4-1247-1 feeder and running in UCEF

March 2022

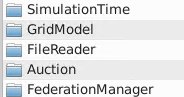
1. Running in UCEF
2. Generating a new feeder in TESP and modifying to run in UCEF.

# (1) Running R4-1247-1 feeder model in UCEF

Starting UCEF

* Start UCEF in Hyper-V Manager
* Remote to the 198.162.0.21 vagrant vagrant

File structure and config files

* /home/vagrant/ucef/transactive-energy/src/tesp/R4-12.47-1/
* Five UCEF components are under src/tesp/R4-12.47-1/
* 
* Also here are all the run.sh scripts for different prices and dates.
* Under /auction/conf is the agent\_dict.json (flat and rtp) that has all the hvac config, but also the average price and stdev. Also the auction.json that references the agent\_dict.json
* Under /FileReader/conf is the filereader.json and the lmp.csv, tou.csv, flat.csv price files.
  + Here is the lmp-July6.csv and lmp-July7.csv.
* The model.glm file is in src/tesp/R4-12.47-1/GridModel/model/
  + Whatever glm file I am running gets put as “model.glm” in this subdirectory.
  + Put the actual name of the glm as the first line in the glm (after “//”)
* Outputs are written to src/tesp/R4-12.47-1/GridModel/model/

Running

* at vagrant@vagrant:~/ucef/transactive-energy/src/tesp/R4-12.47-1/
* ./run-rtp-july67.sh, or ./run-flat-july67.sh

Processing data

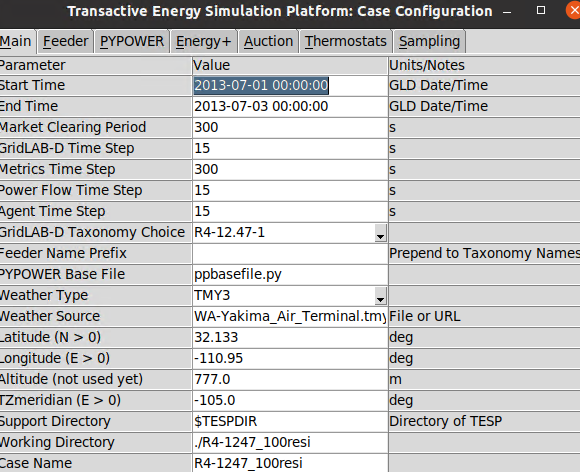
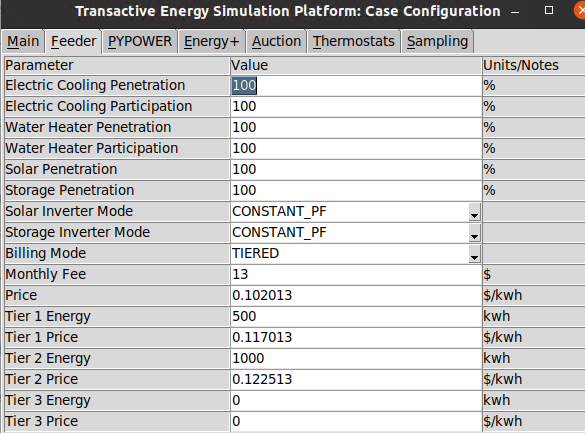
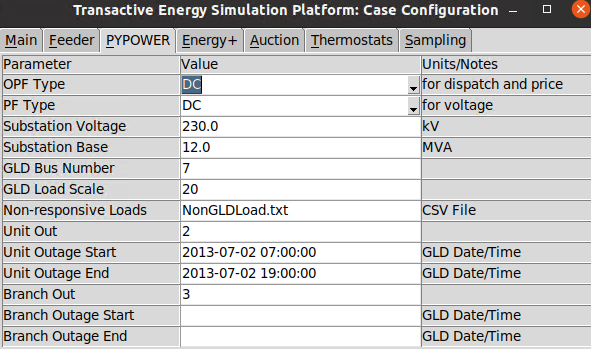
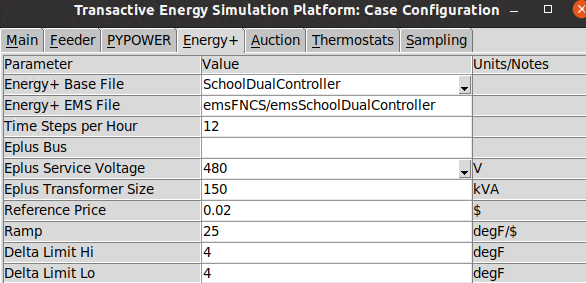
* Instructions here are for the Python scripts for extracting metrics writer data to csv, and plotting script from TomMcD (from <https://github.com/pnnl/tesp/tree/master/examples/ieee8500>): In order to plot results from the JSON files, Python 3 and the matplotlib package can be used:
  + "python3 glm\_dict.py model" will create circuit metadata for plotting.
  + The process\_gld.py model &" failed (was used for that 8 plots on a page together looking at metrics. Plotting failed. Must be different arrangement of data in .json files or whatever. Will not use it for now.
  + Did not try to use the "python3 process\_voltages.py model &" to plot all meter voltages on the same graph.
* Run in VM at: /home/vagrant/ucef/transactive-energy/src/tesp/R4-12.47-1/Gridmodel/model/ where the .json output files and model.glm file are located.
* Prices from auction to GLD are contained in R4-12.47-1/Auction/auction.csv.
* All the data is contained in the metrics\_writer.json files
* Process that data while still on virtual drive
* Still in /Gridmodel/model where the data is, execute at the prompt:
  + python3 json2csv.py model
* this generates 50 files with all the data
* if I want to take data from the PC back to the virtual drive, then use WinSCP on the desktop.

For additional info on GLD and updating VM, see the file /UCEF TE Experiment/Running the UCEF TE with 8500.docx

## Generating TESP Feeder Model

First use Hyper-V to start the TESP VM and then connect to it from Hyper-V (not remote in like for UCEF). pw:vagrant to log in.

From the “GLM Update” email thread with Trevor:

* Generating the feeder
  + cd ~/tesp/repository/tesp/examples/capabilities/feeder\_generator
  + don’t forget to run source ~tespEnv
  + ./config\_feeder.py - Launches a GUI to build a co-simulation. You NEED to change the following in the “Main” tab.
    - “GridLAB-D Taxonomy Choice”: R4-12.47-1
    - “Feeder Name Prefix”: (blank)
    - In the Energy+ tab, set “Eplus Bus” to an empty string so that the Energy+ stuff doesn’t get generated
    - “Support Directory”: $TESPDIR
    - “Working Directory”: ./<Case Name>
    - “Case Name”: whatever you want
      * Use ./R4-1247\_8500comp
      * 100 PV, 100 HP, 50 HWH, but no batteries.
        + For the R4 model there is PV only on resi (this config script doesn’t put any PV on commercial, that is, on BIGBOX or STRIP\_MALL). This is approx same as the 8500 case which was PV on 90% of the houses (and now on all resi, but not all buildings, that is, not commercial buildings). This is as close as I can come to the 8500 case which had 90% of resi with PV.
        + 100% HP. Even the commercial house objects will have HPs, but only those in houses will participate (per Tom’s config)—I still put 100 for participating in the config script and then it gets modified by Tom’s script I think.
        + 50% HWH is the same. This includes 50% of commercial house objects. But none of them participating in auction (like for 8500, only the HPs are price responsive; HWHs follow the schedule for each house).
  + Save Config – Pick whatever name you want, use a “json” extension, and put it in the same directory as the config\_feeder.py script (in feeder\_generator where it goes automatically)
  + ./populate\_feeder.py <name of config file you just created>
  + Glm is generated in the working directory you just created (e.g., R4-1247\_8500comp) and use WinSCP to pull that out of the TESP VM back to the Experiment Runs folder in my UCEF TE Experiments directory.
* Here are images from the config\_feeder.py GUI
  + 
  +  this should be 100, 100, 50, 0, 100, 0
  + I could include batteries at 100% for future use, but I don’t see “storage participation” so maybe that means there automatically all participating. In the 8500 grid there was a switch in the glm to turn battery participation on.
  + I should create a “R4-1247\_8500comp-bat”
  + 
  + These are defaults, but what is the “unit outage” here? Shouldn’t matter.
  + Eplus Bus set to “”
  + I ignored “auction”, “thermostat” and “sampling” tabs in the config GUI. I think we ignore the auction settings and don’t use it. Thermostat tab provided ranges for various temps. Not sure about Sampling. None of the should matter since Tom’s settings override.

After this, a quick edit:

Change “module powerflow{” near top of glm to “module powerflow {” (add space)

Then go into UCEF VM:

* Start UCEF VM (vagrant, vagrant) and then remote in.
* Use WinSCF to copy the glm into src/tesp/tesp\_support directory

# Scripts

From /home/vagrant/ucef/transactive-energy/src/tesp/tesp\_support/

Run the following command to reset the cooling set point / air temperatures after changes to the model:

python3 update\_glm.py <glm\_file> R4\_12\_47\_1\_initial\_data.csv

* This generates a new glm <my file.glm.modified>
* Rename <my file.glm.modified> to <my file-m.glm>

(note: the new R4-1247-1 model with new set of houses – the auction config files have been generated using Toms’ “bash generate\_agent\_dict.sh <my\_file-m\_without\_glm\_extension>“—doesn’t need to be run again for this model (PV, batteries, etc. don’t change houses))

* Files for Farhad also in tesp/testsupport:
  + R4-1247\_8500-m\_house\_data.csv
  + R4\_12\_47\_1\_initial\_data.csv

After that, make the following additional edits in Notepad++

# Required R4-12.47-1 Model Changes

In “object metrics\_collector\_writer”

Comment out filename ${METRICS\_FILE}; (line 27)

Uncomment filename <case name>\_metrics.json

Change the clock value (this only matters for Farhad’s run; doesn’t matter for running in UCEF)

```

clock {

timezone MST7;

starttime '2017-06-23 0:00:00';

stoptime '2017-07-08 0:00:00';

}

```

Add a CSV reader (before the climate object):

```

object csv\_reader {

name CsvReader;

filename "weather/tucson\_minutely.csv";

};

```

Update the climate object:

```

object climate {

reader CsvReader;

tmyfile "weather/tucson\_minutely.csv";

//interpolate QUADRATIC;

};

```

Add group recorders at the end

//recorders

object group\_recorder {

group "class=inverter";

property VA\_Out;

file outputs/inv\_power.csv;

interval 60; // 5 minutes

};

object group\_recorder {

group "class=inverter";

property VA\_Out;

complex\_part REAL; //IMAG, MAG, ANG\_DEG, ANG\_RAD

file outputs/inv\_real\_power.csv;

interval 60; // 5 minutes

};

object group\_recorder {

group "class=house";

property cooling\_setpoint;

interval 60;

file outputs/cooling\_setpoint.csv;

};

object group\_recorder {

group "class=house";

property air\_temperature;

interval 60;

file outputs/air\_temperature.csv;

};

object group\_recorder {

group "class=house";

property mass\_temperature;

interval 60;

file outputs/mass\_temperature.csv;

};

object group\_recorder {

group "class=waterheater";

property actual\_load;

interval 60;

file outputs/hwh\_load.csv;

};

object group\_recorder {

group "class=house";

property hvac\_load;

interval 60;

file outputs/hvac\_load.csv;

};

object group\_recorder {

group "class=triplex\_meter";

property measured\_real\_energy;

interval 60;

file outputs/hse\_real\_energy.csv;

};

object group\_recorder {

group "class=triplex\_meter";

property measured\_real\_power;

interval 60;

file outputs/hse\_real\_power.csv;

};

object group\_recorder {

group "class=meter";

property measured\_real\_energy;

interval 60;

file outputs/commercial\_real\_energy.csv;

};

The above manual changes can be made either before or after running the update\_glm script; the resulting updated and edited file should be renamed as “model.glm” and replace the file in src/tesp/R4-12.47-1/GridModel/model/ (and this file will then be used automatically by the run scripts).