PowerLab Notebook

Midrar Adham

last updated: May 28, 2021

Contents

List of Errors

1. convergence iteration limit reached for object node:34
   - This error mostly refers to transformers ratio (downstream). Make sure the the primary and secondary voltages satisfy the transformer configuration object.
2. Unable to load player file.
   - Make sure player file in the same directory as the glm file.
3. unable to set current process affinity mask, err code 5
   - This is a common error for GridLAB-D version ¡ 3.2. Write gridlabd –version in command window. If it shows any version that is less than 3.8, update your version from sourceforge.com or use GitHub source code.
4. expected ';' at end of property specification
   - This is a general error. If your file has ";" at the end of each property, then this refers to spaces between numerical values. For example, assuming you defined a voltage value for a transformer object as "120 + 0j;", GridLAB-D will produce the above error. The above value should be written as "120+0j;" without any spaces.
5. : sync_player:11: not enough values on line
   - Each property in player object must be indented. If you add a property in any object, make sure you press "tab" on your keyboard instead of adding spaces.
6. sync_node(obj=6;mds_primary_node_SwingNode): All nodes were not properly populated
   - First of all, make sure you add a "name" property to any object in your glm file. Thus, if there is an error in your file, GridLAB-D will tell you the name of the object that is causing the error. Second, to resolve the above error, make sure the indicated node is linked to another object using a link-based object (line object, link object, triplex line object, ..). Nodes in glm file SHALL not be left hanging.
7. ERROR [INIT] : init_transformer(obj=4;tranformer1): transformer:4 (split phase) - tranformer1 has a phase mismatch at one or both ends
   - SINGLE_PHASE_CENTER_TAPPED transformers connects two nodes. The primary node SHALL be a 3ϕ node and the secondary node SHALL be a 1ϕ node. Also, in "phases" property, make sure you add the phase name and the letter "S". For example, for a phase A triplex transformer, "phases" property SHALL be defined as "phases AS;".
8. Newton-Raphson method is unable to converge to a solution at this operation point
   - NR does not work with radial systems. A possible solution to get NR working with radial systems is to set a reference node. That being said, the best option that does not lead to other errors is to use FBS with radial systems and NR with loop systems.

List of Warnings

1. switch object may not behave properly under FBS!

    - This warning is a bit misleading and might cause serious issues in the future. FBS is mostly used with strictly radial topologies. Beside the fact that your system ”should” be radial or meshed, this also means that your link-based objects should use ”from” and ”to” properties to connect objects with each other. If you’re using NR solver instead of FBS, ”from” and ”to” properties can be neglected. Instead, you can simply use ”line” objects to connect nodes with each other.

Fall 2020

Dec 19, 2020 – Water heater object GLD

OBJECTIVE:

Test water heater behavior in GridLAB-D.

OUTLINE:

Using GridLAB-D, a water heater behavior is tested using the following parameter: **PROCEDURE:**

To achieve the goal of this sprint, IEEE_4_Node_Feeder is used. The following objects are needed:

- Triplex objects such as transformers, lines, meter, and water heater.
- Water heater parent. Typically a house object.
- Water heater object.

PARAMETERS:

- Setpoint 120F
- Deadband 2F
- Volume 50 Gallons
- Water demand ELCAP data
- heat_mode ELECTRIC

DATA:

glm file can be found here

https://github.com/psu-powerlab/GridLab-D/blob/master/NeoChargeProject/WH_4_Node_Feeder/Uncontrolled_WH/WH_4_node.glm

Full output data is uploaded to PSU power lab GitHub account, GridLAB-D repository.

https://github.com/MidrarAdham/GridLab-D/blob/master/NeoChargeProject/WH_4_Node_feeder/Water_heater/wh_1.csv

RESULTS:

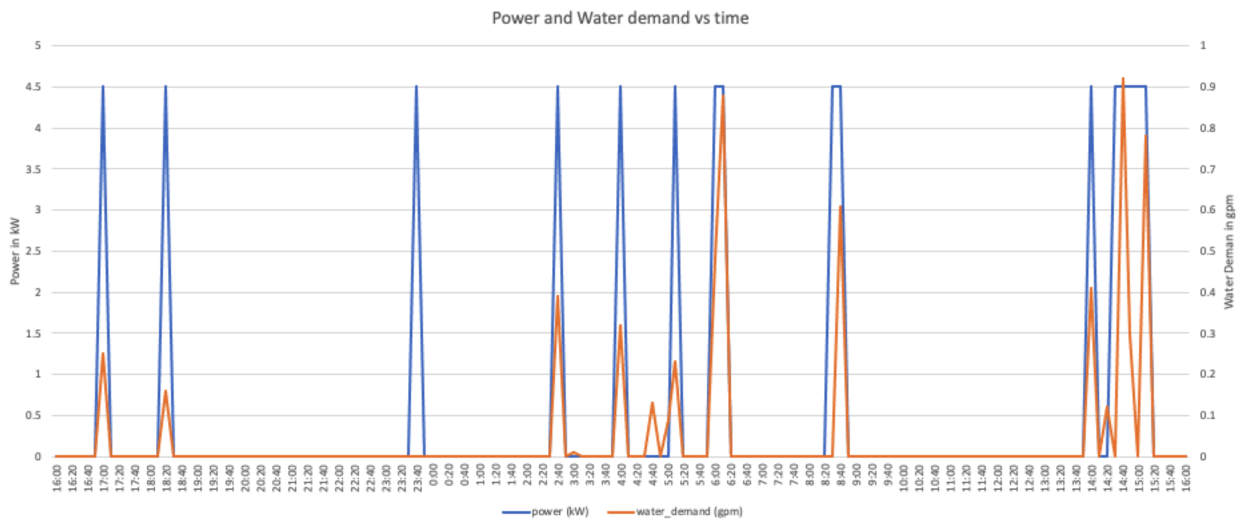

Figure 1: Power and Water Demand vs Time (un-controlled)

Winter 2021

Jan 06, 2021 – Control Water heater using switch and passive controller in IEEE 4 node feeder

OBJECTIVE:

- Test behavior of water heater when controlled by switch object using NR and FBS solvers.
- Test water heater behavior when controlled by passive controller.
- Test water heater behavior when shed command is received.

OUTLINE:

What steps are required?

1. Switch Object
    - Set up a switch object in 4 node feeder.
    - Use player object to control the state of the switch (OPEN OR CLOSED).
    - Define player object timestamp to be compatible with CLOCK object.
2. Passive Controller:

    Passive controller utilizes energy market. When prices are high, water heater turns OFF. When prices are low, water heater turns ON.

    - Set up auction object.
    - Set up passive controller.
    - Set up water heater object as passive controller child.
    - Set up a .player file so auction object can read from it. (Alternative solution: Prices can be scheduled using schedule object.)
3. Shed Command:
    - Change water heater setpoints during simulation to simulate shed command.

PROCEDURE:

1. Switch Object
    - switch object is placed in the triplex section of the feeder (between center tapped transformer and triplex node).
    - Remember, switch object SHALL be placed between link-based nodes.
    - Switch object SHALL be used in INDIVIDUAL mode. It won't work with BANKED mode.
    - Use NR solver. Switch object may behave incorrectly with FBS solver.
2. Passive Controller:
    - Import market module
    - Set up auction object with prices source file.
    - Set up a player object that contains prices data. This object is auction object child.
    - Set up
3. Shed command:
    - Using schedule object, setpoints are scheduled every 10 minutes.

- The water temperature SHALL decrease below the original setpoints.

PARAMETERS:

1. Water Heater parameter (without Shed command):
    - Setpoint 120F
    - Deadband 2F
    - Volume 50 Gallons
    - Water demand ELCAP data
    - heat_mode ELECTRIC
2. Switch object state:
    - At 4:00 pm, switch is CLOSED until 6:00 pm.
    - Switch state changes to OPEN from 6:05 pm until 8:00 pm.
    - Switch state changes to CLOSED from 8:05 pm until the end of the simulation.
3. passive_controller:
    - period 600 seconds. (This property SHALL match simulation time)
    - Control_mode PROBABILITY_OFF. (SHALL be used when der is aggregated.)
    - comfort_level SHALL be set to a high number to force water heater to turn OFF at specified times.
    - state_ property SHALL be override. This is important to force water heater object to stick to parent object parameter.

OBSERVATIONS:

1. Switch object:
    - Water heater did not respond to switch changes with NR solver.
    - When switch is open, water heater still turns ON and consume power (kW).
2. passive_controller:
    - water heater behaves as expected.
3. shed_command
    - Setpoints changed as expected.

DATA:

1. Switch_object:

Timestamp	power (kW)	water_demand (gpm)	is_waterheater_on
2020-01-01 18:00:00 PST	+0	+0	0
2020-01-01 18:10:00 PST	0	0.16	0
2020-01-01 18:20:00 PST	4.5	0	1

Table 1: Water heater controlled by a switch

2. passive_controller
   glm file can be found here: `https://github.com/psu-powerlab/GridLab-D/blob/master/NeoChargeProject/WH_4_Node_Feeder/Controlled_WH/Controlled_WH_4.glm`

   Full output file is uploaded to power lab github account: `https://github.com/psu-powerlab/GridLab-D/blob/master/NeoChargeProject/WH_4_Node_Feeder/Controlled_WH/wh_1.csv`

Timestamp	power (kW)	water_demand (gpm)	is_waterheater_on
2020-01-01 16:00:00 PST	+0	+0	0
2020-01-01 16:10:00 PST	+0	+0	0
2020-01-01 16:20:00 PST	+0	+0	0
2020-01-01 16:30:00 PST	+0	+0	0
2020-01-01 16:40:00 PST	+0	+0	0
2020-01-01 16:50:00 PST	+0	+0.25	0

Table 2: Water heater controlled by passive controller

3. shed_command
   glm file can be found here `https://github.com/psu-powerlab/GridLab-D/blob/master/NeoChargeProject/WH_4_Node_Feeder/Controlled_WH/WH_Shed_command.glm`

   Full data is uploaded to PSU power lab GitHub account `https://github.com/psu-powerlab/GridLab-D/blob/master/NeoChargeProject/WH_4_Node_Feeder/Controlled_WH/wh_shed.csv`

   Does the shed command contain starting and ending time?

Timestamp	power (kW)	water_demand (gpm)	water_temperature (F)	is_waterheater_on
2020-01-01 18:00:00 PST	0	0	119.008	0
2020-01-01 18:10:00 PST	0	0.16	118.944	0
2020-01-01 18:20:00 PST	0	0	117.026	0
2020-01-01 18:30:00 PST	0	0	116.965	0
2020-01-01 18:40:00 PST	0	0	116.904	0
2020-01-01 18:50:00 PST	0	0	116.843	0

Table 3: Water heater controlled with shed command

RESULTS:

1. switch_object

   The above table **??** is a portion of the water heater output file. At the specified timestamps, the switch is open. It can be seen from the last row that the water heater was turned ON and consumed 4.5 kW even though switch was open. I sent a request to GridLAB-D folks regarding this issue. I will resume this work on switch object once I receive a response. Alternatively, a passive controller object was used. The results are shown in table **??**.

2. passive_controller

   A shed command is received at 17:00. The water heater is supposed to turn ON at 18:20 as the water temperature drops below the range (118F). Due to shed command, the water temperature continues to drop as shown in table **??**.

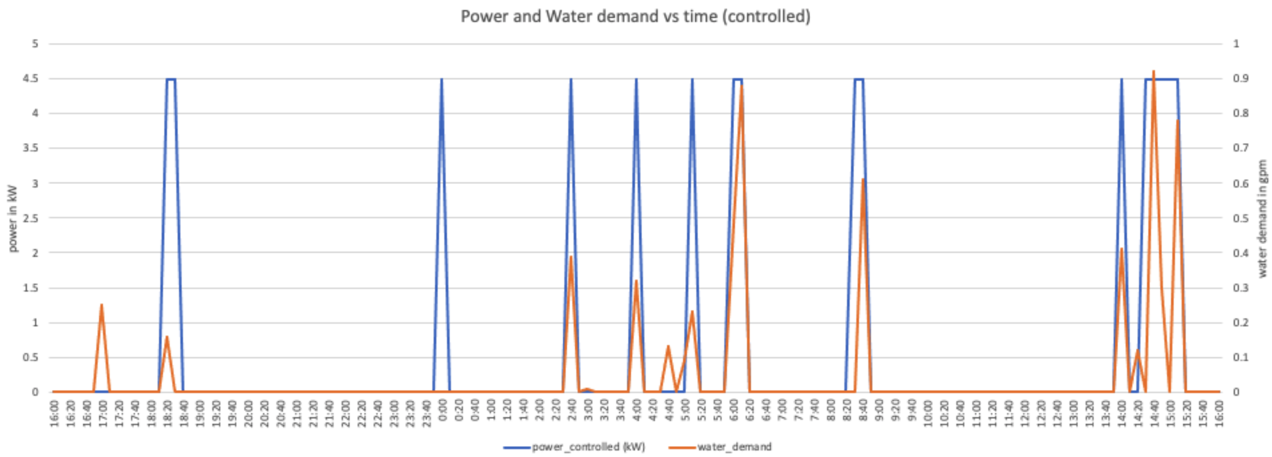

Figure 2: Power and Water Demand vs Time (controlled)

3. shed_command Refer to powerLab github account for glm file and procedure in this link Shed Command in EWH.

Spring 2021

Apr 10, 2021 – Control Heat Pump Water heater using switch and passive controller in IEEE 4 node feeder

OBJECTIVE:

- Test behavior of HPWH.
- Test HPWH behavior when controlled by passive controller.

OUTLINE:

What steps are required?

1. Build a WH object with HEAT_PUMP specified as a heat_mode.
2. Passive Controller:

   Passive controller utilizes energy market. When prices are high, water heater turns OFF. When prices are low, water heater turns ON.

   - Set up auction object.
   - Set up passive controller.
   - Set up water heater object as passive controller child.
   - Set up a .player file so auction object can read from it. (Alternative solution: Prices can be scheduled using schedule object.)
3. Shed Command:
   - Change water heater setpoints during simulation to simulate shed command.

PROCEDURE:

1. HP_WH
   - HP_WH is linked to a house object (Required) as a child.
   - Need to specify the parent in the WH object. (parent House1;)
2. Passive Controller:
   - Import market module
   - Set up auction object with prices source file.
   - Set up a player object that contains prices data. This object is auction object's child.
3. Shed command:
   - Using schedule object, setpoints are scheduled every 10 minutes.
   - The water temperature SHALL decrease below the original setpoints.

PARAMETERS:

1. Water Heater parameter (without Shed command):
   - Setpoint 120F
   - Deadband 2F
   - Volume 50 Gallons
   - Water demand ELCAP data

- heat_mode ELECTRIC

2. Switch object state:
   - At 4:00 pm, switch is CLOSED until 6:00 pm.
   - Switch state changes to OPEN from 6:05 pm until 8:00 pm.
   - Switch state changes to CLOSED from 8:05 pm until the end of the simulation.
3. passive_controller:
   - period 600 seconds. (This property SHALL match simulation time)
   - Control_mode PROBABILITY_OFF. (SHALL be used when der is aggregated.)
   - comfort_level SHALL be set to a high number to force water heater to turn OFF at specified times.
   - state_ property SHALL be override. This is important to force water heater object to stick to parent object parameter.

OBSERVATIONS:

1. HP_WH:
   - Water temperature increases above the setpoint. HP_WH object does **NOT** respond to setpoints as shown in figure **??**.

1	# timestamp	height	tank_setpoint	waterheater_model	power.real	constant_power	constant_current	water_demand	temperature	is_waterheater_on
2	2019-12-31 16:00:00 PST	+3.782	+119	ONEZNODE	+0.951456	+0+0j	+0+0j	+0	+119.977	+1
3	2019-12-31 16:10:00 PST	+3.782	+119	TWONODE	+0	+0+0j	+0+0j	+0	+123.509	+0
4	2019-12-31 16:20:00 PST	+3.782	+119	TWONODE	+0	+0+0j	+0+0j	+0	+126.963	+0
5	2019-12-31 16:30:00 PST	+3.782	+119	TWONODE	+0	+0+0j	+0+0j	+0	+130.333	+0
6	2019-12-31 16:40:00 PST	+3.782	+119	TWONODE	+0	+0+0j	+0+0j	+0	+133.616	+0

Figure 3: HPWH object in GLD does not respond to specified setpoints

- Comparing the HPWH behavior to the EWH, we can see the issue clearly. Figure **??** shows the EWH behavior under the same parameters.

1	# timestamp	height	tank_setpoint	waterheater_model	power.real	constant_power	constant_current	water_demand	temperature	is_waterheater_on
2	2019-12-31 16:00:00 PST	+3.782	+119	ONEZNODE	+0	+0+0j	+0+0j	+0	+119.977	+0
3	2019-12-31 16:10:00 PST	+3.782	+119	ONEZNODE	+0	+0+0j	+0+0j	+0	+119.911	+0
4	2019-12-31 16:20:00 PST	+3.782	+119	ONEZNODE	+0	+0+0j	+0+0j	+0	+119.844	+0
5	2019-12-31 16:30:00 PST	+3.782	+119	ONEZNODE	+0	+0+0j	+0+0j	+0	+119.778	+0
6	2019-12-31 16:40:00 PST	+3.782	+119	ONEZNODE	+0	+0+0j	+0+0j	+0	+119.712	+0

Figure 4: EWH object in GLD responds properly to specified setpoints

Debugging:

Related links

- Here's my conversation with Frank Tuffner, a GridLAB-D developer, regarding the HPWH. Frank's input regarding HPWH issue.
- On your computer, go to GLD folder > residential > open waterheater.cpp file.

Water Heater Dynamic Driving Parameters

- Demand
  - The higher the demand, the more quickly the thermocline drops.
- Voltage
  - The line voltage of the coil. The lower the voltage, the more slowly the thermocline rises.
- Inlet water temperature
  - The lower the inlet water temperature, the more heat needed to raise the temperature to the setpoint.
- Indoor air temperature
  - The higher the indoor temperature, the less heat loss through the jacket.

Heating Element Capacity

The heating element capacity equation in the EWH is voltage dependant as shown in equation **??**.

$$test = HeatingElementCapacity * (ActualVoltage)^2/(NominalVoltage)^2 \tag{1}$$

However, the heating element capacity for the heat pump water heater does not have a voltage dependence as shown in equation **??**.

$$HeatingElementCapacity = (1.09+(1.17-1.09)*(get_Tambient(location)-50)/(70-50))*(0.379+0.00364*Tw) \tag{2}$$

Commented commands by GLD folks

- Heating element capacity (line 1634)
- Water temperature increment for onenode and twonode analysis (lines 1656 and 1684)
- Coesfficient of Performance (CoP) line 1731

Water Heater Source Code Structure

- The code is defined by parameters instead of water heater models.
- Some parameters, such as tank_area, tank_volume, tank_height, etc are global as they work with all water heater models (i.e Electric, heat pump, and gas).
- Other parameters, such as heating element, need to be calculated when using Heat pump water heater model. The heating element in heat pump water heater is used as a backup.

Errors Summary

Running the HPWH object in GLD, we see the following errors:

- The property is_waterheater_on is randomly 1 or 0. For a correct HPWH behavior, it should be 1 when there's sufficient water demand. Otherwise, it should always be zero.
- The waterheater_model property should be ONEZNODE when there is no water demand. When there is water demand, there is inlet water pumped inside the tank. Therefore, both heating element capacity (top and bottom) should turn on. When both heating element capacity are on, the model switches to TWONODE model which is not the case in the HPWH. Refer to figure **??** and figure **??** for a visual analysis.

Questions

- I know the heating element is used as a backup in the HPWH. How is "backup" defined? Is it used where there's a high water demand? How high should the water demand be to turn on the heating element?
  - There are four modes in the A. O Smith units [**?**]. These modes are listed as shown below:
    - **Hybrid Mode:**
      - This mode uses the dead-band algorithm. If the average tank temperature (the weighted temperature of the upper and lower thermostat) drops below 9F below the setpoint, then the HP turns on to heat the water.
      - If the HP fails to heat the water to the setpoint (i.e due to high water demand.) and the average temperature drops more than 20F below the setpoint, then the upper heating element replaces the HP as the heating source.
      - The unit uses the HP until 75% of the available hot water has been depleted.
    - **Efficiency Mode**
      - This mode does not use the electric resistance elements, unless the ambient temperature is outside the safe operating range (45°–109°F) of the heat pump.
    - **EWH Mode**
      - HPWH acts as EWH. Upper element turns ON first to heat the top of the tank and then lower element turns on to heat the bottom of the tank.
    - **Vacation Mode**
      - Reduce the temperature setpoint (default is 60F)

.1 Heating Element Operation Principle in HPWH

The heating element operates under the following circumstances:

- If the air temperature is outside the safe range (45 - 120F)
- If the water in the tank is significantly lower than the set point, the upper element operates. The difference between the tank temperature and the set point depends on the circumstances, but it is generally 25°–30°F.
- If the system senses that the water use is too high, the lower element operates. In general, 25–30 gal within a short time period is considered high water use. Once the lower electric resistance element engages, the entire tank is reheated like a traditional ERWH.

How does a Heat Pump Water-Heater work?

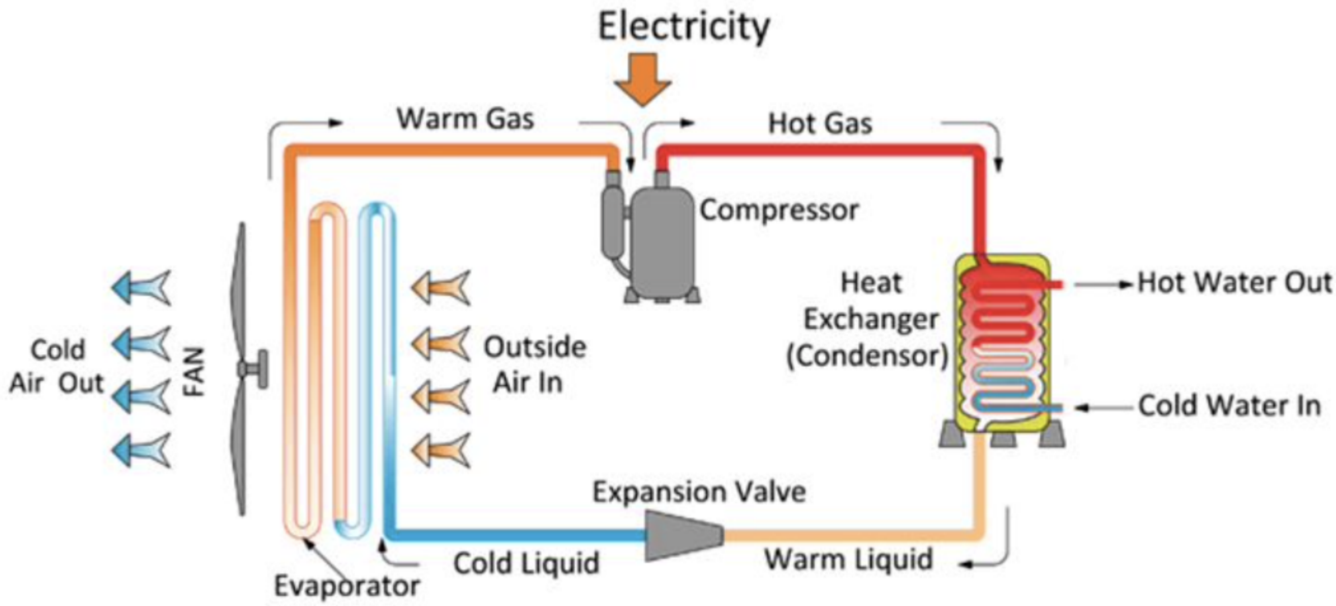

Figure 5: source:nukiengineering.com

I will start with a basic, however, very important thermodynamic principle. **HEAT ALWAYS GOES TO COLD.** Compressor increases the pressure of the gas passing through to make its temperature very high. This coil goes through the tank which heats up the water inside the tank (top portion). The heat of the gas gets released as it goes down the tank. The output of the tank is the same gas but with less hot temperature. The liquid goes through an expansion valve. The expansion valve "release" the liquid (less pressure therefore less heat) to make the liquid temperature less hot (cold). When the liquid goes through the evaporator, the temperature of the liquid is way less than the outside temperature. Therefore, the cold air gets dumped out and heat comes in. Thus, warm gas goes in to the compressor again and the same cycle is repeated.

This subsection is created in python. Let's see if it actually works.

Something

May 12, 2021 – EMCB Use Cases

OBJECTIVE:

Studying the use cases provided by PGE.

1. Electric Vehicle Managed Charging.

The above use case require EV breakers. PGE might receive EV breakers before the summer of 2021. Once received, we can install the EV breakers and connect them to the NHR-9430 (Load Simulator).