Power Lab Notebook

Authors:
Midrar Adham
September 11, 2021

Maseeh College of Engineering and Computer Science
PORTLAND STATE UNIVERSITY

List of Figures

1 Power and Water Demand vs Time (un-controlled) 5
2 Power and Water Demand vs Time (controlled) 12
3 HPWH object in GLD does not respond to specified setpoints 15
4 EWH object in GLD responds properly to specified setpoints 16
5 source:nukiengineering.com 20
6 Heat Pump Water Heater (HPWH) COP: 20 Gallon Water Draw . . 24
7 HPWH COP vs Time: Vacation Mode Recovery 30
8 HPWH COP vs EnergyTake: Vacation Mode Recovery 31
9 CTA-2045 EnergyTake Definition 39
10 HPWH COP vs EnergyTake: Vacation Mode Recovery 41
11 HPWH COP vs EnergyTake: Vacation Mode Recovery 43
12 HPWH COP vs EnergyTake: Vacation Mode Recovery 44
13 HPWH EnergyTake (top) and Energy_in (bottom) line fitting 49
14 HPWH COP vs EnergyTake: Vacation Mode Recovery 50
15 HPWH Watts vs EnergyTake: Vacation Mode Recovery 52
16 HPWH Watts vs EnergyTake: Vacation Mode Recovery 53
17 HPWH EnergyTake vs Time: Vacation Mode Recovery 54
18 HPWH Watts vs EnergyTake: Validation with Energy Management Circuit Breaker (EMCB) Data 55
19 HPWH Idle losses: Validation with EMCB Data 57
20 GLD and Physical Model Validation: Water Draw Validation 61
21 GLD and Physical Model Validation: Idle Losses 62
22 GLD and Physical Model Validation: Heating Sources Switching . . . 63
23 Water Demand Sample 64
24 65
25 66
26 67
27 68
28 69
29 70

30 . . . 71
31 . . . 72
32 . . . 73
33 . . . 74
34 . . . 75
35 . . . 76
36 . . . 77
37 . . . 78
38 . . . 79
39 . . . 80
40 . . . 81
41 . . . 82
42 . . . 83
43 . . . 84
44 . . . 85
45 . . . 86
46 . . . 87
47 . . . 88
48 . . . 89
49 . . . 90
50 . . . 91
51 . . . 92
52 . . . 93
53 . . . 94
54 . . . 95
55 . . . 96
56 . . . 97
57 . . . 98
58 . . . 99
59 . . . 100
60 . . . 101
61 GridAPPS-D Model Conversion . . . 102

62 File Conversion Processing Chart 103

List of Tables

1 Water heater controlled by a switch 10
2 Water heater controlled by passive controller 10
3 Water heater controlled with shed command 11
6 HPWH Data Used For Calculating COP in this Section 40
7 Heating Element for HPWH 42
8 EnergyTake Difference Calculated 45
9 Recorded and Calculated HPWH Porperties 46
10 Recorded and Calculated HPWH Porperties 47
11 Temperature Comparison 60

1 Fall 2020

Water heater object GLD Dec 19, 2020

1.1 objective

Test water heater behavior in GridLAB-D.

1.2 outline

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

1.3 procedures

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.

1.4 parameters

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

1.5 Data

glm file can be found on GitHub. Further, the full output data is uploaded to PSU power lab GitHub account, GridLAB-D repository PSU power lab GitHub account, GridLAB-D repository

1.6 results

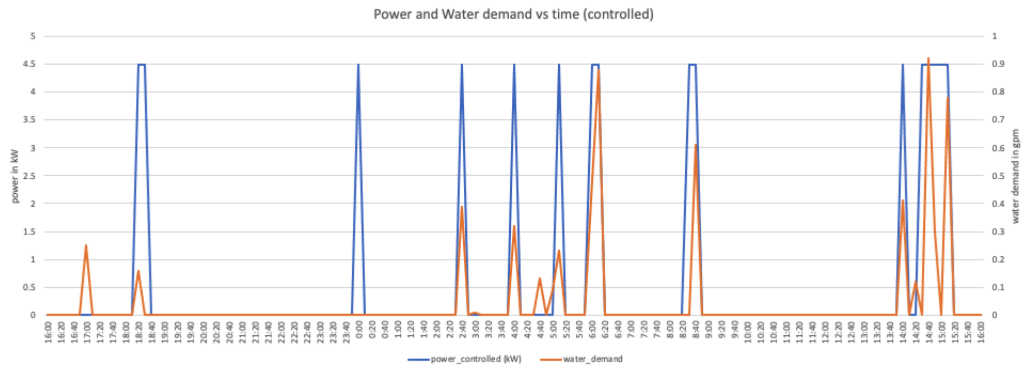


Figure 1: Power and Water Demand vs Time (un-controlled)

2 Winter 2021

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

2.1 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.

2.2 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.

2.3 procedures

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.

2.4 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.

2.5 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.

2.6 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_waterhe
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

2.7 results

1. switch_object

   The above table 1 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.

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 3.

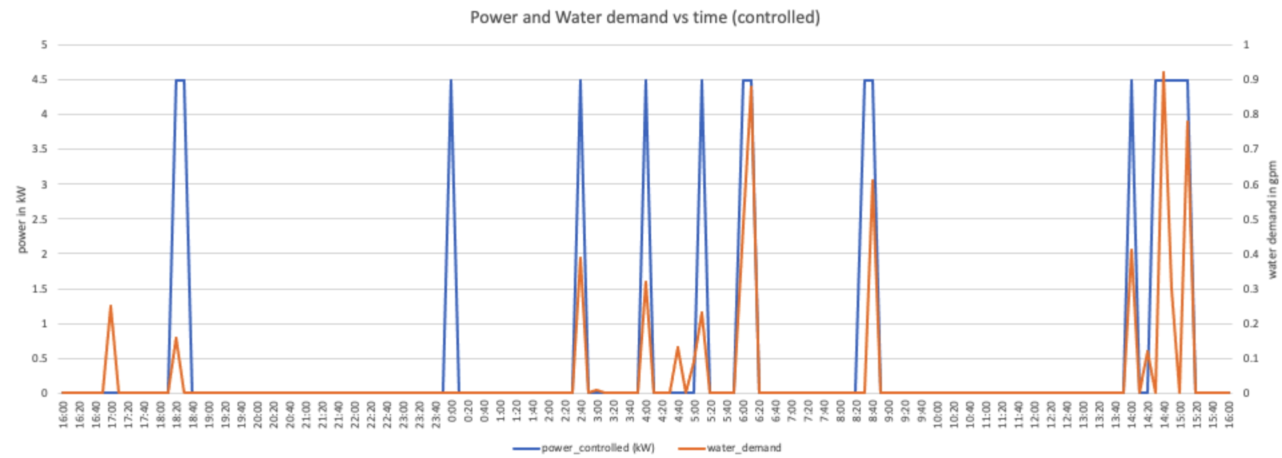


Figure 2: Power and Water Demand vs Time (controlled)

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

3 Spring 2021

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

3.1 objective

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

3.2 outline

What steps are required?

1. Build a WH object with HEATPUMP 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.

3.3 procedures

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.

3.4 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.

3.5 observations

1. HP_WH:

    - Water temperature increases above the setpoint. HP_WH object does **NOT** respond to setpoints as shown in figure 3.

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

3.6 debugging

3.6.1 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.

3.7 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.

3.8 Heating Element Capacity

The heating element capacity equation in the EWH is voltage dependant as shown in equation 1.

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

However, the heating element capacity for the heat pump water heater does not have a voltage dependence as shown in equation 2.

$$HeatingElementCapacity = (1.09+(1.17-1.09)*(get_Tambient(location)-50)/(70-50))*(0.379+0.003 \tag{2}$$

3.9 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

3.10 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.

3.11 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 4 and figure 3 for a visual analysis.

3.12 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)

3.12.1 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.

3.13 How does a Heat Pump Water-Heater work?

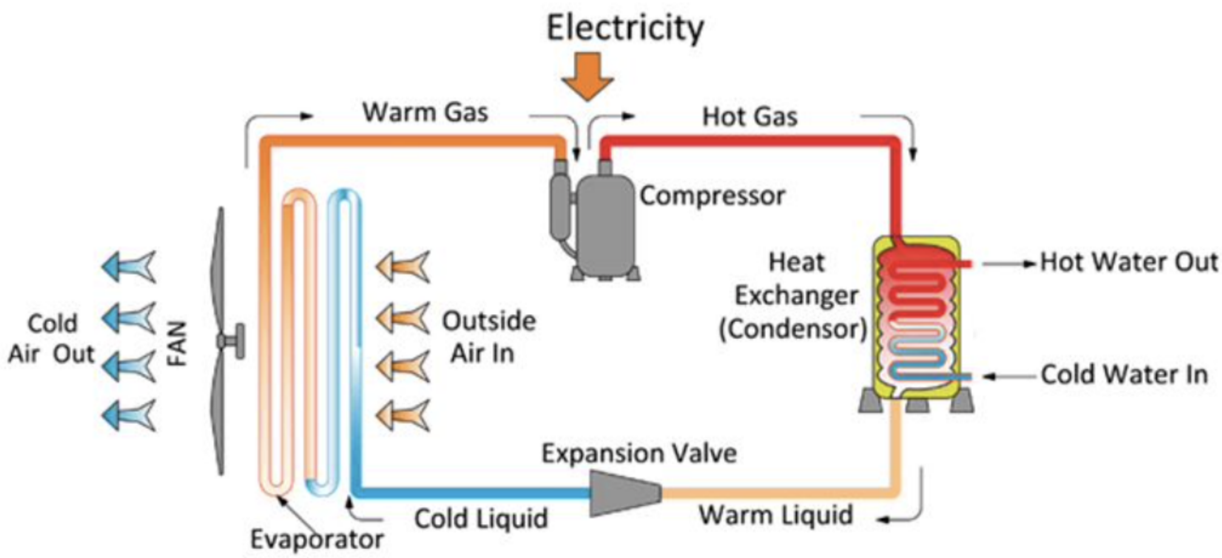


Figure 5: source:nukiengineering.com

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.

3.14 Coefficient of Performance (CoP)

In EWH, we measure the efficiency to understand the performance of the WH. However, in the HPWH, we measure the CoP to see the ratio of useful heating or cooling provided to the required work as shown in equation 3.

In GridLAB-D waterheater.cpp file, the CoP of the HPWH is defined with an equation that contains a set of integers. To make CoP more accessible, we need a more general equation.

$$CoP = \frac{E_{delivered}}{E_{in}} \tag{3}$$

$E_{delivered}$ can be defined as:

$$\Delta E_{delivered} = \frac{(V_i \rho_w C_{p,w}(T_{t,i} - T_{ref})) - (V_i \rho_w C_{p,w}(T_{t-1,i} - T_{ref}))}{t} \tag{4}$$

Where:

- V_i = Volume of the node
- ρ_w = water density of the node
- $C_{p,w}$ = water specific heat capacity.
- $T_{t,i}$ = Temperature of the node measured at time t.
- T_{ref} = reference water temperature (default)
  - For inlet water, the temperature is 60 °.

4 Summer 2021

All data and plots are in PSU Pwrlab Github account. If they're not there, contact midrar@pdx.edu

5 Fall 2021

5.1 Coefficient of Performance: V1

The coefficient of performance is a measure of the useful energy transferred to the water in the tank per the system's supplied work. In other words, how much thermal energy can one get from 100 W input power, for example. The data obtained from this section are from EMCB use cases. There were different equations obtained from different resources [?, ?, ?]. All the aforementioned equations result in the following:

$$COP = \frac{Q}{E_{input}} = \frac{m \cdot C_p \cdot \Delta T}{E_{input}} \tag{5}$$

Where:

- m is the mass of the water in the tank in Pounds (lbm)
- C_p is the specific heat of water ($\frac{Btu}{lbm \cdot \circ F}$)
- ΔT is the difference between the ambient temperature and the tank temperature in F.
- E_{input} is the electrical power input in Watts. This includes the compressor and the heating element.

Equation 5 is applied to the morning shower in the EMCB studies. The morning shower is a 20 gallon water draw. The change in the water temperature during the heating process is linear. Therefore, a cumulative sum of the input power and then the average were calculated which resulted in 1680 W. Here's a list of the numerical value in equation 5:

- E_{input} = 1680 W.
- m = 50 gallon $\times$ 8.34 = 417 lbm.
- C$_p$ = 1.001 $\frac{Btu}{lbm \cdot {}^\circ F}$.
- $T_{ambient}$ = 75 °F

For example, if the current temperature in the tank is 100 $°F$, then the COP can be calculated as follows:

$$COP = \frac{(417[lbm] \times 1.001\frac{Btu}{lbm \cdot F} \times (100 - 75)[F]) \times 0.293}{1680[W]} \& = 1.82 \tag{6}$$

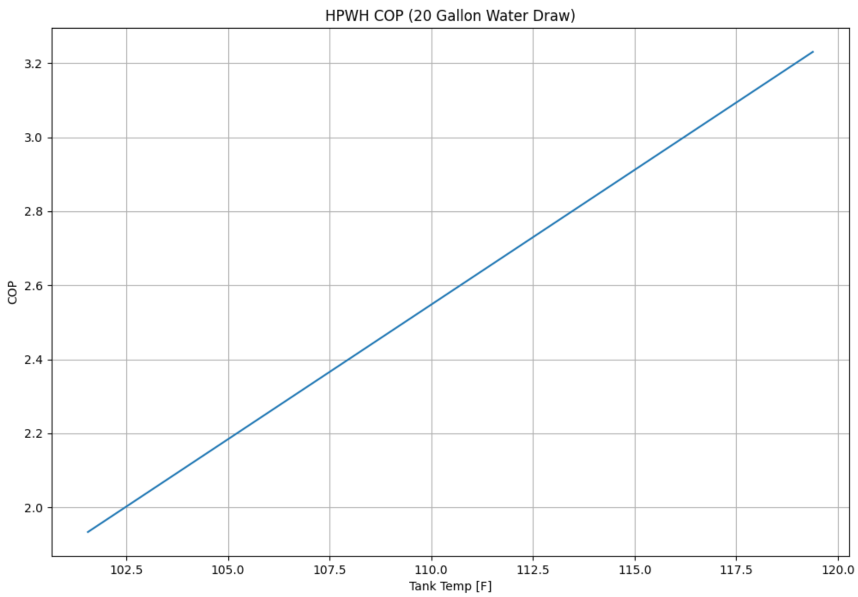


Figure 6: HPWH COP: 20 Gallon Water Draw

5.2 Coefficient of Performance: V2

The equation used to calculate and plot the COP of the HPWH is as follows:

$$COP = \frac{EnergyTake}{Watts} \tag{7}$$

The HPWH was set to vacation mode for three days. After, the HPWH was switched to Hybrid mode. Here's the data when the HPWH switch ON.

time	EnergyTake	Watts
Mon Sep 20 12:17:12 2021	2775	4449.0300
Mon Sep 20 12:18:12 2021	2775	4665.8100
Mon Sep 20 12:19:13 2021	2625	4686.4500
Mon Sep 20 12:20:14 2021	2550	4717.4200
Mon Sep 20 12:21:14 2021	2250	4738.0600
Mon Sep 20 12:22:15 2021	2175	4748.3900
Mon Sep 20 12:23:15 2021	2025	4707.1000
Mon Sep 20 12:24:16 2021	2025	4769.0300
Mon Sep 20 12:25:17 2021	1875	4769.0300
Mon Sep 20 12:26:17 2021	1800	4758.7100
Mon Sep 20 12:27:18 2021	1725	4769.0300
Mon Sep 20 12:28:18 2021	1725	4769.0300
Mon Sep 20 12:29:19 2021	1650	4779.3500
Mon Sep 20 12:30:19 2021	1575	4676.1300
Mon Sep 20 12:31:20 2021	1575	4707.1000
Mon Sep 20 12:32:21 2021	1425	4645.1600
Mon Sep 20 12:33:21 2021	1425	4696.7700
Mon Sep 20 12:34:22 2021	1275	4696.7700
Mon Sep 20 12:35:22 2021	1200	4686.4500
Mon Sep 20 12:36:23 2021	1050	392.2580
Mon Sep 20 12:37:24 2021	1050	402.5810
Mon Sep 20 12:38:24 2021	1050	402.5810
Mon Sep 20 12:39:25 2021	975	402.5810

Mon Sep 20 12:40:25 2021	975	402.5810
Mon Sep 20 12:41:26 2021	975	402.5810
Mon Sep 20 12:42:27 2021	975	402.5810
Mon Sep 20 12:43:27 2021	975	402.5810
Mon Sep 20 12:44:28 2021	975	402.5810
Mon Sep 20 12:45:28 2021	975	402.5810
Mon Sep 20 12:46:29 2021	975	412.9030
Mon Sep 20 12:47:29 2021	975	402.5810
Mon Sep 20 12:48:30 2021	975	402.5810
Mon Sep 20 12:49:31 2021	975	402.5810
Mon Sep 20 12:50:31 2021	900	402.5810
Mon Sep 20 12:51:32 2021	900	412.9030
Mon Sep 20 12:52:32 2021	825	412.9030
Mon Sep 20 12:53:33 2021	825	402.5810
Mon Sep 20 12:54:34 2021	825	412.9030
Mon Sep 20 12:55:34 2021	825	412.9030
Mon Sep 20 12:56:35 2021	825	412.9030
Mon Sep 20 12:57:35 2021	825	412.9030
Mon Sep 20 12:58:36 2021	825	412.9030
Mon Sep 20 12:59:36 2021	825	412.9030
Mon Sep 20 13:00:37 2021	750	412.9030
Mon Sep 20 13:01:38 2021	750	412.9030
Mon Sep 20 13:02:38 2021	750	412.9030
Mon Sep 20 13:03:39 2021	750	412.9030
Mon Sep 20 13:04:39 2021	600	412.9030
Mon Sep 20 13:05:40 2021	600	423.2260
Mon Sep 20 13:06:41 2021	600	423.2260
Mon Sep 20 13:07:41 2021	600	423.2260
Mon Sep 20 13:08:42 2021	600	423.2260
Mon Sep 20 13:09:42 2021	600	423.2260

Mon Sep 20 13:10:43 2021	600	423.2260
Mon Sep 20 13:11:43 2021	600	423.2260
Mon Sep 20 13:12:44 2021	600	423.2260
Mon Sep 20 13:13:45 2021	600	423.2260
Mon Sep 20 13:14:45 2021	525	423.2260
Mon Sep 20 13:15:46 2021	525	423.2260
Mon Sep 20 13:16:46 2021	525	423.2260
Mon Sep 20 13:17:47 2021	525	423.2260
Mon Sep 20 13:18:48 2021	525	433.5480
Mon Sep 20 13:19:48 2021	525	433.5480
Mon Sep 20 13:20:49 2021	525	433.5480
Mon Sep 20 13:21:49 2021	525	433.5480
Mon Sep 20 13:22:50 2021	450	433.5480
Mon Sep 20 13:23:51 2021	450	433.5480
Mon Sep 20 13:24:51 2021	375	433.5480
Mon Sep 20 13:25:52 2021	375	433.5480
Mon Sep 20 13:26:52 2021	375	433.5480
Mon Sep 20 13:27:53 2021	375	433.5480
Mon Sep 20 13:28:53 2021	375	433.5480
Mon Sep 20 13:29:54 2021	225	433.5480
Mon Sep 20 13:30:55 2021	225	433.5480
Mon Sep 20 13:31:55 2021	225	433.5480
Mon Sep 20 13:32:56 2021	225	433.5480
Mon Sep 20 13:33:56 2021	225	433.5480
Mon Sep 20 13:34:57 2021	225	433.5480
Mon Sep 20 13:35:58 2021	225	443.8710
Mon Sep 20 13:36:58 2021	225	443.8710
Mon Sep 20 13:37:59 2021	225	443.8710
Mon Sep 20 13:38:59 2021	75	443.8710
Mon Sep 20 13:40:00 2021	75	443.8710

Mon Sep 20 13:41:01 2021	75	433.5480
Mon Sep 20 13:42:01 2021	75	433.5480
Mon Sep 20 13:43:02 2021	75	433.5480
Mon Sep 20 13:44:02 2021	75	443.8710
Mon Sep 20 13:45:03 2021	75	443.8710
Mon Sep 20 13:46:03 2021	0	443.8710
Mon Sep 20 13:47:04 2021	0	443.8710
Mon Sep 20 13:48:05 2021	0	443.8710
Mon Sep 20 13:49:05 2021	0	454.1940
Mon Sep 20 13:50:06 2021	0	443.8710
Mon Sep 20 13:51:06 2021	0	454.1940
Mon Sep 20 13:52:07 2021	0	454.1940
Mon Sep 20 13:53:07 2021	0	454.1940
Mon Sep 20 13:54:08 2021	0	443.8710
Mon Sep 20 13:55:08 2021	0	454.1940
Mon Sep 20 13:56:09 2021	0	454.1940
Mon Sep 20 13:57:09 2021	0	454.1940
Mon Sep 20 13:58:10 2021	0	454.1940
Mon Sep 20 13:59:11 2021	0	454.1940
Mon Sep 20 14:00:11 2021	0	454.1940
Mon Sep 20 14:01:12 2021	0	454.1940
Mon Sep 20 14:02:12 2021	0	454.1940
Mon Sep 20 14:03:13 2021	0	454.1940
Mon Sep 20 14:04:13 2021	0	454.1940
Mon Sep 20 14:05:14 2021	0	464.5160
Mon Sep 20 14:06:15 2021	0	454.1940
Mon Sep 20 14:07:15 2021	0	454.1940
Mon Sep 20 14:08:16 2021	0	454.1940
Mon Sep 20 14:09:16 2021	0	454.1940
Mon Sep 20 14:10:17 2021	0	454.1940

Mon Sep 20 14:11:17 2021	0	41.2903
Mon Sep 20 14:12:18 2021	0	30.9677
Mon Sep 20 14:13:19 2021	0	41.2903
Mon Sep 20 14:14:19 2021	0	41.2903
Mon Sep 20 15:20:57 2021	0	10.3226

The HPWH was ON for 81 minutes to heat the water up to the setpoints, 120 $°F$. Therefore, the values of watts consumed was converted to Watts-hour as follows:

$$Wh = Watts \times \frac{x-1}{60} \tag{8}$$

Where x is the duration of the heating process.

The following figures show the COP VS:

- time
- EnergyTake
- Line fit

The average COP is 3.2

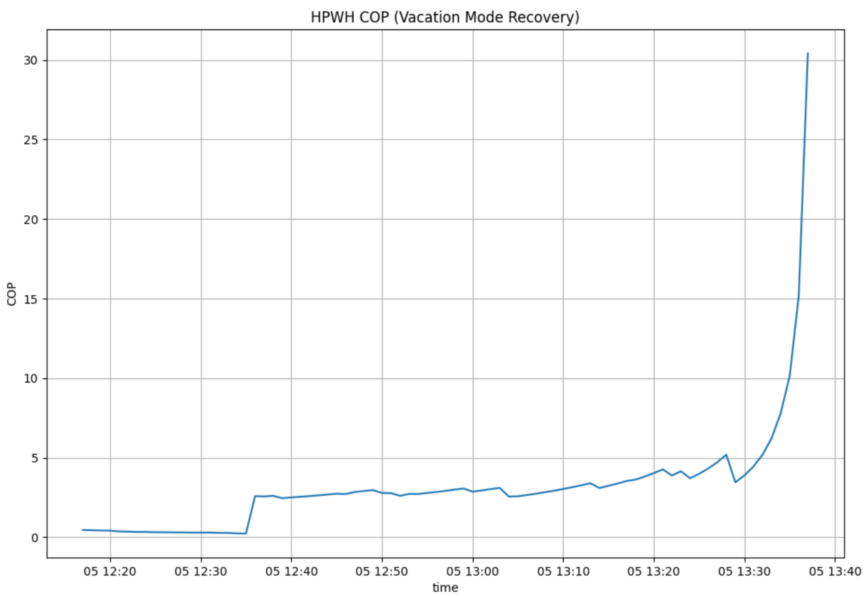


Figure 7: HPWH COP vs Time: Vacation Mode Recovery

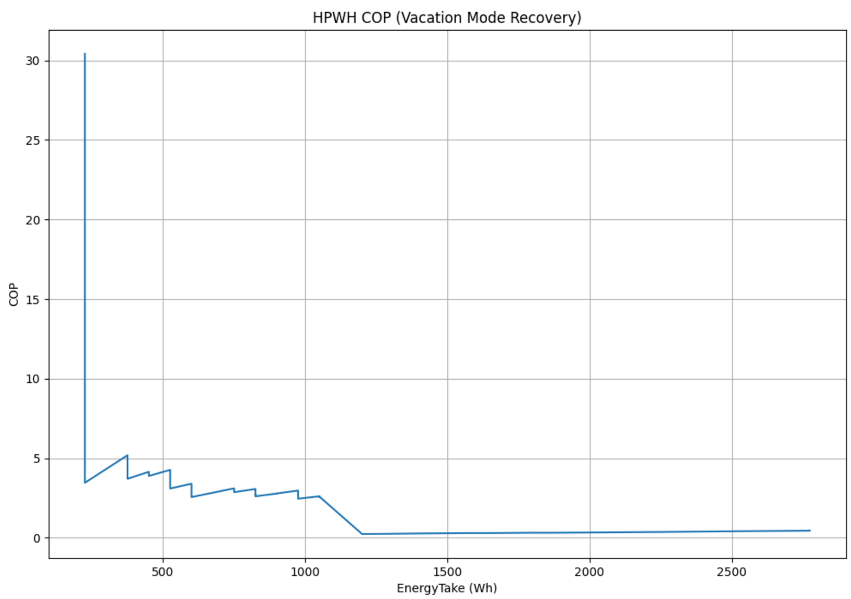


Figure 8: HPWH COP vs EnergyTake: Vacation Mode Recovery

5.3 October 12

In order to proceed with the COP calculations, we need to calculate the watts and watts-hour used by the water heater every time step. The following sections will walk through the process in steps.

5.3.1 Testing

In order to measure the output energy of the water heater, we need to the compressor to run as long as possible. Therefore, the efficiency mode was used. The hot water inside the tank was completely drawn by opening the valve and let the water flow. After that the unit was set in vacation mode for a few days (extra precautions. The unit should be good to go from opening the valve). Then the unit was turned on in efficieny mode. The efficiency mode allows the compressor to do most of the work to heat the water instead of the heating element. The data collected from this testing is shown below:

timestamp	real_available_Wh	consumed_watts
Fri Oct 8 12:26:59 2021	4050	4459.3500
Fri Oct 8 12:28:00 2021	4050	4676.1300
Fri Oct 8 12:29:01 2021	3750	4665.8100
Fri Oct 8 12:30:01 2021	3525	4727.7400
Fri Oct 8 12:31:02 2021	3450	4696.7700
Fri Oct 8 12:32:03 2021	3375	4707.1000
Fri Oct 8 12:33:04 2021	3225	4707.1000
Fri Oct 8 12:34:05 2021	3225	4696.7700
Fri Oct 8 12:35:05 2021	3000	4717.4200
Fri Oct 8 12:36:06 2021	3000	4696.7700
Fri Oct 8 12:37:07 2021	2925	4707.1000
Fri Oct 8 12:38:08 2021	2775	4727.7400
Fri Oct 8 12:39:09 2021	2700	4748.3900
Fri Oct 8 12:40:09 2021	2700	4696.7700

Continued on next page

timestamp	real_available_Wh	consumed_watts
Fri Oct 8 12:41:10 2021	2625	4748.3900
Fri Oct 8 12:42:11 2021	2550	4696.7700
Fri Oct 8 12:43:12 2021	2550	4717.4200
Fri Oct 8 12:44:13 2021	2475	4738.0600
Fri Oct 8 12:45:13 2021	2400	4748.3900
Fri Oct 8 12:46:14 2021	2325	4696.7700
Fri Oct 8 12:47:15 2021	2250	4717.4200
Fri Oct 8 12:48:16 2021	2175	4707.1000
Fri Oct 8 12:49:17 2021	2100	350.9680
Fri Oct 8 12:50:17 2021	2100	350.9680
Fri Oct 8 12:51:18 2021	2025	350.9680
Fri Oct 8 12:52:19 2021	2025	350.9680
Fri Oct 8 12:53:20 2021	1950	350.9680
Fri Oct 8 12:54:20 2021	1950	361.2900
Fri Oct 8 12:55:21 2021	1950	350.9680
Fri Oct 8 12:56:22 2021	2025	350.9680
Fri Oct 8 12:57:23 2021	2025	350.9680
Fri Oct 8 12:58:24 2021	2025	361.2900
Fri Oct 8 12:59:24 2021	2025	350.9680
Fri Oct 8 13:00:25 2021	2025	361.2900
Fri Oct 8 13:01:26 2021	2025	361.2900
Fri Oct 8 13:02:27 2021	2025	361.2900
Fri Oct 8 13:03:28 2021	1950	361.2900
Fri Oct 8 13:04:28 2021	1950	361.2900
Fri Oct 8 13:05:29 2021	1950	361.2900
Fri Oct 8 13:06:30 2021	1950	361.2900
Fri Oct 8 13:07:31 2021	1800	361.2900
Fri Oct 8 13:08:32 2021	1800	361.2900
Fri Oct 8 13:09:32 2021	1725	371.6130

Continued on next page

timestamp	real_available_Wh	consumed_watts
Fri Oct 8 13:10:33 2021	1725	361.2900
Fri Oct 8 13:11:34 2021	1725	361.2900
Fri Oct 8 13:12:35 2021	1725	371.6130
Fri Oct 8 13:13:36 2021	1650	371.6130
Fri Oct 8 13:14:36 2021	1650	371.6130
Fri Oct 8 13:15:37 2021	1650	371.6130
Fri Oct 8 13:16:38 2021	1650	371.6130
Fri Oct 8 13:17:39 2021	1650	371.6130
Fri Oct 8 13:18:40 2021	1575	371.6130
Fri Oct 8 13:19:40 2021	1575	371.6130
Fri Oct 8 13:20:41 2021	1575	381.9350
Fri Oct 8 13:21:42 2021	1575	371.6130
Fri Oct 8 13:22:43 2021	1575	381.9350
Fri Oct 8 13:23:44 2021	1575	381.9350
Fri Oct 8 13:24:44 2021	1575	381.9350
Fri Oct 8 13:25:45 2021	1425	381.9350
Fri Oct 8 13:26:46 2021	1425	381.9350
Fri Oct 8 13:27:47 2021	1425	381.9350
Fri Oct 8 13:28:48 2021	1425	381.9350
Fri Oct 8 13:29:48 2021	1425	381.9350
Fri Oct 8 13:30:49 2021	1425	381.9350
Fri Oct 8 13:31:50 2021	1350	381.9350
Fri Oct 8 13:32:51 2021	1350	381.9350
Fri Oct 8 13:33:52 2021	1275	381.9350
Fri Oct 8 13:34:52 2021	1275	392.2580
Fri Oct 8 13:35:53 2021	1275	392.2580
Fri Oct 8 13:36:54 2021	1275	381.9350
Fri Oct 8 13:37:55 2021	1275	381.9350
Fri Oct 8 13:38:56 2021	1275	392.2580

Continued on next page

timestamp	real_available_Wh	consumed_watts
Fri Oct 8 13:39:56 2021	1275	392.2580
Fri Oct 8 13:40:57 2021	1200	392.2580
Fri Oct 8 13:41:58 2021	1200	392.2580
Fri Oct 8 13:42:59 2021	1200	392.2580
Fri Oct 8 13:43:59 2021	1050	392.2580
Fri Oct 8 13:45:00 2021	1050	402.5810
Fri Oct 8 13:46:01 2021	1050	392.2580
Fri Oct 8 13:47:02 2021	1050	392.2580
Fri Oct 8 13:48:03 2021	1050	402.5810
Fri Oct 8 13:49:03 2021	1050	402.5810
Fri Oct 8 13:50:04 2021	1050	402.5810
Fri Oct 8 13:51:05 2021	1050	402.5810
Fri Oct 8 13:52:06 2021	975	402.5810
Fri Oct 8 13:53:07 2021	975	402.5810
Fri Oct 8 13:54:07 2021	975	402.5810
Fri Oct 8 13:55:08 2021	975	402.5810
Fri Oct 8 13:56:09 2021	975	402.5810
Fri Oct 8 13:57:10 2021	975	402.5810
Fri Oct 8 13:58:11 2021	975	402.5810
Fri Oct 8 13:59:11 2021	900	402.5810
Fri Oct 8 14:00:12 2021	900	402.5810
Fri Oct 8 14:01:13 2021	825	402.5810
Fri Oct 8 14:02:14 2021	825	412.9030
Fri Oct 8 14:03:15 2021	825	412.9030
Fri Oct 8 14:04:15 2021	825	412.9030
Fri Oct 8 14:05:16 2021	825	412.9030
Fri Oct 8 14:06:17 2021	825	412.9030
Fri Oct 8 14:07:18 2021	825	412.9030
Fri Oct 8 14:08:19 2021	750	412.9030

Continued on next page

timestamp	real_available_Wh	consumed_watts
Fri Oct 8 14:09:19 2021	750	412.9030
Fri Oct 8 14:10:20 2021	750	412.9030
Fri Oct 8 14:11:21 2021	750	412.9030
Fri Oct 8 14:12:22 2021	600	423.2260
Fri Oct 8 14:13:23 2021	600	412.9030
Fri Oct 8 14:14:23 2021	600	423.2260
Fri Oct 8 14:15:24 2021	600	423.2260
Fri Oct 8 14:16:25 2021	600	412.9030
Fri Oct 8 14:17:26 2021	600	423.2260
Fri Oct 8 14:18:27 2021	600	412.9030
Fri Oct 8 14:19:27 2021	600	423.2260
Fri Oct 8 14:20:28 2021	600	423.2260
Fri Oct 8 14:21:29 2021	525	423.2260
Fri Oct 8 14:22:30 2021	525	423.2260
Fri Oct 8 14:23:31 2021	525	423.2260
Fri Oct 8 14:24:31 2021	525	433.5480
Fri Oct 8 14:25:32 2021	525	423.2260
Fri Oct 8 14:26:33 2021	525	423.2260
Fri Oct 8 14:27:34 2021	525	423.2260
Fri Oct 8 14:28:35 2021	450	433.5480
Fri Oct 8 14:29:35 2021	450	433.5480
Fri Oct 8 14:30:36 2021	375	433.5480
Fri Oct 8 14:31:37 2021	375	433.5480
Fri Oct 8 14:32:38 2021	375	433.5480
Fri Oct 8 14:33:39 2021	375	433.5480
Fri Oct 8 14:34:39 2021	375	433.5480
Fri Oct 8 14:35:40 2021	225	433.5480
Fri Oct 8 14:36:41 2021	225	433.5480
Fri Oct 8 14:37:42 2021	225	433.5480

Continued on next page

timestamp	real_available_Wh	consumed_watts
Fri Oct 8 14:38:43 2021	225	433.5480
Fri Oct 8 14:39:43 2021	225	443.8710
Fri Oct 8 14:40:44 2021	225	443.8710
Fri Oct 8 14:41:45 2021	225	443.8710
Fri Oct 8 14:42:46 2021	75	443.8710
Fri Oct 8 14:43:47 2021	75	443.8710
Fri Oct 8 14:44:47 2021	75	443.8710
Fri Oct 8 14:45:48 2021	75	443.8710
Fri Oct 8 14:46:49 2021	75	443.8710
Fri Oct 8 14:47:50 2021	75	443.8710
Fri Oct 8 14:48:50 2021	75	443.8710
Fri Oct 8 14:49:51 2021	0	443.8710
Fri Oct 8 14:50:52 2021	0	443.8710
Fri Oct 8 14:51:53 2021	0	443.8710
Fri Oct 8 14:52:54 2021	0	454.1940
Fri Oct 8 14:53:54 2021	0	443.8710
Fri Oct 8 14:54:55 2021	0	443.8710
Fri Oct 8 14:55:56 2021	0	454.1940
Fri Oct 8 14:56:57 2021	0	443.8710
Fri Oct 8 14:57:58 2021	0	443.8710
Fri Oct 8 14:58:58 2021	0	454.1940
Fri Oct 8 14:59:59 2021	0	454.1940
Fri Oct 8 15:01:00 2021	0	454.1940
Fri Oct 8 15:02:01 2021	0	454.1940
Fri Oct 8 15:03:02 2021	0	454.1940
Fri Oct 8 15:04:02 2021	0	454.1940
Fri Oct 8 15:05:03 2021	0	454.1940
Fri Oct 8 15:06:04 2021	0	454.1940
Fri Oct 8 15:07:05 2021	0	464.5160

Continued on next page

timestamp	real_available_Wh	consumed_watts
Fri Oct 8 15:08:06 2021	0	454.1940
Fri Oct 8 15:09:06 2021	0	454.1940
Fri Oct 8 15:10:07 2021	0	454.1940
Fri Oct 8 15:11:08 2021	0	454.1940
Fri Oct 8 15:12:09 2021	0	464.5160
Fri Oct 8 15:13:10 2021	0	464.5160
Fri Oct 8 15:14:10 2021	0	41.2903
Fri Oct 8 15:15:11 2021	0	30.9677
Fri Oct 8 15:16:12 2021	0	41.2903
Fri Oct 8 15:17:13 2021	0	41.2903

5.3.2 EnergyTake and Watts

The COP equation that I've been using so far is the following:

$$COP = \frac{EnergyTake}{Watts} \tag{9}$$

The EnergyTake definition, from $CTA - 2045$ as shown in figure 9, is the current available energy in the tank. It does not represent the output energy with every timestep. However, the output energy can be represented by the difference between each ET value in each timestep, which in this case, would be 75 Wh. Table 5 shows the data collected from the HPWH durint the testing. The watts conversion to Watts-hour is the same way used as before. No changes.

5.3.3 Data Used for Calculations

<u>Total Energy Storage/Take Capacity</u> is the total amount of energy storage that the end device represents. For example, the energy capacity of a water heater would be the total amount of energy (W-hr) to move the tank from its minimum operating temperature (e.g. what it would allow itself to drop to during a curtailment event) to its maximum operating temperature (e.g. what it could run up to when asked to Basic DR "Load-Up" before shutting off). Similarly, for a thermostat/HVAC system, this would be the total energy (W-hr) to move the temperature of the conditioned space from its curtailed state (at the temp offset for example), back to its max cooled/heated state.

<u>Present Energy Storage/Take Capacity</u> is the amount of energy that the end device can take now. This parameter is also represented in (W-hr) and would normally be some portion of the Total Energy Storage Capacity as illustrated in Figure 11-1. It is recognized that under some extraordinary circumstances, the Present Energy Storage/Take Capacity could exceed the Total. For example, if a water heater temperature has fallen well below the normal minimum regulation range.

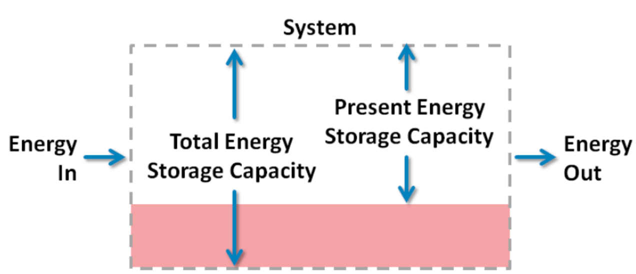


Figure 9: CTA-2045 EnergyTake Definition

Table 6: HPWH Data Used For Calculating COP in this Section

time	EnergyTake (Wh)	consumed_watts (Watts)	time_diff	average_watts	Energy in	cop
12:49:17	2100	350.968	1.02	350.97	5.97	12.56
12:51:18	2025	350.968	2.02	350.97	11.82	6.35
12:53:20	1950	350.968	2.03	350.97	11.87	6.32
13:07:31	1800	361.290	14.18	361.29	85.38	0.88
13:09:32	1725	371.613	2.02	371.61	12.51	6.00
13:13:36	1650	371.613	4.07	371.61	25.21	2.98
13:18:40	1575	371.613	5.07	371.61	31.40	2.39
13:25:45	1425	381.935	7.08	381.94	45.07	1.66
13:31:50	1350	381.935	6.08	381.94	38.70	1.94
13:33:52	1275	381.935	2.03	381.94	12.92	5.80
13:40:57	1200	392.258	7.08	392.26	46.29	1.62
13:43:59	1050	392.258	3.03	392.26	19.81	3.79
13:52:06	975	402.581	8.12	402.58	54.48	1.38
13:59:11	900	402.581	7.08	402.58	47.50	1.58
14:01:13	825	402.581	2.03	402.58	13.62	5.51
14:08:19	750	412.903	7.10	412.90	48.86	1.53
14:12:22	600	423.226	4.05	423.23	28.57	2.63
14:21:29	525	423.226	9.12	423.23	64.33	1.17
14:28:35	450	433.548	7.10	433.55	51.30	1.46
14:30:36	375	433.548	2.02	433.55	14.60	5.14
14:35:40	225	433.548	5.07	433.55	36.63	2.05
14:42:46	75	443.871	7.10	443.87	52.52	1.43

The data for the HPWH including the heating element are as shown in Table 7. I also included the work done by the heating element to the compressor as shown in Figure 10. Furthermore, a line fit to the data to see the average range of the coefficient of performance is shown in Figure 11.

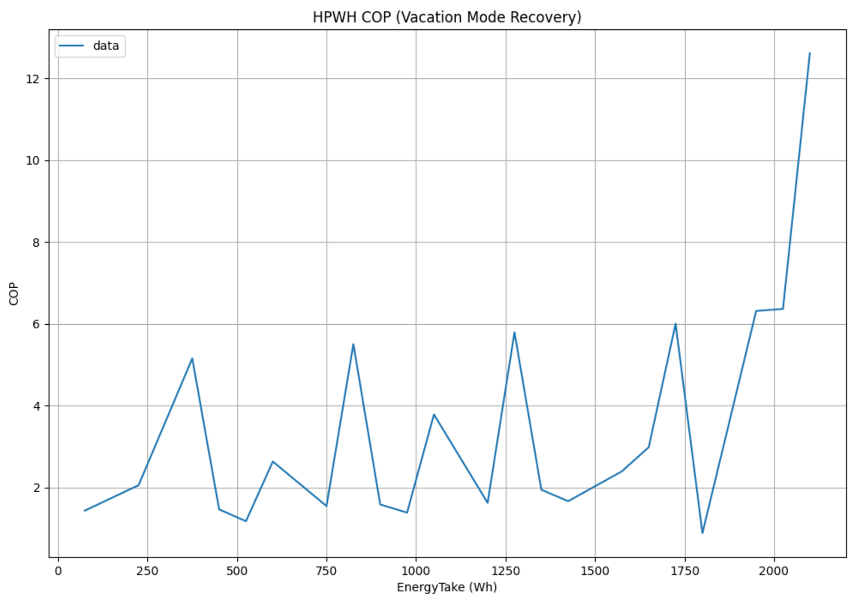


Figure 10: HPWH COP vs EnergyTake: Vacation Mode Recovery

Table 7: Heating Element for HPWH

time	EnergyTake (Wh)	$consumed\_watts$	$time\_diff$ (minutes)	$average\_watts(watts)$	Energy in (Wh)	cop
12:26:59	4050	4459.350	746.50	4459.35	55481.75	0.00
12:29:01	3750	4665.810	2.03	4665.81	157.86	0.48
12:30:01	3525	4727.740	1.00	4727.74	78.80	0.95
12:31:02	3450	4696.770	1.02	4696.77	79.85	0.94
12:32:03	3375	4707.100	1.02	4707.10	80.02	0.94
12:33:04	3225	4707.100	1.02	4707.10	80.02	0.94
12:35:05	3000	4717.420	2.02	4717.42	158.82	0.47
12:37:07	2925	4707.100	2.03	4707.10	159.26	0.47
12:38:08	2775	4727.740	1.02	4727.74	80.37	0.93
12:39:09	2700	4748.390	1.02	4748.39	80.72	0.93
12:41:10	2625	4748.390	2.02	4748.39	159.86	0.47
12:42:11	2550	4696.770	1.02	4696.77	79.85	0.94
12:44:13	2475	4738.060	2.03	4738.06	160.30	0.47
12:45:13	2400	4748.390	1.00	4748.39	79.14	0.95
12:46:14	2325	4696.770	1.02	4696.77	79.85	0.94
12:47:15	2250	4717.420	1.02	4717.42	80.20	0.94
12:48:16	2175	4707.100	1.02	4707.10	80.02	0.94
12:49:17	2100	350.968	1.02	350.97	5.97	12.56
12:51:18	2025	350.968	2.02	350.97	11.82	6.35
12:53:20	1950	350.968	2.03	350.97	11.87	6.32
13:07:31	1800	361.290	14.18	361.29	85.38	0.88
13:09:32	1725	371.613	2.02	371.61	12.51	6.00
13:13:36	1650	371.613	4.07	371.61	25.21	2.98
13:18:40	1575	371.613	5.07	371.61	31.40	2.39
13:25:45	1425	381.935	7.08	381.94	45.07	1.66
13:31:50	1350	381.935	6.08	381.94	38.70	1.94
13:33:52	1275	381.935	2.03	381.94	12.92	5.80
13:40:57	1200	392.258	7.08	392.26	46.29	1.62
13:43:59	1050	392.258	3.03	392.26	19.81	3.79
13:52:06	975	402.581	8.12	402.58	54.48	1.38
13:59:11	900	402.581	7.08	402.58	47.50	1.58
14:01:13	825	402.581	2.03	402.58	13.62	5.51
14:08:19	750	412.903	7.10	412.90	48.86	1.53
14:12:22	600	423.226	4.05	423.23	28.57	2.63
14:21:29	525	423.226	9.12	423.23	64.33	1.17
14:28:35	450	433.548	7.10	433.55	51.30	1.46
14:30:36	375	433.548	2.02	433.55	14.60	5.14
14:35:40	225	433.548	5.07	433.55	36.63	2.05
14:42:46	75	443.871	7.10	443.87	52.52	1.43

5.4 Oct 19

The difference in EnergyTake in each timestemp was calculated. The data calculated is as shown in Table 8

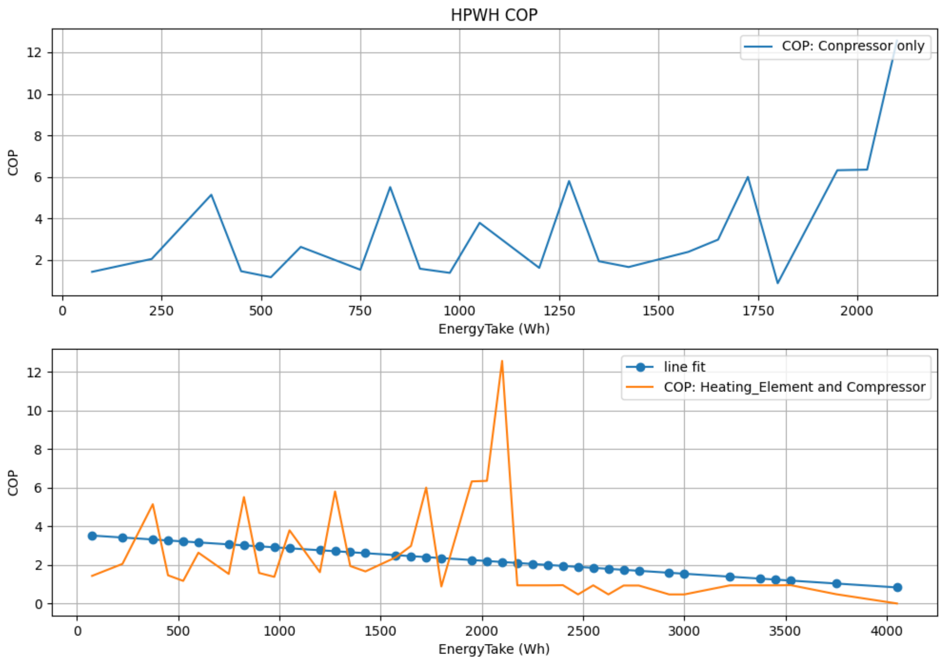


Figure 11: HPWH COP vs EnergyTake: Vacation Mode Recovery

The data above was plotted as shown in the following figure. Note that the spike of COP = 12 has disappeared, which is progress, I guess. But Still the COP plot is not appropriate. The next steps that I have in mind is the following:

- Plot the "EnergyTake diff" values versus time
- Fit a line into the data
- Repeate the same steps for the "power" values in the above table.
- Finally, caluclate the COP from the above lines.

The idea is that these spikes are caused by the fluctuating in the EnergyTake values (increment in 75 Wh). Averaging these values might be a solution to get a nice, clean plot.

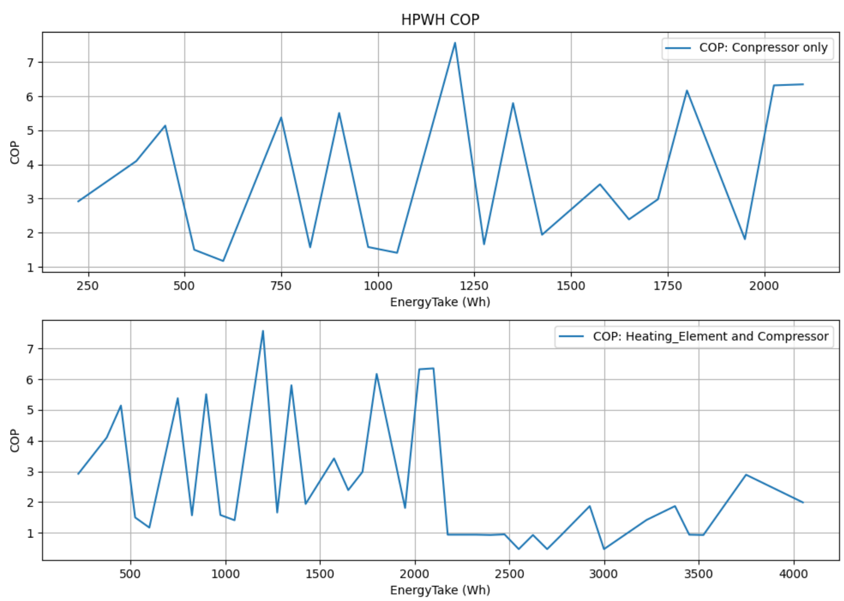


Figure 12: HPWH COP vs EnergyTake: Vacation Mode Recovery

Table 8: EnergyTake Difference Calculated

time	real_available_Wh	consumed_watts	time.1	time_diff	average_watts	power in Wh	EnergyTake_diff	cop
12:26:59	4050	4459.350	12:26:59	2.03	4459.35	150.87	300.0	1.99
12:29:01	3750	4665.810	12:29:01	1.00	4665.81	77.76	225.0	2.89
12:30:01	3525	4727.740	12:30:01	1.02	4727.74	80.37	75.0	0.93
12:31:02	3450	4696.770	12:31:02	1.02	4696.77	79.85	75.0	0.94
12:32:03	3375	4707.100	12:32:03	1.02	4707.10	80.02	150.0	1.87
12:33:04	3225	4707.100	12:33:04	2.02	4707.10	158.47	225.0	1.42
12:35:05	3000	4717.420	12:35:05	2.03	4717.42	159.61	75.0	0.47
12:37:07	2925	4707.100	12:37:07	1.02	4707.10	80.02	150.0	1.87
12:38:08	2775	4727.740	12:38:08	1.02	4727.74	80.37	75.0	0.93
12:39:09	2700	4748.390	12:39:09	2.02	4748.39	159.86	75.0	0.47
12:41:10	2625	4748.390	12:41:10	1.02	4748.39	80.72	75.0	0.93
12:42:11	2550	4696.770	12:42:11	2.03	4696.77	158.91	75.0	0.47
12:44:13	2475	4738.060	12:44:13	1.00	4738.06	78.97	75.0	0.95
12:45:13	2400	4748.390	12:45:13	1.02	4748.39	80.72	75.0	0.93
12:46:14	2325	4696.770	12:46:14	1.02	4696.77	79.85	75.0	0.94
12:47:15	2250	4717.420	12:47:15	1.02	4717.42	80.20	75.0	0.94
12:48:16	2175	4707.100	12:48:16	1.02	4707.10	80.02	75.0	0.94
12:49:17	2100	350.968	12:49:17	2.02	350.97	11.82	75.0	6.35
12:51:18	2025	350.968	12:51:18	2.03	350.97	11.87	75.0	6.32
12:53:20	1950	350.968	12:53:20	14.18	350.97	82.95	150.0	1.81
13:07:31	1800	361.290	13:07:31	2.02	361.29	12.16	75.0	6.17
13:09:32	1725	371.613	13:09:32	4.07	371.61	25.21	75.0	2.98
13:13:36	1650	371.613	13:13:36	5.07	371.61	31.40	75.0	2.39
13:18:40	1575	371.613	13:18:40	7.08	371.61	43.85	150.0	3.42
13:25:45	1425	381.935	13:25:45	6.08	381.94	38.70	75.0	1.94
13:31:50	1350	381.935	13:31:50	2.03	381.94	12.92	75.0	5.80
13:33:52	1275	381.935	13:33:52	7.08	381.94	45.07	75.0	1.66
13:40:57	1200	392.258	13:40:57	3.03	392.26	19.81	150.0	7.57
13:43:59	1050	392.258	13:43:59	8.12	392.26	53.09	75.0	1.41
13:52:06	975	402.581	13:52:06	7.08	402.58	47.50	75.0	1.58
13:59:11	900	402.581	13:59:11	2.03	402.58	13.62	75.0	5.51
14:01:13	825	402.581	14:01:13	7.10	402.58	47.64	75.0	1.57
14:08:19	750	412.903	14:08:19	4.05	412.90	27.87	150.0	5.38
14:12:22	600	423.226	14:12:22	9.12	423.23	64.33	75.0	1.17
14:21:29	525	423.226	14:21:29	7.10	423.23	50.08	75.0	1.50
14:28:35	450	433.548	14:28:35	2.02	433.55	14.60	75.0	5.14
14:30:36	375	433.548	14:30:36	5.07	433.55	36.63	150.0	4.10
14:35:40	225	433.548	14:35:40	7.10	433.55	51.30	150.0	2.92

5.5 Oct 26

In this trial, I'm going to use the cumulative energy divided by the $Energy_{in}$. The $Energy_{in}$ is calculated as following: (As coded in *cop_vmode.py* file)

- Recorded the time difference between each timestep in a column named $time\_diff$
- Then each value in the $time\_diff$ column is converted to seconds (This is because timedelta library. It does not take arguments in minutes. It has to be

seconds.)

- The seconds values is then converted to minutes and saved in the same column, $time\_diff$.
- Then the each value in the $time\_diff$ column is converted to hours by dividing the value by 60.
- Then the consumed power in each timestep ($consumed\_watts$ column) is multiplied by the time difference and saved in a new column named $energy\_in(Wh)$

The cumulative energy values is obvious which I believe there's no need to explain. However, for validation sake, each csv file pulled from the DCS devices, contains a cumulative energy queries. These values are validated with the calculated values.

The following table shows the data recorded for this simulation trial. Note that only the compressor operation is recorded. The heating element operation was ignored in this table.

Table 9: Recorded and Calculated HPWH Porperties

time	cumulative_energy	real_available_Wh	consumed_watts	time.1	EnergyTake_diff	time_diff	average_watts	power	power_cum	cop
12:49:17	0	2100	350.968	12:49:17	75.0	2.02	350.97	11.82	0.00	0.00
12:51:18	26	2025	350.968	12:51:18	75.0	2.03	350.97	11.87	11.82	2.20
12:53:20	52	1950	350.968	12:53:20	150.0	14.18	350.97	82.95	23.69	2.20
13:07:31	234	1800	361.290	13:07:31	75.0	2.02	361.29	12.16	106.64	2.19
13:09:32	260	1725	371.613	13:09:32	75.0	4.07	371.61	25.21	118.80	2.19
13:13:36	312	1650	371.613	13:13:36	75.0	5.07	371.61	31.40	144.01	2.17
13:18:40	390	1575	371.613	13:18:40	150.0	7.08	371.61	43.85	175.41	2.22
13:25:45	481	1425	381.935	13:25:45	75.0	6.08	381.94	38.70	219.26	2.19
13:31:50	559	1350	381.935	13:31:50	75.0	2.03	381.94	12.92	257.96	2.17
13:33:52	585	1275	381.935	13:33:52	75.0	7.08	381.94	45.07	270.88	2.16
13:40:57	676	1200	392.258	13:40:57	150.0	3.03	392.26	19.81	315.95	2.14
13:43:59	715	1050	392.258	13:43:59	75.0	8.12	392.26	53.09	335.76	2.13
13:52:06	819	975	402.581	13:52:06	75.0	7.08	402.58	47.50	388.85	2.11
13:59:11	910	900	402.581	13:59:11	75.0	2.03	402.58	13.62	436.35	2.09
14:01:13	936	825	402.581	14:01:13	75.0	7.10	402.58	47.64	449.97	2.08
14:08:19	1027	750	412.903	14:08:19	150.0	4.05	412.90	27.87	497.61	2.06
14:12:22	1079	600	423.226	14:12:22	75.0	9.12	423.23	64.33	525.48	2.05
14:21:29	1196	525	423.226	14:21:29	75.0	7.10	423.23	50.08	589.81	2.03
14:28:35	1287	450	433.548	14:28:35	75.0	2.02	433.55	14.60	639.89	2.01
14:30:36	1326	375	433.548	14:30:36	150.0	5.07	433.55	36.63	654.49	2.03
14:35:40	1391	225	433.548	14:35:40	150.0	7.10	433.55	51.30	691.12	2.01
14:42:46	1482	75	443.871	14:42:46	75.0	7.08	443.87	52.38	742.42	2.00
14:49:51	1573	0	443.871	14:49:51	0.0	NaN	443.87	NaN	794.80	1.98

As shown in table 9, the first cop values are zero because the cumulative energy is zero. The actual cop values were NaN but they were converted to zero. Anyways, **The cumulative energy values are queries from the unit sent using**

CTA2045. The following table shows the calculated cumualtive energy which is the cumsum of the $EnergyTake\_diff$ column. These values are assigned in a new column that is called $EnergyTake\_cumsum$.

5.6 Nov 16

The calculated COP for this work is as shown in equation 10

$$COP = \frac{EnergyOut}{EnergyIn} \tag{10}$$

The *EnergyOut* is represented by the *EnergyTake* gradient during the heating process. The HPWH was set to *efficiecy mode* and cooled down all the way to inlet water temperature. The heating element turned on for a while and then the compressor turned on. Since the operation of the heating element is predictable and its efficiency is understood, the COP was calculated for the compressor operation.

5.6.1 Data Recorded

During the heating process, several HPWH properties were recorded and calculated as shown in Table 10. This section discusses the properties that were used to caluclate the COP.

Table 10: Recorded and Calculated HPWH Porperties

cumulative_energy	EnergyTake	consumed_watts	time	EnergyTake_diff	time_diff	average_watts	Energy_in	power_cum	ET_fit_comp	W_fit_comp	cop_comp
52	1950	350.968	12:53:20	150	14.18	350.97	82.95	23.69	94.70	34.47	2.75
234	1800	361.290	13:07:31	75	2.02	361.29	12.16	106.64	94.82	34.63	2.74
260	1725	371.613	13:09:32	75	4.07	371.61	25.21	118.80	95.06	34.95	2.72
312	1650	371.613	13:13:36	75	5.07	371.61	31.40	144.01	95.36	35.35	2.70
390	1575	371.613	13:18:40	150	7.08	371.61	43.85	175.41	95.78	35.91	2.67
481	1425	381.935	13:25:45	75	6.08	381.94	38.70	219.26	96.14	36.39	2.64
559	1350	381.935	13:31:50	75	2.03	381.94	12.92	257.96	96.26	36.55	2.63
585	1275	381.935	13:33:52	75	7.08	381.94	45.07	270.88	96.68	37.11	2.61
676	1200	392.258	13:40:57	150	3.03	392.26	19.81	315.95	96.86	37.35	2.59
715	1050	392.258	13:43:59	75	8.12	392.26	53.09	335.76	97.34	37.99	2.56
819	975	402.581	13:52:06	75	7.08	402.58	47.50	388.85	97.76	38.55	2.54
910	900	402.581	13:59:11	75	2.03	402.58	13.62	436.35	97.88	38.71	2.53
936	825	402.581	14:01:13	75	7.10	402.58	47.64	449.97	98.30	39.27	2.50
1027	750	412.903	14:08:19	150	4.05	412.90	27.87	497.61	98.54	39.59	2.49
1079	600	423.226	14:12:22	75	9.12	423.23	64.33	525.48	99.08	40.31	2.46
1196	525	423.226	14:21:29	75	7.10	423.23	50.08	589.81	99.50	40.87	2.43
1287	450	433.548	14:28:35	75	2.02	433.55	14.60	639.89	99.62	41.03	2.43
1326	375	433.548	14:30:36	150	5.07	433.55	36.63	654.49	99.92	41.43	2.41
1391	225	433.548	14:35:40	150	7.10	433.55	51.30	691.12	100.34	41.99	2.39
1482	75	443.871	14:42:46	75	7.08	443.87	52.38	742.42	100.76	42.55	2.37

5.6.2 *EnergyTake* & Energy_in

The time (in minutes) that it takes the HPWH to reports a change in the *EnergyTake* ("EnergyTake_diff" column) and the power consumption ("avergae_watts" column) is shown in Table 10, column "time_diff". For each time window, the consumed watts is converted to energy and reported in the column "Energy_in". As shown in the table, the "Energy_in" values are proportional to the size of the time window. That is, as the time window increases, the "Energy_in" values increase as well.

5.6.3 Linear line fitting

A linear regression algorithm is used to fit lines to two plots. The first plot is the *EnergyTake* VS Time, and the second plot is the *Energy_in* VS Time. Numpy Module in python provides an efficient way of doing such task by simply calling the function ***polyfit*** along with equation's degree. Since our data behaves linearly, a polynomial equaiton with a degree of 1 was used to fit a line into each plot. Figure 13 (Top) shows the *EnergyTake* vs Time in minutes and the *Energy_in* along with a line to fit the data in each plot

5.6.4 COP

The COP is calculated as shown in Equation 10. The EnergyOut is the "EnergyTake_diff" column shown in Table 10 and the EnergyIn is the "EnergyIn" column shown in Table 10. The lines fit of each of the represented data were used to calculate the COP. The COP ranges between 2.3 and 2.76 as shown in Figure 14

Therefore, the equation for the COP is as follows:

$$COP = \frac{0.06 \times EnergyTake + 93.86}{0.08 \times EnergyIn + 33.35} \tag{11}$$

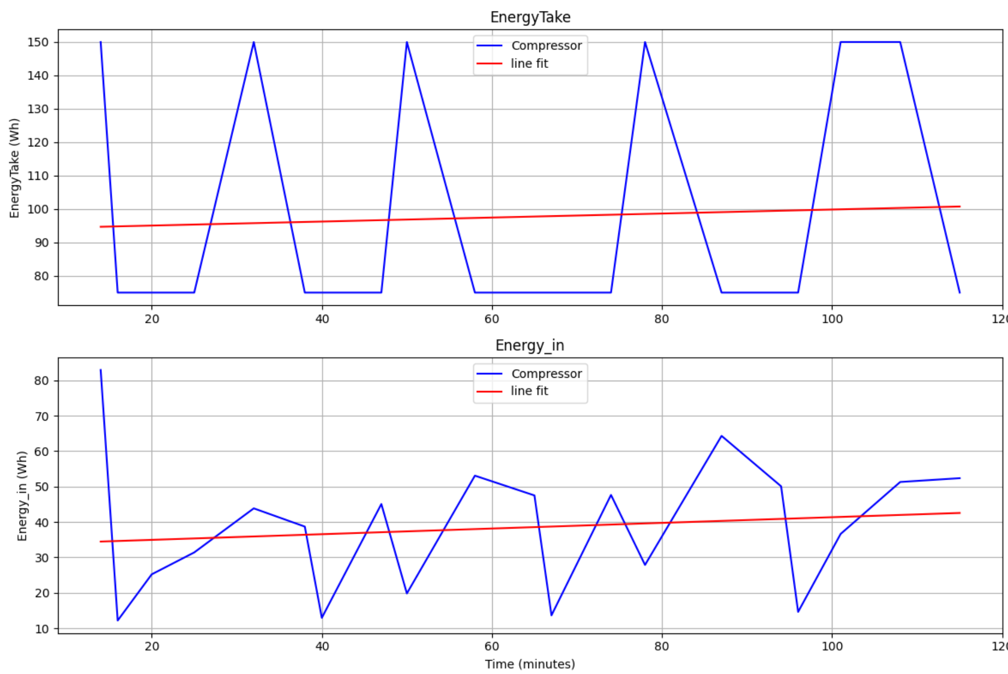


Figure 13: HPWH EnergyTake (top) and Energy_in (bottom) line fitting

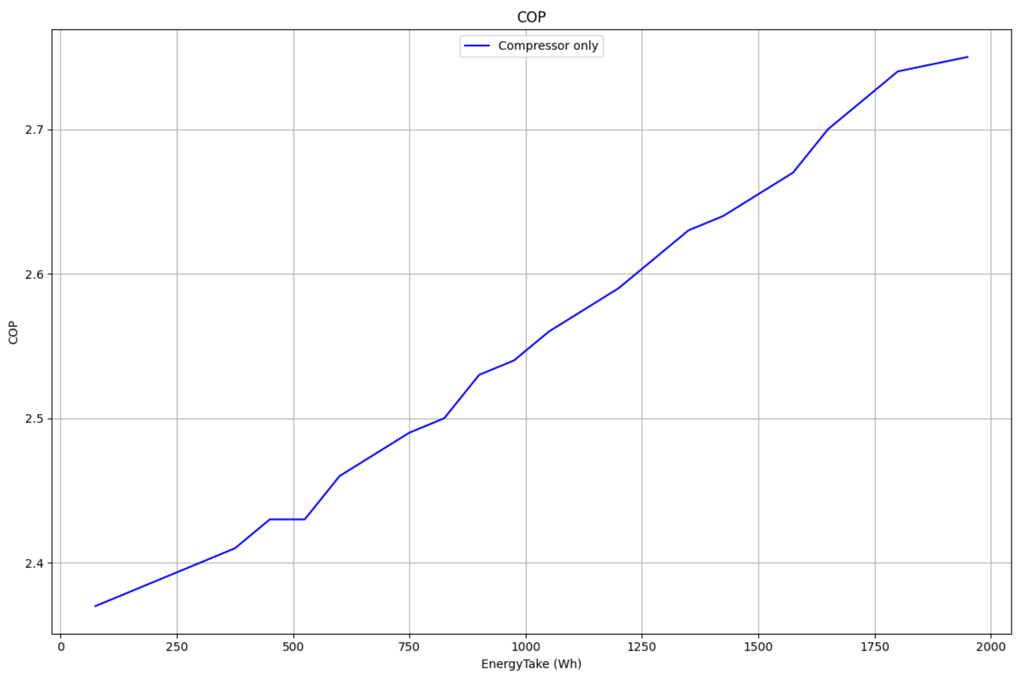


Figure 14: HPWH COP vs EnergyTake: Vacation Mode Recovery

5.6.5 Relationship between Power Consumption (W) and *EnergyTake* (Wh)

This relationship is needed when the *EnergyTake* changes due to the triggering of the heating source, either the heating element or the compressor. To get an equation that is representative for this relationship, both maximum compressor threshold points of the compressor and the heating element are considered. From the EMCB use cases report, case 3, the 20 gallon water draw event caused the *EnergyTake* to increase more than 2000 Wh. The heating element triggered and heated the water until the *EnergyTake* reached 200 Wh. The heating element then switched off and the compressor triggered to heat the water until the set points. Therefore, two equations are needed to represent this behavior. One for the heating element operation, and one for the compressor operation.

For the compressor operation, the *EnergyTake* data were plotted against the power consumption as shown in Figure 15. The same process is repeated for the heating element as shown in Figure 16. To get a representative equation for the compressor and heating element, a line-fit equation was fit to both curves.

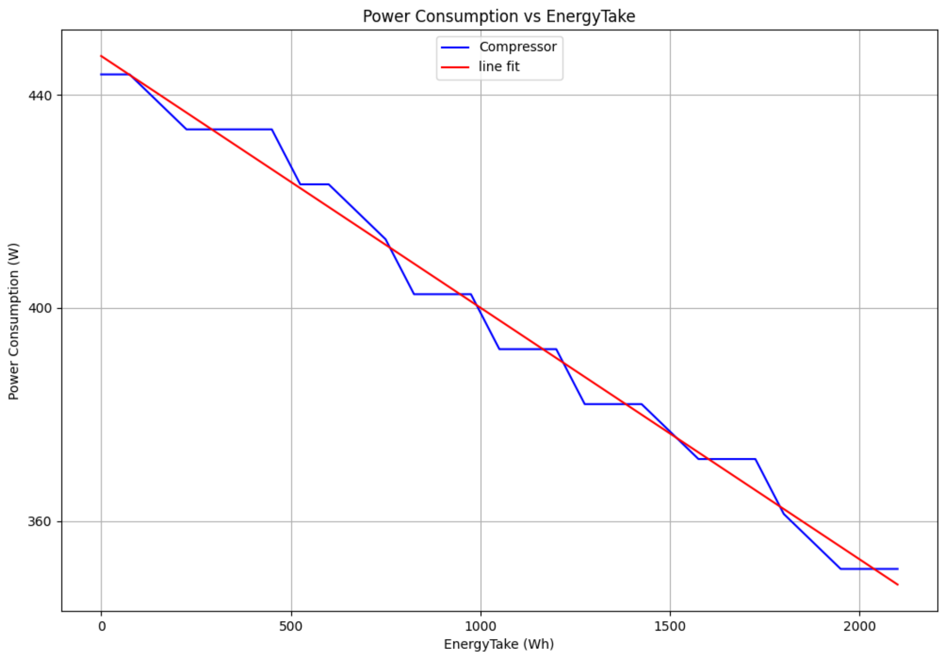


Figure 15: HPWH Watts vs EnergyTake: Vacation Mode Recovery

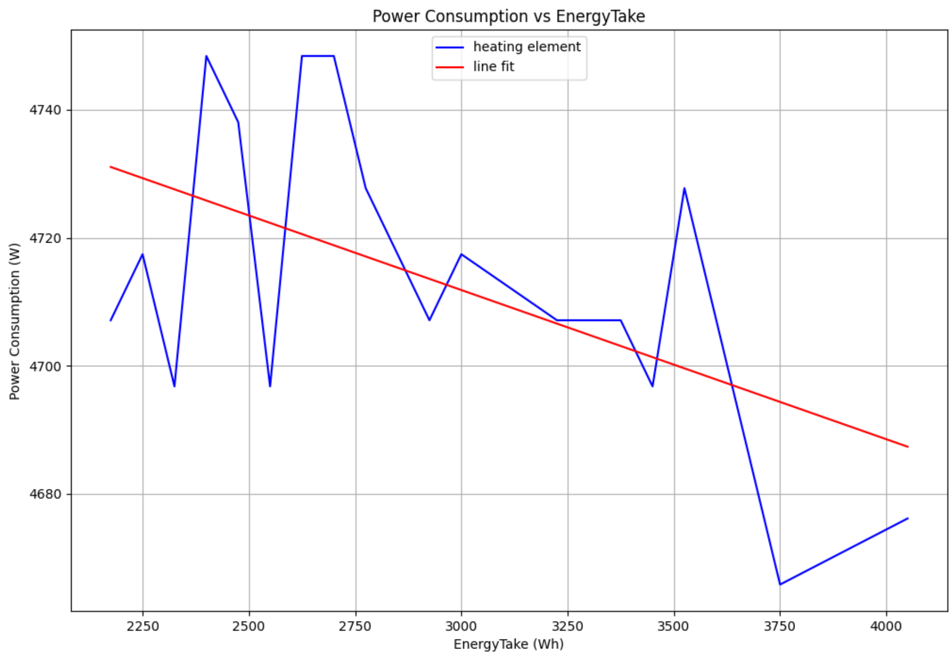


Figure 16: HPWH Watts vs EnergyTake: Vacation Mode Recovery

5.6.6 HPWH Idle Losses

To test the change in *EnergyTake* when the HPWH is idle, a *CPE* command was sent and no water draws were applied. The HPWH was allowed to cool down all the way to 1600 Wh as shown in Figure 17.

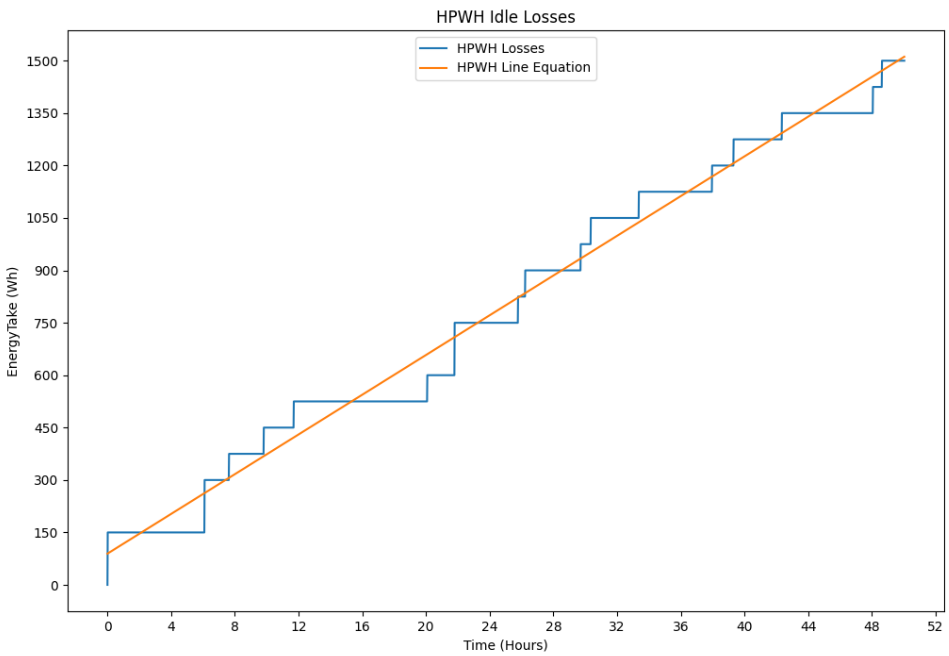


Figure 17: HPWH EnergyTake vs Time: Vacation Mode Recovery

5.6.7 Power Consumption (W) and *EnergyTake* (Wh) Validation Test

The compressor behavior is modeled using the following equation:

$$P(ET) = -0.04729 \times ET + 447.3 \tag{12}$$

The heating element behavior is modeled using the following equation:

$$P(ET) = -0.02331 \times ET + 4782 \tag{13}$$

Both of these equations were tested against the EMCB data. Their behavior is shown in Figure 17.

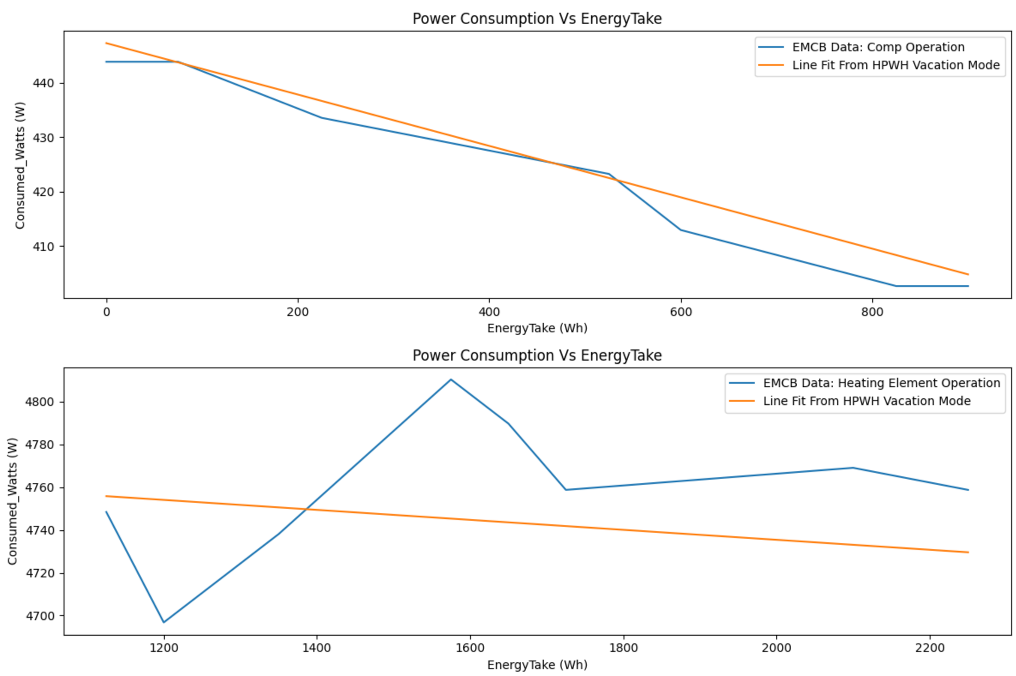


Figure 18: HPWH Watts vs EnergyTake: Validation with EMCB Data

5.6.8 HPWH Idle Losses Validation Test

The idle losses equation was also tested agains the EMCB data. Since the equation was created from a operiod of two days, the equaiton did not perform well with the testing data. Therefore, a modification factor was added to the equation. The modification factor was obtained upon different testings. Figure 19 shows the behavior of Equation 15 when tested with the EMCB data.

The original equation:

$$E(time) = 0.4737 \times time + 89.36 \tag{14}$$

The modified equation:

$$E(time) = 0.8960 \times time + 126 \tag{15}$$

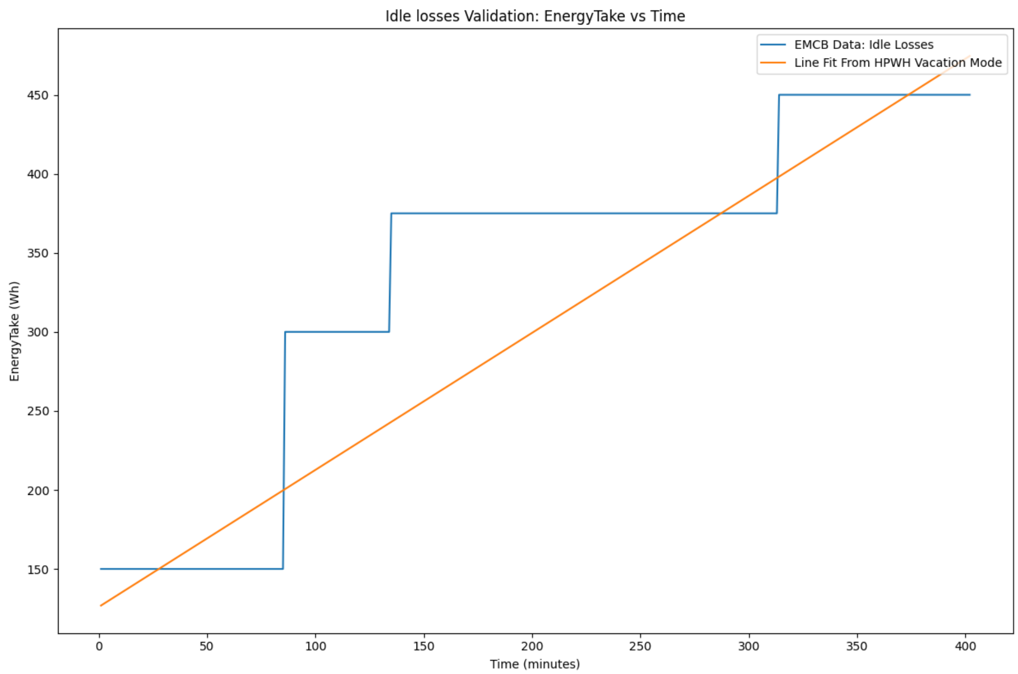


Figure 19: HPWH Idle losses: Validation with EMCB Data

5.6.9 *EnergyTake* as a function of Water Draw Volume

As water draw occurs, the temperature in the tank changes due to influx water. The influx water is the water entering the tank due to water leaving the tank. The influx water temperature is usually around $60°F$. In this test, I'm trying to measure the temperature of the tank as a 20 gallon water volume enters the tank and gets mixed with 30 gallon water in the tank.

As the cold water enters the tank, the heat in the hot water transfers to the cold water and vice-versa. This process continues to happen until both water volumes reach the same temperature. In other words, the hot water loses heat and cold water gains heat. This can be interperted from the first law of thermodynamics: Energy Conservation. This process can be expressed mathematically as shown in equation 16, where Q_{lost} and Q_{gain} are expressed as shown in equations 17 and 18. Equating equations 17 and 18 and solving for T_{New}, we end up with equation 19

$$Q_{lost} = Q_{gained} \tag{16}$$

$$Q_{lost} = V_{WaterTank}(gal) \times \rho_{water}(\frac{lb}{gal}) \times C_p(\frac{Btu}{lb.F}) \times (T_{Setpoint} - T_{New}) \tag{17}$$

$$Q_{gain} = V_{WaterDraw}(gal) \times \rho_{water}(\frac{lb}{gal}) \times C_p(\frac{Btu}{lb.F}) \times (T_{New} - T_{init}) \tag{18}$$

$$T_{New} = \frac{(V_{WaterTank} \times T_{Setpoint}) + (V_{Draw} \times T_{inlet})}{V_{draw} + V_{WaterTank}} \tag{19}$$

Equation 19 approximates the temperature of the tank after both water volumes get mixed together. As the HPWH model detects the drop in the temperature, one of the heating elements will trigger to heat the water. The change in temperature as heat added to the tank is calculated as shown in equation 20. Keep in mind that the heat added to the tank is now heating the whole 50 gallon.

$$\Delta T = \frac{(Q_{added}(btu))}{(C_p(\frac{Btu}{lb.F}) \times V_{WaterTank}(gal) \times \rho_{water}(\frac{lb}{gal}))} \tag{20}$$

Where Q_added is as follows:

$$Q_{added}(btu) = P_{consumed}(W) \times T_{step}(h) * 3.41 \tag{21}$$

5.6.10 Testing

The above procedure was tested on one of the EMCB files. The water draw event was 20 gallon. Table 11 shows comparison between the calculated temperature (inferred from ET) and the temperature calculated from the above equations.

Table 11: Temperature Comparison

Original Temp	Calc_Temp
97.029769	101.557377
98.059539	102.786885
101.148847	105.860656
103.208385	106.475410
104.238155	107.090164
107.327463	108.934426
109.387001	110.163934
110.416770	110.778689
112.242271	112.622951
112.698646	113.237705
114.372021	115.081967
115.589021	115.696721
117.870896	118.155738
119.087896	119.385246
120.000646	119.779253

6 Winter 2022

6.1 GridLAB-D Model Validation

THIS SECTION AND FORWARD NEEDS TO BE REWRITTEN AND EXPLAINED. FOR NOW, I'M POSTING IMAGES HERE TO PRESENT IN THE TECH MEETING

6.1.1 HPWH Water Demand Behavior

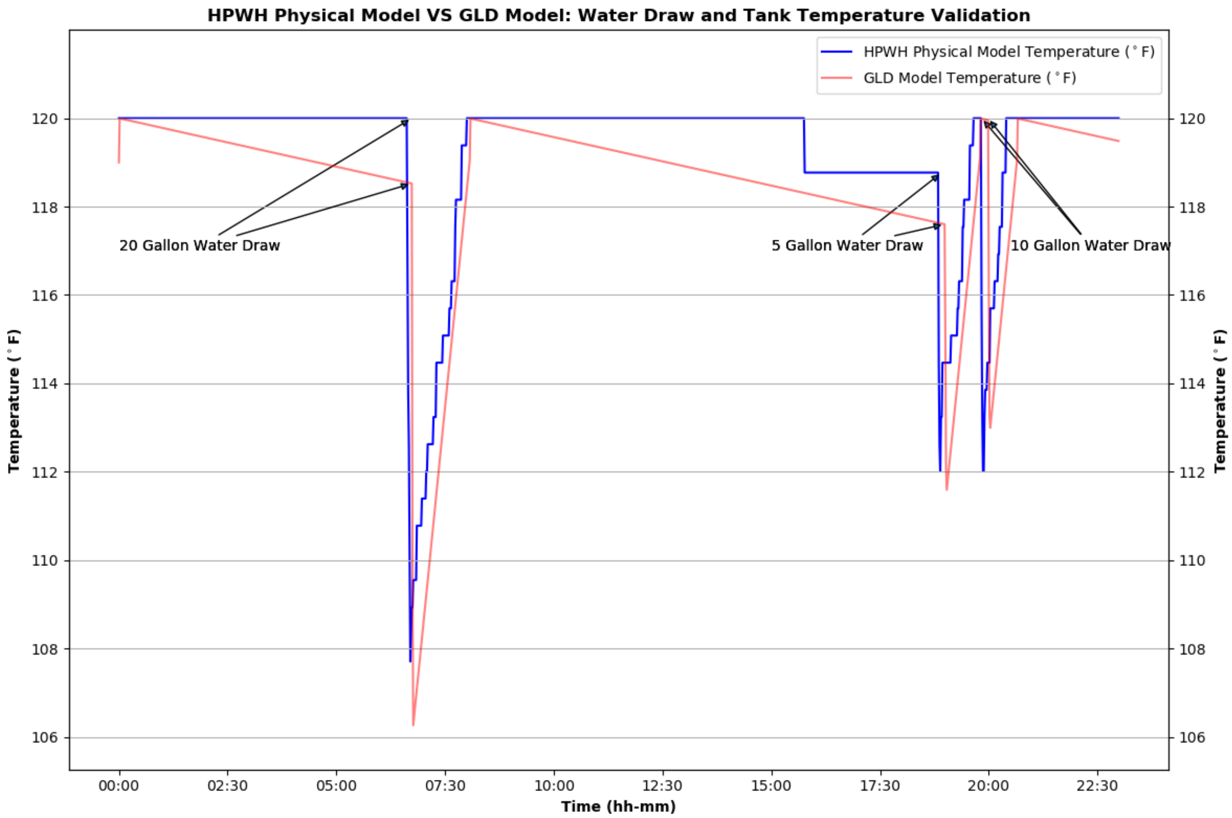


Figure 20: GLD and Physical Model Validation: Water Draw Validation

6.1.2 Idle Losses

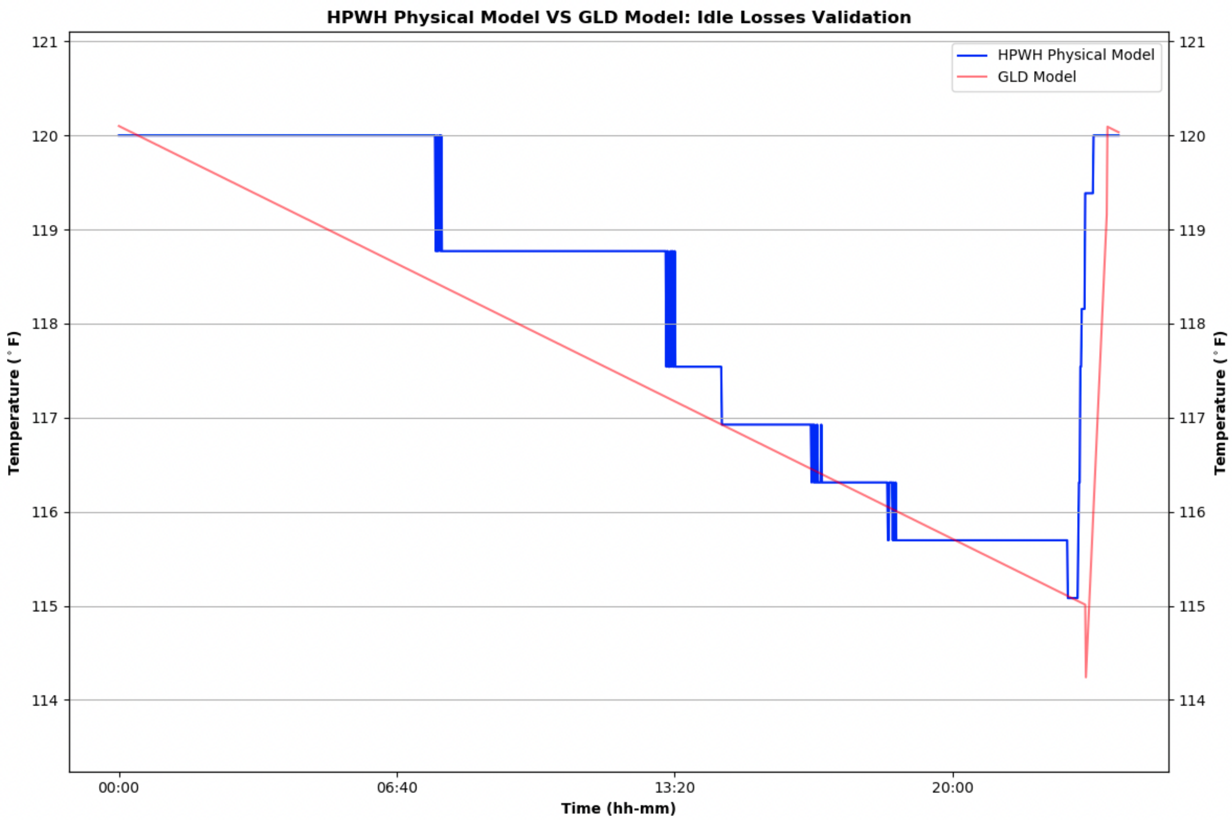


Figure 21: GLD and Physical Model Validation: Idle Losses

6.1.3 Heating Sources Switching

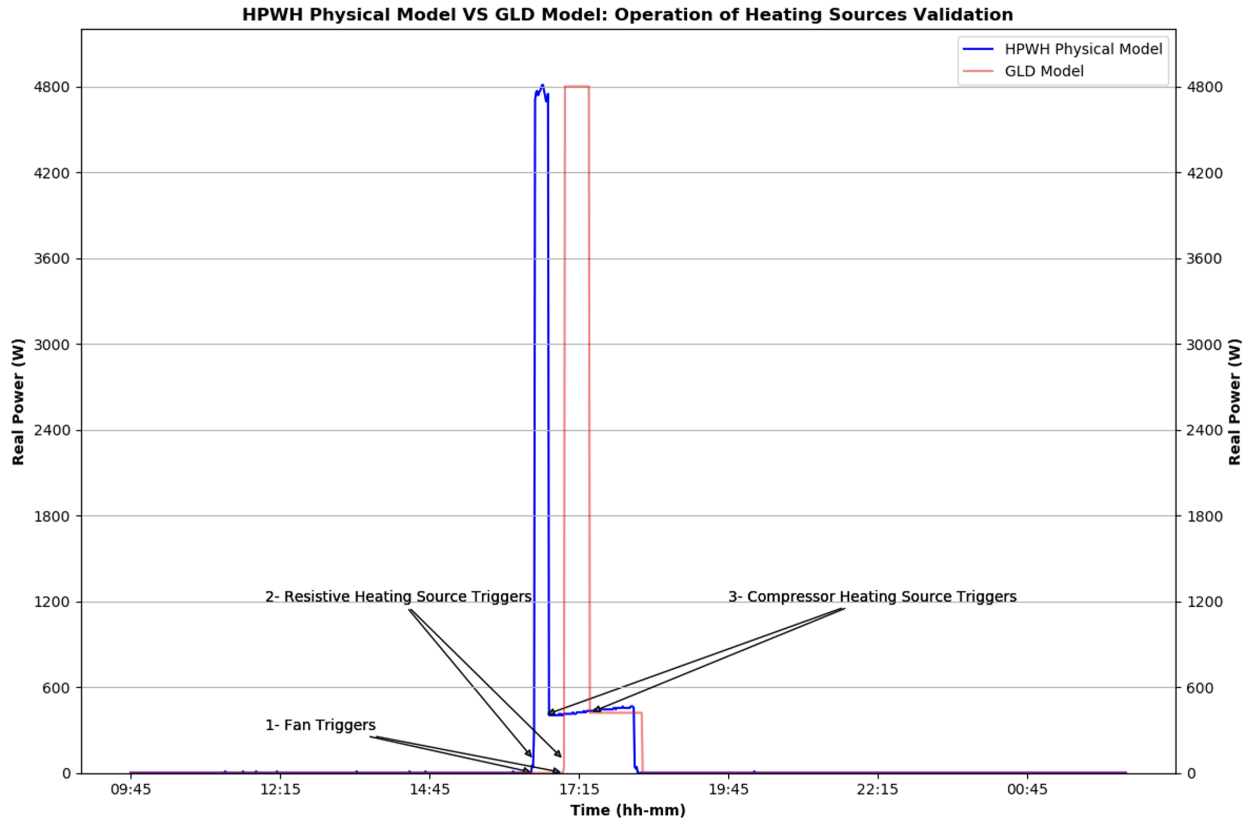


Figure 22: GLD and Physical Model Validation: Heating Sources Switching

6.1.4 Data Samples

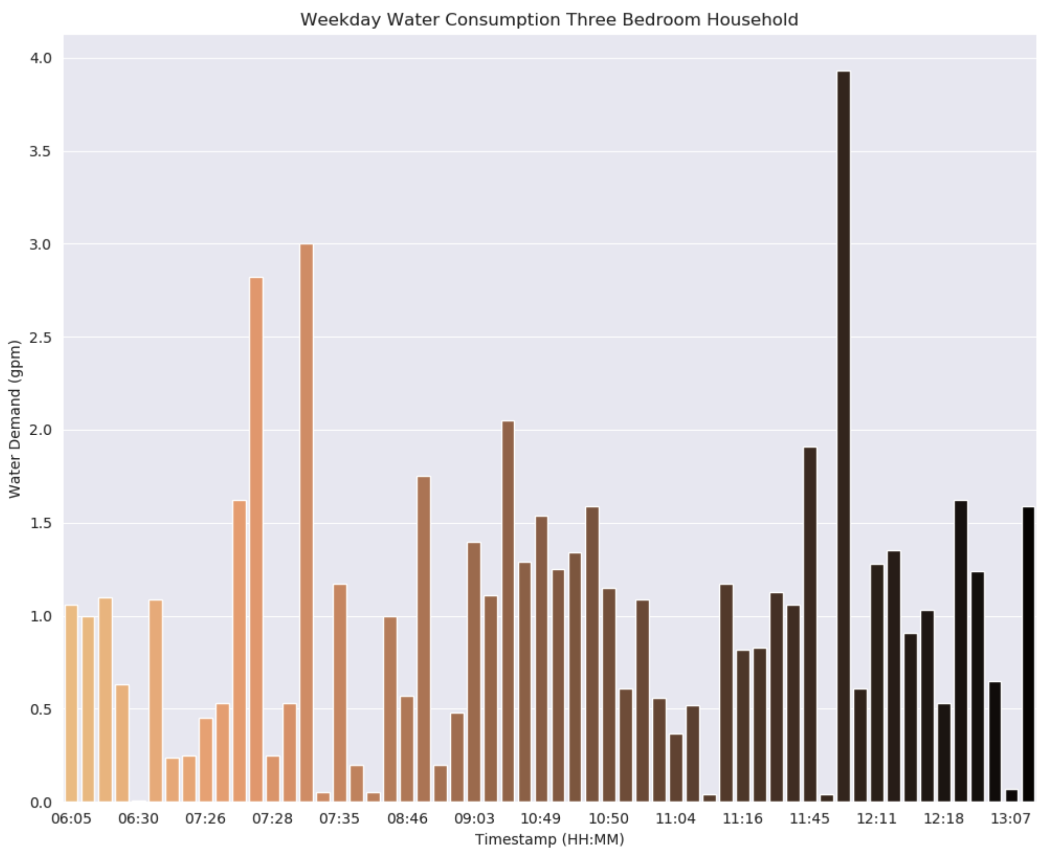


Figure 23: Water Demand Sample

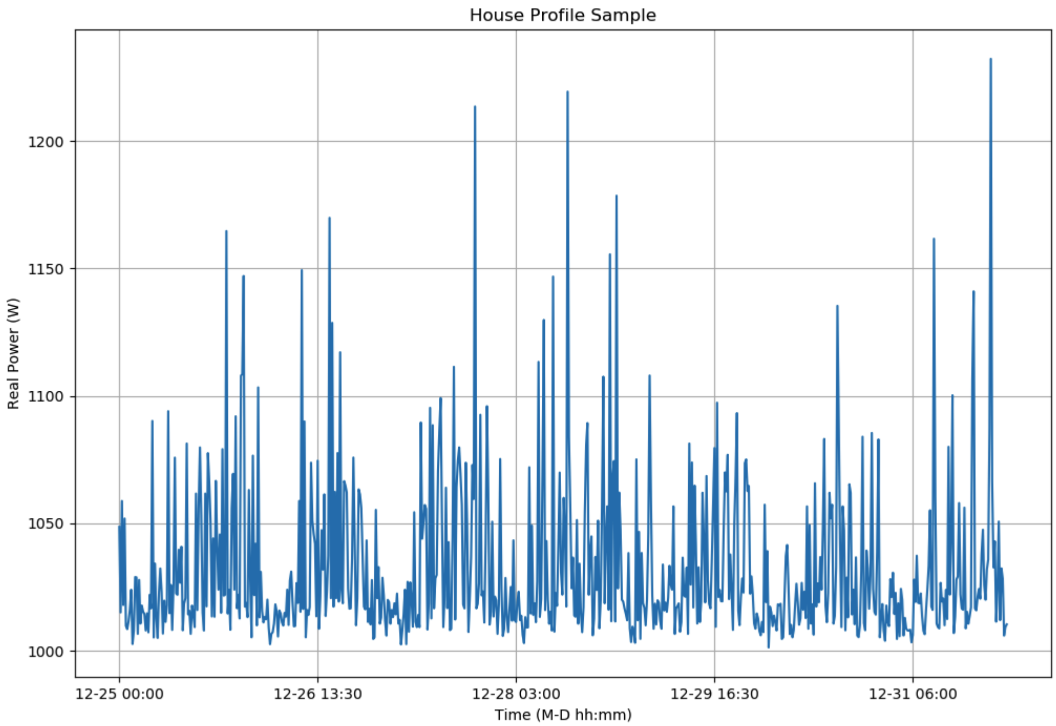
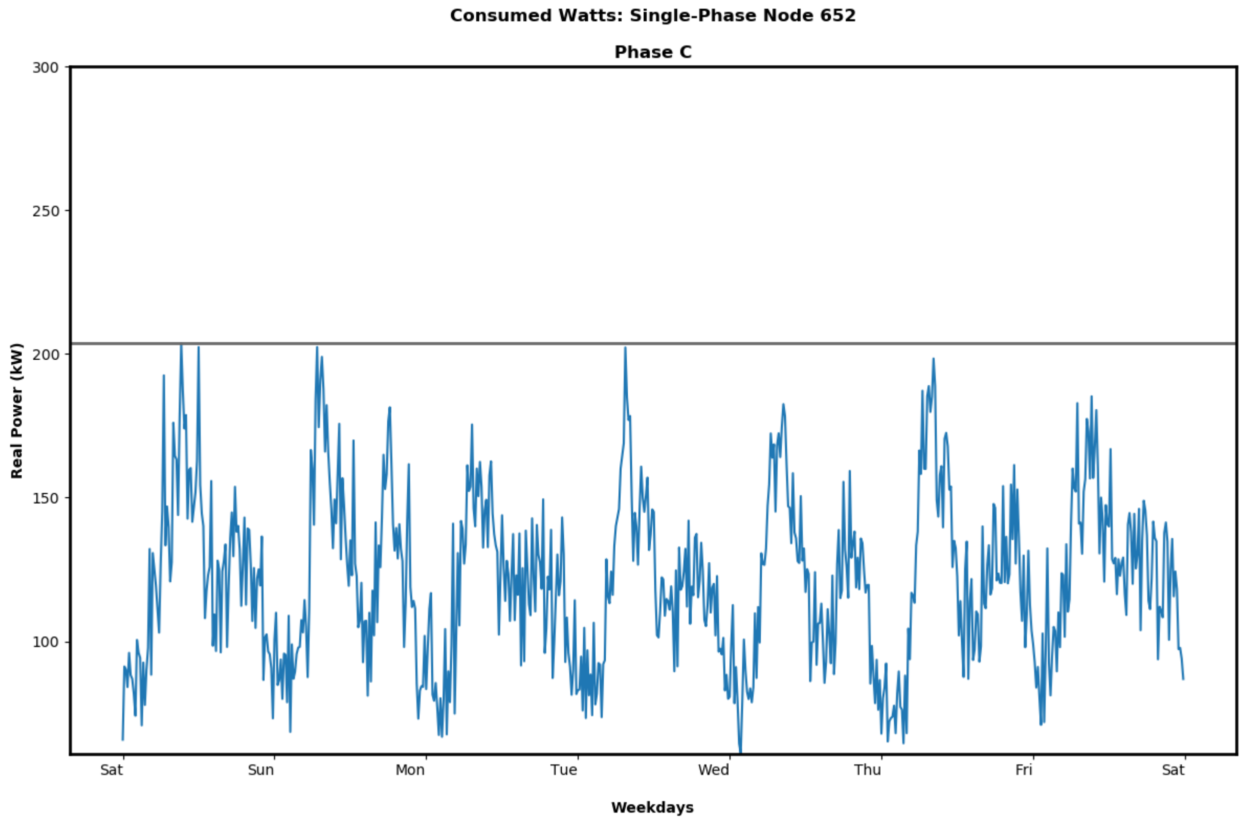

House Profile Sample
Real Power (W)
1000
1050
1100
1150
1200
12-25 00:00
12-26 13:30
12-28 03:00
12-29 16:30
12-31 06:00
Time (M-D hh:mm)

Figure 24

// ==



Consumed Watts: Single-Phase Node 652
Phase C
Real Power (kW)
300
250
200
150
100
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 25

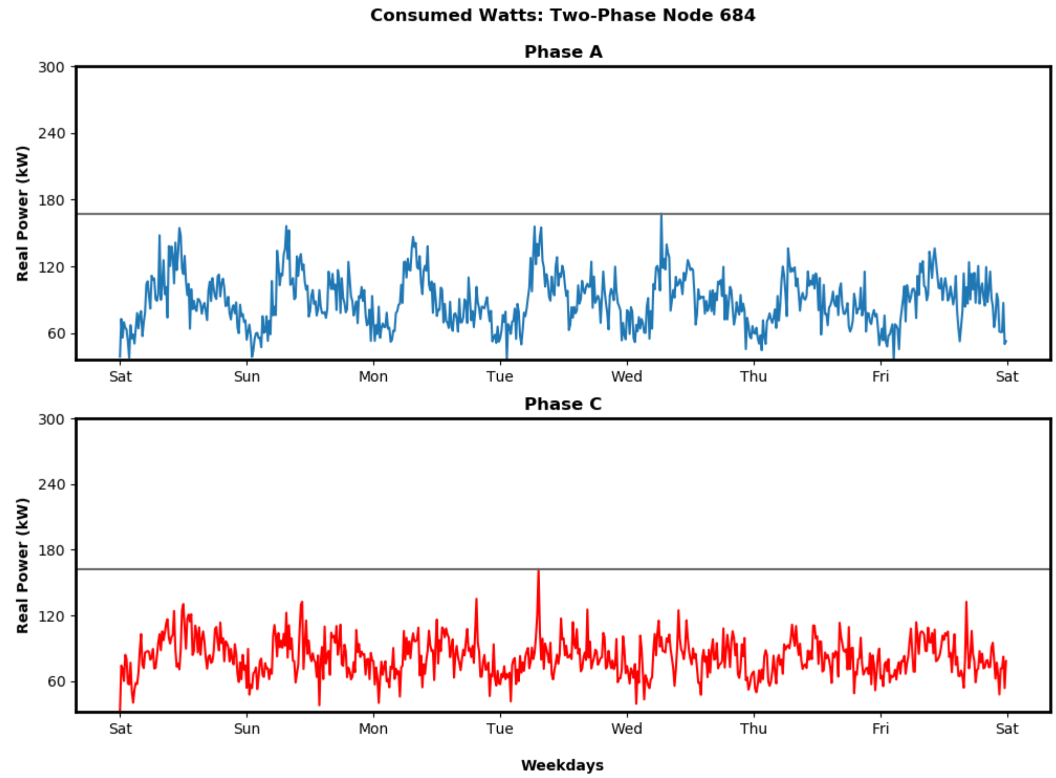

Consumed Watts: Two-Phase Node 684
Phase A
Real Power (kW)
300
240
180
120
60
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Phase C
Real Power (kW)
300
240
180
120
60
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 26

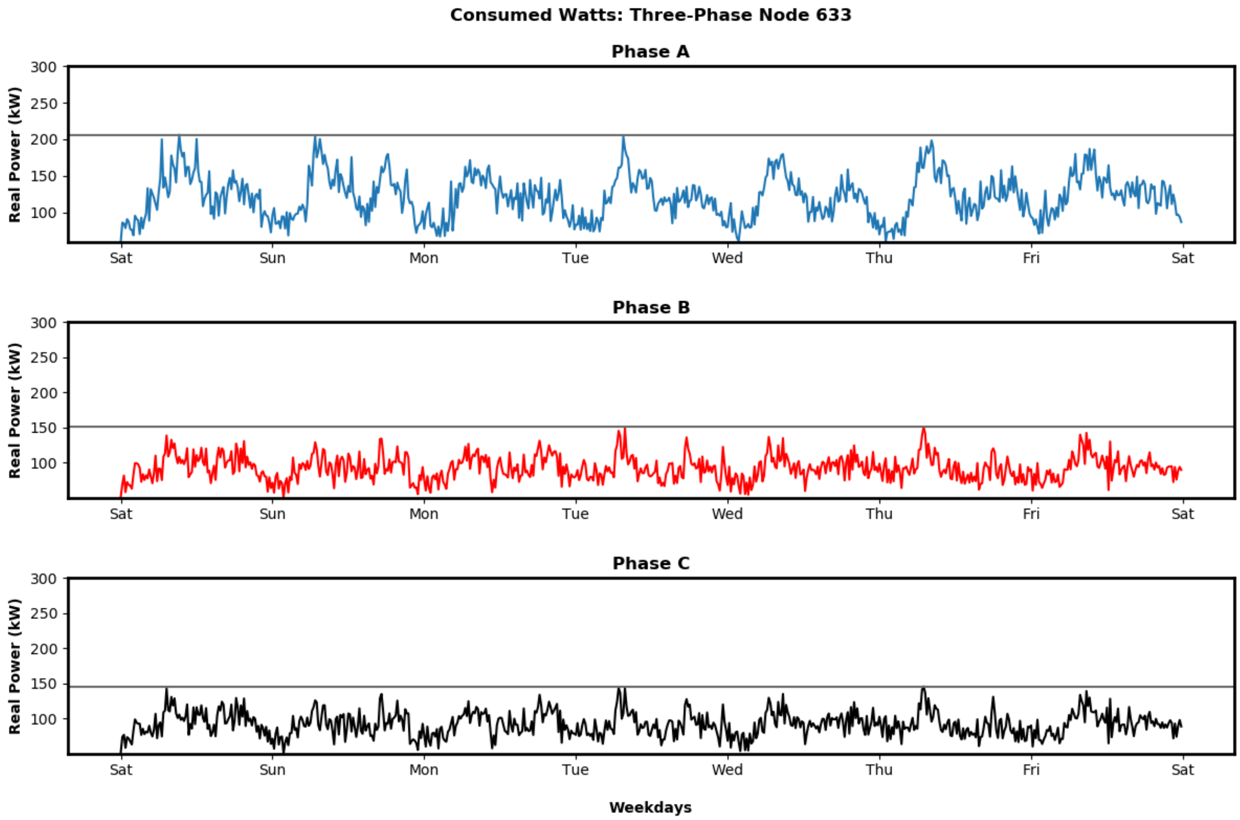

Consumed Watts: Three-Phase Node 633
Phase A
Phase B
Phase C
Real Power (kW)
300
250
200
150
100
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 27

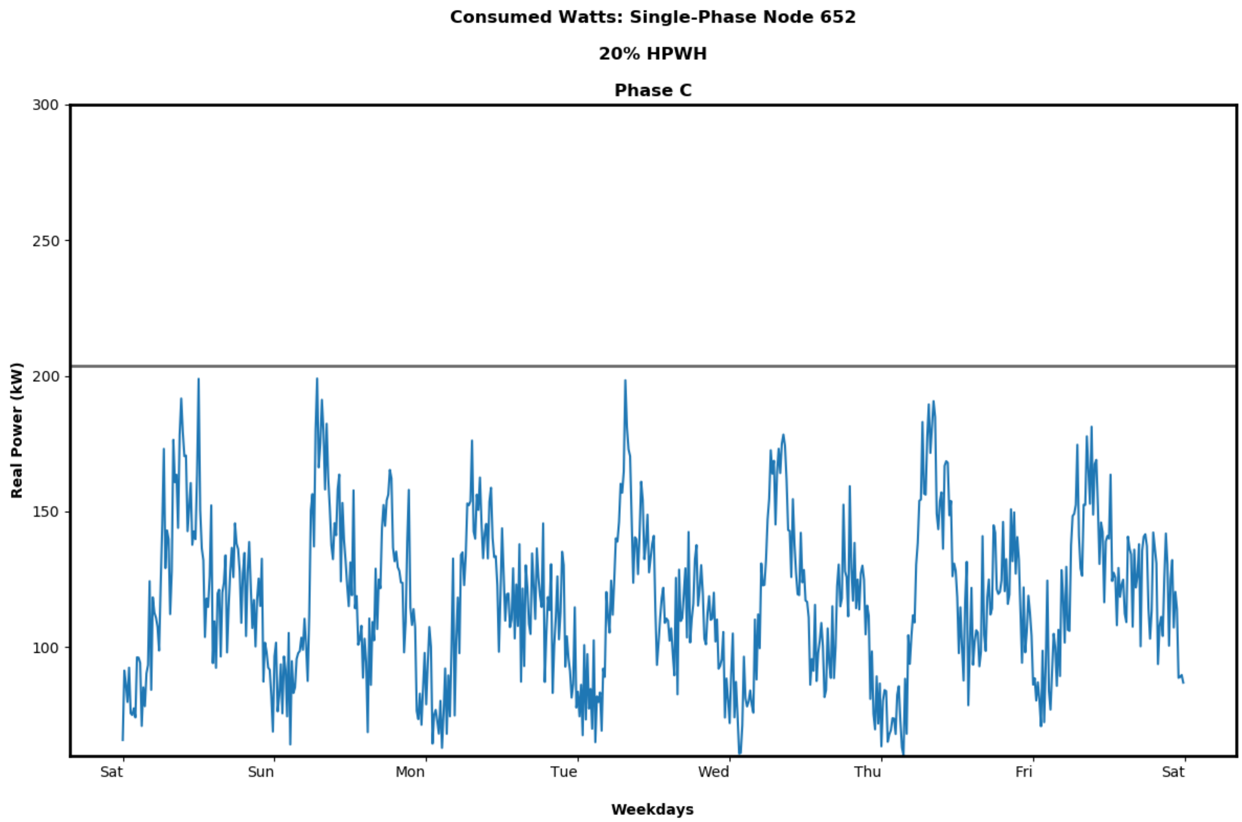

Consumed Watts: Single-Phase Node 652
20% HPWH
Phase C
Real Power (kW)
300
250
200
150
100
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 28

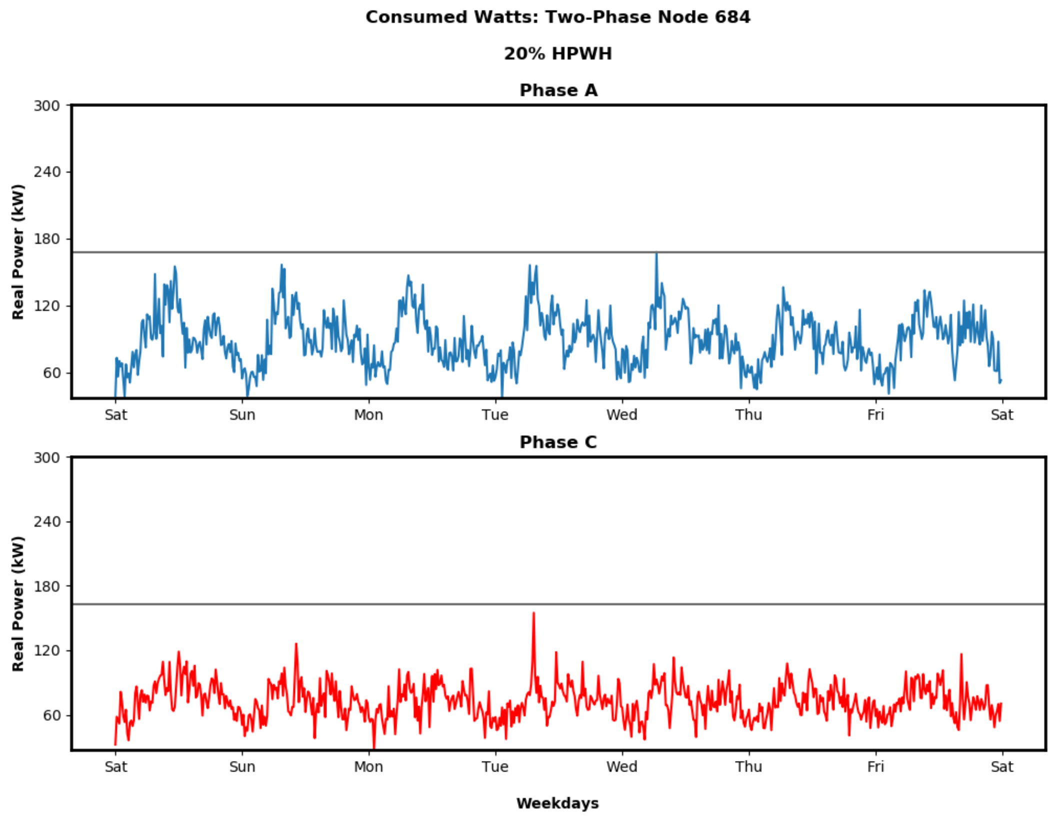

Consumed Watts: Two-Phase Node 684
20% HPWH
Phase A
Real Power (kW)
300
240
180
120
60
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Phase C
Real Power (kW)
300
240
180
120
60
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 29

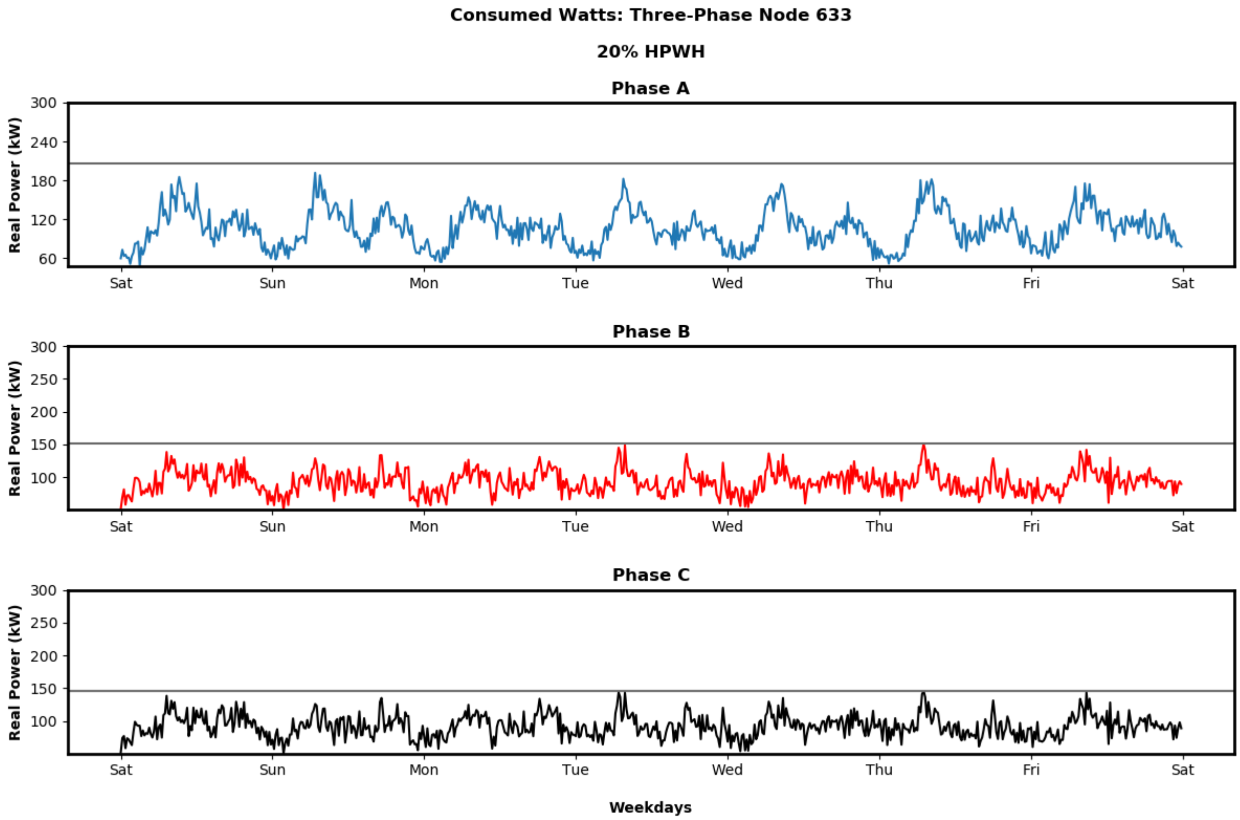

Consumed Watts: Three-Phase Node 633
20% HPWH
Phase A
Real Power (kW)
300
240
180
120
60
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Phase B
Real Power (kW)
300
250
200
150
100
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Phase C
Real Power (kW)
300
250
200
150
100
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 30

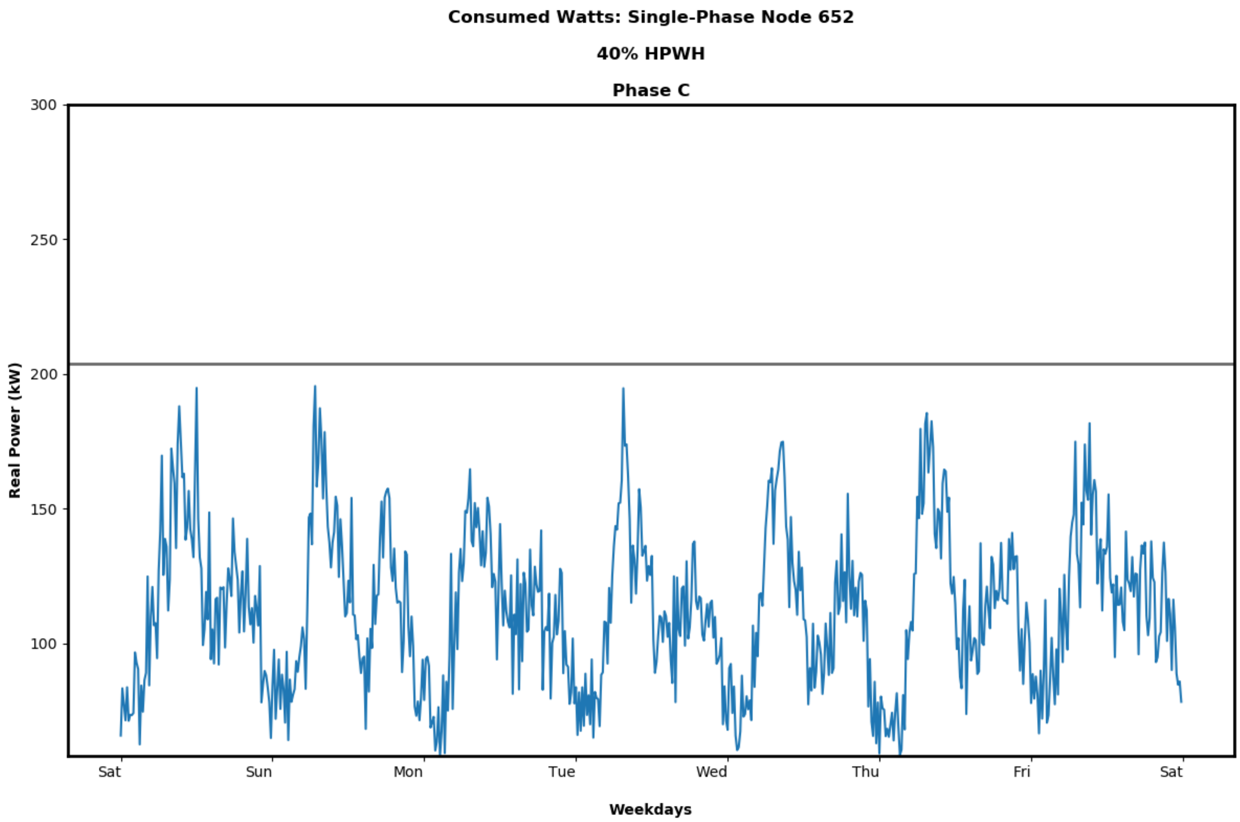

Consumed Watts: Single-Phase Node 652
40% HPWH
Phase C
300
250
200
150
100
Real Power (kW)
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 31

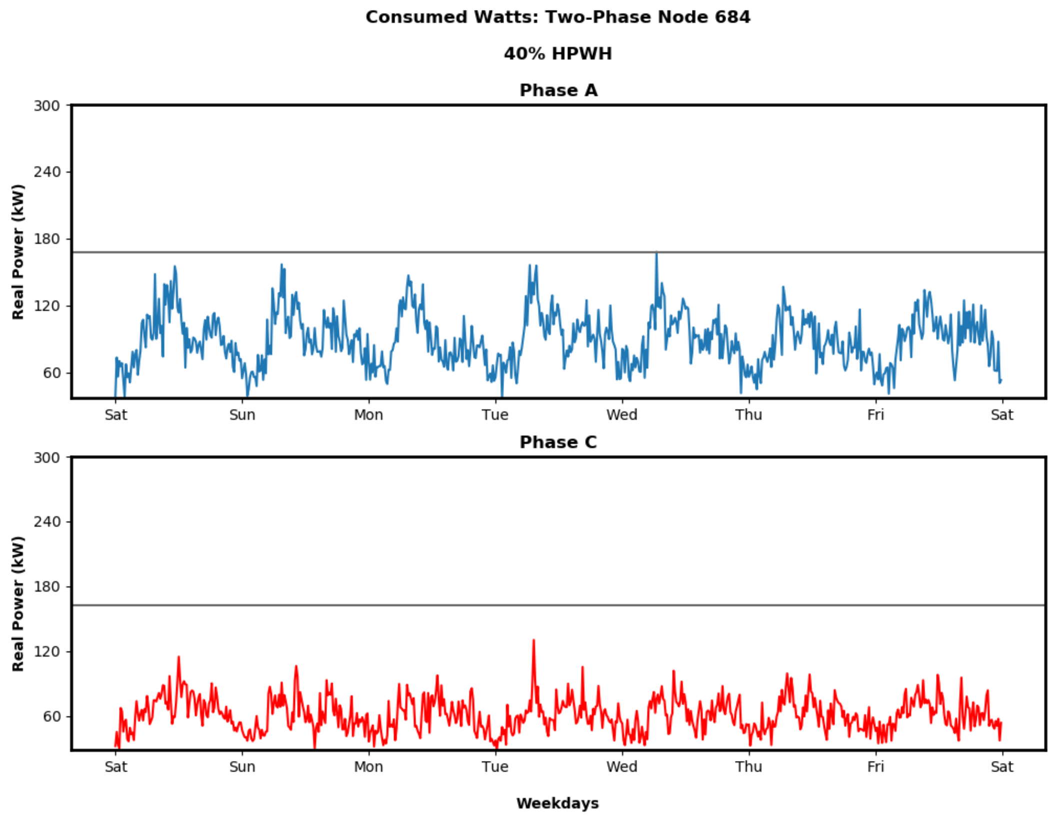

Consumed Watts: Two-Phase Node 684
40% HPWH
Phase A
Real Power (kW)
300
240
180
120
60
Sat Sun Mon Tue Wed Thu Fri Sat
Phase C
Real Power (kW)
300
240
180
120
60
Sat Sun Mon Tue Wed Thu Fri Sat
Weekdays

Figure 32

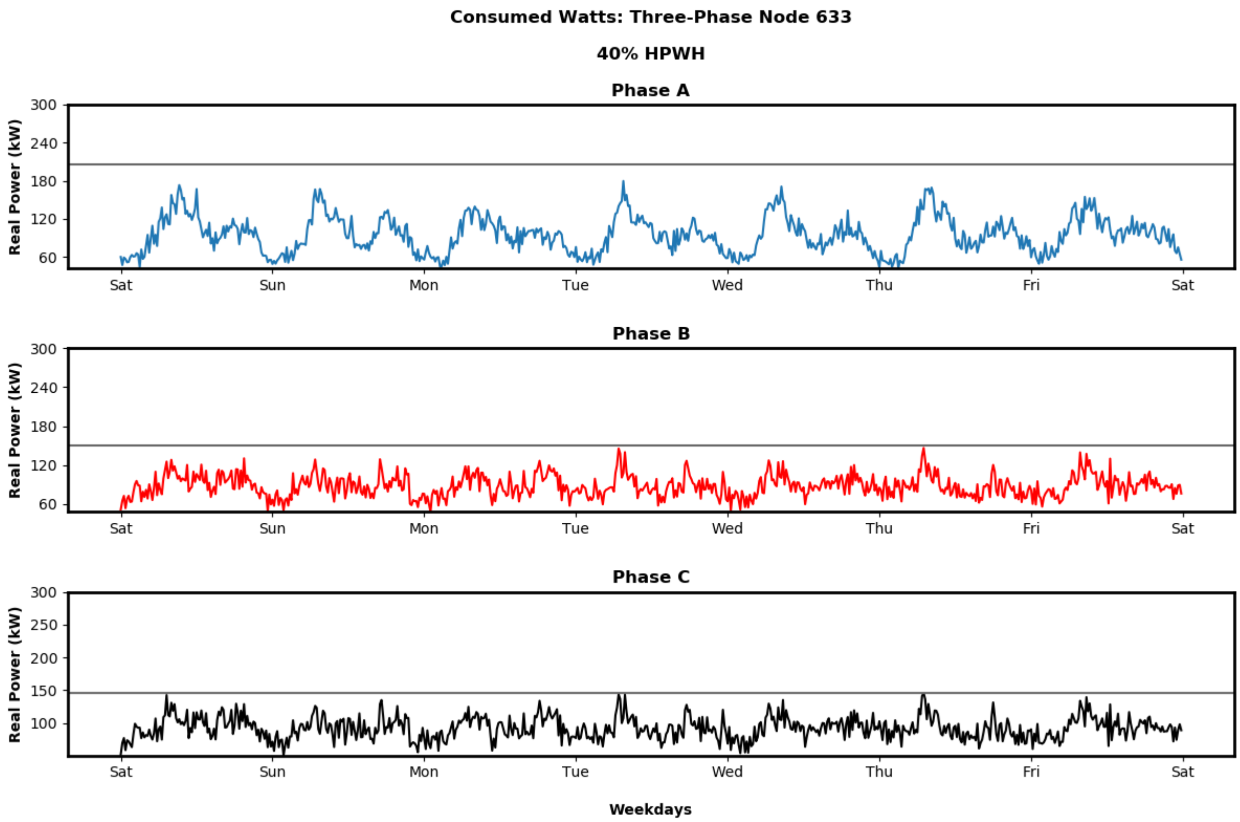

Consumed Watts: Three-Phase Node 633
40% HPWH
Phase A
Real Power (kW)
300
240
180
120
60
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Phase B
Real Power (kW)
300
240
180
120
60
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Phase C
Real Power (kW)
300
250
200
150
100
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 33

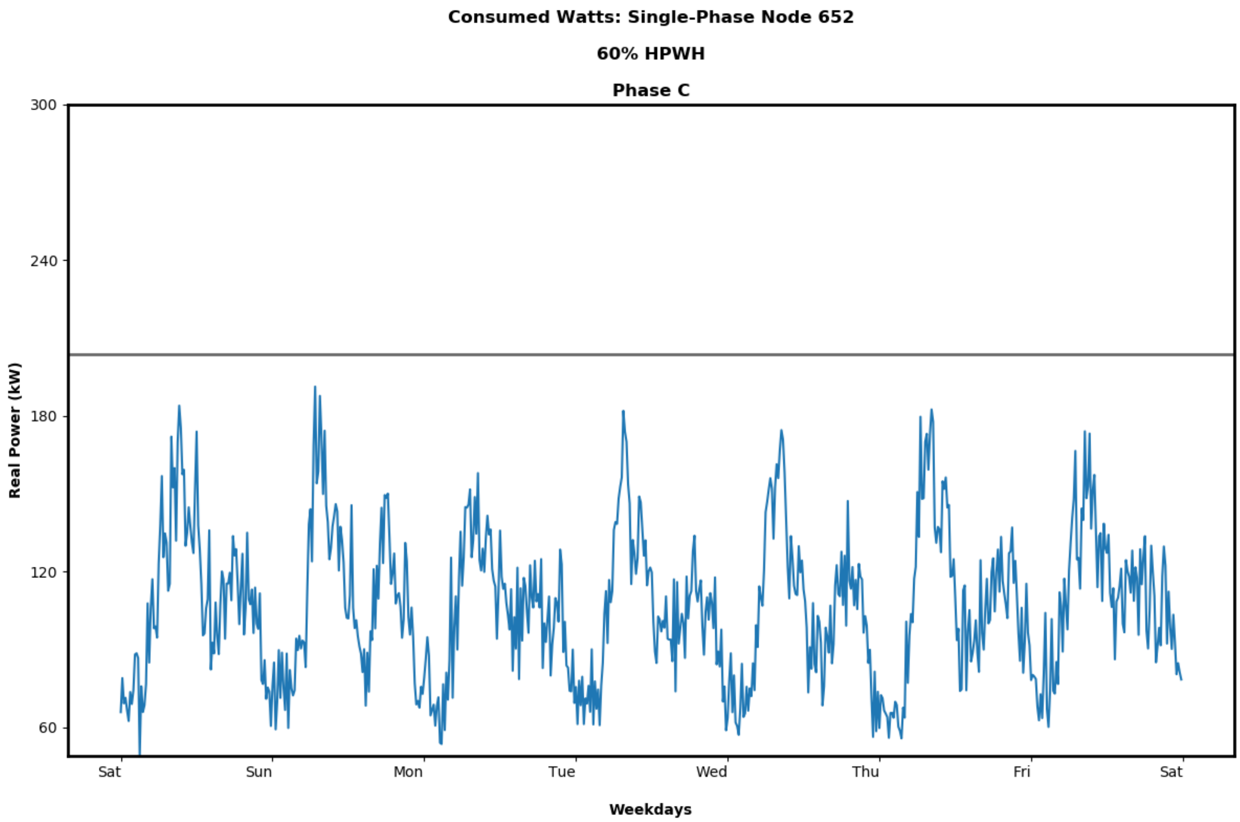

Consumed Watts: Single-Phase Node 652
60% HPWH
Phase C
Real Power (kW)
300
240
180
120
60
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 34

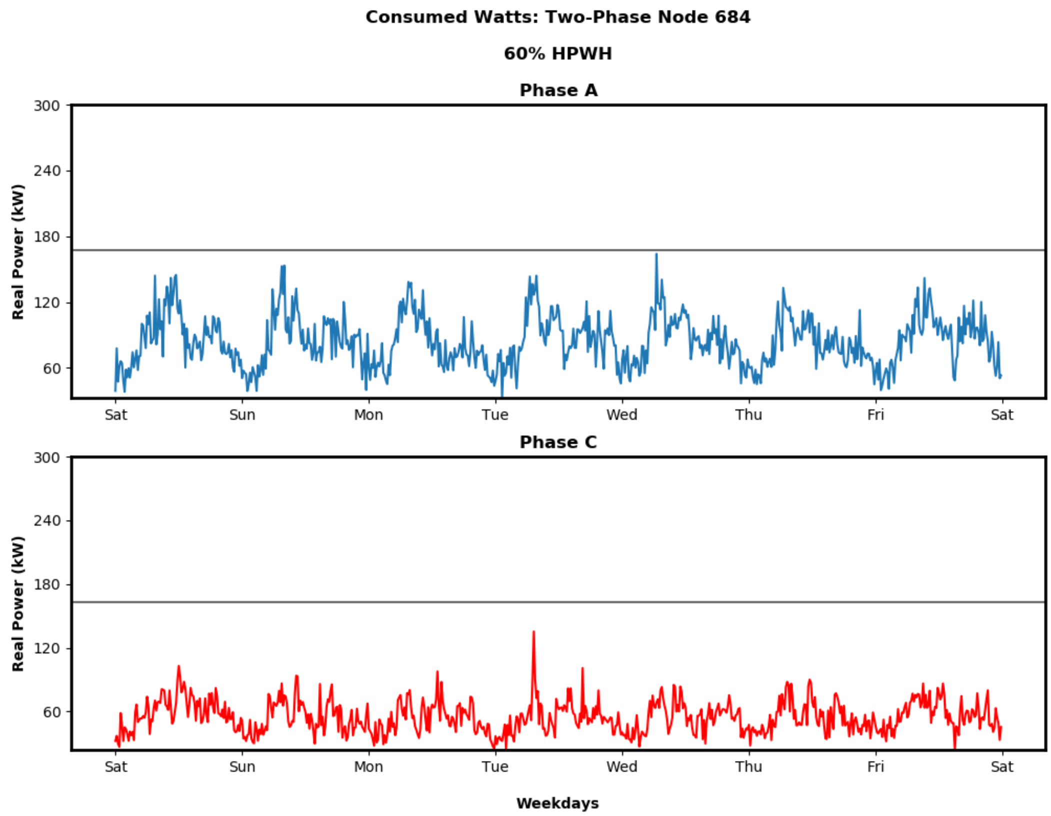

Consumed Watts: Two-Phase Node 684
60% HPWH
Phase A
Real Power (kW)
300
240
180
120
60
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Phase C
Real Power (kW)
300
240
180
120
60
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 35

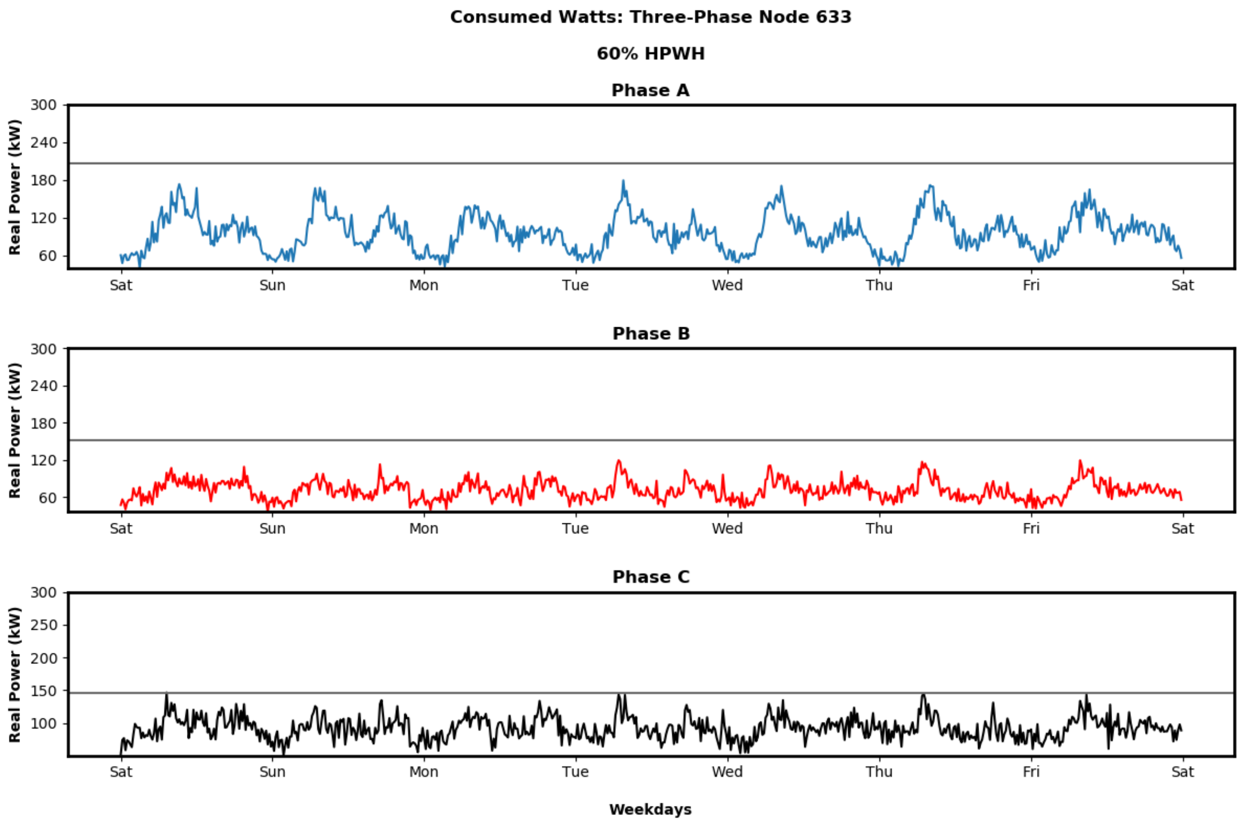

Consumed Watts: Three-Phase Node 633
60% HPWH
Phase A
Real Power (kW)
300
240
180
120
60
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Phase B
Real Power (kW)
300
240
180
120
60
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Phase C
Real Power (kW)
300
250
200
150
100
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 36

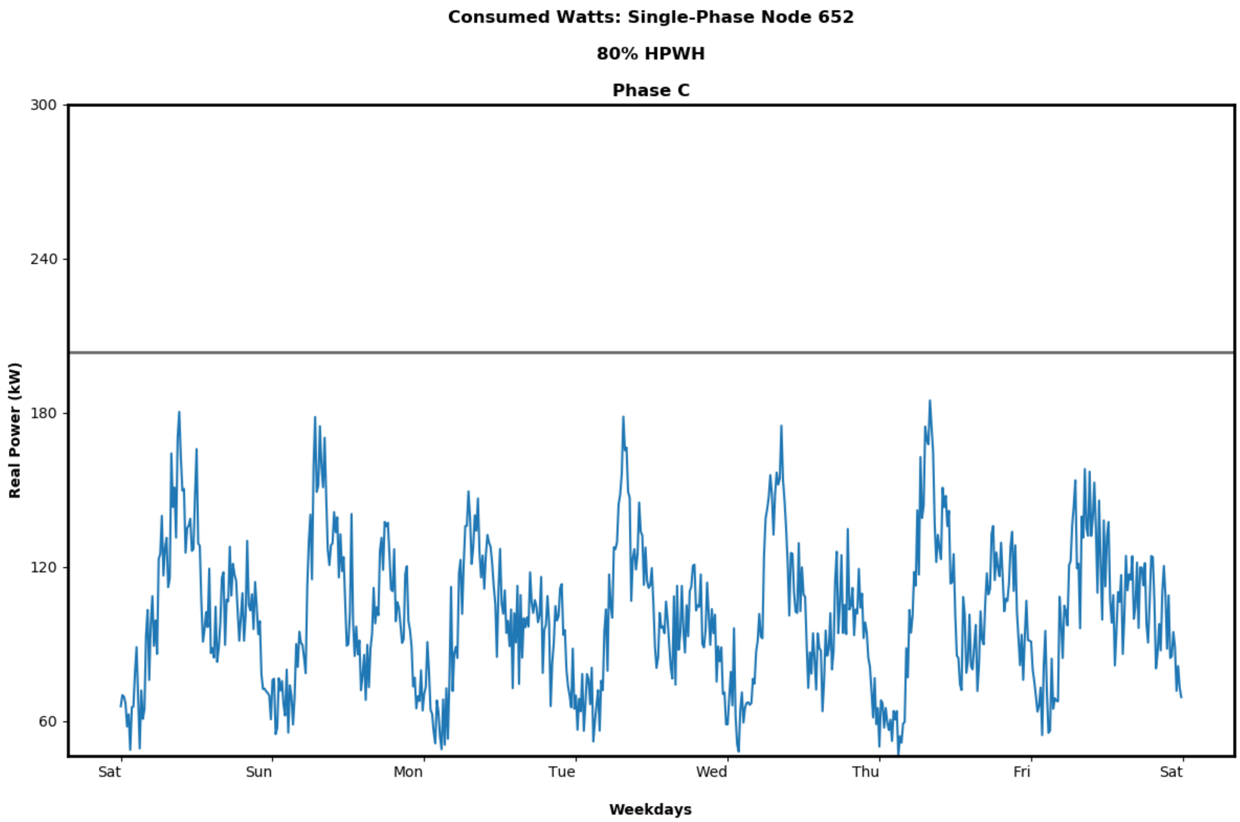

Consumed Watts: Single-Phase Node 652
80% HPWH
Phase C
Real Power (kW)
300
240
180
120
60
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 37

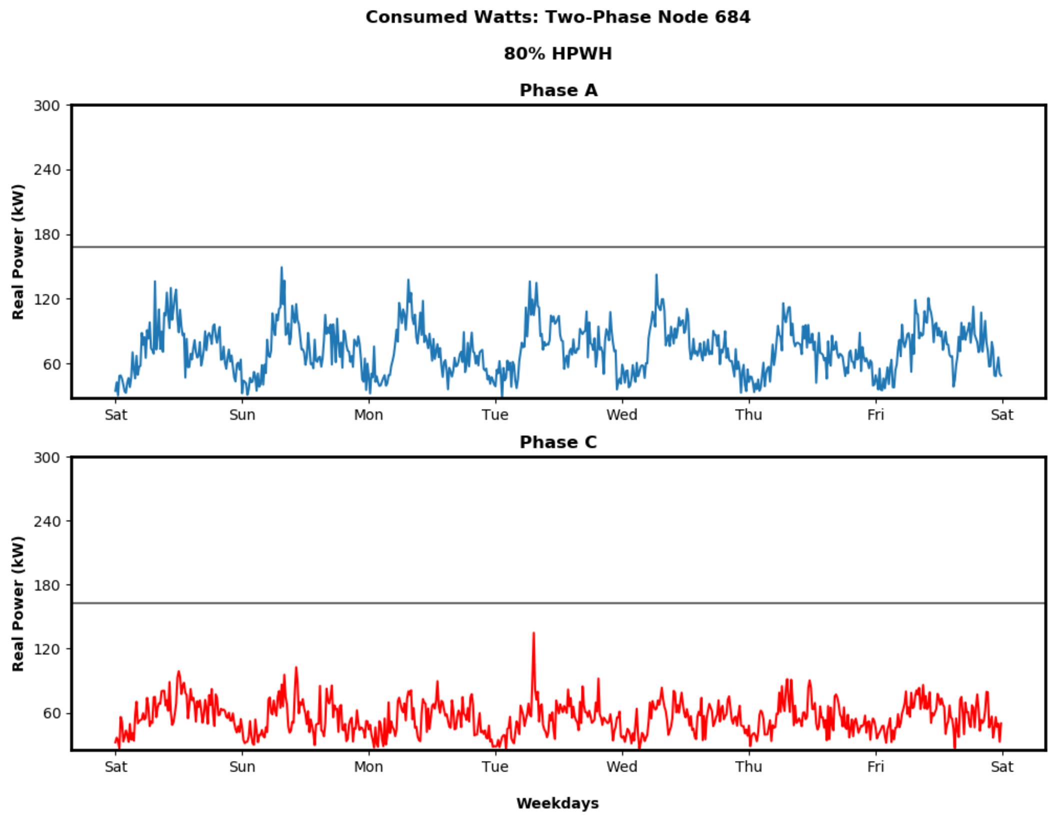

Consumed Watts: Two-Phase Node 684
80% HPWH
Phase A
Real Power (kW)
300
240
180
120
60
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Phase C
Real Power (kW)
300
240
180
120
60
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 38

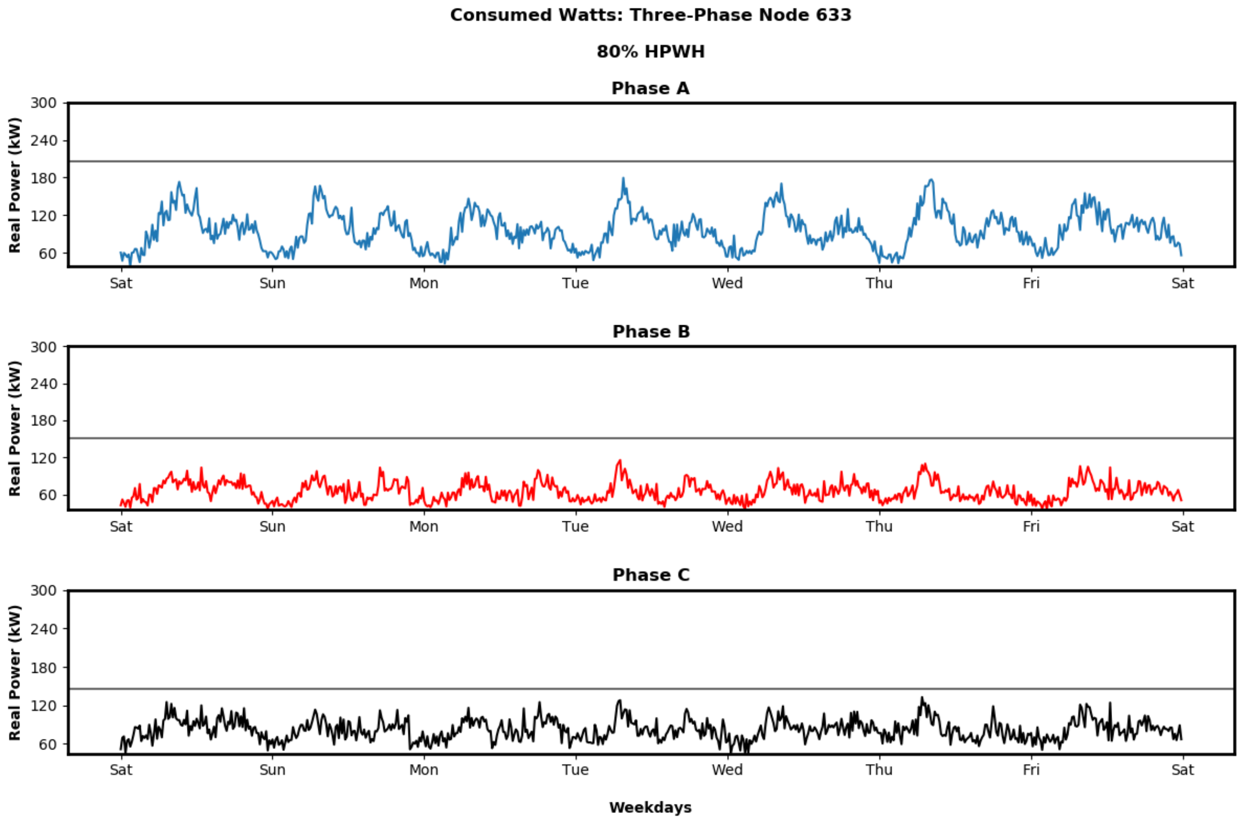

Consumed Watts: Three-Phase Node 633
80% HPWH
Phase A
Phase B
Phase C
Real Power (kW)
300
240
180
120
60
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 39

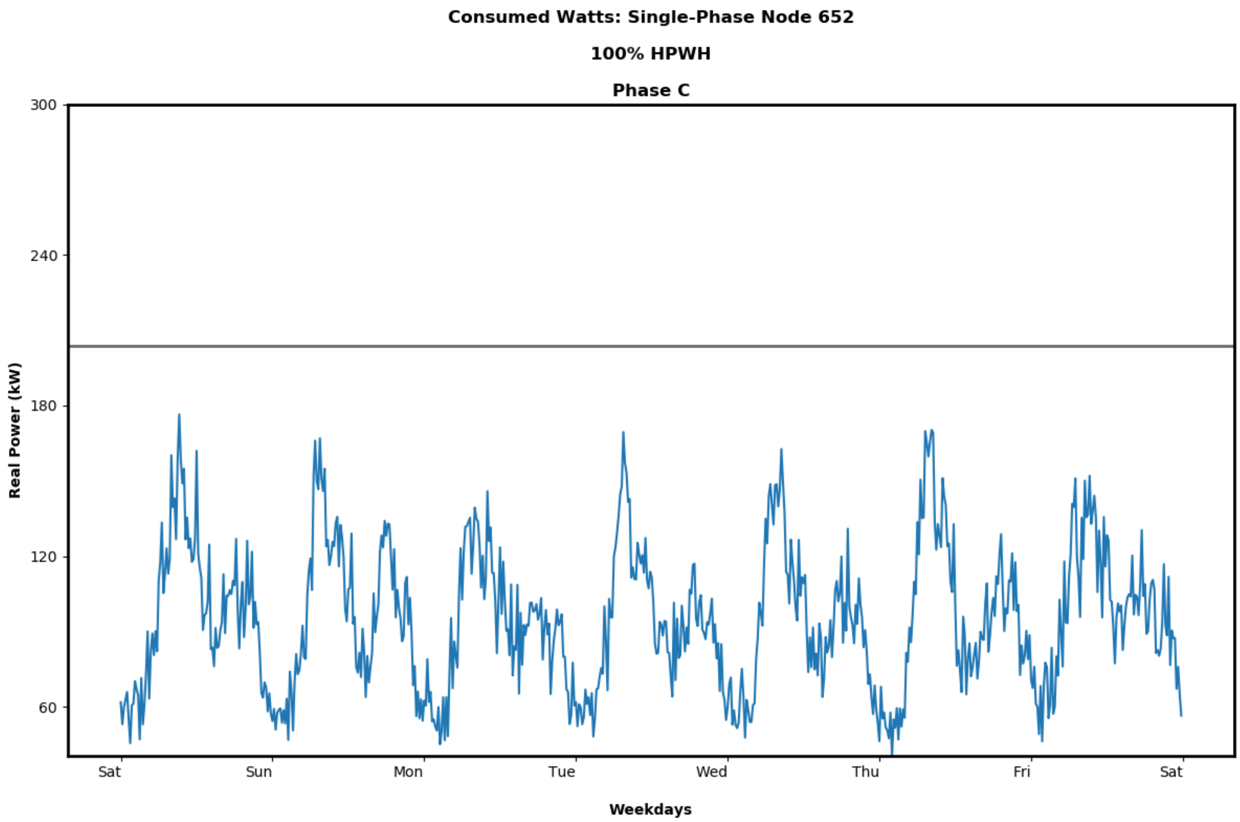

Consumed Watts: Single-Phase Node 652
100% HPWH
Phase C
Real Power (kW)
300
240
180
120
60
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 40

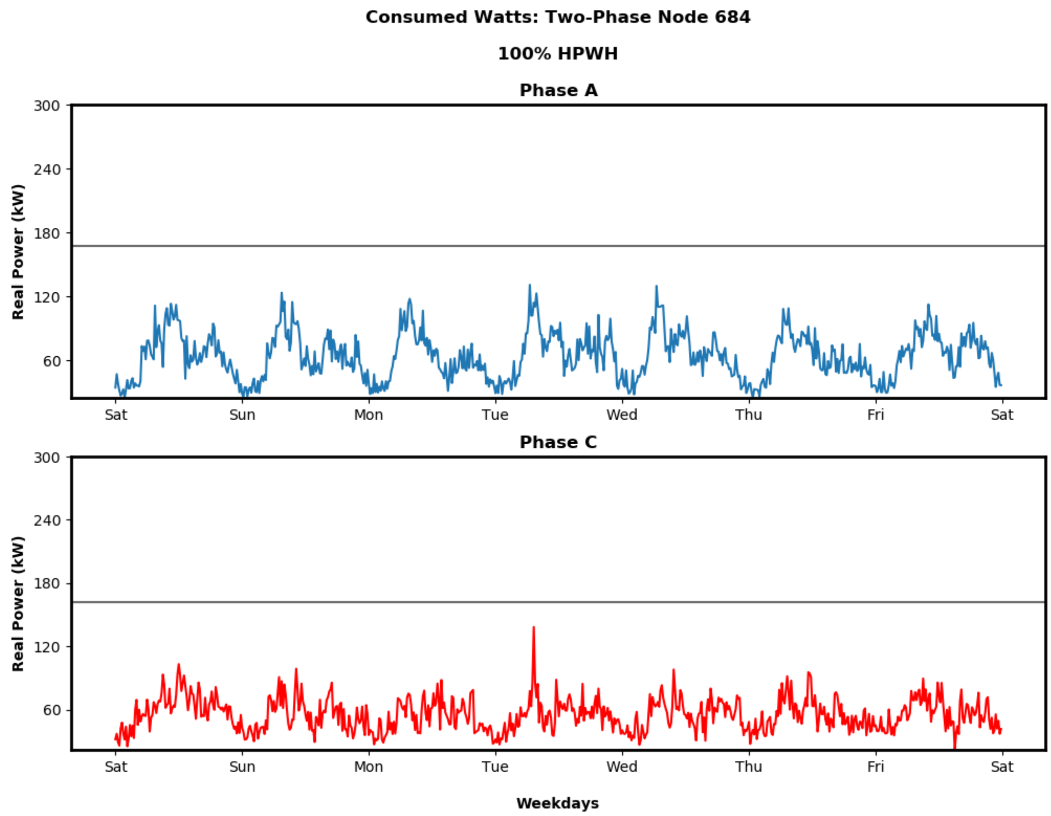

Consumed Watts: Two-Phase Node 684
100% HPWH
Phase A
Real Power (kW)
300
240
180
120
60
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Phase C
Real Power (kW)
300
240
180
120
60
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 41

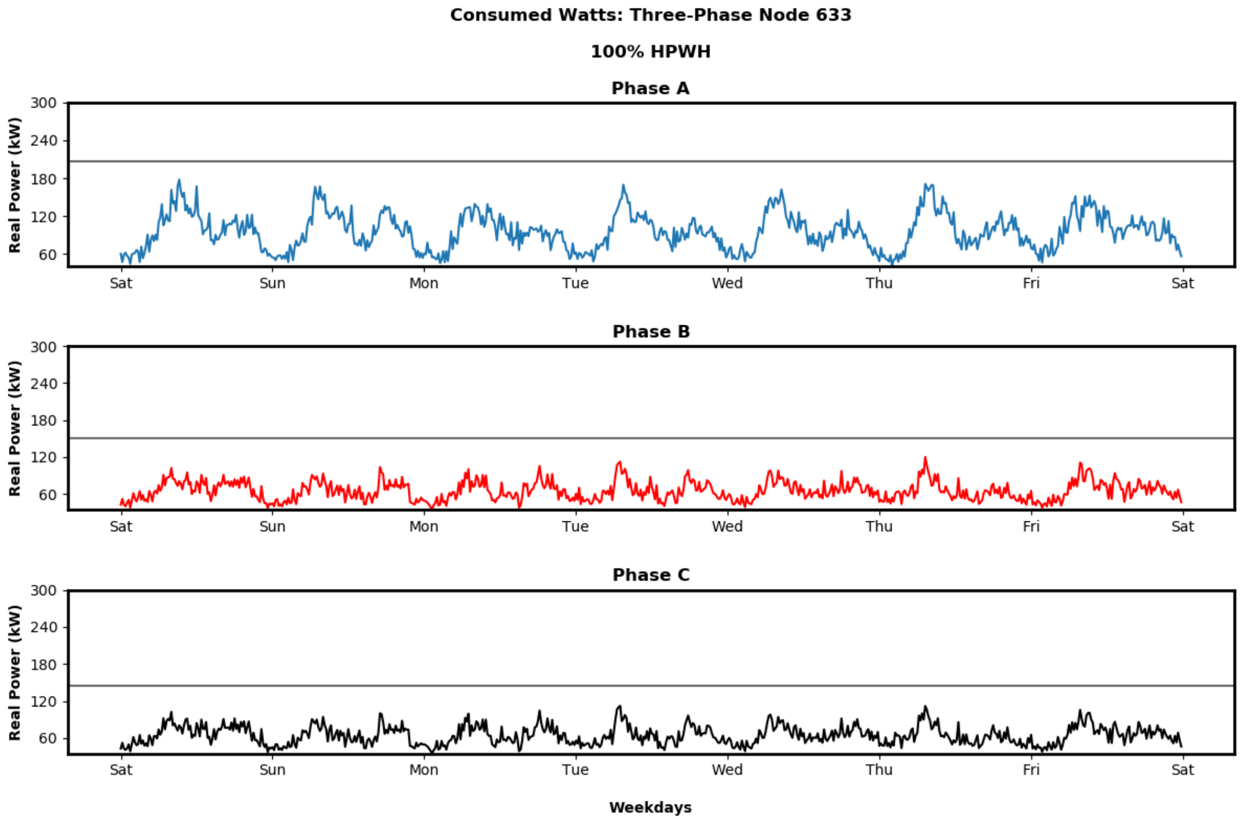

Consumed Watts: Three-Phase Node 633
100% HPWH
Phase A
Phase B
Phase C
Real Power (kW)
300
240
180
120
60
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 42

// ==

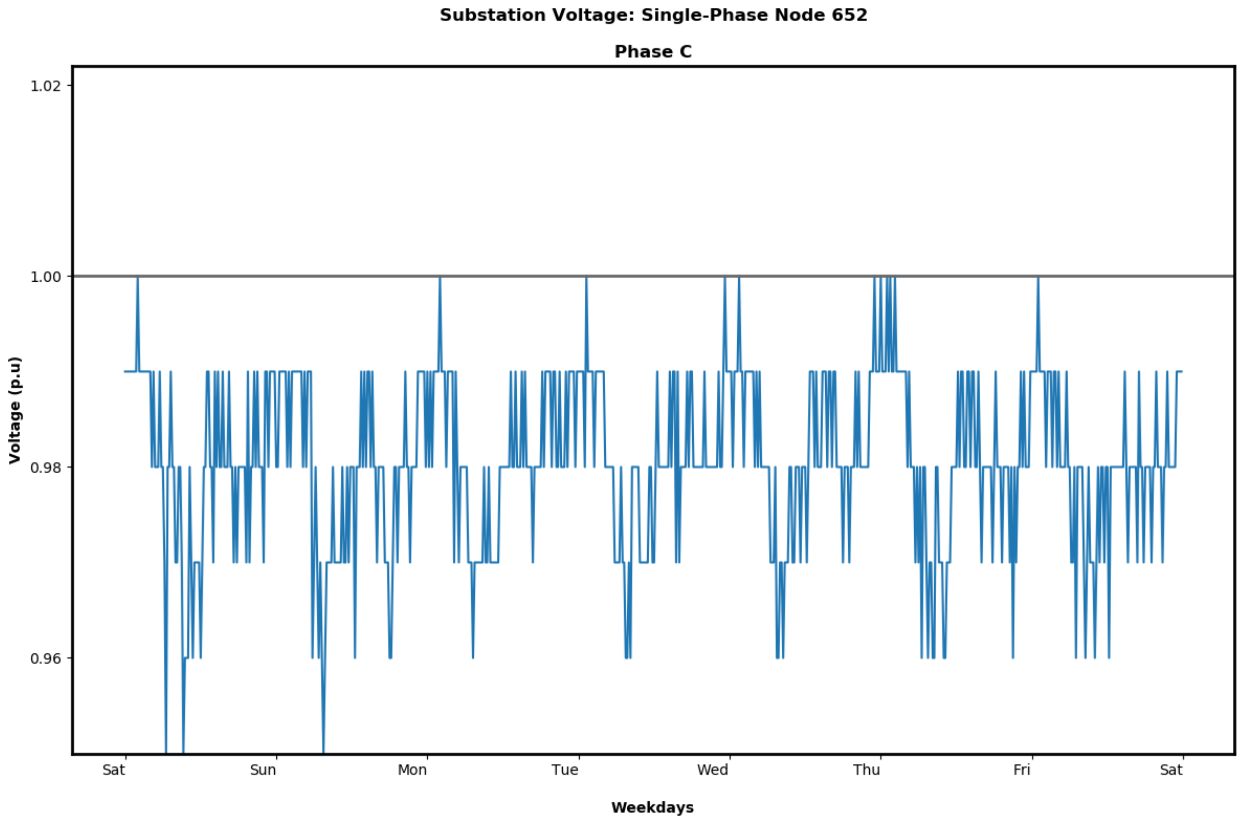

Substation Voltage: Single-Phase Node 652
Phase C
1.02
1.00
0.98
0.96
Voltage (p.u)
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 43

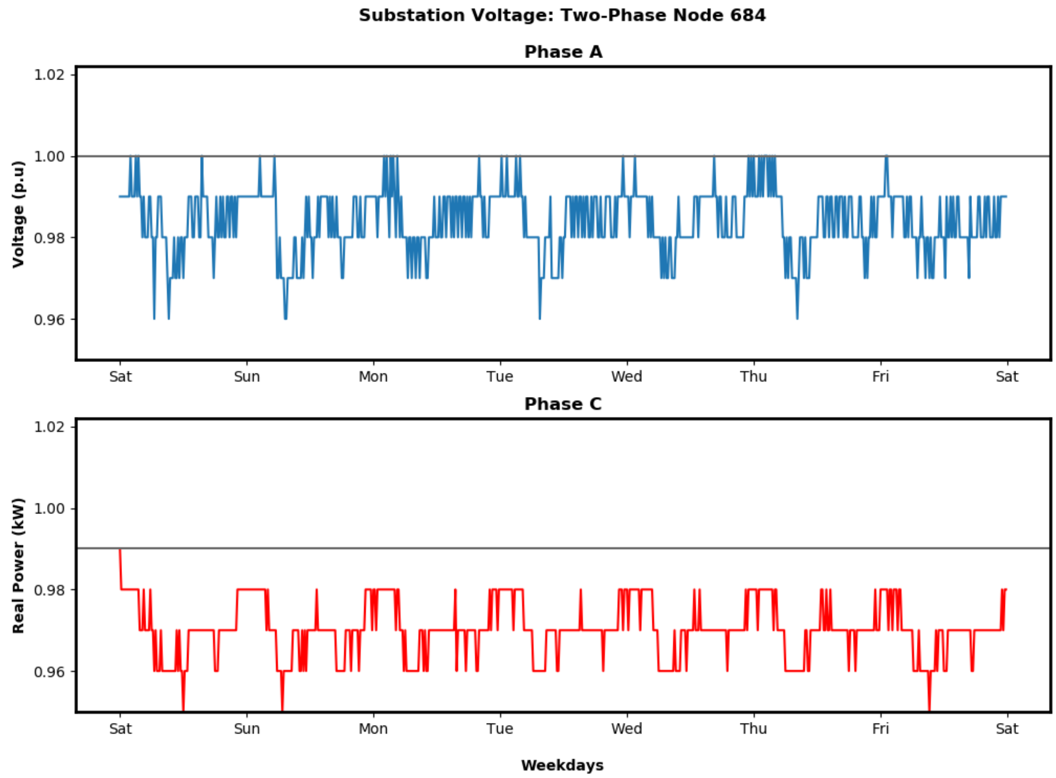

Substation Voltage: Two-Phase Node 684
Phase A
Voltage (p.u)
1.02
1.00
0.98
0.96
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Phase C
Real Power (kW)
1.02
1.00
0.98
0.96
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 44

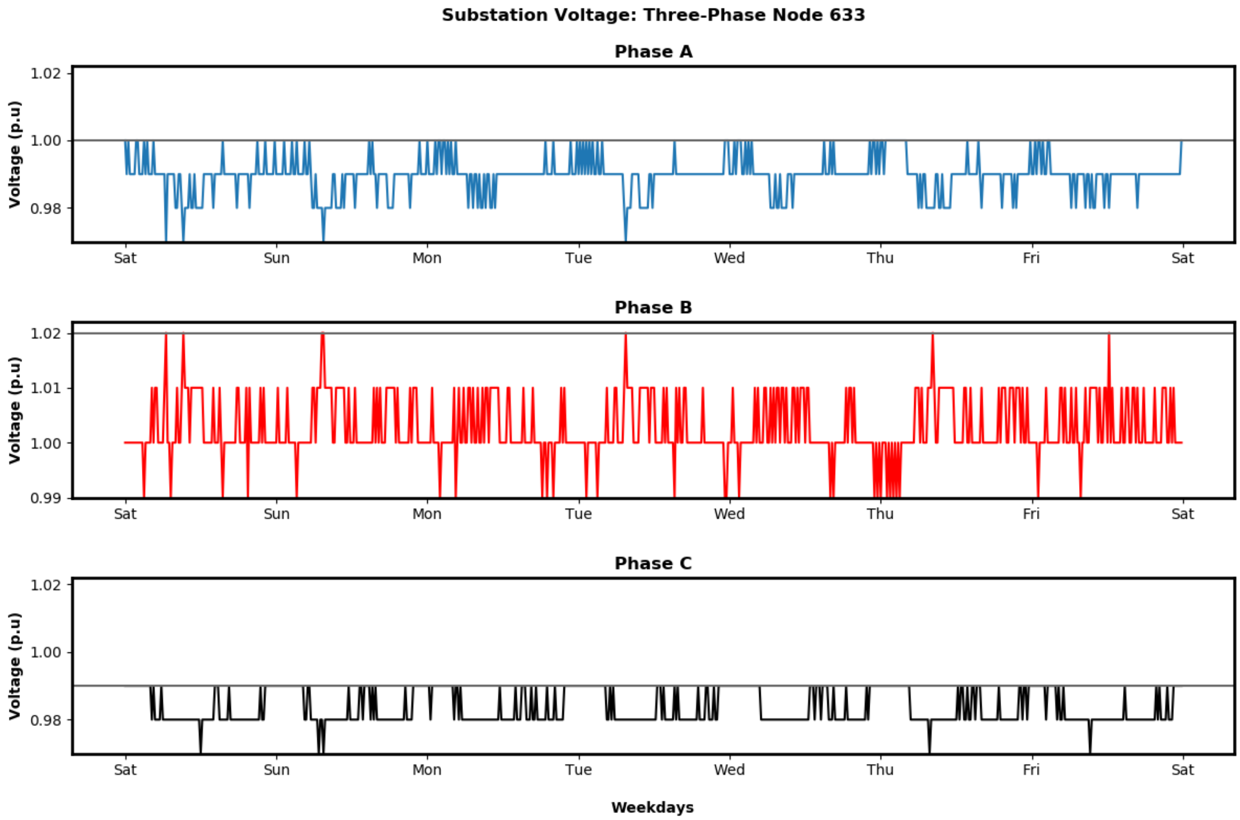

Substation Voltage: Three-Phase Node 633
Phase A
Phase B
Phase C
Voltage (p.u)
1.02
1.01
1.00
0.99
0.98
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 45

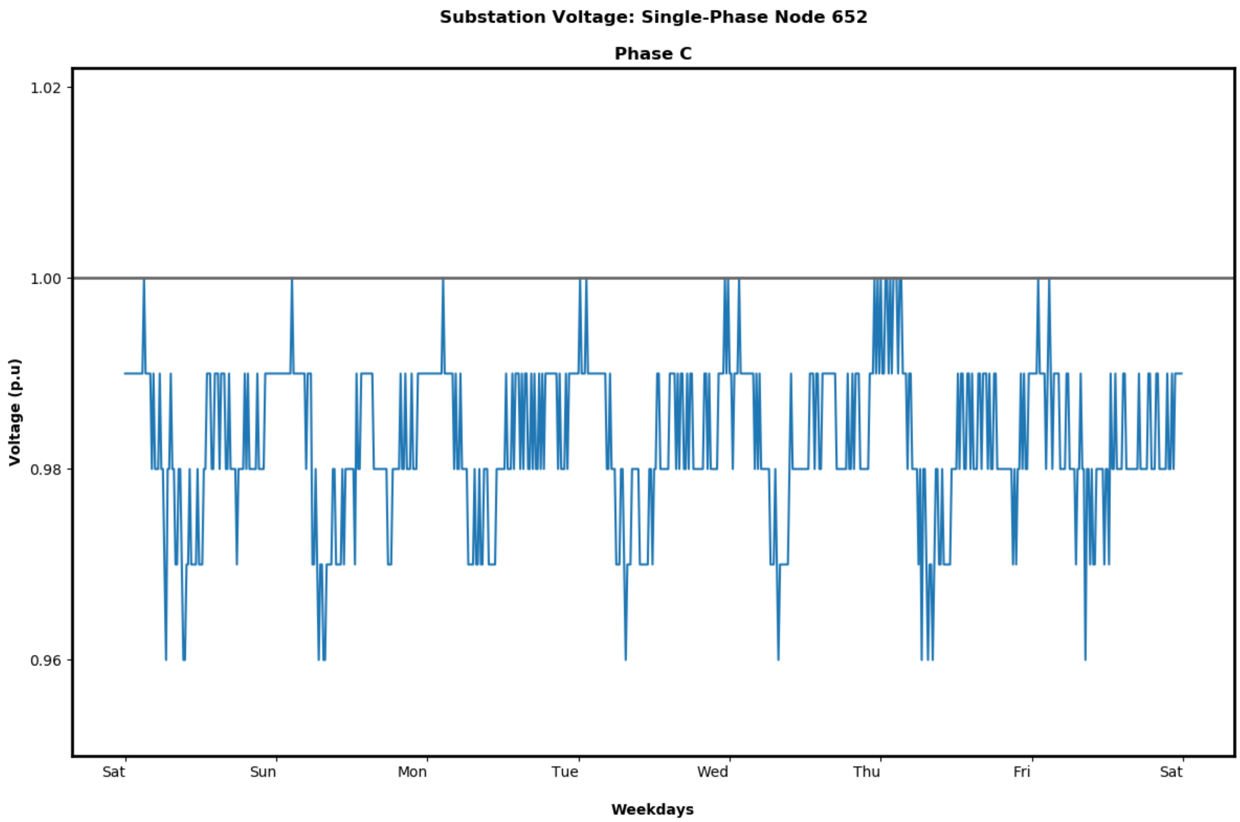
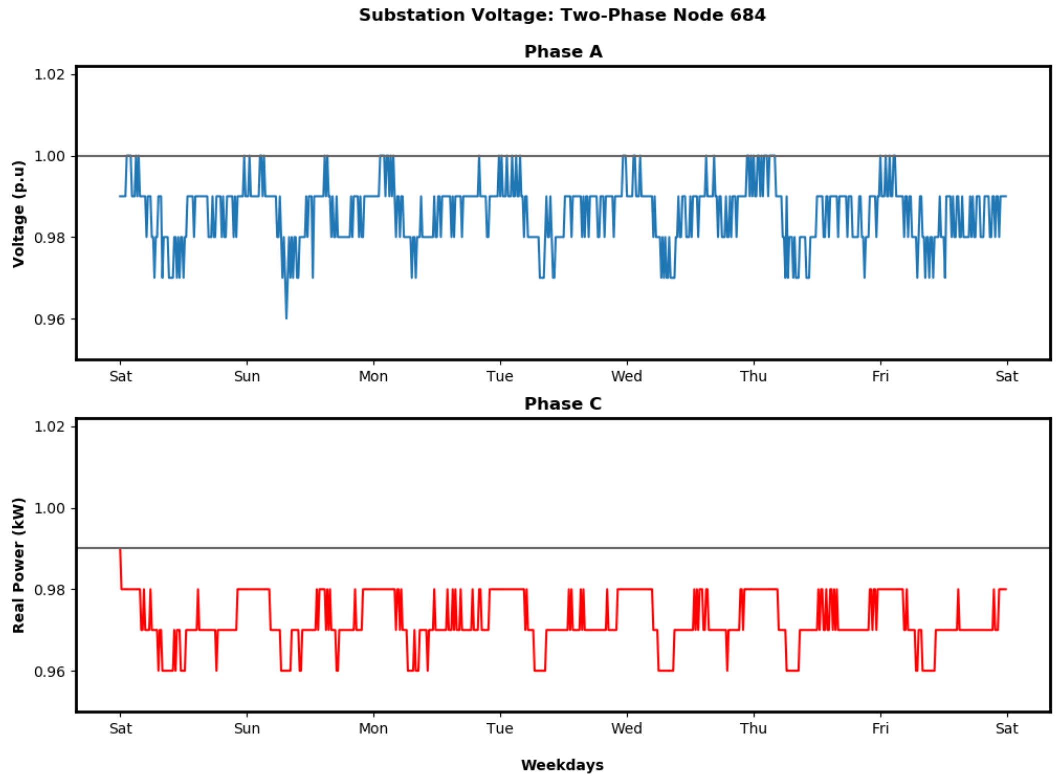

Substation Voltage: Single-Phase Node 652
Phase C
1.02
1.00
0.98
0.96
Voltage (p.u)
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 46



Substation Voltage: Two-Phase Node 684
Phase A
Voltage (p.u)
1.02
1.00
0.98
0.96
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Phase C
Real Power (kW)
1.02
1.00
0.98
0.96
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 47

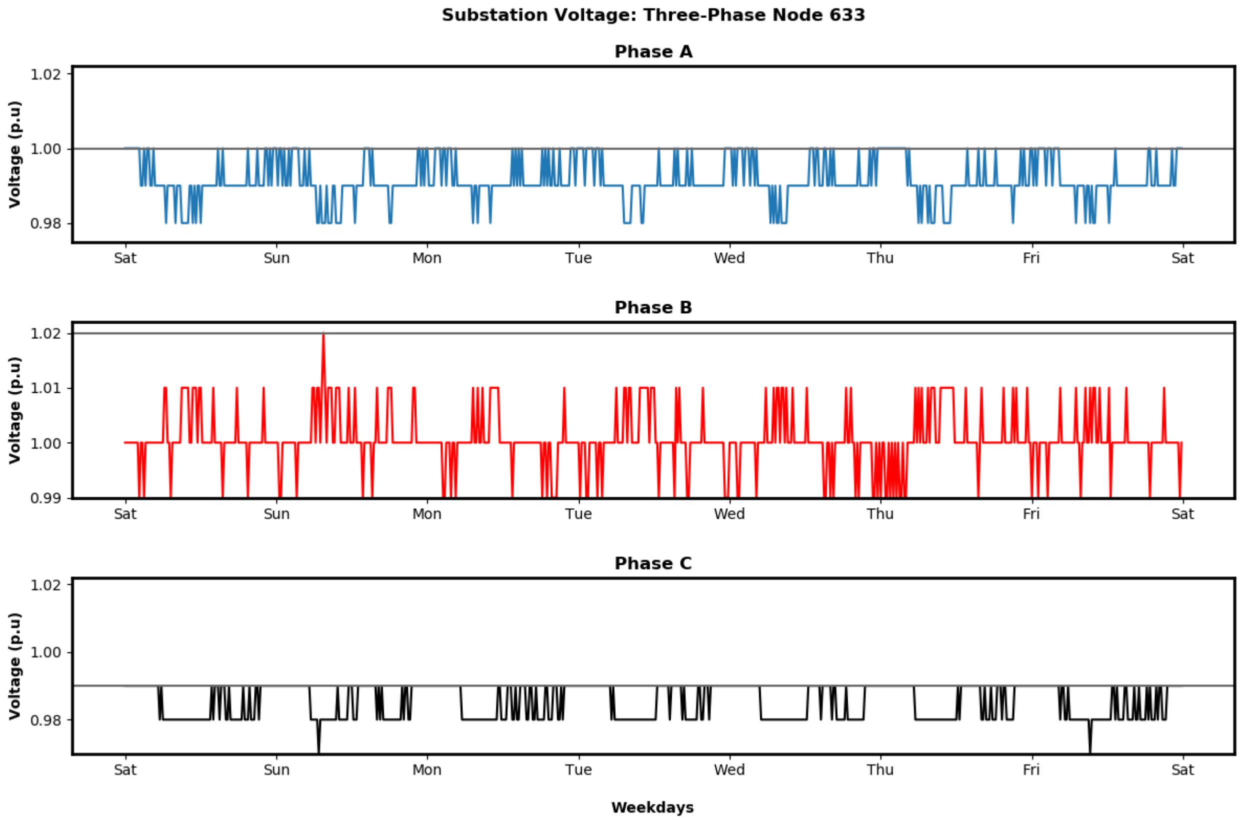

Substation Voltage: Three-Phase Node 633
Phase A
Phase B
Phase C
Voltage (p.u)
1.02
1.00
0.98
1.01
0.99
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Weekdays

Figure 48

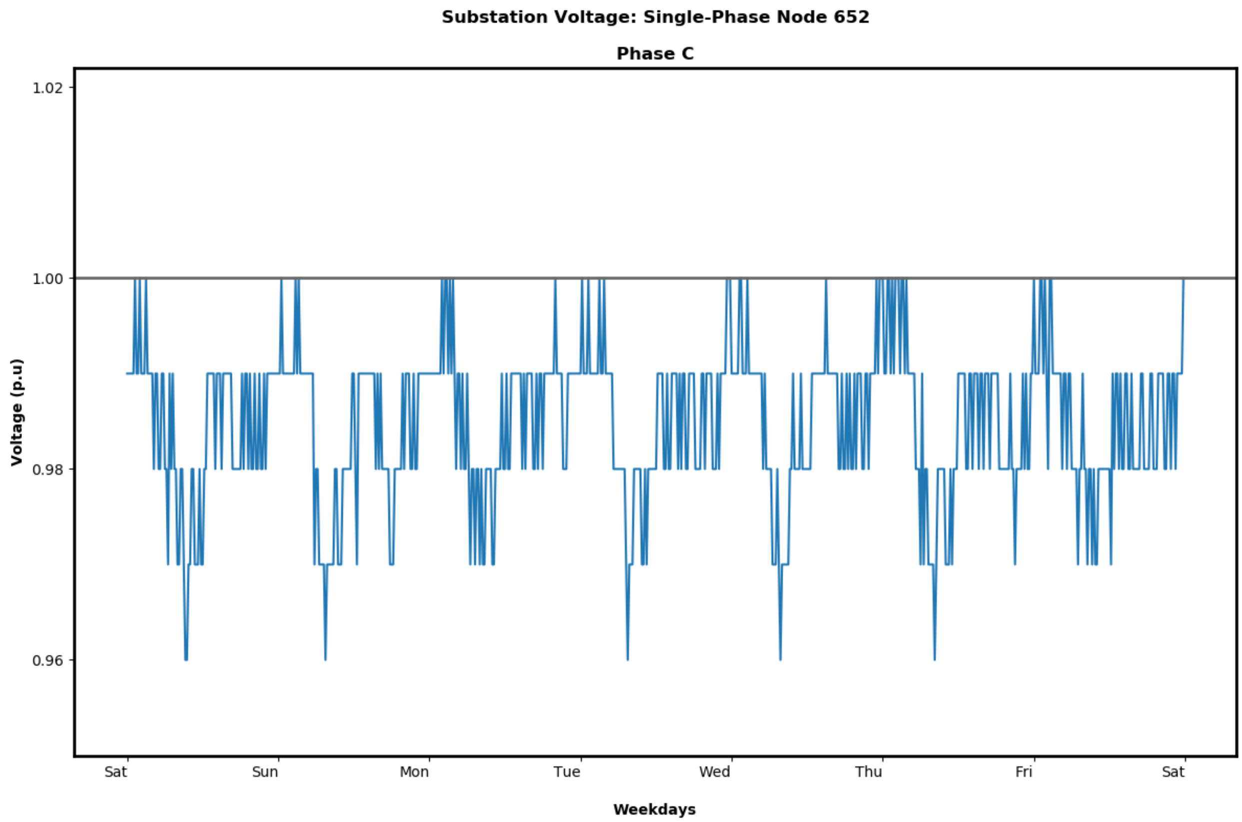

Substation Voltage: Single-Phase Node 652
Phase C
1.02
1.00
0.98
0.96
Voltage (p.u)
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 49

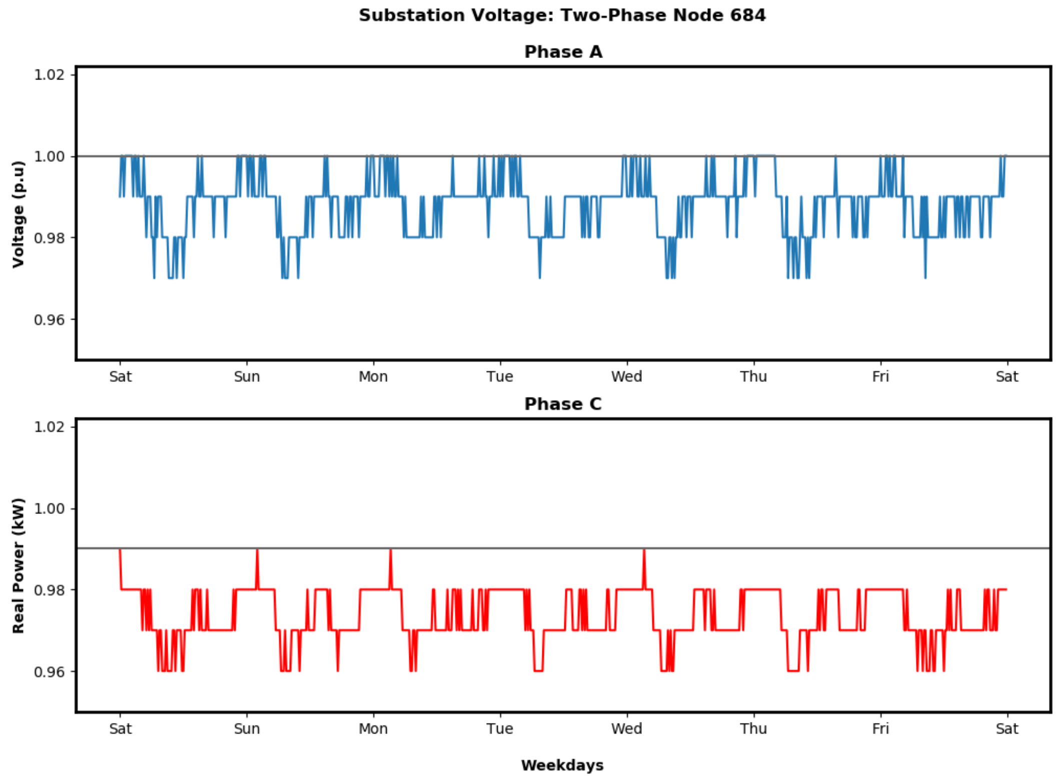

Substation Voltage: Two-Phase Node 684
Phase A
1.02
1.00
0.98
0.96
Voltage (p.u)
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Phase C
1.02
1.00
0.98
0.96
Real Power (kW)
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 50

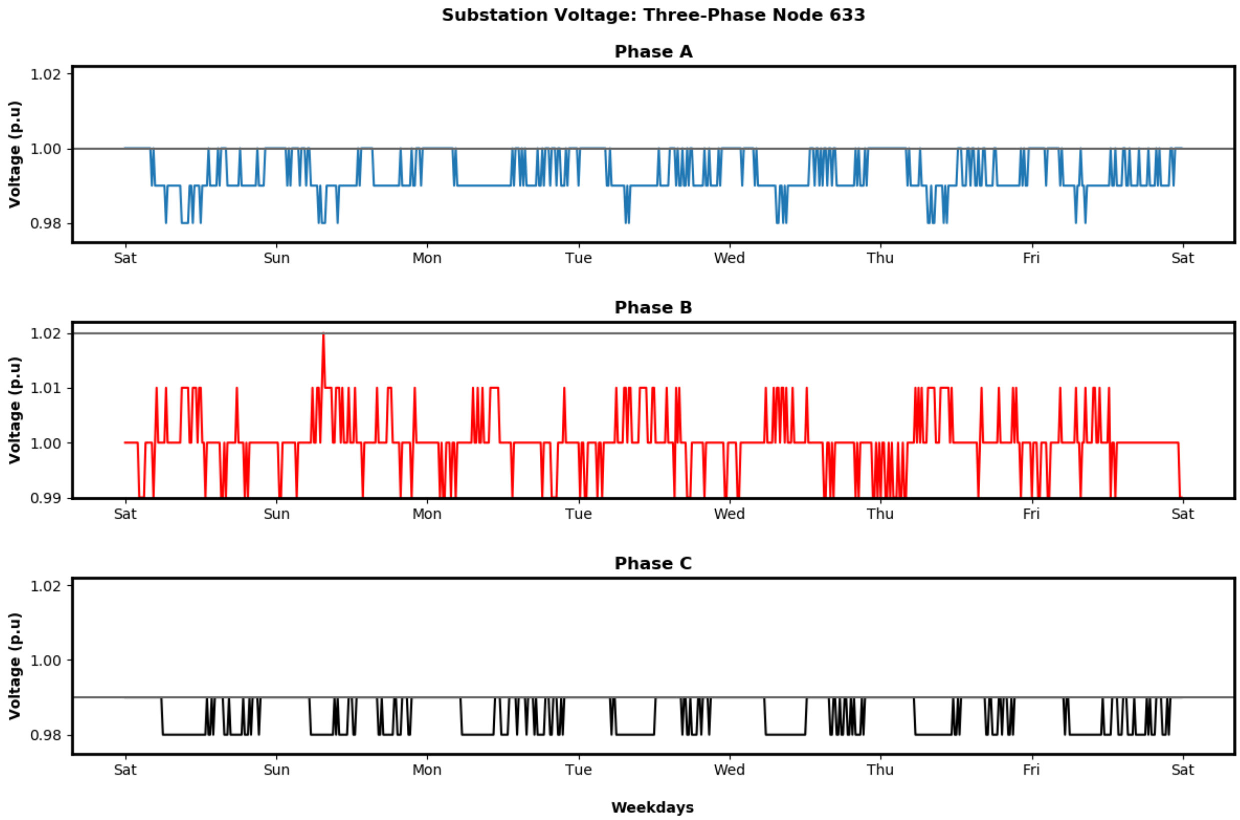

Substation Voltage: Three-Phase Node 633
Phase A
Phase B
Phase C
Voltage (p.u)
1.02
1.01
1.00
0.99
0.98
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 51

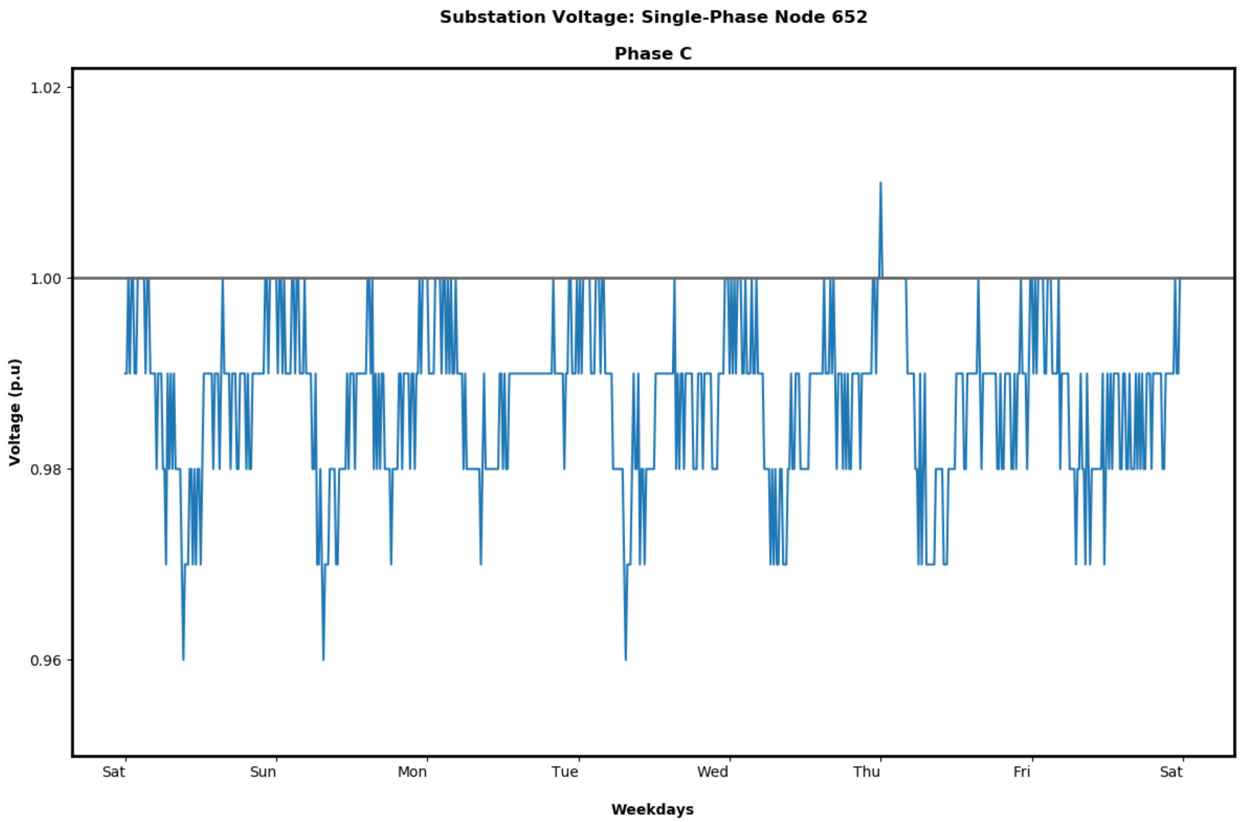

Substation Voltage: Single-Phase Node 652
Phase C
1.02
1.00
0.98
0.96
Voltage (p.u)
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 52

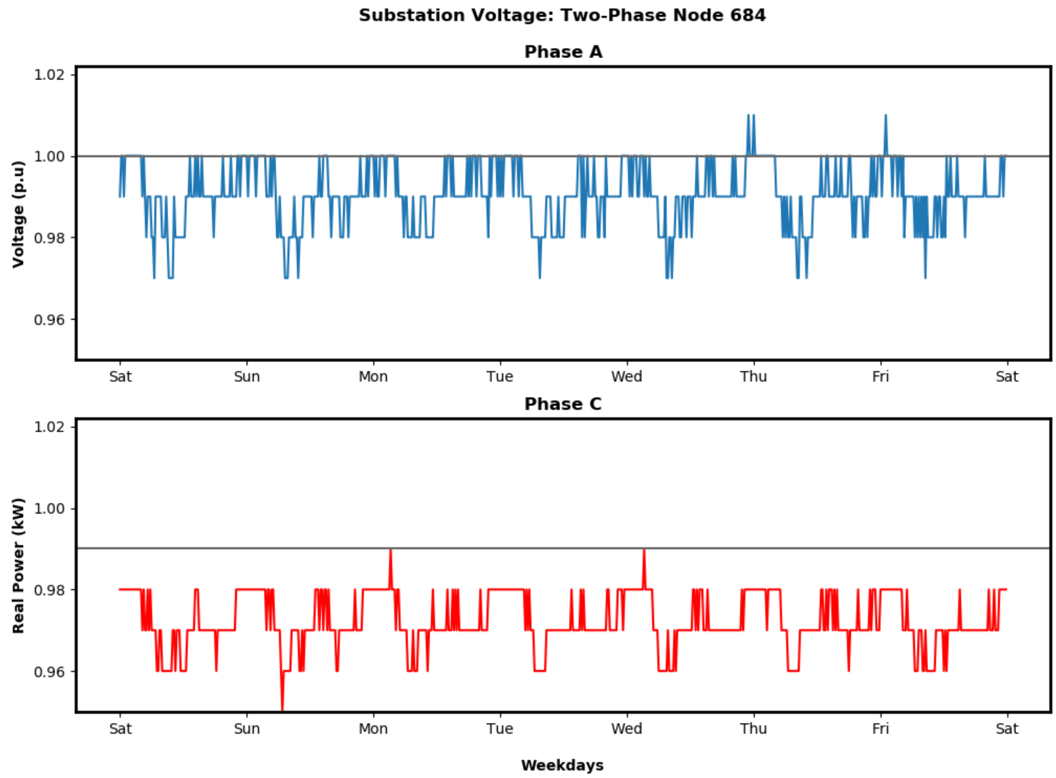

Substation Voltage: Two-Phase Node 684
Phase A
Voltage (p.u)
1.02
1.00
0.98
0.96
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Phase C
Real Power (kW)
1.02
1.00
0.98
0.96
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 53

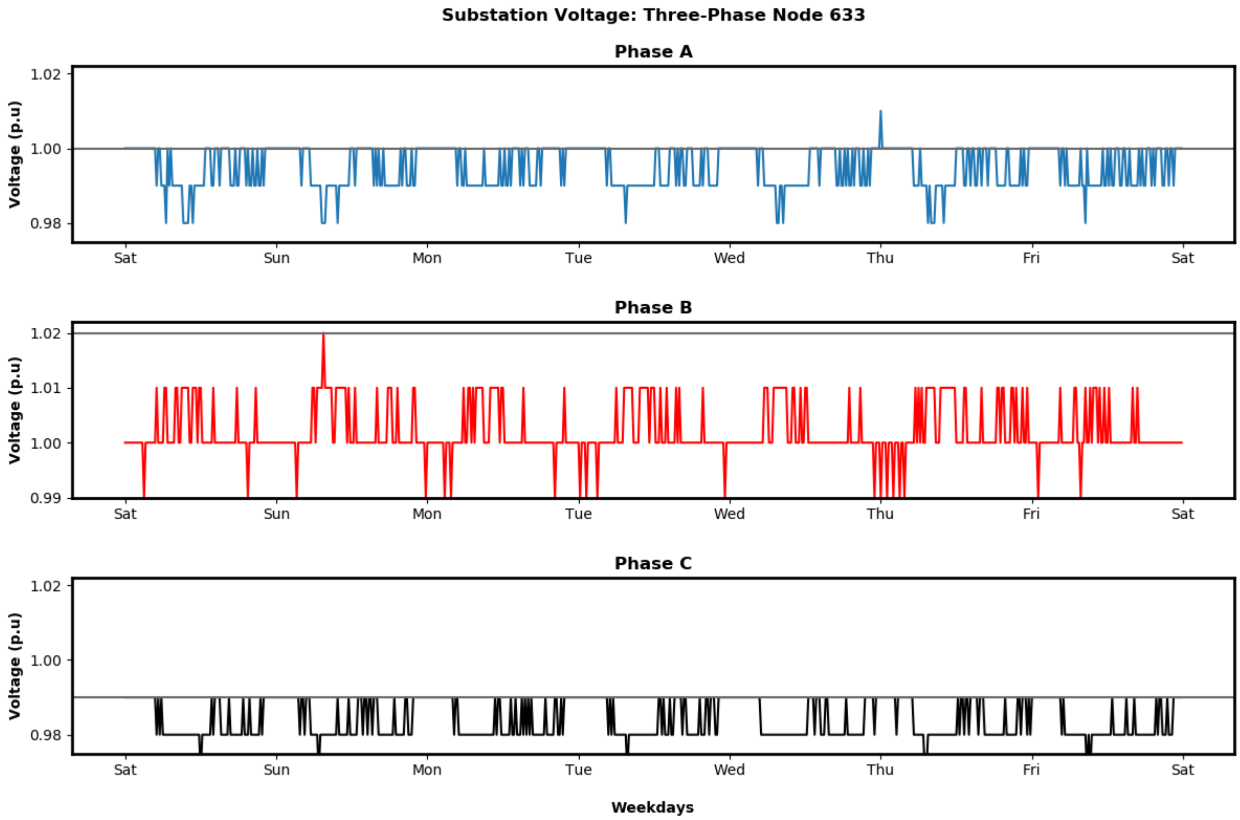

Substation Voltage: Three-Phase Node 633
Phase A
Phase B
Phase C
Voltage (p.u)
1.02
1.01
1.00
0.99
0.98
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 54

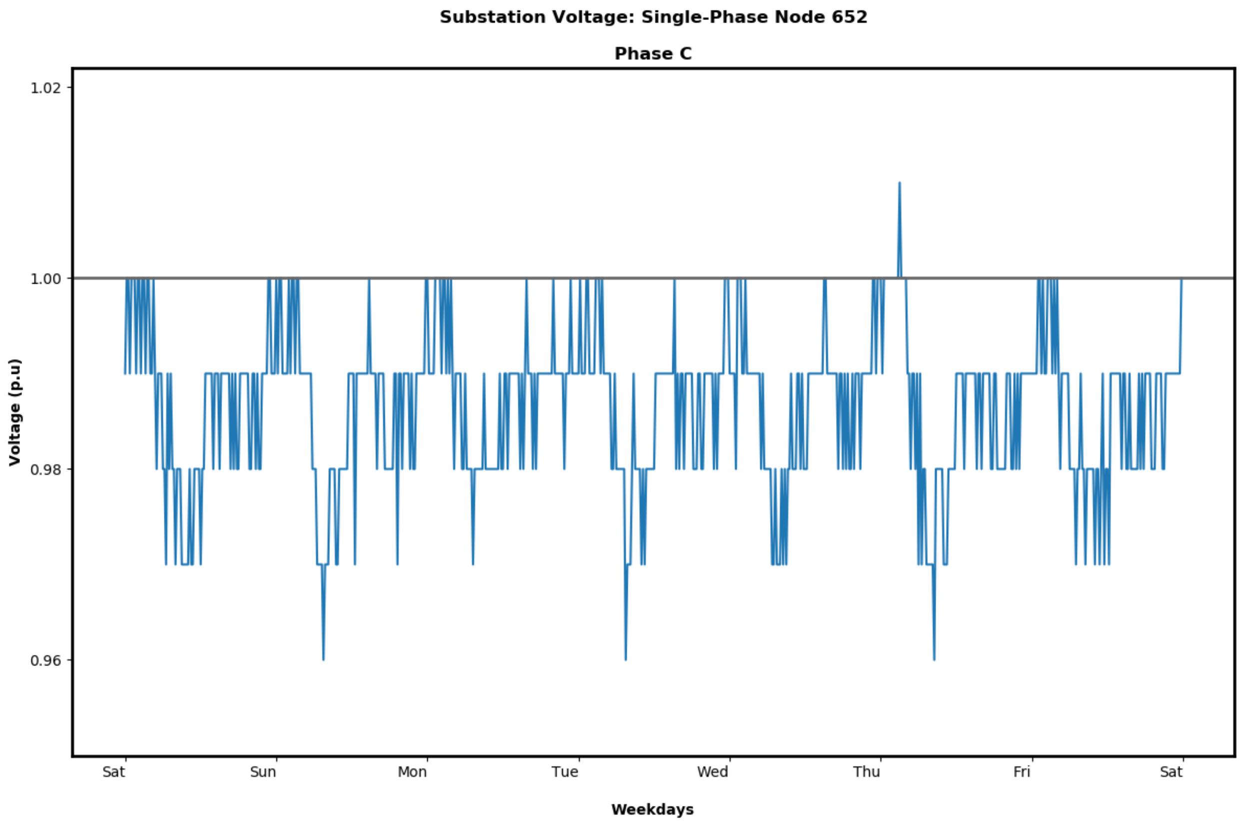

Substation Voltage: Single-Phase Node 652
Phase C
1.02
1.00
0.98
0.96
Voltage (p.u)
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 55

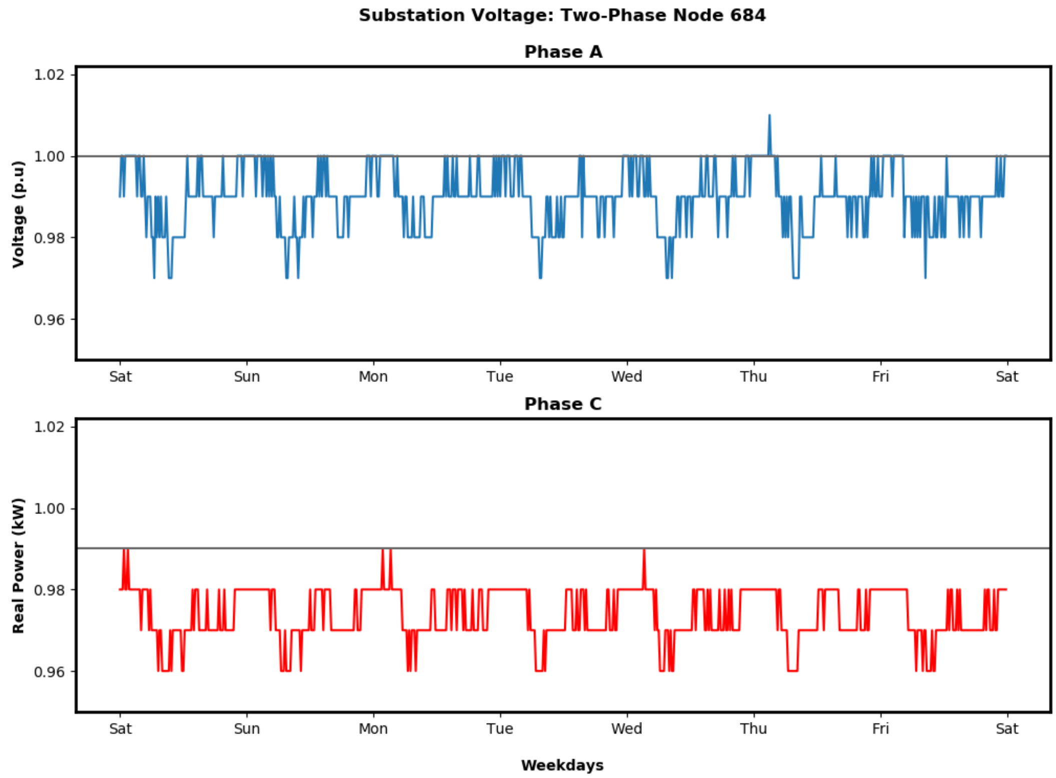

Substation Voltage: Two-Phase Node 684
Phase A
Voltage (p.u)
1.02
1.00
0.98
0.96
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Phase C
Real Power (kW)
1.02
1.00
0.98
0.96
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 56

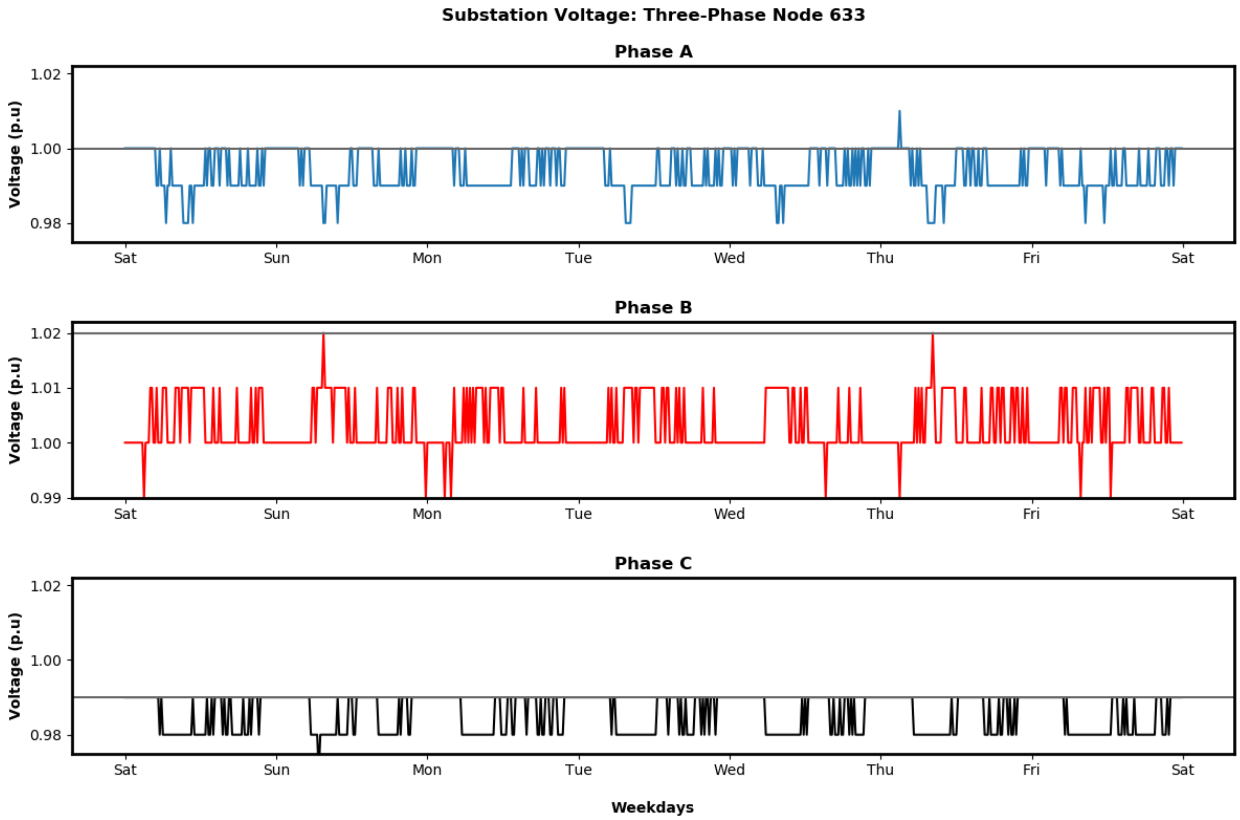

Substation Voltage: Three-Phase Node 633
Phase A
Phase B
Phase C
Voltage (p.u)
1.02
1.01
1.00
0.99
0.98
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 57

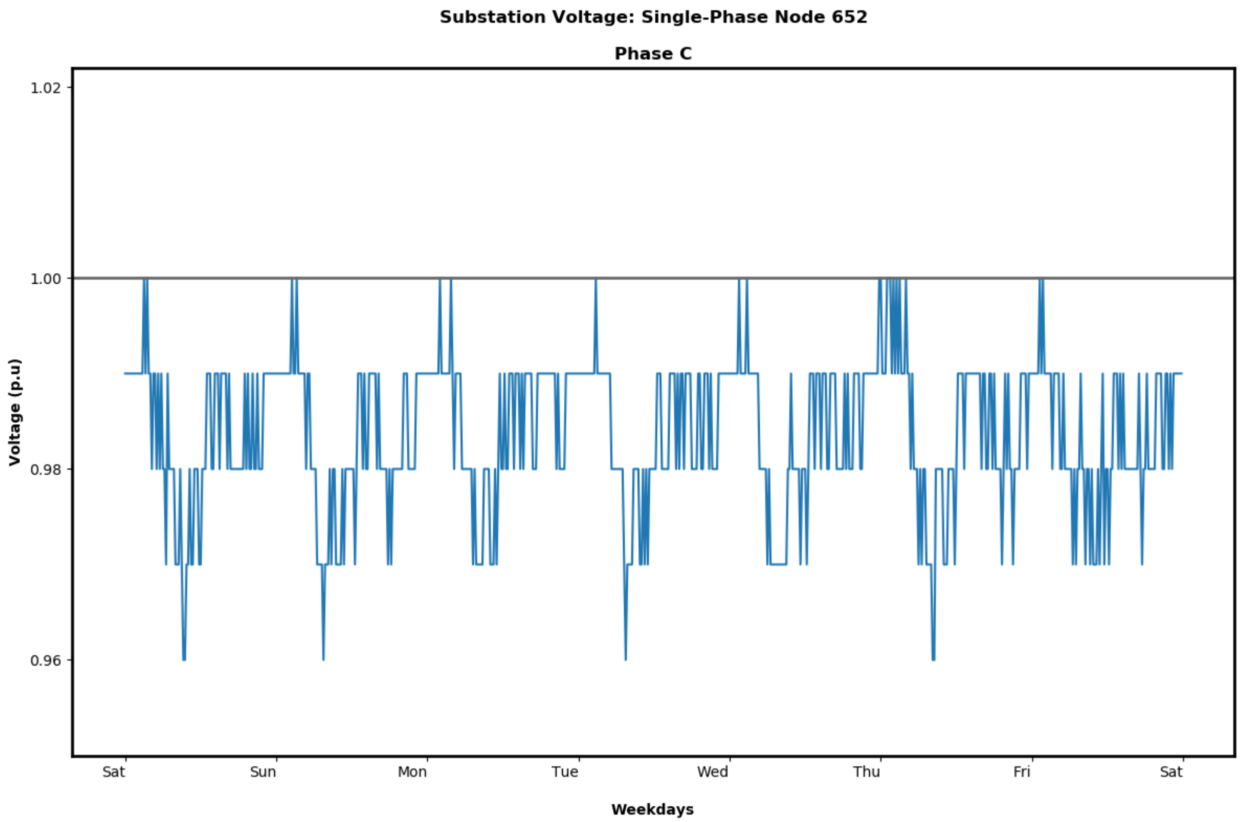

Substation Voltage: Single-Phase Node 652
Phase C
Voltage (p.u)
1.02
1.00
0.98
0.96
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 58

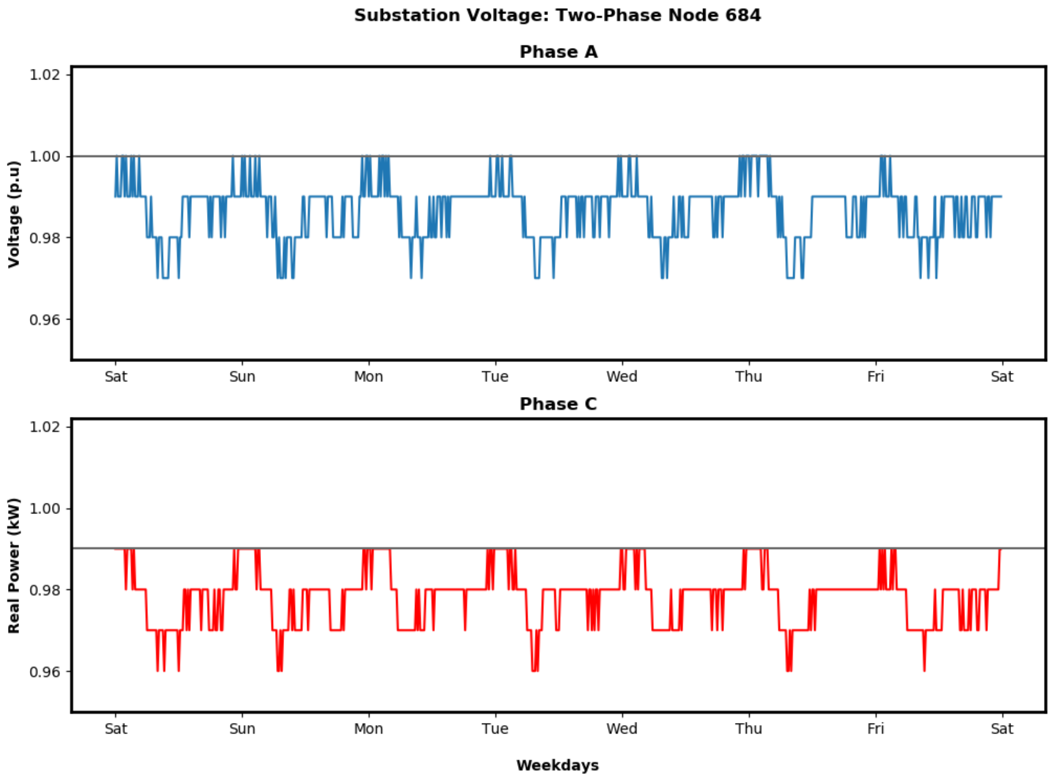

Substation Voltage: Two-Phase Node 684
Phase A
Voltage (p.u)
1.02
1.00
0.98
0.96
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Phase C
Real Power (kW)
1.02
1.00
0.98
0.96
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Weekdays

Figure 59

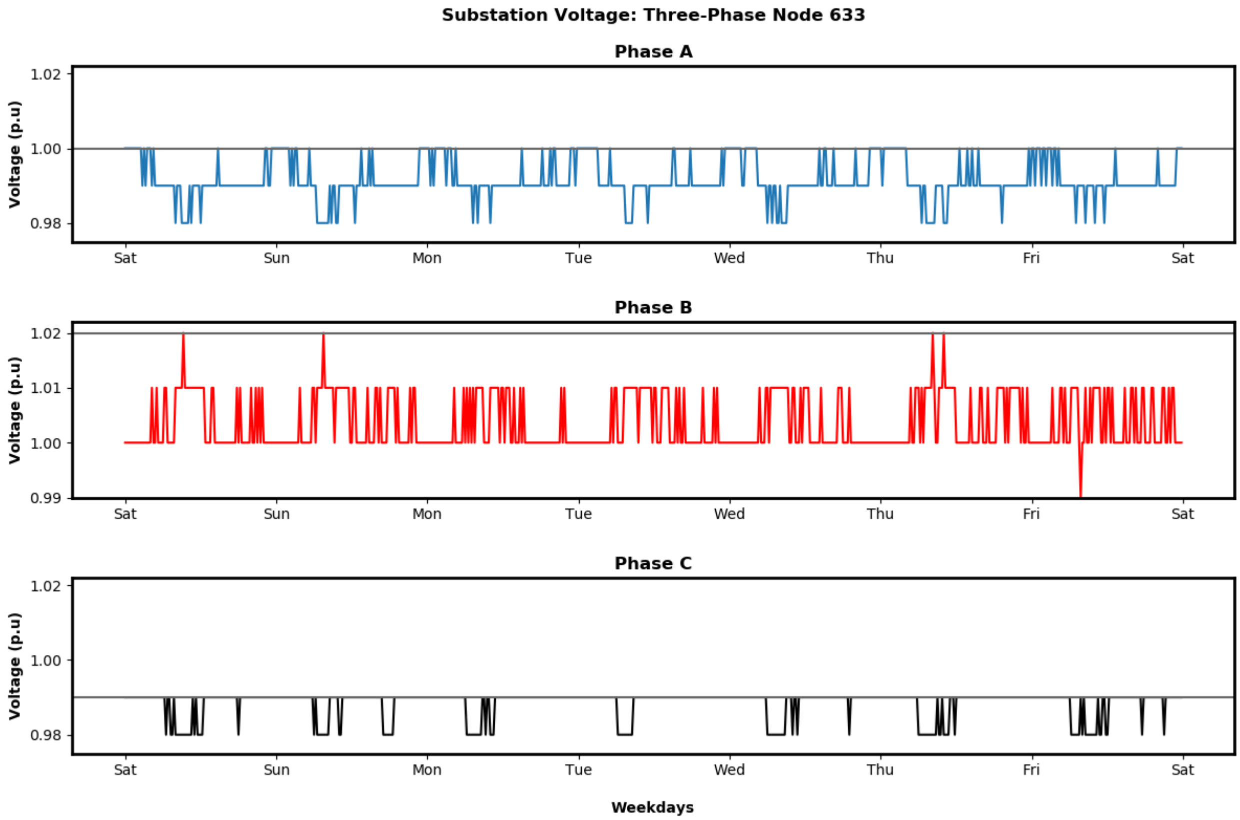

Substation Voltage: Three-Phase Node 633
Phase A
Voltage (p.u)
1.02
1.00
0.98
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Phase B
Voltage (p.u)
1.02
1.01
1.00
0.99
Phase C
Voltage (p.u)
1.02
1.00
0.98
Weekdays

Figure 60

7 Summer 2022

This section addresses the work progress within the ME. Refer to Sean Keene's thesis to get how the system is set up. This document, however, addresses an issue that is related to GridAPPS-D flexibility.

7.1 Problem Statement

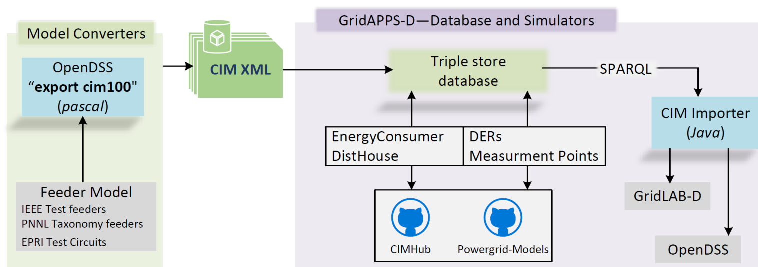


Figure 61: GridAPPS-D Model Conversion

As shown in Figure 61, GridAPPS-D takes an input file (IEEE Test Feeders, PNNL Taxonomy Feeders, or EPRI Test Circuits) as a **dss** format. Within each **dss** file, "export cim100" command is used to export an XML version of the **dss** file. The XML file is then stored in the Triple Store database so it can be adjusted as needed.

Over the years, the PowerLab team has been extensively working with GridLAB-D. The PowerLab team has developed several **glm** scripts to implement studies and cases. Many of these studies are slightly or non-GridAPPS-D-related. The time has come to merge many of these studies within the Modelling Environment (ME). This Section is meant to walk through the progress of implementing scripts that convert **glm** files to **dss**.

7.2 Objectives

- Convert **glm** files to **Common Information Model (CIM)**.

- Insert the appropriate loads in the specified **glm** file using GridAPPS-D.
- Develop communication between entities and demand response capabilities.

7.3 Tools to be Used

- To convert from GridLAB-D to OpenDSS, I used Ditto-CLI
- To edit the **glm** files, I used glm module in Python.
- To convert from **dss** to **CIM**, I used CIMhub

7.4 Processing Steps

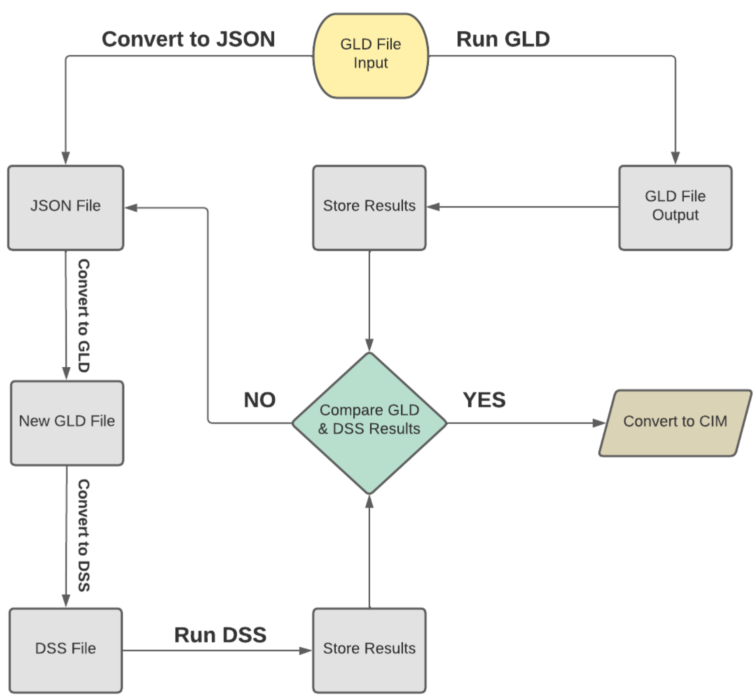


Figure 62: File Conversion Processing Chart

7.4.1 Step 1: Edit glm Files

GridLAB-D and OpenDSS are different software packages and, therefore, they are not expected to have the same models. To find out which GridLAB-D objects are compatible with OpenDSS models, I went through test files within Ditto-CLI. The Node.glm is a **glm** file that can be easily converted to **dss** without errors. The content of this file is compared with the **glm** that I developed to ensure a smooth and accurate transition between GridLAB-D and OpenDSS. These models are as follows:

- Clock
- module powerflow
- overhead line conductor
- line spacing
- line configuration
- transformer configuration
- node object
- overhead line object
- transformer objects
- load object

Note that objects like triplex load and water heaters are not included. Whether these objects can be implemented or not is not the question. The question is, do they align with the final objective of this work? Given that **CIM** files are general models, and their behavior mimics the operation of the aforementioned loads, then the answer is no, they are not needed.

To delete such objects from the GridLAB-D file (or (glm) file), we need a tool to access the (glm) file and perform the required edits. This editor as far as the author's knowledge does not exist. Therefore, I used the GLM module in Python. module in python. This module converts (glm) files to JSON and vice-versa. It's easy to deal with JSON files with python. Once the (glm) file is converted to JSON, the following objects were deleted:

- triplex_load objects.
- triplex_line objects.
- house objects.
- waterheater objects.
- multi_recoreder objects.
- player objects.
- capacitor objects.

Again, as far as the author's knowledge, these objects do not have similar models in OpenDSS. The next step is to convert the JSON file back to a (glm) format. There are several crucial factors to note here:

- During the conversion process, the *(*clock) object has double quotes instead of single quotes. GridLAB-D compiler does not interpret the double quotes correctly, so they need to be changed to single quotes.
- The 13-node feeder contains 13 nodes. However, only 10 nodes are used due to reasons mentioned in Midrar's Thesis. Some branch nodes are not used within the 13-node feeder and these are as follows:
  - Node 6711
  - Node 6321
  - Node 634
  - Node 650
  - Node 630

To ensure an accurate and smooth conversion from GridLAB-D to OpenDSS, run the (glm) file and the (dss) file and make sure they run correctly without errors and have the same results.

7.4.2 Ditto-CLI

To convert a **glm** file to **dss**, the following command may be used:

```
$ ditto-cli convert --input "glm file name" --from gridlabd --to
    opendss --output "output file directory"
```

The Ditto-CLI tool does most of the work for those who want to convert GridLAB-D files to OpenDSS and vice-versa. However, it is not perfect and it has its shortfalls. several errors in particular that have been showing most of the time are the following:

```
$ Matrix Inversion Error for Line " line name".
$ TLineObj.CalcYPrim.
$ Access violation.
```

Such errors are caused by the length and impedance conversions of the overhead and underground lines. In my case, since I'm working on the 13-node feeder, I need to make sure the original length parameters and impedance matrices are the same as the one I have. After the conversion, run the **dss** file and if it outputs similar errors as above, do the following:

- edit "LineCodes.dss" file.
- change the "R" and "X" matrices and ensure they correspond to the original IEEE-13 Node Feeder.
- All original IEEE feeders in **dss** format can be found in DSS Github Repository

7.5 Important opendsscmd Commands

- To export an XML version of the current **dss** feeder, insert the following command within the "Master.dss" file:

```
$ export cim100
```

- To export a CIM version of the current **dss** file, create a script containing the following and name it “export_cim.dss”:

```
redirect Master.dss
solve
export y triplet base_ysparse.csv
export ynodelist base_nodelist.csv
export summary base_summary.csv
```

- Using opendsscmd command in the terminal, run the above script.
- Source of the above info.