Power Lab Notebook

Authors:
Midrar Adham

September 11, 2021

Maseeh College of Engineering and Computer Science
PORTLAND STATE UNIVERSITY

Contents

List of Figures

List of Tables

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 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`

1.6 results

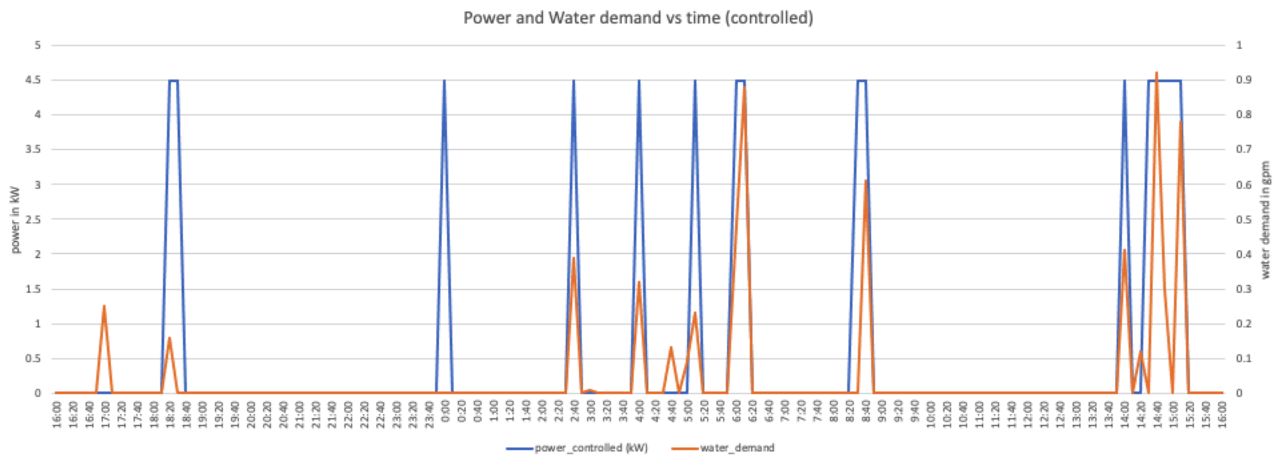

Figure 1: Power and Water Demand vs Time (un-controlled)

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.

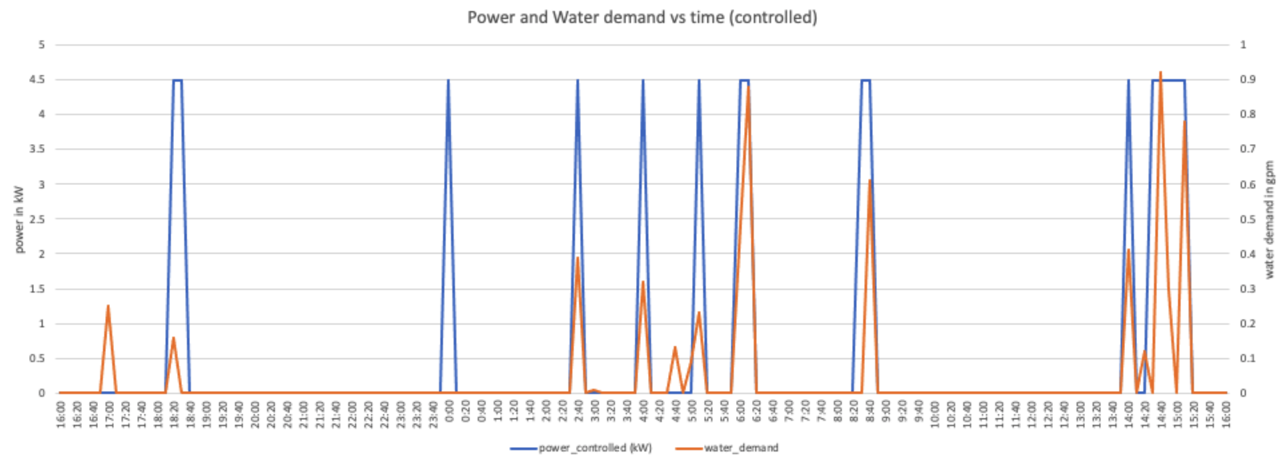

Figure 2: Power and Water Demand vs Time (controlled)

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 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.

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 5 shows the EWH behavior under the same parameters.

# 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	+0+0j	+0+0j	+0	+119.977	+0
2019-12-31 16:10:00 PST	+3.782	+119	ONEZNODE	+0	+0+0j	+0+0j	+0	+119.911	+0
2019-12-31 16:20:00 PST	+3.782	+119	ONEZNODE	+0	+0+0j	+0+0j	+0	+119.844	+0
2019-12-31 16:30:00 PST	+3.782	+119	ONEZNODE	+0	+0+0j	+0+0j	+0	+119.778	+0
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 5 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?

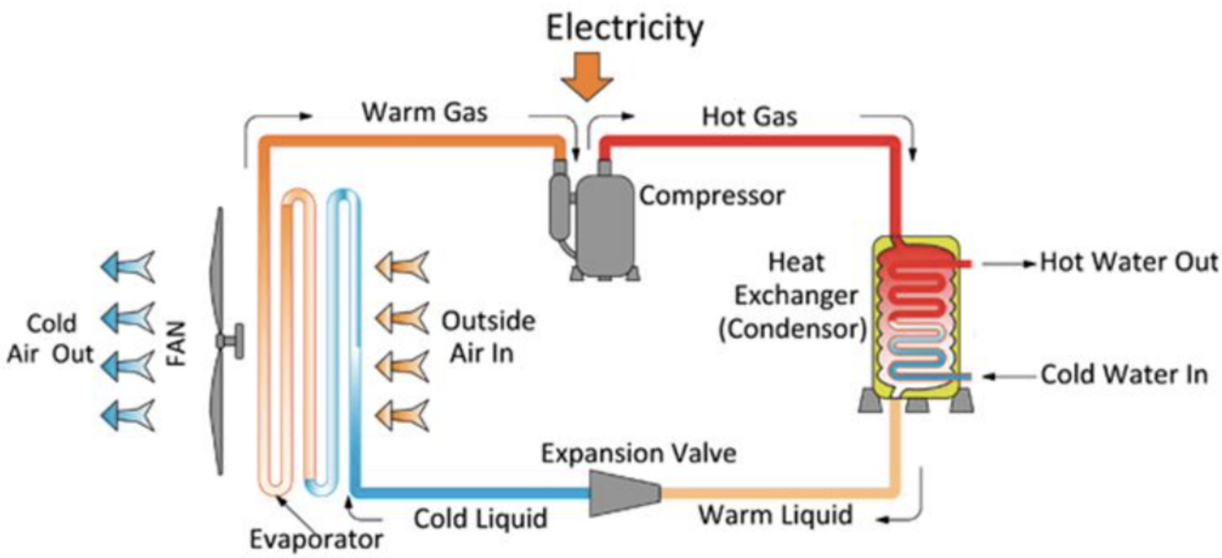

Figure 5: source:nukiengineering.com

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 [?], [?], and [?]. 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 × 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:

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

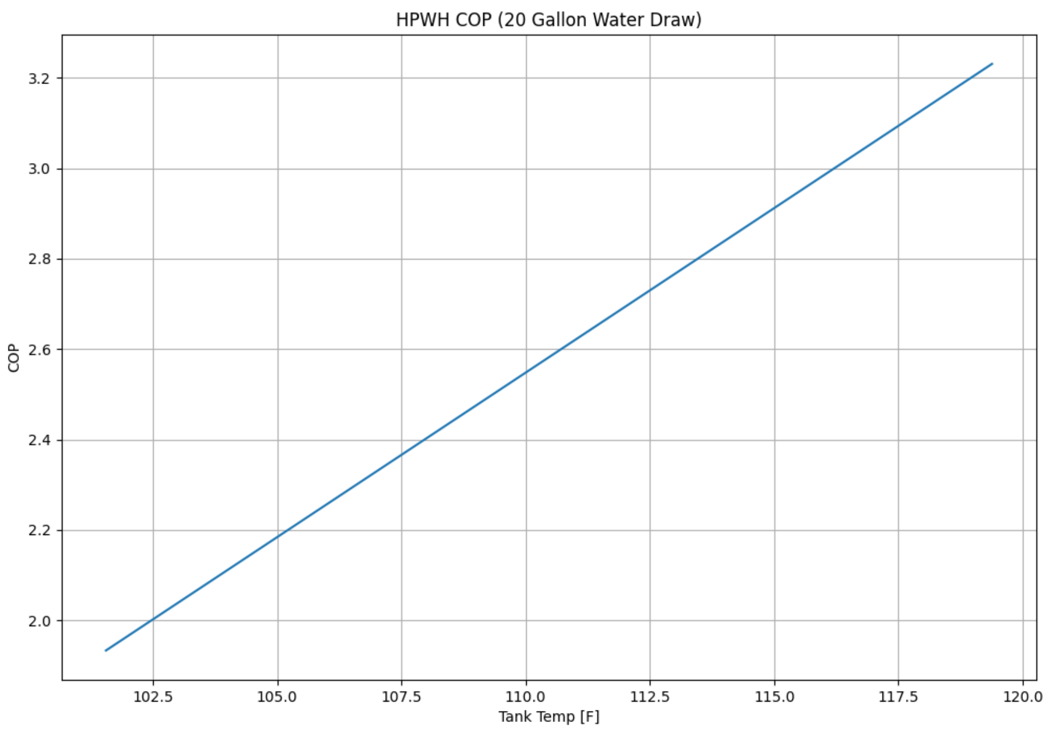

Figure 6: HPWH COP: 20 Gallon Water Draw

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

time	EnergyTake	Watts
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

time	EnergyTake	Watts
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

time	EnergyTake	Watts
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

time	EnergyTake	Watts
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 $^\circ F$. 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

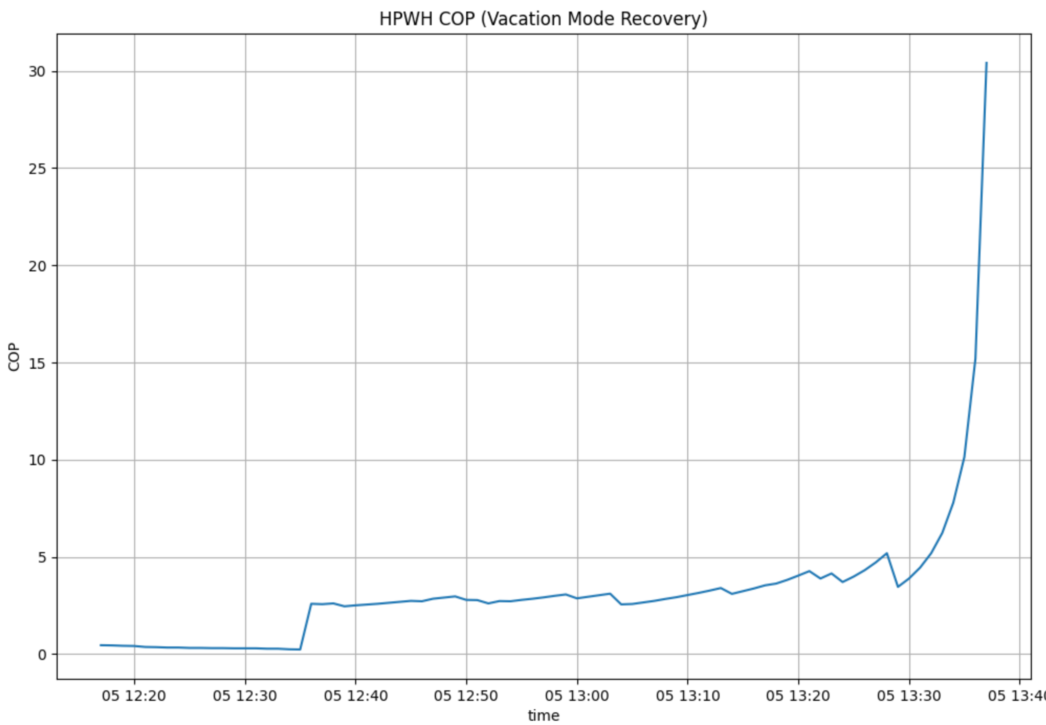

Figure 7: HPWH COP vs Time: Vacation Mode Recovery

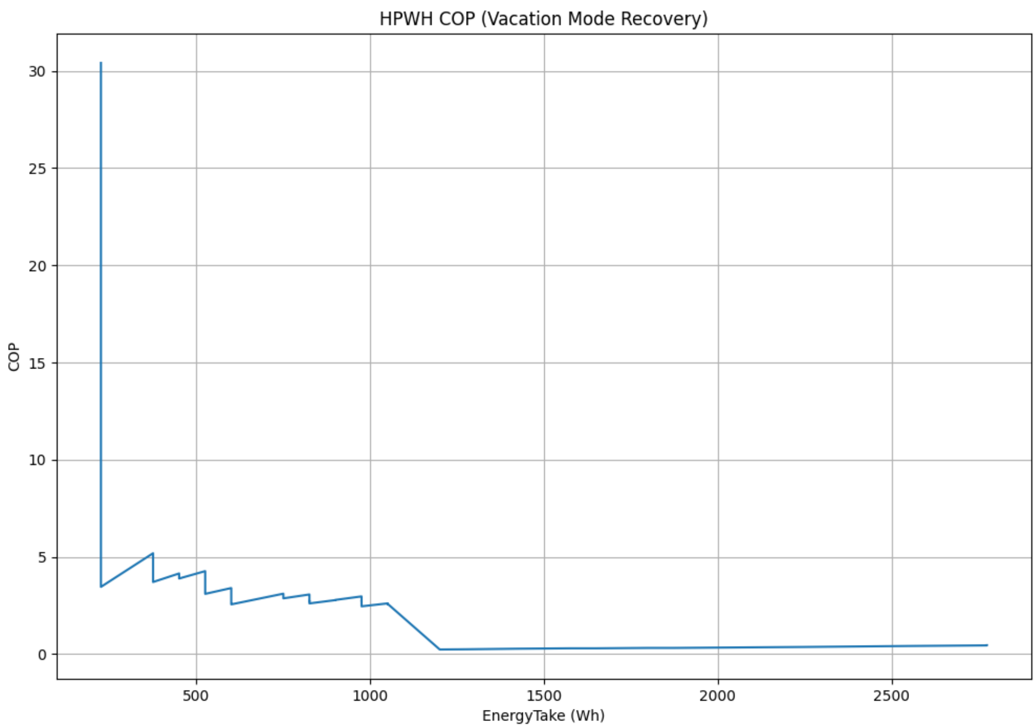

Figure 8: HPWH COP vs EnergyTake: Vacation Mode Recovery

5.3 this should work in a good position