Customer Benefit Analysis and Experimental Study of Residential Rooftop PV and Energy Storage Systems

By

Pavan Etha

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2017 by the
Graduate Supervisory Committee:

George Karady, Chair
Gerald Heydt
Raja Ayyanar
Grant Smedley

ARIZONA STATE UNIVERSITY

December 2017

ABSTRACT

The government support towards green energy sources for the better future of the planet has changed the perspective of the people towards the usage of green energy. Among renewables, solar is one of the important and easily accessible resource to convert energy from sun directly into electricity and this system has gained fame since the past three decades.

SRP has set up a 6.36 kW PV and 19.4 kWh battery system on the rooftop of Engineering Research Center (ERC). The system is grid-connected and ASU (Arizona State University) has developed two load banks with a minimum step of 72 watts to simulate different residential load profiles and perform other research objectives.

A customer benefit analysis is performed for residential customers with photovoltaic (PV) systems and energy storage particularly in the state of Arizona. By optimizing the use of energy storage device, the algorithm aims at maximizing the profit and minimizing utility bills in accordance with the demand charge algorithm of the local utility. This part of the research has been published as a conference paper in IEEE PES General Meeting 2017.

A transient test is performed on the PV-battery during the on-grid mode and the off-grid mode to study the system behavior during the transients. An algorithm is developed by the ASU research team to minimize the demand charge tariff for the residential customers. A statistical analysis is performed on the data collected from the system using a MATLAB algorithm.

ACKNOWLEDGMENTS

I would like to express my gratitude towards Dr. George Karady for offering me the opportunity to work on this project and giving me valuable guidance through the past one and half year. I would like to thank my committee members, Dr. Gerald Heydt, Dr. Raja Ayyanar, and Grant Smedley for their valuable time and the instructions.

Also, I want to thank Mr. Travis Stowers from SRP for providing me the opportunity to work with them and giving me their continued support. The thesis cannot be completed without their technical support and some important data inputs.

In the end, I want to thank all my friends for their help and encouragement during the time I finished my masters. Last but not the least I want to thank my parents Etha Satyanarayana and Etha Vijaya Lakshmi for their unconditional love and enormous support throughout my educational path.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure		Page

Figure Page

Figure Page

Figure Page

Figure		Page

NOMENCLATURE

SRP	Salt River Project
ASU	Arizona State University
SN	Serial Number
ERC	Engineering Research Center
TOU	Time of Use
TOUD	Time of Use Demand
MLP	Mixed Integer Programming
SOC	State of Charge
P_{loadi}	Load demand
P_{pvi}	Historical PV generated
$P_{pv-loadi}$	PV fed to load
$P_{pv-bati}$	PV fed to battery
$P_{pv-gridi}$	PV exported to grid
$P_{bat-loadi}$	Power transferred from battery to grid
$P_{bat-gridi}$	Power exported to grid from battery
$P_{grid-loadi}$	Power imported from grid to load
P_{surpi}	Surplus power

CHAPTER 1- INTRODUCTION

1.1 Battery Assisted Residential PV System

The government support towards the green energy sources for better future of the planet has changed the perspective of the people towards the implementation of renewable energy. Different states have set goals of converting their conventional sources to different renewable sources for the supply of energy. Arizona has a target of reaching 15 % total electricity generation of renewables by 2025 [20]. Solar, wind, and hydro are some of the popular types of renewable sources.

Solar is one of the important and easily accessible resource considering PV to be the most convenient technology to convert energy from sun directly into electricity and the system has gained fame since the past three decades. The energy from the solar is emission less and does not consume any form of natural resources. PV systems range from small, rooftop-mounted or building-integrated systems with capacity varying from a few kilowatts to megawatts. There has been a huge increase in the number of residential PV-battery systems due to the incentives provided by the government.

In Arizona, each year, a total of 30% of the required renewable energy target must come from non-utility distributed generation [20]. Half of the non-utility distributed generation requirement must come from residential sites and the other half from non-residential installations. Arizona allows net metering for a customer-sited renewable generation [20]. In addition, Arizona's energy efficiency standards require investor-owned electric utilities and natural gas utilities to increase energy efficiency in order to reduce consumption of both electricity and natural gas.

A grid-connected PV-battery system has a varied advantage, as the utility grid offers net metering and the feed-in-tariff. The utilities have come up with different tariff plans for the customers having their own generation. SRP has a demand charge component rate plan (E-27) in Arizona for their utility customers [7]. The storage device, batteries play an important role in providing the long term and short term economic benefits for such a utility rate plan. The PV system vendors such as Sunverge have come up with an intelligent algorithm that would try to predict the future load and dispatch the battery accordingly and minimize the demand during the on-peak hours.

1.2 Motivation for Research

The benefits of a residential PV-battery system are still undergoing a lot of research. In order to study the effects of the residential PV-battery system with regards to the Arizona climate and the local tariffs, SRP has installed a 6.36 kW PV and 19.4 kWh battery system. The system is installed on the rooftop of ERC. The system is connected to the grid. ASU will report the experimental results from the system on a monthly basis to SRP.

Economic benefits of the system are constantly updated due to the variation in the cost of solar panels and battery system. Developing an algorithm that will provide an optimal size of PV-battery system with regards to the load characteristics of the customer will help SRP advise their customers on the long-term economic benefits from the PV-battery system.

There has been a debate if the PV-battery system of an average load consuming customer can work in a standalone mode. ASU will try to answer it by conducting the transient test using a 3.5-ton residential air conditioner that can be seen at almost all the individual residential houses in Arizona.

Every utility residential customer has a varying load characteristic. Developing a load bank and simulating different load profiles will help SRP understand the benefits from the system with regards to customer generation plan (E-27 tariff plan).

SRP has a demand charge rate plan for the residential customers. Proper usage of available battery energy is very important in order to reduce the maximum demand during the on-peak hours. Developing an operating strategy will help SRP advise customers in optimal utilization of the available battery capacity during the on-peak hours.

1.3 Literature Review

Rapid reduction of fossil fuel resources, increase in the electricity demand and the environmental concerns associated with conventional generators has led to a worldwide concern on the development of alternative electric energy generation methods. Renewable energy sources have provided an opportunity to address these concerns. Among renewable energy technologies, the grid-connected photovoltaic application has gained a great attention in research because as it is one of the easily available and effective energy source. Fig. 1.1 shows a general model of a grid-connected PV-battery system.

Issues primarily related to the distribution grid including growing end-use demand, transmission and substation limits, voltage drop have moved utility companies towards peak demand management by assessing time-of-use energy pricing and demand charges for industrial and commercial consumers. Recently with the increased implementation of smart meters and electric vehicles, some utility companies introduced TOU rate structure and peak demand charges for residential consumers. The capability of batteries to supply energy generated at one time to a load at some later time can provide financial benefits to the system's owner through peak shaving, load shifting, and demand response applications.

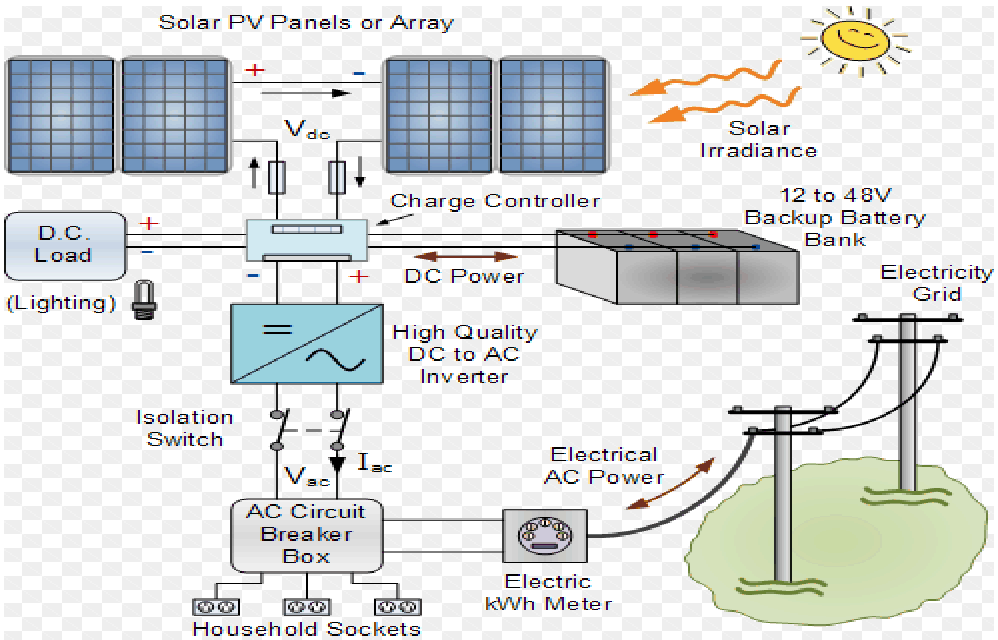

Fig. 1.1. Grid-connected PV-battery System [6]

Customers with PV /storage system that purchase electricity on a TOU basis can use an excess of PV generation early in the day to support a load later in the day. If PV generation is not sufficient to charge the battery, electricity can be purchased from the grid when the time-of-use price is low during the off-peak hours. Electricity purchased from the grid is used to supply all or part of the load when on-peak rates are in effect. For the purpose of peak shaving in a grid-connected PV system, PV provides all required power above a specified threshold during on-peak hours. If the PV output was not sufficient, discharging the battery which energized earlier, reduces the house demand to the desired value. The objective is to investigate effects of the most available peak demand management tariff structures on optimal sizing and dispatching of the battery which minimizes costs of purchasing electricity from the grid. Charging higher prices at times when it is cost-beneficial to reduce customer demand and charging lower prices at the other times (time-varying prices) and charging the customers for the maximum demand (demand charge) are important types of peak demand management tariffs. SRP uses time-of-use with specifying demand charge tariffs for residential customers.

Energy storage system integration in residential buildings equipped with PV generation can help to balance the electrical power grid. The PV-battery system configuration should increase the grid independence of individual households. The research paper [1] provides an optimal dispatch strategy for the PV-battery using linear programming, the objective is to reduce the amount of energy purchased from the grid. Similar studies have been conducted in [2], the author uses MLP to optimize the battery energy by providing a battery dispatch schedule.

The general optimization dispatch schedule for a PV- Battery system as described in [1], [2], [3] is as given below,

$$Objective : \min \sum_{i}^{cycles} C_{buy} \; P_{grid-loadi} - C_{sell} (P_{pv-gridi} + (P_{pvw} \; P_{bat-gridi})).$$

The $\boldsymbol{C_{buy}}$ and $\boldsymbol{C_{sell}}$ are the electricity prices for the feed-in tariff. The objective tries to minimize the amount of power consumed from the grid. The constraints that govern the objective are as follows,

`

$$P_{pvi} = P_{pv-loadi} + P_{pv-bati} + P_{pv-gridi} + P_{surpi} \quad . \tag{1.1}$$

The constraint (1.1) explains PV energy generated is equal to the PV energy supplied to load, battery, grid and surplus energy that is wasted

$$P_{loadi} = P_{pv-loadi} + P_{bat-loadi} + P_{grid-loadi} \, . \tag{1.2}$$

The constraint (1.2) is a power balance equation between load and energy supplied

$$E_{soci} = E_{soci,0} + P_{pv-bati} - P_{bat-loadi} - P_{bat-gridi} - L_{bat} \, . \tag{1.3}$$

The Constraint (1.3) explains the state of charge of battery (SoC) and E_{soci} is the initial state of charge of battery

$$P_{Pv-gridi} + P_{bat-gridi} \leq P_{feed-in} \, . \tag{1.4}$$

The constraint (1.4) is the power balance equation for the feed in energy to the grid

$$P_{bat-loadi} + P_{bat-gridi} \leq \Delta P_{charge} B_{size} \, . \tag{1.5}$$

The constraint (1.5) ensures that the battery discharge rate is within the limits

$$P_{pv,bati} \leq \Delta P_{discharge} B_{size} \, . \tag{1.6}$$

The constraint (1.6) ensures that the battery charging rate is within the limits

$$E_{soci} \leq SoC_{upperlimit} \, . \tag{1.7}$$

The Constraint (1.7) ensures that the battery is not exceeding the upper limit.

Other research papers [4] concentrated on providing a constant charging and discharging rates with respect to the time of the day in order to reduce the evening peak.

`

Fig. 1.2(a) shows a load and PV profile during the day. During this period, a surplus power will be available between 9:00 a.m. and 3:00 p.m. According to the proposed strategy, the storage device will be charged over this period of surplus power. The charging rate will be increased from zero at the start of the period when the storage SOC is at the maximum depth of discharge at a slope of charging rate (SCR) until the SOC reaches a first threshold level ToS_1 . From this point, the storage will be charged at a constant charging rate, once the SOC reaches the second threshold. Less capacity of the storage is available and therefore the charging rate will be decreased using the same SCR, until the storage device attains the state of maximum charge at the end of period. The description of this strategy is as shown in Fig. 1.2.

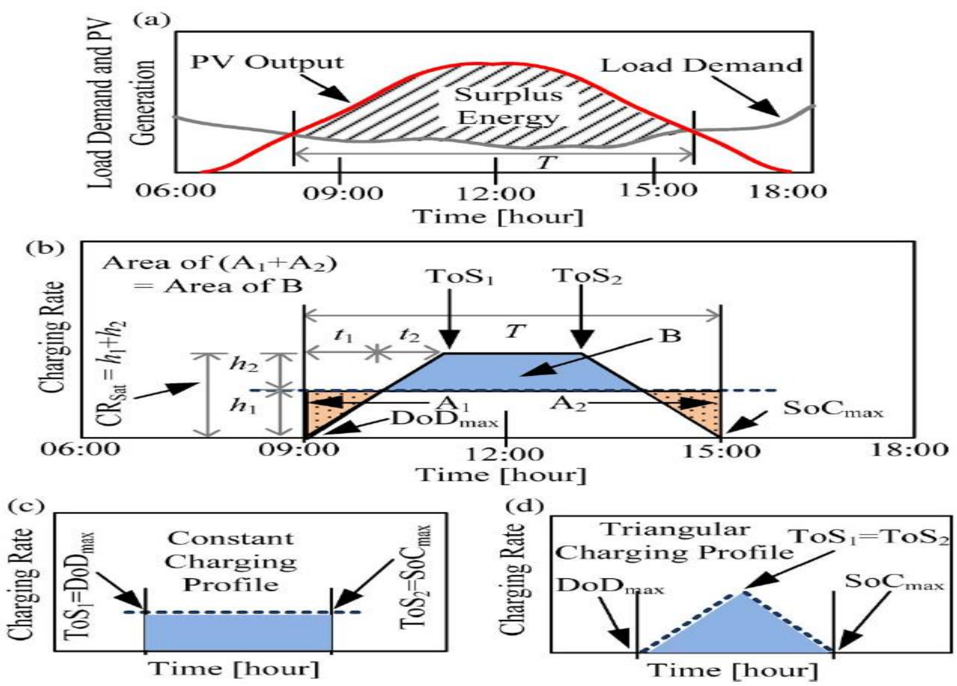

Fig. 1.2. Charging/Discharging Strategy [4]

`

These algorithms require perfect load prediction, which is very difficult for the residential load. Also, the energy charges and tariffs are based on the kWh consumption. However, in the recent times, the electric utilities have started charging the customers based on maximum demand [7] they consume during the on-peak hours. In such cases, the algorithms may not be effective enough to reduce the electricity tariff.

One of the major challenges for PV systems remains in the matching of the intermittent energy production with the dynamic power demand. A solution is to add a storage element to these nonconventional and intermittent power sources. The objective is to reduce operation costs by managing the power flows in the system. Researchers as in [8] have used dynamic programming to optimize power flow from grid-connected PV-battery systems during the on-peak hours. The formulation is as represented as,

$$Objective: Min(CF) = Min \sum_{t=t_o}^{T} [CR(t) + CP(t)]. \tag{1.8}$$

The objective function (1.8) aims at minimizing the final value of the cash flow, CF over the entire studied period. The cash flow is composed of the cash received, CR and the cash paid, CP.

The operating constraints are given by,

$$P_{grid}(t) = P_{PV}(t) + P_{BAT}(t) + P_{LOADS}(t). \tag{1.9}$$

The constraint (1.9) explains the power flow balance between grid, PV, battery and load

$$SOC^{\min} \leq SOC(t) \leq SOC^{\max}. \tag{1.10}$$

`

The constraint (1.10) explains the battery state of charge (SOC) during the operating stage

$$P_{BAT}^{\min} \leq P_{BAT}(t) \leq P_{BAT}^{\max}. \tag{1.11}$$

The constraint (1.11) explains the battery degradation

$$SOH(t) \geq SOH^{\min}. \tag{1.12}$$

The constraint (1.12) explains the state of health of the battery and minimum explains the replacement of battery

$$P_{grid}(t) \leq P_{grid}^{\max}. \tag{1.13}$$

The constraint (1.13) explains the maximum regulated limit for the exchange of power between grid and PV.

K-mean clustering [5] is a learning algorithm that generates an output based on the available dataset. The algorithm creates clusters in the form of subsets with respect to the variables. The K-mean clustering works on multi-dimensional clusters. Depending upon the tolerance the nearest two/three loads of the set points are assigned weights with the closest setpoint having a larger weight than the set point that is farther away. This algorithm is used in the research to select load from the historical dataset of the customer.

Weight calculation and final data point for a cluster of two data points using k-mean clustering is as mentioned below,

$$weight_{first\ point} = 1 - \frac{\text{devation of first point}}{\text{deviation of first point} + \text{deviation of second point}} \tag{1.14}$$

`

$$weight_{second\ point} = 1 - \frac{\text{devation of second point}}{\text{deviation of first point} + \text{deviation of second point}} \quad (1.15)$$

$$Final\ load = (weight_{first\ point}\text{first point}) + (weight_{second\ point}\text{second point})\ . \quad (1.16)$$

1.4 Thesis Outline

a. Chapter 1 provides the basic outline introduction, motivation for the project and the literature review that has been used for the project.

b. Chapter 2 provides the system description and the control method for operating the system on the rooftop of ERC.

c. Chapter 3 provides the algorithm for the optimal PV-battery sizing with regards to the demand charge component rate plan.

d. Chapter 4 provides the results of the transient test conducted on the rooftop system at the ERC.

e. Chapter 5 provides the algorithm for the load selection strategy and the operating strategy of the PV- battery system.

f. Chapter 6 provides an algorithm for the statistical analysis on the data obtained from the PV-battery system.

g. Chapter 7 contains the conclusions and the future work for the project.

CHAPTER-2 SYSTEM DESCRIPTION

SRP is currently funding the installation of a 6.36 kW photovoltaic system with integrated battery storage on the rooftop of ERC at ASU Tempe campus. The system includes a digitally controlled load-bank, which is programmed with an SRP provided load-shape derived from residential customer meter data.

2.1 Photovoltaic System

The PV system is made up of Polycrystalline solar panels each rated at 265 watts and efficiency of 19 %. The details of the panel are as given in Table 2.1.

Table 2.1. Specification of Solar Module

Specification	Value
Current	8.56 A
Voltage	30.96 V
Power	265 W

Fig. 2.1. Polycrystalline Solar Panels Rooftop ERC

The solar modules are arranged in two sub-arrays. Each sub-array consists of solar modules wired 3 in series by 4 in parallel.

2.2 Sunverge System

2.2.1 Charge Controller

The Sunverge system consists of two charge controllers one for each SUB-array described in Section -2.1. The MPPT charge controller specifications are as given in Table 2.2.

Table 2.2. Specification of Charge Controllers

Specification	Value
Maximum output power	3500 W
PV array operating voltage	140 V
Max PV array open-circuit voltage	150 V
Max array short-circuit current	60 A

Fig. 2.2. MPPT Charge Controllers

2.2.2 Inverter

A bidirectional 6 kW 120/240 V utility-interactive inverter is used. It is an adaptable single-phase and three-phase hybrid inverter with grid-tie functionality and dual AC power inputs. The specifications of the inverter are as given in Table 2.3.

Table 2.3. Specification of Bi-directional Inverters

Specification	Value
AC nominal power	6000 W
Battery charging current	100 A
Battery charging voltage	48 V
Peak-efficiency	95.4 %

Fig. 2.3. Utility Scale Inverter

2.2.3 Battery

A lithium-ion type (LiFePO4) rechargeable battery is used. The battery operates on 48V DC with Li-ion energy storage capacity of 19. 4 kWh.

2.3 Load Bank

2.3.1 Load Bank-1

The load bank is made up of 10 number of heating elements. The load bank works on 120 V and has a resistance of 15.3 ohms. At its full capacity, each heating element is capable of dissipating about 880 watts. The total capacity of this load bank is 8800 watts. This load bank works in steps of 880 watts. The electrical connection diagram is as shown in Fig. 2.4. A Raspberry Pi 3 microcontroller activates the relays.

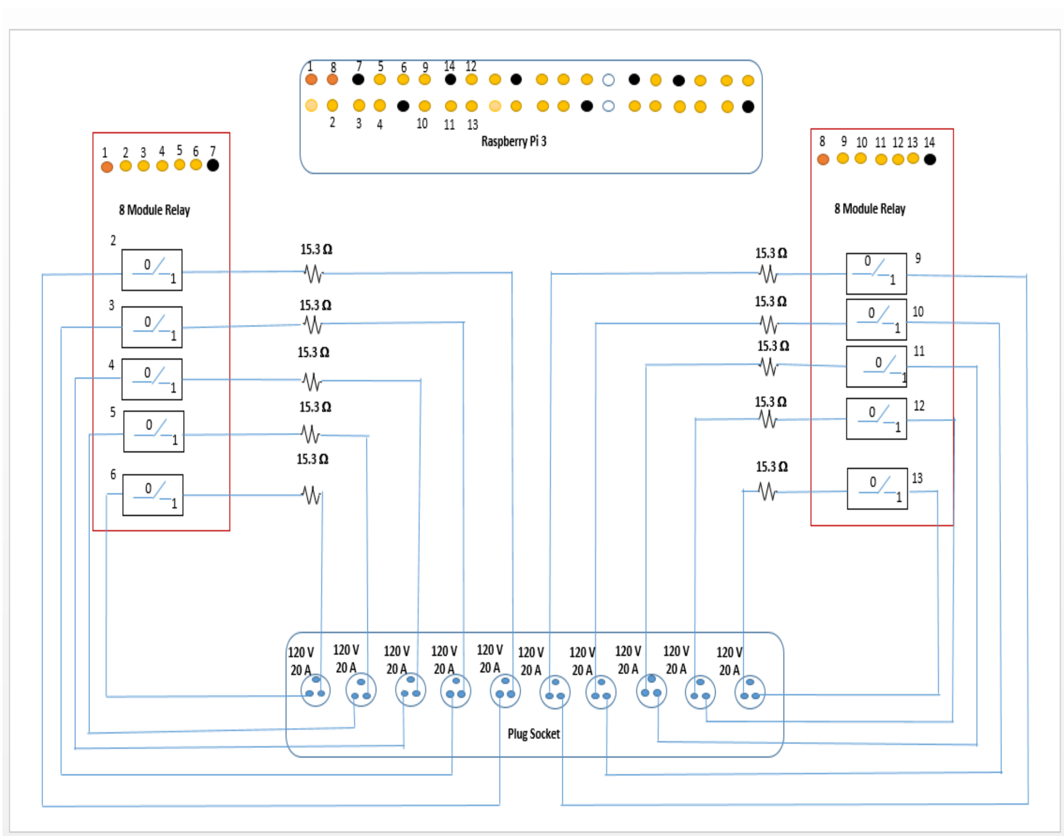

Fig. 2.4. Circuit Connection of Load Bank-1

Fig. 2.5. Load Bank-1 Rooftop ERC

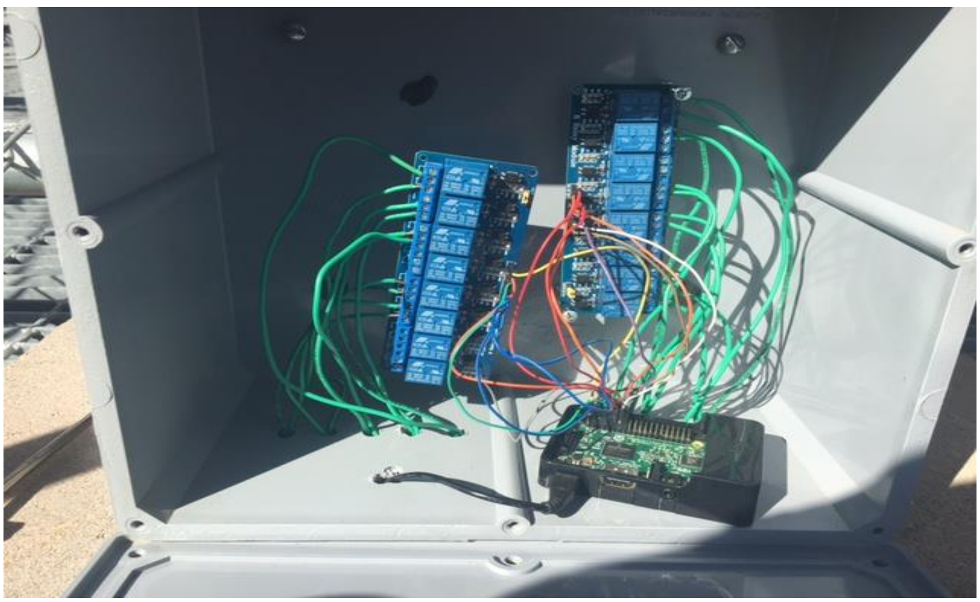

Fig. 2.6. Load Bank-1 Control Circuit

`

2.3.2 Load Bank-2

The load bank is made up of 100 ohms resistor. Two 100 ohms resistor are connected in series to form a set. There are 29 such sets in this load bank. A voltage of 120 V is applied and a current of 0.6 amps passes through each set. The power dissipating capacity of each set is 72 watts. So, the total capacity of this load bank is 2088 watts. The picture of this load bank is as shown in Fig. 2.8. This load bank works in steps of 72 watts. The electrical connection diagram is as shown in Fig. 2.7. The Raspberry Pi 3 energies the relays.

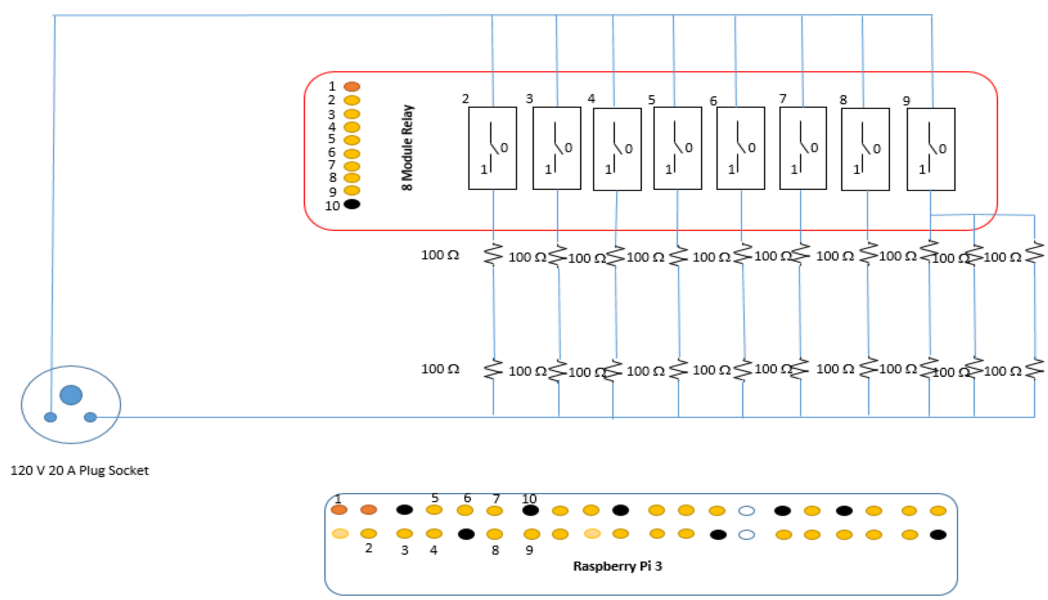

Fig. 2.7. Circuit Connection of Load Bank-2

Fig. 2.8. Load Bank-2 Rooftop ERC

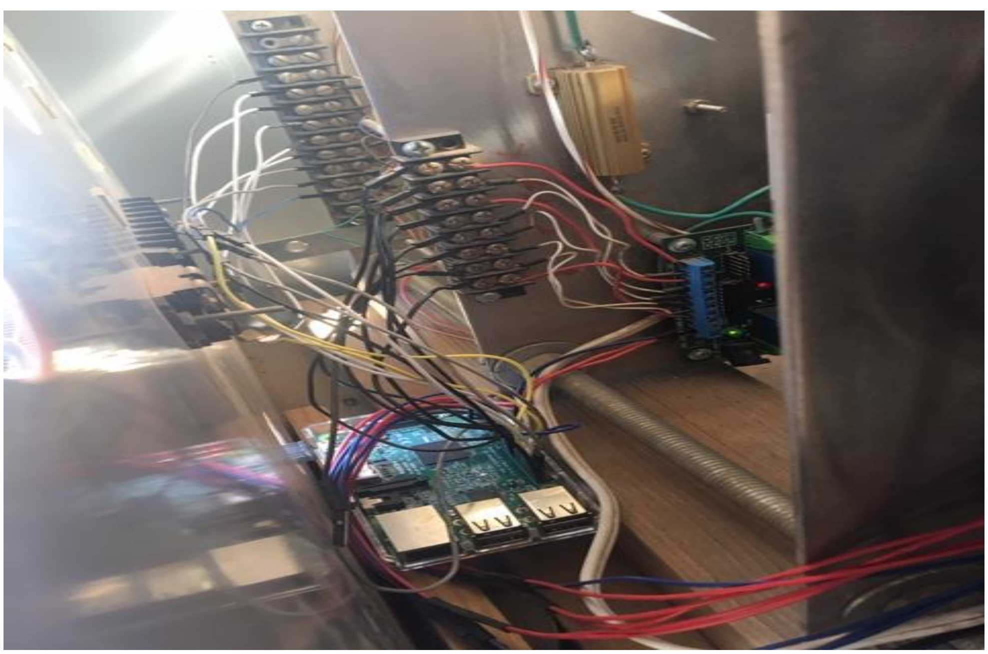

Fig. 2.9. Load Bank-2 Control Circuit

`

2.4 Microcontroller

A Raspberry PI 3 microcontroller is used to operate both the load banks defined in Sec. 2.3.1 and Sec. 2.3.2. The controller is as shown in Fig. 2.10. The available storage capacity of this controller is of 11 GB which enables to simulate residential load data. The Python 3 program is used to run a code to give a low and high input signals to the controller pin. The controller pins are used to energize the digital relays as shown in Fig. 2.12. The pin diagram is as shown in Fig. 2.11. The flowchart shown in Fig. 2.13 demonstrates the steps required to program the code. A sample code is attached in the appendix.

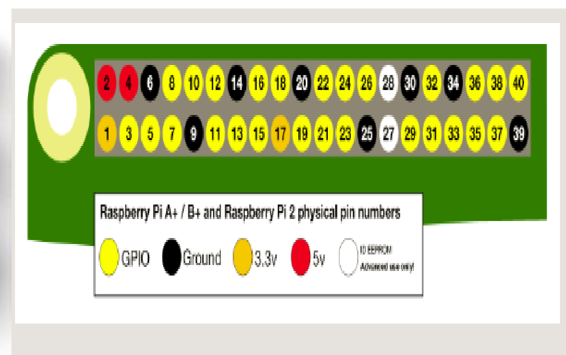

Fig. 2.10. Raspberry Pi 3 Fig. 2.11. Raspberry Pi 3-pin Diagram

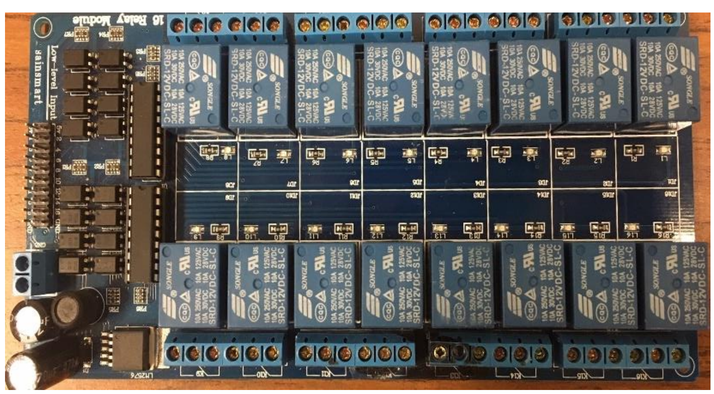

Fig. 2.12. Digital Relay

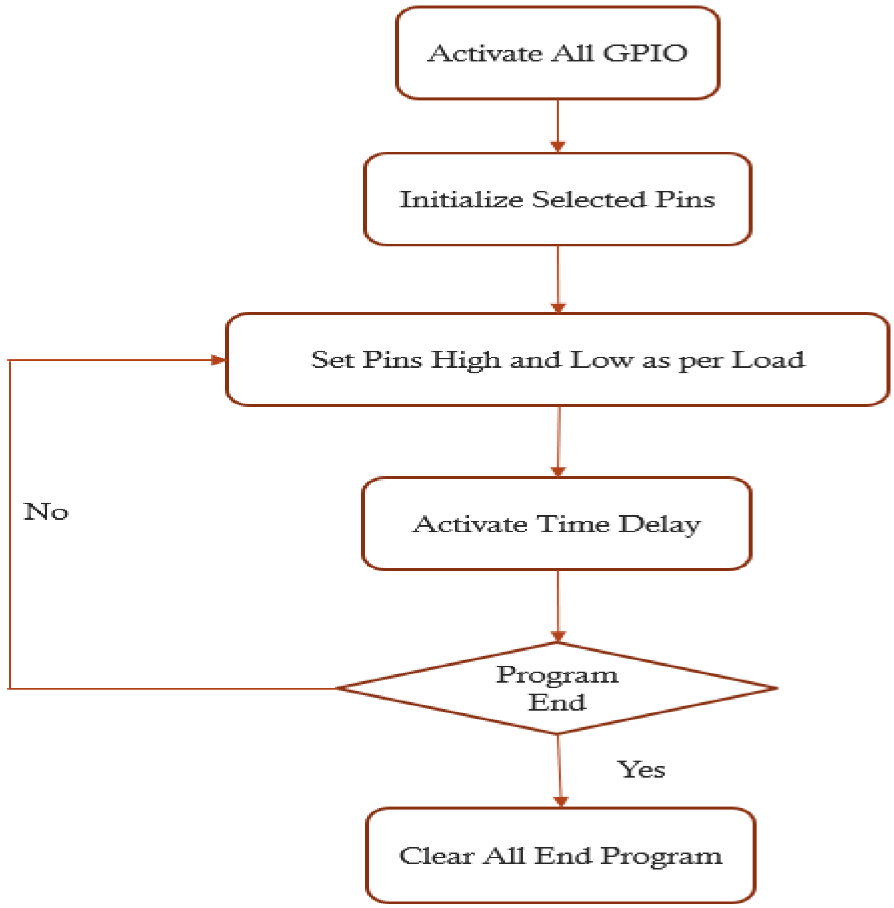

Fig. 2.13. Flowchart Depicting the Python Code Operation

2.5 Load Bank Validation

2.5.1 Step Load

The Fig. 2.14 shows a step curve having a step increase of 1 kW each until 6 kW. The 1 kW step is achieved by triggering one relay output high from load bank-1 and two relay outputs high from load bank-2. Each step is given a delay of 900 seconds.

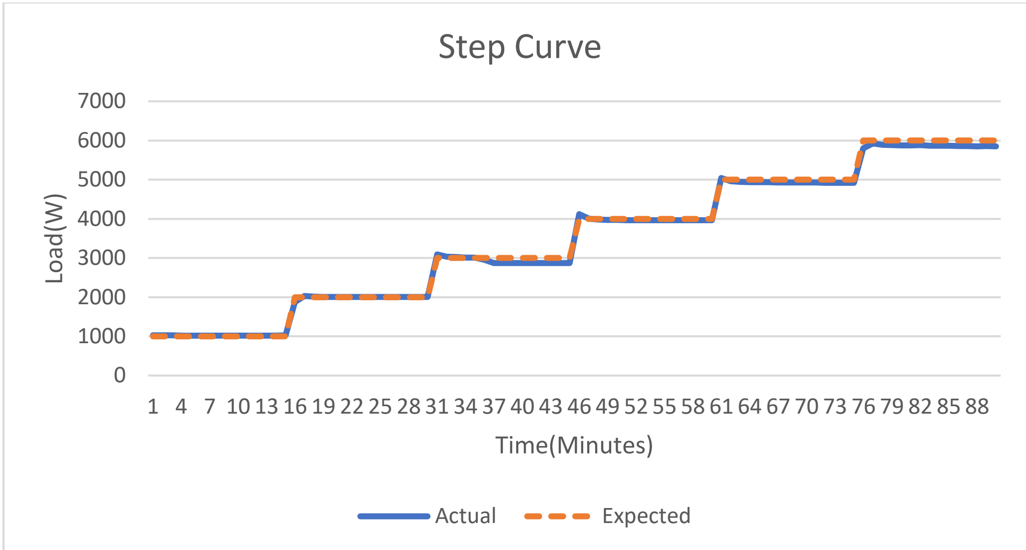

Fig. 2.14. Residential Load with a Step Increase

2.5.2 Hourly Load

The Fig. 2.15 shows an hourly load curve applied for a period of 24 hours to demonstrate how closely and accurately an hourly load curve can be followed.

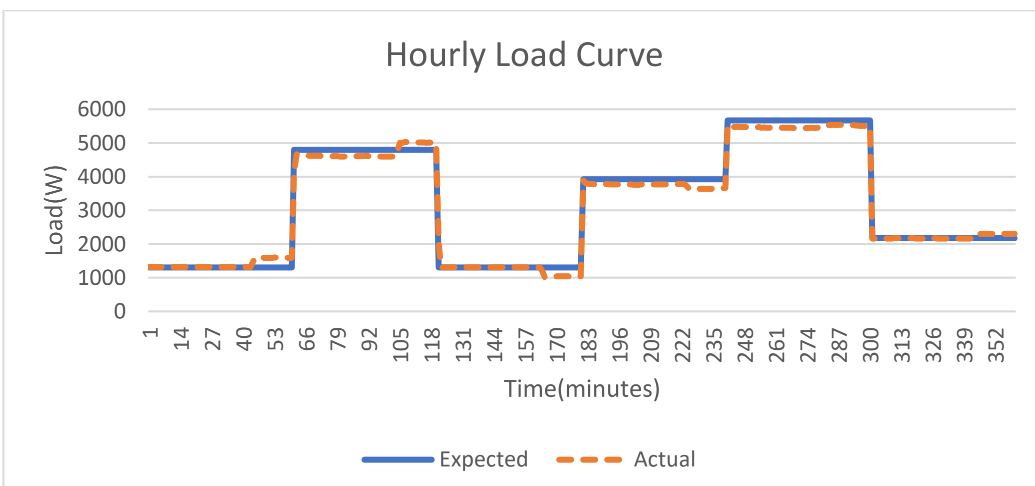

Fig. 2.15. Residential Load with an Hourly Interval

`

2.5.3 15 Minute Interval Load

The Fig. 2.16 shows a 15-min load curve applied for a period of 75 minutes to demonstrate how closely and accurately a 15-min load curve can be followed.

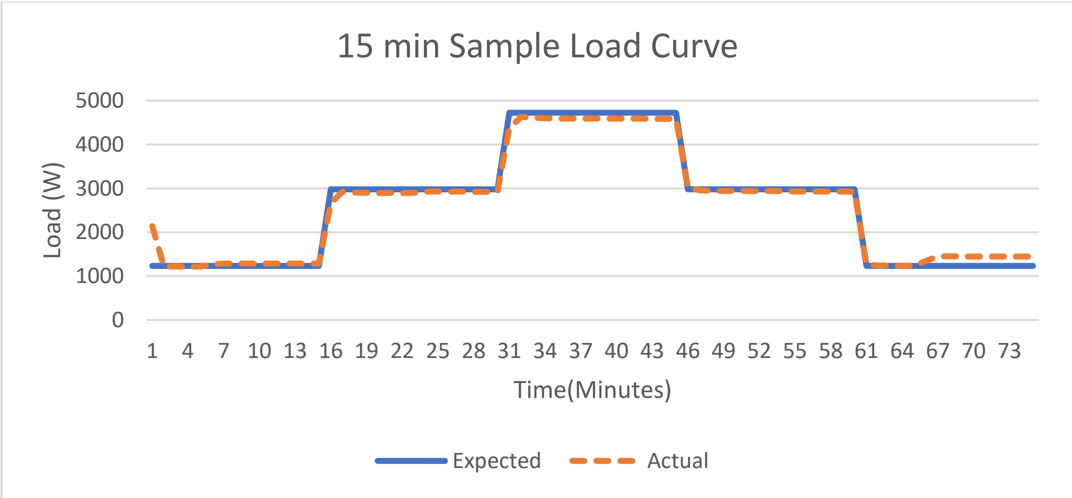

Fig. 2.16. Residential Load with a 15 - Minute Interval

2.5.4 One Minute Interval Load

The Fig. 2.17 shows a 1 min load curve applied for a period of 7 minutes to demonstrate how closely and accurately a 1 min load curve can be followed.

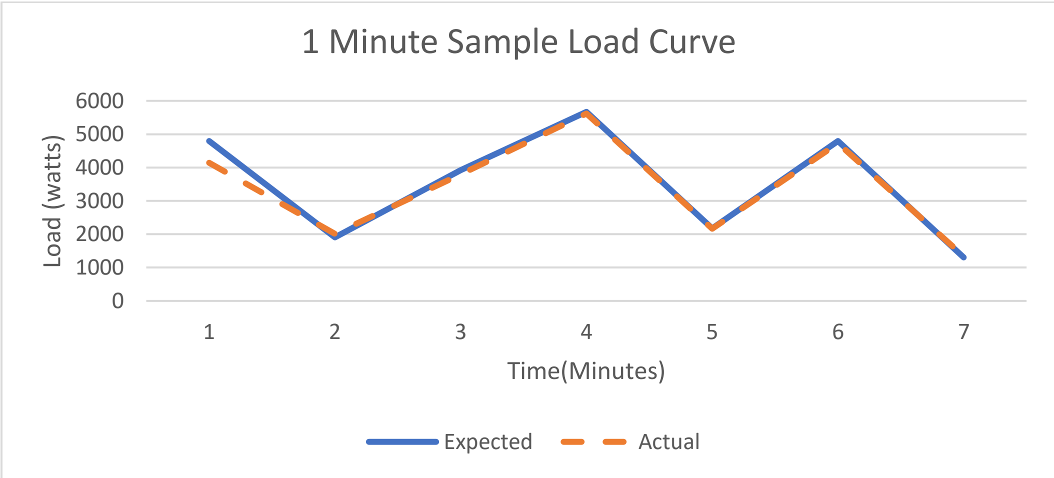

Fig. 2.17. Residential Load with a 1 - Minute Interval

CHAPTER-3 ECONOMIC ANALYSIS

3.1 Introduction

Economic analysis is implemented to find an optimal size of battery and PV system for a residential customer with respect to their load. The analysis is performed on each pair of PV, battery system and the pair that is most profitable after 25 years is considered to be the optimal system. The analysis is performed using the one complete yearly load profile of a residential customer and PV data. The algorithm is implemented using MATLAB.

3.2 Assumptions

The list of assumptions considered for this algorithm is as mentioned,

- The rate of interest is considered to be 4%.
- The inverter is considered to be at 100% efficiency.
- The on-peak-hours are from 1 p.m. to 8 p.m.
- The battery is fully charged from grid and PV-energy during off-peak hours.
- The battery cost is considered to be $350/kWh.
- The PV system cost is considered to be $2.532/W.
- The battery is only used to serve the load during the on-peak hours.
- The lifetime of the PV system is considered to be 25 years.
- The lifetime of the battery is taken as 5000 cycles.
- The inverter level is considered to be same as the PV level.

`

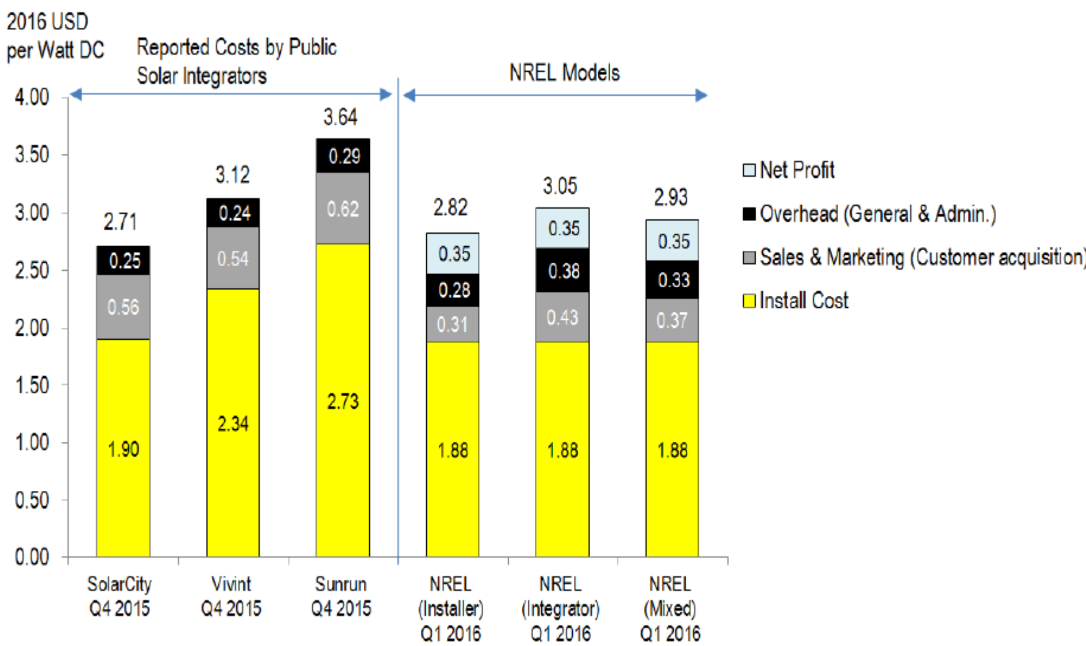

Fig. 3.1. PV System Cost [22]

3.3 Electricity Tariff

3.3.1 Basic Rate Plan

The customer initially stays on the basic rate plan, as shown in Table 3.1. Once the customer sets up his own residential PV or any type of electricity generation, he is charged from demand charge component rate plan as shown in Table 3.2. The profit is a measure of savings that the customer makes after shifting to the new rate plan.

The electricity price for the given load using the basic plan is calculated as,

$$if\ T_l < 700 \tag{3.1}$$

$$E_{bill} = E_c^1 \times T_l \tag{3.2}$$

$$if\ T_l > 700\ and\ T_l < 2000 \tag{3.3}$$

$$E_{bill} = E_c^1 \times 700 + E_c^2 \times (T_l - 700) \quad (3.4)$$

$$if\ T_l > 2000 \quad (3.5)$$

$$E_{bill}^{basic} = E_c^1 \times 700 + E_c^2 \times (1300) + E_c^3 \times (T_l - 2000) \quad (3.6)$$

where

E_{bill}^{basic} = Electricity bill for a particular month

E_c^1, E_c^2, E_c^3 = Electricity charge for a particular month described in Table 3.1

T_l = Total load for a particular month.

Table 3.1. Basic Rate Plan

Specification	Summer ($)	Summer-peak ($)	Winter ($)
Monthly service charge	18.50	18.50	20.00
Per kWh charges (first 700 kWh)	0.1102	0.1168	0.0803
Per kWh charges (700-2000 kWh)	0.1121	0.1180	0.0803
Per kWh charges (above 2000 kWh)	0.1226	0.1331	0.0803

3.3.2 Demand Charge Component Rate Plan

The demand charge component rate plan is used to calculate the annual electricity cost for the customer after the installation of PV-battery system or any other source of electricity generation at the home.

The electricity price for the given load using the demand charge component rate plan is calculated as,

$$p_d = \max((O^{new}_{on-peak})^i + E^i_l)_{i = \text{month start} - \text{month end}} \tag{3.7}$$

$$if \;\; P_d < 3kW \tag{3.8}$$

$$DC = P_d \times ED^1 \tag{3.9}$$

$$else \;\; P_d > 3kW \;\; and \;\; P_d < 10kW \tag{3.10}$$

$$DC = 3 \times ED^1 + \left(P_d - 3\right) \times ED^2 \tag{3.11}$$

$$else \;\; DC = 3 \times ED^1 + (7) \times ED^2 + \left(P_d - 10\right) \times ED^3. \tag{3.12}$$

Table 3.2. Demand Charge Component Rate Plan

Specification	Summer ($)	Summer-peak ($)	Winter ($)
Monthly service charge	30.94	30.94	32.44
Per kWh charges (on-peak-charges)	0.0486	0.0633	0.0430
Per kWh charges (off-peak-charges)	0.0371	0.0423	0.0390
Peak demand charges/kW (first 3 kW)	8.03	9.59	3.55
Peak demand charges/kW (next 7 kW)	14.63	17.82	5.68
Peak demand charges/kW (above 10 kW)	27.77	34.19	9.74

`

3.4 Load

The algorithm is run for different load profiles collected from sources. The results in this chapter are shown for the load details mentioned in Table 3.3.

Table 3.3. Residential Load Details

Specification	Value
Total annual energy	14259.732 kWh
Peak demand	10.82 kW

3.5 Customer Electricity Payment Prediction

The payment made by the customer towards the electric utility for the next 25 years is predicted using the data obtained from EIA (U.S. Energy Information Administration) . The increase in fuel costs is related to the increase in the electricity tariff. The percentage increase or decrease in each year is calculated from Fig. 3.2. The curve in the graph indicates residential end-use price.

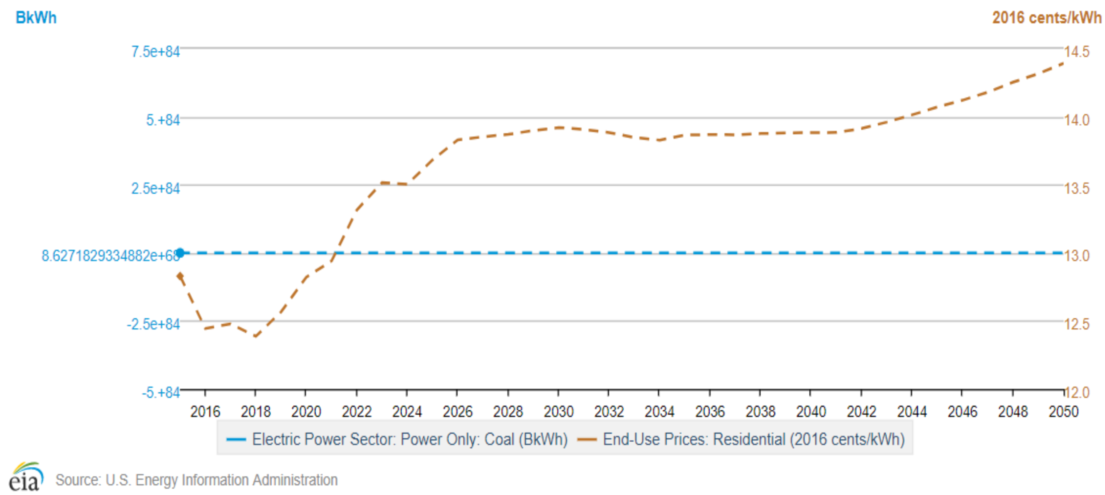

Fig. 3.2. Electricity Price Prediction [20]

3.6 Algorithm Description

The battery is reserved only for the use during the on-peak hours and stays at a fixed state of charge during the off-peak hours. The inverter is assumed to be of the same size as the PV system. The load is divided into three parts. The on-peak-power, off-peak-power and excess load. The excess load is the amount of on-peak-power above the inverter level that must be supplied by the grid and will account as a part of peak-demand power.

The battery is dispatched in such a way that the grid power consumed during the on-peak hours is minimal. If the sum of on-peak-energy for a day, $\left(SE_{on-peak}\right)^{d}$ is less than the total available battery capacity E_b at the start of on-peak-hours, then the new on-peak-energy requirement $O_{on-peak}^{new}$ is zero and the battery is self-sufficient to support the system else the available battery capacity is dispatched in a proportionate manner by dividing the i^{th} hour of the on-peak-energy requirement to the sum of the on-peak-energy requirement allowing the battery to be completely discharged at the end of on-peak-hour. Then a threshold limit is obtained for a month which is the maximum of on-peak requirement during the month. Any on-peak energy above the threshold limit is satisfied by the battery and on-peak energy below the threshold limit is satisfied by the grid.

On-peak energy dispatch logic

$$if\left(SE_{on-peak}\right)^{d} < E_{b} \tag{3.13}$$

$$O_{on-peak}^{new} = 0 \tag{3.14}$$

`

$$else\left(O_{on-peak}\right)^{i}=\left(O_{on-peak}\right)^{i}-\left(\frac{\left(O_{on-peak}\right)^{i}}{\left(SE_{on-peak}\right)^{d}}\right)\times E_{b} \tag{3.15}$$

$$\text{Threshold limit} = \max\left(O_{on-peak}^{new}\right)^{i=1 to\ n} \tag{3.16}$$

$$\left(O_{on-peak}^{new}\right)^{i} = \text{Threshold limit}\ . \tag{3.17}$$

3.7 Sizing of PV-Battery System

The analysis described in Section 3.6 is run for different PV-battery pairs. All the terms mentioned below hereafter in this sub-section will form an array of 15 by 50 matrix where 15 indicates the PV size starting from 1 to 15 kW and 50 indicates the battery size starting from 1 to 50 kWh. Each value mentioned below represent the annual values of the system.

a. Self –consumption –rate

Self-consumption-rate is defined as the ratio between sum of the PV energy which is used by load and for charging the battery divided by the total PV produced

$$S_{co}=\frac{PV_{load}+PV_{battery}}{PV_{total}}\ . \tag{3.18}$$

b. Self-sufficiency-rate

Self-sufficiency-rate is defined as the ratio between the PV energy which is used by load and battery energy used by load divided by the total load

$$S_{cr} = \frac{PV_{load} + L_{battery}}{L_{total}}. \tag{3.19}$$

c. Battery-cycle

Number of battery cycles are calculated by dividing the amount of total energy discharged by the useful battery capacity

$$n_b = \frac{EN_d}{E_u}. \tag{3.20}$$

d. Total system cost

The total cost of the system is the cost of the battery and PV system put together

$$PV_{\text{cost}} = PV_{per - watt - \text{cost}} \times watts + PV_{\text{maintainence}} \tag{3.21}$$

$$Battery_{\text{cost}} = Battery_{\text{cost} / kWh} \times kWh \times n_n + Battery_{\text{maintainence}} \tag{3.22}$$

$$\text{Total system cost} = PV_{\text{cost}} + Battery_{\text{cost}} \tag{3.23}$$

n_n is the number of battery replaced during the period of 25 years.

e. Feed-in-tariff

The feed-in-tariff savings are defined as follows,

$$Feed - in - tarrif = Electricity\ price \times PV_{total} \times \left(1 - S_{co}\right). \tag{3.24}$$

`

f. Pay-back-period

The total savings on the system is the sum of savings on the electricity bill and the reimbursement from the feed-in-tariff. The annual savings at the end of each year are invested in a recurring savings account that provides an interest rate of 4%. At the end of 25 years the net value of the savings account will be amounted to total savings. Pay-back-period is obtained by dividing the system cost and the total savings divided by the lifetime of PV. The total savings in a year is the sum of the savings from bill and the feed-in-tariff

$$\text{Profit} = \text{Total savings} - \text{Total system cost} \tag{3.25}$$

$$\text{Pay} - \text{back period} = \frac{\text{Total system cost}}{\frac{\text{Total savings}}{25}} \tag{3.26}$$

3.8 Final Optimal System

The system with a pay-back period of fewer than 25 years and with the maximum profit is considered as the optimal system. Using this algorithm for the load mentioned in the Section 3.4 the optimum solution is found out to be a 7 kW PV system with a 14 kWh battery.

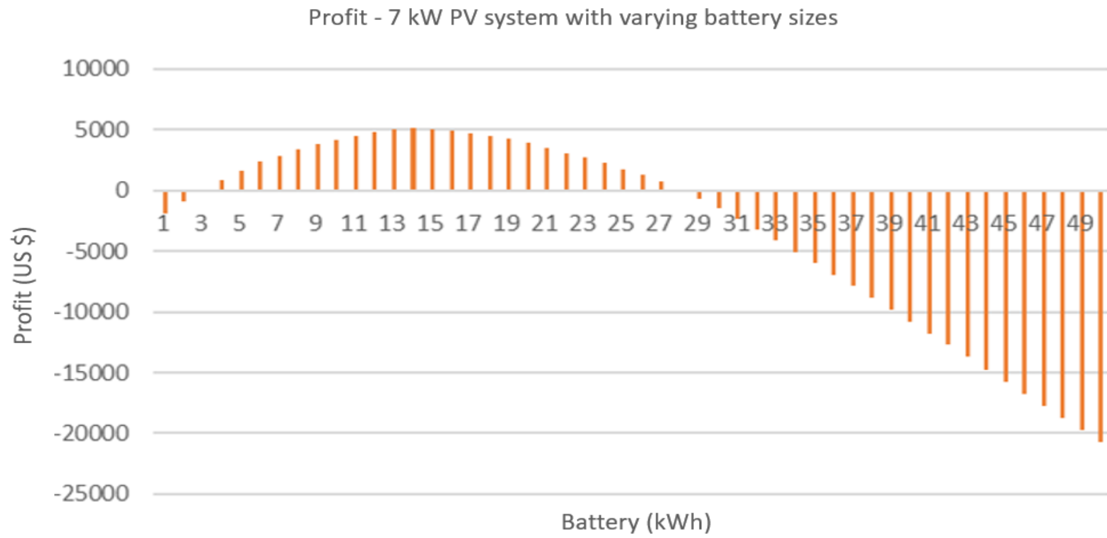

Fig. 3.3. Profit-7 kW PV System with Varying Battery Sizes

`

CHAPTER 4-TRANSIENT ANALYSIS

4.1 Introduction

The transient test is performed on the system using a 3.5-ton residential air conditioner. This test demonstrates the capability of the PV-battery system in providing the transient energy. The test is performed in two stages on-grid mode and off-grid mode. The results of the on-grid mode are shown in Section 4.2, the results of the off-grid mode are shown in Section 4.3. In Section 4.4 the readings are presented with varying battery state of charge (SOC 20%-80%).

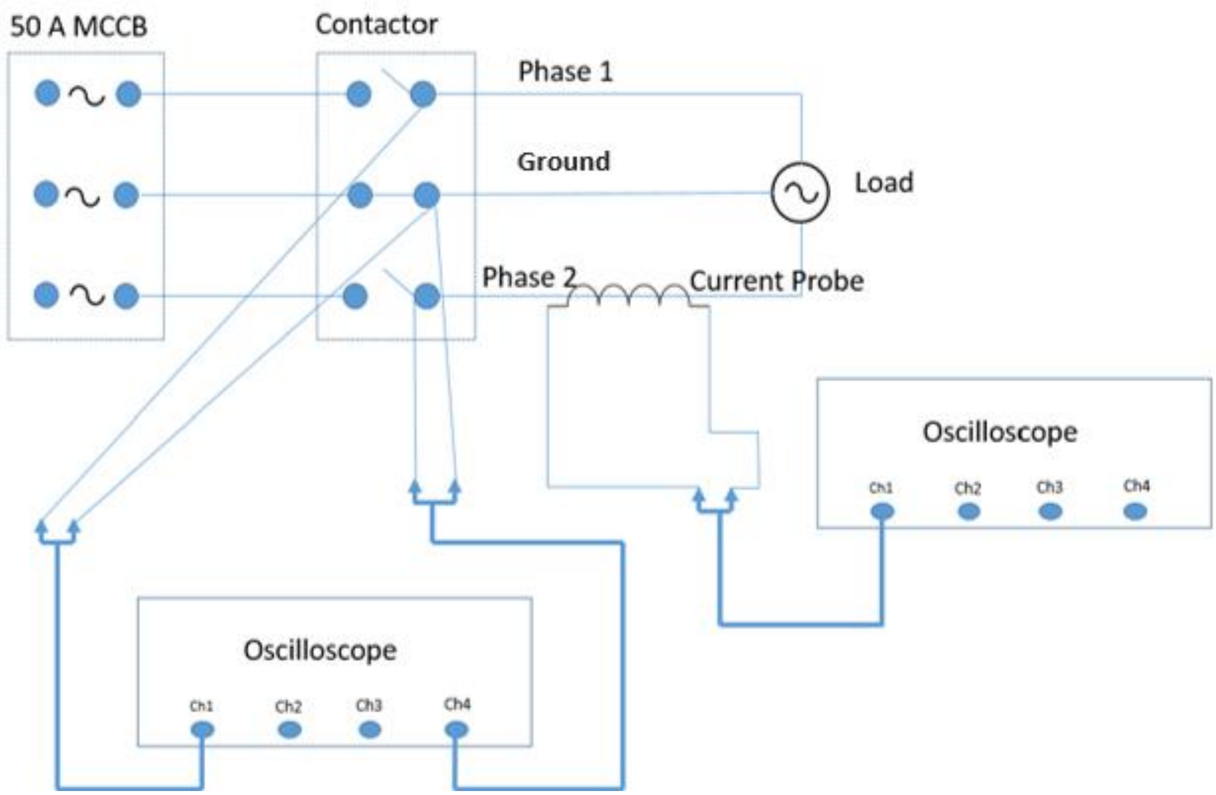

Fig. 4.1. Connection Diagram for Transient Test

4.2 On-Grid Mode

- The first test is performed on 12/19/2016 in between 9:30 a.m. – 11:00 a.m.
- The test is performed in grid-connected mode on a clear sky day with the battery charged to 60 % of its capacity.
- The 3.5-ton residential air conditioner is connected to main load panel at first and the transient test is performed.
- The test is repeated by switching the test load to critical load panel.
- All the other loads connected to the system are switched off to ensure the panels are not overloaded beyond the limit.
- Calibration for current measurement 10 mV /A.
- The bandwidth of the current probe used is 1.5 kHz.

4.2.1 Explanation

- All the load current readings from Fig. 4.6-Fig. 4.9 have a peak transient around 2 V and the current probe is calibrated at 10 mV /A so the peak current is 200 A.
- All the grid current readings from Fig. 4.2-Fig. 4.5 have a peak transient around 1.87 and the current probe is calibrated at 10 mV /A so the peak current is 187 A.
- From Fig. 4.10-Fig. 4.15 it can be inferred that no transient in voltage is observed except for a slight voltage drop initially.
- Initial voltage peak is 155 V.
- The steady state voltage peak is 170 V.

4.2.2 Grid Current

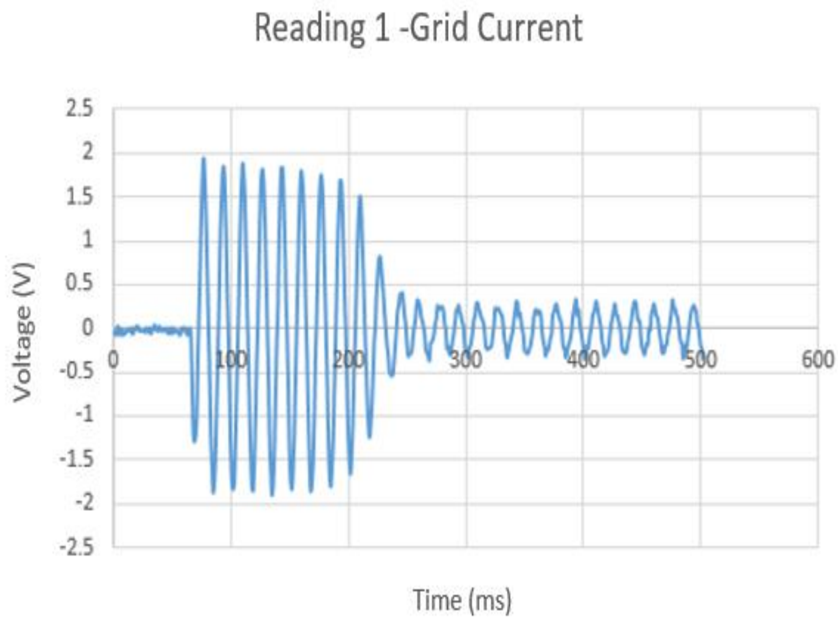

Fig. 4.2. Reading 1 Grid Current (Current Probe Reading)

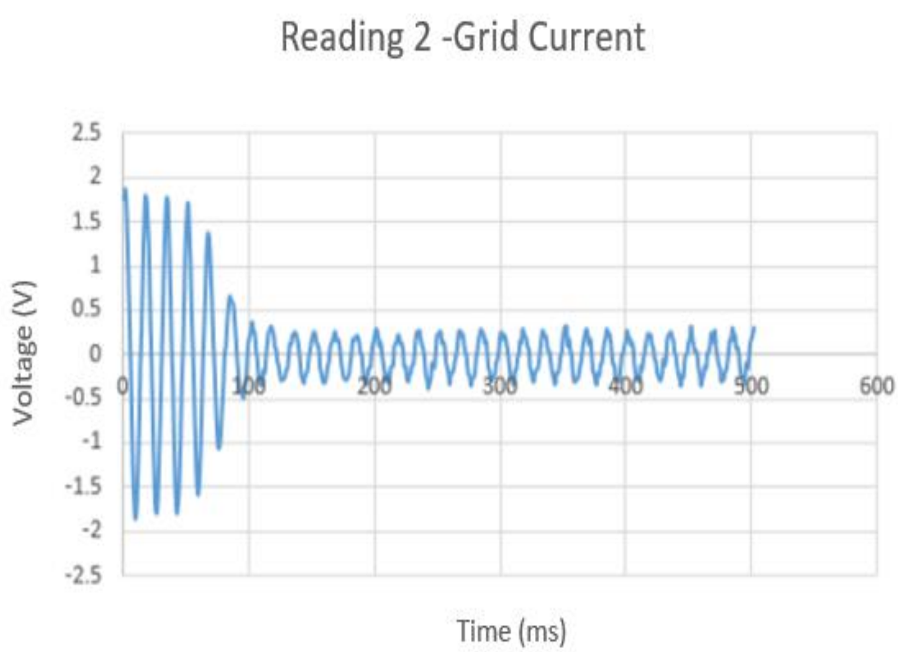

Fig. 4.3. Reading 2 Grid Current (Current Probe Reading)

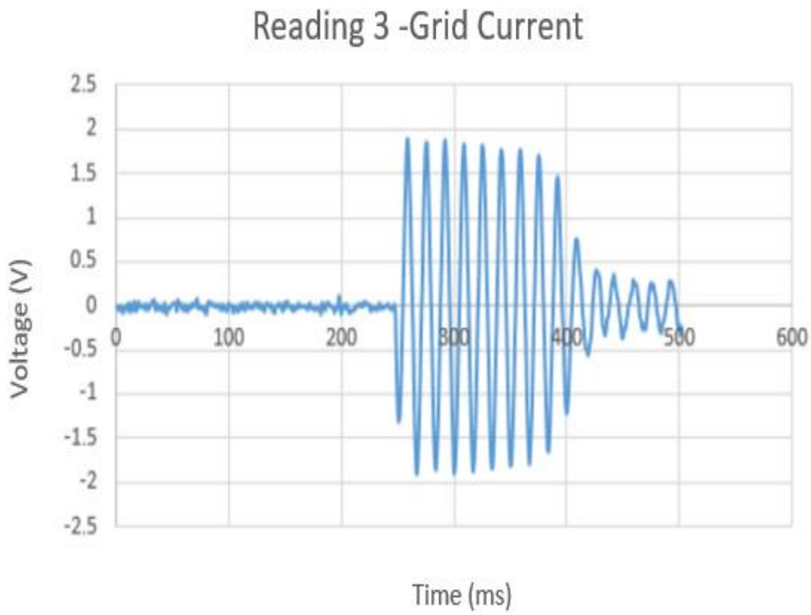

Fig. 4.4. Reading 3 Grid Current (Current Probe Reading)

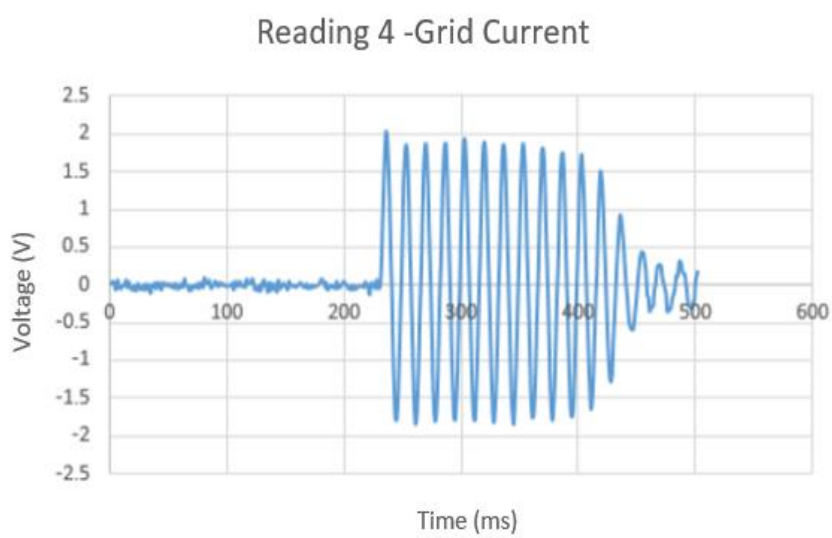

Fig. 4.5. Reading 4 Grid Current (Current Probe Reading)

4.2.3 Load Current

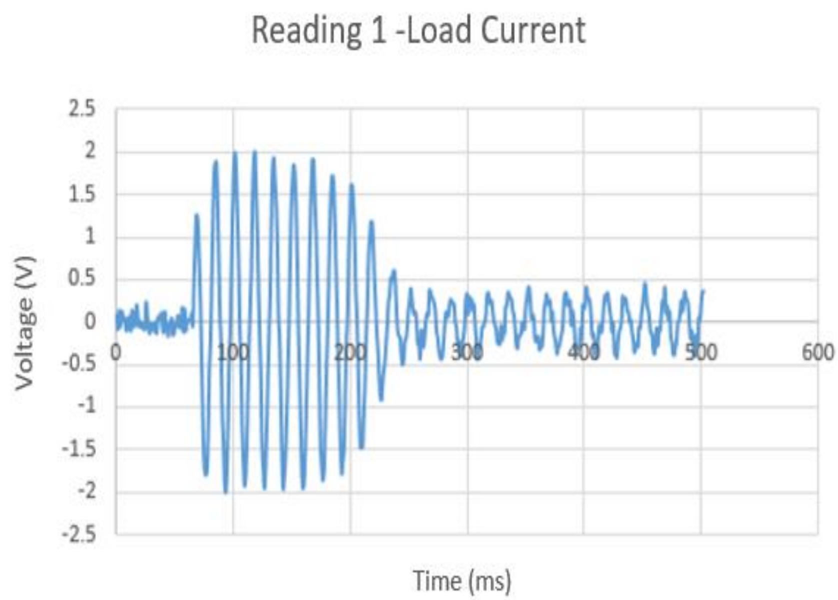

Fig. 4.6. Reading 1 Load Current (Current Probe Reading)

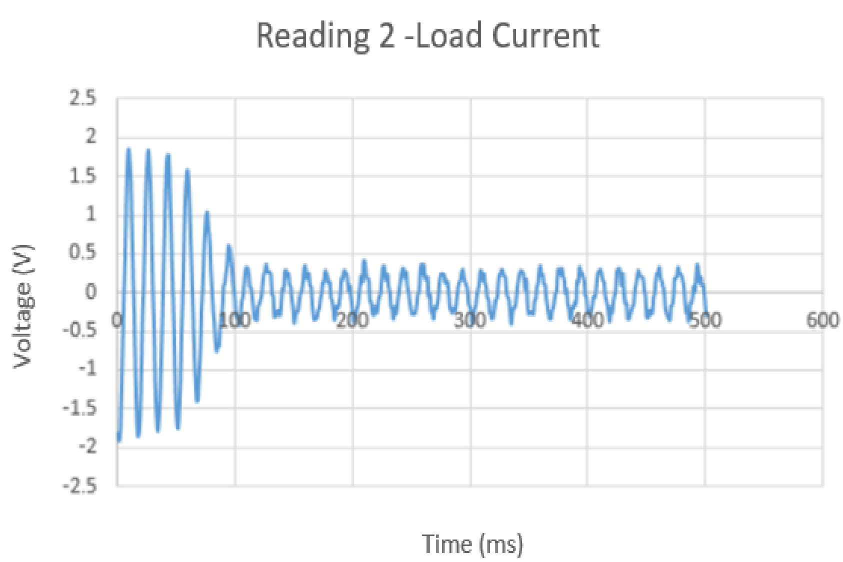

Fig. 4.7. Reading 2 Load Current (Current Probe Reading)

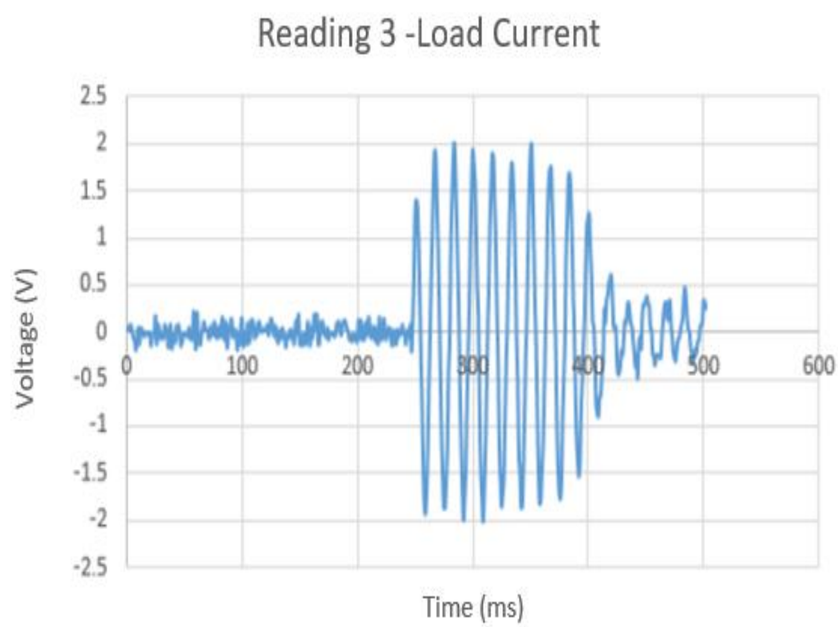

Fig. 4.8. Reading 3 Load Current (Current Probe Reading)

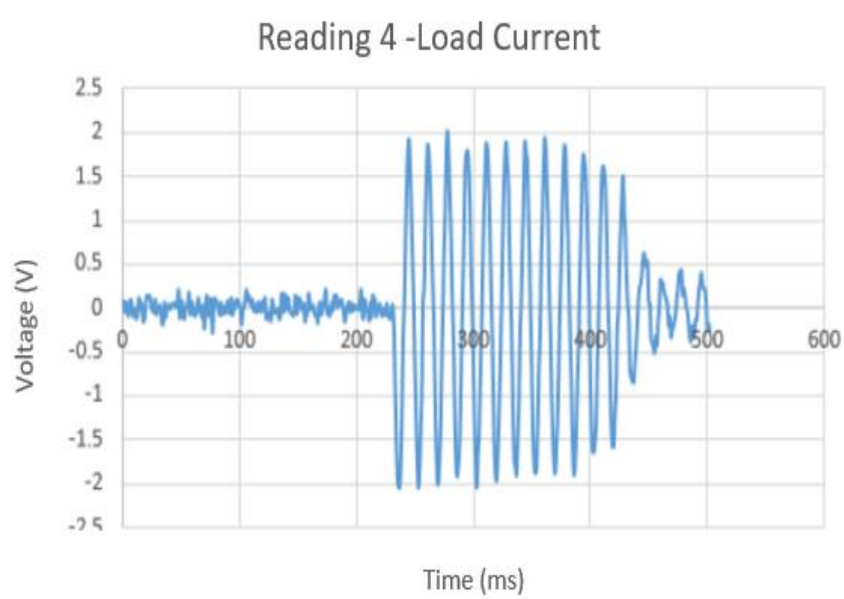

Fig. 4.9. Reading 4 Load Current (Current Probe Reading)

`

4.2.4 Voltage Reading

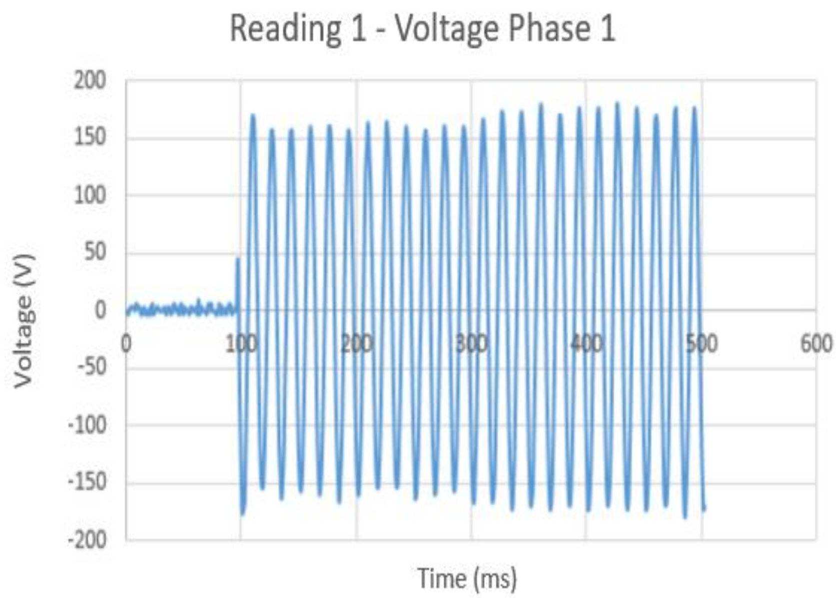

Fig. 4.10. Reading 1 - Voltage Phase 1 - On Grid Mode

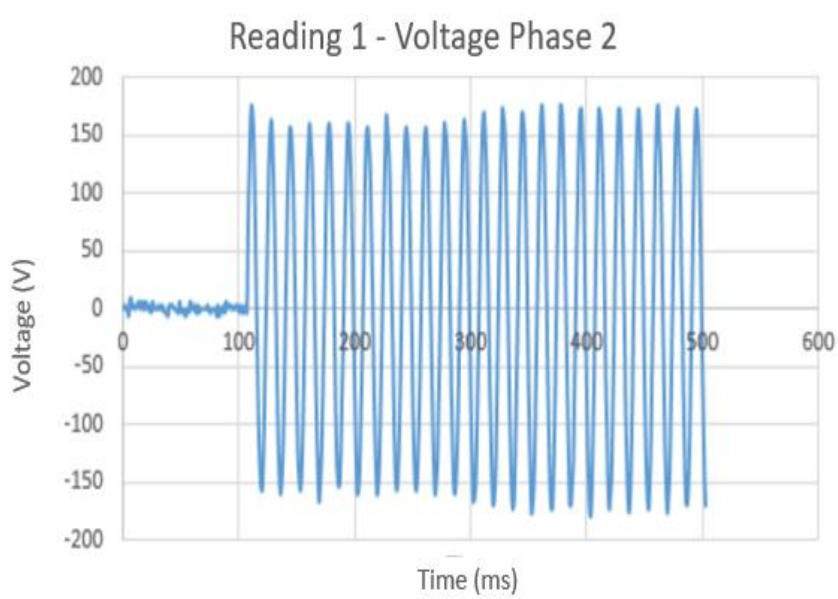

Fig. 4.11. Reading 1 - Voltage Phase 2 - On Grid Mode

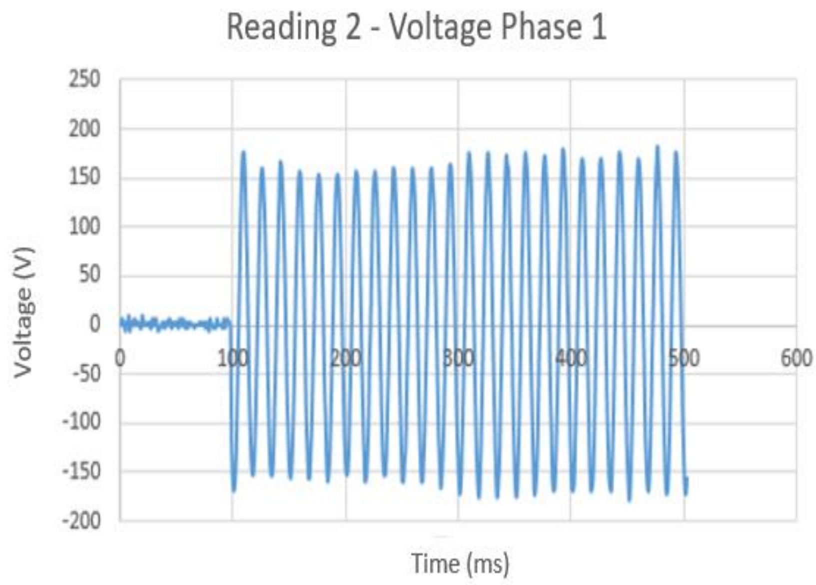

Fig. 4.12. Reading 2 - Voltage Phase 1- On Grid Mode

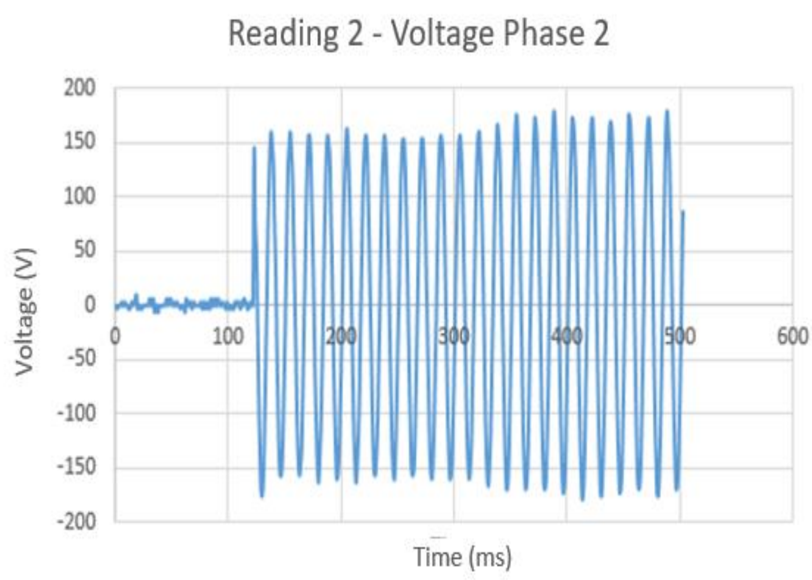

Fig. 4.13. Reading 2- Voltage Phase 2 - On Grid Mode

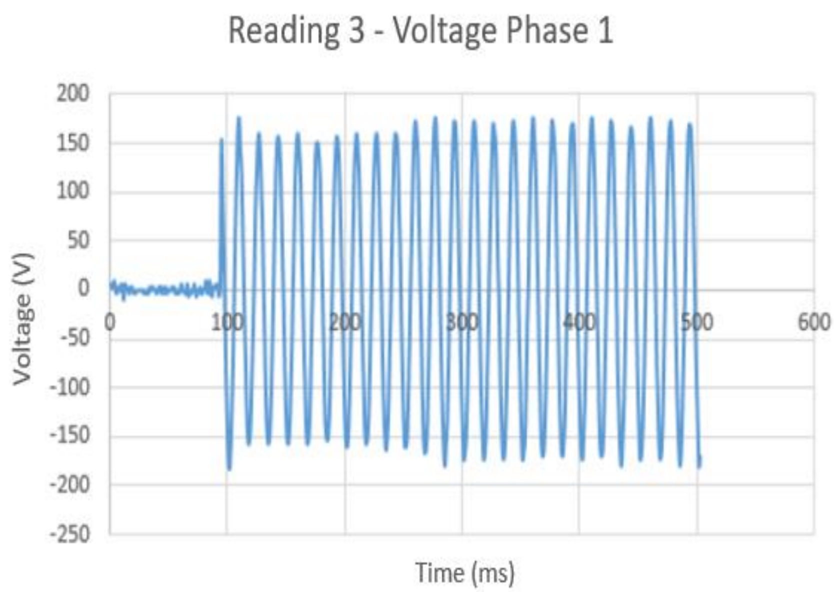

Fig. 4.14. Reading 3- voltage phase 1- On grid mode

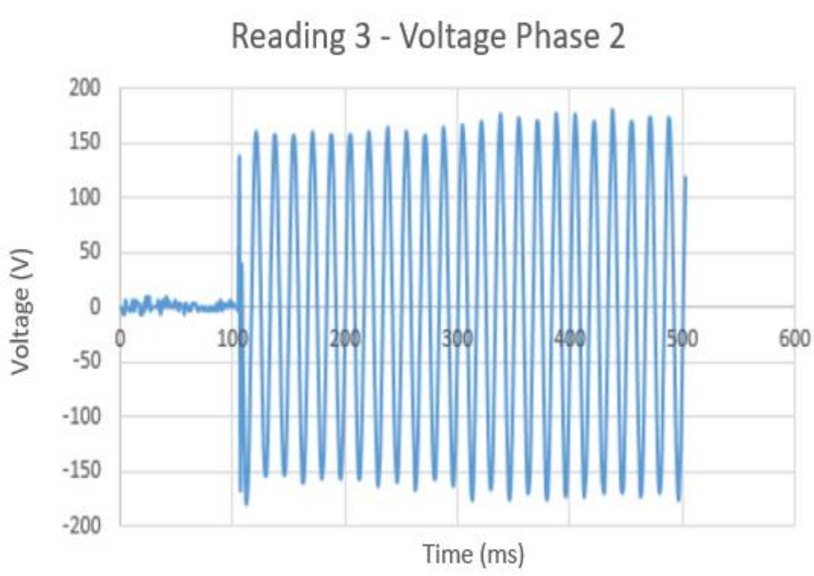

Fig. 4.15. Reading 3 - Voltage Phase 1 - On Grid Mode

`

4.3 Off-Grid Mode

- The test is performed on 02/23/2017 in between 6:30 p.m. – 7:15 p.m.
- The test is performed in an off-grid mode with the battery charged to almost 50 % of its capacity.
- The 3.5-ton residential air conditioner is connected to critical load panel.
- One current measurement (load terminal) and one voltage reading (load terminal voltage) are taken.
- All the other loads connected to the system are switched off to ensure the panels are not overloaded beyond the limit.
- Calibration for current measurement 10 mV /A.
- The Bandwidth of the current probes used are 1.5 kHz and 25 kHz.

4.3.1 Explanation

- The drop-in case of the off-grid mode is above the +/- 10 % limits with the lowest peak around 90 V.
- The load current as shown in Fig. 4.16-Fig. 4.19 is measured using a high bandwidth transformer so that fast transients are captured. The bandwidth of the transformer used is 25 kHz.
- The peak transient current in grid-connected mode is very high as compared to off-grid mode

  Peak transient current (grid connected mode) = 200 A

  Peak transient current (off-grid mode) = 110 A.

- The number of transient cycles is very high in off-grid mode as compared to grid- connected mode

  Transient cycles (Grid-connected mode) = 13

  Transient cycles (Off-grid mode) = 22-23.

- There is no significant voltage drop at the load terminal in grid-connected mode, however in the off-grid mode from Fig. 4.20-Fig. 4.23 the voltage drop lasted for 22-23 cycles upon starting of the 3.5-ton air conditioner.

- All the load currents from Fig. 4.16-Fig. 4.19 are shown as per current probe readings.

4.3.2 Load Current

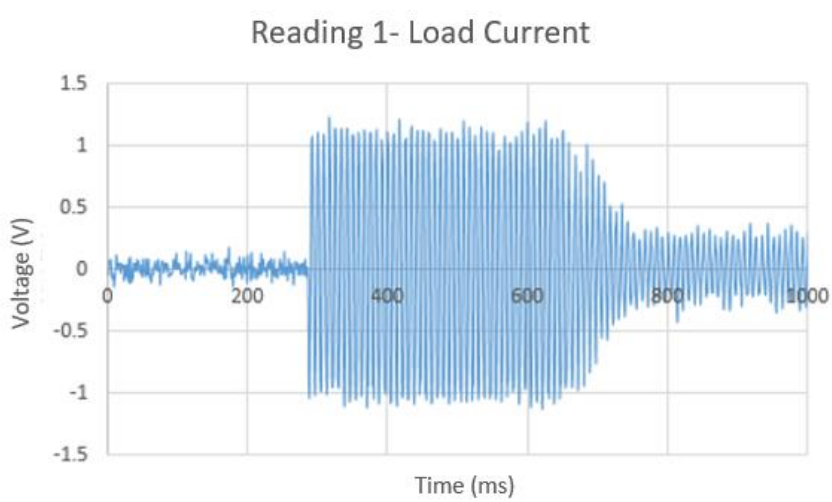

Fig. 4.16. Reading 1 Load Current- Off Grid Mode

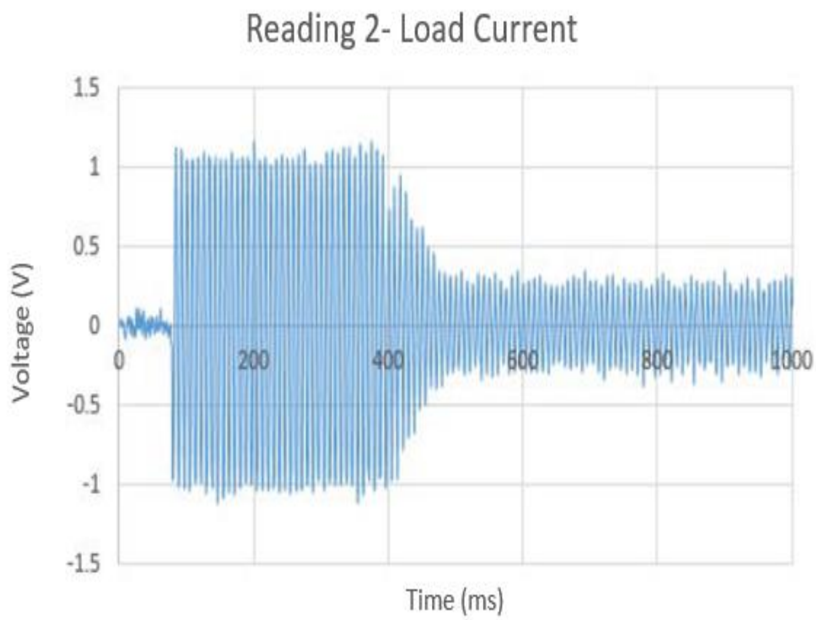

Fig. 4.17. Reading 2 Load Current - Off Grid Mode

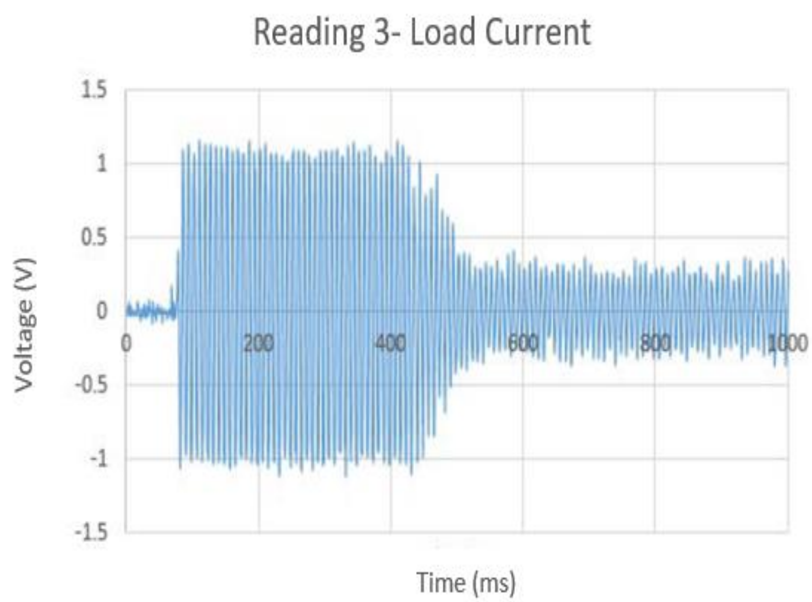

Fig. 4.18. Reading 3 Load Current - Off Grid Mode

`

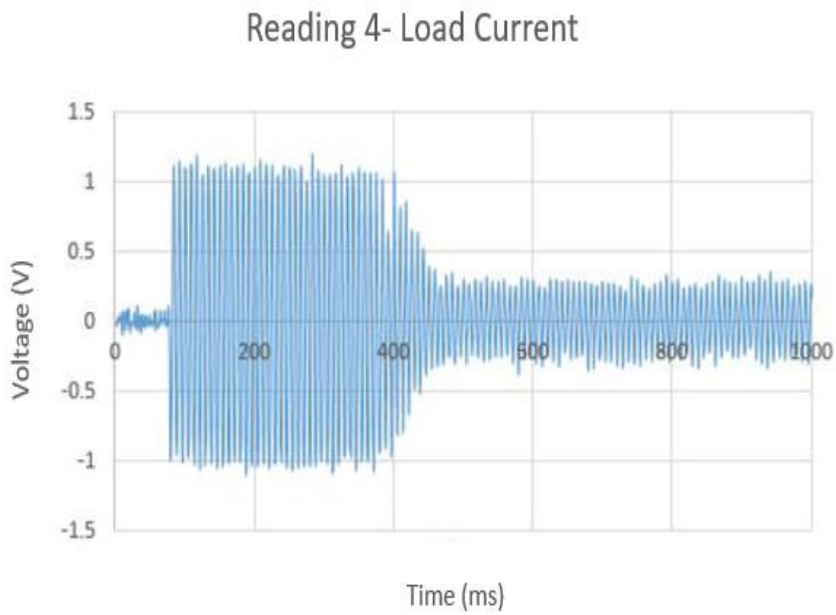

Fig. 4.19. Reading 4 Load Current - Off Grid Mode

4.3.3 Voltage

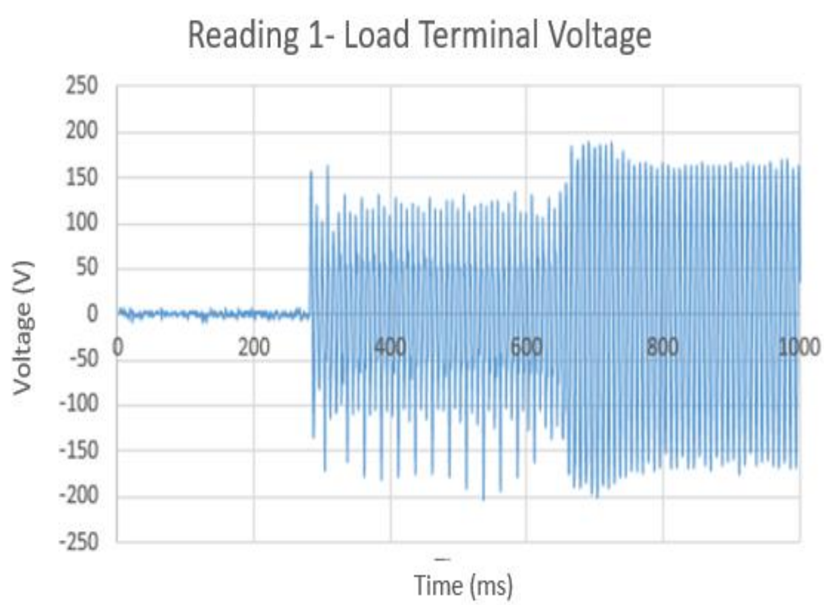

Fig. 4.20. Reading 1 Terminal Voltage- Off Grid Mode

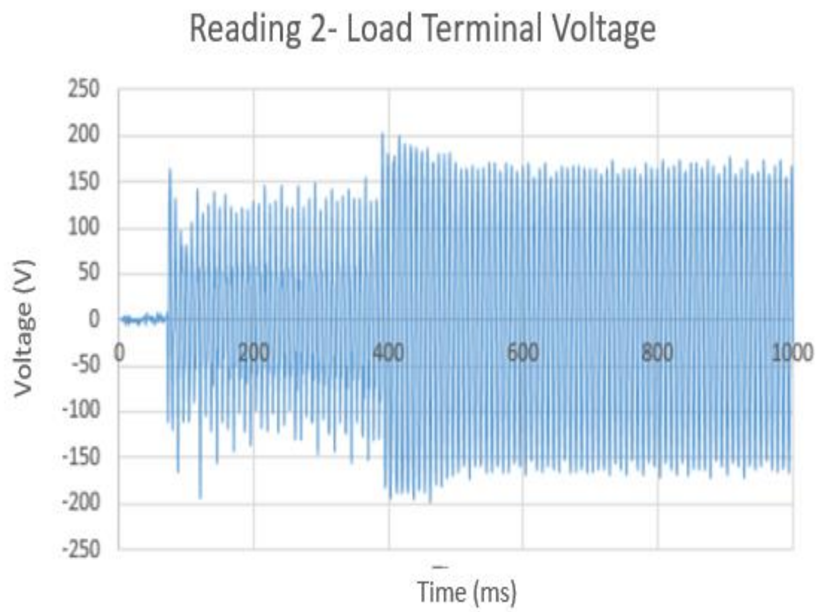

Fig. 4.21. Reading 2 Terminal Voltage - Off Grid Mode

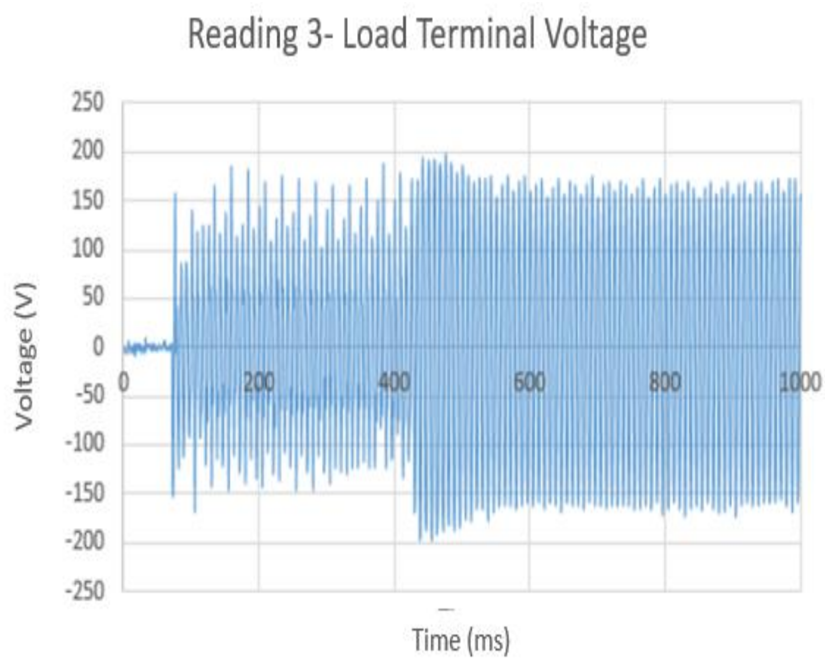

Fig. 4.22. Reading 3 Terminal Voltage - Off Grid Mode

`

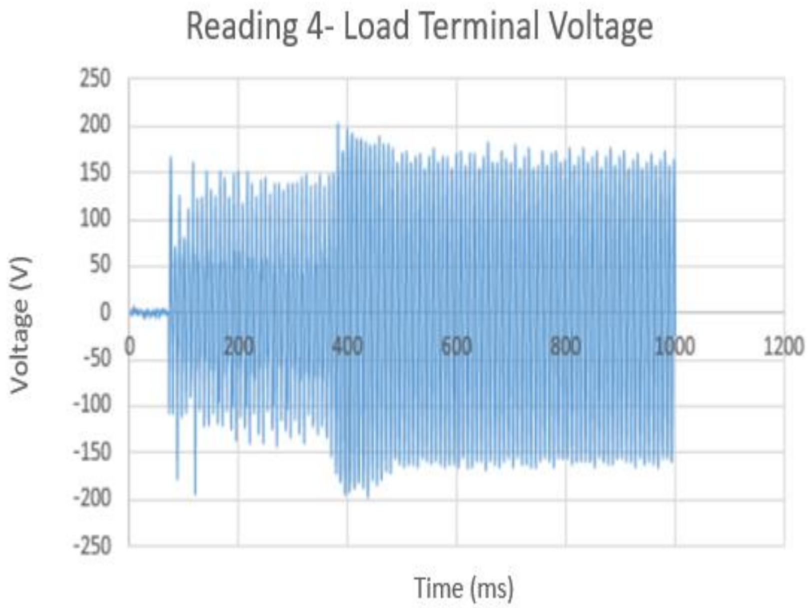

Fig. 4.23. Reading 4 Terminal Voltage - Off Grid Mode

4.4 Off-Grid mode with Varying State of Charge (SOC)

- The experiment is performed to test the significance of battery state of charge and the system performance to transients with an additional load.
- The test is performed in an off-grid mode with battery SOC at different step levels starting from 80 % to 20 %.
- The 3.5-ton residential air conditioner is connected to critical load panel. One current measurement (load terminal) and one voltage reading (load terminal voltage) are measured.
- All the other loads connected to the system are switched off initially and then the loads are added in steps along with the 3.5-ton residential air conditioner and the experiment is repeated.

- Calibration for current measurement 10 mV /A.
- The bandwidth of the current probes used are 1.5 kHz.

4.4.1 Explanation

- The voltage drop-in case of the off-grid mode is above the +/- 10 % limits with the lowest peak around 90 V.
- However, on switching on the motor with additional load the motor tripped the breaker with the overcurrent fault.
- The transient lasted for more than one second and is seen as an overcurrent fault by the breaker and it trips off the PV-battery system.
- Fig. 4.24-Fig. 4.30 shows the load current with varying battery SOC.
- Fig. 4.31-Fig. 4.37 shows the terminal load voltage with varying battery SOC.

Table 4.1. Variation of Transient Cycles with SOC

Battery SOC	Transient cycles
SOC-80 %	59 Cycles
SOC-70%	62 Cycles
SOC-60%	83 Cycles
SOC-50%	83 Cycles
SOC-40%	82 Cycles
SOC-30%	86 Cycles
SOC-20%	85 Cycles

`

4.4.2 Current Reading

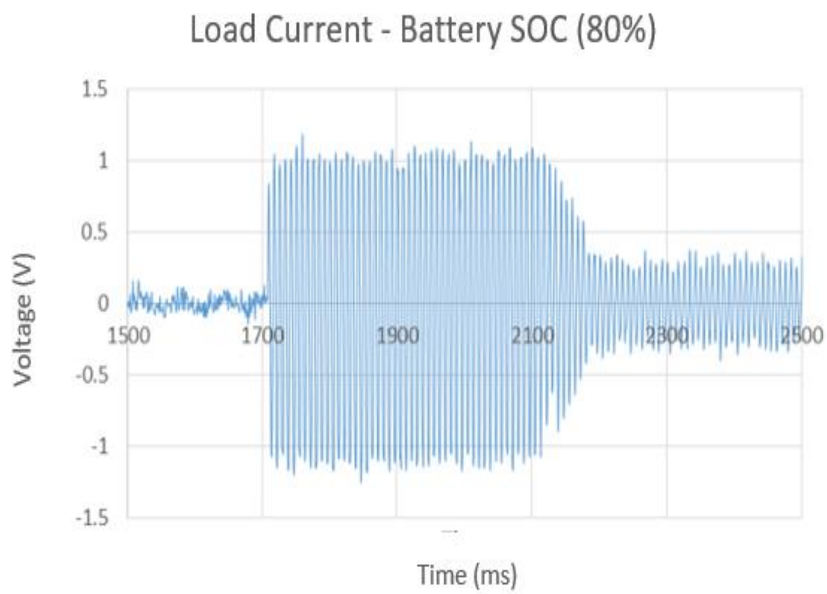

Fig. 4.24. Load Current - Off Grid Mode - SOC-80%

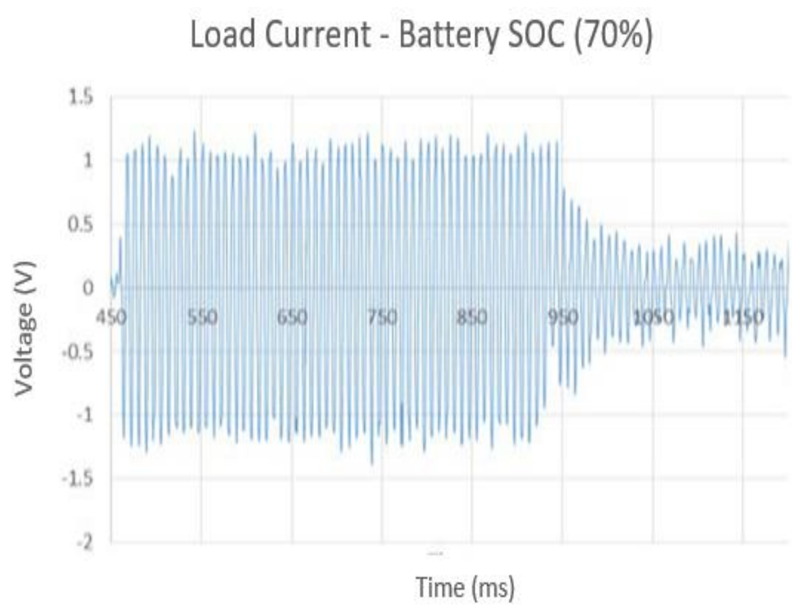

Fig. 4.25. Load Current - Off Grid Mode - SOC-70%

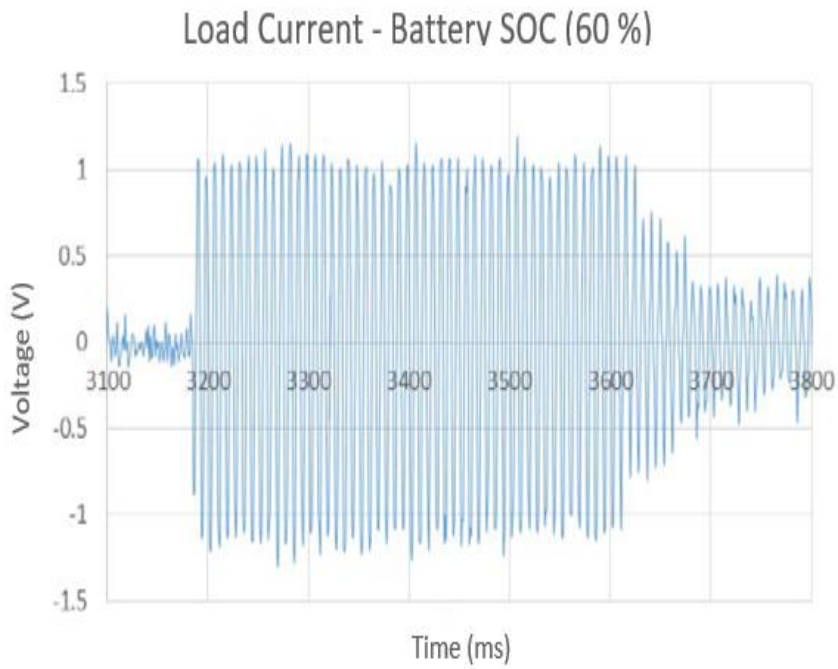

Fig. 4.26. Load Current - Off Grid Mode - SOC-60%

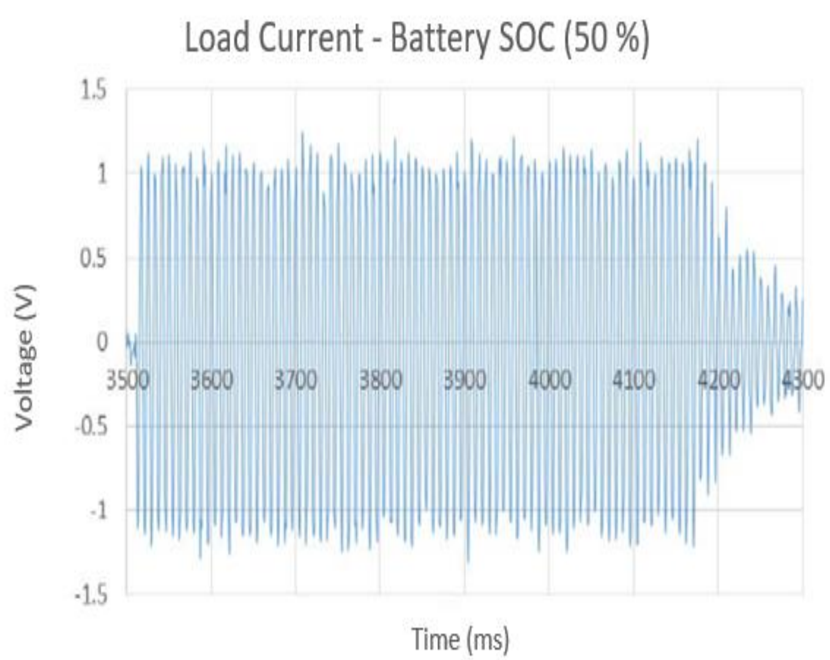

Fig. 4.27. Load Current - Off Grid Mode - SOC-50%

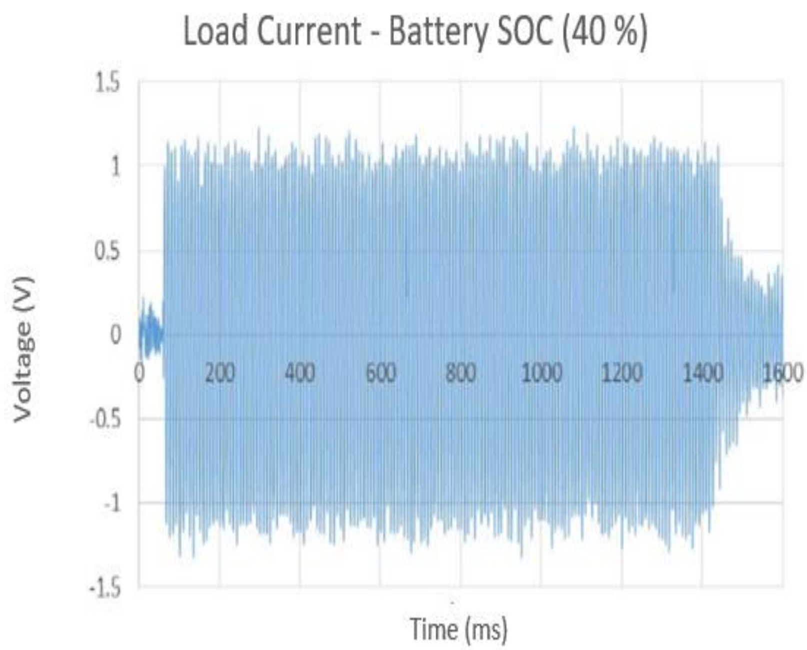

Fig. 4.28. Load Current - Off Grid Mode - SOC-40%

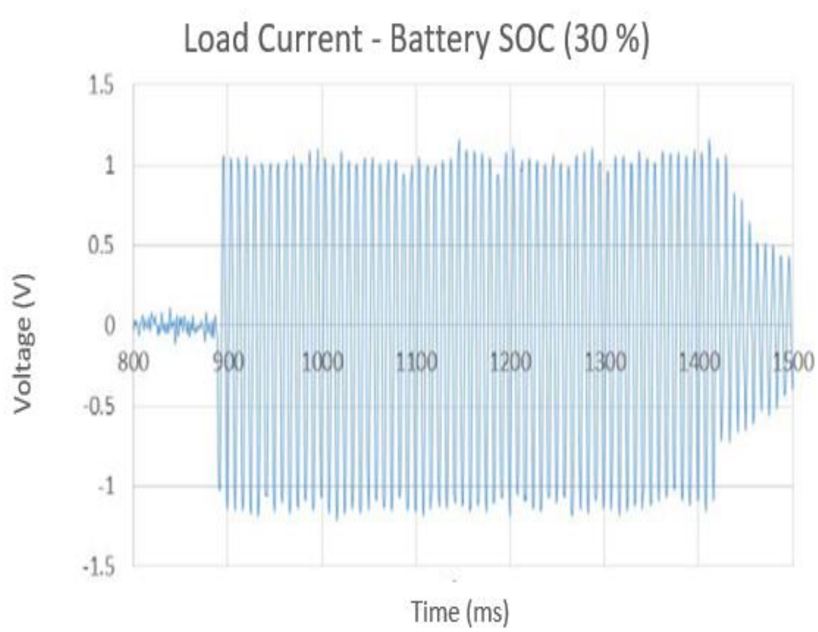

Fig. 4.29. Load Current - Off Grid Mode - SOC-30%

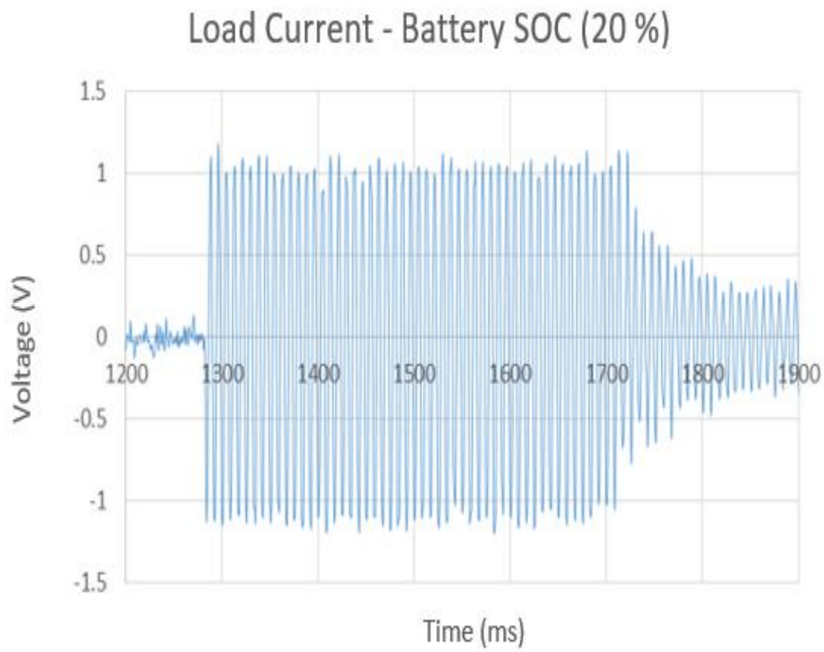

Fig. 4.30. Load Current - Off Grid Mode - SOC-20%

4.4.3 Voltage Reading

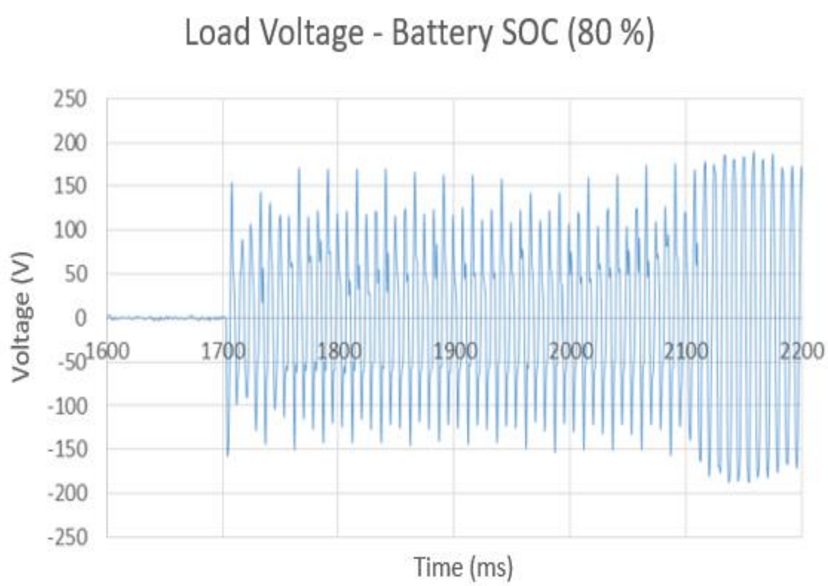

Fig. 4.31. Voltage Reading - Off Grid Mode - SOC-80%

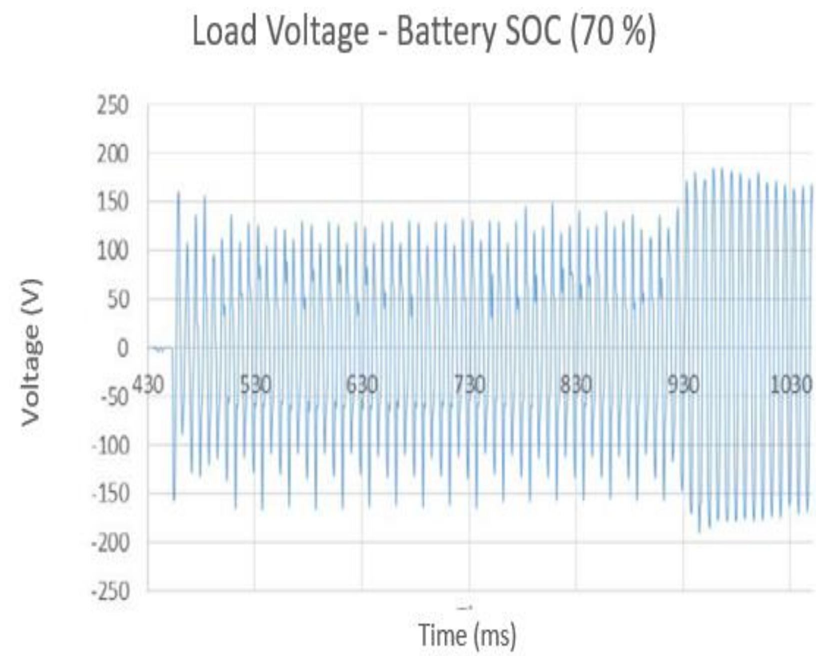

Fig. 4.32. Voltage Reading - Off Grid Mode - SOC-70%

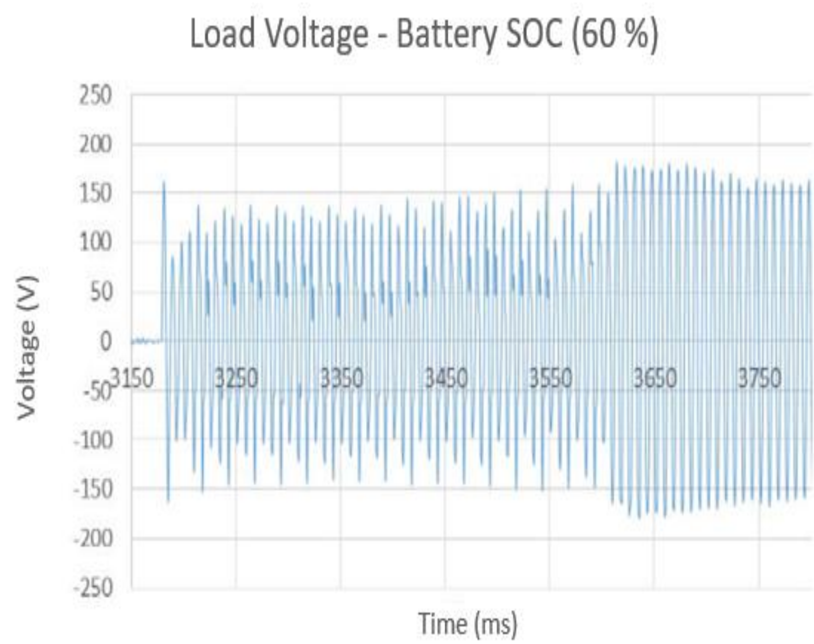

Fig. 4.33. Voltage Reading - Off Grid Mode - SOC-60%

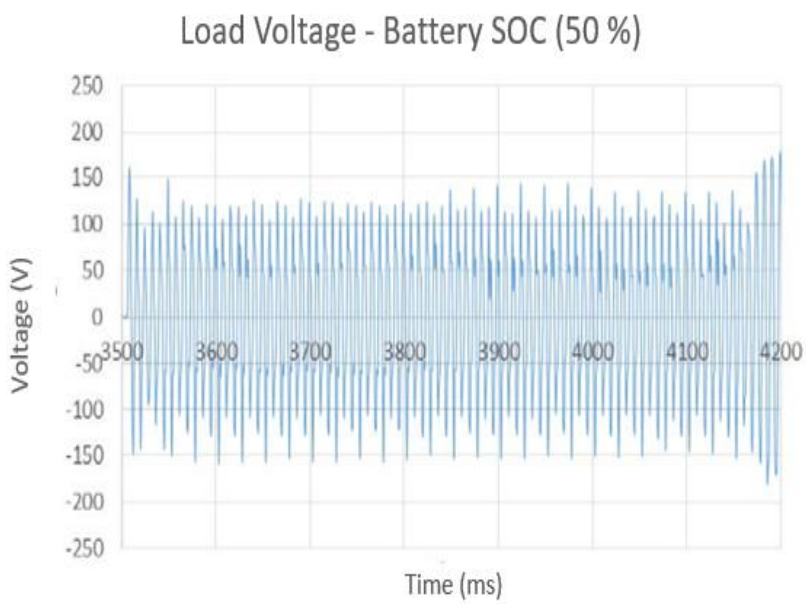

Fig. 4.34. Voltage Reading - Off Grid Mode - SOC-50%

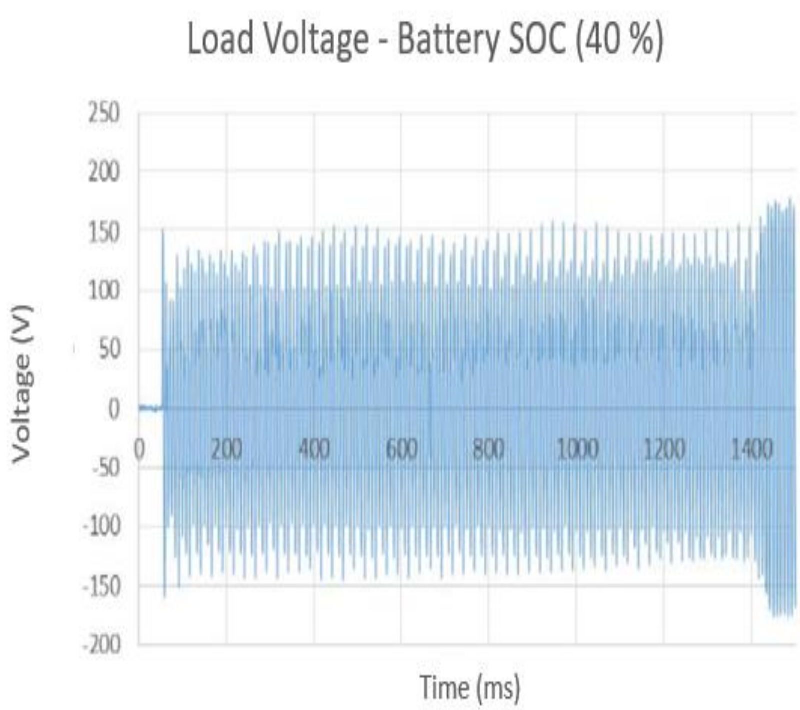

Fig. 4.35. Voltage Reading - Off Grid Mode - SOC-40%

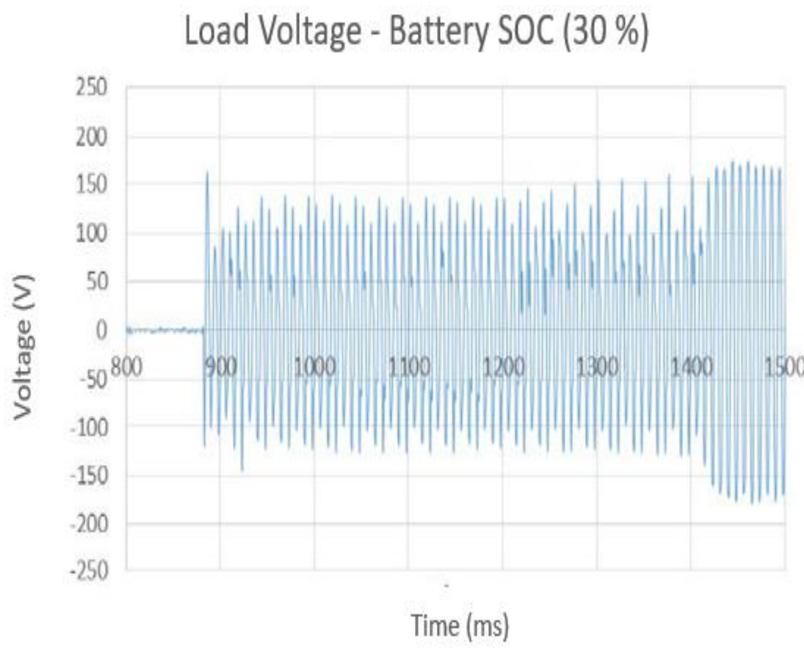

Fig. 4.36. Voltage Reading - Off Grid Mode - SOC-30%

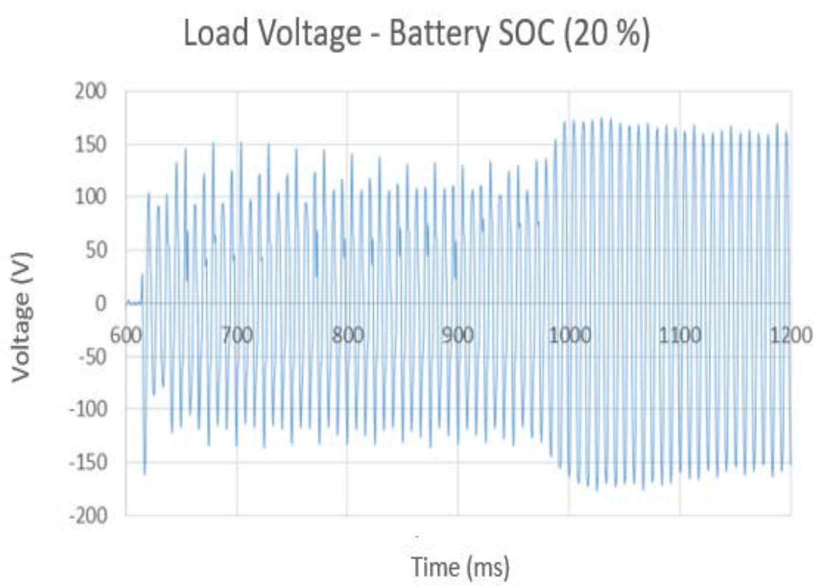

Fig. 4.37. Voltage Reading - Off Grid Mode - SOC-20%

`

CHAPTER 5 – LOAD SELECTION AND DEMAND CHARGE REDUCTION ASU ALGORITHM

5.1 Introduction

The load selection is performed using the historical load data set of one year. The selected load is simulated on the load banks that are present on the rooftop of the ERC. An algorithm has been developed to operate the PV-battery system in order to reduce the residential demand charge during the on-peak hours. A comparison has been made with the results obtained from the commercial algorithm and that of the ASU algorithm.

5.2 Load Selection

Each dotted point in Fig. 5.1 [*x*, *y*] with *x*-axis denoted by temperature and *y*-axis denoted by the load. The nearest loads of the set points are assigned weights with the closest setpoint having a larger weight than the farther one.

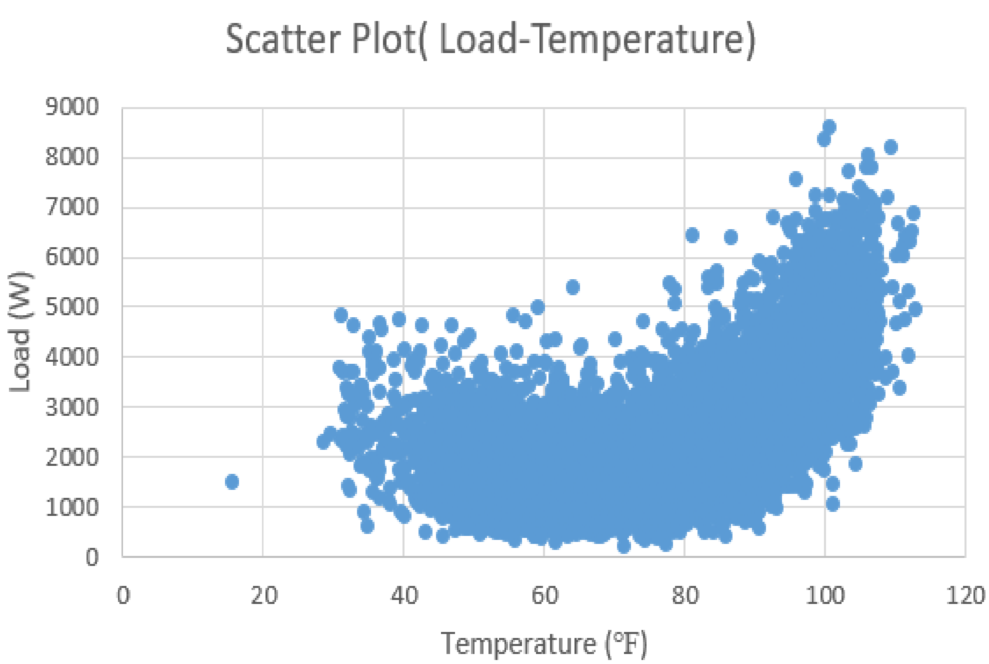

Fig. 5.1. Load vs Temperature

`

- The algorithm would use maximum temperature variable to select the load that is to be run on the load banks at ASU.
- Weighted k-mean clustering is used to find the next day load from the selected group of load sets.
- The selected load is based on the forecasted temperature for the next day.
- The dataset is divided into three clusters winter, summer and summer-peak.
- An hourly average load is obtained for each season.
- Based on the maximum forecasted temperature the subset of data points is selected from the season cluster that are within +/- 0.25 °F of the forecasted temperature.
- Deviation of each data point from the hourly average load is obtained.
- Weights are assigned to each data point, the load with minimum deviation has a maximum weight and the load with maximum deviation has the minimum weight.
- The final load is a weighted mixture of the selected subset.

5.2.1 Results for Load selection

The results are shown with the example. The algorithm is implemented using the MATLAB and the code is attached in the appendix. For simplification only five data points (five hours) in a 24-hour load are shown.

Step 1: Select the maximum forecasted temperature for next day

Temperature = 72 °F.

Step 2: Select Season Winter, summer, Summer-peak

`

Season = winter.

Step 3: Average hourly load of the season.

Table 5.1. Average Load

Time	Average load (watts)
9:00 a.m.	1891.4
10.00 a.m.	1776.1
11.00 a.m.	1755.6
12:00 p.m.	1721.4
1:00 p.m.	1781.2

Step 5: Subsets of loads selected with a tolerance of +/- 0.25 °F

Table 5.2. Load Subset Within Tolerance

Time	Load-1 (watts)	Load-2 (watts)
9:00 a.m.	1078.1	1145.0
10.00 a.m.	1451.9	2457.3
11.00 a.m.	2225.1	2192.7
12:00 p.m.	1150.3	1663.8
1:00 p.m.	2853.2	1431.2

`

Step 6: Deviation of the loads

$$Deviation = Average\ \ Load - (load\,1, load\,2)\,. \tag{7.1}$$

Table 5.3. Deviation of Load Subset

Time	Deviation load-1 (watts)	Deviation load-2 (watts)
9:00 a.m.	813.2	746.3
10.00 a.m.	324.1	681.2
11.00 a.m.	469.4	437.0
12:00 p.m.	571.1	57.6
1:00 p.m.	1072.0	350.0

Step 7: Calculate weights

$$weight_{first\,load} = 1 - \frac{\text{devation of first load}}{\text{deviation of first load} + \text{deviation of second load}} \tag{7.2}$$

$$weight_{\text{second } load} = 1 - \frac{\text{devation of second load}}{\text{deviation of first load} + \text{deviation of second load}}\,. \tag{7.3}$$

`

Table 5.4. Weights of Load Subset

Time	Weights load-1 (ratio)	Weights load-2 (ratio)
9:00 a.m.	0.4786	0.5214
10.00 a.m.	0.6776	0.3224
11.00 a.m.	0.4821	0.5179
12:00 p.m.	0.0916	0.9084
1:00 p.m.	0.2461	0.7539

Step 8: Final load

$$Final\ Load = weight_{first\,load} Load\,1 + weight_{\mathrm{second}\ load} Load\ 2\ . \qquad (7.4)$$

Table 5.5. Final Load

Time	Final load (watts)
9:00 a.m.	1113.0
10.00 a.m.	1776.1
11.00 a.m.	2208.3
12:00 p.m.	1616.7
1:00 p.m.	1781.2

`

5.3 Demand Charge Reduction -Commercial Algorithm

- The objective is to minimize consumption from grid during on-peak hours.
- In regions with net energy metering, tries to maximize export during peak hours.
- Charge from grid during non-peak hours.
- Minimizing demand during demand assessment period.
- Sets the previous calendar months demand as the threshold limit.
- The system has an interface to communicate and set the limits and assessment period.
- Fig. 5.3 indicates the operating conditions.
- On-peak hours are from 1 p.m. to 8 p.m.
- Demand assessment period is one month.
- Demand charge is measured every 15 minutes.
- The battery is charged from grid during off-peak hours (10:00 p.m. – 3:00 a.m.).
- All the figures from Fig. 5.4 – Fig. 5.23 are obtained by implementing the commercial algorithm on the system.
- All the figures from Fig. 5.4 – Fig. 5.23 have *x* axis on a 15-min interval scale.
- Each hour has four 15- min data points. To view the *x*-axis scale on an hourly basis simply divide the *x* axis value by 4, for example data point 40 on x- axis indicate 10 a.m. on an hourly basis.
- The on-peak hours are from 1 p.m. to 8 p.m., on a 15-min interval scale they are indicated between data points 52 and 80.

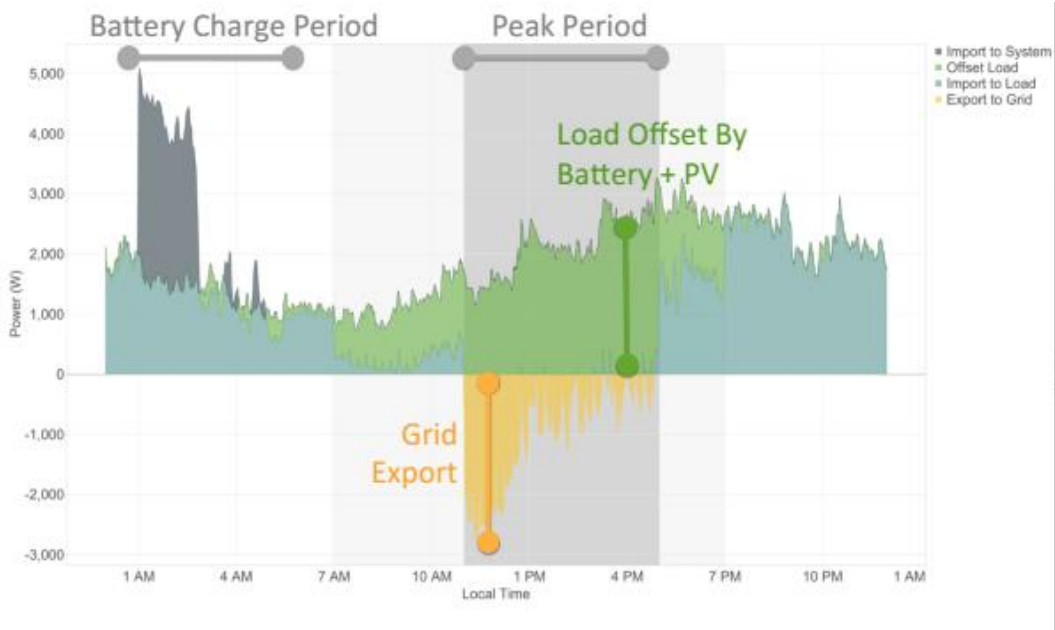

Fig. 5.2. Commercial Algorithm Pictorial Representation

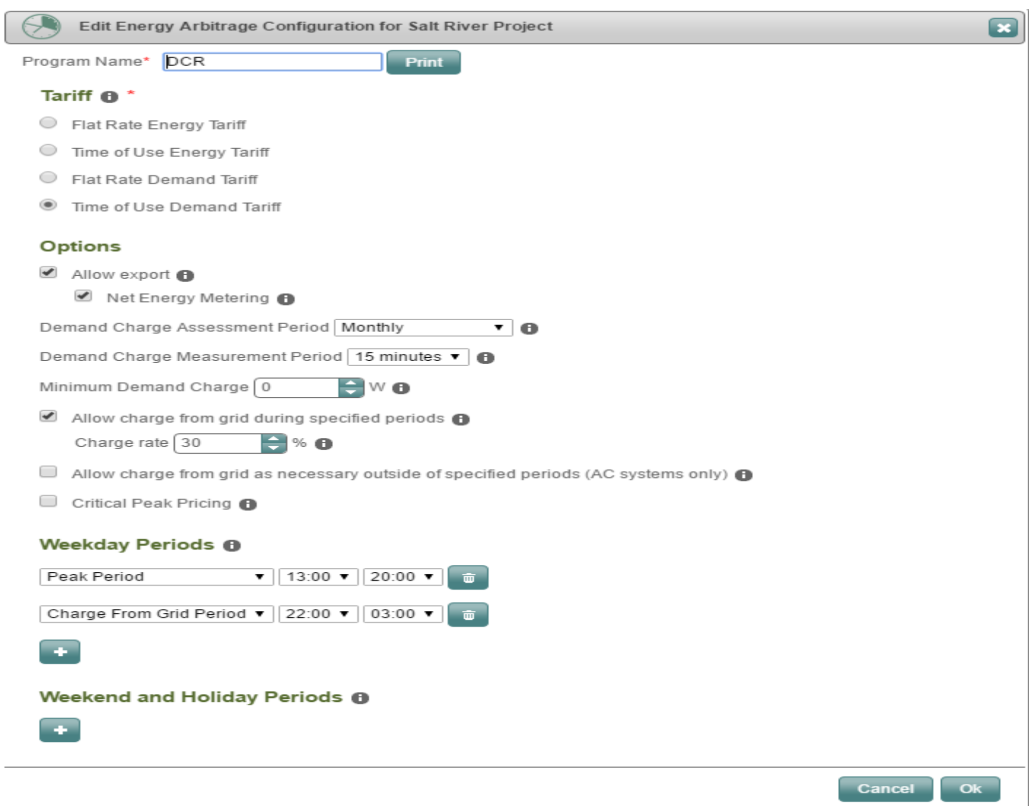

Fig. 5.3. Commercial Algorithm Logic Implementation

`

5.3.1 Results for Commercial Algorithm

a. PV Data

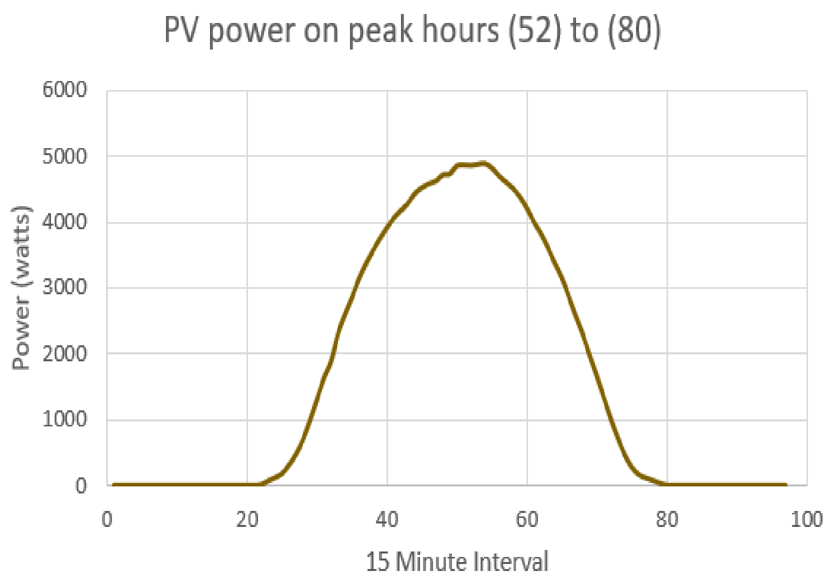

Fig. 5.4. PV-power-day 1 (Commercial Algorithm)

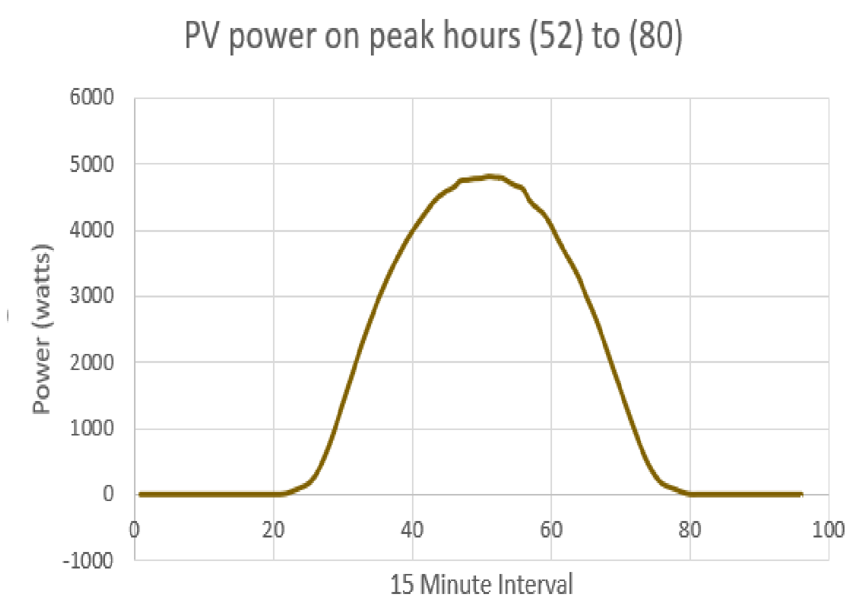

Fig. 5.5. PV-power-day 2 (Commercial Algorithm)

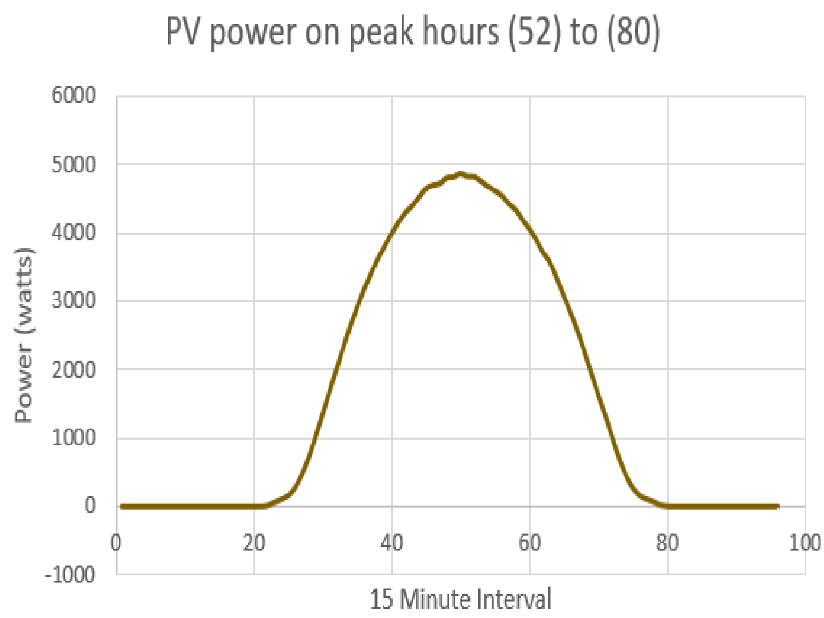

Fig. 5.6. PV-power-day 3 (Commercial Algorithm)

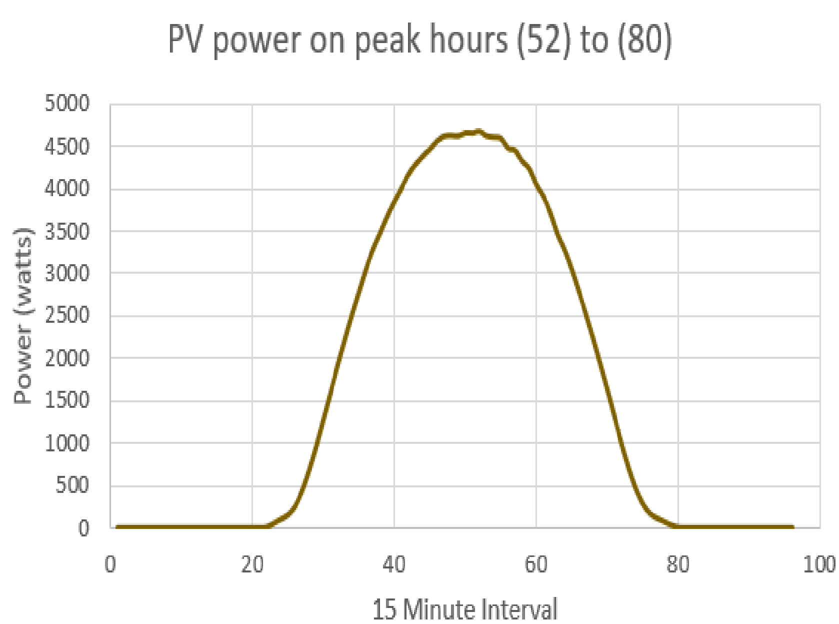

Fig. 5.7. PV-power-day 4 (Commercial Algorithm)

b. Battery Data

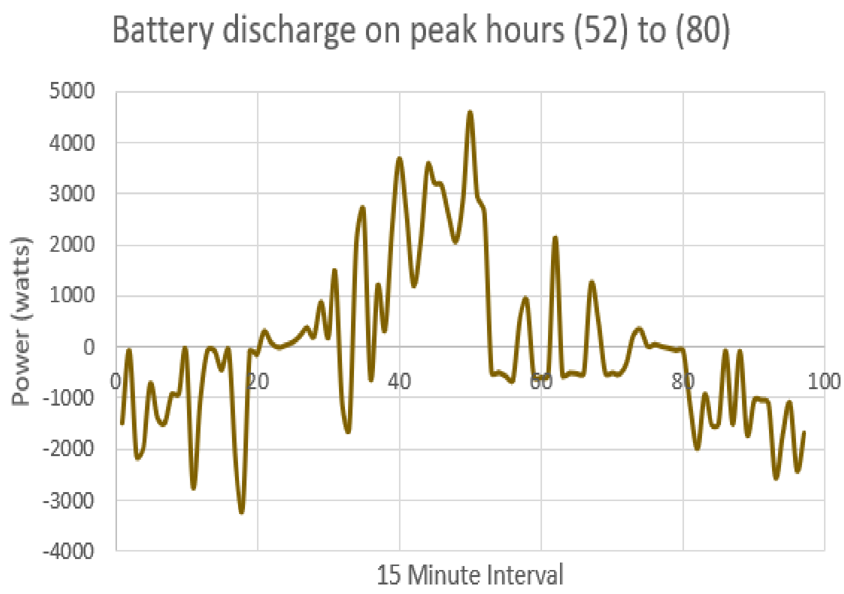

Fig. 5.8. Battery Discharge-day 1 (Commercial Algorithm)

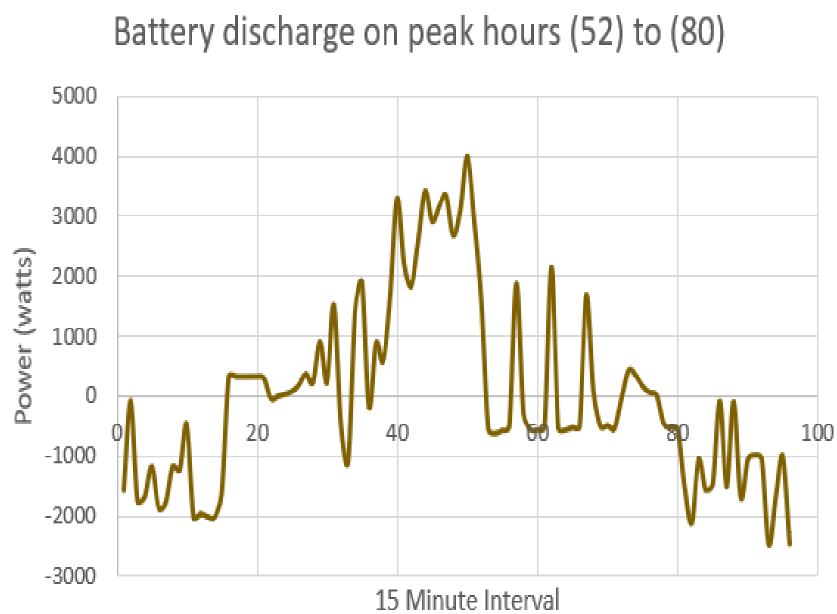

Fig. 5.9. Battery Discharge-day 2 (Commercial Algorithm)

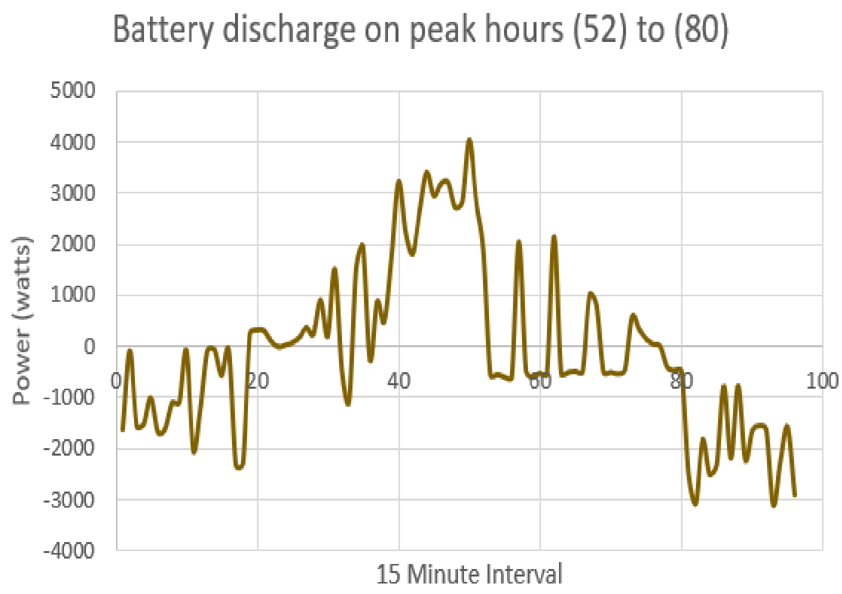

Fig. 5.10. Battery Discharge-day 3 (Commercial Algorithm)

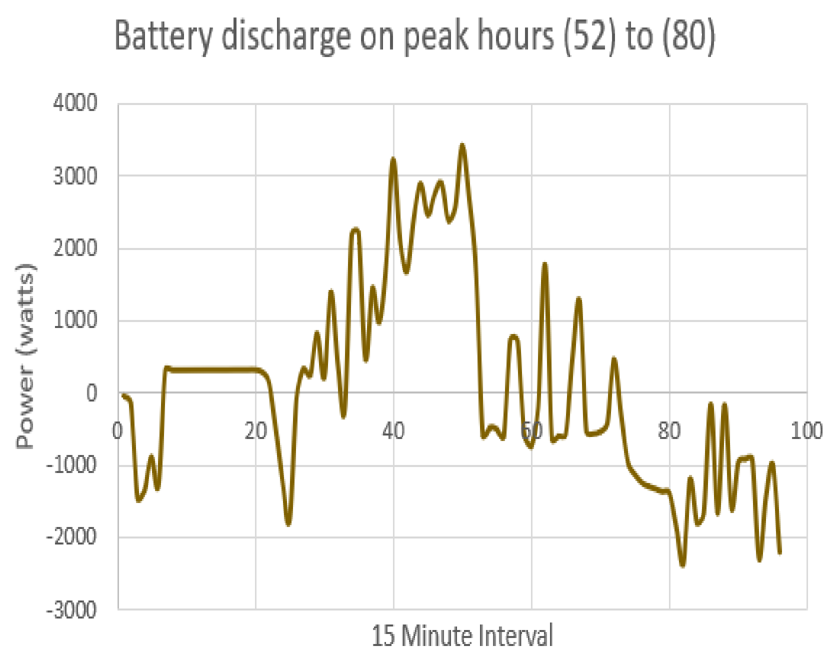

Fig. 5.11. Battery Discharge-day 4 (Commercial Algorithm)

c. Load

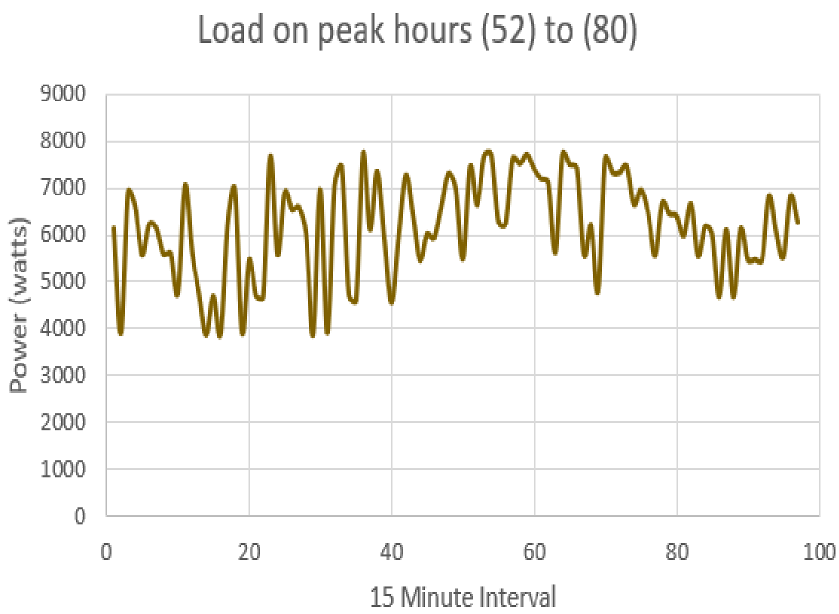

Fig. 5.12. Load-day 1 (Commercial Algorithm)

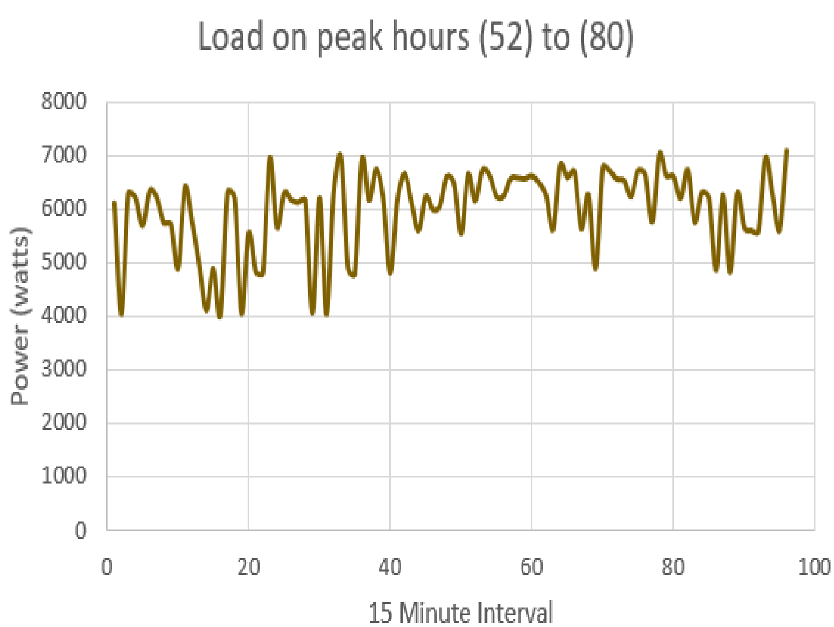

Fig. 5.13. Load-day 2 (Commercial Algorithm)

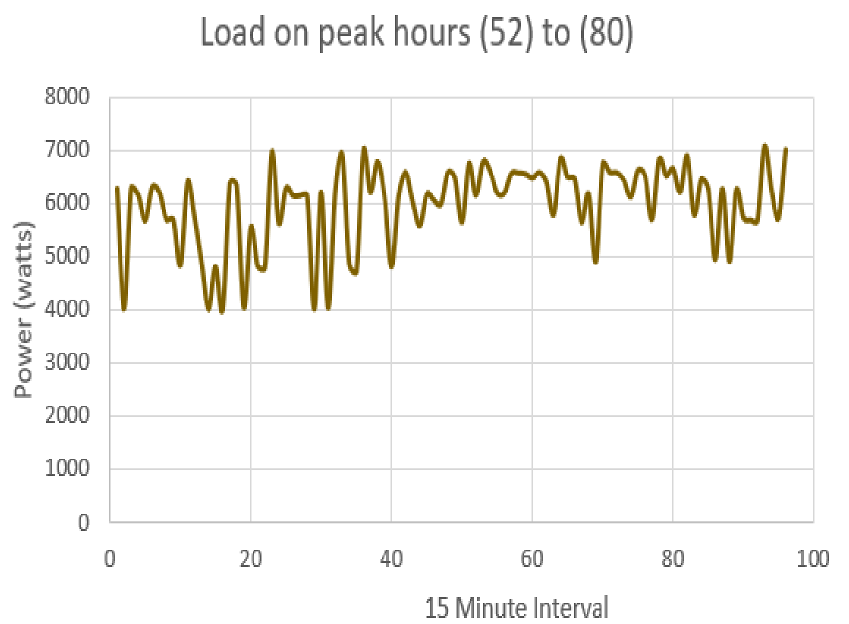

Fig. 5.14. Load-day 3 (Commercial Algorithm)

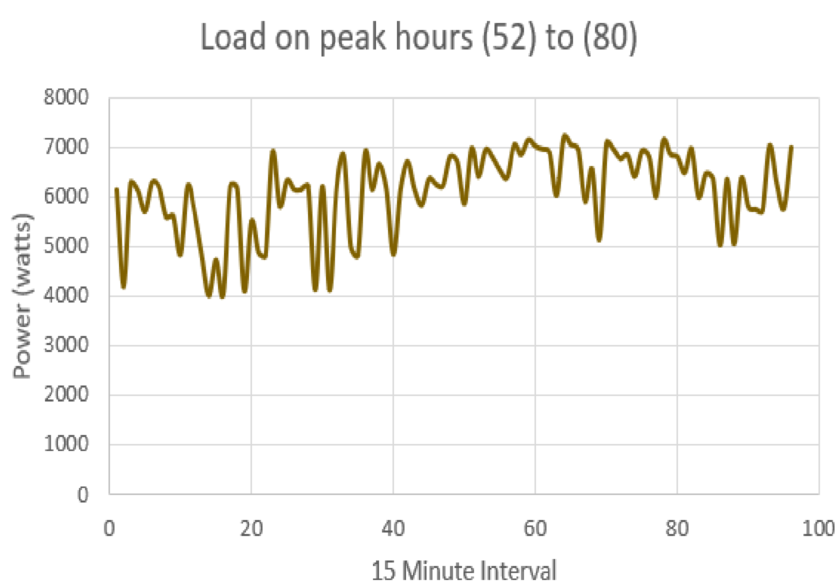

Fig. 5.15. Load-day 4 (Commercial Algorithm)

d. Site Demand

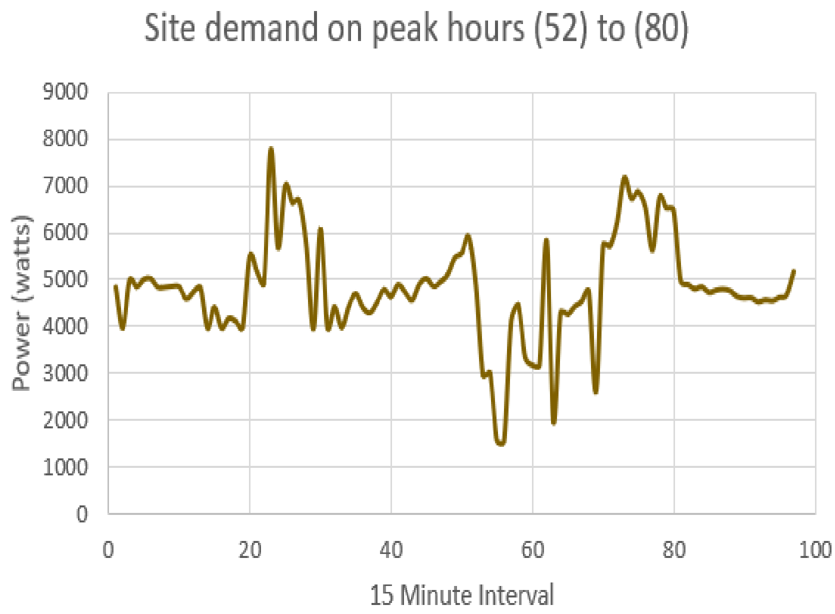

Fig. 5.16. Site Demand-day 1 (Commercial Algorithm)

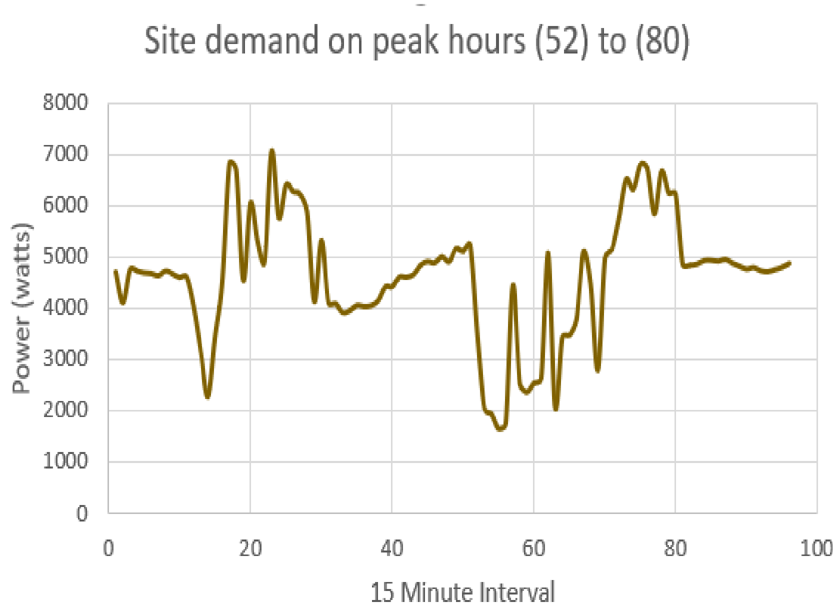

Fig. 5.17. Site Demand-day 2 (Commercial Algorithm)

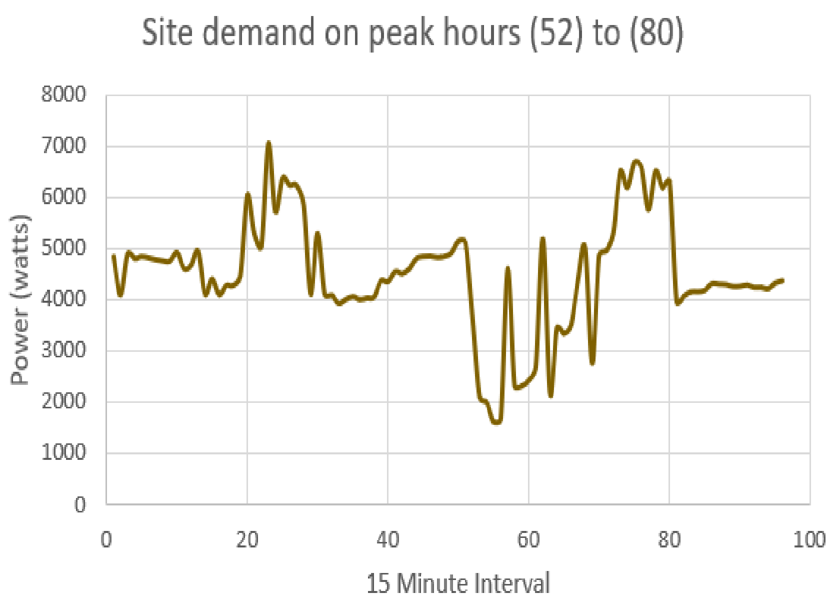

Fig. 5.18. Site Demand-day 3 (Commercial Algorithm)

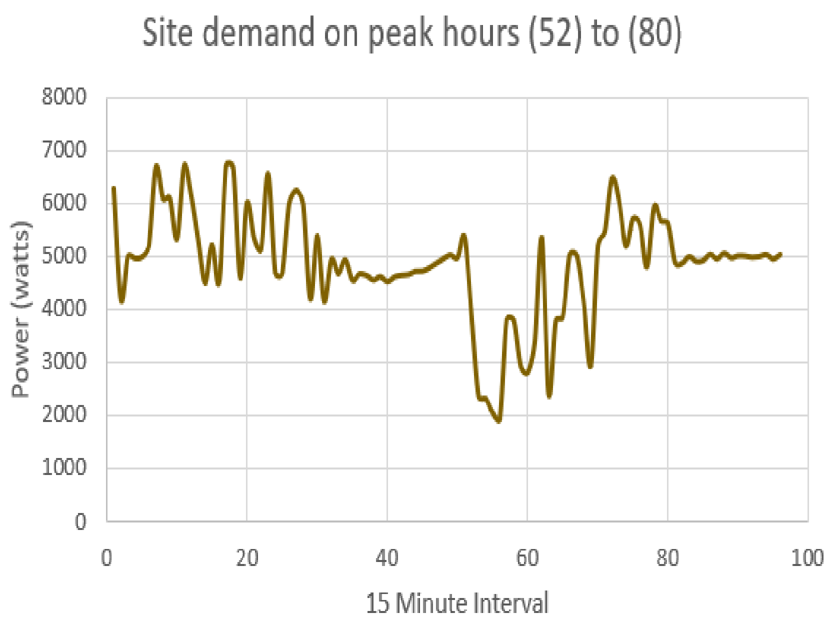

Fig. 5.19. Site Demand-day 4 (Commercial Algorithm)

e. Battery SOC

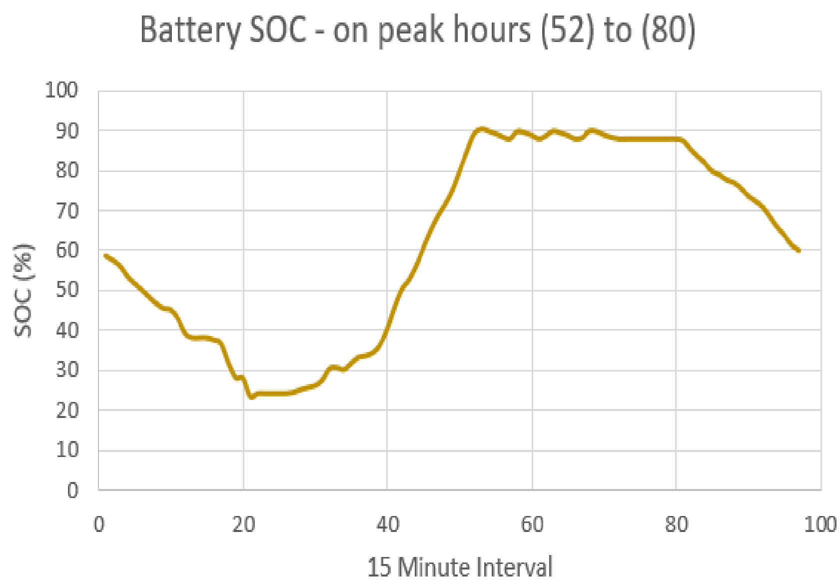

Fig. 5.20. Battery SOC-day 1 (Commercial Algorithm)

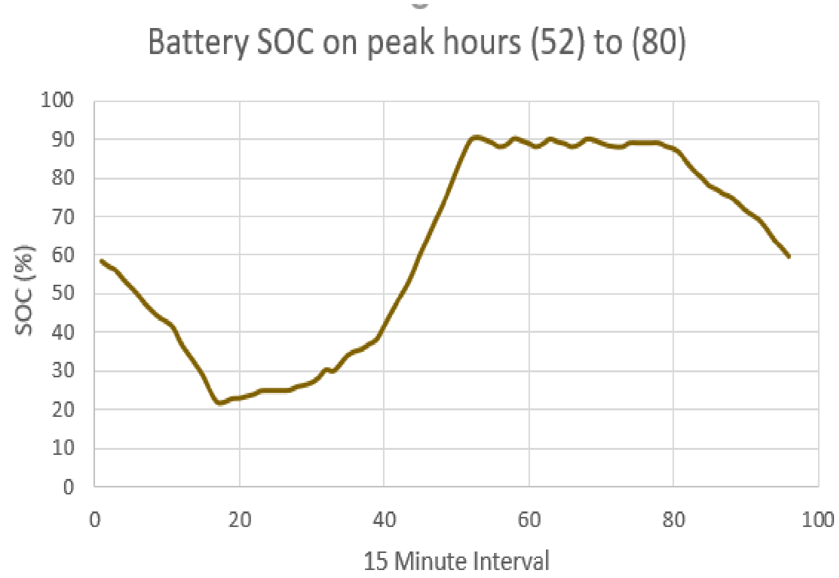

Fig. 5.21. Battery SOC-day 2 (Commercial Algorithm)

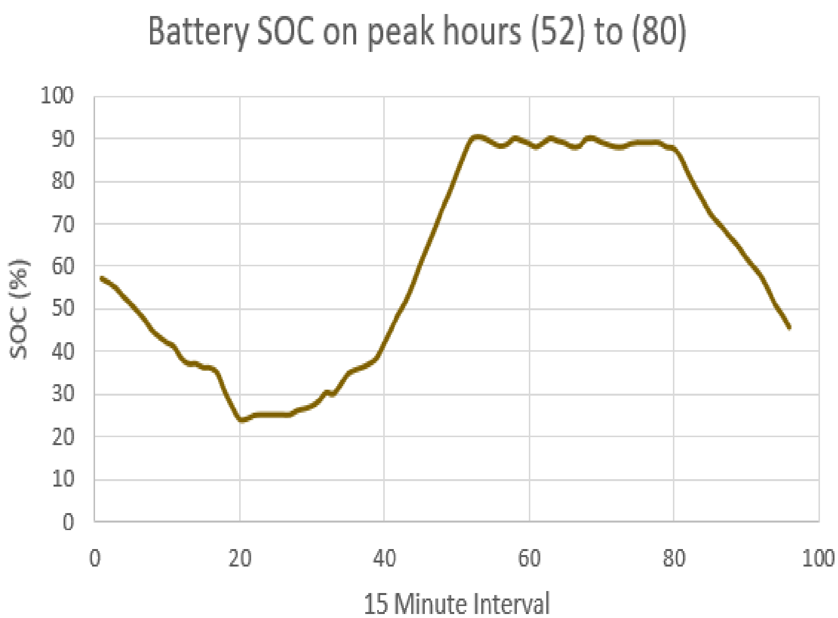

Fig. 5.22. Battery SOC-day 3 (Commercial Algorithm)

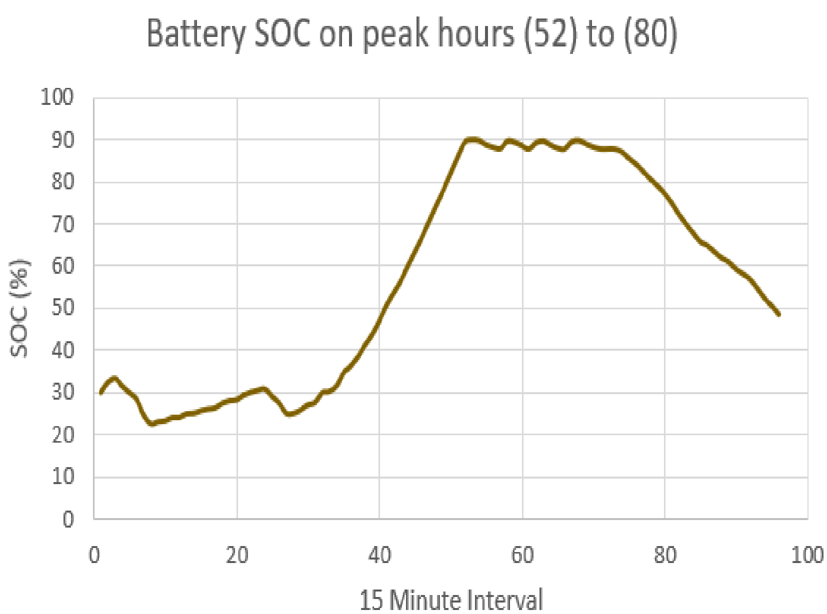

Fig. 5.23. Battery SOC-day 4 (Commercial Algorithm)

Table 5.6. Results from Commercial Algorithm

Days	PV energy	Load energy	Peak demand
Day 1	39.268 kWh	149.753 kWh	7010 W
Day 2	39.688 kWh	143.821 kWh	6980 W
Day 3	39.267 kWh	145.330 kWh	6100 W
Day 4	38.513 kWh	146.594 kWh	6500W

5.4 Demand Charge Reduction- ASU Algorithm

- This algorithm is based on providing a limit for the on-peak-grid-energy.
- The amount of energy that the battery needs to be charged overnight is determined based upon the minimum PV-energy available during the off-peak-hours.
- If the sum of load-difference during the on-peak-hours is less than the battery capacity, then the on-peak-limit is zero.
- If the sum of load-difference during the on-peak-hours is greater than the battery capacity, then the amount the energy by which the battery capacity is exceeded is divided by the number of on-peak-hours to determine the on-peak-limit. Each day would yield a unique on-peak-limit.
- Fig. 5.23 indicates the pictorial representation of ASU algorithm.

- The maximum on-peak-limit during the month is considered as a threshold limit for that month.
- The algorithm is developed in MATLAB and is attached in the appendix.
- All the figures from Fig. 5.25–Fig. 5.42 are obtained by implementing the commercial algorithm on the system.
- All the figures from Fig. 5.25–Fig. 5.42 have *x* axis on a 15-min interval scale.
- Each hour has four 15- min data points. To view the *x*-axis scale on an hourly basis simply divide the *x* axis value by 4, for example data point 40 on x- axis indicate 10 a.m. on an hourly basis.
- The on-peak hours are from 1 p.m. to 8 p.m., on a 15-min interval scale they are indicated between data points 52 and 80.

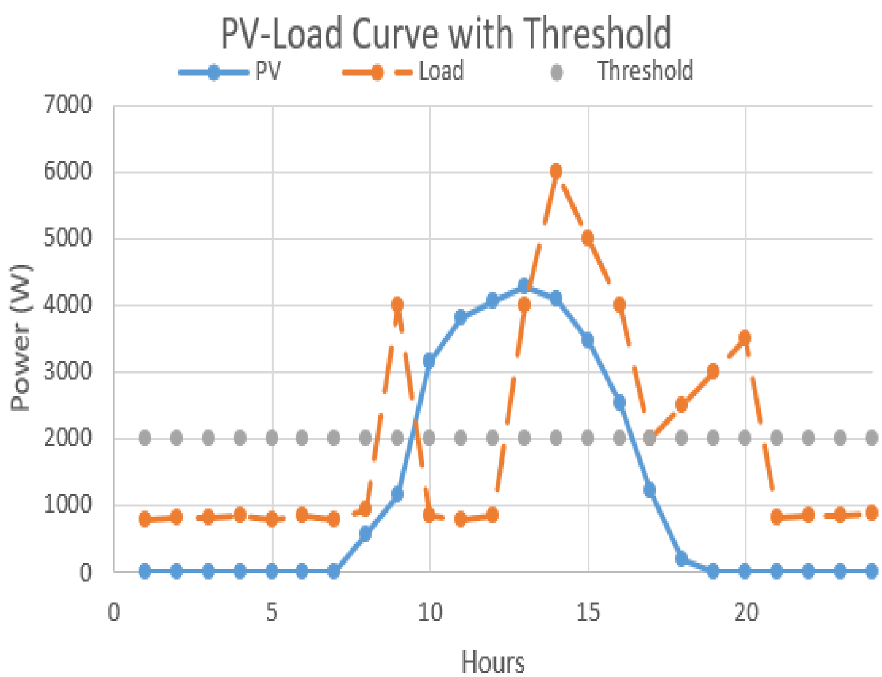

Fig. 5.24. Pictorial Representation of ASU Algorithm

5.4.1 Results for ASU algorithm

Results from the algorithm

- Minimum amount of energy battery needs to be charged on-peak is 15418 watts.
- Threshold limit for the system is 3500 watts.

a. PV data

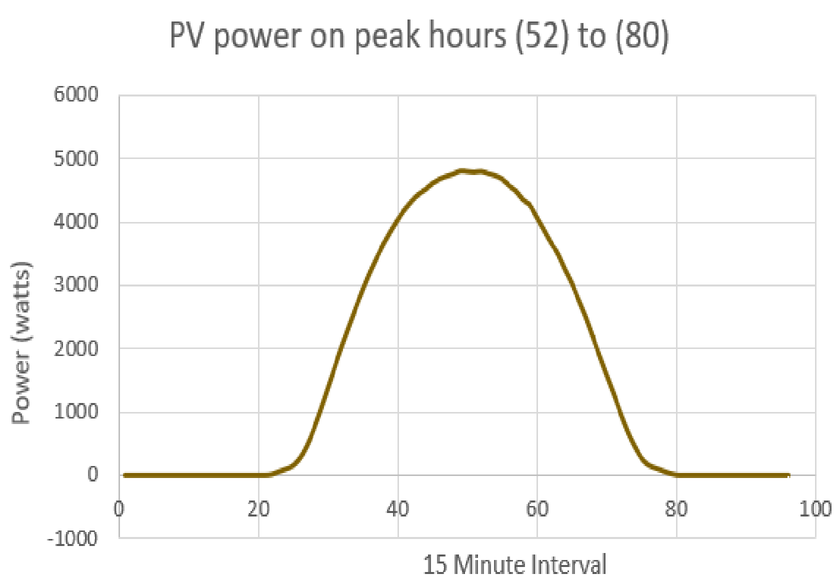

Fig. 5.25. PV-power-day 1 (ASU Algorithm)

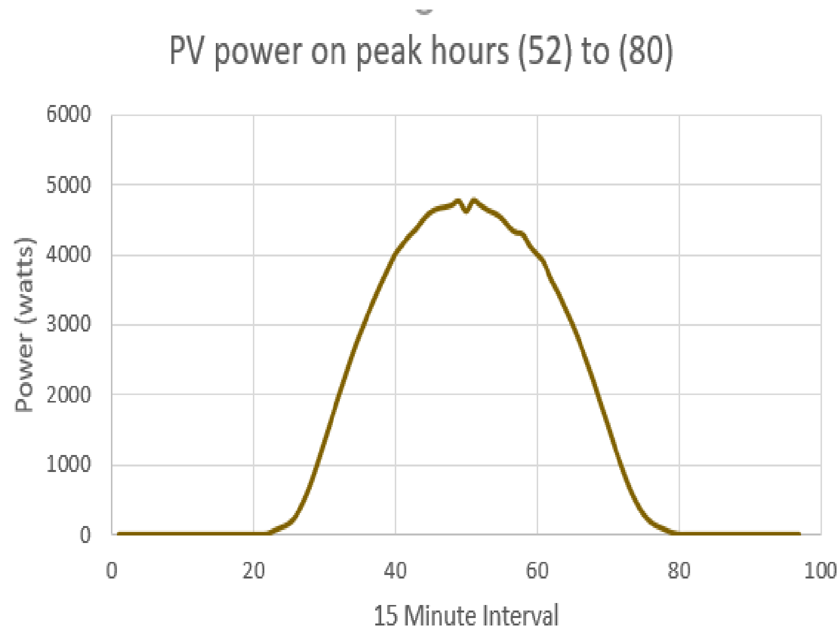

Fig. 5.26. PV-power-day 2 (ASU Algorithm)

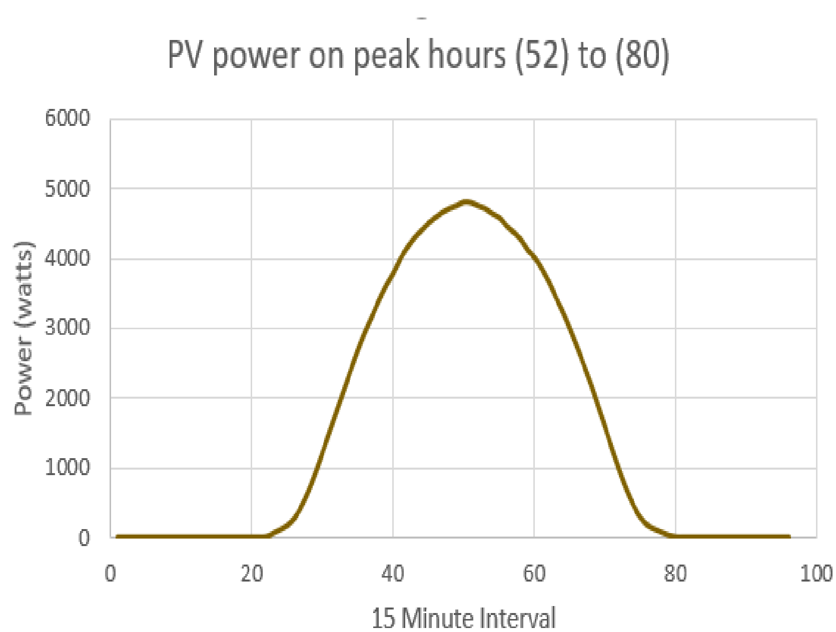

Fig. 5.27. PV-power-day 3 (ASU Algorithm)

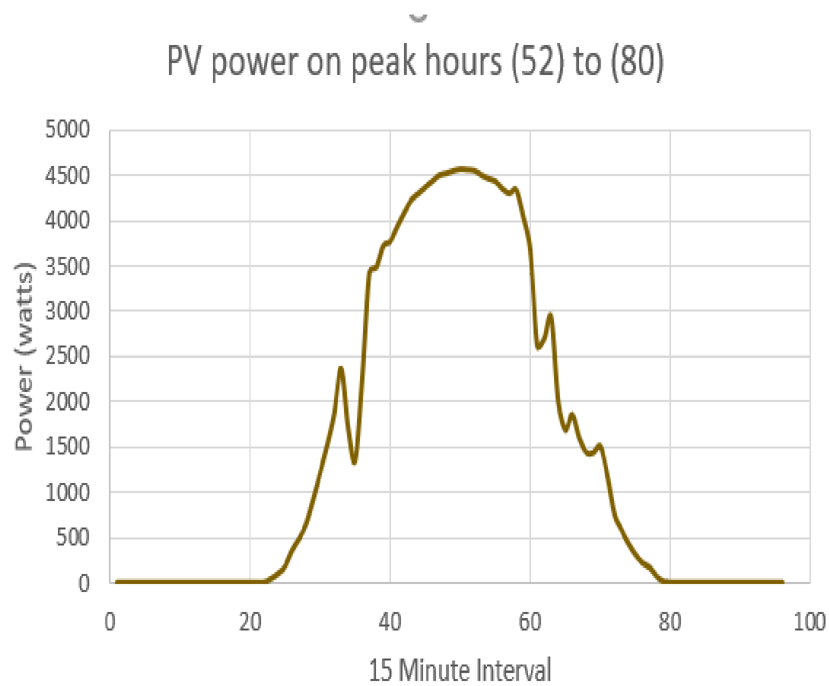

Fig. 5.28. PV-power-day 3 (ASU Algorithm)

b. Load data

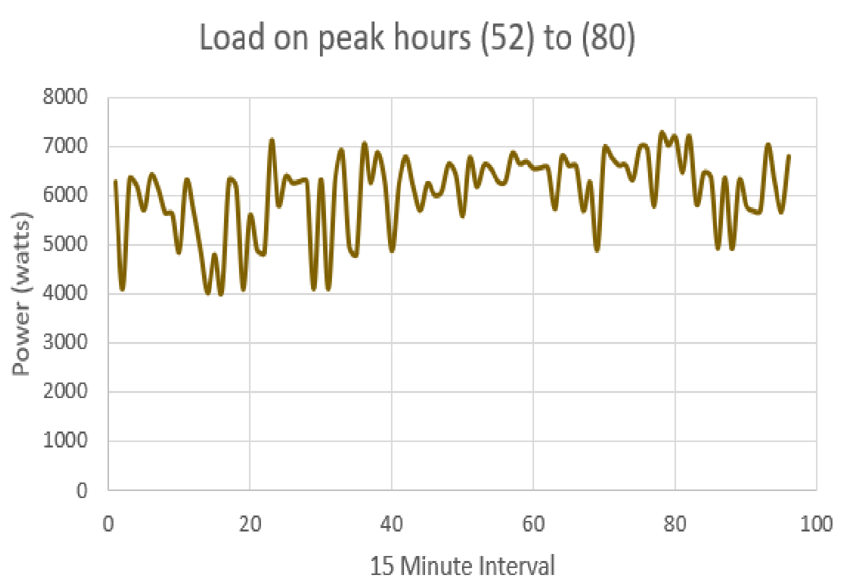

Fig. 5.29. Load-day 1 (ASU Algorithm)

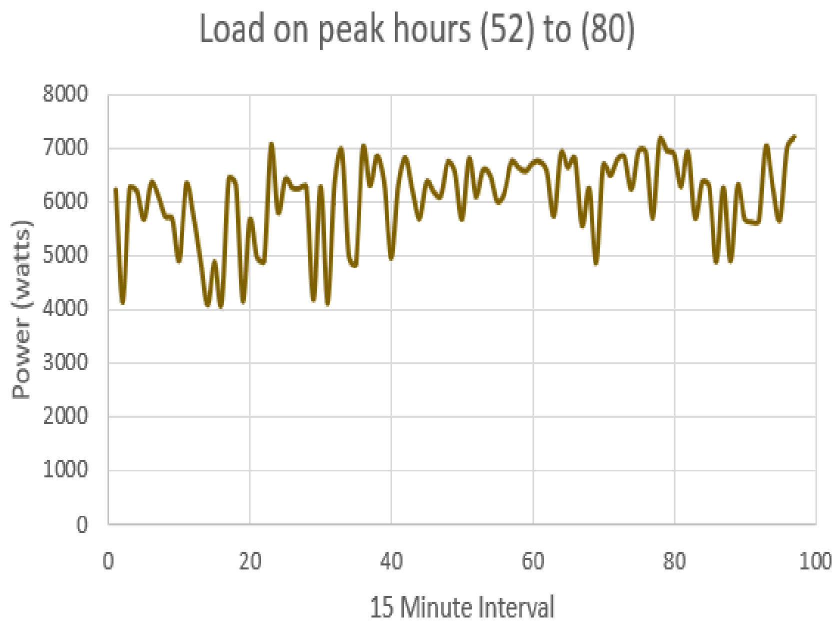

Fig. 5.30. Load-day 2 (ASU Algorithm)

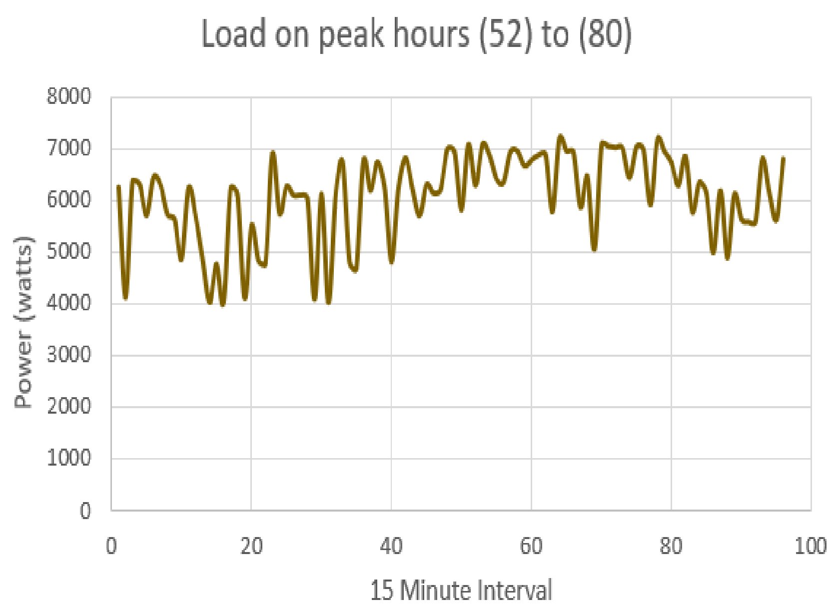

Fig. 5.31. Load-day 3 (ASU Algorithm)

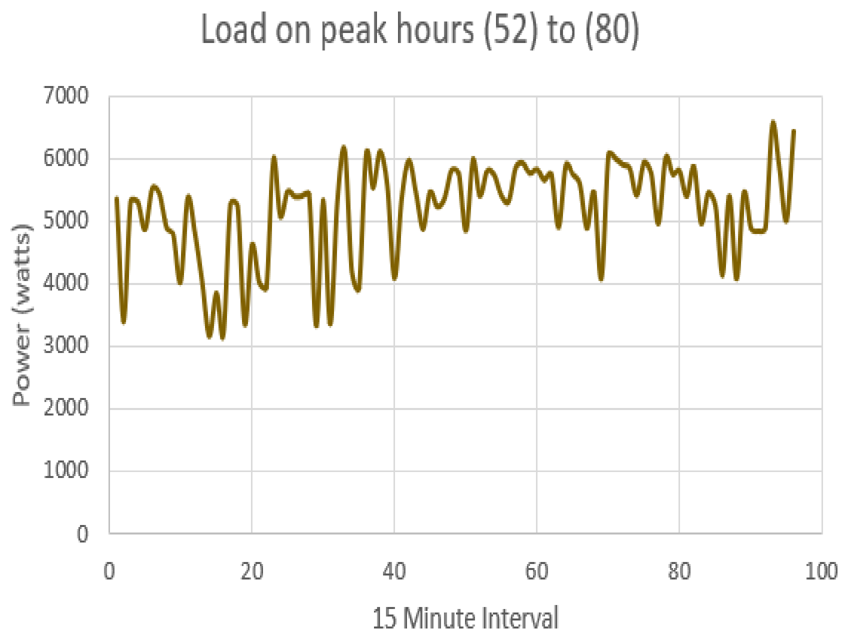

Fig. 5.32. Load-day 4 (ASU Algorithm)

c. Battery Discharge

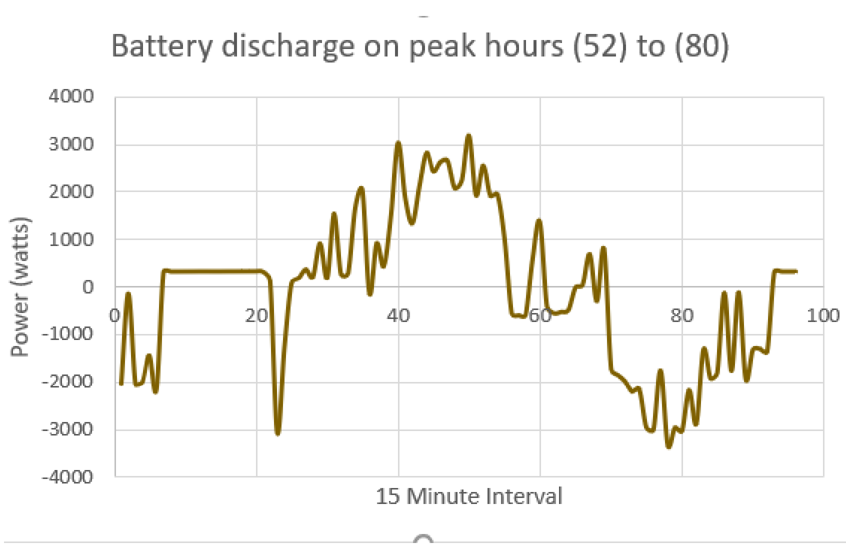

Fig. 5.33. Battery Discharge-day 1 (ASU Algorithm)

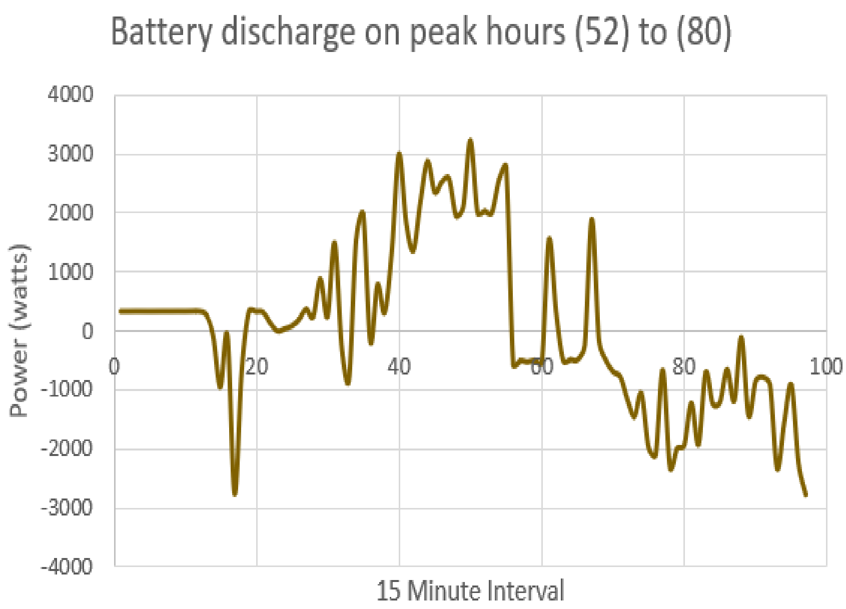

Fig. 5.34. Battery Discharge-day 2 (ASU Algorithm)

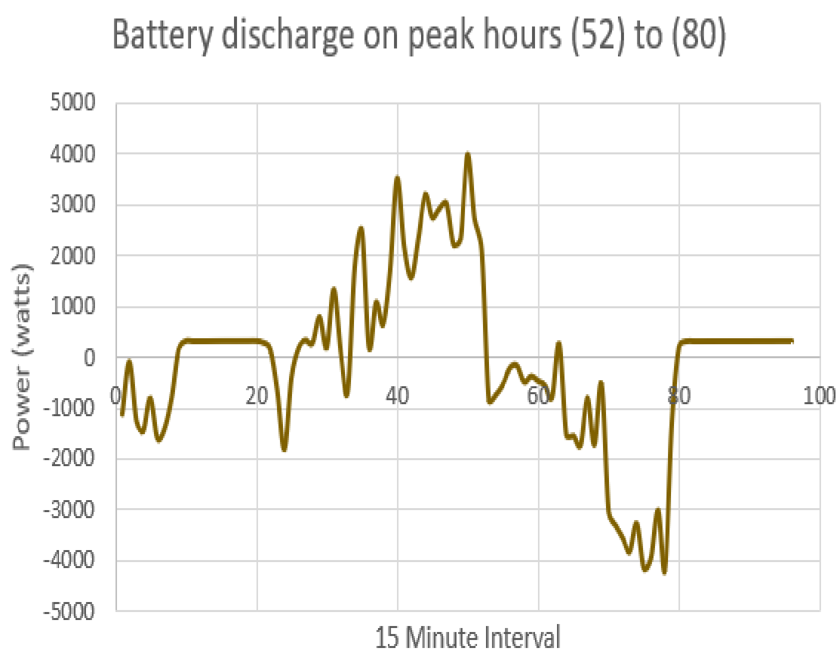

Fig. 5.35. Battery Discharge-day 3 (ASU Algorithm)

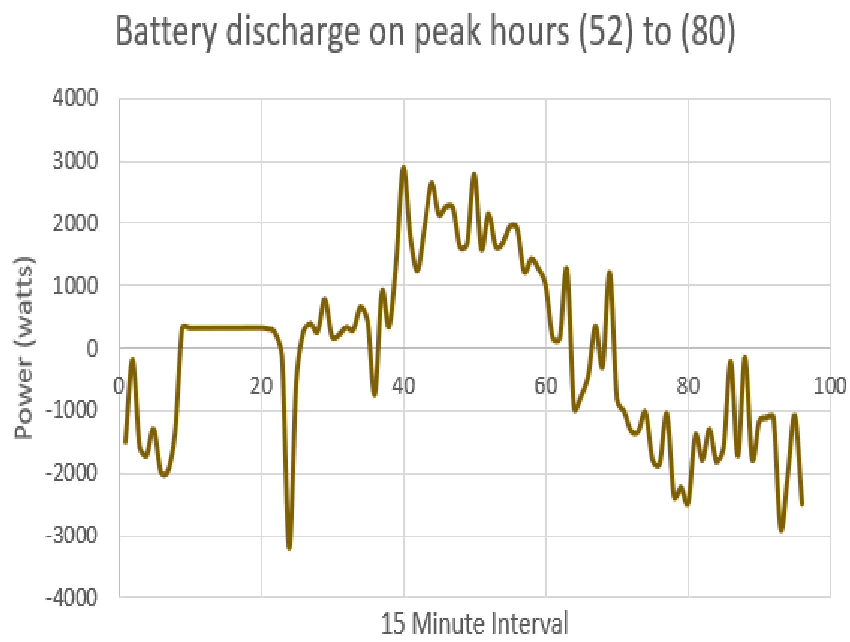

Fig. 5.36. Battery Discharge-day 4 (ASU Algorithm)

d. Site Demand

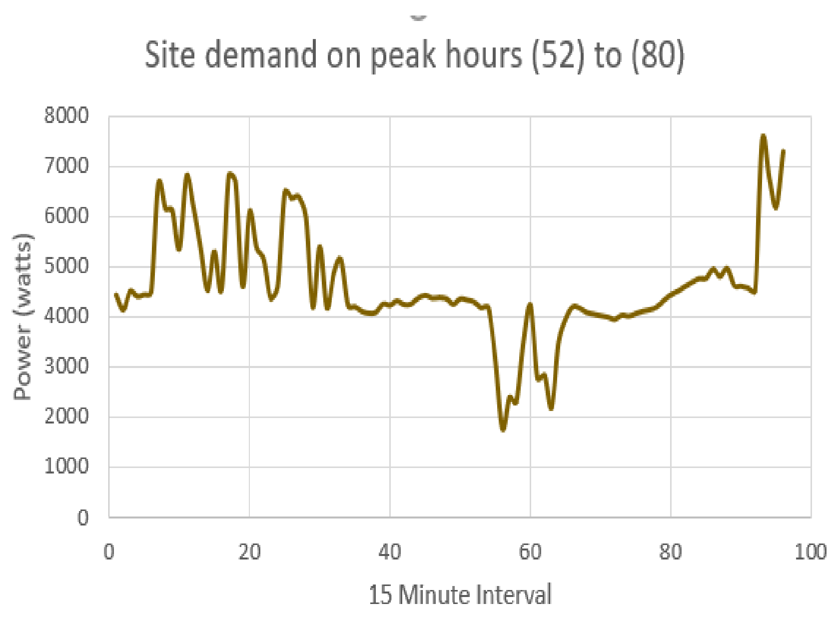

Fig. 5.37. Site Demand-day 1 (ASU Algorithm)

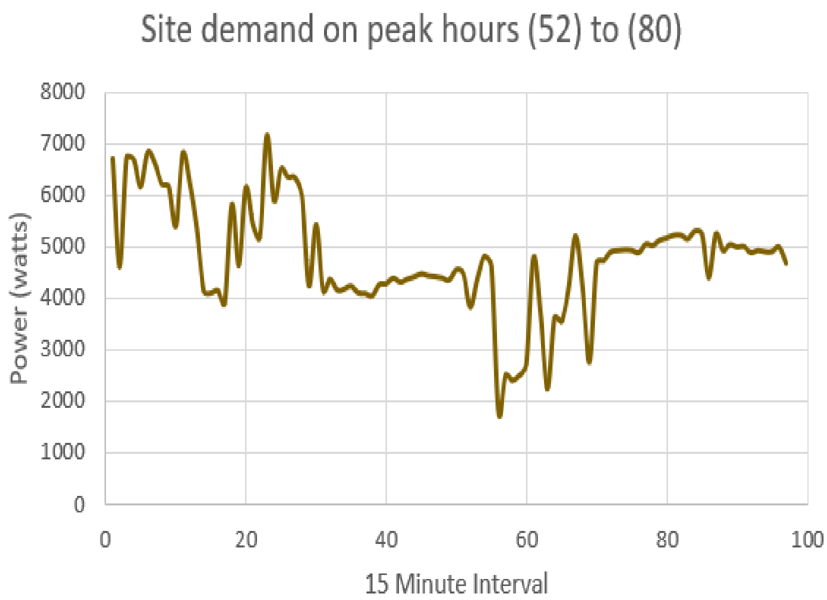

Fig. 5.38. Site Demand-day 2 (ASU Algorithm)

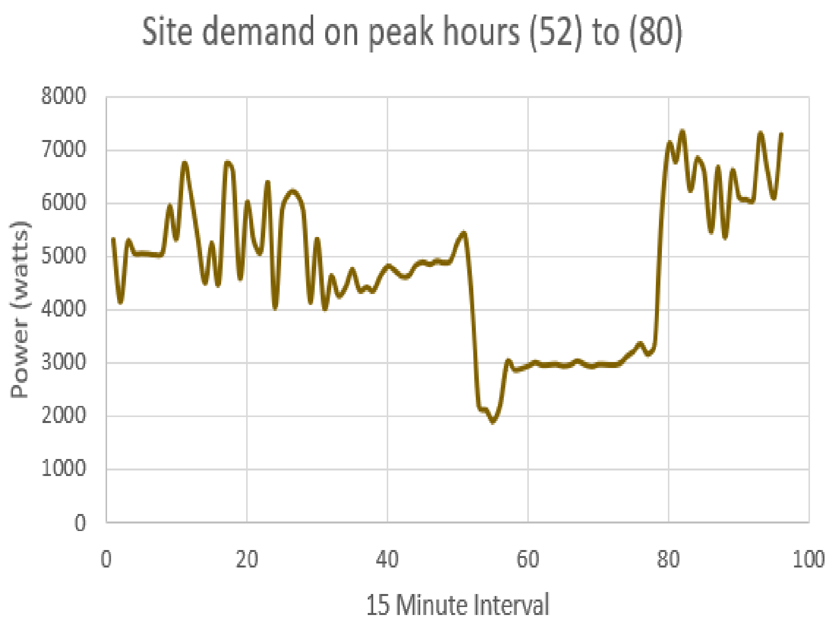

Fig. 5.39. Site Demand-day 3 (ASU Algorithm)

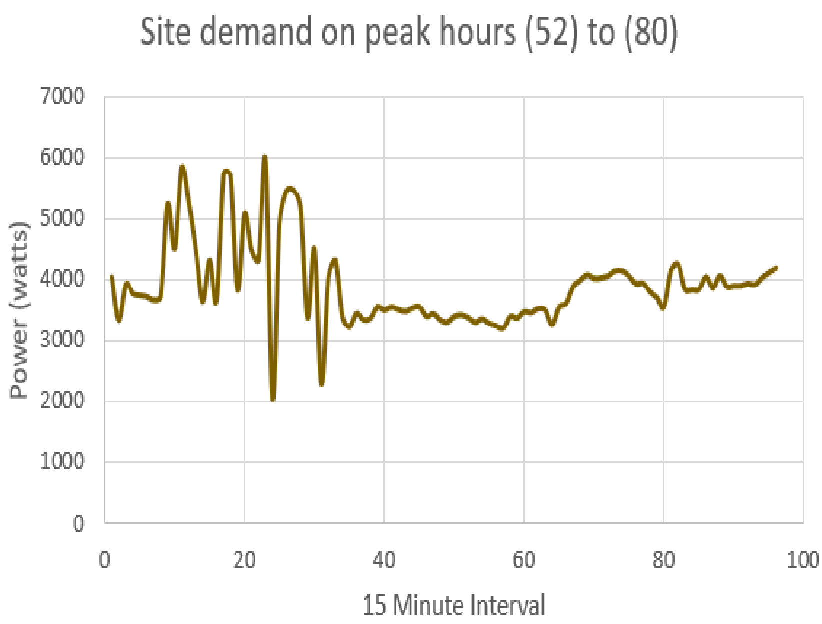

Fig. 5.40. Site Demand-day 4 (ASU Algorithm)

e. Battery SOC

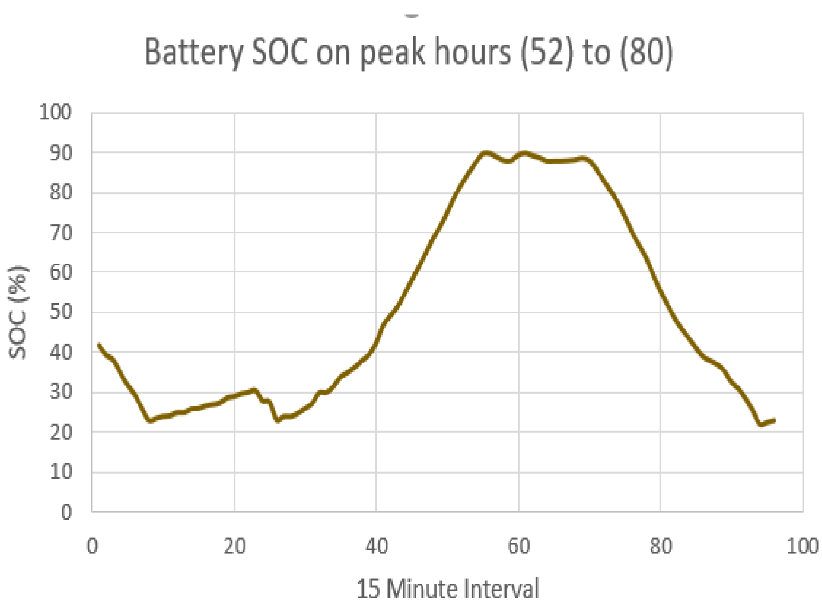

Fig. 5.41. Battery SOC-day 1 (ASU Algorithm)

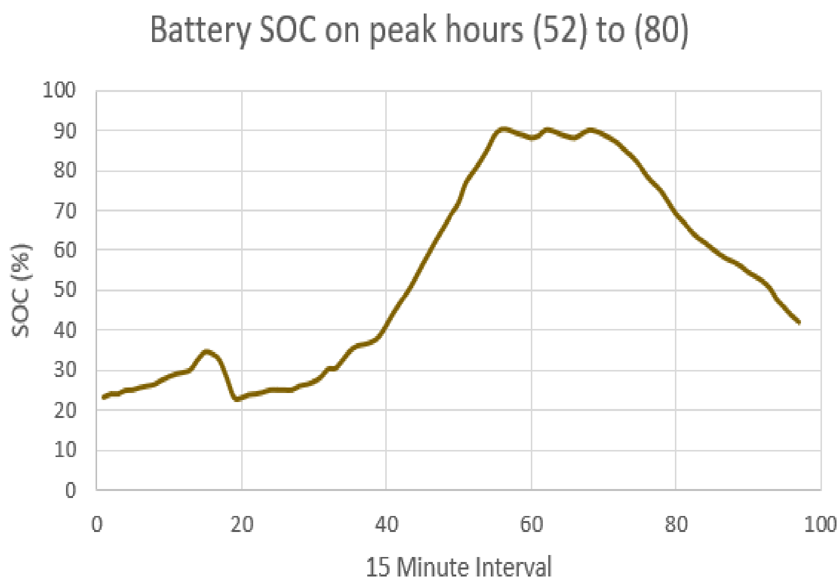

Fig. 5.42. Battery SOC-day 2 (ASU Algorithm)

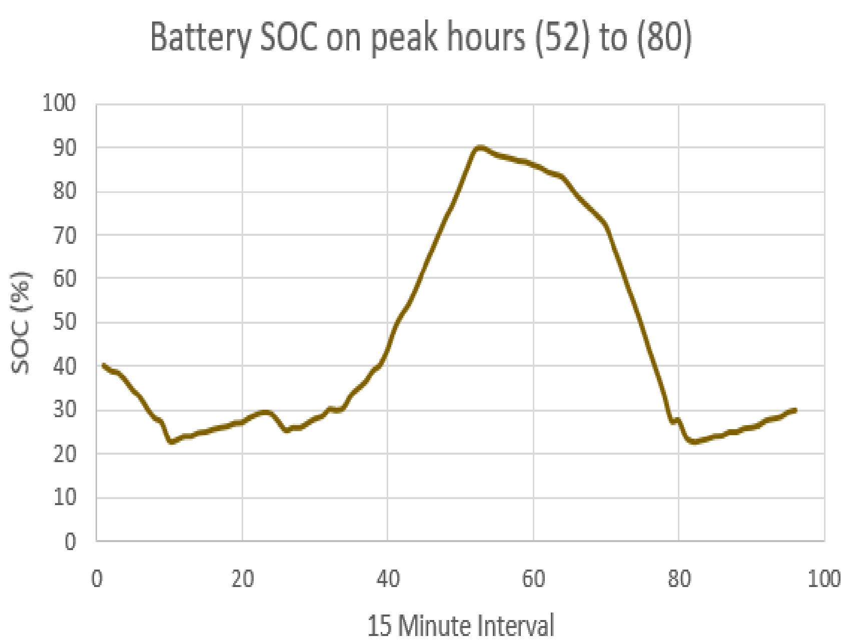

Fig. 5.43. Battery SOC-day 3 (ASU Algorithm)

`

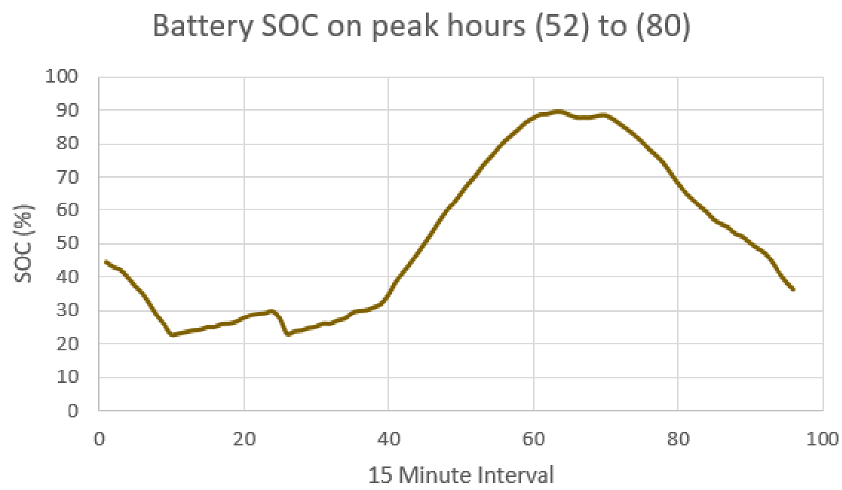

Fig. 5.44. Battery SOC-day 4 (ASU Algorithm)

Table 5.7. Results from ASU Algorithm

Days	PV energy	Load energy	Peak demand
Day 1	38.251 kWh	145.652 kWh	4800 W
Day 2	38.735 kWh	147.226 kWh	5080 W
Day 3	34.947 kWh	138.610 kWh	4100 W
Day 4	39.268 kWh	145.330 kWh	4200 W

5.5 Comparison

The comparison is made between the results obtained using ASU algorithm and those using commercial algorithm. For making this comparison, 8 similar days are chosen. The system is made to run on commercial demand charge reduction algorithm for the first four days and ASU algorithm for the next four days. The load profile is kept constant for all the 8 days. From Table 5.6 and Table 5.7 it can be noted that the load is kept constant and the similarity between the days can be assessed by the amount of PV energy captured in a day.

From the data shown in the above sections from Table 5.6 and Table 5.7 , it can be inferred that the peak power drawn from the grid was comparatively less using ASU algorithm. From the SOC graphs obtained using the two algorithms it can be noticed that ASU algorithm has better usage of battery energy during the on-peak hours.

`

CHAPTER 6- STATISTICAL ANALYSIS

6.1 Introduction

The statistical analysis is performed on the data collected from the PV-battery system. The analysis shown in the report is for the data collected in the month of February. The program to perform the analysis is coded in MATLAB and is attached in the appendix. The algorithm is developed in a step by step procedure for each data point mentioned below so that the future developments can be made easier. The interval between the data points is one hour.

6.2 PV Data

Fig. 6.1 shows the average hourly PV kW for the month of February. The straight line on Fig. 6.1 indicates the average daily PV kW. The Fig. 6.2 shows the total PV energy in a day, the dotted straight line in Fig. 6.2 indicates the average PV energy for the selected data.

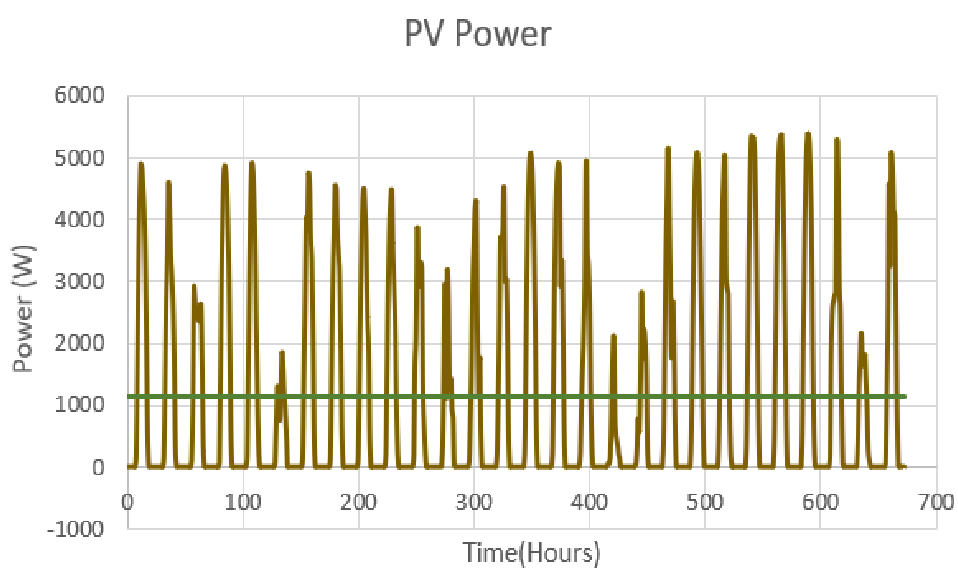

Fig. 6.1. PV Data

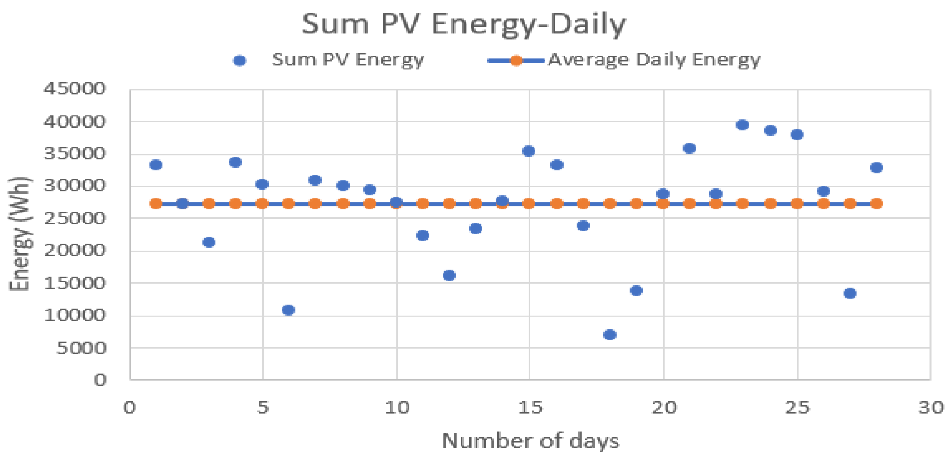

Fig. 6.2. Daily PV Energy

- Maximum sum of daily PV energy for selected data is 39.308 kWh.
- Minimum sum of daily PV energy for selected data is 6.878 kWh.
- Average sum of daily PV energy for selected data is 27.141 kWh.
- Average hourly PV energy in a day is 1130.913 Wh.

6.3 Battery Profile

Fig. 6.3 shows the battery profile for the dataset. +ve indicates charging and –ve indicates discharging for the battery profile.

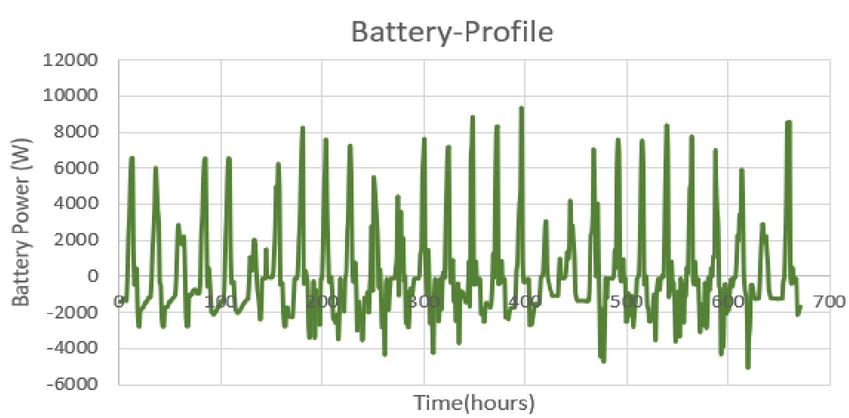

Fig. 6.3. 1-hour Average Battery Data

- Maximum amount of battery energy to grid during on-peak hours 13.337 kWh.
- Minimum amount of battery energy to grid during on-peak hours 3.378 kWh.
- Average amount of battery energy to grid during on-peak hours 8.795 kWh.
- Maximum amount of battery energy to load during on-peak hours 11.163 kWh.
- Minimum amount of battery energy to load during on-peak hours 1.274 kWh.
- Average amount of battery energy to load during on-peak hours 5.721 kWh.

6.4 Site Demand

Fig. 6.4 shows the site demand profile which is the amount of energy drawn or given to grid. +ve direction indicates energy taken from grid and –ve direction indicates energy given to grid.

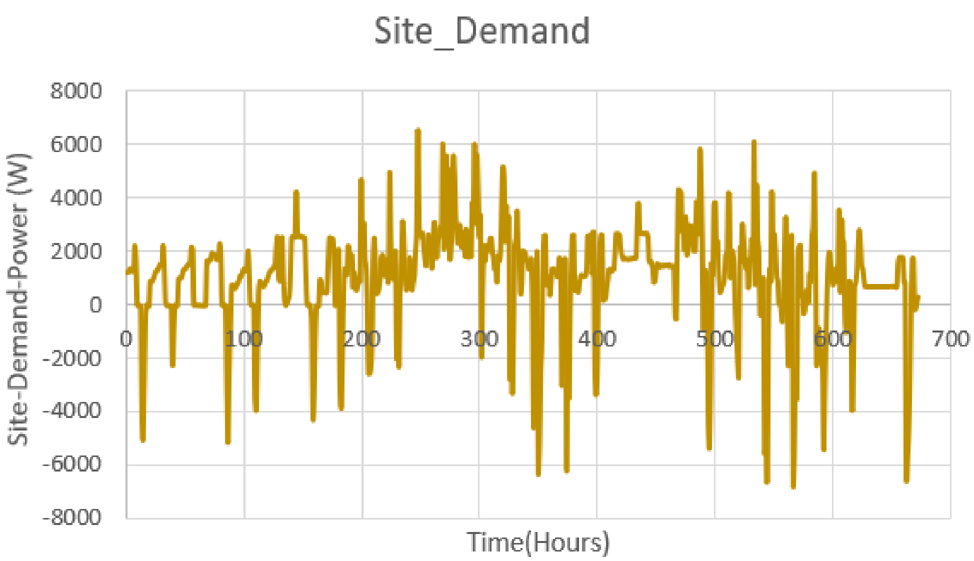

Fig. 6.4. 1-hour Average Site Demand Profile

- Maximum hourly rate of energy taken from grid 6521.441 Wh.
- Maximum hourly rate of energy given to grid 6667.19 Wh.
- Maximum amount of grid energy to load during on-peak hours 27.680 kWh.

- Minimum amount of grid energy to load during on-peak hours 0 kWh.
- Average amount of grid energy to load during on-peak hours 8.201 kWh.
- Maximum amount of energy to grid during on-peak hours 0 kWh.
- Minimum amount of energy to grid during on-peak hours -20.224 kWh.
- Average amount of energy to grid during on-peak hours -7.639 kWh.

6.5 Battery SOC

Fig. 6.5 indicates the battery SOC over the dataset. The plot indicates the days where the battery is completed charged and is completely discharged.

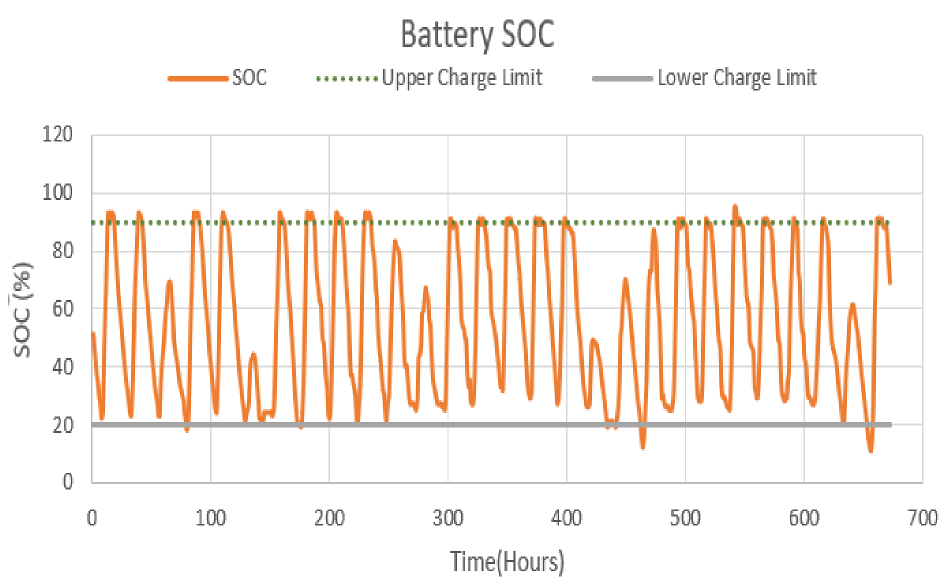

Fig. 6.5. Battery SOC Dataset

- Number of days battery completely charged =20.
- Upper limit=90 and lower limit=20.
- Percentage of days battery is completely charged= 71.4.
- Number of days battery is completely discharged =6.
- Percentage of days battery is completely discharged= 21.4.

6.6 Site Demand Chargeable Day

The customers are charged as per the customer generation price plan which accounts for a major portion of electricity bill coming from the one-time demand in a month. The algorithm allows to find the day that has the maximum demand. The plot for important data points are as shown in Fig. 6.6- Fig. 6.9.

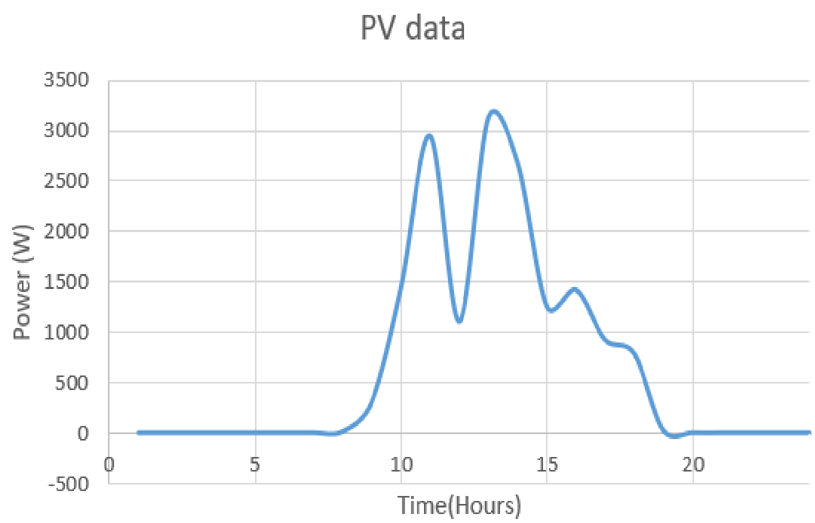

Fig. 6.6. PV Power on Day of Maximum Demand

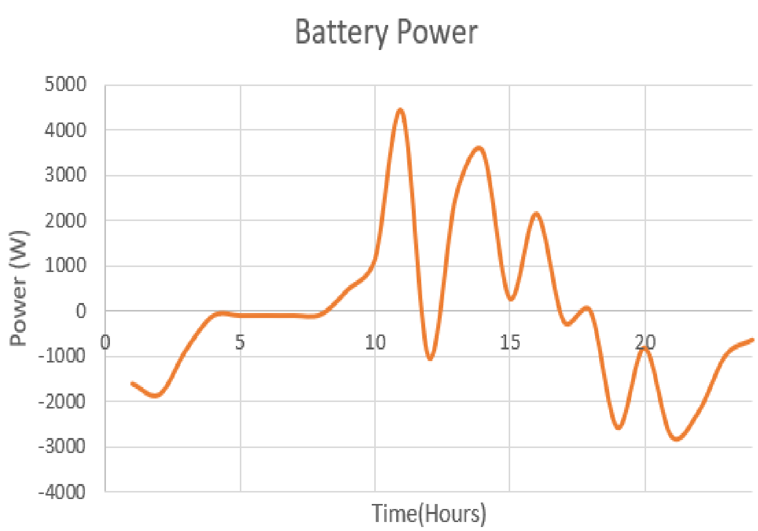

Fig. 6.7. Battery Power on Day of Maximum Demand

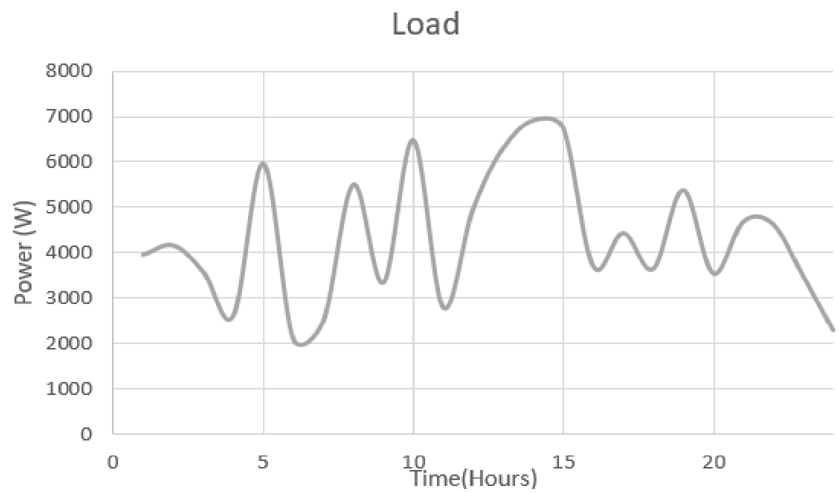

Fig. 6.8. Load on Day of Maximum Demand

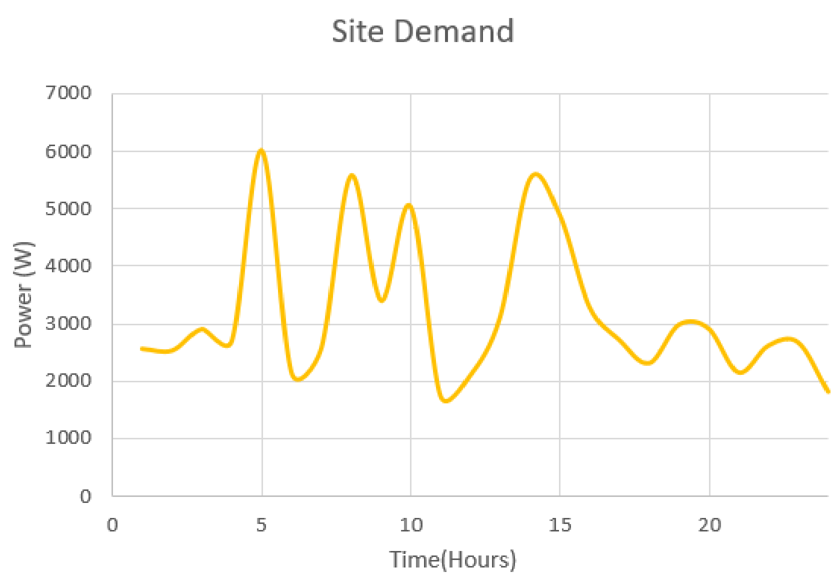

Fig. 6.9. Site Demand Power on the Day of Maximum Demand

- The 12th day is found to have the maximum demand requirement during the on-peak hours for the selected data.
- On-peak hours are from 13 to 20 hours.
- Peak demand during on-peak hours is 5515.28 watts.

CHAPTER 7- CONCLUSION AND FUTURE WORK

7.1 Conclusion

The research sheds light on various aspects in which the residential PV along with the battery system can be used with improved benefits. The research also implies and emphasizes upon the fact that the policies and the tariffs play a major role in maximizing benefits from the battery-assisted PV residential systems. Another important factor contributing to profit maximization could be a significant reduction in system cost especially the battery as its prices are expected to see a downward trend over the years [21].

Chapter 2 explains the experimental set up of the rooftop PV -battery system and load bank integration to the system. The load is a resistive load and is controlled by a microcontroller which in turn controls the digital relay that operates the load steps.

Chapter 3 summarizes that the system sizing and battery discharge strategy may be designed according to the load characteristics of the customers and a smart, artificially intelligent system may be established which would learn the load patterns and alter the battery discharge strategy and in such a system, customers benefit can be maximized.

Chapter 4 documents the results from the transient test performed on the PV-battery system. The test is performed on both the off-grid mode and the on-grid mode. It can be concluded that during the off-grid mode the number of transient cycles might increase and trigger a breaker.

In Chapter 5 a load selection technique for the load banks at ASU is explained with an example. ASU developed demand charge algorithm is explained and a comparison is made with the commercial algorithm available in market and the results are tabulated. In Chapter 6 the results from the statistical analysis algorithm developed in MATLAB are explained.

7.2 Future Work

The customer benefit algorithm developed in Chapter 3 has assumed a new cost to battery after 7000 cycles are complete and efficiency of the system (inverter and battery) are taken as 100%. Integrating the battery and inverter efficiency can develop a more accurate model. By adding inductive load bank to the existing system, ASU and SRP can run different residential load profiles and study the metering effects. Allowing the ASU algorithm and commercial algorithm to run for one year and document the performance of both the algorithms.

REFERENCES

[1] Martins, Rodrigo, Petr Musilek, and Holger C. Hesse, "Optimization of photovoltaic power self-consumption using linear programming," in *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC).*

[2] Gitizadeh, Mohsen, and Hamid Fakharzadegan, "Effects of electricity tariffs on optimal battery energy storage sizing in residential PV/storage systems," in *2013 International Conference on Energy Efficient Technologies for Sustainability (ICEETS).*

[3] Ru, Yu, Jan Kleissl, and Sonia Martinez, "Storage size determination for grid-connected photovoltaic systems," in *IEEE Transactions on Sustainable Energy* 4.1 (2013), vol. 4, pp. 68-81.

[4] Alam, M. J. E., K. M. Muttaqi, and Darmawan Sutanto, "Mitigation of rooftop solar PV impacts and evening peak support by managing available capacity of distributed energy storage systems," in *IEEE Transactions on Power Systems*, vol. 28.4, pp. 3874-3884, 2013.

[5] Ahmad, Amir, and Lipika Dey, "A k-mean clustering algorithm for mixed numeric and categorical data," *Data & Knowledge Engineering*, vol. 63.2, pp. 503-527, Nov 2007.

[6] Hanna, R., J. Kleissl, A. Nottrott, and M. Ferry, "Energy dispatch schedule optimization for demand charge reduction using a photovoltaic-battery storage system with solar forecasting," *Solar Energy*, vol.103, pp. 269-287, May 2014.

[7] Srpnet.com- Customer Generation Price Plan for SRP residential electric customers [online] Available at: https://www.srpnet.com/prices/home/customergenerated.aspx

[8] Riffonneau, Yann, Seddik Bacha, Franck Barruel, and Stephane Ploix, "Optimal power flow management for grid connected PV systems with batteries," *IEEE Transactions on Sustainable Energy*, vol. 2, pp. 309-320, Feb 2011.

[9] Zarezadeh, Esmail, Hamid Fakharzadegan, Ayaz Ghorbani, and Hossin Fathabadi, "A probabilistic approach to determine PV array size and battery capacity used in grid-connected PV systems," in *2015 23rd Iranian Conference on Electrical Engineering (ICEE)*, pp. 1533-1538.

[10] Wai, Rong-Jong, Wen-Hung Wang, and Chung-You Lin, "High-performance stand-alone photovoltaic generation system," *IEEE Transactions on Industrial Electronics*, vol. 55, pp. 240-250, Jan 2008.

[11] Leadbetter, Jason, and Lukas Swan, "Battery storage system for residential electricity peak demand shaving," *Energy and Buildings*, vol.55, pp. 685-692, Dec 2012.

[12] Shimada, Takae, and Kosuke Kurokawa, "Grid-connected photovoltaic systems with battery storages control based on insolation forecasting using weather forecast," *Renewable Energy*, pp. 228-230, 2006.

[13] Karmacharya, Samir, Ghanim Putrus, Chris Underwood, and Khamid Mahkamov, "Evaluation of domestic electrical demand and its effect on low voltage network," in *47th International Universities Power Engineering Conference (UPEC), 2012*, pp. 1-4. IEEE.

[14] Thomson, Murray, and David G. Infield, "Network power-flow analysis for a high penetration of distributed generation," *IEEE Transactions on Power Systems*, vol. 22.3, pp. 1157-1162, 2007.

[15] Tonkoski, Reinaldo, Luiz AC Lopes, and Tarek HM El-Fouly, "Coordinated active power curtailment of grid connected PV inverters for overvoltage prevention," *IEEE Transactions on Sustainable Energy*, vol. 2, pp. 139-147, Apr 2011.

[16] Papic, Igor, "Simulation model for discharging a lead-acid battery energy storage system for load leveling," *IEEE transactions on Energy Conversion*, vol. 21, pp. 608-615, Jun 2006.

[17] Muselli, M. N. G. L. A., G. Notton, and A. Louche, "Design of hybrid-photovoltaic power generator, with optimization of energy management," *Solar Energy*, vol. 65, pp. 143-157, Feb 1999.

[18] Barton, John P., and David G. Infield, "Energy storage and its use with intermittent renewable energy," *IEEE transactions on Energy Conversion*, vol.19, pp. 441-448, Jun 2004.

[19] Sheen, J-N., C-S. Chen, and J-K. Yang, "Time-of-use pricing for load management programs in Taiwan Power Company," *IEEE Transactions on Power Systems*, vol. 9, pp. 388-396, Feb 1994.

[20] Arizona – State Energy Profile Analysis – U.S. Energy Information Administration (EIA). [online] Available at : https://www.eia.gov/state/analysis.php?sid=AZ.

[21] Bloomberg New Energy Finance (2017) – Lithium- ion Battery Costs: Squeezed Margins and New Business Models | Bloomberg New Energy Finance. [online] Available at: https://about.bnef.com/blog/lithium-ion-battery-costs-squeezed-margins-new-business-models.

[22] Nrel.gov-Publications | NREL. [online] Available at: https://www.nrel.gov/research/publications.html

APPENDIX A

CODE: PYTHON CODE TO OPERATE RASPBERRY PI

`

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(3,GPIO.OUT)
GPIO.setup(5,GPIO.OUT)
GPIO.setup(7,GPIO.OUT)
GPIO.setup(8,GPIO.OUT)
GPIO.setup(10,GPIO.OUT)
GPIO.setup(12,GPIO.OUT)
GPIO.setup(11,GPIO.OUT)
GPIO.setup(13,GPIO.OUT)
GPIO.setup(15,GPIO.OUT)
GPIO.setup(16,GPIO.OUT)
time.sleep(900)

GPIO.output(3,1)
GPIO.output(5,1)
GPIO.output(7,1)
GPIO.output(8,0)
GPIO.output(10,0)
GPIO.output(12,0)
GPIO.output(11,0)
GPIO.output(13,0)
GPIO.output(15,0)
GPIO.output(16,0)
time.sleep(900)
```


`

APPENDIX B

CODE: BASIC PLAN E-23 CODE

`

```
%Data plan + load data calculation
filename = 'Loaddata';
hourly_load = xlsread(filename,'B:B');

%E-23 Plan Hourly Prices
Winter_hourly_price = 0.0793;
Summer_hourly_price1 = 0.1082;
Summer_hourly_price2 = 0.1101;
Summer_hourly_price3 = 0.1206;
Summerpeak_hourly_price1 = 0.1148;
Summerpeak_hourly_price2 = 0.1160;
Summerpeak_hourly_price3 = 0.1311;

%January
Jan_total_hours = xlsread(filename, 'B1:B744');
Jan_total_days = (Jan_total_hours/1000) * Winter_hourly_price;
Jan_total = sum(Jan_total_days);
%disp(Jan_total);

%February
Feb_total_hours = xlsread(filename, 'B745:B1416');
Feb_total_days = (Feb_total_hours/1000) * Winter_hourly_price;
Feb_total = sum(Feb_total_days);
%disp(Feb_total);

%March
Mar_total_hours = xlsread(filename, 'B1417:B2160');
Mar_total_days = (Mar_total_hours/1000) * Winter_hourly_price;
Mar_total = sum(Mar_total_days);
%disp(Mar_total);

%April
Apr_total_hours = xlsread(filename, 'B2161:B2880');
Apr_total_days =(Apr_total_hours/1000) * Winter_hourly_price;
Apr_total = sum(Apr_total_days);
%disp(Apr_total);

%May
May_total_hours = xlsread(filename, 'B2881:B3624');
May_total_hours_sum = sum(May_total_hours)/1000;
if May_total_hours_sum <= 700;
May_total_days = May_total_hours_sum * Summer_hourly_price1;
elseif May_total_hours_sum >= 701 && May_total_hours_sum <=2000;
    May_total_days = (May_total_hours_sum-700) *
Summer_hourly_price2+(700*Summer_hourly_price1);
else May_total_hours_sum > 2000;
    May_total_days = (May_total_hours_sum-2000) *
Summer_hourly_price3+(1300*Summer_hourly_price2)+(700*Summer_hourly_pri
ce1);
end
May_total = May_total_days;
%disp(May_total);

%June
```

```
Jun_total_hours = xlsread(filename, 'B3625:B4344');
Jun_total_hours_sum = sum(Jun_total_hours)/1000;
if Jun_total_hours_sum <= 700;
Jun_total_days = Jun_total_hours_sum * Summer_hourly_price1;
elseif Jun_total_hours_sum >= 701 && Jun_total_hours_sum <=2000;
    Jun_total_days = (Jun_total_hours_sum-700) *
Summer_hourly_price2+(700*Summer_hourly_price1);
else Jun_total_hours_sum > 2000;
    Jun_total_days = (Jun_total_hours_sum-2000) *
Summer_hourly_price3+(1300*Summer_hourly_price2)+(700*Summer_hourly_pri
ce1);
end
Jun_total = Jun_total_days;
%disp(Jun_total);

%July
Jul_total_hours = xlsread(filename, 'B4345:B5088');
Jul_total_hours_sum = sum(Jul_total_hours)/1000;
if Jul_total_hours_sum <= 700;
Jul_total_days = Jul_total_hours_sum * Summerpeak_hourly_price1;
elseif Jul_total_hours_sum >= 701 && Jul_total_hours_sum <=2000;
    Jul_total_days = (Jul_total_hours_sum-700) *
Summerpeak_hourly_price2+(700*Summerpeak_hourly_price1);
else Jul_total_hours_sum > 2000;
    Jul_total_days = (Jul_total_hours_sum-2000) *
Summerpeak_hourly_price3+(1300*Summerpeak_hourly_price2)+(700*Summerpea
k_hourly_price1);
end
Jul_total = Jul_total_days;
%disp(Jul_total);

%August
Aug_total_hours = xlsread(filename, 'B5089:B5832');
Aug_total_hours_sum = sum(Aug_total_hours)/1000;
if Aug_total_hours_sum <= 700;
Aug_total_days = Aug_total_hours_sum * Summerpeak_hourly_price1;
elseif Aug_total_hours_sum >= 701 && Aug_total_hours_sum <=2000;
    Aug_total_days = (Aug_total_hours_sum-700) *
Summerpeak_hourly_price2+(700*Summerpeak_hourly_price1);
else Aug_total_hours_sum > 2000;
    Aug_total_days = (Aug_total_hours_sum-2000) *
Summerpeak_hourly_price3+(1300*Summerpeak_hourly_price2)+(700*Summerpea
k_hourly_price1);
end
Aug_total = Aug_total_days;
%disp(Aug_total);

%September
Sep_total_hours = xlsread(filename, 'B5833:B6552');
Sep_total_hours_sum = sum(Sep_total_hours)/1000;
if Sep_total_hours_sum <= 700;
Sep_total_days = Sep_total_hours_sum * Summer_hourly_price1;
elseif Sep_total_hours_sum >= 701 && Sep_total_hours_sum <=2000;
```

```
    Sep_total_days = (Sep_total_hours_sum-700) *
Summer_hourly_price2+(700*Summer_hourly_price1);
else Sep_total_hours_sum > 2000;
    Sep_total_days = (Sep_total_hours_sum-2000) *
Summer_hourly_price3+(1300*Summer_hourly_price2)+(700*Summer_hourly_pri
ce1);
end
Sep_total = Sep_total_days;
%disp(Sep_total);

%October
Oct_total_hours = xlsread(filename, 'B6553:B7296');
Oct_total_hours_sum = sum(Oct_total_hours)/1000;
if Oct_total_hours_sum <= 700;
Oct_total_days = Oct_total_hours_sum * Summer_hourly_price1;
elseif Oct_total_hours_sum >= 701 && Oct_total_hours_sum <=2000;
    Oct_total_days = (Oct_total_hours_sum-700) *
Summer_hourly_price2+(700*Summer_hourly_price1);
else Oct_total_hours_sum > 2000;
    Oct_total_days = (Oct_total_hours_sum-2000) *
Summer_hourly_price3+(1300*Summer_hourly_price2)+(700*Summer_hourly_pri
ce1);
end
Oct_total = Oct_total_days;
%disp(Oct_total);

%November
Nov_total_hours = xlsread(filename, 'B7297:B8016');
Nov_total_days = (Nov_total_hours/1000) * Winter_hourly_price;
Nov_total = sum(Nov_total_days);
%disp(Nov_total);

%December
Dec_total_hours = xlsread(filename, 'B8017:B8760');
Dec_total_days = (Dec_total_hours/1000) * Winter_hourly_price;
Dec_total = sum(Dec_total_days);
%disp(Dec_total);

%Total Annual Price
Annual_total =
(20*12)+Jan_total+Feb_total+Mar_total+Apr_total+May_total+Jun_total+Jul
_total+Aug_total+Sep_total+Oct_total+Nov_total+Dec_total;
disp(Annual_total);

%Electricity price increase from 2017-2037 annualy and total prices of
all added
%Increase#### is the percent increase/decrease between the two years

%2017-2018
Increase1718 = xlsread(filename, 'B8761:B8761');
Annual_inc_1718 = Annual_total * (Increase1718/100) + Annual_total;
%disp(Annual_inc_1718); %decrease

%2018-2019
```

```
Increase1819 = xlsread(filename, 'B8762:B8762');
Annual_inc_1819 = Annual_inc_1718 * (Increase1819/100) +
Annual_inc_1718;
%disp(Annual_inc_1819); %increase

%2019-2020
Increase1920 = xlsread(filename, 'B8763:B8763');
Annual_inc_1920 = Annual_inc_1819 * (Increase1920/100) +
Annual_inc_1819;
%disp(Annual_inc_1920); %increase

%2020-2021
Increase2021 = xlsread(filename, 'B8764:B8764');
Annual_inc_2021 = Annual_inc_1920 * (Increase2021/100) +
Annual_inc_1920;
%disp(Annual_inc_2021); %increase

%2021-2022
Increase2122 = xlsread(filename, 'B8765:B8765');
Annual_inc_2122 = Annual_inc_2021 * (Increase2122/100) +
Annual_inc_2021;
%disp(Annual_inc_2122); %increase

%2022-2023
Increase2223 = xlsread(filename, 'B8766:B8766');
Annual_inc_2223 = Annual_inc_2122 * (Increase2223/100) +
Annual_inc_2122;
%disp(Annual_inc_2223); %increase

%2023-2024
Increase2324 = xlsread(filename, 'B8767:B8767');
Annual_inc_2324 = Annual_inc_2223 * (Increase2324/100) +
Annual_inc_2223;
%disp(Annual_inc_2324); %decrease

%2024-2025
Increase2425 = xlsread(filename, 'B8768:B8768');
Annual_inc_2425 = Annual_inc_2324 * (Increase2425/100) +
Annual_inc_2324;
%disp(Annual_inc_2425); %increase

%2025-2026
Increase2526 = xlsread(filename, 'B8769:B8769');
Annual_inc_2526 = Annual_inc_2425 * (Increase2526/100) +
Annual_inc_2425;
%disp(Annual_inc_2526); %increase

%2026-2027
Increase2627 = xlsread(filename, 'B8770:B8770');
Annual_inc_2627 = Annual_inc_2526 * (Increase2627/100) +
Annual_inc_2526;
%disp(Annual_inc_2627); %increase

%2027-2028
```


`

```
Increase2728 = xlsread(filename, 'B8771:B8771');
Annual_inc_2728 = Annual_inc_2627 * (Increase2728/100) +
Annual_inc_2627;
%disp(Annual_inc_2728); %increase

%2028-2029
Increase2829 = xlsread(filename, 'B8772:B8772');
Annual_inc_2829 = Annual_inc_2728 * (Increase2829/100) +
Annual_inc_2728;
%disp(Annual_inc_2829); %increase

%2029-2030
Increase2930 = xlsread(filename, 'B8773:B8773');
Annual_inc_2930 = Annual_inc_2829 * (Increase2930/100) +
Annual_inc_2829;
%disp(Annual_inc_2930); %increase

%2030-2031
Increase3031 = xlsread(filename, 'B8774:B8774');
Annual_inc_3031 = Annual_inc_2930 * (Increase3031/100) +
Annual_inc_2930;
%disp(Annual_inc_3031); %decrease

%2031-2032
Increase3132 = xlsread(filename, 'B8775:B8775');
Annual_inc_3132 = Annual_inc_3031 * (Increase3132/100) +
Annual_inc_3031;
%disp(Annual_inc_3132); %decrease

%2032-2033
Increase3233 = xlsread(filename, 'B8776:B8776');
Annual_inc_3233 = Annual_inc_3132 * (Increase3233/100) +
Annual_inc_3132;
%disp(Annual_inc_3233); %decrease

%2033-2034
Increase3334 = xlsread(filename, 'B8777:B8777');
Annual_inc_3334 = Annual_inc_3233 * (Increase3334/100) +
Annual_inc_3233;
%disp(Annual_inc_3334); %decrease

%2034-2035
Increase3435 = xlsread(filename, 'B8778:B8778');
Annual_inc_3435 = Annual_inc_3334 * (Increase3435/100) +
Annual_inc_3334;
%disp(Annual_inc_3435); %increase

%2035-2036
Increase3536 = xlsread(filename, 'B8779:B8779');
Annual_inc_3536 = Annual_inc_3435 * (Increase3536/100) +
Annual_inc_3435;
%disp(Annual_inc_3536); %increase

%2036-2037
```

```
Increase3637 = xlsread(filename, 'B8780:B8780');
Annual_inc_3637 = Annual_inc_3536 * (Increase3637/100) +
Annual_inc_3536;
%disp(Annual_inc_3637); %decrease

%2037-2038
Increase3738 = xlsread(filename, 'B8781:B8781');
Annual_inc_3738 = Annual_inc_3637 * (Increase3738/100) +
Annual_inc_3637;
%disp(Annual_inc_3738); %increase

%Total added of all years
Total_added_years =
Annual_inc_1718+Annual_inc_1819+Annual_inc_1920+Annual_inc_2021+Annual_
inc_2122+Annual_inc_2223+Annual_inc_2324+Annual_inc_2425+Annual_inc_252
6+Annual_inc_2627+Annual_inc_2728+Annual_inc_2829+Annual_inc_2930+Annua
l_inc_3031+Annual_inc_3132+Annual_inc_3233+Annual_inc_3334+Annual_inc_3
435+Annual_inc_3536+Annual_inc_3637+Annual_inc_3738;
%disp(Total_added_years);
```


`

CODE: RESIDENTIAL PV-BATTERY SYSTEM

```
filename='sample';
PV_old=xlsread(filename,'A:A');
Load_old=xlsread(filename,'B:B');
[np,mp]=size(PV_old);
[nl,ml]=size(Load_old);
battery=29000;
d=1;
nday=np/24;
Electricity_price=0.0486;
roi=0.04;
x=15;
y=50;
Energy_charging=zeros(x,y);
Energy_discharging=zeros(x,y);
operating_cost=zeros(x,y);
for q=1:x
PV=0;
Load=0;
Load_total=0;
PV_total=0;
threshold=(q*1000)/2;
   for e=1:y
    PV=PV_old*(q/6);
    Load=Load_old;
    Load_total=sum(Load);
    PV_total=sum(PV);
    SOC=zeros(25,365);
    Load_thres=zeros(24,365);
    peak_total=zeros(24,365);
    Excess_Load=zeros(24,365);
    difference=zeros(24,365);
    PV_load=zeros(24,365);
    Total_PV_load=0;
for i=1:nday
        w=1;
        SOC(1,i)=e*1000*0.8;
  for j=(1+(24*(i-1))):(24+(24*(i-1)))
        if w<=12
         off_peak_energy(w,i)=Load(j);
           on_peak_energy(w,i)=0;
          Excess_Load(w,i)=0;

        elseif w>12 && w<=20
         off_peak_energy(w,i)=0;
         if Load(j)>(q*1000)
            on_peak_energy(w,i)=(q*1000);
            Load_thres(w,i)=Load_thres(w,i)+(Load(j)-(q*1000));
            difference(w,i)=on_peak_energy(w,i)-PV(j);
        else
            on_peak_energy(w,i)=Load(j);
            difference(w,i)=on_peak_energy(w,i)-PV(j);
         end
```


`

```
           else
           off_peak_energy(w,i)=Load(j);
              on_peak_energy(w,i)=0;
              Excess_Load(w,i)=0;
          end

          w=w+1;
    end
     sum_off_peak_energy=sum(off_peak_energy)+e*1000*0.8;


      w=1;
      sum_difference=0;
      for j=(1+(24*(i-1))):(24+(24*(i-1)))
          if difference(w,i)>0
              sum_difference=sum_difference+difference(w,i);
          end
          w=w+1;
      end

 w=1;
   for j=(1+(24*(i-1))):(24+(24*(i-1)))
       if w<12 || w>20
       if PV(j)>off_peak_energy(w,i)
          off_peak_energy(w,i)=0;
           PV_load(w,i)=PV(j);
           SOC(w+1,i)=SOC(w,i);
       else
           off_peak_energy(w,i)=off_peak_energy(w,i)-PV(j);
           SOC(w+1,i)=SOC(w,i);
       end
       else
       sum_on_peak_energy=sum(on_peak_energy);
       if PV(j)>on_peak_energy(w,i)
           PV_load(w,i)=on_peak_energy(w,i);
           if SOC(w,i) >=e*1000*0.2 &&SOC(w,i) <=e*1000*0.8
            SOC(w+1,i)=SOC(w,i)+PV(j);
             if SOC(w+1,i)>e*1000*0.8
               Excess_PV(w,i)=SOC(w,i)-e*1000*0.8;
              SOC(w+1,i)=e*1000*0.8;
                if w>1
              Energy_charging(q,e)=Energy_charging(q,e)+SOC(w+1,i)-
SOC((w),i);
                else
             Energy_charging(q,e)= Energy_charging(q,e)+SOC(w+1,i)-
SOC(1,i);
               end
               end
           end
       else
           PV_load(w,i)=PV(j);
           Excess_PV(w,i)=0;
          Excess_Load(w,i)=on_peak_energy(w,i)-PV(j);
          if Excess_Load(w,i)>0
```

```
                if sum_difference<e*1000*0.8
                    SOC(w+1,i)=SOC(w,i)- Excess_Load(w,i);
                     Excess_Load(w,i)=0;

Energy_discharging(q,e)=Energy_discharging(q,e)+SOC(w+1,i)-SOC((w),i);
                else

               SOC(w+1,i)=SOC(w,i)-
(Excess_Load(w,i)/sum_difference)*e*1000*0.8;
               Excess_Load(w,i)=Excess_Load(w,i)-
(Excess_Load(w,i)/sum_difference)*e*1000*0.8;


Energy_discharging(q,e)=Energy_discharging(q,e)+SOC(w+1,i)-SOC((w),i);
                end
            end

          end

        end
        peak_total(w,i)=Excess_Load(w,i)+ Load_thres(w,i);
        Total_PV_load=Total_PV_load+PV_load(w,i);


        w=w+1;
    end




end
sum_peak_total=sum(peak_total);
peak_demand_value=max(peak_total);
%Electricity Bill

%january_bill
peak_demand_jan=0;
for m=2:31
    if peak_demand_value(1,m)>peak_demand_value(1,m-1);
        peak_demand_jan=peak_demand_value(1,m);
    else
        peak_demand_jan=peak_demand_value(1,m-1);
       peak_demand_value(1,m)=peak_demand_value(1,m-1);

    end
end
january_bill=0;
january_bill_final=0;
peak_demand_charge=0;
for m=1:31
    x=m/7;
    if x==5||x==6
```

```
cost_day(m)=0.0390*(sum_off_peak_energy(1,m)/1000)+0.0390*(sum_peak_tot
al(1,m)/1000)+(1.0813);
    else

cost_day(m)=0.0390*(sum_off_peak_energy(1,m)/1000)+0.0430*((sum_peak_to
tal(1,m))/1000)+(1.0813);
    end
    if peak_demand_jan<3000 ||peak_demand_jan==3000;
        peak_demand_charge=((peak_demand_jan)/1000)*3.55;
    elseif peak_demand_jan<10000 ||peak_demand_jan==10000;
        peak_demand_charge=3*3.55+((peak_demand_jan-3000)/1000)*5.68;
    else
        peak_demand_charge=3*3.55+7*5.68+((peak_demand_jan-
10000)/1000)*9.74;
    end
    january_bill=january_bill+cost_day(m);
end
january_bill_final=january_bill+peak_demand_charge;


%febuary_bill
peak_demand_feb=0;
for m=33:59
    if peak_demand_value(1,m)>peak_demand_value(1,m-1);
        peak_demand_feb=peak_demand_value(1,m);
    else
        peak_demand_feb=peak_demand_value(1,m-1);
       peak_demand_value(1,m)=peak_demand_value(1,m-1);

    end
end
febuary_bill=0;
febuary_bill_final=0;
peak_demand_charge=0;
for m=32:59
    x=m/7;
    if x==5||x==6

cost_day(m)=0.0390*(sum_off_peak_energy(1,m)/1000)+0.0390*(sum_peak_tot
al(1,m)/1000)+(1.0813);
    else

cost_day(m)=0.0390*(sum_off_peak_energy(1,m)/1000)+0.0430*((sum_peak_to
tal(1,m))/1000)+(1.0813);
    end
    if peak_demand_feb<3000  ||peak_demand_feb==3000;
        peak_demand_charge=((peak_demand_feb)/1000)*3.55;
    elseif peak_demand_feb<10000 ||peak_demand_feb==10000;
        peak_demand_charge=3*3.55+((peak_demand_feb-3000)/1000)*5.68;
    else
        peak_demand_charge=3*3.55+7*5.68+((peak_demand_feb-
10000)/1000)*9.74;
    end
```

```
    febuary_bill=febuary_bill+cost_day(m);
end
febuary_bill_final=febuary_bill+peak_demand_charge;


%march_bill
peak_demand_mar=0;
for m=61:90
    if peak_demand_value(1,m)>peak_demand_value(1,m-1);
        peak_demand_mar=peak_demand_value(1,m);
    else
        peak_demand_mar=peak_demand_value(1,m-1);
       peak_demand_value(1,m)=peak_demand_value(1,m-1);

    end
end
march_bill=0;
march_bill_final=0;
peak_demand_charge=0;
for m=60:90
    x=m/7;
    if x==5||x==6

cost_day(m)=0.0390*(sum_off_peak_energy(1,m)/1000)+0.0390*((sum_peak_to
tal(1,m))/1000)+(1.0813);
    else

cost_day(m)=0.0390*(sum_off_peak_energy(1,m)/1000)+0.0430*((sum_peak_to
tal(1,m))/1000)+(1.0813);
    end
    if peak_demand_mar<3000  ||peak_demand_mar==3000;
        peak_demand_charge=((peak_demand_mar)/1000)*3.55;
    elseif peak_demand_mar<10000 ||peak_demand_mar==10000;
        peak_demand_charge=3*3.55+((peak_demand_mar-3000)/1000)*5.68;
    else
        peak_demand_charge=3*3.55+7*5.68+((peak_demand_mar-
10000)/1000)*9.74;
    end
    march_bill=march_bill+cost_day(m);
end
march_bill_final=march_bill+peak_demand_charge;

%april_bill
peak_demand_apr=0;
for m=92:120
    if peak_demand_value(1,m)>peak_demand_value(1,m-1);
        peak_demand_apr=peak_demand_value(1,m);
    else
        peak_demand_apr=peak_demand_value(1,m-1);
       peak_demand_value(1,m)=peak_demand_value(1,m-1);

    end
end
april_bill=0;
```

```
april_bill_final=0;
peak_demand_charge=0;
for m=91:120
    x=m/7;
    if x==5||x==6

cost_day(m)=0.0390*(sum_off_peak_energy(1,m)/1000)+0.0390*((sum_peak_to
tal(1,m))/1000)+(1.0813);
    else

cost_day(m)=0.0390*(sum_off_peak_energy(1,m)/1000)+0.0430*((sum_peak_to
tal(1,m))/1000)+(1.0813);
    end
    if peak_demand_apr<3000  ||peak_demand_apr==3000;
        peak_demand_charge=((peak_demand_apr)/1000)*3.55;
    elseif peak_demand_apr<10000 ||peak_demand_apr==10000;
        peak_demand_charge=3*3.55+((peak_demand_apr-3000)/1000)*5.68;
    else
        peak_demand_charge=3*3.55+7*5.68+((peak_demand_apr-
10000)/1000)*9.74;
    end
    april_bill=april_bill+cost_day(m);
end
april_bill_final=april_bill+peak_demand_charge;

%may_bill
peak_demand_may=0;
for m=122:151
    if peak_demand_value(1,m)>peak_demand_value(1,m-1);
        peak_demand_may=peak_demand_value(1,m);
    else
        peak_demand_may=peak_demand_value(1,m-1);
       peak_demand_value(1,m)=peak_demand_value(1,m-1);

    end
end
may_bill=0;
may_bill_final=0;
peak_demand_charge=0;
for m=121:151
    x=m/7;
    if x==5||x==6

cost_day(m)=0.0371*(sum_off_peak_energy(1,m)/1000)+0.0371*((sum_peak_to
tal(1,m))/1000)+(1.0813);
    else

cost_day(m)=0.0371*(sum_off_peak_energy(1,m)/1000)+0.0486*((sum_peak_to
tal(1,m))/1000)+(1.0813);
    end
    if peak_demand_may<3000  ||peak_demand_may==3000;
        peak_demand_charge=((peak_demand_may)/1000)*8.03;
    elseif peak_demand_may<10000 ||peak_demand_may==10000;
        peak_demand_charge=3*8.03+((peak_demand_may-3000)/1000)*14.63;
```

```
    else
        peak_demand_charge=3*8.03+7*14.63+((peak_demand_may-
10000)/1000)*27.77;
    end
    may_bill=may_bill+cost_day(m);
end
may_bill_final=may_bill+peak_demand_charge;

%june_bill
peak_demand_june=0;
for m=153:181
    if peak_demand_value(1,m)>peak_demand_value(1,m-1);
        peak_demand_june=peak_demand_value(1,m);
    else
        peak_demand_june=peak_demand_value(1,m-1);
       peak_demand_value(1,m)=peak_demand_value(1,m-1);

    end
end
june_bill=0;
june_bill_final=0;
peak_demand_charge=0;
for m=152:181
    x=m/7;
    if x==5||x==6

cost_day(m)=0.0371*(sum_off_peak_energy(1,m)/1000)+0.0371*((sum_peak_to
tal(1,m))/1000)+(1.0813);
    else

cost_day(m)=0.0371*(sum_off_peak_energy(1,m)/1000)+0.0486*((sum_peak_to
tal(1,m))/1000)+(1.0813);
    end
    if peak_demand_june<3000  ||peak_demand_june==3000;
        peak_demand_charge=((peak_demand_june)/1000)*8.03;
    elseif peak_demand_june<10000 ||peak_demand_june==10000;
        peak_demand_charge=3*8.03+((peak_demand_june-3000)/1000)*14.63;
    else
        peak_demand_charge=3*8.03+7*14.63+((peak_demand_june-
10000)/1000)*27.77;
    end
    june_bill=june_bill+cost_day(m);
end
june_bill_final=june_bill+peak_demand_charge;

%july_bill
peak_demand_july=0;
for m=183:212
    if peak_demand_value(1,m)>peak_demand_value(1,m-1);
        peak_demand_july=peak_demand_value(1,m);
    else
        peak_demand_july=peak_demand_value(1,m-1);
       peak_demand_value(1,m)=peak_demand_value(1,m-1);
```

```
    end
end
july_bill=0;
july_bill_final=0;
peak_demand_charge=0;
for m=182:212
    x=m/7;
    if x==5||x==6

cost_day(m)=0.0423*(sum_off_peak_energy(1,m)/1000)+0.0423*((sum_peak_to
tal(1,m))/1000)+(1.0813);
    else

cost_day(m)=0.0423*(sum_off_peak_energy(1,m)/1000)+0.0633*((sum_peak_to
tal(1,m))/1000)+(1.0813);
    end
    if peak_demand_july<3000  ||peak_demand_july==3000;
        peak_demand_charge=((peak_demand_july)/1000)*9.59;
    elseif peak_demand_july<10000 ||peak_demand_july==10000;
        peak_demand_charge=3*9.59+((peak_demand_july-3000)/1000)*17.82;
    else
        peak_demand_charge=3*9.59+7*17.82+((peak_demand_july-
10000)/1000)*34.19;
    end
    july_bill=july_bill+cost_day(m);
end
july_bill_final=july_bill+peak_demand_charge;

%August_bill
peak_demand_aug=0;
for m=214:243
    if peak_demand_value(1,m)>peak_demand_value(1,m-1);
        peak_demand_aug=peak_demand_value(1,m);
    else
        peak_demand_aug=peak_demand_value(1,m-1);
       peak_demand_value(1,m)=peak_demand_value(1,m-1);

    end
end
august_bill=0;
august_bill_final=0;
peak_demand_charge=0;
for m=213:243
    x=m/7;
    if x==5||x==6

cost_day(m)=0.0423*(sum_off_peak_energy(1,m)/1000)+0.0423*((sum_peak_to
tal(1,m))/1000)+(1.0813);
    else

cost_day(m)=0.0423*(sum_off_peak_energy(1,m)/1000)+0.0633*((sum_peak_to
tal(1,m))/1000)+(1.0813);
    end
    if peak_demand_aug<3000  ||peak_demand_aug==3000;
```

```
        peak_demand_charge=((peak_demand_aug)/1000)*9.59;
    elseif peak_demand_aug<10000 ||peak_demand_aug==10000;
        peak_demand_charge=3*9.59+((peak_demand_aug-3000)/1000)*17.82;
    else
        peak_demand_charge=3*9.59+7*17.82+((peak_demand_aug-
10000)/1000)*34.19;
    end
    august_bill=august_bill+cost_day(m);
end
august_bill_final=august_bill+peak_demand_charge;

%september_bill
peak_demand_sep=0;
for m=245:273
    if peak_demand_value(1,m)>peak_demand_value(1,m-1);
        peak_demand_sep=peak_demand_value(1,m);
    else
        peak_demand_sep=peak_demand_value(1,m-1);
       peak_demand_value(1,m)=peak_demand_value(1,m-1);

    end
end
september_bill=0;
september_bill_final=0;
peak_demand_charge=0;
for m=244:273
    x=m/7;
    if x==5||x==6

cost_day(m)=0.0390*(sum_off_peak_energy(1,m)/1000)+0.0390*((sum_peak_to
tal(1,m))/1000)+(1.0813);
    else

cost_day(m)=0.0390*(sum_off_peak_energy(1,m)/1000)+0.0430*((sum_peak_to
tal(1,m))/1000)+(1.0813);
    end
    if peak_demand_sep<3000  ||peak_demand_sep==3000;
        peak_demand_charge=((peak_demand_sep)/1000)*3.55;
    elseif peak_demand_sep<10000 ||peak_demand_sep==10000;
        peak_demand_charge=3*3.55+((peak_demand_sep-3000)/1000)*5.68;
    else
        peak_demand_charge=3*3.55+7*5.68+((peak_demand_sep-
10000)/1000)*9.74;
    end
    september_bill=september_bill+cost_day(m);
end
september_bill_final=september_bill+peak_demand_charge;

%october_bill
peak_demand_oct=0;
for m=275:304
    if peak_demand_value(1,m)>peak_demand_value(1,m-1);
        peak_demand_oct=peak_demand_value(1,m);
    else
```

```
            peak_demand_oct=peak_demand_value(1,m-1);
           peak_demand_value(1,m)=peak_demand_value(1,m-1);

        end
    end
    october_bill=0;
    october_bill_final=0;
    peak_demand_charge=0;
    for m=274:304
        x=m/7;
        if x==5||x==6

    cost_day(m)=0.0390*(sum_off_peak_energy(1,m)/1000)+0.0390*((sum_peak_to
    tal(1,m))/1000)+(1.0813);
        else

    cost_day(m)=0.0390*(sum_off_peak_energy(1,m)/1000)+0.0430*((sum_peak_to
    tal(1,m))/1000)+(1.0813);
        end
        if peak_demand_oct<3000  ||peak_demand_oct==3000;
            peak_demand_charge=((peak_demand_oct)/1000)*3.55;
        elseif peak_demand_oct<10000 ||peak_demand_oct==10000;
            peak_demand_charge=3*3.55+((peak_demand_oct-3000)/1000)*5.68;
        else
            peak_demand_charge=3*3.55+7*5.68+((peak_demand_oct-
    10000)/1000)*9.74;
        end
        october_bill=october_bill+cost_day(m);
    end
    october_bill_final=october_bill+peak_demand_charge;

    %november_bill
    peak_demand_nov=0;
    for m=306:334
        if peak_demand_value(1,m)>peak_demand_value(1,m-1);
            peak_demand_nov=peak_demand_value(1,m);
        else
            peak_demand_nov=peak_demand_value(1,m-1);
           peak_demand_value(1,m)=peak_demand_value(1,m-1);

        end
    end
    november_bill=0;
    november_bill_final=0;
    peak_demand_charge=0;
    for m=305:334
        x=m/7;
        if x==5||x==6

    cost_day(m)=0.0390*(sum_off_peak_energy(1,m)/1000)+0.0390*((sum_peak_to
    tal(1,m))/1000)+(1.0813);
        else
    cost_day(m)=0.0390*(sum_off_peak_energy(1,m)/1000)+0.0430*((sum_peak_to
    tal(1,m))/1000)+(1.0813);
```

```
    end
    if peak_demand_nov<3000  ||peak_demand_nov==3000;
        peak_demand_charge=((peak_demand_nov)/1000)*3.55;
    elseif peak_demand_nov<10000 ||peak_demand_nov==10000;
        peak_demand_charge=3*3.55+((peak_demand_nov-3000)/1000)*5.68;
    else
        peak_demand_charge=3*3.55+7*5.68+((peak_demand_nov-
10000)/1000)*9.74;
    end
    november_bill=november_bill+cost_day(m);
end
november_bill_final=november_bill+peak_demand_charge;

%december_bill
peak_demand_dec=0;
for m=336:365
    if peak_demand_value(1,m)>peak_demand_value(1,m-1);
        peak_demand_dec=peak_demand_value(1,m);
    else
        peak_demand_dec=peak_demand_value(1,m-1);
       peak_demand_value(1,m)=peak_demand_value(1,m-1);

    end
end
december_bill=0;
december_bill_final=0;
peak_demand_charge=0;
for m=335:365
    x=m/7;
    if x==5||x==6

cost_day(m)=0.0390*(sum_off_peak_energy(1,m)/1000)+0.0390*((sum_peak_to
tal(1,m))/1000)+(1.0813);
    else

cost_day(m)=0.0430*(sum_off_peak_energy(1,m)/1000)+0.0430*((sum_peak_to
tal(1,m))/1000)+(1.0813);
    end
    if peak_demand_dec<3000  ||peak_demand_dec==3000;
        peak_demand_charge=((peak_demand_dec)/1000)*3.55;
    elseif peak_demand_dec<10000 ||peak_demand_dec==10000;
        peak_demand_charge=3*3.55+((peak_demand_dec-3000)/1000)*5.68;
    else
        peak_demand_charge=3*3.55+7*5.68+((peak_demand_dec-
10000)/1000)*9.74;
    end
    december_bill=december_bill+cost_day(m);
end
december_bill_final=december_bill+peak_demand_charge;
Total_bill(q,e)=december_bill_final+november_bill_final+october_bill_fi
nal+september_bill_final+august_bill_final+july_bill_final+june_bill_fi
nal+may_bill_final+april_bill_final+march_bill_final+febuary_bill_final
+ january_bill_final;
```


`

```
%optimization parameters
system_self_consumption(q,e)=((Total_PV_load+Energy_charging(q,e))/PV_t
otal);
system_self_sufficeincy(q,e)=((Total_PV_load-
Energy_discharging(q,e))/Load_total);
storage_cycles(q,e)=((-Energy_discharging(q,e))/(e*1000*0.6));
npv=25;
tc(q,e)=5000/storage_cycles(q,e);
nb=min(tc,20);

%System_cost
PV_cost(q,e)=2.513*q*1000+(2.02*q*1000*0.02*npv);
if nb(q,e)>=20
Battery_cost(q,e)=350*e+350*e*0.02*nb(q,e);
elseif  nb(q,e)<=20 &&  nb(q,e)>=10
 Battery_cost(q,e)=2*(350*e+350*e*0.02*nb(q,e));
elseif nb(q,e)<=10 &&  nb(q,e)>=6.6
    Battery_cost(q,e)=3*(350*e+350*e*0.02*nb(q,e));
elseif nb(q,e)<=6.6 &&  nb(q,e)>=5
    Battery_cost(q,e)=4*(350*e+350*e*0.02*nb(q,e));
else
    Battery_cost(q,e)=5*(350*e+350*e*0.02*nb(q,e));
end
Total_cost_system(q,e)=PV_cost(q,e)+Battery_cost(q,e);

Feed_in_tarrif(q,e)=Electricity_price*(PV_total/1000)*(1-
system_self_consumption(q,e));

savings_bill(q,e)=(42010)-((Total_bill(q,e))*npv);

amount_year_end=0;

Total_savings_system(q,e)=savings_bill(q,e)+(Feed_in_tarrif(q,e)*npv);
principal(q,e)=(Total_savings_system(q,e))/npv;
for i=1:25
    amount_year_end=principal(q,e)*(1.04);
    principal(q,e)=amount_year_end+(Total_savings_system(q,e))/npv;
end

pay_back_period(q,e)=((Total_cost_system(q,e)*npv)/principal(q,e));
profit(q,e)=principal(q,e)-Total_cost_system(q,e);
end
end
```

APPENDIX C

CODE: LOAD SELECTION

`

```
filename= 'test';
Load=xlsread(filename,'A:A');
Temp=xlsread(filename,'B:B');
[ml,nl]=size(Load);
[mt,nt]=size(Temp);
winter_load=zeros(4344,1);
winter_temp=zeros(4344,1);
summer_load=zeros(2928,1);
summer_temp=zeros(2928,1);
summer_peak_load=zeros(1488,1);
summer_peak_temp=zeros(1488,1);
x=1;y=1;z=1;

    for j=1:2880
       winter_load(x,1)=Load(j,1);
       winter_temp(x,1)=Temp(j,1);
       x=x+1;
    end

    for j=2881:4344
        summer_load(y,1)=Load(j,1);
        summer_temp(y,1)=Temp(j,1);
        y=y+1;
    end

    for j=4345:5832
        summer_peak_load(z,1)=Load(j,1);
         summer_peak_temp(z,1)=Temp(j,1);
        z=z+1;
    end

    for j=5833:7296
        summer_load(y,1)=Load(j,1);
        summer_temp(y,1)=Temp(j,1);
        y=y+1;
    end

    for j=7297:8760
        winter_load (x,1)=Load(j,1);
        winter_temp(x,1)=Temp(j,1);
       x=x+1;
    end


    winter_load_day=zeros(24,181);
    winter_temp_day=zeros(24,181);
    for i=1:181
         x=1;
        for j=1+(24*(i-1)):24+(24*(i-1))
            winter_load_day(x,i)=winter_load(j,1);
            winter_temp_day(x,i)=winter_temp(j,1);
            x=x+1;
        end
    end
```

```
    winter_load_day_avg=mean(winter_load_day,2);
     summer_load_day=zeros(24,122);
     summer_temp_day=zeros(24,122);
     for i=1:122
          y=1;
        for j=1+(24*(i-1)):24+(24*(i-1))
            summer_load_day(y,i)=summer_load(j,1);
            summer_temp_day(y,i)=summer_temp(j,1);
             y=y+1;
        end
     end
     summer_load_day_avg=mean(summer_load_day,2);
      summer_peak_temp_day=zeros(24,62);
    summer_peak_load_day=zeros(24,62);
     for i=1:62
          z=1;
        for j=1+(24*(i-1)):24+(24*(i-1))
            summer_peak_load_day(z,i)=summer_peak_load(j,1);
            summer_peak_temp_day(z,i)=summer_peak_temp(j,1);
            z=z+1;
        end
     end
     summer_peak_load_avg=mean(summer_peak_load_day,2);

prompt='what is the new max temp\n';
Max_Temp_new=input(prompt);
prompt='Select season wintwer=1, summer=2,summer_peak=3\n';
season=input(prompt);
if season==1
    load_set=winter_load_day;
    temp_set=winter_temp_day;
elseif season==2
    load_set=summer_load_day;
    temp_set=summer_temp_day;
elseif season==3
    load_set=summer_peak_load_day;
    temp_set=summer_peak_temp_day;
else
    fprintf('invalid selection\n');
end
[m,n]=size(load_set);
[mt1,nt1]=size(temp_set);
max_temp_set=max(temp_set);
e=1;
for i=1:n
    if max_temp_set(1,i)>=(Max_Temp_new-0.25) &&
max_temp_set(1,i)<=(Max_Temp_new+0.25)
    for j= 1:24
        load_selected(j,e)=load_set(j,i);
    end
     e=e+1;
    end
end
if season==1
```

```
    for i=1:(e-1)
        for j=1:24
            load_deviation(j,i)=abs(load_selected(j,i)-
winter_load_day_avg(j,1));
        end
    end
elseif season==2
    for i=1:(e-1)
        for j=1:24
            load_deviation(j,i)=abs(load_selected(j,i)-
summer_load_day_avg(j,1));
        end
    end
else
    for i=1:(e-1)
        for j=1:24
            load_deviation(j,i)=abs(load_selected(j,i)-
summer_peak_load_day_avg(j,1));
        end
    end
end

   for j=1:24
       Load_deviation_hour_sum=0;
       for i=1:(e-1)

Load_deviation_hour_sum=Load_deviation_hour_sum+load_deviation(j,i);
       end
       Load_deviation_sum(j,1)= Load_deviation_hour_sum;
   end

   if e>2
   division_coefficient=e-2;
  for j=1:24
      for i=1:(e-1)
          weight_load(j,i)=((1-(load_deviation(j,i)/
Load_deviation_sum(j,1)))/division_coefficient);
      end
  end
  else
       division_coefficient=1;
       for j=1:24
      for i=1:(e-1)
          weight_load(j,i)=(((load_deviation(j,i)/
Load_deviation_sum(j,1)))/division_coefficient);
      end
       end
   end

  for j=1:24
      Load_new_calculation=0;
      for i=1:(e-1)
          Load_new_calculation=
Load_new_calculation+(weight_load(j,i)*load_selected(j,i));
```


`

```
    end
    Load_new_final(j,1)=Load_new_calculation;
end
```

CODE: OPERATING STRATEGY

```
filename='sample';
PV=xlsread(filename,'A:A');
Load=xlsread(filename,'B:B');
[n,m]=size(PV);
days=n/24;
PV_data=zeros(24,days);
Load_data=zeros(24,days);
PV_Load_difference=zeros(24,days);
PV_Load_difference_new=zeros(24,days);
PV_sum_off_peak=zeros(1,days);
battery_energy=zeros(24,days);
Excess_Load=zeros(24,days);
Dispatch_energy=zeros(24,days);
on_peak_energy=zeros(24,days);
for i=1:days
    w=1;
    for j=(1+(24*(i-1))):(24+(24*(i-1)))
    PV_data(w,i)=PV(j);
    if Load(j)<6000
    Load_data(w,i)=Load(j);
    else
        Load_data(w,i)=6000;
        Excess_Load(w,i)=Load(j)-6000;
    end
    w=w+1;
    end
end
for i=1:days
    for w=1:13
        PV_sum_off_peak(1,i)=PV_sum_off_peak(1,i)+PV_data(w,i);
    end
end
B=min(PV_sum_off_peak);
PV_data_min=min(PV_data,[],2);
minimum_off_peak_energy=(100-(((min(PV_sum_off_peak))/(4*16000))*100));
formatspec='off_peak_battery_charged to %1.0f ';
value=minimum_off_peak_energy*(16000/100);
formatspe='off_peak_battery_charged to %1.0f ';
fprintf(formatspe,value);
for j=1:days
        for w=1:24
            PV_data_min(w,j)= PV_data_min(w,1);
        end
end
 PV_Load_difference_sum=zeros(1,days);
 on_peak_limit=zeros(1,days);

for i=1:days
    for w=13:20
        PV_Load_difference(w,i)=PV_data_min(w,j)-Load_data(w,i);
        if PV_Load_difference(w,i)<0
```


`

```
            PV_Load_difference_sum(1,i)=PV_Load_difference_sum(1,i)-
PV_Load_difference(w,i);
        end
    end
    if PV_Load_difference_sum(1,i)<16000
        on_peak_limit(1,i)=0;
    else
        on_peak_limit(1,i)=((PV_Load_difference_sum(1,i)-16000)/8);
    end
end
threshold=max(on_peak_limit);

PV_Load_difference_sum_new=zeros(1,days);

for i=1:days
    for w=13:20
         PV_Load_difference_new(w,i)=PV_data_min(w,j)+threshold-
Load_data(w,i);
         if PV_Load_difference_new(w,i)<0

PV_Load_difference_sum_new(1,i)=PV_Load_difference_sum_new(1,i)-
PV_Load_difference_new(w,i);
         end
    end
end

for i=1:days
for w=13:20
    battery_energy(13,i)=16000;
    if PV_Load_difference_sum_new(1,i)<=16001
        if PV_Load_difference_new(w,i)<0
            battery_energy(w+1,i)=
battery_energy(w,i)+PV_Load_difference_new(w,i);
            PV_Load_difference_new(w,i)=0;
        else
            battery_energy(w+1,i)= battery_energy(w,i);
            PV_Load_difference_new(w,i)=0;
        end
    else
        PV_Load_difference_new(w,i)=1;
    end
end
end

for i=1:days
    for w=13:20

on_peak_energy(w,i)=threshold+Excess_Load(w,i)+PV_Load_difference_new(w
,i);
    end
    peak_demand_day=max(on_peak_energy);
end
peak_demand=max(peak_demand_day);
```

APPENDIX D

CODE: STATISTICAL ANALYSIS

`

```
filename= 'Febdata';
PV_Data=xlsread(filename,'B:B');
Bat_SOC=xlsread(filename,'C:C');
Bat_Ene=xlsread(filename,'D:D');
Load_Ene=xlsread(filename,'E:E');
Site_Ene=xlsread(filename,'F:F');
[m,n]=size(PV_Data);
c=24;
x=m/c;
PV_Data_day=zeros(c,1);
Bat_SOC_day=zeros(c,1);
Bat_Ene_day=zeros(c,1);
Load_Ene_day=zeros(c,1);
Site_Ene_day=zeros(c,1);
for i=1:x
    w=1;
     for j=(1+(c*(i-1))):(c*i)
         PV_Data_day(w,i)=PV_Data(j);
         Bat_SOC_day(w,i)=Bat_SOC(j);
         Bat_Ene_day(w,i)=Bat_Ene(j);
         Load_Ene_day(w,i)=Load_Ene(j);
         Site_Ene_day(w,i)=Site_Ene(j);
         w=w+1;
     end
end


%Solar_Data
prompt = 'Do you want Solar Data Analysis? Y/N [Y]: ';
str = input(prompt,'s');
if str=='Y'
    Sum_PV_Data_day=zeros(1,c);
     for i=1:x
         Sum_A=0;
         for j=1:c
             Sum_A=Sum_A+PV_Data_day(j,i);
         end
         Sum_PV_Data_day(1,i)=Sum_A;
     end
     Max_PV_Data_day=max(Sum_PV_Data_day);
     Min_PV_Data_day=min(Sum_PV_Data_day);
     Avg_PV_Data_day=mean(Sum_PV_Data_day);
fprintf('Maximum PV energy in a day for selected data is
%d\n',Max_PV_Data_day);
fprintf('Minimum PV energy in a day for selected data is
%d\n',Min_PV_Data_day);
fprintf('Average PV energy in a day for selected data is
%d\n',Avg_PV_Data_day);
end

% SOC_Data
prompt = 'Do you want SOC Data Analysis? Y/N [Y]: ';
str = input(prompt,'s');
if str=='Y'
```

```
    Max_Bat_SOC_day=max(Bat_SOC_day);
    Min_Bat_SOC_day=min(Bat_SOC_day);
        v=0;
        v1=0;
    for i=1:x
        if Max_Bat_SOC_day(1,i)>=90
            v=v+1;
        end
        if Min_Bat_SOC_day(1,i)<=20
            v1=v1+1;
        end
    end
     P_max=((v/x)*100);
     P_min=((v1/x)*100);
fprintf('Number of days in the selected data battery is completely
charged is %d and percentage is %d\n',v,P_max);
fprintf('Number of days in the selected data battery is completely
discharged is %d and percentage is %d\n',v1,P_min);
end

% Battery_Energy
prompt = 'Do you want Battery Energy Data Analysis? Y/N [Y]: ';
str = input(prompt,'s');
if str=='Y'
    Bat_Ene_day_onpeak=zeros(24,x);
    Bat_Ene_day_toload=zeros(1,x);
     Bat_Ene_day_togrid=zeros(1,x);
    for i=1:x
        w=13;
    for j=(13+(24*(i-1))):(20+(24*(i-1)))
        Bat_Ene_day_onpeak(w,i)=Bat_Ene_day(j);
        if Bat_Ene_day_onpeak(w,i)>0
            Bat_Ene_day_togrid(1,i)=
Bat_Ene_day_togrid(1,i)+Bat_Ene_day_onpeak(w,i);
        else
            Bat_Ene_day_toload(1,i)=
Bat_Ene_day_toload(1,i)+Bat_Ene_day_onpeak(w,i);
        end
        w=w+1;
    end
    end
   Max_Bat_Ene_day=max(Bat_Ene_day_onpeak);
   Min_Bat_Ene_day=min(Bat_Ene_day_onpeak);
   Max_Bat_Ene_day_togrid=max(Bat_Ene_day_togrid);
   Min_Bat_Ene_day_togrid=min(Bat_Ene_day_togrid);
   Avg_Bat_Ene_day_togrid=mean(Bat_Ene_day_togrid);
   Max_Bat_Ene_day_toload=min(Bat_Ene_day_toload);
   Min_Bat_Ene_day_toload=max(Bat_Ene_day_toload);
   Avg_Bat_Ene_day_toload=mean(Bat_Ene_day_toload);
fprintf('Maximum amount of Battery Energy given to grid from selected
data is %d\n',Max_Bat_Ene_day_togrid);
fprintf('Minimum amount of Battery Energy given to grid from selected
data is %d\n',Min_Bat_Ene_day_togrid);
```

```
fprintf('Average amount of Battery Energy given to grid from selected
data is %d\n',Avg_Bat_Ene_day_togrid);
fprintf('Maximum amount of Battery Energy given to load from slected
data is %d\n',Max_Bat_Ene_day_toload);
fprintf('Minimum amount of Battery Energy given to load from slected
data is %d\n',Min_Bat_Ene_day_toload);
fprintf('Average amount of Battery Energy given to load from slected
data is %d\n',Avg_Bat_Ene_day_toload);
prompt = 'Select a day with in range where you want to see the minimum
and maximum discharge of the battery ';
I=input(prompt);
fprintf('Maximum Charge rate in a selected day is
%d\n',Max_Bat_Ene_day(1,I));
fprintf('Maximum Discharge rate in a selected day is
%d\n',Min_Bat_Ene_day(1,I));
end

%Load_Energy
prompt = 'Do you want Load Energy Analysis? Y/N [Y]: ';
str = input(prompt,'s');
if str=='Y'
    Sum_Load_Ene_day=zeros(1,x);
     for i=1:x
         Sum_A=0;
         for j=1:c
             Sum_A=Sum_A+Load_Ene_day(j,i);
         end
         Sum_Load_Ene_day(1,i)=Sum_A;
     end
     Max_Load_Ene_day=max(Sum_Load_Ene_day);
     Min_Load_Ene_day=min(Sum_Load_Ene_day);
     Avg_Load_Ene_day=mean(Sum_Load_Ene_day);
fprintf('Maximum Load Energy in a day for selected data is
%d\n',Max_Load_Ene_day);
fprintf('Minimum Load Energy in a day for selected data is
%d\n',Min_Load_Ene_day);
fprintf('Average Load Energy in a day for selected data is
%d\n',Avg_Load_Ene_day);
end


%Site_Demand
prompt = 'Do you want Site_Demand Data Analysis? Y/N [Y]: ';
str = input(prompt,'s');
if str=='Y'
    Site_Ene_day_tgrid=zeros(c,x);
    Site_Ene_day_fgrid=zeros(c,x);
    for i=1:x
        for j=13:20
            if Site_Ene_day(j,i)<=0
                Site_Ene_day_tgrid(j,i)=Site_Ene_day(j,i);
                Site_Ene_day_fgrid(j,i)=0;
            else
                Site_Ene_day_tgrid(j,i)=0;
```

```
                Site_Ene_day_fgrid(j,i)=Site_Ene_day(j,i);
            end
        end
    end
    Site_Ene_day_tgrid_day=sum(Site_Ene_day_tgrid);
    Site_Ene_day_fgrid_day=sum(Site_Ene_day_fgrid);
    Max_Site_Ene_day_tgrid=min(Site_Ene_day_tgrid);
    Max_Site_Ene_day_fgrid=max(Site_Ene_day_fgrid);
    Max_Site_Ene_day_tgrid_day=max(Site_Ene_day_tgrid_day);
    Min_Site_Ene_day_tgrid_day=min(Site_Ene_day_tgrid_day);
    Avg_Site_Ene_day_tgrid_day=mean(Site_Ene_day_tgrid_day);
    Max_Site_Ene_day_fgrid_day=max(Site_Ene_day_fgrid_day);
    Min_Site_Ene_day_fgrid_day=min(Site_Ene_day_fgrid_day);
    Avg_Site_Ene_day_fgrid_day=mean(Site_Ene_day_fgrid_day);
fprintf('Maximum amount of Grid Energy given to Load during on peak
hours %d\n',Max_Site_Ene_day_fgrid_day);
fprintf('Minimum amount of Grid Energy given to Load during on peak
hours %d\n', Min_Site_Ene_day_fgrid_day);
fprintf('Average amount of Grid Energy given to Load during on peak
hours %d\n',Avg_Site_Ene_day_fgrid_day);
fprintf('Maximum amount of Energy given to Grid during on peak hours
%d\n',Max_Site_Ene_day_tgrid_day);
fprintf('Minimum amount of Energy given to Grid during on peak hours
%d\n',Min_Site_Ene_day_tgrid_day);
fprintf('Average amount of Energy given to Grid during on peak hours
%d\n',Avg_Site_Ene_day_tgrid_day);

     v=0;
    for i=1:x
        if Max_Site_Ene_day_fgrid(1,i)<=0
            v=v+1;
        end
    end
    P_self=((v/x)*100);
prompt = 'Select a day with in range where you want to see the maximum
to and from grid energy ';
I=input(prompt);
fprintf('Maximum to grid energy in a selected day is
%d\n',Max_Site_Ene_day_tgrid(1,I));
fprintf('Maximum from grid energy in a selected day is
%d\n',Max_Site_Ene_day_fgrid(1,I));
fprintf('Number of days battery is self sufficeient is %d and percetage
is %d\n',v,P_self);
end


%Graph Plot
prompt = 'Do you want Graphical Analysis? Y/N [Y]: ';
str = input(prompt,'s');
if str=='Y'
    [Max_Site_ene,D]=max(Max_Site_Ene_day_fgrid);
    Y1=zeros(24,1);
    Y2=zeros(24,1);
    Y3=zeros(24,1);
```

```
    Y4=zeros(24,1);
    Y5=zeros(24,1);
 for j=1:24
     Y1(j,1)= PV_Data_day(j,D);
     Y2(j,1)=Bat_SOC_day(j,D);
     Y3(j,1)=Bat_Ene_day(j,D);
     Y4(j,1)=Load_Ene_day(j,D);
     Y5(j,1)=Site_Ene_day(j,D);
 end
 figure
 plot(Y1,'-o')
 title=('Max_Site_Demand');
 hold on
 plot(Y2,'-o')
 hold on
 plot(Y3,'-o')
 hold on
 plot(Y4,'-o')
 hold on
 plot(Y5,'-o')
 hold off
fprintf('Maximum site demand from selected day is %d and the day is
%d\n',Max_Site_ene,D);
prompt = 'Do you want another graph for some other day? Y/N [Y]: ';
str = input(prompt,'s');
if str=='Y'
prompt = 'Select a day with in range where you want to see the graph ';
I=input(prompt);
for j=1:24
     Y1(j,1)= PV_Data_day(j,I);
     Y2(j,1)=Bat_SOC_day(j,I);
     Y3(j,1)=Bat_Ene_day(j,I);
     Y4(j,1)=Load_Ene_day(j,I);
     Y5(j,1)=Site_Ene_day(j,I);
 end
 figure
 plot(Y1,'-o')
 title=('Other Day');
 hold on
 plot(Y2,'-o')
 hold on
 plot(Y3,'-o')
 hold on
 plot(Y4,'-o')
 hold on
 plot(Y5,'-o')
 hold off

end
end
```


`