

12-2014

# Design of System Level Control for a Residential Power Router

Fahad Hossain

*University of Arkansas, Fayetteville*

Follow this and additional works at: <http://scholarworks.uark.edu/etd>

 Part of the [Controls and Control Theory Commons](#), and the [Power and Energy Commons](#)

---

## Recommended Citation

Hossain, Fahad, "Design of System Level Control for a Residential Power Router" (2014). *Theses and Dissertations*. 2027.  
<http://scholarworks.uark.edu/etd/2027>

This Thesis is brought to you for free and open access by ScholarWorks@UARK. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of ScholarWorks@UARK. For more information, please contact [scholar@uark.edu](mailto:scholar@uark.edu), [ccmiddle@uark.edu](mailto:ccmiddle@uark.edu).

## Design of System Level Control for a Residential Power Router

Design of System Level Control for a Residential Power Router

A thesis submitted in partial fulfillment  
of the requirements for the degree of  
Master of Science in Electrical Engineering

by

Fahad Hossain  
University of Dhaka  
Bachelor of Science in Applied Physics, Electronics and Communication Engineering, 2012

December 2014  
University of Arkansas

This thesis is approved for recommendation to the Graduate Council

---

Dr. H. Alan Mantooth  
Thesis Director

---

Dr. Simon Ang  
Committee Member

---

Dr. Michael Glover  
Committee Member

## **ABSTRACT**

This thesis presents the overall power flow and load management system for a 2 kW residential power router. The objective is to satisfy the residential load demand with cost savings by managing the energy sources including PV, battery, generator, and grid. The power flow management system algorithm is developed both for grid-connected mode and island mode. This algorithm involves the battery state of charge estimation, available PV power, load demand and electricity grid pricing. The residential load management algorithm is also developed using PV power prediction and load power prediction acquired for a day. A Labview GUI is also developed for this load management.

## **ACKNOWLEDGEMENTS**

First of all I would like to thank my advisor, Dr. Alan Mantooth for his guidance and encouragement towards my graduate program. His mentorship helped me a lot to progress in my graduate program. I would also like to thank my thesis committee members, Dr. Simon Ang and Dr. Michael Glover for their valuable participation. I would like to thank the faculty and staff of the Electrical Engineering Department for their cooperation.

I would like to thank my project team members, Yuzhi Zhang, Janvier Umuhiza and Haoyan Liu with whom I worked in this project. I would like to thank Chris Farnell for successfully managing our lab group. I would also like to thank Dr. Roger Dougal from University of South Carolina for his advice and guidance. I would like to acknowledge Hang Xu and other research students from University of South Carolina for their hard work on this project.

I would also like to thank my friends from Dr. Mantooth's lab, Tavis Clemmer, Yusi Liu, Sonia Perez, Stacy Brooks, Shamim Ahmed, Thomas White, Joe Moquin and many more for their continuous help and support.

## **DEDICATION**

This thesis is dedicated to my wife, Shaila Amin Bhuyan, my Father, Md. Shahadat Hossain Bhuiyan, my Mother, Ferdousi Akter, and my siblings Munia Sunjid, Tania Sanjid and Shahriar Hossain.

## TABLE OF CONTENTS

Chapter 1	INTRODUCTION AND OBJECTIVES .....	1
1.1	Need for System Level Energy Management .....	1
1.2	Formulation of System Level Energy Management .....	2
1.3	Smart Green Power Node.....	3
1.4	Thesis Organization.....	3
Chapter 2	THEORETICAL BACKGROUND AND RELATED CONCEPTS .....	4
2.1	System level Control Overview .....	4
2.2	Lead Acid Battery Construction.....	4
2.3	Lead Acid Battery Models .....	5
2.3.1	Simple Battery Model.....	5
2.3.2	Modified Simple Battery Model .....	6
2.3.3	Advanced Simple Battery Model.....	7
2.3.4	Thevenin Battery Model .....	8
2.3.5	Third Order Model.....	8
2.4	PV Prediction Overview.....	12
2.4.1	Clear Sky Radiation Model Overview .....	12
2.5	Load Prediction Overview .....	14
2.5.1	Autoregressive Model (AR).....	14
2.5.2	The Moving Average Process (MA).....	15

2.5.3	The Autoregressive Moving Average Process (ARMA) .....	15
2.5.4	The Autoregressive Integrated Moving Average Process (ARIMA) .....	16
Chapter 3	POWER FLOW MANAGEMENT .....	17
3.1	Power Flow Management Structure .....	17
3.2	Battery SOC Estimation .....	18
3.2.1	Discharge Current Method .....	18
3.2.2	Battery Voltage Method .....	22
3.3	Battery Charge and Discharge Control .....	24
3.3.1	Island Mode Control .....	25
3.3.2	DSP Implementation of Island Mode Control .....	30
3.3.3	Grid-connected Mode .....	31
3.3.4	DSP implementation of Grid-connected Mode .....	40
Chapter 4	LOAD MANAGEMENT .....	42
4.1	Load Management Structure .....	42
4.2	PV Power Prediction .....	43
4.3	Load Prediction .....	52
4.3.1	Electrical Load Prediction Using Auto-regression Model .....	52
4.4	DSP Implementation of PV & Load Prediction .....	58
4.5	Types of Load Management .....	60
4.5.1	Automatic Turn On/Off Load Management .....	60



4.5.2	Labview GUI Demonstration for Load Management.....	63
4.5.3	Automatic Load Management with Temperature Control.....	65
Chapter 5	CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK.....	70
5.1	Concluding Remarks.....	70
5.2	Recommendations for Future Work.....	71
	REFERENCES .....	73
	APPENDIX.....	75
	Appendix A: DSP Code for Power Flow and Load Management .....	75

## LIST OF FIGURES

Fig. 1.1.1. Average electricity price in different sectors from 2012 to 2015 [1].	1
Fig. 2.3.1. Simple battery model circuit diagram [7].	6
Fig. 2.3.2. Modified battery model circuit diagram [8].	7
Fig. 2.3.3. Advanced simple battery model circuit diagram [9].	7
Fig. 2.3.4. Circuit diagram for Thevenin battery model [11].	8
Fig. 2.3.5. Third order battery model circuit diagram [7].	11
Fig. 3.1.1. Block diagram for power flow management structure.	17
Fig. 3.2.1. Top level block diagram for battery SOC estimation using discharge current.	18
Fig. 3.2.2. Simulink <sup>TM</sup> block diagram with capacity model, voltage model and thermal model.	19
Fig. 3.2.3. Simulation results for battery SOC estimation when the battery is charging.	21
Fig. 3.2.4. Simulation results for battery SOC estimation when the battery is discharging.	22
Fig. 3.2.5. Simulation results for battery SOC estimation using voltage method.	24
Fig. 3.3.1. Flow chart for power flow management in island mode.	25
Fig. 3.3.2. Simulation results for island mode.	26
Fig. 3.3.3. State diagram for island mode.	29
Fig. 3.3.4. Power flow and generator status waveform in island mode.	29
Fig. 3.3.5. Output results for DSP implementation of island mode control.	30
Fig. 3.3.6. Flow chart for power flow management in grid-connected mode.	32
Fig. 3.3.7. Simulation results for grid-connected mode.	33
Fig. 3.3.8. Power flow, daily grid power usage and electricity cost in grid-connected mode.	36
Fig. 3.3.9. State diagram for grid connected mode.	38
Fig. 3.3.10. Average monthly grid power usage and electricity cost comparison.	39

Fig. 3.3.11. Average monthly electricity cost comparison. ....	39
Fig. 3.3.12. Average monthly cost savings using SGPN. ....	40
Fig. 3.3.13 Output results for DSP implementation of grid-connected mode control. ....	41
Fig. 4.1.1. Block diagram for load management structure. ....	42
Fig. 4.2.1. PV power prediction model block diagram. ....	44
Fig. 4.2.2. Twenty-four hour solar radiation prediction for the month of February. ....	48
Fig. 4.2.3. Twenty-four hour solar radiation prediction for the month of July. ....	48
Fig. 4.2.4. Twenty-four hour solar radiation prediction for the month of November. ....	49
Fig. 4.2.5. Twenty-four hour PV power prediction for the month of February. ....	51
Fig. 4.2.6. Twenty-four hour PV power prediction for the month of July. ....	51
Fig. 4.2.7. Twenty-four hour PV power prediction for the month of November. ....	52
Fig. 4.3.1. Twenty-four hour average historical load demand for the month of February [20]. ....	53
Fig. 4.3.2. Twenty-four hour average historical load demand for the month of July [20]. ....	54
Fig. 4.3.3. Twenty-four hour average historical load demand for the month of November [20]. ....	55
Fig. 4.3.4. Twenty-four hour load demand prediction for the month of February. ....	56
Fig. 4.3.5. Twenty-four hour load demand prediction for the month of July. ....	57
Fig. 4.3.6. Twenty-four hour load demand prediction for the month of November. ....	57
Fig. 4.4.1. Output results for DSP implementation of PV prediction and load prediction. ....	59
Fig. 4.5.1. Simulation results for automatic turn on/off load management for the month of February. ....	60
Fig. 4.5.2. Simulation results for automatic turn on/off load management during the month of July. ....	61

Fig. 4.5.3. Simulation results for automatic turn on/off load management for the month of November.....	62
Fig. 4.5.4. Hardware connections and Labview GUI for weather and TOU data acquisition.....	63
Fig. 4.5.5. LabVIEW GUI interface automatic load management (February). ....	64
Fig. 4.5.6. LabVIEW GUI interface for automatic load management for (a) July and (b) November.....	65
Fig. 4.5.7. Simulink model for space heater control.....	68
Fig. 4.5.8. Simulation results for automatic load management (space heater control).....	68

## LIST OF TABLES

TABLE 3.1. PARAMETERS FOR BATTERY SOC ESTIMATION.....	20
TABLE 3.2. SUMMARY OF FOUR DIFFERENT CONDITIONS IN ISLAND MODE .....	28
TABLE 3.3. SUMMARY OF DIFFERENT CONDITIONS IN GRID - CONNECTED MODE .....	35
TABLE 4.1. $r_0$ , $r_1$ , and $r_k$ VALUES FOR DIFFERENT CLIMATE TYPES [13] .....	46
TABLE 4.2. DAY TYPE INDEX VALUES FOR DIFFERENT WEATHER [13].....	50
TABLE 4.3. REGRESSION COEFFICIENTS FOR AUTO REGRESSION MODEL ( FEBRUARY WEEKDAYS).....	54
TABLE 4.4. REGRESSION COEFFICIENTS FOR AUTO REGRESSION MODEL (FEBRUARY WEEKENDS) .....	54
TABLE 4.5. REGRESSION COEFFICIENTS FOR AUTO REGRESSION MODEL (JULY WEEKDAYS).....	55
TABLE 4.6. REGRESSION COEFFICIENTS FOR AUTO REGRESSION MODEL (JULY WEEKENDS).....	55
TABLE 4.7. REGRESSION COEFFICIENTS FOR AUTO REGRESSION MODEL (NOVEMBER WEEKDAYS).....	55
TABLE 4.8. REGRESSION COEFFICIENTS FOR AUTO REGRESSION MODEL (NOVEMBER WEEKENDS) .....	56
Table 4.9. LOAD DEMAND PREDICTION ACCURACY .....	58
TABLE 4.10. SPECIFICATIONS FOR HOUSE SPACE HEATER [21].....	66

## **ABBREVIATIONS**

**SGPN** – Smart Green Power Node

**DC** – Direct Current

**AC** – Alternating Current

**PV** – Photovoltaic

**SOC** – State of Charge

**TOU** – Time of Use

# CHAPTER 1

## INTRODUCTION AND OBJECTIVES

### 1.1 Need for System Level Energy Management

Residential electricity prices are increasing day by day and are expected to continue rising in the future [1]. As shown in Fig. 1.1.1, the electricity price was 11 Cents/kWh for residential sector in 2012. However, the price increased to 12 Cents/kWh in 2014 and it is predicted that this price will increase to 12.8 Cents/kWh in 2015 for the residential sector. At the same time, carbon emissions have also increased over the past few years. According to the new “Clean Power Plan”, carbon emissions must be reduced and a new emission threshold is defined by the plan [2].

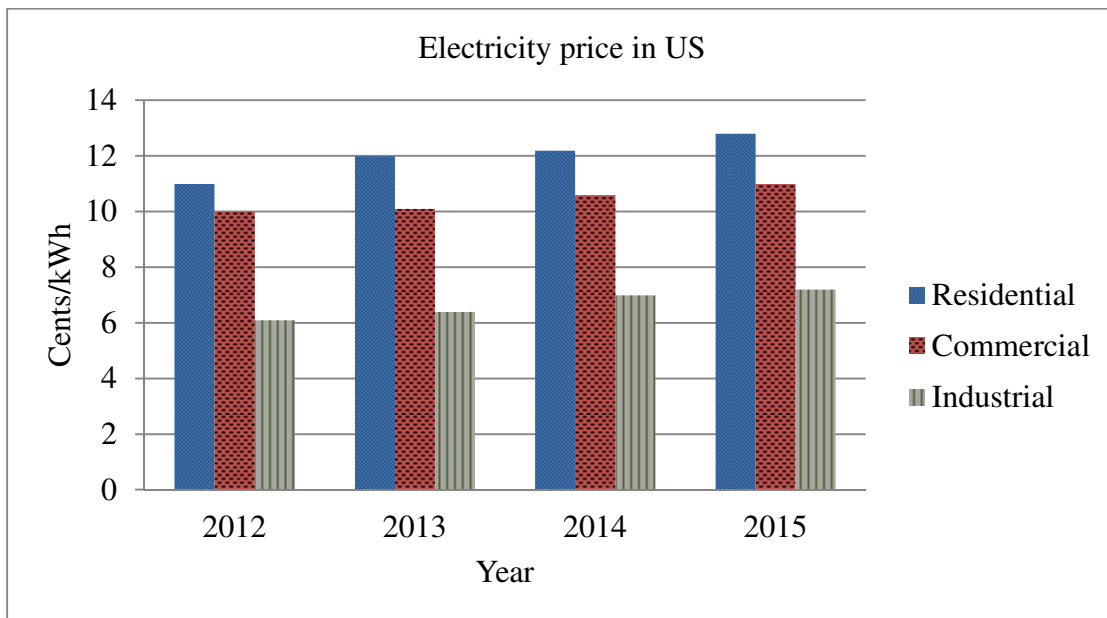


Fig. 1.1.1. Average electricity price in different sectors from 2012 to 2015 [1].

Many companies are trying to develop new technologies which can reduce energy consumption and these emissions which has led to the development of renewable energy sources such photovoltaic (PV) or wind power generation [3][4].

However, there are some concerns regarding the use of PV as a renewable energy source, since there is often a mismatch between load demand and PV power generation [5]. To adjust load demand to match available energy resources an intelligent energy management system is required, which will control the power flow between PV, battery, and grid. This results in an overall reduction of the grid energy consumption for the residence.

## **1.2 Formulation of System Level Energy Management**

This thesis presents an energy management system that includes PV, battery, and grid that accommodates various factors such as TOU pricing, battery state of charge, generated PV power, and user load demand. This system also includes a load management structure which manages the loads according to forecasted PV power, and forecasted load demand. The TOU pricing data is used in the control algorithm along with PV power, load power and battery state of charge information to distribute the generated power to supply the loads. In addition, weather data is used for PV power prediction to further enhance load management. TOU pricing and weather information are acquired directly from Internet sources using serial communication. This information can be remotely monitored via a computer or a smart device. For this thesis a Labview GUI is used for collecting data and monitoring PV power, battery state of charge, and load demand. With all of the collected and monitored data, the system will automatically control the charging and discharging of the battery to maximize cost savings for the end user.



### **1.3 Smart Green Power Node**

The University of Arkansas, in collaboration with the University of South Carolina, started a project in order to build a smart energy management system in January 2013. The system was named the *Smart Green Power Node* (SGPN). This thesis focuses on the design and implementation of the system level control of the SGPN, which includes power flow and load management. This system level control will be interfaced with the hardware during the system integration process.

### **1.4 Thesis Organization**

This thesis is arranged as follows: Chapter 2 covers the theoretical knowledge of battery models for state of charge estimation, solar irradiation model for photovoltaic power prediction, and statistical regression models for electrical load demand prediction. Chapter 3 presents the algorithms, simulation, and DSP implementation of power flow management for both grid-connected and island modes. Chapter 4 covers the design, simulation, and DSP implementation of the PV prediction and load prediction model. It also covers the simulation and Labview GUI implementation of the load management using the PV prediction and load prediction models. Finally, Chapter 5 closes with conclusions and recommendations for future work.

## CHAPTER 2

### THEORETICAL BACKGROUND AND RELATED CONCEPTS

#### 2.1 System level Control Overview

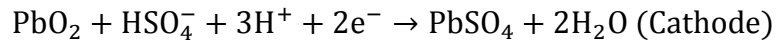
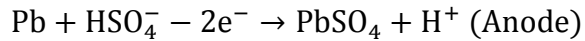
The control structure for this energy management system can be divided into two categories: (a) hardware level control, and (b) system level control. This system level control is then sub-categorized into two sections: (a) Power flow management and (b) Load management. Power flow management system initially monitors the voltage and current for PV, load, grid, and battery. Battery SOC is estimated using battery voltage and current. After that, the system level control algorithm is developed to decide when to charge or discharge the battery based on these data. Lead-acid battery is considered for this project. Hence the construction of lead-acid battery and its equivalent models are studied for the battery SOC estimation. Load management is developed using TOU price, PV power prediction, and load demand prediction. PV power prediction primarily depends on the estimation of solar radiation and weather conditions. Hotell's solar radiation model is studied and described in this thesis. Load demand prediction is a statistical process which depends on historical load profile. Several stochastic time series (STS) methods have been studied and discussed in this thesis paper.

#### 2.2 Lead Acid Battery Construction

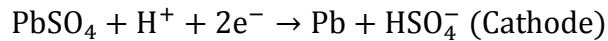
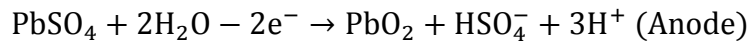
Lead acid batteries are made of lead dioxide, which acts as the positive electrode, and pure lead, which acts as the negative electrode. Sulphuric acid acts as the electrolyte in which the electrodes are immersed. When the battery discharges the electrons travel from the negative

electrode (anode) to positive electrode (cathode) and when the battery charges the anode acts as the positive electrode and the cathode acts as the negative electrode. The chemical reactions in a lead acid battery are given below [6]:

- When discharging:



- When charging:



### 2.3 Lead Acid Battery Models

There are several electrical models commonly applied to lead acid batteries. These models are outlined and described in the subsections that follow.

#### 2.3.1 Simple Battery Model

The simple battery model consists of an electromotive force ( $E_0$ ), internal resistance, and battery terminal voltage [7]. The electromotive force  $E_0$  is obtained from open circuit voltage measurements. In this model, the internal resistance ESR is a constant value. Fig. 2.3.1 shows the circuit diagram for the simple battery model.

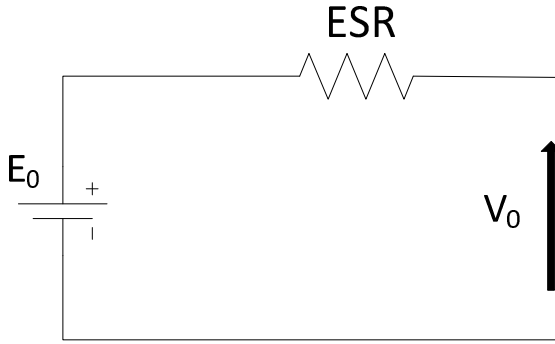


Fig. 2.3.1. Simple battery model circuit diagram [7].

### 2.3.2 Modified Simple Battery Model

The modified simple battery model is quite similar to the simple battery model with one exception: the internal resistance is not constant. It varies with battery state of charge (SOC) [8]. Fig. 2.3.2 shows the circuit diagram for the modified simple battery model. The equation for varying internal resistance is given by:

$$r = \frac{R_0}{SOC^k} \quad (2.1)$$

where,

$r$  is the varying internal resistance,

$R_0$  is the internal resistance when the battery is fully charged,

$SOC$  is the battery state of charge, and

$k$  is battery capacity constant (typically from 1.1 to 1.3 for a lead acid battery).

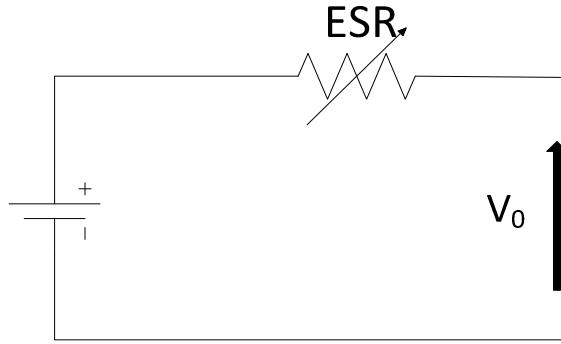


Fig. 2.3.2. Modified battery model circuit diagram [8].

### 2.3.3 Advanced Simple Battery Model

In the advanced simple battery model, the battery terminal voltage depends on the battery open circuit voltage, battery discharge current, and the internal resistance [9]. The equation for the battery terminal voltage is given by:

$$V_{terminal} = V_{oc} - I_{bat} * R_i \quad (2.2)$$

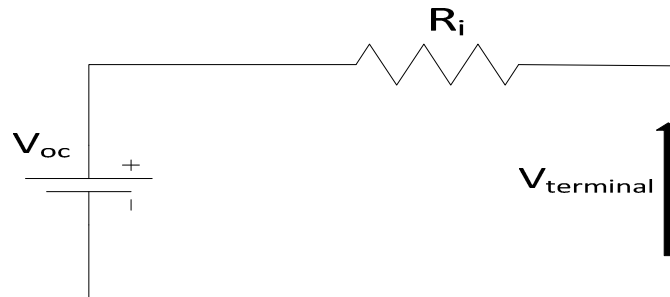


Fig. 2.3.3. Advanced simple battery model circuit diagram [9].

Here, the battery open circuit voltage can be calculated with the knowledge of the battery SOC. The equation for battery open circuit voltage is given by:

$$V_{oc} = a_0 + a_1 * SOC \quad (2.3)$$

Here,  $SOC$  is the state of charge of the battery,  $a_0$  is defined as the terminal voltage when the state of charge is 0% and  $a_1$  can be calculated from getting the value of  $a_0$  and  $V_{oc}$  when  $SOC$  is 100% [10].

### 2.3.4 Thevenin Battery Model

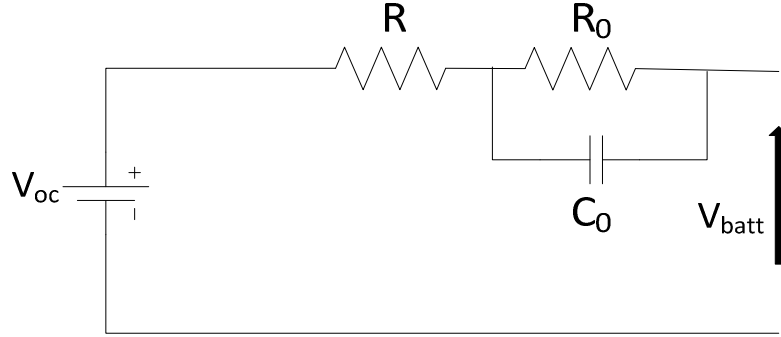


Fig. 2.3.4. Circuit diagram for Thevenin battery model [11].

The Thevenin battery model is composed of the battery open circuit voltage, internal resistance, capacitance, over-voltage resistance, and the capacitance between electrolyte and electrode [11]. The equations for the Thevenin battery model are:

$$V_{bat} = V_{oc} - (I_{bat} * R + V_0) \quad (2.4)$$

$$V_0 = \left( \frac{1}{R_0} + \frac{1}{C_0} \right) * I_{bat} \quad (2.5)$$

### 2.3.5 Third Order Model

The third order model can be divided into three sections. Each of the sections is described below:

### 2.3.5.1 Capacity Model

The battery charge and discharge current is the input for this capacity model. Battery capacity remaining can be calculated by knowing the rated capacity and electrolyte temperature [12]. The equation for battery capacity is given by:

$$C(I_n, \theta_n) = C_0 * \left(1 + \frac{\theta_n}{\theta_f}\right)^\varepsilon \quad (2.6)$$

where  $\theta_f$  is the electrolyte temperature and  $C_0$  is the battery rated capacity. When the battery is discharging the total capacity can be calculated by the following equation:

$$C(I, \theta) = \frac{K_c * C_0}{1 + (K_c - 1) \left(\frac{I}{I^*}\right)^\delta} * \left(1 + \frac{\theta_n}{\theta_f}\right)^\varepsilon \quad (2.7)$$

Here  $K_c$  is a constant,  $\theta_f$  is the electrolyte temperature, and  $I^*$  is the nominal battery current.

The amount of charge extracted from the battery can be calculated by integrating the total current flow from and to the battery [12]. The equation for the extracted charge is given by:

$$Q_e(t) = Q_{int} + \int_0^t -I_m(\tau) d\tau \quad (2.8)$$

Here  $Q_e$  is the charge extracted from the battery,  $Q_{int}$  is the initial extracted charge,  $I_m$  is the current flowing from or to the battery, and  $t$  is the simulation time.

Finally, battery state of charge (SOC) can be calculated by knowing the battery capacity and charge extracted from the battery [12]. The equation is given by:

$$SOC = 1 - \frac{Q_e}{C(0, \theta)} \quad (2.9)$$

### 2.3.5.2 Voltage Model

The state of charge is calculated using the capacity model in [12], which requires the battery electrolyte temperature  $\theta_f$  as an input. To acquire this electrolyte temperature it is required to create a thermal model. The input for the thermal model comes from the voltage model which takes  $SOC$  as an input and generates  $R_0$ ,  $R_1$ ,  $R_2$  and electromotive force  $E_m$  for the thermal model.

The electromotive force  $E_m$  varies with temperature and state of charge. The equation for  $E_m$  is:

$$E_m = E_{m0} - K_e(273 + \theta)(1 - SOC) \quad (2.10)$$

Here  $E_{m0}$  is the open circuit voltage at full charge,  $\theta$  is the electrolyte temperature, and  $SOC$  is the battery state of charge.

The resistance  $R_1$  varies with depth of discharge. The equation for the depth of discharge is:

$$DOC = 1 - \frac{Q_e}{C(I_{avg}, \theta)} \quad (2.11)$$

And the equation for  $R_1$  is given by:

$$R_1 = -R_{10} \ln(DOC) \quad (2.12)$$

Here  $R_{10}$  is a constant and  $DOC$  is the depth of discharge.

The main branch resistance  $R_2$  is dependent on battery state of charge. When the battery is charging it affects this resistance, but it is not affected by battery discharge current. The equation for  $R_2$  is given by:

$$R_2 = R_{20} * \frac{e^{[A_{21}(1-SOC)]}}{1 + e^{\frac{A_{22}I_m}{T^*}}} \quad (2.13)$$



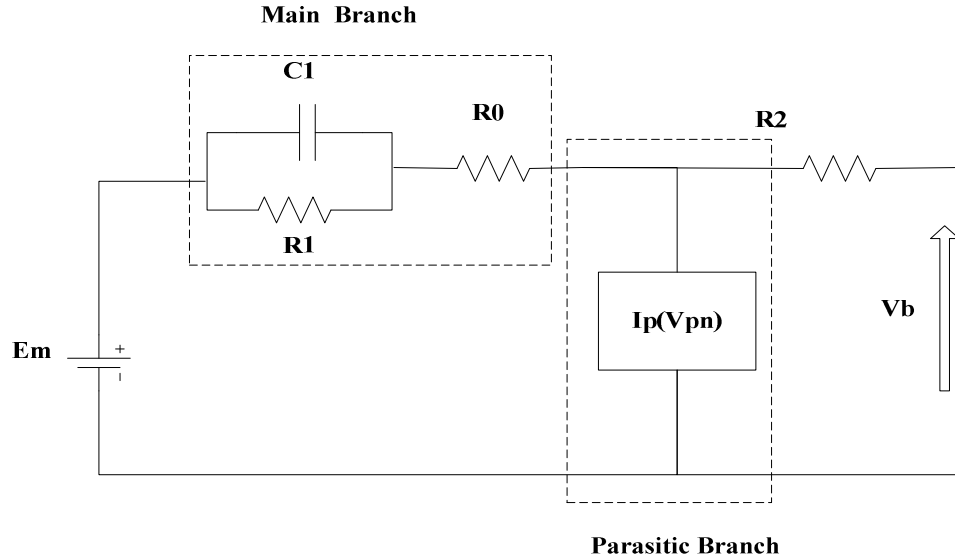


Fig. 2.3.5. Third order battery model circuit diagram [7].

where  $R_2$  is the main branch resistance,  $R_{20}$  is a constant in ohms,  $E_m$  is the open circuit voltage,  $I_m$  is the main branch current,  $I^*$  is the nominal battery current and  $A_{22}$  is an empirical constant.

The terminal resistance  $R_0$  varies with battery *SOC* but can be assumed constant for different temperatures. The equation for  $R_0$  is given by:

$$R_0 = R_{00}[1 + A_0(1 - SOC)] \quad (2.14)$$

where  $R_0$  is the terminal resistance in ohms,  $R_{00}$  is the initial value of  $R_0$  when the state of charge is 100%.

### 2.3.5.3 Thermal Model

The thermal model calculates the battery electrolyte temperature by estimating the internal power losses due to resistance and ambient temperature [12]. The equation for electrolyte temperature is given by:

$$\theta(t) = \theta_{int} + \int_0^t \frac{(P_s - \frac{(\theta - \theta_a)}{R_\theta})}{C_\theta} d\tau \quad (2.15)$$

where  $P_s$  is the power loss due to the resistances. The equation for  $P_s$  is given by:

$$P_s = \frac{V_{r1}^2}{R_1} + I^2 R_0 + I^2 R_2 \quad (2.16)$$

## 2.4 PV Prediction Overview

The output from a PV panel can be predicted based on solar irradiation and temperature forecasted [13]. The equation for the PV power output is given by:

$$P_{pv} = \eta A_{pv} G_t [1 - 0.005(T_c - 25)] \quad (2.17)$$

Here,  $P_{pv}$  is the PV power output in kW,  $A_{pv}$  is the solar panel area in  $m^2$ ,  $G_t$  is the total solar radiation in  $W/m^2$  and  $T_c$  is the weather temperature in  $^\circ C$ .

Weather forecasting data can be found from various Internet weather service websites. The solar irradiation depends on different geographical conditions such as latitude, longitude, time zone, day of the year, local mean time and the altitude of the area [13].

### 2.4.1 Clear Sky Radiation Model Overview

Total solar radiation on a PV panel can be divided into three components: beam radiation, diffuse radiation, and reflected radiation [13]. Beam radiation is one kind of solar radiation which comes straight to the surface of the earth from the Sun. Radiation scattered by molecules and particles in the atmosphere is called diffuse radiation. Reflected radiation comes from the reflection of sunlight from non-atmospheric entities such as the ground. These three radiation components can be calculated using the following equations:

$$G_{bt} = G_{on} * t_b * \cos \theta_s \quad (2.18)$$

$$G_{dt} = G_{on} * t_d * \cos \theta_z * \frac{1 + \cos \beta}{2} \quad (2.19)$$

$$G_{rt} = \rho * G_{on} * t_r * \cos \theta_z * \frac{1 + \cos \beta}{2} \quad (2.20)$$

Here,  $G_{on}$  is the extraterrestrial solar radiation in  $\text{W/m}^2$ ,  $\rho$  is the average reflectance of the ground, and  $t_b$ ,  $t_d$ , and  $t_r$  are atmospheric transmittance for beam, diffuse and reflected radiation.  $\theta_z$  is the zenith angle and  $\beta$  is the inclined angle.  $G_{on}$  is calculated by the following equation:

$$G_{on} = G_{sc} * \left( 1 + .033 \cos \left( \frac{2\pi D}{365} \right) \right) \quad (2.21)$$

where  $D$  is the number of the day of the year and  $G_{sc}$  the solar constant ( $1367 \text{ W/m}^2$ ).

The transmittance for beam, diffuse and reflected radiation can be calculated by the equations proposed by Hottel [13].

$$t_b = (a_0 + a_1 * e^{\left(\frac{-k}{\cos \theta_z}\right)}) \quad (2.22)$$

$$t_d = 0.271 - 0.294 * t_b \quad (2.23)$$

$$t_r = 0.271 - 0.706 * t_b \quad (2.24)$$

$a_0$ ,  $a_1$ , and  $k$  are constants that can be calculated from Hottel's equations [13]:

$$a_0 = r_0 * (0.4237 - 0.00821 * (6 - A)^2) \quad (2.25)$$

$$a_1 = r_1 * (0.5055 - 0.00595 * (6.5 - A)^2) \quad (2.26)$$

$$k = r_k * (0.2711 - 0.1858 * (2.5 - A)^2) \quad (2.27)$$

where  $A$  is the altitude of the area.  $r_0$ ,  $r_l$  and  $r_k$  are the correction factors which vary from climate to climate. The solar zenith angle  $\theta_z$  and incident angle vector  $\theta_s$  can be calculated using the following equations:

$$\cos\theta_z = \cos\delta * \cos\varphi * \cos\omega + \sin\delta * \sin\varphi \quad (2.28)$$

$$\begin{aligned} \cos\theta_s &= \sin\delta * \sin\varphi * \cos\beta - \sin\delta * \cos\varphi * \sin\beta * \\ \cos\alpha + \cos\delta * \cos\varphi * \cos\beta * \cos\omega + \cos\delta * \sin\varphi * \\ \sin\beta * \cos\alpha * \cos\omega + \cos\delta * \sin\alpha * \sin\omega * \sin\beta \end{aligned} \quad (2.29)$$

where  $\delta$  is the solar declination,  $\varphi$  is latitude of the area,  $\omega$  is the hour angle vector, and  $\alpha$  and  $\beta$  are inclination of the surface. The equations for calculating these components are given by:

$$\delta = \frac{\pi}{180} * 23.45 * \sin\left(\frac{2\pi(D + 284)}{365}\right) \quad (2.30)$$

$$\omega = \frac{\pi}{180} * \left( (LMT - TZ - 12) * 15 + longitude + \frac{E_t}{4} \right) \quad (2.31)$$

where  $LMT$  is the local mean time,  $TZ$  is the time zone and  $E_t$  is solar time in minutes.

## 2.5 Load Prediction Overview

There are several methods used for electrical load prediction. Among them, the stochastic time series (STS) method is the most popular approach that is applied for short term load forecasting. The STS method is classified into different models depending on certain characteristics [14]. These models will be outlined and detailed in the following sub-sections.

### 2.5.1 Autoregressive Model (AR)

In the auto-regression process the current value can be expressed linearly in terms of previous value and random noise [14]. The equation for the regression process is given by:

$$y(t) = a(t) + \phi_1 y(t-1) + \phi_2 y(t-2) + \dots + \phi_p y(t-p) \quad (2.31)$$

where  $y(t)$  is the current value,  $\phi_1, \phi_2, \dots, \phi_p$  are the regression coefficients and  $y(t-1), y(t-2), \dots, y(t-p)$  are the previous value. The random noise  $a(t)$  has zero mean and unknown fixed variance. The previous loads can be models using the backshift operator  $B$ .

$$\phi(B)y(t) = a(t) \quad (2.32)$$

### 2.5.2 The Moving Average Process (MA)

In the moving average process the current can be calculated from the previous values of white noise. In the auto-regression process there is only one white noise that is used but in the moving average process, a number of white noise series are used [14]. The series of white noise can be constructed by forecasting errors and residuals. The equation for the moving average process is given by:

$$y(t) = a(t) + \theta_1 a(t-1) + \theta_2 a(t-2) + \dots + \theta_p a(t-p) \quad (2.33)$$

White noise series can be written in terms of the backshift operator  $B$ .

$$y(t) = \theta(B)a(t) \quad (2.34)$$

### 2.5.3 The Autoregressive Moving Average Process (ARMA)

In the autoregressive moving average process the current value  $y(t)$  is calculated from the previous values ( $y(t-1), y(t-2), \dots, y(t-p)$ ) and also from the series of the white noise ( $a(t-1), a(t-2), \dots, a(t-p)$ ) [14]. The equation for autoregressive moving average process is given by:

$$\begin{aligned} y(t) = & \phi_1 y(t-1) + \phi_2 y(t-2) + \dots + \phi_p y(t-p) + a(t) \\ & + \theta_1 a(t-1) + \theta_2 a(t-2) + \dots + \theta_p a(t-p) \end{aligned} \quad (2.35)$$

Using the backshift operator the equation can be written as:

$$\phi(B)y(t) = \theta(B)a(t) \quad (2.36)$$

#### 2.5.4 *The Autoregressive Integrated Moving Average Process (ARIMA)*

The regression processes described earlier are stationary processes where the covariance of the time series does not change with time [14]. However, for a non-stationary process the first step is to convert to a stationary process by introducing the delta operator. Then, the stationary process can be modeled with an AR, MA, or ARMA process. The model equation for the ARIMA process is given by:

$$\phi(B)\nabla dy(t) = \theta(B)a(t) \quad (2.37)$$

## CHAPTER 3

### POWER FLOW MANAGEMENT

#### 3.1 Power Flow Management Structure

The objective for system level control is to satisfy the residential load demand by managing the onsite energy sources including PV, battery, and generator in conjunction with the grid. The rule-based algorithm described in [15] is used for power flow management both for grid-connected and island modes. This algorithm involves the battery state of charge estimation, available PV power, load demand, and electricity time of use (TOU) pricing. The residential load management algorithm is also developed using TOU price, PV power prediction, and load power prediction acquired for one day. The block diagram for the system level control is shown in Fig. 3.1.1.

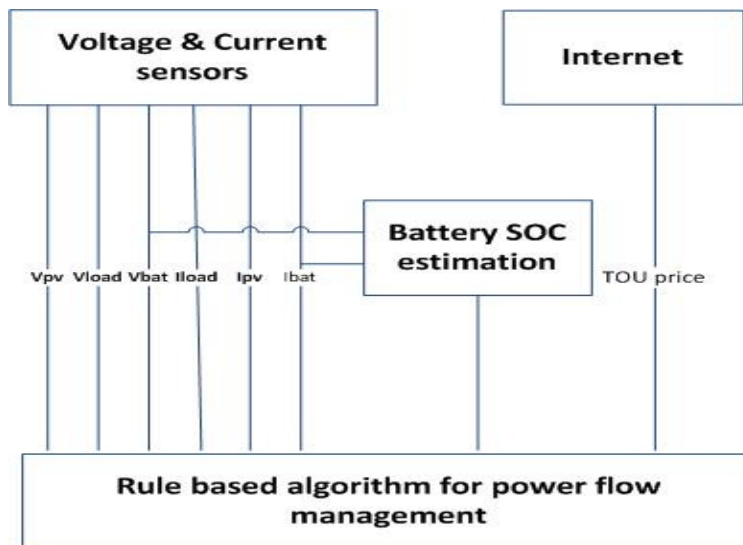


Fig. 3.1.1. Block diagram for power flow management structure.

PV power and load demand can be calculated from their voltage and current values using voltage and current sensors for PV and load. Battery SOC estimation is performed using sensed battery voltage and current. TOU pricing is parsed from the Internet using serial communication and a LabVIEW GUI. Finally, a rule-based algorithm is implemented to make the decision of when to charge or discharge the battery based on this data.

### 3.2 Battery SOC Estimation

Four 12 V lead-acid batteries have been chosen for this power routing system. The lead-acid batteries modeled in Simulink are used in the system level control for battery SOC estimation. There are several known methods for battery SOC estimation. Two are considered here:

- Discharge current method
- Terminal voltage method

#### 3.2.1 Discharge Current Method

In the discharge current method, the SOC is calculated from battery discharge current and the electrolyte temperature. A third order lead acid battery model, as described in Chapter 2, is used for the simulation.

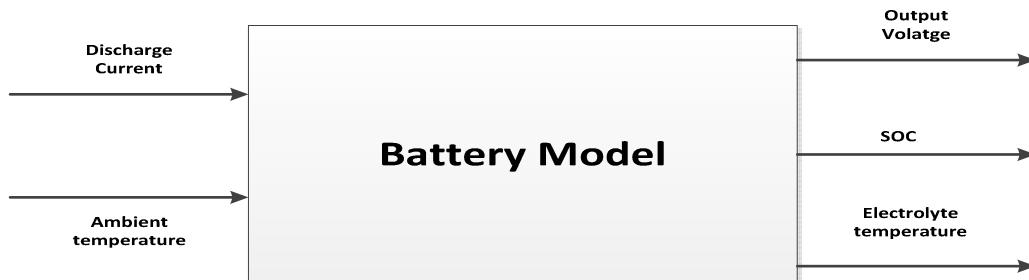


Fig. 3.2.1. Top level block diagram for battery SOC estimation using discharge current.



In Fig. 3.2.1,  $I_b$  is the battery discharge current,  $\theta_a$  is the ambient temperature,  $V_0$  is terminal output voltage,  $Q_e$  is the amount of charge extracted from the battery and  $\theta$  is the battery electrolyte temperature. The amount of charge extracted from the battery can be calculated by integrating the discharging current and the battery SOC is dependent on this extracted charge and the battery capacity. This battery capacity is dependent on the battery electrolyte temperature.

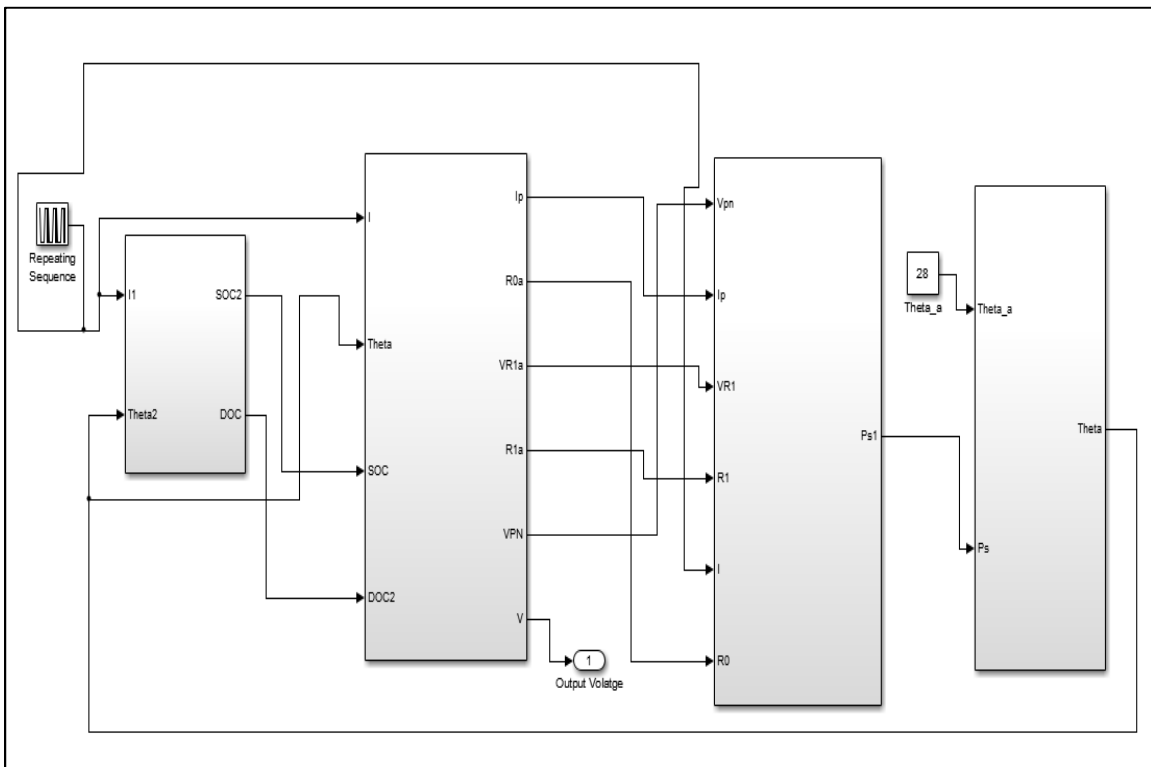


Fig. 3.2.2. Simulink™ block diagram with capacity model, voltage model and thermal model.

Battery electrolyte temperature depends on the amount of power loss due to heat and the ambient temperature. The electrolyte temperature is calculated using the thermal model described in section 2.3.5.3, and power loss due to heat is calculated using the voltage model described in section 2.3.5.2. Once the electrolyte temperature is known, the battery capacity is calculated using the following equation:

$$C(I, \theta) = \frac{K_c * C_0}{1 + (K_c - 1) \left(\frac{I}{I^*}\right)^\delta} * \left(1 + \frac{\theta n}{\theta_f}\right)^\varepsilon \quad (3.1)$$

The amount of charge extracted  $Q_e$ , is calculated by integrating the battery discharge current  $I_m$ . Then, battery *SOC* is calculated by taking into account the amount of charge extracted from the battery and the battery capacity.

$$Q_e(t) = Q_{int} + \int_0^t -I_m(\tau) d\tau \quad (3.2)$$

$$SOC = 1 - \frac{Q_e}{C(0, \theta)} \quad (3.3)$$

The entire battery model is simulated in Matlab/Simulink. The parameters used in this model are taken from [12] and [7] and are listed in Table I below.

TABLE 3.1. PARAMETERS FOR BATTERY SOC ESTIMATION

Parameters for capacity model	$C_0^* = 72.37 \text{ Ah}$
	$K_c = 1.2$
	$\varepsilon = 1.19$
	$\delta = 1.75$
	$K_t = 0.47$
	$\theta_f = -40 \text{ }^\circ\text{C}$
Parameters for voltage model	$R_{00} = 2 \text{ m}\Omega$
	$A_0 = -0.30$
	$R_{10} = 7 \text{ m}\Omega$
	$K_e = 5.8 \times 10^{-4}$
	$A_{21} = 8$
	$A_{22} = -8.45$
Parameters for thermal model	$C_\theta = 15 \text{ Wh/C}$
	$R_\theta = 0.2 \text{ C/W}$

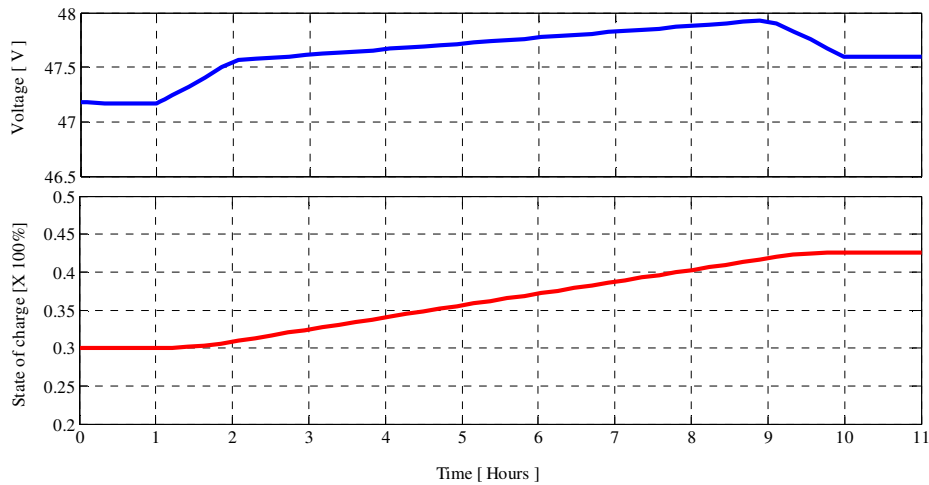


Fig. 3.2.3. Simulation results for battery SOC estimation when the battery is charging.

Fig. 3.2.3 shows simulation results for battery SOC estimation when the battery is charging. Initially, the simulation is performed for a 12 V (6 cells) battery. It is then multiplied by 4 to achieve 48 V from the battery. The charging current limit is 10 A and capacity is 72.37 Ah. So, when the battery is charging, the terminal voltage increases linearly and battery state of charge also increases with this charging current. Initially, the battery voltage was 47.2 V and the state of charge was 30%. The voltage then starts increasing with the charging current. It increases up to 47.9 V and the battery SOC reaches 43% from a starting point of 30%.

Fig. 3.2.4 shows the simulation results for battery SOC estimation when the battery is discharging. From the battery datasheet it is found that the maximum voltage of the battery is limited to 51.2 V. For the simulation, the discharge current limit is -20 A. When the battery discharges, the terminal voltage decreases linearly with the battery discharging current and the battery state of charge also decreases from the previous condition. Initially, the battery voltage was 49 V and the state of charge was 82%. The voltage then starts decreasing as the battery

discharges current. Eventually, it decreases to 48.37 V and the battery SOC reaches 74% from an initial starting point of 82%.

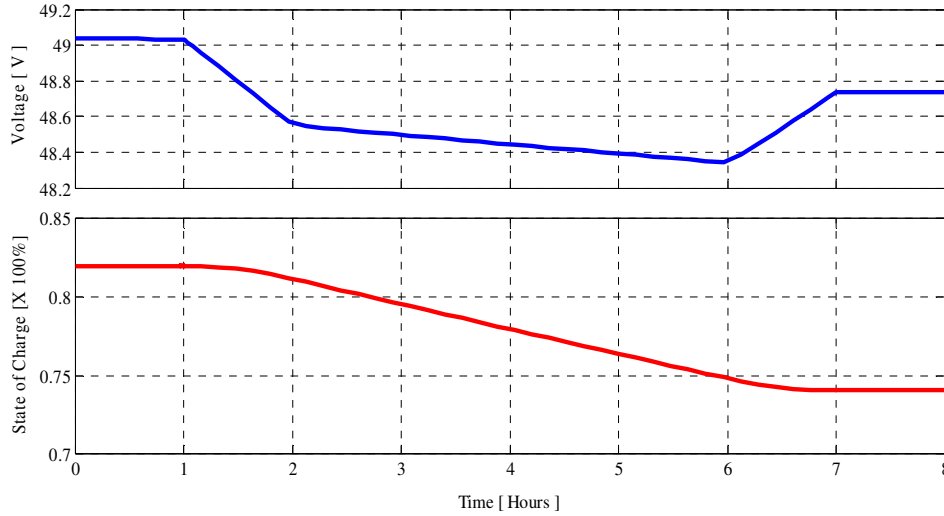


Fig. 3.2.4. Simulation results for battery SOC estimation when the battery is discharging.

The discharge current method of estimating battery SOC is not straightforward, since it depends on many factors such as temperature and battery capacitance. It also requires temperature sensor to monitor the ambient temperature. However, the voltage method is implemented in system level since it only requires voltage and current sensor for battery.

### 3.2.2 Battery Voltage Method

Voltage method is another way to estimate battery state of charge which requires estimating battery open circuit voltage. The equation for battery open circuit voltage is given by:

$$V_{oc} = a_1 * SOC + a_0 \quad (3.4)$$

So, the equation for battery state of charge is:

$$SOC = \frac{V_{oc} - a_0}{a_1} \quad (3.5)$$

Here,  $SOC$  is the state of charge of the battery,  $a_0$  is defined as the terminal voltage when the state of charge is 0% and  $a_1$  can be calculated from getting the value of  $a_0$  and  $V_{oc}$  when  $SOC$  is 100% [10].

To calculate the battery state of charge under closed circuit conditions, battery current and internal resistance are taken into account for accurate output [16]. The equation for battery terminal voltage in a closed circuit condition is given by:

$$V_{terminal} = V_{oc} - I_{bat} * R_i \quad (3.6)$$

$$V_{oc} = V_{terminal} + I_{bat} * R_i \quad (3.7)$$

where  $I_{bat}$  is the battery current and  $R_i$  is the battery internal resistance.

The battery state of charge under closed conditions can be estimated using the equation given by:

$$SOC = \frac{(V_{terminal} + I_{bat} * R_i) - a_0}{a_1} \quad (3.8)$$

Four Optima™ red top 75/25 12 V batteries are used for this residential power router system [17]. The terms  $a_0$  and  $a_1$  are calculated from datasheet information and  $R_i$  is directly given in the datasheet. From the datasheet, it is found that the open circuit voltage at 100% state of charge is 12.8 V for one battery [17]. Since four batteries are used in series, the open circuit voltage at 100%  $SOC$  is 51.2 V and the open circuit voltage at 0%  $SOC$  is 42 V ( $a_0$ ). The term  $a_1$  is calculated as  $(51.2 - 42) / 1 = 9.2$  and the internal resistance,  $R_i$  is .0030  $\Omega$ . The battery terminal voltage and current are measured using current and voltage sensors. Given these parameters, the battery state of charge can be calculated using the above equation.

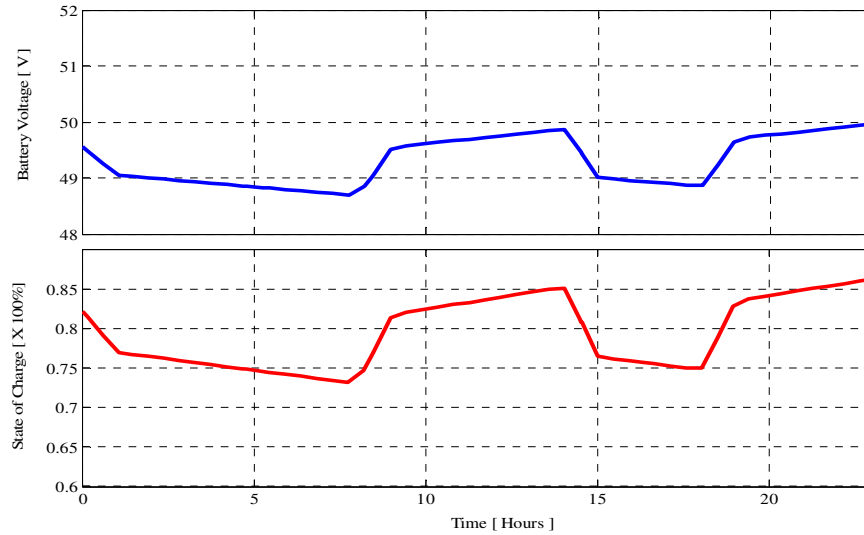


Fig. 3.2.5. Simulation results for battery SOC estimation using voltage method.

Fig. 3.2.5 shows the battery SOC estimation waveform using the voltage method. The battery charging limit is 10 A and the discharge limit is -20 A. The battery voltage was initially at 49.5 V. The battery SOC decreases to 75% from 82%. As the battery charges, the battery voltage starts increasing and the battery SOC logically increases as well.

### 3.3 Battery Charge and Discharge Control

After acquiring battery state of charge information, the next step is to use this SOC information along with the generated PV power data, load demand data, and electricity TOU pricing data to calculate a battery current reference for the hardware level control to decide when to charge or discharge the battery. The rule-based algorithm is used to calculate the battery current reference. This rule-based algorithm is divided into two modes of operation:

- Island mode control
- Grid-connected mode control

### 3.3.1 Island Mode Control

In island mode the system is not connected with the grid. Load demand is served by PV generated power, available battery storage, and the AC generator. Thus, electricity grid pricing is not used for control decisions. Fig. 3.1.1 shows the algorithm that is used for island mode power flow management.

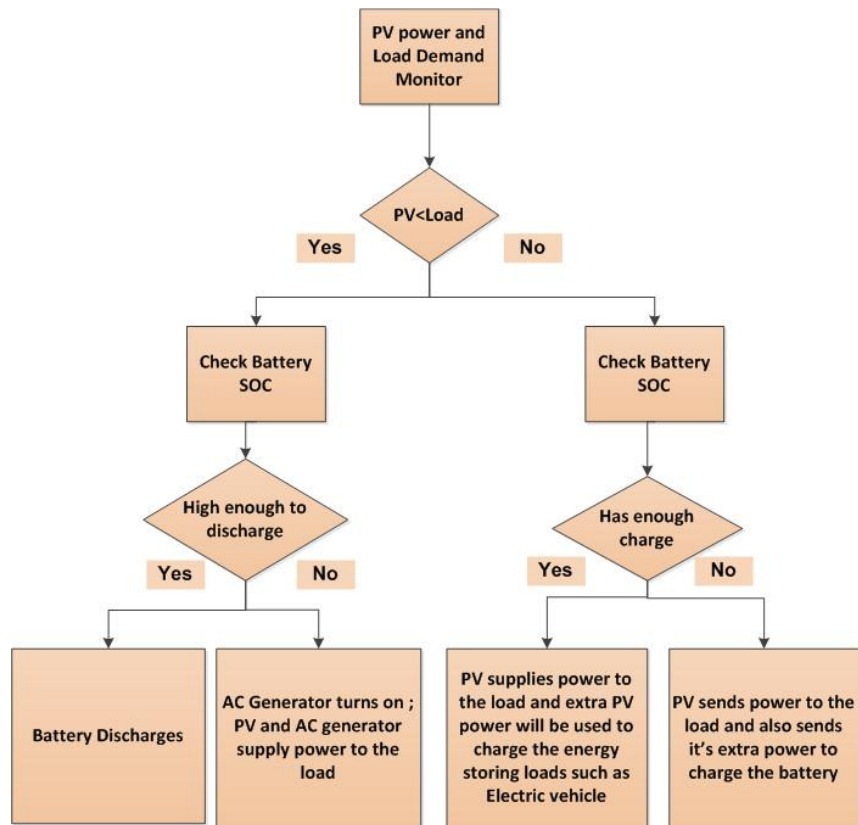


Fig. 3.3.1. Flow chart for power flow management in island mode.

The simulations are performed for a 2 kW system. The output voltage is chosen to be between 50 V and 60 V for a 2 kW PV system and it is assumed that four 12 V batteries are used in series in the system. The simulation was completed for a time period covering one day (24 hours).

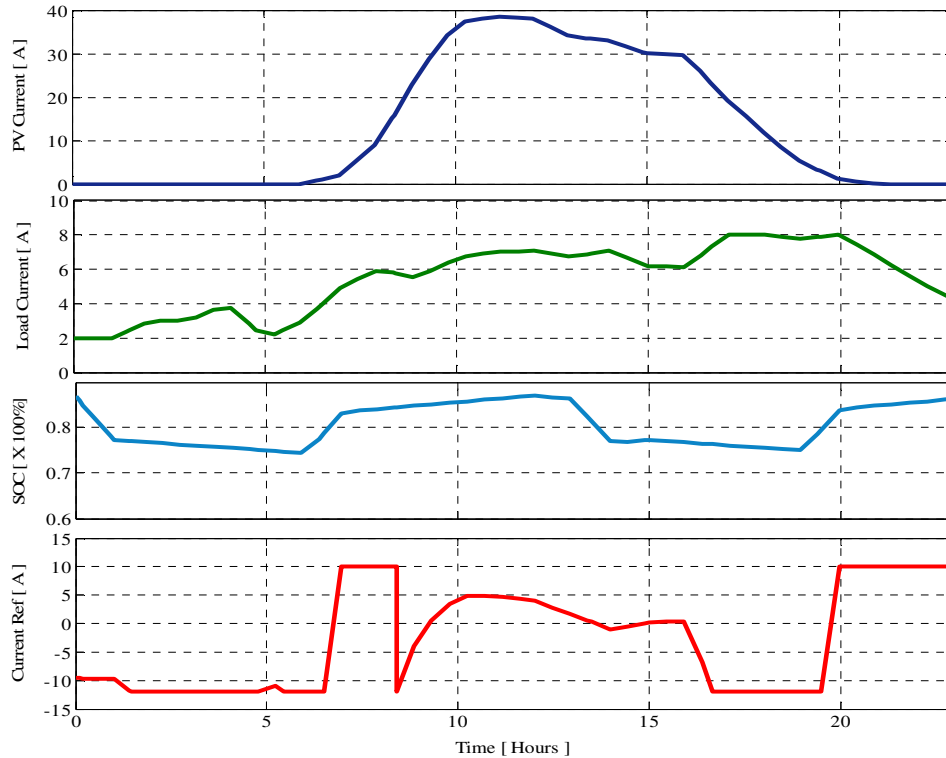


Fig. 3.3.2. Simulation results for island mode.

As is shown in Fig. 3.3.2, at 5:00 am in the morning when there is no solar radiation:

- PV current is zero
- Load current is 2.35 A
- Battery state of charge is 0.74

PV power =  $(0 \text{ A} * 50 \text{ V}) = 0 \text{ W}$ ; Load Power =  $(2.35 \text{ A} * 240 \text{ V}) = 564 \text{ W}$ ; Load power > PV power. Hence, the load needs compensation from the battery with  $\{(0 \text{ W} - 564 \text{ W}) \div 48.8 \text{ V}\} = -11.55 \text{ A}$ , which doesn't cross the battery discharge limit of -20 A. Thus, the generator is not required at that moment and remains off.

At 10:00 am in the morning when there is some amount of solar radiation:



- PV Current is 37 A
- Load Current is 6.7 A
- Battery SOC is 0.85

PV power = (37 A \* 50 V) = 1850 W; Load power = (6.7 A \* 240 V) = 1608 W; Load power < PV power. Hence, the PV can charge the battery using excess power with (1850 W – 1608 W) / 49.9 V = 4.84 A. As in the previous case, the charging limit of 10 A is not crossed, and the generator remains off.

At 3:00 pm in afternoon:

- PV Current is 30 A
- Load Current is 6.2 A
- Battery SOC is 0.77

PV power = (30 A \* 50 V) = 1500 W; Load power = (6.2 A \* 240 V) = 1488 W; Load power < PV power. Now, the PV can charge the battery using excess power with (1500 W – 1488 W) / 49.1 V = 0.24 A. Again, the generator remains off.

At 8:00 pm:

- PV Current is 1 A
- Load Current is 8 A
- Battery SOC is 0.839

Here, PV power = (1 A \* 50 V) = 50 W; Load Power = (8 A \* 240 V) = 1920 W. So, load power > PV power. The load therefore needs compensation of (1920 W – 50 W) = 1870 W, which requires the battery to discharge (-1870 W / 49.75 V) = -37.58 A. So, the generator must be turned on to provide 10 A of current to the battery.

TABLE 3.2. SUMMARY OF FOUR DIFFERENT CONDITIONS IN ISLAND MODE

PV Power (W)	Load Demand (W)	Battery Voltage (V)	SOC (%)	I <sub>ref</sub> (A)
0	564	48.8	74	-11.55
1850	1608	49.9	85	4.84
1500	1488	49.1	77	0.24
50	1920	49.75	83.9	10

TABLE 3.2 shows the summary of the island mode control output according to four different values for PV power, load demand, and battery SOC. Battery current reference calculation is shown according to different operating scenarios for a day. For island mode control, this battery current reference is used by the hardware control to charge and discharge the battery. The control algorithm is able to handle all possible scenarios according to the state diagram shown in Fig. 3.3.3. However, these four scenarios have been shown in this paper since it covers the major operating conditions including charging the battery, discharging the battery, and turning on the generator.

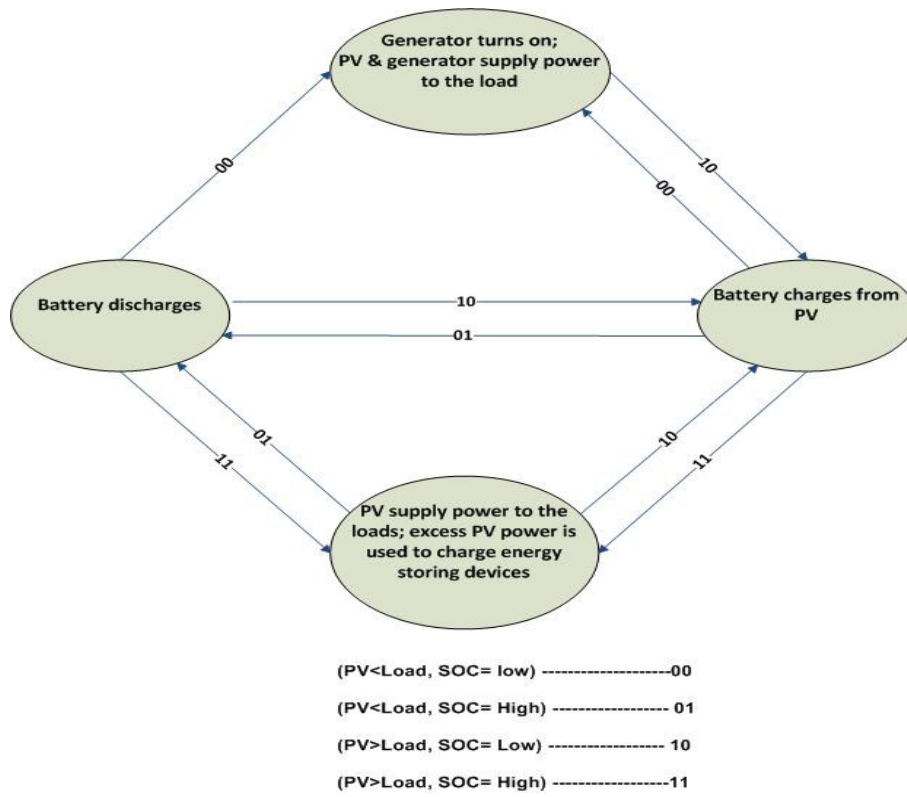


Fig. 3.3.3. State diagram for island mode.

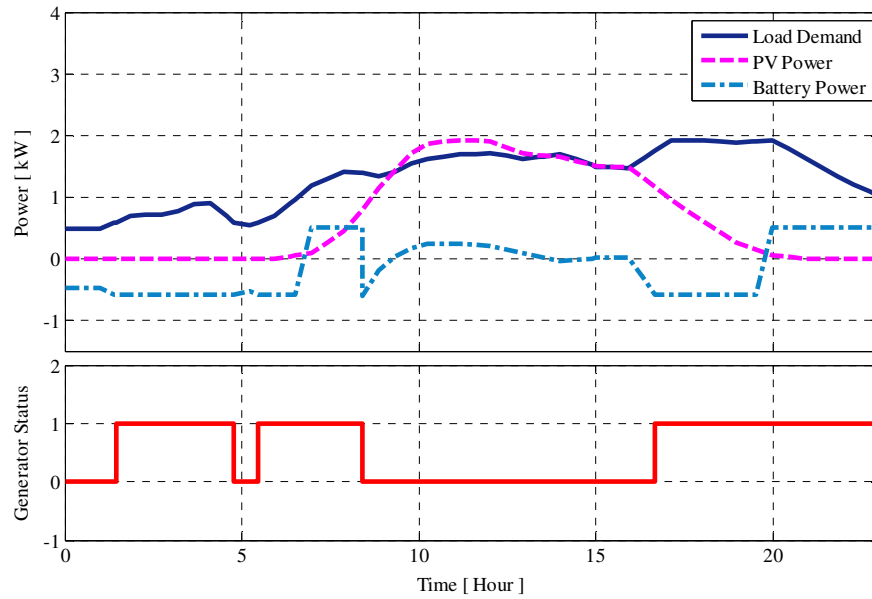


Fig. 3.3.4. Power flow and generator status waveform in island mode.

In Fig. 3.3.4 it is shown that when there is no PV power available the battery discharges and is not sufficient to supply power to the load, so the generator also turns on to compensate for the demand. However, when the PV power is available, it supplies the necessary power to the load and excess power is diverted to charge the battery. At night, when the load demand is high but there is no PV power available, the generator turns on to supply power to the load and charge the battery.

### 3.3.2 DSP Implementation of Island Mode Control

Expression	Type	Value	Address
Ipv	double[4]	0x0000C040@Data	0x0000C040@Data
(*) [0]	double	0.0	0x0000C040@Data
(*) [1]	double	37.0	0x0000C042@Data
(*) [2]	double	30.0	0x0000C044@Data
(*) [3]	double	1.0	0x0000C046@Data
(*) Vpv	double	50.0	0x0000C006@Data
Iload	double[4]	0x0000C01C@Data	0x0000C01C@Data
(*) [0]	double	2.35	0x0000C01C@Data
(*) [1]	double	6.7	0x0000C01E@Data
(*) [2]	double	6.2	0x0000C020@Data
(*) [3]	double	8.0	0x0000C022@Data
(*) Vload_rms	double	240.0	0x0000C004@Data
Vbat	double[4]	0x0000C024@Data	0x0000C024@Data
(*) [0]	double	48.8	0x0000C024@Data
(*) [1]	double	49.9	0x0000C026@Data
(*) [2]	double	49.1	0x0000C028@Data
(*) [3]	double	49.75	0x0000C02A@Data
Ibat	double[4]	0x0000C034@Data	0x0000C034@Data
(*) [0]	double	-10.0	0x0000C034@Data
(*) [1]	double	10.0	0x0000C036@Data
(*) [2]	double	-10.0	0x0000C038@Data
(*) [3]	double	10.0	0x0000C03A@Data
SOC	double[4]	0x0000C048@Data	0x0000C048@Data
(*) [0]	double	0.7423911 (Decimal)	0x0000C048@Data
(*) [1]	double	0.8554351	0x0000C04A@Data
(*) [2]	double	0.7749997	0x0000C04C@Data
(*) [3]	double	0.8391306	0x0000C04E@Data
Iref	double[4]	0x0000C014@Data	0x0000C014@Data
(*) [0]	double	-11.55738	0x0000C014@Data
(*) [1]	double	4.849699	0x0000C016@Data
(*) [2]	double	0.2443992	0x0000C018@Data
(*) [3]	double	10.0	0x0000C01A@Data
(*) mode	int	0	0x0000C000@Data
generator_status	int[4]	0x0000C010@Data	0x0000C010@Data
(*) [0]	int	0	0x0000C010@Data
(*) [1]	int	0	0x0000C011@Data
(*) [2]	int	0	0x0000C012@Data
(*) [3]	int	1	0x0000C013@Data

Fig. 3.3.5. Output results for DSP implementation of island mode control.

The island mode power flow management algorithm is implemented using a TI TMS320F28335 microcontroller. The objective is to calculate the battery current reference based on battery SOC, PV power, and load power. Fig. 3.3.5 shows the DSP output for battery current reference in island mode for different conditions representing 24 hours of operation. Battery SOC is estimated by sensing battery voltage and battery current. PV power and load power is calculated by sensing PV output voltage, PV output current, load voltage, and load current. These values come from the hardware level control algorithm where they are sensed using voltage and current sensors. Finally, the battery current reference is calculated for island mode, which is sent to the hardware level control algorithm to maintain the power between battery and PV. It also sends a switching signal to turn on/off the generator whenever it is needed.

### *3.3.3 Grid-connected Mode*

In grid-connected mode, the system is always connected to the grid. It can pull required power from the grid or it can “sell” back some power to the grid, depending on the power available from the PV and battery. The electricity TOU pricing plays an important role in this control algorithm. Twenty-four hour TOU data is obtained from Internet source. The algorithm calculates the current reference based on battery charging and discharging limit, TOU price information, PV power, load demand, and battery SOC as shown in Fig. 3.3.6.

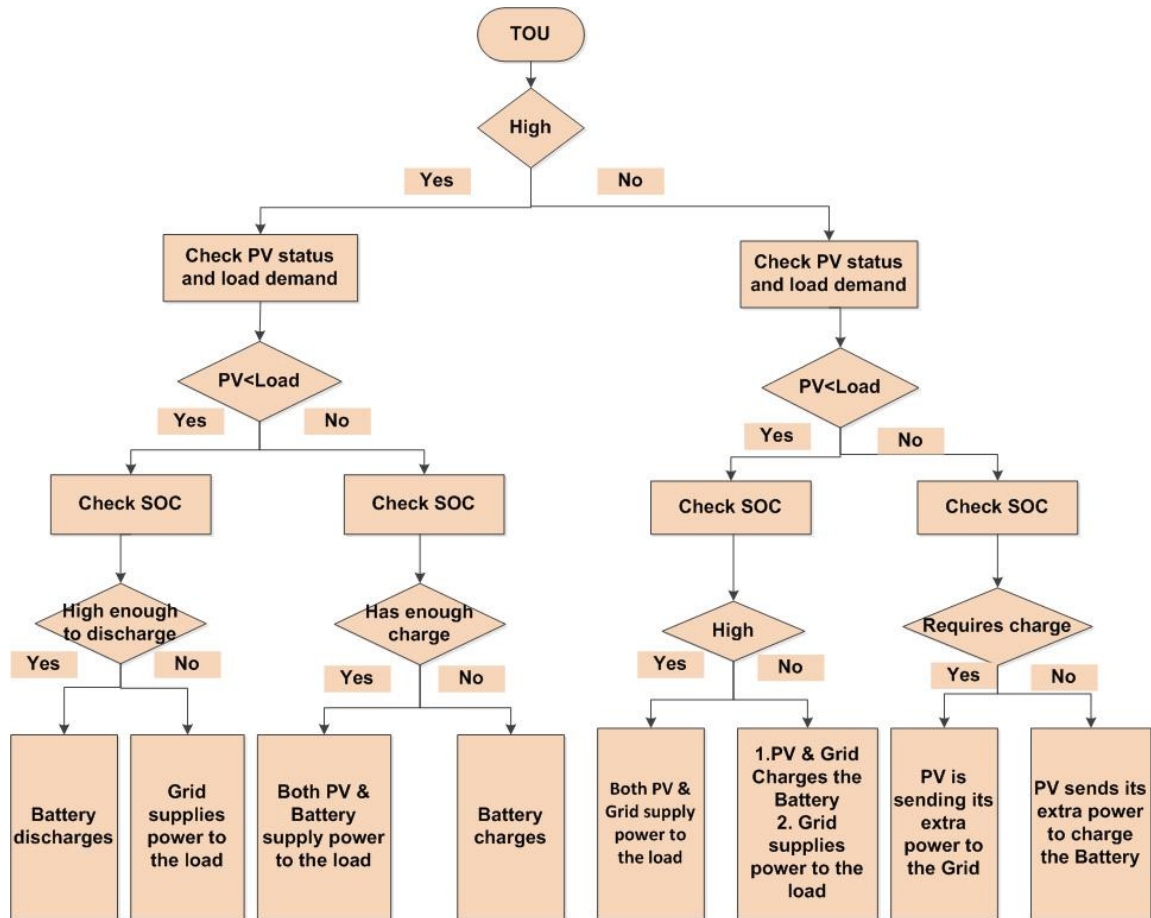


Fig. 3.3.6. Flow chart for power flow management in grid-connected mode.

This simulation is also carried out for a 2 kW system. The TOU pricing information is obtained from an internet website [18]. The simulation is completed for a 24 hour period.

As can be seen in Fig. 3.3.6, at 5:00 am in the morning when there is no solar radiation:

- TOU price is 4.6 cents/kWh
- PV current is zero
- Load Current is 2.35 A
- Battery SOC is 0.85

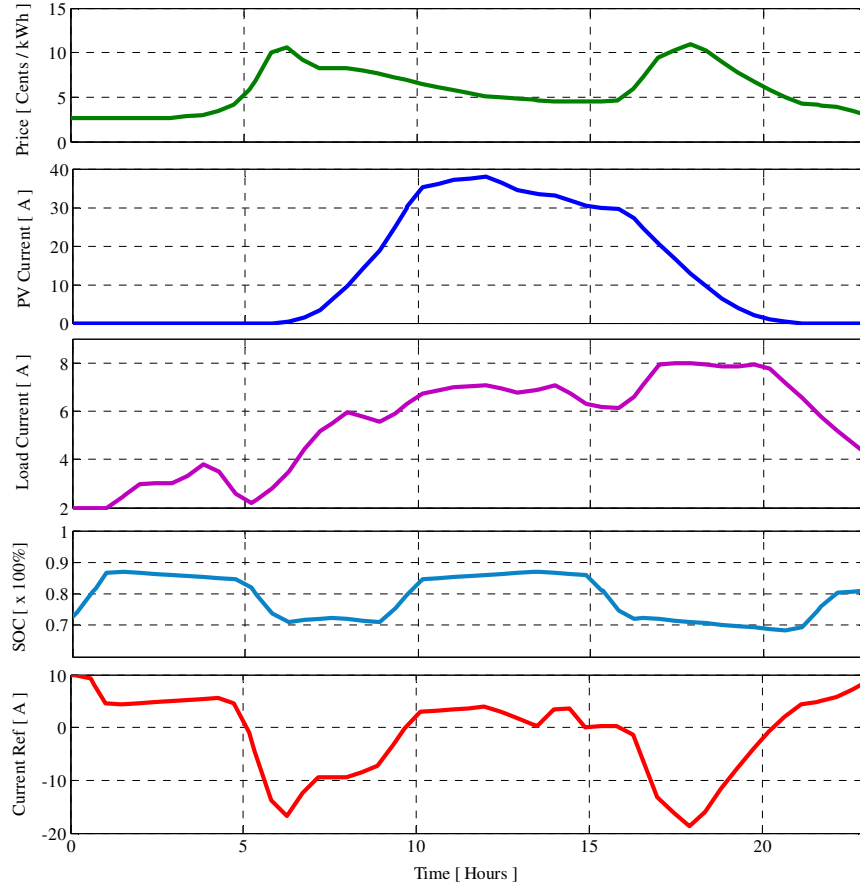


Fig. 3.3.7. Simulation results for grid-connected mode.

Since the TOU price is low at that time, PV power is  $(50 \text{ V} * 0 \text{ A}) = 0 \text{ W}$  and load power is  $(240 \text{ V} * 2.35 \text{ A}) = 564 \text{ W}$ . Hence, the load requires power from grid and, since the battery SOC is 0.85, the grid can charge the battery as per the following equation:

$$\begin{aligned}
 I_{ref} &= I_d + \left[ (I_c - I_d) * \left\{ -\frac{(P_1 - P_{low})}{(P_{up} - P_l)} + 1 \right\} \right] \\
 &= -20 + [10 - (-20)] \\
 &\quad * \left\{ -\frac{(4.6 - 2.6)}{(11.4 - 2.6)} + 1 \right\} = 3.19 \text{ A}
 \end{aligned} \tag{3.9}$$

where,  $I_d$  is the discharging limit,  $I_c$  is the charging limit,  $P_l$  is the TOU price,  $P_{low}$  is the lowest TOU price of the day, and  $P_{up}$  is the highest TOU price of the day.

At 10:00 am in the morning:

- TOU price is 6.6 cents/kWh
- PV current is 35 A
- Load Current is 6.7 A
- Battery SOC is 0.865

Since the TOU price is high, PV power is greater than load power by  $(50 \text{ V} * 35 \text{ A}) - (240 \text{ V} * 6.7 \text{ A}) = 142 \text{ W}$  and SOC is 0.865. Therefore, excess power generated by the PV can be used to charge the battery with a charging current,  $I_{ref} = (142 \text{ W} / 49.99 \text{ V}) = 2.84 \text{ A}$ .

At 3:00 pm in the afternoon:

- TOU price is 4.5 cents/kWh
- PV current is 30 A
- Load current is 6.2 A
- Battery SOC is 0.75

Since the TOU price is low at that time, but PV power is greater than load power by  $(50 \text{ V} * 30 \text{ A}) - (240 \text{ V} * 6.2 \text{ A}) = 12 \text{ W}$  and the battery SOC is 0.85, the grid will not be used and the PV can give its extra power to charge the battery with  $I_{ref} = (12 \text{ W} / 48.9 \text{ V}) = 0.24 \text{ A}$ .



At 8:00 pm:

- TOU price is 6.1 cents/kWh
- PV current is 1 A
- Load current is 8 A
- Battery SOC is 0.866

Here, the TOU price is high and the PV power is lagged by  $(1 \text{ A} * 50 \text{ V}) - (240 \text{ V} * 8 \text{ A}) = -1870 \text{ W}$ , which requires a battery discharge current of  $(-1870 \text{ W} / 50 \text{ V}) = -37.4 \text{ A}$ . However, the battery discharge limit is  $-20 \text{ A}$ , so grid power will be used and the battery will also partially discharge  $I_{ref} = -1.93 \text{ A}$  which is calculated using Eq. (3.9).

TABLE 3.3. SUMMARY OF DIFFERENT CONDITIONS IN GRID - CONNECTED MODE

TOU Price (Cents/kWh)	PV Power (W)	Load Demand (W)	Battery Voltage (V)	SOC (%)	$I_{ref}$ (A)
4.6	0	564	49.82	85.3	3.18
6.6	1750	1608	49.99	86.5	2.84
4.5	1500	1488	48.9	75	0.24
6.1	50	1920	50	86.6	-1.93

TABLE 3.3 shows the summary of grid-connected mode control with different values for TOU price, PV power, load demand, and battery SOC.

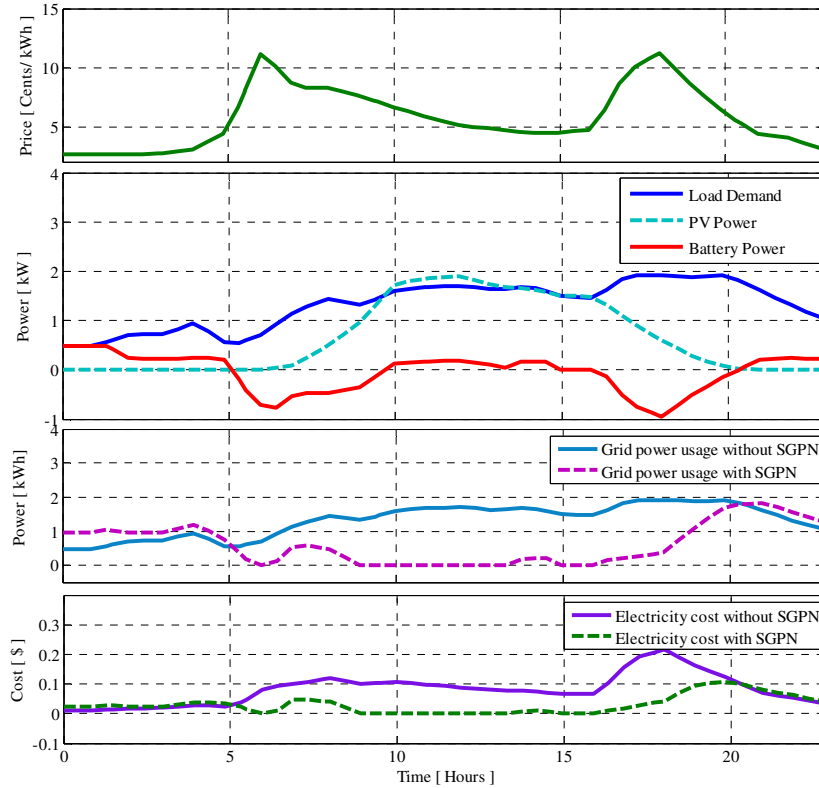
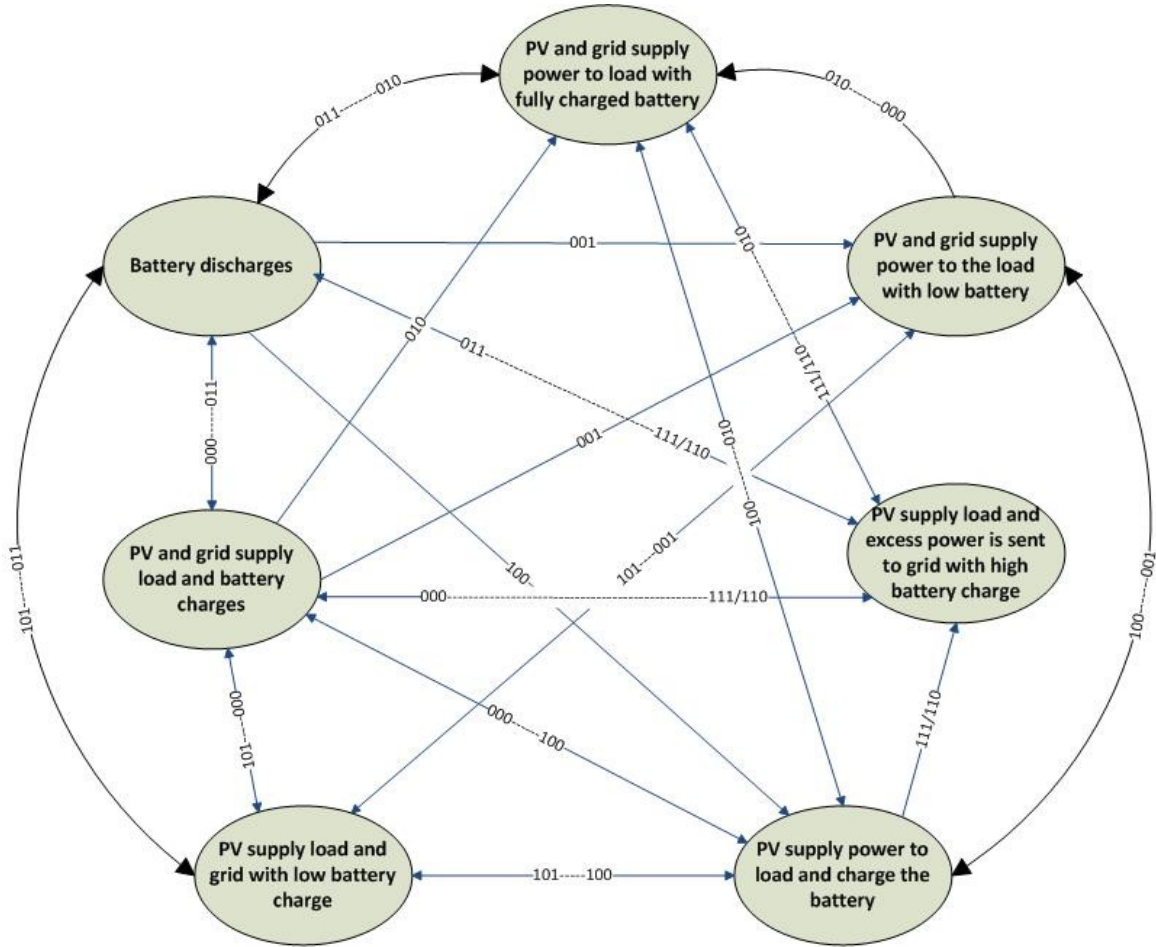


Fig. 3.3.8. Power flow, daily grid power usage and electricity cost in grid-connected mode.

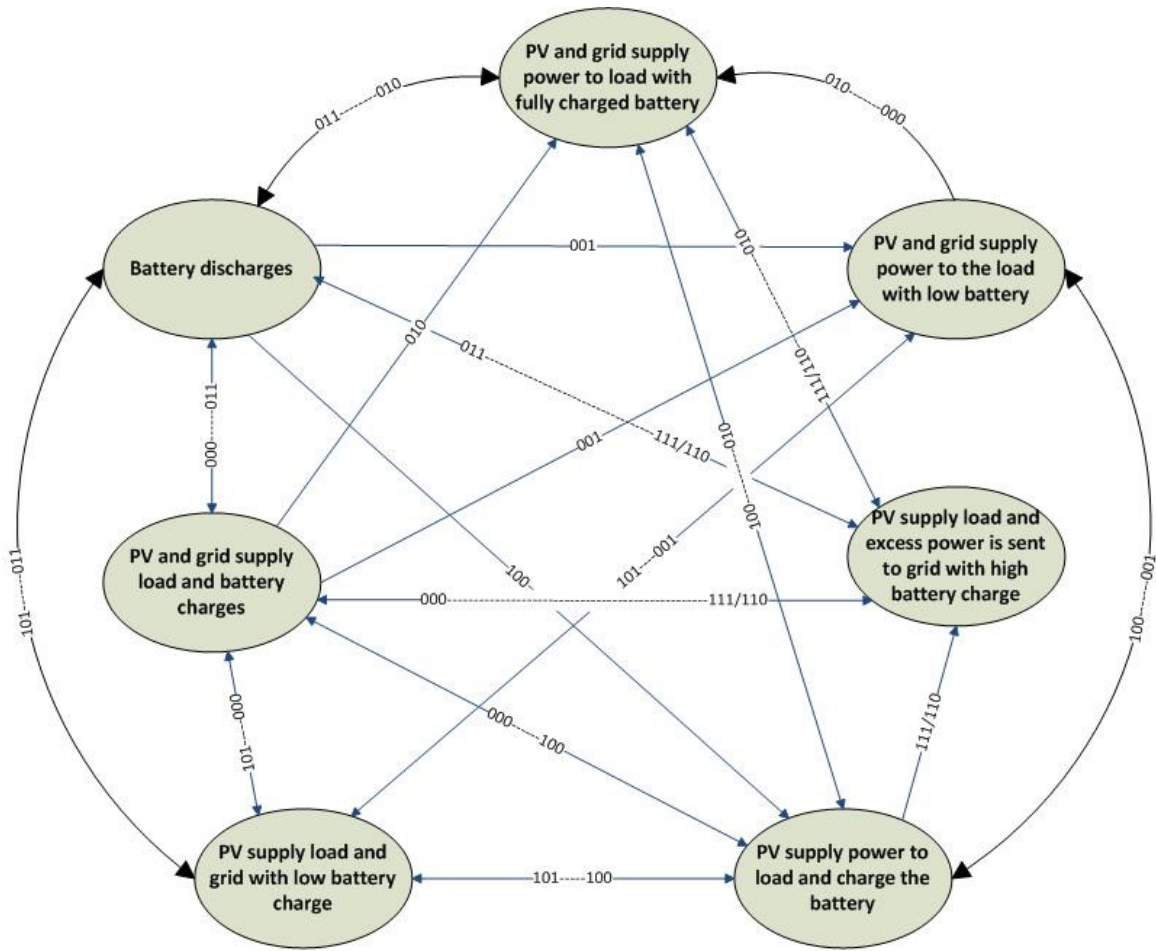
Fig. 3.3.8 shows the power flow for the PV, battery, and load. It also shows a comparison for grid power usage and electricity costs between “with SGPN system” and “without SGPN system”. Based on this power flow and 24 hour TOU pricing information, daily grid power usage is calculated. In the morning, when the TOU price is high, the load demand is compensated by the grid, and the battery is also charged using grid power. When there is available solar radiation, the load demand is compensated by PV power, and the extra PV power is used to charge the battery. At night, when there is no solar power, and the grid TOU price is also high, the battery discharges to compensate load demand. Four different scenarios have been described in this thesis paper but the control algorithm is able to perform all the scenarios according to the state

diagram shown in



- (PV<Load, SOC= low, TOU = Low) -----000
- (PV<Load, SOC= low, TOU = High) ----- 001
- (PV<Load, SOC= High, TOU = Low) ----- 010
- (PV<Load, SOC= High, TOU = High) ----- 011
- (PV>Load, SOC= low, TOU = Low) ----- 100
- (PV>Load, SOC= low, TOU = High) ----- 101
- (PV>Load, SOC= High, TOU = Low) ----- 110
- (PV>Load, SOC= High, TOU = High) ----- 000

Fig. 3.3.9. The monthly average grid power usage and electricity cost savings are also estimated from this daily usage calculation. The average monthly grid power usage and electricity cost are as shown in Fig. 3.3.10 and Fig. 3.3.11.



- (PV<Load, SOC= low, TOU = Low) -----000
- (PV<Load, SOC= low, TOU = High) ----- 001
- (PV<Load, SOC= High, TOU = Low) ----- 010
- (PV<Load, SOC= High, TOU = High) ----- 011
- (PV>Load, SOC= low, TOU = Low) ----- 100
- (PV>Load, SOC= low, TOU = High) ----- 101
- (PV>Load, SOC= High, TOU = Low) ----- 110
- (PV>Load, SOC= High, TOU = High) ----- 000

Fig. 3.3.9. State diagram for grid-connected mode

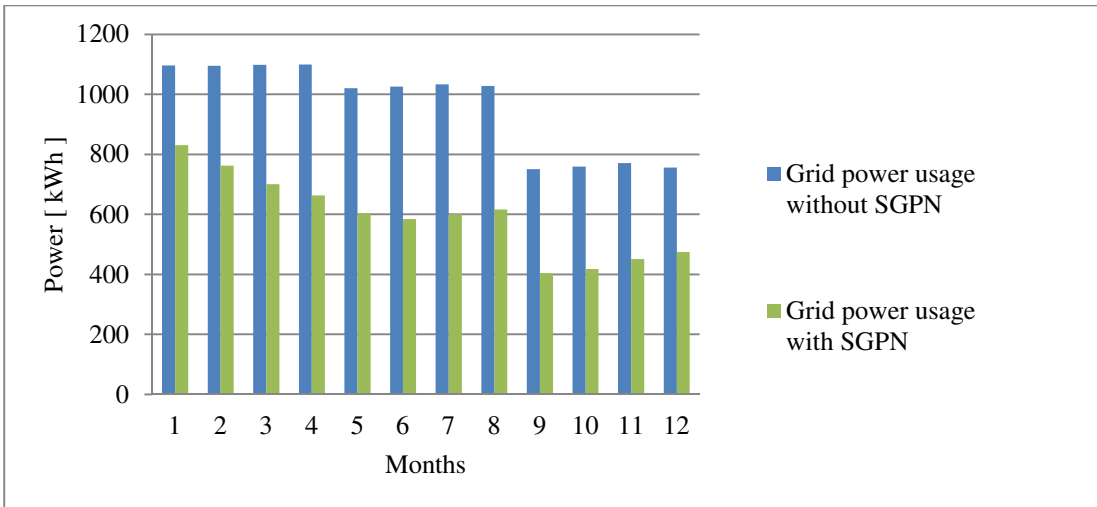


Fig. 3.3.10. Average monthly grid power usage and electricity cost comparison.

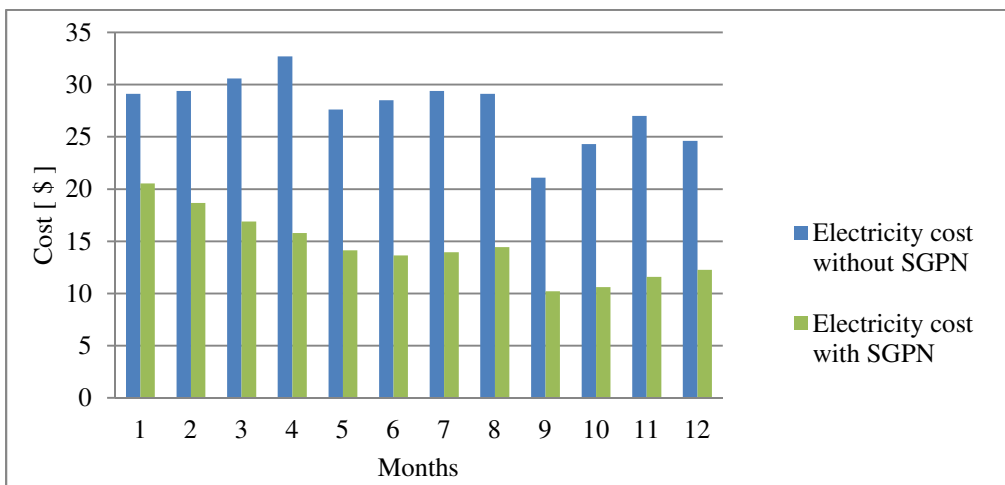


Fig. 3.3.11. Average monthly electricity cost comparison.

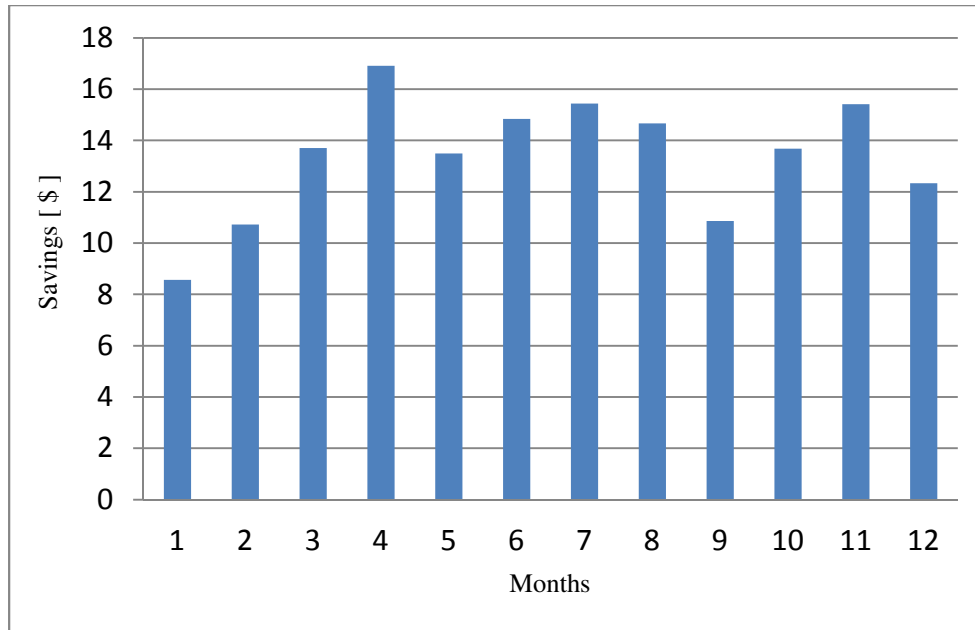


Fig. 3.3.12. Average monthly cost savings using SGPN.

Average savings per month is estimated using grid power usage and electricity cost calculation. From the Fig. 3.3.12, it is shown that, the maximum savings yield in summer season (from April to August) since the PV power is maximum at that time. The average savings per month is around \$15.00 in summer. The average yearly saving is around \$160.00.

#### 3.3.4 DSP implementation of Grid-connected Mode

The algorithm for power flow management in grid-connected mode includes TOU price, PV power, load power, and battery SOC. To acquire 24 hour TOU price information, a communication system algorithm is used along with the system level control algorithm. TOU price data is parsed directly from the internet using LabVIEW. Initially Hypertext Transfer protocol (HTTP) Client Virtual Instruments is used to get XML file from TOU price website. The parsed data is then trimmed with a string function. The Xbee communication protocol is used to transfer this data to the DSP. Twenty-four hour TOU price data is collected from an

internet source and sent to the DSP through the Labview GUI. Then, according to the algorithm, the maximum and the minimum value of TOU pricing data are extracted from the data. PV power, load demand, and battery SOC are also calculated from their voltage and current values by using voltage and current sensors. The algorithm then calculates the battery current reference for grid-connected mode based on the TOU price, battery SOC, PV power, and load demand. Fig. 3.3.13 shows the DSP output for the battery current reference for different conditions representing a 24-hour period. The battery current reference is used at the hardware level control code in the DSP to indicate when to charge and discharge the battery.

Expression	Type	Value	Address
t	int[4]	0x0000C00C@Data	0x0000C00C@Data
t[0]	int	5	0x0000C00C@Data
t[1]	int	10	0x0000C00D@Data
t[2]	int	15	0x0000C00E@Data
t[3]	int	20	0x0000C00F@Data
price	float[4]	0x0000C02C@Data	0x0000C02C@Data
price[0]	float	4.6	0x0000C02C@Data
price[1]	float	6.6	0x0000C02E@Data
price[2]	float	4.5	0x0000C030@Data
price[3]	float	6.1	0x0000C032@Data
pv	double[4]	0x0000C040@Data	0x0000C040@Data
pv[0]	double	0.0	0x0000C040@Data
pv[1]	double	35.0	0x0000C042@Data
pv[2]	double	30.0	0x0000C044@Data
pv[3]	double	1.0	0x0000C046@Data
Vpv	double	50.0	0x0000C006@Data
load	double[4]	0x0000C01C@Data	0x0000C01C@Data
load[0]	double	2.35	0x0000C01C@Data
load[1]	double	6.7	0x0000C01E@Data
load[2]	double	6.2	0x0000C020@Data
load[3]	double	8.0	0x0000C022@Data
Vload_rms	double	240.0	0x0000C004@Data
ibat	double[4]	0x0000C034@Data	0x0000C034@Data
ibat[0]	double	-10.0	0x0000C034@Data
ibat[1]	double	10.0	0x0000C036@Data
ibat[2]	double	-10.0	0x0000C038@Data
ibat[3]	double	10.0	0x0000C03A@Data
Vbat	double[4]	0x0000C024@Data	0x0000C024@Data
Vbat[0]	double	49.82	0x0000C024@Data
Vbat[1]	double	49.99	0x0000C026@Data
Vbat[2]	double	48.9	0x0000C028@Data
Vbat[3]	double	50.0	0x0000C02A@Data
SOC	double[4]	0x0000C048@Data	0x0000C048@Data
SOC[0]	double	0.8532607 (Decimal)	0x0000C048@Data
SOC[1]	double	0.8652177	0x0000C04A@Data
SOC[2]	double	0.7532609	0x0000C04C@Data
SOC[3]	double	0.8663045	0x0000C04E@Data
Iref	double[4]	0x0000C014@Data	0x0000C014@Data
Iref[0]	double	3.181818	0x0000C014@Data
Iref[1]	double	2.840568	0x0000C016@Data
Iref[2]	double	0.2453988	0x0000C018@Data
Iref[3]	double	-1.931818	0x0000C01A@Data

Fig. 3.3.13 Output results for DSP implementation of grid-connected mode control.

## CHAPTER 4

### LOAD MANAGEMENT

#### 4.1 Load Management Structure

The objective of the load management system is to reduce the grid power usage and electricity bill for the user by maximizing the usage of available PV power and by using the grid power when the TOU pricing is low. This load management is dependent on the projected power generation of the PV over a 24 hour period, load demand over the same period, and electricity TOU price.

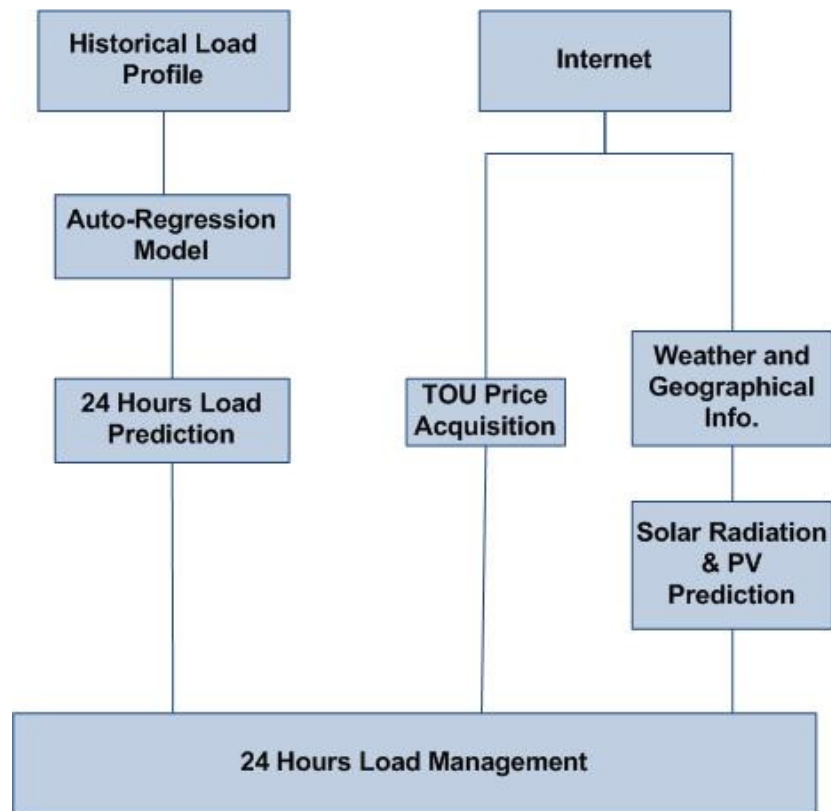


Fig. 4.1.1. Block diagram for load management structure.



Fig. 4.1.1 shows the block diagram for load management. Among the various stochastic time series (STS) methods, general auto-regression model used for twenty-four hour load demand prediction based on historical load demand. PV power is predicted based on radiation model using geographical and weather information of the local area. The weather data parsed from “*National Weather Service*” website and the TOU price data is acquired from “*Power smart pricing*” website using LabVIEW and serial communication [19],[18].

## 4.2 PV Power Prediction

The output power from a PV panel is dependent on solar radiation, temperature, solar panel area, and photoelectric conversion efficiency. Hottel’s clear sky radiation model is used to forecast the solar radiation for 24 hours [13]. Solar radiation varies from place-to-place, since it depends on geographical data such as latitude, longitude, time zone, day of the year, and the local mean time.

The solar radiation model was simulated in Matlab using geographical information for Fayetteville, Arkansas, USA.

A calculation for 12:00 pm, 9<sup>th</sup> July, 2014 is performed as follows:

Perpendicular solar radiation constant,  $G_{sc}$ :

$$G_{sc} = 1367 \frac{W}{m^2} \quad (4.1)$$

The 9<sup>th</sup> July, 2014 is the 190<sup>th</sup> day of the year, so:

$$D = 190 \quad (4.2)$$

Extra-terrestrial solar radiation,  $G_{on}$  is calculated from the perpendicular solar radiation constant and numbered day of the year:

$$G_{on} = 1367 * \left( 1 + .033 \cos\left(\frac{2\pi * 190}{365}\right) \right) = 1320.3 \frac{W}{m^2} \quad (4.3)$$

The calculation for solar declination ( $\delta$ ) and the time in minutes ( $E_t$ ) are calculated by putting the value of  $D$  into the following equations:

$$\delta = \frac{\pi}{180} * 23.45 * \sin\left(\frac{2\pi(190 + 284)}{365}\right) = 0.39 \text{ rad/s} \quad (4.4)$$

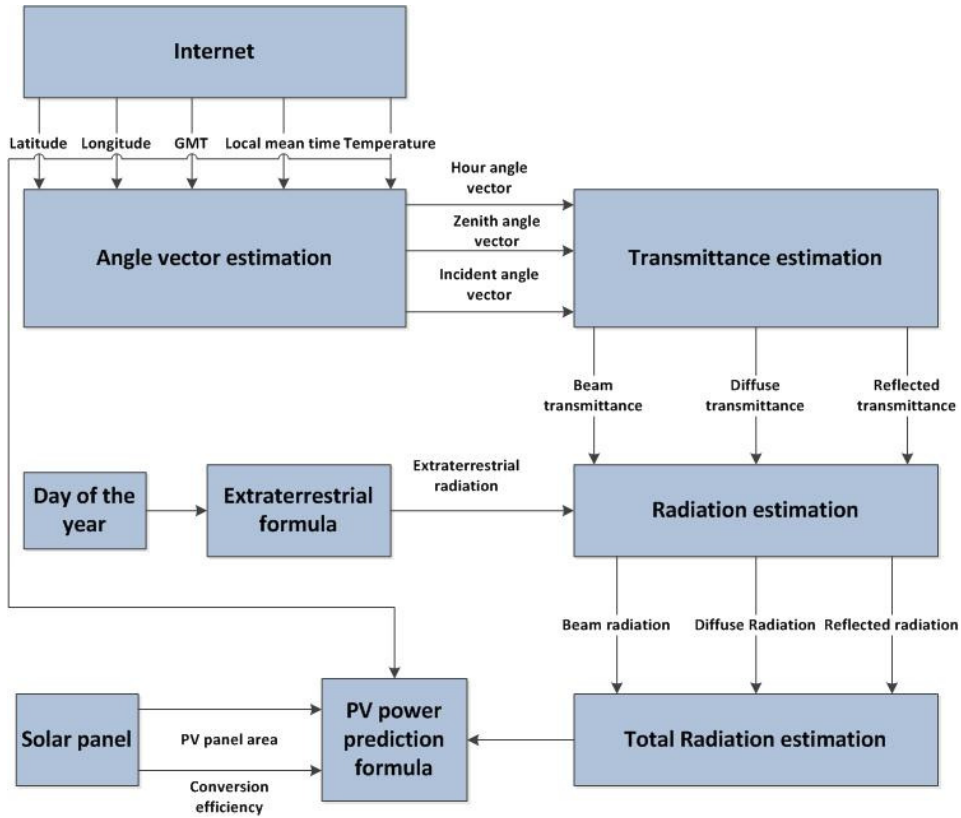


Fig. 4.2.1. PV power prediction model block diagram.

$$\begin{aligned}
E_t &= 229.1831 * (7.5e^{-5} + .001868 * \cos\theta - .032077 * \sin\theta \\
&\quad - .014615 * \cos 2\theta - 0.040849 * \sin 2\theta \\
&= -4.93 \text{ minutes}
\end{aligned} \tag{4.5}$$

where,

$$\theta = \frac{2\pi(D - 1)}{365} = 3.25 \text{ rad/s} \tag{4.6}$$

The calculation is performed for local mean time (LMT) of 12:00 pm in Fayetteville, AR.

$$LMT = 12 \tag{4.7}$$

The time zone, TZ for Fayetteville, AR is -6:

$$TZ = -6 \tag{4.8}$$

The latitude,  $\varphi$  and the longitude,  $\epsilon$  for Fayetteville, AR are  $36^\circ$  North, and  $-94^\circ$  West, respectively.

$$\varphi = 36 * \frac{\pi}{180} = 0.6238 \text{ rad/sec} \tag{4.9}$$

$$\epsilon = -94 \tag{4.10}$$

Hour angle vector, ( $\omega$ ) depends on local mean time, time zone, longitude, and time in minutes.

$$\begin{aligned}
\omega &= \frac{\pi}{180} * \left( (12 - (-6) - 12) * 15 + (-94) + \frac{-4.93}{4} \right) \\
&= -0.09 \text{ rad/sec}
\end{aligned} \tag{4.11}$$

Inclination angle of surface is  $0^\circ$ .

$$\alpha, \beta = 0 \tag{4.12}$$

The next step is to calculate the solar zenith angle vector and the incident angle vector. Both of the angle vectors vary with factors such as solar declination, latitude, inclination angle of surface, and the hour angle.

$$\begin{aligned} \cos\theta_z &= \cos(0.39) * \cos(0.6238) * \cos(-0.09) + \sin(0.39) \\ &\quad * \sin(0.6238) = 0.968 \end{aligned} \tag{4.13}$$

$$\begin{aligned} \cos\theta_s &= \sin(0.039) * \sin(0.6238) * \cos(0) - \sin(0.039) \\ &\quad * \cos(0.6238) * \sin(0) * \cos(0) + \cos(0.039) \\ &\quad * \cos(0.6238) * \cos(0) * \cos(0) + \cos(0.039) \\ &\quad * \sin(0.6238) * \sin(0) * \cos(0) * \cos(-0.09) \\ &\quad + \cos(0.039) * \sin(0) * \sin(-0.09) * \sin(0) = 0.9687 \end{aligned} \tag{4.14}$$

The terms  $a_0$ ,  $a_1$  and  $r_k$  are the constant from Hottel's model [13]. These values vary with altitude and correction factors  $r_0$ ,  $r_1$  and  $r_k$ , which depend on climate types.

TABLE 4.1.  $r_0$ ,  $r_1$ , and  $r_k$  VALUES FOR DIFFERENT CLIMATE TYPES [13]

Climate Type	$r_0$	$r_1$	$r_k$
Tropical	0.95	0.98	1.02
Mid latitude summer	0.97	0.99	1.02
Subarctic summer	0.99	0.99	1.01
Mid latitude winter	1.03	1.01	1

The climate type for Fayetteville, AR is mid-latitude summer (Since its latitude is between 23° N and 66° N). Hence, from the TABLE 4.1, the correction factors for Fayetteville, AR are 0.97, 0.99, and 1.02.

The altitude for Fayetteville, AR is 427 m:

$$A = 0.427 \text{ km} \quad (4.15)$$

$$a_0 = 0.97 * (0.4237 - 0.00821 * (6 - 0.427)^2) = 0.16 \quad (4.16)$$

$$a_1 = 0.99 * (0.5055 - 0.00595 * (6.5 - 0.427)^2) = 0.717 \quad (4.17)$$

$$k = 1.02 * (0.2711 - 0.1858 * (2.5 - 0.427)^2) = 0.358 \quad (4.18)$$

Atmospheric transmittance ( $t_b$ ), diffuse transmittance ( $t_d$ ) and reflected transmittance ( $t_r$ ) are calculated from the following equations by using altitude and the constant values.

$$t_b = (0.16 + 0.717 * e^{\left(\frac{-0.358}{0.986}\right)}) = 0.6596 \quad (4.19)$$

$$t_d = 0.271 - 0.294 * 0.6596 = 0.0771 \quad (4.20)$$

$$t_r = 0.271 - 0.706 * 0.6596 = 0.7367 \quad (4.21)$$

Average reflectance of the ground ( $\rho$ ) is 0.2, so:

$$\rho = 0.2 \quad (4.22)$$

Beam radiation ( $G_{bt}$ ), reflected radiation ( $G_{rt}$ ), and diffuse radiation ( $G_{dt}$ ) are calculated using the following equations using each transmittance value:

$$G_{bt} = 1320 * 0.6596 * 0.986 = 844.88 \text{ W/m}^2 \quad (4.23)$$

$$G_{dt} = 1320 * 0.0771 * 0.986 * \frac{1 + \cos 0}{2} = 98.71 \text{ W/m}^2 \quad (4.24)$$

$$G_{rt} = 0.2 * 1367 * 0.7367 * 0.986 * \frac{1 + \cos 0}{2}$$

$$= 188.72 \text{ W/m}^2$$
(4.25)

The total solar radiation,  $G_t$  is the summation of beam radiation, reflected radiation, and the diffuse radiation.

$$G_t = 844.88 + 98.71 + 188.72 = 1132.3 \text{ W/m}^2$$
(4.26)

The simulation is performed for a 24 hour period for three different times of the year: February, July, and November.

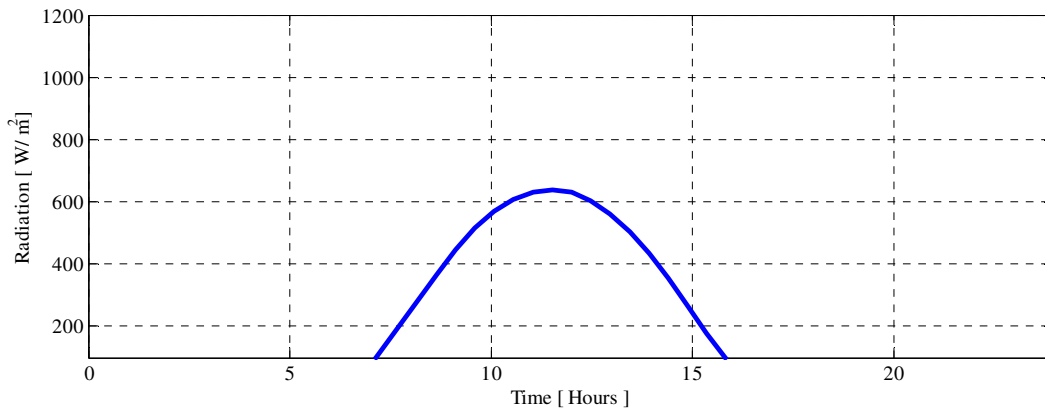


Fig. 4.2.2. Twenty-four hour solar radiation prediction for the month of February.

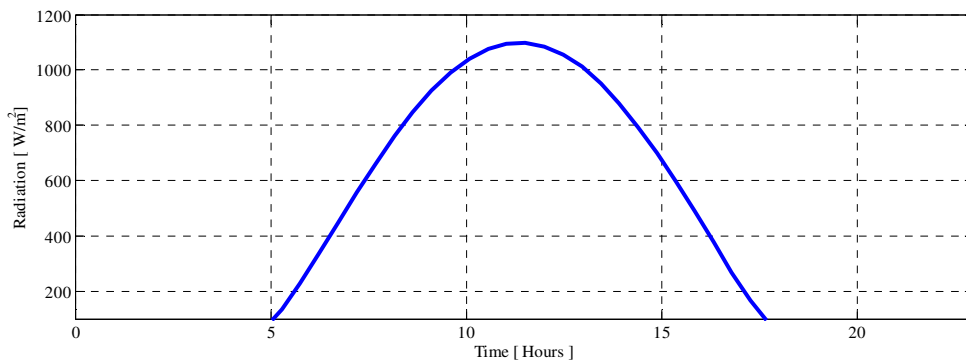


Fig. 4.2.3. Twenty-four hour solar radiation prediction for the month of July.

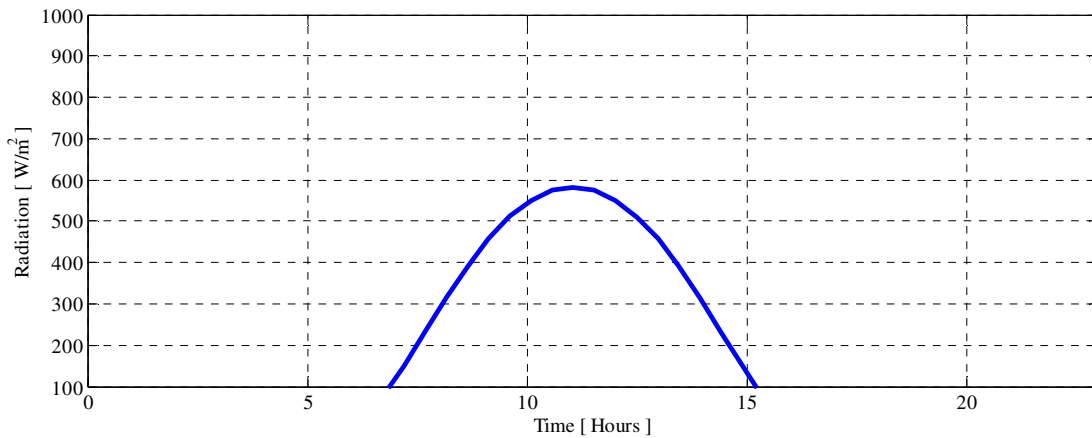


Fig. 4.2.4. Twenty-four hour solar radiation prediction for the month of November.

The total solar radiation in July (shown in Fig. 4.2.3) is greater than the radiation in February and November (shown in Fig. 4.2.3. and Fig. 4.2.4, respectively). The predicted power from the solar panel is calculated using predicted solar radiation in conjunction with other information such as solar panel area, conversion efficiency, and temperature. The area of the solar panel and photoelectric conversion efficiency is calculated from the solar panel datasheet. This SGPN system is designed to have eight 327 W solar panels in series to yield approximately 2.5 kW power from the PV. The photoelectric conversion efficiency is 20%. The equation for solar panel area is given by:

$$\text{solar panel area} = \frac{\text{PV power}}{\text{solar radiation} * \text{conversion efficiency}} \quad (4.27)$$

$$\text{Solar panel area, } A = \frac{(327 * 8)W}{(1132.3 * 0.2)W/m^2} = 11.5 \text{ m}^2 \quad (4.28)$$

TABLE 4.2. DAY TYPE INDEX VALUES FOR DIFFERENT WEATHER [13]

Forecasted Weather Type	Day Type Index
Clear	0.9
Partly Cloudy	0.7
Cloudy/ Foggy	0.3
Rainy/Snowy/Overcast	0.1

The predicted power from the PV panel for a clear day is then calculated using the equation given by:

$$P_{pv} = \sigma * A_{pv} * 0.9 * G_t [1 - 0.005(T_c - 25)] = 2.23 \text{ kW} \quad (4.29)$$

Here, the temperature  $T_c = 34 \text{ }^\circ\text{C}$  is used for calculation. The simulation for PV prediction is also performed for 24 hour periods in February, July, and November since the temperatures are different for each of these months a year.



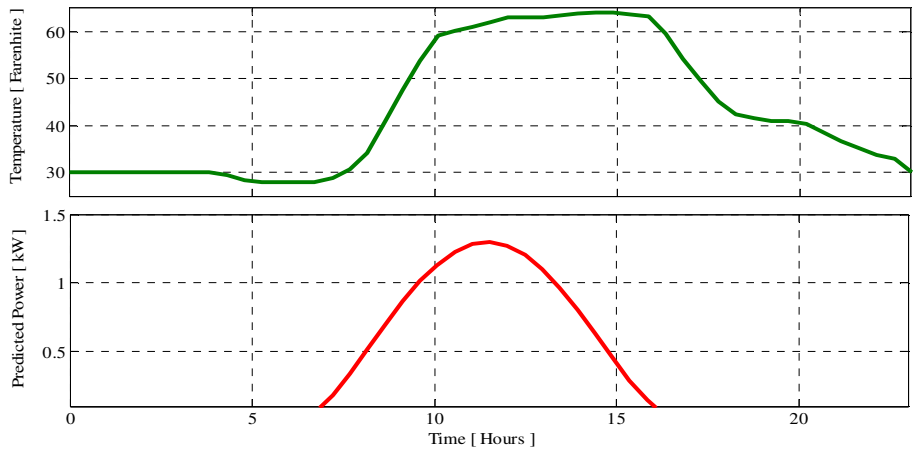


Fig. 4.2.5. Twenty-four hour PV power prediction for the month of February.

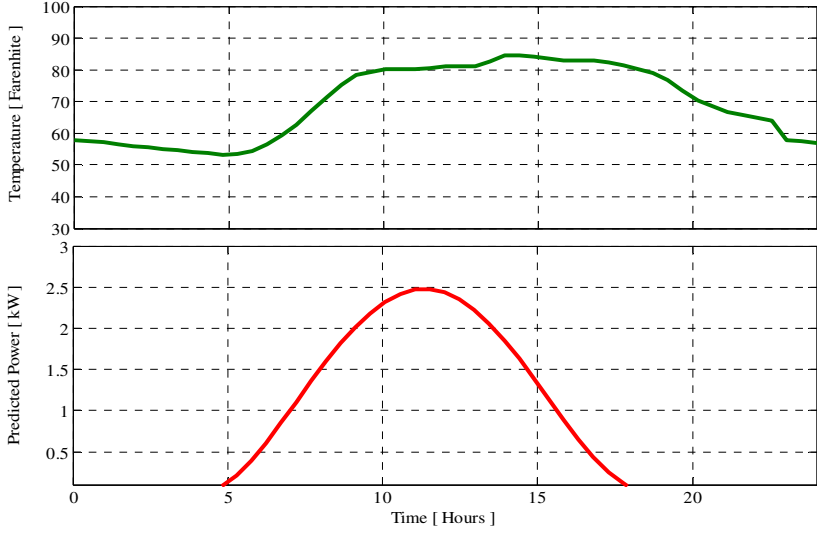


Fig. 4.2.6. Twenty-four hour PV power prediction for the month of July.

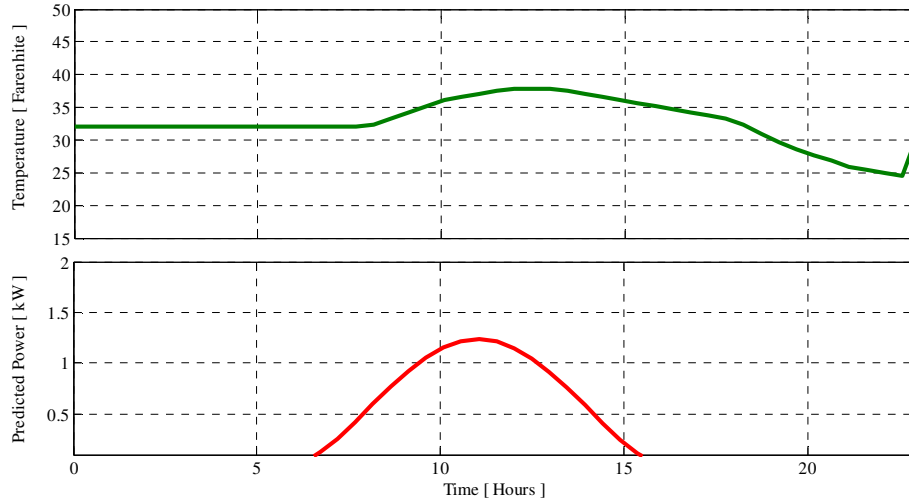


Fig. 4.2.7. Twenty-four hour PV power prediction for the month of November.

The Fig. 4.2.5 shows PV power simulation results for a 24 hour period in February. Here, the PV can generate a maximum of approximately 1.5 kW, since the radiation is not that high and PV power generation begins at 7:00 am and lasts until 4:00 pm. Fig. 4.2.6 shows the same simulation for July. Here, the PV will generate a maximum of approximately 2.2 kW, since the solar radiation is higher in summer. PV power generation starts at 5 am and lasts until 6:30 pm. In November, as shown in Fig. 4.2.7, the PV power production is not as high and produces power only from 7:00 am to 3:30 pm.

### 4.3 Load Prediction

#### 4.3.1 Electrical Load Prediction Using Auto-regression Model

For the auto-regression model, the current load depends on the previous load demand and the auto-regression coefficients. The equation for auto-regression model is given by:

$$y(t) = a(t) + \phi_1 y(t - 1) + \phi_2 y(t - 2) + \dots + \phi_p y(t - p) \quad (4.30)$$

where  $y(t)$  is the value for the current load,  $\phi_1, \phi_2, \dots, \phi_p$  are the regression coefficients, and  $y(t-1), y(t-2), \dots, y(t-p)$  are the value previous hour's load demands. The initial step is to find the regression coefficient values using historical load demand data in the regression analysis. Historical US residential load demand data for three different times of a year (February, July, and November) are considered for this auto-regression model. These data sets are taken “Connecticut Light & Power” website [20]. This website provides hourly historical load demand for twelve months. Separate load demand data for weekdays and weekends are collected from this website.

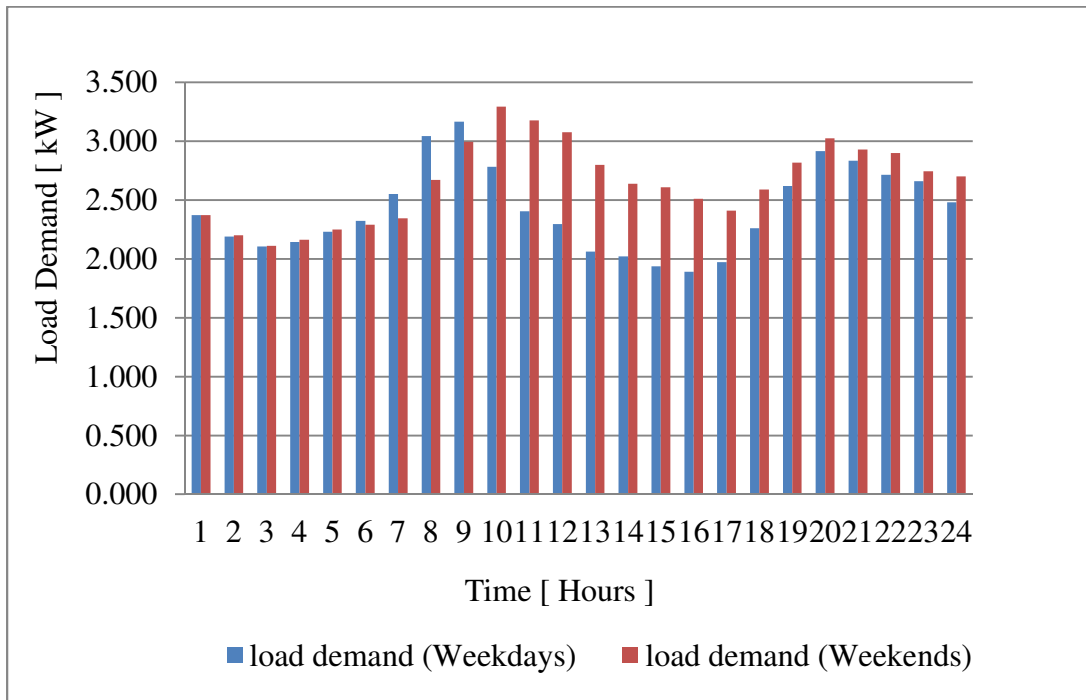


Fig. 4.3.1. Twenty-four hour average historical load demand for the month of February [20].

These historical data sets are used in an MS Excel regression analysis to calculate the regression coefficients.

TABLE 4.3. REGRESSION COEFFICIENTS FOR AUTO REGRESSION MODEL ( FEBRUARY WEEKDAYS)

$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$	$a_{10}$
1.574	1.204	-0.928	0.403	-0.169	-0.133	0.113	-0.339	0.219	-0.27	0.248

TABLE 4.4. REGRESSION COEFFICIENTS FOR AUTO REGRESSION MODEL (FEBRUARY WEEKENDS)

$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$	$a_{10}$
1.13	1.289	-0.326	-0.44	0.488	-0.233	-0.294	0.018	0.244	0.365	-0.343

The regression coefficients for load demand profiles in July and November are calculated in a similar way and are shown in Table 4.5, TABLE 4.6, Table 4.7, and TABLE 4.8 respectively.

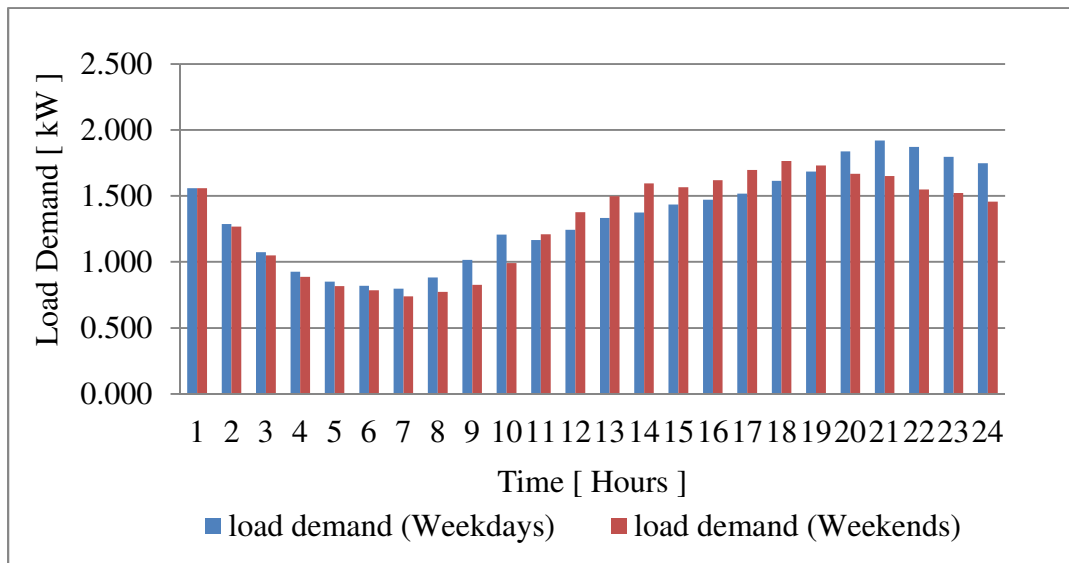


Fig. 4.3.2. Twenty-four hour average historical load demand for the month of July [20].

TABLE 4.5. REGRESSION COEFFICIENTS FOR AUTO REGRESSION MODEL (JULY WEEKDAYS)

<b>a<sub>0</sub></b>	<b>a<sub>1</sub></b>	<b>a<sub>2</sub></b>	<b>a<sub>3</sub></b>	<b>a<sub>4</sub></b>	<b>a<sub>5</sub></b>	<b>a<sub>6</sub></b>	<b>a<sub>7</sub></b>	<b>a<sub>8</sub></b>	<b>a<sub>9</sub></b>	<b>a<sub>10</sub></b>
0.581	1.093	-0.169	0.217	-0.153	-0.265	-0.232	-0.07	0.331	0.049	0.305

TABLE 4.6. REGRESSION COEFFICIENTS FOR AUTO REGRESSION MODEL (JULY WEEKENDS)

<b>a<sub>0</sub></b>	<b>a<sub>1</sub></b>	<b>a<sub>2</sub></b>	<b>a<sub>3</sub></b>	<b>a<sub>4</sub></b>	<b>a<sub>5</sub></b>	<b>a<sub>6</sub></b>	<b>a<sub>7</sub></b>	<b>a<sub>8</sub></b>	<b>a<sub>9</sub></b>	<b>a<sub>10</sub></b>
0.47	1.21	-0.183	-0.464	-0.248	-0.14	-0.538	-0.539	0.069	0.29	-0.255

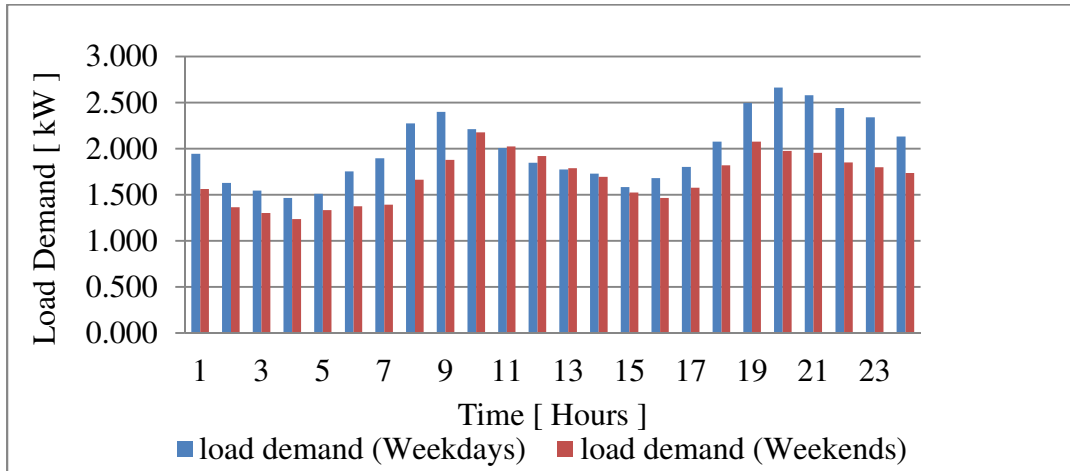


Fig. 4.3.3. Twenty-four hour average historical load demand for the month of November [20].

TABLE 4.7. REGRESSION COEFFICIENTS FOR AUTO REGRESSION MODEL (NOVEMBER WEEKDAYS)

<b>a<sub>0</sub></b>	<b>a<sub>1</sub></b>	<b>a<sub>2</sub></b>	<b>a<sub>3</sub></b>	<b>a<sub>4</sub></b>	<b>a<sub>5</sub></b>	<b>a<sub>6</sub></b>	<b>a<sub>7</sub></b>	<b>a<sub>8</sub></b>	<b>a<sub>9</sub></b>	<b>a<sub>10</sub></b>
0.709	1.2919	-0.527	-0.007	-0.17	0.329	-0.413	0.04	0.099	-0.133	0.311

TABLE 4.8. REGRESSION COEFFICIENTS FOR AUTO REGRESSION MODEL (NOVEMBER WEEKENDS)

$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$	$a_{10}$
0.618	1.207	-0.039	-0.776	0.28	0.408	-0.66	-0.081	0.54	0.036	-0.48

After the regression coefficients are calculated, the next step is to use these regression coefficients with previous hourly load demands in the auto regression model to predict the next 24 hours of load demand. The simulation is performed to predict the load in a 24 hour period using an auto-regression model in Matlab with the calculated regression coefficients.

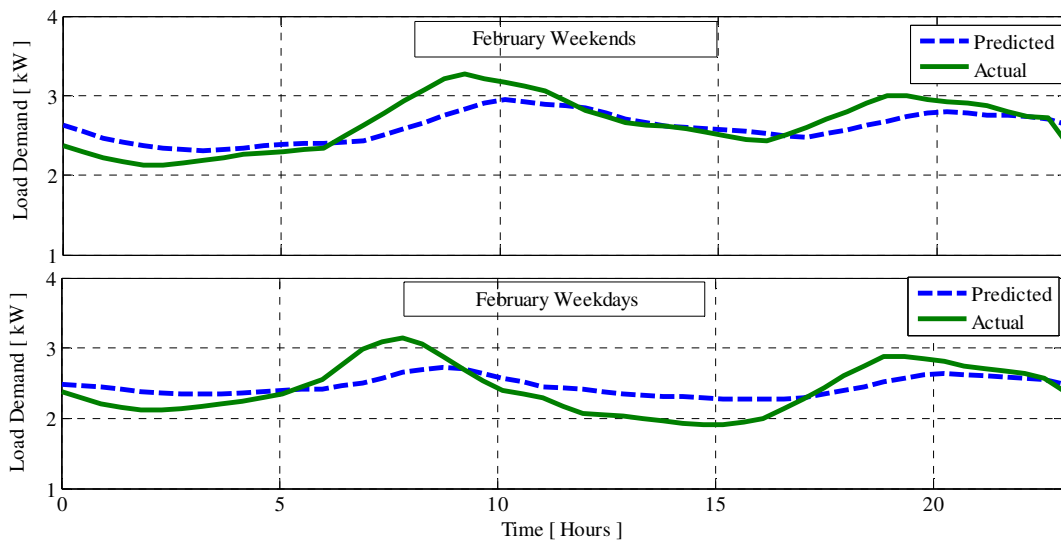


Fig. 4.3.4. Twenty-four hour load demand prediction for the month of February.

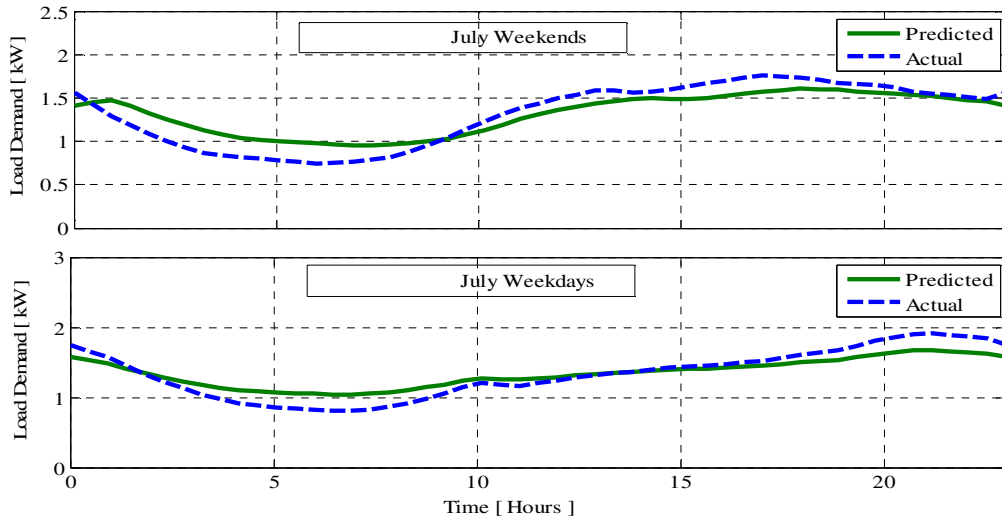


Fig. 4.3.5. Twenty-four hour load demand prediction for the month of July.

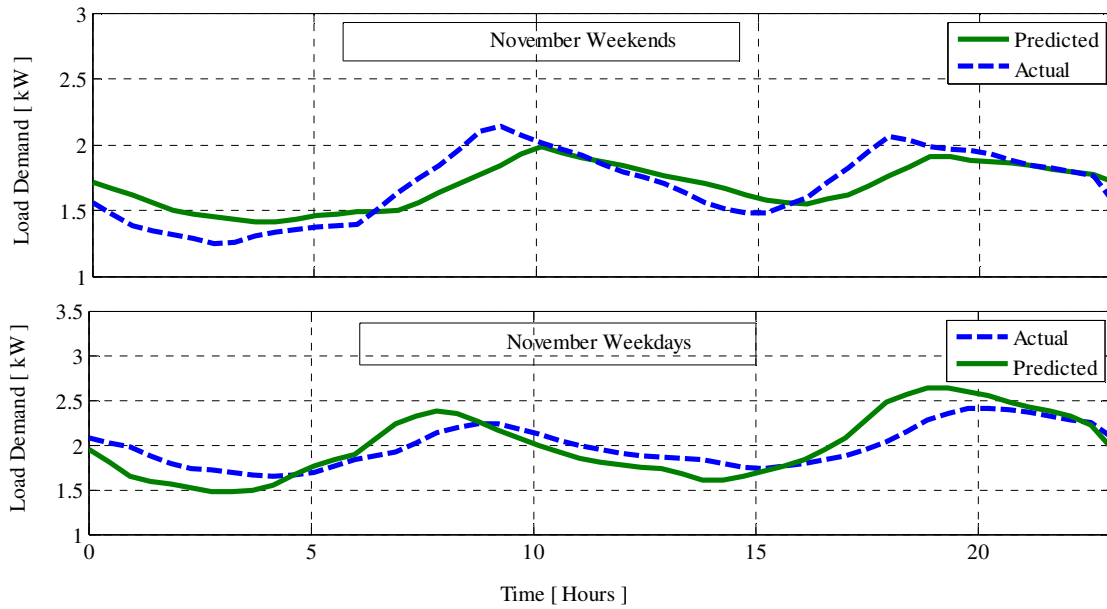


Fig. 4.3.6. Twenty-four hour load demand prediction for the month of November.

Table 4.9. LOAD DEMAND PREDICTION ACCURACY

Month	Accuracy	
	Weekdays	Weekends
February	93.85%	90.49%
July	87.25%	89.11%
November	92.49%	91.5%

Table 4.9. shows the calculated accuracy for predicted load demand for different seasons. For weekdays in February, the prediction for load demand is 93.85% accurate whereas in weekends the accuracy is 90.49%. For July, the prediction is 87.25% accurate for weekdays and 89.11% for weekends. The prediction accuracy for November load demand is also more than 90%.

#### 4.4 DSP Implementation of PV & Load Prediction

Twenty-four hour weather data, along with geographical information (latitude, longitude, time zone, and local time) is acquired from a weather website using LabVIEW using the same method as used for TOU pricing. Xbee communication is used to send these data sets to the DSP. When the DSP receives the weather information, it runs the code to calculate the predicted PV power. It also calculates the predicted load demand by retrieving previous load demand for the home from memory. Fig. 4.4.1 shows the DSP output for the PV predicted power and predicted load demand that are calculated from acquired weather, geographical information, and previous load demand.



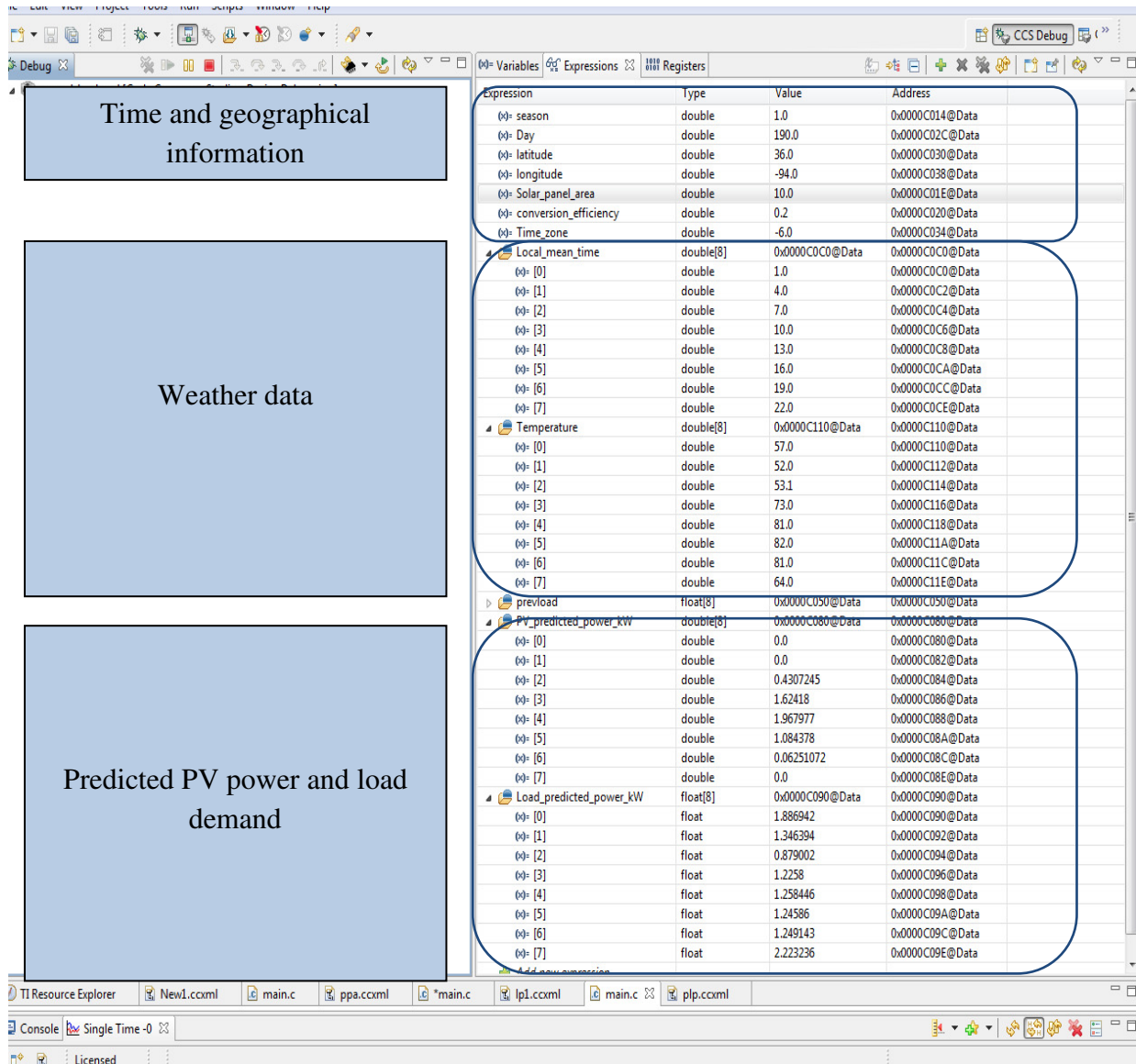


Fig. 4.4.1. Output results for DSP implementation of PV prediction and load prediction.

## 4.5 Types of Load Management

Twenty-four hour load management is performed for three times of the year based on the calculation of PV prediction, load prediction, and TOU price. The load management is categorized into two sections. These sections are described in the following sub-sections.

### 4.5.1 Automatic Turn On/Off Load Management

Automatic turn on/off load management makes a default schedule for twenty-four hours for load usage. It also requires user interaction to turn on/off the load to save electricity. However, the users get an indication of when to turn on or turn off the load depending on PV power prediction, load demand prediction, and TOU price. Three household loads are chosen for simulation whose power ratings are 1 kW, 0.6 kW, and 0.4 kW. The simulation is performed for three different months in a year: February, July, and November.

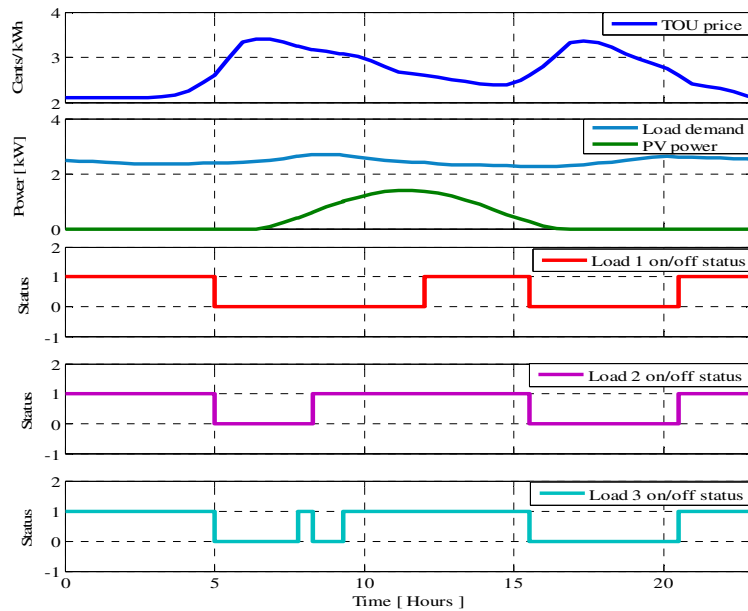


Fig. 4.5.1. Simulation results for automatic turn on/off load management for the month of February.

Fig 4.5.1 shows the automatic load management simulation results obtained for the month of February. In February, the TOU price reaches a peak in the morning (from 6:00 am to 10:00 am) and in the evening (from 5:00 pm to 7:00 pm). PV power is available from 7:00 am to 5:00 pm. To reduce electricity cost, each of the three loads can be turned on when the TOU price is low in the morning. When the TOU price goes high, the loads can be turned off until the PV power is available to supply the loads. The loads can then be turned on again at night when the TOU is low.

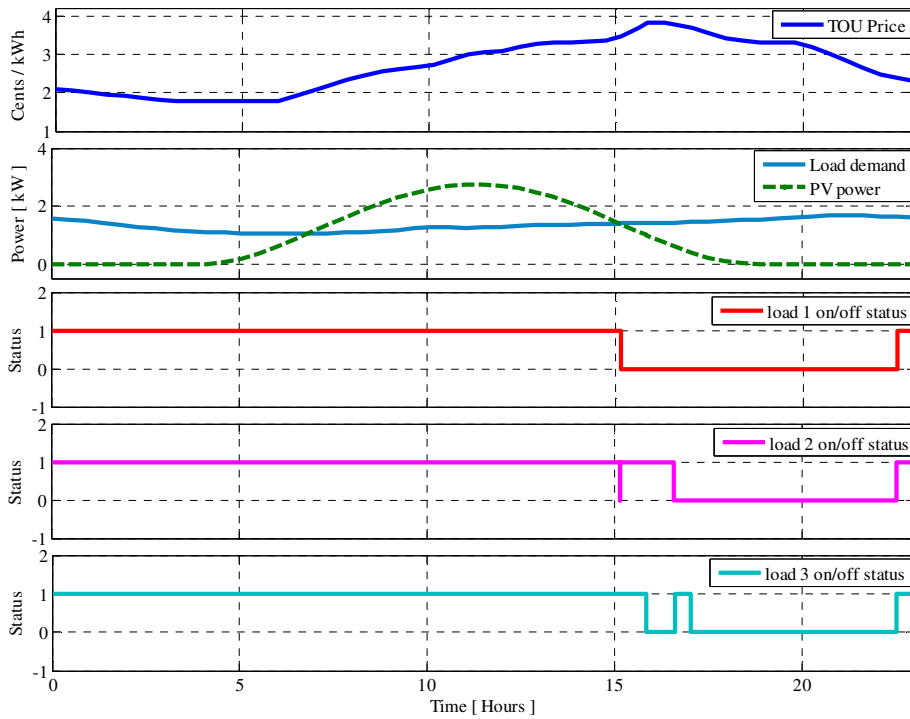


Fig. 4.5.2. Simulation results for automatic turn on/off load management during the month of July.

Fig 4.5.2 shows the automatic load management simulation results obtained for the month of July. Here, the TOU price remains low 11:00 am in the morning. After 11:00 am, the price goes high till 9:00 pm. The PV power is high in July, since the solar radiation is very high and

available from 5:00 am to 6:00 pm. To reduce electricity cost, each of the three loads can be turned on when the TOU price is low until 11:00 am. When the TOU price goes high after 11:00 am, the available PV power supplies the loads. The loads can remain turned on until 6:00 pm. The loads can be turned off again at night when the TOU is high and there is available PV power.

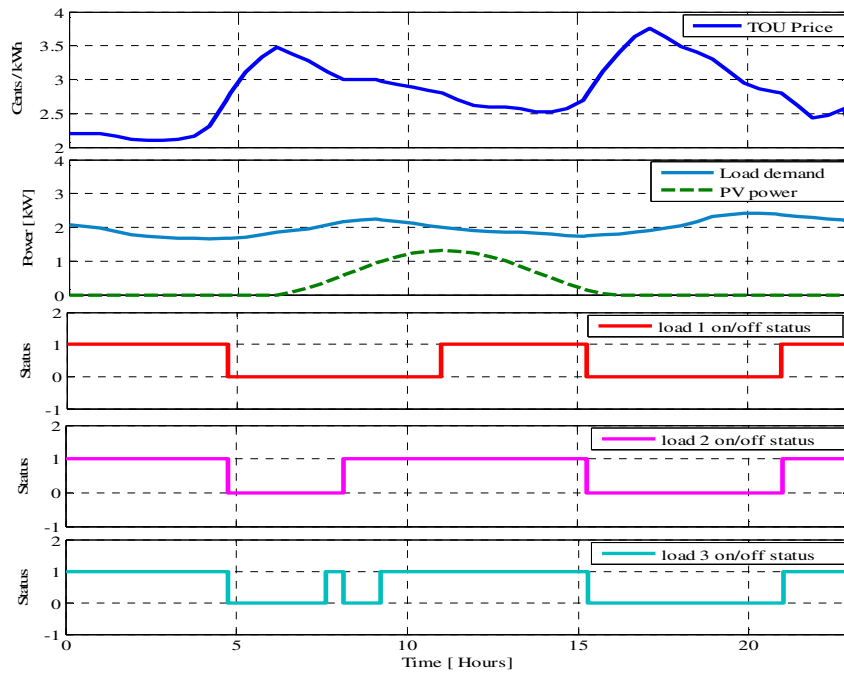


Fig. 4.5.3. Simulation results for automatic turn on/off load management for the month of November.

Fig 4.5.3 shows the automatic load management simulation results obtained for the month of November. In November, the TOU price reaches a peak in the morning (from 5:00 am to 8:00 am) and in the evening (from 4:00 pm to 7:00 pm). PV power is available from 7:00 am to 4:30 pm. To reduce electricity cost, each of the three loads can be turned on when the TOU price is low in the morning. When the TOU price goes high, the loads can be turned off until the PV

power is available to supply the loads. The loads can be turned on again at night when the TOU price is low.

#### 4.5.2 Labview GUI Demonstration for Load Management

A Labview GUI is constructed for manual load management demonstration. The TOU price and weather data is collected from the internet using Labview and sent to DSP using Xbee communication protocol as shown in Fig. 4.5.4.

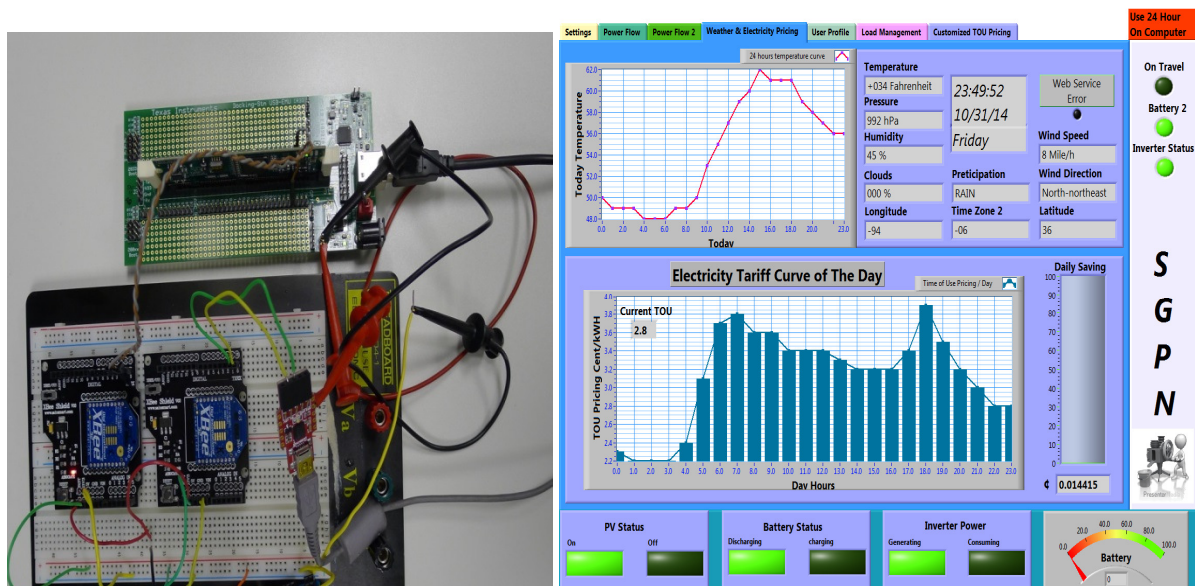


Fig. 4.5.4. Hardware connections and Labview GUI for weather and TOU data acquisition.

PV prediction and load prediction models are programmed in Labview. Weather data and TOU price is acquired at the beginning of the day. Initially these data are parsed from internet using Labview and sent to DSP using Xbee communication protocol. This forecasted weather data is used with the predicted solar radiation and solar panel parameters to calculate the 24 hours of PV predicted power. The load demand for the previous day is also collected for 24 hour load demand prediction. The Matlab simulation is coded into Labview for the manual load

management. From this GUI interface the user can see which time will be better to turn on/off the loads to minimize the cost of electricity. Fig. 4.5.5 and Fig. 4.5.6 show the GUI interface constructed for the month of February, July, and November.

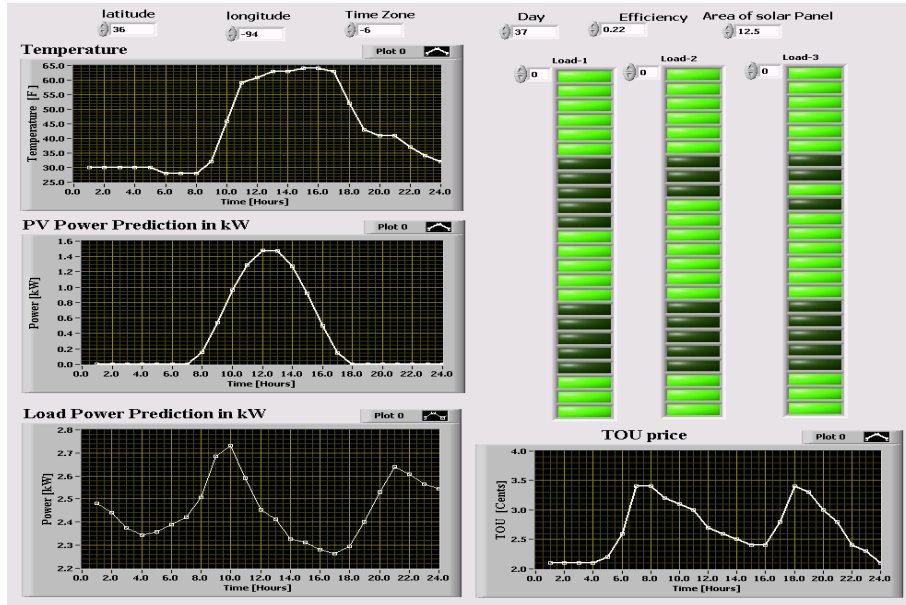
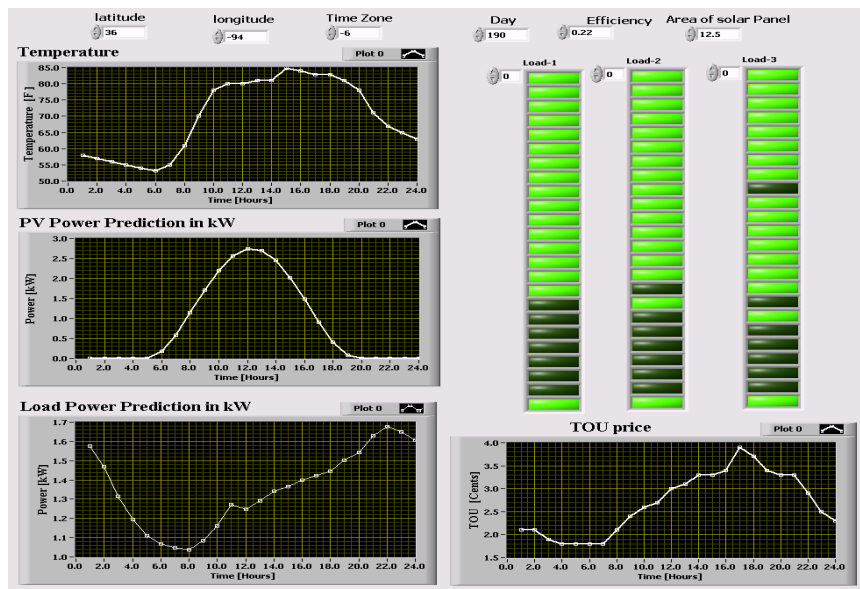
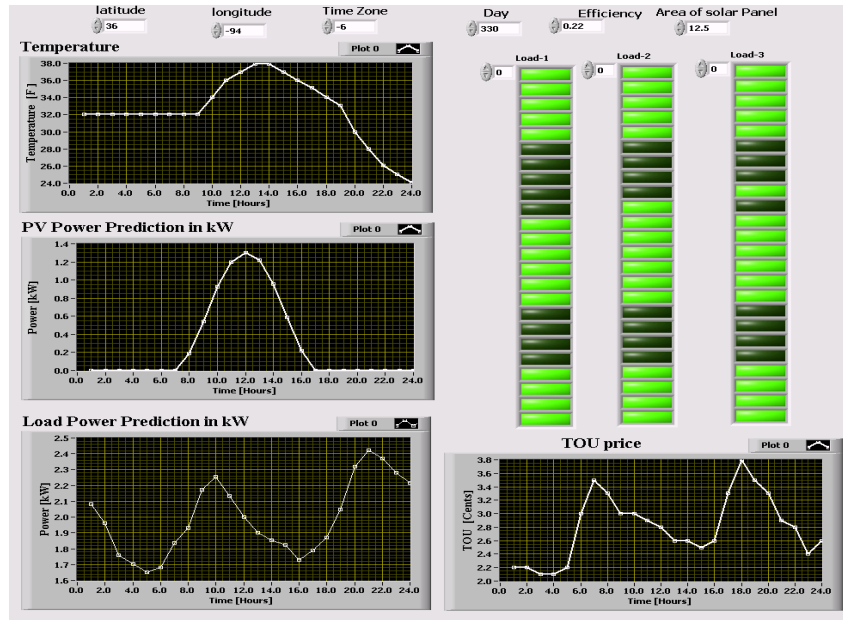


Fig. 4.5.5. LabVIEW GUI interface automatic load management (February).



(a)



(b)

Fig. 4.5.6. LabVIEW GUI interface for automatic load management for (a) July and (b) November.

### 4.5.3 Automatic Load Management with Temperature Control

Automatic load management requires no user interaction. A house space heater was used as the load in this thesis. However, other loads with specific load model can also be added. But the algorithm will be same since it is based on TOU price, PV power prediction, and load demand prediction. A Matlab thermodynamic model of a house heating system is considered for this thesis [21]. The system automatically turns the heater on or off depending on the predicted PV power, predicted load demand, and TOU price. It also controls the indoor temperature of the house. The parameters used for this house heating system are given in TABLE 4.10.

TABLE 4.10. SPECIFICATIONS FOR HOUSE SPACE HEATER [21]

Parameters	Value
Length of the house, $L_{house}$	30 m
Width of the house, $W_{house}$	10m
Height of the house, $H_{house}$	4m
Roof pitch	40°
Number of windows, $N_{win}$	6
Height of the windows, $H_{win}$	1m
Width of the windows, $W_{win}$	1m
$C_p$ value of air	1005.4 J/kg-K
Heater air temperature	50°F
Air flow rate	3600 kg/hr
Density of air, $\delta$	1.2250 kg/m <sup>3</sup>
Temperature set point	70°F

Total area for the windows,  $A_{win}$  is calculated using the following equation:

$$A_{win} = N_{win} * H_{win} * W_{win} \quad (4.31)$$

$$A_{win} = 6 * 1 * 1 = 6 \text{ m}^2 \quad (4.32)$$

Total area of the wall  $A_{wall}$  for the house is calculated using the following:

$$\begin{aligned}
 A_{wall} &= (2 * L_{house} * H_{house}) + (2 * W_{house} * H_{house}) \\
 &+ \left( 2 * \left( \frac{1}{\cos \frac{Pitch \text{ roof}}{2}} \right) * W_{house} * L_{house} \right) \\
 &+ \tan(\text{pitch roof}) * W_{house} - W_{win} = 960 \text{ m}^2
 \end{aligned} \quad (4.33)$$



The next step is to calculate the equivalent thermal resistance,  $R_{eq}$ . The equation for equivalent thermal resistance is calculated using the following equation. The terms  $k_{wall}$  and  $k_{win}$  are two constants whose values are 136.8 J/hr/m/C and 2808 J/hr/m/C, respectively.

$$\text{wall resistance, } R_{wall} = \frac{L_{wall}}{k_{wall} * A_{wall}} = 1.52 \times 10^{-6} \text{ Jhr}^{-1}m^{-1}C^{-1} \quad (4.34)$$

$$\text{window resistance, } R_{win} = \frac{L_{win}}{k_{win} * A_{win}} = 5.93 \times 10^{-7} \text{ Jhr}^{-1}m^{-1}C^{-1} \quad (4.35)$$

And the equivalent thermal resistance of the house is:

$$R_{eq} = \frac{R_{wall} * R_{win}}{R_{wall} + R_{win}} = 4.26 \times 10^{-7} \text{ Jhr}^{-1}m^{-1}C^{-1} \quad (4.36)$$

The next step is to calculate the total internal air mass,  $M$ . The equation for calculating the internal air mass,  $M$  is given by:

$$M = (L_{house} * W_{house} * H_{house} + \tan(\text{pitch roof}) * W_{house} * L_{house}) \quad (4.37)$$

The Matlab house space heater model is simulated by using calculated geometry, thermal properties, thermal resistance of the house, temperature set point, and outdoor temperature. The space heater is controlled to turn on/off depending on PV predicted power, predicted load demand, and TOU price.

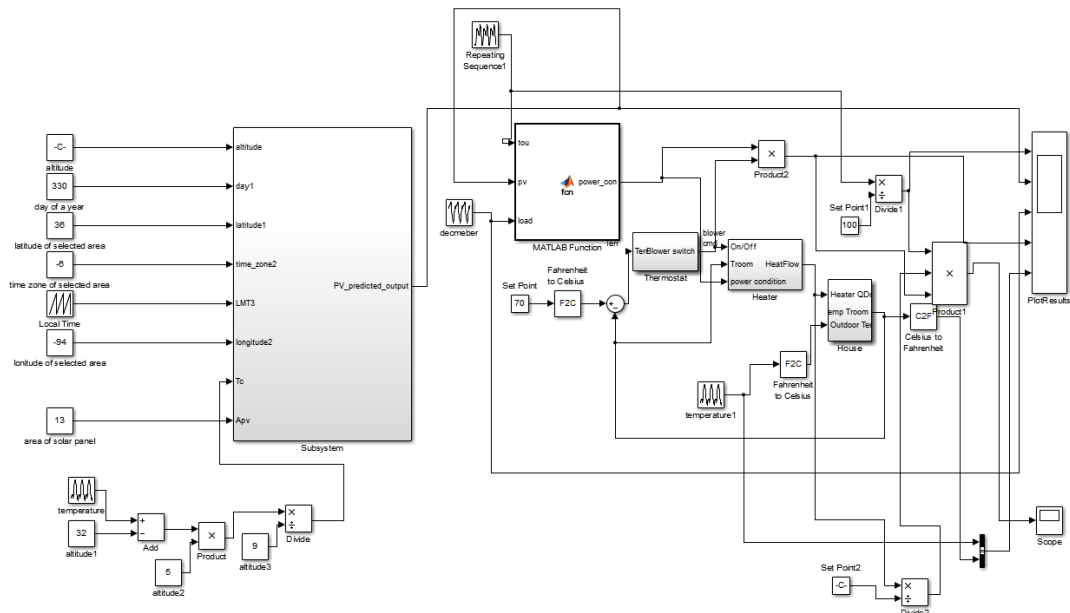


Fig. 4.5.7. Simulink model for space heater control.

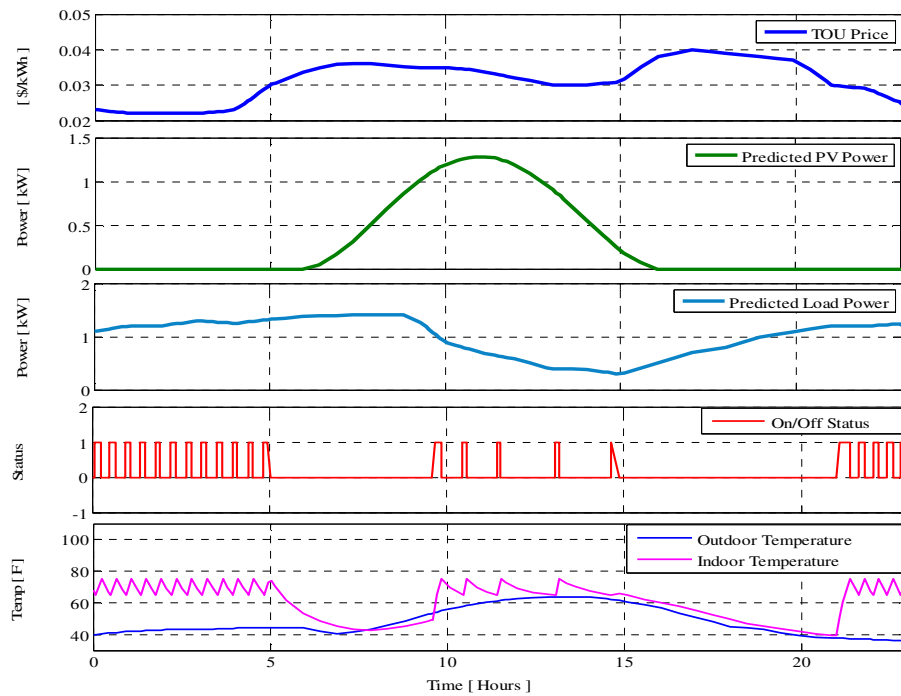


Fig. 4.5.8. Simulation results for automatic load management (space heater control).

Fig. 4.5.8 shows the simulation results for space heater control based on TOU price, 24 hour PV power prediction, and 24 hour load prediction. When the TOU price is low from 9 pm to 5 am, the space heater is continuously turned on/off although there will be no PV power available. It will also control the temperature between 65 °F and 75 °F. Whenever the temperature is beyond 75 °F the heater will be turned off, and when the temperature is below 65 °F the heater will be turned on again. Under any other circumstance, the heater will be turned on/off when the PV power is available to supply the space heater load demand. From 5 am to 9 am, the TOU price goes high and there will be not enough PV power to supply the load. Therefore, the space heater will be turned off until the PV power is available to compensate the load demand.

## CHAPTER 5

### CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

#### 5.1 Concluding Remarks

This thesis has presented the overall system level energy management for a residential power router which includes both power flow and load management to reduce the cost of electricity for the homeowner. Hardware systems were also designed by other graduate students involved in this project.

The power flow management system is designed based on battery SOC, electricity TOU price, PV power, and load demand. Among the various available methods, the voltage method was chosen for battery SOC because it is easier to implement in a DSP. Labview was used as an interface to parse the live TOU price data from the Internet. PV power and load power information were obtained from the hardware system. Using this information, a power flow management algorithm was implemented for both a grid-connected mode and an island mode. The daily, monthly, and yearly average electricity cost savings for the homeowner were also estimated. The total yearly savings for a year is around \$160.00 depending on seasonal PV power availability, different load demand and TOU pricing data. The total cost for the device is around \$1200.00. It will take  $(\$1200.00 / \$160.00) = 7.5$  years for a user to start getting the profit.

The load management system design was based on PV power prediction, load demand prediction, and TOU price information. Twenty-four hour PV power prediction was

implemented using the clear sky solar radiation model, which requires geographical data and weather data for the local area. These data are parsed live from the internet using Labview. A 24 hour electrical load demand prediction model was designed and implemented using a statistical auto regression model based on historical load demand data.

## **5.2 Recommendations for Future Work**

In this project, there were some limitations and challenges in system level design. There are also a few tasks remaining which are in need of completion. In this thesis, the PV prediction is modeled using geographical and weather data of Fayetteville, Arkansas. However, TOU pricing data used in the control algorithm is parsed from a website which shows the pricing data for Illinois, since there is no website available for Arkansas TOU pricing information. For the PV prediction, the solar radiation is estimated successfully. Based on the estimated solar radiation, PV power prediction is also calculated using weather information and solar panel parameters. Real solar panel will be used in further field testing. Currently, the TOU price and weather data obtained from the internet are sent to the DSP using Labview as a GUI interface. This arrangement could be made more robust by developing a smart phone application or web server for the system which directly sends the required data to the DSP. The development of a smart phone application is currently under construction. The system level control is integrated with hardware level control using Labview graphical user interface. However, both the communication and system level control can also be implemented in Raspberry Pi which will not require any computer interface for internal communication. The TOU price and weather data can be parsed directly from the internet to the Raspberry Pi and after completing the system level control tasks, the output will be sent to DSP for hardware level control. After full system

integration, more features can be added into this energy management system to improve accuracy and reliability.

## REFERENCES

- [1] "Short-Term Energy Outlook," U.S. Energy Information Administration, [Online]. Available: <http://www.eia.gov/forecasts/steo/>.
- [2] "Fact Sheet: Clean Power Plan Overview," [Online]. Available: <http://www2.epa.gov/carbon-pollution-standards/fact-sheet-clean-power-plan-overview>.
- [3] Y. Riffonneau and S. Bacha, "Optimal Power Flow Management for Grid Connected PV Systems With Batteries," *IEEE Transactions on Sustainable Energy*, vol. 2, 2011.
- [4] B. Stalling, R. Motte, T. Clemmer, H. A. Mantooth and R. Dougal, "Design and Evaluation of a Universal Power Router for Residential Applications," in *Energy Conversion Congress and Exposition*, Raleigh, 2012.
- [5] Y. Riffonneau, S. bacha, F. barruel and A. Delaille, "Energy flow management in grid connected PV systems with storage - A deterministic approach," in *IEEE International Conference on Industrial Technology*, 2009.
- [6] "Lead-acid Battery," [Online]. Available: [http://en.wikipedia.org/wiki/Lead%E2%80%93acid\\_battery](http://en.wikipedia.org/wiki/Lead%E2%80%93acid_battery).
- [7] M. Ceraolo, "New Dynamical Models of Lead-Acid Batteries," *IEEE transactions on Power Systems*, vol. 15, pp. 1184-1190, 2000.
- [8] J. P. Cun, J. Fiorina, M. Fraisse and H. Mabboux, "The Experience of a UPS Company in Advanced Battery Monitoring," in *Telecommunications Energy Conference*, 1996.
- [9] M. Valvo, F. Wicks, D. Robertson and S. Rudin, "Development and Application of an Improved Equivalent Circuit Model of a Lead Acid Battery," in *Proc. Energy Convers. Eng. Conf.*, 1996.
- [10] S. Pang, J. Farrell, J. Du and M. Barth, "Battery State-of-Charge Estimation," in *Americal Control Conference*, 2001.
- [11] F. M. Gonzalez-Longatt, "Circuit Based Battery Models: A Review," in *Congreso Iberoamericano de estudiantes De Ingenieria Electrica*, 2006.
- [12] S. Barsali and M. Ceraolo, "Dynamic Models of Lead Acid Batteries: Implementation Issues," *IEEE Transactions on Energy Conversion*, vol. 17, 2002.
- [13] D. Shanxu, C. Changsong and C. Tao, "Forecasting Power Output for Grid-Connected Photovoltaic Power System Without Using Solar Radiation Measurement," *IEEE International Symposium on Power Electronics for Distributed Generation Systems*, pp. 773-777, 2010.

- [14] I. Moghram and S. Rahman, "Analysis and Evaluation of Five Short-Term Load Forecasting Techniques," *IEEE Transactions on Power System*, vol. 4, 1989.
- [15] S. Teleke, M. E. Baran, S. Bhattacharya and A. Q. Huang, "Rule-Based Control of Battery Energy Storage for Dispatching Intermittent Renewable Sources," *IEEE Transactions On Sustainable Energy*, vol. 1, p. 117, 2010.
- [16] "How to measure state of charge," Battery University, [Online]. Available: [http://batteryuniversity.com/learn/article/how\\_to\\_measure\\_state\\_of\\_charge](http://batteryuniversity.com/learn/article/how_to_measure_state_of_charge).
- [17] "OPTIMA Batteries Specification," Johnson Controls Inc. Company, [Online]. Available: [http://d26maze4pb6to3.cloudfront.net/optimabatteries/9613/4583/5078/REDTOP\\_Full\\_Specs\\_Sheet.pdf](http://d26maze4pb6to3.cloudfront.net/optimabatteries/9613/4583/5078/REDTOP_Full_Specs_Sheet.pdf).
- [18] "Power Smart Pricing," [Online]. Available: <http://www.powersmartpricing.org/>.
- [19] "National Weather Service," [Online]. Available: <http://www.weather.gov/>.
- [20] "Numeric Profiles," Connecticut Light & Power, [Online]. Available: [http://www.clp.com/ESuppliers/CLPProfileSegments/Numeric\\_Profiles/?MenuID=4294985412](http://www.clp.com/ESuppliers/CLPProfileSegments/Numeric_Profiles/?MenuID=4294985412).
- [21] "Thermal Model Of a House," The MathWorks Inc., [Online]. Available: <http://www.mathworks.com/help/simulink/examples/thermal-model-of-a-house.html>.



## APPENDIX

### Appendix A: DSP Code for Power Flow and Load Management

```
#include "stdio.h"

#include "math.h"

# include "stdlib.h"

//for power flow management
float Iload=2.35;
float Vpv=50;
float Vbat=49.32;
float Vload_rms=240;
float Ipv=0;
float price=0;
int t;
float Iref;
float SOC;
float Ibat=-20;
int grid_status;
int mode=1; //mode=0 island ; mode=1 is grid-connected
int generator_status;
float TOU[24]= {2.6, 2.6, 2.6, 2.7, 3.1, 4.6, 11.4, 8.3, 8.3, 7.6, 6.6, 5.8,
5.1, 4.8, 4.5, 4.5, 4.7, 9.6, 11.2, 8.4, 6.1, 4.3, 4, 3};

//for pv prediction
float month;
float Day;
float GSC;
float fValue;
float Gon;
float theta;
float ET;
float latitude;
float longitude;
float latitude_1;
float longitude_1;
float solardec;
int Local_mean_time[24]= {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
15, 16, 17, 18, 19, 20, 21, 22, 23};
float Time_zone;
float omega[24];
float alpha;
float beta;
float Costhetaz[24];
float costhetas[24];
float r0;
float r1;
float rk;
```

```

float a0;
float a1;
float kappa;
float tr[24];
float td[24];
float tb[24];
float ro;
float A;
float Grt[24];
float Gdt[24];
float Gbt[24];
float Gt[24];
float conversion_efficiency;
float Solar_panel_area;
float Temperature[24];
float PV_output[24]={0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0};
float PV_OUTput[24]={0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0};
int PV_Output[24]={0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0};

// for load prediction and load scheduling
float TOUsum;
float TOUavg;
float prevload[24] = {1.946, 1.629, 1.545, 1.464, 1.512, 1.754, 1.898, 2.273,
2.398, 2.211, 2.006, 1.848, 1.776, 1.730, 1.584, 1.680, 1.803, 2.075, 2.498,
2.662, 2.582, 2.441, 2.340, 2.131};
float tempe[24]={2.131, 1.946, 1.629, 1.545, 1.464, 1.512, 1.754, 1.898,
2.273, 2.398, 2.211, 2.006, 1.848, 1.776, 1.730, 1.584, 1.680, 1.803, 2.075,
2.498, 2.662, 2.582, 2.441, 2.340};
float tempe3[24]={2.340, 2.131, 1.946, 1.629, 1.545, 1.464, 1.512, 1.754,
1.898, 2.273, 2.398, 2.211, 2.006, 1.848, 1.776, 1.730, 1.584, 1.680, 1.803,
2.075, 2.498, 2.662, 2.582, 2.441};
float tempe4[24]={2.441, 2.340, 2.131, 1.946, 1.629, 1.545, 1.464, 1.512,
1.754, 1.898, 2.273, 2.398, 2.211, 2.006, 1.848, 1.776, 1.730, 1.584, 1.680,
1.803, 2.075, 2.498, 2.662, 2.582};
float tempe5[24]={2.582, 2.441, 2.340, 2.131, 1.946, 1.629, 1.545, 1.464,
1.512, 1.754, 1.898, 2.273, 2.398, 2.211, 2.006, 1.848, 1.776, 1.730, 1.584,
1.680, 1.803, 2.075, 2.498, 2.662};
float tempe6[24]={2.662, 2.582, 2.441, 2.340, 2.131, 1.946, 1.629, 1.545,
1.464, 1.512, 1.754, 1.898, 2.273, 2.398, 2.211, 2.006, 1.848, 1.776, 1.730,
1.584, 1.680, 1.803, 2.075, 2.498};
float tempe7[24]={2.498, 2.662, 2.582, 2.441, 2.340, 2.131, 1.946, 1.629,
1.545, 1.464, 1.512, 1.754, 1.898, 2.273, 2.398, 2.211, 2.006, 1.848, 1.776,
1.730, 1.584, 1.680, 1.803, 2.075};
float tempe8[24]={2.075, 2.498, 2.662, 2.582, 2.441, 2.340, 2.131, 1.946,
1.629, 1.545, 1.464, 1.512, 1.754, 1.898, 2.273, 2.398, 2.211, 2.006, 1.848,
1.776, 1.730, 1.584, 1.680, 1.803};
float tempe9[24]={1.803, 2.075, 2.498, 2.662, 2.582, 2.441, 2.340, 2.131,
1.946, 1.629, 1.545, 1.464, 1.512, 1.754, 1.898, 2.273, 2.398, 2.211, 2.006,
1.848, 1.776, 1.730, 1.584, 1.680};
float tempe10[24]={1.680, 1.803, 2.075, 2.498, 2.662, 2.582, 2.441, 2.340,
2.131, 1.946, 1.629, 1.545, 1.464, 1.512, 1.754, 1.898, 2.273, 2.398, 2.211,
2.006, 1.848, 1.776, 1.730, 1.584};

```

```

float tempe11[24]={1.584, 1.680, 1.803, 2.075, 2.498, 2.662, 2.582, 2.441,
2.340, 2.131, 1.946, 1.629, 1.545, 1.464, 1.512, 1.754, 1.898, 2.273, 2.398,
2.211, 2.006, 1.848, 1.776, 1.730};
float tempe12[24]={1.730, 1.584, 1.680, 1.803, 2.075, 2.498, 2.662, 2.582,
2.441, 2.340, 2.131, 1.946, 1.629, 1.545, 1.464, 1.512, 1.754, 1.898, 2.273,
2.398, 2.211, 2.006, 1.848, 1.776};
int load[24]={0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0};
float aa=0;
float bb=0;
float cc=0;
float dd=0;
float ee=0;
float ff=0;
float gg=0;
float hh=0;
float ii=0;
float jj=0;
float kk=0;
float ll=0;
float load1;
float load2;
float load3;
char load1_status[24];
char load2_status[24];
char load3_status[24];

//Fahad

void main(void)
{
int i;
float upper_price = TOU[0];
float lower_price = TOU[0];
for (i = 0; i < 24; i++)
{
if (TOU[i] > upper_price)
{
upper_price = TOU[i];
}
else if (TOU[i] < lower_price)
{
lower_price = TOU[i];
}
}

// determining the current TOU price according to time input in hour

if (t<1)
{price=TOU[0];}
else if (t<2)
{price=TOU[1];}
else if (t<3)
{ price=TOU[2];}
else if (t<4)
{ price=TOU[3];}
else if (t<5)

```

```

{ price=TOU[4];}
else if (t<6)
{ price=TOU[5];}
else if (t<7)
{ price=TOU[6];}
else if (t<8)
{ price=TOU[7];}
else if (t<9)
{ price=TOU[8];}
else if (t<10)
{ price=TOU[9];}
else if (t<11)
{ price=TOU[10];}
else if (t<12)
{ price=TOU[11];}
else if (t<13)
{ price=TOU[12];}
else if (t<14)
{ price=TOU[13];}
else if (t<15)
{ price=TOU[14];}
else if (t<16)
{ price=TOU[15];}
else if (t<17)
{ price=TOU[16];}
else if (t<18)
{ price=TOU[17];}
else if (t<19)
{ price=TOU[18];}
else if (t<20)
{ price=TOU[19];}
else if (t<21)
{ price=TOU[20];}
else if (t<22)
{ price=TOU[21];}
else if (t<23)
{ price=TOU[22];}
else
{price=TOU[23];}

SOC= (((Vbat-(Ibat*0.0030))-31.5)/6.9);

//battery current reference estimation for grid-connected mode

    if (mode==1)
{
//battery current reference estimation for grid-connected mode

if (5+((-( (((price*100)-(lower_price*100))/((upper_price*100)-(lower_price*100)))+1)*(-2-5))>0)

    {
        if (Ipv*Vpv < Iload*Vload_rms)
        {
            if (SOC > 0.5 )
            {

```

```

        if (((Iload*Vload_rms-Ipv*Vpv)/Vbat) <= 5+((-((price*100)-
(lower_price*100))/((upper_price*100)-(lower_price*100))+1)*(-2-5)))
        {
            Iref=((Iload*Vload_rms-Ipv*Vpv)/Vbat);
            grid_status=1;
        }
        else
        {
            Iref= (5+((-((price*100)-(lower_price*100))/((upper_price*100)-
(lower_price*100))+1)*(-2-5)));
            grid_status=1;
        }
        }
        else
        {
            Iref = 0;
            grid_status=1;
        }
        }
        else
        {
            Iref = 0;
            grid_status=0;
        }
        }
        else
        {
            if (Ipv*Vpv < Iload*Vload_rms)
            {
                if (SOC < 0.8)
                {
                    Iref = -((5 + (-2 -5)*(-((price*100) -
(lower_price*100))/((upper_price*100)-(lower_price*100))+1)));
                    grid_status=1;
                }
                else if (SOC < 1)
                {
                    if ((100000*exp(-5*((Vbat-(-Ibat*0.0030))-31.5)/6.9)*1) <= 5 +
(-2 -5)*(-((price*100) - (lower_price*100))/((upper_price*100)-
(lower_price*100))+1))
                    {
                        Iref=-(-100000*exp(-5*((Vbat-(-Ibat*0.0030))-31.5)/6.9)*1));
                        grid_status=1;
                    }
                    else
                    {
                        Iref= (5 + (-2-5)*(-((price*100) -
(lower_price*100))/((upper_price*100)-(lower_price*100))+1));
                        grid_status=1;
                    }
                }
                }
            else
            {
                Iref = 0 ;
            }
        }
        else

```

```

        {
            if (SOC < 0.9)
            {
if ((Ipv*Vpv-Iload*Vload_rms)/Vbat > 2)
{
                Iref = -2;
            }
else
{
                Iref=-(Ipv*Vpv-Iload*Vload_rms)/Vbat;
            }
        }

        else
        {
            Iref = 0 ;
        }
    }

}
else
{
    if (SOC <= 0.5)
    {
        if (Ipv*Vpv-Iload*Vload_rms > 0)
        {
            if ( ((Ipv*Vpv-Iload*Vload_rms)/Vbat) <= 2)
            {
                Iref = -(Ipv*Vpv-Iload*Vload_rms)/Vbat;
                generator_status=0;
            }
            else
            {
                Iref = -2;
                generator_status=0;
            }
        }
        else
        {
            Iref = 0;

            generator_status=1;
        }
    }
    else if (SOC >= 0.99)
    {
        if (Ipv*Vpv < Iload*Vload_rms)
        {
            if (((Ipv*Vpv-Iload*Vload_rms)/Vbat) >= -5)
            {
                Iref = -(Ipv*Vpv-Iload*Vload_rms)/Vbat;
                generator_status=0;
            }
            else
            {
                Iref = 5;
                generator_status=1;
            }
        }
    }
}

```

```

}
else
{
Iref = 0;
generator_status=0;
}
}
else if (SOC > 0.8)
{
if (Ipv*Vpv > Iload*Vload_rms)
{
if (((Ipv*Vpv-Iload*Vload_rms)/Vbat) > 1e5*exp(-5*SOC*2.30))
{
if ((Ipv*Vpv-Iload*Vload_rms)/Vbat > 2)
{
Iref = -2;
generator_status=0;
}
else
{
Iref = -(1e5*exp(-5*SOC*2.30));
generator_status=0;
}
}
}
else
{
if ((Ipv*Vpv-Iload*Vload_rms)/Vbat > 2)
{
Iref = -2;
generator_status=0;
}
else
{
Iref = -(Ipv*Vpv-Iload*Vload_rms)/Vbat;
generator_status=0;
}
}
}
}
else if (((Ipv*Vpv-Iload*Vload_rms)/Vbat) < -5)
{
Iref = 5;
generator_status=1;
}
else
{
Iref = -(Ipv*Vpv-Iload*Vload_rms)/Vbat;

generator_status=0;
}
}
else
{
if (Ipv*Vpv > Iload*Vload_rms)
{
if (((Ipv*Vpv-Iload*Vload_rms)/Vbat) > 2)
{

```

```

Iref = -2;

generator_status=0;

}
else
{
Iref = (Ipv*Vpv-Iload*Vload_rms)/Vbat;
generator_status=0;
}
}
else if (((Ipv*Vpv-Iload*Vload_rms)/Vbat) < -5)
{
Iref = 5;
generator_status=1;
}
}
else
{
Iref = -(Ipv*Vpv-Iload*Vload_rms)/Vbat;
generator_status=0;
}
}
}

//load management code//
if(t == 17)
{
int j;
for (j=0; j<24;j++)
{
GSC=1370;
// time in minutes
fValue = cos(2 * 3.1416 * Day / 365);
Gon = GSC * (1 + (0.033 * fValue));
theta = 2 * 3.1416 * (Day - 1) / 365;
ET = 229.1831
      * (7.5e-5 + .001868 * cos(theta) - .032077 *
sin(theta)
      - .014615 * cos(2 * theta) - .040849 *
sin(2 * theta));
latitude_1 = 3.1416 / 180 * latitude;
longitude_1 = longitude;
// solar declination
solardec = 3.1416 / 180 * 23.45 * sin(2 *3.1416 * (Day + 284) /
365);
// hour angle vector
omega[j] = 3.1416/180*((Local_mean_time[j]-Time_zone-
12)*15+longitude_1+ET/4);
// inclination angle of surface
beta = 0;
alpha = 0;
// solar zenith angle vector
Costhetaz[j] = cos(solardec) * cos(latitude_1) * cos(omega[j])
+ sin(solardec) * sin(latitude_1);
// incident angle vector

```



```

        costhetas[j] = sin(solardec) * sin(latitude_1) * cos(beta)
                    - sin(solardec) * cos(latitude_1) * sin(beta) *
cos(alpha)
                    + cos(solardec) * cos(latitude_1) * cos(beta) *
cos(omega[j])
                    + cos(solardec) * sin(latitude_1) * sin(beta) *
cos(alpha)
                    * cos(omega[j])
                    + cos(solardec) * sin(alpha) * sin(omega[j]) *
sin(beta);
// correction factors for this climate type
r0 = .97;
r1 = .99;
rk = 1.02;
A=0.0960;
a0 = r0 * (0.4237 - 0.00821 * pow((6 - A), 2));
a1 = r1 * (.5055 + .00595 * pow((6.5 - A), 2));
kappa = rk * (.2711 + .01858 * pow((2.5 - A), 2));
tb[j] = (a0 + a1 * exp(-kappa / Costhetaz[j]));
td[j] = .271 - .294 * tb[j];
tr[j] = .271 + .706 * tb[j];
ro = .2;
Gbt[j] = Gon * tb[j] * costhetas[j];
Gdt[j] = Gon * Costhetaz[j] * td[j] * ((1 + cos(beta)) / 2);
Grt[j] = ro * Gon * Costhetaz[j] * tr[j] * ((1 + cos(beta)) /
2);
Gt[j] = Gbt[j] + Gdt[j] + Grt[j];
conversion_efficiency=0.2;
Solar_panel_area=10;
PV_OUTput[j]= conversion_efficiency*Solar_panel_area*Gt[j]*(1-5e-
3*((Temperature[j]-32)*0.556)-25))*0.5*(sin(Costhetaz[j])+1)/1000;
PV_output[j] = (PV_OUTput[j]*1000);

if (PV_output[j]<=0)
{PV_Output[j]=0;}
else
{PV_Output[j]=(int) (PV_output[j]);}
    if (month<= 4)
    {
aa=0.5816;
bb=1.09;
cc=-0.169;
dd=0.217963;
ee=-0.15357;
ff=-0.26548;
gg=-0.23235;
hh=-0.07016;
ii=0.33199;
jj=0.04902;
kk=0.3059;
ll=-0.5379;
    }
    else if (month<=8)
    {
aa=0.70;
bb=1.29;
cc=-0.52;

```

```

dd=-0.007;
ee=-0.17;
ff=0.329;
gg=-0.413;
hh=0.04;
ii=0.099;
jj=-0.133;
kk=0.311;
ll=-0.175;
    }
else
{
aa=1.57;
bb=1.20;
cc=-0.9;
dd=0.40;
ee=-0.16;
ff=-0.13;
gg=0.11;
hh=-0.33;
ii=0.219;
jj=-0.27;
kk=-0.24;
ll=0.24;
}
load[j] =
(aa+(bb*tempe[j])+(cc*tempe3[j])+(dd*tempe4[j])+(ee*tempe5[j])+(ff*tempe6[j])
)+(gg*tempe7[j])+(hh*tempe8[j])+(ii*tempe9[j])+(jj*tempe10[j])+(kk*tempe11[j])
)+(ll*tempe12[j]))*1000);
    load1=(0.87*1000);
    load2=(0.62*1000);
    load3=(0.47*1000);
TOUsum=(TOU[0]+TOU[1]+TOU[2]+TOU[3]+TOU[4]+TOU[5]+TOU[6]+TOU[7]+TOU[8]+TOU[9]
+TOU[10]+TOU[11]+TOU[12]+TOU[13]+TOU[14]+TOU[15]+TOU[16]+TOU[17]+TOU[18]+TOU[
19]+TOU[20]+TOU[21]+TOU[22]+TOU[23]);
TOUavg=(TOUsum/24);
    if (TOU[j]<TOUavg)
    {
        load1_status[j]='1';
        load2_status[j]='1';
        load3_status[j]='1';
    }
else
{
    if (PV_Output[j] > load[j]) //if (PV_output[j] > load[j])
    {
        load1_status[j]='1';
        load2_status[j]='1';
        load3_status[j]='1';
    }
else if (PV_Output[j]>load1+load2+load3)
{ load1_status[j]='1';
    load2_status[j]='1';
    load3_status[j]='1';}
else if (PV_Output[j]>load1+load2)
{ load1_status[j]='1';
    load2_status[j]='1';

```

```

        load3_status[j]='0';
        else if (PV_Output[j]>load1+load3)
        { load1_status[j]='1';
          load2_status[j]='0';
          load3_status[j]='1';}
        else if (PV_Output[j]>load3+load2)
        { load1_status[j]='0';
          load2_status[j]='1';
          load3_status[j]='1';}
        else if (PV_Output[j]>load1)
        {load1_status[j]='1';
          load2_status[j]='0';
          load3_status[j]='0';}
        else if (PV_Output[j]>load2)
        {load1_status[j]='0';
          load2_status[j]='1';
          load3_status[j]='0';}
        else if (PV_Output[j]>load3)
        {load1_status[j]='0';
          load2_status[j]='0';
          load3_status[j]='1';}
        else
        { load1_status[j]='0';
          load2_status[j]='0';
          load3_status[j]='0';}
    }
}
}
}
}

```