‘Peak Shave’ Energy Managing Tool For Energy Subsystems Design of AMS Sustainable Food Truck

Shao (Raymond) Hou

University of British Columbia

EECE 490L

April 08, 2014

Contents

Glossary

Peak Shave Energy System	An energy managing system
GHG	Green House Gas
SOC	Battery State of Charge
Battery Banks	A stacks of batteries connected

ABSTRACK

Sustainability is significantly affecting people's lives. Sustainability ensures that the future generations would have sufficient clean resources to develop a sustainable society. Engineers are responsible for making sure they are directing the world to the correctly direction. The UBC Alma Mater Society (AMS) has provide a chances to the undergraduate engineering students to explore their interested research topics such as the sustainable project called "Energy Subsystems of AMS Sustainable Food Truck". In the project, an engineering will research, design and implement an energy managing system to help the AMS and the society eliminate the Green House Gas (GHG) emissions in the near future.

1.0 INTRODUCTION

UBC made a commitment at the GLOBE 2010 conference on reducing GHGs by 33%, 67% and 100% from 2007 levels by 2015, 2020 and 2050 respectively. Sustainable technology is important to achieve UBC's goal and it will also significantly increase the quality of lives for the future generations. This project is an initial research into the AMS Sustainable Food Truck Energy System Design.

1.01 STATEMENT OF PURPOSE

The project provides a tool to assess energy consumption and energy supply of a specific system based on the energy availability, device specifications, usage patterns and solar condition.

1.02 OBJECTIVE

The objective of this project is to provide a 'Peak Shave' Energy Managing Tool will be developed using the tools of MATLAB and Simulink to help the AMS access the behavior of different power systems of the food truck and provide details how the systems are operating based on the input parameters of the power system and conditions of solar radiations, the availability of the fuel cell energy source and energy stored in the battery banks.

1.03 SIGNIFICANCE

Firstly, the results of the project will provide significant supports to achieve AMS's goal of designing the sustainable food truck systems to reduce the GHG emissions footprints. Furthermore, the results of the project will also significantly raise awareness of using new green technologies to eliminate the food truck greenhouse gas emissions.

1.04 SCOPE

The scope of the project defines the design and implementation of a ‘Peak Shave’ Energy Managing Tool, simulations of different system scenarios and generates energy usage profiles of a defined system based on different scenarios such as Sunny Day Operation, Cloudy Day Operation in the summer time. The scope also includes the design and implementation of the circuitry behavior of the energy sources such as solar cells and energy storage control system.

2.0 HOW DOES THE ‘PEAK SHAVE’ ENERGY MANAGING WORK

The system consists a few main components, the power source, energy storage system, energy conversion systems and the loads. The system use solar energy, fuel cell energy and battery as the main power source. In addition, the battery can be charged from the electric grid at the end of the operation so that the battery will be fully charged at the beginning of the operation in the next day. The goal of the project is to design and implement a ‘Peak Shave’ Energy Managing Tool for the AMS to access their chosen power equipment in order to optimize their design and reduce the Green House Gas (GHG) emissions.

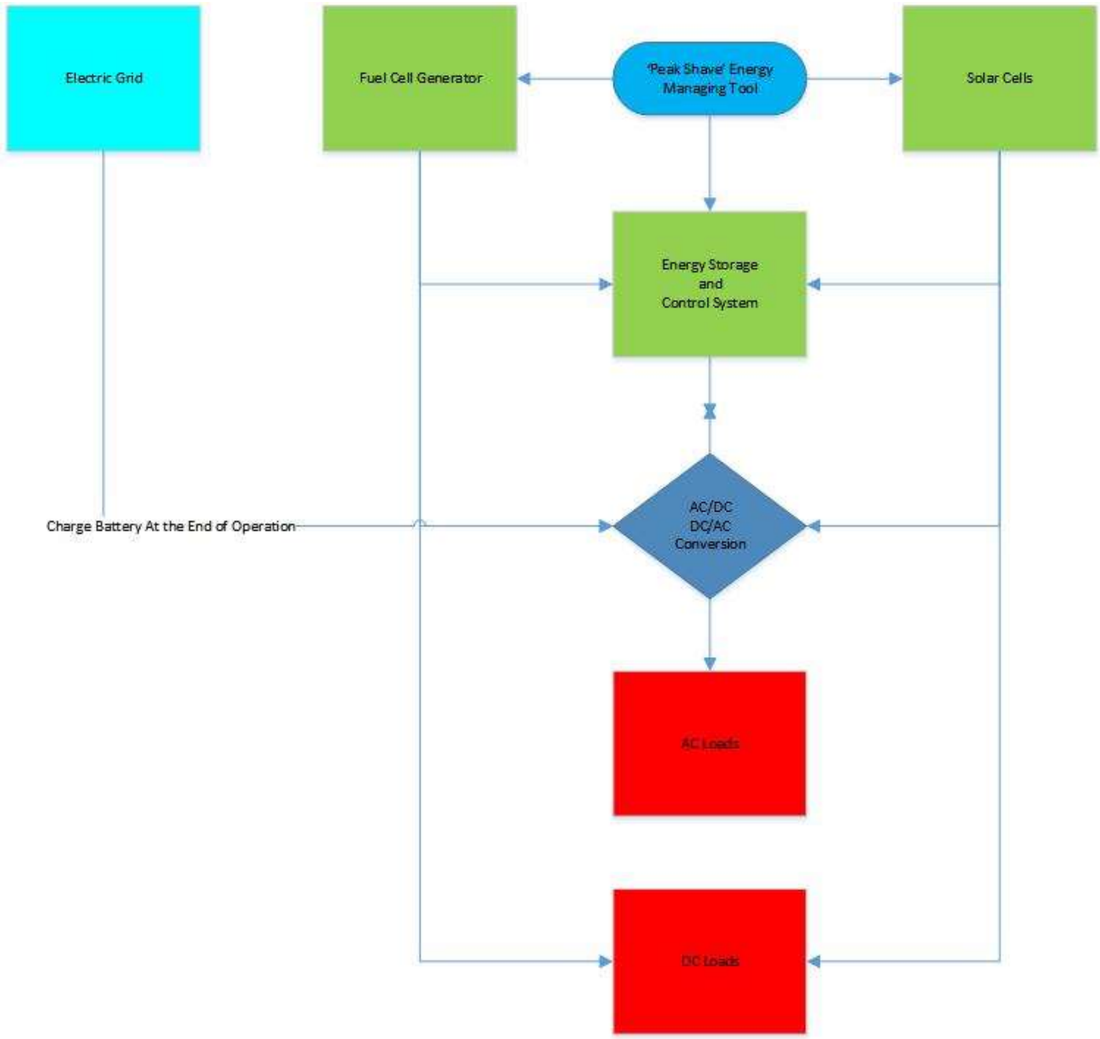

Figure 1'Peak Shave' Energy Managing Tool Overview

2.1 PEAK SHAVE ENERGY MANAGING TOOL

A 'Peak Shave' Energy Managing Tool is a system developed using the tool of Matlab and Simulink that prioritizes the use of the energy sources.

2.2 SYSTEM COMPONENTS AND OPERATION

In this project the system defines the solar cells, fuel cell generator, and the battery banks as the primary, secondly and backup energy source respectively. The system will manage to provide energy to the load according to the available of energy from the sources and the energy demand of the loads. The following figure shows the algorithm of the system. When the system is on, the system will determine how much energy is available from the solar, the fuel cell and the battery. Then it will determine how much energy will need to supply to the loads and check if there is sufficient energy to supply to the loads. Once it confirms that it has enough energy to operate the system, it will consistently monitor the energy usage patterns of the subsystems and the energy supply patterns of the sources. All the energy absorbed from the solar panels will be supplied to the load as the primary energy; if there is not enough energy from the solar cells, the system will turn on the fuel cell generator to generate additional energy to the loads. Meanwhile, the system will determine the excessive energy and makes sure the battery will always store the excessive energy if the battery is not fully charged. In addition, when the system determines that the solar cells and fuel cells cannot satisfy the energy demand of the loads, the battery will be turn on to provide additional energy to the loads if the state of charge(SOC%) of the battery is above the reserved level. Otherwise, will circuit breaker in the system will be triggered to disconnect the loads so that the power source will not be overloaded. During the operation the system will consistently save the energy usage data such that it can visualize how the systems consume energy and therefore help the AMS to analyze their designs and optimize the design.

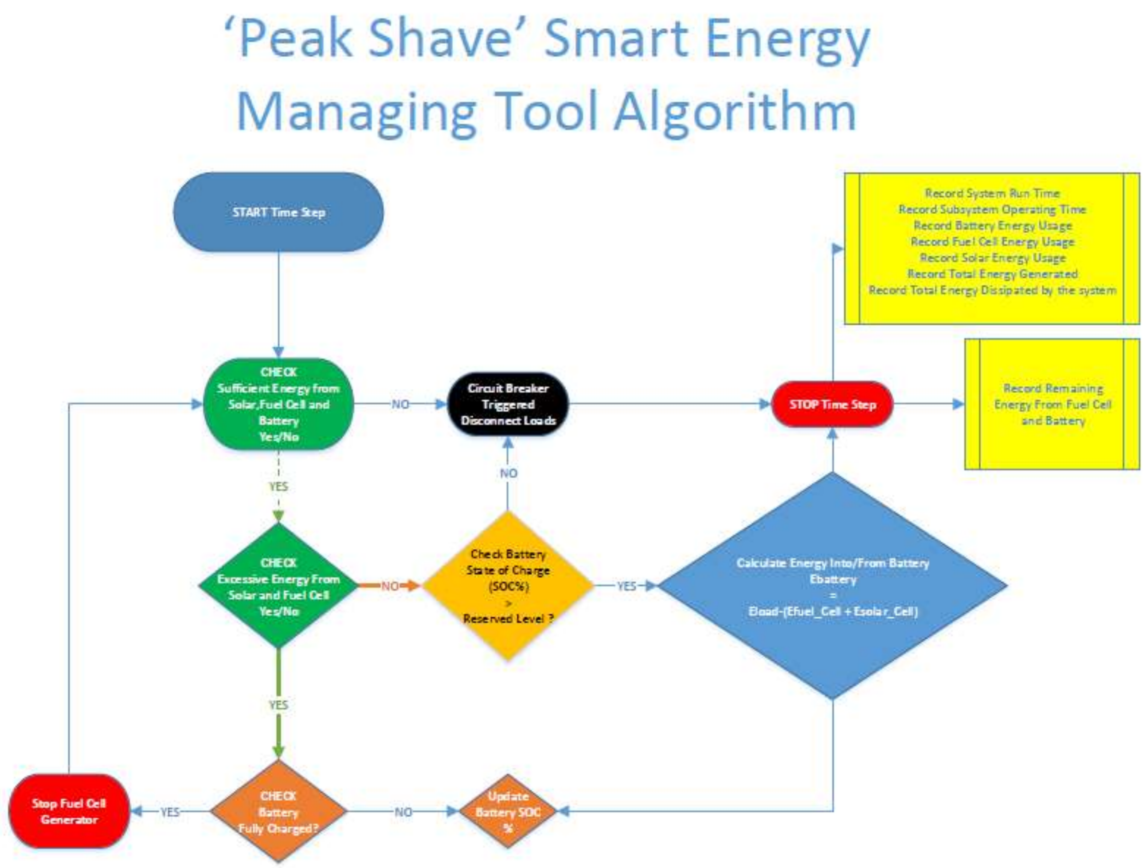

Figure 2'Peak Shave' Energy Managing Tool Algorithm

4.0 SIMULATIONS

It requires the user to input the system specifications and solar radiation conditions and energy usage patterns of the subsystems of the food truck. This parameters can be generated using MATHLAB program as input profiles to the system. Table 1 in section 4.1 summarizes the system settings. Once the subsystems are defined, it is ready for starting the simulation and produce significant results helping the AMS and other users optimize their designs, for example the AMS Food Truck Energy System Design.

4.1 SCENARIO 1: SUNNY BUSY DAY

During the development of the project, lots of experiments and trials have been done in order to produce sufficient evidence to support the results. In this report, two simulation based on different

Table 1 Simulation Scenario 1 Parameters

Simulation Scenario 1

Busy Sunny Day Operation Profiles

Power Source	Max Output Power	Operating Time(Hours)
Solar Energy	320*10=32000 W/h	6AM to 8PM (Summer)
Fuel Cell Power	30KW @5000W/h Output Rate	Depends on Power Output Rates
Battery Banks	10KWh	Depends on Power Output Rates

Loads	Power(Watts/h)	Simulation Interval(5 minutes/cycle)		Operating Time(Hours)
Griddle	3000	ON All the Time		10AM to 6Pm
Deep Fryer	1200	ON 90 minutes	OFF 30 minutes	10AM to 6PM
Fridge	100	ON 10 minutes	OFF 50 Minutes	10AM to 6PM
Microwave	1000	ON 5 Minutes	OFF 5 Minutes	10AM to 6PM
Ventilation	200	ON all the Time During Operation		10AM to 6PM
Others	110	ON all the time		10AM to 6PM

energy usage patterns of the subsystems, specifications of the systems and the initial conditions of the systems. These simulations represent the operation during a sunny day for busy operation and cloudy day for normal. The results of a sunny day operation can be summarized by the following graphs. Table 1 summarizes the considerations and assumptions of the subsystems. The sun rises at 6AM and goes down at 8PM. Operation time is 8 hours from 10AM to 6PM.

The top part of figures 3 shows the energy usage patterns of the loads, solar cells, fuel cells and battery. Please note that the battery state of charge (SOC%) is scaled by a factor of 100 in order to provide a clear idea of how the system behaves. As it shows in figure 3, the horizontal axis indicates the hours of the day from 00:00AM to 20:00PM. The sun rises from 6AM but the loads are not operating until 10AM. The fuel cell generator is set to switch on when the battery is not fully charged. Since the battery is fully charged at the beginning of the operation, it is not absorbing any energy from the solar cells. When the load systems start operating, the power supply system start providing energy to the loads. The energy of the battery start drops as more energy is required by the loads and rises when operation time finishes at 18:00PM. The simulation accurately shows how the loads and power source behaves. The bottom part of the figure show how the battery and fuel cell operates when it is set to trigger the fuel cell to operate when the battery SOC% is less than 70%. When the trigger is set to 70%, the system might be able to save some fuel cell energy; but it would probably decrease the output energy efficiency since the fuel cell is switching on and off frequently. Thus, it is a good idea to switch the fuel cell on only if the battery is not fully charged and the loads need more energy from it.

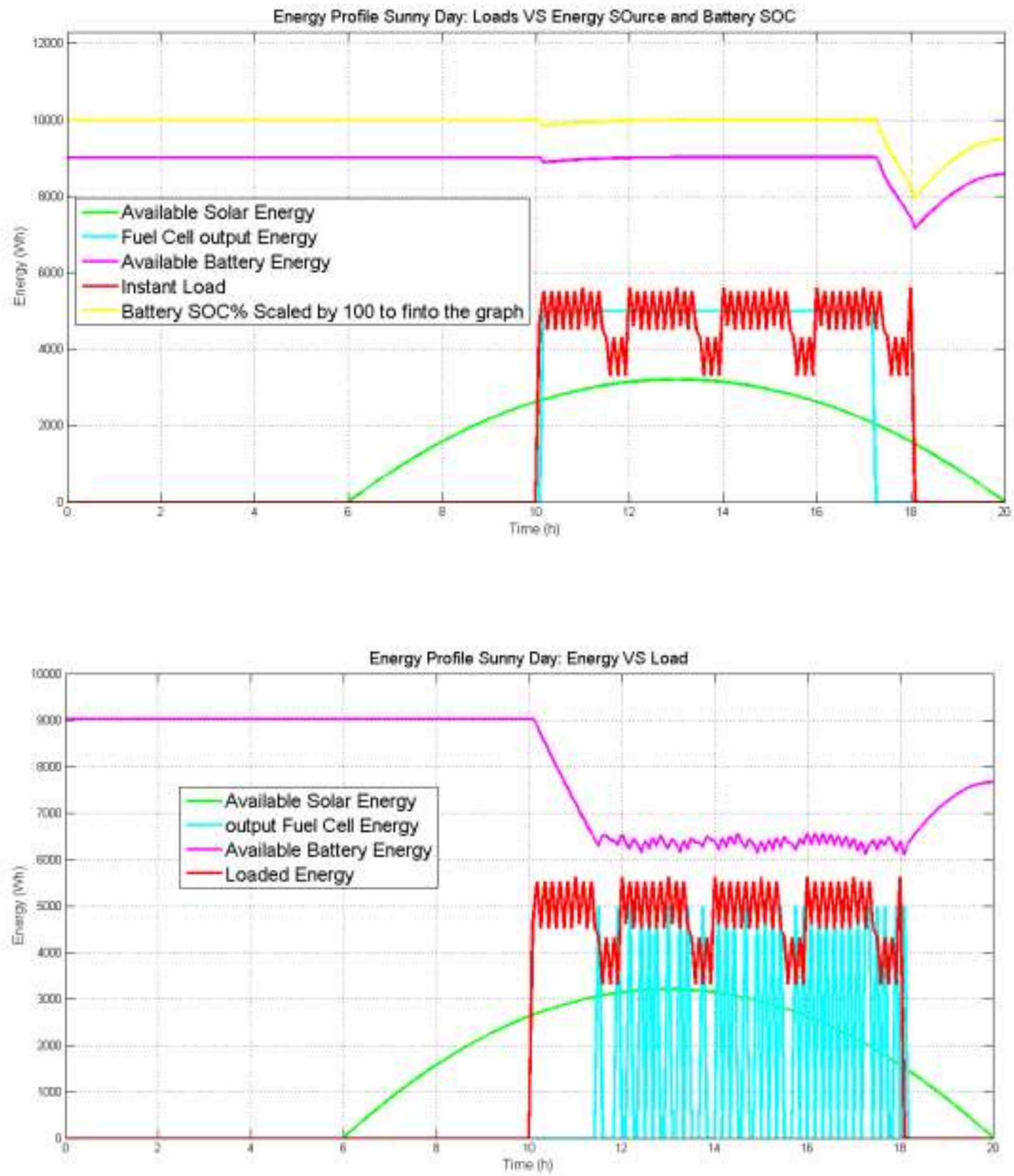

Figure 3Load pattern and energy usage for the loads, battery, solar cell and fuel cells Setting Fuel Cell Triggered at SOC 100% and 70% respectively

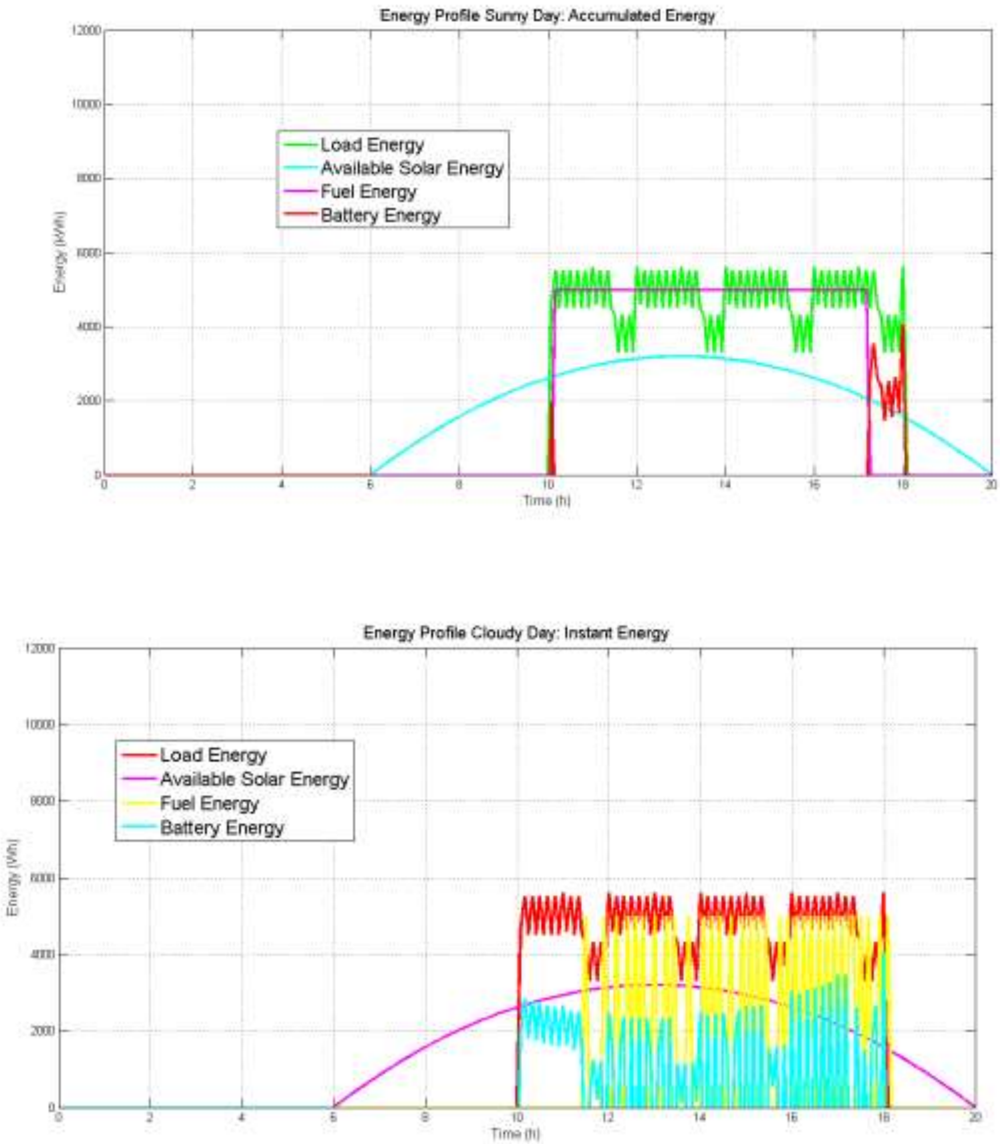

Figure 4 Instant energy from power source and to the loads Setting Fuel Cell Triggered at SOC 100% and 70% respectively

Similarly, figure 4 shows the instant energy usage of the loads and the energy source for different fuel cell generator trigger settings. Similar results are concluded. It is better to keep the fuel cell on for some time by setting the fuel cell trigger to 100% SOC. Same conclusion are made for other simulations. Figure 5 represents the accumulated energy used and the accumulated available energy from the sun. Figure 6 show how the system monitors the remaining energy of battery and the fuel cell generator. This feature definitely helps the users easily decide how much fuel they should prepare for the day of operation.

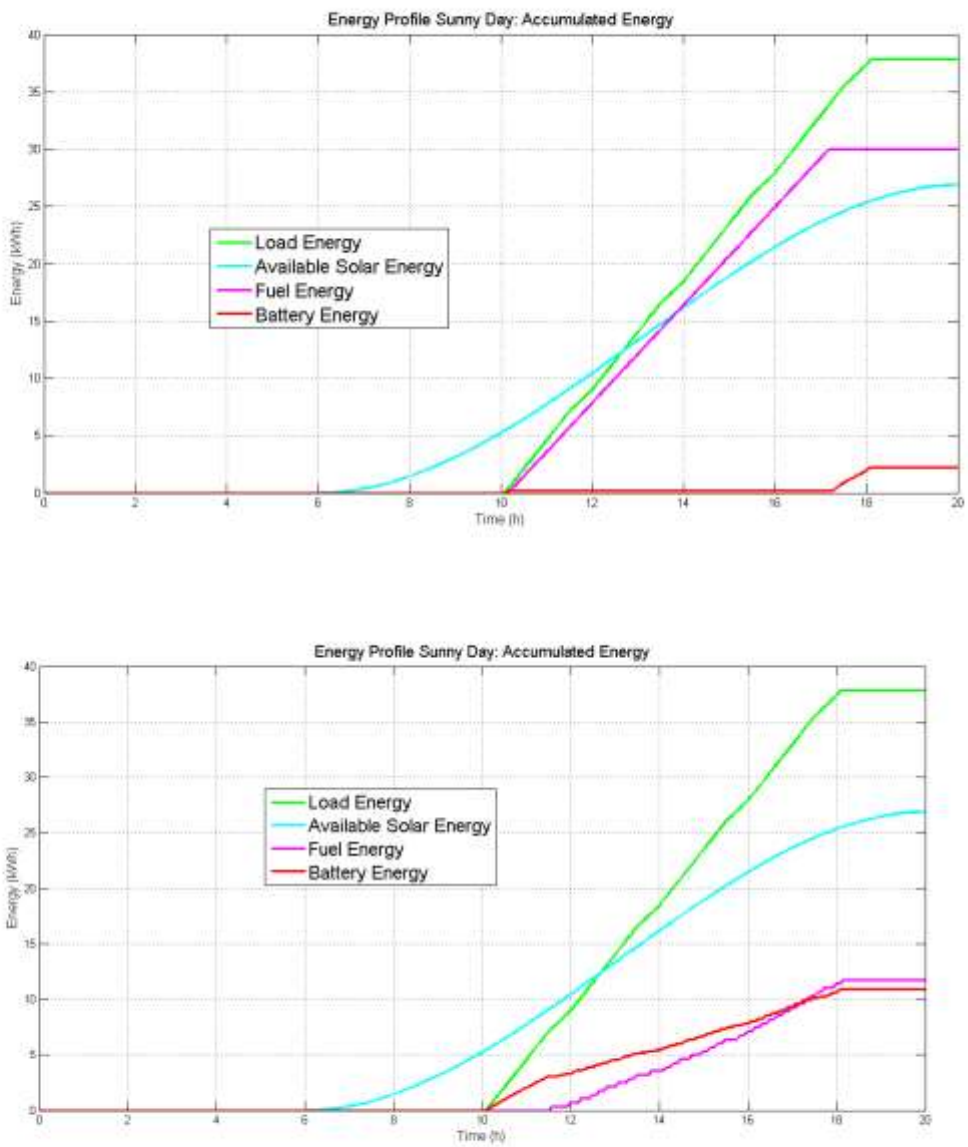

Figure 5 Available Energy and Accumulated Energy of Loads and Sources Setting Fuel Cell Triggered at SOC 100% and 70% respectively

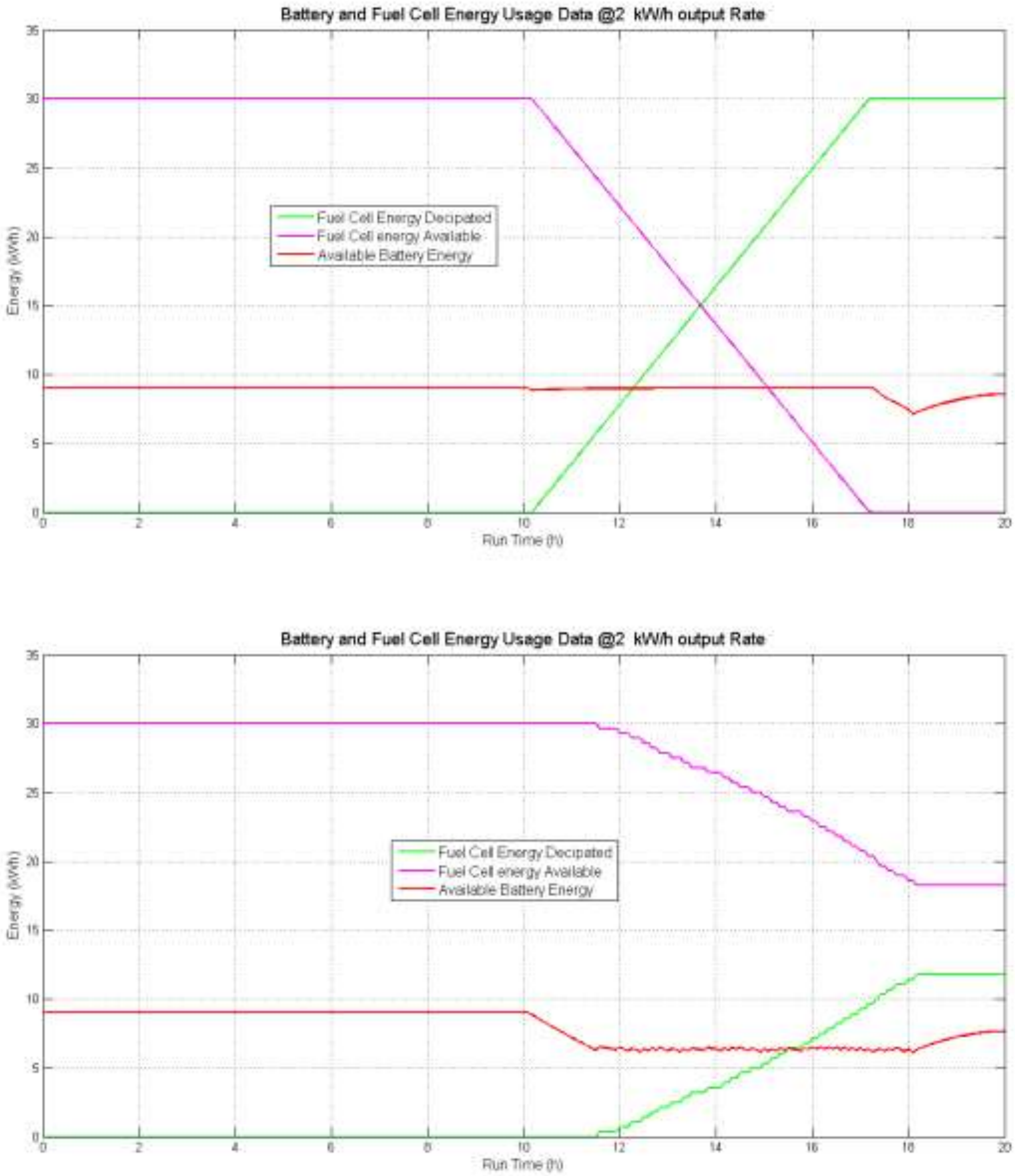

Figure 6 Energy Available and Used from Fuel Cell and Battery Setting Fuel Cell Triggered at SOC 100% and 70% respectively

4.2 SCENARIO 2: CLOUDY DAY

In scenario 2 it shows similar results. The total accumulated energy dissipated by the loads is about 35kWh while it is about 38kWh in the sunny day in scenario 1. Both simulation summarizes the energy usage patterns, energy availability before and after operation and as well as during the operation.

Table 2 Simulation Scenario 2 Parameters

Simulation Scenario 2

Normal Cloudy Day Operation Profiles

Power Source	Max Output Power	Solar Radiation Hours
Solar Energy	200*10=2000 W/h	8AM to 6PM (Cloudy)
Fuel Cell Power	10KW @5000W/h Output Rate	Depends on Power Output Rates
Battery Banks	10KWh	Depends on Power Output Rates

Loads	Power(Watts/h)	Simulation Interval(5 minutes/cycle)		Operating Time(Hours)
Griddle	3000	ON All the Time		10AM to 6Pm
Deep Fryer	1200	ON 60 minutes	OFF 30 minutes	10AM to 6PM
Fridge	100	ON 10 minutes	OFF 50 Minutes	10AM to 6PM
Microwave	1000	ON 5 Minutes	OFF 10 Minutes	10AM to 6PM
Ventilation	200	ON all the Time During Operation		10AM to 6PM
Others	80	ON all the time		10AM to 6PM

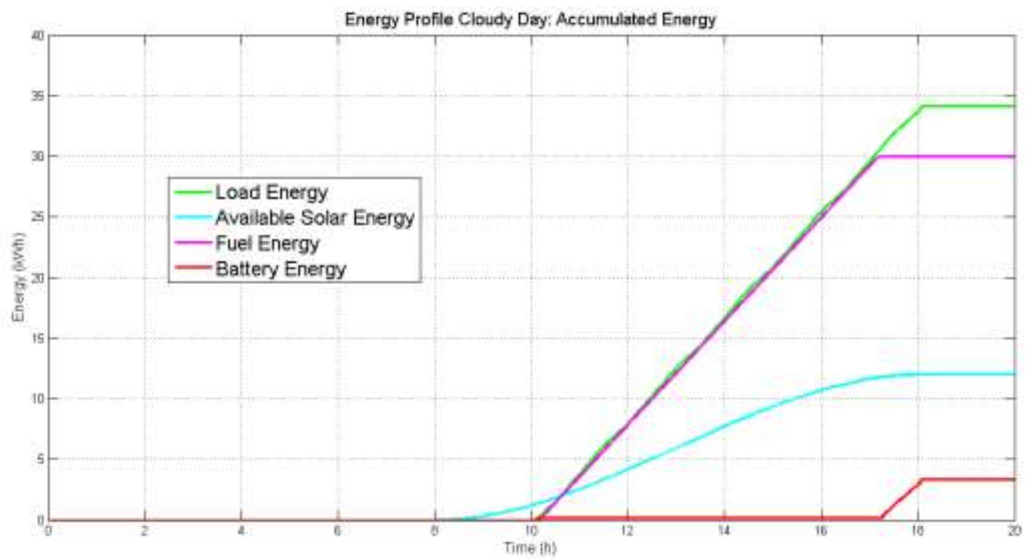

Figure 7 Accumulated Energy of different systems and available energy from sun radiations

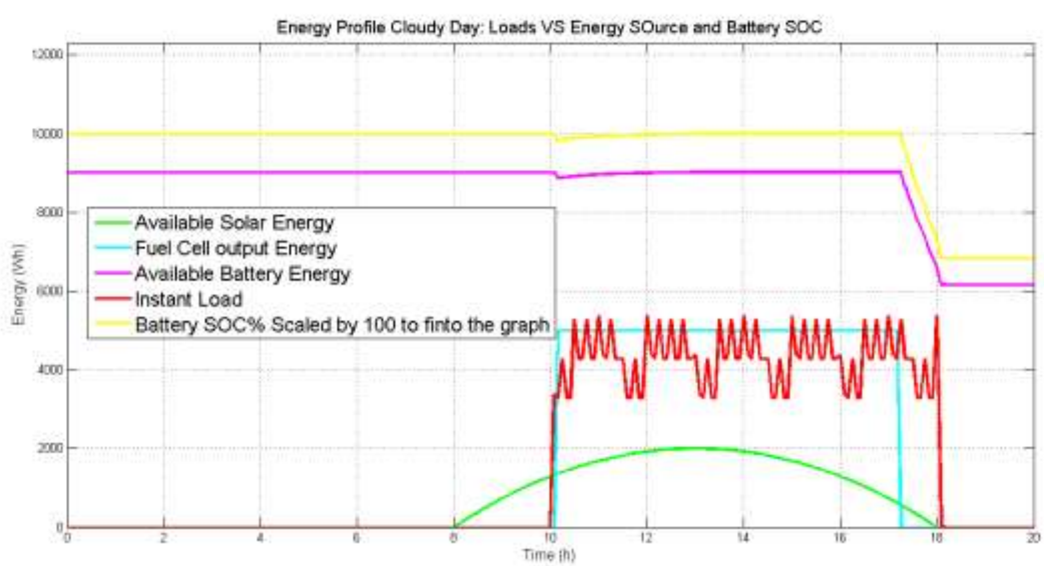

Figure 8 Load pattern and energy usage for the loads, battery, solar cell and fuel cells

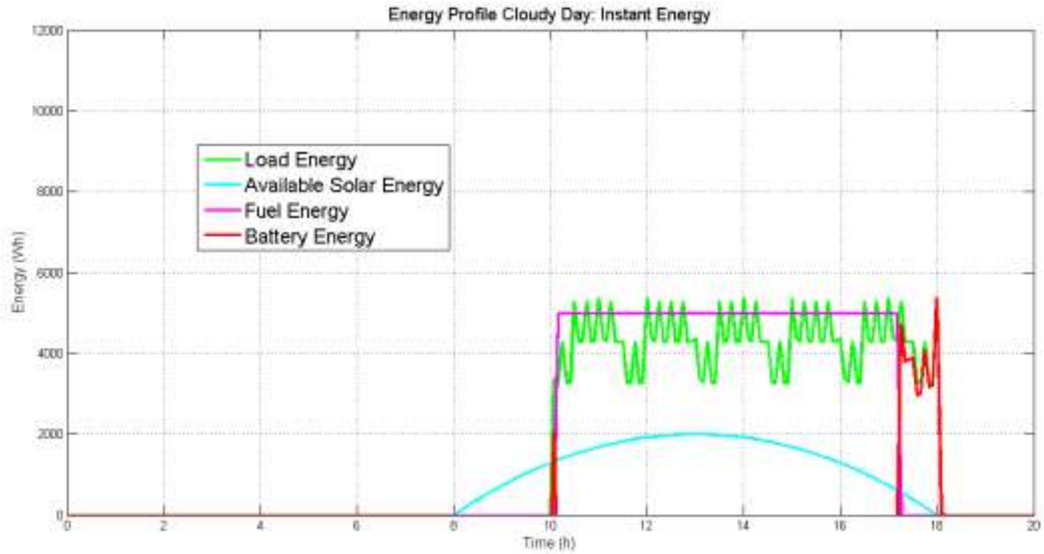

Figure 9 Instant Load pattern and energy usage for the loads, battery, solar cell and fuel cells

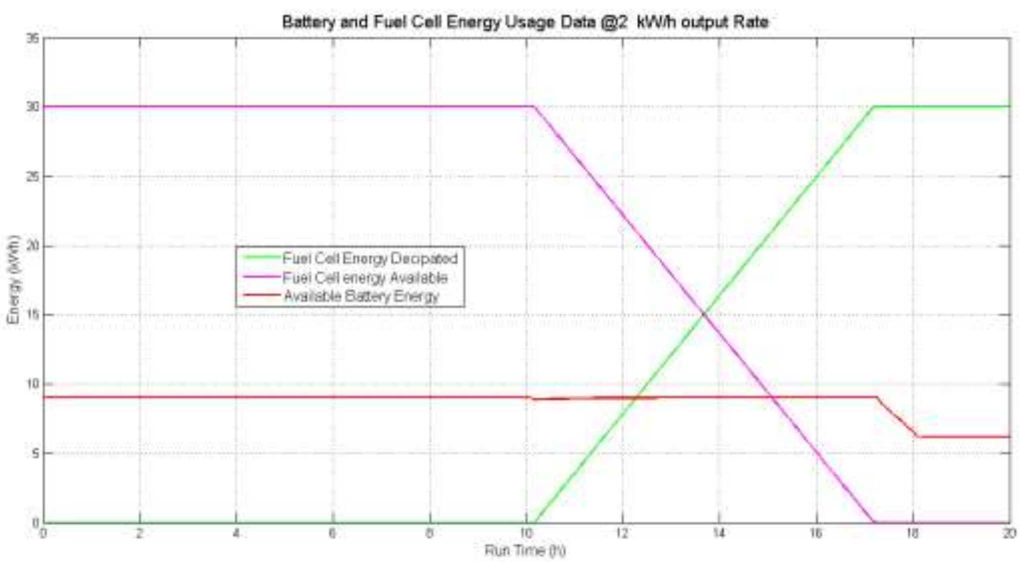

Figure 10 Available Energy of Battery and Fuel Cell

5.0 Implementation

5.1 Energy Storage Control System

To implement the system, Simulink and Matlab are used. The reason for choosing Matlab and Simulink to implement the system is because ones can define the energy usage patterns any defined systems using Matlab and build the modules using Simulink and simulate the design based on the specifications of the system and other input parameters related to the design. The simulation is set to sample the data per 5 minutes interval and the time step is set to be 1 hour for representing the operating time of the system in real time. Figure 10 and 11 in Appendix 1 shows the system block of the Energy Storage Control System and the parameter input window for initializing the initial energy stored in the battery, maximum capacity of the battery, sample time for sampling the data and the efficiency of charging the battery and the energy conversion efficiency of the regulator. These conditions are manually defined in the early stage of the project. But it fully represents the actually characteristics of the energy storage system. Figure 12 shows the implementation of the energy storage system. Later developments of this subsystem will involve in implementing the detailed design. As mention in the previous section, the system determines when the battery will be charged and discharged and constantly update the energy storage level during the operation. The initial SOC% of the battery is set to be 100 to make sure the system have sufficient energy at the beginning of the operation. The SOC% will be consistently updated during the operation to represent the real time operating in real world. During the operation, the charging system will catch the excessive energy from other sources so that the battery can absorb as much energy as possible so that it can provide maximum output energy to the loads whenever is necessary. The updated information will be monitored by the Energy Monitoring System.

5.2 Energy Monitoring System

Figure 16 shows the block of the Energy Monitoring Subsystem. It checks the status of each subsystem including the power source, loads, device specifications and solar radiations and shows the information to the users on a display. This subsystem monitors the energy in and out of each system and updates the systems synchronously by sending energy usage information to them. This makes sure that the systems are managed closely in order to increase the efficiency of the overall system and resulting in saving more energy and being more sustainable.

5.3 Fuel Cell Energy Control System

The Fuel Cell Energy Control Systems manages when the fuel cell should be switched on or off. The fuel cell generator must satisfy the following two conditions in order to be switched on.

1. When the solar cell generates not enough energy to the loads
2. When the battery is less than 100%

This setting makes sure the fuel cell is operating as the secondary energy source and result in implementing the 'Peak Shave' Energy Managing Tool.

5.4 System Integration

Once all the subsystems, including the Energy Storage Managing System, Energy Monitoring System and the Fuel Cell Energy Control System are fully functioning, the subsystems are integrated as a 'Peak Shave' Energy Managing Tool. As shown in the simulations in section 4, the integrated system operates optimally when the following conditions and settings are satisfied.

1. Fuel Cell Trigger is set to 100% SOC
2. Fuel cell total available energy of 30kWh with output energy rate is set to 5000watts/hour
3. Installation of 10 pieces of solar panels with max output energy of 320w/h
4. Battery Capacity of 10kWh
5.

6.0 Additional Work Done

In addition, some of the electric circuits are mathematically implemented using Simulink. These implementations include the actually solar cells with temperature and solar radiation input control. This system can be integrated into this system when the next person takes over the project.

7.0 CONCLUSION

In summary, the project provided a tool to access and simulate the AMS food truck systems based on the input parameters and the specifications of devices and the energy availability of the power source system. This project would help the AMS visualized how their design behaves so that they can get a more idea to optimize their designs. Moreover, the project investigated how a specific power system behaved based on two scenarios, sunny day operation and cloudy day operation. Based on these two scenarios it shown accurate results and therefore verified the design worked. The challenge of the project was the implementation of the Energy Storage Control System and the Fuel Cell Energy Control System. Since the system is a real time managing tool, it needs to track the status of the system consistently in order to accurately manage the subsystems. The foreseeable challenges for the later development of the project would be the implementation of the electric components and the integrations into this simulation tool. It is important for whoever carry on this project to understand how this system works and how to debug the system if there is any.

Appendix 1

System Implementation

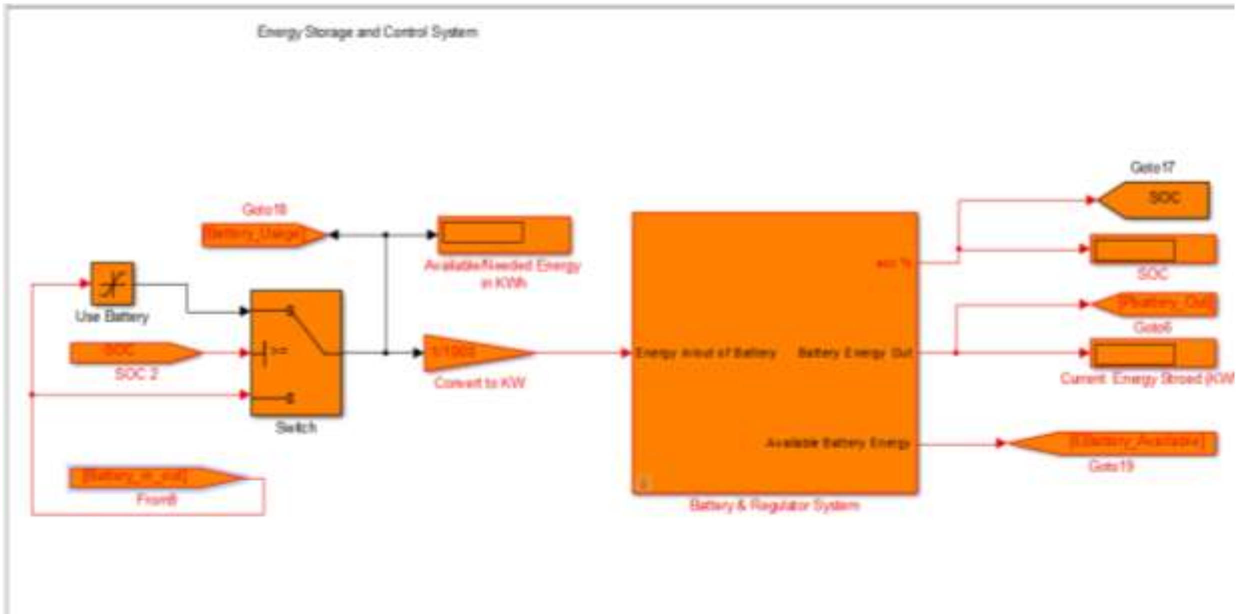

Figure 11 Energy Storage Control System

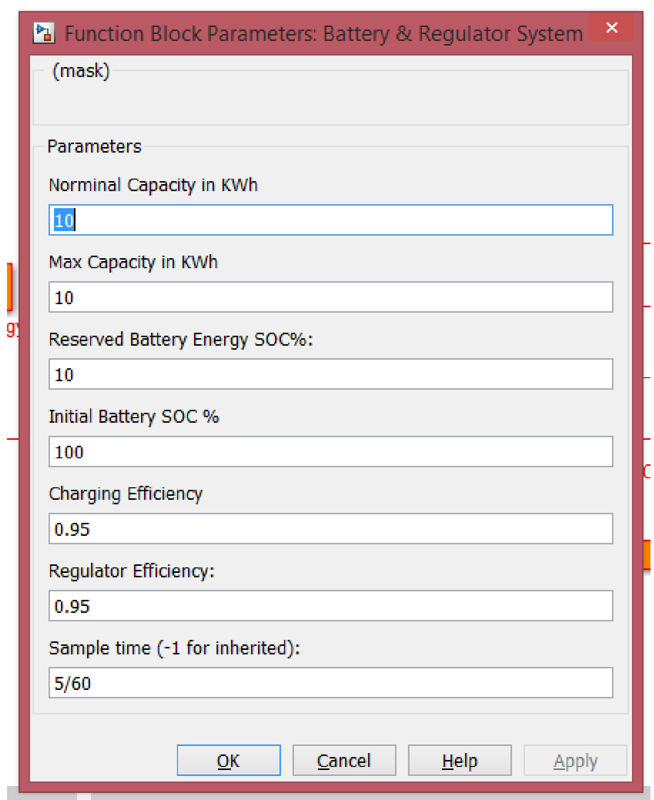

Figure 12 Energy Storage Managing Tools

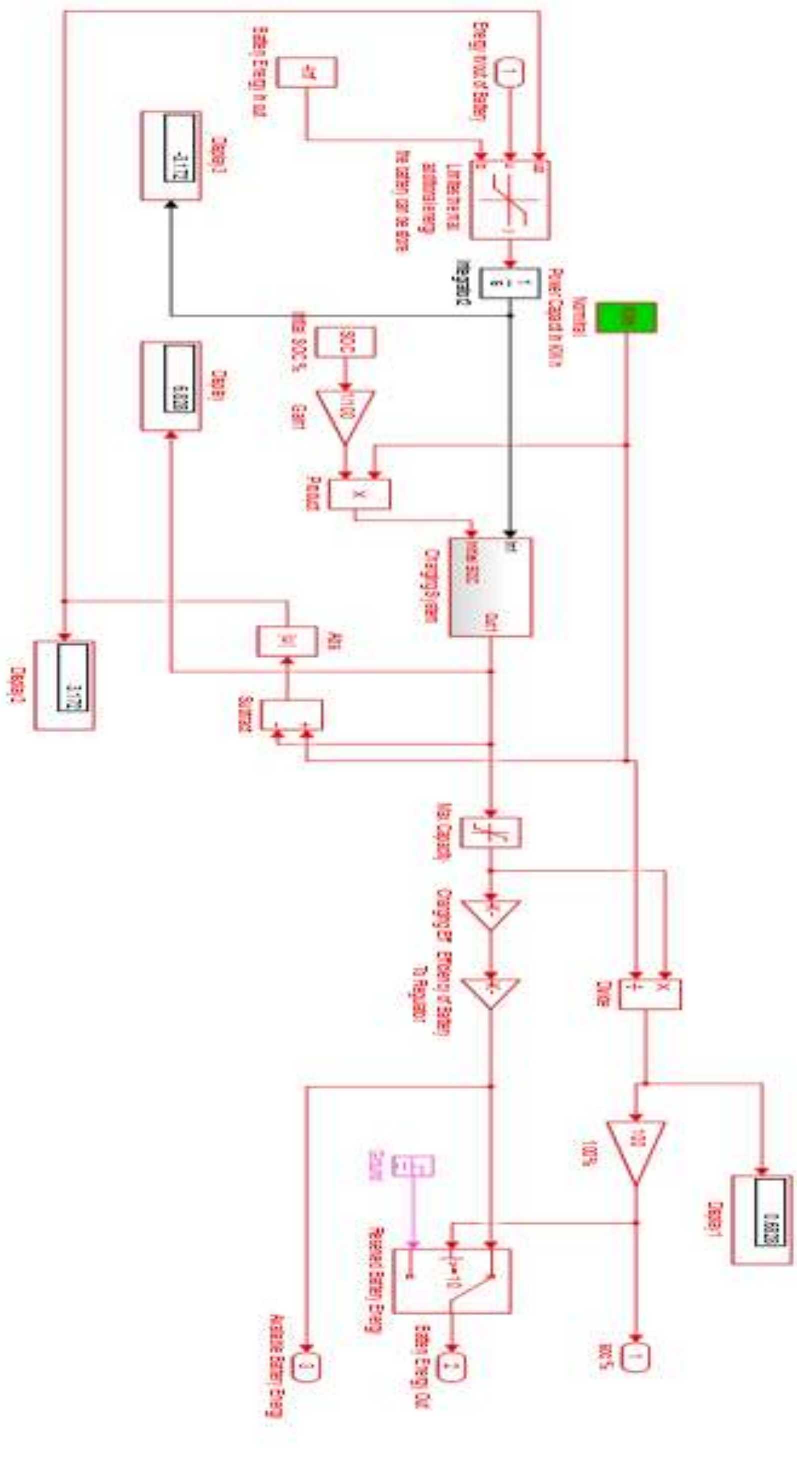

Figure 13 Energy Storage Control System Implementation Details

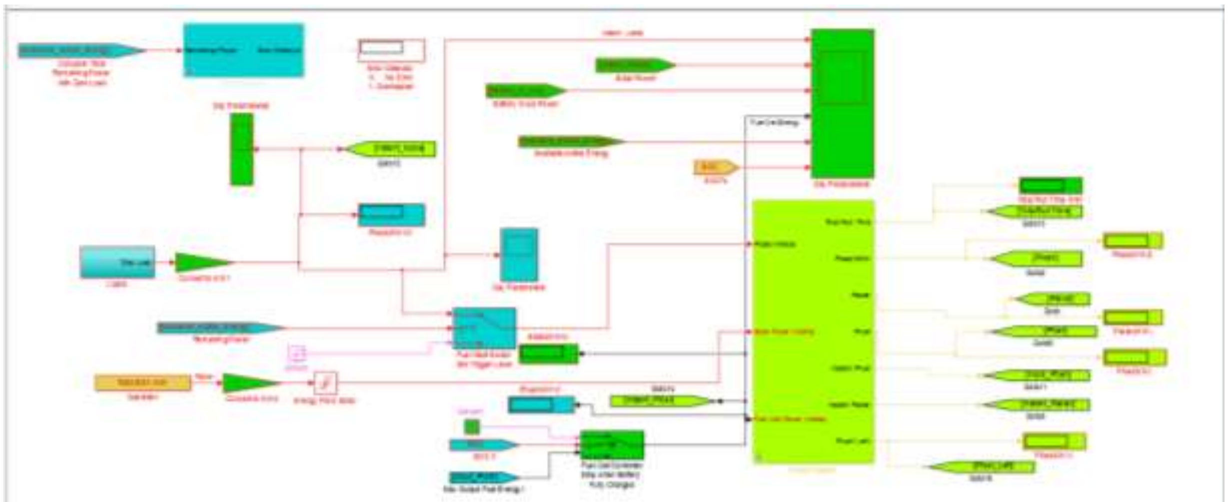

Figure 14 Fuel Cell Control System

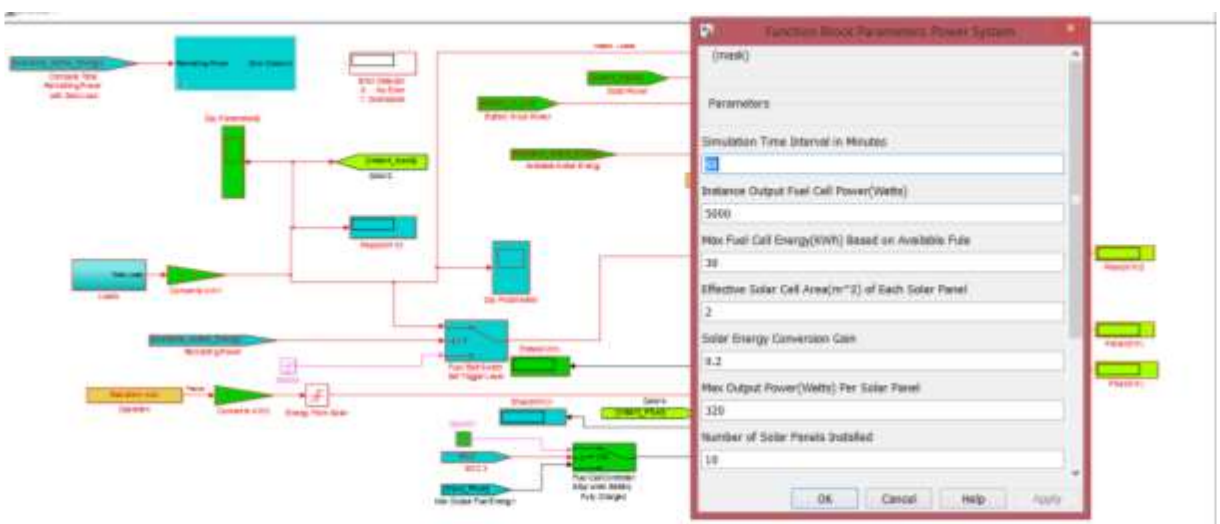

Figure 15 Fuel Cell initialization

Figure 16 Energy Monitor

Appendix 2

Mathematical Model of the Solar Cells Energy Storage System

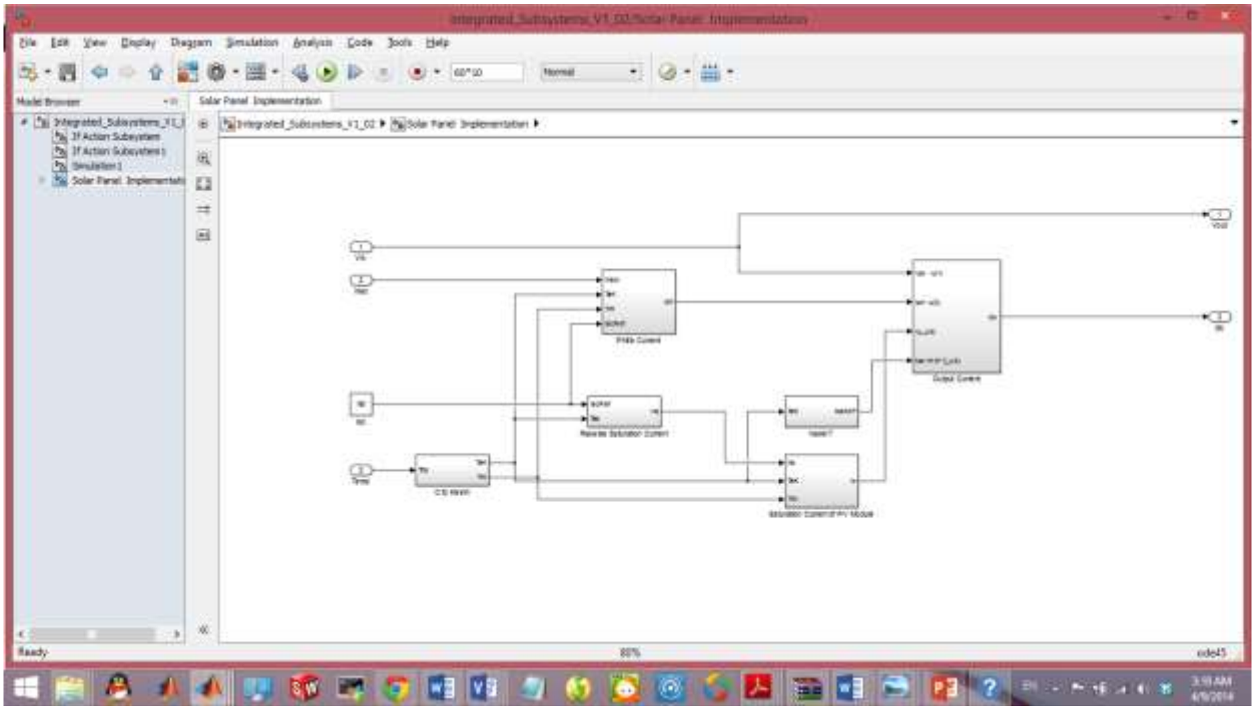

Figure 17 Mathematical Module of the Solar Cell Implementation

Appendix 3

MATLAB codes for generating input profiles

```
% EECE 490L - AMS Sustainable Food Truck
% Source / Load Profile Generation Script
% March 2014
% Paul Lusina

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Operating Conditions
% Daily Vector, 5 min intervals
clear all
Scenario_Name = 'Scenario_Files';
Sample_Time = 5; % minutes
Sample_Time_Hours = 5/60;

%Check if scenario directory exists, otherwise create it
Dir_List = ls;
Dir_Exists = 0; % flag set to 1 if directory exists
for indx = 3:size(ls,1)
    if eval(['strcmp(''' Scenario_Name ''',''' ...
            Dir_List(indx,1:size(Scenario_Name,2)) ''')'])
        Dir_Exists = 1;
    end
end

if Dir_Exists == 0
    mkdir('.',Scenario_Name);
end

% Day time vector (24 hours)
Day_Start = 0; % 24 hour time
Day_End = 24; % 24 hour time
Day_Window = (Day_End - Day_Start)*60; % minutes
Day_Number_of_Samples = floor(Day_Window / Sample_Time); % samples
Day_Vector_Hours = Day_Start:Sample_Time/60:Day_End-(Sample_Time/60);
Day_Vector_Min = Day_Start*60:Sample_Time:Day_End*60 -(Sample_Time);
Day_Vector_Steps = 1:size(Day_Vector_Min,2);

% Operating time vector
Operation_Start = 10; % 24 hour time
Operation_End = 18; % 24 hour time
Operation_Start_Sample_Number = floor(Operation_Start*60/Sample_Time)+1;
Operation_End_Sample_Number = floor(Operation_End*60/Sample_Time)+1;

% Operation Vector
Zeros_Start=zeros(1,Operation_Start_Sample_Number);
Ones_Operation=ones(1,(Operation_End_Sample_Number)...
    -Operation_Start_Sample_Number);
Zeros_End=zeros(1,(Day_Number_of_Samples)...
    -Operation_End_Sample_Number);
Operation_Vector = [Zeros_Start Ones_Operation Zeros_End];
```

```
% Save Data
Operation_Data = save_time_data( ...
    'Operation', Day_Vector_Hours, Operation_Vector, 'units', Scenario_Name);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Generation Solar Profile, Variables
Sun_Up_Hour = 6;
Sun_Down_Hour = 20;
Radiation_Max = 800; %W/m^2

% Solar Operating time vector

Solar_Operation_Start_Sample_Number = floor(Sun_Up_Hour*60/Sample_Time)+1;
Solar_Operation_End_Sample_Number = floor(Sun_Down_Hour*60/Sample_Time)+1;

% Solar Operation Vector
Solar_Zeros_Start=zeros(1,Solar_Operation_Start_Sample_Number);
Solar_Ones_Operation=ones(1,(Solar_Operation_End_Sample_Number)...
    -Solar_Operation_Start_Sample_Number);
Solar_Zeros_End=zeros(1,(Day_Number_of_Samples)...
    -Solar_Operation_End_Sample_Number);
Solar_Operation_Vector = [Solar_Zeros_Start Solar_Ones_Operation
Solar_Zeros_End];

Solar_Operation_Data = save_time_data( ...
    'Solar_Operation', Day_Vector_Hours, Solar_Operation_Vector, 'units',
Scenario_Name);


% Solar Profile, Parabolic Equation
Sun_Up = floor(Sun_Up_Hour*60/Sample_Time);
Sun_Down = floor(Sun_Down_Hour*60/Sample_Time);
Sun_Shine = (Sun_Down-Sun_Up);
Sun_Mid = Sun_Shine/2+mod(Sun_Shine+1,2);
Sun_Shine_Vector = [0:Sun_Shine];
Radiation_Points_X = [0 Sun_Mid Sun_Shine];
Radiation_Points_Y = [0 Radiation_Max 0];
Radiation_Coefficients = ...
    polyfit(Radiation_Points_X,Radiation_Points_Y,2);
Radiation_Curve = ...
    Radiation_Coefficients(1)*(Sun_Shine_Vector).^2+ ...
    Radiation_Coefficients(2)*(Sun_Shine_Vector);
Radiation_Vector = [...
    zeros(1,Sun_Up) ...
    Radiation_Curve ...
    zeros(1,Day_Number_of_Samples-Sun_Down-1)].* ...
    Solar_Operation_Vector;

% Save Data
Radiaion_Data = save_time_data( ...
    'Radiation', Day_Vector_Hours, Radiation_Vector, 'W/m^2', Scenario_Name);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Load, Microwave, Variables
```

```
Microwave_Power_Max = 1000; %Watts
Microwave_On = 5; % Minutes
Microwave_Off = 5; % Minutes

% Microwave Profile - Duty Cycle
Microwave_Vector = cycle_function(Microwave_Power_Max, ...
    Microwave_On,Microwave_Off, ...
    Sample_Time, Day_Number_of_Samples, Operation_Vector);

% Save Data
Microwave_Data = save_time_data( ...
    'Microwave', Day_Vector_Hours, Microwave_Vector, 'W', Scenario_Name);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Load, Fridge, Variables

Fridge_Power_Max = 100; %Watts
Fridge_On = 10; % Minutes
Fridge_Off = 50; % Minutes

% Fridge Profile - Duty Cycle
Fridge_Vector = cycle_function(Fridge_Power_Max, ...
    Fridge_On,Fridge_Off, ...
    Sample_Time, Day_Number_of_Samples, Operation_Vector);

% Save Data
Fridge_Data = save_time_data( ...
    'Fridge', Day_Vector_Hours, Fridge_Vector, 'W', Scenario_Name);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Load, Air Conditioning, Variables

AC_Power_Max = 1200; %Watts
AC_On = 90; % Minutes
AC_Off = 30; % Minutes

% AC Profile - Duty Cycle
AC_Vector = cycle_function(AC_Power_Max, ...
    AC_On,AC_Off, ...
    Sample_Time, Day_Number_of_Samples, Operation_Vector);

% Save Data
AC_Data = save_time_data( ...
    'AC', Day_Vector_Hours, AC_Vector, 'W', Scenario_Name);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Load, Ventilation, Variables

Ventilation_Power_Max = 200; %Watts
Ventilation_On = 5; % Minutes
Ventilation_Off = 0; % Minutes

% Ventilation Profile - Duty Cycle
Ventilation_Vector = cycle_function(Ventilation_Power_Max, ...
```

```
    Ventilation_On,Ventilation_Off, ...
    Sample_Time, Day_Number_of_Samples, Operation_Vector);

% Save Data
Ventilation_Data = save_time_data( ...
    'Ventilation', Day_Vector_Hours, Ventilation_Vector, 'W', Scenario_Name);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Load, Other Equipment, Variables

Other_Power_Max = 110; %Watts
Other_On = 5; % Minutes
Other_Off = 0; % Minutes

% Ventilation Profile - Duty Cycle
Other_Vector = cycle_function(Other_Power_Max, ...
    Other_On,Other_Off, ...
    Sample_Time, Day_Number_of_Samples, Operation_Vector);

% Save Data
Other_Data = save_time_data( ...
    'Other_Equipment', Day_Vector_Hours, Other_Vector, 'W', Scenario_Name);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Load, Grill, Variables

Grill_Power_Max = 3000; %Watts
Grill_On = 5; % Minutes
Grill_Off = 0; % Minutes

% Grill Profile - Duty Cycle
Grill_Vector = cycle_function(Grill_Power_Max, ...
    Grill_On,Grill_Off, ...
    Sample_Time, Day_Number_of_Samples, Operation_Vector);

% Save Data
Grill_Data = save_time_data( ...
    'Grill', Day_Vector_Hours, Grill_Vector, 'W', Scenario_Name);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plots

Day_Matrix = Day_Vector_Hours' * ones(1,8);
Profile_Matrix = [ ...
    Operation_Vector' ...
    Radiation_Vector' ...
    Microwave_Vector' ...
    Fridge_Vector' ...
    AC_Vector' ...
    Ventilation_Vector' ...
    Other_Vector'...
    Grill_Vector'];
```

```
figure(1)
clf
subplot(1,1,1)
plot(Day_Matrix,Profile_Matrix)
xlabel('time (h)')
ylabel ('power (W or W/m^2')
title('Operation Vector')
axis([0 20 0 5000])

% % Save Routine
% save AMS_Food_Truck_Source_Load_Profiles.mat

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function Cycle_Vector = cycle_function(Power_Max,Cycle_On,Cycle_Off,
Sample_Time, Num_Samples, Op_Vector)

% Power_Max = maximum power used (Watts)
% Cycle_On = number of minutes the device is on
% Cycle_Off = number of minutes the device is off
% Sample_Time = sample time in minutes
% Num_Samples = total number of samples to be simulated
% Op_Vector = vector in which the system is on

%Sample variables
%clear all
%Power_Max = 1000; %Watts
%Cycle_On = 5; % Minutes
%Cycle_Off = 10; % Minutes
%Sample_Time = 5;
%Num_Samples = 90;
%Op_Vector = [zeros(1,30) ones(1,30) zeros(1,30)];

% Profile - Duty Cycle
On = floor(Cycle_On/Sample_Time);
Off= floor(Cycle_Off/Sample_Time);
Cycle = [ones(1,On) ...
    zeros(1,Off)];
Num_Cycles = floor(Num_Samples/ ...
    length(Cycle));
Num_Points = Num_Cycles*length(Cycle);
Curve = Power_Max*reshape( ...
    Cycle'*ones(1,Num_Cycles), ...
    1,Num_Points);
if Num_Points == Num_Samples
    Cycle_Vector = Curve.*Op_Vector;
else
    Buffer_Zeros = zeros(1,Num_Samples-Num_Points);
    Cycle_Vector = [Curve.*Op_Vector ...
        Buffer_Zeros];
end % End If

end % End Function
```

```
function Vector_Data = save_time_data( ...
    VectorName, VectorTime, VectorData, VectorUnits, DataFolder)
%
% This function creates and saves information in a time series format which
% can be read by a simulink module.
%
% VectorName - name describing the data
% VectorTime - vector with the time samples
% VectorData - data points corresponding to the time sample
% VectorUnits - units of the data points
% DataFolder - location where the file is stored
%

% Test Data
% clear all
% VectorName = 'Test_Vector';
% VectorTime = 1:10;
% VectorData = ones(size(VectorTime));
% VectorUnits = 'm';
% DataFolder = 'Example_Files';

% Construct the timeseries vector.
Vector_Data= timeseries;
Vector_Data.Name = VectorName;
Vector_Data.Time = reshape(VectorTime,max(size(VectorTime)),1);
Vector_Data.Data = reshape(VectorData,max(size(VectorData)),1);
Vector_Data.DataInfo.Units = VectorUnits;

eval(['cd ' DataFolder])
eval(['save(''' VectorName ''', ''Vector_Data'', ''-v7.3'')'])
eval(['cd ..'])



function FigureData = save_figure(DirectoryName, CurrentFigure,
FigureFormat, ...
    FigureFile)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% This function loads time data
% DirectoryName:  Directory where data is stored
% FileName: File name with the data
%
% DirectoryName = 'Example_Files';
% FigureFile=strcat('AC','_','Profile');
% CurrentFigure = 1;
% FigureFormat = '-dpng';

eval(['cd ' DirectoryName]);
print(CurrentFigure,FigureFormat, FigureFile)
eval(['cd ..']);

FigureData = 1;
```

Scenario Generation input

```
% EECE 490L - AMS Sustainable Food Truck
% Plotting Template Script
% April 2014
% Paul Lusina
%Modified by Raymond Hou April 2014

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File Parameters
clear all
FileFolder = 'Scenario_Files';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Global Plotting Parameters - change
TimeAxisMin=0;
TimeAxisMax=20;

EnergyAxisMin = 0;
EnergyAxisMax = 40;
TimeLabel = 'Time (h)';
AxisFontSize = 12;
TitleFontSize = 14;
LegendFontSize = 16;
LineWidth = 2.5;
MarkerSize = 10;
MarkerSpacing = 0;
FigureFormat = '-dpng';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Profile - two graphs
FileName1 = 'Loads.mat';
FileName2 = 'Esolar.mat';
FileName3 = 'Efuel.mat';
FileName4 = 'Ebattery_Out.mat';
%FileName5 = 'Ebattery_in_out.mat';
%FileName6 = 'Available_Energy.mat';
%FileName6 = 'Available_Energy.mat';

Profile1 = load_time_data(FileFolder, FileName1);
Profile2 = load_time_data(FileFolder, FileName2);
Profile3 = load_time_data(FileFolder, FileName3);
Profile4 = load_time_data(FileFolder, FileName4);
%Profile5 = load_time_data(FileFolder, FileName5);
%Profile6 = load_time_data(FileFolder, FileName6);
eval(['YLabel=strcat(''Energy ('',''kWh'','')'');'])
eval(['GraphTitle=strcat(''Energy Profile Sunny Day: Accumulated
Energy'');'])
eval(['FigureFile=strcat(''PowerLoads_Profile'');'])

% Plot the figure
figure(1)
clf
PlotValues = plot( ...
```

```
Profile1.Time(1:MarkerSpacing:end),Profile1.Data(1:MarkerSpacing:end),'gd', .
..

Profile2.Time(1:MarkerSpacing:end),Profile2.Data(1:MarkerSpacing:end),'co', .
..

Profile3.Time(1:MarkerSpacing:end),Profile3.Data(1:MarkerSpacing:end),'mo', .
..
    Profile4.Time(1:MarkerSpacing:end),-
1*Profile4.Data(1:MarkerSpacing:end),'ro', ...
    ...Profile5.Time(1:MarkerSpacing:end),Profile5.Data(1:MarkerSpacing:end),
'yo', ...
    ...Profile6.Time(1:MarkerSpacing:end),Profile6.Data(1:MarkerSpacing:end),
'co', ...
    Profile1.Time,Profile1.Data,'g', ...
    Profile2.Time,Profile2.Data,'c',...
    Profile3.Time,Profile3.Data,'m',...
    Profile4.Time,-1*Profile4.Data,'r');
    ...Profile5.Time,Profile5.Data,'y');
    %Profile6.Time,Profile6.Data,'c');

% Graph Options
grid on
axis([TimeAxisMin TimeAxisMax EnergyAxisMin EnergyAxisMax]);

%LegendValues = legend(Profile1.Name, Profile2.Name)
%LegendValues = legend('Loads','Fuel','Solar','Instant Total Power','Battery
in/out(-ve:Charging +ve:Discharging)','Instant Battery Power');
LegendValues = legend('Load Energy', 'Available Solar Energy','Fuel
Energy','Battery Energy');
XValues = xlabel(TimeLabel);
YValues = ylabel(YLabel);
TitleValues = title(GraphTitle);

% Formatting Values
set(XValues, 'FontSize', AxisFontSize)
set(YValues, 'FontSize', AxisFontSize)
set(TitleValues, 'FontSize', TitleFontSize)
set(PlotValues, 'LineWidth', LineWidth)
set(PlotValues, 'MarkerSize', MarkerSize)
set(LegendValues, 'Location', 'SouthEast')
set(LegendValues, 'FontSize', LegendFontSize)

% % Save the figure
% FigureData = save_figure(FileFolder, gcf, FigureFormat, ...
%     FigureFile);
```

Reference

[1] Matlab and Simulink, The Math Works Inc. as of April 2014, http://www.mathworks.com/

[2] Samer Alsadi, Basim Alsayid, Electrical Engineering Department, Palestine Technical University-Kadoorie (PTU),Palestine-Tulkarm, as of January, 2014

[3] Tiezhou Wu, Qing Xiao, Linzhang Wu, Jie Zhang, Mingyue Wang, School of Electrical & Electronic Engineering, Hubei University of Technology, Wuhan, China, file:///C:/Users/Raymond/Downloads/SGRE20110400002_39755267%20(1).pdf , as of February, 2014

[4] Solar Cells, Northen Arizona Wind & Sun, http://www.solar-electric.com/kyocera-kd320gx-lfb-320-watt-polycrystalline-solar-panel.html, as of Feb, 2014

[5] 12000 Watts Inverter, AIMS Power, http://www.aimscorp.net/12000-Watt-Inverter-Charger-48-volt-120-240vac.html, as of March 2014

[6] Top 10 Energy-Saving Tips for the Commercial Kitchen, http://www.foodservicewarehouse.com/education/going-green/top-10-energy-saving-tips-for-the-commercial-kitchen/c28108.aspx, as of March 2014

http://ecee.colorado.edu/~ecen2060/matlab.html

http://www.academia.edu/3624997/Maximum_Power_Point_Tracking_Simulation_for_Photovoltaic_Systems_Using_Perturb_and_Observe_Algorithm