Small Scale Stand Alone Hybrid Solar PV Generation System for EE 452 Lab

FINAL DOCUMENTATION

Team Number: sddec19-06
Client/Adviser: Professor Ajjarapu
Team Members:
Chief Engineer: Adam Schroeder
Test Engineer/ Webmaster: Hussein Ghitan
Meeting Facilitator: Anna Schulte
Team Email: sddec19-06@iastate.edu
Team Website: http://sddec19-06.sd.ece.iastate.edu

Contents

1 Introduction

1.1 Project statement

The focus of this project is for the use of a stand alone PV system in the EE 452 lab for future students to learn the characteristics of stand alone solar PV systems, and maximum power point tracking (MPPT). The senior design team's main objective was to re-engineer the system to ensure its functionality as an effective tool for students.

1.2 Purpose

In recent years, renewable energy sources, such as solar, have gained more prominence in the power industry. This has lead to an increased demand for their development, putting a greater importance on the understanding of solar energy for students as they move into the workforce. The re-engineered PV system and labs for the EE 452 power systems course are designed to allow students to understand these concepts by providing hands on experiments.

1.3 Goal

The original goal of our project was to add on to an already working solar PV system. However, when our team received the system, it did not operate. This transitioned our main goal to re-engineer the solar PV system and safely demonstrate how it operates. To do this, the team had to redesign the hardware of the physical system and its lab manuals. These lab manuals use Simulink simulations of a solar PV system and compare them to a physical system. In order for this goal to be obtained, the senior design team had to:

- Gain a complete understanding of PV systems by diving through documents from previous groups as well as outside sources in order to explain the system's functionality to students via lab manuals and the PV system.
- Re-engineer the system to get the system operating, safe, and easily understandable for students to use.
- Evaluate and revise lab manuals allowing them fully grasp the characteristics of solar PV systems and the properties of MPPT so that students are able to learn these concepts.

2 Deliverables

Re-engineer the system:
None of the system's components worked properly when the team received the project assignment. Complete understanding of the PV system is needed in order to identify the technical issues and re-engineer the entire system from the solar panel to the load.

Functionality of the system:
The system is made up of two fundamental circuits, a power circuit and a measurement circuit. Each individual component is within one of these circuits. The power circuit includes solar panels, batteries, wiring, relays, a MPPT, an inverter, and a three-phase converter. The measurement circuit is made up of multimeters, both temperature and irradiance sensors, Arduino, and displays. The entire system functions properly, and an accurate schematic of the circuitry has been created for future students and faculty.

Testing:
Tested both the Simulink simulation and physical system to ensure they both follow the correct characteristics of a solar PV system. The simulation and hardware accurately show the relationships they should and match one another with limited error.

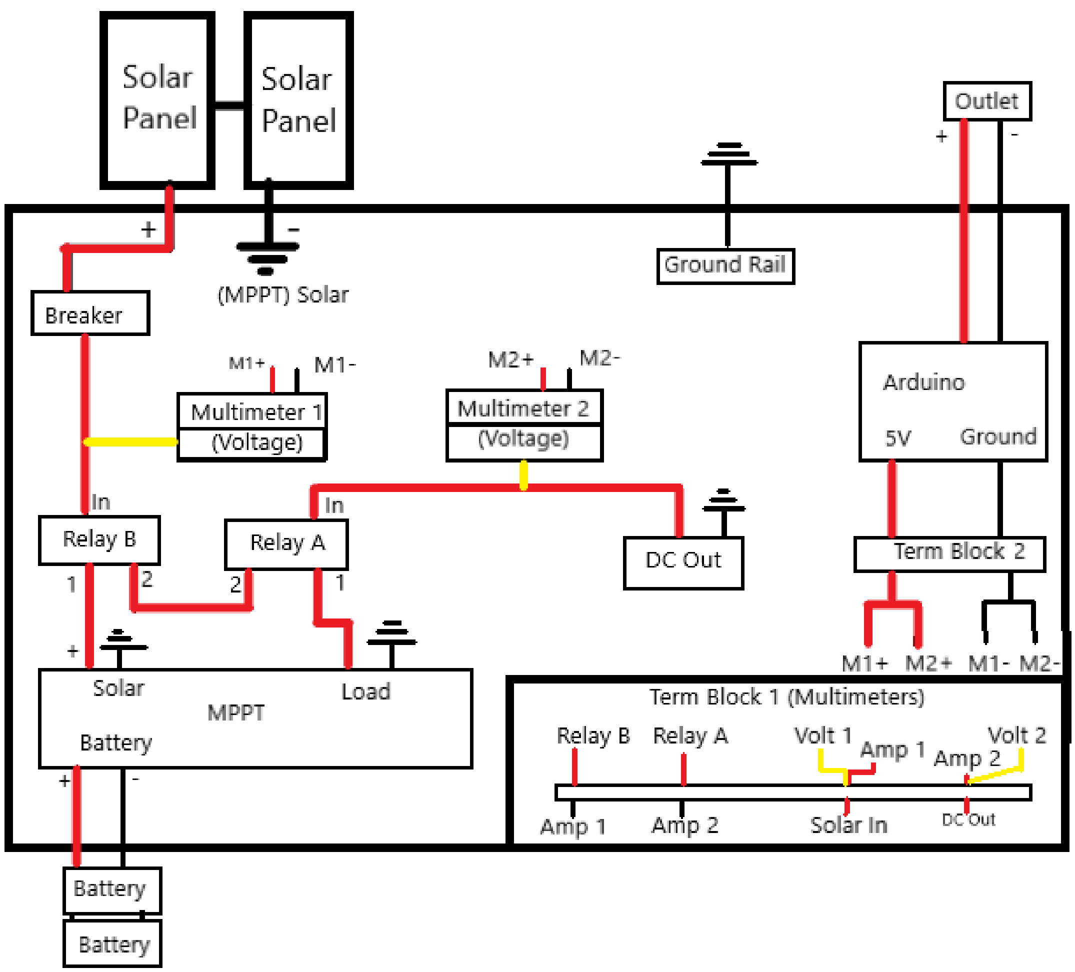
Lab manuals and first session of EE 452 lab:
The lab manuals contain clear and accurate instructions for the experiments that will be conducted in the EE 452 lab. They need to match any changes or corrections our team has made to the system. All portions of the lab instructions are clear and relevant for the students understanding. From our lab manuals and knowledge of the system we should be able to successfully run the EE 452 lab.

Safety:
The re-engineering of both fundamental circuits impacts the safety of the system. The team successfully checked the system's safety by running tests confirming short circuits do not occur, breakers work properly, each component is grounded properly, and system is up to IEEE safety standards.

Safety is a large part of this project. A standalone solar PV system has potential risks as there can be up to 270 Watts of power flowing through the devices and into the load. With up to 8 Amperes of current, touching an open terminal can seriously harm a person and even possibly be fatal.

3 Design

The design of the PV system consists of multiple components, each playing a key role in the objective. Two solar panels (each 135W) are connected to two relays, controlled by a switch, allowing for voltage to flow either to the Maximum Power Point Tracker, or directly to the load. This MPPT allows the PV arrays to generate maximum power at all times. It does this continually by repeatedly changing the load seen by the PV array to get maximum power output. The ideal load for maximum power output depends upon irradiance and temperature, both of which vary each second. When the system is in the direct solar setting, an external buck/boost converter can be implemented before the load in order to change the duty ratio to obtain the maximum power point for the load without using the MPPT. Two 12V batteries in series Tare connected to the MPPT, as well as the two solar panels. The power from the solar panels is used for the load as well as charging the batteries. Together, through the MPPT, they can power a 600-watt inverter and thus generate 120V, AC for any load through the single phase output. The inverter is also connected to a three-phase motor controller allowing the system to produce a three-phase AC voltage.



Photovoltaic System Schematic

3.1 System Specifications

- Workstation should be organized and clean.
- System to be completely safe.
- No external tools should be used other than what specified in the lab manuals.
- Simulation and software model should be consistently comparable to the physical system.
- Hardware should be clearly labeled and easy to operate.

3.1.1 Non-Functional

- Understand the main components of the physical system such as MPPT, and I-V and P-V relationships.
- Knowledge of each part of the system in order to do the labs safely and according to the provided manuals.
- The physics of how batteries charging and discharging, using the inverter, buck-chopper converter and open circuit voltage of the solar panels.

3.1.2 Functional

- Solar Panels provide an acceptable amount of power to run experiments that encompass the systems functions.
- Proper wiring and connections to allow for running direct solar and MPPT to a load.
- Batteries with correct voltage rating that can be charged and provide the correct amount of voltage to an MPPT in conjunction with solar power.
- Ability to change loads through various supply standards such as DC, AC, and three phase.
- Ability to measure voltages and currents from solar and load.

3.1.3 Standards

There are a variety of standards that apply to this project. Since solar energy is a rapidly growing concept and is making noticeable improvements to renewable energy sources in the industry, standards are always being refined and developed. There are several IEEE guides and recommended practices that pertain to this project:

- IEEE Recommended Practice for Testing Insulation Resistance of Electricity outlines the process of testing the insulation resistance of polarized machine windings. It goes into definitions, reasoning for importance, testing conditions and procedures, and expected outcome of different systems.
- IEEE Guide for Insulation maintenance of Electric Machines is an overview of the standard procedure that goes into testing and maintaining different electric machines. It is focused around how one can prevent problems or how to solve issues as they arise.

- IEEE Guide for Array and Battery Sizing in Stand-Alone Photovoltaic (PV) Systems is focused on providing a structure for determining the battery capacity needed to have an independent PV system. It also covers a lot of definitions, and information on what it will take to have a PV as your only power source with batteries as your storage.
- IEEE Guide for Selecting, Charging, Testing, and Evaluating Lead-Acid Batteries Used in Stand-Alone Photovoltaic (PV Systems) mainly discusses lead-acid battery charging requirements in relation to operational parameters of the PV system and battery performance. It provides a guideline to select and test the battery. It also gives very thorough descriptions of hazards that can occur when handling batteries, how to prevent these problems, and how to deal with situations when a problem does occur.

If the project were to cease to be stand-alone and connect to the grid, then the standards of importing solar energy to the grid by the City of Ames would apply. Existing local, state, and federal code makes it unlikely that the standards above will apply exactly outside of a classroom setting; it is important that students learn about IEEE standards and follow them to ensure safety and ethics in the classroom as well as in the field. Safety of operators comes first, and it is our responsibility to ensure that safety throughout use of the project.

3.2 Design Analysis

A complete disassembly and redesign of the entire system was necessary. The wiring of both power and display circuits had several different short circuits within them. This caused there to be no voltage across the DC output for any type of load. None of the displays operated since the original wiring for the multimeters fed them a higher voltage than what they were rated for. This resulted in both of the original multimeters getting fried and being no longer operational. Safety during the design process was a big focus since the solar panels and batteries both produce a significant voltage. This provided a threat of electric shock as well as the potential to send too much voltage through components in the system. The redesign of the system had to ensure sending the proper voltage to all of the components in the system as well as taking measurements of voltage and current from the correct locations within the system for proper analysis of the system.

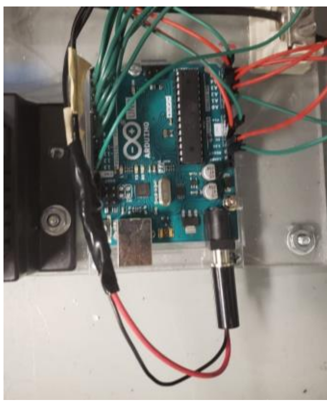
3.3 Safety Recommendations for Upcoming Students:

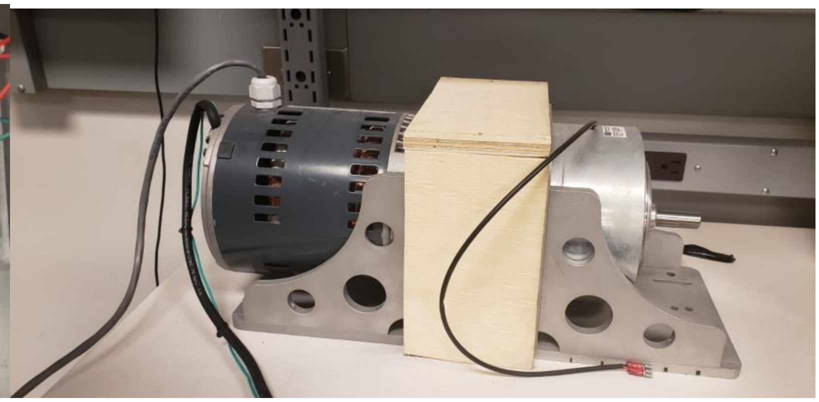
- Enclosure ground.
- Move the 24V batteries off the table (e.g. underneath the table).
- Install more stationary voltmeters to avoid using external measurements tool during the lab implementations.
- Install battery circuit breaker.

4 Testing/Development

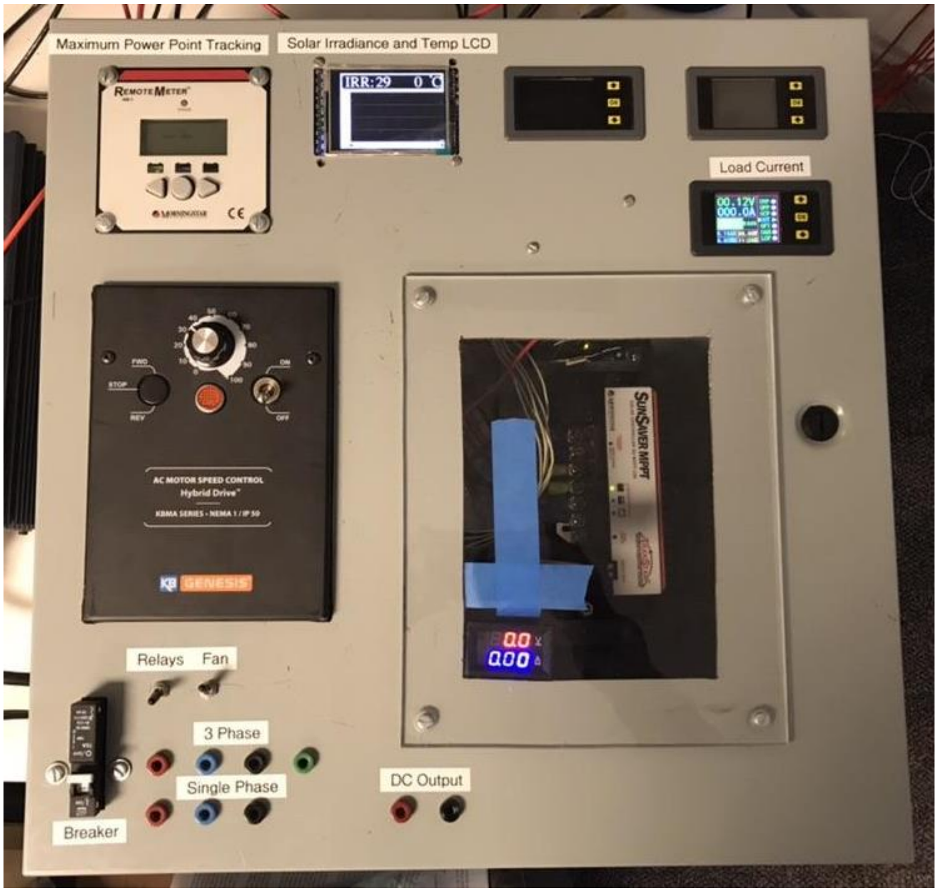
4.1 Interface/Hardware Specifications

Students now have the project's workstation safe and ready to use. The workstation included but not limited to the enclosure where the Arduino, irradiance sensor, solar panel input, temperature sensor input, relays, circuit breaker, solar open circuit voltmeter, load current sensor and displays are connected inside. The batteries, 3-phase motor and the loads are also at the workstation. Students can do their measurements and implement the lab experiments safely. They can observe and understand the power generated by the panels and how that power flows into the system and charge the batteries. Students may gain a good amount of understanding the PV system by following the provided lab manuals of the Power Electronics lab.





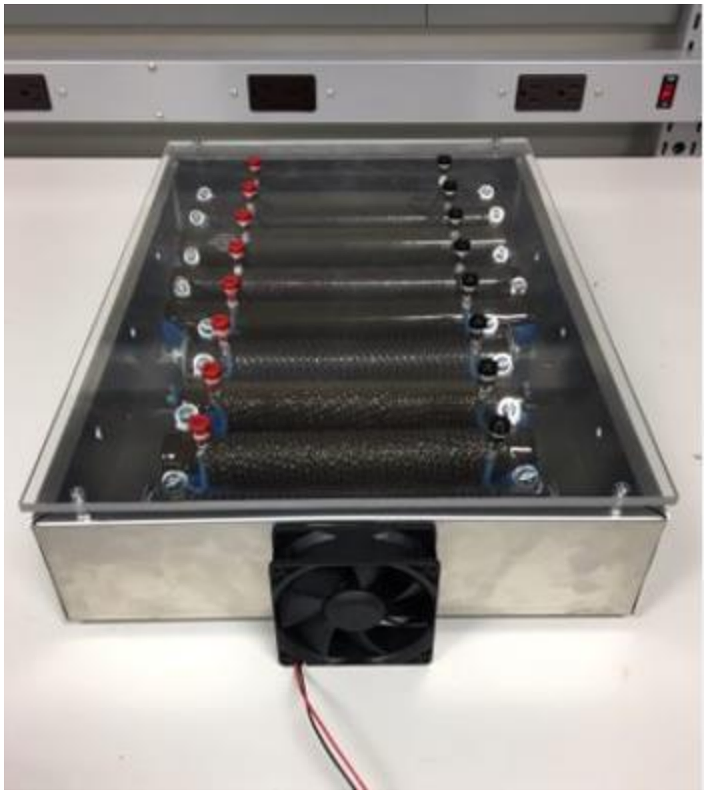
Arduino UNO **3-Phase Motor**



Top View of The Enclosure

Enclosure specifications:

- Current setup includes an Arduino connected to a thermometer and pyranometer.
- Data is stored and displayed using 2 libraries (GFX and TFT LCD).
- A TFT LCD screen displays the irradiance and temperature.
- Voltmeter to display the solar open circuit voltage.
- 3-phase speed control unit.
- Solar Panel characteristics and load characteristics for an AC load are also displayed.



Resistor Bank Used As Load

Resistors Load specifications:

- Aluminum Enclosure 17" x 2"
- Resistors: 8" x 2" and 10" x 2"
- Resistance Needed: 0.5Ω - 200Ω
- Current Power Rating: 300 W
- Resistor Values: 1, 2, 5, 16, 25, 100, 150, 200 Ω
- Provides Max Resistance: 499 Ω
- Provides Min Resistance: 0.54 Ω

Circuit Safety Design

- Fully Enclosed Circuit Container
- Locking Access Panel
- Wire Management, and High Voltage Isolation
- Circuit Breaker to Disconnect Solar Panels
- Wire Insulation
- Grounding the Ground Rail

4.2 Software

There are two portions of the project that contain software. The system uses modules in Simulink and an Arduino that contains a code that outputs the current irradiance and temperature onto a display. The code has the display track the irradiance over time on a graph. The previous team has modeled Simulink simulation and the Arduino code. However, our group had to make changes to the code so that the display would show the temperature in Celsius instead of Fahrenheit. Our team also had to program the code into a new Arduino and install it into the system when the old Arduino test failed. Below are the provided modules:

- Booster_Inverter_Combo
- Complete System
- PV_cell
- PV_cell_array
- MPPT _Battery_ Controller
- MPPT_Manual
- Resistive_Load_Only

4.3 Process

To begin the project we determined that research on how solar PV systems work was necessary in order to get the system functioning. We looked at previous groups information as well as external sources in order to gain a complete understanding of PV systems. More importantly, how the given, non-operational, system in the EE452 lab was intended to work. The importance of learning this is to be able to help students grasp important concepts as they interact with our system for the first time. As we researched and began to understand the details of how the system should work, we were able to identify errors in the wiring and programming.

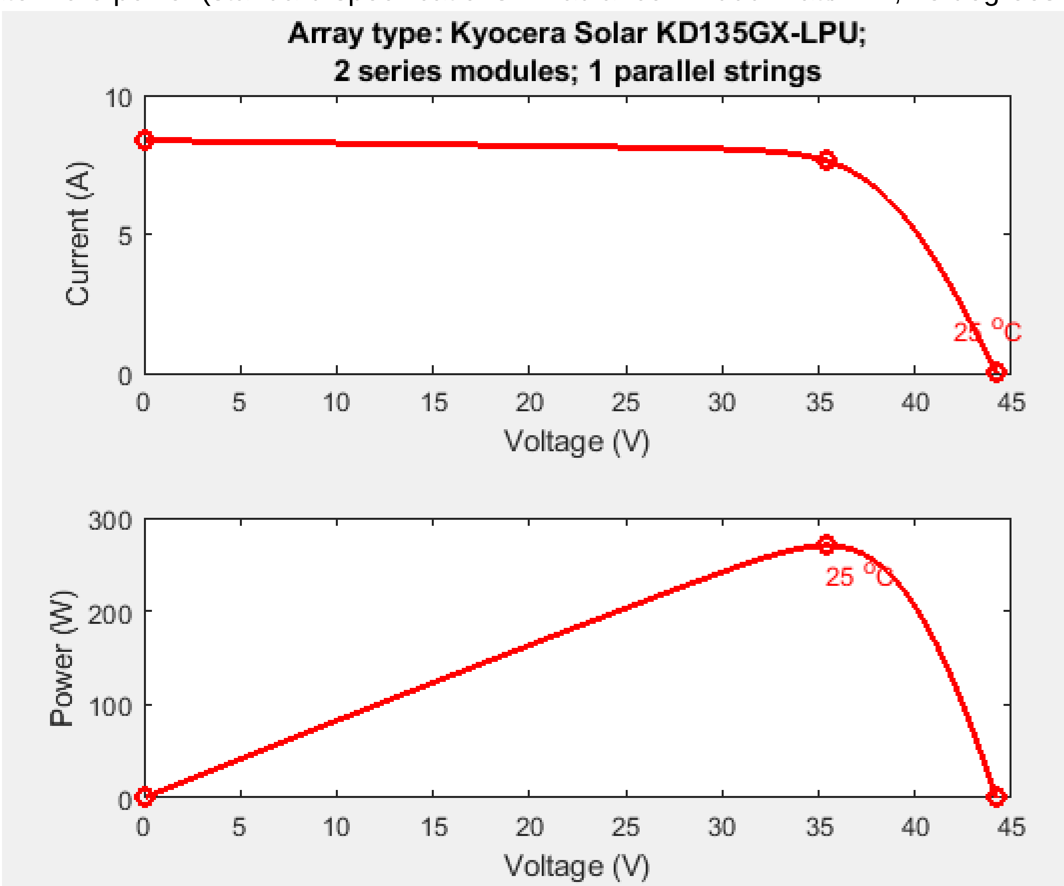
When our team received the system, the first thing we noticed was that there was no voltage across any outputs and none of the displays turned on. As we began looking into what we originally thought to be a simple problem, our team discovered several problems with the system. This meant that re-engineering would be necessary. Safety was an extremely important topic that we needed to be aware of while carrying out this process. As we worked to resolve the issues, we made sure to disconnect the solar panels and batteries, check voltages, and cover open wires before working on any part of the system.
The first task our team needed to do was replace one of the 12V batteries that was no longer holding a charge. The main breaker for the solar input was flipped in the opposite direction not allowing for the voltage to be disconnected from the system. Voltage was going into the old multimeters but was not coming out implying a short circuit within them. The top of one multimeter was burned and melted, due to being fed a higher voltage than rated and were not grounded correctly.

In Spring semester our team was pressed for time, since the labs using this system in the EE 452 course were planned for late April. Our team had to quickly disassemble the system and temporarily redesign it so that the main power circuit worked without the displays. Allowing the experiments could be run at the end of the spring 2019 semester. The first task our team needed to accomplish was to replace one of the 12V batteries that was no longer holding its charge. Next, we had to replace the main solar breaker in the correct direction so the input solar voltage could be disconnected from the system easily. Then, our team had to flip the polarity of the solar input which was causing our MPPT to not function properly. The MPPT was reading a negative voltage from the solar panels and when compared to the 24V from the batteries the system pulled only from the batteries. Finally making a few more connections the main power circuit was fully operational in time for the experiments to be run.

In the Fall semester our team turned to drawing up a schematic to include new multimeters and displays. Our team then ordered new multimeters, and once received, our team externally tested these multimeters to ensure they functioned correctly. We then installed them to read the voltage from the solar and from the load. Unfortunately, we installed one multimeter incorrectly and broke it. Doing more research on the old ampere, we were able to correctly power it by connecting it to the batteries. When all set and done our new display measures the direct solar voltage at all times and the load current.

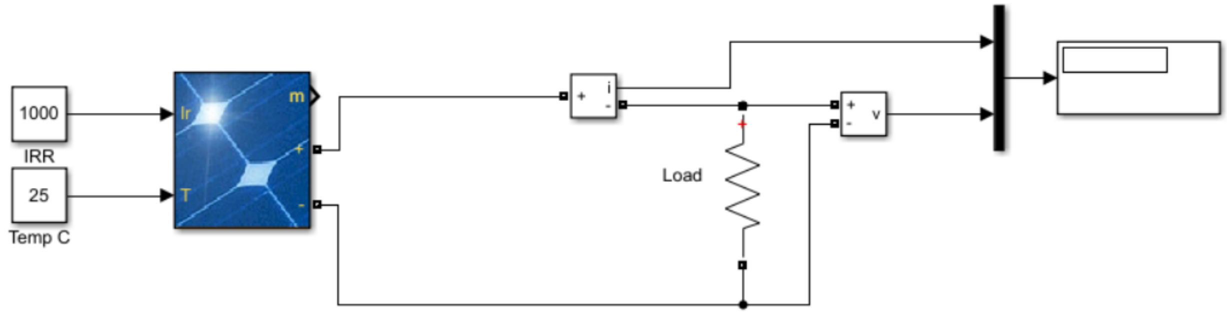
Each component of the system was tested prior to implementation so that each part of the system was able to function cohesively.

This project is weather dependent, ideally we would have tested out the system on high irradiance and low temp days as these conditions are the perfect conditions for the solar cells to generate more power (standard specifications : Irradiance = 1000 Watt/m^2, 25 degrees C).



I-V and P-V Characteristics of The Standard Specifications

The team had the simulations as a reference however, measurements from hardware and from the model in MATLAB Simulink never completely matched due the error percentage but the pattern is always the same.



Solar-Load Simulink Model

R	Irr	Vsim	Vhard	Psim	Phard	Temp = 25
OC	75	39.74	39.03	0	0	
200	73	38.51	35.66	7.4151005	6.358178	
143.2	69	38.24	32.37	10.2115754	7.31715712	
100	67	37.34	25.13	13.942756	6.315169	
25	65	13.45	6.804	7.2361	1.85177664	
16	64	8.523	4.364	4.54009556	1.190281	
5	63	2.682	1.397	1.4386248	0.3903218	

Max Power Measurements from Physical System And Simulink

Lab manuals

A majority of the team was not enrolled in the EE452 course, making it difficult for the team to determine if the lab manuals encompassed the concepts of this course. Even this being the case, with their understanding of stand alone systems the team was still able to successfully revise the lab manuals to contain the deliverables needed for understanding of a stand alone solar system.

5 Results

Through our team's research and understanding of solar PV systems, we were able to determine the Simulink model provides accurate I-V and P-V graphs as well as the maximum power tracking point for every load. In order to verify that our design of the hardware operates correctly, our team compared the system to the simulation in Simulink. Results showed that the Simulink model and the hardware had similar results under the same irradiance and temperature conditions. After further testing it was determined that when comparing the simulation and hardware, there is an inverse relationship between the irradiance and percent error.

6 Conclusions

Over the past year our team has safely disassembled, redesigned, and implemented a solar PV system. The re-engineered system effectively explains how solar PV systems work using multiple experiments in the EE 452 Power Systems lab. Our team discovered that there is a certain degree of error between the simulation in Simulink and the hardware. The error is dependent on the amount of irradiance. This is most likely because the physical system does not operate at one-hundred percent efficiency like the simulation does. Through this senior design process our team has gained a complete understanding of maximum power point tracking, how to construct a solar PV system, and how to effectively explain how the system works to individuals.

7 Appendices

7.1 Arduino Code

```
// IMPORTANT: Adafruit_TFTLCD LIBRARY MUST BE SPECIFICALLY
// CONFIGURED FOR EITHER THE TFT SHIELD OR THE BREAKOUT BOARD.
// SEE RELEVANT COMMENTS IN Adafruit_TFTLCD.h FOR SETUP.

#include <Adafruit_GFX.h>    // Core graphics library
#include <Adafruit_TFTLCD.h> // Hardware-specific library
#include <Fonts/FreeSerif24pt7b.h>
#include <Fonts/FreeSerif9pt7b.h>

// The control pins for the LCD can be assigned to any digital or
// analog pins...but we'll use the analog pins as this allows us to
// double up the pins with the touch screen (see the TFT paint
example).
#define LCD_CS A3 // Chip Select goes to Analog 3
#define LCD_CD A2 // Command/Data goes to Analog 2
#define LCD_WR A1 // LCD Write goes to Analog 1
#define LCD_RD A0 // LCD Read goes to Analog 0

//#define LCD_RESET A4 // Can alternately just connect to Arduino's
reset pin
//A4 is used for irradiance ADC input

// When using the BREAKOUT BOARD only, use these 8 data lines to the
LCD:
// For the Arduino Uno, Duemilanove, Diecimila, etc.:
//   D0 connects to digital pin 8  (Notice these are
//   D1 connects to digital pin 9   NOT in order!)
//   D2 connects to digital pin 2
```

```
//   D3 connects to digital pin 3
//   D4 connects to digital pin 4
//   D5 connects to digital pin 5
//   D6 connects to digital pin 6
//   D7 connects to digital pin 7
// For the Arduino Mega, use digital pins 22 through 29
// (on the 2-row header at the end of the board).

// Assign human-readable names to some common 16-bit color values:
#define  BLACK   0x0000
#define BLUE    0x001F
#define RED     0xF800
#define GREEN   0x07E0
#define CYAN    0x07FF
#define MAGENTA 0xF81F
#define YELLOW  0xFFE0
#define WHITE   0xFFFF
#define GRAY    0x5AEB
#define SCREEN_WIDTH 320
#define SCREEN_HEIGHT 220
#define PLOT_WIDTH 290
#define PLOT_HEIGHT 160
#define PLOT_X_OFFSET 30
#define PLOT_Y_OFFSET 60
#define PLOT_Y_WIDTH 10
#define MAX_IRRADIANCE 1150
#define SCREEN_ROTATION 1
#define TFT_SCREEN_IDENTIFIER 0x9341

Adafruit_TFTLCD tft(LCD_CS, LCD_CD, LCD_WR, LCD_RD, 0);
// If using the shield, all control and data lines are fixed, and
// a simpler declaration can optionally be used:
// Adafruit_TFTLCD tft;
uint16_t irradiance[33] = { };
long firstTaskMillis = 0;
long secondTaskMillis = 0;
long thirdTaskMillis = 0;
long oneSecondTask = 100;
long twoSecondTask = 2000;           // interval at which to run
(milliseconds)
//change this variable to extend the plot refresh rate. It is at 5 sec
now
long fiveSecondTask = 5000;
float sensorReading;
long averageReading = 0;
```

```
void setup(void) {
  Serial.begin(9600);
  Serial.println(F("TFT LCD test"));
  Serial.print("TFT size is "); Serial.print(tft.width());
Serial.print("x"); Serial.println(tft.height());
  tft.reset();
  tft.begin(TFT_SCREEN_IDENTIFIER);
  initializeDisplay();

  Serial.println(F("Done!"));
}

//Main function where program runs
void loop(void) {
 unsigned long currentMillis = millis();


  //two hundred millisecond task
  if(currentMillis - thirdTaskMillis > oneSecondTask){
    thirdTaskMillis = currentMillis;
    //read voltage
    averageReading = averageReading + analogRead(A5);
  }

  //two second task
  if(currentMillis - firstTaskMillis > twoSecondTask) {
      // save the last time you blinked the LED
      firstTaskMillis = currentMillis;

      //double sensorReading = averageReading/5;
      //averageReading = 0;
      // Convert the analog reading (which goes from 0 - 1023) to a
voltage (0 - 5V):
      //float voltage = analogRead(A4) * (5.0 / 1023.0);
      //shift array values for plot data

      //sensorReading = (voltage * 245); //use Example 3 for
conversion

      int sensorValue = analogRead(A4);
      // Convert the analog reading (which goes from 0 - 1023) to a
voltage (0 - 5V):
      float voltage = sensorValue * (5.0 / 1023.0);
      float temp = voltage * 245; //use Example 3 for conversion
```

```
      sensorReading=temp;


      float value=averageReading/20;          //read from A0
      averageReading = 0;
      float volts=(value/1024.0)*5.0;      //conversion to volts
      float something= volts*100.0;             //conversion to temp
Celsius
      float tempC=something;
      displayIRR(temp, tempC);

      Serial.print("temperature= ");
      Serial.println(tempC);
      Serial.print("Voltage: ");
      Serial.println(voltage);
      Serial.print("Irradiance: ");
      Serial.println(temp);

  }

  //five second task
  if(currentMillis - secondTaskMillis > fiveSecondTask) {
      secondTaskMillis = currentMillis;
      for (uint8_t j = 0; j<33; j++){
        irradiance[j] = irradiance[j+1];
      }
      irradiance[32] = sensorReading;
      drawPlot(irradiance);
  }


}

//Initialization instructions
void initializeDisplay() {
  tft.fillScreen(WHITE);
  tft.setRotation(SCREEN_ROTATION);

  //Draw black round rectangle for top text
  tft.fillRoundRect( 5, 5, 310, 45, 8, BLACK);

  //Setup text and cursor for title
  tft.setCursor(15, 36);
  tft.setTextColor(WHITE);
  tft.setTextSize(1);
```

```
  tft.setFont(&FreeSerif24pt7b);
  tft.print("IRR:");
  tft.setCursor(284, 42);
  tft.print("C");

  //Draw plot background
  initPlot();
}

void drawPlot(uint16_t data[]){
  //Clear current plot before drawing new values
  initPlot();
  //divisor value for taking care of varible type mismatches
  double divisor = (double)MAX_IRRADIANCE/(SCREEN_HEIGHT-
PLOT_Y_OFFSET);
  //draw all new values
  for(uint16_t i = PLOT_X_OFFSET; i<=(SCREEN_WIDTH - PLOT_Y_WIDTH);
i=i+PLOT_Y_WIDTH){
      //Draw each line. Calculations are done to keep the plot size
relative
      tft.drawLine(i, SCREEN_HEIGHT-((data[i/PLOT_Y_WIDTH])/divisor),
                   i+PLOT_Y_WIDTH,
                   SCREEN_HEIGHT-((data[(i/PLOT_Y_WIDTH)+1])/divisor),
                   WHITE);
  }
}

void displayIRR(uint16_t i, float temperature){
  //Clear the previous IRR value for printing
    tft.fillRect(105, 7, 180, 43, BLACK);
    //Move the cursor to the correct printing location
    tft.setFont(&FreeSerif24pt7b);
    tft.setCursor(110, 42);
    tft.setTextColor(WHITE);
    tft.setTextSize(1);
    //Print the irradiance value
    tft.print(i);
    //temperature = 72;
    //set cursor to appropriate location
    if(temperature>99){
      tft.setCursor(212, 42);
    }
    else{
      tft.setCursor(230, 42);
    }
```

```
    //convert digital value to analog temp and print
    //math goes here
    tft.print((int)temperature);
    //Draw degrees circle here
    tft.drawCircle(281,10,3,WHITE);
    tft.drawCircle(281,10,2,WHITE);


}

void initPlot() {
  //Set font to default
  tft.setFont();
  //Set font color to black
  tft.setTextColor(BLACK);
  //Set font text size to 1
  tft.setTextSize(1);
  //Draw black rectangle for plot
  tft.fillRect(PLOT_X_OFFSET,PLOT_Y_OFFSET,SCREEN_WIDTH-PLOT_X_OFFSET,
                SCREEN_HEIGHT-PLOT_Y_OFFSET, BLACK);

  //Draw middle grid lines
  tft.drawLine(PLOT_X_OFFSET,(PLOT_HEIGHT/2)+PLOT_Y_OFFSET,
SCREEN_WIDTH,
                (PLOT_HEIGHT/2)+PLOT_Y_OFFSET, GRAY);
  tft.drawLine(PLOT_X_OFFSET,(PLOT_HEIGHT/2)+PLOT_Y_OFFSET+1,
SCREEN_WIDTH,
                (PLOT_HEIGHT/2)+PLOT_Y_OFFSET+1, GRAY);
  //The offset here of -7 determines how high the text is above the
lines
  tft.setFont(&FreeSerif9pt7b);
  tft.setCursor(2, (PLOT_HEIGHT/2)+PLOT_Y_OFFSET);
  tft.print(MAX_IRRADIANCE/2);

  //Draw legend text
  tft.setCursor(2, (PLOT_HEIGHT/4)+PLOT_Y_OFFSET);
  tft.print((int)MAX_IRRADIANCE*.75);
  tft.setCursor(19, 218);
  tft.print("0");
  tft.setCursor(0, 69);
  tft.print("1");
  tft.setCursor(5, 69);
  tft.print("1");
  tft.setCursor(11, 69);
```

```
  tft.print("5");
  tft.setCursor(20, 69);
  tft.print("0");
  tft.setFont();
  tft.setCursor(0,212);
  tft.print("IRR");
  tft.setFont(&FreeSerif9pt7b);

  //Draw top grid lines
  tft.drawLine(PLOT_X_OFFSET,(PLOT_HEIGHT/4)+PLOT_Y_OFFSET,
SCREEN_WIDTH,
                (PLOT_HEIGHT/4)+PLOT_Y_OFFSET, GRAY);
  tft.drawLine(PLOT_X_OFFSET,(PLOT_HEIGHT/4)+PLOT_Y_OFFSET+1,
SCREEN_WIDTH,
                (PLOT_HEIGHT/4)+PLOT_Y_OFFSET+1, GRAY);


  //Draw bottom grid lines
  tft.drawLine(PLOT_X_OFFSET,(PLOT_HEIGHT*.75)+PLOT_Y_OFFSET,
SCREEN_WIDTH,
                (PLOT_HEIGHT*.75)+PLOT_Y_OFFSET, GRAY);
  tft.drawLine(PLOT_X_OFFSET,(PLOT_HEIGHT*.75)+PLOT_Y_OFFSET+1,
SCREEN_WIDTH,
                (PLOT_HEIGHT*.75)+PLOT_Y_OFFSET+1, GRAY);
  //Draw legend text
  tft.setCursor(2, (PLOT_HEIGHT*.75)+PLOT_Y_OFFSET);
  tft.print(MAX_IRRADIANCE/4);

  tft.setCursor(130, 235);
  tft.print(((double)(fiveSecondTask/1000)*31)/60);
  tft.print(" minutes");
  tft.drawLine(35, 230, 120, 230, BLACK);
  tft.drawLine(230, 230, 310, 230, BLACK);
  tft.fillTriangle(25, 230, 35, 223, 35, 237, BLACK);
  tft.fillTriangle(320, 230, 310, 223, 310, 237, BLACK);
}
```

7.2 Lab Manual

Stand-Alone PV System Lab

Experiment 1: Basics of PV - Simulation

The objective of this first experiment is to simulate the basic characteristics of photovoltaic (PV) cells. We will test the effects of temperature, irradiance, and load resistance on the solar panels, and the resultant change in the I-V characteristics. All tests for this experiment will use the provided Simulink model called “Resistive Load Only.” You must open Simulink prior to opening the file in order for the file to work properly.

1. Changing temperature
    a. Set the resistor to 10Ω, and irradiance to 1000 W/m2
    b. Run the simulation several times with varying input temperature (into the PV Array).
        i. Hint: use a range between 0 - 200 °C
    c. Record the Voltage, Current, and Power output, and take note of any trends.

DELIVERABLE 1: Document your observations about the effects of both changing temperature.

- What patterns did you notice?
- What do you notice as Temperature increases/decreases?
- Does this make sense, why or why not?
- What temperature provides max power?

2. Changing irradiance
    a. Set the resistor to 10Ω, and temperature to 25 °C
    b. Run the simulation several times with varying input irradiance (into the PV Array).
        i. Hint: use a range between 0 - 1000 W/m2
    c. Record the Voltage, Current, and Power output, and take note of any trends.

DELIVERABLE 2: Document your observations about the effects of both changing irradiance.

- What patterns did you notice?
- What do you notice as irradiance increases/decreases?
- Does this make sense, why or why not?
- What irradiance provides max power?

3. Changing resistance
    a. Set the irradiance to 1000 W/m2 and temperature to 25 °C
    b. Run the simulation several times with varying resistance
        i. Start with these values: 1, 2, 3, 4, 5, 7.5, 10, 15, 20, 50, 100, 200
    c. Record the Voltage, Current, and Power output
    d. Plot both Voltage vs. Current and Voltage vs. Power (Voltage on x-axis)
        i. Use plots to estimate the resistance value that draws the maximum power

e. Based on your value from the last part, run the simulation a couple more times and try to get close to the maximum power point.

- DELIVERABLE 3: Include both plots created in your report.
    - What happened when you changed the load resistance?
    - What resistance value did you find that drew the most power from the solar panels?

4. Effect of irradiance and temperature on the maximum power point
    a. Find the MPP for each of the values given in the tables below.
    b. For each step you do not need to plot as many points as part 3 – 5 to 7 for each should suffice.

Irradiance (W/m^2)	Temperature (°C)	Resistance for MPP (Ω)		Irradiance (W/m^2)	Temperature (°C)	Resistance for MPP (Ω)
200	25			1000	0	
400	25			1000	10	
600	25			1000	20	
800	25			1000	30	
1000	25			1000	40	

DELIVERABLE 4: Include both tables from part 4. You do not need to include each individual graph created to find the MPP for each of the values. Describe any trends you noticed as you filled out the tables.

Experiment 2: Solar Panel with MPPT – Simulation
As seen in the first experiment, solar panels are nonlinear devices. The curve obtained in Experiment 1 from varying the load resistance is called the I-V characteristics. As we have already seen, these characteristics vary with changes in temperature and irradiance, each with its corresponding maximum power point. When using solar panels in real life, however, the temperature, irradiance, and load can all be changing – sometimes all at the same time! This makes always drawing the maximum power from a solar panel very difficult, because as the variables around it all change, so does its maximum power point. To remedy this issue, an impedance matching circuit is used to remain at the maximum power point. The device that does this, is called Maximum Power Point Tracking (MPPT). It utilizes a DC conversion circuit, along with a logic circuit to alter the duty ratio continuously. In our case, we will use a Boost Converter. All tests for this experiment will use the provided Simulink model called "MPPT_MANUAL" and the "Duty Cycle" page in the "Simulink SpreadSheet".

1. Duty Cycle (D)

a. You should see a similar model to Experiment 1, except there is an additional circuit in the middle
b. To begin, make sure that the Irradiance is set to 1000 W/m2 and temperature to 25 °C
c. Find the MPP to the nearest tenth.
    i. Hint: This is very similar to your work in part 3 of Experiment 1.
    ii. This is your equivalent resistance
d. Set the load to 50Ω, and using the formulas provided above, solve for the duty cycle required to reach the maximum power point.
e. Run the simulation several times, with a varying duty cycle, and record the Voltage, Current, and Power output. You should also calculate the effective resistance using Ohm's Law.
    i. Be sure to use your value from part d as one of the duty cycles!
    ii. Note: the input to the switch is 1-D, not D

DELIVERABLE 5: With the irradiance and temperature unchanged, what is the duty cycle needed to reach maximum power? Hint: you want to keep the equivalent resistance constant
DELIVERABLE 6: Document your observations about the effects of changing the duty cycle of the converter. What patterns did you notice?

HARDWARE BACKGROUND

The general layout of the equipment is as follows: the PV array takes inputs of irradiance and temperature, which feed into the MPPT, which adjusts voltage and current inputs to find the maximum power output of the panels. From there, power is outputted from the MPPT to an inverter and DC output. The inverter converts the DC power into single phase. The single phase has its own output and also enters a VFD that converts the single phase to three phase. When the load requires more power than what solar can currently provide, supplemental power will be taken from the batteries. The relays are used to allow the solar to bypass the MPPT and go directly to the DC Output terminals at the flick of a switch. The Arduino powers the graphical display showing IRR and Temp. Lastly the switch turns the measurement devices on or off.

Main Circuit: All the circuitry shown above, is contained in a metal enclosure that has three outputs – DC, single phase AC, and three phase AC. We will use each of these in the following three experiments.

Solar Panels: Two Kyocera KD135GX-LPU panels with maximum power rating of 135W per panel.

Batteries: Two 12V batteries in series for 24V output.

Inverter: Peak efficiency of the inverter is 85%. Converts DC voltage to 120VAC to power load.

MPPT (maximum power point tracker): A DC to DC converter that optimizes the match between the solar panels and battery of the PV array. Its functionality is to convert higher DC output voltage from solar panels to lower voltage output needed to charge batteries. The maximum power point tracking comes from the MPPT reading the output of the solar panels in order to compare it to the battery voltage. From these readings, it figures out what the optimum power the panels can output to charge the battery.

Load: A resistor box with 8 different resistors (1, 2, 5, 16, 25, 100, 150, and 200 ohm)

Note:
This experiment is extremely weather dependent. The available irradiance will determine how many measurements you will be able to take. If you attempt to increase the load, or isolate PV and battery to power the load, it may pull the inverter under its required power. You will know that this is happening because the inverter will beep before it cuts power to the load. If this happens, power all of the equipment down and decrease the load.Turn the inverter off and back on again. If this issue persists with only one light bulb powered by both PV and battery, you will not be able to complete this portion of the lab.

Safety:
Refer to the circuit diagram provided for the graphical connections and locations of each switch for the equipment. For the safety of everyone using the hardware, it is very important that the equipment is hooked up correctly. Verify with the TA that everything is correctly connected before moving on to the next steps. Do not manipulate the circuit while it is powered on.

Experiment 3: Solar to Resistive Load – Hardware
In the first 2 experiments, you were working only with simulated models of the solar panels and MPPT. In this experiment you will begin working with these devices in the real world. The objective of this experiment is to see how the hardware compares to the ideal simulated models. In the first part you will test the characteristics of our solar panels by connecting them directly to the load.

1. PV directly to load
    a. The system should be set up correctly for you to use, the only switch you need to touch is the breaker on the far left side
        i. NOTE: DO NOT touch any of the other switches on the hardware
    b. Flip the breaker and measure the DC output voltage using a voltmeter, this is the open circuit voltage
    c. This voltage will vary throughout the day as it is dependent on temperature and irradiance as we learned in the previous experiments
    d. Look at the irradiance and temperature display to observe the current irradiance and temperature.
    e. Open the “Resistive_Load_Only” file in Simulink and input the current temperature and irradiance

        i. NOTE: Remember you have to open Simulink first in Matlab before opening the file
    f. Double click on the solar panel symbol and change the “Series-Connected modules per string” to 2 instead of 1, this is because we have 2 solar panels outside in the courtyard
    g. Find the correct resistance value for the MPP under these conditions in simulink (Very similar to Experiment 1 part 3 “Changing Resistance”)
    h. This value is the MPP for the current condition outside
    i. Using the resistance box load and the DC output, connect the PV to several different values of resistance, half above and half below the maximum power point
    j. Be sure to flip the breaker when switching resistances.
        i. Note: include the simulated maximum power point as one of your values!
        ii. This resistance value may not be the same for the hardware
    k. Record voltage, current, and power for each different load value

DELIVERABLE 7: Create 2 plots like you did in experiment 1, Voltage vs. Current and Voltage vs. Power. Was the simulated MPP the same as you found with the hardware? What do you think accounts for any discrepancies?

Experiment 4:

1. Buck-Boost Converter
    a. Connect the DC output to the Buck-Boost Converter, and from the converter to the resistance box. Ensure that the resistance is significantly higher than the maximum power point, and REMAINS UNCHANGED for all duty ratios tested.
    b. Record voltage, current, and power output from solar panels while varying the duty ratio.
    c. For each different duty ratio, calculate the equivalent resistance using Ohm’s Law and the solar output. Graph Equivalent Resistance vs. Power.
2. Repeat all these steps but use a resistance value that is smaller than the maximum power point.

DELIVERABLE 7: Include both plots created in your report. What duty ratio provided the maximum power for each value of resistance tested?
DELIVERABLE 8: How did the power output with the converter compare to when the solar output was connected directly to the resistors? Can you calculate the efficiency of the Buck-Boost Converter?

Experiment 5: 3-Phase Induction Motor
This part of the lab will show that a three phase AC load can be operated using a series of DC to AC and AC to AC conversions. Using the built in modules and other tools, the power and speed will be measured and calculations regarding the efficiency and power output will be made and observed. This lab will use the KBMA Induction motor drive and the ½ HP Marathon induction motor.

1. Setup the motor by plugging in the appropriate pins.

2. Using the Solar in display panel write down the voltage at the solar terminals. Also record the IRR using the appropriate display module. Using these two measurements and the Simulink information that was gathered in part 1, what is the max power available.
    a. It should be noted that if we just had a PV system without batteries our system would be limited to the amount of power we are getting from the solar panel but since we have batteries and a charge controller the MPPT module will automatically pull power from the batteries to compensate for any power that is not being produced by the solar panels that we need. That leads us to one of the core ideas for this lab. The idea is that we can run multiple different loads from a PV system in conjunction with a battery and charge controller if the load is properly sized. (We need to be careful to not fully drain the battery).
    b. Measured V:
    c. Measured IRR:
    d. Calculated P_In:
3. Calculate the amount of solar power that should be output at the end of Inverter (Note: The inverter has an efficiency of 85%)

Estimated solar power available based on max power calculation at the output of the Inverter:

4. Turn on both the power supply and the KBMA drive. There will be some power being drawn when the motor drive is on but the motor is not running. Record this power using the load display module.

KBMA Drive Power Loss:

***MAKE SURE THAT THERE IS NOTHING THAT COULD GET CAUGHT IN THE TURNING OF THE MOTOR!!**

***SAFETY IS THE MOST IMPORTANT WHEN OPERATING THE MOTOR!!**

5. In order to control the speed of an Induction motor in general a V/F control is used where V = voltage and F = frequency. The drive outputs a percentage of the rated Hz of the setting that we have it at. The drive is set at 60 HZ this means if I have the main speed Potentiometer set at 10 it will output 10% of the rated nameplate speed and frequency.
    a. Calculate the speed of the motor at two different points one from 0-50 and one from 51-100

Calculated Speed
0-50:
51-100:

6. With the KBMA drive switched on turn the drive spots of your choice one from 0 – 50 and one measurement from 51-100. For each of these two drive speeds measure the speed of the motor using a tachometer. Also record the power being drawn by the motor

using the load display module and the power injected by the solar panel using the Solar In display module.

Measured Speed
0-50:
51-100:

Does the measured speed match the calculated speed in Step 12 if not give reasons as to why?
Measured Power

Solar In
0-50:
51-100:

Solar Load
0-50:
51-100:

Subtract the load power from the solar power.

Power Difference
0-50:
51-100:

How much power is the motor pulling from the battery or is the solar panel injecting into the battery?

What is the max speed that the motor can be operated at so that it is only drawing power from the solar panel?

7. As you can probably guess running a motor this way is not very efficient. Calculate the Efficiency of running the motor using this system.

$Eff\ \% = (P_{In} - P_{Loss})/P_{In} * 100$

Where P_{Loss} is the amount of power lost in the inverter as well as the KBMA drive.

Eff %:

References:

Below are the references that included the previous groups worked on this project. These also includes all Arduino code, lab manual, and documentations.

- Anca D. Hensen, Poul Sorensen, Lars H. Hansen, Henrik Binner, “Models for a Stand-Alone PV System”, Riso National laboratory, Roskilde, pp. 1-78, 2000.
- G.Bhucaneswari, R. Annamalai, “Development of a Solar Cell Model In Matlab For PV Based Generation System”, India Conference (INDICON), 2011 Annual IEEE, pp. 1-5, 2011.
- "IEEE Guide for Array and Battery Sizing in Stand-Alone Photovoltaic (PV) Systems," in IEEE Std 1562-2007 , pp.i-22, May 12 2008.
- "IEEE Guide for the Selection of Monitoring for Circuit Breakers," in IEEE Std C37.10.1-2000, 2001.
- Merrifield, T., Chartouni, N., Korhonen, E., Ukpan, S. and Pachl, J. (2017). Senior Design - Renewable Energy Lab Project Team - Iowa State University. [online] Available at: http://dec1706.sd.ece.iastate.edu/
- Mishra, Bibek, and Bibhu Prasanna Kar. "Matlab Based Modeling of Photovoltaic Array Characteristic." Thesis. National Institute of Technology, Rourkela, 2012.
- Ting-Chung Yu, Yu-Cheng Lin, “A Study on Maximum Power Point Tracking Algorithms for Photovoltaic Systems”, Lunghwa University of Science and Technology, pp. 1-10, 2010.
- Tyson DenHerder, “Design and Simulation of Photovoltaic Super System Using Simulink” California Polytechnic State University, San Luis Obipo, 2006.
- Senior Design Team sddec18-19. “Design and Implementation of a small scale stand alone Hybrid Solar PV and Wind Energy Generation system”. Iowa State University . EE 491 . http://sddec18-19.sd.ece.iastate.edu/ . Dec 2018.