Electrical Engineering Department Project, 2018 Spring Semester

Bill Chen, Alex Yao, Angela Wang, Callie Singer, Bennet Brown

March 27, 2018

1 Design

1.1 System Overview:

Columbia secondary school has requested a solar panel system that will accommodate the needs of their school garden. Our solar panel system will be located in front of Pupin Hall and consists of 12 solar panels with dimensions $39.7 \times 26.7 \times 1.4$ inches. Students from the secondary school will come daily to collect their Windy Nation 12V charged battery. They will also bring back their uncharged battery from the previous day to be recharged.

Our design has been greatly simplified to accomodate to the available materials and simplifying the design process. However, an overview of our calculations regarding the the materials can be found in the calculations section. Our goal is to be able to power the Garden AC System for 2 hours at 120V and 20 amps.

1.2 The Solar Panel

We narrowed our panels down to two candidates:

Table 1: Renogy 100 Watt 12 Volt Polycrystalline Solar Panel

Maximum Power	100W
Optimum Operating Voltage	17.8 V
Optimum Operating Current	5.62 A
Weight: 16.5 lbs	Dimensions: 39.7 x 26.7 x 1.4 inches
Cost	$105

Table 2: 100 Watt Flexible Solar Panel with SunPower Solar Cells from Windy Nation

Maximum Power	100W
Optimum Operating Voltage	17.8V
Maximum Power Point	5.62A
Weight	4.1 lbs (1.85 kg)
Module Dimension ($L \times W \times H$)	$(41.7 \times 21.3 \times 0.1)$
Cost	$179.99

We can see from these two tables that the performance ratings of the two are similar but the monocrystalline 100 Watt Solar Panel is flexible, thinner and lighter. It also costs a lot more (179 vs 105). Due to our needs for a long term power solution that will probably be mounted permanently. We chose the polycrystalline solar panel to save on costs.

1.3 Other Materials and Setup

As for other materials, we choose the following for simplicity and compatibility after careful research. Batteries are chosen to 1) have deep cycling compatibility, 2) be able to hold the necessary 4800 WHr energy needed to power the garden AC system.

The batteries will be connected in parallel. During charging, it will be connected to the regulator which utilizes MPPT algorithm to maxmize the power output of the solar cell array. The regulator will be connected to the solar cells. The solar cells will be connected in parallel as there will be very little distance between the regulator and the cells: loss due to high amperage energy transfer is minimized. The system is also more immune to the shading problem - one panel shaded will not affect the whole string, a problem worth

Table 1: Other parts

Batteries	12V 100 Amp-Hour Deep Cycle AGM Sealed Lead Acid Battery $\times 6$	$\$185 \times 6 \times 2 = \$2220$
Inverter	VertaMax 3000 Watt 12V Pure Sine Wave Power Inverter DC to AC	$416
Regulator	TrakMax 30L LCD MPPT 30A Solar Charge Controller Regulator	$200

considering as our system will be surrounded by buildings. 6 batteries will be able to store a total of 7200 WHr of energy. Even though technically only 4 batteries are needed for our need of 4800 WHr, the extra two batteries should be able to hold excess charge in times when the sun has more available energy and can be used as backup power for rainly and the winte days.

When discharging, the batteries will be hooked up to the power inverter which boosts the 12V battery voltage to 120V, allowing the garden AC system to draw the power.

2 Model

2.1 Assumptions

We assume the following statements to be true: solar panels are charging the battery throughout the whole duration of sunlight from sunrise to sunset; solar panels are placed at a constant angle horizontal to the ground throughout the day; students from Columbia Secondary School switch the set of batteries once per day; and lastly, temperature and weather conditions do not have a significant effect on the amount of solar energy obtainable.

2.2 Solar Energy

2.2.1 Basic Parameters

$$\begin{aligned} \text{astronomical unit } AU &= 1.496 \times 10^{11} m \\ \text{solar power } P_0 &= 3.8 \times 10^{26} w \\ \text{atmosphere shielding } s &= 0.75 \\ latitude &= 40.8107 \\ longitude &= 73.9561 \end{aligned}$$

2.2.2 Angle of the solar panel

The distance between the sun and earth at a given day, is given by

$$r = 1 - 0.01672 \times \cos(2 \times \frac{\pi}{365.256363} \times (day - 4)))$$

The solar declination angle in degrees is

$$\phi = 23.45 \times (\pi \sin((284 + day) \times 2 \times \frac{\pi}{365.256363})$$

The max solar zenith angle in radian is

$$\psi_{max} = latitude \times \frac{\pi}{180} \times \phi$$

The hour angle and altitude angle are, in degrees

$$\begin{aligned} &\theta = (time - 12) \times 15 \\ &\alpha = \cos\theta \times \cos\phi \times \cos(atitude) + \sin\phi \times \sin(latitude) \end{aligned}$$

To optimize the efficiency of PV cells, the solar panel should be positioned at an angle α relative the the earth surface.

2.2.3 Solar energy per area

The ambient irradiation is given by $G(w/m^2)$

$$G = \frac{P_0}{4\pi r^2}$$

The availiable solar energy per area $P(J/m^2)$ over a period of time is given by

$$P = \int_{t_1}^{t_2} G \times \sin\alpha \times s\mathrm{d}t$$

2.3 Area of the Solar Panel

We calculated the amount of Solar Energy expected per square meter in New York with Python. We first calculated the distance of Earth from the Sun as a function. We then took into account of New York's location and Earth's declination angle. Python source code of our calculation is available on github [1] The permittivity of the atmosphere to solar radiation was adjusted to be 0.75 to fit the lowerbound of our model to available official data [2]. A quick reference to other resources finds our assumption rational [3]. We assumed that we are able to obtain around 6 hours of perfect sunlight everyday as our conservative estimate. The advantage of using a model like ours is that we can not only predict how many solar panels we would need to charge our batteries to 4800 watt-hours, we can effectively predict how much solar energy we can expect from the sun each and everyday of the year.

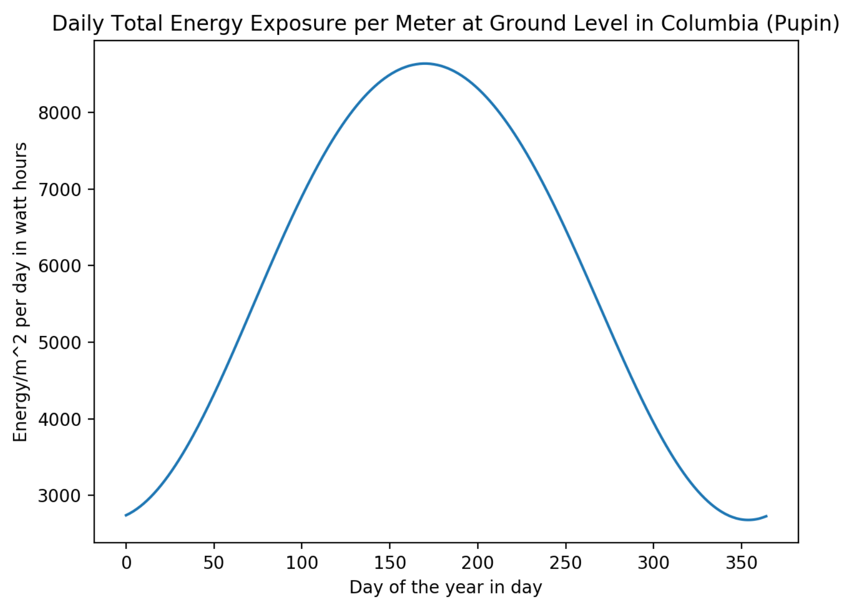

Figure 1: Daily solar energy over a year

Taking the lowest of the year: 2700 Watt-Hours/m^2, which roughly occurs at the 350th day mark, and assuming the efficiency coefficient of the solar panels to be 0.2 and taking into account each solar panel's area: $0.687m^2$, we can roughly approximate how many solar panels we would need.

Each panel output during the day of lowest sun exposure:

$$0.687m^2 \times 0.2 \times 2700WHr/m^2 = 370.98WHr$$

$$4800WHr/370.98WHr = 12.938Panels$$

Since our estimate is extremely conservative (we did not take into account of the left over charge we may have from the previous day or any other time of the year and we have vastly underestimated daylight time), we should be able to round down our estimation and 12 solar panels should be sufficient to run our garden.

A few select points of our data is calculated here and put onto a table:

Table 2: Select Data Points

Day	Sunrise	Sunset	Total Energy (WHr/m^2)	Functional Hours
1	7:33	16:33	2741.96	1.88
50	6:17	17:17	4274.31	2.94
100	5:25	18:25	6856.48	4.71
150	4:20	19:20	8477.49	5.82
200	4:16	19:16	8334.43	5.73
250	5:19	18:19	6519.67	4.48
300	6:11	17:11	3993.43	2.74
350	7:32	16:32	2691.41	1.85

2.4 PV System Design

We have so far calculated the number of PV cells required (12) to provide enough energy for Columbia Secondary School. Now we can use IV-curve modeling to optimize the efficiency of our PV system. We will provide a circuit diagram to demonstrate how the system should be installed.

2.4.1 Power Maximization

From IV charastics we know that the PV array should have the same number of cells per row and per column. We denote the number of cells conected in series by n_s and the number of cells conected in parallel by n_p. They should satisfy $n_s \times n_p = 12$. We model the diode current by Shockly equation,

$$I = I_0(e^{\frac{qV}{kT}} - 1)$$

where the thermal voltage is approximated to be $\frac{kT}{q} = 25.85mV$. We calculated the maximum power $P = IV$ for each permutation of n_s and n_p. Thus each row should have 4 cells.

Table 3: Maximum Output for each layout

n_s	P (w)
1	13.17
2	22.46
3	91.3643
4	1096.37
6	730.914

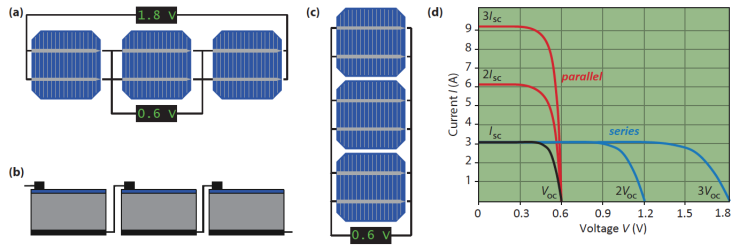

Figure 2: Illustrating (a) a series connection of three solar cells and (b) realisation of such a series connection. (c) Illustrating a parallel connection of three solar cells. (d) I-V curves of solar cells connected in series and parallel.

2.4.2 Circuit Diagram

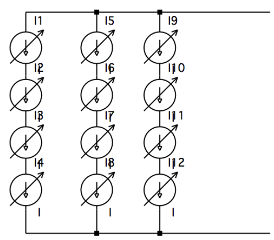

Figure 3: PV modules circuit design

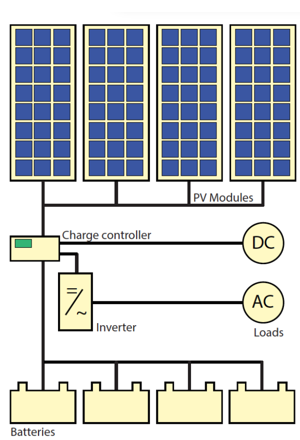

Figure 4: PV system design

2.5 Cost

Total cost of Solar Panels:

$$\$105 \times 12 = \$1260$$

The total cost including all required components is:

$$\$1260 + \$416 + \$200 + \$2220 = \$4096$$

The total cost including all required components without two sets of batteries is:

$$\$1260 + \$416 + \$200 + \$1110 = \$2986$$

Along with installation fees and such, we are estimating a budget proposal of around $4500 for the full system and $3500 for the system with one set of batteries.

3 Conclusion

The proposed solar panel solution comprises of 12 Renogy 100 Watt Polycrystalline solar panels with two sets of batteries. We arrived at this number of panels by computing the energy obtained from the sun at the Pupin building given an elliptical, rotating model of the Earth's movement around the sun. While more expensive, we believe our proposal will provide the greatest convenience for Columbia Secondary School students to retrieve and switch batteries and allow the panels to function for the entirety of sunrise to sunset. The final cost is $4096. We hope our proposal will aid in supplementing some of Columbia and Columbia Secondary School's energy demands with clean solar energy.

4 References

[1] https://github.com/billchen99/AOE_SolarPanel/blob/master/solar_alex1.py
[2] *Solar Energy Resource Throughout New York*, http://www.asrc.cestm.albany.edu/perez/publications/
[3] *Solar Energy To Earth http://energyeducation.ca/encyclopedia/Solar_energy_to_the_Earth*

5 Appendix

5.1 Python code

```
import math
import matplotlib.pyplot as plt
import scipy.integrate as integrate

AU = 1.496*10**11 # m
P0 = 3.8*10**26 # W
SHIELD = 0.75
latitude_deg = 40.8107
longitude_deg = 73.9561

V = 120 #VAC
I = 20 #amps
T = 2 #hrs
E = V*I*T #w-hr

def get_distance(day):
    return (1-0.01672*math.cos(2*math.pi/365.256363*(day-4))) # AU

def get_intensity(day):
    r = get_distance(day)*AU
    return P0/(4*math.pi*r**2)

def get_declination_angle(day):
    return 23.45*(math.pi/180)*math.sin((284+day)*2*math.pi/365.256363) # radians

def get_max_zenith_angle(day):
    return latitude_deg*math.pi/180 - get_declination_angle(day) # radians

def get_hour_angle(time):
    return (time-12)*15*math.pi/180

def get_altitude_angle(day,time):
    # elevation angle in deg
    hour_angle = get_hour_angle(time)
    latitude = latitude_deg*math.pi/180
    declination_angle = get_declination_angle(day)
    x = math.cos(hour_angle)*math.cos(declination_angle)*\
    math.cos(latitude)+math.sin(declination_angle)*math.sin(latitude)
    return math.asin(x)*180/math.pi

def get_effective_intensity(day,time):
    elevation = get_altitude_angle(day,time)*math.pi/180
    return get_intensity(day)*SHIELD*math.sin(elevation)

def get_surface_energy_horizontal(day,time1,time2):
    # energy per square meter when panel is flat on the ground
    result = integrate.quad(lambda t: get_effective_intensity(day,t), time1, time2)
    return result[0]

def get_surface_energy_tracking(day,duration):
```

```
    # energy per square meter when panel is elevated
    result = integrate.quad(lambda t: get_effective_intensity(day,t), \
    12-duration/2, 12+duration/2)
    return result[0]
    # panel's angle of elevation = get_altitude_angle(day,time)
```

5.2 Mathematica code

```
    In[91]:= Plot[5.62/6 - 10^-24.9*(Exp[v*38.6/2] - 1), {v, 2.5, 3.},
 Filling -> Axis]

In[104]:= Plot[5.62 - 6*10^-24.9*(Exp[v/6*38.6/2] - 1), {v, 2.5*6, 18}]

In[107]:= Plot[
 5.62*4 - 4*6*10^-24.9*(Exp[v/3/6*38.6/2] - 1), {v, 2.5*18, 3*18}]

In[110]:= Plot[{5.62*3 - 3*6*10^-24.9*(Exp[v/4/6*38.6/2] - 1)}, {v,
  60, 4*18}]

In[111]:= MaxValue[(5.62*3 - 3*6*10^-24.9*(Exp[v/4/6*38.6/2] - 1))*
  v, v]

Out[111]= 1096.37

In[112]:= MaxValue[5.62*4 - 4*6*10^-24.9*(Exp[v/3/6*38.6/2] - 1)*v, v]

Out[112]= 22.48

In[120]:= MaxValue[(5.62 - 6*10^-24.9*(Exp[v/12/6*38.6/2] - 1))*v, v]

Out[120]= 1096.37

In[121]:= MaxValue[(2*5.62 - 2*6*10^-24.9*(Exp[v/4/6*38.6/2] - 1))*
  v, v]

Out[121]= 730.914
```