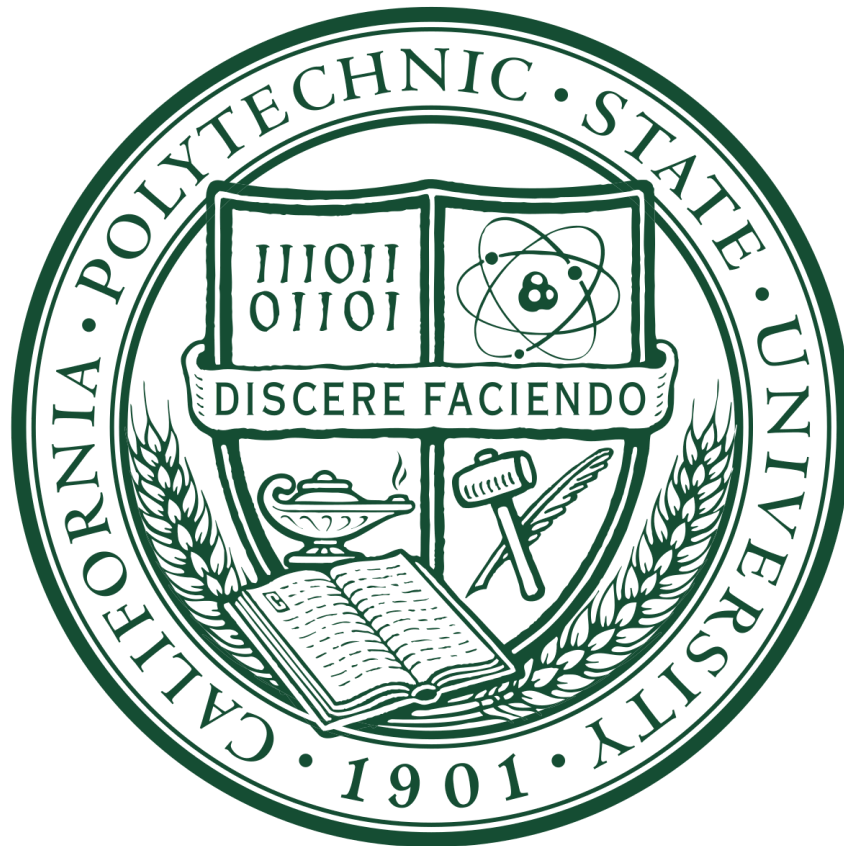


# Smart Solar Charge Controller

Aria Pegah

Miclaine Emtman



Senior Project

Electrical Engineering Department

California Polytechnic State University

San Luis Obispo, CA

2020

## **Table of Contents**

List of Figures	ii, iii
List of Tables	iv
Abstract	v
Chapter 1 Introduction	1
Chapter 2 Background	3
Chapter 3 Customer Needs and Specifications	6
Chapter 4 Functional Decomposition	8
Chapter 5 Project Planning	11
Chapter 6 Design and Simulations	13
Chapter 7 Conclusion	31
References	32
Appendix A: MPPT Simulink Schematic with Direct Voltage Feedback	34
Appendix B: MPPT Simulink Schematic with Buck Converter	34
Appendix C: Improved P&O MPPT Code	35
Appendix D: PSO Code	37
Appendix E: Senior Project Analysis	41

## List of Figures

Figure 1.1: IV and PV Curve of Solar Cell	1
Figure 1.2: Solar Powered Trailer	2
Figure 2.1: Basic Perturb and Observe Flow Chart	4
Figure 2.2: Effect of Partial Shading on the IV and PV Curves	4
Figure 2.3: Swarm Intelligence Models	5
Figure 4.1: Level 0 Block Diagram	8
Figure 4.2: Level 1 Block Diagram	9
Figure 5.1: EE 461 Gantt Chart	11
Figure 5.2: EE 462 Gantt Chart	11
Figure 6.1: Simulink Model	13
Figure 6.2: Buck Converter	14
Figure 6.3: Visual Representation of Current Flow in Buck Converter	16
Figure 6.4: Irradiance Switcher	17
Figure 6.5: P&O with Fixed Large Step	18
Figure 6.6: P&O with Fixed Small Step	19
Figure 6.7: P&O with Successive Approximation Flowchart	20
Figure 6.8: Modified P&O Power Results	21
Figure 6.9: Duty Cycle Measurement	22
Figure 6.10 PSO Flowchart:	25
Figure 6.11: PSO Simulink Block	26
Figure 6.12: PSO Output Characteristics	26
Figure 6.13: Duty Cycle and Power Output Delay	28



## **List of Tables**

Table 3.1: Requirements and Specifications	7
Table 4.1: Functional Requirements Level 0	8
Table 4.2: Functional Requirements Level 1	10
Table 5.1: Cost Analysis	12
Table E.1: Cost Analysis for Physical Implementation of Project	42

## **Abstract**

The technology of photovoltaics has been quickly evolving as we move toward a future with more clean energy. Energy conversion efficiency is key to making these systems the viable option amongst other sources of power. This project proposes a way to increase the power yield from a solar system by implementing smarter algorithms in the maximum power point tracking (MPPT) component of the system. MPPT is critical because environmental conditions may vary significantly for any given system due to irradiance and shading. The proposed method for MPPT utilizes a controller that uses swarm intelligence to do the power tracking.. This controller makes use of the Particle Swarm Optimization (PSO) algorithm to find the optimal duty cycle used by the charge controller. Matlab/Simulink was used as a simulation resource to test the various MPPT designs.

## Chapter 1 Introduction

The photovoltaic cells within a solar panel have a strong and elaborate relationship with the environment in which they are used. All photovoltaic cells follow what is known as the current-voltage (IV) curve. This curve defines how much power a solar cell or module will produce for a given load. At some point along this curve, the module is able to produce its maximum power-point (MPP). The operating point along this curve is adjusted by the load provided across the panel. However, this IV curve is dynamic as any set of external conditions such as solar irradiance, shading, and temperature causes the curve to shift. Therefore, with a variable IV curve, the load producing the MPP at one set of conditions, may not provide the maximum power at another set. Hence, the necessity for MPPT arises, so that the module may always operate at its maximum power.

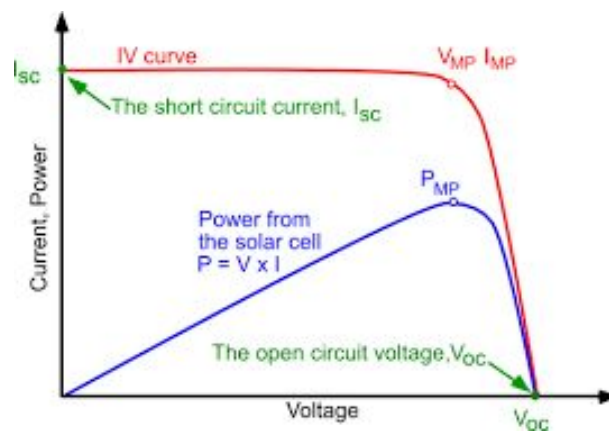


Figure 1.1: IV and PV Curve of Solar Cell [1]

In off-grid solar panel implementations, such as the solar powered trailer shown in Figure 1.2, the use of batteries as energy storage is necessary to provide reliable power at any time of day. However, the introduction of batteries in a solar power system increases the complexity of the system. The issue is in the deviation between the voltage of the array and the voltage of the

battery. While most solar panels operate around 18 volts [2], a typical battery voltage ranges from 11.8-12.8 volts [3]. A smaller voltage produces less power at the same current, therefore, the difference in voltage induces a significant power loss to the system.



Figure 1.2: An example of an off-grid solar powered trailer where MPPT would be used

The power loss due to voltage deviation has been combated with the MPPT solar charge controller. An MPPT controller operates by determining the necessary voltage required to charge the battery at maximum power and then utilizes a DC-DC converter to adjust the voltage to that level. The decrease in voltage in turn increases the current to the battery, resulting in more power and less losses between the solar array and battery storage. Other charge controllers, such as the PWM, also step down voltage, but do not utilize logic to track the MPP of the system.[4]



## Chapter 2 Background

Commercial MPPT charge controllers have utilized various methods to track the maximum power point. The goal of any algorithm implemented for MPPT is to reach the maximum power-point as quickly as possible while being able to adjust to the varying operating conditions present. Generally speaking, the more involved the algorithm, the better it is at finding the MPP. The tradeoff of a more complex controller is the processing power requirements, research, development time, and potentially an increase in cost.

The simplest methods for maximum power-point tracking revolve around using predefined lookup tables that include values such as temperature and corresponding voltage values. Although this keeps necessary processing power at a minimum and is cheap to implement, it lacks a lot of the robustness of higher-end solutions when it comes to adjusting to varying levels of irradiance, especially at the lower end.

The conventional algorithmic methods for MPPT are the Perturb and Observe (P&O) and Incremental Conductance methods. Both are referred to as “hill climb” methods because they begin with an arbitrary operating point and then work towards finding the MPP.

P&O begins by varying the operating voltage by adjusting the duty cycle of the DC-DC converter attached to the output of the solar array. A power measurement at the load is then taken to determine if the duty cycle adjustment caused an increase or decrease in power. If power follows the direction of the duty cycle change, then further changes in that direction are needed, if not then the duty cycle needs to change direction. A visual representation of this process can be seen in Figure 2.1.

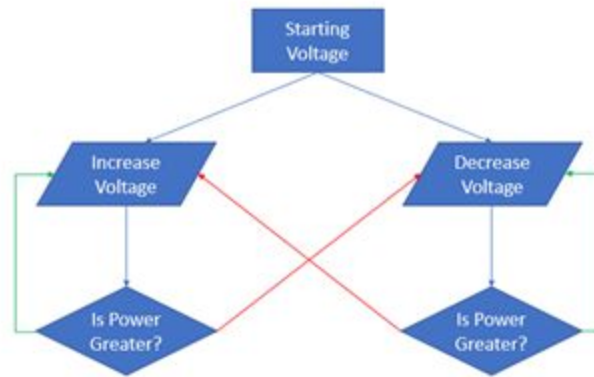


Figure 2.1: Basic Perturb and Observe Flow Chart

P&O's weakness though, is that the exact maximum power-point is unlikely to ever be found. The output of many P&O controllers is a voltage that oscillates around the MPP, wasting available energy. In addition, in environmental conditions where partial shading occurs, the IV curve does not follow typical characteristics as shown in Figure 2.2. In the case of a partially shaded module, MPPT controllers can get stuck tracking the local maxima versus the global maxima, meaning the controller isn't operating at the optimal power. Tracking at the wrong power point may cause a significant reduction in power efficiency. [5-6]

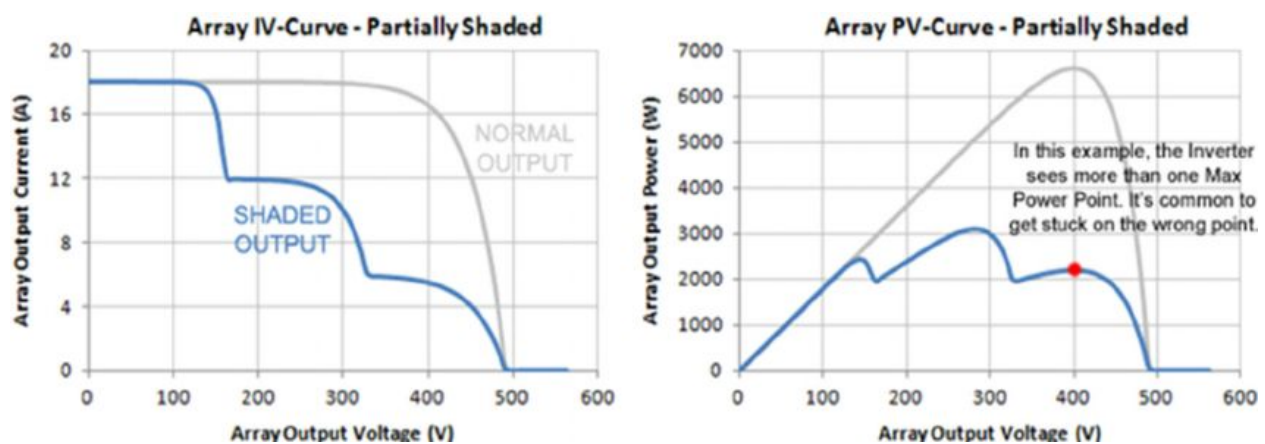


Figure 2.2: Effect of Partial Shading on the IV and PV Curves [7]

A largely unexplored area in MPPT has been the use of machine learning. Swarm intelligence is a form of machine learning that bases its algorithmic principles on mimicking the way animals behave in groups. Typically, an algorithm that utilizes swarm intelligence creates some form of the agent or animal within the group. The individual algorithms that fall under swarm intelligence give a mathematical model for the relationship between the agent and the rest of the swarm. The collective behavior of these systems are often used to solve optimization problems. One example of swarm intelligence is the Ant Colony Optimization algorithm. Each ant is rather unintelligent but through using pheromones as a guidance system, colonies of ants are able to find food. This example is used to demonstrate that these swarms are easily modeled because of their simplicity of their individual agents and intuitive social behavior. Many other models have been created inspired by other natural optimizing phenomena, however this paper will make use of the Particle Swarm Optimization algorithm. Figure 2.3 provides a visual representation of how these algorithms are modeled.

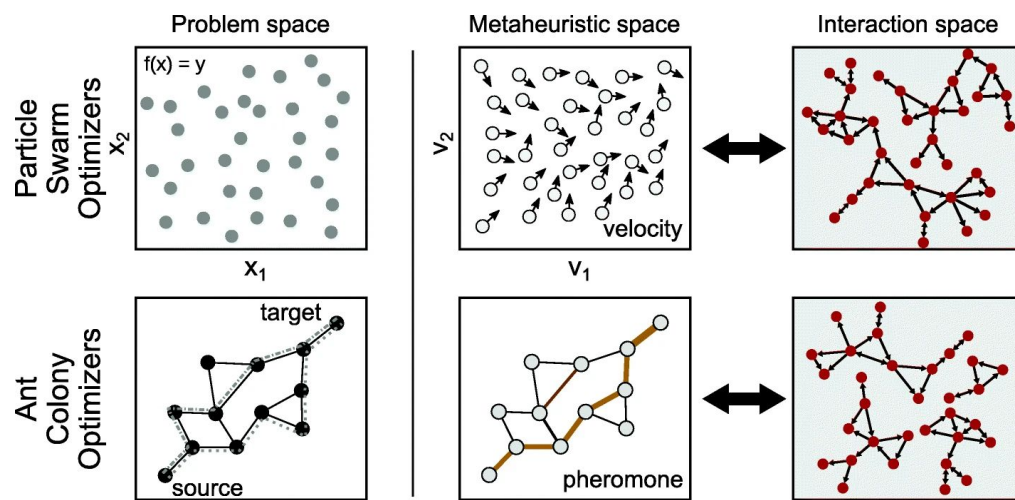


Figure 2.3: Swarm Intelligence Models [8]

### **Chapter 3 Customer Needs and Specifications**

This project's design is geared for a person who camps or lives off-grid and makes use of solar energy as their primary power source. Ultimately, the consumer will require that this device provide them a more efficient alternative to other solar charge controllers.

The solutions provided in this report must provide the customer with reliable tracking of the maximum power point in their respective solar array. This device must be able to perform under a variety of different operating conditions such as varying levels of irradiance, shading, and temperature. As described in Table 3.1, Irradiance may span between 0 and  $1500\text{W/m}^2$  during a given day and the charge controller must be able to operate anywhere within that range. Varying temperatures will also change the respective IV curve and the MPPT must perform within the bounds of extreme conditions. The changes provided by the MPPT controller must happen quickly to react to potentially drastic changes in these environmental factors to mitigate power losses due to time delay. When the MPPT is in steady state, the MPPT must have above a 98% efficiency, meaning that the MPP detected must be within 2% of the true MPP. The customer will also want whatever is implemented in simulation to be able to easily and cheaply be transferred over to a programmable microcontroller that is fast enough to execute the necessary calculations.

Table 3.1: Requirements and Specifications

Marketing Requirements	Engineering Specifications	Justification
1, 2, 3, 4	Follows irradiance changing irradiance levels from $\sim 0\text{W/m}^2$ to $\sim 1500\text{W/m}^2$	The MPPT system must be able to quickly adjust to a large variety of irradiance levels due to the environmental variability it may ensue.
1, 4	Follows changing temperatures from $0^\circ\text{C}$ to $50^\circ\text{C}$	Day to day variations in temperature are expected and this device must be applicable in various climates.
2	Does not deviate more than 2% from the maximum power point in steady-state (no change in irradiance or temperature)	One weakness of certain MPPT algorithms is that they tend to oscillate around the maximum power point in steady-state. The improved algorithms will need to improve upon this so that there is very little deviation.
1, 2	Find the maximum power point within 1 second of a major change in either temperature or irradiance.	A quick response of the controller is necessary to ensure that as much power is being provided to the system in a given time span.
2	Able to take samples and complete computations and adjustments once every 0.5 seconds.	A fast controller is needed to order to adjust the impedance presented to the solar panel in order to find the maximum power point quickly.
2, 5	Implementation on an MSP432	The improved perturb and observe algorithm is anticipated to be able to be implemented on the MSP432, but the neural network may not be able to, so an addition of an offsite computing server and a wireless adapter to the MSP432 could be implemented.
<b>Marketing Requirements</b> <ol style="list-style-type: none"> <li>Usable in a variety of environments</li> <li>Efficient</li> <li>May be implemented with a 12V battery system</li> <li>Works in a variety of array sizes</li> <li>Cost Effective</li> </ol>		

## Chapter 4 Functional Decomposition

Overall, the requirements of this system are to output the optimal voltage and current characteristics at the load given any solar module conditions. The MPPT controller takes the current and voltage of the solar module and uses the algorithms to output maximum power.

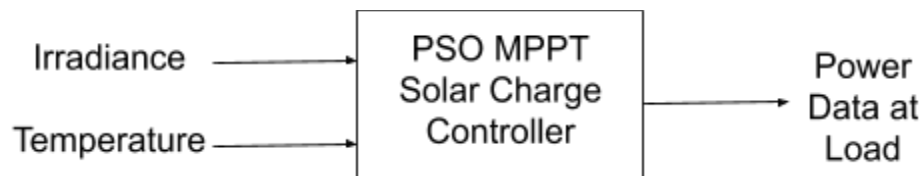


Figure 4.1: Level 0 Block Diagram

Table 4.1: Functional Requirements Level 0

Module	MPPT Solar Charge Controller
Inputs	- Irradiance -Temperature
Outputs	Power Data at Load
Functionality	The user will define the environmental solar module conditions. With the corresponding current and voltage, the system will calculate the optimal power characteristics and adjust the voltage accordingly. This is demonstrated as the power output across the load.

The diagram shown in Figure 4.2, shows the modules within the simulation. The user will input the temperature and irradiance conditions of the solar array which the solar panel simulator is able to turn into the corresponding current and voltage at this environment. This voltage and current is fed into the algorithm. The algorithm will use the PV and load data to find the optimal

load voltage for maximum power and send the corresponding duty cycle to the buck converter. With the duty cycle, the buck converter is able to step down the PV voltage and adjust the current. At the output of the buck converter is the load which receives the adjusted circuit parameters. Across the load a power measurement may be taken and displayed as the output power.

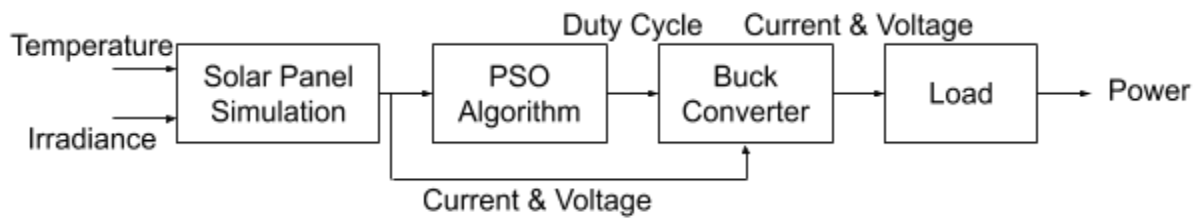


Figure 4.2: Level 1 Block Diagram

Table 4.2: Functional Requirements Level 1

Module	Solar Panel Simulation
Inputs	-Temperature -Irradiance
Outputs	-Current -Voltage
Functionality	This simulator determines the solar modules output voltage and current given the environmental conditions.
Module	Algorithm
Inputs	-Current -Voltage
Outputs	Duty Cycle
Functionality	The algorithm takes the PV current and voltage and determines the voltage required by the load to maximize power. The algorithm outputs the corresponding duty cycle to the calculated voltage.
Module	Buck Converter
Inputs	-Duty Cycle -Current -Voltage
Outputs	-Current -Voltage
Functionality	The buck converter takes the voltage from the solar modules and steps it down according to the duty cycle. Outputting an adjusted voltage and current.
Module	Load
Inputs	-Current -Voltage
Outputs	Power
Functionality	The load receives the current and voltage and allows for a power measurement to take place across it.



## Chapter 5 Project Planning

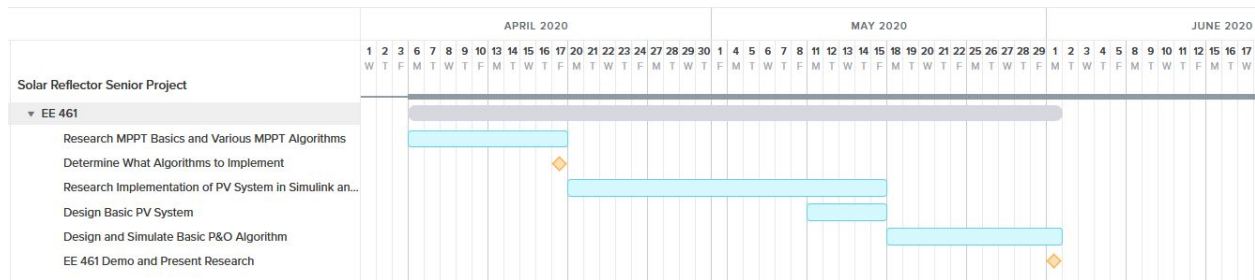


Figure 5.1: EE 461 Gantt Chart

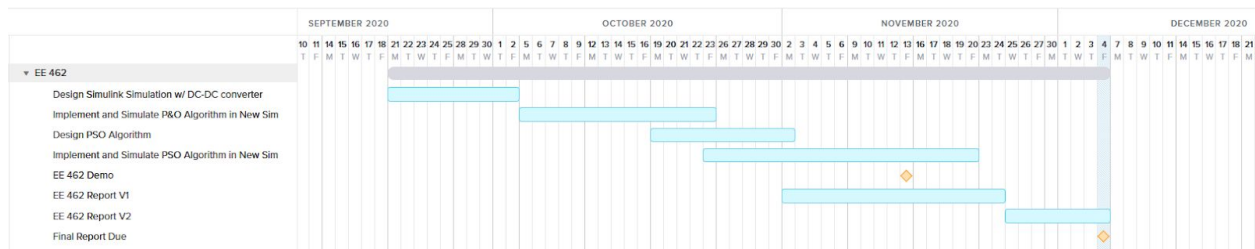


Figure 5.2: EE 462 Gantt Chart

At the end of winter quarter 2020 the COVID-19 pandemic hit with classes switching entirely to online. Due to this, EE 461 was dedicated towards finding a new project idea and research rather than implementation, with only a basic PV array and P&O algorithm implemented at the end of the quarter as shown in the Gantt chart in Figure 5.1.

Summer quarter was not dedicated to the project due to other commitments, so during fall the project was picked up again. Here the majority of the design, implementation, and simulation for each algorithm took place, with the report being compiled and edited here as well as shown in Figure 5.2

Table 5.1: Cost Analysis

<b>Component</b>	<b>Estimated Cost</b>	<b>Justification</b>
Matlab License	\$0	Matlab/Simulink/Simscape were all used in the creation of this simulation.
Labor	\$1,600	Wage is set at \$16/hr for expected 100 hours
<b>Total</b>	<b>\$1,600</b>	

The cost of this project was very simple to break down as seen in Table 5.1. Due to the nature of the COVID-19 pandemic and the limited ability of the project participants to cooperate in any in-person activities related to the project, the project was done entirely within simulation. Matlab was provided at no cost, provided by California Polytechnic State University to the engineers. The labor cost was estimated at \$16/hour, currently close to the minimum wage within San Luis Obispo, CA.

## Chapter 6 Design and Simulations

Since this project's goal is to improve upon existing solar charge controller architecture, quantifying and comparing the performance between the new design and classic MPPT algorithms is necessary. The Perturb and Observe method is the most commonly available MPPT algorithm on the market, so this was used as the reference model. This model was created with Matlab and Simulink. All blocks within the Simulink model excluding the MPPT Matlab block which includes the algorithm under test is either a Simulink/Simscape model or a MathWorks file exchange model [9].

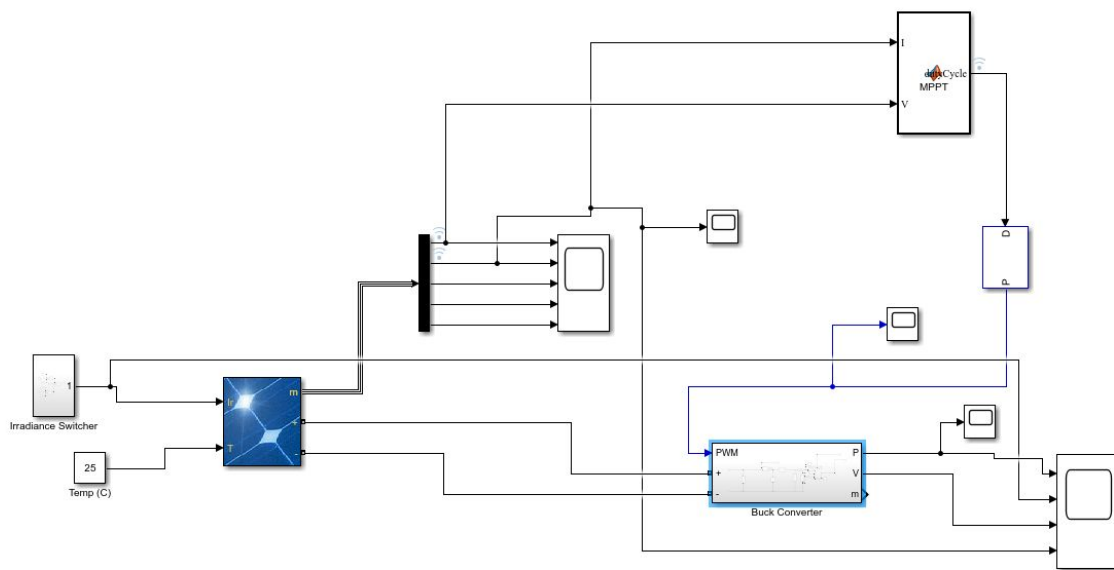


Figure 6.1: Simulink Model

In Figure 6.1 above, the inputs to the PV model are irradiance in  $\text{W/m}^2$  and temperature in Celsius. The PV simulator then feeds into a measurement block that quantifies the output characteristics of the PV array. From these measurements, the current and voltage are outputted into the PO MPPT algorithm block. Within this block, Matlab code, resembling the PO

algorithm, operates on the input variables. The algorithm outputs duty cycle to the buck converter. After making a duty cycle adjustment, the algorithm observes the change in power at the output, and continues to adjust in the same fashion.

### Buck Converter Operation

MPPT relies on adjusting the output voltage of the solar panels to the load. The way the controller accomplishes this is with a DC-DC converter attached to the output of the panel. Since solar panels typically operate at higher voltage than the battery bank requires for charging, the buck or step-down converter is used. The buck converter consists of four major components: a transistor that acts as a switch, a diode, an inductor, and a capacitor.

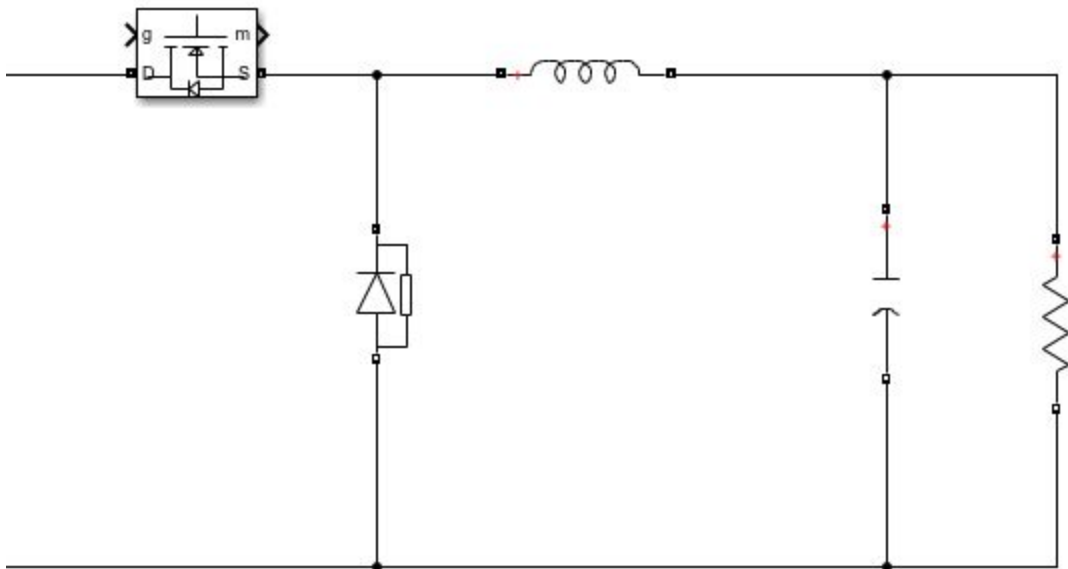


Figure 6.2: Buck Converter [10]

In simple terms, a buck converter is used to step down DC voltage while stepping up current. It does this through the use of a pulse-width modulated (PWM) signal being sent to the transistor into terminal “g” in the transistor in Figure 6.2. When the PWM signal is high, the transistor acts as a closed switch and current flows through the inductor. When the PWM is low, the transistor is moved to the OFF position and the inductor begins discharging. The diode is also forward biased at this point which forces the current direction from the inductor toward the load. Once the inductor has discharged enough, the load voltage begins to fall and the charge stored in the capacitor keeps current flowing until another ON cycle comes. This capacitor combats any voltage ripple at the output. Due to the voltage switching of the system, the load does not receive the PV voltage and instead receives only a fraction of the input voltage. This relationship is shown in equation 6.1, where  $t$  is the ON time of the PWM signal and  $T$  is the period of the signal. This fraction may also be referred to as the duty cycle.

$$V_{LOAD} = V_{IN} * \frac{t}{T} \quad (6.1)$$

Due to the inductor providing current even when the switch turns the voltage supply off, the decrease in voltage may be considered converted into current. A visual representation of this process can be viewed in Figure 6.3

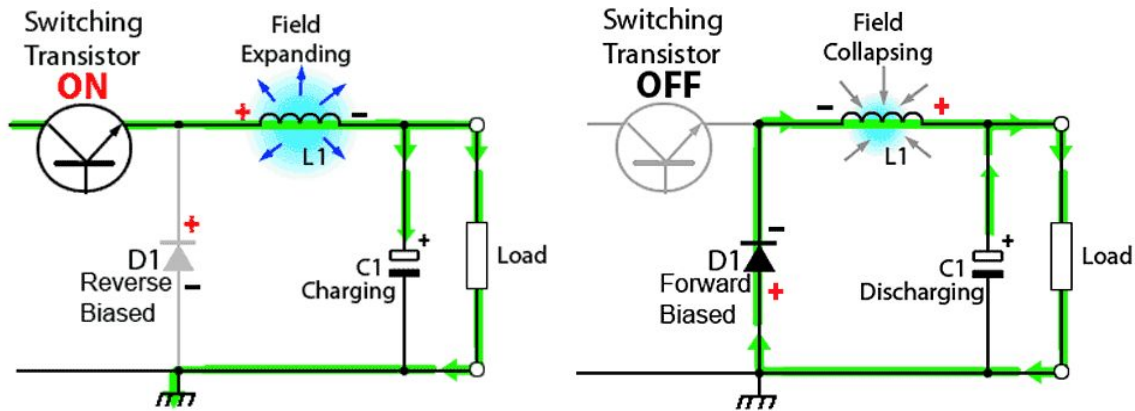


Figure 6.3: Visual Representation of Current Flow in Buck Converter [11]

### Irradiance Switcher

In order to mimic a sudden change in irradiance due to shading, an irradiance switcher, shown in Figure 6.4, is attached to the irradiance input into the PV array. This is accomplished by attaching the irradiance inputs, which are Simulink doubles, to a switch controlled by a step function set to go from “low” to “high” after a set time value. Figure 6.4 shows four separate irradiance values. Readers should keep in mind that the same approach can be taken for temperature.

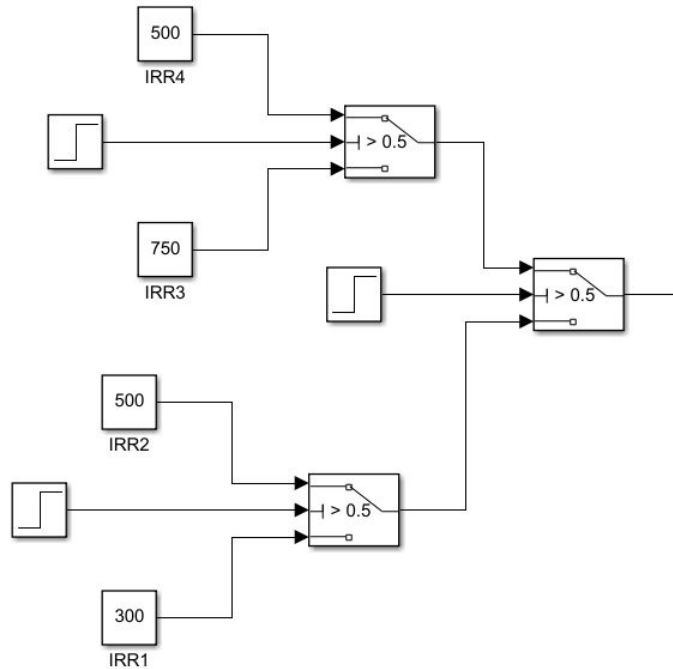


Figure 6.4: Irradiance Switcher

### P&O with Fixed Step

The first algorithm tested was a basic P&O algorithm with a fixed voltage step, the change in voltage every sample. The key advantage is that this method is the simplest to implement. There, however, are some key disadvantages to this approach that future methods will hope to correct. The first being the significant oscillations in the output power as mentioned in Chapter 2. As shown in Figure 6.5 below, the algorithm can find the maximum power point quickly at a low sample frequency and large voltage step but oscillates heavily around the correct maximum power point. These oscillations lead to an undesirable loss in energy. An issue like this may be corrected by reducing the size of the voltage step, but also increases the time it takes to adjust to a new MPP. The correction time issue could be remedied through the use of a faster sample time as shown in Figure 6.6 which has a sampling frequency of 0.1 MHz, but this would require the use of a significantly faster microcontroller for data processing, driving up cost.

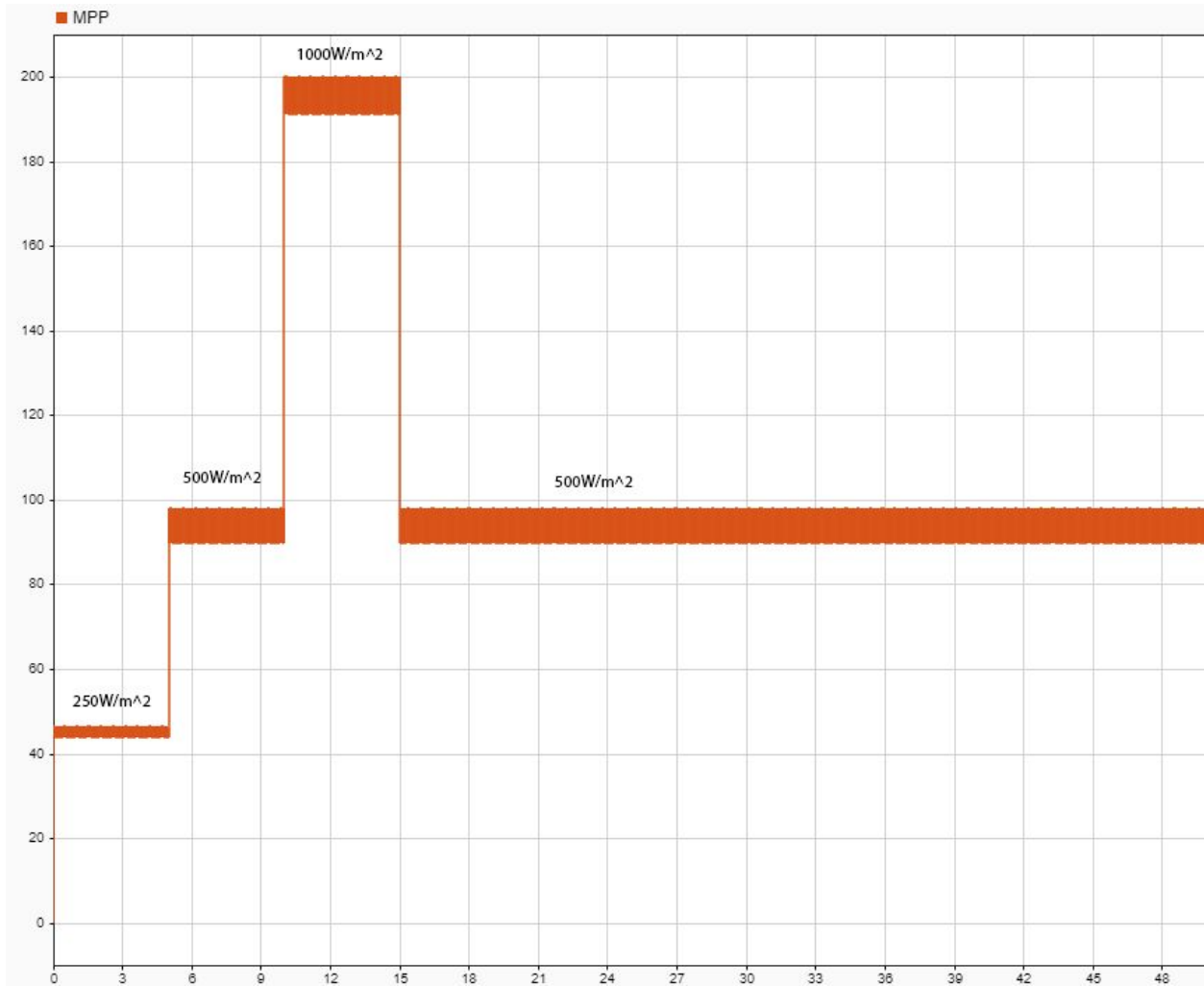


Figure 6.5: P&O with Fixed Large Step ( $T_s = 0.001s$ )



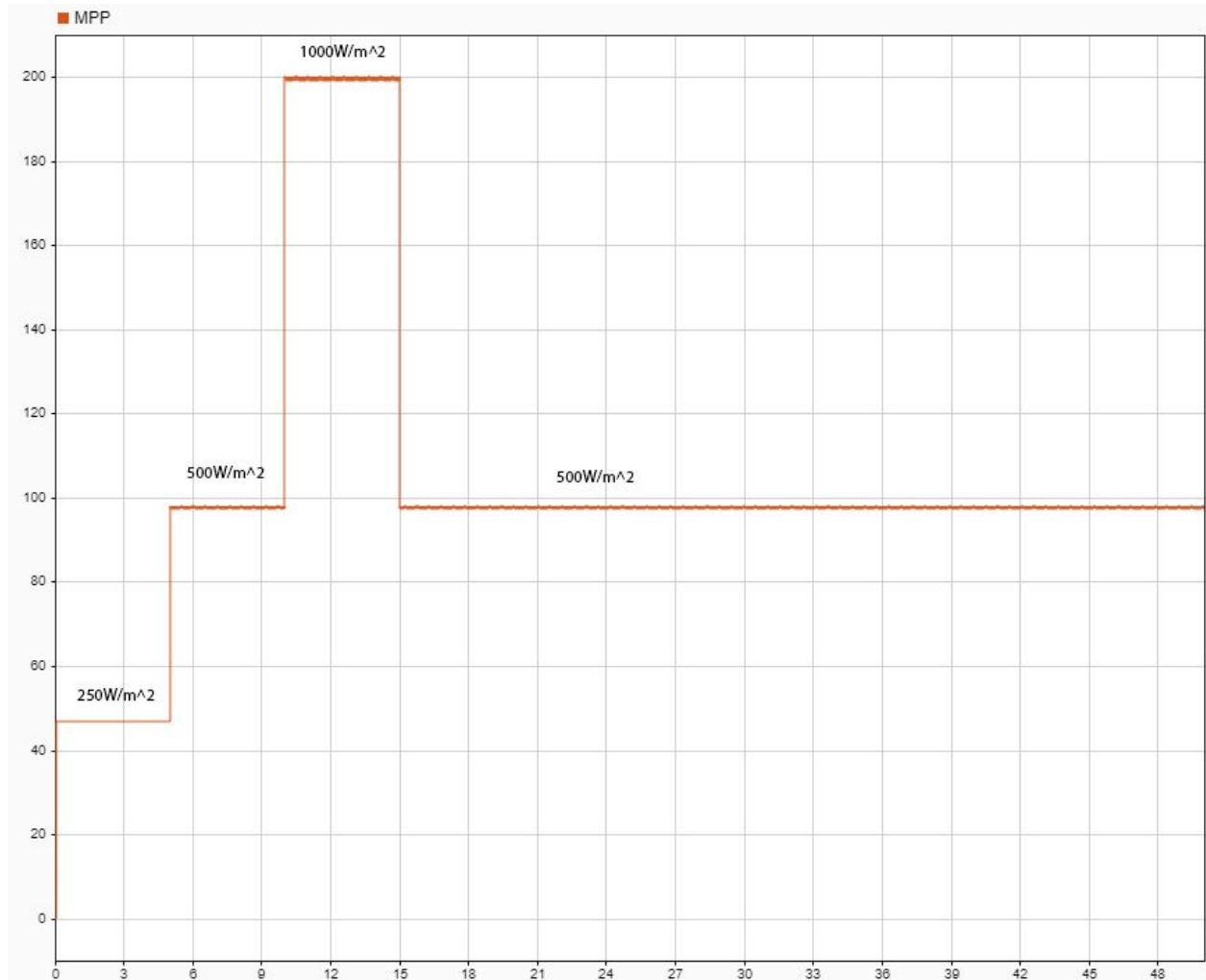


Figure 6.6: P&O with Fixed Small Step ( $T_s = 0.000001s$ )

### P&O with Successive Approximation

The goal with the second algorithm was to improve upon the original P&O by addressing the major issues with design. As shown in our marketing requirements, the original P&O algorithm failed to address requirements 2 and 3 from Table 1 in Chapter 3. The adjusted algorithm uses a variable duty cycle adjustment, meaning that depending on the past observations and actions of the controller, the duty cycle sent to the transistor in the buck converter would either increase or decrease.

Like the original P&O, the duty cycle step starts out at a set value, and as the maximum power point has not been hit, the step will continue to grow each sample. Once a sample hits where the voltage has surpassed the maximum power point voltage ( $V_{mpp}$ ), the process of successive approximation takes place, where the duty cycle step will continue to decrease as the voltage oscillates around its maximum power point until it eventually settles to a single value at steady-state. Figure 6.7 demonstrates the logical flow of this algorithm.

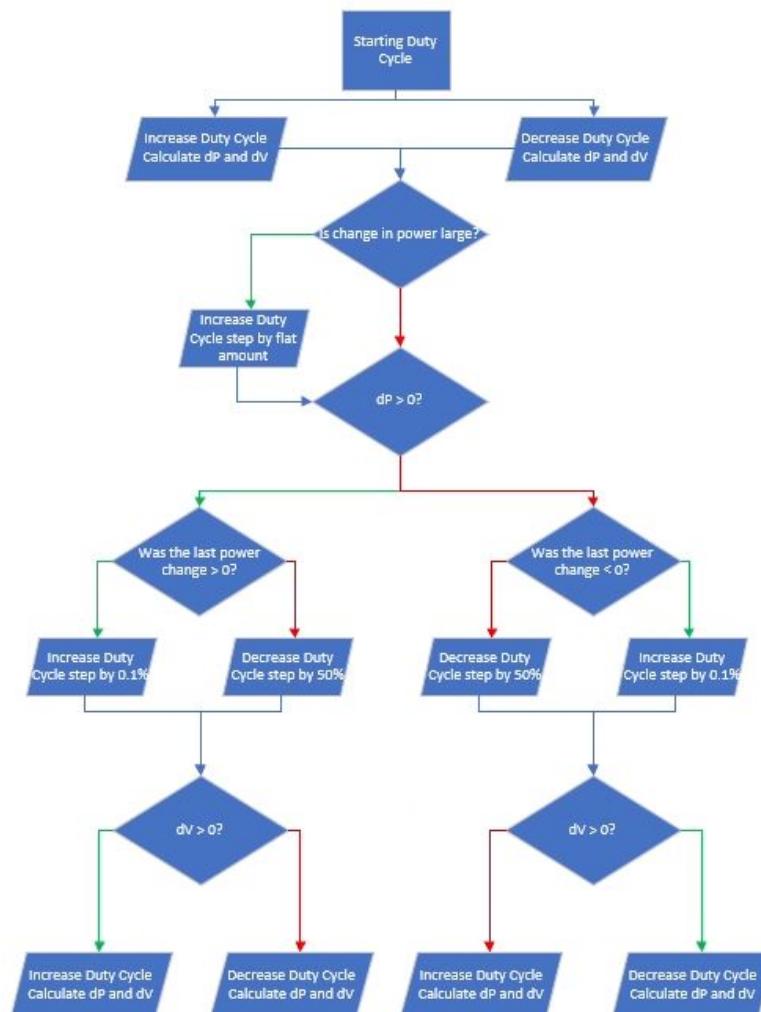


Figure 6.7: P&O with Successive Approximation Flowchart (Red - No/Green - Yes)

The advantages of this method vs. a traditional P&O are in both accuracy and speed to achieve the maximum power point. The only disadvantage this would have would be in processing power required, but due to the fact that this algorithm only remembers the action from the last sample the additional overhead would be minimal.

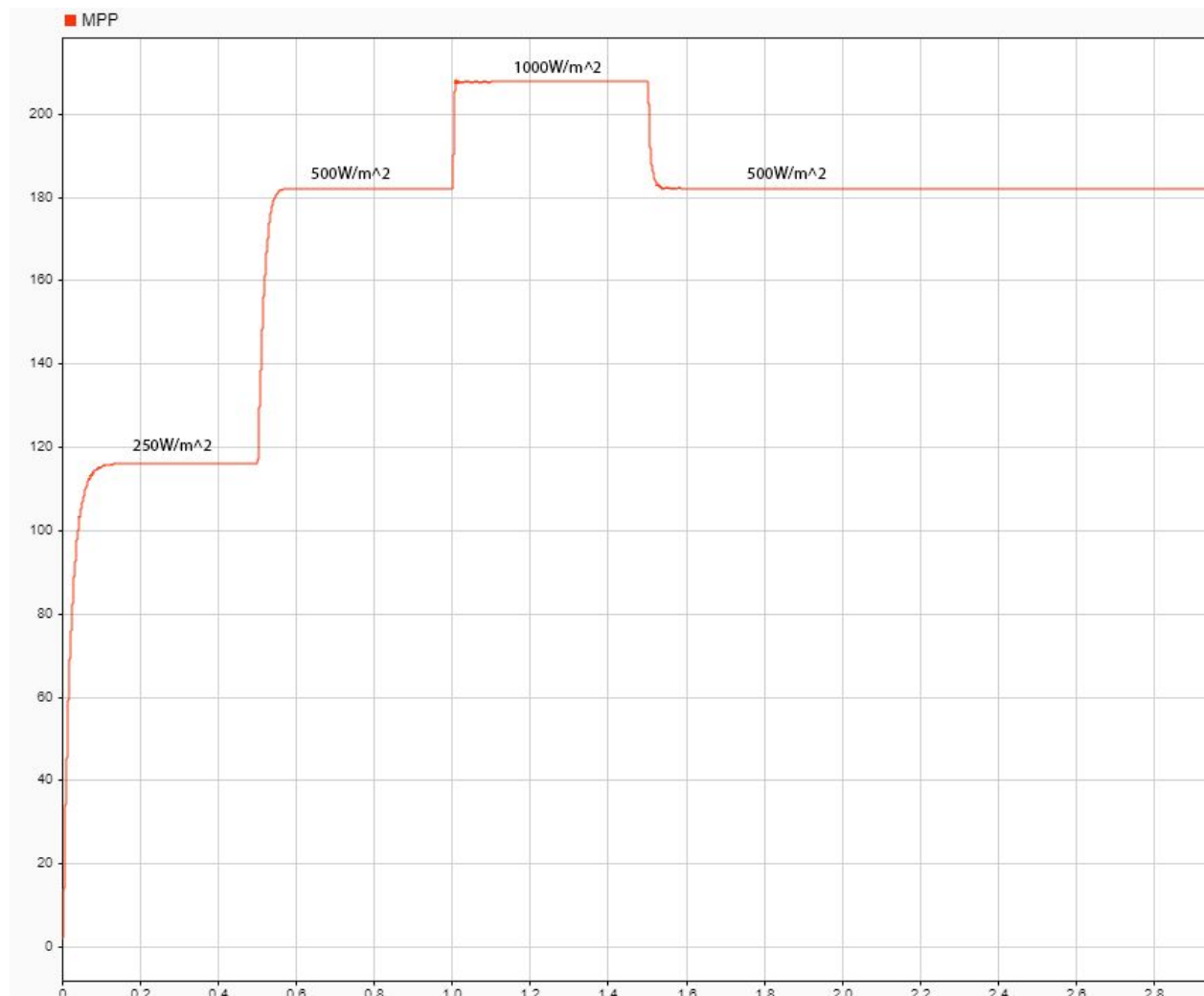


Figure 6.8: Modified P&O Power Results (Continuous Samples)

In Figure 6.8 the power at the load is measured. Like the large step P&O, the power quickly climbs with the change in irradiance and begins to oscillate as the duty cycle step begins

to fall. As the duty cycle settles at a set number, the power then settles. Shown in Figure 6.9 is the duty cycle out of the MPPT algorithm block.

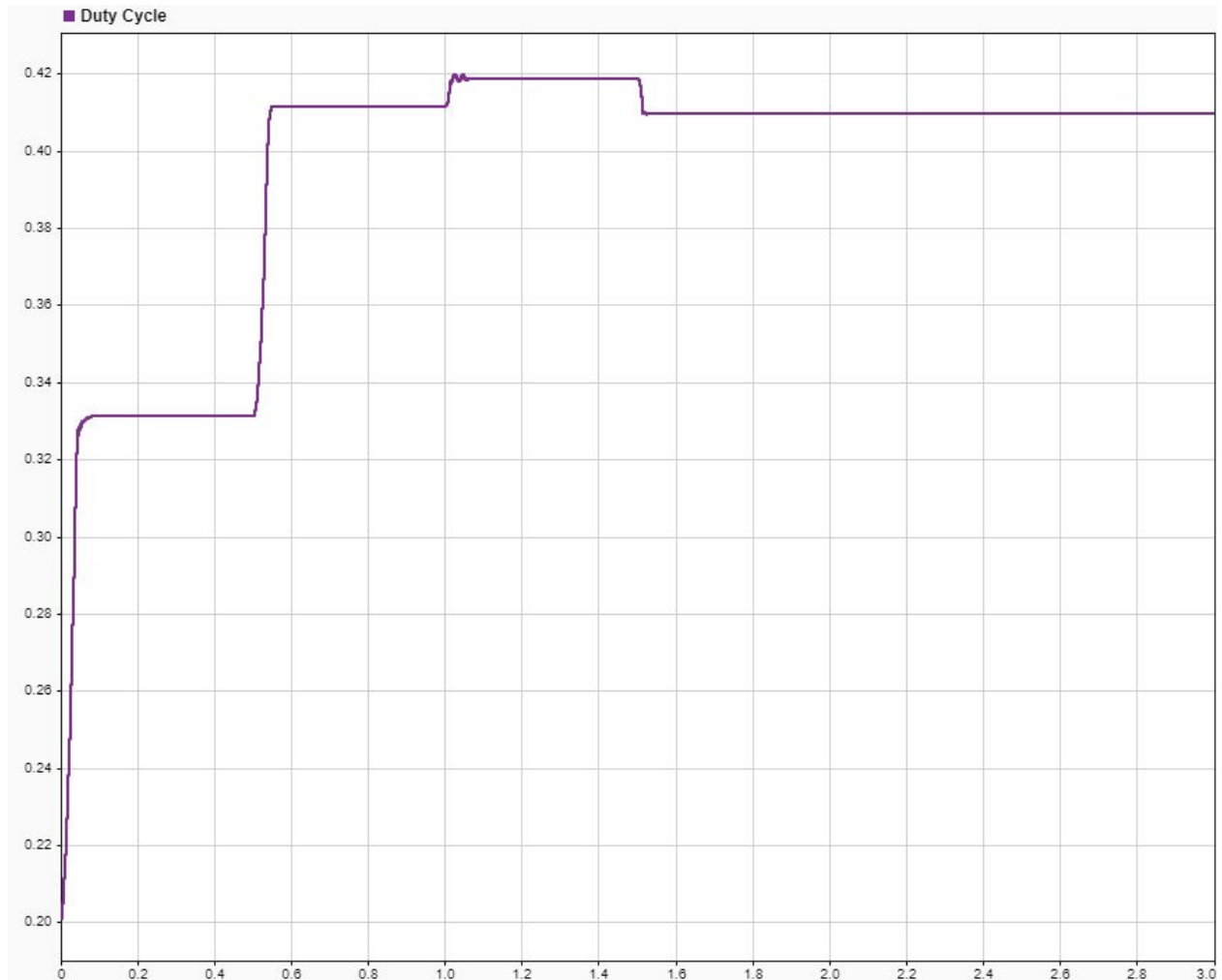


Figure 6.9: Duty Cycle Measurement

The reader should note that a continuous sample (variable step) model needs to be used here due to the fact that our future models will need to use a continuous modeling scheme. This means that the solar panel model needed to be changed to the built in Simulink Model, so although the results are generally comparable to the original P&O algorithm, the exact numbers are not. In addition, the overall simulation time needed to be made much shorter due to errors

that were arising when longer simulations were occurring as well as the significantly increased compute time for a continuous sample mode. The original P&O model data looks steeper because of this.

There is one significant shortcoming of this design. To start, the amount that the duty cycle step increases and decreases greatly impacts the results of the simulation. For example, allowing the duty cycle step to increase at a higher rate when successive voltage changes in one direction happen would decrease the number of samples needed to hit that maximum power point, but it also leads to a significantly greater overshoot when it hits the maximum power point.

Additionally, adjusting the duty cycle step value to anything higher than what is shown will result in the duty cycle never hitting the right value and jumping between the maximum and the minimum duty cycle allowed.

### **Particle Swarm Optimization Design**

The goal of the PSO based MPPT is to provide an algorithm that has an increased efficiency and improved response to partial shading of solar arrays as compared to the conventional PO algorithm.

The PSO algorithm is an application of swarm intelligence based on the behavior of birds in groups. This method of optimization was created by Kennedy and Eberhart in 1995 and has been adopted into many different applications [12]. The premise of the PSO algorithm is based on various particles trying to find the solution to a problem and ultimately coming to a conclusion through cooperation.

To begin the algorithm, the particles are initialized with arbitrary locations. The algorithm keeps track of the personal best,  $P_{best}$ , for each particle as well as the global best,  $G_{best}$ ,

the best solution that any of the particles have yet found. The velocity of the particle may be viewed as the vector that the particle uses to change location. The algorithm runs through a set number of iterations wherein each particle updates their location with the knowledge of current velocity,  $P_{best}$  and  $G_{best}$ . The equations for velocity and position are shown by equation 2 and 3, respectively.

$$v_i^{k+1} = wv_i^k + c_1r_1(P_{best\ i} - s_i^k) + c_2r_2(G_{best\ i} - s_i^k) \quad (2)$$

$$s_i^{k+1} = s_i^k + v_i^{k+1} \quad (3)$$

In the velocity and position equations the subscript,  $i$ , refers to the index of the particle and the superscript,  $k$ , refers to the index of iterations. The parameter,  $w$ , is the inertial weight constant which controls the contribution of the previous velocity to the new one. The parameters  $c_1$  and  $c_2$  weigh the importance of the individual cognition, the second term, and social learning, the third term. The  $r_1$  and  $r_2$  are random values ranging between 0 and 1. In the implementation of the PSO algorithm, the duty cycle is the position of the particle and the power output is the fitness of the particle. The weights chosen for this algorithm are  $w=0.5$   $c_1=1$ , and  $c_2=1$ .

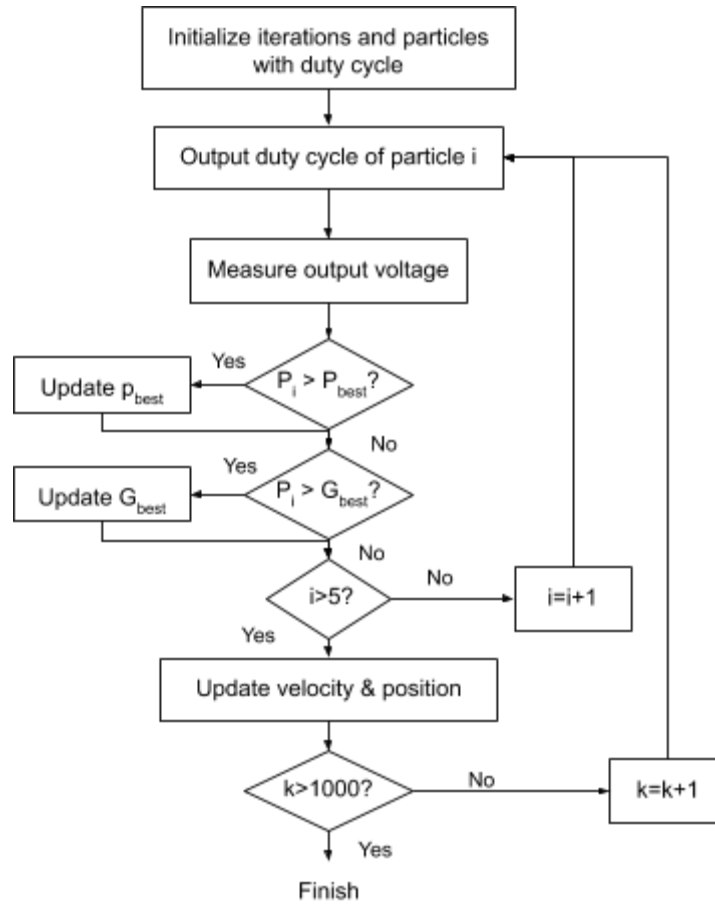


Figure 6.10: PSO Flowchart

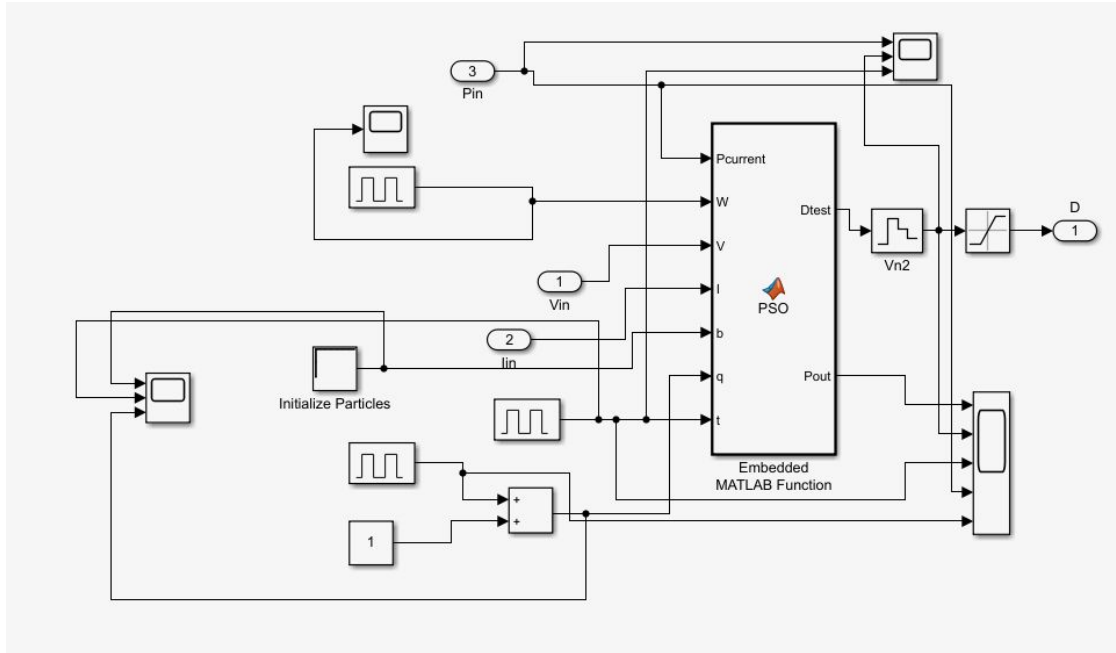


Figure 6.11: PSO Simulink Block

The input of the PSO block takes voltage and current from the PV module and uses initializing blocks to sequence through the various steps of the algorithm as well as initialize the particles' position. The embedded Matlab function code may be found in the Appendix.

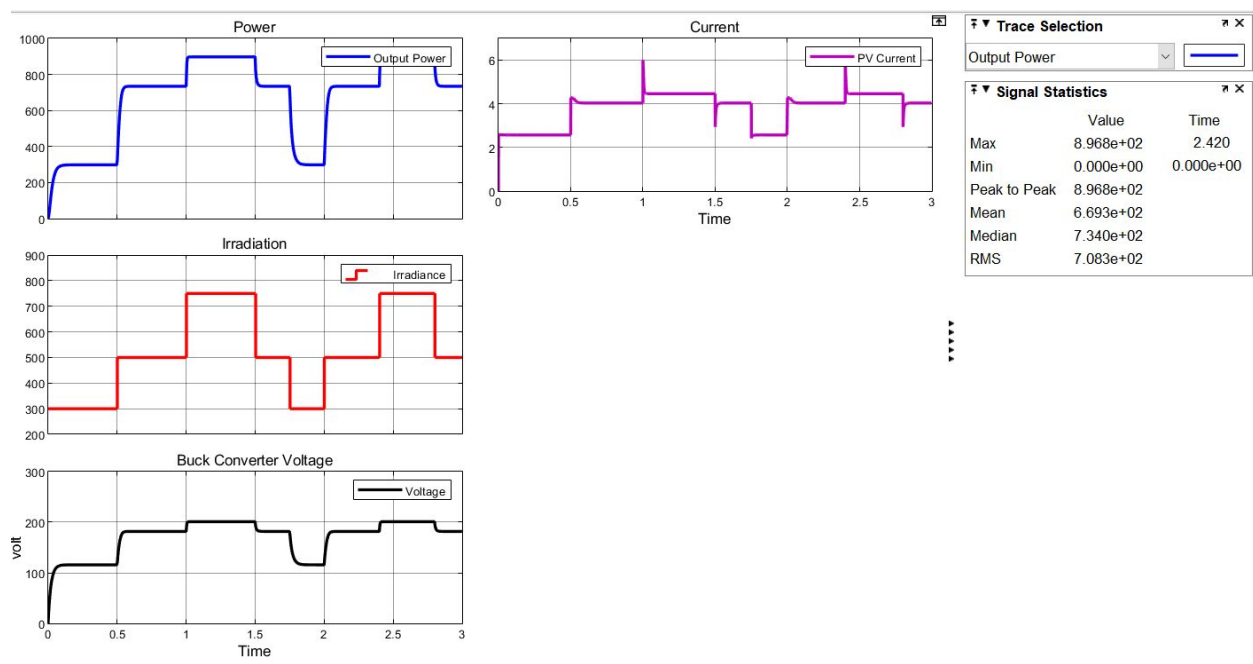




Figure 6.12: PSO Output Characteristics

When comparing the PSO and PO algorithm in typical irradiance scenarios, their performance is very similar. The PSO was set to use testing iterations of the same time as the PO, creating a standard of comparison between the two algorithms. Since the typical irradiance IV curves are fairly simplistic in nature, the algorithm's efficiency had more dependence on sampling time than effective algorithms. However, in a partially shaded scenario, the PO still has much more potential to fail to find the MPP than the PSO. The PSO is able to reach the global maxima because of the various particles spanning the range of duty cycles. To fully actualize the potential of the PSO a partially shaded simulation may be required in which the differences between the algorithms would become apparent.

The PSO outperforms the PO algorithm as it pertains to output oscillations. The PO algorithm often gets stuck oscillating around the MPP where the PSO algorithm always converges on a single MPP. As shown in the PO design portion of this report, the PO oscillations also vary with step size. Due to the nature of the project being in a simulation format, the PO step and period were able to be chosen to be very small, but in the real world, delays would be present that would limit the algorithm from performing at this capacity, whereas the PSO performs relatively better at lower sample rates.

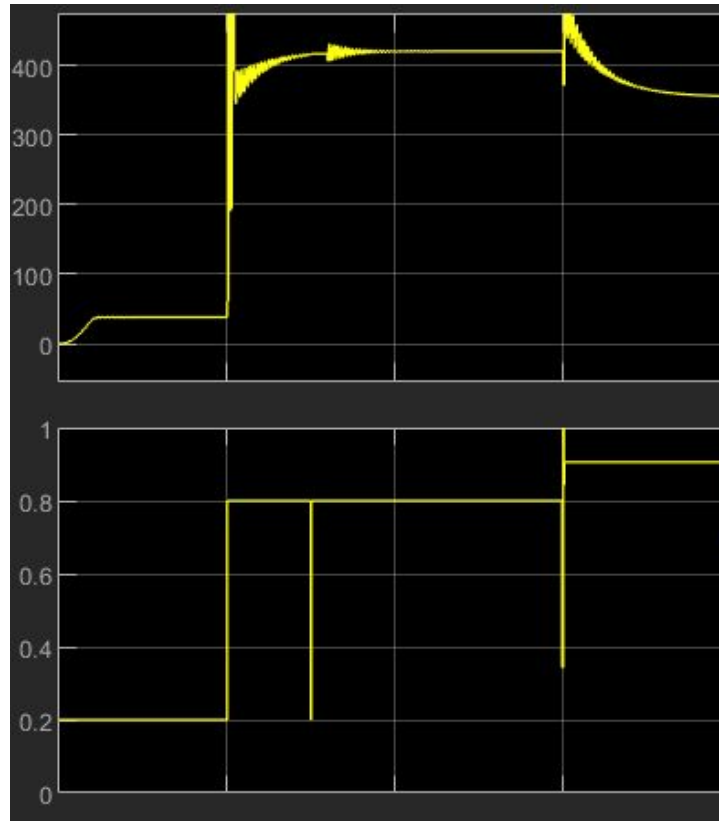


Figure 6.13: Duty Cycle and Power Output Delay (Top Power, Bottom Duty Cycle)

One of the limitations of the speed of the PSO controller is in the fundamental timing delays of the buck converter. A buck converter utilizes an inductor and a capacitor, both introducing a time delay in the output power of the converter as it pertains to a varying duty cycle. This is due to the inductor not being able to change its current instantaneously and the capacitor not being able to produce instantaneous changes in the voltage. The issue is demonstrated in Figure 6.13. The PSO algorithm updates the duty cycle instantaneously, however the velocity function is updated with the output power at that instant. In this plot, you may see that although the 0.2 duty cycle was tested, the output power only showed a slight disruption and did not accurately capture the output power for this duty cycle values.

Figure 6.14 shows the effect that the output power delay has on the conclusion of the PSO algorithm. In this case, the particles were using past power to update the algorithm because the circuit had not yet been able to produce the new output power as it corresponds to the duty cycle at test. This resulted in the algorithm being unable to conclude the most efficient duty cycle because the power readings were inconsistent with the changes. This requires that an intentional time delay be added to the algorithm so that the output power had enough time to respond.

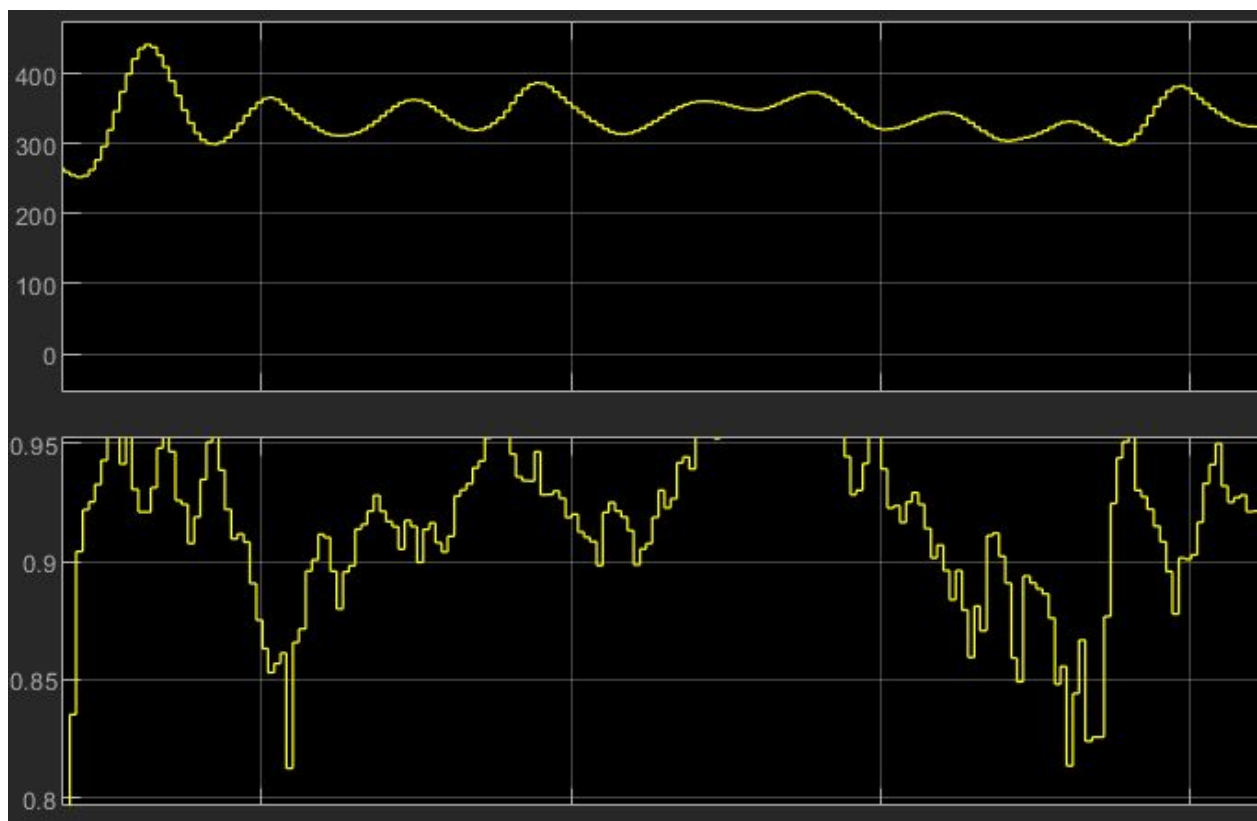


Figure 6.14: Inconclusive Duty Cycle Due to Time Delay

Another limitation acquired by the PSO algorithm is the additional time required by using many particles. While in many applications, using more particles allows for a wider span of positions, in this case the use of more particles actually slows the algorithm down because of the

initiation time for each particle. If each particle requires a set amount of initiation time, at some point the particles are still initializing while the algorithm could be making more meaningful predictions. This makes it more beneficial in the cases of delayed testing to use only two particles, as used in this simulation.

## Chapter 7 Conclusion

The goal of this project was to develop a model of a PSO based MPPT solar charge controller and simulate the power and tracking capabilities. In Chapter 6, we successfully designed and modeled both the PSO and PO based controllers with Matlab and Simulink software. Both of the models were able to successfully track the MPP given a solar panel with varying irradiance.

Due to COVID-19 and the nature of this project being required to move completely online, there are limitations to the accuracy of the data as it pertains to the real world. While the converter and solar arrays were modeled to the best of our ability, certain attributes of the physical system cannot be properly accounted for. Largely, the compatibility of the algorithms and microcontrollers should be physically tested because of the impact this may have on system speed and functionality.

Additionally, more simulations testing the partially shaded condition would help to further grasp the performance of the PSO algorithm. The similarities in outcomes between the two methods calls for a more extensive testing to evaluate the applicability of each algorithm. The PSO algorithm also may take many forms with its ability to change the social and cognitive coefficients. Moving forward, these coefficients will be further studied and evaluated in different applications. A physical prototype is also a future goal of this project so that modifications can be made for the device's true functionality.

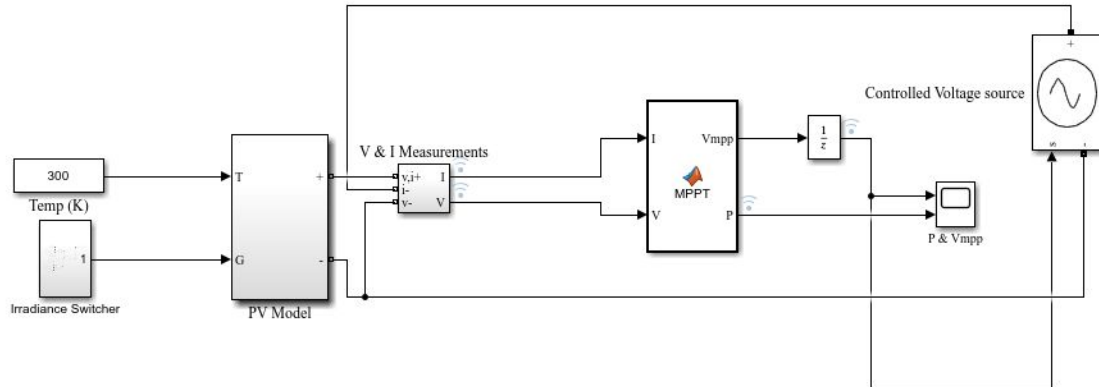
## References

- [1] F. A. Lindholm, "IV Curve," *PVEducation*. [Online]. Available: <https://www.pveducation.org/pvcdrom/solar-cell-operation/iv-curve>. [Accessed: 24-Nov-2020].
- [2] Wattneed, "Poly solar panel 150Wp 12V," *Wattneed*. [Online]. Available: <https://www.wattneed.com/en/solar-panels/8992-solar-panel-170wp-12v-polycrystalline-0712971138889.html>. [Accessed: 07-Oct-2020].
- [3] Don, "Deep cycle battery voltage & state of charge," *Energy Matters*. [Online]. Available: <https://www.energymatters.com.au/components/battery-voltage-discharge/>. [Accessed: 21-Nov-2020].
- [4] "Buyer guide - do I need a PWM or MPPT solar charge controller?," *Solar 4 RVs*. [Online]. Available: <https://www.solar4rvs.com.au/buying/buyer-guides/choosing-the-right-solar-charge-controller-regulat/>. [Accessed: 21-Nov-2020].
- [5] D. Matuszko, "RMets Journals," *Royal Meteorological Society (RMets)*, 10-Oct-2011. [Online]. Available: <https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1002/joc.2432>. [Accessed: 21-Nov-2020].
- [6] M. Chang, Y. Yao, G. Li, Y. Tong and P. Tu, "Cloud tracking for solar irradiance prediction," *2017 IEEE International Conference on Image Processing (ICIP)*, Beijing, 2017, pp. 4387-4391, doi: 10.1109/ICIP.2017.8297111.
- [7] M. R. Maghami, H. Hizam, C. Gomes, S. Hajighorbani, J. A. Kadhum, K. S. Rida, A. Alwaeli, K. A. H. Al-Asadi, M. O. A. Aqel, M. H. Marhaban, M. I. Saripan, A. Khmag, S. Alrwais, X. Liao, X. mi, D. McCoy, C. Davila-Peralta, J. Hyatt, D. Alfred, and R. Angel, "Fig. 9. Characteristics of the PV array under partial shading condition. ," *ResearchGate*, 07-Oct-2020. [Online]. Available: [https://www.researchgate.net/figure/Characteristics-of-the-PV-array-under-partial-shading-condition\\_fig8\\_292275071](https://www.researchgate.net/figure/Characteristics-of-the-PV-array-under-partial-shading-condition_fig8_292275071). [Accessed: 24-Nov-2020].
- [8] Oliveira, M., Pinheiro, D., Macedo, M. *et al.* Uncovering the social interaction network in swarm intelligence algorithms. *Appl Netw Sci* 5, 24 (2020). <https://doi.org/10.1007/s41109-020-00260-8>
- [9] "P&O based MPPT algorithm Tester," *P&O based MPPT algorithm Tester - File Exchange - MATLAB Central*. [Online]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/63406-p-o-based-mppt-algorithm-tester>. [Accessed: 21-May-2020].

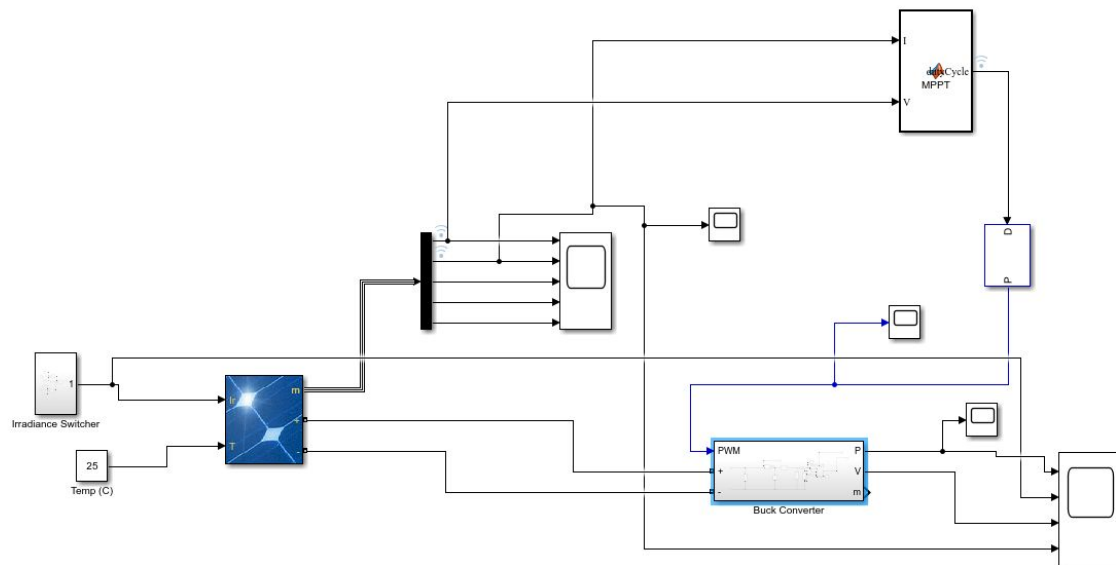
- [10] “Buck Converter,” *Buck Converter - MATLAB & Simulink*. [Online]. Available: <https://www.mathworks.com/help/physmod/sps/ug/buck-converter.html>. [Accessed: 23-Sep-2020].
- [11] “Learnabout electronics,” *Buck Converters*. [Online]. Available: <https://learnabout-electronics.org/PSU/psu31.php>. [Accessed: 25-Sep-2020].
- [12] B. Gomes de Almeida, “Swarm Intelligence - Recent Advances, New Perspectives and Applications,” in *Particle Swarm Optimization: A Powerful Technique for Solving Engineering Problems*, V. Leite, Ed. Intechopen, 2019.
- [13] D. R. Baker, “End of an atomic era: PG&E to close Diablo Canyon nuclear plant,” *SFGATE*, 22-Jun-2016. [Online]. Available: <https://www.sfgate.com/business/article/End-of-an-atomic-era-PG-E-to-close-Diablo-Canyon-8314258.php>. [Accessed: 21-Nov-2020].
- [14] “San Luis Obispo California Housing data,” *Towncharts Housing data*. [Online]. Available: <https://www.towncharts.com/California/Housing/San-Luis-Obispo-city-CA-Housing-data.html>. [Accessed: 21-Nov-2020].
- [15] “IEEE Code of Ethics,” *IEEE*. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html>. [Accessed: 24-Nov-2020].
- [16] N. Femia, G. Petrone, G. Spagnuolo and M. Vitelli, "Optimization of perturb and observe maximum power point tracking method," in *IEEE Transactions on Power Electronics*, vol. 20, no. 4, pp. 963-973, July 2005, doi: 10.1109/TPEL.2005.850975.

## Appendix

### Appendix A: MPPT Simulink Schematic with Direct Voltage Feedback (No DC-DC Converter)



### Appendix B: MPPT Simulink Schematic with Buck Converter





## Appendix C: Improved P&O MPPT Code

```
% MPPT Block in Simulink
function dutyCycle = MPPT(I,V)

persistent Vpast Ipast Ppast highFlag lowFlag pastDuty deltaD;
    % set initial values
    if isempty(Vpast)
        highFlag = 0;
        lowFlag = 0;
        Vpast = 0;
        Ipast = 0;
        deltaD = .001;
        pastDuty = 0.2;
        Ppast=Vpast*Ipast;
    end

    % calculate power
    P = I * V;
    % calculate changes in power and voltage
    dV = V - Vpast;
    dP = P - Ppast;

    % for immediate large changes in power
    if dP > 1 || dP < -1
        deltaD = deltaD + .0001;
    end
    % MPPT Algorithm
    if dP~=0
        % if power went down
        if dP < 0
            lowFlag = 1;
            % if the power is oscillating begin reducing deltaD
            if highFlag == 1
                % reset flag
                highFlag = 0;
                % reduce deltaD by 50% to settle the value
                deltaD = deltaD/1.5;
                % if voltage also went down
                if dV < 0
                    dutyCycle = pastDuty + deltaD;
                else
                    dutyCycle = pastDuty - deltaD;
                end
            end
        else
            dutyCycle = pastDuty - deltaD;
        end
    else
        dutyCycle = pastDuty;
    end
end
```

```

        % change deltaD by .1%
        deltaD = deltaD * 1.001;
        % if voltage also went down
        if dV < 0
            dutyCycle = pastDuty + deltaD;
        else
            dutyCycle = pastDuty - deltaD;
        end
    end
else
    highFlag = 1;
    % if the power is oscillating begin reducing deltaD
    if lowFlag == 1
        % reduce deltaD by 50% to settle the value
        deltaD = deltaD/1.5;
        lowFlag = 0;
        % if voltage also went down
        if dV < 0
            dutyCycle = pastDuty - deltaD;
        else
            dutyCycle = pastDuty + deltaD;
        end
    else
        % change deltaD by .1%
        deltaD = deltaD * 1.001;
        if dV < 0
            dutyCycle = pastDuty - deltaD ;
        else
            dutyCycle = pastDuty + deltaD;
        end
    end
end
end
else
    % keep current duty cycle if mpp reached
    dutyCycle = pastDuty;
end

% set limits for duty cycle and reset deltaD
if dutyCycle > .8
    dutyCycle = .8;
    deltaD = 0.001;
end
if dutyCycle < .2
    dutyCycle = 0.2;
    deltaD = 0.001;
end

```

```

end

% store values for use next sample
pastDuty = dutyCycle;
Ipast = I;
Vpast = V;
Ppast = P;
end

```

## Appendix D: PSO Code

```

function [Dtest, Pout] = PSO(Pcurrent,W,V,I,b,q,t)
persistent P iterations current_duty velocity current_pwr
pbest_pwr pbest_duty gbest_duty gbest_pwr

%Duty Cycle values for the global best value and the value
temporarily being tested
if isempty(gbest_duty)
    gbest_duty=0;
end
Dtest=gbest_duty;
if isempty(P)
    P=zeros(1,5);
end
if isempty(gbest_pwr)
    gbest_pwr=0;
end
Pout=Pcurrent;

if isempty(iterations)
    iterations=0;
end
if isempty(current_duty)
    current_duty=zeros(1,5);
end

if isempty(velocity)
    velocity=zeros(1,5);
end
if isempty(current_pwr)
    current_pwr=zeros(1,5);
end

```

```

if isempty(pbest_pwr)
    pbest_pwr=zeros(1,5);
end
if isempty(pbest_duty)
    pbest_duty=zeros(1,5);
end
c=0;

w= 0.2;
c1= 0.5;
c2=0.5;
R1=-1+rand(1,1)*2;
R2=-1+rand(1,1)*2;

%Initialize Particle 1
if b==1
    if t==0
        current_duty(1)= 0.2;
        Dtest= current_duty(1);
        pbest_duty(1)= current_duty(1);
        P(1)=Pcurrent;
        %if Delay==1
        end
        if t==1

Pout=P(1);
Dtest=pbest_duty(1);
pbest_pwr(1) = P(1);
gbest_pwr=pbest_pwr(1);
gbest_duty=pbest_duty(1);

        end

end

%Initialize Particle 2
if b==2

    if t==0
        current_duty(2)= 0.8;
        Dtest= current_duty(2);

```

```

pbest_duty(2)= current_duty(2);
P(2)=Pcurrent;
%if Delay==1
    end
    if t==1

Pout=P(2);
pbest_pwr(2) = P(2);

if pbest_pwr(2)>pbest_pwr(1)
    gbest_pwr=pbest_pwr(2);
    gbest_duty=pbest_duty(2);
end
    end
end

if b==3

%%Run Loop when W is externally set
if W==1

    for iterations=1:1

if t==0
    % Update Velocity
        velocity(q) = w*velocity(q) ...
            + c1*R1.*(pbest_duty(q) - current_duty(q)) ...
            + c2*R2.*(gbest_duty - current_duty(q));

    % Apply Velocity Limits
        velocity(q) = max(velocity(q), -0.2);
        velocity(q) = min(velocity(q), 0.2);

    % Update Position
current_duty(q)=current_duty(q)+velocity(q);

    % Apply Lower and Upper Bound Limits
        current_duty(q) = max(current_duty(q), 0);
        current_duty(q) = min(current_duty(q), 1);
%Evaluate
Dtest=current_duty(q);

```

```

%while Delay==0
    Dtest=current_duty(q);
%end
%if Delay==1
P(q)=Pcurrent;
%end
if t==1

Pout=P(q);
%Update Personal Best
if Pout>pbest_pwr(q)
    pbest_pwr(q)=P(q);
    pbest_duty(q)=current_duty(q);
end

%Update Global Best
if pbest_pwr(q)>gbest_pwr
    gbest_pwr=pbest_pwr(q);
    gbest_duty=pbest_duty(q);
end

end
end
end

if W==0

    Dtest=0.5;

end
end

```

# Appendix E

## **1. Summary of Functional Requirements**

The project is designed to find the maximum power point of a PV array given a varying set of operating conditions. The total system is provided an irradiance and temperature value to operate at and the output is the power provided to a load.

## **2. Primary Constraints**

The major constraint of our project was the impact of the COVID-19 pandemic. The original project plan that was created in EE 460 had to be reworked completely due to the fact that the original project involved a lot of in-person work, something that was at best inconvenient, and at worst impossible for collaboration. Finding a new idea in a short amount of time and spending most of spring quarter 2020 doing research that would have been done in EE 460 greatly constrained the scope of the project.

Another constraint was the limitations within the software that was used. Numerous models from Simulink and Simscape were used within a Simulink model, whilst integrating Matlab code as well. Matlab code outputs a Simulink signal and not an actual “electrical” signal, so the blocks that could be used together were quite limited. Unclear Mathworks documentation failed to help alleviate a lot of these issues.

## **3. Economic**

The project will ultimately have a positive economic impact on anyone who decides to implement what was created in a new or existing solar array where a very inefficient or primitive

charge controller is used. Solar energy is one of the largest growing forms of renewable energy around the world, and being able to better optimize the conversion of solar energy into usable electricity is something being studied and explored constantly.

The cost of the project is entirely research and development costs. Due to the fact that this project was done in simulation, the major economic impact on the authors' was the time put in to research the topics covered, such as the basis for why MPPT is needed, how MPPT works, the various ways to implement it, research how to use the software used, and the developmental time of the various algorithms. This time was valued at \$16/hour, a rough estimate for what a minimum hourly salary would be for an Electrical Engineering intern within San Luis Obispo, CA. Given an estimate of 100 hours of work, which was done at the start of the project, this equates to a total cost of \$1,600. An accurate record of time spent on the project was not kept.

If the project were to be continued further, a microcontroller, solar panel, and accompanying accessories, such as wires, a dummy load, and a DC-DC converter, would need to be purchased in order to design a physical implementation. Additional development time would also be needed in order to write, debug, and deploy C code on the microcontroller, as well as build a proper test setup. The following cost estimate was calculated:



Table E.1: Cost Analysis for Physical Implementation of Project

<b>Component</b>	<b>Estimated Cost</b>
Solar Panel	\$100
TI MSP432 Microcontroller	\$20
DC-DC Converter	\$5
Accessories	\$20
Labor	\$1,600
<b>Total</b>	<b>\$1,745</b>

There is no physical maintenance of this project but in the case of software issues, the \$16/hr rate of the engineer would be required. If this project were to proceed into physical construction, the potential for maintenance may arise but still minor in comparison to the upfront cost of the project.

The ultimate goal of a MPPT controller implementation is to increase the amount of usable power of a solar array. The increase in efficiency that this project provides, gives the customer an economic benefit by decreasing the cost of the necessary system and thereby reducing the time it takes for the system to recoup the upfront costs of build.

#### **4. If manufactured on a commercial basis:**

This project is meant to appeal to anyone who wants to use solar power as a significant source of their energy source. As solar becomes increasingly popular around the world, especially in the United States where weather conditions around the country can vary wildly, we anticipate the need for improvements in maximum power point tracking algorithms to increase.

That being said, because we are based in San Luis Obispo, CA, the residential communities within city limits would be our primary market. Like the rest of the U.S., the San Luis Obispo city and county governments are trying to move away from non-renewable energy sources in favor of clean energy, like solar. Additionally, with the Diablo Nuclear Plant, San Luis Obispo's primary source of energy, closing down by 2026, the demand for more alternative energy is pressing [13]. The city has over 20,000 housing units [14] and about 9% of homes are estimated to use solar to keep in line with California's residential solar average, with an additional 5% being added over the next year. If 20% of those systems use our solution, about 200 units will be sold.

The total manufacturing cost of the device will be \$65, which is broken down in section 3. The DC-DC converter, microcontroller, and accessories will all be included in the manufacturing cost, totaling \$45, with an additional \$20 for labor and manufacturing costs added on.

The system will be marked up by about 20%, bringing the total cost for purchase up to \$80, meaning the profit will be \$15 per device, or \$3000/year. The cost to use the device would be negligible because of the small power consumption of the device.

## **5. Environmental**

Since the project was strictly conducted in a simulation format, the environmental impact of the product was negligible. The workstations and laptops that were used did have a variety of natural resources that were used, such as steel, copper, plastics, and silicon. These systems also consumed power that was primarily produced from solar energy.

If the controller were to be manufactured and implemented, then the environmental impact would increase. We are aware that some manufacturing and transportation techniques will have a negative environmental impact, with energy required potentially coming from nonrenewable fuel sources. However, because our system works with solar, a clean renewable resource, we anticipate the negative impact to the environment to be negligible once the system is implemented. It should not harm any other living species directly.

## **6. Manufacturability**

The only foreseeable challenge with manufacturing this product would be acquiring enough microcontrollers to sell the product. At this time we do not know if Texas Instruments will allow us to use an MSP432 in a commercial product. Either getting a license from them or, worse case, designing our own microcontroller could drive up production and manufacturing costs significantly.

## **7. Sustainability**

Two major challenges arise when thinking about how the controller would be maintained. It is likely the device will be placed outside, meaning that there is a chance that water, dust, heat or insects could potentially cause damage to the electronics of the enclosure. To ensure that the device does not fail under these conditions, a rugged enclosure would need to be made around the device using aluminum or a weather proof plastic. The next challenge is the upgradability of the device. We believe the microcontroller itself is robust enough to handle any additional algorithm improvements in the future, but the challenge lies in pushing those updates to the device. Writing a software utility may be needed in order to push updates to a users PC that they can then use to push updates to the device via USB. For future iteration, a wireless module could be added in order to ensure that users can have an earlier time updating the device and potentially monitor statistics or diagnostic data in real time.

## **8. Ethical**

This project involves the collection of data from an end user's solar array system. In today's world, there is no shortage of individuals, companies, or governments that will pay to get a hold of any data from the general populous. That is why we held firm the IEEE Code of Ethics, specifically the 4th item: "to avoid unlawful conduct in professional activities, and to reject bribery in all its forms". [15]

If this project were to become a physical product, it has the power to collect and transmit data of the user's energy usage patterns. We aim to disclose any data that will be collected to the

customer and to not divulge any of it to any other entity, regardless of whether any form of compensation is offered.

Ethical code 5: “to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest and realistic in stating claims or estimates based on available data, and to credit properly the contributions of others” is also directly applicable to the project. As designers of the system, we welcome honest feedback for our work and have ensured that any outside resource used towards the completion of this project is credited.

The project does have the potential to allow for a malicious person or entity to inject malicious code into the microcontroller that could cause the solar system to behave erratically or to shut it down completely.

## **9. Health and Safety**

Due to the simulated nature of this project, we did not consider the health and safety of either us or any potential customers to be at risk. Therefore, no extra measures were taken to ensure a safer environment for any party involved.

## **10. Social and Political**

The major political issue this project could be implicated upon is the issue of climate change as it directly deals with renewable energy technology. The side of debate that the project would lie on is pro-renewable energy, meaning it impacts any individual or entity that uses solar power. These stakeholders would benefit from the algorithms used here because it increases the

amount of power that a solar system produces for a given load. The project should not harm any stakeholder.

Due to the hobbyist nature of the algorithm, the stakeholders that benefit the most are individuals or families that own homes or trailers rather than businesses. Businesses, especially larger ones, are more likely to use a more sophisticated system or a system built in-house that is more catered to their needs.

## **11. Development**

Because of the COVID-19 pandemic, the project had to be converted to an entirely virtual one. This meant that in order to accomplish our goals of creating and testing these MPPT algorithms we needed to learn how the function of a solar system with MPPT, simple power electronics knowledge such as how a DC-DC converter works, and how to use Matlab and Simulink to simulate the total system.