Design and Optimization of a Teensy-Based Embedded Solar Panel Controller

Ian Henriques
ECE464 Project, Spring 2023

***Abstract*—The aim of this project is to develop a Teensy-based solar controller, for use in powering a fixed resistive load, with maximum power efficiency. A closed-loop control system is implemented, using an INA3221 current-and-voltage sensor for power metrics, a Teensy 4.0 microcontroller for processing and decision-making, and a student-designed buck (voltage step-down) converter as an actuator to regulate the load voltage of an 18V (225mA) solar panel. Once the system is able to stably find the maximum power point (MPP), optimizations will be added to allow for more efficient scheduling and complex decision-making. This project compared two previously simulated (but not practically compared) approaches, perturbation-and-observation and gradient descent, and determined their strengths and weaknesses in terms of steady-state and transient performance, as well as consistency and power conversion efficiency, by controlling the load of a solar cell under simulated sunlight from a lamp.**

I. Introduction

The amount of solar panels in use today is ever-increasing, which has heightened the importance of efficient light-to-electricity conversion. Much like other electronic components, solar cells have output voltages and currents which depend highly on the load impedance [1]. In addition, the power of a solar cell varies significantly according to environmental conditions, including the luminance and spectra of incident light rays [2]. In many cases, due to fluctuations in light levels and non-uniform temperatures, the electrical characteristics of a solar cell may drastically change, which can adversely impact efficiency if the load remains the same [3].

While the environment is difficult to control in practical solar applications (due to external factors like clouds and shadows), there are ways to use DC-to-DC converters to tune the load of the solar cell [4]. These DC-to-DC converters use a combination of capacitors, inductors, and switching to adjust the effective impedance of the load, from the perspective of the solar cell, changing the solar cell's power output as a result [5]. Applying this load-tuning idea, an approach called **Maximum Power Point Tracking (MPPT)** allows an embedded system to approach the maximum power output given the environment [6]. Several algorithms, including classical and artificially intelligent ones, may be used to regulate the power output of the solar cell in different ways [7].

Various papers in the surrounding literature have proposed and simulated techniques for implementing closed-loop MPPT algorithms. The simplest of these is the perturb-and-observe algorithm [8] [9], which adjusts the load impedance incrementally and uses constant directional feedback to approach the maximum power output for the given environment. Another alternative, gradient descent [10], and other similar techniques like steepest descent with centered differentiation [11], may use models of solar cell parameters and root-finding algorithms to converge more quickly to the maximum power.

The aforementioned algorithms in the literature have been primarily tested in idealized simulation environments including MATLAB Simulink and Spice. While each algorithm has its own merits, they have rarely been compared in controlled real-world, real-time environments. Therefore, the question arises: *In the real world, with a physical solar cell, DC-DC converter, power sensor, and microcontroller, how well do the perturbation-and-observation (P&O) and gradient descent (Gradient) algorithms perform against each other?*

The goal of this project is less to test the practicality of these MPPT methods for large industrial solar cell arrays, and more to develop a proof-of-concept implementation which identifies the strengths and weaknesses of each algorithm in an embedded, low-power, microcontroller-based setting. Below are some of the accomplishments of this project:

- A fully functional buck converter was designed using a bootstrap gate driver circuit and logic-level MOSFET, to adjust the effective load seen by the a solar cell when powering a 100Ω resistor.
- An INA3221 industrial current-and-voltage monitoring sensor was connected to the solar cell and configured to sample accurate solar panel power readings for testing.
- Two algorithms were written, the P&O algorithm and a student-developed variant of the Gradient algorithm, for testing in a simulated sunlight environment.
- Power time-series were collected for both algorithms in 4 initial duty cycle conditions (3 trials each), to test both steady-state and transient performance, as well as consistency of MPPT convergence between the trials.
- Using multimeter measurements at steady state for each of the trials, output powers across the load were calculated and compared against the measured solar panel powers to determine the power conversion efficiency of the circuit when using the two algorithms.

Section II (Background) provides information about general MPPT system design and the algorithms which were tested. Section III (Motivation) demonstrates the tradeoffs of both algorithmic approaches by using a simplified, linearized example. Section IV (Methodology) discusses the circuit design, firmware decisions, and algorithmic details which are tested in the project. Section V (Experimental Data, Results, & Analysis) provides detailed results and analysis of experimental findings, accompanied by the detailed time-series charts in the Appendix (Section IX). Section VI (Discussion & Limitations) describes the accomplishments and suggestions for future work for extending the results of the project. Finally, Section VII (Conclusion) summarizes the findings of the project. References can be found in Section VIII.

II. Background

MPPT control has been extensively used for controlling large-scale solar systems. Though specific design choices vary, the circuit usually consists of a solar cell, connected to voltage and current sensors, which are connected to a computer that actuates a DC-to-DC voltage converter (Fig. 1).

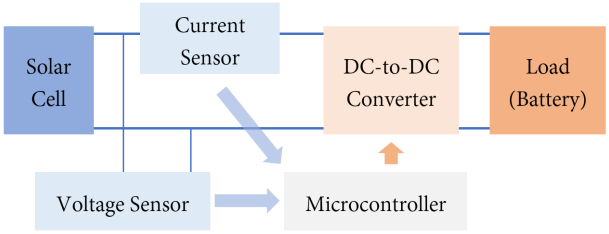

Fig. 1: Typical Design of an MPPT System

This project consists of four main design phases: hardware design for the DC-to-DC converters, hardware design for power sensing, sensor and actuator firmware, and high-level MPP tracking. Several existing MPPT controller designs in the surrounding literature have influenced the choices made during this small-scale MPPT Teensy implementation.

However, before delving into those details, it is first necessary to understand the theory behind solar cells to see why MPPT helps achieve maximum power in the first place.

The underlying principle behind MPPT is illustrated by the I-V characteristic of an ideal solar cell in Figure 2. When a solar cell is short-circuited, it will produce a short-circuit current I_{sc} with zero voltage, resulting in zero electrical power. Similarly, an open-circuited solar cell will produce an open-circuit voltage of V_{oc} with no current, once again with no power output. Between these two extremes, for an increasing load impedance, the current gradually decreases and voltage gradually increases, so that at some load voltage V_{mpp} (the **maximum power point**), the power output (area of the blue rectangle in Fig. 2) is maximized.

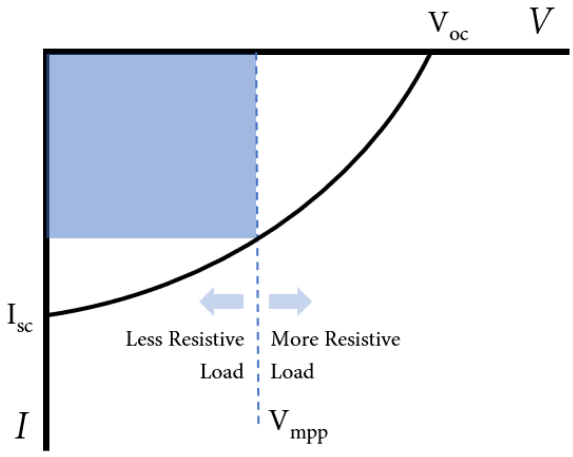

Fig. 2: I-V Characteristic of a Solar Cell in its Operating Region

In more realistic situations, non-uniform irradiance due to partial shading may lead to multiple 'local maximum' power points, due to some areas of the cell receiving more light than others [6]. In these cases, the MPPT algorithm simply settles on one, even if it is not the global maximum [6]. Much of the research effort in the surrounding literature focuses on the partial shading condition, as it is more difficult to model and is less of a clear-cut optimization problem due to its introduction of multiple local peaks of power [13].

It is important to note that the shape of the I-V characteristic itself is dependent solely on the environment of the solar cell. The goal of MPPT is simply to move along the I-V characteristic curve without changing it, to find the load at which the power is maximized for the environment.

Now that a rationale has been provided for changing the load of the solar cell to maximize the solar output power, it is important to address the hardware that makes this possible.

A. *DC-to-DC Converter Design*

Hardware design of MPPT controllers is typically focused on the DC-to-DC converter. Common DC-to-DC converters include buck converters (to step down voltage), boost converters (to step up voltage), and boost-buck converters (which can step voltage up or down) [4]. Each of these uses a combination of diodes, capacitors, and inductors, with high frequency MOSFET switching to generate high voltage inductor spikes for voltage shifting. The values of the inductors and capacitors should be carefully set, along with the clock frequency, to limit the output voltage and current ripple within a certain range [4].

This project utilizes a buck DC-to-DC converter (Fig. 3). This is because the solar cell provides a small current, so it would be better to step down the voltage and increase the current to supply more power to less resistive loads.

It is important to note that driving the MOSFET (Fig. 3) on the high-side means that a gate driver with a bootstrap circuit must be used to ensure the FET conducts adequately (i.e. guaranteeing the gate-to-source voltage is greater than threshold). As will be described in the Methodology, the IR2110 gate driver is used for this purpose.

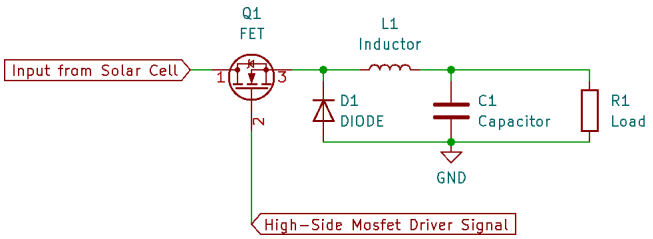

Fig. 3: A simple KiCAD schematic for a buck converter

To control the voltage level of buck converters, which effectively adjusts the total impedance seen by the solar cell, it is necessary to manipulate D, the duty cycle of the switching, defined by the fraction of the time the FET is on (conducting) [14]. The switching frequency should remain the same.

For a buck converter, a higher duty cycle leads to a higher (i.e. less impedance, in a power electronics context) load seen by the solar cell, as effectively, the solar cell is connected to the load for a longer period of time, drawing more current and

dropping its voltage. To alter the duty cycle of a buck converter MOSFET, a variable-duty-cycle pulse-width modulation (PWM) input from the Teensy microcontroller is utilized.

B. Power Sensing

Another hardware consideration for building a Teensy-based MPPT controller is voltage and current sensing. To enable a feedback loop that approaches maximum power, high-precision voltage and current sensors for calculating power are necessary. Voltage measurement is simple, as it just involves measuring the difference between the potentials of the solar cell rails. Current measurement for MPPT systems is often done by measuring the voltage across a low-resistance, high power shunt resistor in series with the solar cell.

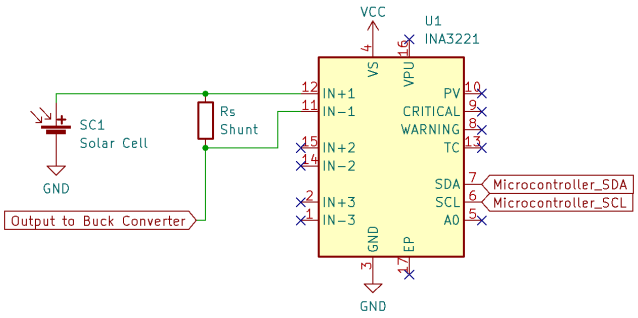

Fig. 4: KiCAD schematic for INA3221 power sensing

To sense voltage and current, this project uses the INA3221 sensor, which can sense between 0V and 26V. By using two pins for each voltage measurement, a shunt-resistor can be added to the sensor to measure both voltage and current simultaneously. These values are then sent, via the I^{2}C bus, to the microcontroller to enable processing and closed-loop control by actuating the buck converter (Fig. 4).

C. Sensor & Actuator Firmware

As previously described, the MPPT cyberphysical system in this project involves reading current and voltage data from the INA3221 power sensors, processing the data, deciding whether to raise or lower the voltage, and then making an appropriate adjustment to the duty cycle of the buck converter to approach the maximum power point voltage, V_{mpp}.

To communicate with the INA3221 sensors, it is necessary to use an I^{2}C driver. I^{2}C drivers are freely available online, and one has been used for the INA3221 firmware in this project.

Fast I^{2}C communication is done at 400 kHz, implying at least a 2.5μs time gap between successive measurements. However, since the illumination level of the solar cell is unlikely to fluctuate greatly at high frequencies under a constant light source, the sensor data will likely be frequent enough to capture and react to changes in V_{mpp} within a few iterations.

Actuation of the buck converter is done through a PWM library on the Teensy microcontroller, which generates a waveform based on a specified frequency, resolution, and duty cycle. High PWM frequencies are generally useful for controlling buck converters, as they decrease the voltage output ripple for a given inductor and capacitor combination [5]. It is important to note there is a tradeoff between PWM frequency and resolution, as higher PWM frequencies imply fewer microcontroller clock cycles dedicated to each PWM pulse, reducing the granularity of the duty cycle ratio. The duty cycle itself will be adjusted by the high-level MPPT algorithm in real time, so that the PWM signal sent to the buck converter can be changed quickly in response to environmental changes.

D. High-level MPPT Algorithms

With respect to the MPPT algorithm itself, there are many possible approaches for tackling the problem of tracking V_{mpp}. However, the general problem to solve is the same: for a unimodal Power vs Voltage curve (termed a **P-V characteristic**), the algorithm must find and stablize to the maximum value of the function as quickly and accurately as possible (Fig. 5).

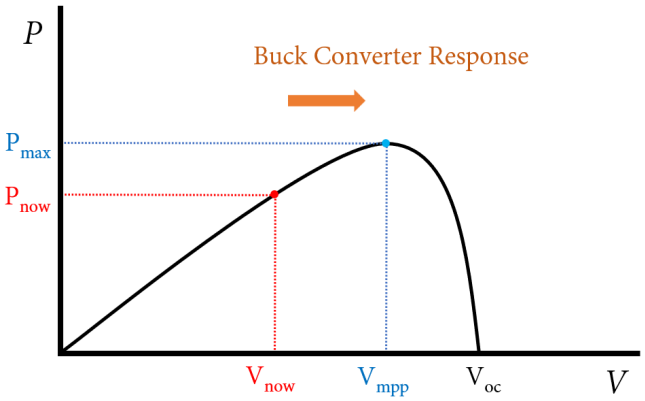

Fig. 5: An illustration of a buck converter response under MPPT

1) Perturbation and Observation: One of the simplest and most common implementations of MPPT is the perturb-and-observe algorithm [8]. Since the goal is to optimize the power, a unimodal function for ordinary solar cells, the power values at two different voltages can be compared against each other to determine whether to raise or drop the voltage seen by the load. By this principle, a system can find the maximum power point simply by nudging the voltage in one direction, measuring the impact on power delivery, and reversing the direction of actuation if necessary. This approach is simple to implement and often quickly converges to the maximum power point [9]. However, it has the disadvantage of oscillating around V_{mpp}, as constant perturbations around the maximum power point can often overshoot or knock the voltage away, resulting in efficiency losses compared to other methods [12].

2) Gradient Descent: Gradient descent, just like for neural networks, is especially useful for finding the extremum of a cost function, and this feature makes it useful for MPPT [10]. Effectively, the 'cost' function of gradient descent in this context is the negative of the power function, so that the cost will be minimized when the power is maximized. Unlike the perturb-and-observe approach, it uses the measured slope of the Power-Voltage characteristic in Figure 5 (which can be estimated as the change in power over the change in voltage) to decrease the strength of the perturbations as the power

converges to the maximum. This is more computationally complex compared to the perturb-and-observe algorithm, but it has the advantage of smoothly converging to the maximum power point without much oscillation. Other similar differentiation-based algorithms, including the Newton-Raphson root-finding method and steepest descent by centered differentiation [11], have been explored in the literature and are worth considering as variants of gradient descent.

E. *Comparing Performances of Algorithms*

Once the hardware and firmware are implemented, the objective of this project is to select and program versions of the perturb-and-observe and gradient descent approaches for testing in different illumination and environmental conditions. Considerations including the steady-state behavior, transient behavior, inter-trial consistency, and power efficiency are taken into account when judging the algorithms.

III. Motivation (Simplified Example of Approach)

To illustrate the MPPT approach, it is useful to consider an application with a pre-determined solar cell P-V characteristic curve, to both understand the role of the buck converter in the control loop and detail how the high-level algorithms would approach the MPPT problem differently.

In essence, MPPT is an incremental algorithm. It chooses to slightly change the buck converter duty cycle, which ultimately changes the voltage level of the load, as well as the impedance from the point of view of the solar panel. Then, based on the outcome of the duty cycle change, i.e. the new voltage and power measurements from the power sensors, the algorithm must determine which change to make next.

As an ideal approximation, assume that locally, the P-V characteristic looks roughly linear. This assumption is valid when considering small adjustments made to the duty cycle (and hence small adjustments to the voltage) with each iteration of MPPT, slightly adjusting the position on the solar panel's P-V characteristic. Recall that the I-V (and by extension, P-V) characteristic of a solar cell is solely environment-dependent, so the buck converter's variable load will not change the shape of the curve itself.

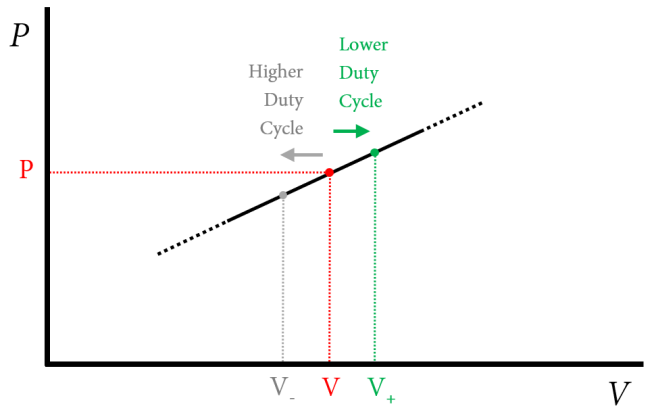

Fig. 6: An locally linearlized segment of the P-V characteristic

Figure 6 displays an example of what the linearized P-V characteristic curve may look like. To increase the voltage (move farther right along the linearized curve, to the green point), the system must drop the effective load of the buck converter, from the perspective of the solar panel. This means the duty cycle, which is proportional to the strength of the load, should decrease. Conversely, it can be seen that a higher duty cycle will drop the operating voltage of the solar cell, moving towards the gray point on the left.

In this example (Fig. 6), all versions of the MPPT algorithm should be able to determine that the green (lower duty cycle) operating point increases the power output of the solar cell, bringing it closer to a local maximum power point. If the segment of the P-V characteristic were instead downward sloping, it could be inferred that the MPP is at a lower voltage level (higher duty cycle of the buck converter).

What differs between the two algorithms is how much they decide to adjust the operating point of the solar cell:

- The perturb-and-observe algorithm [9], the simplest option, adjusts the duty cycle by a fixed step size (maybe $1/1024 \simeq 0.1\%$) in the direction of higher power. This may cause an overshoot past the desired operating voltage level, if too close to the MPP [12]. Note that smaller steps may fix the overshoot, but at the cost of tracking the MPP more slowly in case environmental conditions change.
- The gradient descent algorithm [10], meanwhile, would adjust proportionally to the slope of the P-V characteristic segment, adjusting the duty cycle more drastically if the segment is steeper. This may help to decrease overshoot, by slowing down the perturbations near the MPP when the curve is less steep, but it may also experience difficulties correcting in regions where the P-V characteristic has a low slope but is not maximized.

Each of the MPPT algorithms attempt to move towards a higher power operating point, but they do so in different ways. This project aims to compare their real-world performances at tracking the maximum power point, both in in terms of steady-state and transient behavior.

IV. Methodology

As described in the Background section, the design approach in this project consists of four main phases:

A) The first goal is to complete hardware organization and wiring for the buck converter, including any necessary MOSFET drivers or other auxiliary components.
B) Second, wiring of the INA3221 sensor must be planned out and added to the circuit, in series (for current readings) and parallel (for voltage) with the solar panel.
C) Third, firmware is implemented so that the Arduino can independently communicate with the power sensor and produce a PWM signal to drive the buck converter, without any control algorithms (i.e. with manual control from a potentiometer input for testing purposes).
D) Fourth, the two "paths" (INA3221 sensor reading and buck converter actuation) converge, allowing for the implementation of various versions of MMPT algorithms,

including perturb & observe gradient descent (the two algorithms of interest).

Then, the performances of the various algorithms are compared against each other, and any necessary optimizations are made to maximize the efficiency, reaction speed, accuracy, and other characteristics of the MPPT algorithms.

Shown below is a top-level flowchart (Fig. 7), which sums up the milestones of the approach:

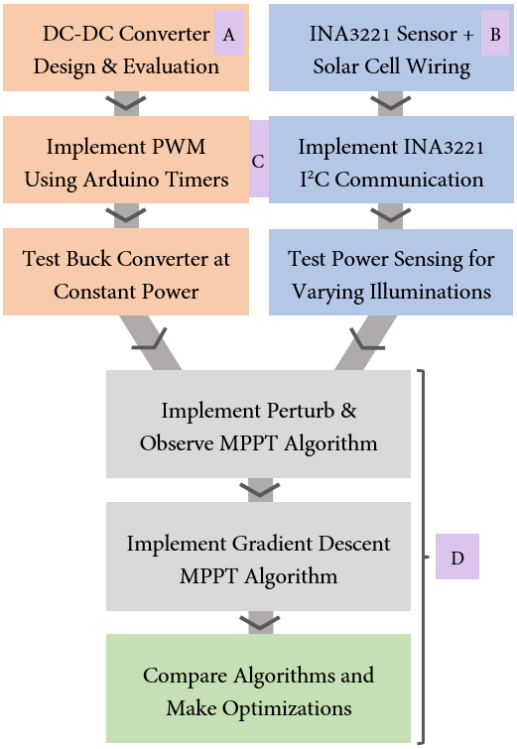

Fig. 7: Flowchart of tasks for the MPPT project

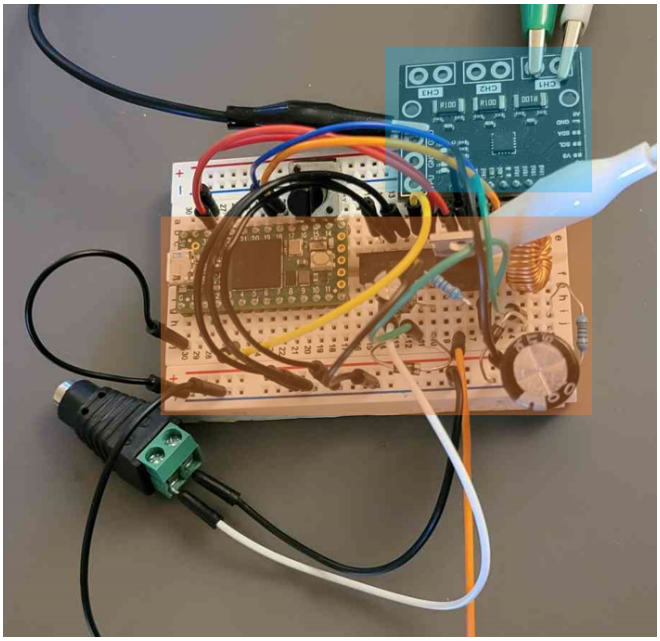

Fig. 8: An image of the final breadboard hardware layout (Orange = DC-to-DC Converter, Blue = INA3221)

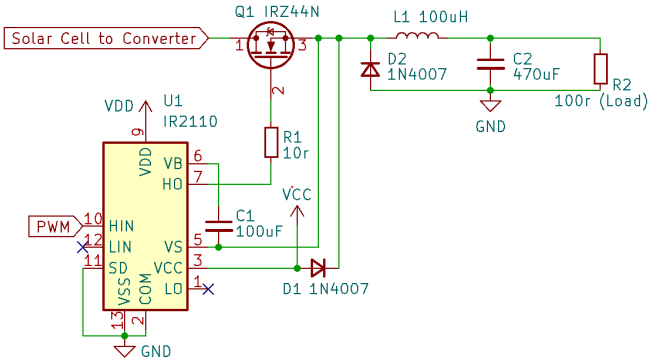

Fig. 9: A complete schematic for the DC-to-DC converter

A. *DC-to-DC Converter Design*

The first step of the circuit design involved building the DC-to-DC converter (see the orange region of Fig. 8). Due to the design choice to use a buck converter, a MOSFET gate driver chip (**IR2110**) was necessary. The function of the gate driver is to quickly and reliably drive the gate of the FET using an input PWM waveform from the Teensy. It has high-side floating output and return paths to bias the FET gate and source (respectively), to ensure that enough gate-to-source voltage can be provided to fully turn on the FET when the input signal is high. See Figure 9 for a more detailed schematic of the buck converter designed in this project.

In conjunction with the gate driver, a "bootstrap" circuit powered by V_{CC} (an external 12 V wall power supply) was also necessary. Since the gate driver itself was not able to produce enough voltage (relative to circuit ground) to open the high-side MOSFET, a diode (D1) and capacitor (C1) were utilized as a charge pump [15], ensuring the FET would open completely (much beyond its 2V threshold) when the PWM signal went high. A 10Ω (R1) was added between the gate of the FET and the high-side driver output, to dampen out parasitic oscillations from the high-frequency switching.

The FET chosen for this project was the **IRZ44N**, a logic-level (low threshold) and low resistance MOSFET, for connecting and disconnecting the buck converter to the solar panel according to the Teensy duty cycle.

Note the use of the alligator clip in Figure 8, between the INA sensor low-side (effectively the solar panel voltage) and the heat-sink of the FET. Since the heat-sink of the IRZ44N is tied to its drain terminal, the alligator clip provides a way to power the buck converter without jumper wires.

To test the DC-to-DC converter under a constant power source, the alligator clip connection to the solar panel was replaced with a connection to V_{cc} (an external power supply for driving the gate driver chip). After several rounds of testing to ensure optimal power conversion (i.e. continuous conduction mode of the inductor, minimal ripple of the output waveform, and a low-latency response to duty cycle adjustments made using a potentiometer input), a 100Ω resistor was chosen as the load, and the PWM frequency was set to be 100kHz.

B. Power Sensing

Power sensing was done by attaching the INA sensor terminals to the solar panel and the output to the buck converter, exactly as illustrated previously in Figure 4. Once again, alligator clips (Fig. 8) were used to allow for easily removable, low-resistance connections between the solar panel and the inputs to the INA3221 and buck converter FET.

The hardware connections for the INA3221 sensor were relatively simple: the I^2C (SDA and SCL) pins were connected to those from the Teensy, and the sensor was powered using the Teensy's voltage and ground. External warning and alert pins of the INA3221 were left unconnected.

Since the INA3221 breakout board comes with a pre-soldered 100mΩ shunt resistor, no additional modifications were necessary to provide current sensing capabilities to the sensor. Simply, every time a shunt voltage was read, it would be divided by the fixed 100mΩ to calculate the current passing through the circuit (i.e. the solar panel).

The INA3221 sensor was tested in series with the solar panel in various lighting conditions, to ensure that it could calculate currents and voltages accurately and reliably. Results were tested against a multimeter to make sure that the INA3221 voltage and current readings were properly calibrated.

C. Sensor & Actuator Firmware

The firmware for the buck converter was straightforward, as it merely involved setting a PWM frequency and resolution for the Teensy's internal PWM timer, and then adjusting the duty cycle when necessary (such as when a duty cycle-selecting potentiometer was turned during testing):

```
pwm_duty ← 0
while (manual control) do:
    pwm_duty ← analogRead(15); // 10 bit value
    analogWrite(pwm_duty);
end while
```

Writing and testing the INA3221 firmware, however, was more complex. An INA3221 sensor library was used, to sample the voltage and shunt voltage values periodically. Due to occasional spikes in the INA3221 sensor voltage, it was necessary to create a software-based low-pass filter to smoothen out the input voltage and current waveforms to prevent spurious feedback from reaching the MPPT algorithm:

```
lp_volt[SIZE]; // Low-pass ring buffer for voltage
lp_curr[SIZE]; // Low-pass ring buffer for current
index ← 0;
while (sampling) do:
    // Sensing:
    lp_volt[index] ← lp_volt[index] + getVolts();
    lp_curr[index] ← lp_curr[index] + getCurr();
    // Averaging:
    V ← (lp_volt[index] - lp_volt[index+1]) / SIZE;
    I ← (lp_curr[index] - lp_curr[index+1]) / SIZE;
    P ← I * V;
    index ← index + 1;
end while
```

D. High-level MPPT Algorithms

As described in the Motivation section, the core idea of the MPPT algorithm is to determine which direction to adjust the duty cycle to maximize power. The perturb-and-observe algorithm will do this adjustment by a constant amount each time, whereas gradient descent will correct proportionally to the estimated slope of the P-V characteristic in that region.

1) Perturbation and Observation: The pseudocode for the perturb-and-observe algorithm is shown below:

```
V_last ← 0
P_last ← 0
while (running MPPT) do:
    if (P > P_last) then
        if (V > V_last) then
            D ← D - 1; // D = Duty Cycle
        else
            D ← D + 1;
        end if
    else if (P < P_last) then
        if (V > V_last) then
            D ← D - 1;
        else
            D ← D + 1;
        end if
    end if
    V_last ← V;
    P_last ← P;
end while
```

As seen, the algorithm simply uses the directions of power change and voltage change to infer which direction the duty cycle should be pushed towards to approach the maximum of the P-V characteristic.

Note that in order to allow the low-pass filter for voltage and current measurements to stabilize, the algorithm does not run every clock cycle. Instead, it waits a fixed number (10) of INA3221 samples before making another change to the duty cycle, in order to ensure that noise in the power measurements will not affect the decisions made by the algorithm.

2) Gradient Descent: Inspired by the approaches in [10] and [11], the following pseudocode is written to perform the gradient descent version of the MPPT algorithm:

```
D_last ← 0; // D = Duty Cycle
P_last ← 0;
// one of 2 states, STAY or MOVE
gradient_state ← STAY;
while (running MPPT) do:
    if (gradient_state == STAY) then
        D_last ← D;
        P_last ← P;
        D ← D + 1; // slightly perturb to find derivative
        gradient_state ← MOVE;
    else if (gradient_state == MOVE) then
        D += STEP * (P - P_last) / (D - D_last) - 1;
        gradient_state ← STAY;
    end if
end while
```

From empirical testing, a value of STEP (ensuring that the algorithm will adjust the duty cycle when necessary, but not too much) that works well with the algorithm is 1.0. Other similar step sizes will likely converge to the maxima of the power curve similarly, as the algorithm slows down its corrections significantly when the slope is close to 0.

Note that this version of the gradient descent algorithm, while similar in behavior to those in the source material, is very different in terms of its design. Rather than measuring temperature and semiconductor characteristics of the solar cell and inferring the relationships between voltages and duty cycles, this novel student-created variant of the algorithm uses a slight perturbation, every other cycle, to find the derivative of the power, directly with respect to duty cycle (skipping the chain rule evaluation with respect to the voltage altogether).

By transitioning between the STAY and MOVE states repeatedly, this algorithm effectively acts like a rudimentary state machine, cycling between two actions which must be executed in alternating order. Note that to compensate for the 1 added to the duty cycle in the STAY state, a 1 is subtracted from the duty cycle in the MOVE state, to allow for steady-state convergence of the algorithm when the power stops changing.

E. *Environmental Setup for Comparing Algorithms*

Fig. 10: Setup of solar cell (left) and camping light (right)

To simulate a constant source of sunlight so that the two algorithms could be compared in a controlled environment, a camping LED light (Fig. 10) was used for this project. While it had a tendency to get dimmer after staying continuously on for several minutes, it generally kept the same brightness level when powering it off and then on again for each trial.

The solar panel that was tested was an 18V, 225mA rated solar panel, placed parallel to the light source to decrease partial shading and maximize the incident light. To keep conditions as controlled as possible for all trials, the solar panel was never moved between trials and it was left connected to a 470μF capacitor to smoothen out any noise from the light source or surrounding testing environment.

To compare the peformances of the two algorithms, they were started from various initial duty cycles (10%, 20%, 30%, 60%), and their power time-series were measured (3 trials per algorithm per initial duty cycle condition) to determine their steady-state power, transient power, inter-trial consistency, and power conversion efficiency. More detail on these metrics are provided along with the data and results in the next section.

V. Experimental Data, Results, & Analysis

A. *P-V Characteristic*

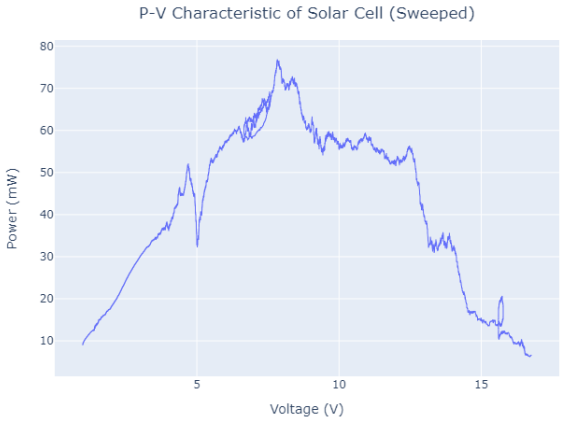

Fig. 11: Power vs Voltage for the solar cell in simulated sunlight

Above (Fig. 11) is a P-V characteristic curve much like the one displayed in Figure 5, obtained by sweeping across all possible duty cycles of the buck converter while the solar panel was illuminated by the camping light. Note the clearly multimodal shape of the distribution, likely due to factors including partial shading and external lighting. Also, note that the sweep done here had a much larger time between samples to avoid low-pass filtering effects, but the time-series plots which will be shown later did use low-pass filtering, so the power values measured at local maxima may not exactly match. However, the peak locations should still match, as the illumination remained the same.

This poses a realistic challenge which is often overlooked in idealized MPPT simulation environments: since MPPT is an iterative algorithm, it may get "stuck" in the neighborhood of a sub-optimal, local maximum of the power, if the curve is flat enough in that region or contains several bumps.

Considering that the true independent variable being controlled by the MPPT algorithms is the duty cycle, it may also be useful to visualize the relationship between the power and the duty cycle of the buck converter (Fig. 12).

Once again, note in Figure 12 the presence of several local maxima of the power in different duty cycle regions. Four initial duty cycle conditions were chosen for testing the MPPT algorithm, to gauge the performance in different scenarios:

- **10%** was chosen as an initial condition, since it is at a lower duty cycle than the global maximum but with several local maxima in between. The goal would be to test whether either algorithm can overcome these local maxima and converge to the much larger maximum at around D = 15%, with $\sim$ 50 mW (Fig. 12).

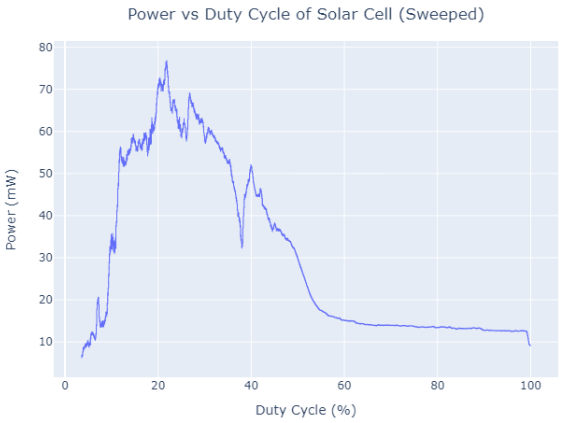

Fig. 12: Power vs Duty Cycle for the solar cell in simulated sunlight

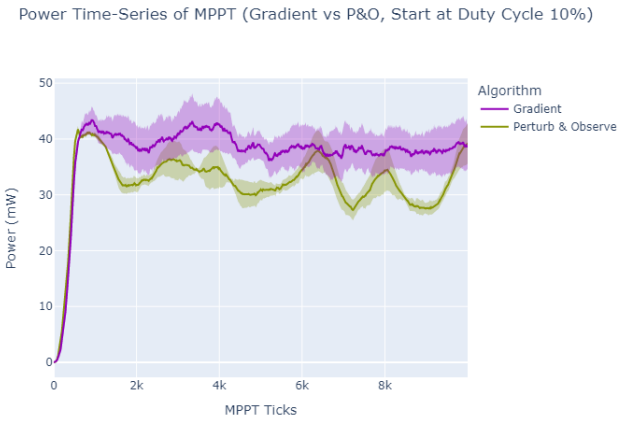

Fig. 13: Compared Gradient and P&O MPPT, starting at D = 10%

- **20%** was also selected, since it is nearby to the global maximum of the P-V characteristic. The goal would be to determine if either algorithm successfully finds this global maximum, or compare how close they get.
- **30%** was selected as an initial condition for the same reason as 10%, to test whether either algorithm would get "stuck" at one of the lower-power maxima or reach higher operating powers of $\sim$ 60 mW.
- **60%** was selected mainly to test the effectiveness of the MPPT algorithms when the power-duty curve is roughly flat. The hypothesis for this region would be that the perturb-and-observe can achieve higher powers due to its non-reliance on the slope of the curve, but the gradient descent algorithm would get stuck at powers below $\sim$ 20 mW as the duty cycle would be adjusted too slowly.

For each of these four categories, graphs of the power vs MPPT ticks (i.e. time units of 0.5 ms, so that 10000 ticks = 5 seconds in real-time) were generated to compare the performances of each MPPT algorithm.

B. *Time-series Charts & Summaries*

*1) **10%**:* Figure 13 shows the (average) performances of the Perturbation & Observation (P&O) and Gradient Descent (Gradient) algorithms when starting at duty cycle 10%. The continuous error bars represent the mean-absolute deviations (MAD, used over standard deviations for inter-trial comparisons due to a small amount of trials) calculated from 3 trials per algorithm (see Appendix to see the individual trial-wise time-series for each initial condition).

Neither algorithm surpasses 50 mW in the steady state, and they have similar steady-state (10k ticks = 5s) performances, but the P&O algorithm is much more erratic and oscillatory, failing to surpass 40 mW most of the time. While the gradient descent algorithm has a more steady average performance, it has much larger error bars, likely due to the gradient algorithm overshooting suboptimal local maxima in some cases (due to the steep slope of the power vs duty cycle curve for low duty cycles, Fig. 12) and failing to in others.

Below are the power summary values based on the 3 trials for each algorithm, starting with duty cycle 10% (Fig. 14):

Power Category	Gradient	P&O
Steady-State Avg (mW)	38.8	39.2
Steady-State MAD (mW)	4.05	3.65
Transient Avg (mW)	39.0	32.8
Transient Std Dev (mW)	1.62	3.06

Fig. 14: Summary of performances, starting at D = 10%

Note that in the above table and also in the similarly formatted tables below, the "Steady-State" Average and MAD refer to the average and mean-absolute deviation of power measured, between the 3 trials, at exactly 10k ticks (5 seconds in real time). On the other hand, the "Transient" statistics measure the means and standard deviations of the trial-averaged time-series over the period of 1k ticks (0.5 seconds, to allow the buck-converter and low-pass power measurements to stabilize initially) to 10k ticks (5 seconds).

*2) **20%**:* In Figure 15, it can be seen that the P&O algorithm has a better average performance near the global maximum, but it is much more oscillatory and erratic in terms of the power level compared to the gradient descent algorithm, likely due to overshooting local maxima at several points.

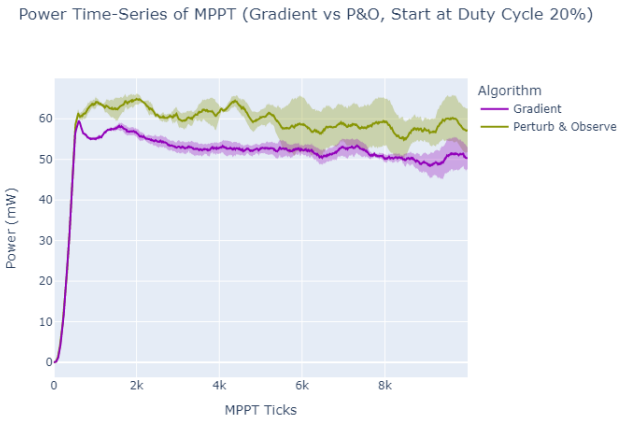

Fig. 15: Compared Gradient and P&O MPPT, starting at D = 20%

Below are the power summary values based on the 3 trials for each algorithm, starting with duty cycle 20% (Fig. 16):

Power Category	Gradient	P&O
Steady-State Avg (mW)	50.2	57.3
Steady-State MAD (mW)	2.38	5.40
Transient Avg (mW)	52.6	59.9
Transient Std Dev (mW)	2.15	2.15

Fig. 16: Summary of performances, starting at D = 20%

*3) **30%**:* In Figure 17, the gradient descent algorithm barely moves the operating point once the power stabilizes, and this steady-state power does not differ much among the trials. Likely, the gradient descent algorithm is failing to find higher local maxima due to operating in the less steep region of the Power vs Duty Cycle curve (Fig. 12) at higher duty cycles.

On the other hand, P&O can much more easily escape some of the lower local maxima of the Power vs Duty Cycle curve (Fig. 12) due to moving more erratically, allowing it to achieve higher (but more varied among trials) steady-state powers.

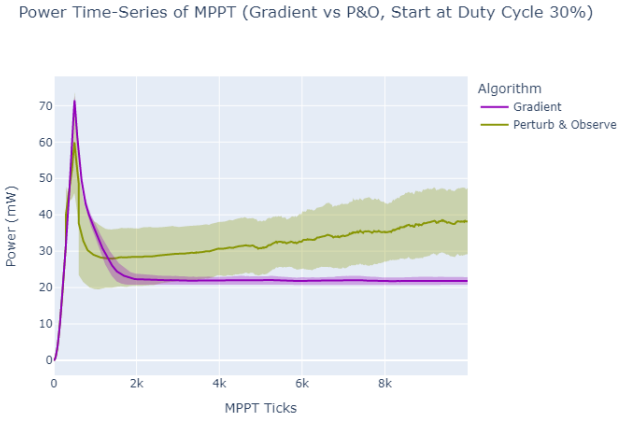

Fig. 17: Compared Gradient and P&O MPPT, starting at D = 30%

Below are the power summary values based on the 3 trials for each algorithm, starting with duty cycle 30% (Fig. 18):

Power Category	Gradient	P&O
Steady-State Avg (mW)	21.8	38.2
Steady-State MAD (mW)	1.05	9.06
Transient Avg (mW)	22.4	32.6
Transient Std Dev (mW)	1.91	3.31

Fig. 18: Summary of performances, starting at D = 30%

*4) **60%**:* Much like with 30%, the gradient descent algorithm barely changes the operating point due to the low steepness of the power relative to duty cycle, while the P&O algorithm successfully finds higher maxima due to its erratic behavior and ability to overshoot local maxima.

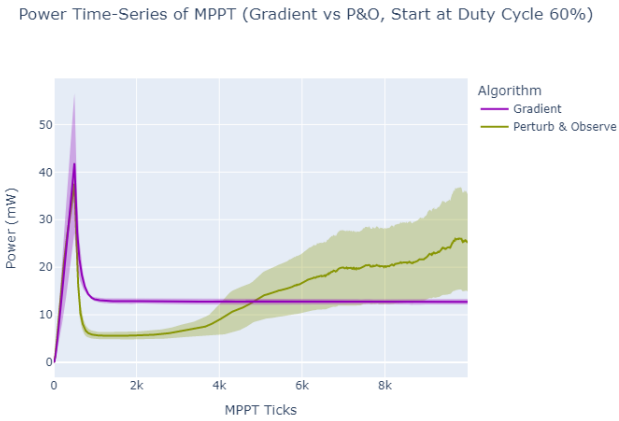

Fig. 19: Compared Gradient and P&O MPPT, starting at D = 60%

Below are the power summary values based on the 3 trials for each algorithm, starting with duty cycle 60% (Fig. 20):

Power Category	Gradient	P&O
Steady-State Avg (mW)	12.53	25.15
Steady-State MAD (mW)	0.56	10.22
Transient Avg (mW)	12.8	14.3
Transient Std Dev (mW)	0.06	6.68

Fig. 20: Summary of performances, starting at D = 60%

From the above data, several metrics can be analyzed and compared, as will be done in the following sub-sections.

C. *Steady-state Average Power*

Steady-state power gives an idea of how well each algorithm would perform in fixed, unchanging environmental conditions once the MPPT algorithm has been able to stabilize. Practically, this can be approximated to be true when there is no

cloud cover or shading and there is a large amount of direct sunlight on the solar cell (for example, in the afternoon).

Given the steady-state average powers tabulated in Figures 14, 16, 18, and 20, the P&O algorithm seems to always perform better or closely to the Gradient algorithm, regardless of initial duty cycle conditions.

A likely reason is that the Perturb and Observe algorithm always changes the duty cycle a fixed amount towards the maximum power, rather than slowing down the adjustments to the duty cycle once local maxima are found. When the derivative of power with respect to the duty cycle is very low, as can be found at local maxima or regions where the power is relatively unchanging (like at higher duty cycles), the Gradient algorithm is susceptible to slowing down completely, without exploring other potentially higher maxima.

Note that before reaching the steady state time (10k ticks, 5 seconds) in Figure 13, the Perturb & Observe algorithm power was much lower on average compared to the Gradient algorithm's. Also, it likely that at a later stopping time, the Gradient algorithm power would stay high but the steady-state P&O time might drop significantly, due to oscillations. In this case, the Gradient algorithm would be better in the steady state when starting at 10% duty cycle.

A likely reason for the gradient descent algorithm performing better at this initial 10% duty cycle is that the Power vs Duty Cycle curve (Fig. 12) is very steep in this region. Since the gradient descent algorithm makes sharp adjustments when the slope is high, this could lead to a large overshoot and then convergence towards a potentially better local maximum of the power. However, this is more serendipitous than replicable when considering P-V characteristics in general.

D. *Transient Average Power*

The transient average power, on the other hand, would be a good measurement of solar cell performance when the environmental conditions rapidly change. Starting at different duty cycles simulates clouds or shading appearing or disappearing, forcing the MPPT algorithm to converge to a new maximum power which may not be the same. As the algorithm adjusts, the transient average power measures the amount of power generated while adjusting to the new operating point.

For duty cycle 10%, the Gradient algorithm has a higher transient average power. This would be because the P&O power curve starting at 10% duty cycle is very oscillatory, forgoing higher power points on average due to overshooting past them. The higher transient standard deviation power of the P&O algorithm in Figure 14 supports this explanation too.

For the other 3 initial conditions, however, the transient average power of P&O is higher. A reason for this could be that the P&O algorithm, on average, is more erratic in terms of duty cycle adjustments. Therefore, by overshooting local maxima, it may be able to achieve higher power operating points. However, this is a more random, uncertain process (note the higher transient standard deviations of P&O on average), and it may not always be true that the P&O algorithm has higher average powers when adjusting from a sub-optimal duty cycle (or, more realistically, when environmental conditions and the P-V characteristic rapidly change).

E. *Inter-trial Consistency (Steady-State MAD)*

As mentioned before, the mean absolute deviation (MAD) was used to show the variation in power time-series between trials, since only 3 trials were done per algorithm per initial condition (not enough for the standard deviation to be directly meaningful). However, the standard deviations of the power curves across trials could also be calculated, and similar error bars would likely be obtained to those in the plots above.

As expected, the more erratic and unpredictable behavior of the Perturb & Observe algorithm makes it more inconsistent between trials, both in the transient and steady-state times. For the three initial duty cycles higher than 10%, the steady-state MAD of the P&O algorithm was significantly higher.

For the initial duty cycle of 10%, as mentioned before, the beneficial overshooting of the gradient descent algorithm when the slope of the Power vs Duty Cycle curve (Fig. 12) is steep could be a reason for increased inconsistency among trials for the gradient descent algorithm. However, even for the 10% duty cycle initial condition, the inter-trial MAD at steady-state for P&O was also high, due to its oscillatory behavior.

F. *Steady-State Power Conversion Efficiency*

While measuring the power across the solar cell (input) is valuable to understand, it may also be useful to consider the **output** power consumed by the 100Ω load, to see how much usable power was delivered to the load at steady state.

To calculate this, a simple multimeter voltage measurement was done across the load resistor at steady state for every trial. Since $P = V_R^2 / R$ for resistors, the output power can be easily calculated and compared for all 3 sets of trials. The average results for each set of trials are summarized below (Fig. 21):

Initial Condition	Gradient	P&O
D = 10%	13.5 mW	9.8 mW
D = 20%	18.3 mW	16.0 mW
D = 30%	12.0 mW	10.7 mW
D = 60%	10.7 mW	8.25 mW

Fig. 21: Summary of output (load) powers, at all initial conditions

Therefore, using the results from Figures 14, 16, 18, and 20, the output power efficiencies can be computed (Fig. 22):

Initial Condition	Gradient	P&O
D = 10%	34.7%	25.2%
D = 20%	36.4%	27.9%
D = 30%	55.0%	28.0%
D = 60%	64.1%	32.8%

Fig. 22: Summary of output (load) efficiencies, at all initial conditions

From Figure 22, two interesting details are worth discussing:

- At higher duty cycles (regardless of the algorithm used), there is a general trend of higher power conversion

efficiency (from the solar cell to the load). This might have to do with a higher buck converter efficiency at higher loads due to higher current draw. However, it may also be a result of fewer parasitic losses in the circuit when the power input from the solar cell is low. Further testing would need to be done to know for sure.

- Overall, the power conversion efficiencies of the gradient descent algorithm are significantly higher than the P&O algorithm. However, this is not as impressive when considering that the output (load) powers themselves were roughly the same, but that the lower input (solar cell) powers of the gradient descent algorithm were much lower. As suggested in the previous remark, this may have to do with parasitic losses in the circuit decreasing the conversion efficiency of higher solar input powers, but more experimentation would need to be done to confirm.

VI. Discussion & Limitations

A. *Accomplishments*

This project successfully designed, built, and tested a full closed-loop MPPT system, with INA sensors, a buck converter as an actuator, and two MPPT algorithms which were tested against each other using various metrics, including steady-state performance, transient performance, inter-trial consistency, and power conversion efficiency.

The buck converter was student-designed, implementing a charge-pump bootstrap circuit using a gate driver, to open and close a logic-level MOSFET, which would itself drive a breadboard-based buck converter based on a signal generated by the Teensy. Firmware was written to ensure periodic INA3221 sensor sampling and accurate PWM adjustments in response to both manual potentiometer inputs and algorithmic commands to change the duty cycle.

A student-designed small-perturbation version of the popular gradient descent algorithm was implemented and optimized, and was compared against a version of the Perturb & Observe algorithm from the surrounding literature. This project identified the strengths and weaknesses of each algorithm through a practical test with various initial conditions.

Overall, it was found that the Perturb & Observe algorithm had a higher steady-state average performance and had a higher transient average performance as well, in most cases. In low duty cycle regions where the Power vs Duty Cycle curve (Fig. 12) was steep, however, the Gradient algorithm's high overshoot may have helped reach higher power operating points. In terms of the consistencies of the algorithms over 3 trials, the Gradient algorithm was shown to have less overall transient variation and more consistency between trials, due to converging more smoothly to local maxima of power. The measured power conversion efficiencies of the Gradient algorithm were found to be higher than those of the P&O algorithm, likely due to settling on higher duty cycles, which may be efficient in terms of buck converter power conversion. Further testing would be necessary to rule out whether the effect was simply due to parasitic losses from the breadboard and other reactive circuit elements dropping the power efficiencies when the measured solar cell power was higher.

B. *Limitations & Future Work*

Due to time constraints, a breadboard was used for the design and implementation of the buck converter circuit. This may have affected the power conversion efficiency of the MPPT system due to parasitic losses, and stray capacitance in the board could have slightly affected the high-frequency output PWM waveform before it was used to drive the buck converter. Precautions (like gate resistors and decoupling capacitors) to taken to help decrease these effects, but they were not preventable. Given more time, it would be beneficial to design and print a PCB for testing this circuit with the two algorithms in question, to determine whether the buck converter and MPPT algorithms would perform better.

Also, as a light source, a high-brightness camping light was used to test the MPPT algorithms, but external lighting in the room may have slightly affected the results of the experiment by slightly changing the power. Also, the power reading was not calibrated to the brightness of the camping light, so it is possible that slight variations in the camping light illumination (though these effects were ameliorated by using a bypass capacitor across the solar cell's terminals) may have affected the power measurements from the solar cell for the MPPT algorithm. Testing in a more controlled, fixed-illuminance simulated sunlight environment may be useful to confirm the results from this project.

Finally, an initial plan in this project was to develop and test a reinforcement learning (RL) approach for MPPT, to determine whether it would be able to successfully predict and overcome sub-optimal local maxima of the P-V characteristic. However, due to time constraints, this test could not be performed. A good follow-up study would be to develop an RL-based approach for MPPT, and compare its MPP-tracking ability against the algorithms tested in this project.

VII. Conclusion

Overall, this project has created a closed-loop cyberphysical system which is capable of regulating the load a solar panel to achieve a locally maximum power operating point, even under difficult partial shading conditions. Two algorithms which were present in the literature but only tested against each other via simulation, P&O and Gradient, have been written and compared over several trials during the course of this research, by focusing on steady-state and transient performance, as well as inter-trial consistency and power conversion efficiency. The weaknesses and strengths of each algorithm have been evaluated and discussed, with the hope of shedding light on the practicality of each in real-world solar MPPT settings.

VIII. References

[1] Benghanem, M. & Alamri, S. N.. (2009). Modeling of photovoltaic module and experimental determination of serial resistance. *Journal of Taibah University for Science*. 2. 94-105. doi: 10.1016/S1658-3655(12)60012-0.

[2] El-Maksood, A.. (2018). Performance dependence of (I-V) and (C-V) for solar cells on environmental conditions. *Journal of Advances in Physics*. 14. doi: 5331-5351. 10.24297/jap.v14i1.7315.

[3] Li, G. & Xuan, Q. & Pei, G. & Su, Y. & Ji, J.. (2017). Effect of non-uniform illumination and temperature distribution on concentrating solar cell - A review. *Energy*. 144. doi: 10.1016/j.energy.2017.12.067.

[4] Abid, R. & Masmoudi, F. & Ben Salem, F. & Derbel, N.. (2016). Modeling and simulation of conventional DC-DC converters deticated to photovoltaic applications. *International Renewable Energy Congress*. 1-6. doi: 10.1109/IREC.2016.7478937.

[5] Peng, C. & Wang, C.. (2013). An analysis of buck converter efficiency in PWM/PFM mode with Simulink. *Energy and Power Engineering*. 05. 64-69. 10.4236/epe.2013.53B013.

[6] Thangamuthu, L. & SenthilKumar, A.. (2014). A review of maximum power point tracking algorithms for photovoltaic systems under uniform and non-uniform irradiances. *Energy Procedia*. 54. 228–235. doi: 10.1016/j.egypro.2014.07.266.

[7] Bollipo, R. & Mikkili, S. & Bonthagorla, P.. (2020). A critical review on PV MPPT Techniques: classical, intelligent and optimization. *IET Renewable Power Generation*. 14. doi: 10.1049/iet-rpg.2019.1163.

[8] Pakkiraiah, B. & Sukumar, G.. (2016). Research survey on various MPPT performance issues to improve the solar PV system efficiency. *Journal of Solar Energy*. 2016. 1-20. doi: 10.1155/2016/8012432.

[9] Salman, S. & Ai, X. & Wu, Z.. (2018). Design of a P-&-O algorithm based MPPT charge controller for a stand-alone 200W PV system. *Protection and Control of Modern Power Systems*. 3. doi: 10.1186/s41601-018-0099-8.

[10] Zhang, J., Wang, T., & Ran, H. (2009). A maximum power point tracking algorithm based on gradient descent method. *IEEE Power & Energy Society General Meeting, Calgary, AB, Canada, 2009*. 1-5. doi: 10.1109/PES.2009.5275737.

[11] Xiao, W. & Dunford, W. & Palmer, P. & Capel, A.. (2007). Application of centered differentiation and steepest descent to maximum power point tracking. *IEEE Transactions on Industrial Electronics*. 54. 2539 - 2549. doi: 10.1109/TIE.2007.899922.

[12] Ma, J. & Man, K. & Ting, T. & Lee, H. & Jeong, T. & Sean, J. & Guan, S. & Wong, P.. (2012). Insight of direct search methods and module-integrated algorithms for maximum power point tracking (MPPT) of stand-alone photovoltaic systems. *IFIP International Conference on Network and Parallel Computing*. 7513. 463-471. doi: 10.1007/978-3-642-35606-3_55.

[13] Baba, A. & Liu, G. & Chen, X.. (2020). Classification and evaluation review of maximum power point tracking methods. *Sustainable Futures*. 2. 100020. doi: 10.1016/j.sftr.2020.100020.

[14] Baharudin, Nor & Tunku Mansur, Tunku Muhammad Nizar & Hamid, F & Ali, Rosnazri & Irwanto, Muhammad. (2018). Performance Analysis of DC-DC Buck Converter for Renewable Energy Application. Journal of Physics: Conference Series. 1019. 012020. 10.1088/1742-6596/1019/1/012020.

[15] Shatla, H. & Said, E. & Abd Elwhab, A. (2022). Performance of MOSFET driven via a bootstrap capacitor for dynamic load continuity enhancement. *Hindawi Journal of Engineeering*. doi: 10.1155/2022/2273819

IX. Appendix (Detailed Time-Series Charts)

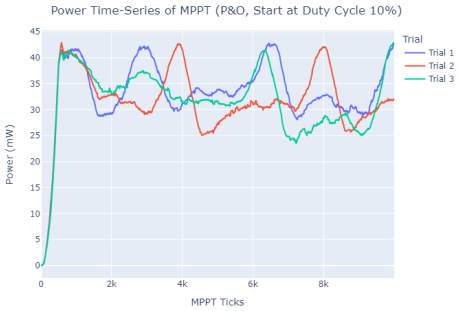

Fig. 23: Performance of P&O MPPT, starting at D = 10%

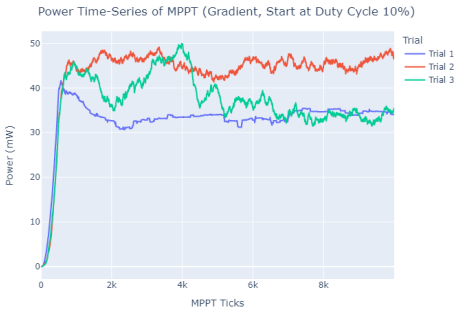

Fig. 24: Performance of Gradient MPPT, starting at D = 10%

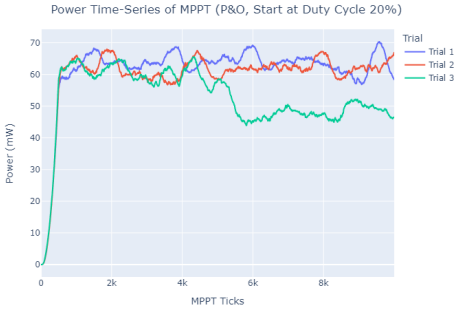

Fig. 25: Performance of P&O MPPT, starting at D = 20%

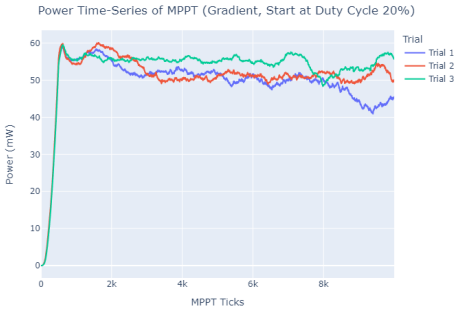

Fig. 26: Performance of Gradient MPPT, starting at D = 20%

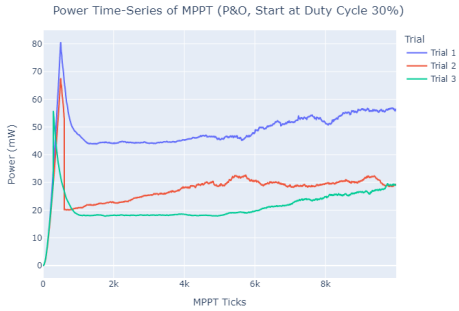

Fig. 27: Performance of P&O MPPT, starting at D = 30%

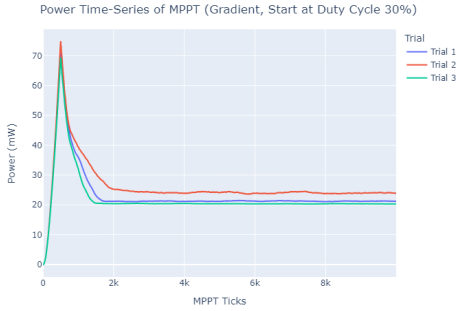

Fig. 28: Performance of Gradient MPPT, starting at D = 30%

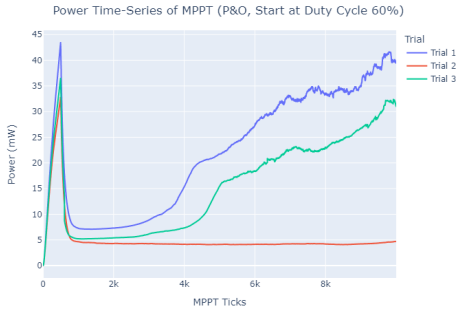

Fig. 29: Performance of P&O MPPT, starting at D = 60%

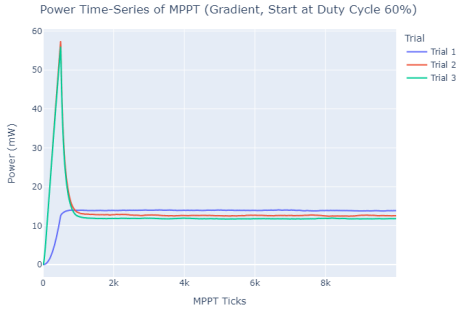

Fig. 30: Performance of Gradient MPPT, starting at D = 60%