Solar powered weather station

Power management

Tibbe van der Biezen and Maarten van der Meulen

Group 2

TU Delft

Solar powered weather station

Power management

by

Tibbe van der Biezen and Maarten van der Meulen

to obtain the degree of Bachelor of Science

Supervised by Dr. Patrizio Manganiello and Dr. Mirco Muttillo
Proposed by Dr. Patrizio Manganiello
Department of Electrical Sustainable Energy

Defended before Patrizio Manganiello, Mirco Muttillo, Mohamad Ghaffarian Niasar and Marco Antonio Zúñiga Zamalloa

Bachelor Graduation Thesis
June 18, 2021

Delft University of Technology

Faculty of Electrical Engineering, Mathematics and Computer Science

Electrical Engineering Programme

Abstract

Due to the current trend of urbanization more people will come and live in cities than ever before [1]. This increases the need to monitor the urban environment, to be able to improve the living conditions of the urban dwellers. Our project combines the need to monitor the local environment, with a portable, self-powered and wireless weather station. The final design consists of a solar powered and WiFi-connected weather station with autonomous functionality for one year. For our research, the focus was on the Netherlands. Part of such a project is the power management which will be dealt with in this thesis.
In the thesis, different maximum power point tracking algorithms will be discussed and compared. Furthermore, some more research is done in the hardware design, the use of different evaluation boards and battery configurations. Finally, a prototype using the incremental conductance algorithm is constructed and tested. Simulations lead to an efficiency of around 80%, which means autonomous functionality for a minimum of one year. The physical system had a lower efficiency, but autonomous functionality for one year was still achieved.

Preface

Before you lies the thesis ”Solar powered weather station, power management”, which is based on research into maximum power point tracking for IoT weather stations. It has been written to graduate the electrical engineering bachelor at the TU delft. The project was proposed by Patrizio Manganiello. The research was fun and interesting, but the COVID-19 pandemic posed some interesting challenges for the physical prototype and the available research avenues.

We would like to thank first and foremost our proposer and supervisor Patrizio Manganiello for his expert feedback and support. We also want to thank Mirco Muttillo for his feedback and help during the days we were able to work in the Solar Lab. Lastly we thank Ton Slats and Martin Schumacher of the Tellegen Hal for their help in finding components, and their support in the lab.

Contents

1 Introduction

1.1. System overview

The goal of this project is to make an autonomous weather station which only runs on solar energy. A large number of these stations are to be placed in and around urban areas where they will gather data about humidity, temperature, noise and particulate matter. This information will be uploaded via Wi-Fi to a database where it can be analysed. The results may be used in traffic planning, city-planning or to make some weather predictions. In order to make it feasible to have a lot of these stations, the stations themselves should be as small as possible, cost efficient and maintenance free for at least one year.

1.1.1. Structure

The project is divided into three groups, each focusing on a particular part of the station (summarised in figure 1.1).
The first group is responsible for the power generation of the Photovoltaic (PV) array. An additional task is the design of the housing and support structure of the station, as well as a mounting system to allow it to be attached to an object like a light pole.
The second group will focus on the part which connects the output of the PV-array to the micro-controller. The main subject is maximum power point tracking (MPPT), to get the maximum power transfer between the panel and the controller. This group is also responsible for the power management when it comes to battery charging and discharging.
The final group takes care of the choice and implementation of the sensors. Based on the power generation of the solar panels and consumption of the system, an algorithm is made to optimize the frequency of measurement. After collecting the data, they make sure it is periodically uploaded to a database via Wi-Fi.

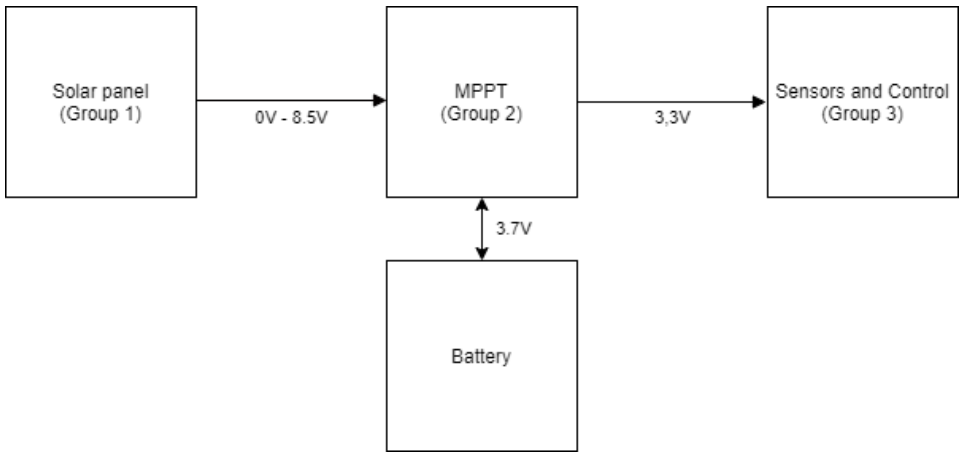


Figure 1.1: System overview with sub-group division.

1.2. Problem definition

Scoping analysis

In modern cities both pollution and the local climate have a large effect on its inhabitants. The urban heat island effect can decrease comfort and pollution costs lives. If one wants to be able to influence either of these factors, it is necessary to first be able to measure these factors. The weather station needs to collect data on these environmental conditions to facilitate decision making. The weather station needs to function autonomously, off the grid for at least one year, and be able to communicate the data over the air to a server. The MPPT system that is to be designed must be able to charge a battery so that the weather station can operate during the night and during low irradiance days. During the winter, the system needs to keep functioning even with consecutive low solar power days.

Bounding analysis

Due to the COVID-19 pandemic, the available time in the lab was limited, therefore the decision was made that instead of designing an entire MPPT system from scratch, an existing of the shelf board would be bought and where necessary improved. Besides the limits imposed by the pandemic, it is also important to keep in mind the MPPT system should work in the overall weather station. The solar panel imposes boundaries on the input range of the voltage and the current, and the Arduino microcontroller for measuring the environmental data imposes constraints on the output voltage and current range of the MPPT system.

1.3. State of art analysis

In this section the reader will get a short overview on the available literature on both the fundamentals of MPPT algorithms, and currently available technology solutions for MPPT on the market. MPPT is a method to achieve a higher power transfer between a PV panel and a load. Figure 1.2 shows both the current-voltage (I-V) curve and the power-voltage (P-V) curve of a PV panel. There is one point on this curve where the peak power is generated, the maximum power point (MPP). The MPP shifts under different irradiation and temperature conditions. The MPPT tries to find the MPP under all different conditions to make sure maximum power is extracted from the PV generator.

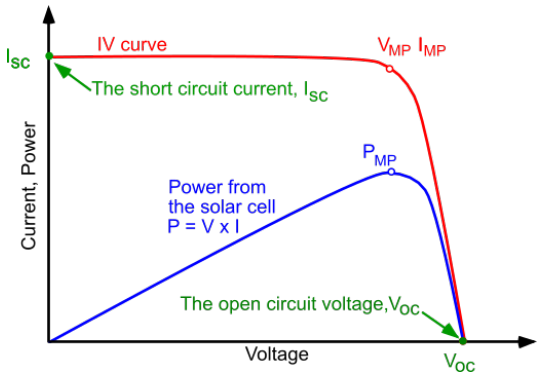


Figure 1.2: I-V curve of a solar panel [2]

A MPPT system is implemented using a dc/dc converter together with a MPPT controller. The controller measures the current and voltage, and calculates the power. The controller runs the MPPT algorithm and controls the duty cycle of dc/dc converter, using a control signal. The control signal is often the PWM of the converter itself.

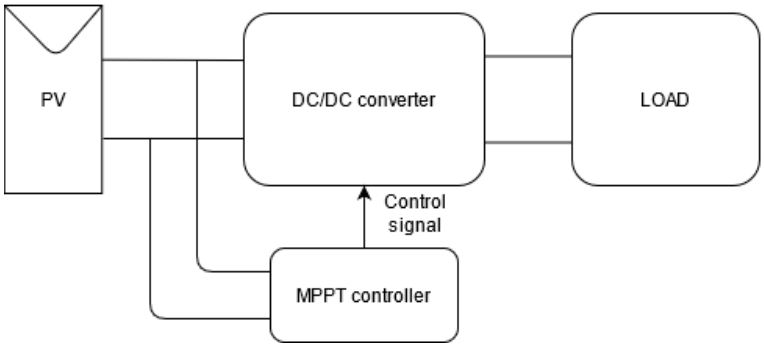


Figure 1.3: MPPT system with DC/DC converter

1.3.1. DC/DC converters

From chapter 1.3.2 it is clear that almost all algorithms make use of the variable duty-cycle of the dc/dc-converters. This section will therefore highlight some of the characteristics of dc/dc-converters. First of all, there are four commonly used converters, see figure 1.5. Buck, to lower the voltage and increase the current, boost, which does the opposite and finally the buck-boost and Ćuk, which can do both. The functionality between these converters differs, however the principal is the same everywhere. Input power is stored in an inductor (buck, boost and buck-boost) or capacitor (Ćuk) and then released to the output. Via a switch the time difference (duty-cycle) between charging and releasing can be set. The duty-cycle is the amount of time that a signal is on within a single period.

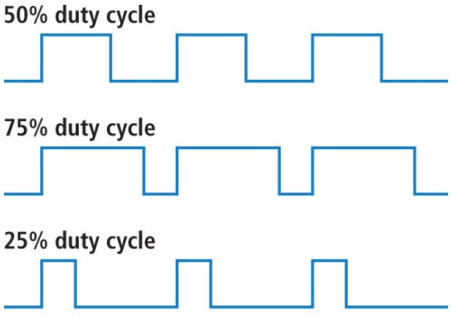


Figure 1.4: Duty cycle

This directly translates to the amount of voltage drop or boost at the output. However, in our case the output voltage is defined by the battery and therefore a varying duty-cycle will not effect output voltage but the input voltage instead. Meaning a MPPT algorithm can be implemented. Depending on the input values and the requirements for the load, in our case the battery, the correct converter type is chosen.

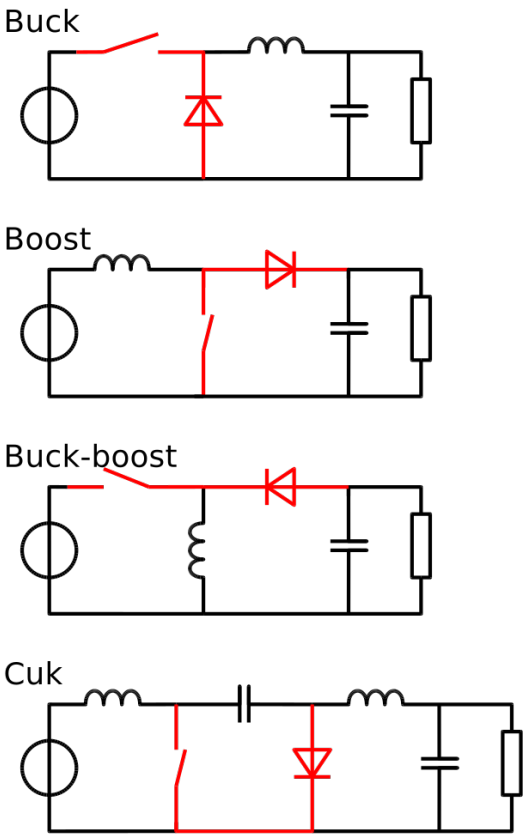


Figure 1.5: Basic converter types

Buck:

$$\frac{V_o}{V_{in}} = D \tag{1.1}$$

Boost:

$$\frac{V_o}{V_{in}} = \frac{1}{1-D} \tag{1.2}$$

Buck-Boost:

$$\frac{V_o}{V_{in}} = \frac{-D}{1-D} \tag{1.3}$$

Cuk:

$$\frac{V_o}{V_{in}} = \frac{D}{1-D} \tag{1.4}$$

When dealing with ultra low voltage inputs, a standard boost converter is not enough. To charge a Li-ion battery the minimum voltage is often around 4.2V [3]. In [4] an converter is build which can amplify the input-voltage by a factor of nine. Other methods involve different circuits for different operating modes [5], or the more general approach of using charge pumps. When designing a converter whose input voltage ranges from below the battery charging voltage to above this voltage, a buck-boost converter is needed. After all, the battery needs a constant charging voltage. Again, there are numerous implementations of which the system described in [6] is one of them. By taking care of the polarity change and minimizing the amount of components an efficient converter is designed that operates in a wide range.

1.3.2. Maximum power point tracking algorithms

The papers [7–9] give a good overview of the existing MPPT algorithms. They identify 17 different algorithms, not all of these will be discussed for the sake of brevity. For instance the techniques relating to inverters for a grid connection or techniques that make use of array reconfiguration are irrelevant for the scope of the project. Finally, there are a lot more techniques but often it is found that they are modifications of the ones mentioned below. A distinction can be made between true or full MPPT algorithms and false MPPT algorithms [10]. A true MPPT algorithm can track the MPP over the entire I-V curve of a solar panel. A false MPPT algorithm does not track the entire curve and therefore can get stuck in a local 'false' MPP. A different approach is, what Analog Devices calls maximum power point control (MPPC) [10]. These algorithms find an operating point close to the actual MPP, but do not fully track it. An example of MPPC is for instance the fractional open-circuit voltage algorithm.

Hill climbing and perturb & observe

Hill climbing and perturb & observe (P&O) are algorithms where the MPP is found by changing the duty cycle of the converter or the voltage respectively, then measuring the output power of the PV array or at

the battery. The logic of the P&O algorithm is shown as a block diagram in figure 1.8. Both algorithms try to do the same thing, that is to change the input voltage of the converter and measure the change in output power. When you are on the left of the MPP you try to move to the right and vice versa. This process is continued until the MPP is reached. After that the algorithm dithers around the MPP.

Figure 1.6 shows how the P&O algorithm would move across the power curve.

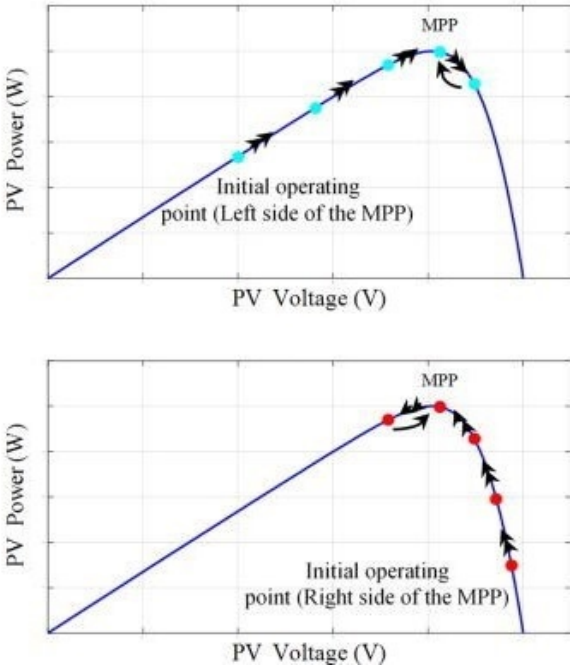


Figure 1.6: Power voltage characteristics of the PV operating points using the P&O algorithm [11]

The amount of voltage change at the input is also called the step size. When the step size is small, the algorithm will be relatively slow, when it arrives at the MPP the oscillations will be at a minimum. A large step size will give opposite results namely a fast algorithm with high oscillations at MPP. Some versions of these two algorithms use variable step sizes to increase the tracking speed of the algorithm, as described in [12, 13]. Some of the disadvantages of P&O and hill climbing is their inability to deal with a rapidly changing MPP, and partial shading. Partial shading occurs when a part of the solar panel is in the shade. The panel will have one or more local optima and one global optimum, as shown in figure 1.7. This can lead to a case where the algorithm gets stuck in a local optimum, and does not supply maximum power to the load.

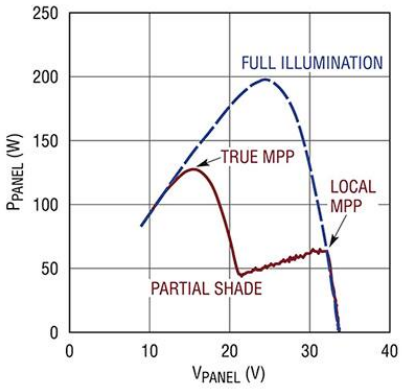


Figure 1.7: Local vs global MPP under partial shading conditions

In [14] a solution is proposed for the shading problem by changing the step size dynamically, and thereby increasing the tracking efficiency. Another possible solution is proposed in [15], where an extra

checking step is implemented to deal with partial shading and improve stability of the output power. In paper [16] P&O is compared to a lookup table based approach. The lookup table compares temperature and irradiance measurements to find a pre-calculated MPP current, and fixes on that point. The performance is slightly better, but their P&O implementation itself is poor.

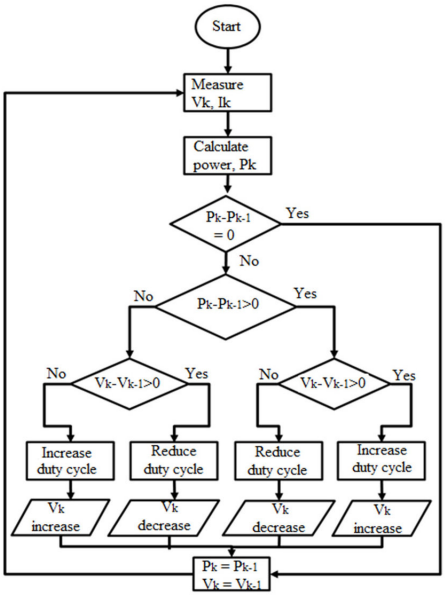


Figure 1.8: P&O algorithm

Incremental conductance

The incremental conductance algorithm uses the fact that the slope of the PV power curve (shown in figure 1.2) is zero at the MPP, negative on the right of the MPP and positive on the left of the MPP.

$$\begin{array}{ll} \text{dP/dV} = 0 & \text{at MPP} \\ \text{dP/dV} > 0 & \text{left of MPP} \\ \text{dP/dV} < 0 & \text{rigth of MPP} \end{array} \tag{1.5}$$

Since

$$\frac{dP}{dV} = \frac{d(IV)}{dV} = I + V\frac{dI}{dV} \cong I + V\frac{\Delta I}{\Delta V} \tag{1.6}$$

Equation 1.5 can be rewritten as:

$$\begin{array}{ll} \Delta I/\Delta V = \text{-I/V} & \text{at MPP} \\ \Delta I/\Delta V > \text{-I/V} & \text{left of MPP} \\ \Delta I/\Delta V < \text{-I/V} & \text{rigth of MPP} \end{array} \tag{1.7}$$

The tracking speed is determined by the step size, meaning the amount by which the PV voltage is changed to get to the new operating point. When the MPP is reached, it is maintained until ΔI changes. In the paper by S. Khadidja, M. Mountassar, and B. M'hamed, incremental conductance is compared with P&O [17]. They showed that the incremental conductance algorithm has a better performance, because the oscillations around the MPP where reduced therefore minimizing the power loss.

Fractional open-circuit voltage

The relation between the V_{OC} and V_{MPP} is nearly linear (1.8). To utilize this relation the fractional open voltage method is devised. The method works by periodically measuring the V_{OC} and then calculating V_{MPP}.

$$V_{MPP} \approx k_1 V_{OC} \tag{1.8}$$

k_1 is dependant on the characteristics of the solar panel, and has to be computed in advance. This is done by measuring the panel under different irradiance and temperature levels. The main advantage

of this method is the low complexity and relatively low cost. It is however not true MPPT since the maximum power point is only found by approximation. In case of partial shading, the value found for k_1 also becomes invalid. The V_{OC} can be determined in two ways. The first being temporarily disconnecting the converter from the panel and measuring the voltage. This however imposes some drawbacks, the main one being the loss of power during the disconnection. The second option is to use or add some cells which represent the PV array. By measuring the voltage over them there is no more power loss. It does however adds more uncertainty, adds complexity and needs extra space.

Fractional short-circuit current

This algorithm is, like Fractional open-circuit voltage, based on the fact that I_{MPP} behaves almost linear to I_{SC}. Equation (1.9) shows this relation where k_2 is determined by the characteristics of the PV-array.

$$I_{MPP} \approx k_2 I_{SC} \tag{1.9}$$

The short-circuit current I_{SC}, is usually measured by adding a switch to short the PV array. During this time, there is no power going to the system and therefore the efficiency decreases. Like the fractional open-circuit voltage algorithm this also isn't true MPPT. The operating point will never match the MPP perfectly.

Fuzzy logic control

Fuzzy logic for MPPT was first described in [18]. Fuzzy logic consists of three stages: fuzzification, rule based lookup and defuzzification. During fuzzification the input values are converted to linguistic variables based on a membership function. In the paper, 5 fuzzy logic levels were used and 3 membership functions.

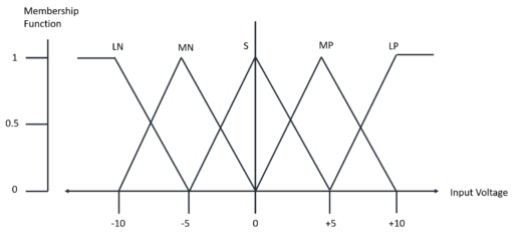


Figure 1.9: A generic membership function

Fuzzy logic uses as input E and ΔE. In the paper these are defined as follows.

$$E(k) = \frac{P(k) - P(k-1)}{I(k) - I(k-1)} \tag{1.10a}$$

$$\Delta E = E(k) - E(k-1) \tag{1.10b}$$

The fuzzyfied variables are used in a lookup table 1.1. With the help of another membership function the output of the lookup table is defuzzified. This deffuzified signal is used to control the MPPT tracker.

Table 1.1: Lookup table

E\CE	NB	NS	ZO	PS	PB
NB	ZO	ZO	NB	NB	NB
NS	ZO	ZO	NS	NS	NS
ZO	NS	ZO	ZO	ZO	PS
PS	PS	PS	PS	ZO	ZO
PB	PB	PB	PB	ZO	ZO

Fuzzy logic has the advantage that it can deal with non-linearity and abrupt changes in irradiance and temperature. The disadvantage is however that it is difficult to construct an efficient, well working logic control.

Neural network

Another MPPT technique is to use a neural network [19]. The neural network is trained on the specific solar cell, and can take into account all kinds of data, such as: irradiance, temperature, V_{in} and I_{in} of the converter. The neural network then manages the duty cycle that drives the converter. Because over time the characteristics of the PV array may change, the model has to be retrained periodically to keep a high performance.

RCC

Ripple current control (RCC) is based around the current and voltage ripple that arises when the dc/dc-converter switches. By comparing the time derivative of the power $\dot{p}$ to that of the current $\dot{i}$ or voltage $\dot{v}$, the MPP can be found. In other words, let $\dot{i} > 0$ or $\dot{v} > 0$ then, when $\dot{p} > 0$ the operating point is below the MPP and when $\dot{p} < 0$ it is above the MPP. Like most of the algorithms, RCC enforces the MPP by changing the duty cycle. The equations (1.11) and (1.12) describe this control mechanism.

$$d(t) = -k_3 \int \dot{p}\dot{v}dt \tag{1.11}$$

$$d(t) = k_3 \int \dot{p}\dot{i}dt \tag{1.12}$$

In both equations, k_3 is a positive constant. To implement these equations, specifically the time derivative, multiple options are available like using ac-coupled measurements to circumvent the derivatives all together, high-pass filters or measurements of the inductor voltage. In [20] RCC is compared with P&O during a sudden variation of irradiance. Their measurements showed that RCC had a lower settling time, and greater tracking speed compared to P&O. The ripple current at MPP was almost the same. RCC is completely analog which makes it is cheap to implement but also more difficult to adapt and more time consuming to design.

Current sweep

The goal of the current sweep is to obtain the I-V-characteristic of the PV-array. Once this is known the V_{MPP} can then be calculated. The following arithmatic is based on the paper by T. Esram and P. L. Chapman [7]. The sweep-waveform is defined by the function $f(t)$ where $f(t)$ is chosen in such a way that its directly proportional to its derivative, as shown in equation (1.13).

$$f(t) = k_4 \frac{df(t)}{dt} \tag{1.13}$$

where k_4 is a proportionality constant. During the current sweep the power of the PV array is then given by equation (1.14).

$$p(t) = v(t)i(t) = v(t)f(t) \tag{1.14}$$

At MPP it is known that $\frac{dp(t)}{dt} = 0$. Combining equations (1.13) and (1.14) gives the following equation.

$$\frac{dp(t)}{dt} = v(t)\frac{df(t)}{dt} + f(t)\frac{dv(t)}{dt} = \left[v(t) + k_v \frac{dv(t)}{dt}\right]\frac{df(t)}{dt} = 0 \tag{1.15}$$

Solutions for $f(t)$ given equation (1.13) are of the form:

$$f(t) = Ce^{\frac{t}{k_4}} \tag{1.16}$$

where C is chosen such that it is equal to I_{max} of the PV array and k_4 is negative, giving a decreasing exponential function with $\tau = -k_4$. By looking at equation (1.16) it can be seen that $\frac{df(t)}{dt}$ is nonzero, meaning that equation (1.15) can be simplified by applying simple division. Note that $f(t) = i(t)$ still holds.

$$\frac{dp(t)}{di(t)} = v(t) + k_4 \frac{dv(t)}{dt} = 0 \tag{1.17}$$

To summarize, with the current sweep the I-V-characteristic is found. This results in the V_{MMP}. To check whether this is actually the MPP, equation (1.17) can be used. In [21] an algorithm based on these principles is proposed. The MPPT uses analog electronics to find the MPP over a period of 50 ms, and then waits for five minutes before scanning again. During the scan the power output is decreased. The solution is only feasible if the extra power gain is higher then the consumption of the device.

Load current or voltage maximization

Like other MPPTs, load current or voltage maximization also aims for maximum power. However for this algorithm it is assumed that a maximum power output of the dc/dc-converter is the result of the maximum power output of the PV array. In other words, the converter is lossless. Achieving maximum power at the load therefore means operating at the MPP of the PV array. For this case, the load can be modelled in four different ways. As a current source, a voltage source, a resistive load and as any combination of those three. In [22] it is shown that for a current source type load, maximizing the voltage results in the MPP and for the voltage source type load maximizing the current will give the MPP. As for the other types, either one will do. Resulting in the need for only one sensor.

dP/dV or *dP/dI* feedback control

Most of the MPPT-algorithms are based around finding dP/dV or dP/dI. This algorithm does exactly that by using digital signal processing and a microcontroller. Over the years, many methods are developed to calculate the slope. A positive or negative slope indicates that the operating point is respectively left or right of the MPP. By adjusting the duty-cycle of the dc/dc-converter the operating point can then be moved to the MPP [7].

Comparison of the MPPT-algorithms

In table 1.2 the above discussed algorithms are evaluated. Important to note is that there are much more algorithms in use. For example algorithms which make use of the temperature or irradiance to calculate the MPP [23, 24]. The columns shown in the table indicate some of the important aspects to take into account when selecting a MPPT. Others like cost, efficiency or the ability to deal with multiple local maxima are heavily dependent on particular implementations and are thus not mentioned.

Table 1.2: Overview of the discussed algorithms based on the paper by T. Esram and P. L. Chapman [7]

MPPT Algorithm	PV Array Dependent	True MPPT	Analog or Digital	Periodic Tuning	Convergence Speed	Implementation Complexity	Sensed Parameters
Hill-climbing/P&O	No	Yes	Both	No	Varies	Low	Voltage, Current
Incremental conductance	No	Yes	Digital	No	Varies	Medium	Voltage, Current
Fractional V_{OC}	Yes	No	Both	Yes	Medium	Low	Voltage
Fractional I_{SC}	Yes	No	Both	Yes	Medium	Medium	Current
Fuzzy logic control	Yes	Yes	Digital	Yes	Fast	High	Varies
Neural network	Yes	Yes	Digital	Yes	Fast	High	Vaires
RCC	No	Yes	Analog	No	Fast	Low	Voltage, Curent
Current sweep	Yes	Yes	Digital	Yes	Slow	High	Voltage, Current
Load I or V maximization	No	No	Analog	No	Fast	Low	Voltage, Current
dP/dV or dP/dI feedback control	No	Yes	Digital	No	Fast	Medium	Voltage, Current

1.3.3. Maximum power point tracking solutions

In this section a short overview is given of the available technology implementations from major electronics companies. The focus here lies on development boards, not so much on the individual chips, because the time frame does not allow for the implementation of a chip. In looking for a good board there are some things to keep in mind, such as but not limited to: ”the MPPT algorithm, voltage operating range, charge current and battery temperature sense capabilities.”

Sunny buddy/DC1568A

The Sunny buddy is based on the LT3542 by Analog Devices [25]. It uses a feedback loop to fix the voltage of the solar panel to the MPP. The off-the-shelf evaluation board does not implement true MPPT. However, the voltage reference pin can be used to implement our own MPPT algorithm.

BQ25798EVM

This evaluation board is based the BQ25798 chip from Texas Instruments [26]. It is an integrated buck-boost converter. The chip uses fractional open voltage for its MPPT control. And there is no way to modify the behaviour of the algorithm.

DC2038A-j

The DC2038A-j is based on the LTC4162 by Analog Devices [27]. It is a buck converter. The chip uses a current sweep for its MPPT algorithm. The board itself is equipped with I^2C to interface with other devices. The MPPT cannot be modified.

STEVAL-ISV012V1

The STEVAL-ISV012V1 is based on the SPV1040 by STMicroelectronics [28]. The SPV1040 is a low input voltage boost converter. The chip uses an proprietary P&O algorithm, and can charge one lithium battery cell. It has no battery temperature control, and the behaviour of the MPPT algorithm cannot be modified.

Comparison table

In table 1.3 the most important characteristics of the different boards are summarized.

Table 1.3: Comparison table of different boards

Board	Vin	MPPT	Bat_{temp}*	topology
BQ25798EVM	5-26	Fractional V_{oc}	yes	Buck-Boost
Sunny buddy	4.95-20	Constant voltage	yes	Buck
DC2038A-j	4.5-35	Current sweep	yes	Buck
STEVAL-ISV012V1	0.3-4.2	P&O	no	Boost

* Bat_{temp} means the availability to measure the temperature of the battery

1.3.4. Batteries and power management

In this section some remarks about Li-ion batteries are discussed and a short overview of power path is given. First of all, the batteries themselves have certain specifications which need to be met [3]. The most important ones being the required charge voltage and maximum charge current the battery can handle. However, to extend the lifetime of the battery, load matching and the charging curve are also important to look at [29, 30]. Power management makes sure the battery is charged and discharged in the most optimal way [31]. When this is used in combination with power-path it also makes sure that the battery can be used to power the load when the power generation is not enough [32].

2

Programme of requirements

2.1. Mandatory requirements

Functional requirements:

- The system must measure at an interval of at least one hour.
- The system must transmit the measured data at least once a week.
- The system must send its measured data over Wi-Fi.
- The system must measure:
  - Temperature
  - Humidity
  - Air pressure
  - PM2.5/PM5/PM10 pollution

Non-functional requirements:

- The system must function in an urban environment for at least 1 year.
- The only external power source must be PV.
- The maximum size of the box must be 250x250x300mm.

2.2. Trade-off requirements

- The system should contain a mounting system.
- It is preferable to also measure:
  - NOx
  - O_3
  - SO_2

2.3. Requirements for the MPPT system

2.3.1. Mandatory requirements

From the overall programme of requirements, the programme of requirements for the subgroup can be derived. Functional requirements

- System should transfer the power from the PV array to the battery
- True MPPT

- Charge a Li-ion battery
- Supply energy to the Arduino MKR1010
- Handle input voltage and current of the PV array
- Battery temperature measurement

Non-functional requirements

- Protect the battery

2.3.2. Trade-off requirements

- System should be as energy efficient as possible
- System should be independent of the Arduino MKR1010 to facilitate orthogonal design

3

Design

3.1. Design overview

This chapter will discuss all the design choices that were made before the system could be implemented and tested. In section 3.2, the general concept of what the design should do and how it will achieve that will be discussed. Section 3.3 goes more into detail about the specific components that are used.

3.2. Design framework

In this section the general overview of the whole MPPT system is discussed. Figure 3.1 shows the power management system, its different components and the relation between these components.
In this system there are three components which are already determined, namely the PV panel, the Arduino MKR1010 and the battery. These form the outline in which the MPPT system will operate. The other two components, the micro-controller and Sunny Buddy, form the MPPT system. The way they interact can be summarised as follows.
On the micro-controller a MPPT algorithm is implemented which, based on the incoming current and voltage measurements of the PV panel, determines the operating point of the panel. It communicates this information via a control signal to the Sunny Buddy. This board controls the power flow to the micro-controller, the Arduino MKR1010 and battery and can also set the operating point of the PV panel. This is done by reading out the control signal and then adapting the duty-cycle of the embedded buck-converter accordingly.

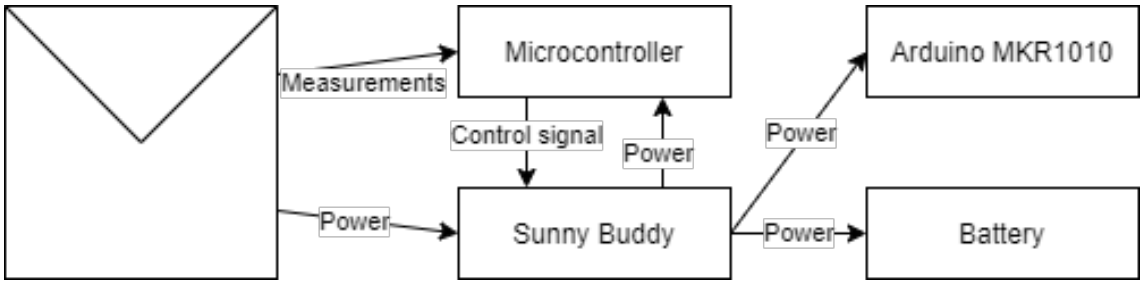


Figure 3.1: General overview of the MPPT system

3.2.1. PV characteristics

The PV panel is at the base of the system. One of the goals of this design is after all to extract the most power out of the panel. In order to do so it is important to keep its characteristics in mind and build the system around those. A short summary can be found in table 3.1.

Table 3.1: Important characteristics PV panel

Characteristic	Value
$Vmpp_{stc}$	5.758V
$Impp_{stc}$	1.5848A
Voc_{stc}	7.3V
Isc_{stc}	1.754A
Vmax	8.5V
Imax	1.5848A

In the table the 'stc' stands for 'under standard test conditions'. Meaning that these values were found at a temperature of $25°C$ and an irradiance of $1000W/m^2$. Most important are the maximum ratings and the voltage and current since these determine the boundaries in which the system must be able to operate.

3.2.2. Arduino MKR1010

At the output of the system, the Arduino MKR1010 is located. This micro-controller, which is required from the project description, will be used to collect the data from the different sensors and send it over Wi-Fi to get it processed. This micro-controller requires stable 3.3V input power at all times. When there is not enough solar energy available, the battery will be used to still supply the Arduino with enough power.

3.2.3. Battery

As mentioned in the previous section, the battery will function as a backup when there is not enough solar energy available. This will be at night or during periods with low irradiance, typically found in winter. The design choices for the battery are rather straightforward.
The type will be determined by the evaluation board. A large part of the efficiency of the system is determined by the evaluation board. To keep the options for the board as open as possible it is more logical to keep the battery type free. Later, in section 3.3.1, it is decided to use the Sunny Buddy which means Li-Ion batteries will be used.
Secondly the configuration is to be determined. Again this heavily relies on the evaluation board. The required charging voltage for Li-Ion batteries is $4.2V$. The evaluation board is able to down-convert the PV panel voltage to that level. However, placing two or more batteries in series requires a higher voltage which the board is not able to provide, leaving only a parallel connection as an option. Besides the charging voltage a parallel connection also has the advantage that the batteries always charge at the same rate. Meaning they are always operating at the same level of discharge which is important for the safety of the batteries [33].
Finally the total capacitance is found by analysing the generation, consumption, weather data, efficiency and to a lesser extend the control. From a configuration perspective the best solution is one very large battery however after looking at what is available in the current market a $3500mAh$ battery gives the best balance between capacity and costs.
In the analysis of sub group 1, which looks at the irradiance levels in Delft across 12 years, it can be seen that most of the time a battery of $3500mAh$ is fully charged. Extra capacity is only needed a few times per year. It can thus be concluded that in order to ensure full autonomous operation for one year, two or three parallel connected $3500mAh$ Li-Ion batteries are the best option.

3.3. Hardware

3.3.1. Converter choice

The first choice to be made is which converter board to use. In the state of the art analysis a short overview of available development boards is given. To choose one of these boards the following design requirements need to be taken into account: "The voltage and current range of the panel, the efficiency of the board, the required voltage of the Arduino MKR1010 and the MPPT implementation". Since all cells are connected in series, the voltage of the panel will be higher than the battery voltage. So down conversion will be needed.
In table 1.3 presented in the introduction the most suitable evaluation boards are summarized. Given

the just explained need for a buck-converter there are three options suitable. Each has a different but already implemented MPPT algorithm. For this project the choice was made to use the Sunny Buddy. It operates within the required ranges but most importantly has the possibility to replace the built-in MPPT algorithm.
In short, the built in constant voltage algorithm, is based on taking a fraction of the panel voltage via a potentiometer. This is then used as a control signal to move the operating point. By disconnecting this potentiometer and replacing it by a set voltage this behaviour can be manually controlled. This is not possible with the other boards where the algorithms are embedded into the chips, and no external feedback mechanism is used.

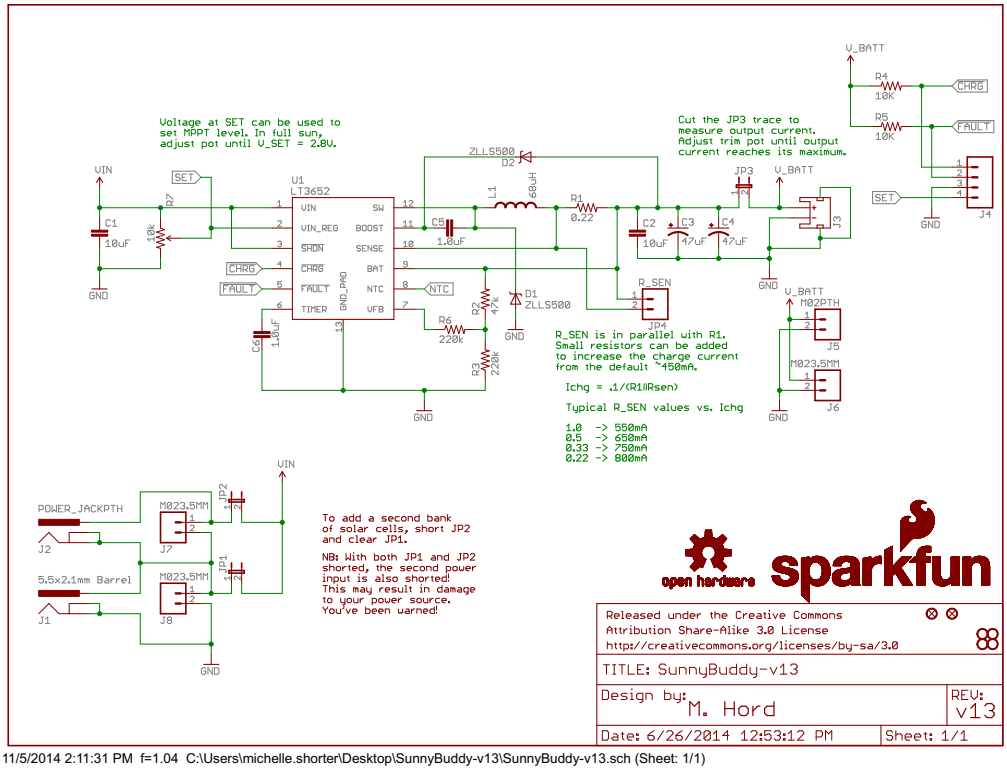


Figure 3.2: Schematic of the Sunny Buddy by Sparkfun

3.3.2. LT3652

The LT3652 [25] chip by Analog Devices is the brain of the Sunny Buddy. The LT3652 is a monolithic step-down battery charger. The LT3652 provides a constant-current/constant-voltage charge characteristic. It is programmable up to 2A. The LT3652 can be configured to terminate charging when charge current falls below 1/10 of the programmed maximum charge current. Once charging is terminated, the LT3652 enters a low-current (85µA) standby mode. An auto-re-charge feature starts a new charging cycle if the battery voltage falls 2.5% below the programmed float voltage. The LT3652 also contains a programmable safety timer, used to terminate charging after a desired time is reached. This allows top-off charging at currents less than one tenth of the charge current.

3.3.3. Controller

In order to implement a true MPPT algorithm the decision was made to add a microcontroller. This means that a complete analog circuit will not be designed. The deciding factors were the added complexity in combination with time constraints. One could argue that this solution does not result in the most energy efficient system however, a digital implementation gives more flexibility when it comes to testing different algorithms and/or adapting them. Therefore increasing the change of implementing the most efficient algorithm. The micro-controller that will be used in this project is the Arduino Zero.

It has one digital to analog converter and multiple analog to digital converters. These are needed to read the current and voltage measurements which are in turn needed for the algorithm to determine the MPP. The DAC will be used to output a control signal which is used to get the operating point at the MPP.

3.3.4. Measurement

To be able to implement MPPT, the power of the panel needs to be measured. This is done by measuring the PV panel voltage and current. Because the input of the Arduino Zero can handle a maximum of 3.3V, extra circuitry needs to be implemented.

Voltage measurements

The most simple to implement is the voltage measurement setup. A voltage divider will suffice. Important to keep in mind is the maximum voltage rating of the panel since that determines the ratio of the divider. Furthermore, large resistances are preferred to keep the energy efficiency as high as possible. Of course the input of the Arduino also has a high input resistance to minimise the losses. For this design 3.3 a potentiometer with a series resistance of $10k\Omega$ is used. This gives the flexibility to connect different panels with different maximum ratings.
When the buck-converter in the Sunny Buddy goes into discontinuous mode the signal quality drops and high frequency noise starts to appear. To prevent this, a low pass filter is added with a cut-off frequency less than $1MHz$ which is the operating frequency of the buck-converter. The capacitance of C1 is 100nF. The resulting sub-circuit can be found below.

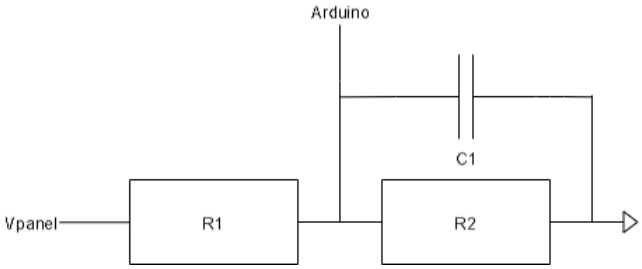


Figure 3.3: Voltage divider including filter

Current measurements

To measure the current, a current sensor is placed between the output of the panel and the input of the Sunny Buddy. The sensor used in this project is the INA169 which was already available. The INA169 generates a voltage which is linear to the current. With a $10k\Omega$ resistor, this relation will be $1V \approx 1A$. This means that, given the maximum charge current of $2A$, the output voltage of the current sensor will never exceed the maximum input voltage of the Arduino, equal to $3.3V$.
Like with the voltage measurements, also the current is subject to the high frequency components caused by the buck-converter in discontinuous mode. Therefore a filter is placed between the current sensor and the Arduino. C2 is 100 nF, and R3 = 1kΩ. This results in the sub-circuit shown below 3.4, with a cut-off frequency of 10kHz.

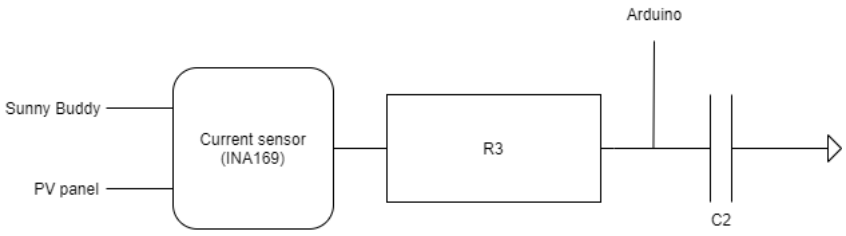


Figure 3.4: Current sensor including filter

3.3.5. Operational amplifier

The final component to design is the op-amp. The problem to solve is the voltage range difference between the output of the Arduino Zero ($0 - 2.3V$) and the input of the Sunny Buddy ($2.65 - 2.75V$). In order to preserve the accuracy of the output of the Arduino while keeping the implementation relatively simple the solution was to use a biased negative feedback circuit. Figure 3.5 shows the general concept.

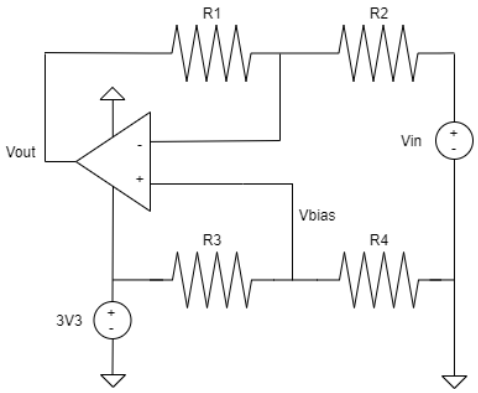


Figure 3.5: Simple overview of the op-amp design

The supply, desired input and desired output voltages are all known. They are respectively $3.3V$, $0 - 2V$ (for simplicity the last $0.3V$ is not used) and $2.65 - 2.75V$. With this information the values of the resistors can be calculated. The parameters mentioned in this calculation can be found in figure 3.5. First the bias voltage is determined.

$$V_{bias} = V_{out}/(1 + adjustment\_factor) \tag{3.1}$$

Since a range of $2V$ should be mapped to a total range of $0.1V$ the conversion per $1V$ is equal to $0.05V$. Filling this in 3.1 gives $V_{bias} = 2.619V$ Now, the first resistor value can be calculated.

$$R_3 = R_4 * (V_{supply}/V_{bias} - 1) \tag{3.2}$$

All values are known except that of R_4. In the simulations the value of $1k\Omega$ is chosen because it is readily available, the other resistor values are also available and it is high enough to ensure minimal losses. For now this will be kept variable and can be chosen freely at the end.
With R_3 known, the calculations for R_1 and R_2 are relatively simple. R_1 should be equal to the parallel resistance of R_3 and R_4 to ensure the op-amp functions as a differential amplifier. Finally R_2 can be determined by applying the wanted adjustment_factor. Both relations are shown in equations 3.3 and 3.4.

$$R_1 = R_3 * R_4/(R_3 * R_4) \tag{3.3}$$

$$R_2 = R_1/adjustment\_factor \tag{3.4}$$

One final observation to be made is that the required output voltage of the op-amp comes really close to the supply voltage. The difference here is only $3.3 - 2.75 = 0.55V$ while most regular op-amps need a overhead of around $1V$. Since there are no such voltage levels available (using the pv panel voltage is not desired because of its unstable behaviour and possible efficiency loss), a special op-amp needs to be used called a rail-to-rail op-amp. These kinds of amplifiers can generate outputs up to the supply voltage.

3.4. Software

The only place software can be found in the system is on the micro-controller, for this project the Arduino Zero is used. There are however a number of requirements the software should meet. Referencing back to the programme of requirements (2.3) the software should implement a true MPPT and be as energy efficient as possible.
The software will receive two inputs via two ADC's. One representing the PV panel current and the other

one the voltage. These converters have an adjustable accuracy between 10 to 12 bits. The output will be an analog signal varying between $0 - 2.3V$. This will be produced by an 8-bit DAC and corresponds to digital values ranging between $0 - 187$. The DAC can however also be adjusted to support 10 bits input. In the next sections the three main functions of the software will be discussed namely the implementation of true MPPT, the optimization regarding energy efficiency of the micro-controller and some additions to make the code more robust.

3.4.1. MPPT algorithm

Since true MPPT is required the number of possible algorithms is already limited. In section 1.3.2 an overview can be found of the most used algorithms. The ones suitable for this application given the requirements are listed below.

1. Hill-climbing/P&O
2. Incremental conductance
3. Fuzzy logic control
4. Neural network
5. RCC
6. Current sweep
7. dP/dV or dP/dI feedback control

A few of those can already be crossed of. A neural network algorithm would be far to complex for the time frame of this project and would also be too advanced given the conditions in which the total system is operating. The current sweep algorithm, although its relatively easy to implement, brings a lot of efficiency losses. Since power consumption should be minimised, better options are available. The last one to cross of is the RCC algorithm. Because the RCC algorithm is purely analog it would be a great fit for this project. However due to time constraints and the limited control options on the Sunny Buddy, this algorithm is also not implemented. See also the reasoning in section 3.3.3 In the end the choice was made to implement incremental conductance.

3.4.2. Micro-controller energy optimization

The micro-controller used in this project is the Arduino Zero. This controller however has a lot of functionality that won't be used like certain LED's and most of the analog to digital converters (ADC). The micro-controller will be continuously analysing whether the MPPT is reached or not. This means that, in order to minimize the power consumption, all other functions should be switched off or even soldered off. In practice this comes down to two steps. First the three LED's can be soldered of. Their only function is to indicate when the board is switched on, when it receives and sends data via the data-pin (rx and tx). The second step is to program the standby mode. In this mode the board goes into power-saving mode potentially switching everything off by disabling their clocks. The benefit of this mode is that it is possible to specify which modules shouldn't be switched off. In our case those would be two ADC's, the DAC and the IC module.

3.4.3. Robustness

The implemented algorithms will use the incoming current and voltage values to base their calculations on. These values are converted via the input ADC's and are thus subject to noise and limited accuracy. To make sure the algorithm functions reliably the choice is made to implement a rolling average in combination with the filters mentioned in the previous section (3.3). The length of this average will be determined in the implementation part and depends on the quality of the incoming signals.

4

Simulation and implementation

In this chapter the previous discussed theory is put into practise. At the start a number of simulations are carried out to confirm that the system is able to function. After that, two steps were taken in parallel. The first one being actually building a prototype. At the same time some more detailed simulations were done to get information about the generation and consumption resulting in the most important characteristic, the efficiency.

4.1. Simulation

The simulations were done in two environments. First LTSpice was used. This environment is developed by Linear Technologies and already has a blackbox model of the chip used in the Sunny Buddy. From these simulations a lookup table of efficiencies against irradiance is constructed which can be used in the second set of simulations. These are done in Matlab and the goal of these simulations is to predict an accurate efficiency of the complete system with the use of real weather data.

4.1.1. LTSpice

To simulate the system four separate models were made. Because the system will be solar powered, the first model that will be discussed is the single diode model. This is used to simulate a solar panel under varying irradiance and temperature. The second model used, is the model of the Sunny Buddy itself. By combining them, the third model is created. This model will be used for most simulations since it gives the most accurate representation of the system working. Finally a more simple model was made to design and confirm the working of the operational amplifier, discussed in section 3.3.5. The next few sections will go into more detail about the models.

Model of solar panel

One of the problems that became more and more important, was to accurately simulate a system that is generating a correct input. The most effective way was to make a complete model of the PV-panel that is going to be used. With the help of sub group 1, consisting of M. Haas en A.W. van der Knaap and with the use of this paper [34] a PV model was created. This model, based on the often used single diode model, simulates the output of the PV panel based on the temperature and irradiance. In figure 4.1 the schematic is shown.

The voltage source $V1$ is not part of the single diode model but is used to sweep the panel and construct the I-V characteristics of the panel. The code used to describe the current source, resistors and diode can be found in Appendix B.1.

Model of the Sunny Buddy

Another quick simulation that was done is that of the Sunny Buddy itself. The goal of this simulation is to verify the described behaviour of the board. Note that in this case the standard potentiometer is used at the V_{in-reg}-pin and that the input is determined by a simple PV model connected to the V_{in}-pin. Apart from the chip, all other components were placed as described in the schematic of the Sunny Buddy which can also be found in figure 3.2.

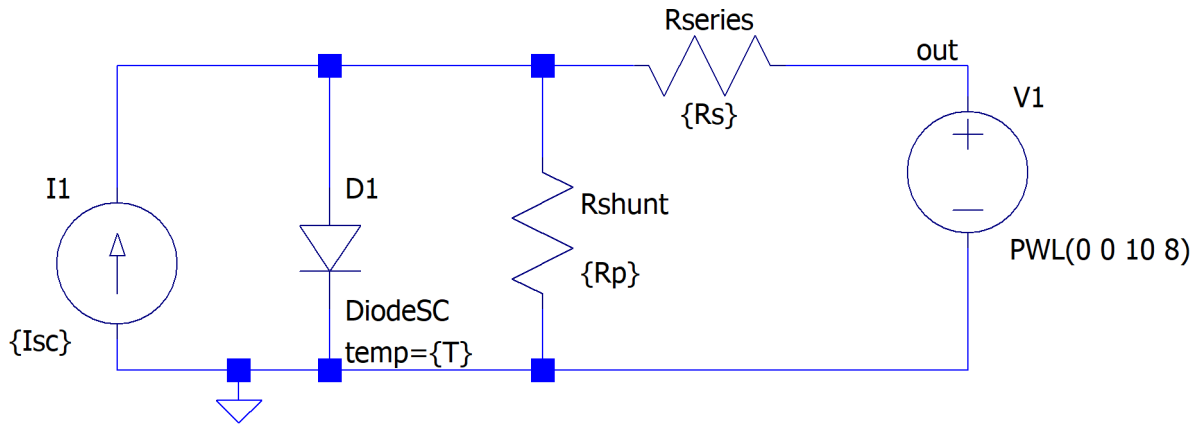


Figure 4.1: LTSpice single diode model

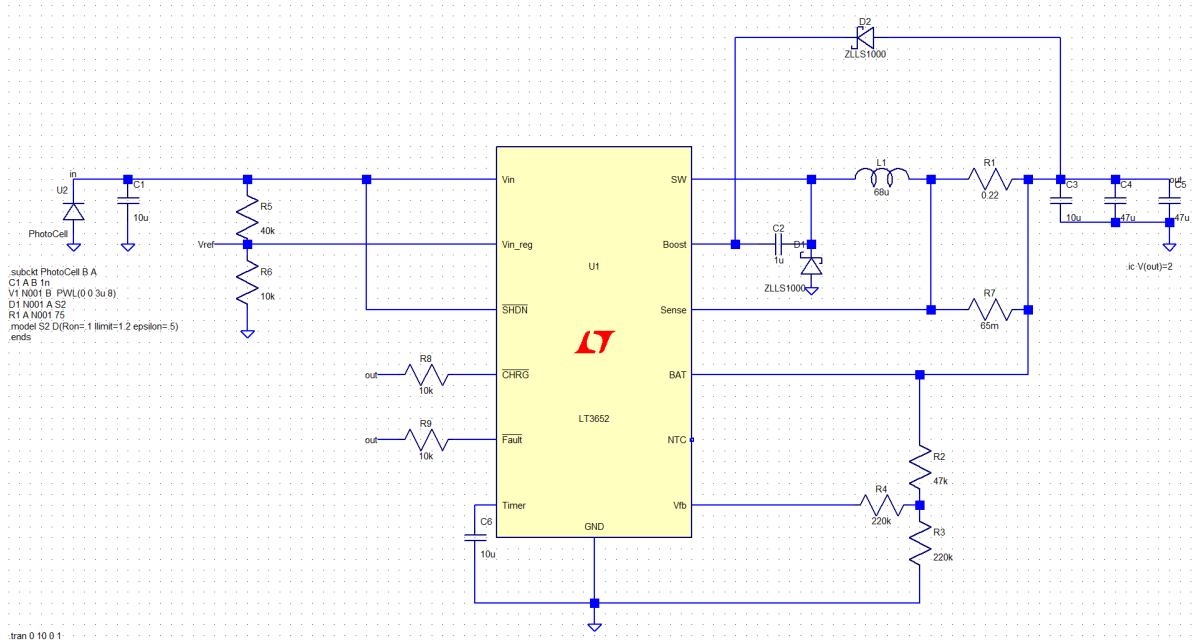


Figure 4.2: LTspiceXVII model of the Sunny Buddy

Model combining Sunny Buddy with PV panel

In figure 4.2 the model of the Sunny Buddy is shown. A number of changes have been made with respect to the previous model which will be discussed next. The most obvious one is the addition of the PV model at the input (indicated in brown). Staying at the input, the next change is made at the V_{in-reg}-pin. The potentiometer is switched out for a voltage source (indicated in green). This source will represent the signals coming from the Arduino Zero after the op-amp. The final change can be found at the output where a battery has been connected (indicated in red). With this model the effect of various control signals on the operating point can be simulated. Furthermore, the temperature but more importantly the irradiance levels can be changed to create simulations under all environmental circumstances. This simulation is used to find the maximum power points at different irradiance levels by sweeping the control signal. Since the generated power (input) and the power going into the battery (output) are known, as a result of such simulations, the efficiency under varying irradiance and temperature can be calculated. With this information a lookup table can be constructed to be used in the Matlab simulations.

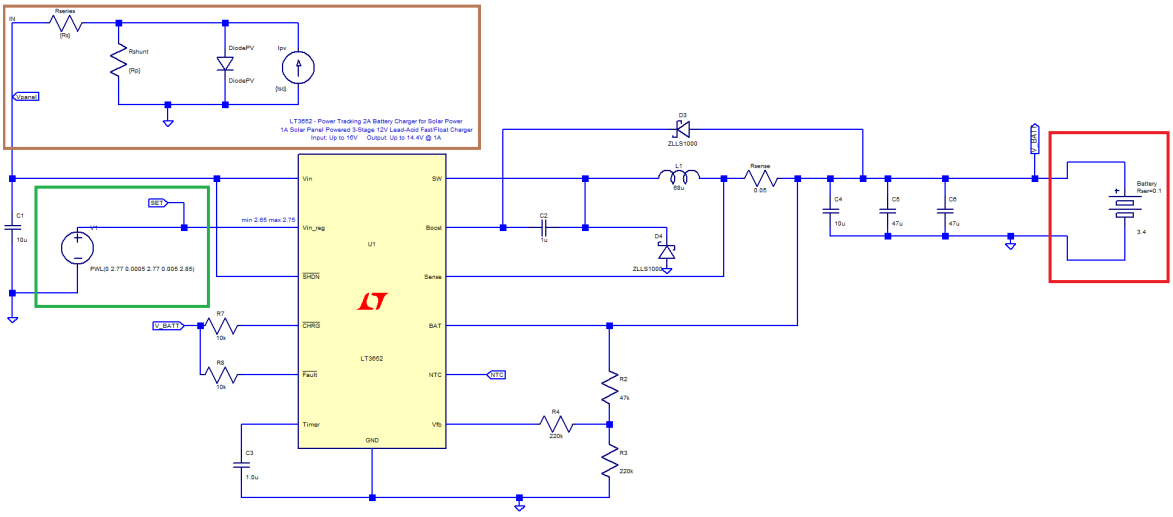


Figure 4.3: Model of the PV panel and Sunny Buddy combined

Operational amplifier

Last but not least also the behaviour of the designed op-amp circuit was simulated. For this simulation the OPA365 is used, which is a rail to rail op-amp. Furthermore, as discussed in section 3.3.5, the resistor values are R1 = 220 Ω, R2 = 3900 Ω, R3 = 270 Ω and R4 = 1000 Ω

In figure 4.4 the simulation setup is shown. A sinusoidal input voltage is used to 'sweep' through all the values.

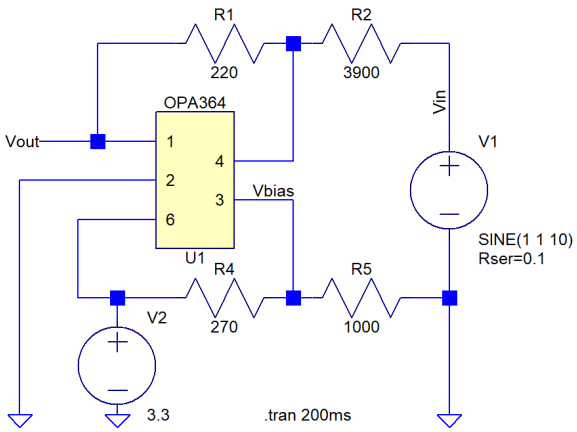


Figure 4.4: Simulation setup for the op-amp

4.1.2. Matlab

The matlab model uses the lookup table of the different efficiencies to calculate the power that is generated based on the weather data from Delft over the past 12 years. With the lookup table and the weather data, the total power generation can be calculated. The code used for the model can be found in the appendix B.2

The tracking speed has also an influence on the efficiency used by this model. This parameter however is yet to be determined but early testing suggests a negligible influence.

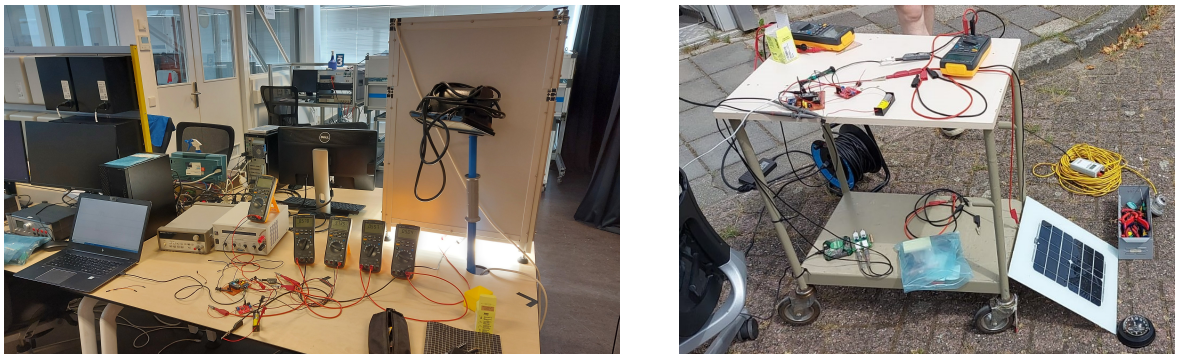
5

Results and validation

In this chapter the results of the simulation of the design are discussed. The simulation is validated using measurements of the physical system.

5.1. Testing setup and calibrations

To gather data from the prototype the two test setups shown in figure 5.1 were used. The solar panel is placed inside a solar simulation chamber which is a box with mirrors on each side and a set of lamps in the top. This way all kinds of irradiance levels can be simulated. Both setups can be found in Appendix A.1. The goal of tests is to compare and verify the simulations with the prototype. Values that are recorded are the irradiance, voltage and current levels of the PV panel and the voltage and current levels of the battery. Additionally the control signal at the V_{in-reg}-pin of the Sunny Buddy is monitored.



(a) Measurement setup indoor

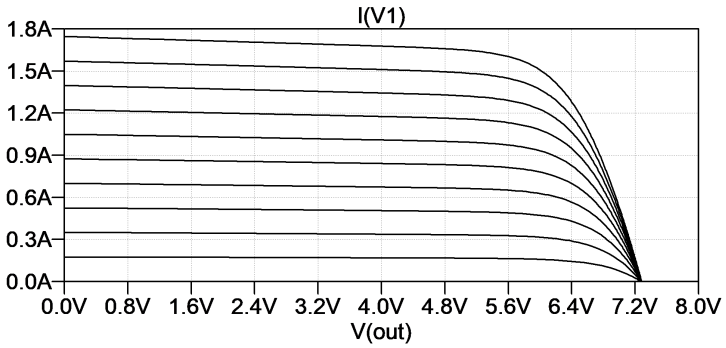
(b) Measurement setup outdoor

Figure 5.1: Measurement setups for the prototype

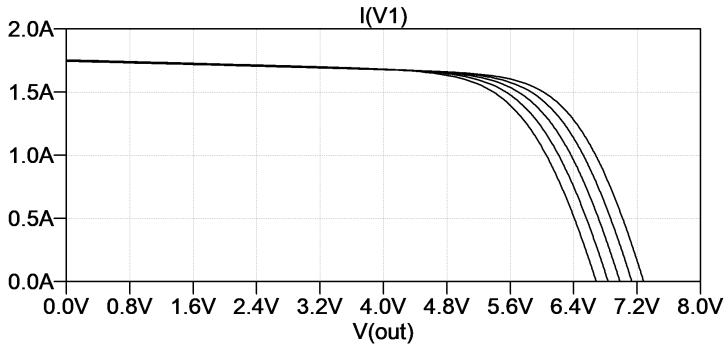
When testing outdoor there is no irradiance simulator needed. However to store all the data some additional steps had to be taken. First of all, to record the data from the Arduino for longer periods of time an additional program was used called CoolTerm [35]. It stores all the incoming data into one text file. Secondly the current and voltage of the battery also had to be monitored. This is done via a portable oscilloscope called the Analog Discovery 2 by Digilent [36]. One channel was used to measure the voltage over the battery and the other to measure the current. The latter one is accomplished by putting a 0.1Ω-resistance in series with the battery and measuring the voltage. This will negatively effect the efficiency of the system but since it replaces a multi-meter the comparisons between measurements are still valid. At $1000W/m^2$ the maximum efficiency loss is found which will be around $0.1W = 1.1\%$. The resistor value has been verified by applying a known voltage over the resistance and measuring the current with a multi-meter.

5.2. PV generation

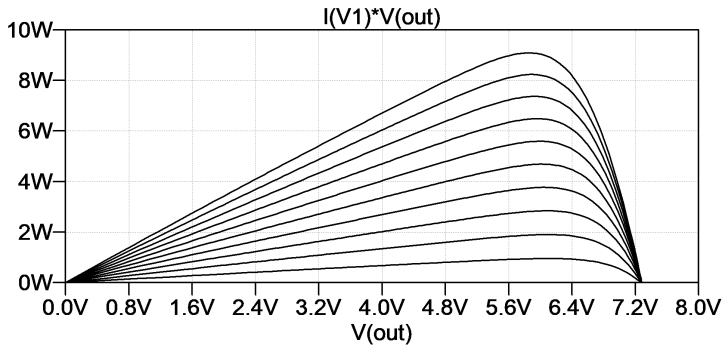
The simulation discussed in 4.1.1 was run by applying a voltage over the PV panel varying between 0 and $8V$. Then there are two main variables when it comes to the MPP, namely the irradiance and the PV panel temperature. In figure 5.2 this behaviour is plotted. On the left side the irradiance is varied between 0 and $1000W/m^2$ in steps of $100W/m^2$ and the temperature is kept at $25°C$. On the right side the irradiance is kept at $1000W/m^2$ and the temperature varies between $25°C$ and $45°C$ in steps of $5°C$. The bottom two graphs show the output powers. Those peaks also show that for different irradiance levels the MPP stays around the same voltage but has significant changes in magnitude whereas for different temperatures the MPP voltage shifts quite linear but varies the power no more than $1W$. The MPP-power versus irradiance is also summarized in table 5.1. This data will be used to determine the efficiency of the system later on. In table 5.2 the MPP's at different temperatures are summarized. For the next part of the results, temperature is neglected because the impact is about ten times smaller than the impact of range of irradiances, and is thereby only important when implementing a MPPT algorithm.



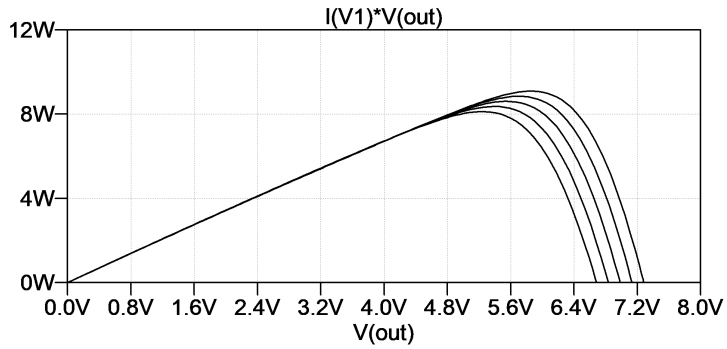
(a) I-V-curve at different irradiance levels



(b) I-V-curve at different temperature levels



(c) Power-curve at different irradiance levels



(d) Power-curve at different temperature levels

Figure 5.2: PV characteristics

Table 5.1: Overview of MPP's at different irradiance, temperature is $25^{\circ}C$

Irradiance (W/m^2)	MPP (W)
0	0
100	0.954414
200	1.90799
300	2.85268
400	3.78398
500	4.70339
600	5.61407
700	6.51438
800	7.37831
900	8.25938
1000	9.08727

Table 5.2: Overview of MPP's at different temperature, irradiance is $1000W/m^2$

Temperature ($^{\circ}C$)	MPP (W)
25	9.0914
30	8.85044
35	8.60898
40	8.36629
45	8.12319

5.3. Sunny Buddy

Figure 5.3 shows the results of the simulations for the panel combined with the Sunny Buddy, for the irradiances mentioned in table 5.1. The power output of the panel (upper figure) and power going into the battery (lower figure) are shown. On the x-axis time is set, however important to know is that the control signal goes linearly from $2.77V$ at $0.5ms$ to $2.85V$ at $5ms$. This means that within $5ms$ a sweep is done to get the complete controllable and thus accessible power values. Some interesting behaviour can be seen in these graphs. First of all, the panel has indeed a maximum power point which, according to the simulations, is reachable. Quickly after this point is reached the power drops until the device goes in to discontinuous mode with a switching frequency of $1MHz$. This corresponds to the operating frequency found in the datasheet and is also why the voltage and current filters are implemented. In the bottom it can be seen that the device is able to transfer most of its power to the battery. At this point an efficiency table can be constructed where the MPP of the panel is compared against the power going into the battery, see table 5.3. The *PV MPP Ideal* column is the same as in table 5.1 and is the MPP of the simulated PV panel. Similar simulations have shown that adding the Sunny Buddy to the PV panel has no effect on these values. Column $P_{battery}$ *at MPP* shows the power going into the battery at the MPP of the panel. The last column shows the efficiency which is inherently the expected efficiency of the Sunny Buddy. The negative effects on the efficiency, because of the consumption of the Arduino and the possible imperfections of the algorithm, are not yet taken into account. What can be concluded from this table is that for lower irradiance the efficiency increases. This can in part be explained by the current sense resistor, where higher currents generate proportionally more power loss. This resistor can be found in the schematic shown in figure 3.2.

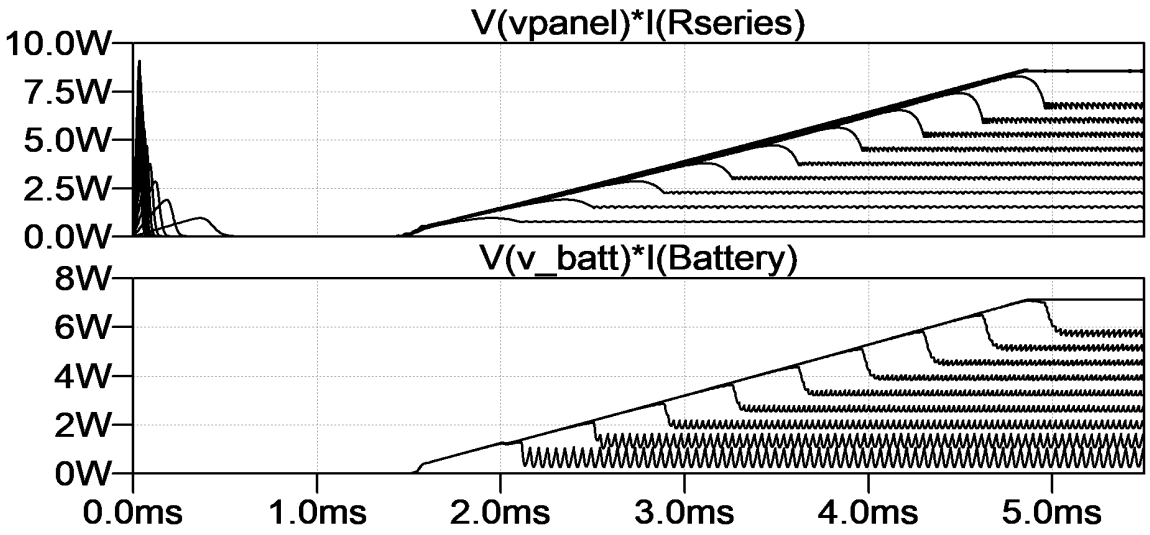


Figure 5.3: Simulation results of the Sunny Buddy combined with the pv-panel

Irradiance (W/m^2)	PV MPP Ideal (W)	$P_{battery}$ at MPP (W)	Efficiency Battery (%)
0	0	0	-
100	0.954414	0.852	89.2
200	1.90799	1.656	86.8
300	2.85268	2.454	86.0
400	3.78398	3.271	86.4
500	4.70339	4.077	86.6
600	5.61407	4.796	85,4
700	6.51438	5.505	84.5
800	7.37831	6.234	84.2
900	8.25938	6.952	84.0
1000	9.08727	7.124	82.5

Table 5.3: Simulated efficiency overview Sunny Buddy and PV panel

The same type of tests were also conducted in the real world to compare the simulated efficiency with the efficiency of the prototype. The difference between the simulation and test of the prototype is that the duration of the ramp signal was longer for the prototype. The figures 5.4 and 5.5 show the sweep of the control signal (green) and the corresponding power received from the PV panel (blue). The first observation is the position of the maximum power point or rather the lack of it. Using the complete range available for the regulation pin at maximum accuracy it still does not show a maximum power point. The reason why can be explained by looking at the current and voltage levels. These are shown in figure 5.6 for $760W/m^2$ but look the same for all irradiance levels. Again the control signal is indicated in green.
There are three distinctive phases in this figure. The first and last show a plateau of the current and voltage and in between there is a transition phase. The unsuspected behaviour starts at the end of phase two, where the system goes from decreasing the voltage and increasing the current to phase three, the discontinuous mode. With the oscilloscope the voltage and current were measured to verify this. There are two reasons why the Sunny Buddy goes into this phase. One, the maximum charge current is reached and second the minimum voltage level is reached. Neither of those seems to be the case according to the datasheet. The maximum charge current set to $2A$ which is not reached and the minimum voltage is the battery voltage plus $0.75V$. This would be, using a maximum charged battery, at a maximum of $4.2 + 0.75 = 4.95V$. The voltage in the graph is nevertheless never below $5V$. Up to now, the most valid explanation is that the minimum voltage is in reality higher than described in the datasheet. This in turn is probably due to the different configuration of the Sunny Buddy compared to the configuration used by Linear Technologies. In other words, the LT3652 chip used in the Sunny Buddy is wired differently then the setup used in the datasheet provided by Linear Technologies. As a result, reaching the maximum power point is not feasible while using the current panel, causing a significant efficiency drop. It also explains the flat line of phase three at the end of figures 5.4 and 5.5.

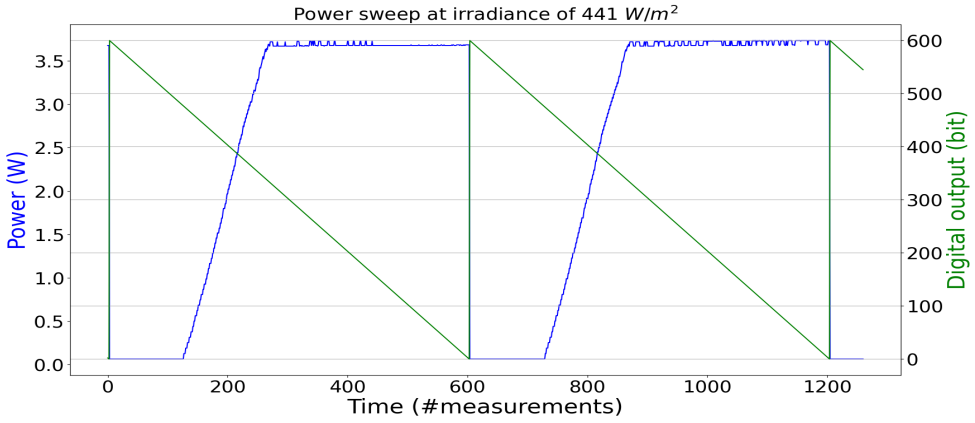


Figure 5.4: Power sweep at 441 W/m^2 irradiance

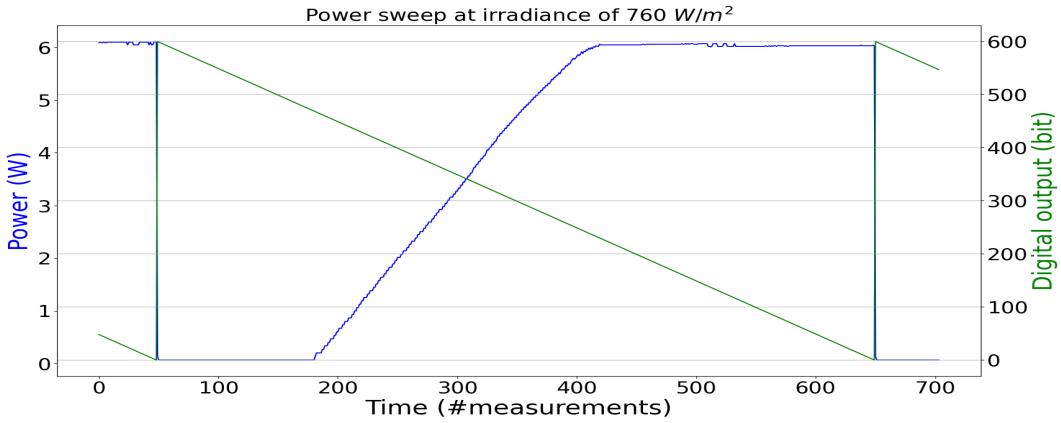


Figure 5.5: Power sweep at 760 W/m^2 irradiance

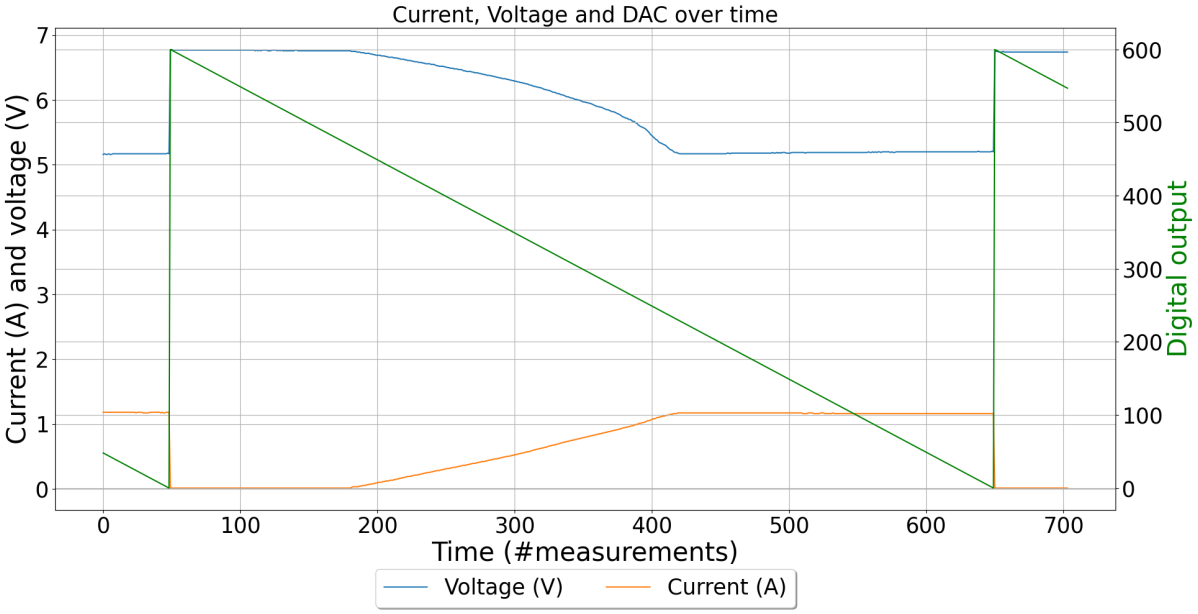


Figure 5.6: Current and voltage at 760 W/m^2 irradiance

Back to figures 5.4 and 5.5, maximum power of $3.72W$ was found for $441W/m^2$ irradiance. The same is done for $750W/m^2$ irradiance with a maximum power of $6.09W$. Using table 5.3, it can be shown that the achieved MPP is respectively at 89.4% and 86.6%, of the actual MPP of the panel.

Measurements regarding the efficiency of the Sunny Buddy were also carried out, resulting in table 5.4. the power consumption of the microcontroller is measured at $20mA$ continuously at $3.3V$. This can be much lower when it is put into deep sleep at night however for the sake of simplicity it is kept at a constant power consumption of $66mW$. Using the microcontroller also means that the panel operates at the maximum power point thanks to the MPPT algorithm. The efficiencies without the Arduino are relatively stable around 80%. Meaning that the converter is around 80% efficient. When the Arduino is connected, the efficiency drops with lower irradiance. This is due to the fact that the power consumption of the Arduino is constant and thus has a bigger effect on the efficiency when the generated power is lower. The voltage drop of the battery is due to the effects of charging the battery. At low irradiance levels the battery is charged less, resulting in a lower voltage across the battery. Averaging the*efficiency with Arduino* gives an estimated efficiency of around 74.8% of the converter combined with the Arduino. The last two columns give the efficiency of the panel to the battery when the MPP of the panel is also taken into account. For a given irradiance the MPP of the PV panel is known from the model of the PV panel. The difference between the achieved operating point of the panel and the calculated MPP is

used to calculate the extra inefficiency of the prototype. The difference between the simulated efficiency and achieved efficiency are mostly due to the fact that the Sunny Buddy cannot operate the PV panel at MPP in practice, but can in simulation.
In combination with the average efficiency of the panel of around 87%, an average efficiency of the complete system of 65% is found.

Irradiance (W/m^2)	P_{Panel} (W)	P_{bat} (W)	Efficiency without Arduino (%)	Efficiency with Arduino (%)	Efficiency without Arduino with ideal MPP (%)	Efficiency with Arduino with ideal MPP (%)
68	0,654	0,519	79,3	69,2	80.0	69.86
181	1,749	1,412	80,7	76,9	82.1	78.3
262	1,871	1,503	80,3	76,8	60.6	57.9
374	2,982	2,358	79,0	76,8	66.9	65.0
488	3,698	2,900	78,4	76,6	63.4	61.9
678	3,811	3,002	78,7	77,0	47.7	46.69
750	3,898	2,985	76,5	74,8	43.1	42.1

Table 5.4: Efficiency Sunny Buddy at different irradiance levels

5.4. OPA365

The results of the simulation described in 4.1.1 are shown in figure 5.7. The input is a sinusoidal voltage source with an average of $1V$ and an amplitude of $1V$ thus sweeping the complete range of the DAC output of the Arduino. At the output a similarly shaped sinusoidal voltage can be seen. The amplitude and average differ compared to the input, making the output suited for the control-pin.

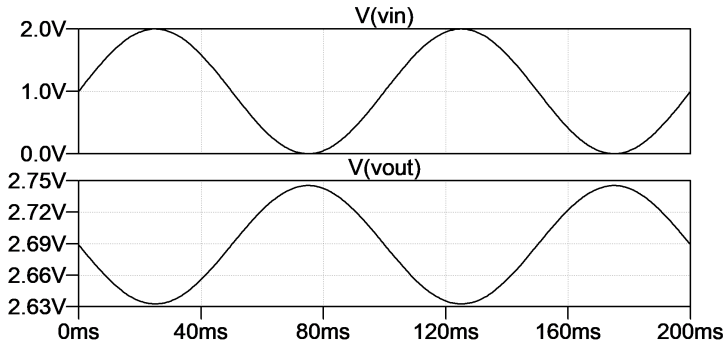
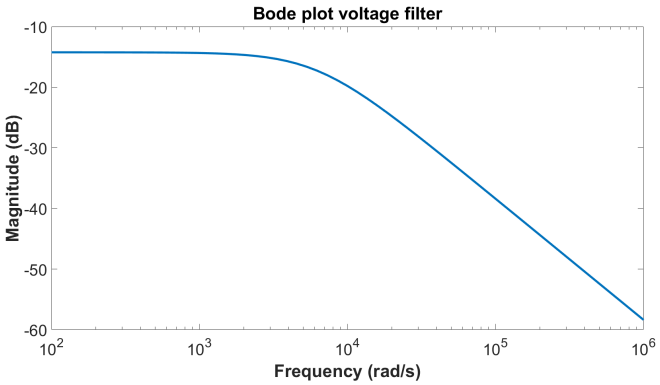


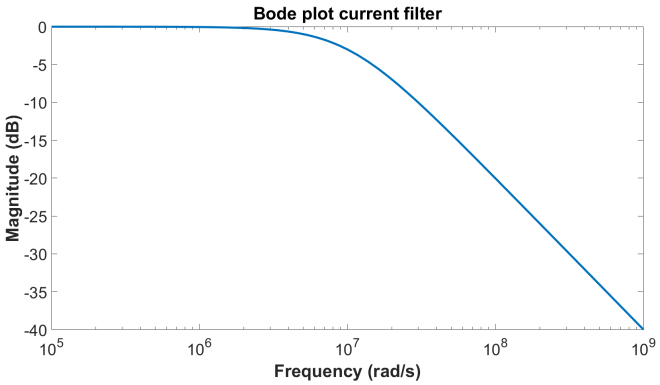
Figure 5.7: Simulation results op-amp

5.5. Filters

In the design the current and voltage measurement circuits are discussed, see section 3.3.4. These filters were also simulated and gave the following frequency responses, see figure 5.8. The cutoff frequency should be lower or at most equal to the $1MHz$, since that is operating frequency and thus the noise in discontinuous mode. This is the case for the voltage filter but for the current filter compromises had to be made due to limited available parts. Having said this, in practise it does filter out enough noise to get clean input current values. This can be seen in figure 5.6.



(a) Bode plot for the voltage filter, $f_{cutoff} = 985Hz$



(b) Bode plot for the current filter, $f_{cutoff} = 1.59MHz$

Figure 5.8: Frequency response of the filters

Furthermore, a conversion factor is determined to convert the bit readout at the ADC's to voltage floats. These factors are purely used for convenience and only exist in the Arduino code. This way, the current and voltage levels of the PV can easily be verified with the values read on the multi-meters used in the test setup. The results are shown in tables A.1 and A.2. The conversion factors found are 235.6 for the voltage and 1290 for the current.

5.6. Tracking speed

To track the MPPT the incremental conductance algorithm was coded and programmed onto the Arduino. The code can be found in the appendix B.3.2. The data acquired from real world testing shows that the tracking speed of the Sunny Buddy is almost instantaneous. This means that there is no energy lost in the Sunny Buddy because of slow tracking. As a result the tracking efficiency relies only on the algorithm. The implementation as of now takes an average of 50 samples in a time frame of $1ms$. These averages are then compared and processed according to the IC-algorithm described in section 1.3.2. The tracking speed of $1ms$ is fast enough to neglect any efficiency losses that may result from not being able to track the MPP.
This is also valid because the actual power point is always at low control values and because the algorithm makes use of adjustable step sizes. The latter means that it wont oscillate around the MPP because at MPP the step size will be reduced. On the other hand, when the MPP is not reached fast enough, it will increase the step size.

5.7. Battery charge

Two simulations are done in Matlab, shown in figure 5.9 and 5.10, that show the charge of the battery over a simulated period of twelve years, in Delft. The first simulation is based on the LTSpice model of the system, and the second is based on the measurements of the prototype. For both simulations a lookup table was generated, which give the power generation for a given MPP of the PV panel. At the end the power consumption is simulated using the algorithm design by group 3, consisting of Roan Föllings and Henk van Grootheest. In both simulations two batteries of 3000mAh, a nominal voltage of 3.7v, and a maximum depth of discharge of 80% are used. The code can be found in appendix B.2. The simulations show that both system can function autonomously for at least one year.

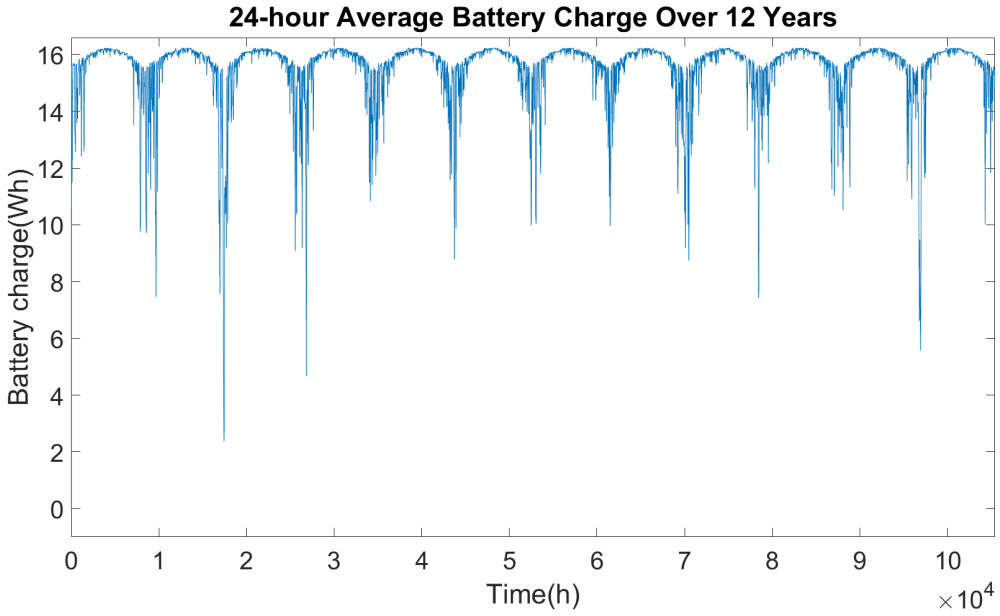


Figure 5.9: Charge in battery from simulation based on LTSpice

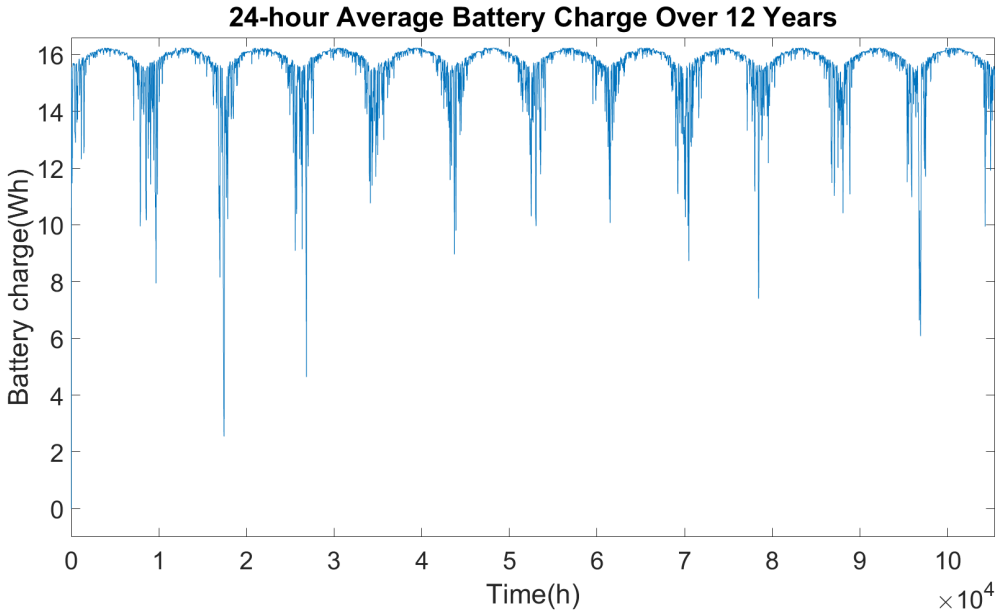


Figure 5.10: Charge in battery from simulation based on the measurements of the prototype

6

Discussion and conclusion

6.1. Conclusion

From the simulation, and from the testing, it can be concluded that the power management system can extract enough power from the solar panel to guarantee the adequate functioning of the weather station, for a whole year.
The system as designed and built meets the programme of requirements. It transfers the energy from the solar panel to the battery. The system charges the battery and it supplies energy to the Arduino MKR1010. It can handle input voltage and current range from the solar panel and the algorithm can find the MPP of the solar panel. There are also some parts however that don't function yet or that well. The first being the battery temperature control. The second one is the complete implementation of true MPPT.
The simulated system works better than the physical prototype. According to the datasheet from Analog Devices, the system should be able to track the MPP to a accuracy of 98%, and the simulation shows the system should achieve 80% efficiency. However test of the prototype show that the system does not perform as well as designed, and only about 65% of energy of the panel can be delivered to the battery. Due to the shortcomings of the Sunny Buddy it was in practice impossible to reach the MPP of the solar panel.
That being said, the system is independent of the MKR1010 and the Sunny Buddy comes with embedded battery protection. The final requirement to meet is the energy efficiency which can be improved in two ways. The most simple option is to optimize the current system (put the Arduino into deep sleep and get the Sunny Buddy to work with lower voltages) while the more effective choice would be to design a new system from scratch using all the knowledge gained from this project.

6.2. Discussion

The system works and meets the programme of requirements but leaves a lot to be desired. Due to the limitation of the Sunny Buddy, MPPT is not achieved in practice. This means another converter has to be chosen or designed from scratch. One possible solution is to implement a simple buck converter using a few components, and using the Arduino as a direct controller by driving the PWM to the switching mosfet, this would eliminate the need for the LT3652 chip. This solution gives far more control over the behavior of the converter than using the voltage control pin of the Sunny Buddy. It would also eliminate some unexplained behaviour, because the model of the LT3652 in LTspice is a black box. One disadvantage to this approach is the need for additional battery protection circuitry.

If the converter is able to more closely reach the MPP of the panel, then other parts of the system can also be improved. For instance the PV-panel could be made smaller, or cheaper technology could be used. On the other hand with improved tracking and efficiency it should be possible to increase the amount of measurements of the weather station, because more power would be available to do the measurements.

A Hardware

A.1. Testing setups

A.1.1. Testing setup indoor

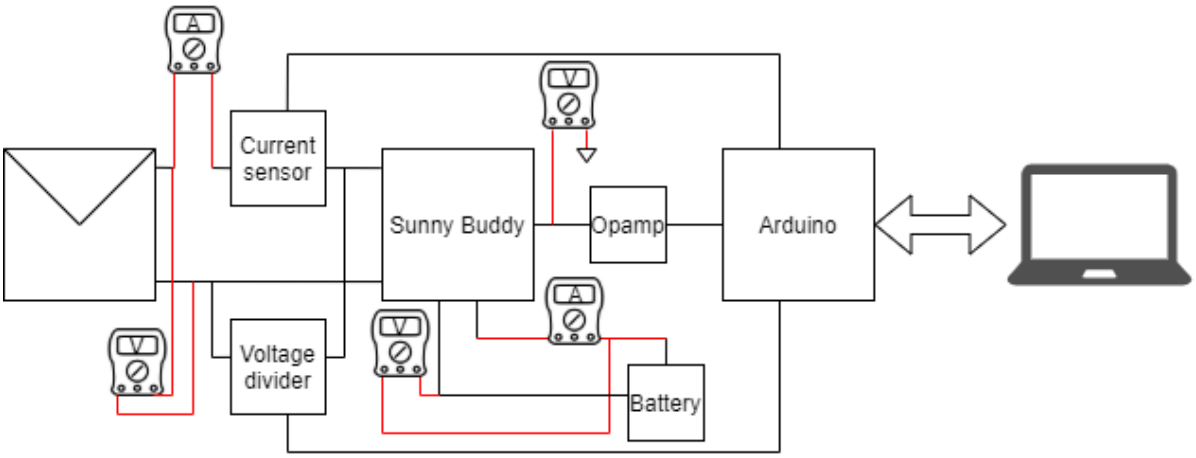


Figure A.1: Measurement setup indoor

In A.1 the rough schematic of the measurement setup for indoor is shown. To clarify, five multimeters were used in addition to a laptop which is able to interact with the Arduino. The line through the opamp indicates the control signal going into the V_{in-reg}-pin of the Sunny Buddy.

A.1.2. Testing setup outdoor

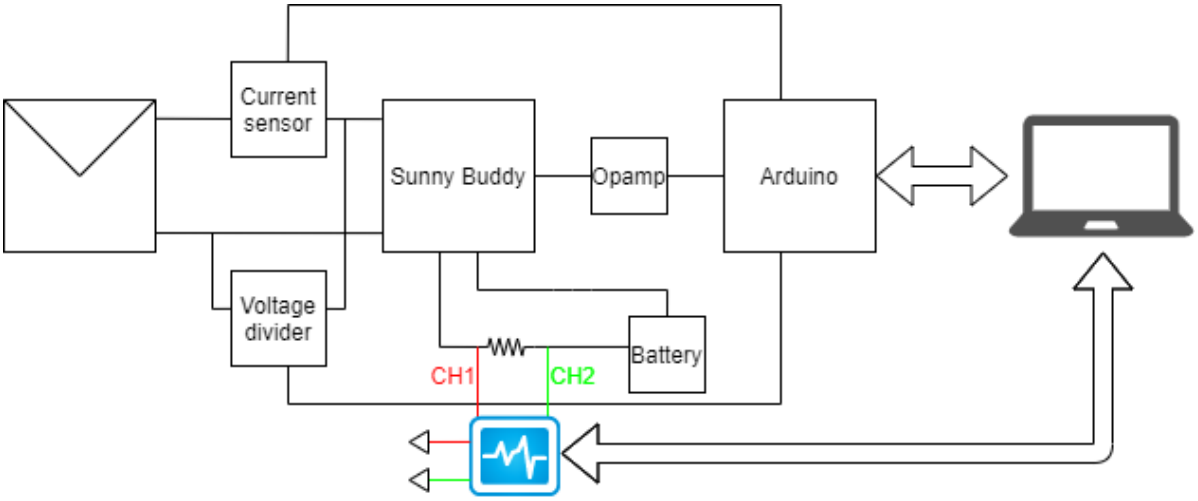


Figure A.2: Measurement setup outdoor

In A.2 the rough schematic of the measurement setup for outdoor is shown. Because in this case the measurements were done for longer periods of time, it was no longer possible to record the measurements by hand. As a solution, a portable oscilloscope is implemented and connected to the laptop. This way, both the PV panel (via the Arduino) and the battery (via the oscilloscope) can be monitored while still saving all the measurements. Because the oscilloscope can not measure currents and there were no current sensors available at the time, a small resistor (0.1Ω) is put in series with the battery. By applying differential voltage measurements the current can then be calculated.

A.2. ADC conversion factor

Tables used to determine the conversion factors. These are used within the algorithm to make the values used more readable.

Table A.1: ADC verification for $PV_{voltage}$

$PV_{voltage}$ (V)	ADC (bit)
5	1177
5.5	1297
6	1415
6.5	1532
7	1650
7.5	1763
8	1886

Table A.2: ADC verification for $PV_{current}$

$PV_{current}$ (A)	ADC (bit)
0.081	111
0.299	382
0.494	633
0.560	717

A.3. Table with measurements

This is the table from the results, but with the voltage and current instead of power for completeness.

Irradiance (W/m^2)	Vpv (V)	Ipv (A)	Vbat (V)	Ibat (A)	Efficiency without Arduino (%)	Efficiency with Arduino (%)	Efficiency without Arduino with ideal MPP (%)	Efficiency with Arduino with ideal MPP (%)
68	4,88	0,134	3,787	0,137	79,3	69,2	80.0	69.86
179	4,89	0,155	3,857	0,156	79,3	70,6	35.4	31.5
181	4,640	0,377	3,795	0,372	80,7	76,9	82.1	78.3
262	4,76	0,393	3,865	0,389	80,3	76,8	60.6	57.9
374	5,02	0,594	3,873	0,609	79,0	76,8	66.9	65.0
488	5,57	0,664	3,877	0,748	78,4	76,6	63.4	61.9
678	5,9	0,646	4	0,775	78,7	77,0	47.7	46.69
750	6,09	0,64	4	0,771	76,5	74,8	43.1	42.1

Table A.3: Efficiency Sunny Buddy at different irradiance levels

B

Simulation code

B.1. Single diode model

```
*Author:    Tibbe van der Biezen (4686608)
*Author:    Aru van der Knaap (4693531)
*Date:  04/06/2021
.param Isc_stc = 1.754 ;Short circuit current
+Rs_stc = 0.275408 ;Series resistance
+Rp_stc = 58.331508 ;Shunt resistance
+Voc_stc = 7.3 ;Open circuit current
+Irs = 1.546782n ;Reverse saturation current
+Ipv = 1.762  ;Photo current
+G_stc = 1000 ;Standard irradiance
+G = 1000 ;irradiance
+a_stc = 1.14 ;Ideality factor
+Kv = -0.356 ;percent per Celsius
+Ki = 0.024 ;percent per Celsius
+T_stc = 25 ;Standard temperature
+T = 25 ; Temperature
+Vmpp_stc = 5.758 ;Voltage at mpp
+Impp_stc = 1.5848 ;Current at mpp
+k =1.380653e-23 ;Boltzman constant
+q=1.60217646e-19 ;Electric charge
+N=1  ;Number of cells
*Formulas********

+Rs = Rs_stc
+Rp = {Rp_stc*G_stc/G}
+Isc = {Isc_stc*G/G_stc+Ki/100*Isc_stc*(T-T_stc)}
+vt = {a_stc*k *T*N/q}
+dV = {vt*ln({G}/G_stc)}
+Voc={Voc_stc+Kv/100*Voc_stc*(T-T_stc)+{dV}}
+Ec = {q/(k*(T+273.15))*{Voc}/ln({Isc}/{Irs})}

.model DiodeSC D(Is={Irs} N={Ec})
```

B.2. Matlab code

This matlab script simulates the complete power generation and consumption and construct plots indicating the power reserves left in the battery.

```
%Author Roan Föllings
%Date: 18-6-2021
```

```
%Edited by Tibbe van der Biezen
%Date: 18-6-2021

clear all;
load('Pmax_Data.mat');

%MPPT efficiency parameters
x = [0.64816 1.71895 2.48243 3.52691 4.57575 6.28847 6.92672];
y_physical = [0.80 0.821 0.606 0.669 0.634 0.477 0.431];
y_ideal = [0.793 0.807 0.803 0.790 0.784 0.787 0.765];
mppt_efficiency = polyfit(x, y_physical, 4); %Change to physical to get the
 ↪ corresponding efficiencies
arduino_consumption = 0.022; % 22mW consumption (night time is already
 ↪ accounted for)
% generation parameters
data = Pmax38(1:8784,1:12);
data = reshape(data,[],1);
efficiencies = polyval(mppt_efficiency, data); % production in Watt,
 ↪ includes solar panel efficiency and area
production = efficiencies.*data - arduino_consumption;
% production = 0.65*data;
% consumption parameters
percentage_per_month = 3; % percentage of total battery charge dissipated
 ↪ per month
idle_consumption = 0.0165; % baseline power use
short_consumption = 0.0000231; % additional average power use for one short
 ↪ measurement per hour
pm_consumption = 0.005147; % additional average power use for one pm
 ↪ measurement per hour
gas_consumption = 0.01419; % additional average power use for one gas
 ↪ measurement per hour
complete_consumption = 0.02052; % additional average power use for one
 ↪ pm+gas measurement per hour

% tune the number of batteries and number of measurements to be done per
 ↪ hour
batteries = 2; % number of batteries
short_measurements = 225; % amount of measurements per hour without PM or
 ↪ gas sensor (takes 2 seconds), set to 1800 for continuous sensing.
pm_measurements = 15; % amount of measurements per hour with PM without gas
 ↪ sensor (takes 60 seconds), set to 60 to simulate continuous sensing.
gas_measurements = 0; % amount of measurements per hour without PM with gas
 ↪ sensor (takes 60 seconds), set to 60 to simulate continuous sensing.
complete_measurements = 0; % amount of measurements per hour with PM and gas
 ↪ sensor (takes 60 seconds), set to 60 to simulate continuous sensing.

% tune the algorithm
algorithm = true; % turn on the algorithm
full_battery_threshold = 0.95; % upper threshold for the algorithm
empty_battery_threshold = 0.1; % lower threshold for the algorithm
wifi_spam = false; % continuously send data over wifi above full battery
 ↪ threshold
wifi_consumption = 0.4; % consumption of wifi module if constantly sending
measuring_freq_factor = 0.1; % fraction of measurements to be done just
 ↪ above 10% battery
```

```
a = (1-measuring_freq_factor)/(full_battery_threshold-
 ↪  empty_battery_threshold); % slope of
 ↪  algorithm
b = measuring_freq_factor - a*empty_battery_threshold; % intersect of
 ↪  algorithm

battery_selfdissipation = 0.00296*batteries*percentage_per_month;
consumption = battery_selfdissipation + idle_consumption +
 ↪  short_measurements*short_consumption + pm_measurements*pm_consumption
 ↪  + gas_measurements*gas_consumption +
 ↪  complete_measurements*complete_consumption; % If the algorithm is
 ↪  turned off, consumption is constant
consumption_tracker(1) = consumption; % for a consumption plot
batt_cap_max = 8.3*batteries; % effective battery capacity in Wh
batt_cap(1) = 0.9*batt_cap_max; % battery charge at start
for i = 1:length(data(:,1))
    if algorithm == true
        if batt_cap(i) > full_battery_threshold*batt_cap_max
            if wifi_spam == true
                consumption = battery_selfdissipation + idle_consumption +
                 ↪  short_measurements*short_consumption +
                 ↪  pm_measurements*pm_consumption +
                 ↪  gas_measurements*gas_consumption +
                 ↪  complete_measurements*complete_consumption +
                 ↪  wifi_consumption;
            else
                consumption = battery_selfdissipation + idle_consumption +
                 ↪  short_measurements*short_consumption +
                 ↪  pm_measurements*pm_consumption +
                 ↪  gas_measurements*gas_consumption +
                 ↪  complete_measurements*complete_consumption;
            end
            measuring_freq = 1;
        elseif (batt_cap(i) >= empty_battery_threshold*batt_cap_max) &&
         ↪  (batt_cap(i) <= full_battery_threshold*batt_cap_max)
            measuring_freq = a*batt_cap(i)/batt_cap_max + b;
            freq_tracker(i) = measuring_freq;
            consumption = battery_selfdissipation + idle_consumption +
             ↪  short_measurements*short_consumption +
             ↪  measuring_freq*(pm_measurements*pm_consumption +
             ↪  gas_measurements*gas_consumption +
             ↪  complete_measurements*complete_consumption);
        else
            consumption = battery_selfdissipation + idle_consumption;
            measuring_freq = 0;
        end
        measuring_freq_tracker(i) = measuring_freq;
    end
    batt_cap(i+1) = batt_cap(i) + production(i) - consumption;
    if batt_cap(i+1) > batt_cap_max
        batt_cap(i+1) = batt_cap_max;
    else
        batt_cap(i+1) = batt_cap(i+1);
    end
    consumption_tracker(i+1) = consumption;
end
```

```
% make the graph more readable by taking the average of multiple hours
reduction_factor = 24;
for i = 1:(length(batt_cap)-reduction_factor)/reduction_factor
    temp_avg = 0;
    for j = 0:reduction_factor-1
        temp_avg = temp_avg + batt_cap(reduction_factor*i+j);
    end
    temp_avg = temp_avg/reduction_factor;
    for j = 0:reduction_factor-1
        avg_batt_cap(reduction_factor*i+j) = temp_avg;
    end
end

% make the graph more readable by taking the minimum of multiple hours
reduction_factor = 24;
for i = 1:(length(batt_cap)-reduction_factor)/reduction_factor
    local_minimum = batt_cap_max + 1;
    for j = 0:reduction_factor-1
        if local_minimum > batt_cap(reduction_factor*i+j)
            local_minimum = batt_cap(reduction_factor*i+j);
        end
    end
    for j = 0:reduction_factor-1
        low_batt_cap(reduction_factor*i+j) = local_minimum;
    end
end

plot(avg_batt_cap)
title(['24-hour Average Battery Charge Over 12 Years'], 'FontSize', 40)
xlim([0 length(data(:,1))])
ylim([-1 batt_cap_max])
xlabel('Time(h)', 'FontSize', 40)
ylabel('Battery charge(Wh)', 'FontSize', 40)
set(gca,'FontSize',30)


% plot(low_batt_cap)
% title(['24-hour Low Battery Capacity Over 12 Years'])
% xlim([0 length(data(:,1))])
% ylim([-1 batt_cap_max])
% xlabel('Time(h)')
% ylabel('Capacity(Wh)')

% % make the graph more readable by taking the average of multiple hours
% reduction_factor = 24;
% for i = 1:(length(consumption_tracker)-reduction_factor)/reduction_factor
%     temp_avg = 0;
%     for j = 0:reduction_factor-1
%         temp_avg = temp_avg + consumption_tracker(reduction_factor*i+j);
%     end
%     temp_avg = temp_avg/reduction_factor;
%     for j = 0:reduction_factor-1
%         consumption_tracker(reduction_factor*i+j) = temp_avg;
%     end
% end
```

```
%
% % plot power consumption
% plot(consumption_tracker)
% title('24-hour Average Power Consumption')
% ylim([0 max(consumption_tracker)+0.1])
% xlim([0 length(consumption_tracker)])
% xlabel('Time(h)')
% ylabel('Consumption(W)')

% make the graph more readable by taking the average of multiple hours
% reduction_factor = 24;
% for i =
 ↪  1:(length(measuring_freq_tracker)-reduction_factor)/reduction_factor
%     temp_avg = 0;
%     for j = 0:reduction_factor-1
%         temp_avg = temp_avg +
 ↪  measuring_freq_tracker(reduction_factor*i+j);
%     end
%     temp_avg = temp_avg/reduction_factor;
%     for j = 0:reduction_factor-1
%         measuring_freq_tracker(reduction_factor*i+j) = temp_avg;
%     end
% end

% % plot measuring frequency
% plot(measuring_freq_tracker)
% title('Relative PM Measuring Frequency')
% ylim([0 max(measuring_freq_tracker)+0.1])
% xlim([0 length(consumption_tracker)])
% xlabel('Time(h)')
% ylabel('Relative Frequency')
```

B.3. Arduino code

B.3.1. Sweep code

```
//Author: Tibbe van der Biezen
//Date: 16-6-2021

// Global variables
int V_ref; //0-255 (Mapped to 0-2.26V) V_ref > 127 voltage 'too' high,
 ↪  V_ref <127 indicates voltage 'too' low
float Isum = 0;
float Vsum = 0;
uint16_t V_in = 0;
uint16_t I_in = 0;
float Vavg = 0;
float Iavg = 0;
float Pavg = 0;
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  Serial.begin(115200);
  pinMode(A0, OUTPUT);
  pinMode(A2, INPUT); //Voltage PV
  pinMode(A4, INPUT); //Current PV
  //Initialize variables
```

```
  int V_ref = 127; //0-255 (Mapped to 0-2.26V) V_ref > 127 voltage 'too'
   ↪  high, V_ref <127 indicates voltage 'too' low
  analogReadResolution(12);
  analogWriteResolution(10);

}

// the loop function runs over and over again forever
int a = 50;
float Vconv = 235.6;
float Iconv = 1290;
void loop() {
  for(int i = 600; i>= 0; i--){
    analogWrite(A0, i);
    Vsum=0;
    Isum=0;
    for (int j = 0; j <= a; j++) {
      V_in = analogRead(A2); //0-1023*4
      I_in = analogRead(A4); //0-1023*4
      Vsum = Vsum+V_in;
      Isum = Isum+I_in;
      // delayMicroseconds(10);
      // delay(10);
    }
    Vavg = Vsum/a/Vconv;
    Iavg = Isum/a/Iconv;
    Pavg = Vavg*Iavg;


    SerialUSB.print("Panel Voltage: ");
    SerialUSB.print(Vavg);
    SerialUSB.print(" Current: ");
    SerialUSB.print(Iavg);
    SerialUSB.print(" Power_in: ");
    SerialUSB.print(Pavg);
    SerialUSB.print(" Digital DAC value: ");
    SerialUSB.println(i);
  }
}
```

B.3.2. IC - Algorithm

```
//Author: Tibbe van der Biezen
//Date: 16-6-2021

// Global variables
int control = 600; //0-255 (Mapped to 0-2.26V) V_ref > 127 voltage 'too'
 ↪  high, V_ref <127 indicates voltage 'too' low
float Isum = 0;
float Vsum = 0;
uint16_t V_in = 0;
uint16_t I_in = 0;
float Vavg = 0;
float Iavg = 0;
float Pavg = 0;
float Vavg_sweep = 0;
float Iavg_sweep = 0;
```

```
float Pavg_sweep = 0;
float Pavg_sweep_old = 0;
int control_sweep = 0;
int sample_count = 50;
float Vconv = 235.6;
float Iconv = 1290;
//MPPT Variabls
float deltaV = 0;
float deltaI = 0;
float Vavg_old = 0;
float Iavg_old = 0;
int step_size = 1;
unsigned long StartTime = micros();
//unsigned long CurrentTime = micros();
unsigned long ElapsedTime = micros();
// the setup function runs once when you press reset or power the board
void setup() {
  Serial.begin(115200);
  pinMode(A0, OUTPUT);
  pinMode(A2, INPUT); //Voltage PV
  pinMode(A4, INPUT); //Current PV
  analogReadResolution(12);
  analogWriteResolution(10);
}

// the loop function runs over and over again forever
void loop() {
  analogWrite(A0, control);
  Vsum = 0;
  Isum = 0;
  for (int j = 0; j <= sample_count; j++) {
    V_in = analogRead(A2); //0-1023*4
    I_in = analogRead(A4); //0-1023*4
    Vsum = Vsum + V_in;
    Isum = Isum + I_in;
    // delayMicroseconds(10);
    // delay(10);
  }
  Vavg = Vsum / sample_count / Vconv;
  Iavg = Isum / sample_count / Iconv;
  Pavg = Vavg * Iavg;
  ElapsedTime = micros() - StartTime;
  SerialUSB.print("Panel_Voltage:");
  SerialUSB.print(Vavg);
  SerialUSB.print(" Current:");
  SerialUSB.print(Iavg);
  SerialUSB.print(" Power_in:");
  SerialUSB.print(Pavg);
  SerialUSB.print(" Step_size:");
  SerialUSB.print(step_size);//*0.003);
  SerialUSB.print(" Digital_DAC_value:");
  SerialUSB.println(control);

  deltaI = Iavg - Iavg_old;
  deltaV = Vavg - Vavg_old;
```

```
  if (deltaI / deltaV == -1 * Iavg / Vavg) { //at MPP
    control = control;
  } else if (deltaI / deltaV > -1 * I_in / V_in) { //left of MPP -> V++ =>
   ↪  I-- => Vref-- (=> Ipv--)
    if (control < 800) {
      control = control+step_size;
    }
  } else if (deltaI / deltaV < -1 * I_in / V_in) { //right of MPP
    if (control > step_size) {
      control= control-step_size;
    }
  }
  if ((Vavg*Iavg < 1) and (Vavg*Iavg >= 0.5)){
    step_size = step_size+1;
  }else if ((Vavg*Iavg < 0.5) and (Vavg*Iavg >= 0.2)){
    step_size = step_size+5;
  }else if (Vavg*Iavg <0.2){
    step_size = step_size+10;
  }else if (Vavg*Iavg >= 1){
    if (step_size >= 100){
      step_size = step_size-10;
    } else if ((step_size <= 100) and (step_size > 50)){
      step_size = step_size-5;
    } else if ((step_size <= 50) and (step_size > 1)){
      step_size = step_size-1;
    } else {
      step_size = 1;
    }
  }
  if (step_size > 100){
    step_size = 100;
  }
  if (step_size <= 0){
    step_size = 1;
  }
//  if (control < 10){
//    SerialUSB.print("Att MPP, with time elapsed in seconds: ");
//    SerialUSB.println(ElapsedTime/1000);
//  }
//
//  //Do global sweep
//  if (ElapsedTime > 10000000){ //Every 10 seconds
//    Pavg_sweep_old = 0;
//    for (int c = 0; c <= 800; c++) {
//      analogWrite(A0, c);
//      Vsum = 0;
//      Isum = 0;
//      for (int j = 0; j <= sample_count; j++) {
//        V_in = analogRead(A2); //0-1023*4
//        I_in = analogRead(A4); //0-1023*4
//        Vsum = Vsum + V_in;
//        Isum = Isum + I_in;
//        // delayMicroseconds(10);
//        // delay(10);
//      }
//      Vavg_sweep = Vsum / sample_count / Vconv;
```

```
//      Iavg_sweep = Isum / sample_count / Iconv;
//      Pavg_sweep = Vavg * Iavg;
//      if (Pavg_sweep > Pavg_sweep_old){
//        Pavg_sweep_old = Pavg_sweep;
//        control_sweep = c;
//      }
//    }
//  }
//  SerialUSB.print("Control value out of the global sweep: ");
//  SerialUSB.print(control_sweep);
//  control = control_sweep;


  Iavg_old = Iavg;
  Vavg_old = Vavg;
}
```

Bibliography

[1] United Nations Department of Economic and Social Affairs Population Division (2019), "World urbanization prospects: The 2018 revision," *World Urbanization Prospects*, 2019. https://population.un.org/wup/Publications/Files/WUP2018-Report.pdf.

[2] F. Lindholm, J. Fossum, and E. Burgess, "Application of the superposition principle to solar-cell analysis," *IEEE Transactions on Electron Devices*, vol. 26, no. 3, pp. 165–171, 1979.

[3] S. S. CO., "Specification of product for lithium-ion rechargeable cell model : Icr18650-30a," 11 2009.

[4] T. Ozaki, T. Hirose, H. Asano, N. Kuroki, and M. Numa, "A fully-integrated, high-conversion-ratio and dual-output voltage boost converter with mppt for low-voltage energy harvesting," in *2015 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, pp. 1–4, 2015.

[5] S. Matsumoto, T. Shodai, and Y. Kanai, "A novel strategy of a control ic for boost converter with ultra low voltage input and maximum power point tracking for single solar cell application," in *2009 21st International Symposium on Power Semiconductor Devices IC's*, pp. 180–183, 2009.

[6] C. L. Espinosa, "Asynchronous non-inverter buck-boost dc to dc converter for battery charging in a solar mppt system," in *2017 IEEE URUCON*, pp. 1–4, 2017.

[7] T. Esram and P. L. Chapman, "Comparison of photovoltaic array maximum power point tracking techniques," *IEEE Transactions on Energy Conversion*, vol. 22, no. 2, pp. 439–449, 2007.

[8] A. Haque and Zaheeruddin, "Research on solar photovoltaic (pv) energy conversion system: An overview," in *Third International Conference on Computational Intelligence and Information Technology (CIIT 2013)*, pp. 605–611, 2013.

[9] M. A. G. de Brito, L. P. Sampaio, L. G. Junior, and C. A. Canesin, "Evaluation of mppt techniques for photovoltaic applications," in *2011 IEEE International Symposium on Industrial Electronics*, pp. 1039–1044, 2011.

[10] A. Devices, "Battery charging and management solutions," 10 2018.

[11] B. Talbi, F. Krim, T. Rekioua, A. Laib, and H. Feroura, "Design and hardware validation of modified p&o algorithm by fuzzy logic approach based on model predictive control for mppt of pv systems," *Journal of Renewable and Sustainable Energy*, vol. 9, no. 4, p. 043503, 2017.

[12] N. Femia, G. Petrone, G. Spagnuolo, and M. Vitelli, "Optimizing sampling rate of p o mppt technique," in *2004 IEEE 35th Annual Power Electronics Specialists Conference (IEEE Cat. No.04CH37551)*, vol. 3, pp. 1945–1949 Vol.3, 2004.

[13] N. Femia, G. Petrone, G. Spagnuolo, and M. Vitelli, "Optimizing duty-cycle perturbation of p o mppt technique," in *2004 IEEE 35th Annual Power Electronics Specialists Conference (IEEE Cat. No.04CH37551)*, vol. 3, pp. 1939–1944 Vol.3, 2004.

[14] A. Mohapatra, B. Nayak, and K. B. Mohanty, "Current based novel adaptive p o mppt algorithm for photovoltaic system considering sudden change in the irradiance," in *2014 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, pp. 1–4, 2014.

[15] R. Alik, A. Jusoh, and N. A. Shukri, "An improved perturb and observe checking algorithm mppt for photovoltaic system under partial shading condition," in *2015 IEEE Conference on Energy Conversion (CENCON)*, pp. 398–402, 2015.

[16] V. R. Kota and M. N. Bhukya, “A simple and efficient mppt scheme for pv module using 2-dimensional lookup table,” in *2016 IEEE Power and Energy Conference at Illinois (PECI)*, pp. 1–7, 2016.

[17] S. Khadidja, M. Mountassar, and B. M’hamed, “Comparative study of incremental conductance and perturb observe mppt methods for photovoltaic system,” in *2017 International Conference on Green Energy Conversion Systems (GECS)*, pp. 1–6, 2017.

[18] C.-Y. Won, D.-H. Kim, S.-C. Kim, W.-S. Kim, and H.-S. Kim, “A new maximum power point tracker of photovoltaic arrays using fuzzy controller,” in *Proceedings of 1994 Power Electronics Specialist Conference - PESC’94*, vol. 1, pp. 396–403 vol.1, 1994.

[19] T. Hiyama, S. Kouzuma, and T. Imakubo, “Identification of optimal operating point of pv modules using neural network for real time maximum power tracking control,” *IEEE Transactions on Energy Conversion*, vol. 10, no. 2, pp. 360–367, 1995.

[20] A. Trivedi, A. Gupta, R. K. Pachauri, and Y. K. Chauhan, “Comparison of perturb observe and ripple correlation control mppt algorithms for pv array,” in *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, pp. 1–5, 2016.

[21] M. Bodur and M. Ermis, “Maximum power point tracking for low power photovoltaic solar panels,” in *Proceedings of MELECON ’94. Mediterranean Electrotechnical Conference*, pp. 758–761 vol.2, 1994.

[22] D. Shmilovitz, “On the control of photovoltaic maximum power point tracker via output parameters,” *Electric Power Applications, IEE Proceedings* -, vol. 152, 03 2005.

[23] Ángel Antonio Bayod-Rújula and J.-A. Cebollero-Abián, “A novel mppt method for pv systems with irradiance measurement,” *Solar Energy*, vol. 109, pp. 95–104, 2014.

[24] R. F. Coelho, F. M. Concer, and D. C. Martins, “A mppt approach based on temperature measurements applied in pv systems,” in *2010 IEEE International Conference on Sustainable Energy Technologies (ICSET)*, pp. 1–6, 2010.

[25] A. Devices, “Power tracking 2a batterycharger for solar power,” 12 2015.

[26] T. Instruments, “Bq25798 i2c controlled, 1-4 cell, 5-a buck-boost solar battery charger with dual-input selector and mppt for small photovoltaic panel,” 12 2020.

[27] A. Devices, “35v/3.2a multicell lithium-ion step-down battery charger with powerpath and i2c telemetry,” 10 2018.

[28] STMicroelectronics, “High efficiency solar battery charger with embedded mppt,” 02 2021.

[29] R. Gogoana, M. Pinson, M. Bazant, and S. Sarma, “Internal resistance matching for parallel-connected lithium-ion cells and impacts on battery pack cycle life,” *Journal of Power Sources*, vol. 252, p. 8–13, 04 2014.

[30] S. S. Zhang, “The effect of the charging protocol on the cycle life of a li-ion battery,” *Journal of Power Sources*, vol. 161, no. 2, pp. 1385–1391, 2006.

[31] J.-n. Lou, X.-b. Wu, and M.-l. Zhao, “Switch-mode multi-power-supply li-ion battery charger with power-path management,” in *2010 10th IEEE International Conference on Solid-State and Integrated Circuit Technology*, pp. 527–529, 2010.

[32] M. Diaz-Corrada, “Dynamic power-path management and dynamic power management,” 01 2007.

[33] M. Daowd, N. Omar, P. Van Den Bossche, and J. Van Mierlo, “Passive and active battery balancing comparison based on matlab simulation,” in *2011 IEEE Vehicle Power and Propulsion Conference*, pp. 1–7, 2011.

[34] V. Stornelli, M. Muttillo, T. de Rubeis, and I. Nardi, “A new simplified five-parameter estimation method for single-diode model of photovoltaic panels,” *Energies*, vol. 12, Feb 2019.

[35] “Coolterm.” https://freeware.the-meiers.org/.

[36] “Analog discovery 2: 100ms/s usb oscilloscope, logic analyzer and variable power supply.” https://store.digilentinc.com/analog-discovery-2-100msps-usb-oscilloscope-logic-analyzer-and-variable-power-supply/.