June 21, 2019

PV system design for an RC plane

Bachelor thesis

Delft University of Technology

Sumeet Sharma
Tom Wemelsfelder
BAP group F

Project supervisor:
Olindo Isabella

Abstract

In this thesis, the design and implementation of a photovoltaic (PV) system for application in solar airplane has been developed with the objective of extending its flight time. Firstly, a literature review was conducted to study the basics of solar technology and its applications. Additionally, the code for maximum power point tracking was developed and verified. After that, a commercially available small aircraft was obtained, and the potential areas on which the solar cells can be installed were determined. Based on this, the layout of the PV system was designed and available monocrystalline PV cells were cut into the desired dimensions. For the provided area of solar cells, the I-V curves were also measured. This study showed that there is some potential to use PV cells to improve the power availability in aircraft, however, this technology currently has several limitations.

Contents

1 Introduction

Solar energy is being increasingly used in many sectors as it is clean energy which requires minimum operation and maintenance costs [1]. One of the potential fields of applications of solar energy is the aviation industry, specifically in small aircraft and monitoring devices. Many research projects have been conducted in developing solar energy based aircrafts, like Helios and Solar Impulse [2][3]. The total energy that can be harnessed by using solar panels in the aircraft may not alone be sufficient to provide enough power required for the aircraft due to several reasons including additional weights of components (battery, converter) and requirements of large surface area [4][5]. Therefore, one of the attractive options is to use solar technology as an additional source of energy for extending the flight time or range of an aircraft. PV technology offers the ability to provide an uninterrupted longer term power supply as energy can be continuously harvested during flight and stored for later use. This extension possibility of flight time has become increasingly important in various fields such as environmental monitoring and military applications in manned and unmanned aircrafts.

This BSc thesis is a part of the 'Solar Plane' bachelor graduation project. The main goal of the project was to extend the flight time of a small commercially available aircraft using photovoltaic (PV) technology. The project was split into 3 sub-groups: PV (Photovoltaic), power electronics (PE) and control. This thesis describes the design and implementation of the PV system/layout and the software used for maximum power point tracking (MPPT).

This thesis consists of 7 chapters. The first chapter introduces the subject and presents the main goal as well as the outline of the thesis. In chapter 2, the programme of requirements is elaborated. Chapter 3 focusses on the conducted background information and research. Chapters 4 and 5 are devoted to the design process and the prototype development respectively. In chapter 6 the obtained results are discussed while the key conclusions and recommendations for future work are in chapter 7.

2 Programme of requirements

Examples of requirements (incomplete):

- Control the power transfer between the PV array, the battery pack and the load. Always work towards generating maximum solar power, as long as this is safe.
- The battery must never be overcharged or undercharged
- The battery must never receive or output overcurrent
- The efficiency of the MPPT must be larger than 95%
- The PV cells should not prevent the plane from being easily disassembled
- The PV cells and MPPT algorithm must be maintenance free during flight
- The MPPT must have stabilized within 1 second
- Oscillations of the duty cycle around the MPP must not exceed 1%
- The microcontroller performing MPPT must log the measurements done by the voltage and current sensor
- Duty cycle must never exceed 0.7 as this could break the DC/DC converter.

3 Background information and research

3.1 Basics of solar cells

A solar cell uses energy from light to generate electric power. It can be modeled with the equivalent circuit seen in figure 1. The solar cell produces a DC current that is dependent on the light intensity or solar irradiance. The solar cell thus behaves like a current source. However, because solar cells are implemented as a PN-junction, not all power is delivered to the load. Thus, a diode is added in forward-active mode in the model to model this characteristic. A series resistance (R_s) and shunt resistance (R_{sh}) are added to model additional losses within the PV cell (contact resistances, etc.) By measuring the output voltage while varying the output current, an I-V curve like figure 2 can be constructed. Using this I-V curve we can calculate the power delivered by the solar cell. This is denoted by the P-V curve in the same figure. This P-V curve clearly shows that the power reaches a maximum point at a certain voltage. This point is called the Maximum Power Point (MPP) and the corresponding power, voltage and current are called P_{mp}, V_{mp} and I_{mp} respectively. Other important operating points are the V_{oc} and the I_{sc}. These denote the voltage at open circuit and the current at short circuit respectively. A metric often used to determine the efficiency of a solar cell is the fill factor (FF). The FF can be calculated using equation 1.

$$FF = \frac{V_{mp} * I_{mp}}{V_{oc} * I_{sc}} \tag{1}$$

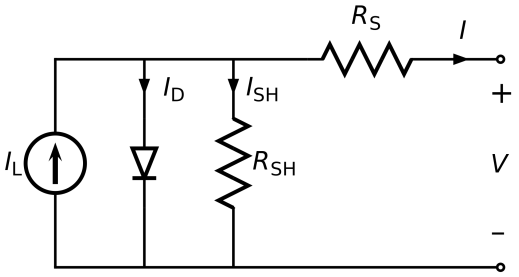

Figure 1: Equivalent circuit of a solar cell

The current generated by the PV cell can be found using the equation [6]:

$$I_{pv} = (I_{pv\_n} + K_l * dT) * G/G_n \tag{2}$$

where

- I_{pv} is generated current
- $I_{pv,n}$ is the generated current under test conditions (T = 25 degrees celsius, G = 1000 W/m^2
- K_l is a constant
- dT is the difference between the (outside) air temperature and room temperature (25 degrees celsius)
- G is the solar irradiance
- G_n is the nominal radiation (1000 W/m^2)

If we assume dT to be negligible, the equation shows that the current delivered (I_{pv}) linearly depends on the solar irradiance. This information can be used to estimate the I-V and P-V characteristics while flying outside based on the measurements conducted inside.

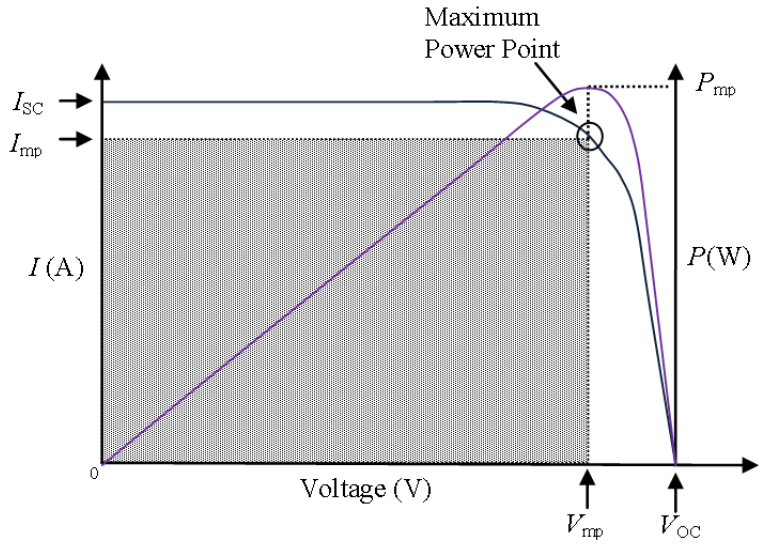

Figure 2: I-V curve of a solar cell

3.2 Maximum power point tracking

Maximum power point tracking (MPPT) is a process which aims to allow a solar cell to output as much power as possible. It can be implemented using two parts: a microcontroller to keep track of the MPP and a DC-DC converter that shifts the operating point of the PV cells by changing the effective load seen by the solar cell.
By controlling the solar cell in this way, the maximum power point (MPP) - as seen in figure 2 - can be reached. This point is dependant on multiple factors such as the type of solar cell, its interconnection with other cells, incident light and partial shading conditions.
An example of such a DC-DC converter is a boost converter. It is depicted in figure 3. This type of converter can theoretically set the ratio of V_{out}/V_{in} anywhere from 1 to ∞. This ratio is controlled by flipping the switch (denoted by S) using a Pulse Width Modulation (PWM) signal. The duty cycle of this PWM signal denotes the ratio of V_{out}/V_{in}. In case of a boost converter, the relationship between this ratio and the duty cycle is given by equation 3.

$$\frac{V_{out}}{V_{in}} = \frac{1}{1-D} \tag{3}$$

Where D ($0 \leq D \leq 1$) is the duty cycle. The MPPT is the system that will control this duty cycle, based on readings from voltage and/or current sensors. There are many different MPPT algorithms, all with their advantages and disadvantages. A comparison will be made in section 4.1.1, after which the most applicable method is chosen.

3.3 Comparison of solar cells

There exist a couple of PV technologies such as c-Si (crystalline silicon), GaAs (galium arsenide) and Copper Indium Gallium Selenide (CIGS). Since one of the main goals of the project is to extend the flight-time of an aircraft, the added weight of these PV technologies play a crucial role in determining which PV technology would be best. Additionally, the available surface area for installing PV panels is limited to certain relatively flat sections of the plane. Therefore it would be advantageous to use a PV technology with a high power generation per unit area. Figure 4 compares a number of PV technologies based on their power-to-surface-area and power-to-mass ratios. From this figure it can clearly be seen that thin film GaAs would be the best PV technology for this purpose [7]. Of course, there are also other factors that are important such as weather sensitivity and price. A more in-depth

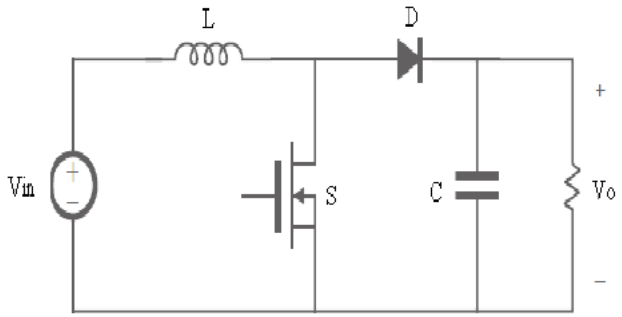

Figure 3: Circuit diagram of a boost converter

search showed us that while thin film GaAs may be the optimal solution, its use is fairly limited to certain sectors as their cost is extremely high compared to other types of PV technology.

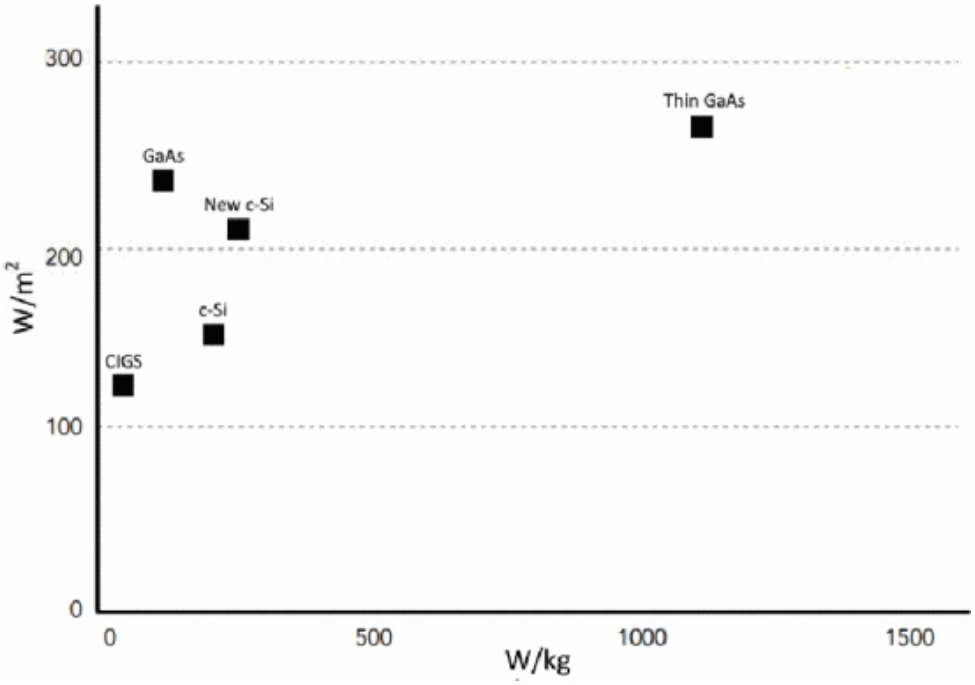

Figure 4: Comparison of PV technologies

3.4 External influences on solar irradiance

Solar cells need sunlight directly from the sun, from the sky, or reflected sunlight to be able to deliver power. A solar cell will generate the most power when the angle of incidence of sunlight is orthogonal to the cell area. As the sun changes its position through the day, this means that the power generated by the solar cells will also vary. Because the Earth's axis of rotation is tilted at 23,5 degrees, the time of the year also influences solar irradiance. Additionally, latitude also has an impact on the solar irradiance: the closer you get to the arctic, the higher the variations and the lower the average solar irradiance. As expected, weather conditions also play a role in the performance of PV cells. For example, clouds may obstruct the sun and reduce the solar irradiance incident on the cell. Based on this information we can conclude that to improve the performance of the PV system, flying above

the height of clouds would be preferable. However, clouds may also have a positive impact on the performance: When a cell is in direct sunlight from a hole between clouds it receives that sunlight as well as some light reflected from the clouds [8].

While having solar irradiance that is as high as possible may seem preferable, this is not always the case. An important thing to notice is that the temperature of a PV cell increases when it absorbs sunlight. This may cause the I-V curve of the PV cell to deviate. Because solar cells contain a lot of silicon, and silicon is an amazing heat conductor, this heating process is sped up. The temperature specifications/coefficients for a mono-crystalline solar cell can bee seen table 1. As explained later in section 4.2.1, partial shading may result in hot spots which raise the temperature of a PV cell even more.

There are multiple methods to decrease the influence of temperature on the PV cell. Examples of passive cooling methods are ensuring proper ventilation or ensuring that the color of the background is light (e.g. white or lightly colored roofs reflect sunlight more which causes the solar panels on it to heat up less). Examples of active cooling methods are blowing fans or circulating water. In our case, the plane will be flying through the sky. Depending on the weather conditions (solar radiance/ cloudiness and wind speed) the PV cells could heat up or cool off. Additionally, if you were to fly the aircraft in higher altitudes the ambient temperature would cool down the PV cells and the I-V characteristics would also be influenced.

Table 1: Temperature coefficients

Current (Alpha)	0.043 percent/ degrees celsius
Voltage (Beta)	-0.317 percent/ degrees celsius
Power (Gamma)	-0.439 percent/ degrees celsius

4 Design Process

4.1 MPPT and charge controller

4.1.1 Comparison of MPPT algorithms

There are multiple methods to track the MPP of a PV system. This section provides a brief overview of the different methods.

P&O (Perturb & Observe) is the most commonly used algorithm for MPPT. In this algorithm, the duty cycle of the converter is perturbed, causing the operating point on the I-V curve to shift. The voltage and current are then measured and used to calculate the power generated by the PV system. This generated power and measured voltage are compared to the values measured/calculated in the previous cycle and the duty cycle D is then increased/decreased accordingly. A basic diagram of the P&O algorithm can be seen in figure 5. The major disadvantage of the P&O algorithm is that the performance is influenced by the step size; a small step size may result in longer delays in e.g. quickly changing solar irradiance levels, which may result in losing sight of the MPP. A large step size, on the other hand, may result in large oscillations around the MPP.

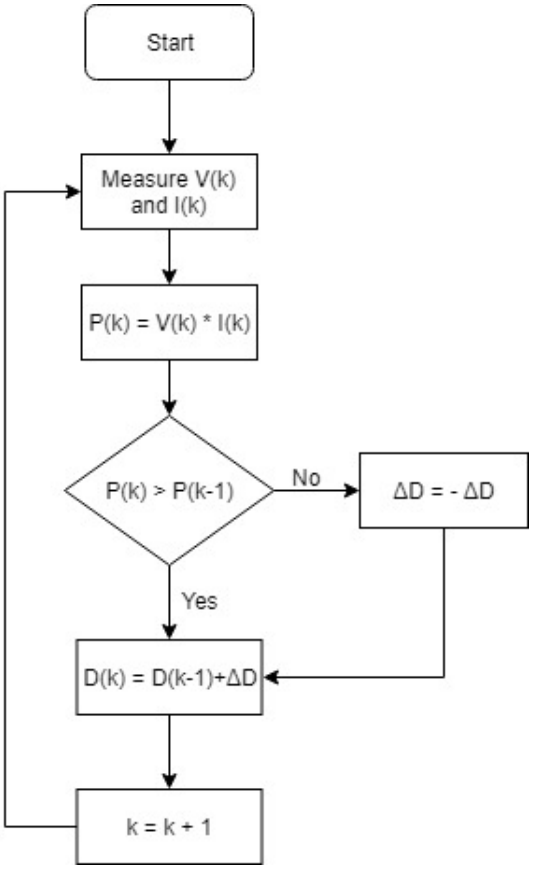

Figure 5: Flow chart of the P&O algorithm

Incremental Conductance (IncCon) is another method to implement MPPT. It makes use of the fact that the general shape of the I-V curve is known. By comparing $\frac{dI}{dV}$ with $\frac{I}{V}$ it determines its next duty cycle. A basic diagram of the IncCond algorithm can be seen in figure 6. The major advantages are that the IncCon algorithm is very good for quickly changing solar irradiances while being less susceptible to oscillations compared to P&O [9]. A disadvantage is that it is more vulenerable to

noise.

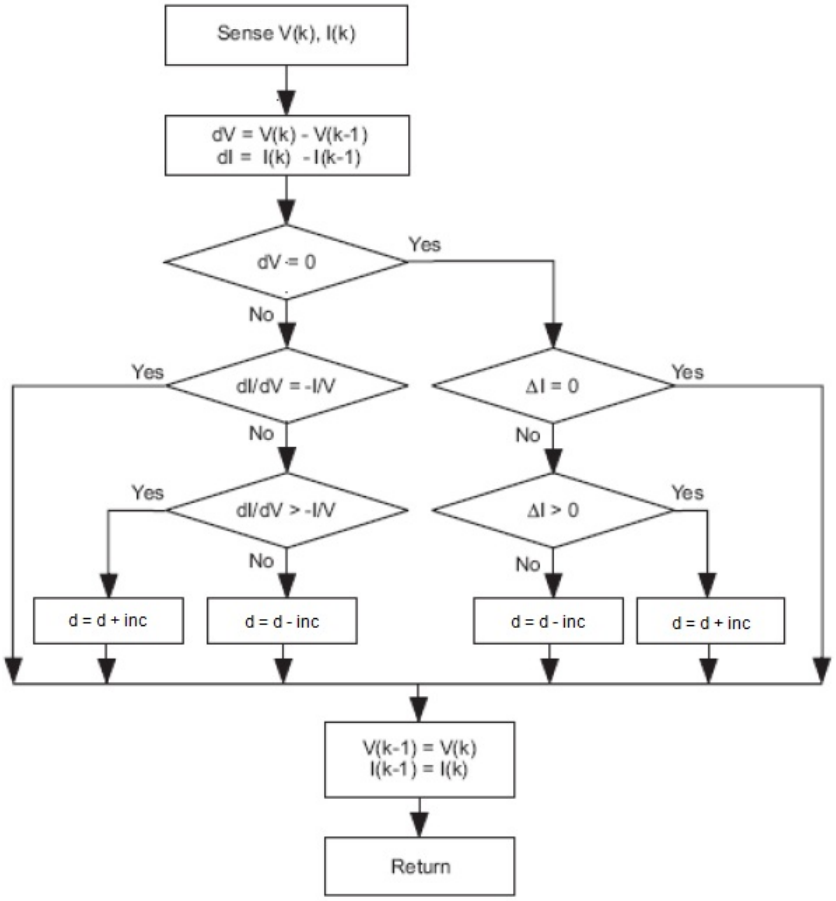

Figure 6: Flow chart of the IncCond algorithm

Fractional voltage/Open-circuit voltage is another method to determine the MPPT. However, unlike the previously mentioned algorithms, this method does not provide any feedback at all. The fractional voltage method is based on the fact that V_{mp} is approximately located at the same fraction of V_{oc}. This method thus requires the PV array to be connected in open circuit in a repeating time interval. To increase accuracy under changing irradiances/temperatures this repeating time should be as short as possble. This however has the disadvantage that the circuit is repeatedly disconnected, resulting in less power being transferred from the converter.

One of the major disadvantages of the tracking methods mentioned above is that there is no way to accurately determine whether the observed maximum is a local or a global maximum. A couple of methods exist to determine the global maximum. One such method is the current sweep method. The current sweep method is performed by varying the current delivered by the PV array from open circuit to short circuit. To keep the system operating at MPP, this current sweep is repeated at a fixed time interval. The main disadvantage of this method is that the system is not operating at MPP while the current sweep is performed.

Out of the methods mentioned above, the decision was made to implement P&O. This is because this method is the simplest method to implement while being less susceptible to noise and not requiring a reset at a repeated time interval. However, the problems related to performance vs step size still hold. To reduce this problem, an option would be to make the system adaptive: use a large step size when far from the MPP, and reduce the step size around the MPP. This way the MPP would be

reached more quickly, while the oscillations around it are smaller.

4.1.2 Simulink simulation

Firstly, a basic simulation of the entire system was made in SIMULINK. The complete model can be seen in figure 7. The PV cells were modelled using the standard MATLAB subsystem called 'PV array'. A variable irradiance was set as an input to the model. The results can be seen in figure 8. As expected the algorithm was closely oscillating around the MPP.

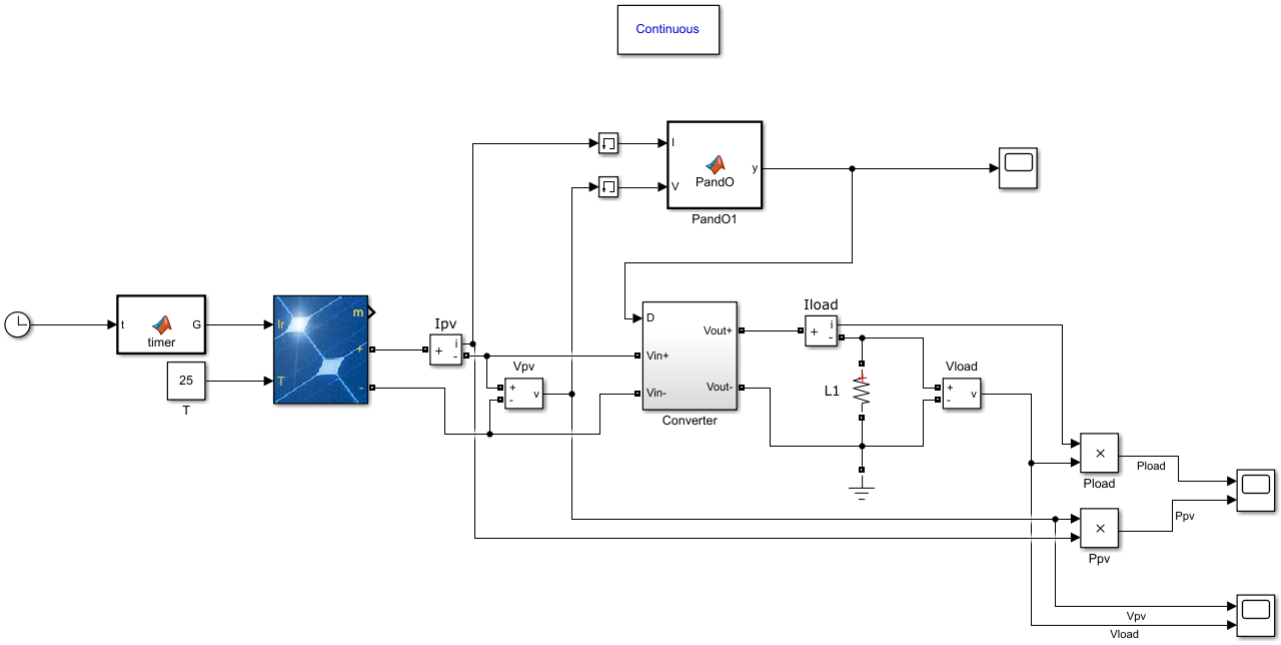

Figure 7: Model of the system in Simulink

4.1.3 Design

To be able to design the MPPT and charge controller (CC), it is important to first know their interconnections with the rest of the system. An illustration of the electrical system layout is given in figure 9. Here, the blue lines indicate switch control signals, and the yellow lines indicate communication between the MPPT and CC. The idea is to program the MPPTs and the CC on the same microcontroller, so their communication will not pose many problems.
The CC controls the switch to the battery pack. This way, the batteries can be disconnected from the rest of the system in case they are over- or undercharging, or in case of overcurrent.

As stated in the Programme of requirements (section 2), the main function of the MPPT and charge controller combination is to: control the power transfer between the PV array, the battery pack and the load. It should always work towards generating maximum solar power, as long as this is safe. With this main requirement in mind, a decision chart was created for the MPPT/CC system, as shown in figure 10. The CC will make the decisions, which then have their effect on the way the MPPT unit should behave. In the figure, 'pv' refers to the photovoltaics/solar cells; 'SoC' stands for State of Charge of the battery pack and 'bat' refers to the battery pack as well. The subscripts 'min' and 'max' refer to minimum and maximum allowed values. 'Danger' refers to dangerously high values that

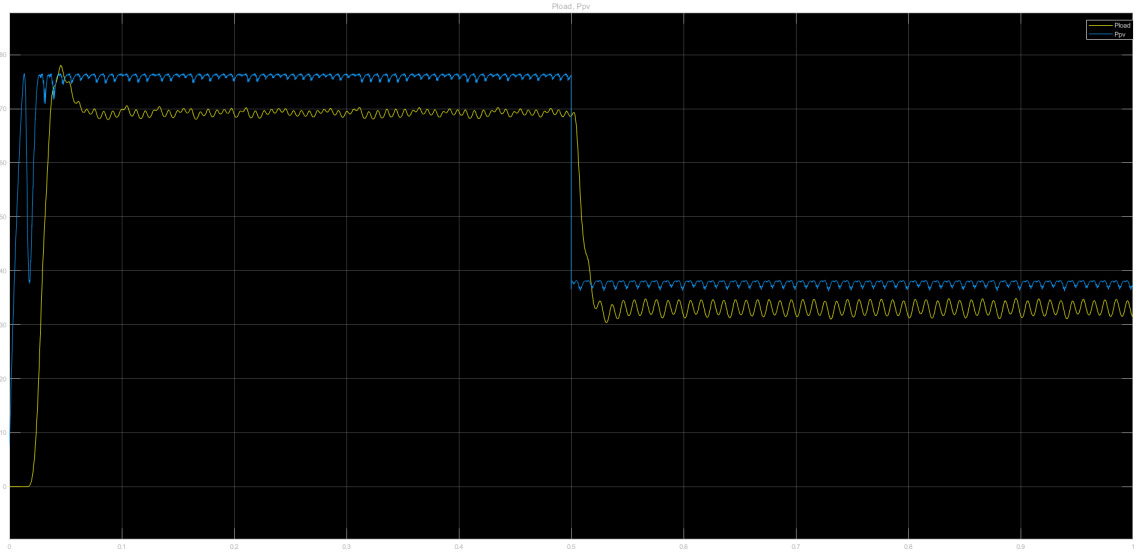

Figure 8: Graph of the simulation results

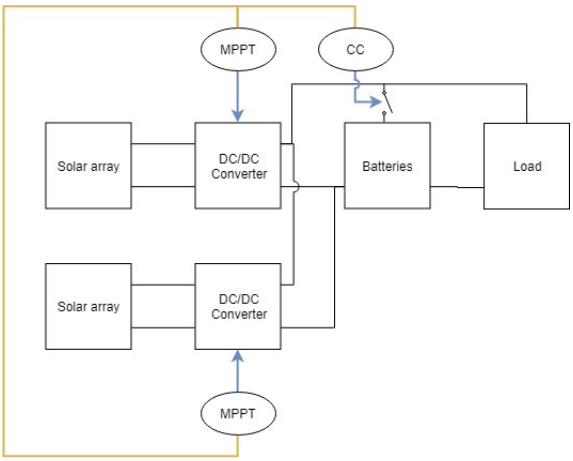

Figure 9: Layout of the electrical system

should technically not arise. They are however implemented in the decision tree because emergency measures need to be taken in case they do arise.
By 'Track P_{load}' and 'Track $P_{load} + V_{bat} * I_{batMax}$' is meant that the MPPT unit should apply the P&O algorithm to track said power, instead of the maximum power point. This is required because in both cases the battery can not handle the amount of solar power, as the battery is either fully charged or it is being charged with a current that is too high. By tracking this lower power point, the solar cells can supply the maximum amount of power that is deemed safe in these situations.

Having decided on the Perturb and Observe algorithm, requirements for the MPPT and CC can now be set. A C++ class will be created for the MPPT. Its requirements are:

- Class arguments to set the sensor pins and PWM output pin
- A function executing the P&O algorithm
  standard: track MPP ; optional: track lower power point

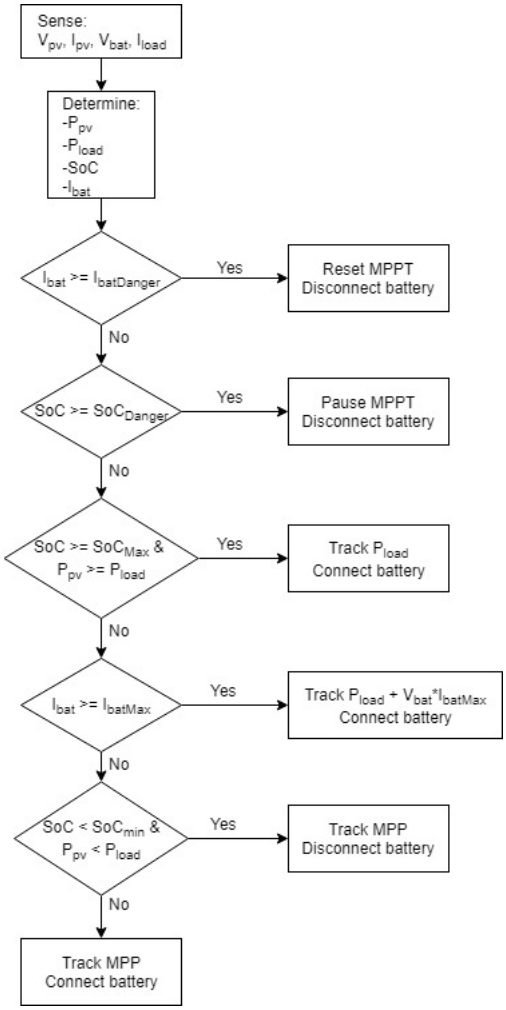

Figure 10: Decision chart of the CC in combination with MPPT

- A function to pause the MPPT algorithm (saving the duty cycle)
- A function to reset the MPPT (set duty cycle to zero, and stop algorithm)
- A function to allow the solar power to be read

Referencing to the Programme of requirements, the quantitative requirements and restrictions of the MPPT are:

- Oscillations of the duty cycle around the MPP must not exceed 1%
- The efficiency of the MPPT must be larger than 95%
- The MPPT must have a settling time within 1 second
- The duty cycle must never exceed 0.7

Another class must be created for the charge controller. Its requirements are:

- Class arguments to set sensor and switch control pins
- A function to run the charge control algorithm

- Must be able to read the load current and battery voltage
- A function to read the control signal
- A function to read the target power
- Must be able to receive data on the produced solar power

Referencing to the Programme of requirements, the quantitative requirements and restrictions of the charge controller are:

- The battery must never be overcharged or undercharged
- The battery must never receive or output overcurrent

These two classes will be imported into the main cpp file, which must control their intercommunication and execute their member functions. Looking back at figure 10, the charge controller will make all the decisions based on the sensor readings. It must then send an appropriate control signal to the main code. A communication schematic can be established as shown in figure 11.

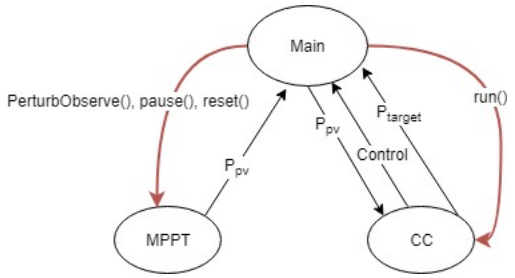

Figure 11: Internal communication schematic of the microcontroller

4.1.4 Programming environment

The code for the MPPT and charge controller (CC) will have to be programmed on a microcontroller. It will receive analog sensor readings as input, and use PWM an digital switch control signals as output. The type of microcontroller has been decided on by the Control Team: the STM32 NUCLEO-F303K8 [10].
This microcontroller supports three programming languages: Arduino, mbed and STM32Cube. Arduino was considered first, but this option was abandoned when it turned out Arduino does not provide good support for changing the PWM frequency. The standard PWM frequency is set to 50Hz. This is too low for controlling a DC-DC converter switch, as the Power Electronics team set the required switching frequency in the range of 100 - 500 kHz. This frequency is easily set using mbed, which also provides easy implementation of digital and analog in- and output ports. That is why the choice was made to program the MPPT and CC in mbed.

4.1.5 Voltage and current sensors and their inaccuracies

For the MPPT controller to determine the next duty cycle to achieve approach the MPP, the voltage and current delivered by the solar module in the previous sample has to be compared to the voltage and current delivered measured in the new sample. For this purpose, a current sensor and voltage sensor were designed by the PE team. The main challenges during this design were to accurately measure the voltages and current while keeping their power dissipation and their influence on the rest of the system as low as possible. The sensors will both output a voltage to an analog input port of the microcontroller. This voltage is assumed to be linear with the sensor readings. However, it should be normalized to stay within the bounds of 0 - 3.3V as 3.3V is considered as a digital 1 by the microcontroller. These

scaling factors should be implemented in the code to retrieve the appropriate sensor reading from a certain voltage at an analog input port. Furthermore, the sensors and microcontroller could also have systematic errors. these could either be proportional to the input signal, or it could be a bias; a constant deviation from the actual value. By doing measurements on the sensors, the system can be calibrated and so the systematic errors can be mitigated. These measurements will be described in section 5.3.2.

4.2 PV layout

4.2.1 Introduction to PV layout

The PV cells of different shapes used in the prototype solar plane were cut from crystalline 15,6 x 15,6 cm PV cells. In these cells, the current caused by the irradiance is approximately proportional to the surface area of the cell. This means that, for example, cutting a cell in half would result in I_mp being twice as small and V_mp staying constant. The power delivered by the two cut cells combined would thus equate to the power delivered by the original uncut cell. Also, this would mean that a cut cell could be modelled as a larger cell where the solar radiance is multiplied by a scaling factor.

The placement of PV cells in parallel produces a larger output current whereas the placement of PV cells in series produces a larger output voltage. All in all, under uniform radiance and constant temperature (ideal conditions), the power available that can be harvested would always be a constant. In practice, however, the radiance may not be uniform and slight temperature variations due to e.g. heating may be present. Because the solar plane is mostly going to be used in open areas or at reasonably high altitudes (above buildings/trees) it would be safe to assume that the radiance on the solar cells on the wings of the solar plane is approximately uniform.

In case of non-uniform shading, the effects would drastically depend on the topology. If there was partial shading in a series connection of PV cells, the current flowing through the solar cells would be limited by the current produced by the solar cell with the lowest solar radiance. To prevent this from happening, a bypass diode is often placed in parallel. This bypass diode 'bypasses' the connection with the lower current [11]. This however, may result in multiple local maximums in the I-V curve. Since the P&O algorithm may settle around a local maximum, this would be highly undesirable.

The diodes usually used as bypass diodes are Schottky barrier diodes. The reason they are most commonly used compared to P-N junction diodes are because they have a lower forward voltage drop (0.4 V instead of 0.6 V). This results in less power loss when a current is flowing through them during partial shading.

Although bypass diodes improve the performance of PV modules, having one parallel to each PV cell is unneccessary because of the additional space and costs it requires. Generally, a bypass diode are only placed at points where a local groups voltage drop exceeds an individual solar cells reverse breakdown threshold voltage (typically around 25 to 30V). Since our plane does not go up to that voltage, bypass diodes would are not a requirement. However, if a cell breaks during flight or a crash the entire series connection would fail. To reduce the chance of an entire series connection from failing it would be advisable to have several bypass diodes.

An even better alternative has been developed by certain manufacturers. 'Active bypass diodes' generally consist of a FET-based switching transistor and have a forward voltage drop which generally almost 20 times as small. This results in much less power dissipation in bypass circuitry.

4.2.2 PV layout design (version 1)

The maximum voltage (V_{oc}) that can be produced by one cell is generally around 0.6 V. One of the preferences of the PE group was to have a voltage above 5 V, since the efficiency of the boost converter

is greatly reduced for voltage levels lower than that. This would mean that we would want as many cells in series as possible. The plane itself consists of a lot of different 'compartments' of different sizes. Placing PV panels of different sizes in series would result in current restrictions as mentioned in the previous subsection. This would mean that there is a compromise to make. Possible solutions we thought of were:

- Using certain fixed sizes. This however would mean that some compartments may have space left over (where there aren't any PV cells) and would mean that a lot of power that may have been yielded gets lost.
- Splitting up each 'compartment' in a multitude of series connected smaller PV cells. This, however, would result in a lot of work to connect each PV mini cell properly which is undesirable. On top of that, cutting very small sections of PV cells may result in other unaccounted effects such as V_{oc} being slightly reduced.

Another design consideration is that the to be connected PV cells must be physically close to each other. The longer the distance between the PV cells, the more influence the resistance of the wires between the PV cells has on the I-V characteristics of the PV cells. The effect of this resistance can be modelled by increasing the series resistance R_s in the PV cell model described in subsection 3.1. As expected, the maximum power that can be delivered by the PV cell is reduced when the series resistance increases.

Figure 12 shows the first layout we came up with. Each group highlighted there consists of 6 compartments of roughly the same size which are connected in series. Each colored block on each wing was then also connected in series with the colored block on the other wing. This was done to be as close to the nominal battery voltage as possible: 11.1 V. As V_{mpp} of a single solar cell is around 0.49 V, a 12 series connection has a V_{mpp} of roughly 5.88 V. Since almost all of these modules have compartments placed right next to each other, a series connection would not be difficult to implement. In this layout, we used roughly 94 percent of the surface area of the compartments of the plane.

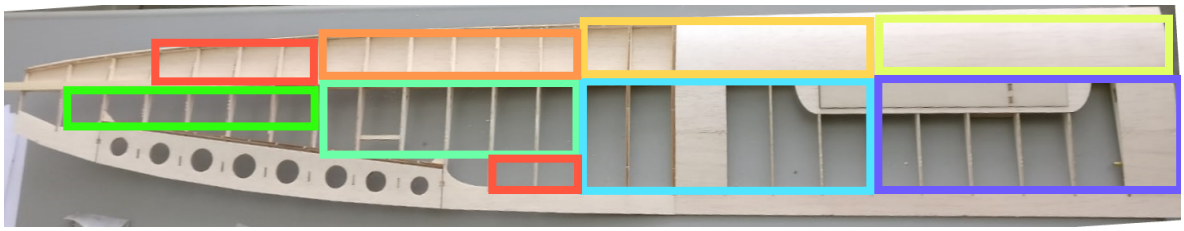

Figure 12: Layout of PV cells (version 1)

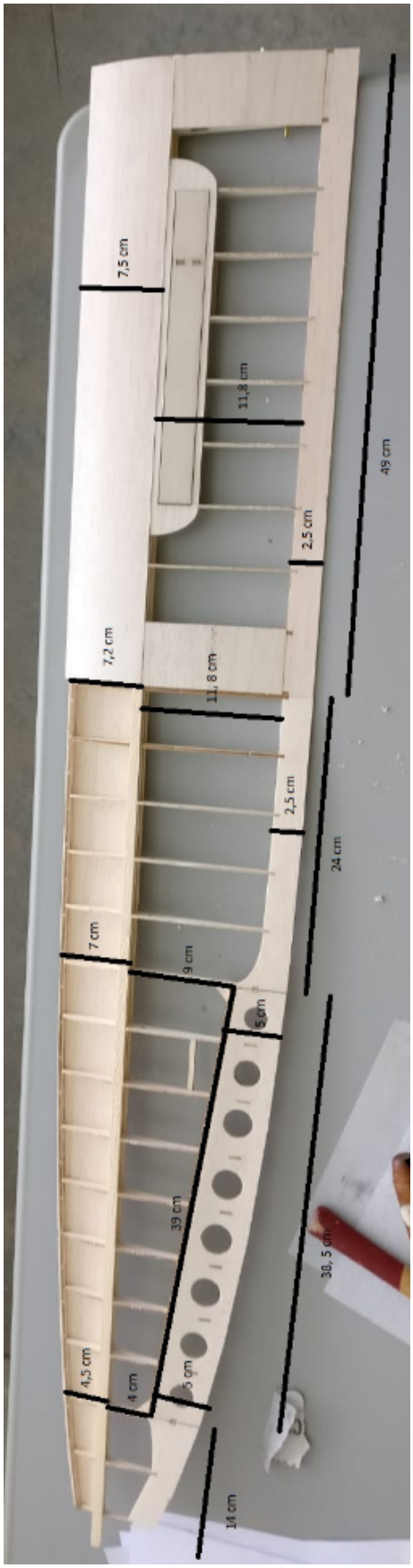

Figure 13: Measurements of the wing

4.2.3 PV layout design (version 2)

The previously designed PV layout has a couple of fallacies. First of all, the series connection between the two wings would result in the complete system failing in case a wing/the connection between the wing and the main body fails. Thus, for safety and failure-proof reasons, it would be more ideal to have the wings connected to each other in parallel, as this would prevent a complete system failure in such case. There were two main reasons for using these compartments: First of all this way we would minimize the effects on the aerodynamic properties of the plane. Secondly, the type of solar cells we were using were very brittle, and not flexible at all, which would make it impossible for them to follow the curvature of the wings. This is why at first we wanted to place everything within the wings instead of on top. Another issue was the long wires required to connect the compartments in the different wings in series with each other.

At the Greenlight Assessment we discovered that more flexible cells were available. This meant they could be folded over the wings, to minimally influence the aerodynamics. Another thing we had not taken into account was extra spacing for the connections. As the number of pieces of cells was relatively large, a lot of area would be wasted for connecting. Thus, we came up with several new layouts that were no longer restricted to compartments, consequently covering a much larger surface area and thus having more available output power.

During the design process we realized that the components on the plane would also operate at a lower voltage level. This led to us getting a battery with a nominal voltage of 7.4 V.

Firstly we measured the relevant dimensions of the wing. These can be seen in figure 13. One of the requirements for the PV layout was that it should not prevent the wings from being disassembled for ease of transportation.This meant that we could not place PV cells accross the joint. Another restriction was the size of the PV cells; as the cells used were 15,6 x 15,6 cm that set the maximum dimensions of a single cell.

One of the layouts we came up with can be seen in figure 14 (dimensions of aircraft and cells are not according to scale). In this layout, on each wing, 11 solar cells with the same area are connected in series, resulting in a V_{mpp} of $0.49 * 11 = 5.39V$. 9 of those cells are rectangular with dimensions of 15,6 x 10 cm. The remaining 2 pieces are trapezoidal. The main advantage of this layout was that it only required simple wiring/connections. On top of that the rectangular pieces have the exact same dimensions, making the cutting process for them very easy.
Another option we considered was to cut up the solar cells into pieces of 15,6 cm by 6,8 cm. As the width of the inner part of the wings is roughly 21 cm, where then angle of elevation is positive for 14 cm and then negative for 7 cm, this would result in less strain on the solar cells. This layout was later omitted as having 11 of these cells on the plane covered far less area.

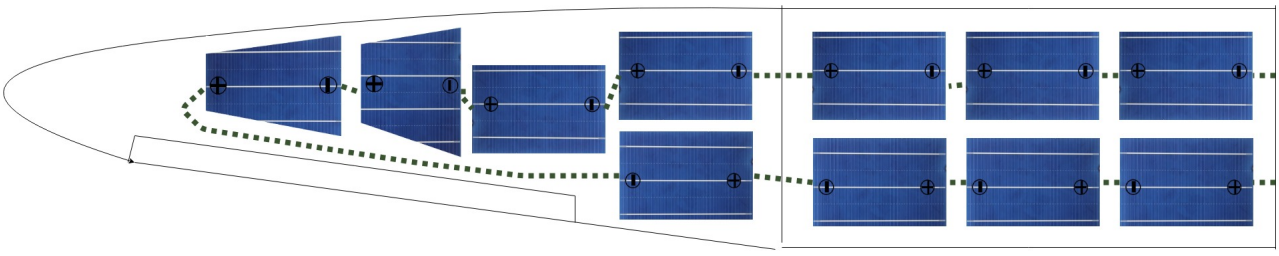

Figure 14: Layout of PV cells (version 2)

The next step was to accurately determine the dimensions for the trapezoidal PV cells. This was done by treating the curved edges of the wing as a slope (see figure 15) and setting up an equation that

specifies the width of the wing as a function of the distance from the joint (equation 4). Since we didn't want the current to be limited by the other cells in series, the areas of the trapezoidal sections had to be the same as the rectangular sections: $156cm^2$ (15,6 cm by 10 cm).

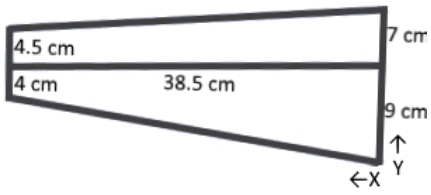

Figure 15: Measurements of the trapezoidal part of the wing

$$y = (9+7) - (\frac{9-4}{38.5} + \frac{7-4.5}{38.5})x = 16 - (\frac{5}{38.5} + \frac{2.5}{38.5})x = 16 - \frac{15}{77}x \tag{4}$$

Equation 4 can now be integrated and set equal to 156 cm^2 to get the required lenght of the trapezoidal cells. In figure 14 it can be seen that a rectangular cell partially occupies the trapezoidal part of the wing. To leave space for this cell, the first trapezoidal cell is started at X = 8 cm.

$$\int_8^{d1} 16 - \frac{15}{77}x\,dx = 156 \tag{5}$$

Equation 5 gives $d1 = 19.73cm$, which sets the length of the cell to $19.73 - 8 = 11.73cm$. Equation 4 is then used again to calculate the other dimensions of the cell. This results in the shape shown in figure 16.

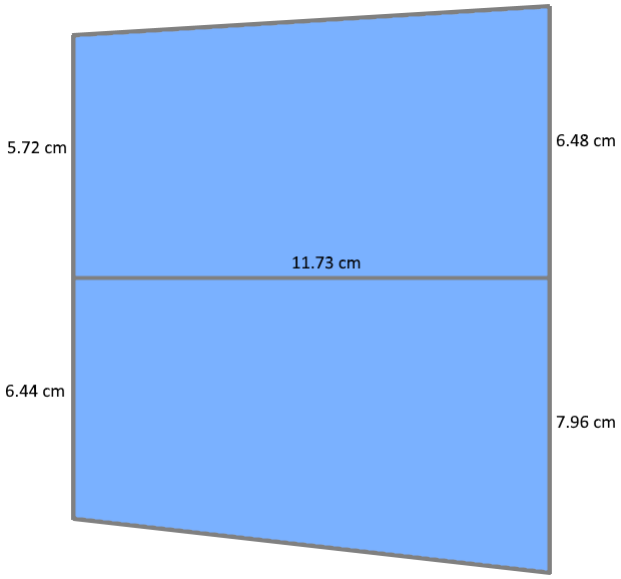

Figure 16: Measurements of the first trapezoidal cell

For the second cell we can evaluate the integral in equation 6. Because the cell of figure 16 ends at d = 19.73 cm, the second cell is started at d = 20 cm to leave some space between them.

$$\int_{20}^{d2} 16 - \frac{15}{77}x\,dx = 156 \tag{6}$$

which leads to $d2 = 34.61cm$. This means the length of this cell has to become $34.61 - 20 = 14.61cm$. Equation 4 is then used to calculate the other dimensions of the cell. The resulting shape can be seen in figure 17.

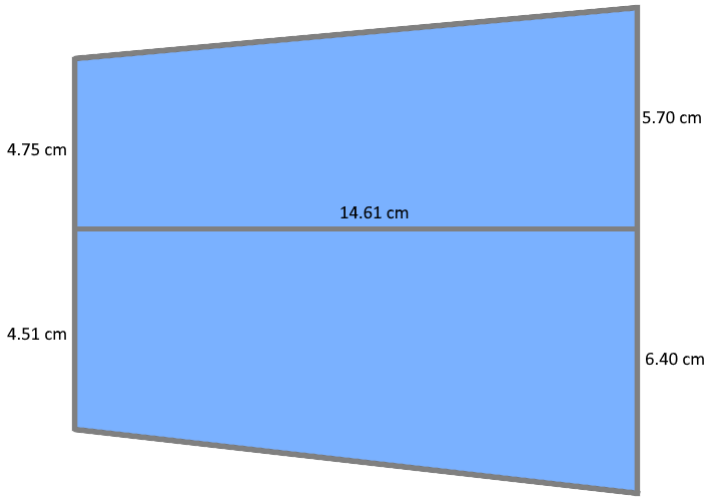

Figure 17: Measurements of the second trapezoidal cell

5 Prototype Implementation and Results

5.1 Cutting and testing of the solar cells

The solar cells had to be cut individually from cells of 15,6 x 15,6 cm using a laser cutter. While testing the cutting process using dummy cells, we found that the cells were very difficult to break properly. This resulted in us changing the parameters a bit. Although we did find the cells easier to snap, we later found out that the original parameters would also have worked. The reason that the cells kept breaking with the original parameters is that the lasering resulted in pressure being exerted on the very light solar cells, causing them to slightly bend. This resulted in the laser no longer cutting in the exact same spot. By placing weights near the edges of the cutting line, this problem was no longer present.

5.1.1 Solar cell measurements

To test if the cells were working properly, their I-V curve had to be measured. The expected result is that up to a certain threshold voltage, a cell will output a constant current equal to the short circuit current (I_{sc}). After this point, the current will drop until it reaches zero at the open circuit voltage (V_{OC}). To get a good idea of the performance of a solar cell, a fill factor can be established according to equation 7. where I_{mpp},V_{mpp} and P_m are respectively the current, voltage and power at the maximum power point. Typical fill factors range from 50% to 80% ; the higher the fill factor the better the performance.

$$FF = \frac{I_{mpp} * V_{mpp}}{I_{sc} * V_{oc}} = \frac{P_m}{I_{sc} * V_{oc}} \tag{7}$$

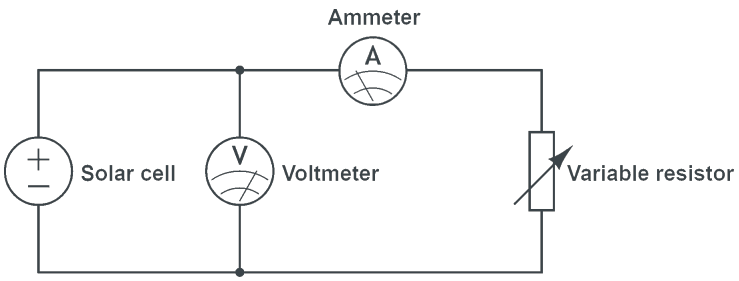

Figure 18: IV curve setup

To measure the IV curve, the setup in figure 18 was used. By illuminating the solar cell under test with a constant illumination, a reliable measurement can be done. To create a representative image, a matlab script was written to create a least-squares curve fit for the measured data. This matlab script is added in Appendix A.

The first measurements were done using a lamp with adjustable light intensity and a box containing a potentiometer. The results of one of these measurements is shown in figure 19.

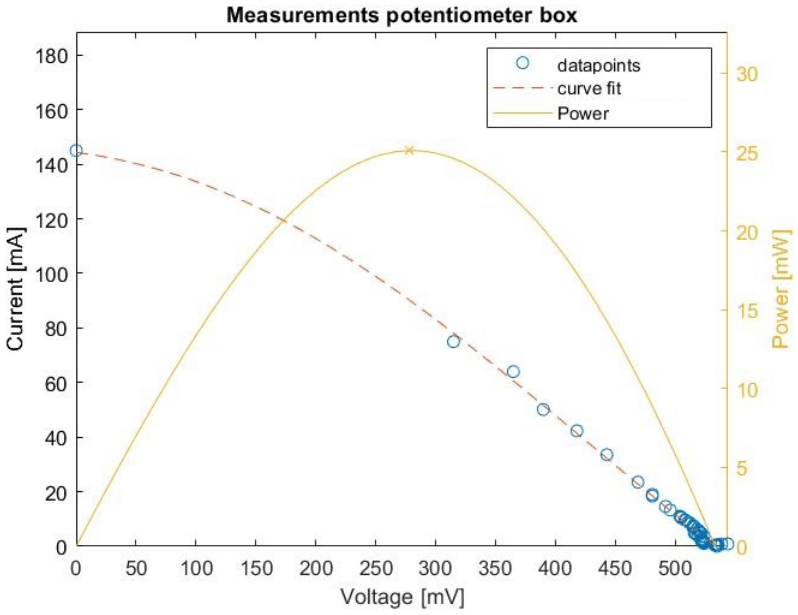

Figure 19: IV curve measurement using the potentiometer box

This result is not as expected, as the output current quickly drops when increasing the voltage. The fill factor of this cell can be determined as $FF = \frac{25m}{544m*144m} = 32\%$. This is clearly below the typical range.
To make sure the measurement setup was correct, a measurement was done on a small commercially available pre-laminated solar array. The results of this small cell can be seen in figure 20.

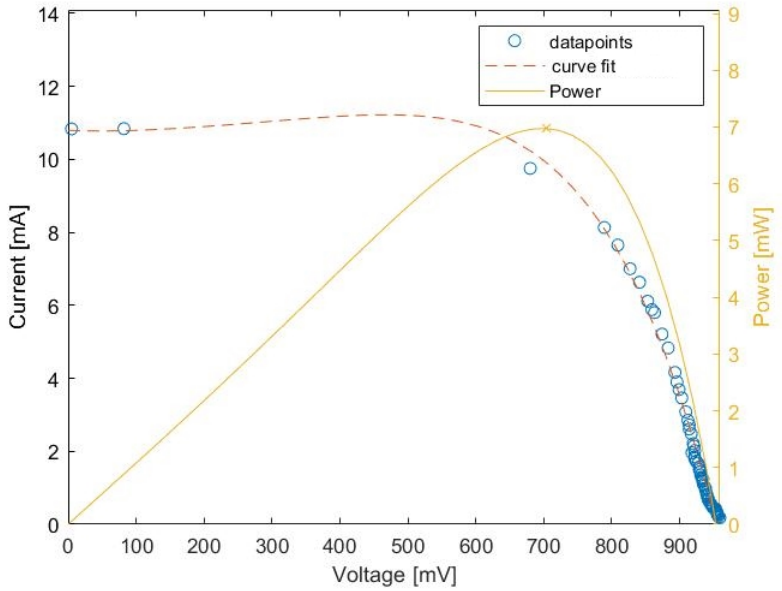

Figure 20: IV curve of own cell, using the potentiometer box

The fill factor of this cell can be determined as $FF = \frac{7.0m}{959m*10.8m} = 67\%$. This is much better and within the typical range. So the measurement setup seemed to be working properly.

There were two main hypotheses on why the cell from figure 19 gave such bad results. Firstly, it could be because of bad soldering. Metal strips have to be soldered to the bottom (positive) and the top (negative) of the solar cells in order for easy connections. Excessive heat due to the soldering iron being on the surface for too long could have damaged it.
The second option was that the cutting parameters were not right for this type of cell. The cells are cut using a laser cutting machine, and settings such as frequency, speed and power can all be modified. Cutting the cell too lightly could make it difficult to break it correctly, but cutting with too much power could damage the cell.
With these hypotheses in mind, new cells were cut with better parameters; these were also soldered more carefully. The new parameters were recommended by more experienced members in the PV lab. Also, a modified testing setup was used, as a lot of the potentiometer boxes were broken and thus weren't very reliable. Instead of these boxes, an electric load was used. Instead of the lamp, a 'Solar simulator' was used, which can emit light at very constant intensities up to $1000W/m^2$. Measurements of the newly cut cells, as seen in figure 21, did improve. However, they did still not provide desired results as the fill factor $FF = \frac{404m}{1.73*567m} = 41\%$ is not within the typical range. A possible explanation was that the solar cells might have heated up too much due to long exposure to high irradiance, which would have decreased their efficiency.

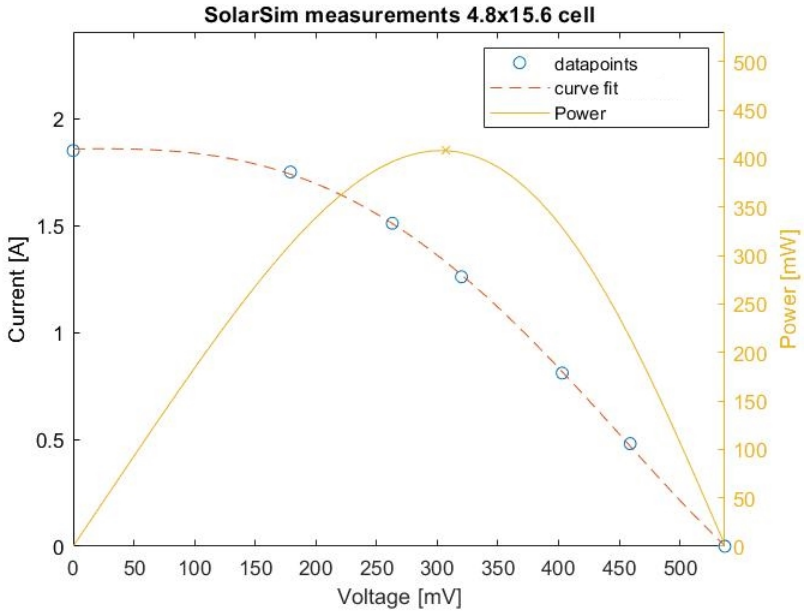

Figure 21: solar simulator measurement of a cell cut with new parameters

And so the testing setup was adapted once more. The solar simulator was replaced with the lamp again. The electronic load was replaced with a rheostat adjustable resistor. Finally, this resulted in good measurements on multiple cells, as can be seen in figure 22 and figure 23. Their fill factors are respectively $FF = \frac{302m}{0.91*550m} = 61\%$ and $FF = \frac{298m}{0.86*535m} = 64\%$.

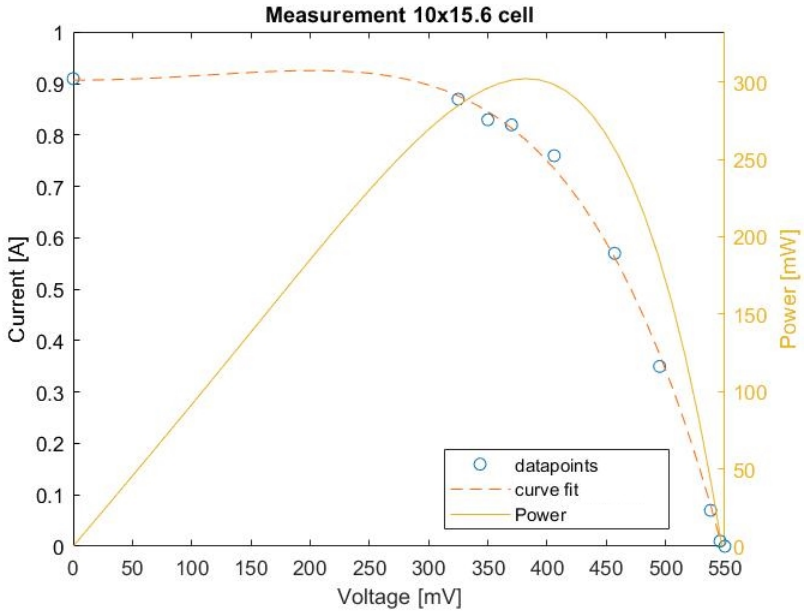

Figure 22: IV curve measurement of a 10x15.6 cm cell

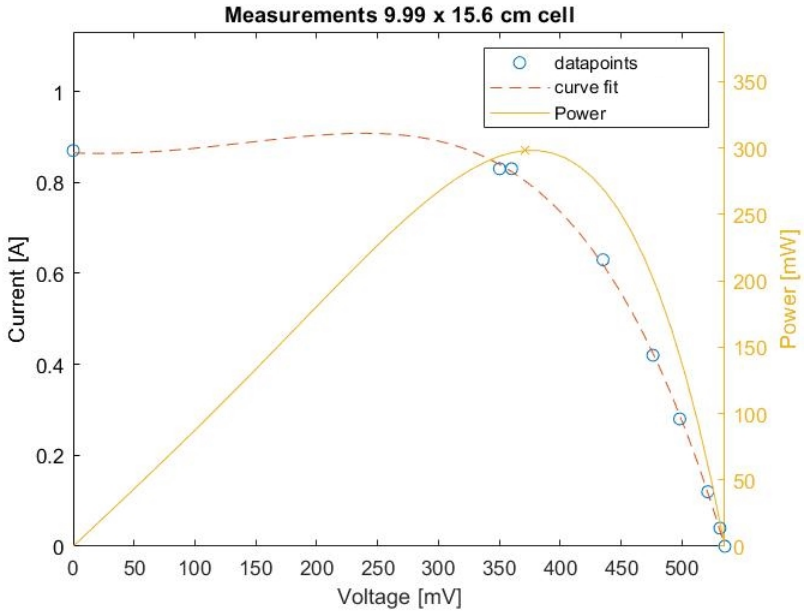

Figure 23: IV curve measurement of a 9.99x15.6 cm cell

5.2 Transparent foil measurements

In the final product, the plane's wings - and thus the solar cells - will be wrapped in transparent foil. This foil might reflect some of the incident light, which will then not reach the solar cells. Therefore it is important to know the transmissivity of the wrapping film, to be able to make a good estimate of the actual solar output power. For this test, firstly the IV curve of a 3x15.6 cm cell was measured. Then it was measured again, but now the cell was covered by the transparent foil. It is important to

note that the film was not wrapped tightly around the cell, and had some crinkles in it. This might have had influence on the transmissivity. The IV curves are plotted in figure 24.
The current was then multiplied by the voltage to create a figure showing the difference in power output (see figure 25). Important measurement values are shown in table 2. From this it can be concluded that in our measurements, the foil had a transmissivity of $\frac{331}{395} = 84\%$. This means that 16% of the incoming light is reflected.

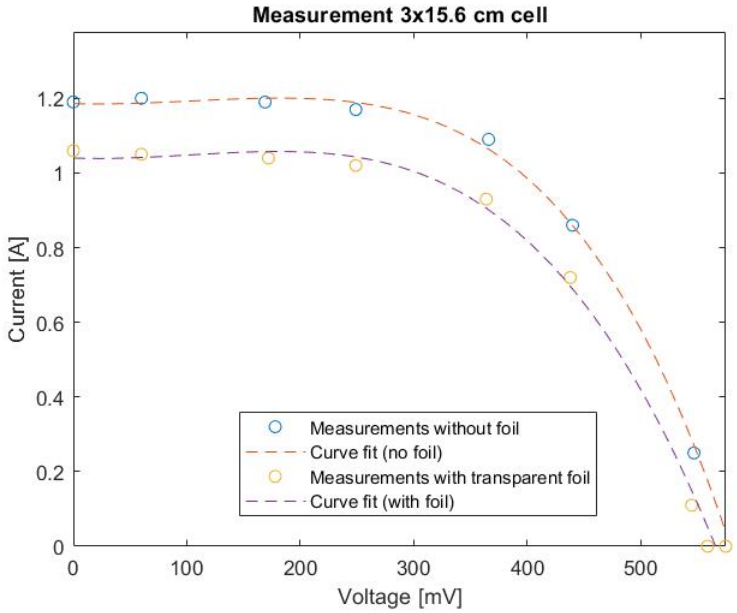

Figure 24: Measurements of a 3x15.6 cm cell, with and without covering film

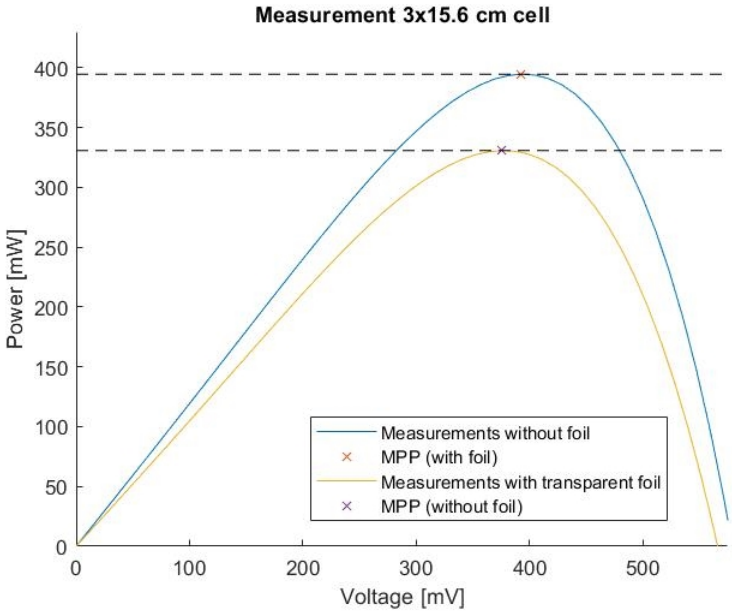

Figure 25: Power curve of a 3x15.6 cm cell, with and without covering film

Table 2: Foil measurement results

	MPP [mW]	I_{sc} **[A]**	V_{OC} **[mV]**
without foil	395	1.19	580
with foil	331	1.01	550

5.3 MPPT and charge controller

Following all the requirements set in section 4.1.3, the MPPT and charge controller code was written. This was programmed in Visual Studio Code using the mbed programming language.

5.3.1 Sensor tests

To fully test the MPPT and CC, it is important that the current and voltage sensors work correctly. That is why these need to be tested and calibrated. The voltage sensor provided by the PE team is shown in figure 26, where V_{in} will be connected to a voltage generator and V_{out} to an analog input port of the microcontroller. According to this schematic, V_{out} should be defined as

$$V_{out} = \frac{2.2}{2.2 + 5.6} V_{in} = 0.28205 V_{in} \tag{8}$$

And so, to retrieve the actual input voltage from the voltage read by the microcontroller, V_{out} must be divided by 0.28205. Furthermore, the microcontroller considers 3.3V as a digital 1. So when 3.3V is set over an analog input port, the software will read this as a one. Therefore the value read from the analog input port should also be multiplied by 3.3. Measurements were done by varying the input voltage from 0 to 10 volt in steps of 1V, the results can be seen in Appendix B. On average, the measurements did not deviate from the actual input voltage by more than 1%.

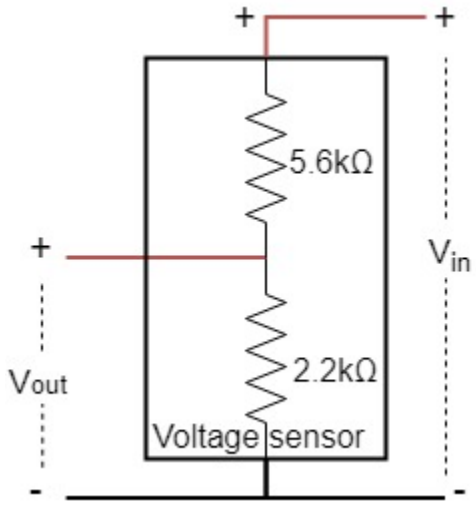

Figure 26: Test setup of the voltage sensor

The schematic of the current sensor, provided by the PE team, is shown in figure 27.

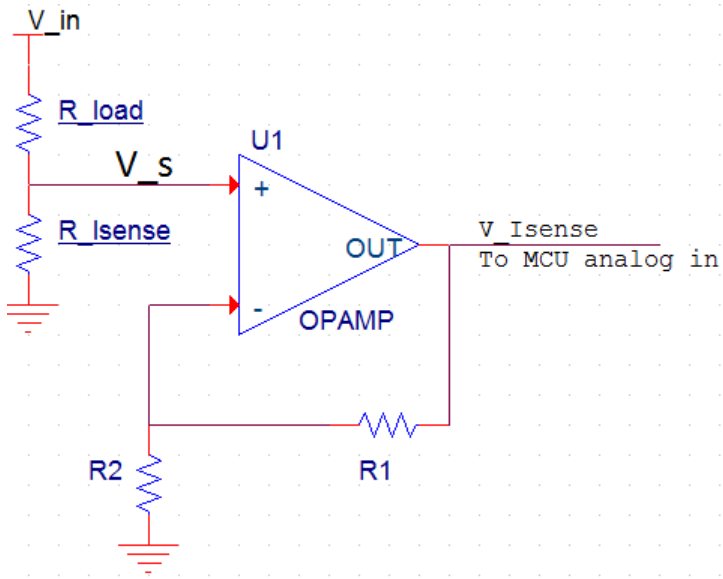

Figure 27: Test setup of the current sensor

When trying to test this sensor, the sensing resistance seemed to malfunction; the sensor did not show expected behavior. When removing the sensing resistance and simply putting a voltage over the opamp, the sensor did work as expected. For testing this part of the sensor, a voltage source was connected to the opamp and the ground shown in the figure. The output of the sensor was connected to an analog input port of the microcontroller. The opamp amplifies V_s by a factor of 5.57, and so the value read by the microcontroller was multiplied by 5.57 and 3.3 to retrieve V_s. A sensing resistance of 0.1Ω is assumed, so V_s is multiplied by 0.1 to get I_s, the current that would flow through this resistor. The results of this test are shown in Appendix B. The results show that the average read current has a bias of $\frac{0.02-0.02-0.02-0.01-0.02-0.02}{5} = -0.014$ as compared to I_s, the virtual input current . This bias can be mitigated in future measurements by subtracting it from the calculated current.

5.3.2 MPPT and charge controller testing

The code for the MPPT and CC was done relatively early in the project, but could not be physically tested as the sensors of the PE team were not yet fully working. This is why another testing method was designed, which involved simulating a solar cell on another microcontroller and using its analog output ports as stand-in voltage and current sensors.
The design of the PV simulator is as follows: It will receive the value of the MPPT duty cycle as input. Based on the duty cycle it will output the appropriate voltage and current values according to an approximation of a typical IV curve. The voltage is directly calculated based on the duty cycle. The output current is constant up to a certain threshold voltage, after which it drops linearly to zero at the open circuit voltage. The code can be seen in Appendix D. In case of the given code, the MPP is exactly at the transition point between constant current and linear decrease as shown in figure 28.

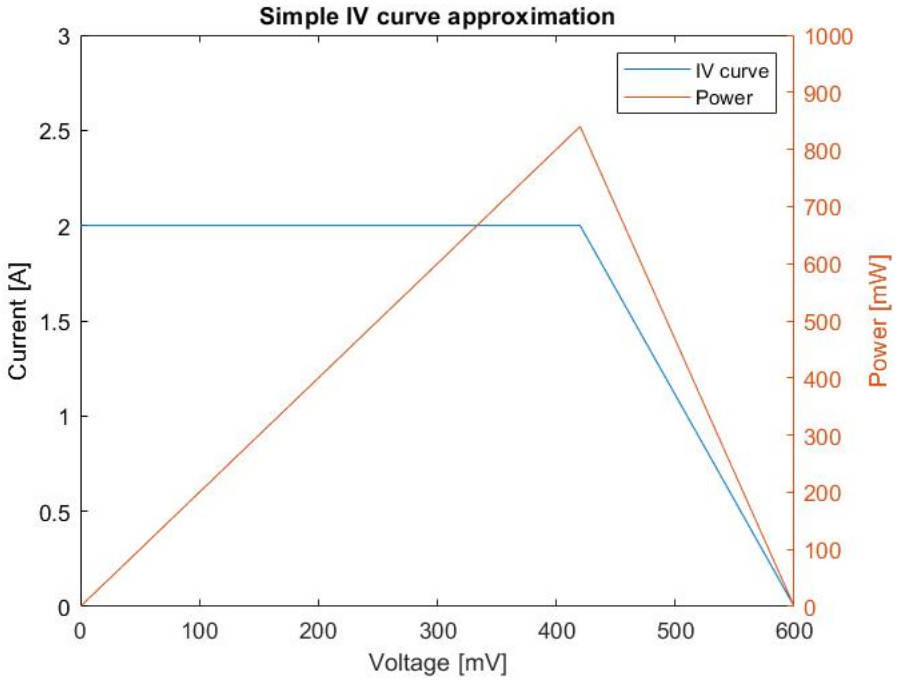

Figure 28: Simplified approximation of an IV curve

The MPP is in this case exactly at 70% of the V_{oc}. According to the boost converter formula $V_{in} = (1 - D) * V_{out}$ - where in this case $V_{out} = V_{oc}$ and V_{in} is the solar panel voltage - duty cycle D must be $D = 1 - \frac{V_{mpp}}{V_{oc}} = 1 - 0.7 = 0.3$. So, if the MPPT unit settles at a duty cycle of 0.3, the algorithm works.
The test was executed and the data is plotted in figure 29.

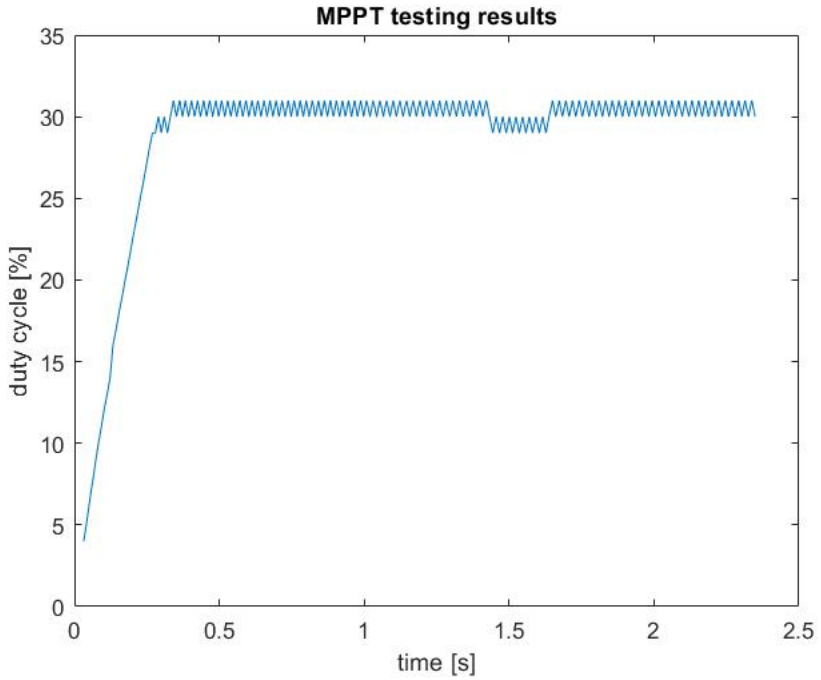

Figure 29: Test results of MPPT unit tracking MPP at 30% duty cycle

From this data it can be seen that the MPP at 30% duty cycle is reached in about 0.3 seconds. This would mean it would take one second to go from 0% duty cycle to 100% duty cycle, which in its turn would mean that the settling time is always lower than one second.
The oscillation magnitude around the MPP is equal to the perturbation size $\Delta D = 1\%$. These measurements exactly fulfill the demands set in the Programme of requirements. The efficiency of the MPPT must still be tested, but this is dependent on the IV curve of the final solar array that will be implemented. At the moment of writing this report, we have not been able to do this measurement yet.
The charge controller was also tested in combination with the solar cell simulator. Relevant information like the control signal value and the states of the switches were printed to the console, and the CC proved to work correctly in this simulation.

5.4 Integration with the rest of the system

The integration of the PV system has two main aspects: mounting the solar panels on the plane, and properly connecting the electrical system.

The used cells are flexible, but only to a certain extent. The cells need to follow the curvature of the wing. The biggest challenge during cell placement is to divide the pressure on the cells as evenly as possible, as to not break them. One of the problems is that the solar cells have connectors (metal strips) soldered onto them on the top and bottom. When placing them on the wings, the pressure would be concentrated at these points. This is illustrated in figure 30. To solve this, incisions were made into the wood of the wings. The connectors fit in to these grooves so the pressure is more evenly divided.

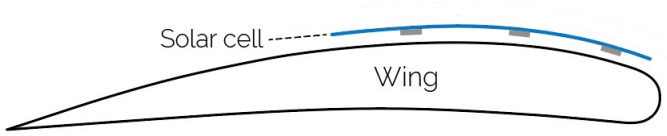

Figure 30: Visualisation of cell following wing curvature

The cells would then be glued to the wood. This method seemed to work when first trying it out. But as the glue was drying up the placed cells broke. This might also have been because pressure was still being applied to them by hand to keep them in place.
To spread this pressure more evenly over the cells while the glue is drying up, the next plan is to use a small bag of sand, as shown in figure 31.

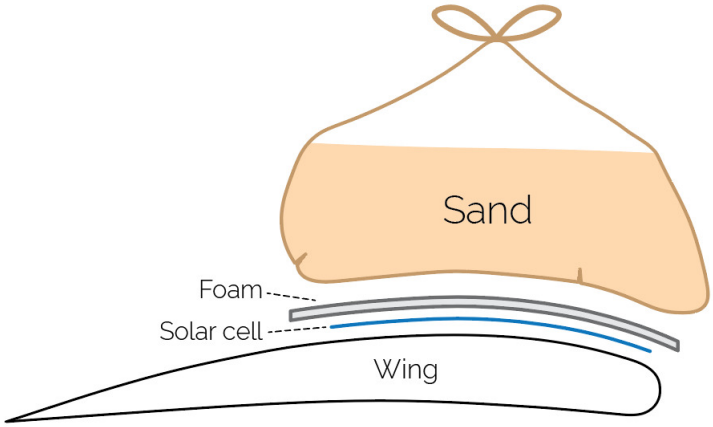

Figure 31: New placement method

This method has been tested without glue first, and the cells did not break. However no cells have actually been placed yet.

5.5 Validation of the entire system

As the system hasn't been fully integrated yet, it is hard to properly validate it. However, a test plan can still be made.

5.5.1 Test plan

Components and systems that still need to be fully tested and/or validated are:

- Complete current sensor
- MPPT and CC
- Solar cell placement
- Full system

Firstly it is important that the current sensor gets finished. Then it can be tested and calibrated in a similar way as described in section 5.3.1. When this works, the MPPT and CC can be fully tested. Testing the MPPT will be done using the circuit shown in figure 32. Using a constant light source, firstly the full IV curve of the cell can be measured. This will reveal the maximum power point of that

cell with that specific light intensity. Then the maximum power point tracking can be activated, and it should settle at the measured MPP. The power dissipated by the load will be the MPP times the efficiency of the converter.

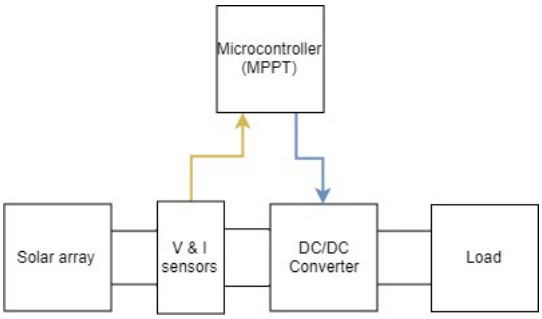

Figure 32: Test setup for MPPT

To test the CC properly, all the cases in figure 10 must be tested in mostly the same configuration as the full system. The full system circuit including sensors is shown in figure 33.

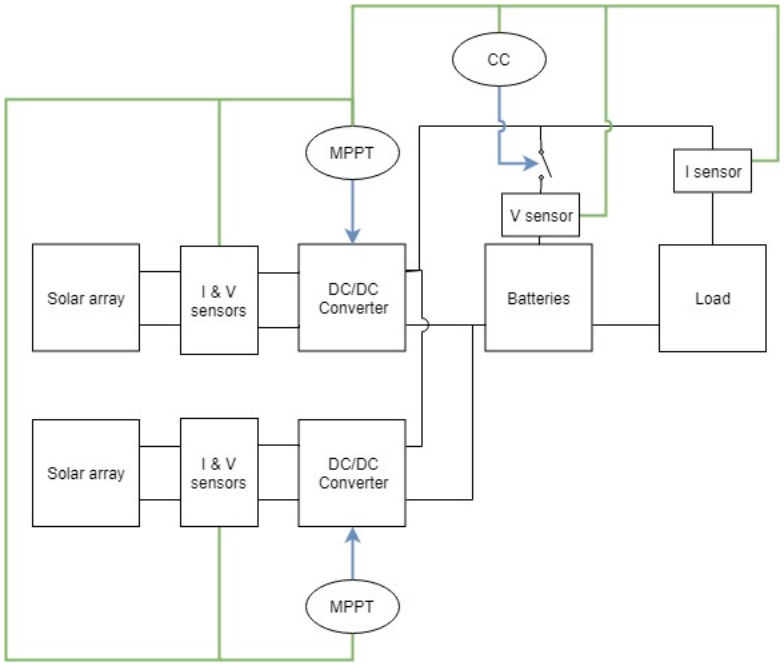

Figure 33: Electrical system layout including sensors

For testing the CC it is however not necessary to use two full solar arrays. For example only one cell or small array can be used in combination with one converter. Also the load can simply be a variable resistor.

Once all the solar cells are properly placed without breaking, the I-V curves of the wings can be measured for different irradiances. Afterwards we will be able to do a test outside to see the amount of power generated. Based on this we will be able to estimate the additional power delivered by the PV system and its effect on extending the flight time of the RC plane.

6 Conclusions and discussion of future work

In this thesis, a design was made from implementing solar panels on an RC plane, and code for MPPT and a charge controller were designed and written. Furthermore many solar cells were tested in different circumstances.
The cut solar cells have shown to work properly, but their placement on the wings is still a challenge. The MPPT is working according to the set requirements, however it still needs to be tested on an actual solar cell/array. The CC code acts as desired but has yet to be implemented in the full system.

Much was learned from working with these systems, but the results at the moment of writing this report are not very substantial. A lot of time was spent on cutting and testing solar cells, and trying to find explanations on their unexpected behaviour. Also, the dependency on the development of the current and voltage sensors stalled our progress concerning the testing of the MPPT and CC. In hindsight it would have been a good idea to order our own current and voltage sensors online.
In the two weeks left in the project, the test plan described in section 5.5.1 will be used as a guideline for our system tests. The sensors of the PE team are almost working properly and so the system tests can begin soon. With two weeks remaining, we are still confident we can create a working product.

7 Bibliography

References

[1] (2018) Levelised cost of electricity renewable energy technologies. [Online]. Available: https://www.ise.fraunhofer.de/content/dam/ise/en/documents/publications/studies/EN2018_Fraunhofer-ISE_LCOE_Renewable_Energy_Technologies.pdf

[2] (2014) Nasa armstrong fact sheet: Helios prototype. [Online]. Available: https://www.nasa.gov/centers/armstrong/news/FactSheets/FS-068-DFRC.html

[3] Solar impulse. [Online]. Available: https://www.solvay.com/en/innovation/encourage-science/solar-impulse

[4] C. Blais. Is it possible to make solar-powered airplanes? [Online]. Available: https://engineering.mit.edu/engage/ask-an-engineer/is-it-possible-to-make-solar-powered-airplanes/

[5] P. Chmielewski and W. Wróblewski. (2017) Selected issues of designing and testing of a hale-class unmanned aircraft.

[6] M. G. Villalva, J. R. Gazoli, and E. R. F. (2009) Modeling and circuit-based simulation of photovoltaic arrays. [Online]. Available: https://www.researchgate.net/profile/Marcelo_Villalva/publication/224087574_Modeling_and_circuit-based_simulation_of_photovoltaic_arrays/links/0deec53372df72e836000000/Modeling-and-circuit-based-simulation-of-photovoltaic-arrays.pdf

[7] D. Karabetsky. (2016) Solar rechargeable airplane: Power system optimization. [Online]. Available: https://ieeexplore.ieee.org/document/7783146

[8] A. Hart. Effects of clouds on a solar panel. [Online]. Available: https://www.streetdirectory.com/travel_guide/125965/computers/effects_of_clouds_on_a_solar_panel.html

[9] S. E. Babaa, M. Armstrong, and V. Pickert. (2014) Overview of maximum power point tracking control methods for pv systems. [Online]. Available: https://file.scirp.org/pdf/JPEE_2014082811233330.pdf

[10] Datasheet stm32f303. [Online]. Available: https://www.st.com/resource/en/datasheet/stm32f303k8.pdf

[11] Bypass diode for solar panel protection. [Online]. Available: http://www.alternative-energy-tutorials.com/energy-articles/bypass-diode.html

[12] D. Y. Dube and R. K. Munje. (2015) Modeling and control of unmanned aerial vehicle. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7503428

A IV curve fitting

A.1 IVcurves.m

```
close all;

function [] = plotIV (IV)
figure()
I = IV(:,2);
V = IV(:,1);

V = [-500;-100;V];
I = [2*max(I);1.01*max(I);I];
x = linspace(0,max(V),50);
n = 5;
p = polyfit(V,I,n);
f = polyval(p,x);
plot(V,I,'o'); hold on;
plot(x,f,'--');
xlabel('Voltage [mV]'); ylabel('Current [A]');

xlim([0 max(V)]);ylim([0 max(I)/2 * 1.3]);
legend('Current datapoints','Current curve fit');


FF = plotPV(x,f)
end


function [FF] = plotPV (V,I)
P = I.*V;
mpp = max(P)
Voc = V(end)
Isc = I(1)

hold on;
yyaxis right;
plot(V,P);
ylabel('Power [mW]');

ylim([0 1.3*mpp]);
VmppIndex = find(P == mpp, 1);
Vmpp = V(VmppIndex);
plot(Vmpp,mpp,'x');
legend('Current datapoints','Current curve fit','Power');


FF = mpp/(Voc*Isc); % return fill factor
end
```

B Sensor measurements

B.1 Voltage sensor measurements

Table 3: Voltage sensor readings

Input Voltage [V]	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Avg. read voltage [V]	0,01	0,99	2,01	3,01	4,02	4,99	6,06	7,03	8,03	9,03	10,03
Read voltage [mV]	0	965	2008	2982	4008	5011	6331	7017	8039	9025	10034
	0	997	1988	2985	4030	5008	6031	6971	8045	9017	10059
	5	1006	1993	2968	4053	5031	6014	7031	8019	9014	10140
	0	831	2014	3008	4006	5002	6037	7053	8042	9028	10053
	5	980	1996	2982	4008	4996	6002	7031	8034	9034	10065
	5	957	1977	3014	4020	4996	6011	7003	8028	9051	10014
	8	1017	2019	3040	3997	5025	6014	7036	8279	9037	10048
	5	1048	1999	3002	4003	5014	7042	7036	8028	9051	10042
	0	974	2051	3017	3997	5019	6002	7076	8005	9025	10014
	5	994	2005	3008	3951	5051	6014	7059	8016	9051	10059
	23	994	2002	2906	4006	5039	6020	6985	8031	9022	10034
	0	994	1991	3020	4017	4996	6025	7026	8011	9002	10026
	8	988	2188	3008	3991	5016	6350	7026	8005	9031	10014
	0	991	2002	3014	4011	4990	6020	7026	8068	9082	10003
	23	1003	2011	2991	3997	5025	6037	7026	8034	9037	10071
	0	991	1980	2968	4011	5014	6031	7042	7971	9040	10020
	23	983	2002	2999	4020	4996	5991	7053	7999	9060	10036
	20	974	2002	3014	4028	5019	6362	6997	8048	9037	10036
	54	1014	2002	2997	4030	5014	6025	7036	8042	9034	10003
	0	1017	2008	2997	4448	5014	5997	7008	8042	9011	10026
	11	1008	1996	3023	4006	4785	6043	7020	8034	9079	9962
	0	1014	1996	3025	3983	4980	5997	7003	8016	9034	10071
	0	968	2011	2991	4003	4985	6031	7020	8085	9025	10036
	0	980	2005	2997	4065	4996	6031	7017	7977	9031	10042
	40	983	1991	3066	4023	4990	6002	7031	8051	9034	10059
	99	968	1980	3008	4030	5031	6031	7053	8028	9051	10082
	0	937	1996	3054	4020	5019	6011	7020	8028	9045	10036
	2	1000	1922	3014	3997	4996	6054	7014	8051	9031	10020
	0	1006	2011	3014	4011	4137	6031	7048	8016	9017	10036
	0	965	2042	3008	4030	4756	6020	7051	8016	9011	10020
	0	1011	2008	3073	4008	5025	6014	7048	8060	9051	10065
	0	1045	2014	2888	4023	5019	6020	7014	8028	8979	10042
	31	1008	1991	2994	4020	5060	6034	7020	8028	9063	10036
	2	994	1966	3048	3988	5014	6014	7017	8034	9020	10020
	0	1011	1996	3031	4014	5025	5997	7053	7986	9037	10048
	0	1006	1996	3005	4000	5042	5991	7014	8031	9051	10036
	2	965	2008	3002	4017	5014	6037	7031	8002	8982	10026
	0	988	1951	3014	4017	4980	6105	7214	8062	8991	9997
	0	1000	2019	2997	4028	5223	6011	7036	8005	9008	9814
	20	991	1980	3008	4011	4948	6020	7026	8051	9014	10076
	0	985	2008	3017	4028	5060	6057	7045	8051	9074	9968
	0	1008	2042	3017	3997	5039	5979	7020	8028	9051	10094
	151	983	1993	3020	3994	5019	5991	7020	8019	9014	10036
	0	965	1999	3011	4008	5025	6011	7014	8028	9040	10111
	0	1023	2231	3023	4074	4940	6020	7042	8019	9034	9968
	40	1008	2005	2979	3983	5031	6031	7042	8019	9069	9997
	2	983	1980	3008	4006	4990	6025	7008	8019	9045	10068

B.2 Current sensor measurements

Table 4: Current sensor readings, virtual resistance of 0.1Ω

Input voltage [V]	**0**	**0,1**	**0,2**	**0,3**	**0,4**	**0,5**
Virtual input current [A]	0	1	2	3	4	5
Avg. read current [A]	0,02	0,98	1,98	2,99	3,98	4,98
Read current [mA]	53	814	2000	2987	3983	4978
	26	986	1799	2990	3988	4989
	18	976	1987	2971	3985	4978
	27	988	1998	2994	3974	4972
	14	992	1987	2994	3985	4988
	15	988	1974	2977	3997	4984
	11	989	1982	3000	3954	4965
	27	985	1977	2973	3968	4984
	15	967	1993	2984	3985	5013
	30	986	1973	2994	4000	5004
	20	995	1964	3003	3977	4984
	17	960	1986	3012	3983	4976
	33	986	1989	2990	3962	4975
	21	988	1990	2984	3983	4978
	17	959	1995	2970	3983	4995
	15	982	1990	2955	4010	4995
	26	989	1974	2976	3993	4966
	10	982	1983	2990	3971	4989
	20	970	1983	2993	3969	4998
	26	988	1973	2981	3971	4981
	28	999	1970	2993	3991	4966
	28	988	1993	2987	3980	4968
	21	991	1987	2984	3988	4963
	21	979	1958	2983	3983	5037
	17	988	1987	2993	3994	5037
	21	988	1993	2987	3987	4777
	21	965	1957	2987	3988	4982
	21	960	1984	2981	3980	4979
	23	989	1966	3004	3988	4956
	20	988	1990	2987	3993	4894
	24	988	1983	2987	3980	4976
	21	978	1990	2828	3985	4985
	89	979	1982	2993	3980	4984
	20	985	1971	2986	3965	4979
	20	985	1995	2987	3985	4988
	18	982	1967	2973	3985	4953
	0	989	1992	2970	3969	4985
	21	972	1982	2967	3980	4987

C MPPT/CC code

C.1 main.cpp

```
#include "mbed.h"
#include "MPPT.h"
#include "ChargeController.h"

/*
Tracks the maximum power point for multiple solar arrays (solar cell groups),
And controls the power flow in the system, to maximize solar power usage, without overch
*/

/* Control signal to be read from ChargeController.
0 = track MPPT
1 = pause MPPT
2 = track Ptarget
3 = reset MPPT
*/
int Control;

float Target; // Target variable to be sent to MPPTs in case they need to track a lower

// Pins for tracking group 1
#define I1 PA_0 // Current sensor
#define V1 PA_1 // Voltage sensor
#define PWM1 PA_3 // Pulse width modulation output

/* Pins for tracking group 2 */

//Pins for charge Controller
#define Vbat PA_5 // Battery voltage sensor
#define Iload PA_6 // Load current sensor
#define BatSw PA_8 // Switch which can enable battery charging


MPPT MPPT1(I1,V1,PWM1); // Create Maximum Power Point Tracker 1
/* Create Maximum Power Point Tracker 2 */

ChargeController CC(Vbat,Iload,BatSw); // Create Charge Controller object


// define the Serial object, for debugging
Serial pc(USBTX, USBRX);

int main() {
    while(1) {
        CC.Ppv = MPPT1.readP(); // Send measured solar power to the charge controller
        CC.run(); // execute charge controller algorithm
        Control = CC.readControl(); // read control variable
        if (Control == 0) { // Track MPPT
            MPPT1.PerturbObserve(); // Execute the P&O algorithm for tracking group 1
            /* Execute the P&O algorithm for tracking group 2 */
```

```
        }
        else if (Control == 1) { // Pause the MPPT algorithm
            MPPT1.pause();
            /* pause MPPT2 */
        }
        else if (Control == 2) { // Track Target power
            Target = CC.readPtarget(); // read Pload as target power.
            MPPT1.PerturbObserve(Target);
            /* Track target MPPT2 */
        }
        else if (Control == 3){ // Prevent overcurrent in battery
            MPPT1.reset();
        }

        // MPPT doesn't need to happen very fast
        wait_ms(1); // Track the power point every 10 ms
    }
}
```

C.2 MPPT.h

```
#include "mbed.h"

const float PERTURB_CONST = 0.01;
const float PERIOD_US = 100; // [us] PWM period
const float CURRENT_SENSOR_MAX = 11; // [A] maximum current the sensor is able to sense
const float VOLTAGE_SENSOR_MAX = 9; // [V] maximum voltage the sensor is able to sense p

float I, V, P; // Variables for Current, Voltage and Power


/* Maximum Power Point Tracker Class
Control the duty cycle of a DC-DC converter switch using PWM,
to optimize the power flow.
Power is calculated using current and voltage sensors.
*/
class MPPT{
  private:
    AnalogIn *CurrentSensor;
    AnalogIn *VoltageSensor;
    PwmOut *PwmOutput;
    float PreviousPower; // Power measured previous iteration
    float DutyCycle; // Duty cycle of PWM output signal
    float Perturbation; // Change in the duty cycle every iteration

    float readI(); // read from current sensor
    float readV(); // read from voltage sensor

  public:
    MPPT(PinName I_pin,PinName V_pin,PinName PWM_pin);

    float readP(); // return last calculated power

    /* Perturb and Observe MPP Tracking algorithm.
```

```
    Optional target power argument in case a lower power point needs to be tracked.*/
    void PerturbObserve(float Target);

    void pause(); // stop tracking power point, open PWM switch
    void reset(); // stop tracking power point, reset duty cycle to zero
};

MPPT::MPPT(PinName I_pin,PinName V_pin,PinName PWM_pin){
  CurrentSensor = new AnalogIn(I_pin);
  VoltageSensor = new AnalogIn(V_pin);
  PwmOutput = new PwmOut(PWM_pin);

  Perturbation = PERTURB_CONST;// the change in the duty cycle
  PwmOutput->period_us(PERIOD_US); // set PWM frequency to 10kHz (1/100us = 10kHz)
};

float MPPT::readI(){
  return CurrentSensor->read() * CURRENT_SENSOR_MAX; // return read current
}

float MPPT::readV(){
  return VoltageSensor->read() * VOLTAGE_SENSOR_MAX; // return read voltage
}

float MPPT::readP(){
  return PreviousPower; // return last calculated power
}

void MPPT::PerturbObserve(float Target = 0){
  //observe
  I = readI(); // read current sensor
  V = readV(); // read voltage sensor
  P = I*V; // calculate power

  /*perturb*/
  if (Target == 0) { // No target given -> Track MPPT
    if (P < PreviousPower) { // if previous perturbation resulted in loss of power
      Perturbation = -Perturbation; // reverse perturbation direction
    }

  }
  else { // Track target
    if (P>Target){
      Perturbation = -abs(Perturbation); // set perturbation to negative
    }
    else {
      Perturbation = abs(Perturbation); // set perturbation to positive
    }
  }
  DutyCycle = DutyCycle+Perturbation; // Apply Perturbation
  /*       */

  // Constrain duty cycle to be in between 0 and 1
```

```
  if (DutyCycle < 0) {DutyCycle = 0;}
  else if (DutyCycle > 1){DutyCycle = 1;}

  PwmOutput->write(DutyCycle); // write PWM

  PreviousPower = P; // store calculated power
}

void MPPT::pause(){
  /* Don't set DutyCycle variable to zero,
  because we want to save it for when MPPT is resumed.
  Instead just write a zero to PwmOutput*/
  PwmOutput->write(0); // Open MOSFET
  I = readI(); // read current sensor
  V = readV(); // read voltage sensor
  P = I*V; // calculate power (should be zero)
  PreviousPower = P;
}

void MPPT::reset(){
  /* reset DutyCycle to zero */
  DutyCycle = 0;
  PwmOutput->write(DutyCycle);
  I = readI(); // read current sensor
  V = readV(); // read voltage sensor
  P = I*V; // calculate power (should be zero)
  PreviousPower = P;
}
```

C.3 ChargeController.h

```
#include "mbed.h"

#define open 0
#define close 1

float Ibat, Vbat, Iload, Vload, Pbat; // Variables for current, voltage for battery and
float SoC; // Variable for State of Charge of the battery pack
const float VbMax = 8; //[v] Maximum Battery voltage (SoC = 100)   //// Need to verify v

const int SoCMax = 95; // Maximum allowed battery State of Charge //// Need to verify va
const int SoCDanger = 97; // Dangerously high SoC //// Need to verify value
const int SoCMin = 20; // Minimum allowed battery SoC //// Need to verify value

const float BatCap = 3800; // [mAh] //// Need to verify value
const float IbatDanger = 2*BatCap/1000; // [A] maximum safe charging current
const float IbatMax = IbatDanger*0.8;

const float LOADCURRENT_SENSOR_MAX = 11; // [A] maximum current the sensor is able to se
const float BATVOLTAGE_SENSOR_MAX = 9; // [V] maximum voltage the sensor is able to sens

/*
Control the power flow in the system using switches
*/
```

```
class ChargeController{
    private:
        /* Control signal to be read by the main code. Will tell it how to control the M
        0 = track MPPT
        1 = pause MPPT
        2 = track Ptarget
        3 = reset MPPT
        */
        int Control;

        float Pload; // Power used by the load
        float PbatMax; // Maximum allowed battery charging power
        float Ptarget; // Target tracking power, to be read by main loop

        AnalogIn *BatVoltageSensor; // Pointer to the Battery Voltage Sensor object
        AnalogIn *LoadCurrentSensor; // Pointer to the Load Current Sensor object


        DigitalOut *BatSwitch; // This switch can disconnect the Batteries from the PV s

        float readVbat(){
            return BatVoltageSensor->read()*BATVOLTAGE_SENSOR_MAX;
        };
        float readIload(){
            return LoadCurrentSensor->read()*LOADCURRENT_SENSOR_MAX;
        };

    public:
        float Ppv; // Incoming power from solar panels
        ChargeController(PinName BatVPin, PinName LoadIPin, PinName BatSwPin);
        void run(); // Execute charge controller algorithm
        int readControl(); // Allows the control signal to be read
        float readPload(); // Allows the load power to be read
        float readPbatMax(); // Allows PbatMax to be read
        float readPtarget(); // Allows Ptarget to be read
};

ChargeController::ChargeController(PinName BatVPin, PinName LoadIPin, PinName BatSwPin){
    BatVoltageSensor = new AnalogIn(BatVPin);
    LoadCurrentSensor = new AnalogIn(LoadIPin);
    BatSwitch = new DigitalOut(BatSwPin);
};

int ChargeController::readControl(){
    return Control; // Return control signal
}

float ChargeController::readPload(){
    return Pload; // Return Pload
}

float ChargeController::readPbatMax(){
    return PbatMax;
```

```
}

float ChargeController::readPtarget(){
    return Ptarget;
}

void ChargeController::run(){
    Vbat = readVbat(); // read battery voltage
    SoC = Vbat/VbMax*100; // Determine state of charge [%] //// This formula is incorrec
    Iload = readIload(); // read load current
    Vload = Vbat; // Load is in parallel with the batteries
    Pload = Iload*Vload; // Calculate power used by the load
    Ibat = (Ppv - Pload)/Vbat; // Calculate current flowing into battery
    PbatMax = IbatMax*Vbat; // Calculate maximum allowed battery charging power

    if(Ibat >= IbatDanger){ // Battery charging current is dangerously high
        Control = 3; // Reset MPPT (open MOSFET and reset duty cycle to zero)
        BatSwitch->write(open); // Disconnect battery

    }
    if (SoC >= SoCDanger) { // Battery charge is dangerously high
        Control = 1; // Pause MPPT (open MOSFET and pause algorithm)
        BatSwitch->write(open); // Disconnect battery
    }

    else if (SoC >= SoCMax && Ppv >= Pload) { // Both SoC and Ppv are too high
        Control = 2; // Tell main loop to read Ptarget. MPPTs must try to set Ppv equal
        Ptarget = Pload;
        BatSwitch->write(close); // Keep Battery connected to absorb excess current
    }

    else if (Ibat >= IbatMax){ // Battery charging current is very high, but within limi
        Control = 2; // Track lower power point
        Ptarget = Pload + PbatMax;
        BatSwitch->write(close); // charge battery

    }

    /* Might need to implement some control algorithm in case of low voltage */
    else if (SoC <= SoCMin && Ppv < Pload){ // Battery charge is too low
        Control = 0;
        BatSwitch->write(open); // prevent further discharging
    }

     else
    {
        Control = 0;
        BatSwitch->write(close);
    }
}
```

D PV simulator code

D.1 main.cpp

```
#include <mbed.h>

const float Vbat = 11;
const float Imax = 1;

DigitalOut myled(LED1);

AnalogIn D(PA_3); // Duty Cycle
AnalogOut V(PA_5); // Voltage
AnalogOut I(PA_4); // Current

int main() {
  float Vpv;
  float Ipv;
  float Iir=0.6;

  while(1) {
    myled = 0;

    Vpv = (1-D)*Vbat;

    // calculate Current
    if (Vpv <= 15*Iir) {
      Ipv = Iir; // PV cell acts as current source until a certain point
    }
    else {
      Ipv = (Iir/(Vbat-15*Iir))*(Vbat - Vpv); // after that (in this simulation), I decr
    }

    V=Vpv/Vbat; // write normalized Voltage
    I=Ipv/Imax; // write normalized Current

  }
}
```