# Justifications for design choices

Any figures that do not have sources have been created by the authors.

To do now:

Implement CV

SOC

MPPT, for testing the different configurations.

Schematic:

Connections directly between SMPS and Arduino shield have been omitted (internal connections), this is a simplified schematic.

Need to make a state diagram

Need to make a functional diagram

**Characterising components:**

When designing a system it is necessary to know the behaviour and limitations of its constituent components. There are three main components that make up the energy subsystem: the battery cells, the PV panels and the SMPS.

**Battery cells:**

WHAT IS THE BATTERY CELL NUMBER (ID) AND SUCH?

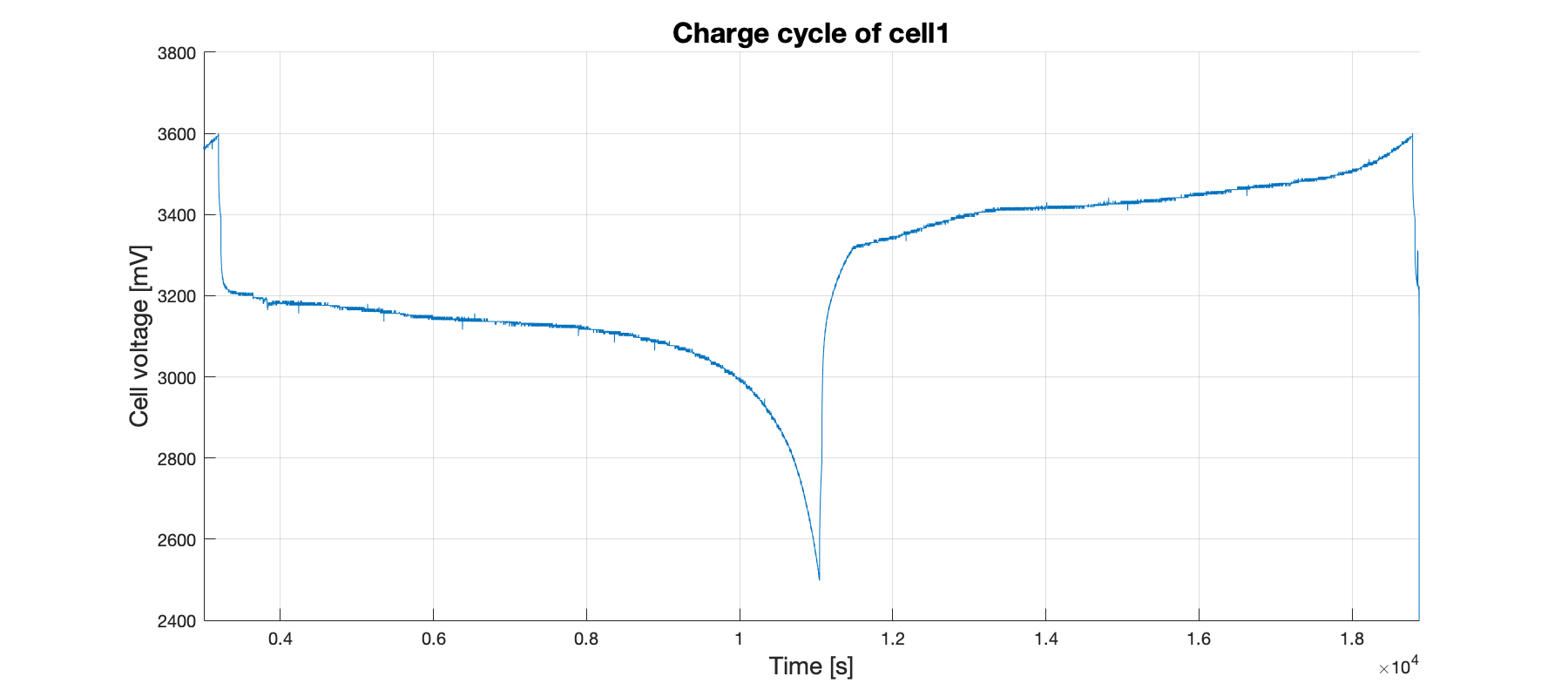
To determine the behaviour of the battery cells they were all tracked through a full charge cycle using the provided “Battery\_Charge\_Cycle\_Logged\_V1.1.ino” code. Every cell behaved similarly in terms of the cell voltage compared to time. The cell voltage of cell1 over a full charge cycle is shown below:

Figure 1: One full charge cycle of battery cell1. Note that the time axis starts at 3000 seconds.

Note the following important points on the graph. At 3190 s the cell is done charging and enters an idle state for 30 s after which it starts discharging. At 11000 s the cell is done discharging and enters an idle state for 30 s after which it starts charging. Finally, at 18800 s the cell is once again fully charged and the charge cycle is completed. The specific behaviour in each region of the graph will be discussed in later sections.

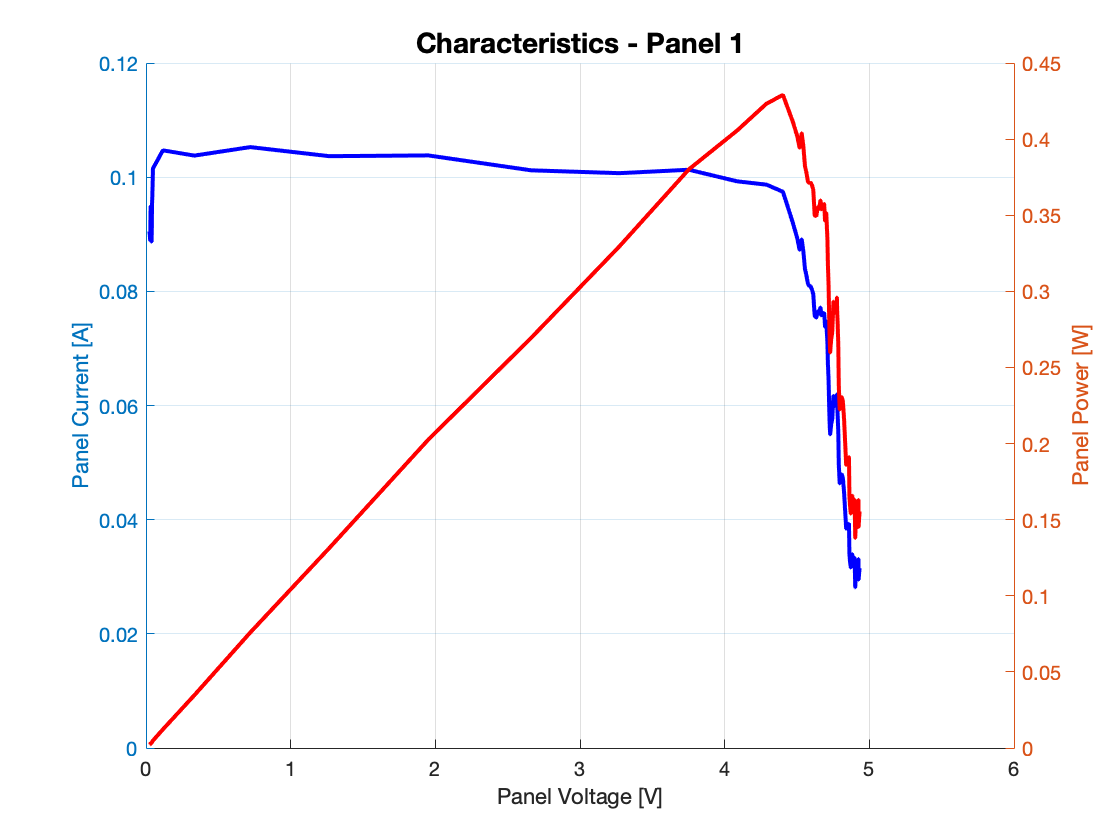
The provided charging algorithm also logs the current into the cell. By integrating said current for a full charge or discharge section it is then possible to determine the cell capacity in terms of mAh. The results of this analysis is presented in the table below:

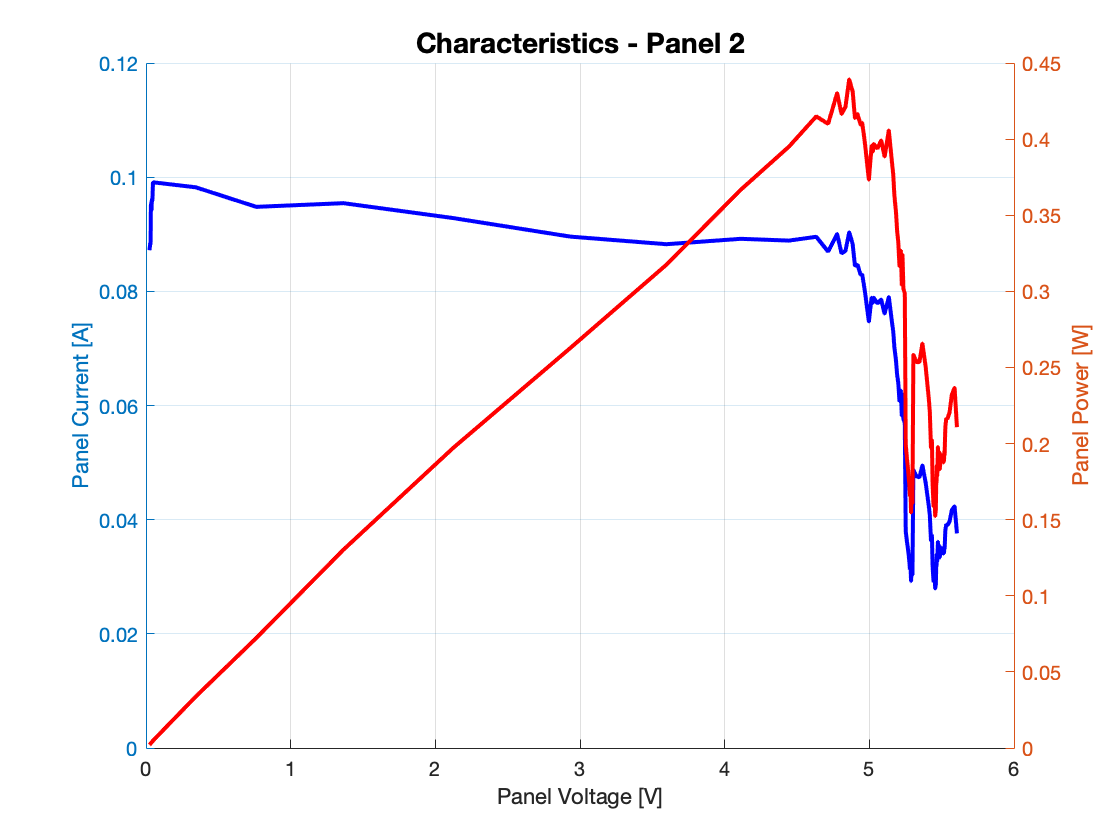
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cell number | 1 | 2 | 3 | 4 | 5 |
| Capacity (mAh) | 542.7 | 526.1 | 519.5 | 530.1 | 543.7 |

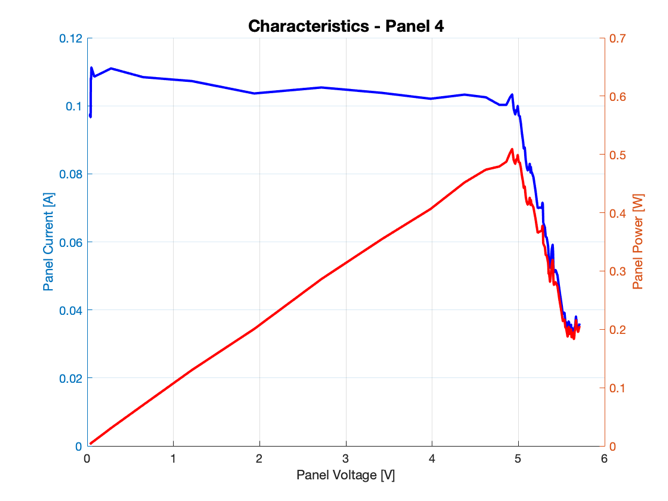
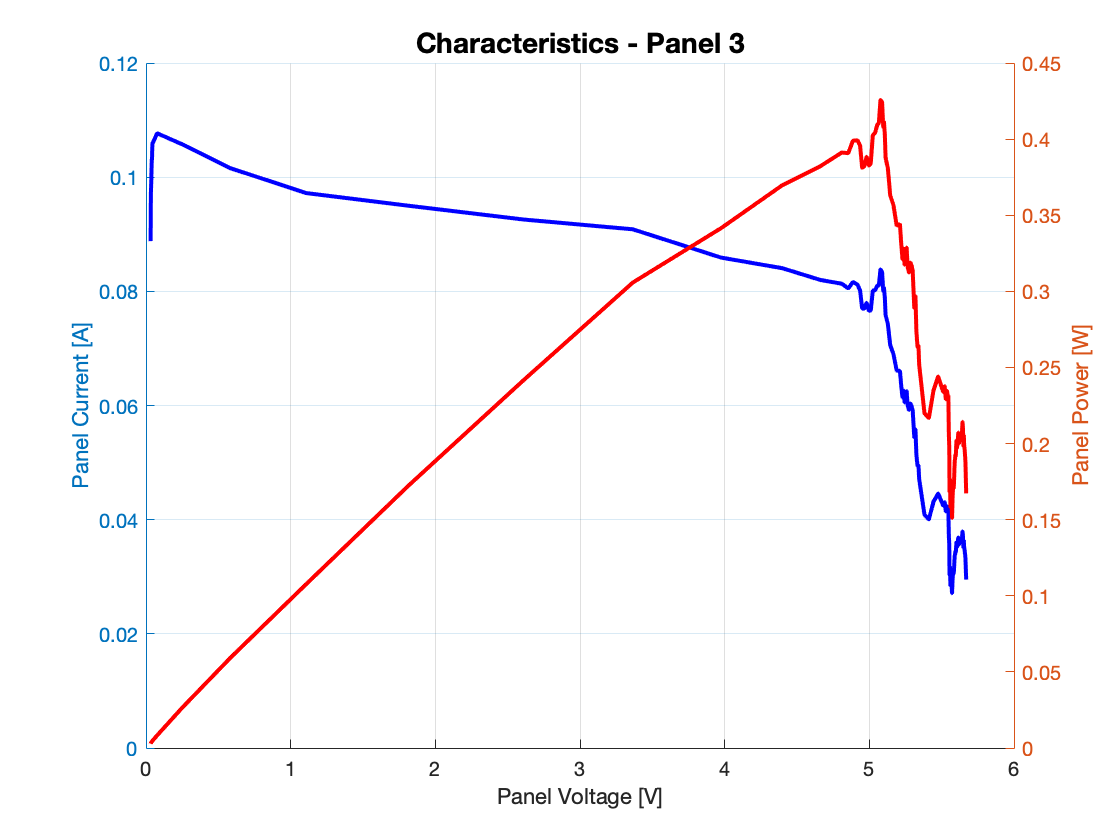
What is important to notice is that the cells all have different capacities. Also behavour at nearly charged an nearly fully discharged. Behaviour in rest part.

**PV panels**

The provided PV panels are each rated for a maximum power of 1.15 W at a voltage of 5.0 V and current 230 mA. Away from the maximum power point the performance of the panels can be determined from their I-V curves. To find the I-V curves each panel was connected to the SMPS operating in non-synchronous boost, they were then lit by the lamp and the duty cycle of the SMPS was varied while measurements of panel current and voltage were taken. After processing the resulting data it was plotted in figure ?:

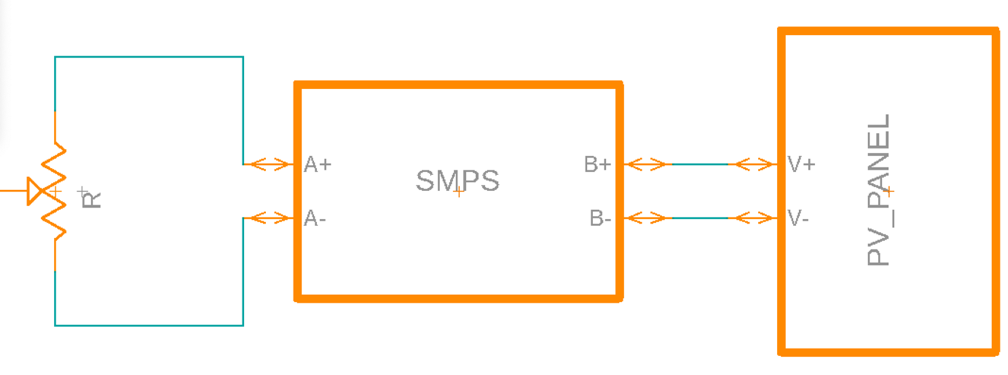






Though the data is noisy, it is clear that all panels exhibit the standard I-V characteristics of a PV cell. That is, they behave as non-ideal current sources with a nearly constant current at low voltages and a rapid current reduction at high voltages (2). Moreover, we see that the provided lamp activates the panels poorly as the peak power for each of the panels is only ~0.5 W.

To determine their performance away from the maximum power point we can find the I-V curves of the panels. To do t



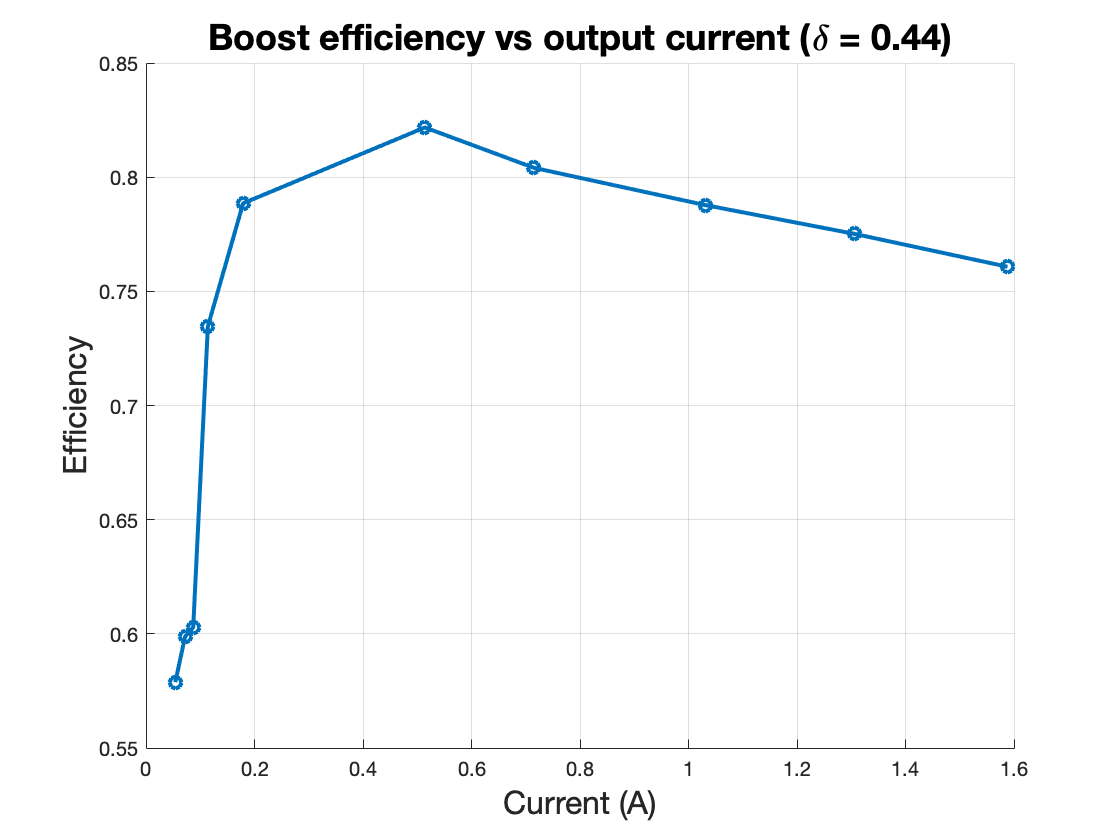
…, they are rated for … at … What is rated power etc?

The PV panels are characterised by their I-V curves. To determine the I-V curves each panel was connected to the SMPS in the manner shown below. To get consistency between panels, each panel was activated using the provided lamp and not direct sunlight.

The SMPS was used in a boost configuration such that the voltage and current of the panels could be measured directly. Using the “PV\_characterisation.ino” code (1), the input current was swept and the corresponding input voltage logged. The resistance on the output was changed at set currents such as to not exceed the maximum output voltage of the SMPS. The produced I-V curves and power output graphs are shown below:

Though the data is a noisy, it is clear that all panels exhibit the standard I-V characteristics of a PV cell. That is, they behave as non-ideal current sources with a nearly constant current at low voltages and a rapid current reduction at high voltages (2).

**SMPS**



The provided SMPS is rated for 10 W throughput with a maximum boost output voltage of 35 V and maximum output current of 10 A [lab specification]. All these ratings are far higher than needed and neither is expected to impose limitations on the design of the energy module.

The many characteristics of the SMPS have been thoroughly examined in 2nd year labs. However, for the energy submodule the most important characteristics will be the SMPS efficiency during non-synchronous boost operation. A graph of efficiency versus output current is shown in figure ?.

The characteristics vary with mode of operation, input power, output voltage and many other factors. However, as will be discussed in later sections, for the energy submodule the SMPS will be operated in non-synchronous boost mode.

and are not expected to com Neither of these ratings are expected to be

The SMPS has been thoroughly characterised in 2nd year labs.

Its characteristics vary with mode of operation, input power, output voltage and many other factors. As will be discussed in later sections, for the energy submodule the SMPS will be operated in non-synchronous boost mode. The most important part of the SMPS performance for the energy submodule is its efficiency. attribute *efficiency* is

only non-synchronous boost mode will be most important.

SMPS will be operating in non-synchronous boost mode. For this project the

The SMPS will be controlled by a closed loop controller and as such

buck mode with an output voltage in the range 2.4 - 3.6 V.

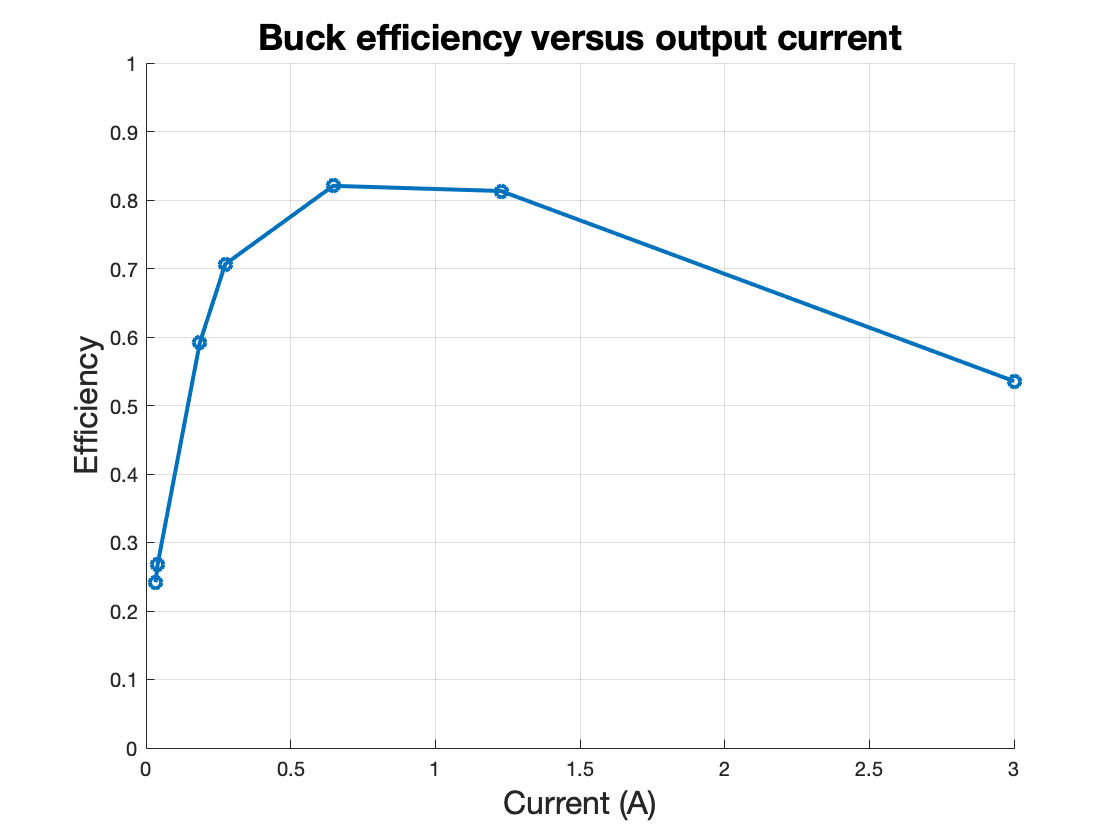


Figure 2 details the efficiency of the SMPS in closed loop, synchronous, buck mode, 5 V input with a target output voltage of ~2.7 V (3). The efficiency of the buck SMPS varies greatly with the output current, with very low efficiency at low output currents.

The SMPS has a power rating of 10 W and maximum input/output voltage of 20 V. When run of the USB power supply the input current is limited to 2.5 A. (4). However, when power is being provided directly at the SMPS ports the current limit is far higher at in our out, and as such is unlikely to impact the operation of the circuit.

Figure 2: Buck SMPS efficiency at V\_out = 2.685 V

**Configuration of PV panels:**

To facilitate the highest power output all the four provided PV panels will be used. There are four different ways in which the four panels can feasibly be connected. The different arrangements are shown in figure ?.

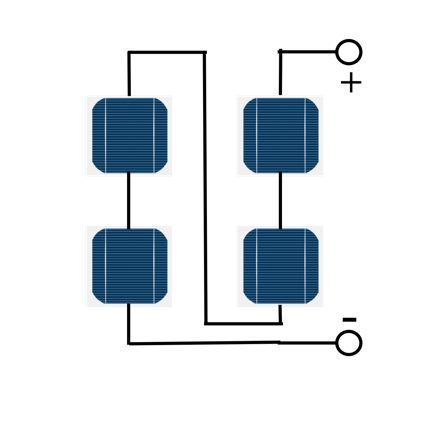
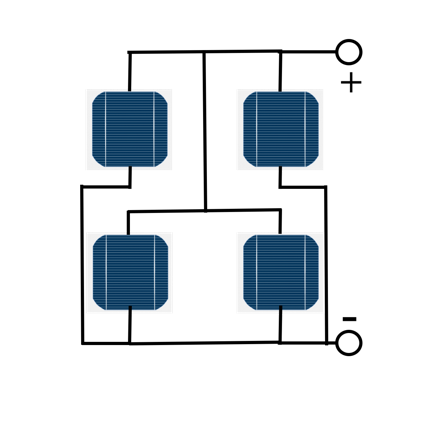
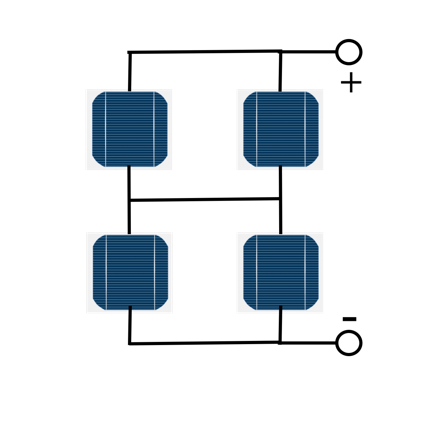
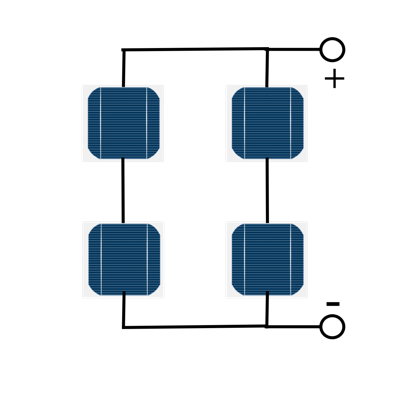


Figure 3: From left to right: Series (S), Series-Parallel (SP), Total-Cross-Tied (TCT), Parallel (P)

However, considering the array voltages it is clear that the only viable arrangement is the purely parallel connection. Consider first a pure series connection. In section 2.5.1 each PV panel was found to have a max voltage of around 5.5 V. The series connection will therefore have a total voltage of about 20+ V. The nominal voltage of the series battery pack is 4\*3.2 = 12.8 V. As the array voltage is higher than the battery voltage, the SMPS must be used in the buck configuration. However, the maximum buck input voltage is only 7 V [pmos datasheet] and therefore a series connected PV array cannot be used. Similarly for the Series-Parallel and Total-Cross-Tied arrangements the maximum array voltage will be about 11 V. However, as will be discussed in section 2.5.6 the battery pack voltage will swing between 10 V and 14.4 V in a charge cycle. The problem is that the 11 V array voltage is lower than the highest battery voltage, but higher than the lowest battery pack voltage. For the array configuration to charge the battery pack it would then be necessary to have a power converter which can both step up and down voltage. Depending on how the SMPS is configured it can either function in buck or boost mode, both not both at the same time. Thus it is not possible to use either the Series-Parallel or Total-Cross-Tied configuration. This leaves a purely parallel battery pack as the only viable option, which is why it has been chosen.

Though there are some differences in the power

CAN THE SMPS TAKE OVER 20V IF THE POWER IS FLOATING?

For series connected PV panels the PMOS will turn on automatically and not work. Look on datasheet for PMOS

The PV panels will provide the power used to charge the battery. The PV panels will perform their job well if they:

1. Provide a high average power output. This will allow us to charge batteries faster.
2. Provide a stable power supply. As will be discussed in later sections, for most of a charging cycle the battery will be fed a constant current. This will need close to constant power.
3. Can interface appropriately with other circuit components.

As we want the PV array to have a high average output all four PV panels will be used. Using all four PV panels there are four different ways that the panels could feasibly be connected.

Of the four proposed configurations we can readily reject a pure series connection. Though the PV panels are rated at 5 V, the measured I-V curves reveal that even under illumination purely from the provided lamp, the output voltage can go significantly above 5V. In that case the total voltage of the series connection would be 20+ V, which is higher than what the SMPS power converter is rated for. A pure series connection is therefore not a suitable configuration of the PV panels. (Can actually handle 100 V if connected the right way?)

This leaves three configurations, all of which produce voltages and currents that the SMPS can handle (integrate well with the SMPS). Assuming identical illumination and identical panels, all of the configurations produce the same amount of power. However, most likely each panel will not experience the exact same illumination. Especially on Mars where the deposition of dust over times leads to partial shading of each panel (6). In partial shading conditions a TCT configuration consistently outperforms the SP configuration in terms of power output, and is therefore preferred over and SP configuration (7). (Also makes power output more stable)

Research on the partial shading of solar

TCT has a problem with voltage, after leaving headroom on the power output for inefficiencies in the SMPS; the solar panels are only able to charge 4 cells at standard charging current. 4 cells in series give voltages in the range [10, 14.4] V, might not work well with voltages of solar panel.

Non-uniform shading, lamp does not activate array well. 8 days of consecutive rain in London, could not test outside.

Mismatch losses in Series are far higher:

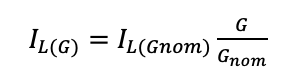
<https://ieeexplore.ieee.org/abstract/document/6104128>

How will series battery pack impact what voltage we need?

However, not only are all panels slightly different as can be seen on the I-V curves. Nor are they likely to experience the same illumination.

Moreover, for the TCT configuration no current would flow through the central wire and the array would therefore be identical to the SP configuration.

PV cells should not be connected directly in series. In a series connection partial shading can have a huge impact. From lectures we have:



Meaning that if a single PV cell has an incoming irradiance of half the others then the current will be halved and the power of the array will drop to almost half of peak power. On Mars partial shading is expected to occur as dust covers parts of the solar array.

This leaves two configuration total-cross tied connection (TCT) and full parallel. Both configurations are shown below:

Et bilde som inneholder tekst, klokke

Automatisk generert beskrivelse

TCT has the advantage over series connection that the drop off in the current of one cell has a far smaller impact on total output power.

Could also have done two parallel connections of two PV panels in series, but this has the disadvantages of series connection without the advantages of TCT. See linked article to see comparison.

Disadvantage of pure parallel connection is that voltage is lower. To charge the batteries we use the SMPS in a buck configuration, and the voltage is therefore stepped down. Thus, if the voltage of the PV array is not higher than the charging voltage of the battery then we won’t be able to charge at all. The advantage of parallel connections is that a change in the current of one cell will not impact the current of other cells. However, at the maximum power point of the array as a whole, each individual cell might not be operating at its own maximum power point. Disadvantage, SMPS losses might be high at low voltages.

Remember that capacitor on SMPS input will be able to hold some energy. At 62.5 kHz it holds enough power that input power is constant no matter the duty cycle.

What is the maximum power drawn by the batteries? There are 5 batteries and we charge at 250 mA:

In addition to this there will be losses in the circuit. However, the rated power of the PV cells together is:

So we need to draw as much power as possible out of the solar panels. Might not be able to use the full 5 battery cells.

<https://www.sciencedirect.com/science/article/pii/S0360544211001484?casa_token=aN6AlhJsx9IAAAAA:yUMOdvzscbw5ltokpvOcWVfY8IOHd0nr_6eLwivW_ZHVWAsjFMjRJ7ihyQtg2kn25_U9QIG5yg> , configuration of solar panels

<https://www.sciencedirect.com/science/article/pii/S0038092X16300111>

Testing both configurations and comparing them. Test each multiple times.

**Configuration of battery cells:**

When designing the battery pack there are two principal choices that need to be made. Firstly, how many battery cells should the battery pack consist of and, secondly, in what manner should these cells be connected?

The optimal number of cells is in large part set by the energy and power needs of other submodules. During testing the total power draw of the rover was found to be about 2 W. Each battery cell has a nominal voltage of 3.2 V and a maximum peak discharge current of 1 A [battery datasheet], giving a maximum power draw of 3.2 W per cell. Thus, to meet the power requirements of the rover it is enough to use only one cell. However, each cell can only store about 3.2 V \* 0.5 A \* 3600 s = 5760 J, which would only power the rover for 48 minutes. It is desirable to have the rover be operational for as many hours as possible each day. Assuming that 12 hours a day are completely without sunlight it is clear that the rover cannot work through the night even if all 5 available battery cells were used. To give the rover the most operational hours we would then want to use all 5 battery cells. However, connecting 5 battery boards to the Arduino would use at least 11 of the 12 free Arduino pins, leaving only one pin for all other purposes. As such it might be wise to use less cells. To accommodate other circuit functions, only 4 cells will therefore be used.

Now consider the PV array. Each PV panel is rated for 1.15 W, meaning that the array as a whole should produce 4.6 W. As shown in figure ? the peak efficiency of the SMPS in boost mode is about 80% giving a maximum usable power of 0.80\*4.6 W ≈ 3.7 W. Assuming 12 hours of sunlight in a day, this means that the PV array will produce less energy each day than the rover uses in 24 hours. It is therefore of high priority to capture as much of the solar power as possible. The battery cells have a standard charging current of 250 mA [battery datasheet]. The power needed to charge the at this current is shown in the table below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number of cells | 1 | 2 | 3 | 4 | 5 |
| Nominal charging power (V\_cell = 3.2V) | 0.8 | 1.6 | 2.4 | 3.2 | 4.0 |
| Peak charging power (V\_cell = 3.6 V) | 0.9 | 1.8 | 2.7 | 3.6 | 4.5 |

A 4 cell battery pack is the largest battery pack which has a power requirement of less than 3.7 W at standard charging current and little power goes unused. Adding to this that the rover has most operational hours with 4 cells, using 4 cells is the clear choice. Note that the cells used are number 1, 2, 4 and 5 as these were found to have the highest capacity.

There are several ways to connect the 4 cells into a power pack, however the design brief advised against mixing parallel and series connections [phils video]. As such only fully series and parallel battery packs will be considered.

A series battery pack has some obvious disadvantages compared to a parallel battery pack. In section 2.5.1 it was shown that all the battery cells have different capacities. In a parallel battery pack this is not a big problem, as it is possible to draw different currents from each cell and thereby use the full capacity of each cell. For a series battery pack however, the total battery capacity will be limited by the cell with the lowest capacity. As such, a series battery pack is able to store less usable energy than an equivalent parallel battery pack. Moreover, to check the OCV of each cell they need to be switched out of circuit using the battery board relay. For a series battery pack this leads to an open circuit and any charging/discharging of the battery must halt while the voltage is measured. For a parallel battery pack on the other hand, switching the relay of a cell only takes that one cell out of circuit and it is possible to charge/discharge the battery pack while taking voltage measurements. Being able to switch single cells out of circuit can also enables switching faulty cells out of circuit, meaning that one cell failing would not mean that the battery pack as a whole fails. Finally, a parallel battery pack is self-balancing and energy automatically flows between cells when necessary [needs reference]. Series cells on the other hand need to be balanced by switching on and off balancing resistor. This not only is more complex, but leads to power being lost in the balancing resistors.

There is however one major weakness to a parallel battery pack: it is very hard to track the current going into individual cells. For safe operation it is necessary to prevent over-current into each individual cell. However, the current sensor on the SMPS can only measure the current flowing into the battery pack as a whole. Each battery cell is only rated for a rapid charging current of 500 mA [battery cell datasheet], and seeing as there is no way to know how the current splits into each of the cells one must operate one must operate with the assumption that all the current can flow into a single cell. As such, no more than 500 mA can be allowed to flow into the battery pack as a whole. At this current the nominal charge power is only 3.2V\*0.5W = 1.6 W meaning charging will be slow and it is only possible to use less than half of the available solar power. This is such a large drawback of using a parallel battery pack, that despite all the previously stated disadvantages of a series battery pack, a series battery pack has been deemed the best option.

To prevent over-current

There is only one proper current sensor available in the circuit, which can only measure the current flowing into the battery pack as a whole. For safety reasons it is necessary to prevent over-currents into each individual cell. With only a curr

, but this is needlessly

This is a problem as for

In series can’t measure cell voltage while charging as current has nowhere to go. In parallel we can switch out a single cell and keep charging the others

No way to detect over-current in parallel

Despite the many disadvantages of

the current flowing through one cell must also flow through the other cells. In a series battery pack the battery capacity will therefore be limited by the cell with the lowest capacity. As such, a series battery pack is able to store less usable energy than an equivalent parallel battery pack. Moreover,

Moreover, a parallel pack is self-balancing making it easier to perform

When the weakest cell has been fully discharged there will

This means that a cell can be fully discharged while others still have some charge in them. This remaining charge cannot flow out of the battery pack as doing so would discharge the weaker cell beyond empty. As such, a series battery pack is able to store less energy than

NEED less SOH maintenance

Parallel battery packs have many advantages. They are self-balancing

In parallel we can disconnect a faulty cell, in that way the battery pack as a whole can keep functioning at a reduced capacity, even if one or multiple of the cells die.

(Constant voltage hard in series power pack., need not mention)

No way to detect over-current in parallel

In series can’t measure cell voltage while charging as current has nowhere to go. In parallel we can switch out a single cell and keep charging the others

A choice of 4 cells also works well with the PV array. As each panel is rated for 1.15 W, put together the PV array is expected to provide 4.6 W. As shown in figure ? the peak efficiency of the SMPS is about 80% giving 0.80\*4.6 W ≈ 3.7 W of usable power on the SMPS output. When the batteries are close to fully charged they have a voltage of about 3.6 V, charging at the standard charging current of 250 mA [battery datasheet] a 4 cell battery pack will then need 4\*3.6\*0.25 = 3.6 W. Which is very close to the 3.7 W of usable power. As such, with a battery pack of 4 cells

is the maximum number of cells that can be used while still being able to use a using four cells high degree of utilisation.

battery cells have a standard charging current of 250 mA

A choice of 4 cells therefore

The drive submodule has the highest energy consumption with a peak power of 1.1 W, the DE10 FPGA used for vision consumes 0.4 W and the ESP32 from control. Factoring in .

However, each cell can only store about 3.2 \* 0.5 \* 3600 = 5760 J, which would only power the rover for …. Hours. It is desirable to have the rover be operational for as many hours as possible each day. Assuming each

A choice of 4 battery cells also works well with the PV array.

submodules have a peak power draw of about Of the other submodules the drive module has the highest energy consumption with a peak power of about 1.1 W.

Each battery cell has a nominal voltage of 3.2 V and a maximum peak discharge current of 1 A [battery datasheet], giving a maximum power draw of 3.2 W per cell. To cover the power needs of the drive system we therefore need to use at least 4 cells, giving a maximum power draw of 4\*3.2 V \* 1 A = 12.8 W. Thus with … cells the power requirements are meet.

Have also been advised not to use all cells at once.

Testing of the drive revealed that

To determine the number of cells it is necessary to consider the energy and power needs of the other submodules.

Testing on the drive module

Note that it is best to use the cells with the highest capacity, so will use cells 1,2,4 and 5

Use all available energy

Provide enough power

For the longest operation in a day

We want to capture as much as possible of the solar power

The rover has a quite high power consumption and drive alone can reach a peak power of ~10 W. 12.75 W. For high power output we need many cells

USE THE LED ON THE ARDUINO ITSELF TO GIVE STATE, number of blinks indicate state\_num. Got rid of LEDs to get more ports, (or maybe I will just use 3 batteries?)

Using boost also has the advantage of being able to test the charging part of the system with the power supply rather than with the solar panels, which do not perform that well when not directly lit with sunlight.

Two choices: either use 0.5 ohm load as current sensor or have parallel battery pack with pulse charging. Or use series connected pv panels (Choose one based on what leads to the smallest loss, using series connected PV panels or using 0.5 ohm resistor as current sensor. Use an op-amp unit gain to give voltage difference, what is it powered by? Simply power it from 5V output of Arduino. 0.5 ohm load current sensor is very hard to implement in a way that gives accurate measurements for both positive and negative currents (with the equipment we have available).

Actually use internal current sensor, assuming the inductor current is constant, which it is on the time scales we are working on. Use formula I\_out = I\_L\*duty\_cycle

<https://www.electronics-tutorials.ws/opamp/opamp_5.html>

How much energy can the batteries hold and how much energy can be produced in a sol. What is the maximum power output of solar panel versus how much power is needed to charge batteries.

Maybe not use the full 5 because then there are no spares. Probably going to use 4. Use the cells with the highest capacity. Also not enough Arduino outputs anyway

In design brief we have been advised as to not mix parallel and series connection and will abide by the advice.

Check if charging at a lower current has any effect on capacity, if not, then we can charge more cells, just at a lower rate. Then we can use the full battery-pack.

Can also provide less power if in series, as maximum current is far lower, no because voltage is higher.

In parallel we can disconnect a faulty cell, in that way the battery pack as a whole can keep functioning at a reduced capacity, even if one or multiple of the cells die.

No way to detect over-current in parallel

In series can’t measure cell voltage while charging as current has nowhere to go.

**Charging algorithm**

Constant voltage is not very power hungry and is only used to top up on the charge.

CC/CV is the recommended charging method mentioned on the datasheet:

<https://www.ampsplus.co.uk/ampsplus-14500-3-2v-500mah-battery-button>

Pulsed CC/CV, for CV charging in a battery pack the voltage across individual cells are not necessarily constant due to different impedances responding differently to current decreasing. Would lead to some cells racing away from 3600 mV, often hitting more than 3700 mV triggering an error. Switched from keeping the output voltage stable to try to keep the individual cells voltages stable at 3600 mV.

In balance state, we first try to get every cell up to 3600 mV, then we keep them at 3600 mV until the current has fallen below 30 mA, which is about the current that maximum can be dissipated by the internal resistances. Current slowly decreases.

Balancing doesn’t take too much power or time as they tend to stay fairly balanced after being balanced once.

Actually measured higher capacity for pulsed! Charging time is the same

There are many ways to charge batteries: (can I call it a lithium-ion, don’t think so)

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6360973>

However, many of them are hard to implement directly in software and either need hardware, which we do not have available, or need extensive knowledge of the battery behaviours, that could be a project in itself in determining. Easy to implement and widely used is, CC/CV.

Maybe do pulse charging for parallel cells. Maybe offset relay switching by a couple of milliseconds as to limit current drawn from Arduino. (staggered switching)

Probably do some integral thingy to determine if a cell is consistently higher voltage then other cells, which will then trigger the disp output and initiate passive balancing.

SMPS more efficient at higher currents?

Relay: <http://www.farnell.com/datasheets/1717878.pdf>

6 ms to turn on, 8 ms to stabilise voltage, 2ms hold time, 4 ms to turn off

Constant current is used to eliminate imbalance of cells.

Actually need to measure Vb as well so that we can use it for constant voltage.

Could probably tell if light on solar panels increase by looking at the duty cycle. Maybe change reference if under 90% of current reference current after some time.

<https://pubs.rsc.org/en/content/articlepdf/2018/ta/c8ta00962g>

pulse charging

Even improves charging speed:

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6646717>

As shown in this paper pulse charging can be combined with constant voltage at the end, but still pulsating. Little negative impact. A lot of good stuff in this article.

<https://www.researchgate.net/publication/224326763_Design_of_Duty-Varied_Voltage_Pulse_Charger_for_Improving_Li-Ion_Battery-Charging_Response>

More efficient charging, but a bit lower capacity

<http://www.kohl.chbe.gatech.edu/sites/default/files/linked_files/publications/2001_The%20effects%20of%20pulse%20charging%20on%20cylcing%20characteristics%20of%20commercial%20lithium-ion%20batteries.pdf>

Faster charging, and longer lifetimes.

1 Hz frequency: (This uses lead battery, so depending on chemistry of battery), 0.8 duty cycle is used here.

<https://core.ac.uk/download/pdf/61010268.pdf>

50% duty cycle is the best

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8294255>

Should do with multiple duty cycles and see if 80% is better.

Everything from 1 Hz, to many kHz

<https://www.researchgate.net/figure/Optimal-pulse-charge-frequency_fig7_3218839>

Change to a Mealy machine, so that outputs can change at a faster frequency than state.

Measure voltage only at certain times, but measure current all the time.

Stop constant voltage charging when average current is less than 3% or something of normal current.

**0.5 Ohm current sensor**

Need to low-pass filter either digitally or in circuit.

**SOC:**

The state of charge (SOC) of a battery is defined as the remaining usable charge given as a percentage of the battery’s total charge capacity [10]. There are many methods for estimating the SOC of a battery, the most common of which rely on measurements of the voltage and/or current of the battery [11]. The perhaps simplest SOC estimation method is to measure the open circuit voltage (OCV) of the battery and then calculate the SOC using a formula or a lookup table. For many types of battery this is a good estimation method. An example is lead-acid batteries, for which the OCV varies approximately linearly with the SOC. However, as was shown figure ~\ref{fig:efficiency}, this is not the case for LiFePO4 batteries. For LiFePO4 cells the voltage is nearly constant for a majority of each charge/discharge cycle. Any measurement error or change in OCV due to current recently flowing through the battery, would then produce large SOC estimation errors. This holds true for all SOC methods relying purely on voltage measurements and they are therefore not good alternatives.

An alternative SOC estimation method is Coulomb counting, where the current flowing through the battery is integrated to find the net charge that has left or entered the battery. Seeing as charge is a conserved quantity nearly all charge put into the battery will be available during discharge. As such, the SOC will vary nearly perfectly linearly with the integrated current. The sources of error for Coulomb counting are mainly the Coulombic efficiency of the batteries and current measurement errors. However, using correction methods the error can be kept small, on the order of 1-2% [12]. Given the simplicity of the estimation method, this is a very small error. Other estimation methods, such as Kalman filters and neural networks, are claimed to give higher estimation accuracies [11]. However given the already high accuracy of Coulomb counting the improvement is marginal. Moreover, they are far more complex both computationally and in implementation. As such, Coulomb counting was deemed the best option for SOC estimation.

SOC is a measure of charge and not energy. It is necessary to estimate the range of the rover. The obvious way of doing this is to track the power consumption and speed of the rover. Dividing the energy left in the batteries by the power and multiplying by the speed gives an estimate of the range under the current operating conditions. The problem however, is that the SOC only gives the usable current and not the usable energy left in the battery. As the SOC of the battery decreases, so too does it voltage, meaning that at lower states of charge each mAh will provide less energy than at higher states of charge. To account for this, during discharging the BMS logs the energy output of the battery alongside the SOC. From this data it is possible to create a lookup table which relates the current SOC to how much energy it is still possible to draw from the battery. This is the method which will be used to find the energy left in the battery.

Current through the dissipation resistors will be so small that it is not necessary to

take into account.

The SOC is not directly proportional to the usable energy in the battery. SOC gives how much current and not how much

Though

However, it still holds true that the charge towards the end of the cycle has less energy as the cell voltages are lower, by as much as 1/3 compared to the maximum voltage. (i.e. maximum voltage is 50% larger). Thus, we can make the estimation better by making SOC table non-linear. (BUT WE NEED TO KNOW HOW MUCH ENERGY IS LEFT DO RANGE ESTIMATION).

Adaptive system, 1% update each cycle

Range estimation

[10]

<https://www.sciencedirect.com/science/article/pii/B9780444527455008777>

[12] Coulombic efficiency is usually larger than 99% and with correction methods this will not have a larger impact, even after many cycles.

<https://www.sciencedirect.com/science/article/pii/S0306261908003061#aep-section-id25>

Would have logged more often but gave issues with speed of programme. 17% slower than it should be. Log the average current for the past second instead of actual current at any given point.

Voltage measurements only have a accuracy of ~4 mV

Method for SOC:[11]

<https://www.sciencedirect.com/science/article/pii/S2352146519301905>

Methods relying solely on the cell voltages does not work well due to the flat part of the voltage curve during charging/discharging.

Impedance method could have worked, but SMPS might not be quick enough to give the current accurately in 10ms. Very hard to measure the internal resistance accurately.

Coulomb counting is good and easy to implement, the problem is that it gives the charge stored, not the energy, which we need to estimate the range of the rover and other stuff. Energy stored is more useful than current stored. We therefore use a hybrid method combining current and voltage.

Note however, that energy in is not necessarily equal to energy out. Measure how much energy we are able to draw from the battery. The problem with having the SOC rely on voltage measurements is that the voltage depends heavily on what current is currently being drawn or has recently been drawn. We see this from the characterisation of the batteries for which in the charging rest state the battery voltage drops immensely even though power is being drawn from the battery. However, it still holds true that the charge towards the end of the cycle has less energy as the cell voltages are lower, by as much as 1/3 compared to the maximum voltage. (i.e. maximum voltage is 50% larger). Thus, we can make the estimation better by making SOC table non-linear. (BUT WE NEED TO KNOW HOW MUCH ENERGY IS LEFT DO RANGE ESTIMATION).

Adaptive system, 1% update each cycle

Range estimation

**SOH:**

The state of health of a battery is a measure of its current condition and performance compared to when it was new [1]. Indicators of a battery’s state of health include battery charge capacity, energy capacity, cell voltage balance, and the number of completed charge/discharge cycles [2]. All these indicators are tracked during operation and stored on the SD-card as to retain data when the Arduino is not being powered. Over the course of its lifetime the SOH of a battery will naturally degrade. However, through SOH maintenance the degradation can be slowed significantly. Most importantly for a series battery pack is to keep the battery cells balanced, as unbalanced battery cells lead to lower capacity and faster cell degradation [3].

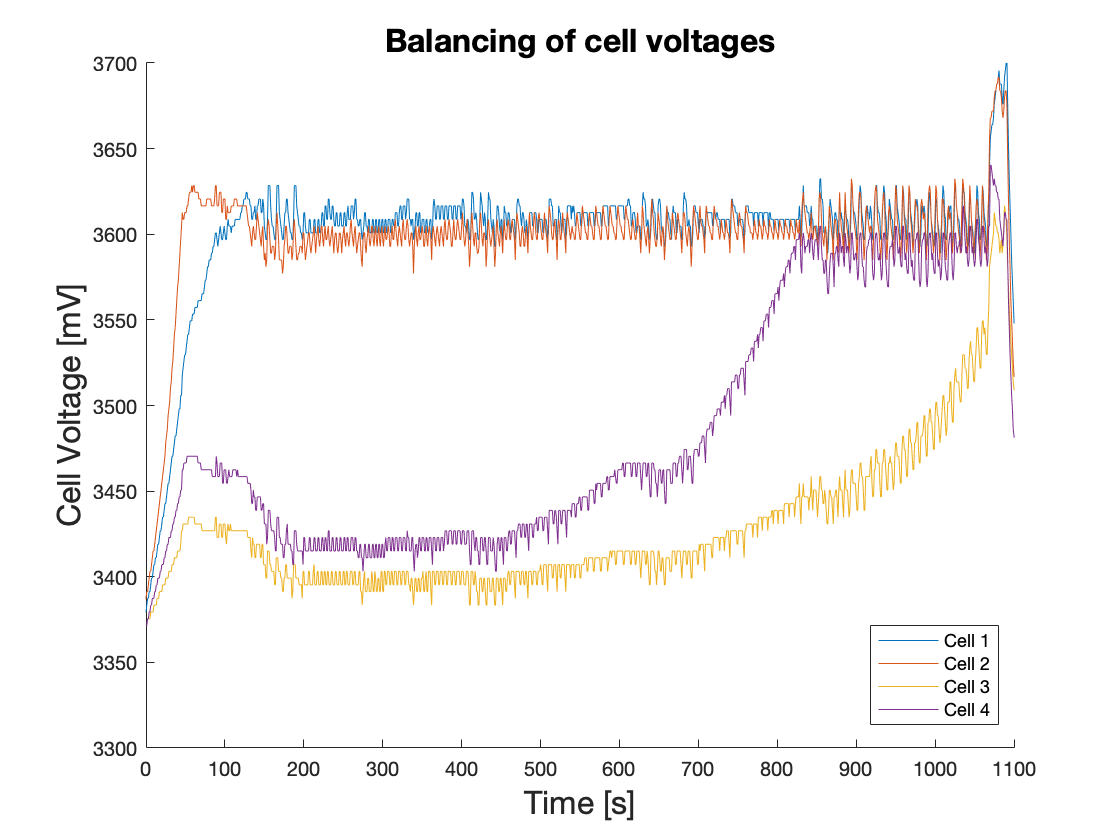
To facilitate balancing, each of the provided battery boards have mounted resistors which through a MOSFET can be connected to the battery cell. Connecting a cell to the resistors on the battery board will lead to a small extra current flowing out of the cell, slowly discharging it. During operation one must decide when to switch said resistors on and off to keep the cells balanced. Usually, balancing is only done towards the end of a charge cycle [3]. There are several reasons for this. Firstly, passive balancing requires energy to be expended and will therefore reduce the total amount of usable energy in a battery if done during discharging. Secondly, differences in impedance and charge curves between cells might make it look as though a cell is charged more than others, but the voltage difference might disappear naturally as the battery is charged more or as charge current is reduced towards the end of a charge cycle. As such, the implemented charging algorithm only does balancing during the constant voltage part of charging. The balancing is done using the code shown in Figure ?. During CV charging it is desirable that all cells have a voltage of 3600 mV. If any cell has a voltage which is higher than 3600 mV its voltage is too high and the dissipative resistors are switched on to lower the voltage to the voltage set point.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Alongside the constant voltage charging algorithm this code was extremely successful at balancing the cells. The balancing is shown in action in figure ?. One by one, the cell voltages are brought up to and kept at 3600 mV. Even with an initial voltage difference of 200 mV, balancing only takes about 1000 seconds ≈ 17 minutes.

To track the SOH the charging and discharging

Disregard spike at end, not part of algorithm.

between batteries might make a battery appear as though it is at a higher

during discharging lowers the maximum usable energy in a cell as some energy will be dissipated as heat.

The data used to keep track of the SOH of the battery is

To keep track of the SOH data regarding

h can be switched on

For the series battery pack passive balancing is performed using the resistors on the battery boards. During charging and discharging all cell voltages are continually tracked. Once the battery is nearly fully charged, balancing is initiated.

Note that for the given battery pack it is not necessary to balance the cells during discharging. Takes power and makes them unbalanced when we start to charge again.

Cells can

How I balanced:

Using passive balancing

Only at high SOC

Only during charging.

How I tracked

What is sent to command

Stored on SD card

Passive balancing battery cells

To track the SOH

To prevent degrading

Balancing not only keeps batteries from degrading, but also allows the storage of more energy.

[1]

<https://www.mpoweruk.com/soh.htm>

[3]

<http://www.amarketplaceofideas.com/wp-content/uploads/2018/09/Topic20220-20Battery20Cell20Balancing20-20What20to20Balance20and20How1.pdf>

If not balanced, some cells will degrade faster.

compared with the con compared to initial conditions. There

Many different way to estimate SOH: [2]

<https://onlinelibrary.wiley.com/doi/epdf/10.1002/er.3598?saml_referrer>

However, most methods require a lot of data about the battery cells, especially lifetime measurements which we do not have access to as there is not enough time in the project to go through a full life cycle of even a single cell.

Coulomb counting, i.e. comparing current capacity to initial capacity is good and easy to implement.

OCV does not work due to the large amount of data about the battery cells needed.

Maybe interpolate inbetween.

Other sign of health is whether the cells are balanced which is why we store data about this.

We store:

How much capacity has already been used, assuming continuous operation or only brief interruptions so this will work well, we can reset once we are fully charged or discharged.

Number of cycles,

Current maximum capacity

Initial maximum capacity

Cell1 voltage

Cell2 voltage

Cell3 voltage

Cell4 voltage

Write every ten seconds just in case the system loses power, that way information will still be retained.

**Cell balancing:**

Balancing state is only used if there is a severe unbalance between the cells (more than 50 mV difference between the cells), if the unbalance is small we move on to the constant voltage state where some balancing is also done.

Only for when charging:

<https://www.batterypoweronline.com/blogs/why-proper-cell-balancing-is-necessary-in-battery-packs/>

Cell balancing during discharging would just use energy and decrease usable power.

Discharging stops when the first cell reaches the minimum acceptable voltage.

Gives a higher capacity and allows us to extract more energy from the battery pack.

Hysterisis loop is not very computationally expensive (was already pushing Arduino to its limits), very easy to implement.

At a balancing coefficient of 0.1, there is no net current through cells with too high a voltage, but current flowing through other cells.

<http://www.amarketplaceofideas.com/wp-content/uploads/2018/09/Topic20220-20Battery20Cell20Balancing20-20What20to20Balance20and20How1.pdf>

Unbalanced cells gives:

Premature cell degradation

Safety hazard

Reduced capacity

Early discharge termination

* **Balancing at high states of charge only** is used to decrease the effect on SOC balancing that can come from impedance unbalance.

Will probably have very similar charge profile as last time, which will hopefully get us to the correct voltage without balancing.

**SMPS configuration**

As the voltage of the battery pack is higher than the voltage of the series connected battery cells, the voltage of the solar panels must be stepped up when used for charging. This means that the SMPS must be operated in boost mode with the solar panels on port B and battery on port A. Moreover, as the SMPS has a max output voltage of 7 V in synchronous boost operation [power logbook], it must be operated in non-synchronous mode.

Talk about current sensor here.

s the if the SMPS is operated in synchronous mode then the PMOS would turn itself on with

as the PMOS will turn itself on if the [power logbook]

the SMPS must be operated in non-synchronous mode

a parallel solar array and a series battery pack is used it is necessary to step up the voltage

Can’t be in synchronous mode due to high output voltage?

Also if power to Arduino is lost then immense currents would flow, destroying the batteries.

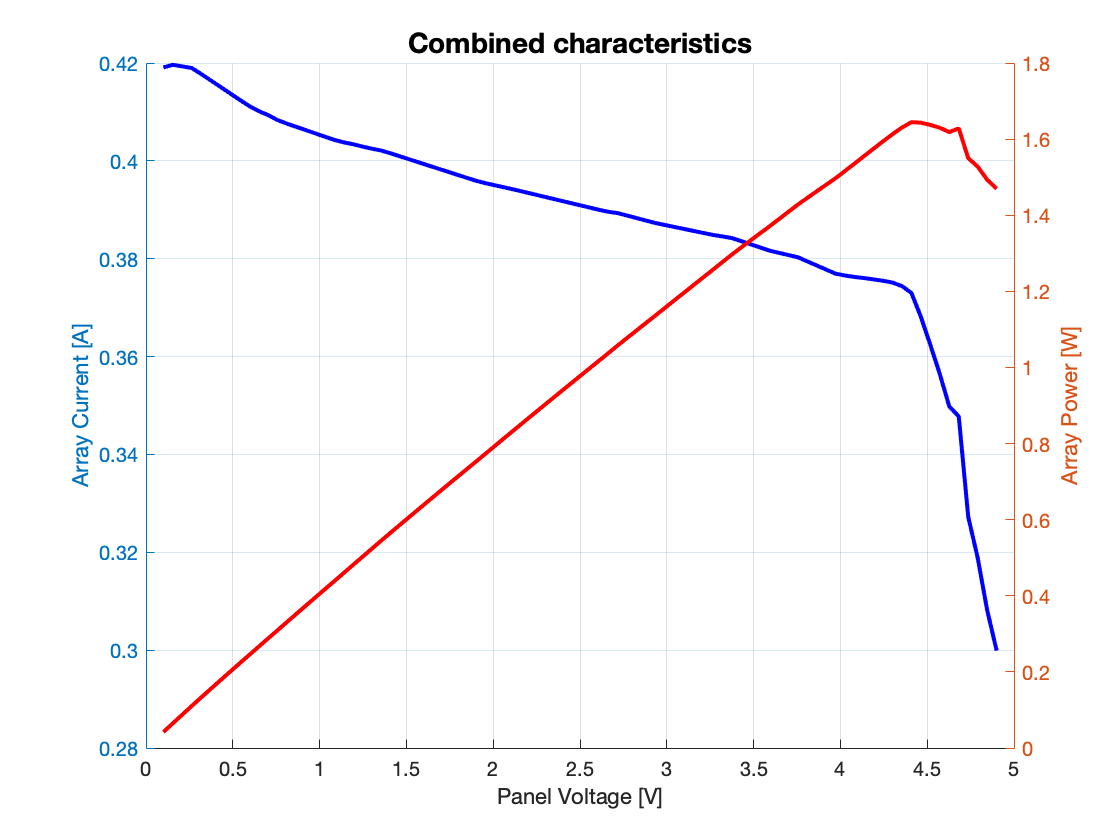
Current sensor on wrong side if using boost. Need to find way around that

**MPPT:**

When power is provided by solar panels it is common to use two-stage power conversion before exporting the power. The first stage of the power converter does maximum power point tracking, so that we can get the most power out of the solar panels, the second stage conditions the power to the correct voltage or current (2). For the energy submodule only one SMPS device was available. Therefore it was not possible to implement the standard two-stage power converter. However, this is not a problem seeing as the goal of the PV panels is not to output the maximum amount of power.

For the charging algorithm for the most part the output power is fairly constant. It then does not matter to us if the solar panels can provide a bit more power than they do currently. In total the provided power is:

This is for parallel



Though we do not need maximum power point tracking, we still need some way to determine the necessary duty cycle. The charging algorithm has two parts for which we need to consider the operation of the SMPS: constant current mode and constant voltage mode.

Assuming boost configuration:

Now let us make the following considerations; in constant current mode, if the current is a bit low we would try to increase the current by increasing the output voltage, by increasing the duty cycle. As both the current and the voltage increases on the output, the output power increases. Assuming 100% efficiency, so too must the power on the input increase. If the duty cycle of the SMPS increases that means that the input impedance decreases, no matter if it is buck or boost. When the input impedance decreases, the input current will increase.

Likewise for constant voltage mode, if the voltage is a bit low, we would increase the output voltage, by increasing the duty cycle, which would increase the output current. As both output current and output voltage increases the output power must increase. When the duty cycle increases, the input impedance decreases and so the input current will increase.

In both situations the output power increases with increased duty cycle, while the input current increases. Assuming 100%, the increase in output power means that the input power must increase. Now let us inspect the curve above. If more current is drawn, then the power only increases if we are to the right of the maximum power peak. If we are to the left of the power peak the increase in current will cause the input power to decrease, which will cause the output power to decrease, the opposite of what we wanted.

From this analysis we see that we want to operate to the right of the power peak, then any time the current reference increases we will move closer to the power peak. If we try to demand more power from the solar panels than they can increase, then the duty cycle will raise to 1. By detecting whether the duty cycle is 1 then we can determine whether the PV array is able to supply the demanded power. If the duty cycle is 1, we know that the PV array cannot supply the demanded current. What we do then is set a lower current reference and reset the duty cycle to 0 for which the output power is 0, the duty cycle will then increase until we find an equilibrium. If we find an equilibrium, great, if not the duty cycle will go to 1 again and we need to set an even lower current reference.

Also every now and then we can check if we can use a higher current by doing simply setting a higher current reference and seeing if an equilibrium exists. The test should come less and less frequently if stepping up the power has failed many times before.

(Maybe roll duty cycle all the way back to 0, when searching for another current which works!, this might be the easiest to implement, if duty cycle is 1 then we need lower current-reference, maybe use a current reference multiplier)

<https://www.sciencedirect.com/science/article/pii/S1364032117305750>

**Statement of own performance**

Energy was pretty separate, for the most part needed access to hardware to do stuff. By the nature of being in different countries, a lot of the work needed hardware. Felt like I was given a workload which was perhaps too large and I know of many energy students who severely struggled to finish on time.

Ended up doing the entire energy module all by myself.

**Integrating with the rest of the rover**

Due to students being in different locations the energy module will not be physically integrated with the full rover. However, outlined below is one way the energy module could be physically connected if necessary.

The energy module can either be integrated as a charging station or, as is more common for mars rovers, be mounted directly on the rover itself. The advantage of implementing it as a charging station is that one would only ever charge or discharge the battery at a given time. This makes it easier to track the current and power in and out of the system and create charging/discharging algorithms. There are two major drawbacks to using charging station. Firstly, the rover would get a severely reduced range as it always needs to be able to make it back to the charging station before its battery is depleted. Secondly, if the rover is detached from the charging station, how will the energy module be able to track the cell voltages and SOC. It might be possible to create a system where different microcontrollers keep track of energy when the rover is connected and when it is not connected to the charging station. However, a likely simpler solution is to mount the energy module on the rover. A mounted system will increase the range, but requires the ability to charge and discharge the battery at the same time.

Assuming that a mounted solution is used there are two main design problems to be solved. Firstly, how will power be supplied to each of the rover components and secondly, how is simultaneous charging and discharging facilitated?

As shown in Figure ? the rover has 4 separate voltage regions. The battery and PV array have already been discussed. The new regions are the 5V node used to power the FPGA and micro-controllers, and the variable voltage node used to power the motors. Currently, the 5V used to power the control circuitry is also used to power the drive SMPS. The most straight forward way to connect the energy sub-module to the rest of the rover would therefore be to step down the battery voltage to 5V using a buck SMPS and connect the SMPS output where the battery pack is currently connected. However, this is very energy inefficient as the power for the motor needs to pass through two SMPS devices before being used, potentially leading to high power losses. A better way of connecting the energy module would involve separating the drive SMPS from the 5V node. Instead the drive SMPS could be directly connected to the battery. Then another SMPS or linear voltage regulator would be connected to the battery to create 5V power for the control circuitry. A sketch diagram of the connections is shown in Figure ?. One problem exists however with this solution: The 10.0 – 14.4 V of the battery is higher than the SMPS maximum input voltage. A way to solve this would be to exchange the PMOS on the SMPS board for another NMOS, which would increase the maximum input voltage.

In this design there are three systems through which current either flows in or out of the battery. To keep track of the SOC and ensure that the batteries are being charged or discharged at an appropriate it is therefore necessary to collect current data for all the power converters. This data could be relayed through control and read in on the energy Arduino through UART. The net battery current can then be calculated by simply adding together the current passing through each of the power converters. The rest of the charge algorithm would not be affected by this. This implementation also integrates well with the integrated power point tracking algorithm and will make sure that the demanded charge current is not higher than what can be provided while simultaneously powering the rover.

To adapt the charging algorithm

would therefore simply be to connect a buck SMPS to the battery, use it to step down the battery voltage to 5V and connect

To achieve these voltages, the battery voltage needs to be stepped down.

are each of the rover components powered and secondly, how

there will be 4 separate voltage regions on the rover

This will increase the range, but make as will be seen

Remotely connect to the Arduino

include decreased range,

making for easier charge/discharge algorithms and

Of the two implementations, a charging station is likely to be simpler to implement. With a stationary charging station one would only need to worry about either charging or discharging at a given time. There are however some drawbacks to using a charging station. The first is the impact on the range of the rover. If the rover always has to return to a charging station before its batteries run out, it will never be able to move further than half the estimated range at full charge in any direction away from the charging station. Secondly, if the charging station is separate from the rest of the rover, then there is harder to monitoring the battery pack when it is not connected to the rover. It could be done by the ESP32, not ideal and negates the simplicity of the solution.

With an onboard implementation however, the system would need to handle charging and discharging at the same time.

Simple circuit diagram?

The advantage of implementing the energy module as a charging station is that it is much simpler.

How the energy module is integrated depends on whether

Every submodule requires some power, all of which must in some way originate in the energy submodule. Four separate voltage regions are nee

How these devices connect to one another depends on whether the

There are three voltage regions on the rover

of power and many parts of the rover use different parts of the

Track current from battery, all arduinos need to talk together. Logging net current out of the battery region.

Need to make the SMPS devices able to take the high input voltage.

Mount on rover

Could probably use a simple regulator for control logic.

Need 3 SMPS devices due to bad part selection. Arduino can’t run directly of battery pack either in series or parallel. An easy fix would be to simply buy an H-bridge which supports speed control directly. Actually, could do a PWM signal from the H-bridge to the motors and have it run of 5V, then we would only need 2 SMPS devices.

Everything might actually be able to run directly of 4 cells in series. No, think I need 5V for FPGA.

In real life would connect through UART, but as we can’t make physical connection we need to connect to database or something. USE UART CONNECTION ON USB, this will control section.

Would not use a Buck-Boost configuration, but simply a buck configuration. That way we could take the full 14 V input voltage.

Would have used the UART ports and not relayed through the computer.

Would probably also put a bigger capacitor on output.

**Communicating with Other Modules**

Though it is not necessary to fully integrate the energy module with the rest of the rover, other submodules, specifically command, needs access data such as the battery SOH and SOC. For communicating with other modules the Arduino shield has a set of UART ports. However, as group members were not in the same location it was not possible to physically connect the energy module to the rover, which is necessary to use UART. As such, an alternative approach was employed. First the Arduino was connected to a computer via USB. On the computer a Python script was run [8]. At the start the Python script establishes a connection to a server created by running a similar script on the command module [9]. After a connection has been established the Python script starts reading the serial data coming from the Arduino and transmits it using TCP to the command module. Each message coming from the Arduino is in CSV form where the first entry is the message ID, which allows the command script to decode what type of data is being sent.

Send information about the estimated amount of joules left in the system. Command takes power usage and speed from drive and estimates range.

**Implementation**

How is

Get SOC from SD-card on start up, update battery capacity, reset SOC and energy capacity every cycle. Updates both as the battery degrades and as the discharge profile becomes more clear. Measurement error was not found to be a big problem.

An op-amp was added to buffer the voltage as after passing through the second voltage divider, the current drawn from the Arduino affected the measured voltage.

Should probably change to 10k and 15k. If smaller, will affect first voltage divider

Calculate current rather than using 0.5 ohm resistor as current sensor. Current sensor is on the wrong side.

**Safety mechanisms:**

Needs to shut itself down when too much power has been drawn, if not we might damage batteries.

Switch one of the cells out of circuit if current is too high for too long. Or simply stop charging, then wait until balanced using internal resistors.

**Sources:**

(2) <https://bb.imperial.ac.uk/bbcswebdav/pid-2060823-dt-content-rid-8486224_1/courses/10435.202020/2%20Notes%20-%20Photovoltaic%20Energy%20-%20ELEC50012%2020-21%281%29.pdf>

(3) Edvard’s power logbook

(4) Power lab instructions v0.99

(5) <https://static.rapidonline.com/pdf/502676_v1.pdf>

(6) <https://d1wqtxts1xzle7.cloudfront.net/1759451/2006Pruessner_Solar_Panel_book_chapter.pdf?response-content-disposition=inline%3B+filename%3DSolar_Panel_Obscuration_by_Dust_and_Dust.pdf&Expires=1621616316&Signature=JwZzU8EQLWboK57iyasZbxDiV4Gi8jSoq9Hr0M5q4eA6G4VtWfkDxhSe~-OG~xerMmS24AdTGWpZV-74hYKt-0jOZhFXLNZr6K3B69Sck5HvhhblMlI1oGC5PrtGi8LDKh5l1iYvNsZH8DMIaob79VVOwP8g3U0nrq1o4Gtwb0xvh3WuWcMH0wNe4URsHGrHGn5v2sfjwVHGhK6fvdrRrJDwEG6BtQcN7CWz3P1~kBeSSwQ10eY8YVsLAR1~xGbJ2yLayR4rAZWIZZCo1EB7MTZxJ3TLOS-4bcaX1l7pdJ4Xu2jjsYqX32gh6FUBdbKnykJ1ZO0CThi9MIXMIS-a3w__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA>

Dust on Mars

(7) <https://www.sciencedirect.com/science/article/pii/S0038092X16300111#s0055> Solar panel, partial shading

[8] Raghav’s Arduino script

**Appendix:**

# Final version:

The SMPS has been thourgohly characterised in power labs and will therefore not be further examined here.

**Requirements**

The energy submodule must:

1. Design a battery pack and charge it using solar power
2. Track SOC data and use it to estimate the range of the rover
3. Track SOH data and complete necessary SOH maintenance
4. Integrate into the main data mailbox
5. Consider how to physically integrate with the rest of the rover

**Characterising Components**

The energy system consists of three main components: the battery cells, the PV panels and the SMPS. To ensure that appropriate design choices are made it is necessary to determine the behaviour and limitations of these components.

*Battery Cells*

To determine their behaviour, each cell was tracked through a full charge/discharge cycle using the provided “Battery\\_Charge\\_Cycle\\_Logged\\_V1.1.ino” code\cite{chargeCode}. Plotting the obtained data, all cells were found to have similar graphs for voltage versus time. Figure~\ref{fig:charge\_cycle} shows the voltage evolution of a battery cell for a full discharge/charge cycle.

In addition to logging the cell voltage, the provided charging algorithm also logs the charging current. By integrating said current for a full charge or discharge section we can determine a cells capacity in mAh. The results from performing this analysis on each cell are shown in the table below:

As expected all cells have a capacity of about 500 mAh. However, some cells have a higher capacity than others, which may have implications for the performance for certain battery configurations.

*PV panels*

The PV panels are rated for a maximum power of 1.15 W at a voltage of 5.0 V and current 230 mA. Away from the maximum power point the performance of the panels can be determined from their IV curves. To find the IV curves, each panel was connected on the B-side of the SMPS operating in non-synchronous boost. They were then lit by the lamp and the duty cycle of the SMPS was swept while measurements of panel current and voltage were taken. The resulting data was processed and is plotted in Figure~\ref{fig:IV\_curve}.

Even though the data is noisy, it is clear that all panels exhibit the standard IV characteristics of a PV cell. That is, they behave as non-ideal current sources with a nearly constant current at low voltages and a rapid current reduction at high voltages\cite{green}. Moreover, it is clear that the provided lamp activates the panels poorly as the peak power for each of the panels is only \approx0.5 W.

*SMPS*

The SMPS is rated for 10 W throughput with a maximum boost output voltage of 35 V and maximum output current of 10 A\cite{SMPS\_lab}. These ratings are far higher than what is needed to charge the battery and are not expected to impose limitations on the design of the energy module.

The many characteristics of the SMPS have been thoroughly examined in 2nd year labs. However, for the energy submodule the most important characteristic will be the SMPS efficiency during non-synchronous boost operation. A graph of efficiency versus output current is shown in Figure~\ref{fig:efficiency}.

**Initial Design**

*Battery configuration*

When designing the battery pack there are two principal choices that need to be made. Firstly, how many cells should be in the battery pack and secondly, in what manner should these cells be connected?

The optimal number of cells is in large part set by the power and energy needs of other submodules. During testing the total power draw of the rover was found to be about 2 W. Each battery cell has a nominal voltage of 3.2 V and a maximum peak discharge current of 1 A \cite{batteryDatasheet}, giving a maximum power draw of 3.2 W per cell. Thus, even with only a single cell the power requirements of the rover are met. However, each cell can only store about \(3.2 \unit{V} \* 0.5 \unit{A} \* 3600 \unit{s} \approx 5760 \unit{J} \), which would only power the rover for 48 minutes. It is desirable to have the rover be operational for as many hours as possible each day. Assuming that 12 hours a day are completely without sunlight it is clear that the rover cannot work through the night even if all 5 available

battery cells are used. To give the rover the most operational hours we would then want to use all 5 battery cells. However, connecting 5 battery boards to the Arduino would use at least 11 of the 12 free Arduino pins, leaving only one pin for all other purposes. As such it might be wise to not use all of the available cells in the battery pack.

Now consider the PV array. Each PV panel is rated for 1.15 W, meaning that the array as a whole should have a peak power of 4.6 W. As shown in Figure~\ref{fig:efficiency} the

peak efficiency of the SMPS in boost mode is about 80\% giving a maximum usable power on the SMPS output of \(0.8 \* 4.6 \unit{W} \approx 3.7 \unit{W} \). Assuming 12 hours of sunlight in a day, this means that the PV array produces less energy each day than the rover

uses in 24 hours. It is therefore of high priority to capture as much of the solar power as possible. The battery cells have a standard charging current of 250 mA \cite{batteryDatasheet}. The power needed to charge the battery pack at this current based on the number of cells is shown in table \ref{table:2}.

From the table we see that a battery pack using 4 cells has the highest power consumption without going over the limit of 3.7 W. Moreover, 4 cells is a good compromise between having enough free Arduino pins and providing enough energy for long operation. Taking all these factors into account, 4 cells seems like the appropriate number for the battery pack. Moreover, as cells 1, 2, 4 and 5 were found to have the highest capacity, these are the cells that will be used.

There are several ways to connect the 4 cells into a power pack, however the design brief advised against mixing parallel and series connections\cite{energyBrief}. As such only fully series and parallel battery packs will be considered.

A series battery pack has some obvious disadvantages compared to a parallel battery pack. Firstly, for a series battery pack the battery capacity is limited by the weakest cell. This is not the case for a parallel battery pack as it is possible to draw different currents from each cell. As a consequence, a series battery pack is able to store less usable energy than an equivalent parallel battery pack. Moreover, to check the OCV of each cell they need to be switched

out of circuit using the battery board relay. For a series battery pack this leads to an open circuit and any charging/discharging must halt while the voltage is measured. For a parallel battery pack on the other hand, switching the relay of a cell only takes that one cell out of circuit and it is still possible to charge/discharge while measuring cell voltages. Finally, a parallel battery pack is self-balancing\cite{batteryBalancing}, while series cells need to be manually balanced using, for example, balancing resistors. This not only is more complex, but also leads to energy being lost as heat during balancing.

There is however a major weakness to a parallel battery pack: it is very hard to track the current into individual cells. The current sensor on the SMPS can only measure the current

flowing into the battery pack as a whole and there is no way to know how the current splits between cells. To prevent over-current one must therefore operate with the assumption that all the current can flow into a single cell. Each cell is only rated for a max charging current of 500 mA\cite{batteryDatasheet}, which is therefore the maximum charging current. At 500 mA the nominal charge power is only \(3.2 \unit{V} \* 0.5 \unit{A} = 1.6 \unit{W} \), which is less than half the available solar power and with a parallel battery pack a lot of power will therefore go unused. As previously determined, energy is a precious resource and only being able to use half of the available solar power is such a large drawback, that despite the disadvantages of a series battery pack it is still deemed the best option.

*PV panels*

There are four ways in which the four PV panels can feasibly be connected. The possible arrangements are shown in figure~\ref{fig:arrayConfigurations}.

Consider first a pure series connection. During characterisation each PV panel was found to have a max voltage of $\sim$5.5 V. In a series connection the array voltage will then be 20+ V. The nominal voltage of the series battery pack is \(4 \* 3.2 \unit{V} = 12.8 \unit{V} \). As the array voltage is higher than the battery voltage, the SMPS must be used in the buck configuration. However, the maximum buck input voltage is only 7 V\cite{PMOS} and therefore a series connected PV array cannot be used. Similarly, for the Series-Parallel and Total-Cross-Tied arrangements the maximum array voltage will be about 11 V. However, as the battery voltage can swing between 10 V and 14.4 in a charge cycle, neither a buck nor a boost configuration will be able be able to provide the necessary voltage range with an 11 V input voltages. Thus it is not possible to use either the Series-Parallel or Total-Cross-Tied configuration. This leaves a purely parallel connected PV array as the only viable option, which is why it has been chosen.

*Maximum Power Point Tracking*

The energy submodule only has access to a single SMPS device. At any time it will therefore only be possible to either perform MPPT or have the PV power be outputted at the correct current/voltage. This is not a problem, as the goal of the PV array is not to output the maximum amount of power, but simply to provide the power demanded by the charging algorithm. As such, the system does not need conventional MPPT. However, if the PV panels cannot provide the demanded power some sort of power tracking must be used.

Consider an SMPS being used to charge a battery pack at a set current. If the actual current on the output is lower than the setpoint, one would attempt to increase the output current by increasing the output voltage. This is achieved by increasing the duty cycle. Similarly, if the output current is too high one would attempt to lower the output current by lowering the duty cycle. From these considerations we see that increasing and decreasing the duty cycle is associated with higher and lower output power respectively. Now compare this with the IV and power characteristics of the parallel PV array shown in Figure ~\ref{fig:parallelArray}. For an SMPS, increasing the duty cycle will lower the input resistance, causing the input current to increase. As increasing the duty cycle increases the output power so too must increased input current lead to increased input power for an equilibrium to exist. However, increased input current only gives increased output current if the PV panels are operating in the region to the right of the maximum power point. Thus this is the region one would want the panels to operate in.

*State of Charge*

The state of charge (SOC) of a battery is defined as the remaining usable charge given as a percentage of the battery’s total charge capacity\cite{DICKINSON2009452}. There are many methods for estimating the SOC of a battery, the most common of which rely on measurements of the voltage and/or current of the battery\cite{DANKO2019186}. The perhaps simplest SOC estimation method is to measure the open circuit voltage (OCV) of the battery and then calculate the SOC using a formula or a lookup table. For many types of battery this is a good estimation method. An example is lead-acid batteries, for which the OCV varies approximately linearly with the SOC. However, as was shown figure ~\ref{fig:efficiency}, this is not the case for LiFePO4 batteries. For LiFePO4 cells the voltage is nearly constant for a majority of each charge/discharge cycle. Any measurement error or change in OCV due to current recently flowing through the battery, would therefore produce large SOC estimation errors. This holds true for all SOC methods relying purely on voltage measurements and they are therefore not good alternatives.

An alternative SOC estimation method is Coulomb counting, where the current flowing through the battery is integrated to find the net charge that has left or entered the battery. Seeing as charge is a conserved quantity nearly all charge put into the battery will be available during discharge. As such, the SOC will vary nearly perfectly linearly with the integrated current. The sources of error for Coulomb counting are mainly the Coulombic efficiency of the batteries and current measurement errors. However, using correction methods the error can be kept small, on the order of 1-2\%\cite{NG20091506}. Given the simplicity of the estimation method, this is a very small error. Other estimation methods, such as Kalman filters and neural networks, are claimed to give higher estimation accuracies\cite{DANKO2019186}. However given the already high accuracy of Coulomb counting the improvement is marginal. Moreover, they are far more complex both computationally and in implementation. As such, Coulomb counting was deemed the best option for SOC estimation.

*State of Health*

The state of health of a battery is a measure of its current condition and performance compared to when it was new\cite{mpower}. Indicators of a battery’s state of health include battery charge capacity, energy capacity, cell voltage balance, and the number of completed charge/discharge cycles\cite{https://doi.org/10.1002/er.3598}. Over the course of its lifetime the SOH of a battery will naturally degrade. However, through SOH maintenance the degradation can be slowed significantly. Most importantly for a series battery pack is to keep the battery cells balanced, as unbalanced battery cells lead to lower capacity and faster cell degradation\cite{texas}. To facilitate balancing, each of the provided battery boards have mounted resistors which through a MOSFET can be connected to the battery cell. During operation one must decide when to switch said resistors on and off to keep the cells balanced. Usually, balancing is only done towards the end of a charge cycle. There are several reasons for this. Firstly, passive balancing requires energy to be expended and will therefore reduce the total amount of usable energy in a battery if done during discharging. Secondly, differences in impedance and charge curves between cells might make it look as though a cell is charged more than others, but the voltage difference might disappear naturally as the battery is charged more or as charge current is reduced towards the end of a charge cycle.

*Integration of the Energy Submodule*

It is not necessary to physically integrate the energy module with the full rover. However, if it were, the energy module could either be integrated as a charging station or be mounted directly on the rover itself. The advantage of a charging station is that the battery is only ever charged or discharged at a given time. This makes it easier to track current and power and leads to simpler charging and SOC algorithms. A drawback of using a charging station is a reduced range, as the rover always needs to get back to the charging station before its battery is depleted. Moreover, if the rover is detached it is difficult for the energy module to track the cell voltages and SOC during discharging. This could be fixed by using the microcontroller of another subsystem to track the battery while the rover is not connected to the charging station. However, a likely simpler solution is to mount the energy module on the rover. This would increase range, but does require the battery to charge and discharge at the same time. To be able to track the battery current, it would then be necessary to collect current and power data from other submodules. This data could be relayed through control and read in on the energy Arduino through UART.

The rover has four separate voltage regions. The battery and PV array have already been discussed. In addition there is a 5 V node used to power the FPGA and microcontrollers, and a variable voltage node used to power the motors. This power must originate in the battery and as such the 5 V and motor voltages must be connected to the battery through voltage converters. The obvious implementation is to use two switch mode power supplies in buck mode, one to provide 5 V and one to power the motors. One problem exists however with this solution: The 10.0 – 14.4 V of the battery is higher than the SMPS maximum input voltage\cite{powerLogbook}. A way to solve this would be to exchange the PMOS on the SMPS board for another NMOS, which would increase the maximum input voltage.

**Implementation**

*SMPS Configuration*

The voltage of the series battery pack is higher than the voltage of the parallel connected PV panels. The voltage of the PV array must therefore be stepped up before it can be used for charging. The SMPS is therefore operated in boost mode with the PV panels connected on port B and the battery on port A. Moreover, as the SMPS has a max output voltage of 7 V in synchronous boost mode\cite{powerLogbook}, non-synchronous mode was used. Finally, the on/off switch was set to emph{off} such that all power comes from the PV array and not from the USB connection powering the Arduino.

*Measurements*

Measurements of the circuit currents and voltages are used to track the operation of the subsystem. During charging, the most important measurements are the output voltage, the cell voltages, and the battery current.

The battery voltage is usually in the range 10.0 to 14.4 V. However, the Arduino can only measure voltages up to 4 V. To solve this issue the output voltage was passed through a voltage divider. The voltage divider consist of two parts. Firstly, a 560 $\Omega$ and a 330 $\Omega$ resistor create $V\_{pd} \approx 0.37 \* V\_{out}$. $V\_{pd}$ is then reduced even further using a 15 $k\Omega$ resistor and a 10 $k\Omega$ resistor to $\sim 0.22 \*V\_{out}$, which is then passed to the Arduino. With this voltage divider it should be possible to measure output voltages up to 18 V. The second stage uses far large resistors than the first so that the two do not impact one another. However, due to the large resistance values, the current drawn by the Arduino during sampling would cause a significant voltage drop. To prevent this an op-amp from the Circuits lab was used to buffer the voltage before it was connected to the Arduino.

The measurement port on the battery boards is used to measure cell voltages. To use this port the battery board relay is be switched by setting the \emph{*RLY*} input \emph{*HIGH*}. While the relay is switched the battery cell will be disconnected from its power terminal. For a series battery pack this means that the cells can only ever be measured or charged/discharged at a given time. To have the smallest impact on operation the measurements are therefore completed as quickly as possible and taken at a low frequency. To measure the voltage of a cell the first step is to set the output current to 0 mA to prevent voltage spikes while the battery is disconnected. The relay is then switched, the voltage is sampled and the relay is switched back. When abiding by the switching times of the relay and giving time for the current/voltage to stabilise this process takes about 40 ms. To minimise the number of Arduino ports needed for measurements, the cell voltages are not measured at the same time. Instead, one cell is measured every second. As there are 4 battery cells this means that each cell is measured with 4 second intervals. Due to the cell voltages changing very slowly the long update time was not found to be a problem.

The current in and out of the battery needs to be tracked to determine the SOC. However, the current sensor is on the wrong side of the SMPS to measure the current flowing into the batteries. Initially this problem was solved by connecting a 0.5 $\Omega$ resistor between the SMPS output and the battery. By measuring the voltage across this resistor the current into the battery could be determined. Initial tests of this solutions gave promising results. However, an excessively complex circuit was needed to ensure the sensor worked for both positive and negative currents. It was determined that a better solution was to simply calculate the output current from the input current using the formula \( I\_{out} = (1 - \delta)\*I\_{in} \). A problem with this solution is that it assumes CCM. At the standard charging current of 250 mA this assumption holds, but for smaller current, below about 150 mA, the SMPS might enter DCM. In this case the calculated current will be somewhat higher than its true value, which might impact SOC tracking. However, as will be discussed, the SOC algorithm has methods for correcting itself and this will hopefully mean that the error does not get too big. Unfortunately, it was not possible to test this due to the ban on working with batteries.

*Controlling the SMPS*

The dual loop controller from the sample charging code\cite{chargeCode} was used as the starting point for the controller design. However, some changes were necessary to accommodate other design choices. As previously discussed, the current needs to be set to 0 for only a couple of milliseconds when cell voltages are being measured. This is not possible using the 1 Hz Moore machine from the original controller. As such the slow loop was changed to a Mealy machine, where the outputs could change with the clock speed of the fast loop. This had the unfortunate side effect of creating timing issues as the resulting code was too complex for the Arduino to successfully run the fast loop at 1 kHz. For this reason the clock speed of the fast loop was reduced to 500 Hz. It should also be noted that the PI-gains from the sample code were changed as they did not work optimally in boost mode. The new gains were found through trial and error.

*Charging Algorithm*

The battery cells are designed to be charged first with a constant current and then with a constant voltage of 3.6 V\cite{batteryDatasheet}. The implemented charging algorithm follows this specification as much as possible. The state machine starts in state 0 where the output current is set to 0. It waits in this state until the OL/CL switch is set to 1, after which it moves to the constant current state. In the constant current state the current is nominally

250 mA. However, as will be discussed later, if the solar panels do not produce enough

power to charge at 250 mA, the MPPT algorithm might impose another lower value for

the constant current in this state. Moreover, technically the current is not constant as 40 ms of each second is used to measure cell voltages, and for a short period there will be zero current. These short charge breaks were not found to have an impact on the performance of the charging process. In fact, charging $LiFePO\_{4}$ battery cells with a pulsating current has even been found to severely reduce battery degradation\cite{PulseCharging}, which might mean better performance than truly constant current in the long term. The charging algorithm leaves the constant current state for the constant voltage state once any cell reaches a cell voltage of 3600 mV.

Initially the output voltage was kept constant in the constant voltage state. However, even if the voltage across the battery pack as a whole is constant this does not mean that the voltage across individual cells is constant. Testing revealed that with a constant output voltage, weaker cells could sometimes reach a cell voltage over 3700 mV as current died down. To prevent this, the constant voltage state was redesigned from keeping the voltage across the battery pack constant, to trying to keep the voltage of each individual cell constant. This was done in two steps. Firstly, the battery input current is regulated so as to keep the highest cell voltage in the range 3600 to 3630 mV. Secondly, balancing ensured that cells of lower voltage catch up to the highest voltage. The code used to control the constant voltage state is shown in Figure~\ref{fig:Constant\_Voltage}. Initial test of this code gave promising results, but it could not fully be tested due to the ban on working with batteries. However, Figure~\ref{fig:balancing} shows the ability of the code to stabilise all cell voltages

around 3600 mV. After all cell voltages have reached 3600 mV and the current has died

to below 30 mA, the charging algorithm is done and the state machine enters an idle state.

*SOH and Cell Balancing*

To monitor the battery’s SOH the charge capacity, energy capacity, cell voltage balance, and the number of completed charge/discharge cycles are all tracked. To retain the data between sessions, it was logged to the SD card every 60 seconds. The relevant data was then read in from the SD-card on start up.

To maintain SOH the cell voltages were balanced every charge cycle. As previously explained balancing should only occur at high SOC, which is why balancing is only done in the constant voltage state. The code use for balancing is shown in Figure~\ref{fig:balancingCode}. During CV charging it is desirable that all cells have a voltage of 3600 mV. If any cell has a voltage which is higher than 3600 mV, its voltage is too high and the dissipative resistors are switched on to lower the voltage. When used in conjunction with the constant voltage charging algorithm this code was extremely

successful at balancing the cells. The balancing is shown in action in Figure~\ref{fig:balancing}. As can be seen, the cell voltages are one by one brought up to and kept at 3600 mV. Even with an initial voltage difference of 200 mV, the balancing only takes about 1000 seconds = 17 minutes.

*MPPT*

As previously discussed, the PV array should be operated in the region where the current is lower than at the maximum power point. To ensure this, the following algorithm was used. First, the duty cycle was set to 0. This makes the input impedance of the SMPS very high, leading to a near 0 PV array current. The duty cycle is then increased until the output current/voltage reaches its desired value. If the desired operating point requires less power than what the PV array can provide, then an equilibrium should be found as the duty cycle is increased. If however the desired operating point requires more power than the PV panels can provide, then no equilibrium can exist and the duty cycle will increase to 1 without ever reaching the desired operating point. Thus, if the duty cycle ever reaches 1 during operation the charging algorithm must be trying to draw more power from the PV array than is available. Whenever the duty cycle reaches 1, the variable \emph{max\\_power}, is reduced by 100 mW and a new current setpoint is found through the formula $i\_{setpoint} = max\\_power / V\_{out}$. The duty cycle is then reset to 0 and the process repeats until an equilibrium is found. When at an equilibrium, the algorithm increases \emph{max\\_power} every 60 seconds to see if it is possible to operate at a higher power point than it currently is. If not, the algorithm will simply fall back to the same equilibrium. This algorithm was found to be very successful at tracking the maximum power point of the PV array. It is shown in action in the demo video.

*SOC*

Implementing the Coulomb counting was done by integrating the current flowing into the battery. The integral value gives how much current has flowed in or out of the battery. The SOC as a percentage can then be found by dividing the net charge into the battery by the charge capacity of the battery and then multiplying by 100. However, due to effects such as the Coulombic efficiency of the battery, the integral can start to stray when done over many cycles. As such, the SOC is reset to the maximum capacity every time the battery is fully charged. The maximum capacity will of course reduce over time as the battery degrades. This is why the maximum capacity is measured and updated with every discharge cycle. Moreover, as the Arduino is sometimes switched off before a charging/discharging cycle is complete, it is necessary to know the SOC on startup. As such, the current SOC is continually logged to the SD card during operation. When the Arduino is then switched on again, it simply fetches the SOC data from the SD card and continues at the SOC at which it left off.

The SOC should be used to estimate the range of the rover. The obvious way of estimating the range is to divide the remaining energy in battery by the power consumption of the rover and multiply by the speed. However, the SOC does not directly give how much energy is in the batteries at any given time. To find the energy in the battery, the total expended energy was logged alongside SOC data for a full discharge of the battery. From this data a 600 element lookup table was created which related the SOC of the battery to the remaining energy in the battery. The lookup table data is plotted in Figure~\ref{fig:lookup}.

*Discharging*

The energy module is also tasked with keeping track of the SOC and SOH during discharging.

This was done using the same algorithms as during charging. In addition the battery power was continually tracked as a function of time and SOC. When the battery is done discharging, this information is used to update the table relating the SOC of the battery to the remaining usable energy. This is done using the formula \( New~Energy~Array = 0.99\*(Old~Energy~Array) + 0.01\*(Measured~Energy~Array) \). By having the energy array update itself it is hoped that the energy array will be a good approximation even after many battery cycles as the battery performance starts to deteriorate.

*Communicating with other Modules*

Though it is not necessary to fully integrate the energy module with the rest of the rover, other submodules, specifically command, needs access data such as the battery SOH, SOC and remaining energy. For communicating with other modules the Arduino shield has a set of UART ports. However, as group members were not in the same location it was not possible to physically connect the energy module to the rover, which is necessary to use UART. As such, an alternative approach was employed. First the Arduino was connected to a computer via USB. On the computer a Python script was run. At the start, the Python script establishes a connection to a server created by running a similar script on the command module. After a connection has been established the Python script starts reading the serial data coming from the Arduino and transmits it using TCP to the command module. Each message coming from the Arduino is in CSV form where the first entry is the message ID, which allows the command script to decode what type of data is being sent.