Energy Management for IoT

Lab 3: Energy storage, generation and conversion

Alfredo Paolino: s295152@studenti.polito.it
Davide Fogliato: s303497@studenti.polito.it

February 22, 2023

Contents

1 Introduction

In this technical report, lab 3 of the course "Energy Management for IoT" will be described.
The goal of this last lab is to analyze a simple system focusing this time not on the data flow, but on the power perspective of the system.
More specifically, we used a simulated IoT device implemented in SystemC/SystemC-AMS. Requests are to populate the system, analyze and optimize its behavior.
The report is divided into 3 sections, each of them addressing problems of increasing complexity. Each section is in turn composed of a first part where we briefly present the approach chosen to face the problem and fulfill the requirements, and a second part where we present the obtained results by means of tables and graphs.
A brief theoretical introduction has been inserted prior to the core of the report, in order to have a common groundwork for all three sections, thus avoiding spreading information here and there between the *Workflow* and the *Results* parts.

2 Background

In this section, we condensed some theoretical aspects that could be useful to better understand the lab experience and the obtained results.

IoT device model

During this lab, we analyzed all three main aspects of energy management:

- Energy storage, computing the appropriate Energy Storage Device (ESD) size requested to guarantee sustained service
- Energy generation, computing how much power should be provided by the power sources
- Energy conversion, computing the amount of power loss due to conversion while distributing it to the various units

Every simulation has been launched on an IoT device model (shown in figure 1), made of:

- 4 sensors, modeled as current states
- Memory and Control Unit (MCU)
- A module to transmit data over ZigBee (RF Radio)
- A Li-ion battery with a DC-DC converter to ensure voltage compatibility
- A thin-film photovoltaic module operated at the Maximum Power Point (MPP) with a DC-DC converter to ensure voltage compatibility
- A DC bus to aggregate loads requests and determine who provides energy

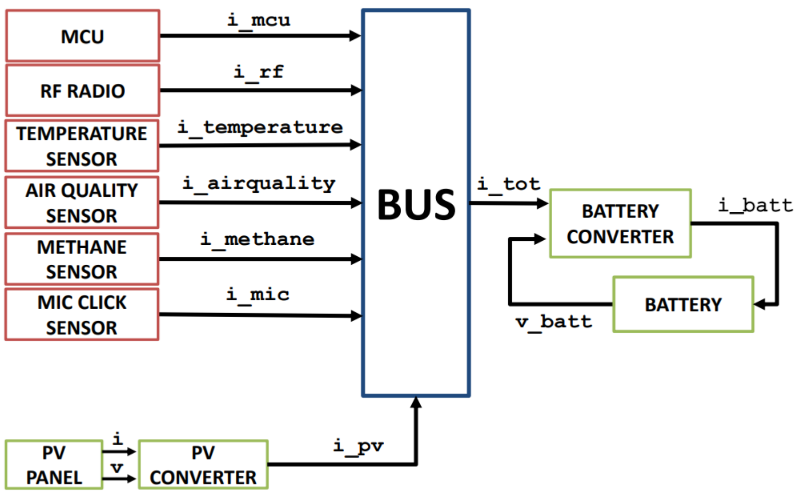


Figure 1: IoT device model used for simulations

Scavenging from radiation

Scavenging means extracting energy from the environment. A large variety of sources can be used to extract energy, and for this lab we used a photovoltaic module that exploits solar radiation. It is based on the photovoltaic effect, which results in electric voltage/current in a material upon exposure to light. Typically, the light is sunlight, and the electrode material is a p-n junction. For photovoltaic cells, the amount of current produced at a given voltage level strongly depends on solar irradiance (surface power density received from the Sun) and in turn on the weather and the day/night cycle.

Characterization of power sources

According to Thevenin's theorem, any combination of ideal voltage sources, current sources, and resistors with two terminals is electrically equivalent to a single voltage source in series with a single resistor.
Although approximate, this is a good model for power sources too.
The P-V curve of any power source can be obtained from the I-V curve, by keeping V on the X-axis and putting I*V=P on the Y-axis.
Due to the convexity of the P-V curve, there exists a point at which the extracted power is maximum, the MPP. Not extracting at the MPP strongly reduces the efficiency of the generation, and the mismatch increases for low irradiance values when using photovoltaic cells.

C-rate

Charge and discharge rates of a battery are governed by C-rates. The capacity of a battery is commonly rated at 1C, meaning that a fully charged battery rated at 1Ah should provide 1A for one hour. The same battery discharging at 0.5C should provide 500mA for two hours, and at 2C it delivers 2A for 30 minutes. The C-rate has then a double interpretation, of both time and current.
Losses at fast discharges reduce the discharge time and these losses also affect charge times. Capacity reduction at high discharge rates occurs because the transformation of the active chemicals cannot keep pace with the current drawn.

Voltage regulators and efficiency

Suppose to have in input an unregulated DC voltage, and want in output a regulated DC voltage, having a magnitude (and possibly polarity) that differs from the input. In this case, a DC-DC voltage regulator is needed to adapt the input voltage to the required output voltage.
The problem with voltage regulators is that the conversion process is not "perfect". We define the efficiency η as the ratio between output power and input power.

$$\eta = \frac{P_{out}}{P_{in}} = \frac{P_{in} - P_{losses}}{P_{in}} = 1 - \frac{P_{losses}}{P_{in}}$$

Battery model

In this lab, we used a circuit equivalent model for the battery. This approach aims at developing an equivalent circuit that models all the detailed characteristics through passive elements, voltage sources, and lookup tables.
The battery model used is represented in figure 2.
Actual values of all the elements are fitted from measured data or from curves extracted from the datasheet.

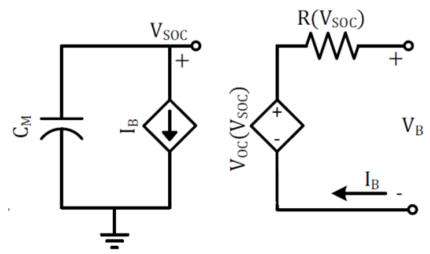


Figure 2: battery model used for simulations

3 Part 1: Model of the photovoltaic module

The goal of this first part is to fill two Look Up Table (LUT), present in the file *inc/pv_panel.h*, to build a model of the photovoltaic module. The LUTs store voltage and current at the MPP for a given irradiance value.

3.1 Workflow

Performing this task is pretty simple and requires only a little bit of patience.
The first thing to do is to import in Matlab the I-V curve present in the datasheet. Thanks to the Matlab digitizer tool, it is possible to extract some I-V points directly from the image and then put them in a text file.
With this done, we can proceed to extrapolate the MPPs. To do that, we build the corresponding power curves (I*V-V $\rightarrow$ P-V curve), find the maximum of the curves, and then extrapolate V and I at the MPP. These operations have been performed through a Matlab script.
Finally, we populate the LUTs with the obtained values.

3.2 Results

We report here the results obtained for this first part.
I-V matrices obtained with the Matlab digitizer tool:

$E_e = 250W/m^2$

V [V]	I [mA]
0.0031	17.6027
0.4456	17.6027
0.8882	17.2844
1.34	17.0144
1.8101	16.423
2.3366	15.0574
2.8993	13.1831
3.4611	9.5261
3.8666	5.8604
4.1793	0.1592

$E_e = 500W/m^2$

V [V]	I [mA]
0.0101	34.9569
0.6021	34.4654
1.2254	34.1654
1.7987	33.574
2.4178	31.5041
2.9452	28.2513
3.4611	22.0415
3.8166	14.7488
4.1032	6.9561
4.2981	0

$E_e = 750W/m^2$

V [V]	I [mA]
0	52.2036
0.5489	52.0079
1.1337	51.7121
1.7455	50.9207
2.3719	48.3594
2.9567	43.9238
3.3923	37.1225
3.7936	26.7685
4.1376	12.2831
4.344	0.0592

$E_e = 1000W/m^2$

V [V]	I [mA]
0	69.8503
0.5489	69.6503
1.2369	68.6674
1.9362	67.2846
2.4636	64.919
3.1401	57.5263
3.5758	47.4722
3.9427	32.6869
4.1949	16.7102
4.4013	0

P-V matrices are obtained by simply multiplying I*V:

$E_e = 250W/m^2$

P [mW]	I [mA]
0.0031	0.0537
0.4456	7.8445
0.8882	15.3526
1.34	22.7993
1.8101	29.7273
2.3366	35.1831
2.8993	38.2218
3.4611	32.9708
3.8666	22.6598
4.1793	0.6653

$E_e = 500W/m^2$

P [mW]	I [mA]
0.0101	0.3531
0.6021	20.7516
1.2254	41.8663
1.7987	60.3896
2.4178	76.1706
2.9452	83.2057
3.4611	76.2878
3.8166	56.2903
4.1032	28.5423
4.2981	0

$E_e = 750W/m^2$

P [mW]	I [mA]
0	0
0.5489	28.5471
1.1337	58.626
1.7455	88.8821
2.3719	114.7037
2.9567	129.8695
3.3923	125.9307
3.7936	101.549
4.1376	50.8226
4.344	0.2572

$E_e = 1000W/m^2$

P [mW]	I [mA]
0	0
0.5489	38.231
1.2369	84.9347
1.9362	130.2764
2.4636	159.9344
3.1401	180.6383
3.5758	169.7511
3.9427	128.8746
4.1949	70.0976
4.4013	0

I-V values at MPP:

$E_e = 250W/m^2$

V [V]	I [mA]
2.8993	13.1831

$E_e = 500W/m^2$

V [V]	I [mA]
2.9452	28.2513

$E_e = 750W/m^2$

V [V]	I [mA]
2.9567	43.9238

$E_e = 1000W/m^2$

V [V]	I [mA]
3.1401	57.5263

4 Part 2: Model of DC-DC converter and battery

This second part is focused on filling two LUTs, one for the photovoltaic module DC-DC converter, and the other one for the battery DC-DC converter. These LUTs store the corresponding efficiency given a certain input voltage or current. In addition, we reconstructed the values of R and V_{OC} as a function of V_{SOC} for the battery model.

4.1 Workflow

The initial part of this task follows the same steps as the previous one. We imported again in Matlab the curves from the datasheets and then, using the digitizer tool, we extracted the points from the images and set the corresponding parameters of the LUTs with the digitized samples. For the battery curve, we used the one with $V_{IN} = 2.4V$, sampling the points corresponding to known values of the logarithmic scale of the X-axis.
For what concerns the battery model parameters, we considered two different discharge curves (one with a discharge rate of 1C, and the other one of 0.5C), and, after using the digitizer to extract the samples, we interpolated the data.
We derived the voltage V_{OC} and the resistance R using the functions:

$$R = \frac{V_{0.5C} - V_{1C}}{I_{1C} - I_{0.5C}}$$

$$V_{OC} = V_{1C} + R * I_{1C}$$

Taking into account different values in the State Of Charge (SOC) range, we obtained two data sets for R and V_{OC} to populate the battery model in the simulator. These resulting samples were then used for the interpolation using the CurveFit app provided by Matlab, opting for third-degree polynomials to better represent the two functions.

4.2 Results

We report here the results obtained for the second part.
$\eta - V_{IN}$ and $\eta - I_B$ matrices are obtained with the Matlab digitizer tool:

η [%]	V_{IN} [V]
64.7607	0.6571
75.9544	0.8686
83.7838	1.2187
85.8583	1.8216
85.6575	1.9966
90.4756	2.5839
68.5266	2.8017
68.5266	3.1012
66.9206	3.3385
61.8348	3.7625
47.4475	4.9799

η [%]	V_{IN} [V]
44.1436	0.02
59.3160	0.04
71.3492	0.08
74.0959	0.10
81.4205	0.20
86.1291	0.50
88.3527	1.00
89.7914	2.00
90.1838	4.00
90.9686	10.0
90.8378	30.0
90.7070	70.0
91.0994	100
92.5382	200

V-SOC matrices respectively for curve 1C and for curve 0.5C:

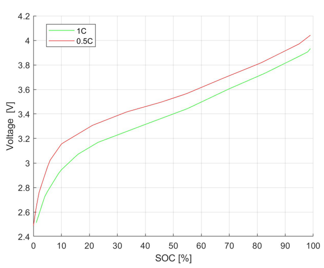
I_B = 3200 mA (1C)

V [V]	SOC [%]
2.4887	0.0000
2.7340	1.6453
3.0132	5.6673
3.1566	10.0548
3.3075	21.1152
3.4170	33.3638
3.4962	45.6124
3.5642	54.5704
3.6925	67.9159
3.8170	81.3528
3.9717	94.9726
4.0585	99.9086

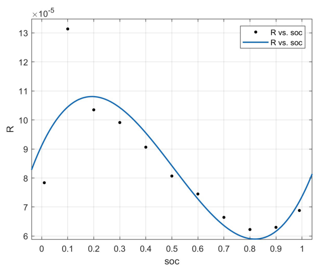
I_B = 1600 mA (0.5C)

V [V]	SOC [%]
2.4887	0.0000
2.7340	1.6453
3.0132	5.6673
3.1566	10.0548
3.3075	21.1152
3.4170	33.3638
3.4962	45.6124
3.5642	54.5704
3.6925	67.9159
3.8170	81.3528
3.9717	94.9726
4.0585	99.9086

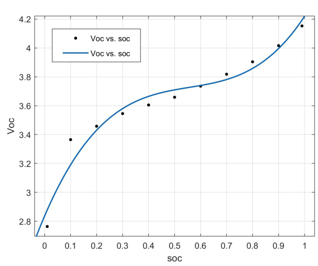
We report here the two curves obtained by the interpolation of the samples with different discharge currents and the graphs of R and V_{OC} obtained from the CurveFit tool.



(a) Curves 1C and 0.5C



(b) R [kΩ] vs. SOC



(c) V_{OC} [V] vs. SOC

The model derived by the interpolation performed by the CurveFit tool for both data sets is:

$$f(x) = p1 * x^3 + p2 * x^2 + p3 * x + p4$$

The coefficients for the R-SOC data set are:

$$p1 = 4.003e - 04 \quad p2 = -6.078e - 04 \quad p3 = 1.899e - 04 \quad p4 = 9.118e - 05$$

The coefficients for the V_{oc}-SOC data set are:

$$p1 = 4.189 \quad p2 = -7.010 \quad p3 = 4.204 \quad p4 = 2.836$$

5 Part 3: Load modeling and scheduling

This is the last but also most important part of this lab.
The goal is to launch some simulations of the entire system in order to extract important figures of merit like:

- The overall duration with meaningful quantities (loads power, photovoltaic power, battery power, and battery SOC for example)
- Efficiency of the converters
- Times the battery is used to provide energy

Each sensor is implemented as a simple Power State Machine (PSM) with an active state and a sleep state.

Load	P_{active} [mA]	P_{sleep} [mA]	T_{active} [s]
Air quality sensor	48.2	0.002	30
Methane sensor	18	0.002	30
Temperature sensor	3	0.002	6
Mic click sensor	0.15	0.002	12
ZigBee transmission	0.1	0.001	24
Memory and control	13	0.002	6

5.1 Workflow

The first simulation is performed with all sensors active in parallel (workload in figure 4).
The positive aspect of this distribution is a higher idle time, in which the power produced by the photovoltaic panel can be entirely used to recharge the battery. Of course, the downside is the higher peak power required by the sensors, which could deplete the battery faster.

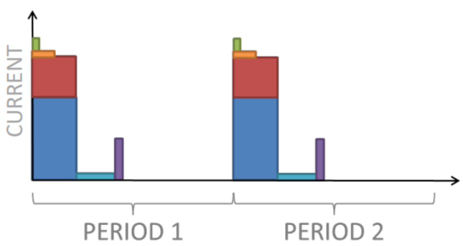


Figure 4: Parallel workload

Then we modified the simulation settings to switch from a parallel activation of the sensors to a sequential one (workload in figure 5).
With this setup, the workload is more balanced during the entire period. A lower peak power is required by the sensors, but for a longer time.

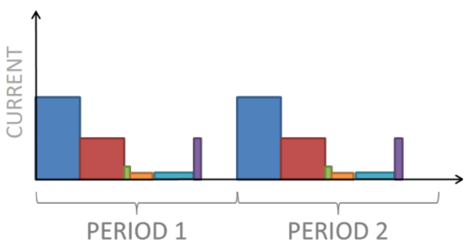


Figure 5: Sequential workload

We will see in section 5.2 if any difference will be present, and, in that case, which configuration will perform better.
For the last simulation, we modified the system by adding a second photovoltaic panel in parallel. The idea behind this solution is simple but effective: a higher power production means that the battery will be less stressed during low irradiance periods, and it will be charged more during high irradiance periods.

5.2 Results

Some meaningful values have been obtained through Matlab scripts and then plotted on the graphs reported below.

Parallel sensors activation

Graph 6 shows the periodic activation of sensors/MCU/radio; this trend is periodically repeated for the entire simulation. In the graph is indicated the required current at each instant of time. This current is either provided by the photovoltaic panel or by the battery. It is also possible for the two power sources to work together, in this case, the battery is still depleted, but at a lower rate.

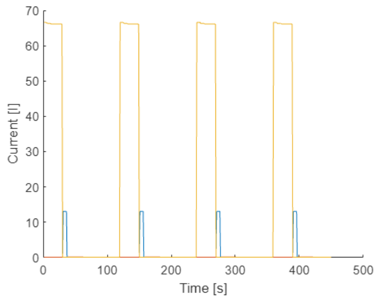
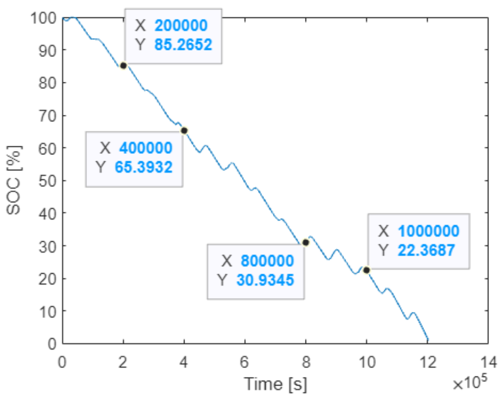
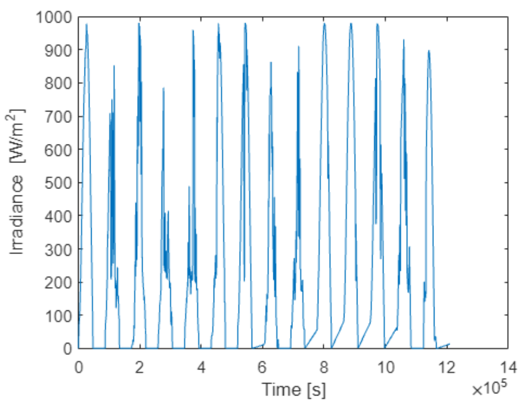


Figure 6: Periodic current required during the simulation with parallel sensors activation

The first important thing to highlight is how long the IoT system will be able to survive. In order to calculate this duration, the simulator has been designed so that the simulation does not stop as long as it is possible to provide the required power, without caring too much if it is generated by the photovoltaic module or the battery. At the moment both the photovoltaic module and the battery are no longer able to provide the required energy, the simulation stops.
The graph 7a shows that the simulation lasts for around 1200000s ($\approx$ 14 days), but what is interesting to notice is the dependence of the battery SOC with the irradiance of the photovoltaic panel.



(a) SOC over time with parallel activation



(b) Irradiance over time of the photovoltaic panel

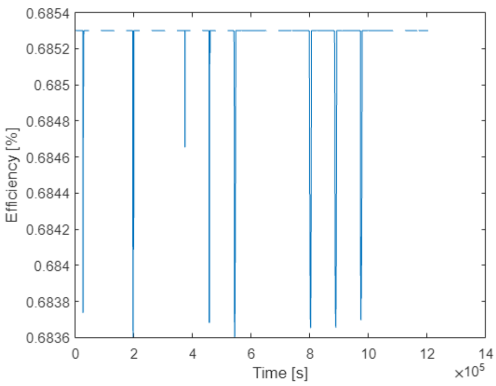
Two couples of points have been marked on the graph to focus our attention on. These two couples have in common the time difference (200000s). What it is important to notice is that, in the same period of time, the SOC has diminished of $\approx$ 20% during the period 200000s-400000s and of only $\approx$ 8% during the period 800000s-1000000s.
If the workload remains unchanged during the entire simulation, the only possible thing affecting the SOC is the photovoltaic panel and in turn its irradiance.

More in detail, we can calculate the so-called solar exposure, which is the integral of irradiance during the two periods, to see if it is compliant with our assumption.
The computation has been done approximately with the *trapz* Matlab command, but it is more than enough to highlight the difference.

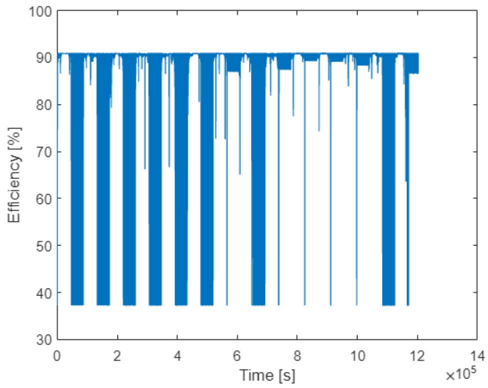
```
>> I24 = trapz(Gmonth(23:46,1), Gmonth(23:46,2))
I24 = 1.6618e+07 [J/m^2]

>> I810 = trapz(Gmonth(503:637,1), Gmonth(503:637,2))
I810 = 7.0170e+07 [J/m^2]
```

The result shows a much higher solar exposure in the second period, so the photovoltaic panel can produce more power, stress the battery less and recharge it more.
Another interesting metric for the system is the efficiency of the two converters, shown in figure 8a and 8b.



(a) Efficiency of photovoltaic panel converter



(b) Efficiency of battery converter

For the photovoltaic panel, whenever the irradiance is 0, the efficiency has not been reported because the panel does not produce any energy. For the battery instead, we see a much more variable efficiency, depending on the load and the power produced by the panel.
With a single panel, the battery often has to provide energy to the system, it is in fact used 46.48% of the entire simulation time, meaning that for nearly half of the time the system is powered (entirely or partially) by the battery.
To find this value is enough to check for how many simulation steps the *i_ batt* value is negative, and divide it by the total simulation steps.

```
>> 100*height(sim_trace(:,9))/height(sim_trace(:,9))
ans = 46.48
```

Sequential sensors activation

In this case, our goal was to check whether the sequential activation of sensors could bring an increase in simulation time, for example by stressing less the battery.
However, obtained results do not show any significant improvement. In fact, we obtain a simulation time just 1000s longer (0.0013% more).
In terms of SOC and efficiency, the results are basically identical to those already discussed for the parallel setting.
The only substantial difference is the total current plot, which of course has lower peaks but a longer duration in each period, as shown in figure 9.
This behavior was expected because the overall power required by the system is the same. Still, this solution could be useful in some cases, for example, if the energy harvester is able to provide a smaller but more constant power.

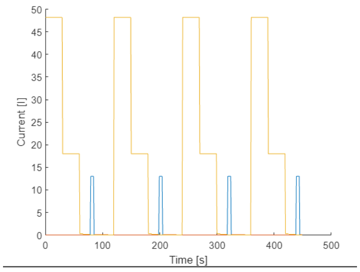


Figure 9: Periodic current required during the simulation with sequential sensors activation

5.3 Double photovoltaic panel

The last simulation was performed by adding a new photovoltaic panel in parallel to the original one. At a first glance, we would expect that doubling the power production also means doubling the simulation time. But actually, the results are even better. The discharge rate is not linear indeed. As shown in figure 10, performing smaller charge and discharge cycles lets the battery charge more, discharge slower, and thus last more.
What is even more astonishing is that a battery usage of only ≈10% less w.r.t. the 1-panel version, ends up lasting more than 4 times longer.

```
>> 100*height(sim_trace(:,9))/height(sim_trace(:,9))
ans = 54.08
```

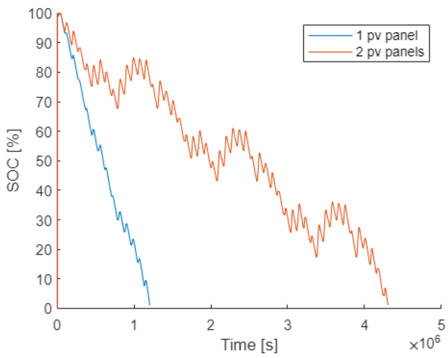


Figure 10: SOC comparison between the system with 1 panel and the one with 2 panels

Another possible solution that came to our minds was to use 2 panels, but devote one of them only to the battery recharge. This solution could be interesting but it has not been explored during this lab because it requires some modifications in the system.

6 Conclusion

To sum up, with this laboratory we learned how crucial it is to model and simulate a system to validate and estimate its behavior in terms of energy management beforehand.
We created models for different components to be able to launch simulations and trace different quantities.
The preliminary part required to populate some LUTs for the photovoltaic module by digitizing its I-V curves found in the datasheet. A similar approach was used to populate efficiency LUTs for the DC-DC converter and the battery too.
The core part was instead focused on performing some simulations of an IoT system to obtain meaningful quantities representing its behavior. For this part, we tried different approaches to see the pros and cons of each of them. Obtained results are satisfying and they have been reported and fully commented on.

Acronyms

MPP Maximum Power Point

LUT Look Up Table

SOC State Of Charge