Berliner Hochschule für Technik

Projekt-Labor: Automatisierungstechnik

Development and validation of a SoC algorithm for solarbatterychargecontroller Mppt-1210-hus by Libre.solar with lead acid
WS2122

handed in

Emile Schons 903606

EDV.Nr.:123 456

of the faculty VII – Elektrotechnik –
of Berlin University of Applied Sciences and Technology handed in Projektarbeit
to finalize the Projekt-Labor
Master of Engineering (M.Eng.)
im Studiengang
Energie und Automatisierungstechnik

Tag der Abgabe November 7, 2021

Gutachter
Gutachter1 Berlin University of Applied Sciences and Technology

Contents

List of Figures

Introduction

The scope of the project is to produce an algorithm for the solar charge controller (mppt-1210-hus) produced by libre.solar, which outputs the state of charge (SoC) of the lead acid battery attached to it as seen in schematic 3.1.
A key principle followed throughout the project is to use, wherever possible, Free Open-source Hardware (FOSH) and Free Open-source Software (FOSS). Consequently, the whole project has been released as source code [Schons 2021] with a license adhering to these principles and is available here. As the SoC can only be measured indirectly, it has to be deduced from available measurements. The charge controller exposes battery current and voltage measurements once a second for this purpose.
There are three main approaches for estimating the SoC: coulomb-counting w/ Open circuit voltage (OCV), model-based, data-driven [Espedal et al. 2021]. Best results seem to be achieved by combining the approaches to profit from their different strengths.

Current integration method - Coulomb Counting /w OCV

Equation 1.1 shows a simple way to define the SoC:

$$z(t) = z(0) - \frac{1}{Q_C} \int_0^{\Delta t} i(t)\, dt \tag{1.1}$$

where:

z	= the SoC
i	= battery current
Q_C	= battery capacity
Δt	= the time interval between two current measurements

The initial SoC $z(0)$ is determined by consulting a OCV-SoC lookup table. Starting from there the charge in relation to total capacity entering and leaving the battery is subtracted from SoC. This method has several disadvantages:

- The initial SoC incorporates already an error, which won't be corrected till recalibration
- Error in counting charges and measurement errors accumulate quickly
- The algorithms only recalibrate reliably on full charge, which especially in solar systems does not occur regularly, producing incorrect SoC until recalibration.

The coulomb counting method is most often combined with ongoing voltage measurements, while the battery is at rest (OCV) and a resulting OCV-SoC-lookup as shown in figure 1.1. This mitigates the recalibration problem in case the battery gets some rests, but this enhancement still does not

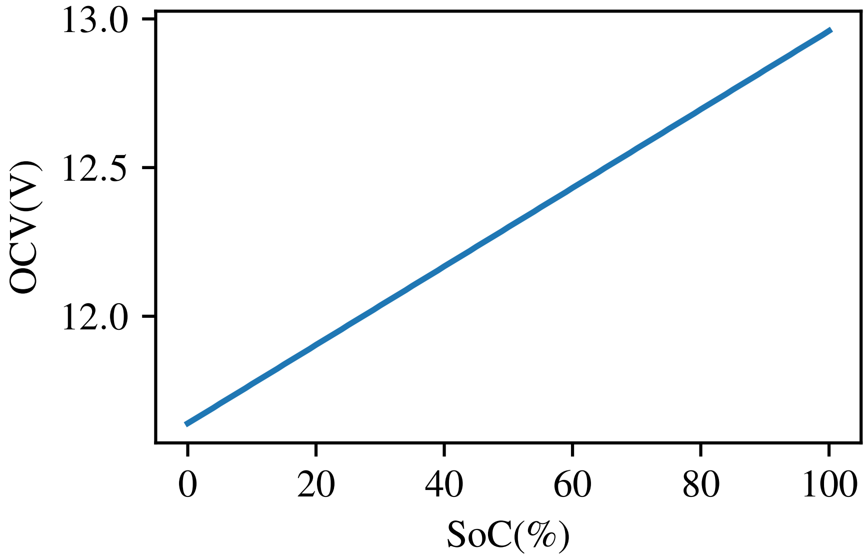


Figure 1.1: OCV-SoC lead acid battery

take hysteresis into account, which make OCV-SoC-lookup even with voltages at rest unreliable [Tjandra et al. 2014] [Khan und Choi 2016].

The existing SoC function $Charger :: update_soc$A embebbed in mppt-1210-hus uses OCV-SoC-lookup function based on linear interpolation of ocv_empty and ocv_full.

Equivalent circuit based models

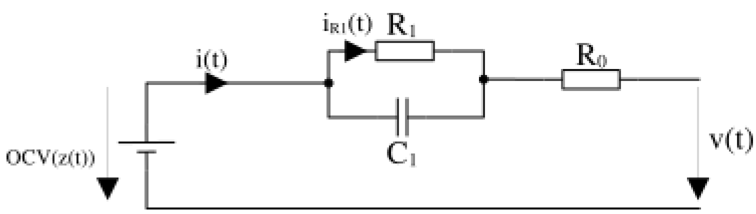


Figure 1.2: Equivalent circuit model

The figure 1.2 shows an empirical model known as Thévenin model or equivalent circuit model (ECM), adhering to the equation:

$$v(t) = OCV(z(t)) - R_0 i(t) - R_1 i_{R_1}(t) \tag{1.2}$$

where:

v = voltage measured on battery terminals under load
z = SoC
v_1 = $R_1 i_{R_1}(t)$
OCV = a function based on a SoC-OCV-lookup table
R_1, C_1, R_0 = don't exist as physical components inside the battery, their values are chosen to fit battery test data for example that of a hybrid pulse power characterization (HPPC) battery test

Other empirical models are the auto regressive exogenous (ARX) model and dual polarisation (DP) model.

Physics based models

As the professor Dr. Gregory L. Plett puts it in his lectures notes of course ECE5720 [1] : "Battery management and controls using physics-based models is a major focus of our present research efforts and (I hope) of a future course" in 2015. The first papers on the topic are available now [Miguel et al. 2021], but physics based models (PBM) still are hard to implement and only little material or software code is available. Thus, the ECM model is preferred for now. Moreover, the PBM based models have a higher computational complexity than empirical models, making it non suitable for the mppt-1210-hus micro-controller unit (MCU) being a 32bit ARM STM32L072.

Data-driven

These approaches are based on artificial
intelligence (AI) and need good training data and the right training parameters. Combined with PBM it is a very promising research field [Miguel et al. 2021]. First successes have been presented, which reduce the computational complexity so online estimation of SoC on MCU becomes possible.

[1] http://mocha-java.uccs.edu/ECE5720/ECE5720-Notes02.pdf

Implementation

It is popular to combine any of the approaches introduced in 1 with a kalman filter (KF) in order to reduce the influence of white (uncorrelated) noise in measurements and to combine voltage and current measurements known as sensor fusion. The second principle makes the algorithm resilient to inaccurately chosen ECM parameters. In this project the combination of coulomb counting 1 and ECM 1 has been chosen.

As the ECM's equation 1.2 is non-linear an extended kalman filter (EKF) needs to be used. The EKF linearizes the non-linear system by calculating the Jacobian matrix, which is the Taylor series expansions of first order.

2.1 EFK implementation

A reasonable step by step guide to implement the EKF is [Rzepka et al. 2021]. The algorithm refined in this project is based on a fork of the algorithm named $kalman - soc$[1] designed by Matthew Johnston for the company okra-solar.
The refinements consists of improvements or adaptions to libre.solar use case and code base.

Changelog[2]:

- adoption of libre solar code style guide
- adoption of TinyEKF notation [3]
- change from energy counting to coulomb counting
- use of float instead of integer in order to guarantee correct calculations, with no overflow.

2.1.1 Definition of EKF equations

The EKF consist of two equations describing the system the SoC is to be predicted for:

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, i_k) + \boldsymbol{w}_k \; (state\ space\ equation) \tag{2.1}$$

$$y_k = h(\boldsymbol{x}_k, i_k) + \boldsymbol{v}_k \quad (control\ output\ equation) \tag{2.2}$$

where:

[1] `https://github.com/mulles/kalman-soc`

[2] note exact changes are traceable with the commits to the fork.

[3] `https://github.com/simondlevy/TinyEKF/`

$f(\boldsymbol{x}_k, i_k,) = x_k[0] - \frac{\Delta t}{Q_C} i_k$ (discrete coulomb counting equation 1.1)

$h(\boldsymbol{x}, i_k) = OCV(SoC) + i_k R_0 + v_1$ (discrete ECM equation 1.2)

$\boldsymbol{x}_k = [SoC, R_0, v_1]$

y_k = battery terminal voltage, v_k is common
i_k = battery current, u_k is common
w_k = noise
v_k = measurement noise

This equations need to be linearized in order to apply the EKF to them.
This leaves us with the matrices defining the system :

$$\boldsymbol{x}_k = \begin{bmatrix} SoC \\ R_0 \\ v_1 \end{bmatrix} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & e^{\frac{\Delta t}{R_1 C_1}} \end{bmatrix}}_{\boldsymbol{F}_k} \underbrace{\begin{bmatrix} SoC \\ R_0 \\ v_1 \end{bmatrix}}_{\boldsymbol{x}_k} + \underbrace{\begin{bmatrix} \frac{\Delta t}{Q_C} \\ 0 \\ R_1(1 - e^{\frac{\Delta t}{R_1 C_1}}) \end{bmatrix}}_{\boldsymbol{B}_k} i_k \tag{2.3}$$

$$\boldsymbol{H} = \frac{h(\boldsymbol{x}_k, i_k,)}{d\boldsymbol{x}} = \frac{OCV(SoC) + i_k R_0 + v_1}{d\boldsymbol{x}_k} = \begin{bmatrix} \frac{dOCV}{dSoC} & i_k & 1 \end{bmatrix} \tag{2.4}$$

EFK Steps

It is common to separate the EKF algorithm into two steps. In the prediction step the next SoC is predicted using coulomb counting method (current measurement) with function f and the matrix $\boldsymbol{P}$ is calculated. In the update step the measurement is updated based on the predicted SoC. Then the gain matrix G is calculated, which is used to adjust the SoC and other states depended on if the voltage measurement is trusted more than the current measurement and vice-versa. Wikipedia offers a comprehensive explanations of the different equations

Predict

$\hat{\boldsymbol{x}}_k = f(\hat{\boldsymbol{x}}_{k-1}, i_k)$

$\boldsymbol{P}_k = \boldsymbol{F}_{k-1} \boldsymbol{P}_{k-1} \boldsymbol{F}_{k-1}^T + \boldsymbol{Q}_{k-1}$

Update

$\hat{\boldsymbol{y}}_{\boldsymbol{k}} = h(\hat{\boldsymbol{x}}, i_k)$
$\boldsymbol{G}_k = \boldsymbol{P}_k \boldsymbol{H}_k^T (\boldsymbol{H}_k \boldsymbol{P}_k \boldsymbol{H}_k^T + R)^{-1}$
$\hat{\boldsymbol{x}}_{\boldsymbol{k}} = \hat{\boldsymbol{x}}_{\boldsymbol{k}} + \boldsymbol{G}_k (v_k - \hat{y}_k)$
$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{G}_k \boldsymbol{H}_k) \boldsymbol{P}_k$

2.1.2 Parametrization of the EKF

The parametrization is particularly challenging. It is typically done offline [4] and requires quite advanced equipment.
No effort has been undertaken to find good parameters, it has been initialized with the idea that the EKF will adjust to bad chosen parameters. The parameters used listed in 2.5 and 2.6 are found in literature.

$$\boldsymbol{F} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{P} = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \boldsymbol{Q} = \begin{bmatrix} 0.0001 & 0 & 0 \\ 0 & 0.0001 & 0 \\ 0 & 0 & 0.0001 \end{bmatrix} \tag{2.5}$$

$$\boldsymbol{x}_k = \begin{bmatrix} SoC \\ R_0 \\ v_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad R = 0.1 \tag{2.6}$$

[4] not generated by the micro controller unit (MCU) the algorithm is deployed to, thus not on the system running the final algorithm. Offline means on more advanced hardware, which is not available in the final product. See `https://en.wikipedia.org/wiki/Online_model`

Experimental Validation

The developed kalman-soc algorithm 2 has been feed with two different datasets and compared to the current libre.solar SoC algorithm A. The first dataset has been generated with a charge controller named mppt-hus-1210 by a laboratory measurement setup as in figure 3.1. The second dataset has been generated by libre.solar by the same charge controller named EVLKpN being deployed in the field in Africa.

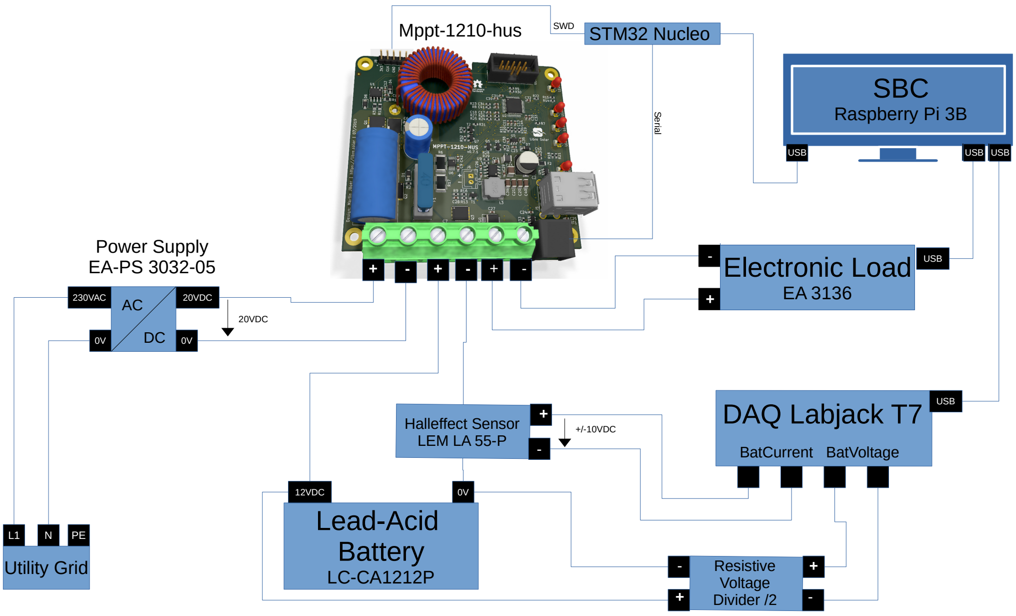


Figure 3.1: Setup to validate the SoC algorithm

3.1 Dataset from mppt-hus-1210 laboratory measurement setup

As visualized in figure 3.2 the dataset consists of charging the battery completely, discharging it for 20min with 1A, fully charging again, discharging 10min with 1A and 40min with 3A, finally fully charging again. The length of the dataset is 220min. The same graph can be seen more interactively here

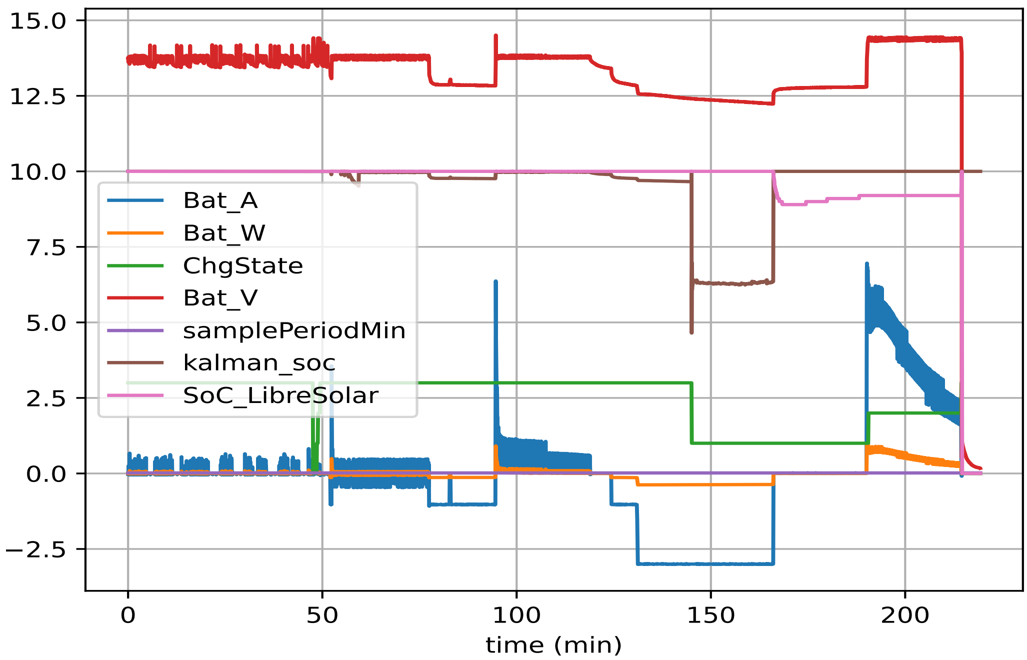


Figure 3.2: All measurements with laboratory device mppt-1210-hus

In figure 3.3 one can see that both algorithm have huge deviations, which could be explained by a probably wrongly set capacity in the charge controller. So comparison does not much sense. One can see that the kalman-soc algorithm follows well what one would expect by analyzing the charge state as well as the current and voltage.

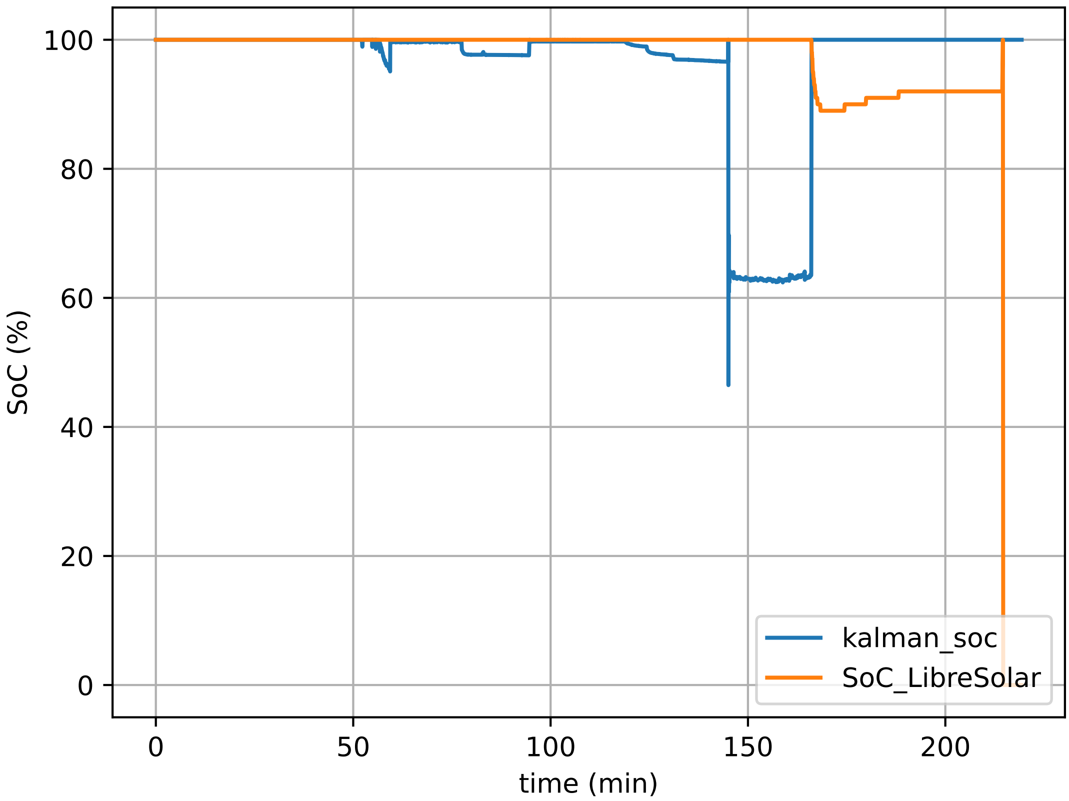


Figure 3.3: SoC libre.solar algorithm vs kalman-soc on device mppt-1210-hus

3.2 Dataset from EVLKpN of field test

The dataset introduced in figure 3.4 was produced by real world usage during 13 days in an african household. The same graph can be seen more interactively here.

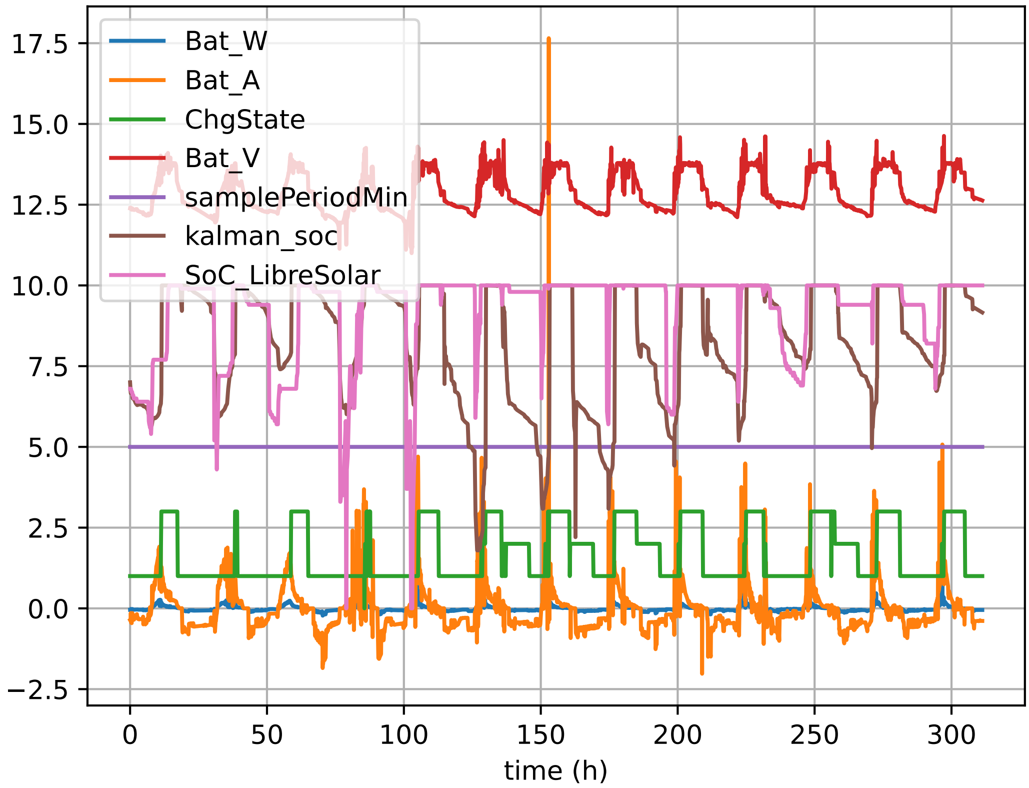


Figure 3.4: All measurements of device EVLKpN in the field from 2021/08/17 to 2021/08/30

In figure 3.5 it is sure that the capacity is set correctly. When comparing both the kalman-soc performs much better, this is especially clear when one looks at the battery voltage, which is followed very well by the algorithm compared to the libre.solar algorithm. The libre.solar algorithm performs so poorly as it only updates SoC on current <0.2A corresponding roughly to OCV.

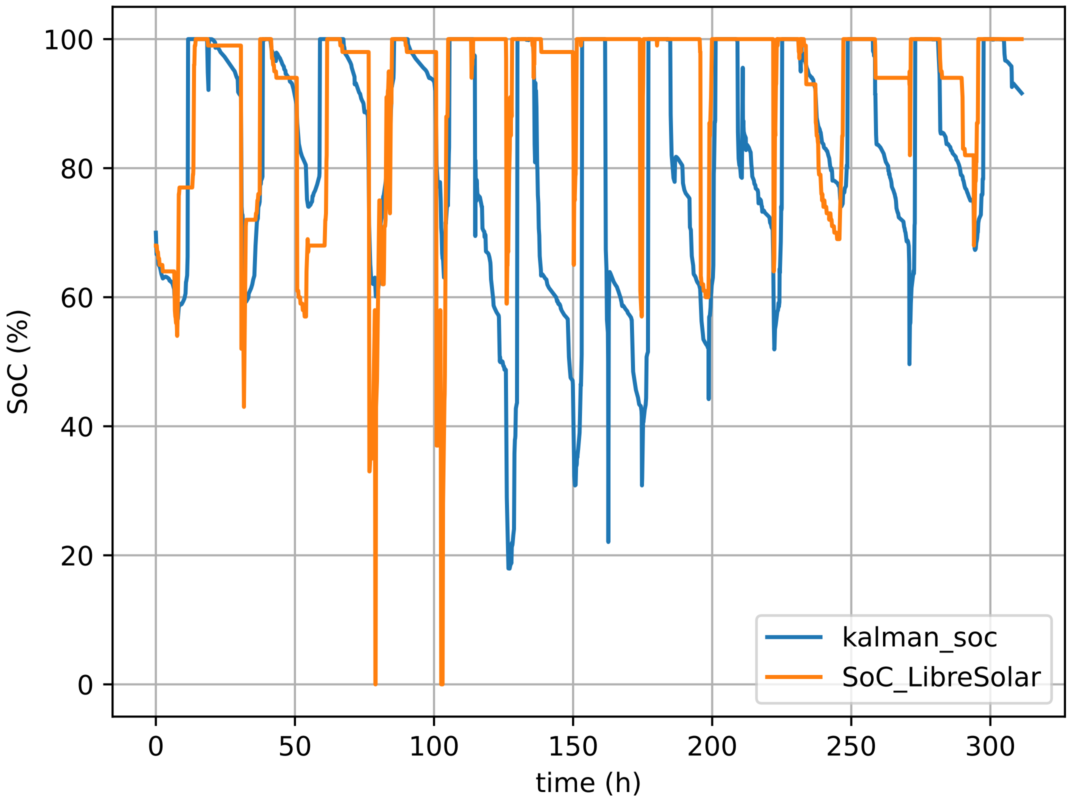


Figure 3.5: SoC libre.solar algorithm vs kalman-soc on device EVLKpN in the field from 2021/08/17 to 2021/08/30

Conclusion

The key to success for a good SoC is a good definition of both models and a good parametrization. In case no FPU is available on the micro-controller unit (MCU) fix point arithmetic might be an option. The Adaptive extend Kalman Filter (AEKF) is an option, where the parameters are determined online, but still AEKF needs to be initialized with parameters not necessary fitting all batteries of a given chemistry, because of different capacities or tolerances and specific designs. TODO It needs to be researched if it only affects the convergence time or if convergence is not reached at all? A further advantage of an adaptive filter is that the state of health (SOH) can be determined along the side. When simple observation and control-input models are used as in the okra kalman-soc algorithm, the parametrization might be much easier? TODO to check if the initial parameters, in this case OCV in function of SoC data can be used generally or only on the batteries, the data has been generated with.

Citations concerning the on/off-line parametrization

"A sigma-point Kalman filter is further used to manage inaccuracies generated by the reduction process and experimental-related issues such as measurement error (noise) in the current and voltage sensors."

Offline parametrization:
'As a result, many experimental pretests investigating the effects of the internal and external conditions of a battery on its parameters are required, since the accuracy of state estimation depends on the quality of the information regarding battery parameter changes'

'Therefore, some tests data must be available in advance to find the parameters of these models.' [Hussein und Batarseh 2011]

Gregory Plett says:
'Two possible approaches:

First, an algorithm might somehow adapt the parameter values of the model during operation to match presently observed current–voltage behaviors; but, this **must be done very carefully to avoid making the model unstable or physically nonmeaningful**.

Alternately, a set of models could be pre-computed at different feasible aging points and the model from this set that most closely predicts presently observed current–voltage dynamics could be selected from the set. This second approach guarantees stable and physically meaningful models since all models in the pre-computed set meet these criteria. We propose such an approach here.' —https://www.sciencedirect.com/science/article/abs/pii/S2352152X18301385?viaIn order to have online first algo, the parametrization needs to be done online!.

4.0.1 Outlook

Promising articles and/or methods for online parameter estimation:

online parameter estimation from the ARX model tran2017state
[Wang et al. 2021] xia2018online

Appendix

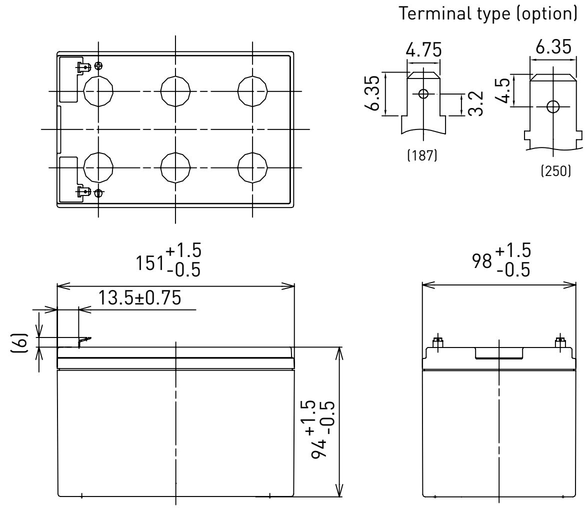
```
void Charger::update_soc(BatConf *bat_conf)
{
    static int soc_filtered = 0;       // SOC / 100 for better filtering

    if (fabs(port->current) < 0.2) {
        int soc_new = (int)((port->bus->voltage - bat_conf->ocv_empty) /
                   (bat_conf->ocv_full - bat_conf->ocv_empty) * 10000.0);

        if (soc_new > 500 && soc_filtered == 0) {
            // bypass filter during initialization
            soc_filtered = soc_new;
        }
        else {
            // filtering to adjust SOC very slowly
            soc_filtered += (soc_new - soc_filtered) / 100;
        }

        if (soc_filtered > 10000) {
            soc_filtered = 10000;
        }
        else if (soc_filtered < 0) {
            soc_filtered = 0;
        }
        soc = soc_filtered / 100;
    }

    discharged_Ah += -port->current / 3600.0F; // current>0:change sign
}
```

LC-CA1212P

**FOR MAIN POWER SUPPLIES.
CYCLE LONG-LIFE TYPE**

DIMENSIONS (MM)

Contents indicated (including the recycle marking, etc.) are subject to change without notice.

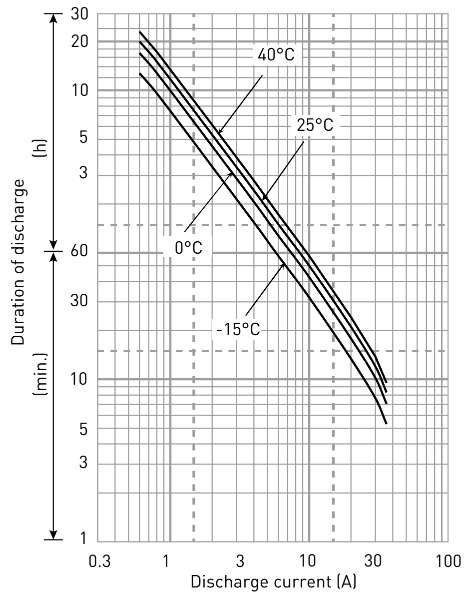


Battery case resin: standard (UL94 HB)

SPECIFICATIONS

Name		LC-CA1212P/P1
Nominal voltage		12V
Nominal capacity (20 hour rate)		12Ah
Dimensions	Length	151mm
	Width	98mm
	Height	100mm
Approx. mass		3.80kg
Terminal		Faston 187/250
Capacity (25°C)	20 hour rate	12.0Ah
	10 hour rate	11.0Ah
	3 hour rate	9.3Ah
	1 hour rate	8.5Ah
Impedance	Fully charged battery (25°C)	15mΩ
Temperature dependency of capacity (20 hour rate)	40°C	102%
	25°C	100%
	0°C	85%
	-15°C	65%
Self-discharge (25°C)	After 3 month	91%
	After 6 month	82%
	After 12 month	64%

DURATION OF DISCHARGE VS. DISCHARGE CURRENT



WATT TABLE (25°C)

(Wattage/battery)

Cut-off	3min.	5min.	10min.	15min.	20min.	30min.	45min.	1h	1.5h	2h	3h	4h	5h	6h	10h	20h
9.6V	679	559	384	298	247	183	137	105	70.3	54.5	38.1	28.8	24.1	21.7	13.3	7.22
9.9V	649	537	373	288	241	177	135	104	69.9	54.2	37.8	28.8	24.1	21.7	13.3	7.22
10.2V	607	506	363	282	235	177	134	102	69.1	53.9	37.5	28.8	24.0	21.6	13.2	7.21
10.5V	556	475	343	271	231	172	133	100	68.5	53.3	36.9	28.7	24.0	21.6	13.2	7.20
10.8V	495	434	321	261	225	166	123	98	66.1	52.1	36.3	28.4	23.8	21.5	13.1	7.18

AMPERE TABLE (25°C)

(Ampere/battery)

Cut-off	3min.	5min.	10min.	15min.	20min.	30min.	45min.	1h	1.5h	2h	3h	4h	5h	6h	10h	20h
9.6V	61.1	50.1	34.3	25.9	21.3	15.6	11.7	8.90	5.95	4.60	3.20	2.41	2.01	1.81	1.11	0.602
9.9V	58.4	48.2	33.3	25.0	20.8	15.1	11.5	8.80	5.92	4.58	3.18	2.41	2.01	1.81	1.11	0.602
10.2V	54.6	45.4	32.4	24.5	20.3	15.1	11.4	8.70	5.85	4.55	3.15	2.41	2.00	1.80	1.10	0.601
10.5V	50.0	42.6	30.6	23.6	19.9	14.7	11.3	8.50	5.80	4.50	3.10	2.40	2.00	1.80	1.10	0.600
10.8V	44.5	38.9	28.7	22.7	19.4	14.2	10.5	8.30	5.60	4.40	3.05	2.38	1.99	1.79	1.09	0.598

All mentioned values are average values

LC-CA1212P

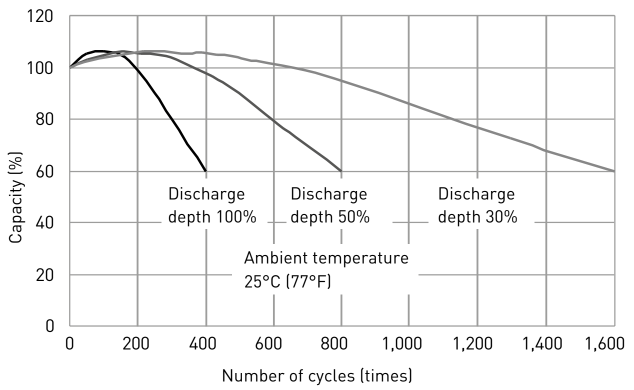
CHARGING METHOD (25°C)

Cycle use	Control voltage 14.5V - 14.9V Initial current 4.8A or smaller

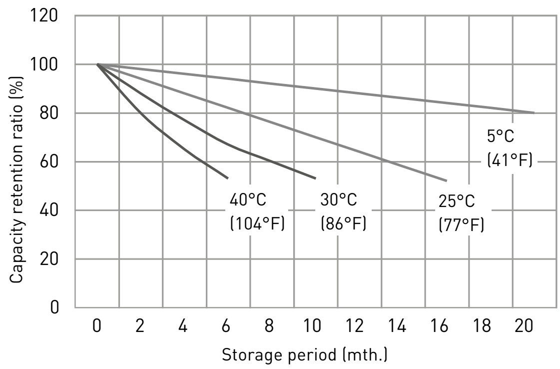
CUT-OFF VOLTAGE

Discharge current	0.6A - 2.4A	2.4A - 6A	6A - 12A	12A - 24A	24A - 36A
Cut-off voltage	10.5V	10.2V	9.9V	9.3V	8.7V

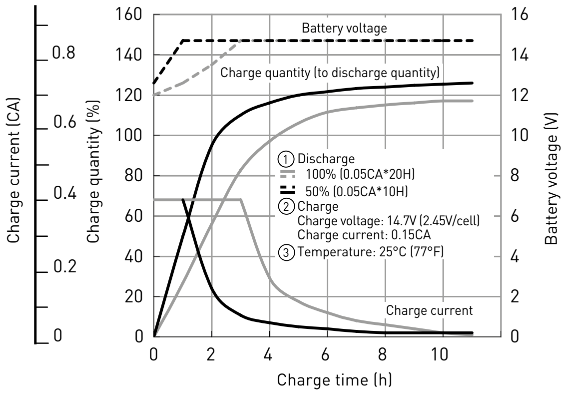
CYCLE LIFE VS. DEPTH OF DISCHARGE (ACC. IEC 61056)



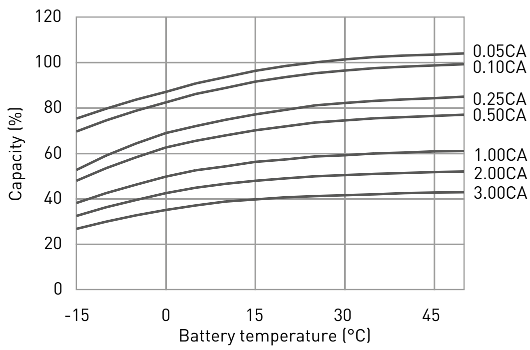
RESIDUAL CAPACITY TEST RESULT



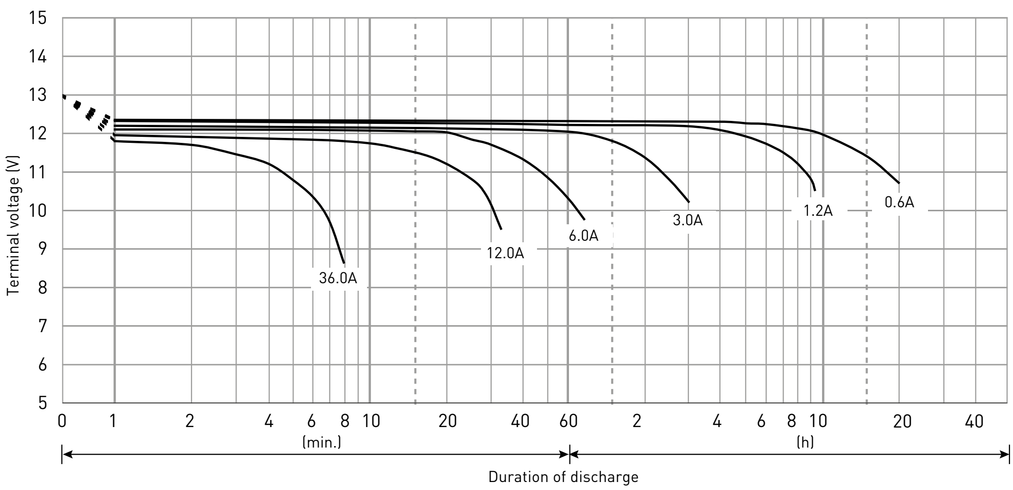
CONSTANT-VOLTAGE CONSTANT-CURRENT CHARGE CHARACTERISTICS FOR CYCLE USE



DISCHARGE CAPACITY BY TEMPERATURE AND BY DISCHARGE CURRENT



DISCHARGE CHARACTERISTICS



The data in this document are for descriptive purposes only and are not intended to make or imply any guarantee or warranty. Regarding handling and safety please consult our VRLA technical handbook chapter 1.

Bibliography

[Espedal et al. 2021] Espedal, I. B., Jinasena, A., Burheim, O. S., und Lamb, J. J., Current trends for state-of-charge (soc) estimation in lithium-ion battery electric vehicles. *Energies*, 14(11):3284.

[Hussein und Batarseh 2011] Hussein, A. A.-H. und Batarseh, I., An overview of generic battery models. In *2011 IEEE Power and Energy Society General Meeting*, Seiten 1–6. IEEE.

[Khan und Choi 2016] Khan, A. B. und Choi, W., Hysteresis modeling of the sealed flooded lead acid battery for soc estimation. In *Proceedings of the KIPE Conference*, Seiten 309–310. The Korean Institute of Power Electronics.

[Miguel et al. 2021] Miguel, E., Plett, G. L., Trimboli, M. S., Lopetegi, I., Oca, L., Iraola, U., und Bekaert, E., Electrochemical model and sigma point kalman filter based online oriented battery model. *IEEE Access*, 9:98072–98090.

[Rzepka et al. 2021] Rzepka, B., Bischof, S., und Blank, T., Implementing an extended kalman filter for soc estimation of a li-ion battery with hysteresis: A step-by-step guide. *Energies*, 14(13):3733.

[Schons 2021] Schons, E., Development and validation of a SoC algorithm for solarbatterychargecontroller Mppt-1210-hus by Libre.solar with lead acid.

[Tjandra et al. 2014] Tjandra, R., Thanagasundram, S., Tseng, K. J., und Jossen, A., Improved lithium-ion battery model with hysteresis effect. In *2014 IEEE Transportation Electrification Conference and Expo (ITEC)*, Seiten 1–8.

[Wang et al. 2021] Wang, Y., Huang, F., Pan, B., Li, Y., und Liu, B., Augmented system model-based online collaborative determination of lead–acid battery states for energy management of vehicles. *Measurement and Control*, 54(1-2):88–101.