



Projekt-Labor: Automatisierungssysteme

SS2021

vorgelegt von

Emile Schons 903606

EDV.Nr.:123 456

dem Fachbereich VII – Elektrotechnik –
der Beuth Hochschule für Technik Berlin vorgelegte Projektarbeit
zum Abschliessen des Projekt-Labors
Master of Engineering (M.Eng.)
im Studiengang
Elektronik und Kommunikationssysteme

Tag der Abgabe 23. August 2021

Gutachter

Gutachter1 Beuth Hochschule für Technik

Inhaltsverzeichnis

1	Introduction	3
1.1	why Kalman, why EKF????	3
2	Implementation	5
2.1	Hardware Setup for validation	5
2.2	EKF Implementation	5
2.2.1	Definition Mathematics Equations	5
2.3	Kalman-SOC	7
3	Validation	9
A	Angehängtes: Die Dateien des Pakets	11
	Literatur- und Quellenverzeichnis	13

Abbildungsverzeichnis

Kapitel 1

Introduction

(4d) (mostly offline)

1.1 why Kalman, why EKF????

Other methods based on auto regressive exogenous (ARX) model

Kapitel 2

Implementation

2.1 Hardware Setup for validation

2.2 EKF Implementation

date: (4d)

To design an extended kalman filter algorithm three steps are necessary: 1. The control-input model and the observation model need to be well designed. 2. The constants inside the models expressed as matrixes need to be defined, which is as well known as parametrisation of the model. 3. The model needs to be implemented as code running on embedded systems.

The key to success for a good SOC is a good definition of both models and a good parametrisation. The implementation is independent of the use case and depends on the skills of the programmer, in his hands lies the optimization of the code. In case no FPU is available on the micro-controller unit (MCU) fix point arithmetic might be an option.

Particular challenging is the parametrization, which is typically done offline* and required quite advanced equipment. Adaptive extend Kalman Filter 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 (different capacities, different manufacturer). 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 algo, the parametrisation might be much easier? TODO to check if the only parameters OCV(SOC) data can be used generally or only on the batteries they have been measured on.

As the OCV curve is non-linear the kalman filter can not be used.

state transition and observation models

* TODO implement foot note -> not done on the MCU the algo is deployed to, not on the system running the final algo.

2.2.1 Definition Mathematics Equations

TODO Make a diagram where you can see the input output and steps

Most general form of state observer equations:

$$x_{k+1} = Ax_k + Bu_k \text{ (state observer equation)}$$

$$y_k = Cx_k + dDu_k \text{ (output equation)}$$

$u_k \equiv i_k$ a control vector (input), defined as the measurement of current through the battery at time k .

$y_k \equiv v_k$ an observation (or measurement), defined as a voltage measurement of the battery at time

k .

$A = I$ identity matrix f.i $I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$B = \begin{bmatrix} -\frac{\Delta t}{Q_C} & 0 \\ 0 & 1 \end{bmatrix}$ is the **control-input model**, deduced from $f(x_k, i_k, \Delta t) = x_k - \frac{1}{Q_C} \int_0^{\Delta t} i_k dk$

$C = [0, -R_1, -R_2, \dots, M] = H(x_k)$ and $H(x_k)$ is the **observation model**, which maps the state space into the observed space and TODO understand the relationship to $h(x_k)$

$D = R_0$

Most general form of Kalman Filter equations:

$x_{k+1} = f(x_k, u_k) + w_k$ (State Space equation)

$z_k = h(x_k) + v_k$ (Output equation)

$\hat{x}_{k+1} = (A - BK) \hat{x}_k + L(y_k - \hat{y}_k)$ (here u_k seems missing)

Predicted variables \hat{y}_k and \hat{x}_k are commonly denoted by a "hat" to distinguish them from y_k and $x(k)$ of the physical system. As the state of charge (SOC) denoted as $\hat{x}_k = SOC_k$ cannot be measured directly it is always a predicted variable, opposed to $y(k)$, which is the measured circuit voltage. Consequently $\hat{y}_k = v_k$ is the predicted circuit voltage also known as measurable output:

$\hat{y}_k = (C - DK) \hat{x}_k$

Input to the extend kalman filter (EKF) is current and voltage measurement and the period of time between these measurements. The initialization of the EKF outputs the initial estimate state of charge $x_{k|k=0}$ another input to the EKF.

System's dynamic model

The $f()$ function is defined as the $f(x_k, i_k, t) = x_k - \frac{1}{Q_C} \int_0^t i_k dk$, thus current measurement i_k . The $h(x_k)$ function is defined as the OCV lookup table. A general OCV lookup table for the battery chemistry can be used or for better results a specific OCV lookup established by offline measurements for the given battery should be used. To further improve the OCV prediction a correction of it can be performed by a equivalent circuit model (ECM) of the battery feeded by the current measurement used to predict the SOC: $v_k = D OCV(z_k) + Cx_k + Di_k$ with $C = [0, -R_1, -R_2, \dots, M]$ and $D = R_0$

-Measurement equation, input (measured voltage, OCV lookup table, current if the correction with a Enhanced Self-Correcting (ESC) Cell Model /ECM) -> output SOC) equation should be use standard letters: filterpy, wikipedia, gregoryPlett, Step by Step Guide

After having defined the observation and control-input model as matrices and described their meaning in case of SOC estimation we proceed to the functioning of the EKF, which is typically divided into to steps, Predict and Update.

In the **predict** step a future state of charge estimate \hat{x}_{k+1} is predicted based on the current state of charge estimate \hat{x}_k and the current i_k during the period Δt (between k and $k + 1$) by calculating $\hat{x}_{k+1} = Ax_k + Bu_k$. Moreover an estimate of the covariance P_{k+1} is calculated based on noise covariances Q_k and current P_k (wiki: a measure of the estimated uncertainty of the prediction of the system's state) $P_{k+1} = B_k P_k B_k^T + Q_k$

In the **update** step

$K_k = P_{k+1} H_k^T (H_k P_{k+1} H_k^T + R_k)^{-1}$ is the kalman gain, which weights whether the SOC based on the measurement of the circuit voltage v_k is more trusted than the SOC prediction based on current i_k

$\hat{x}_k = (I - K_k H_k)(\hat{x}_{k+1}) + (K_k)(H_k \hat{x}_k + v_k)$ TODO update \hat{x}_k because one can not now want one want to calculate

$P_{k+1} = (I - K_{k+1} H_{k+1}) P_k$ update of the covariance TODO is H_k the same as OCV?

\mathbf{R}_k the covariance of the observation noise

Most general equation of extended kalman filter EKF (state transition and observation models):

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k$$

$f()$ -> predicted state from the previous estimate $h()$ -> compute the predicted measurement from the predicted state

2.3 Kalman-SOC

A Fork of Okra-Solar Algorithm.

Features:

- Works without the input of a Equivalent Circuit Model (ECM) specific to the physical battery, which would need to be parameterized doing advanced measurements during charging and discharging of the battery.

- Inputs: Current and Voltage Measurements and OCV Lookup Table

- Outputs: SOC in %Wh

TODO Discuss the Code Snippets? But Some Code in Appendix or reference Github Repo only?

Kapitel 3

Validation

Integrated matplotlib inline with pythontex
Discuss results. (4d)

Kapitel 4

Outlook

Quote: **online parameter estimation** from the ARX model

State of Charge and State of Health Estimation of AGM VRLA Batteries by Employing a Dual Extended Kalman Filter and an ARX Model for Online Parameter Estimation

Anhang A

Angehängtes: Die Dateien des Pakets

Stylefile

Die Styledatei für diese Abschlussarbeit ist `bhtThesis.sty`, die in der Archivdatei vorliegt. Diese muss von \LaTeX auffindbar sein, muss also in einem \LaTeX bekannten Ordner liegen:

- Ubuntu-Linux: `$HOME/texmf/tex/latex/bhtThesis/bhtThesis.sty`
- MikTeX: `c:\localtexmf\tex\latex\bhtThesis\bhtThesis.sty`

Beispieldokument

Dieses Dokument befindet sich im Unterordner `tryout` des zip-files. Sie können diese Dateien in einen Ordner kopieren, in dem Sie schliesslich arbeiten werden. Die Dateien sind die folgenden

- `abstract_de.tex` Kurzfassung in deutscher Sprache
 - `abstract_en.tex` Kurzfassung in englischer Sprache
 - `anhang.tex` der Anhang
 - `bhtThesis.bib` beinhaltet die zu zitierenden Literaturstellen und wird von $\text{bib}\text{\TeX}$ ausgewertet
 - `main.pdf` ist die Ausgabendatei mit der Druckvorlage
 - `main.tex` beinhaltet das Hauptdokument
 - `makefile` realisiert das automatische mehrfache Übersetzen, hierfür muss `make` auf dem System installiert sein.
 - `myapalike.bst` beinhaltet die Formatierung für das Literaturverzeichnis
 - `personalMacros.tex` kann einzelne, persönliche Macros beinhalten, die das Schreiben erleichtern
 - `titelseiten.tex` realisiert alle Seiten bis zum Beginn des ersten Abschnittes
 - Ordner `pictures`
 - `BHT-Logo-Basis.eps`
 - `BHT-Logo-Basis.pdf`
 - Ordner `kapitel1`
 - `ch1.tex` Quelltext des Kapitel 1
-

- Ordner `pictures`
 - * `schaltbild.pdf`
- Ordner `kapitel2`
 - `ch2.tex` Quelltext des Kapitel 2
 - Ordner `pictures`
 - * `leer`

Literaturverzeichnis