BVM13TMO HW10: Online Parameter Estimation

Vojtěch Michal, michavo3

Abstract

This report presents an implementation of two methods for online estimation of battery model parameters. Recursive Least Squares (RLS) take a discrete-time transfer function and search for values of its coefficients. An Extended Kalman Filter (EKF) on the other hand considers the system in the state-space augmented by auxiliary states representing the – possibly time-varying – system parameters and estimates those. Both methods were derived for the battery equivalent circuit model with a single RC element and tested using a recording of a dynamic discharge profile.

Index Terms

Online parameter estimation, Recursive Least Squares, Extended Kalman Filter.

I. RECURSIVE LEAST SQUARES

For simplicity, consider the RLS algorithm for a simple ARX model, i.e. a general discrete-time transfer function in the time-domain delay operator d

$$G(d) = \frac{Y(d)}{U(d)} = \frac{b_0 + b_1 d + \cdots + b_{n_b} d^{n_b}}{1 + a_1 d + \cdots + a_{n_a} d^{n_a}} = \frac{\sum_{i=0}^{n_b} b_i d^i}{\sum_{i=0}^{n_a} a_i d^i}, \tag{1}$$

usually normalized to unity a_0, i.e. the coefficient belonging to $y(k)$ in the corresponding difference equation

$$y(k) + \sum_{i=1}^{n_a} a_i y(k-i) = \sum_{i=0}^{n_b} b_i u(k-i) + e(k) \tag{2}$$

relating past samples of the input u and output y up to the discrete time k. The variable $e(k)$ encapsulates stochastic phenomena such as noise, as well as unmodelled system dynamics.

The most recent output $y(k)$ can be predicted from (2) using past inputs and outputs and system parameters θ as

$$\hat{y}(k) = -\sum_{i=1}^{n_a} a_i y(k-i) + \sum_{i=0}^{n_b} b_i u(k-i), \tag{3}$$

that can be seen as a dot product

$$\hat{y}(k) = \varphi^T(k)\theta(k) \tag{4}$$

of the regressor vector

$$\varphi(k) = \begin{bmatrix} -y(k-1) & -y(k-2) & \ldots & -y(k-n_a) & u(k) & u(k-1) & \ldots & u(k-n_b) \end{bmatrix}^T \tag{5}$$

containing past inputs and outputs and the vector of parameters

$$\theta(k) = \begin{bmatrix} a_1 & a_2 & \ldots & a_{n_a} & b_0 & b_1 & \ldots & b_{n_b} \end{bmatrix}^T. \tag{6}$$

At each iteration k of the RLS algorithm, the actual measured output $y(k)$ is compared with the prediction $\hat{y}(k)$ from (4) to find an error $e(k)$. Then the Kalman gain K is calculated using the current estimate covariance matrix $\mathbf{P}$ and finally both the estimated system parameters $\hat{\theta}$ as well as the covariance matrix $\mathbf{P}$ are updated using K and e. The algorithm can be extended by a real forgetting factor $\lambda \in [0,1]$ used to artificially inflate the covariance matrix $\mathbf{P}$ to facilitate tracking of time-varying parameters.

The algorithm can be generalized and applied far beyond the limitation of linear ARX model. To use the RLS algorithm for any general problem, one needs to

1) express the prediction task using an equation linear-in-parameters ,
2) determine what expressions are to be estimated and belong to the parameter vector θ
3) determine what expressions are known and belong to the regressor φ,
4) find the (generally non-linear) mapping between parameters θ and physically meaningful parameters useful to the application.

A. RLS for battery parameters

The continuous-time 1 RC equivalent circuit model reads

$$U_{\text{bat}} = U_{\text{OC}} - i\,R_0 - i\underbrace{\frac{R_1}{C_1\,R_1\,s + 1}}_{U_1}, \tag{7}$$

where i is the flowing current, U_{bat} is the measured terminal voltage, U_{OC} is the cell's open-circuit voltage, R_j, C_j are parameters of the model and s is the Laplace transform operator. By relabeling $E = U_{\text{bat}} - U_{\text{OC}}$, (7) can be rewritten as a transfer function

$$G_{\text{cont}}(s) = -R_0 - \frac{R_1}{C_1\,R_1\,s + 1} \tag{8}$$

from the flowing current i to the voltage drop E. The Tustin (bilinear) discretization of $G_{\text{cont}}(d)$ with step T gives

$$G(d) = \frac{(R_0\,T + R_1\,T - 2\,C_1\,R_0\,R_1)\,d + R_0\,T + R_1\,T + 2\,C_1\,R_0\,R_1}{(2\,C_1\,R_1 - T)\,d - T - 2\,C_1\,R_1}. \tag{9}$$

Coparing (9) to a general first-order discrete-time transfer function

$$\frac{b_1 d + b_0}{a_1 d + 1} \tag{10}$$

yields substitutions

$$a_1 = \frac{T - 2\,C_1\,R_1}{T + 2\,C_1\,R_1}, \tag{11}$$

$$b_1 = -\frac{R_0\,T + R_1\,T - 2\,C_1\,R_0\,R_1}{T + 2\,C_1\,R_1}, \tag{12}$$

$$b_0 = -\frac{R_0\,T + R_1\,T + 2\,C_1\,R_0\,R_1}{T + 2\,C_1\,R_1}. \tag{13}$$

Using the discretized transfer function $G(d)$, (7) can be rewritten in several steps

$$U_{\text{bat}} - U_{\text{OC}} = iG(d), \tag{14}$$

$$(U_{\text{bat}} - U_{\text{OC}})(a_1 d + 1) = i(b_1 d + b_0), \tag{15}$$

$$U_{\text{bat}} + U_{\text{bat}}a_1 d - U_{\text{OC}}(a_1 d + 1) = i(b_1 d + b_0), \tag{16}$$

$$U_{\text{bat}} = U_{\text{OC}}(a_1 d + 1) - U_{\text{bat}}a_1 d + i(b_1 d + b_0). \tag{17}$$

Finally, assuming that the ΔU_{OC} is negligible on each sampling interval and therefore $U_{\text{OC}}(k) \approx U_{\text{OC}}(k - 1)$, (17) can be converted from the d-operator domain to index shifts

$$U_{\text{bat}}(k) = (a_1 + 1)U_{\text{OC}} - a_1 U_{\text{bat}}(k - 1) + b_0 i(k) + b_1 i(k - 1). \tag{18}$$

Equivalently in the short vector notation

$$U_{\text{bat}}(k) = \varphi^T(k)\theta, \tag{19}$$

where

$$\theta = \begin{bmatrix} (a_1 + 1)U_{\text{OC}} & a_1 & b_0 & b_1 \end{bmatrix}^T, \tag{20}$$

$$\varphi(k) = \begin{bmatrix} 1 & -U_{\text{bat}}(k - 1) & i(k) & i(k - 1) \end{bmatrix}^T. \tag{21}$$

The problem at hand was successfully described using an equation (18) that is linear in parameters θ and hence can be subject to RLS. Equations (20) and (21) describe the role of individual variables (θ are estimated) and equations (12) through (13) describe the nonlinear mappings from physically meaningful battery parameters to elements of θ with the inverse mapping given by

$$R_0 = \frac{b_0 - b_1}{a_1 - 1}, \tag{22}$$

$$R_1 = \frac{2\,(b_1 - a_1\,b_0)}{a_1{}^2 - 1}, \tag{23}$$

$$C_1 = -\frac{T\,a_1{}^2 - 2\,T\,a_1 + T}{4\,(b_1 - a_1\,b_0)}. \tag{24}$$

Performance of the estimation algorithm is presented in Section III.

II. EXTENDED KALMAN FILTER

The EKF was discussed multiple times in previous reports, mainly as a great tool for SoC estimation. This time, it is used to estimate states of a state-space model augmented with model parameters. The 1 RC ECM from (7) can be rewritten as a state-space

$$\dot{U}_1 = -\frac{1}{R_1 C_1} U_1 + \frac{1}{C_1} i \tag{25}$$

$$U_{\text{bat}} = U_{\text{OC}} - i R_0 - U_1 \tag{26}$$

with input i, output U_{bat} and state U_1 (the polarization voltage). To estimate some model parameter M, it must be added to the list of system states. Without any apriori knowledge of parameter evolution, $\dot{M} = 0$ is assumed such that the parameter evolution is driven only by a random walk of the process noise. Two distinct cell state-space representations based on the 1 RC equivalent circuit model were implemented and tested – the reference model "Hongwen" with six states from [1] and my own manually derived model "Michal" with five states. Both models have the flowing current i as input u and the terminal voltage U_{bat} as output y.

A. Hongwen model

The Hongwen model considers six states

$$x = \begin{bmatrix} U_{\text{OC}} & U_{\text{bat}} & U_1 & \frac{1}{C_1} & \frac{1}{R_1} & R_0 \end{bmatrix}^T \tag{27}$$

with state equations

$$
\begin{aligned}
\dot{x}_1 &= 0, \\
\dot{x}_2 &= x_1 x_4 x_5 - x_2 x_4 x_5 - (x_4 x_5 x_6 + x_4) u - x_6 \dot{u}, \\
\dot{x}_3 &= x_4 u - x_3 x_4 x_5, \\
\dot{x}_4 &= 0, \\
\dot{x}_5 &= 0, \\
\dot{x}_6 &= 0.
\end{aligned} \tag{28}
$$

This model includes the terminal voltage U_{bat} as state x_2, making the output equation trivial

$$y = x_2. \tag{29}$$

This model is quite complex and requires a non-standard addition of $\dot{u}$ (the derivative of input) that may introduce complications of its own.

B. Michal model

The Michal model eliminates the terminal voltage U_{bat} and considers only five states

$$x = \begin{bmatrix} U_{\text{OC}} & U_1 & \frac{1}{C_1} & \frac{1}{R_1} & R_0, \end{bmatrix}^T \tag{30}$$

with state equations

$$
\begin{aligned}
\dot{x}_1 &= 0, \\
\dot{x}_2 &= x_3 u - x_2 x_3 x_4, \\
\dot{x}_3 &= 0, \\
\dot{x}_4 &= 0, \\
\dot{x}_5 &= 0
\end{aligned} \tag{31}
$$

and a non-linear output equation

$$y = x_1 - x_2 - u x_5. \tag{32}$$

This model has fewer states, does not require the numeric differentiation of input u and is generally much simpler than the Hongwen model. Results of online parameter estimation with both models are presented in Section III.

III. ESTIMATION RESULTS

All three methods (RLS estimation from Section I, EKF for Hongwen model from Section II-A and EKF for Michal model from II-B) were compared using waveforms from a dynamic discharge profile. The achieved prediction error is shown in Fig. 1; Fig. 1(a) shows predictions throughout several DDP cycles whereas Fig. 1(b) focuses on one DDP period. Low prediction error less than 10 mV confirms good convergence of model parameters to the behavior of the actual battery, although there is certainly some room for improvement (e.g. by introducing a second RC element).
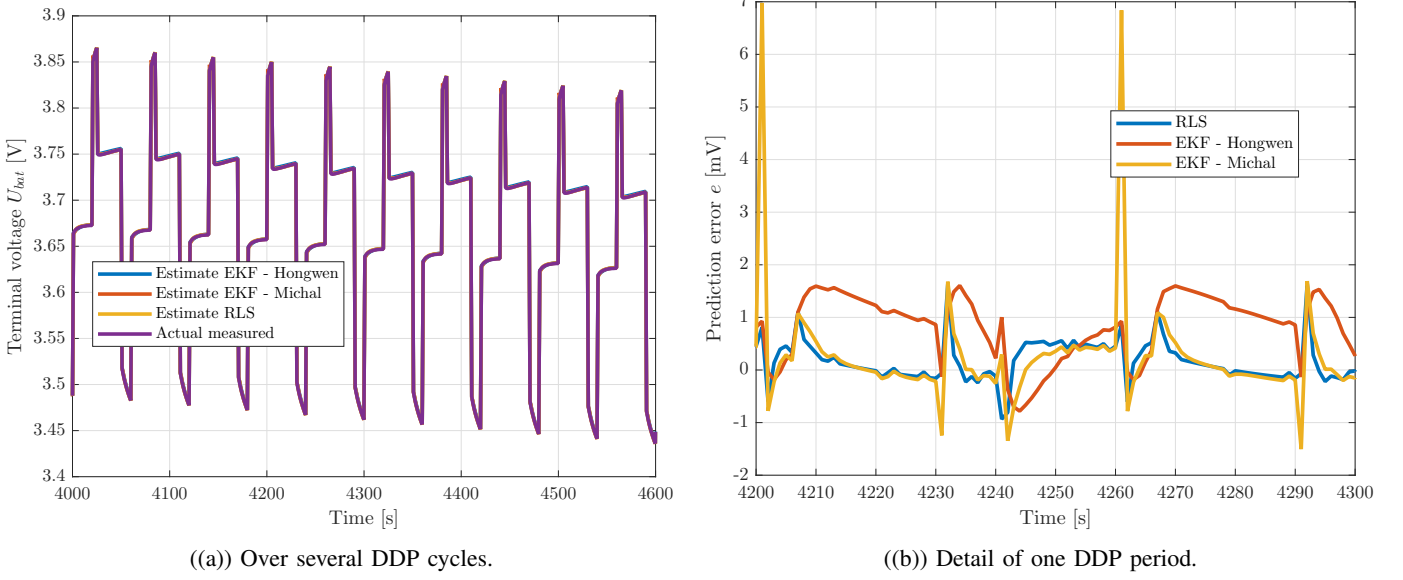

((a)) Over several DDP cycles.

((b)) Detail of one DDP period.

Fig. 1: Comparison of terminal voltages predicted by individual algorithms.

((a)) Estimates of internal impedance R_0.

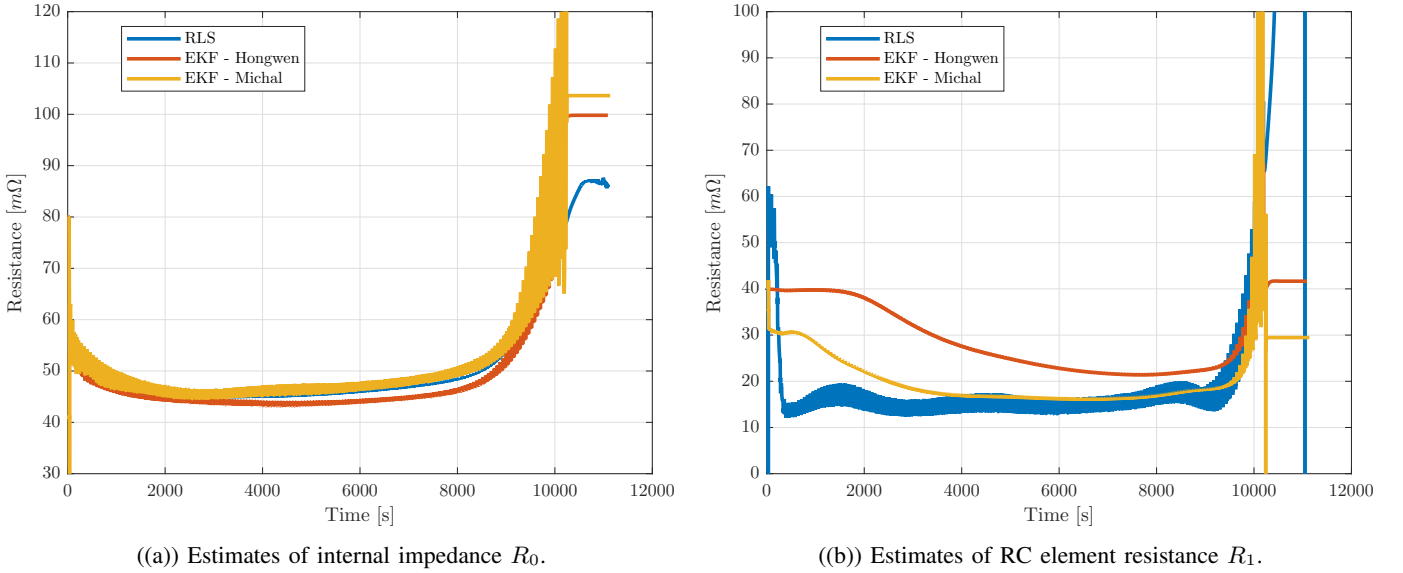((b)) Estimates of RC element resistance R_1.

Fig. 2: Estimates of resistive ECM parameters.

Estimates of both components of the ohmic (real) impedance R_0 and R_1 are shown in Fig. 2(a) and Fig. 2(b), respectively. Estimates of R_0 yielded by various algorithms are quite similar and they demonstrate the expected (approximately) flat plateau roughly in the range of 80 % to 20 % of SoC with growing value towards extremes. On the other hand, estimates of R_1 shown in Fig. 2(b) did not converge as satisfactorily with a significant discrepancy between both EKFs.

The estimated capacitance of the RC element shown in Fig. 3 can be used to demonstrate an important difference between the RLS- and EKF-based parameter estimation. The EKF method allows the user to directly specify variance for individual physically meaningful parameters (albeit possibly inverted, such as C_1^{-1}). In contrast, the RLS works in terms of some artificial parameters b_i and a_i, whose dependence on physically meaningful parameters is non-linear even for a 1 RC ECM and becomes
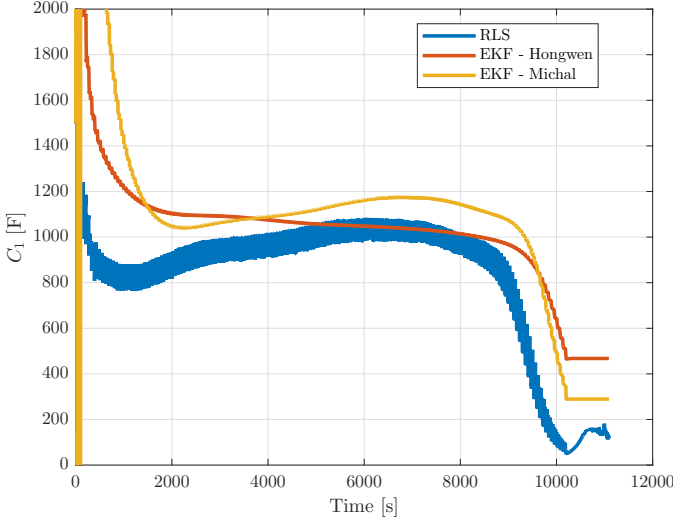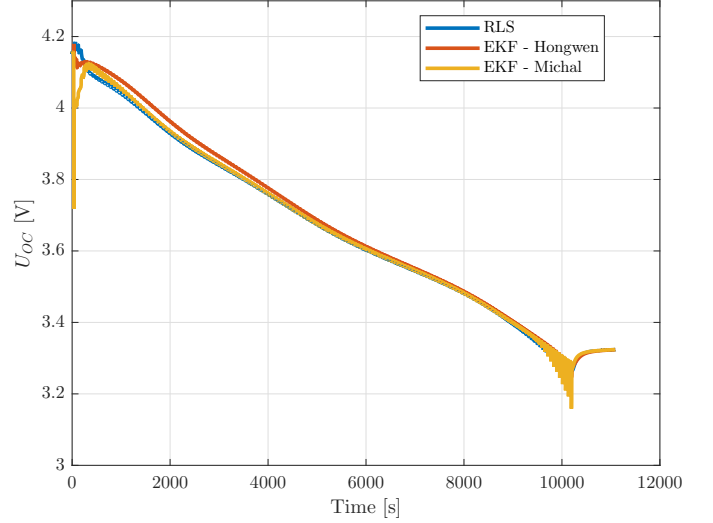
Fig. 3: Estimates of RC element capacitance C_1.

Fig. 4: Estimates of the open circuit voltage U_{OC}.

increasingly more entangled with the growing number of parameters. Therefore, it is possible to get a smooth waveform for some given parameter (here C_1) with the EKF simply by reducing the parameter's variance. When estimating using RLS on the other hand, such fine control is very hard to do and the resulting estimate tends to be more noisy. A similar phenomenon can be observed in Fig. 2(b).

All three methods estimate the U_{OC} (shown in Fig. 4) similarly well. The main difference from the implementation point of view is that the EKF method was far simpler to set up and debug owing to the tight bound of individual states to underlying physics. In contrast, the RLS method uses artificial (physically meaningless) parameters whose values must be post-processed to obtain any physically useful information. Typos such as an incorrect sign in some of the RLS expressions may cause a lot of headaches. Nevertheless, it is computationally less demanding than the EKF estimation and the forgetting factor λ allows an intuitive configuration of the steady-state covariance matrix.

It is still unknown to me why the authors of [1] decided to use an unnecessarily complicated state space model for the parameter estimation task. Despite my best efforts in hyperparameter tuning (especially tuning the process noise covariance matrices Q), the simpler state-space model Michal achieved more credible results (closer matching the RLS estimates that I see as the best available reference in the absence of some ground truth parameter values) while being easier to debug and implement.

REFERENCES

[1] Hongwen He, et al. *"Online estimation of model parameters and state-of-charge of LiFePO4 batteries in electric vehicles"*, Applied Energy, Volume 89, Issue 1, 2012, https://doi.org/10.1016/j.apenergy.2011.08.005.