Tutorial on modelling and identification

The most widely used model-type for vehicle batteries is the equivalent circuit model, since its dynamics can be regarded as linear and because appropriately parameterized it can mimic and predict the electrical behaviour with high accuracy. The linearity implies that there is a large toolbox of computationally efficient methods that can be applied, both when it comes to estimation of the model parameters from data (also called identification) and for state estimation.

Assume that the relation between a system input u and output y can be described by the simplest possible linear mapping

$$y(k) = u(k)\theta + e(k), \tag{1}$$

where k is the sample and e is a noise term. If we collect samples from $k = 1$ to $k = N$ from this system, this can be written as

$$Y_N = U_N\theta + E_N,$$

where

$$\begin{aligned} Y_N &= \begin{bmatrix} y(1) & y(2) & \dots & y(N) \end{bmatrix}^T \\ U_N &= \begin{bmatrix} u(1) & u(2) & \dots & u(N) \end{bmatrix}^T \\ E_N &= \begin{bmatrix} e(1) & e(2) & \dots & e(N) \end{bmatrix}^T \end{aligned}$$

Let the model of the system be

$$\hat{y}(k) = u(k)\hat{\theta}.$$

This model is on linear regression form, and given the data U_N and Y_N, and assuming the noise has zero mean, the model that gives the best fit can be determined by least squares. Minimizing the squared residuals, i.e. $\sum_{k=1}^{N}(\hat{y}(k) - y(k))^2$, gives

$$\begin{aligned} \hat{\theta} &= \left[\frac{1}{N}\sum_{k=1}^{N} u^2(t)\right]^{-1} \frac{1}{N}\sum_{k=1}^{N} u(t)y(t) \\ &= (U_N^T U_N)^{-1} U_N^T Y_N \end{aligned}$$

which can be proven by setting the derivative w.r.t. $\hat{\theta}$ to zero.

Exercise 1

The simplest possible equivalen circuit model of a battery cell is a resistor in series with a voltage source (OCV), see Figure 1.

Download the file `ECM1.mat` *(*`load ECM1`*) where we have measured time-series of battery current from about half an hour of city driving in a BMW i3*[1]*, their time*

[1] https://ieee-dataport.org/open-access/battery-and-heating-data-real-driving-cycles

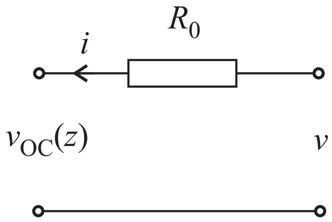

Figure 1: Static battery model. v_{OC} is the open circuit voltage, which depends on the state of charge z, i is the current and v is the terminal voltage.

stamp and simulated voltage using the model in Figure 1 and assuming a normal distributed voltage error with a standard deviation of 10 mV. Also in the file is the SoC-OCV map (`OCV`), where the first column is the state of charge and the second column is the open circuit voltage.

The current profile for this trip is shown in Figure 2.

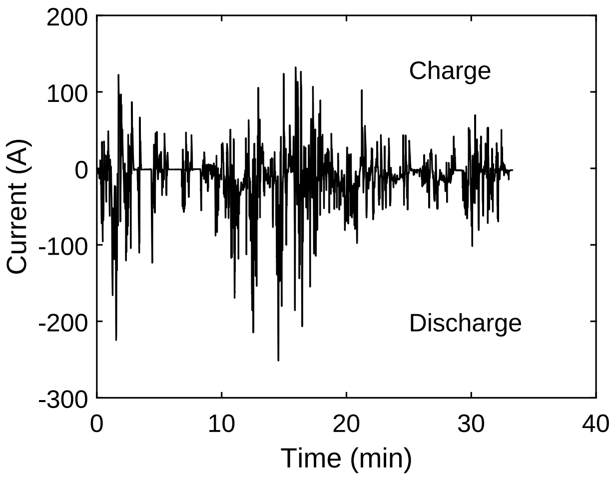

Figure 2: Current during a city driving trip (TripA27).

(a) Plot v_{OC} vs. z. Assume that SoC is 50% and try to use Ohm's law directly to calculate the resistance R_0 at each time instant.

Plot the calculated resistances at each time instant. What is the mean of R_0? The "true" value of the resistance is $R_0 = 2.5m\Omega$. Why does this not work?

(b) Use standard least squares to estimate R_0 from the current-voltage data. Why is this estimate much better, but why is it still wrong?

The above system was both scalar and static. As argued in the lecture we often need to regard the batteries as dynamical systems. To proceed we, therefore, assume we have a linear continuous-time system on state-space form

$$\frac{d}{dt}x(t) = A_c x(t) + B_c u(t) \tag{2}$$

$$y(t) = Cx(t) + Du(t) \tag{3}$$

The equivalent circuit models can often be written on this form, with the modification that the output will also depend nonlinearly on the state of charge through the open circuit voltage.

Like all processors, however, the BMS works in sampled time (discrete time). The above system can be translated to discrete time given what the input is between the samples. The analytical solution to Eqn. (2) is

$$x(t) = e^{A_c(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A_c(t-\tau)}B_c u(\tau)d\tau.$$

Usually we can regard the input as being constant between the sampling instants, i.e.,

$$u(t) = u(k), \quad t \in [kT_s, (k+1)T_s),$$

where T_s is the sampling time.

Setting $t_0 = kT_s$ and $t = (k+1)T_s$ we can then get the discrete-time state-space model corresponding to Eqns. (2) and (3):

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k) && (4)\\ y(k) &= Cx(k) + Du(k) && (5) \end{aligned}$$

where

$$A_d = e^{A_c T_s} \quad \text{and} \quad B_d = \int_0^{T_s} e^{A_c \tau} d\tau B_c.$$

In Matlab you can easily do this translation using the command `c2d`.

To estimate the parameters in this state space model from measured input (current) and output (voltage) data we would like to have the model on the parameter-linear model form corresponding to (1). As a step on the way there, we start by determining the transfer function for the input-output relation of (4) and (5). For that purpose we introduce the shift operator q, defined by

$$q^m y(k) = y(k+m),$$

where m is an integer that can be negative. Applying this to (4) we can write

$$(qI - A_d)x(k) = B_d u(k)$$

and, consequently, using (5)

$$y(k) = \underbrace{\left(C(qI - A_d)^{-1}B_d + D\right)}_{G(q)} u(k). \tag{6}$$

If we are regarding only one battery cell, with input being current and output being voltage, the transfer function $G(q)$ will have the following form

$$\begin{aligned} G(q) &= \frac{b_0 q^n + b_1 q^{n-1} + \ldots + b_n}{q^n + a_1 q^{n-1} + \ldots + a_n} \\ &= \frac{b_0 + b_1 q^{-1} + \ldots + b_n q^{-n}}{1 + a_1 q^{-1} + \ldots + a_n q^{-n}} = \frac{B(q^{-1})}{A(q^{-1})}, \end{aligned}$$

where n is the number of states, i.e. $\dim(x)$. Multiplying (6) from the left with the denominator polynomial operator $A(q^{-1})$ gives

$$(1 + a_1 q^{-1} + \ldots + a_n q^{-n})y(k) = (b_0 + b_1 q^{-1} + \ldots + b_n q^{-n})u(k)$$

or, equivalently,

$$y(k) + a_1 y(k-1) + \ldots + a_n y(k-n) = b_0 u(k) + b_1 u(k-1) + \ldots b_n u(k-n),$$

which we can write on the regressor form

$$y(k) = \varphi^T(k)\theta, \tag{7}$$

where

$$\begin{aligned} \theta &= \begin{bmatrix} a_1 & \ldots & a_n & b_0 & b_1 & \ldots & b_n \end{bmatrix}^T \\ \varphi(k) &= \begin{bmatrix} -y(k-1) & \ldots & -y(k-n) & u(k) & u(k-1) & \ldots & u(k-n) \end{bmatrix}^T. \end{aligned}$$

Stacking N consecutive outputs in one column vector Y_N and the regressors φ^T in a corresponding matrix Φ_N, we get

$$Y_N = \Phi_N \theta,$$

where

$$\begin{aligned} Y_N &= \begin{bmatrix} y(n+1) & y(n+2) & \ldots & y(n+N) \end{bmatrix}^T \\ \Phi_N &= \begin{bmatrix} \varphi^T(n+1) \\ \vdots \\ \varphi^T(n+N) \end{bmatrix} \end{aligned}$$

This is now on the same form as we had in the static case, and we can therefore proceed in the same manner as then to estimate the parameter vector θ using least squares. Inserting the measurements y and u into (7) will give rise to an error

$$\varepsilon(k) = y(k) - \varphi^T(k)\hat{\theta}. \tag{8}$$

The parameter estimate that minimizes the loss function

$$V(\theta) = \frac{1}{2} \sum_{k=n+1}^{n+N} \varepsilon^2(k) \tag{9}$$

is the least squares solution

$$\hat{\theta} = (\Phi_N^T \Phi_N)^{-1} \Phi_N^T Y_N.$$

Exercise 2

In the estimation of R_0 in Figure 1 we assumed that the SoC was constant and that the model was

$$v(k) - v_{OC}(z) = R_0 i(k).$$

Since the SoC (z) actually changes during the data collection we should get a better estimate of R_0 if we also include a varying (dynamic) SoC in our model.

(a) *Make the previous model dynamic by including the state of charge as a dynamic state. If you assume the coulumbic efficiency is 1, you can use the definition*

$$\frac{d}{dt} z(t) = \frac{1}{Q} i(t),$$

which in discrete time can be approximated by

$$z(k+1) = z(k) + \frac{T_s}{Q} i(k).$$

The capacity of the cell is $Q = 60\ Ah$.

(b) *Now, you can estimate R_0 using least squares in (at least) two different ways. One is to calculate $z(k)$ in open loop from i, and from that calculate the corresponding v_{OC} at each time instant. Subtracting that from the measured terminal voltage then gives an output that is linear w.r.t to the current in the same way as before. Try this method to see if the estimate of R_0 improves compared to the result of Exercise 1.*

(c) *Another method is to use that the relationship between z and v_{OC} is approximately linear for this SoC range[2]. Derive a regression model for this case and apply least squares to the resulting model.*

What do you think are the advantages/disadvantages of the methods in (a) and (b)? Scrutinizing the parameter vector in the second case you will notice that the effect of the SOC-OCV relation is negligible. Why is that?

There are two issues with the least squares method as presented. The first one is that the problem grows out of hand as N gets larger and larger. The number of calculations and the required memory keeps on increasing as the number of measurements increases. This, however, can be elegantly solved by reformulating the solution such that it becomes recursive, which means that the estimate at time k can be expressed in terms of the estimate at time $k-1$ and the additional measurements at time k only. The other complication is that the parameters we are estimating are in reality not constants. If they were, we would simply estimate them once, early in the life of the cells, and then use these values for all times. Unfortunately, the

[2]You may assume $v_{OC}(k) \approx Kz(k)$. More correctly we could use $v_{OC}(k) \approx K_1 + K_2 z(k)$, but that will not affect the results here. When identifying the parameters in a transfer function, it is recommended to subtract the mean of the signals and then the effect of K_1 will be removed.

dynamics of the cells keep changing as the cells age and they also change with operating point (temperature and level of current). In fact, it is an automatic tracking of these changes that we actually want to achieve.

One way to achieve such adaptivity is to modify the least squares such that, instead of putting equal weight on all residuals ε in the loss function (9), less and less weight is put on them the older they are. The modified loss function to be minimized at time k takes the form

$$V(k) = \frac{1}{2} \sum_{i=n+1}^{k} \lambda^{k-i} \varepsilon^2(i)$$

where $0 \ll \lambda < 1$ is a so-called *forgetting factor*.

Minimizing this loss function can be formulated as a recursive algorithm:[3]

$$\begin{aligned}
\varepsilon(k) &= y(k) - \varphi^T(k)\hat{\theta}(k-1) \\
K(k) &= \frac{P(k-1)\varphi(k)}{\lambda + \varphi^T(k)P(k-1)\varphi(k)} \\
P(k) &= \frac{1}{\lambda}\left(I - K(k)\varphi^T(k)\right) P(k-1) \\
\hat{\theta}(k) &= \hat{\theta}(k-1) + K(k)\varepsilon(k)
\end{aligned}$$

Note that to start the algorithm, one has to provide initial values for P and $\hat{\theta}$. Here, ε should be considered a prediction error, and K a correction factor saying how much the prediction $\hat{\theta}(k-1)$ should be adjusted based on the prediction error. $P(k)$ is a measure of the (co)variance of $\hat{\theta}(k)$. In fact, if $\lambda = 1$ and the residuals are uncorrelated and have variance σ^2, then the variance of $\hat{\theta}(N)$ is $P(N) = \sigma^2(\Phi_N^T \Phi_N)^{-1}$.

Exercise 3

(a) If the forgetting factor is set to $\lambda = 1$, how should the estimated parameter vector using recursive least squares (RLS) compare to normal least squares after the last data point?

(b) Apply RLS to either of your previous problems and try different values of λ, which should be fairly close to 1 (e.g. 0.97-0.999). What is the effect on the convergence rate and the noise attenuation.

(c) The inital value of P reflects how certain your inital guess of θ is. Try different initial values of P (change in powers of 10). What is the effect? Can you relate this to how certain you are of your initial $\hat{\theta}$?

To achieve good tracking of the voltage at least a first order equivalent circuit model is usually required, see Figure 3. We will now do parameter identification on this

[3] See, e.g., *System Identification* by Söderström and Stoica.

model. As in the previous example we can use (at least) two different situations: One where we can consider the state of charge as known, either by coloumb counting or by estimation with a Kalman filter, for example. Another option is that we go ahead and simply assume the relation between v_{OC} and z is (locally) linear, as in the previous exercise.

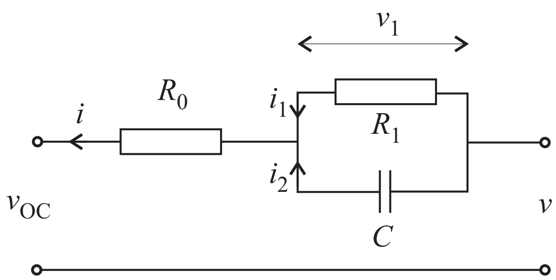

Figure 3: First order equivalent circuit model. v_{OC} is the open circuit voltage, which depends on the state of charge z, i is the current and v is the terminal voltage.

Exercise 4

(a) Consider the state of charge as known and derive a regression model for the system in Figure 3. The corresponding state-space model you may find on pages 19-20 in the lecture slides.

(b) Download the data `ECM2.mat` *that has been generated using such a first order equivalent circuit model and the same current as before, and try first to estimate the parameters using standard least squares. The "true" values of the components are $R_0 = 0.001\Omega$, $R_1 = 0.0015\Omega$ and $C = 10000F$. The capacity is still 60Ah. How well is the estimation working, is it worse or the same as before?*

(c) Run the file `ECM2data` *(which creates* `ECM2.mat`*), reducing the voltage noise* `w` *by a factor 10. How does the estimation accuracy change?*

(d) Keep the lower noise level and try recursive least squares on the new data.

(e) Create your own battery data with measurement noise by modifying `ECM2data` *such that you get gradual changes to the parameters. See how well the RLS can capture the changes and what effect you get of the noise level.*

(f) Try the same thing not using the knowledge of z.