Case Study: Cell Model by Equivalent Circuit Model (ECM)

Aydin Emre Utku

The solution is composed of below parts:

1) ECM Parameter Extraction and LuT Generation
2) ECM R0 Correction
3) ECM RC Correction
4) Simulation and Validation of Extracted ECM Parameters
5) EKF Based SOC Estimator

1) ECM Parameter Extraction and LuT Generation (EcmParamExtrctnAndLutGen.py)

This task is done by using the HPPC and WLTP data from Cycle 0, 206 km case (10-24-19_16.28 960_WLTP206a.mat and 10-16-19_20.16 948_HPPC.mat). The resulting lookup table contains the ECM parameters. Firstly, the pulses from HPPC data are processed. The currents below a certain threshold are eliminated to smooth the input data. Candidate pulses are detected and confirmed if the any pulse is detected to continuous and flat enough. R0 (Ohmic Resistance) values are calculated from the start of the pulses since the transient RC voltage response hasn't evolved yet. The detected voltage jump during these step beginnings is mostly caused by the R0 terms, thus these data is used for R0 calculation. Then, the script checks for the relaxation window, to be used for RC parameter fitting. Voltage responses after the end of each pulse is fit to two RC ECM model:

$$V(t) = OCV(SOC) - IR_1\left(1 - e^{-\frac{t}{R_1C_1}}\right) - IR_2\left(1 - e^{-\frac{t}{R_2C_2}}\right) \tag{1}$$

For the OCV table, corresponding OCV data from the Excel file 1-A123 Aging Tests_Data Summary is used. By using equation (1), a nonlinear least squares fit is used to extract the 2-RC parameters. The resulting lookup table is saved as ecm_lookup_table_hppc.csv.

The second part of this task is refining the lookup table, generated based on HPPC data, further by using the WLTP cycle data. WLTP is observed to be varying between 65% - 45% SOC cyclically. The data between this range is used to refine the lookup table without disrupting earlier entries. By using interpolation, additional data is created and added to the earlier lookup table, which resulted in a richer ECM lookup table. The resulting lookup table is saved as ecm_lookup_table_refined_with_wltp.csv.

However, the performance of this lookup table was not enough. The refined lookup table is simulated by using another WLTP data (10-25-19_11.29 960_WLTP206b.mat) from Cycle 0 and another HPPC data from Cycle 900 as shown in Figure 1 and Figure 2. (ECM implementation and simulation part will be covered in section 4). The resulting mean absolute error was 146 mV, which signals a poor performance of ECM parameters. In Figure 1 and 2, it can be observed that error spikes up when the current also spikes up. This highlights that the extracted R0 parameters cause exaggerated voltage responses.

2) ECM R0 Correction (R0Corrn.py)

To overcome the exaggeration of voltage responses during high currents, I decided to further improve the R0 parameters using R0Corrn.py script without touching RC parameters. By using the WLTP and HPPC data used in Section 1, the ECM is simulated and voltage errors are calculated.

By using this voltage error, a correction model $\Delta R_0(SOC, C-rate)$ is fitted by using Weighted Ridge Regression, which is an iterative weighted least squares fit method that applies bounded

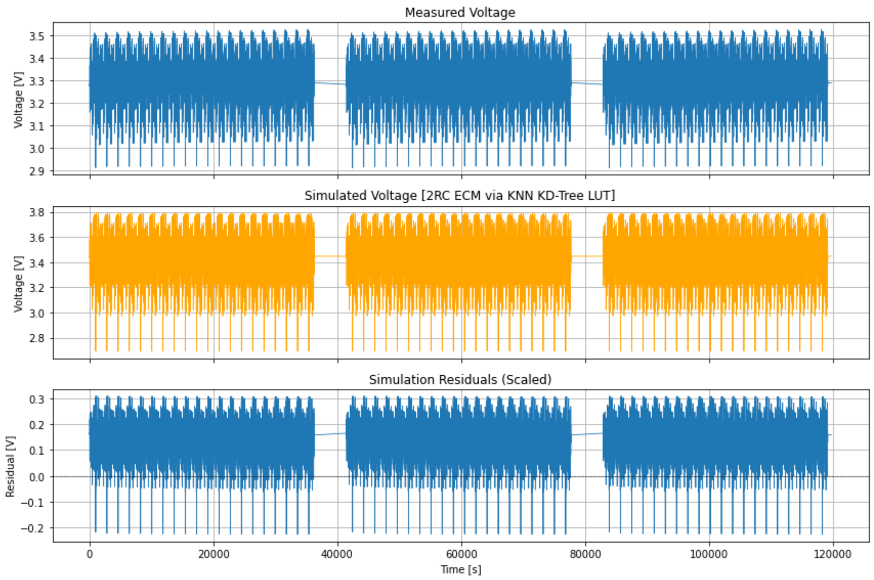

Figure 1 – Simulation result for ecm_lookup_table_refined_with_wltp.csv by using 10-25-19_11.29 960_WLTP206b.mat

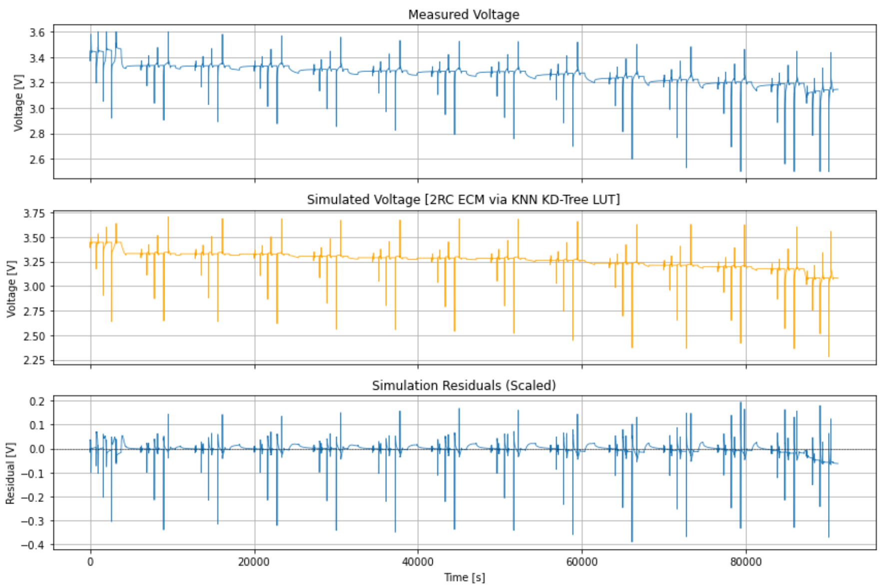

Figure 2 – Simulation result for ecm_lookup_table_refined_with_wltp.csv by using 11-17-19_17.23 1030_HPPC.mat

updates. The resulting R0 parameters are saved as ecm_lookup_table_r0_corrected.csv, along with the previously available RC parameters. The corresponding voltage error plots are shown in Figure 3 and 4. The comparison between the result of base ECM parameter set and R0 corrected one shows clear reduction of voltage estimation errors.

3) ECM RC Correction (EcmRCCorrn.py)

After improving the R0 parameters in Section 2, I wanted to try further optimizing the latest ECM parameter set. Considering that the dimensionality of the problem is decreased after fixing the R0 parameters, it was worth giving it a try to optimize RC parameters without touching R0. This actually makes sense because the same Weighted Ridge Regression method can be used for this purpose after making some adjustments. The concept, weight, is used in this algorithm to construct the matrices and scaling variables to be used in the optimization solution. Constructing

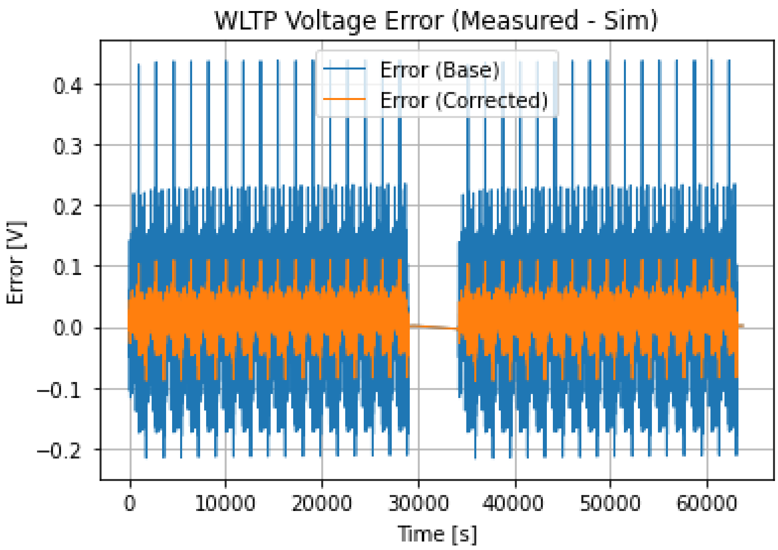

Figure 3 – Voltage error for ecm_lookup_table_r0_corrected.csv *by using 10-24-19_16.28 960_WLTP206a.mat*

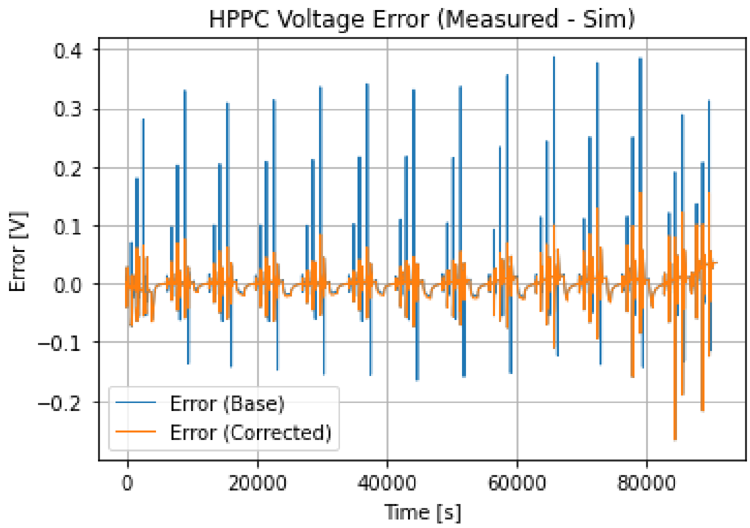

Figure 4 – Voltage error for ecm_lookup_table_r0_corrected.csv *by using 10-16-19_20.16 948_HPPC.mat*

these weights not based on $|Current|$ but on $|\Delta Current|$ is an important trick because ohmic error presents itself during high currents, whereas RC dynamics caused by transients. Thus, the edges that cause transient dynamics are captured better. After implementing the same algorithm by changes like this, the resulting final ECM lookup table is saved as ecm_lookup_table_rc_corrected.csv. The resulting voltage error plots are shown in Figure 5 and 6. The results seem improved based on the previous parameter set, especially for the WLTP cycle data.

4) Simulation and Validation of Extracted ECM Parameters (EcmSimAndValidatn.py)

The resulting ECM parameter lookup table is validated by using HPPC and WLTP data (different data from the previous sections). The equation below is used to simulate the ECM.

$$V[k] = OCV(SOC[k]) - I[k]R_0 - V[k-1]e^{-\frac{dt}{R_1C_1}} - R_1(1 - e^{-\frac{dt}{R_1C_1}})I[k] - V[k-1]e^{-\frac{dt}{R_2C_2}} - R_2(1 - e^{-\frac{dt}{R_2C_2}})I[k] \quad (2)$$

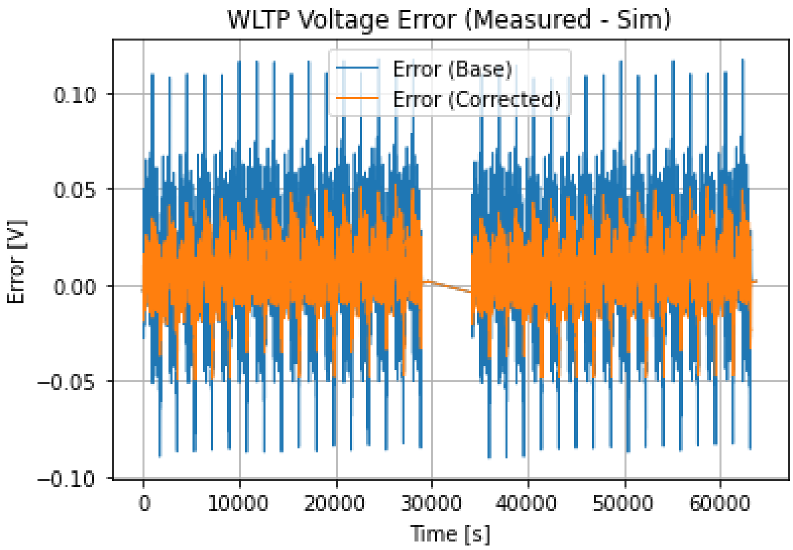

Figure 5 – Voltage error for ecm_lookup_table_rc_corrected.csv *by using 10-24-19_16.28 960_WLTP206a.mat*

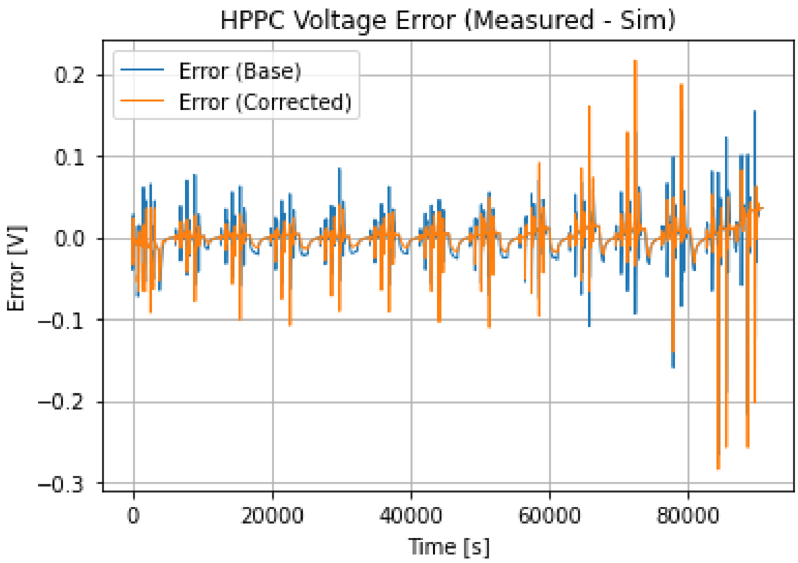

Figure 6 – Voltage error for ecm_lookup_table_rc_corrected.csv *by using 10-16-19_20.16 948_HPPC.mat*

The simulation results are shown in Figure 7 and 8. The used data are WLTP data (10-25-19_11.29 960_WLTP206b.mat) from Cycle 0 and another HPPC data (11-17-19_17.23 1030_HPPC.mat) from Cycle 900. Root Mean Squared Error, Mean Absolute Error and Mean Error values attached below the figures. For a high performance ECM, MAE should be < 10 mV. The Mean Absolute Error for the overall solution is around 12 and 14 mV for WLTP and HPPC data respectively, which yields good, but not a perfect ECM parameter fitting.

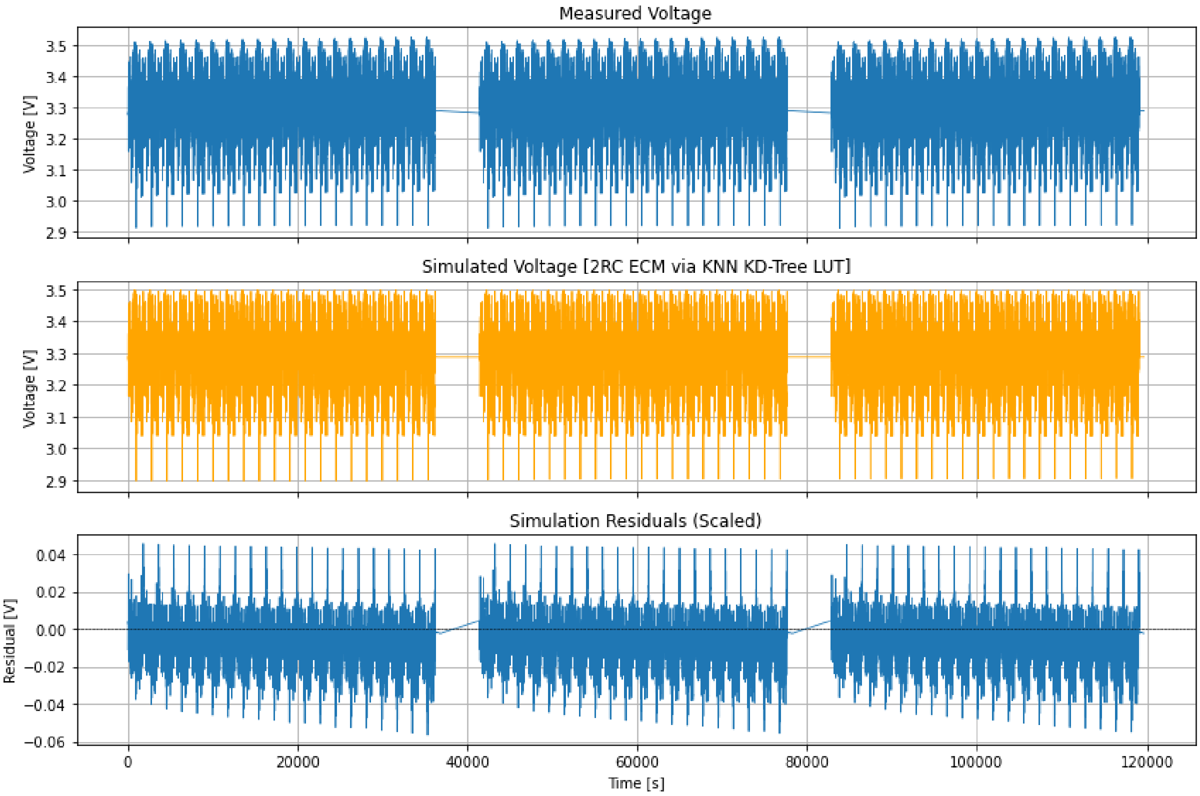

Figure 7 – Simulation result for ecm_lookup_table_rc_corrected.csv *by using 10-25-19_11.29 960_WLTP206b.mat*
RMSE = 0.0148 V, MAE= 0.0123 V, ME = -0.0094 V

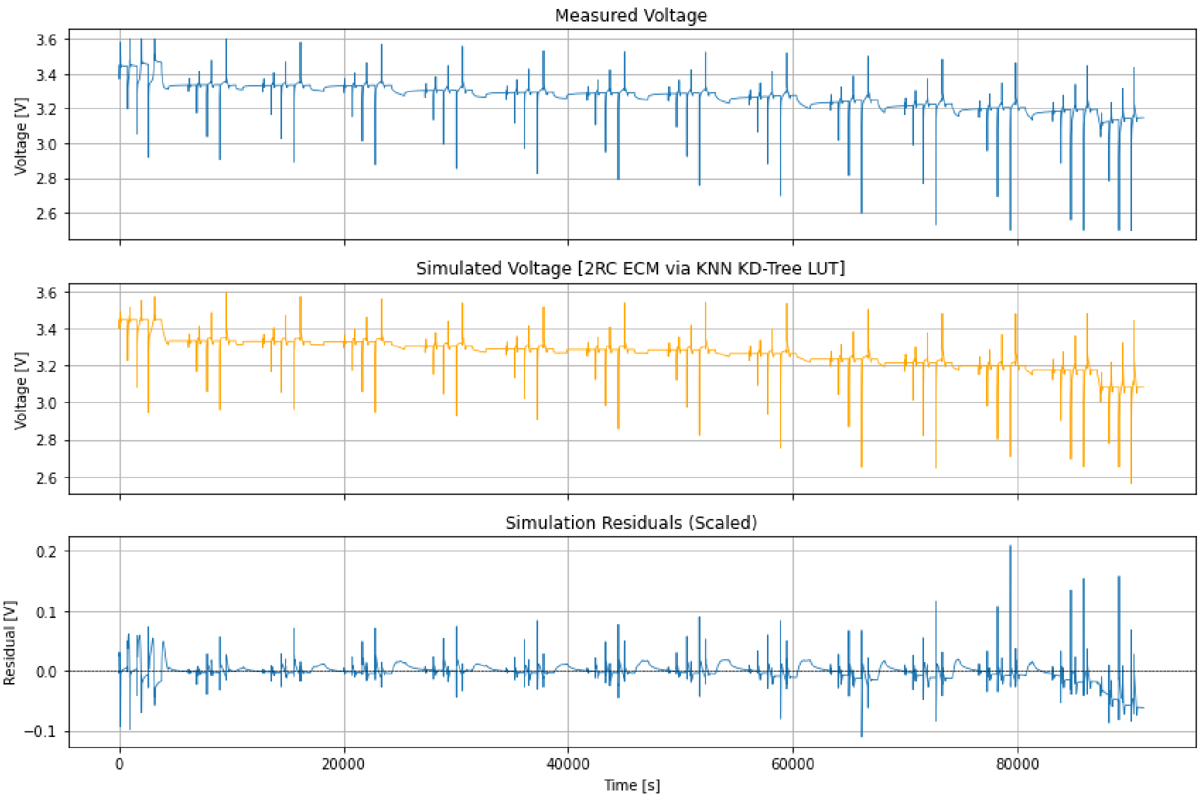

Figure 8 – Simulation result for ecm_lookup_table_rc_corrected.csv *by using 11-17-19_17.23 1030_HPPC.mat*
RMSE = 0.0234 V, MAE= 0.0139 V, ME = -0.0036 V

5) EKF Based SOC Estimator (EkfSocEstimator.py)

An EKF based SOC estimator is built by using Coulomb counting in state prediction step and by using the resulting ECM lookup table ecm_lookup_table_rc_corrected.csv in the measurement update state. The OCV table is again constructed by using the available data from 1-A123 Aging Tests_Data Summary.xlsx. I actually tried to build a new OCV table from scratch but the result did not look very different than the available one, so I thought it won't worth using it. I continued to use the previously available OCV table.

The UDDS file is loaded and EKF is simulated. The result for the first trial can be seen in Figure 9.

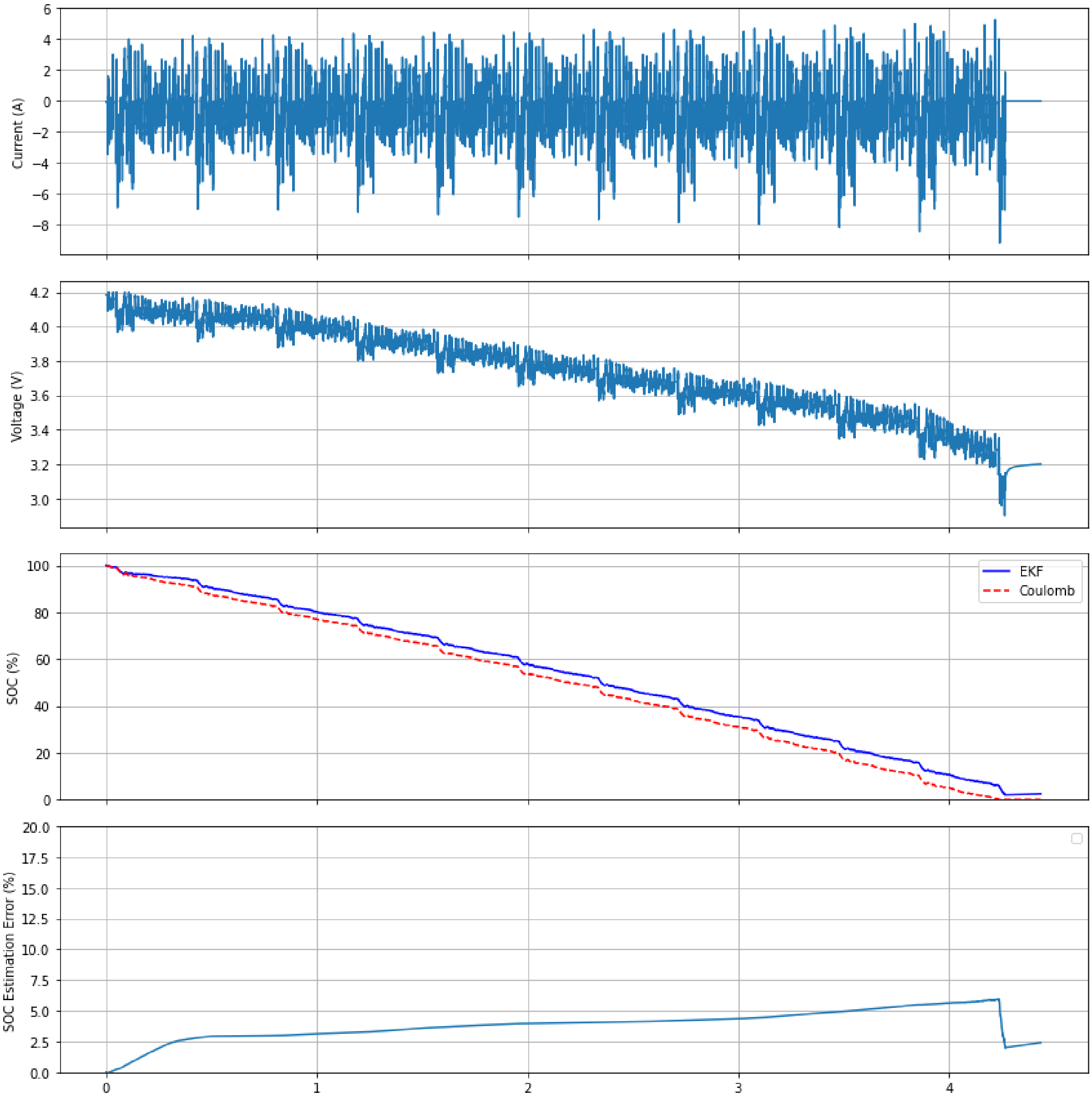

Figure 9 – Simulation result for EKF by using ecm_lookup_table_rc_corrected.csv *and UDDS_25degC.*

It can be observed that the voltage actually starts from 4.2 V, which is way above our OCV table. I was expecting a voltage around 3.8-3.9 V at most (even for an aged cell), so the voltage gap between the UDDS test data and our current OCV table and ECM parameter lookup table is huge. Then we can tweak the measurement noise parameter (R) of the EKF. Moreover, we can make this parameter a variable, which changes during runtime. Since we do not trust our OCV table and ECM parameters when the voltage is higher than 3.9 V, we can set R to a very high number when voltage is greater than 0. Thus, we would be trusting the voltage measurements more than we trust our ECM. When the voltage is relaxed below 3.9, we can set R to a small number to make the EKF trust ECM parameters more. By using this trick, the final result for the EKF is shown in Figure 10.

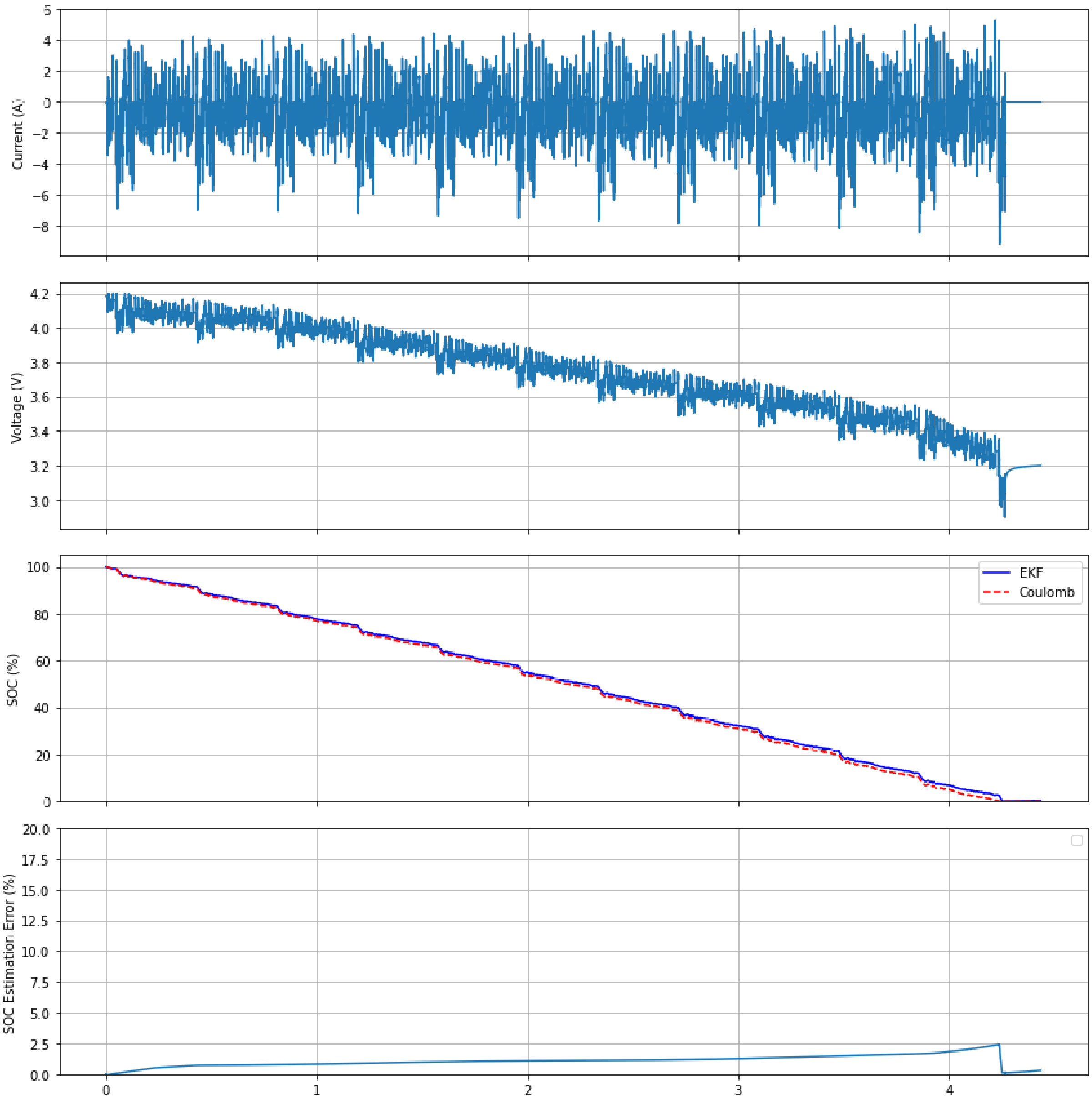

Figure 10 – Simulation result for EKF by using ecm_lookup_table_rc_corrected.csv *and UDDS_25degC, after tweaking R when voltage is higher than 3.9 V.*

In the results, it can be clearly seen that the voltage error that was introduced at the beginning, when the voltage is above 3.9, dropped significantly. After the voltage relaxed below 3.9, our ECM started to behave more actively and we do not see huge errors. However, the ECM parameters are not perfect and it causes a drift towards a 2.5% SOC error at the end. Considering that we operate in the flat region, it is somewhat normal to experience an error drift. The ECM model can be improved by using better test data which includes different low C-rates that sweep through complete SOC range. This data can be helpful to extract much more accurate OCV table. Additionally, hysteresis effect is neglected in this study due to time constraints (I tried to fit some hysteresis models, but could not get better results). By obtaining a specific hysteresis minor loop testing data, which includes low current(C/30) pulses whose directions symmetrically decrease to sweep less SOC over time can also be useful to fit an accurate hysteresis model. By implementing these additional steps, the performance can be improved more.