

# Battery Predictions for Electronic Prognostics

Imre Nagi

Carnegie Mellon University  
imre.nagi@sv.cmu.edu

Ali Yousafzai

Carnegie Mellon University  
ali.yousafzai@sv.cmu.edu

Darren Yin

Carnegie Mellon University  
darren.yin@sv.cmu.edu

Dimitrios Tzannetos

Carnegie Mellon University  
dimitrios.tzannetos@sv.cmu.edu

## ABSTRACT

With the growth of electrical components used in vehicles such as aircrafts, cars, and satellites and the critical role these systems control such as self-driving, navigation, and sensors, it is important to be able to understand that the lifetime of electrical components especially batteries for improved prognostics. In this paper, we investigate and evaluate the performance of two approaches: Gaussian Process Regression (GPR) and the Physical State Model of the battery to predict the Remaining Useful Lifetime (RUL) of Lithium-ion batteries. By using GPR we were able to create an accurate model with RMSE of 0.1058, facing, however, certain limitations which are described in the paper. By modeling the physical battery, we were able to accurately predict the State of Charge (SOC) and capacity of the battery, which can be used to determine the health of the battery. An alarm system can further be built on this model.

## KEYWORDS

Kalman Filtering, Gaussian Process Regression, State of Charge, Remaining battery capacity

## 1 INTRODUCTION

A large amount of electronic components exist in aircrafts, satellites, and cars, many of which perform increasingly critical roles autonomously. This trend of autonomous vehicles will continue to grow for the foreseeable future, such as the growth of self-driving cars and UAVs many of which utilize all electric powertrains.

Understanding the behavior of components over its lifetime must be developed to monitor the health of these components. Of particular note to prognostics is the capability to anticipate failures and predict the remaining life of these components. This allows maintenance and part replacement to be planned optimally, the fixing of components before they fail, and allow aircrafts to optimally plan missions.

Due to the safety management, estimating the remaining useful life (RUL) of Lithium-ion batteries has become a challenging problem in the fields of reliability, automatic test, power sources, and electric vehicles, etc.

This problem can be broken into two parts:

- (1) building a model that can accurately estimate the future state of the electrical component
- (2) creating an alarm system that triggers before said component is estimated to fail.

The accuracy of the models estimates directly impact the effectiveness of the alarm system.

Prior research has been performed in this field where a system was presumed to be a linear dynamic system specified in discrete time. A dynamic Bayes network (DBN) was used to model the linear dynamic system (LDS) to predict future output of a selected feature of the component. An optimal alarm system was then constructed where the alarm would trigger if the conditional probability of a critical event was above a certain threshold probability.

This paper focuses entirely on two different machine learning approaches, created in MATLAB, to model a specific electrical component, a 18650 Lithium-ion battery, using a large dataset of experimental measurements recorded by NASA.

The rest of the document is organized as follows: Section 2 presents recent related work and studies on Predictive Models for Lithium-ion batteries. The problem description and the addressed challenges are explained in Section 3. Our approach is presented in Section 4. Subsequently, the evaluation results are discussed in Section 5. We conclude with a view on future work in Section 6.

## 2 RELATED WORK

At present, the various approaches of battery State of Charge (SOC) estimation and RUL prediction can be generally classified into two categories [8]: statistical data-driven and model based [17], [15]. There are lots of research work focusing on performance degradation, SOC assessment, RUL estimation for the Lithium-ion battery [12], [5], [18]. Especially for the Lithium-ion battery prognostics, the prediction uncertainty, and the applicability of the model-based and data-driven methods have always been challenging problems in this area.

Lots of researchers such as Bhaskar Saha and Kai Goebel and others researchers in the Prognostics Center of Excellence (PCoE) of the NASA AMES Center achieved the battery RUL prediction as well as the uncertainty representation and management with particle filter (PF) algorithm [12], [5]. Moreover, the Artificial Neural Networks (ANN) [9], Extended Kalman Filter (EKF) [4], Support Vector Machine (SVM) [14], Relevance Vector Machine (RVM) [13] and other machine learning and statistical algorithms [11], [2] are applied in Lithium-ion battery prognostics. Finally, lots of physical models, chemistry model and other related empirical model are developed or applied in the battery RUL estimation [3].

All of the RUL estimation frameworks have become an effective and practical approach for Lithium-ion battery degradation analysis and estimation. However, to obtain the model or to identify the model parameters is very difficult in most cases due to complicated operating conditions. The main drawback is that these modeling approaches do not consider the varied operation condition for an

on-line application, a fact that leads to sub-optimal results. On the other hand, most of the data-driven prognostics algorithms require high computing complexity even in the case of simple application, calling for more time-efficient techniques.

### 3 METHODOLOGY

#### 3.1 Dataset

To evaluate the performance of two approaches analyzed in the following parts, we used a dataset provided by NASA<sup>1</sup> [1]. The dataset used was derived from a NASA experiment where a set of four 18650 Lithium-ion batteries, labeled from RW9 to RW12, were continuously operated using a sequence of charging and discharging currents between -4.5A and 4.5. This type of charging and discharging operation is referred to in the experiment as a random walk (RW) operation. Each of the loading periods last 5 minutes or whenever the voltage goes outside the range of 4.2V to 3.2V, and after 1500 periods (about 5 days) a series of reference charging and discharging cycles were performed in order to provide reference benchmarks for battery state health.

The dataset includes the relative time the measurements were taken from the start of the experiment along with the temperature, current, and voltage of the battery. Figures 1 and 2 show the measurements for voltage, current and temperature for the first and last 50 random walk cycles performed.

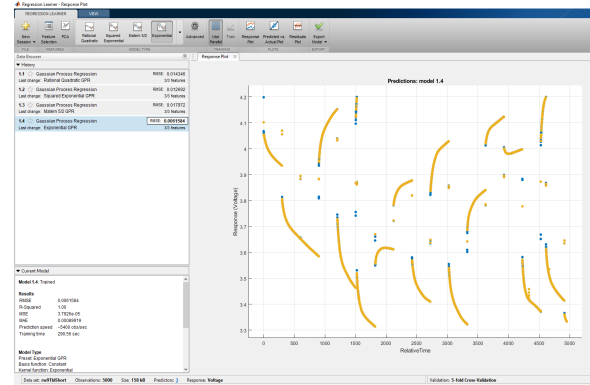
Note that due to random nature of charging and discharging cycles performed during this experiment, the stable state preconditions required by the approach outlined in [10] are violated and hence, non-linear methods should be adopted instead.

**3.1.1 Infrastructure.** We used a machine with an Intel Core™ i5-2557M Processor, 1.7 GHz, 4GB of RAM memory and MATLAB, version R2017b.

#### 3.2 Gaussian Process Regression

Gaussian process regression (GPR) is a statistical model where observations are continuous, such as time [7]. The model is a powerful non-linear multivariate interpolation technique used in fields as far ranging as mining, remote sensing, and real estate appraisal. In machine learning, GPR models are nonparametric kernel-based probabilistic models that utilize lazy learning and a kernel function, which uses the similarity between points, to predict an unseen point in training data, along with the uncertainty of the prediction.

Observed values generally do not match the exact function values, but a noisy realization of them. GPR models, which are probabilistic, are able to compute the prediction intervals using these observed values and predict all values between the observed values. This means that, the model becomes more accurate as more observed values are given to the model as training data as it is easier to find the function that fits the data. However, since GPR models are created through lazy learning, more data dramatically increases the time to train a model. As an addendum, GPR model



**Figure 3: MATLAB's Regression Learner used to find best fit GPR Model**

predictions have drastically increased error when predicting values outside of the time range that the training data is in, as the prediction interval bounds drastically increase due to the lack of observed values for future time. Fortunately, in our case, our dataset includes observed data of the entire lifetime of the battery and so prediction intervals are expected to remain consistent.

In this paper, voltage was selected as the output of the GPR model, with time, temperature, and current as inputs. Given the sheer size of data for a single battery, over 8 million data rows, and the limitation of MATLAB, roughly 30 thousand data rows were randomly chosen over the entire RW9 dataset to be used as training data. Multiple GPR models were trained using different kernel functions: rational quadratic, squared exponential, matern 5/2, and exponential. Each model used 5-fold cross validation. The best model was selected based on the lowest root mean square error (RMSE), the model that best fit the training data. RW10 data was used as a test set for the trained model. The models were created using MATLAB's Statistics and Machine Learning Toolbox™, as shown in Figure 3.

#### 3.3 Battery Modeling Using Battery Equivalent Circuit

In this approach, we can predict the state of the battery by modeling its physical characteristics. There are two important requirements that must be provided before the battery internal parameters are estimated. First, we need the battery equivalent circuit. This equivalent circuit later will be used to construct MATLAB's Simscape<sup>2</sup> and Simulink<sup>3</sup> model. The second requirement is the real data from experiments performed on the battery. In this case, the dataset will be discharging data from RW9 dataset provided by NASA. Even though the dataset used in GPR experiment is still applied, in this approach, the total random walk cycles are not required for the parameter estimation. To construct the battery model and estimates its internal parameters, we only need the very first discharging cycle of the battery in the initial experiment since we only need the discharging data where the battery condition is

<sup>1</sup><https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>

<sup>2</sup><https://www.mathworks.com/products/simscape.html>

<sup>3</sup><https://www.mathworks.com/products/simulink.html>

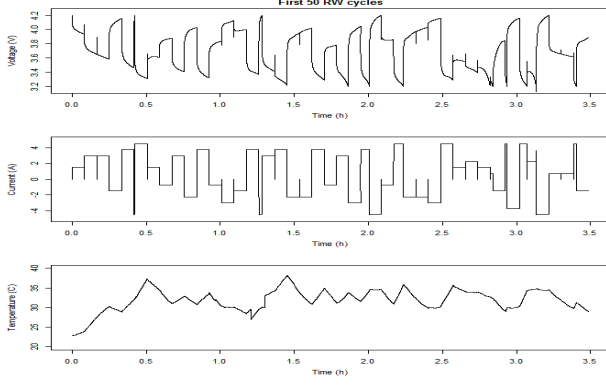


Figure 1: First 50 RW cycles of RW9

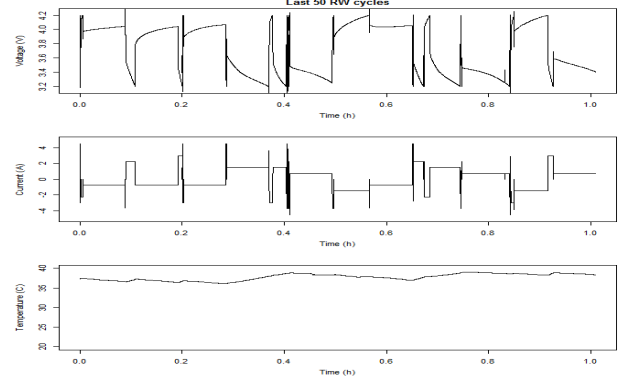


Figure 2: Last 50 RW cycles of RW10

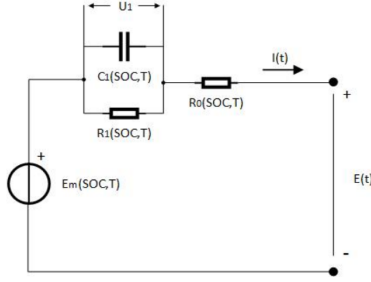


Figure 4: Equivalent circuit of battery using a first RC circuit

still good and optimal so that we can create a proper model for our experiment. By using these two requirements, parameter estimation will be performed in the Simulink Design Optimization tool provided by MATLAB. The goal of this estimation is to estimate the internal parameter values of the battery, e.g.  $R_0$ ,  $R_1$ ,  $C_1$ , and  $E_m$ .

**3.3.1 Battery Model.** In this project, we chose a first order RC circuit as the equivalent circuit of the battery. This circuit consists of  $E_m$ ,  $R_1$ ,  $R_0$  and  $C_1$ , where  $E_m$  is the electromotive force from the voltage source,  $R_0$  and  $R_1$  is the resistor, and  $C_1$  is the battery capacitance. This experiment could be conducted by using more than one RC component to obtain more accurate battery parameter results, however, since the focus of the paper is on the machine learning aspect, we refrained from additional complexity in the battery model.

In the model shown in Figure 4,  $R_0$ ,  $R_1$ ,  $C_1$ , and  $E_m$  are dependent on State of Charge (SOC) and battery temperature (T). Thus, these parameters will be 2D look-up tables that accept SOC and temperature as the matrix index.

The dependency between each battery component with SOC and temperature should be reflected in the Simscape model. In the Figure 5, each components will accept SOC and temperature as the input. Each component on this model is created by using Simscape model language that has been implemented by MATLAB. In addition, to make the Simscape model as close to the real battery

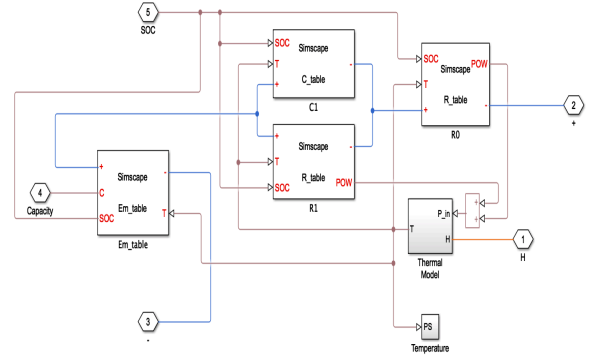


Figure 5: Battery Model in Simscape

model, a thermal model has also been implemented. The thermal model accepts ambient temperature and the total heat or power generated by all resistors on this model. By implementing the thermal model, the experiment becomes more flexible and closer to the real conditions of the battery.

**3.3.2 Battery Parameter Estimation.** The goal of battery parameter estimation is to estimate the value of  $R_0$ ,  $R_1$ ,  $C_1$ , and  $E_m$  in order to minimize the error between the experiment data and simulated data. In Figure 6, the battery voltage of discharging cycle with a sequence of discharge current is given as the result of the experiment. With using known experiment data such as current and battery voltage, this parameter estimation technique will try to train the value of battery parameters so that the predicted voltage can mimic the real experiment data.

As you can see on the Simulink Diagram in Figure 7, we are going to feed current as the input of this simulation by using the current used in the real experiment. In addition, we also try to feed ambient temperature as another input for the battery to simulate its internal thermal condition. Later on, the voltage measurement of the battery is conducted so that error between the simulation and the real dataset can be calculated.

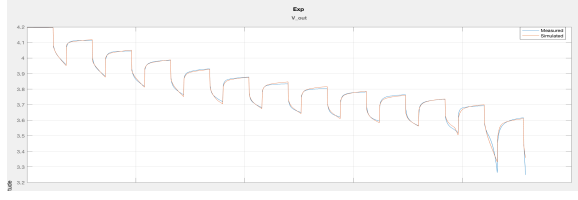


Figure 6: Discharging Voltage of RW9

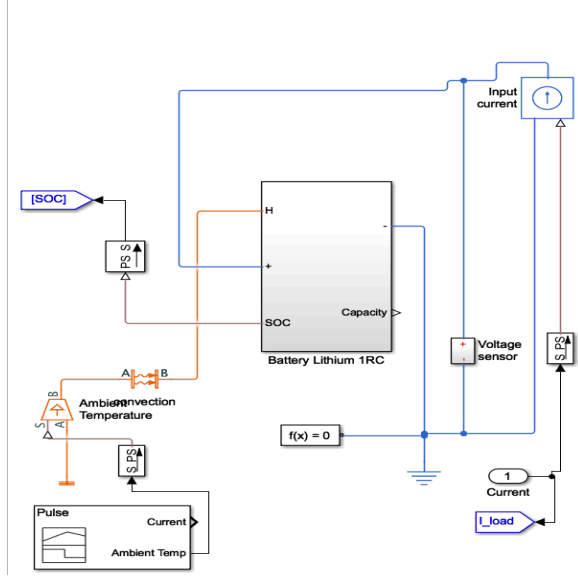


Figure 7: Simulink diagram for parameter estimation

**3.3.3 Kalman Filter and Battery Capacity Estimation.** Kalman Filtering is a common algorithm used in many applications including guidance and navigation of vehicles. It takes into account measures over time that might include noise and estimates future values based on its previous values.

To determine the remaining RUL of a battery, we use SOC and capacity of the battery. We followed the example<sup>4</sup> provided in [6] with the help of MATLAB, Simulink and Simscape. The capacity of the battery degrades with every discharge-charge cycle, giving an inaccurate SOC estimation. Therefore, we can use an event-based linear Kalman filter to estimate the battery capacity when the battery transitions between charging and discharging. Consequently, we can use the estimated capacity to indicate the health condition of the battery (SOC).

*Estimating SOC.* From the previous Section on the battery model, we can derive our state transition equation as below:

$$\frac{d}{dt} \begin{pmatrix} SOC \\ U_1 \end{pmatrix} = \begin{pmatrix} 0 \\ -\frac{1}{R_1(SOC, T_b) * C_1(SOC, T_b)} U_1 \end{pmatrix} + \begin{pmatrix} -\frac{1}{3600 * C_q} \\ \frac{1}{C_1(SOC, T_b)} \end{pmatrix} I + W$$

<sup>4</sup><https://www.mathworks.com/help/control/examples/nonlinear-state-estimation-of-a-degrading-battery-system.html>

where  $V_1$  is the voltage across capacitor  $C_1$ ,  $I$  is input current,  $T_b$  is the battery temperature,  $C_q$  is the battery capacity and  $W$  is the process noise.

We will use Kalman Filtering to estimate the SOC of the battery with the help of Simulink functions. Since SOC is non-linear, we will use an Unscented Kalman Filter, details of which can be found in [6] and [16].

Applying Euler discretization of the state transition equation will give us:

$$\begin{pmatrix} SOC_{T+1} \\ U_{1T+1} \end{pmatrix} = \begin{pmatrix} -\frac{1}{3600 * C_q} I \\ -\frac{1}{R_1(SOC, T_b) * C_1(SOC, T_b)} U_1 + \frac{1}{C_1(SOC, T_b)} I \end{pmatrix} T_s + \begin{pmatrix} SOC_T \\ U_{1T} \end{pmatrix} + W_T$$

where  $T_s$  is the sampling time. The Kalman filter can be configured in Simulink, details of which can be found in [6].

*Estimating Battery Capacity.* The decreasing capacity  $C_q$  of the battery models the battery degradation. The battery is configured to charge when its capacity is at 30% and discharge when it is at 90%. Thus the measurement equations can be derived as:

$$\begin{aligned} C_{qk}^{measured} &= C_{qk} + V_{C_q} = \frac{\int_{t_{k-1}}^{t_k} Idt}{(\Delta SOC)_{nominal}} \\ &= \frac{\int_{t_{k-1}}^{t_k} Idt}{|0.9 - 0.3|} = \frac{\int_{t_{k-1}}^{t_k} Idt}{0.6} \end{aligned} \quad (1)$$

where  $I$  is the input current.

The state transition and measurement equations are shown below:

- State Transition:

$$C_{qk+1} = C_{qk} + W_{C_q}$$

- Measurement Equation:

$$C_{qk}^{measured} = C_{C_q} C_{qk} + V_{C_q}$$

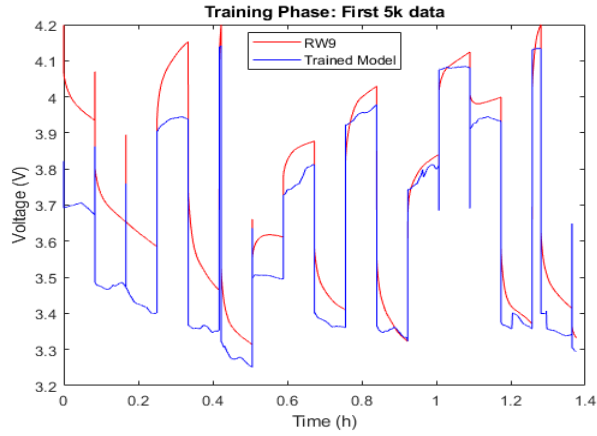
where  $W_{C_q}$  is the process noise,  $V_{C_q}$  is the measurement noise,  $A_{C_q}$  and  $C_{C_q}$  are constants that are set to 1.

## 4 RESULTS

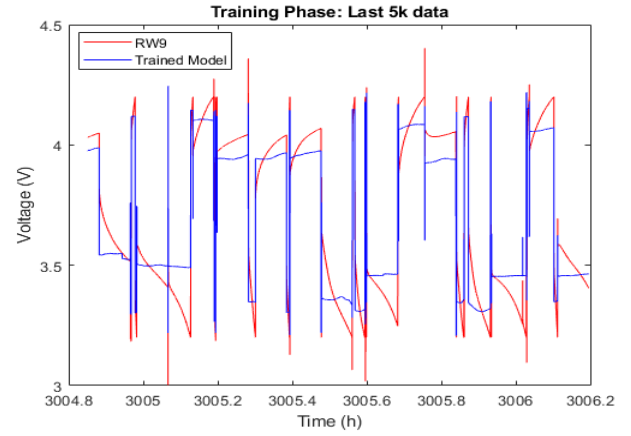
### 4.1 Gaussian Process Regression

The best GPR model created has a RMSE of 0.1058, where voltage ranges from 4.2V to 3.2V, and uses an exponential kernel function. The trained model performs well on both the train and test set, with almost the entire training (RW9) and testing (RW10) dataset falling inside the 95% confidence bounds of the trained GPR models predictions. An interesting characteristic of the model to note is that the predicted peaks are lower and the dips are higher than the actual values as shown in Figures 8 to 13.

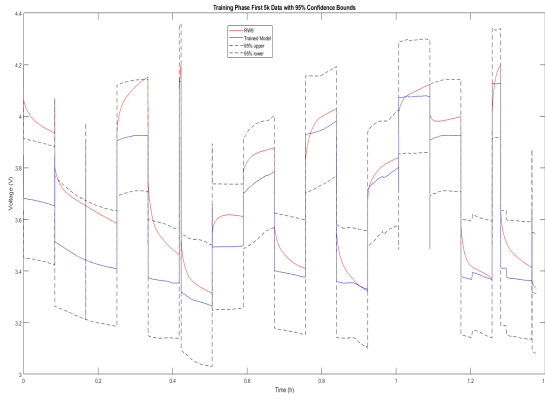
The error in the trained models prediction, measured as the absolute difference in voltage as shown in Figures 16 to 19, against both the training and test data does not go down in time, rather it remains constant throughout the life of the battery. However, the average error is quite low, only 0.0772V against the train set (RW9) and 0.0777V against the test set (RW10). The similar error between



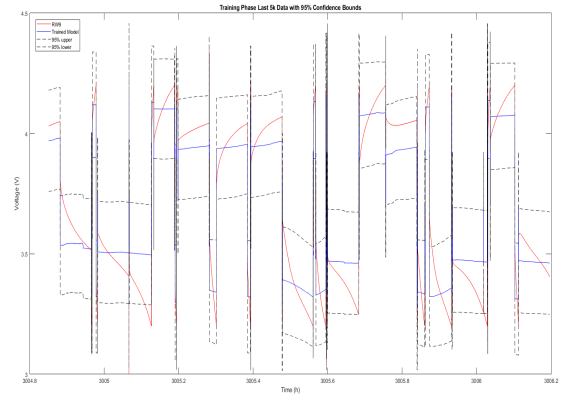
**Figure 8: GPR Model: First 5K data of Model vs. Training Set (RW9)**



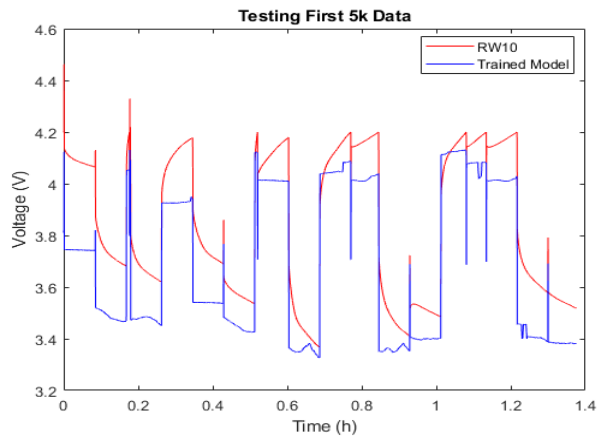
**Figure 9: GPR Model: Last 5k data of Model vs. Training Set (RW9)**



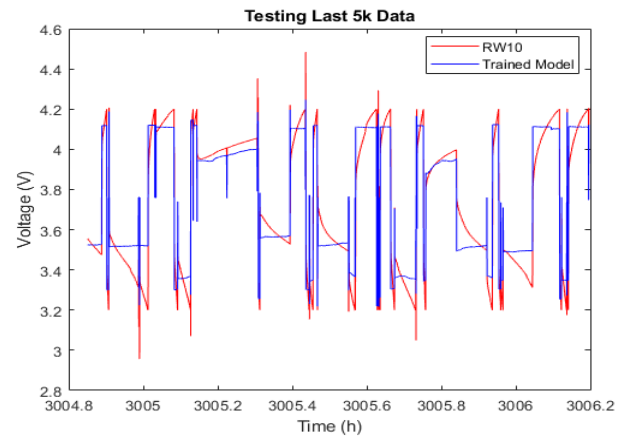
**Figure 10: GPR Model: First 5K data of Model vs. Training Set (RW9) with 95% conf bounds**



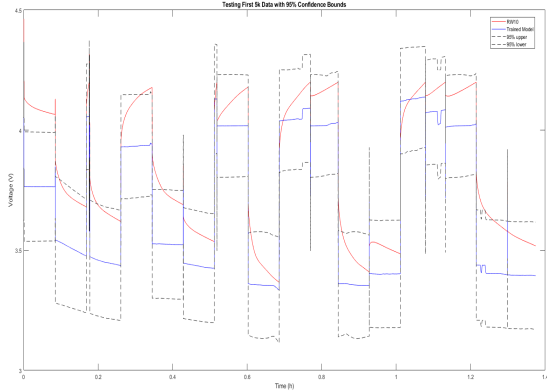
**Figure 11: GPR Model: Last 5k data of Model vs. Training Set (RW9) with 95% conf bounds**



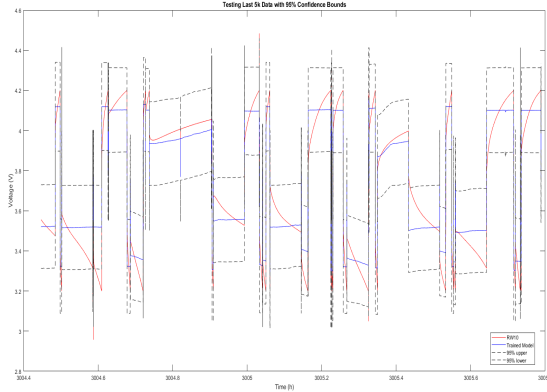
**Figure 12: GPR Model: First 5K data of Model vs. Test Set (RW10)**



**Figure 13: GPR Model: Last 5K data of Model vs. Test Set (RW10)**



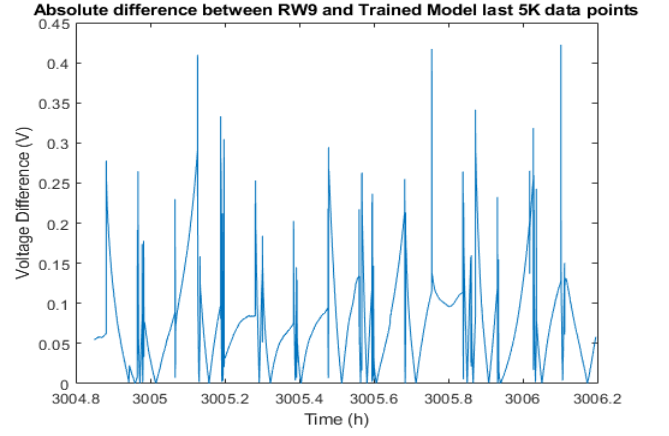
**Figure 14: GPR Model: First 5K data of Model vs. Training Set (RW10) with 95% conf bounds**



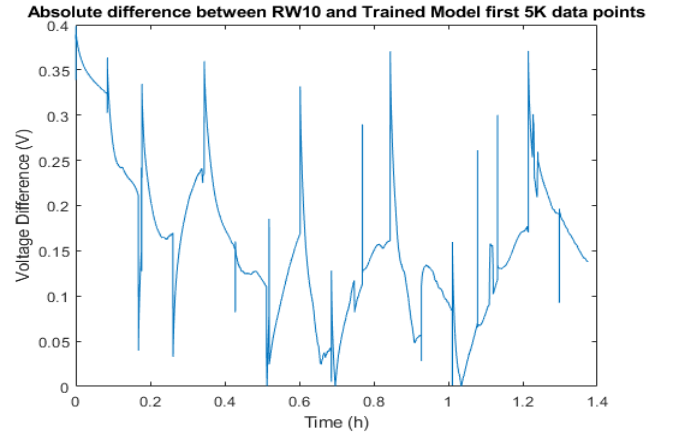
**Figure 15: GPR Model: Last 5K data of Model vs. Training Set (RW10) with 95% conf bounds**



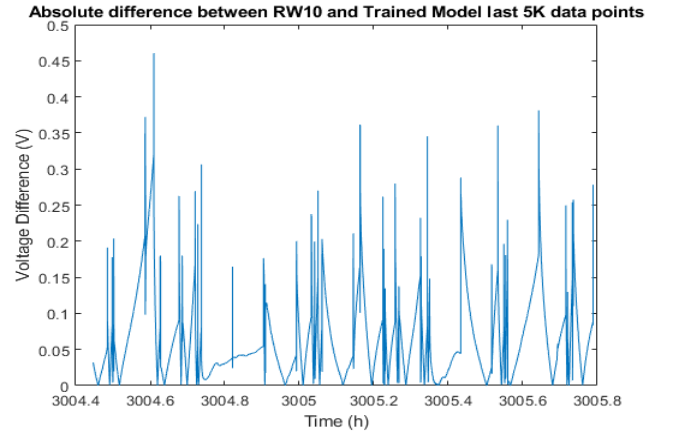
**Figure 16: GPR Model: Absolute difference between RW9 and GPR model for first 5K data points**



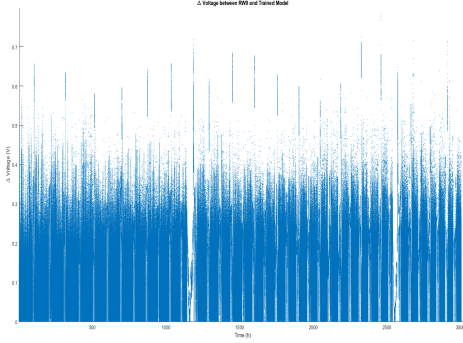
**Figure 17: GPR Model: Absolute difference between RW9 and GPR model for last 5K data points**



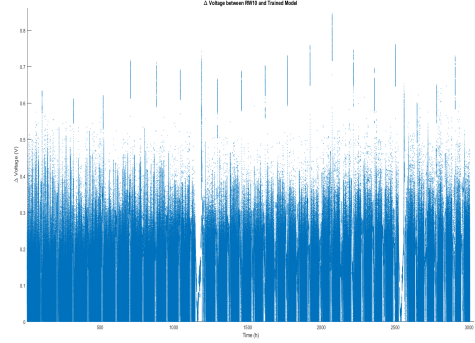
**Figure 18: GPR Model: Absolute difference between RW10 and GPR model for first 5K data points**



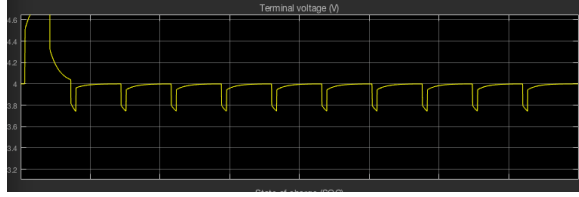
**Figure 19: GPR Model: Absolute difference between RW10 and GPR model for last 5K data points**



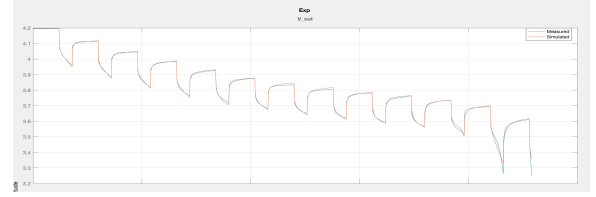
**Figure 20: GPR Model: Scatter Plot of Absolute difference between RW9 and GPR model**



**Figure 21: GPR Model: Scatter Plot of Absolute difference between RW10 and GPR model**



**Figure 22: Simulated data of battery Voltage using initial parameters value**



**Figure 23: Simulated and Measured Voltage of the experiments using trained parameters value**

the training and testing set is a good implication that the trained model is not overfitted.

Despite the ability for the GPR model to predict the state of the battery with relatively high predictive power, there are issues with how the model predicts future data that makes it impractical for prognostics. Due to the fact that temperature and current are used as inputs along with time, if the model is at time  $t$  and needs to predict voltage for  $t + 1$ , the model requires the current and temperature at  $t + 1$  to be able to predict voltage at  $t + 1$ . This is obviously impractical to use for an alarm system as the future current and temperature is unknown to the model until it is measured at  $t + 1$ , but at that point there is no value added to predict voltage for  $t + 1$ , as it has already occurred and can be measured.

## 4.2 Battery Modeling Using Battery Equivalent Circuit

**4.2.1 Battery Parameter Estimation.** The simulated battery voltage of the battery with the initial parameter values is shown in Figure 22. These initial values are manually chosen so that the transient state of the simulated data is close to the transient state of the real experiment. This setup is necessary in order to optimize the training efficiency. If the incorrect initial values are chosen, the parameter training could take more than 50 iterations to converge. Otherwise, it would take less than 10 iterations to reach at a converging point.

Figure 23 shows the simulation result after we successfully estimated the values of battery internal components. As shown on the image below, the measured and simulated lines are almost overlapped because the error of the simulation is quite small.

As shown in Figure 24, the value of  $C_1$  is changing over the iteration during the parameter training until it converges. In addition, when the iteration is completed, we obtain the following results as the parameter values:

$$E_m = \begin{bmatrix} 4 & 3.5917 & 3.1752 \\ 4 & 3.7324 & 3.5411 \\ 4 & 3.7794 & 3.6231 \\ 4 & 3.8961 & 3.7661 \\ 4 & 4.1431 & 3.9838 \\ 4 & 4.2894 & 4.0639 \\ 4 & 4.3648 & 4 \end{bmatrix} \quad C_1 = \begin{bmatrix} 1000 & 9385 & 10308 \\ 1000 & 10139 & 10224 \\ 1000 & 12506 & 11170 \\ 1000 & 11854 & 10517 \\ 1000 & 12403 & 96195 \\ 1000 & 10945 & 94656 \\ 1000 & 10623 & 10000 \end{bmatrix}$$

$$R_1 = \begin{bmatrix} 0.07 & 0.0769 & 0.0712 \\ 0.07 & 0.0655 & 0.0641 \\ 0.07 & 0.0562 & 0.0626 \\ 0.07 & 0.0654 & 0.05697 \\ 0.07 & 0.0753 & 0.0738 \\ 0.07 & 0.0723 & 0.0732 \\ 0.07 & 0.0700 & 0.07 \end{bmatrix} \quad R_0 = \begin{bmatrix} 0.1 & 0.1297 & 0.1091 \\ 0.1 & 0.1024 & 0.0940 \\ 0.1 & 0.1035 & 0.0998 \\ 0.1 & 0.1039 & 0.1043 \\ 0.1 & 0.1087 & 0.0977 \\ 0.1 & 0.1006 & 0.1011 \\ 0.1 & 0.100 & 0.1 \end{bmatrix}$$

Later on, these parameter values of the battery will be used to estimate the SOC of the battery and the remaining useful capacity of the battery.



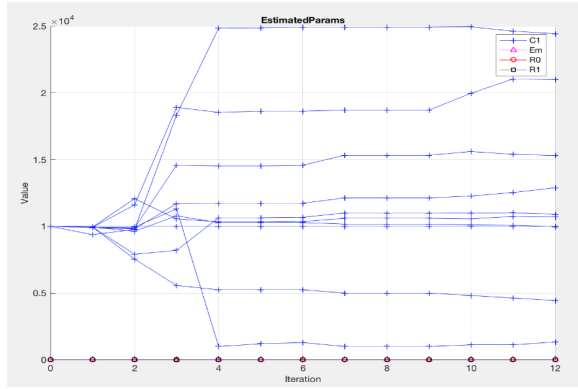


Figure 24: Changes of parameters during training per iterations

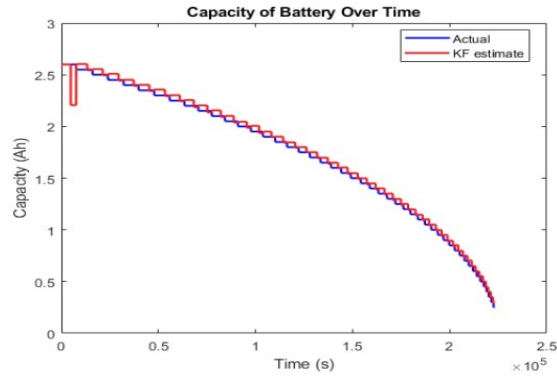


Figure 25: Actual and Estimated Capacity of Battery over time

**4.2.2 Kalman Filter and Battery Capacity Estimation.** Performing Kalman Filtering on SOC and capacity produced the results shown in Figures 25 and 26. It is important to observe that the capacity of the battery decreases as the battery runs out of charge with time. It is also important to note that the actual capacity and estimated capacity are similar. There is a slight lag between the actual and estimated values, which is due to how the capacity is calculated, in that, we require the previous value of the capacity to estimate the next. In Figure 26, it can be seen that as the battery discharges, the frequency of the SOC pattern increases. This means that it takes less time for the battery to charge and discharge, which supports our results from the Figure 25. Note that because of this decrease in charging and discharging time, our estimates for SOC begin showing increased error as the sample rate is greater than the reduced charging/discharge time.

## 5 CONCLUSION

In conclusion, both the GPR and State Space Modeling of the battery can be used to predict the RUL of the battery. However there are tradeoffs between these methods.

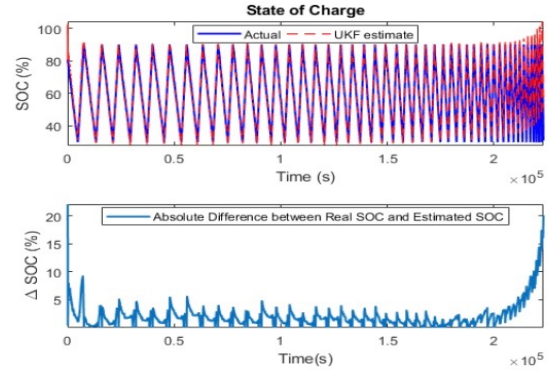


Figure 26: Actual and Estimated SOC with time

The GPR predicts the voltage battery with a relatively small error. However, this method shows serious time and space complexity limitations as it is slow to train (approx. 1 hour) and can accept only a limited training set (less than 1% of available data points). It also requires the inputs of the next time step to predict the output, and thus a usable alarm cannot easily be built. The State Space Model of the Physical battery predicts the state of the battery through simulation. It is faster to train (approx. 20 mins). It is shown to produce, with high accuracy, estimates of SOC and capacity. An alarm system can also be built using the fact that the charging or discharging time decreases as the battery state degrades.

## 6 FUTURE WORK

As of future work, the authors plan to further investigate the capabilities of Gaussian Process Regression (GPR) so that the model will not require future inputs to predict future outputs, but will aim to predict each individual feature using temperature as an alternative output feature. Moreover, the goal is to construct an alarm system that will leverage the results obtained from the Kalman Filtering approach : using the estimated capacity curve and the measured State of Charge behavior of the battery, an alarm will be triggered when the required recharge period drops below a certain threshold, indicating deteriorating capacity.

## REFERENCES

- [1] Brian Bole, Chetan S Kulkarni, and Matthew Daigle. 2014. *Adaptation of an electrochemistry-based li-ion battery model to account for deterioration observed under randomized use*. Technical Report. SGT, Inc. Moffett Field United States.
- [2] M.F. BRONDANI, A.T.Z.R. SAUSEN, P.S. SAUSEN, and M.O. BINELO. 2017. Battery Model Parameters Estimation Using Simulated Annealing. *TEMA (SACarlos)* 18 (04 2017), 127 – 137. [http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S2179-84512017000100127&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S2179-84512017000100127&nrm=iso)
- [3] Christian Fleischer, Wladislaw Waag, Hans-Martin Heyn, and Dirk Uwe Sauer. 2014. On-line adaptive battery impedance parameter and state estimation considering physical principles in reduced order equivalent circuit battery models: Part 1. Requirements, critical review of methods and modeling. *Journal of Power Sources* 260, Supplement C (2014), 276 – 291. <https://doi.org/10.1016/j.jpowsour.2014.01.129>
- [4] Lijun Gao, Shengyi Liu, and R. A. Dougal. 2002. Dynamic lithium-ion battery model for system simulation. *IEEE Transactions on Components and Packaging Technologies* 25, 3 (Sep 2002), 495–505. <https://doi.org/10.1109/TCAPT.2002.803653>
- [5] K. Goebel, B. Saha, A. Saxena, J. R. Celaya, and J. P. Christophersen. 2008. Prognostics in Battery Health Management. *IEEE Instrumentation Measurement Magazine* 11, 4 (August 2008), 33–40. <https://doi.org/10.1109/MIM.2008.4579269>



- [6] T. Huria, M. Ceraolo, J. Gazzarri, and R. Jackey. 2012. High fidelity electrical model with thermal dependence for characterization and simulation of high power lithium battery cells. In *2012 IEEE International Electric Vehicle Conference*. 1–8. <https://doi.org/10.1109/IEVC.2012.6183271>
- [7] Lingling Li, Pengchong Wang, Kuei-Hsiang Chao, Yatong Zhou, and Yang Xie. 2016. Remaining Useful Life Prediction for Lithium-Ion Batteries Based on Gaussian Processes Mixture. *PLOS ONE* 11, 9 (09 2016), 1–13. <https://doi.org/10.1371/journal.pone.0163004>
- [8] Datong Liu, Yue Luo, Jie Liu, Yu Peng, Limeng Guo, and Michael Pecht. 2014. Lithium-ion Battery Remaining Useful Life Estimation Based on Fusion Nonlinear Degradation AR Model and RPF Algorithm. *Neural Comput. Appl.* 25, 3-4 (Sept. 2014), 557–572. <https://doi.org/10.1007/s00521-013-1520-x>
- [9] Jie Liu, Abhinav Saxena, Kai Goebel, Bhaskar Saha, and Wilson Wang. 2010. An Adaptive Recurrent Neural Network for Remaining Useful Life Prediction of Lithium-ion Batteries. (01 2010).
- [10] R. A. Martin. 2010. A State-Space Approach to Optimal Level-Crossing Prediction for Linear Gaussian Processes. *IEEE Transactions on Information Theory* 56, 10 (Oct 2010), 5083–5096. <https://doi.org/10.1109/TIT.2010.2059930>
- [11] Robert R. Richardson, Michael A. Osborne, and David A. Howey. 2017. Gaussian process regression for forecasting battery state of health. *Journal of Power Sources* 357, Supplement C (2017), 209 – 219. <https://doi.org/10.1016/j.jpowsour.2017.05.004>
- [12] Bhaskar Saha and Kai Goebel. 2009. Modeling Li-ion battery capacity depletion in a particle filtering framework. (01 2009), 1–10.
- [13] Bhaskar Saha, Kai Goebel, and Jon Christophersen. 2009. Comparison of prognostic algorithms for estimating remaining useful life of batteries. *Transactions of the Institute of Measurement and Control* 31, 3-4 (2009), 293–308. <https://doi.org/10.1177/01423331208092030>
- [14] B. Saha, K. Goebel, S. Poll, and J. Christophersen. 2009. Prognostics Methods for Battery Health Monitoring Using a Bayesian Framework. *IEEE Transactions on Instrumentation and Measurement* 58, 2 (Feb 2009), 291–296. <https://doi.org/10.1109/TIM.2008.2005965>
- [15] Xiao-Sheng Si, Wenbin Wang, Chang-Hua Hu, and Donghua Zhou. 2011. Remaining useful life estimation - A review on the statistical data driven approaches. *European Journal of Operational Research* 213 (2011), 1–14.
- [16] E. A. Wan and R. Van Der Merwe. 2000. The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*. 153–158. <https://doi.org/10.1109/ASSPCC.2000.882463>
- [17] Jingliang Zhang and Jay Lee. 2011. A review on prognostics and health monitoring of Li-ion battery. *Journal of Power Sources* 196, 15 (2011), 6007 – 6014. <https://doi.org/10.1016/j.jpowsour.2011.03.101>
- [18] Enrico Zio and Giovanni Peloni. 2011. Particle filtering prognostic estimation of the remaining useful life of nonlinear components. *Reliability Engineering and System Safety* 96, 3 (March 2011), 403–409. <https://hal-supelec.archives-ouvertes.fr/hal-00609502>