Bayesian methods for battery state of health estimation

Antti Aitio

Department of Engineering

University of Oxford

This dissertation is submitted for the degree of

Doctor of Philosophy

Hertford College

January 2023

To Sally, Suvi and Henri.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and the acknowledgements and publications pages. Parts of the work have been published in journal papers and presented at conferences and seminars. These are specified in the text and referenced as appropriate.

Antti Aitio
January 2023

Acknowledgements

Spending the last four years doing my DPhil has been tremendously rewarding. It was a decision long in the making to leave behind my previous career and take the chance to immerse myself again in science. I owe thanks to my wife Sally, whose understanding and support over the years, especially during the COVID-19 pandemic, were invaluable. Without her, this work would undoubtedly not have been completed.

I was lucky to do my research with excellent people in the Battery Intelligence Lab. First and foremost, I would like to thank Professor David Howey for his support, guidance and enthusiasm for my research project throughout. I would also like to give special thanks to the post-doctoral researchers in our group and beyond, Pedro Ascencio, Luis Couto, Volkan Kumtepeli, Nicola Courtier, Ross Drummond and Valentin Sulzer, whose technical knowledge and willingness to answer my endless questions is greatly appreciated. Last but not least, these four years would not have been the same without sharing the ups and downs with my fantastic fellow students and colleagues, Samuel Greenbank, Malgorzata Wojtala, Jorn Reniers, Trishna Raj, Adam Lewis-Douglas, Taeho Jung, Zihao Zhou, Masaki Adachi, Rebecca Perriment and Ralph Lane. With them, the hours spent at the office were an absolute delight.

Abstract

Estimating the state of health of battery energy storage systems is key to their operational safety and reliability, both of which affect lifetime cost. However, accurate estimation of state of health remains challenging, as measurement techniques used in laboratory environments are not available in real-world operating environments. In this work, a framework is developed that combines the relative strengths of commonly applied model- and data-driven approaches to state of health estimation. Gaussian process regression, a flexible Bayesian method of learning arbitrary functions from input-output data, is applied to estimate the parameters of low-order battery models as functions of internal states, operating conditions and lifetime. The approach is first motivated by the difficulty of parameter identification for physics-based battery models from real-world data. Then it is shown how electrical equivalent circuit models can be extended to include parameter dependencies on operating conditions and lifetime in a data-driven manner. The framework is then applied to two different usage scenarios. First, internal resistance is estimated for a fleet of solar-connected lead-acid batteries located in sub-Saharan Africa, where the resulting health metric is shown to provide an early indication of end-of-life failure. Second, a first-order RC circuit, coupled with a one-state thermal model, is parameterised in a joint process that also simultaneously estimates battery states, using data from a Li-ion cell under laboratory conditions. The only prerequisite was the cell-level open-circuit voltage versus charge curve and, in the case of the Li-ion cell model, a single thermal parameter. Given this, the method is agnostic to chemistry and battery construction.

Enabling robust and fast state of health estimation for large fleets of batteries has the potential to 'close the loop' in terms of battery energy storage system design. Instead of performing laboratory-based ageing experiments, field data can be used directly to determine factors that affect battery life. By incorporating this information into fault diagnosis and health-aware battery management systems, safety and reliability will be improved. Furthermore, with deeper understanding of degradation in the real world, better design of energy storage systems will ultimately lead to better cost efficiency through reduced over-engineering.

Publications

The work presented in this thesis draws mostly on the following publications:

[1] Aitio, A., Marquis, S. G., Ascencio, P., Howey, D. A. (2020). Bayesian parameter estimation applied to the Li-ion battery single particle model with electrolyte dynamics. IFAC-PapersOnLine, 53(2), 12497–12504.

[2] Aitio, A., Howey, D. A. (2020). Combining non-parametric and parametric models for stable and computationally efficient battery health estimation. Proceedings of the ASME 2020 Dynamic Systems and Control Conference.

[3] Aitio, A., Howey, D. A. (2021). Predicting battery end of life from solar off-grid system field data using machine learning. Joule, 5(12), 3204–3220.

[4] Aitio A., Jöst. D., Sauer, D.U., Howey, D. A. (2022). Learning battery model parameter dynamics from data with recursive Gaussian process regression. (**To be submitted**)

[5] Sulzer, V., Mohtat, P., Aitio, A., Lee, S., Yeh, Y., T., Khan, M. U., Steinbacher, F.,Lee, J.W., Siegel, J.B., Stefanopoulou, A. G., Howey, D. A. (2021). The challenge of battery lifetime prediction from field data. Joule, 1–20.

[6] Couto, L. D., Zhang, D., Aitio, A., Moura, S., Howey, D. A. (2020). Estimation of Parameter Probability Distributions for Lithium-Ion Battery String Models Using Bayesian Methods. Proceedings of the ASME 2020 Dynamic Systems and Control Conference, 1–10.

For publications [1–4] I was the primary author. Apart from the derivation and code for the forward simulation of the SPMe in [1] which is from a previous publication [7], I contributed

the theory, code and analysis for all four papers. For publication [5] I contributed to the writing and editing, mostly in sections relating to real-world data and the discussion on uncertainty. For the final publication [6] I helped with the development of the theory and implementation of the adaptive parallel tempering algorithm [8].

Table of contents

List of figures

List of tables

List of Algorithms

List of symbols

Acronyms and Abbreviations

(U/E)KF Unscented / Extended Kalman filter

ARD Automatic relevance detection

BESS Battery energy storage system

BFGS Broyden-Fletcher-Goldfarb-Shanno algorithm

BMS Battery management system

ECM Equivalent circuit model

EHM Equivalent hydraulic model

ESS Effective sample size

FEC Full equivalent cycles

GPC Gaussian process classification

GPR Gaussian process regression

LTI-SDE Linear time-invariant stochastic differential equation

MAP Maximum-a-posteriori estimate

MCMC/HMC Markov chain/Hamiltonian Monte Carlo

MH Metropolis-Hastings

MLE Maximum-likelihood estimate

MMSE Minimum mean-square error estimate

NLML Negative log-marginal likelihood

NN Neural network

NUTS No U-turn sampler

OCP/V Open circuit potential/voltage

P2D Pseudo-2-dimensional model

PBROM Physics-based reduced order model

RTSS Rauch-Tung-Striebel smoother

RUL Remaining useful life

RVM Relevance vector machine

RWM Random-walk Metropolis

SE Squared exponential kernel

SLI Starting, lighting and ignition

SOC State of charge

SOH State of health

SPM(e) Single particle model (with electrolyte dynamics)

SVM Support vector machine

VRLA Valve-regulated lead-acid battery

WV Wiener velocity kernel

Equivalent circuits

$\alpha(.)$ Inverse RC time constant

$\beta(.)$ Inverse capacitance

$\hat{c}$ Estimated acid concentration

ζ Extent of degradation

I Applied current

$Q^{-1}(.)$ Inverse Capacity

$R_0(.)$ Series resistance

T_c Cell temperature

$V_0(.)$ Terminal open-circuit voltage

V_1 Voltage across RC pair

z State of charge

Gaussian processes

γ Inverse squared length scales of ARD kernel

$\mathbf{F}$ Continuous-time state transition matrix

$\mathbf{K}$ Gramian matrix

$\mathbf{P}$ State covariance matrix

$\mathbf{Q}$ Process noise covariance matrix

$\mathbf{R}$ Measurement noise covariance matrix

$\mathbf{X}$ Training input

$\mathbf{X}_*$ Test input

$k(.)$ Kernel/covariance function

Monte Carlo methods

α Acceptance probability

ε Leapfrog integrator step size

$\hat{R}$ (H/MC)MC convergece statistic

$\mathbf{M}$ HMC (inverse) mass matrix

$\mathbf{p}$ HMC momentum

$K(.)$ HMC kinetic energy

L Leapfrog integrator step count

Physics-based models

$\Delta\Phi_{\text{Elec}}$ Electrolyte Ohmic loss

$\Delta\Phi_{\text{Solid}}$ Solid state Ohmic loss

ε_{k} Electrolyte volume fraction

η_{c} Concentration overpotential

$\eta_{\mathrm{r,k}}$ Reaction overpotential

κ Electrolyte conductivity

Ω_{k} Electrolyte subdomains

σ_{k} Solid state conductivity

a_{k} Electrode cross-sectional area

b Bruggeman's coefficient

$c_{\mathrm{e,k}}$ Electrolyte lithium concentration

D_{e} Electrolyte lithium diffusivity

$D_{\mathrm{s,k}}$ Solid state lithium diffusivity

F Faraday's constant

I Applied current density

$j_{0,\mathrm{k}}$ Exchange current density

L_{k} Electrolyte subdomain width

m_{k} Reaction constant

R Gas constant

R_{k} Particle radius

T Temperature

t^{+} Cation transference number

$U_{0,\mathrm{k}}$ Half-cell open-circuit potential

V Terminal voltage

$c_{\mathrm{s,k}}$ Solid state lithium concentration

$\mathrm{k} \in \{\mathrm{n}, \mathrm{p}, \mathrm{s}\}$ SPMe subdomain labels for negative electrode, positive electrode, separator

Probabilities

$\chi(.)$ Chi distribution

$\Gamma(.)$ Gamma distribution

$\Gamma^{-1}(.)$ Inverse Gamma distribution

$\mathbb{E}[.]$ Expectation

$\mathbb{V}[.]$ (Co)variance

$\mathcal{N}(.)$ (Multivariate) normal distribution

$\phi(.)$ Energy function - logarithm of unnormalised negative posterior probability

ρ_{l} Autocorrelation lag l

$\mathrm{Beta}(.)$ Beta distribution

KL Kullback-Leibler divergence

p(.) Probability distribution

Chapter 1

Introduction

1.1 State of health estimation in batteries

The demand for battery energy storage systems (BESSs) is set to dramatically expand in the coming years as global efforts to minimize CO_2 emissions gather pace. Batteries play an increasingly important role as the share of intermittent renewable energy sources in the grid increases [9] and the electrification of transport occurs [10]. Additionally, BESSs are central in the efforts to achieve universal access to electricity in the developing world [11] while avoiding local air pollution and CO_2 emissions. The rate of deployment of these systems in all applications will depend on their cost-effectiveness, safety and reliability. As the prices of Li-ion batteries have decreased substantially since their initial commercialisation [12], they are becoming the technology of choice across the board, although older technologies, namely lead-acid, still command substantial market share in applications such as starting, lighting and ignition (SLI) systems and off-grid solar storage systems in the developing world [13, 14]. Regardless of the choice of chemistry, the costs and reliability of BESSs need to be further improved to increase their rate of adoption.

Diagnosis and prognosis of battery state of health (SOH) will play a key role in delivering improvements in the cost-effectiveness of BESSs. Alongside ongoing advances in cell, module

and pack construction, improved battery SOH estimation (referring to both diagnosis and prognosis) in real-world operating environments is crucial for many reasons. First, it allows for efficient timing of preventative maintenance, significantly reducing downtime, especially in situations where BESSs are installed in remote locations. Consequently, the supply chain for spare battery parts may also be better managed. Additionally, battery control algorithms can be made health-aware, prolonging life [15], which may lead to reduced over-engineering, thus lowering the overall cost of the system. Given reliable diagnosis of SOH, use of batteries in second-life applications becomes more viable, and warranties may be priced more accurately. Finally, SOH diagnosis and prognosis using real-world data enables leveraging large amounts of data available from a wide range of operating conditions. Ultimately, this will help in elucidating battery degradation in relevant target applications. As battery testing in lab environments is time-consuming and expensive, it is impossible to replicate all the operating conditions faced in the real world. Often lab-based testing protocols are designed to cause accelerated ageing [16], but such conditions are rarely met in actual operation and it is not clear whether results from such tests correlate with real-world experience. Estimating state of health from large fleets of batteries can therefore dramatically expand our understanding of battery behaviour and usage.

The state of health of a battery is loosely defined as its ability to deliver power and energy, relative to either beginning of life or manufacturing specification. The two most commonly used metrics to describe SOH are (discharge) capacity and internal resistance, where the target application determines the relative importance of each. For example, in large grid-connected BESSs, discharge rates are usually constant and relatively low, implying that discharge capacity is the more important metric. On the other hand, the variable of most interest for electric vehicles is arguably the available range, which is affected by both capacity and internal resistance—under dynamic loading patterns, lower voltage limits are reached earlier with increased internal resistance (e.g. when accelerating at low state of charge) as well as reduced capacity. Battery degradation over (calendar) time and repeated cycling causes capacity and

power fade (i.e. increased resistance), and the end of life is usually defined by some preset threshold, for example, when the battery drops to 80% of its original capacity or its internal resistance doubles.

Currently, most of the work on battery SOH estimation has been done in laboratory conditions. Many high quality publicly available lab-generated battery degradation datasets exist [17]. These are commonly generated by cycling batteries over long periods of time, with periodic standardised reference performance tests to measure battery health [18, 19]. In real-world conditions, on the other hand, it is common that reference performance test data are not available because the required service interruption and cost of testing equipment are prohibitive. Therefore, capacity and/or internal resistance have to be estimated rather than measured directly. This is challenging, as operating conditions in real systems tend to change with varying loads and temperatures over the lifetime of a battery, and health metrics such as capacity and resistance are not independent from those influences. Additionally, data quality in real-world operations is poorer, as gaps in the data are more frequent, and sensor quality is not as good. Further challenges arise when health has to be estimated for modules or packs rather than single cells. All these factors mean that SOH estimation is inherently more difficult than under laboratory conditions, as illustrated in Fig. 1.1—the determination of current state of health and its trajectory using real-world data are both less accurate than in the lab, substantially increasing the uncertainty of any prognosis.

In the remainder of this chapter, the state-of-the-art SOH estimation methods in the absence of direct measurements are reviewed and their applicability for real-world scenarios is assessed. Using field data comes with challenges limiting the usability of many of the existing techniques, which is discussed in depth.

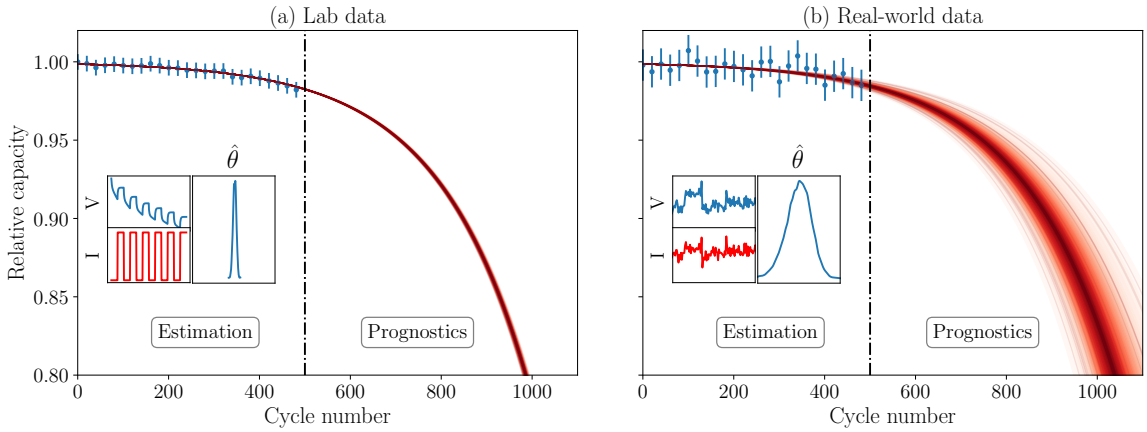


Fig. 1.1 In laboratory conditions, SOH can be accurately estimated with the availability of standardised reference performance tests, which allow for precise determination of health parameters $\hat{\theta}$. In real-world environments, due to lack of direct measurements, poorer data quality and varying operating conditions, estimates of parameters $\hat{\theta}$ are less precise, resulting in a substantial increase in uncertainty in both diagnosis and prognosis. Source: Sulzer et al. [5]

1.2 Methods of battery state of health estimation

The ultimate objective of state of health estimation is to provide both a reliable diagnosis of the current state of health and a prognosis for the remaining useful life (RUL) of the battery. The two tasks are also closely related, as diagnosis is a prerequisite for a prognosis, which involves the fitting of a model of future SOH behaviour using data from diagnosis. In lab conditions, diagnosis is comparatively easy. With controlled charge and discharge tests and current pulse tests, discharge capacity and internal resistance may be accurately measured. Also, more detailed information on battery health can be gained from electrochemical voltage spectroscopy [18] or impedance measurements [19]. In addition, cell teardown can be performed to analyse battery health separately on each electrode. Currently, the above techniques are not available in the field, so diagnosis has to be performed directly from the measured data—consisting of terminal voltage, applied current and surface temperature—the granularity of which varies depending on the battery management system (BMS). The standard BMS configuration logs the minimum and maximum values of each of the measurements at the module level to reduce data

storage requirements, although some systems allow more detail. Although emerging techniques could provide additional information on battery health in the field, such as acoustic time-of-flight analysis [20] or cell expansion measurements [21], these have not been implemented in real-world operating environments. Given a diagnosis, the extra difficulty of estimating the RUL of a battery comes from the complexity and path dependency of battery degradation [22–24], which is compounded by unknown future operating conditions and usage patterns.

There are multiple reviews on the estimation of battery state of health [25–28], and these commonly categorise estimation into direct (experimental) methods or more indirect model- and data-driven methods. As noted earlier, direct measurement of either capacity or internal resistance is commonly not possible in the field. Hence, the focus here will be on model- and data-driven methods for SOH estimation. Fig. 1.2 briefly outlines these using the current commonly-used taxonomy. Each of the methods is discussed in detail below, with special consideration given to their applicability in real-world situations.

1.2.1 Model-driven SOH estimation

Model-driven battery SOH estimation involves the parameterisation of a battery model that describes the input/output response of a battery. The input is commonly the applied current and the output(s) are the terminal voltage and/or temperature. As the parameterisation task is repeated throughout the lifetime of a battery, the change in model parameters reflects degradation. Parameters of interest are in general those related to the (discharge) capacity and internal resistance of the cell. Prognosis in this scenario then involves fitting a model to the evolution of the battery model parameters and projecting this forward in time.

To implement model-driven SOH diagnosis, the choice of battery model is key. More specifically, the model must be chosen so that it contains a parameter set that is identifiable (in the practical sense, given data constraints) and reflects the ageing process in the battery. The two main types of model considered in the literature are electrical equivalent circuits (ECMs)

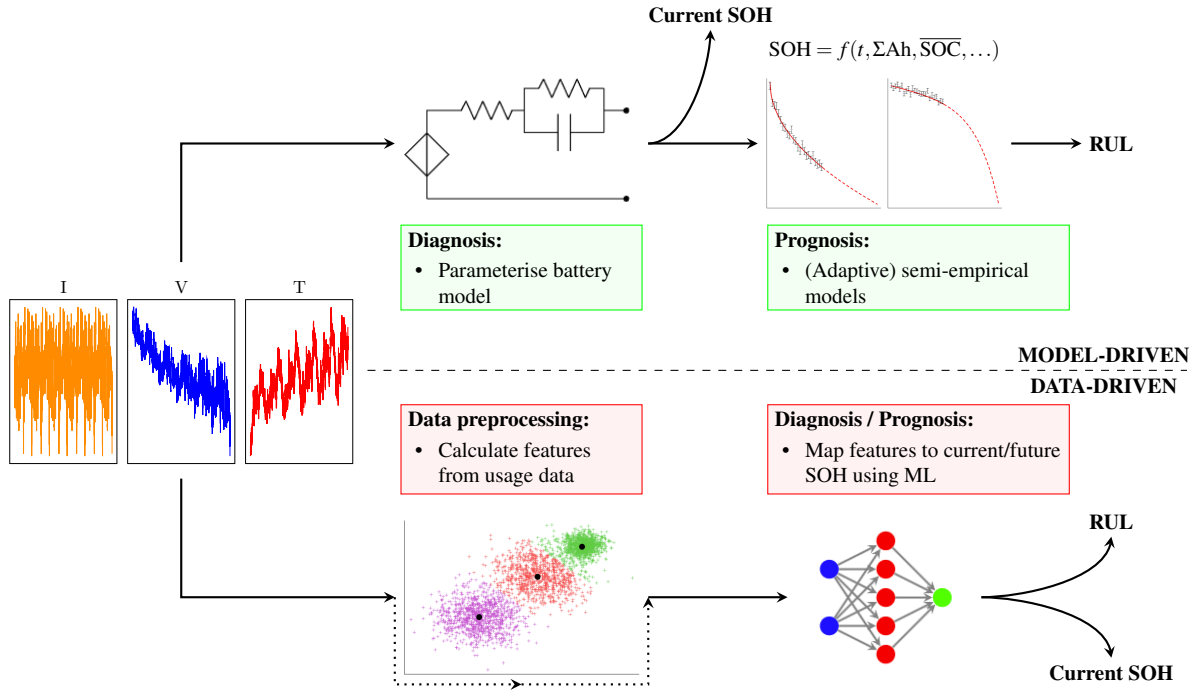


Fig. 1.2 Schematic of SOH estimation methods in the absence of direct measurements. The literature commonly uses the distinction between model- and data-driven paradigms. The key difference between the two is that model-driven approaches employ explicit models for battery input-output response and degradation, whereas a (purely) data-driven framework uses machine learning methods to map battery data directly to SOH estimates. Data pre-processing in the data-driven approach is commonly used to generate lower dimensional features, which serve as inputs to the estimator.

and physics-based reduced order models (PBROMs). Below, each type of model is discussed in terms of identifiability, expressivity and computational requirements. The identifiability of the model, that is, whether it is possible to uniquely determine its unknown parameters from the observed data [29], is the limiting factor in model-driven SOH diagnosis due to the types of data available from real operating environments.

Equivalent circuit models

Due to their simplicity, the most common battery model choice for the estimation of SOH in the literature is ECMs [30–35]. Consisting of ideal circuit elements such as voltage sources, resistors and capacitors, these models usually contain 2-3 ordinary differential equations

describing battery dynamics, depending the number of circuit elements used, therefore requiring a relatively small parameter set. Coupled thermal models have also been incorporated [36] into ECMs, which can improve SOH estimates. Given the simplicity, ECMs are robust and fast to execute.

Compared to more complex battery models, the identifiability of ECMs can be determined relatively easily. In the frequency domain, for example, the *structural* identifiability of (linear) RC networks has been studied in detail [37]. However, while it can be shown that a given ECM can *in principle* be parameterised with a given input signal, this theoretical result does not necessarily hold in practice. For example, an RC network containing elements yielding very short time constants is not in practice identifiable if the data sampling frequency is too low. Also, numerical ill-conditioning may be an issue [38], for example in the case where the series resistor is being estimated with very small applied currents, meaning that small errors in current or voltage result in large errors in resistance. Another more substantial obstacle is the lack of knowledge of initial conditions in field data. As telemetry data, at best, includes SOC, current, voltage and temperature, accurately initialising an ECM can only be done after a battery has been at rest for longer than the longest time constant in the model. However, this challenge can be overcome by jointly estimating the states and parameters of the battery [30, 31], in which case there is less sensitivity of parameter estimates to incorrect initialisation. The structural identifiability of the joint state-space system may be analysed considering the observability of an augmented state vector that includes the model parameters [39, 40]. However, since the joint state/parameter system is usually nonlinear, this type of analysis, relying on the computation of Lie derivatives, does not guarantee global identifiability (i.e. across all combinations of states and parameters).

Finally, the expressivity and accuracy of ECMs may be limited due to their low dimensionality. Parameterising an RC network can yield time constants that roughly correspond to those of physical processes such as solid-state diffusion in Li-ion cells, but parameterisation is affected

by the number of parallel RC elements in the system—in other words, changing the number of RC pairs in the network also changes their time constants, making it difficult to interpret them in relation to physical phenomena. Furthermore, ECMs suffer from decreased accuracy when applied to a broad range of operating conditions, as model parameters depend on internal battery states such as the state of charge (SOC) and temperature [41], as well as the applied current. This means that SOH estimation through repeated parameter estimation throughout the lifetime of the battery may yield unstable or inaccurate results as varying operating conditions affect parameters. This may be mitigated, for example, by imposing an explicit temperature and/or SOC dependency of circuit elements [42, 35] or by simultaneously including multiple models to cover a range of operating points [43].

Physics-based battery models

The physics-based battery models in the literature are derived from porous electrode theory [44]. For Li-ion cells, the current gold standard is the so-called pseudo-2-dimensional (P2D) model by Doyle, Fuller and Newman [45, 46], but similar approaches apply to other battery chemistries such as lead-acid [47, 48]. Constructed from first principles, the models consist of coupled sets of partial differential equations, which accurately model the thermodynamics, reaction kinetics and the various transport processes occurring inside the cell. However, this accuracy comes with a high computational cost relative to ECMs. This has prompted a large volume of work exploring computationally efficient solutions to the P2D model [49] as well as the creation of reduced order physics-based models [7, 50–52] which attempt to reduce computational complexity while maintaining high accuracy. Currently, large open-source libraries of computationally efficient physics-based models are available [53]. Some very low-order physics-based models [47, 48, 51, 52] are also on a par with ECMs in terms of computational complexity while providing more accuracy and better interpretability of model

parameters. In comparison to ECMs, PBROMs in theory provide better insight to cell SOH, as changes in their parameters can be linked to changes in different physical processes in the cell.

However, in practice, the parameterisation of these models from commonly measured battery data is difficult. It is well known that even the simplest PBROMs, such as the single particle model, are not fully identifiable [54]. Although this particular problem may be mitigated by lumping subsets of parameters, a fundamental (i.e. structural) identifiability problem arises when fitting PBROMs to battery data. As these models express terminal open-circuit voltage as a sum of the half-cell components, each a function of surface concentrations of active species at the respective electrodes, the initial conditions can not be uniquely identified from a single resting voltage (this applies to Li-ion – in lead-acid this can in fact be done, since open-circuit voltage is a function of acid concentration in the electrolyte only). This problem would be solved by using a reference electrode, measuring each half-cell voltage separately, but these do not exist in commercial cells. Alternatively, it is possible to constrain the problem with prior knowledge of the total lithium inventory [55] so that a unique solution exists with respect to the initial conditions. However, the total Li inventory does not stay constant over the lifetime of a cell. In addition, to identify model parameters, other assumptions are still required, such as fast diffusion kinetics on one electrode [52, 56], making it possible to uniquely attribute overpotentials to either the anode or cathode. Once an appropriate set of assumptions have been made to make the system identifiable, the resulting system becomes similar to an equivalent circuit. An example of this is the so-called equivalent hydraulic model [52], which is a physics-based model with the dimensionality of a 1st order RC circuit, derived through a low order Padé approximation of the solid state diffusion transfer function in one electrode. This system is akin to an ECM with a single, nonlinear RC pair and nonlinear series resistor. Notwithstanding this simplicity, even low-order physics-based models need a substantial parameter set to be experimentally determined [57] as a starting point, before a subset can be identified that is SOH dependent. While these models may be more accurate across a broader range of operating

conditions than ECMs, they still suffer from state dependency of some parameters, namely the exchange current densities, which have a strong temperature dependency, and the electrode solid state diffusivities, which are functions of temperature and local lithium concentration [57]. Therefore, parameterising these models across varying operating conditions may still be challenging or impossible.

With field data, using joint state/parameter approaches with PBROMs is significantly more complex than ECMs as the governing equations are spatially resolved, meaning that the resulting state-space representation after spatial discretisation is of higher dimensionality than those in ECMs. Such approaches with PBROMs have been demonstrated with simulated data [56], but, for the reasons listed above, the advantage of using PBROMs for model-driven SOH diagnosis over ECMs is not a given, even though using PBROMs to forecast the voltage response of a battery is likely to give more accurate results and theoretically the ability to provide SOH diagnosis at a granularity including the half-cells would be advantageous [22].

Hybrid models incorporating machine learning

Recently, there has been an increasing interest in combining machine learning methods with ECMs [58, 59] or PBROMs [60, 58] to improve their performance. This so-called grey box approach intends to take the basic structure of the underlying model and use data-driven techniques to fill in the missing physics. So far, the focus has been to improve the voltage predictions of the underlying model, but these techniques can equally be applied to SOH estimation.

1.2.2 Fitting battery models to data

If the initial conditions of a battery model are precisely known, the parameters of the model may be fitted by minimising the error between predicted and measured outputs by techniques such as least squares [61] or Bayesian methods, which incorporate prior beliefs of model parameters

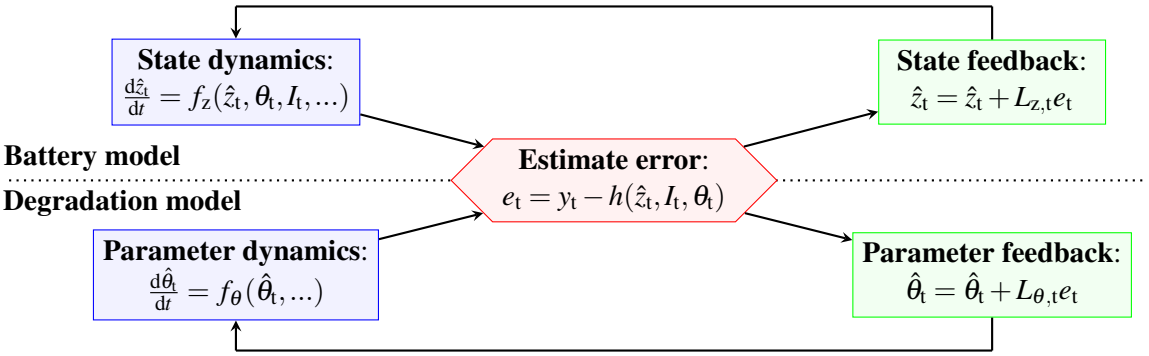


Fig. 1.3 Schematic for the recursive joint state/parameter estimation method used in online approaches. This framework may be implemented using either Bayesian methods [30, 31] or deterministic methods from control theory [56, 32].

into the estimation process [1]. In the absence of data on initial conditions, they also have to be estimated or adaptive methods used which converge the states over time to their true values. As discussed above, in this case the practical *joint identifiability* of parameters and states is a prerequisite for SOH estimation, where factors such as sensor accuracy, logging frequency and numerical conditioning are important. The workflow for joint state/parameter estimation is shown schematically in Fig. 1.3. Following initialisation, the method involves recursively propagating a state-space system defined by the battery model forwards through time series data while using error feedback to adaptively adjust states and parameters at each time step.

This adaptive framework has been implemented using Bayesian filtering methods [30, 31, 62, 33] and deterministic adaptive observers [32, 63, 64, 36]. The choice of approach determines the method of calculation of the feedback gains L_z and L_θ. Using the deterministic approach, applying Lyapunov analysis to construct an observer, stability and convergence of states and parameters may be guaranteed [65] for nonlinear systems, while Bayesian filtering methods offer no such guarantee. Nevertheless, nonlinear filtering methods such as the extended Kalman filter (EKF) [30], unscented Kalman filter (UKF) [31], or particle filter (PF) [62] have been used for state/parameter estimation. Of these, the EKF is computationally least demanding, as the number of states in the state space system is equivalent to the original system [66], compared to the UKF which has dimensionality $2n$+1, where n is the number states in the

original state space. The computational effort of PFs on the other hand is determined by the number of particles propagated through the system, often of the order of 1,000+, where more are needed with increasing dimensionality of the underlying state-space system. Although the EKF is computationally efficient, it has been shown that first-order approximations used by the EKF can result in loss of accuracy with respect to the UKF [67] in battery state estimation. With some nonlinear systems, the UKF will also be inaccurate and/or computationally unstable if, for example, the state-space system contains bound variables. In these cases, due to the Gaussian approximation in the UKF, it is possible that the filter will evaluate the state-space system out of bounds. In these cases, the use of particle filters is the most robust approach.

An additional difficulty with Bayesian filtering arises from the need to tune the process and noise covariance matrices that jointly determine the gains for parameters and states (i.e. speed at which parameters/states adapt given the measurement error). This is equivalent to directly designing the gain in the deterministic approach (e.g. the Luenberger observer). Although adaptive methods to tune the covariances exist [68], in the battery literature, very little attention has been given to obtaining better estimates for them in a principled manner. Methods to obtain rigorous Bayesian estimates for process and noise covariances exist however [66, 69], within the limitations of identifiability. This type of approach has not, however, commonly been used in the context of battery parameterisation.

1.2.3 Model-driven battery health prognosis

Diagnosis of battery SOH in the model-driven framework consists of estimating battery parameters, usually internal resistance and/or capacity. Prognosis in this framework consists of estimating the dynamics of parameter behaviour over battery lifetime. In many of the joint state/parameter estimators (see Fig. 1.3), the 'degradation model' consists of a random walk [30, 31, 33] or a very simple linear model [32], limiting these models' abilities to extrapolate.

A more robust prognosis involves fitting a function to the history of parameter estimates to forecast their future behaviour, which can be done within the same recursive framework.

The most common approach to model-driven prognosis has been to use degradation models originating from semi-empirical ageing maps, which parameterise degradation (usually in terms of capacity) as a function of calendar time and total ampere hour throughput, with adjustments for the average SOC, temperature and voltage [70–72]. Their aim is to be able to replicate the prognosis of more detailed degradation models constructed from first principles [73], while avoiding the difficulty in parameterising them. Constructing these maps is a time-consuming process, as the number of combinations of experimental conditions required to identify the degradation functions over many inputs increases rapidly with the number of experimental parameters. These maps also depend on the specific chemistry and do not address the possible path dependence of degradation [24].

To overcome the difficulty of building empirical ageing maps, adaptive methods are used, involving the recursive reparameterisation of degradation models of the same functional form as ageing maps, where time is the only input. Bayesian filters have also been popular for this purpose, as they readily provide uncertainty estimates around degradation model parameters and RUL prognosis. As the degradation models are nonlinear in their parameters, nearly all studies employ particle filters for the recursive updates. The commonly used functions in degradation models for capacity predictions are in the exponential family [74–77], although variations have also been shown to work, such as logarithmic functions [78] or Verhulst-type models [79]. For internal resistance, exponential [35] and polynomial [76] models have been used. Additionally, it has been noted that the inverse relationship between battery capacity and internal resistance/impedance may be used to yield predictions in both, while only having to diagnose one of them [74, 35]. While some of the work for adaptive RUL prediction takes capacity data as given [75, 76], some work performs simultaneous diagnosis as well, usually by fitting simple circuit models to either EIS data [74, 80] or drive cycle data [35].

1.2.4 Data-driven SOH estimation

In contrast to the model-driven method, data-driven SOH estimation consists of providing a diagnosis and/or prognosis of battery SOH without using an explicit battery dynamics or degradation model. The aim is to map measured operating data directly to state of health, using a range of nonlinear fitting methods. Either raw data in the form of current, terminal voltage and temperature (and SOC if available), or specific features calculated from these may be used, but the principle is to use supervised machine learning techniques to construct a nonlinear map from a set of inputs to a target output, which is most often battery capacity. Reflecting a broader trend in the adoption of ML/AI based methods in all branches of science [81], these methods have become increasingly popular in SOH estimation. As battery degradation is a complex path-dependent process [22, 23], data-driven approaches to prognosis are attractive due to the flexibility and ease of parameterisation they provide in comparison to purely mechanistic models, which can be difficult to parameterise and become very complex when accounting for multiple interacting mechanisms [73]. On the other hand, purely data-driven methods offer limited insight into degradation owing to their 'black-box' approach, and their performance in SOH estimation is limited by the training data they are presented with, as extrapolation outside the training data may be unreliable.

Similarly to model-driven methods, in the literature there is a distinction between SOH diagnosis and prognosis using data-driven approaches. However, unlike model-driven methods where there are explicit models for each stage of estimation, the data-driven approach has less of a distinction between diagnosis and prognosis, as it is only a matter of changing the target output of the fitting procedure to be the 'future' capacity (or indeed RUL directly) rather than the present capacity.

The use of data pre-processing to generate lower dimensional inputs for ML approaches is more common than direct use of raw data, although the latter has also been demonstrated in the context of neural networks (NNs) [82–84], Gaussian process regression (GPR) [85]

and support vector machines (SVM) [86]. Data pre-processing is used to generate features from data, which provide a human input to ML algorithms, reducing the search space and reducing computational effort. For instance, features calculated from galvanostatic charging and discharging voltage curves, such as time or Coulomb count between equi-spaced voltages [87, 88], or the position and values of peaks from incremental capacity analysis [89], have all been demonstrated as inputs. Alternatively, characteristics of the joint distribution of current, voltage and temperature recorded during drive cycles have also been used to diagnose SOH [90]. These types of inputs, calculated from single charge / discharge data, tend to lend themselves to diagnostics rather than prognosis. In contrast, using input features summarising battery usage up to a point in time are common in prognostics. As such, data-driven prediction is very similar to the use of empirical ageing maps, the difference being that the former needs to make fewer assumptions on the functional form between inputs and outputs. This then means that the set of input features can be very rich: common stress factors known to affect battery health are used, such as time, temperature, average SOC, charge/discharge rates, total Ah throughput, cycle count and cumulative times spent at given current, temperature and voltage ranges [91–100]. Machine learning techniques applied to map inputs to outputs to forecast capacity include GPR [85, 89, 91–98], NN [101, 102], SVM [100, 103], and relevance vector machines (RVM) [104]. Each technique has strengths and weaknesses in terms of robustness, computation time, and model storage requirements, which are important in settings where models have to be adaptively re-trained over the lifetime of a battery [105]. However, as each approach can be designed to deal with arbitrary nonlinear functions, arguably one of the most distinguishing criteria becomes whether a model is probabilistic. Probabilistic (i.e. Bayesian) methods not only output uncertainties around point SOH estimates but also provide principled frameworks for inclusion of prior beliefs and model selection, encouraging model parsimony. Consistent uncertainty estimation becomes especially important if models are used to extrapolate outside the training data, where confidence in the prediction therefore should be lower. Out of the

commonly applied methods, GPR and RVM are probabilistic. Bayesian approaches can also be applied to NNs, although the resulting complexity [106] means that they have not been widely applied in the battery literature. SVMs, in turn, are not probabilistic. Furthermore, it should be noted that although advanced ML techniques have been shown to be effective in the prognosis of SOH, simple linear models can also perform well [16] given the right input features.

The good performance of data-driven techniques has been shown in the existing work with laboratory-generated datasets. However, it is not clear to what extent these methods generalise when applied to other datasets. In the prognostics case, for example, battery degradation is highly dependent on usage profiles. The use of accelerated ageing tests [16] to generate capacity data to fit models runs the risk of poor correlation with real-world observations, where operating regimes are likely less onerous. In the diagnostic case the problem is similar. For example, using methods relying on partial charging curves requires at least periodic discharges that are deep enough for the relevant features to be calculated. In addition, in normal applications, other variables such as (ambient) temperature are not controlled, meaning that extrapolation is often required, unless the training dataset contains varying temperatures. For better reliability, exhaustive sets of degradation experiments would be required to train ML based models, which is costly and time consuming. The path dependence of battery degradation [24] exacerbates the problem by further increasing the number of experiments required.

1.2.5 Existing work on SOH estimation from field data

Most of the work on battery SOH estimation has been done using data generated under controlled conditions in labs, using cell-level data. As a result, it is not clear how well existing data- and model-driven approaches translate to real-world applications, where varying operating conditions, cell-to-cell variability, poorer sensor accuracy and poorer data quality and granularity (battery pack data rarely include cell-by-cell measurements) affect the performance of diagnostics and prognostics alike. The reason for the lack of research using field data is

a lack of publicly available real-world data due to commercial secrecy. However, big data platforms collecting battery field data do exist [107] and recently some studies have emerged using large data sets. For example, a feedforward neural network was used to estimate EV battery SOH at a pack level, using as inputs features from usage data from 700 battery and hybrid electric vehicles in Shanghai [99]. Another large-scale study showed the evolution of cell-to-cell variability over pack lifetime using data from 8032 EVs operating in Beijing [108]. On a smaller scale, data collectors were installed on 16 electric taxis in Beijing to obtain validation data for a Bayesian approach with an empirical model for SOH estimation [109]. However, the data used from these studies is not readily available.

1.3 Summary

The current state of the art in model- and data-driven SOH estimation is based mainly on data retrieved from experimental labs under controlled conditions, where the largest campaigns use up to $\sim$ 200 cells [17]. Carefully designed experiments are used to fit both conventional models (ECMs, PBROMs, and empirical aging maps) and data-driven models, and both approaches have been shown to work. However, the use of larger-scale real-world data for the diagnosis of SOH has only received limited attention. Real-world data presents additional challenges to both types of approach to SOH estimation. For model-driven diagnostics, the identifiability of the underlying battery model states and parameters is limited, as field data typically do not contain enough information to fully parameterise even relatively low-order models, limiting insight into degradation. Also, the lowest order models such as ECMs struggle with varying operating conditions, making diagnosis unreliable. On the data-driven side the problem is two-fold. Firstly, if a feature-driven approach is used, the data for calculating features might not be regularly available over the lifetime of a battery. Second, it is likely that varying operating conditions require an extrapolation beyond training data, again reducing the reliability of

diagnosis. This then mandates a flexible approach that estimates uncertainty in the predictions accurately.

Similar arguments apply to both paradigms in the prognostic case. Parameterising empirical ageing maps to account for all possible degradation scenarios is unfeasible in lab settings, and the problem is essentially the same for the data-driven methods, where degradation training data might not sufficiently cover the space of degradation paths observed in real-world operating data. Herein lies the big opportunity for using field data to complement lab experiments in SOH estimation [5]. Being able to estimate battery SOH using data from large fleets of batteries in real-world operating environments offers the potential to greatly improve understanding of degradation. Specifically, with data-driven models, using this type of data implies that the training data covers the relevant cases, making it likely that they would over-perform their model-driven counterparts due to their increased flexibility to better account for the inherent nonlinearities in the aging process.

To take advantage of the large amounts of data obtained from real-world operating conditions in battery prognostics, a reliable method of diagnosis must be developed. As outlined above, due to the properties of field data, this is a significant challenge. This is the main focus of this work; to determine whether it is indeed possible to accurately estimate state of health directly from real-world operating data. Specifically, a hybrid method that combines model- and data-driven methods is developed here, re-casting SOH estimation as a Bayesian parameter estimation problem. It uses as a starting point simple equivalent circuit models and a probabilistic machine learning method (GPR) to model the circuit parameters making them non-parametric functions of internal states and operating conditions, providing a robust flexible framework for SOH diagnosis that can be applied directly to operating data. The method is chemistry-agnostic and is demonstrated for lead-acid and Li-ion batteries with very different use cases, the former in an off-grid solar application and the latter in an automotive application.

Furthermore, it is demonstrated how the hybrid diagnostic method can be used in conjunction with data-driven prognostics to successfully predict end-of-life failure in the off-grid solar case.

Thesis outline

In Chapter 2, the Bayesian approach to parameterisation of a battery model is introduced. Specifically, the limited identifiability of reduced-order physics-based models is discussed in detail. Following this, it is shown in Chapter 3 how simple ECMs may be extended, by treating their parameters as functions battery internal states and degradation, to make them more robust and better suited to SOH estimation in real-world operating environments. The approach relies on a computationally efficient implementation of GPR [110, 111], which allows for a recursive framework in which computational effort scales linearly with the number of datapoints. The GPR method is applied to a fleet of 1027 PV-connected valve-regulated lead-acid batteries (VRLAs) operating in sub-Saharan Africa, where the approach is shown to detect the onset of battery end-of-life failure several weeks before occurrence.

Chapter 4 extends the estimation process by offering a more comprehensive parameterisation of a battery model for a Li-ion cell, which is validated using a publicly available lab generated degradation dataset [112]. Specifically, it is shown how the joint state/parameter estimator using a GP to model the parameters can be used to parameterise a low order electrothermal model. Finally, Chapter 5 concludes the work.

Chapter 2

Physics-based battery model parameter estimation using Bayesian methods

In Chapter 1, parameterisation of battery models was discussed in the context of determining state of health, where tracking the evolution of model parameters over battery lifetime is the cornerstone of model-driven battery SOH estimation. In this chapter, the groundwork is laid for Bayesian parameter estimation of battery models by considering parameter estimation of a physics-based model.

There are two schools of thought on parameter estimation. The *frequentist* approach considers model parameters unknown but fixed and constructs estimators for them, such as the maximum likelihood estimator (MLE). Following this, quantifying the uncertainty around these estimators is done post hoc using, for example, the Cramér-Rao lower bound [113]. The resulting confidence interval, calculated typically at the 95% or 99% confidence level, then represents the upper and lower limits where parameter estimates would lie at the given confidence level, fitting the parameters over different datasets. This frequentist approach to parameter estimation has also been applied to lithium-ion batteries [114, 115].

In contrast, the *Bayesian* approach assumes that the model parameters are themselves random variables over which a prior probability distribution can be defined, encapsulating

previous knowledge of them. Parameter estimation then becomes a task of determining their posterior distribution, which is the prior distribution conditioned by (i.e. updated with) observed data. The two methods yield the same result in the asymptotic case, where infinite data are available for inference, as described by the Bernstein-von Mises theorem [116]. However, in the case of model parameter estimation, this may not be the case, giving different results depending on the approach taken [1]. Throughout the work presented in this thesis, the Bayesian approach is adopted, as incorporating prior information about parameters provides stability to estimates in cases where identifiability is limited. Moreover, the Bayesian framework also provides a rigorous approach for estimating parameters as functions of states and/or operating conditions, by means of Gaussian process regression (see Chapters 3 and 4).

In this chapter, the Bayesian parameter estimation framework is applied to estimate the (joint) posterior distributions of a subset of model parameters for a reduced-order physics-based model, the single particle model with electrolyte dynamics (SPMe) [7]. In principle, parameterising a physics-based model over battery lifetime gives insight to battery health at the electrode level. Here the cases of diffusivities (or diffusional timescales) in the solid phase are considered in detail. These are parameters that are expected to change over lifetime due to degradation. However, it is shown that their limited identifiability, depending on SOC, means that they can not be reliably estimated in real-world scenarios, especially if they also vary significantly over SOC. This is not the case for all model parameters and it is shown that other model parameters can be estimated accurately. The results are consistent with previous theoretical work considering the structural identifiability of the simpler single particle model (SPM) [54]. This chapter relies largely on the previous publication by Aitio et al. [1], but uses improved sampling methods with a more in-depth analysis of the convergence of all estimates. The parameter estimation exercise in this chapter motivates the approach in the next chapter, where an extension of the Bayesian method allows for the incorporation of parameter dependencies on battery states and operating conditions in an equivalent circuit model.

2.1 Bayesian parameter estimation

Bayes' theorem is the starting point for estimating model parameters. With a given model, denoting the observed input-output data as $\mathcal{D}$ and the parameter vector as θ, the posterior distribution of θ given $\mathcal{D}$ is given by

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int_{\theta} p(\mathcal{D}|\theta)\,\mathrm{d}\theta} = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})}, \tag{2.1}$$

where $p(\theta)$ is the prior parameter distribution, $p(\mathcal{D}|\theta)$ the likelihood function relating $\mathcal{D}$ and θ and $p(\mathcal{D})$ the evidence, that is, the probability of observing $\mathcal{D}$ marginalised (i.e. averaged) over θ. Once an estimate of the posterior distribution $p(\theta|\mathcal{D})$ is recovered, parameter point estimates may be calculated. For example, the minimum mean-square error estimate (MMSE) for θ is the posterior mean, and the maximum a posteriori (MAP) estimate is the posterior mode. The identifiability problem, given observed data, is addressed simultaneously, as identifiability is reflected in the characteristics of the posterior distribution and may be quantified by credible intervals, which are defined as the intervals within which a given probability mass lies for the parameters.

Identifiability can be analysed at two different levels. First, *structural* identifiability of dynamic systems addresses the question of whether the states and/or model parameters may theoretically be uniquely identified from a set of input/output data [117]. For linear dynamic systems, identifiability applies *globally*, that is, solutions to states or parameter values can always be defined uniquely if the system is identifiable. For nonlinear systems, the analysis becomes significantly more complex. Often only the weaker form, *local* identifiability, may be shown. This refers to the idea that state and/or parameters may have multiple solutions and that identifiability depends on the locality in the state/parameter space. When states and parameters have to be simultaneously estimated, the problem often becomes nonlinear, in which case local identifiability may be shown by the observability of an augmented state-space system

where parameters become additional states [39, 40]. While necessary for parameter estimation, structural identifiability analysis does not address the issue of the quality of input and output data. *Practical* identifiability, in contrast, does consider data quality, incorporating factors such as measurement noise and sampling frequency. Therefore, practical identifiability is at best limited by the structural component which is encapsulated in it. Structural identifiability of battery models including the SPM, P2D and equivalent circuits has been extensively studied using frequency domain methods [54, 37, 118]. In contrast, Bayesian methods, used here in the time domain, address practical identifiability, reflected in the parameter posterior distribution, as discussed below.

The challenge for the Bayesian approach is to estimate the posterior distribution of parameters. For all but the simplest cases, the right-hand side of (2.1) is computationally intractable, so it needs to be approximated. There are several ways to do this, such as the Laplace approximation, Monte Carlo methods, variational inference or advanced quadrature methods. In this work, the first two are considered in the context of parameter estimation of a physics-based model, which typically represent the opposite ends in terms computational effort and accuracy. First, the simplest method is used, whereby a Gaussian approximation of the posterior is estimated by first finding its mode, then approximating the variance using the so-called Laplace approximation (referred to as the MAP + Laplace method). This method is compared with a state-of-the-art Hamiltonian Monte Carlo approach, which efficiently samples the parameter posterior by simulating Hamiltonian dynamics in parameter space to yield an estimate of the posterior while making no assumptions on its functional form. Beyond these two methods, other methods of posterior approximation such as variational inference [119] or Bayesian quadrature [120] have in some cases been shown to achieve a good balance between speed and accuracy, but these are not considered here.

2.1.1 MAP + Laplace approximation

The starting point for finding MAP estimates of the posterior parameter $p(\theta|\mathcal{D})$ is to note that the normalising constant, i.e. the denominator in equation (2.1), is independent of θ. Hence for the purposes of estimating the mode of $p(\theta|\mathcal{D})$ it is sufficient to consider only the unnormalised posterior probability. Taking the negative logarithm of both sides, we arrive at the so-called *energy function*,

$$\phi(\theta) = -\log p(\mathcal{D}|\theta) - \log p(\theta) \propto -\log p(\theta|\mathcal{D}), \tag{2.2}$$

that is, the negative logarithm of the unnormalised posterior probability. The MAP estimate is the minimum of the energy function, giving the mode of the posterior. Following this, a local Gaussian approximation of the posterior can be made using a second-order Taylor expansion of the energy function [121], so that

$$p(\theta|\mathcal{D}) \sim \mathcal{N}(\theta_{\text{MAP}}, \Sigma^{-1}) \text{ , where } \Sigma = \left.\frac{\partial^2 \phi(\theta)}{\partial\theta\partial\theta^{\mathrm{T}}}\right|_{\theta=\theta_{\text{MAP}}}, \tag{2.3}$$

which is the so-called Laplace approximation. In other words, the inverse Hessian of the energy function gives an approximation of the covariance matrix at the MAP estimate point, the latter being the mean of the Gaussian. This method is commonly used in conjunction with second-order optimisation algorithms (i.e. Newton or quasi-Newton methods), which rely on the calculation of the Hessian in the optimisation process. While the quickest method, this local Gaussian approximation may in some cases be very poor, e.g. when the true posterior is severely asymmetric or discontinuous, or in the case where ϕ has multiple local minima. Also accurate calculation of the Hessian of ϕ for a high-dimensional vector θ is nontrivial—quasi-Newton methods which approximate it 'on-the-fly' [121] may yield inaccurate results for the parameter posterior covariance. It is also possible that gradient-based optimisers terminate at a 'saddle-point' where gradient convergence criteria are met but the Hessian yields a covariance matrix that is not positive definite, making the Laplace approximation invalid.

2.1.2 Hamiltonian Monte Carlo

While the MAP + Laplace approach is commonly used due to its computational efficiency, a more reliable approximation of the full posterior may be obtained by making fewer assumptions about its form. The most comprehensive approach to estimate the posterior is MCMC methods, which use a Markov chain to stochastically explore the posterior [122]. There exist a wide variety of MCMC algorithms [123], but due to its simplicity, the most commonly used is the Metropolis-Hastings (MH) method [124, 125] and the so-called Gibbs sampler, which can be considered a special case of MH [126].

The principle of MCMC methods is to construct a Markov chain (i.e. a sequence of random values where the probability of observing a value is conditional only on the previous state of the chain), which in its stationary state samples efficiently from the target distribution (the parameter posterior in this case). For example, MH constructs a Markov chain where candidate vectors θ_c are sampled from a proposal distribution q conditional on the current state, θ_t. If this proposal distribution is Gaussian, so that $q(\theta_c|\theta_t) \sim \mathcal{N}(\theta_t, \Sigma)$, the method is called the random walk Metropolis (RWM) algorithm. The candidate θ_c is then accepted with a probability determined by the ratio of the posterior probabilities between the candidate and the current state of the chain, $p(\theta_c|\mathcal{D})/p(\theta_t|\mathcal{D})$ (the so-called Metropolis acceptance criterion). If θ_c is accepted, it then becomes the current state. As only the ratio of posterior probabilities is used to determine the transition, this method of MCMC also eliminates the need to calculate the normalising constant in (2.1). Given the transition kernel described by q and the acceptance criterion, the chain drifts towards the 'typical set', that is, the area with the most probability mass. Once the chain converges to its stationary state, i.e. ceases to drift, expectations of arbitrary functions $f(\theta)$ over the posterior $p(\theta|\mathcal{D})$ may be retrieved by a discrete approximation using a simple algebraic average,

$$\mathbb{E}[f(\theta)] = \int_\theta f(\theta) p(\theta|\mathcal{D}) \, \mathrm{d}\theta \approx \frac{1}{N} \sum_{n=1}^{N} f(\theta_n), \tag{2.4}$$

where N is the length of the Markov chain used to compute the expectation. Under ideal circumstances, the discrete approximation approaches the true expectation as $N \to \infty$. In other words, the ideal MCMC estimator satisfies the central limit theorem [122],

$$f_{\mathrm{N}} \sim \mathcal{N}(\mathbb{E}[f(\theta)], \sigma^2_{\mathrm{MCMC}}), \tag{2.5}$$

where the MCMC standard error is given by

$$\sigma_{\mathrm{MCMC}} = \sqrt{\frac{\mathbb{V}[f(\theta)]}{\mathrm{ESS}}}, \tag{2.6}$$

where $\mathbb{V}[f(\theta)]$ is the variance in the Markov chain in the stationary state and the degree to which 'ideality' is reached is determined by the effective sample size,

$$\mathrm{ESS} = \frac{N}{1 + 2\sum_{\mathrm{l}=1}^{\infty} \rho_{\mathrm{l}}}, \tag{2.7}$$

where ρ_{l} denotes the autocorrelation of the chain for lag l. Therefore, if the chain has high autocorrelation, the ESS is smaller and the the MCMC estimator is less reliable. Here lies the key challenge for building and tuning MCMC algorithms: on the one hand, the algorithm has to be such that it reaches its steady state at an acceptable rate, and on the other hand it has to sample the target distribution efficiently once it is in the stationary state. The performance of random walk methods is limited by high dimensionality of the parameter space because average acceptance rates tend to be low because "guessing" the next step randomly is factorially more difficult with parameter dimensionality. Also, if the geometry of the target distribution is complex, and a random walk may well become "stuck" [122], which has the same negative effect on ESS. Although in theory RWM methods will eventually explore all of the parameter space, practical limitations result from these factors. The key method to improve performance in terms of sampling efficiency is to adjust the proposal distribution covariance matrix Σ, but

this is time consuming (and not always sufficient) to perform for high-dimensional parameter vectors, requiring many trial runs including thousands of evaluations of the likelihood function $p(\mathcal{D}|\theta)$. Some algorithms exist to adjust the covariance of the proposal during the process [127], but they may not be enough to deal with pathological cases of complex geometries.

Hamiltonian Monte Carlo methods seek to address these shortcomings of standard random-walk methods. Both classes of methods are inspired by the concept of a canonical distribution, originating from statistical mechanics, which is defined by the energy function,

$$p(\theta) = \frac{1}{Z}\exp\left(-\frac{E(\theta)}{T}\right), \tag{2.8}$$

where temperature T is set to 1 in this case. In standard MCMC (e.g. Metropolis-Hastings), $E(\theta) = \phi(\theta)$, the negative logarithm of the unnormalised posterior, where $\phi(\theta)$ is conceptually equivalent to a position-dependent potential energy that is repeatedly sampled to converge to the stationary state around its minimum. Hamiltonian Monte Carlo, as the name implies, defines energy using the Hamiltonian, including the potential and kinetic energies, giving the canonical distribution

$$p(\theta, \mathbf{p}) = \frac{1}{Z}\exp\left(-\frac{\phi(\theta) + K(\theta, \mathbf{p})}{T}\right), \tag{2.9}$$

where $\mathbf{p}$ represents the hypothetical momentum of the parameter vector θ in phase space. The form of kinetic energy is user-defined. Here the Euclidean-Gaussian [122] function is used,

$$K(\theta, \mathbf{p}) = K(\mathbf{p}) = \frac{1}{2}\mathbf{p}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{p}, \tag{2.10}$$

where $\mathbf{M}^{-1}$ is the so-called mass matrix, which is optimised in the warm-up phase of the sampling process. Exploration of the target distribution proceeds in three steps: first, momentum $\mathbf{p}$ is sampled from a normal distribution, followed by propagation of the pair $(\theta, \mathbf{p})$ over

(hypothetical) time, using Hamiltonian dynamics,

$$\begin{aligned} \frac{\mathrm{d}\theta}{\mathrm{d}t} &= \frac{\partial K}{\partial \mathbf{p}}, \\ \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} &= -\frac{\partial \phi}{\partial \theta}. \end{aligned} \tag{2.11}$$

This is followed by calculating the acceptance probability of $(\theta, \mathbf{p})$ using the standard Metropolis criterion. Since the canonical distribution (2.10) factorises over θ and $\mathbf{p}$, its marginal distribution over θ is then the target distribution of interest. The basic implementation is shown in Algorithm 2.1. In discrete time, Hamiltonian dynamics (2.11) are numerically evaluated by the 'leapfrog integrator' [128], applying alternating updates to position and momentum with step size ε. The integrator is repeated L steps ahead. A key requirement for using Hamiltonian dynamics is the calculation of the potential energy gradient, $\nabla_\theta \phi(\theta)$. This means that the $\phi(\theta)$ must be differentiable. For this purpose, sampling is often done using monotonic transformations of θ which result in the support of both the prior $p(\theta)$ and the likelihood function $p(\mathcal{D}|\theta)$ being unbound.

Standard HMC is, in principle, a much more efficient sampler than RWM because propagation with Hamiltonian dynamics preserves energy and therefore θ can travel further away from its starting point at each iteration without affecting the acceptance rate. This increases the ESS because the autocorrelation between samples is reduced. However, HMC also suffers from the need to tune the parameters ε and L. The problem here is analogous to tuning the proposal covariance in RWM—if L is too small, exploration is very slow as successive θ are too close. If it is too high, the sampler revisits locations in $(\theta, \mathbf{p})$. Step length ε in turn affects energy conservation in the leapfrog integrator; the integrator error in energy grows as a function of ε, resulting in low acceptance rates with large ε. If ε is too low, the system takes many steps that are likely too small [129]. The tuning of the mass matrix $\mathbf{M}^{-1}$ can also be performed on the fly during the early phases of HMC which can improve performance [122] by empirically setting it as the covariance over samples in a warm-up phase.

Algorithm 2.1 Standard HMC algorithm: exploration occurs by first sampling from a distribution over momenta, followed by simulating Hamiltonian dynamics for a predetermined number of steps L with step length ε and finished by using Metropolis acceptance criteria.

Initialize: $\theta = \theta_0 \in \mathbb{R}^D$, $\theta_{\text{h}} = [n_{\text{iter}}, D]$
for $i = 1 \dots n_{\text{iter}}$ **do**
  $\mathbf{p}^\star \sim \mathcal{N}(0, \mathbf{M})$ ▷ Sample from momentum distribution
  $\mathbf{p} = \mathbf{p}^\star$, $\theta = \theta^\star$
  for $l = 1 \dots L$ **do** ▷ Hamiltonian dynamics using leapfrog integrator
    $\mathbf{p} = \mathbf{p} + \frac{\varepsilon}{2} \nabla_\theta \phi(\theta)$
    $\theta = \theta + \varepsilon \mathbf{M} \mathbf{p}$
    $\mathbf{p} = \mathbf{p} + \frac{\varepsilon}{2} \nabla_\theta \phi(\theta)$
  end for
  $\alpha = \min \left\{ 1, \frac{\exp(\phi(\theta) + K(\mathbf{p}))}{\exp(\phi(\theta^*) + K(\mathbf{p}^*))} \right\}$ ▷ Metropolis acceptance criterion
  if $\alpha > \mathcal{U}(0,1)$ **then**
    $\theta^* = \theta$
  end if
  $\theta_{\text{h}}[\text{i}, :] = \theta^*$
end for

To remove the difficult and time-consuming need to manually tune ε and L through trial HMC runs, which is especially problematic for high-dimensional target distributions, Hoffman and Gelman [129] developed a method called the no U-turn sampler (NUTS). For the sake of brevity only the main aspects of the NUTS algorithm, which eliminates the need to hand-tune HMC parameters, are summarised here at a high level. A detailed description may be found in the original publications [129, 122].

The NUTS method first eliminates the need to predetermine the number of leapfrog integrator steps L. The principle is to replace simple forward propagation of the Hamiltonian with building a binary tree. This is done by first choosing a direction of simulation for the Hamiltonian, either backward or forward in hypothetical time, and taking 2^{j-1} leapfrog steps in that direction starting from the edge of the tree, where j is the current 'doubling count' (starting at 1). This process is repeated recursively until the direction of travel at either edge node of the

binary tree starts turning back on itself (i.e. 'does a U-turn'), whereupon a candidate point is sampled from the set generated during the doubling process. By doing this, NUTS performs a wide search over θ at the given energy level. The step length ε is in turn updated iteratively, targeting a usual preset acceptance rate of 65-80%. The updating scheme is based on stochastic gradient descent [130], which guarantees a convergence to the target acceptance rate.

The basic NUTS method has been subsequently extended [122], and it has now become a popular MC sampler in many probabilistic programming environments [131–133]. Outside of the original work, it has also been shown to compare favourably to other samplers such as Metropolis-Hastings and Gibbs [134, 135]. For the purposes of parameter estimation here, the implementation of NUTS in the `julia` programming language, `AdvancedHMC.jl` [132], was used because it is computationally advantageous to calculate the gradients of the energy function using automatic differentiation which is readily available in `julia`.

2.2 Single particle model with electrolyte dynamics

The SPMe schematic Fig. 2.1 shows a Li-ion cell in which the positive and negative electrodes consist of sets of identical particles of radii R_p and R_n respectively and a separator of width L_s. The macroscopic domain is divided into three subdomains Ω_n, Ω_s, Ω_p over the negative electrode, the separator and the positive electrode respectively where

$$\Omega_\mathrm{n} = [0, L_\mathrm{n}] \ , \ \Omega_\mathrm{s} = [L_\mathrm{n}, L - L_\mathrm{p}] \ , \ \Omega_\mathrm{p} = [L - L_\mathrm{p}, L]$$

The total cell thickness L is given by the sum of each subdomain, so that $L = L_\mathrm{n} + L_\mathrm{p} + L_\mathrm{s}$. As in the SPM, the concentration of lithium in the solid state in each electrode $\overline{c}_{\mathrm{s,k}}(r,t)$ over the radial dimension $r \in [0, R_\mathrm{k}]$ is governed by Fickian diffusion with dynamics described by

$$\frac{\partial \overline{c}_{\mathrm{s,k}}}{\partial t} = \frac{D_{\mathrm{s,k}}}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial \overline{c}_{\mathrm{s,k}}}{\partial r} \right), \tag{2.12a}$$

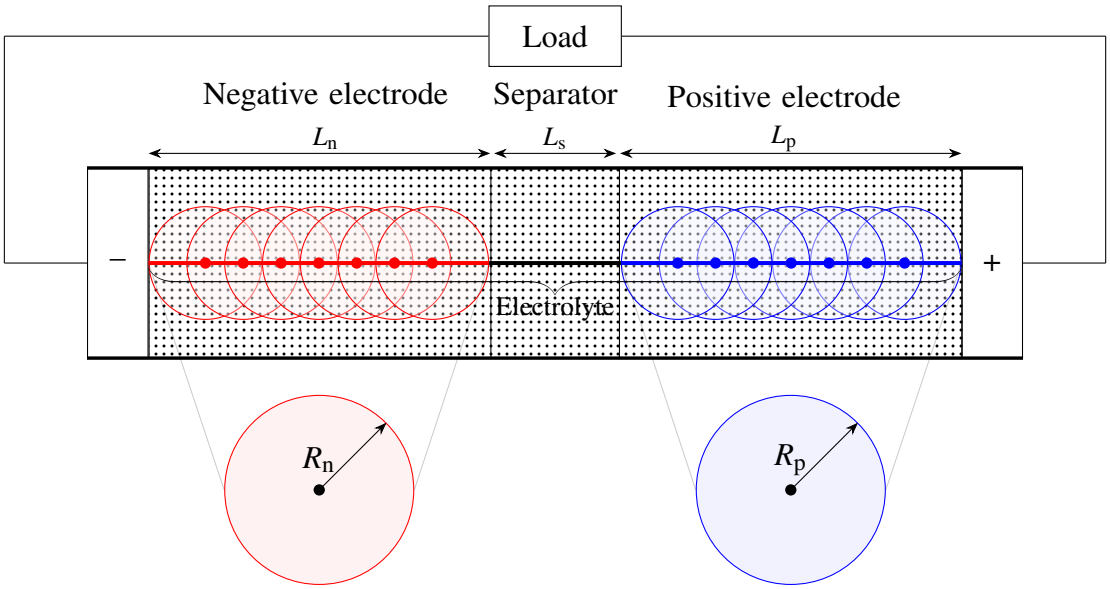


Fig. 2.1 A schematic of a Li-ion cell as modelled by the SPMe, showing the positive and negative electrodes as uniform spherical particles of radius R_p and R_n respectively and the separator in between. The electrolyte domain spans the total thickness of the cell, given by the sum of the electrode and separator thicknesses, so that $L = L_n + L_p + L_s$.

where $\bar{c}_{s,k}$ denotes the electrode-average Li-ion concentration within the particle(s), k denotes the subdomain (n = negative electrode, p = positive electrode, s = separator) and $D_{s,k}$ is the diffusivity of lithium in the solid phase of each electrode. The Neumann boundary conditions for the centres and surfaces of particles are given by

$$\left.\frac{\partial \bar{c}_{s,k}}{\partial r}\right|_{r=0} = 0, \tag{2.12b}$$

$$-D_{s,k}\left.\frac{\partial \bar{c}_{s,k}}{\partial r}\right|_{r=R_k} = \begin{cases} \frac{I}{Fa_n L_n}, & k = n, \\ -\frac{I}{Fa_p L_p}, & k = p, \end{cases} \tag{2.12c}$$

where F is Faraday's constant, a_k and L_k represent the specific active area (i.e. area per unit volume) and thickness of each electrode respectively, and I is the applied current density. The concentration of lithium over the macroscopic dimension $x \in [0, L]$ (i.e. in the electrolyte), $c_e(x,t)$, is governed by the Onsager-Stefan-Maxwell relations, where the Li source/sink is the

electrode-averaged Li-ion exchange with the particles, so that

$$\varepsilon_{\mathrm{k}}\frac{\partial c_{\mathrm{e}}}{\partial t}=\varepsilon_{\mathrm{k}}^{b}D_{\mathrm{e}}\frac{\partial^2 c_{\mathrm{e}}}{\partial x^2}+\begin{cases}\frac{(1-t^+)I}{FL_{\mathrm{n}}}, & \mathrm{k}=\mathrm{n},\\ 0, & \mathrm{k}=\mathrm{s},\\ -\frac{(1-t^+)I}{FL_{\mathrm{p}}}, & \mathrm{k}=\mathrm{p},\end{cases} \tag{2.13a}$$

where D_{e} is the diffusivity of lithium in the electrolyte, ε_{k} the electrolyte volume fraction of the sub-domain k, b the Bruggeman coefficient, and t^+ the cation transference number. Neumann boundary conditions are specified at the edges of the electrode sandwich,

$$\left.\frac{\partial c_{\mathrm{e}}}{\partial x}\right|_{x=0}=\left.\frac{\partial c_{\mathrm{e}}}{\partial x}\right|_{x=L}=0. \tag{2.13b}$$

The initial conditions are set to a typical average value of Li concentration in the electrolyte,

$$c_{\mathrm{e}}(x,0)=c_{\mathrm{e,typ}}. \tag{2.13c}$$

Given the dynamics of the SPMe, governed by the PDEs (2.12)-(2.13), the terminal voltage V is expressed in electrode-averaged quantities,

$$V=\overline{U}_0+\overline{\eta}_{\mathrm{r}}+\overline{\eta}_{\mathrm{c}}+\overline{\Delta\Phi}_{\mathrm{Elec}}+\overline{\Delta\Phi}_{\mathrm{Solid}}, \tag{2.14a}$$

where $\overline{U}_0$ is the open circuit potential, $\overline{\eta}_{\mathrm{r}}$ and $\overline{\eta}_{\mathrm{c}}$ are the reaction and electrolyte concentration overpotentials and $\overline{\Delta\Phi}_{\mathrm{Elec}}$ and $\overline{\Delta\Phi}_{\mathrm{Solid}}$ are the Ohmic losses in the electrolyte and solid phases, respectively. The potential $\overline{U}_0$ is given by the difference in half-cell potentials, which are functions of the electrode-average surface concentrations on each particle,

$$\overline{U}_0=U_{0,\mathrm{p}}(\overline{c}_{\mathrm{s,p}}|_{r=R_{\mathrm{p}}})-U_{0,\mathrm{n}}(\overline{c}_{\mathrm{s,n}}|_{r=R_{\mathrm{n}}}) \tag{2.14b}$$

and $\overline{\eta}_{\mathrm{r}}$ is similarly

$$\overline{\eta}_{\mathrm{r}} = \overline{\eta}_{\mathrm{r,p}} - \overline{\eta}_{\mathrm{r,n}}, \tag{2.14c}$$

where the reaction overpotential for each electrode follows the Butler-Volmer relation

$$\overline{\eta}_{\mathrm{r,k}} = -\frac{2RT}{F}\sinh^{-1}\left(\frac{I}{a_{\mathrm{k}}\bar{j}_{0,\mathrm{k}}L_{\mathrm{k}}}\right), \quad \mathrm{k} = \{\mathrm{p},\mathrm{n}\}, \tag{2.14d}$$

where R is the ideal gas constant and T temperature. The electrode-averaged exchange current density is given by

$$\bar{j}_{0,\mathrm{k}} = m_{\mathrm{k}}(\overline{c}_{\mathrm{s,k}}|_{r=R_{\mathrm{k}}})^{1/2}(\overline{c}_{\mathrm{s,k,max}} - \overline{c}_{\mathrm{s,k}}|_{r=R_{\mathrm{k}}})^{1/2}(\overline{c}_{\mathrm{e,k}})^{1/2}, \tag{2.14e}$$

where m_{k} is the reaction constant. The electrolyte concentration overpotential $\overline{\eta}_{\mathrm{c}}$ in turn is given by

$$\overline{\eta}_{c} = \frac{2RT}{Fc_{\mathrm{e,typ}}}(1-t^{+})\left(\overline{c}_{\mathrm{e,p}} - \overline{c}_{\mathrm{e,n}}\right). \tag{2.14f}$$

Finally, the Ohmic loss for the electrolyte is given by

$$\overline{\Delta\Phi}_{\mathrm{Elec}} = -\frac{I}{\kappa_{\mathrm{e,typ}}}\left(\frac{L_{\mathrm{n}}}{3\varepsilon_{\mathrm{n}}^{b}} + \frac{L_{\mathrm{s}}}{\varepsilon_{\mathrm{s}}^{b}} + \frac{L_{\mathrm{p}}}{3\varepsilon_{\mathrm{p}}^{b}}\right) \tag{2.14g}$$

for the electrolyte, where κ_{e} is the conductivity of the electrolyte, and the solid phase Ohmic losses are given by

$$\overline{\Delta\Phi}_{\mathrm{Solid}} = -\frac{I}{3}\left(\frac{L_{\mathrm{p}}}{\sigma_{\mathrm{p}}} + \frac{L_{\mathrm{n}}}{\sigma_{\mathrm{n}}}\right), \tag{2.14h}$$

where σ_{k} are the conductivities of the electrodes.

2.2.1 Spatial discretisation of the SPMe

Generally, to use HMC methods to estimate posterior distributions of parameters one requires a fast calculation of the energy function (2.2) and its gradient because the number of iterations

required is high (order 10^4 in this case). To estimate the posterior distributions of the SPMe parameters one must evaluate the likelihood term $p(D|\theta)$ through forward simulation of the model and comparison with experimental data (or simulated ground truth in this case). For this purpose, spatial discretisation of the SPMe dynamics (2.12), (2.13) was done by pseudospectral methods using Chebyshev orthogonal collocation [136], allowing for a relatively low-dimensional spatial grid in comparison to simple finite differences. The discretisation scheme for the SPMe dynamics for the solid phase and the electrolyte is the same as that of Bizeray et al. [49, 137], using radial symmetry in the negative and positive particle to further reduce dimensionality. Here, particles contain 3 collocation nodes each, and the electrolyte contains 3 collocation nodes in each of the electrode subdomains and 2 in the separator. To obtain the electrode averaged quantities in the negative and positive electrode subdomains in the electrolyte, numerical integration (Clenshaw-Curtis quadrature) was used [136, 138]. As both the solid-phase and electrolyte diffusivities were assumed independent from lithium concentration, spatial discretisation resulted in a linear state-space system with nonlinear output equation due to Butler-Volmer kinetics in the reaction overpotential and the open-circuit potential. Overall, the system is described by

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) \tag{2.15a}$$

$$V(t) = h(\mathbf{C}\mathbf{x}(t), \mathbf{D}u(t)), \tag{2.15b}$$

where $\mathbf{A} \in \mathbb{R}^{9\times 9}$, $\mathbf{B} \in \mathbb{R}^{9\times 1}$, $\mathbf{C} \in \mathbb{R}^{4\times 9}$, $\mathbf{D} \in \mathbb{R}^{4\times 1}$, $u(t)$ is the applied current and $h(\mathbf{C}\mathbf{x}(t), \mathbf{D}u(t))$ is given by (2.14). The linear function inside h returns the electrode-averaged surface concentrations and electrolyte concentrations in the electrode subdomains.

Parameter	Units	Description	Ω_n	Ω_s	Ω_p
ε_k	-	Electrolyte volume fraction	0.3	1	0.3
$c_{k,max}$	mol/m^3	Maximum lithium concentration	2.4983×10^4	-	5.1218×10^4
σ_k	S/m	Solid conductivity	100	-	10
$D_{s,k}$	m^2/s	Electrode diffusivity	3.9×10^{-14}	-	1×10^{-13}
R_k	μm	Particle radius	10	-	10
a_k	μm^{-1}	Electrode surface area density	0.18	-	0.15
m_k	$(A/m^2)(m^3/mol)^{1.5}$	Reaction rate	2×10^{-5}	-	6×10^{-7}
L_k	μm	Thickness	100	25	100
$U_{k,ref}$	V	Reference OCP	0.18	-	3.94
$c_{e,typ}$	mol/m^3	Typical lithium-ion concentration in electrolyte		1×10^3	
$D_{e,typ}$	m^2/s	Typical electrolyte diffusivity		2.8×10^{-10}	
$\kappa_{e,typ}$	S/m	Typical electrolyte conductivity		1.1	
F	C/mol	Faraday's constant		96485	
R	$J/(mol\,K)$	Universal gas constant		8.314472	
T	K	Temperature		298.15	
b	-	Bruggeman coefficient		1.5	
t^+	-	Transference number		0.4	
I_{typ}	A/m^2	Typical current density		24 (1 C)	

Table 2.1 Model parameters with values taken from Moura et al. [140].

2.3 Parameter estimation of the SPMe

To perform a Bayesian parameter estimation and identifiability analysis of the SPMe, a subset of model parameters was chosen to be analysed in a series of simulations. It is well known that even the simpler SPM, without including electrolyte dynamics, is over-parameterised, i.e. not all parameters may be independently identified from input-output data [54]. Here, the subset of parameters under investigation was chosen to illustrate the limitations of identifiability due to the excitation signal, an issue that is relevant when using reduced-order physics-based models in real-world settings. For each excitation signal, a ground truth input-output dataset was first generated using a known parameter set (Table 2.1) and open-circuit potential functions [139].

In total, 12 excitation signals were considered. Illustrated in Fig. 2.2, the first 11 consisted of applying a zero bias multi-harmonic sinusoidal pulse for a period of 1000 s, containing frequencies of 1 mHz, 10 mHz and 100 mHz and a peak-to-peak amplitude of 50 mV, giving a quasi-steady SOC. This time-limited signal was applied as the model input at 11 discrete SOC points. In contrast, the 12th case used a single-frequency sinusoidal pulse biased by 1C discharge, giving a broad excursion in SOC (from 100%→5%) over the excitation. In each case, identically and independently normally distributed (i.i.d.) noise with variance σ_n^2 was

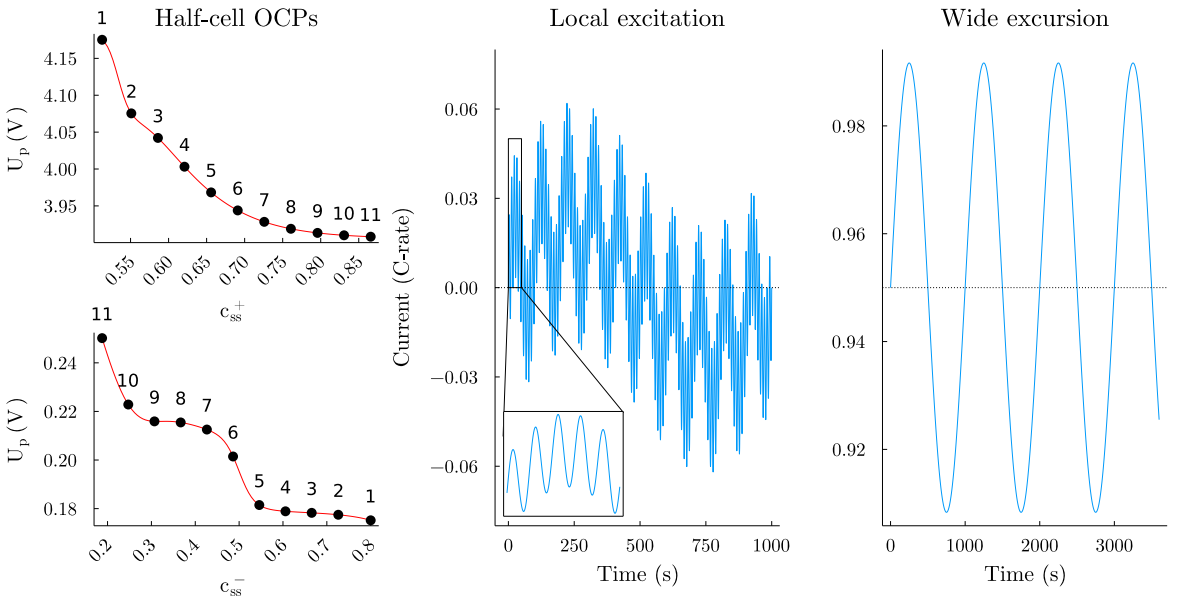


Fig. 2.2 Excitation signals for parameter identification: In the first 11 cases (left and middle), a zero-bias multi-harmonic sinusoidal pulse was applied at 11 discrete points in SOC for a duration of 1000 s. In the final case, a 1C biased single-frequency pulse applied to give wide excursion in SOC (right).

added to the voltage output, where 2σ corresponded to approximately 1% of the amplitude of the voltage response.

Following the generation of the ground-truth data, a subset of model parameters was chosen to be identified. Four model parameters were chosen: solid phase diffusivities D_n, D_p, electrolyte diffusivity D_e and transference number t^+. Furthermore, the noise variance σ_n^2 was estimated in each case.

The Bayesian framework allows for the incorporation of existing knowledge of model parameters by defining prior distributions over them—this is the equivalent to the use of regularization in the frequentist framework; prior distributions were specified as

$$p(\theta) = \begin{cases} \Gamma(\alpha,\beta), & \theta \in \{\widetilde{D}_\mathrm{n}, \widetilde{D}_\mathrm{p}, \widetilde{D}_\mathrm{e}\}, \\ \mathrm{Beta}(\alpha,\beta), & \theta \in \{t^+\}, \end{cases} \tag{2.16}$$

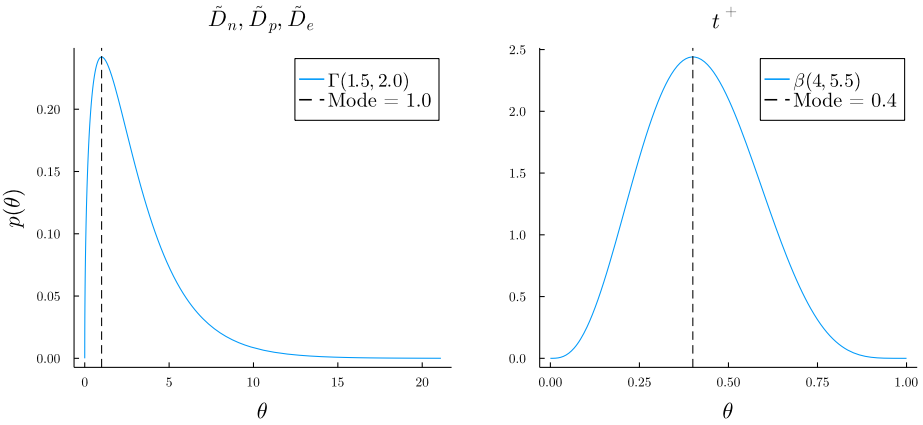


Fig. 2.3 Prior parameter distributions for each of the diffusivities was chosen to be a Gamma distribution where the mode was equal to 1.0 (using re-scaled values $\widetilde{D}_\mathrm{n}$, $\widetilde{D}_\mathrm{p}$, $\widetilde{D}_\mathrm{p}$). The transference number prior was chosen as a Beta distribution, where the mode was again equal to the ground-truth value and 80% of the probability mass lay between 0.2 and 0.6.

where Γ is the Gamma distribution. Schematically shown in Fig. 2.3, these reflect both the physical constraints and reasonable prior knowledge of the parameters. Each of the diffusivities was constrained to be positive, and the Γ parameters α, β were set to 1.5 and 2.0 respectively to give a mode of 1.0 on a rescaled basis where each diffusivity was scaled to a unit range (so that the ground truth for $\widetilde{D}_\mathrm{n}, \widetilde{D}_\mathrm{p}, \widetilde{D}_\mathrm{p}$ equalled 3.9, 1.0 and 2.8 respectively). To enforce positivity, an equally valid approach would have been to use a log-normal prior over diffusivities. Here, the more straight-forward interpretability and parameterisation of the Γ distribution in real parameter space was preferred. For the transference number t^+ a Beta distribution was defined so that 80% of the probability mass lay in the range [0.2,0.6] and the mode was again equal to the ground truth. The prior distribution of the noise variance σ_n was uniform. Given the SPMe parameterised by the vector $\theta = [D_\mathrm{n}\ D_\mathrm{p}\ D_\mathrm{e}\ t^+]$, the voltage response of the model (2.15) is denoted by $f(\theta)$. As normally distributed noise was added to the ground-truth voltage response, the likelihood function is Gaussian, so the energy function is given by

$$\phi(\theta) = -\log p(\theta) - \log p(y|\theta) = -\log p(\theta) + \frac{n}{2}\log 2\pi\sigma_\mathrm{n}^2 + \sum_{i=1}^{n}\frac{(y - f(\theta))^2}{2\sigma_\mathrm{n}^2}, \qquad (2.17)$$

where y is the observed voltage, n the number of data points, and $p(\theta)$ the parameter vector prior (2.16). To explore the posterior parameter distribution, the NUTS algorithm was run for 10,000 iterations with a 1,000 iteration burn-in period, and 4 chains were run in parallel for each of the 12 scenarios, with random initialisation. To guarantee the differentiability of $\phi(\theta)$ (which is required for HMC, see Section 2.1.2), a nonlinear monotonic transformation was applied to the samples θ before passing them to the forward model, using the logistic function, so that $0 \leq t^+ \leq 1$ and $0.1 \leq D_\mathrm{n} \leq 100$ for $n \in \{n, p, e\}$. The non-zero lower bound for the diffusivities was necessary to guarantee numerical stability of the forward SPMe simulation, where the effect is to introduce lower and upper bounds on the prior. As mentioned above, using a log-normal distribution over the diffusivities would naturally enforce positivity, but the logistic transformation would still have been necessary to address the same stability issue. Point estimates from HMC were retrieved by taking the minimum mean-squared estimates (MMSE) from the posterior (i.e. by calculating the posterior mean). For an easier interpretation of the results, diffusional timescales $R_\mathrm{n}^2/D_\mathrm{n}$, $R_\mathrm{p}^2/D_\mathrm{p}$ and L^2/D_e are considered rather than diffusivity directly, where $R_{n,p}$ and L are considered known. For comparison, in each case, MAP estimates for the parameters were also found by minimising the energy function using the gradient-based Broyden–Fletcher–Goldfarb–Shanno (BFGS) minimiser in Julia's `Optim.jl` [141] and the Laplace approximation (2.3) calculated by evaluating the Hessian matrix at the optimum value using automatic differentiation [142].

2.3.1 Results

Parameter identifiability

Both the identifiability and the parameter point estimates of $\theta = [R_\mathrm{n}^2/D_\mathrm{n}\ R_\mathrm{p}^2/D_\mathrm{p}\ L^2/D_\mathrm{e}\ t^+]$ were inferred from the posterior distributions estimated by the NUTS algorithm and the MAP + Laplace method. To illustrate, both the marginal distributions of each single parameter and the pairwise joint posteriors were considered. First, Fig. 2.4 shows the individual marginal

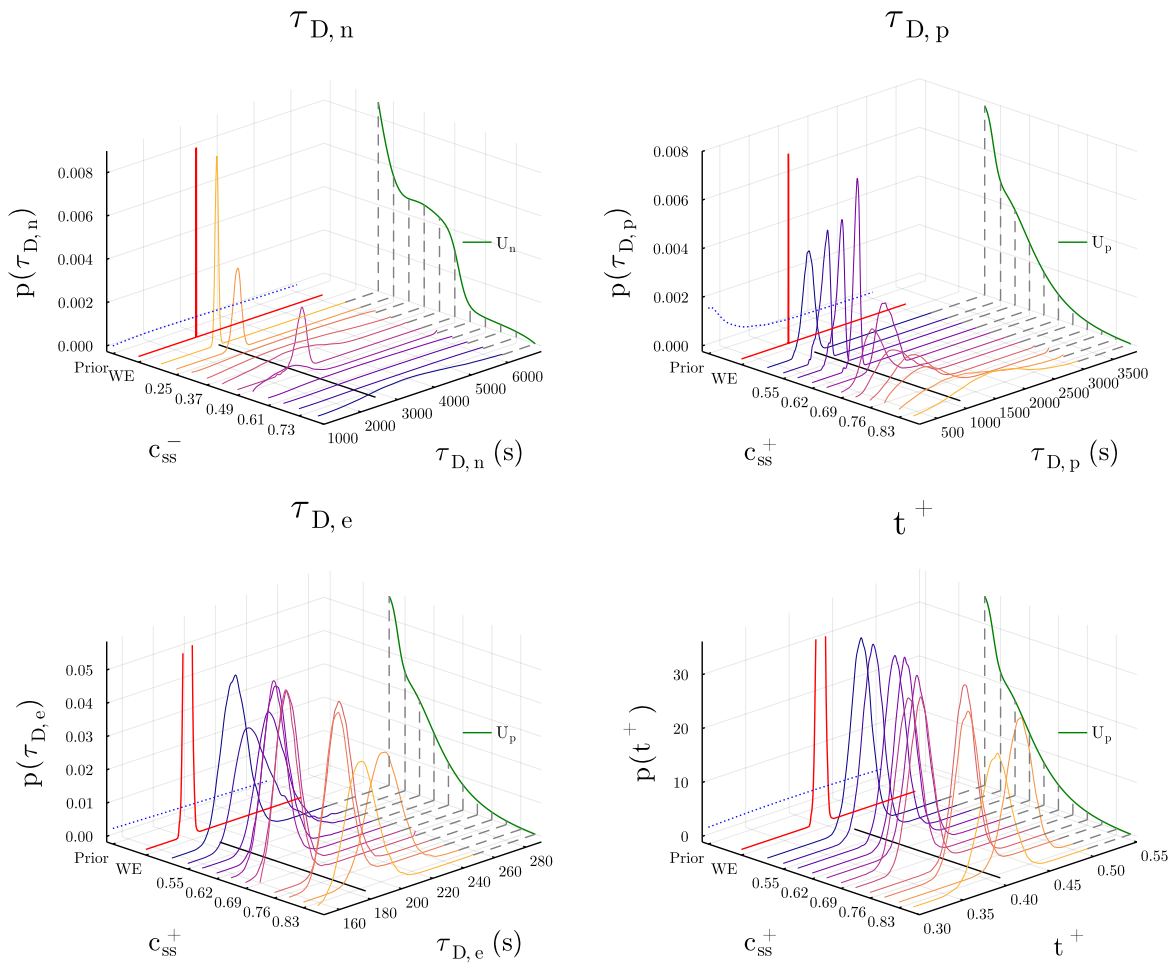


Fig. 2.4 Kernel density estimates of marginal posteriors for the four model parameters for each estimation scenario, with the specific point of excitation on half-cell OCPs indicated by black dashed lines.

posteriors from NUTS of each parameter in each excitation case. It is clear from the width of the distributions that the identifiability of the parameter vector is dependent on the quasi-steady surface concentrations of the particles at each point of excitation. Specifically, the variances of $R_{\mathrm{n}}^2/D_{\mathrm{n}}$ and $R_{\mathrm{p}}^2/D_{\mathrm{p}}$ are related to their respective half-cell OCP gradients at the point of excitation, which is apparent from Fig. 2.4. To quantify the effect of OCP gradients on the identifiability of solid-state diffusivities, the relative entropy of the marginal posteriors of diffusional time scales may be used. Relative entropy, defined by the Kullback-Leibler

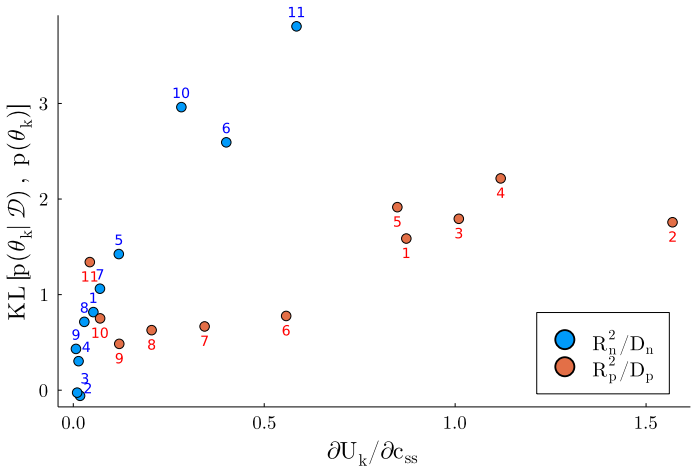


Fig. 2.5 Dependence of the identifiability of solid-phase diffusivity, expressed as the relative entropy of the marginal posterior distributions estimated with NUTS vs. their priors, as a function of half-cell OCP gradient for positive and negative electrodes. Numbers denote local excitation case.

divergence between the prior and (marginal) posterior,

$$\mathrm{KL}[p(\theta|\mathcal{D}) \;,\; p(\theta)] = \int_{\theta} p(\theta|\mathcal{D}) \log \left[\frac{p(\theta|\mathcal{D})}{p(\theta)} \right] \mathrm{d}\theta, \tag{2.18}$$

measures the information gain [143] going from the prior to posterior. The larger the value, the more is learned from data. Fig. 2.5 shows the relative entropy for the two diffusional timescales, calculated using kernel density estimates of the marginal posterior $p(\theta|\mathcal{D})$, as a function of half-cell OCP gradient. The positive relation to identifiability is evident for both electrodes. The dependency on $\partial \mathrm{U}/\partial c_{\mathrm{ss}}$ is stronger for the negative electrode, which is likely due to its longer timescale, making the voltage output more sensitive to this parameter over the length of the excitation. This result agrees with the analysis by Bizeray et al. [54] of solid phase diffusivities for the SPM—as either of the half-cell OCP gradients tends to zero, the voltage response to changes in surface concentration for that electrode drops to zero, rendering the diffusivity unidentifiable. For the other two parameters, D_{e}, t^{+}, there is no such effect. However, in all cases, the signal that results in a wide SOC excursion, corresponding to a full

		1	2	3	4	5	6	7	8	9	10	11	Global
	c_{ss}^{-}	0.8	0.73	0.67	0.61	0.55	0.49	0.43	0.37	0.31	0.25	0.19	
	c_{ss}^{+}	0.51	0.55	0.59	0.62	0.66	0.69	0.73	0.76	0.8	0.83	0.87	
τ_n (2600)	μ_{NUTS}	3600	7500	7500	4500	1900	2700	2800	3300	4200	2600	2500	2600
	σ_{NUTS}	1000	4300	4900	2300	510	160	750	1200	1700	110	45.0	2.6
	μ_{MAP}	3700	7400	6100	4700	2100	2700	3000	3200	4200	2600	2500	2600
	σ_{MAP}	1000	4800	4400	2600	510	130	730	1200	2000	110	45.0	2.7
τ_p (1000)	μ_{NUTS}	900	900	900	950	1200	780	870	860	820	1200	1500	1000
	σ_{NUTS}	93.0	84.0	83.0	55.0	140	200	260	280	330	560	520	1.0
	μ_{MAP}	890	900	930	950	1100	750	810	900	890	1300	1700	1000
	σ_{MAP}	94.0	91.0	75.0	59.0	140	170	250	290	370	570	500	1.1
τ_e (180)	μ_{NUTS}	190	190	200	190	180	180	170	190	180	200	180	180
	σ_{NUTS}	8.2	12.0	11.0	8.0	7.5	7.4	7.1	7.9	7.9	11.0	10.0	1.8
	μ_{MAP}	190	190	190	190	180	180	170	190	180	200	180	180
	σ_{MAP}	8.5	14.0	9.8	8.1	7.6	7.3	7.3	7.9	8.0	11.0	10.0	1.9
t^+ (0.4)	μ_{NUTS}	0.41	0.4	0.41	0.41	0.4	0.38	0.37	0.41	0.4	0.44	0.39	0.39
	σ_{NUTS}	0.011	0.011	0.012	0.012	0.013	0.013	0.013	0.012	0.014	0.015	0.017	0.0036
	μ_{MAP}	0.41	0.4	0.41	0.41	0.4	0.38	0.38	0.41	0.4	0.44	0.39	0.39
	σ_{MAP}	0.011	0.012	0.012	0.012	0.013	0.013	0.013	0.012	0.013	0.015	0.017	0.0038
σ_n^2 (2.5e-4)	μ_{NUTS}	2.5e-4	2.5e-4	2.5e-4	2.5e-4	2.5e-4	2.5e-4	2.5e-4	2.5e-4	2.4e-4	2.5e-4	2.5e-4	2.5e-4
	σ_{NUTS}	2.8e-6	2.8e-6	2.9e-6	2.8e-6	2.8e-6	2.8e-6	2.8e-6	2.8e-6	2.7e-6	2.8e-6	2.8e-6	2.7e-6
	μ_{MAP}	2.5e-4	2.5e-4	2.5e-4	2.5e-4	2.5e-4	2.5e-4	2.5e-4	2.5e-4	2.4e-4	2.5e-4	2.5e-4	2.5e-4
	σ_{MAP}	2.8e-6	2.8e-6	2.8e-6	2.8e-6	2.8e-6	2.8e-6	2.8e-6	2.8e-6	2.7e-6	2.8e-6	2.8e-6	2.9e-6

Table 2.2 NUTS and MAP + Laplace estimates for chosen SPMe parameters for each excitation signals. Numbers in brackets in the first column indicate true values.

discharge in this case, identifies the parameters (which are assumed constant over SOC) with high precision, as shown by the red lines indexed "WE" in Fig. 2.4.

Comparison of NUTS and MAP + Laplace estimates

Table 2.2 summarises the MMSE estimates for each case, and the variances calculated from the NUTS distributions, together with the corresponding MAP estimates and Laplace approximations for the parameter variances. The two sets of estimates are in close agreement in most cases both in terms of mean and variance, which is also illustrated in Fig. 2.6. These show pairwise joint distributions $p(\tau_{D,n}, \tau_{D,p})$ in the top row and $p(\tau_{D,e}, t^+)$ in the bottom row, for a subset of local points (points 2, 6, 11 in Fig. 2.2) and the 'wide excursion' in the right column.

However, the MAP + Laplace approximation approach applied to the estimation of SPMe parameters has shortcomings. Illustrated in the joint posterior plot of $p(\tau_{D,n}, \tau_{D,p})$ in Fig. 2.7, it can be seen that the Laplace approximation, being a symmetric Gaussian approximation of

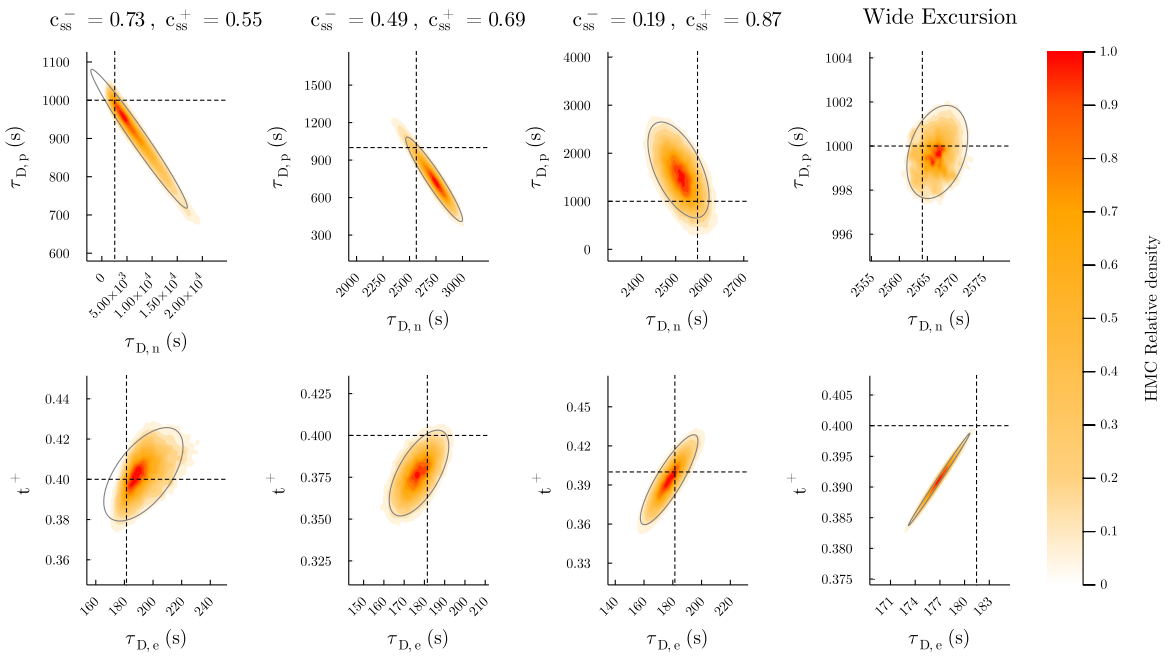


Fig. 2.6 NUTS Kernel density estimates of pairwise joint posteriors p(D_n,D_p) and p(D_e,t^+) for 3 local excitation cases and the wide excursion case. Grey contours indicate the 2σ bounds calculated using the MAP+Laplace approximation.

the posterior, sometimes allocates probability mass to physically infeasible space (i.e. where $\tau_{D,n} < 0$)—a result of the Laplace approximation being calculated in the neighbourhood around the MAP estimate. The HMC estimate does not suffer from this problem. Furthermore, the identifiability problem is not invariant to (nonlinear) transformations in the parameters. For example, the joint distribution of $p(D_n, D_p)$ in Fig. 2.7 shows the effect of applying the MAP + Laplace and NUTS estimators directly on diffusivities rather than diffusional timescales. In the case of solid-state diffusivities, local estimation cases show strongly non-elliptical contours of pairwise joint posteriors, which are not captured by the MAP + Laplace method, again giving a non-negligible probability mass at $D_n < 0$.

Reliability of NUTS estimates

To verify the reliability of NUTS estimates, 4 chains were initialised at random in each case. From these four chains, the MC error (2.6) can be empirically calculated. Specifically, the

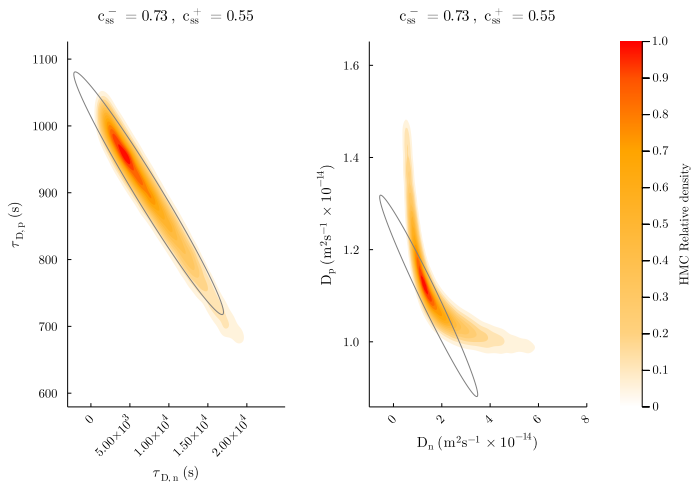


Fig. 2.7 Effect of parameter transformation on posterior and the Gaussian approximation. Directly estimating solid state diffusivities results in non-elliptical contours of the posterior, where the Laplace approximation fails. In both cases, the variance implied at the MAP estimates is such that Gaussian approximations have probability mass in physically unfeasible locations.

'initial monotone sequence estimator' [144] is used, whereby the ESS (2.7) and MC errors are calculated by only using a window of autocorrelations for lags in the chain as long as they are both positive and decreasing. Furthermore, the scale reduction factor $\hat{R}$ [145] may also be calculated for each parameter, where $\hat{R}$ is effectively given by the ratio of variances between chains to variances within the chains. This describes the extent to which additional iterations in each chain would improve the estimate of the target distribution. As $N \to \infty$, $\hat{R} \to 1$, indicating that adding additional iterations to the chains would not improve the estimate of the target distribution. These statistics across the four chains in each estimation case are shown in Table 2.3. The cases where $\hat{R} > 1.01$ are highlighted in red. The convergence statistic in most cases implies that all chains return the same distribution (all available in Appendix A). The notable exception is one of the chains in the wide excursion case, where the target distribution did not sample efficiently, as shown in Fig. A.3. In general, the MC error (that is, the error in the mean estimate of each parameter in the sense of the central limit theorem) is low compared to the estimates of parameter variance in Table 2.2, indicating that NUTS estimates are accurate. In case 2 however, particularly for $\tau_{D,n}$, the ESS is only 271 (calculated over 4×9,000=36,000

		1	2	3	4	5	6	7	8	9	10	11	Global
τ_n	ESS	3288	271	538	1028	2231	1529	1899	3576	7839	8965	12668	373
	σ_{MC}	18.0	200.0	180.0	63.0	11.0	4.1	17.0	20.0	19.0	1.1	0.4	0.089
τ_p	ESS	3347	270	578	1097	2264	1539	1860	3282	6192	6866	9603	564
	σ_{MC}	1.6	3.8	2.9	1.5	2.9	5.1	6.0	4.9	4.2	6.6	5.3	0.028
τ_e	ESS	7155	360	864	2528	7437	6665	5338	9525	19073	15850	17604	3013
	σ_{MC}	0.098	0.5	0.33	0.15	0.087	0.096	0.099	0.082	0.062	0.09	0.075	0.033
t^+	ESS	10880	3039	4206	11357	8401	7711	7315	10538	15232	17590	16980	3080
	σ_{MC}	1e-4	0.0002	1e-48	1e-4	1.5e-4	1.6e-4	1.6e-4	1.3e-4	1.2e-4	1.2e-4	1.3e-4	6.7e-5
σ_n	ESS	10525	3446	4926	10829	7310	11112	7333	10857	20202	15591	16894	846
	σ_{MC}	2.9e-8	4.8e-8	4.3e-8	2.9e-8	3.3e-8	3.0e-8	3.1e-8	2.7e-8	1.9e-8	2.4e-8	2.4e-8	6.3e-8

Table 2.3 Effective sample sizes and MC standard errors of NUTS sampler applied to SPMe parameter estimation problem. Red cells indicate $\hat{R} > 1.01$

iterations), indicating that the HMC estimates are unreliable. In this case, both estimates (Map + Laplace, NUTS) of the parameter variance are also very high, as shown in Table 2.2.

Effect of parameter uncertainty on forward predictions

To gauge the effect of parameter uncertainty in voltage prediction, Fig. 2.8 shows predictive voltage distributions over a constant current 95% depth-of-discharge cycle for three different cases. Compared to voltage prediction using the parameter prior, $p(f(\theta))$, the posterior in the local excitation case c=2, $p(f(\theta)|\mathcal{D}_{c=2})$ shows the effect of the lack of identifiability of the solid state diffusivity of the negative electrode. The credible intervals widen substantially in areas where the negative half-cell OCP is steepest. In comparison to the local excitation case, the wide excursion excitation accurately identifies all parameters, so the credible intervals for voltage prediction are narrow for discharge throughout.

2.4 Discussion

A Bayesian approach was applied to estimate a subset of model parameters for the SPMe, comparing HMC and MAP estimates with the Laplace approximation. It is clear that some parameters of the model are more identifiable than others, and that identifiability for some parameters depends on the type of excitation signal. The relative degree of identifiability

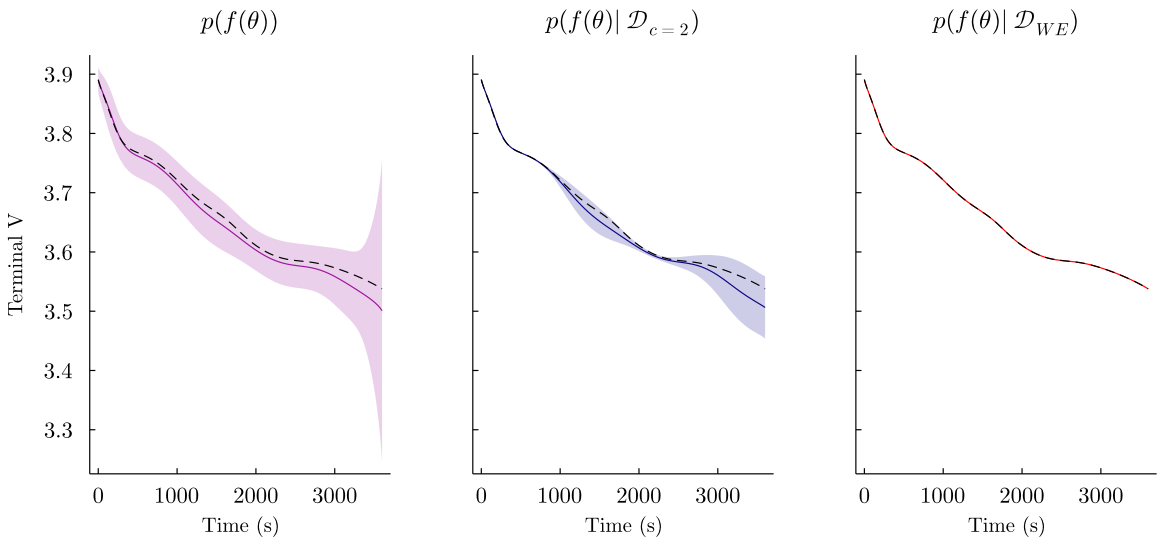


Fig. 2.8 Predictive voltage distributions showing the mean $\pm 2\sigma$ using 1,000 samples from parameter prior and posteriors generated from NUTS in two of the estimation cases.

may be quantified by the relative entropy, given by the KL divergence between the prior and posterior distributions of the parameters. The analysis above was, in a sense, a 'best case' scenario, as all model parameters apart from the ones singled out to perform the analysis were considered known, and initial conditions and half-cell OCPs were also known.

Under real-world conditions, the weak identifiability of model parameters and dependence on excitation creates a two-fold problem. Firstly, if parameters are a function of internal states, such as the state of charge and temperature [57], then limited identifiability means that one cannot simply 'connect the dots' on the basis of local measurements and fits. Moreover, real-world operation conditions do not guarantee that operating points where the parameters are most identifiable are visited consistently throughout the battery life. Using naive approaches, these factors together will lead to very noisy estimates of model parameters over time for batteries.

While identifying parameters as functions of operating conditions and internal states is in principle possible, doing this with physics-based Li-ion models in real-world conditions in practice is not. The most significant hurdle here is the lack of observability of battery states in the system dynamics. As (commercial) cells do not contain a reference electrode, state

observers can only use error feedback from the terminal voltage (and temperature), which is not sufficient to indepenently identify surface concentrations on the positive and negative electrodes.

Theoretically, accurate knowledge of the total lithium inventory allows coupling of half-cell surface concentrations [55], giving a unique solution. However, since cyclable lithium is 'lost' in the ageing process [22], estimating the lithium inventory is also not trivial. Furthermore, even if batteries in a given application routinely have a long enough resting segment and the lithium inventory is known, the effect of temperature on the open-circuit voltage through entropic effects [146] and voltage hysteresis [147] add additional complication.

Given these difficulties, simpler dynamic models are needed for the real-world diagnosis of SOH. However, simpler models, such as ECMs, will naturally exhibit more variation in their parameters due to changes in states and operating conditions. Given that the identifiability problem of initial conditions is not as severe for ECMs (as the OCV is expressed at the terminal level as a function of the state of charge), a Bayesian approach to identifying parameter dependencies as functions of states and operating conditions is more feasible with ECMs.

One principled approach to estimate parameters as functions rather than point estimates is to use Gaussian process regression. Analogous to the method above, the method imposes a prior over 'parameter functions' directly, and observed input data is used to condition the prior to yield a posterior, which describes parameter behaviour over the inputs in a non-parametric manner. Moreover, the GP method also answers the identifiability problem in a similar way, yielding lower precision estimates in cases where there is less informative data available. This method may be applied to any type of model. It will be shown subsequently that applying GPR in this manner to ECMs can both improve their accuracy and condition the parameter estimation problem so as to yield a more reliable state of health diagnosis in real-world operating conditions.

Chapter 3

State of health diagnosis with field data

The previous chapter highlighted the difficulties in applying physics-based models to SOH diagnosis in real-world applications. The major limitations arise from poor parameter identifiability and observability of battery states from standard terminal voltage and current measurements over time.

Due to these challenges, equivalent circuit models are more applicable to real-world SOH diagnosis through grey-box parameter tracking. However, owing to their simplicity, they lose accuracy if operating conditions are substantially different from those when the model parameters were originally fitted. This is reflected in the variation of circuit parameters when, e.g., temperature and/or state of charge is changed [41]. Consequently, applying ECMs to real-world data SOH diagnosis with large fleets of batteries will result in unstable estimates over the lifetime of each battery and will not be consistent across different batteries. To mitigate this effect, work has been done to account for some of the dependencies on operating conditions [42, 43]. However, taking a first-principles parametric approach to assess all the dependencies of circuit elements on operating conditions is time-consuming and difficult, as the number of parameters required is high and all functional forms must be determined *a priori*. This parameterisation has to be performed separately for each cell chemistry, and as the cell degrades, exacerbating the situation. In addition, the parameterisation will depend on

the ageing mechanism, making a full experimental determination of circuit parameters as a function of all inputs (including degradation) very costly.

To address these issues, the framework proposed in this chapter uses a Bayesian non-parametric approach to model ECM parameters as functions of operating conditions and lifetime in a real-world application. Specifically, a computationally efficient implementation of Gaussian process regression (GPR) is formulated to learn functions describing circuit parameters directly from operating data. In essence, this combines model-driven and data-driven SOH diagnosis, whereby the model-driven part comes from the structure provided by the ECM and the data-driven part completes the picture by making the circuit more adaptable to changing operating conditions. Using this approach, the system is robust to changes in operating conditions and gaps in measured data as well as numerical ill-conditioning. Also, the framework is easily transferred between chemistries and can be applied from the cell level to the pack level, because the only laboratory-based measurement required is the open-circuit voltage curve. The rest of this chapter describes background information about Gaussian process regression and classification, then gives an analysis of a large battery field operation dataset where it is demonstrated that GPs may be used both to infer smooth health estimates and to diagnose/predict end of life.

3.1 Gaussian Processes

A Gaussian process is defined as a collection of random variables where any finite subset has a joint Gaussian distribution [148]. It is completely described by the mean and covariance of a multivariate Gaussian distribution, which for a GP with input $\mathbf{x}$ are defined as

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] \\ k(\mathbf{x},\mathbf{x}') &= \mathbb{E}[(f(\mathbf{x})-m(\mathbf{x}))\left(f(\mathbf{x}')-m(\mathbf{x}')\right)], \end{aligned} \tag{3.1}$$

and the Gaussian process $f(\mathbf{x})$ is denoted by

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right). \tag{3.2}$$

If we consider $f(\mathbf{x})$ to describe a function over inputs $\mathbf{x}$, a GP then defines a distribution over functions. Without loss of generality, the mean function $m(\mathbf{x})$ can be assumed to be 0, so that the GP is completely parameterised by its covariance given by a *kernel function*, $k(\mathbf{x}, \mathbf{x}')$.

A brief note on notation

The Gaussian process literature tends to use varying conventions to denote vectors and matrices in training and test sets. In the following sections, outputs are assumed scalars, where n observations are represented by column vector $\mathbf{y} \in \mathbb{R}^n$ indexed by i. The input for each y_i is a vector $\mathbf{x}_\mathrm{i} \in \mathbb{R}^D$. The training input set is denoted by $\mathbf{X} \in \mathbb{R}^{n \times D} = [\mathbf{x}_1\ \mathbf{x}_2\ \ldots\ \mathbf{x}_\mathrm{n}]$, the test input set by $\mathbf{X}_*$ and target output $\mathbf{y}$ always refers to the training set. In general, lower-case bold symbols denote vectors, upper-case bold symbols matrices, and italic symbols either scalars or functions, where the difference between should be clear from the context. Kernel functions k are usually defined in terms of an input pair $\mathbf{x}$, $\mathbf{x}'$ returning a scalar. In practical use, when applied to (arbitrary) pairs of input matrices $\mathbf{X}_0 \in \mathbb{R}^{n \times D}$, $\mathbf{X}_1 \in \mathbb{R}^{m \times D}$, they return a $n \times m$ matrix by applying the function to all combinations of rows in the inputs. When applied to the training input pair $\mathbf{X}$, $\mathbf{X}'$ the result is the $\mathbb{R}^{n \times n}$ Gramian matrix, denoted as $\mathbf{K}$.

3.1.1 Gaussian process regression

Using Bayes' rule, a prior distribution over functions $f(\mathbf{x})$ may be updated with observed data to give the corresponding posterior distribution. This is the principle behind Gaussian process regression, a supervised machine learning technique, where the target variable y_i is modelled

by a Gaussian process such that

$$y_i = f(\mathbf{x}_i) + \varepsilon_i \ , \ \varepsilon_i \sim \mathcal{N}(0, \sigma_n^2), \tag{3.3}$$

where error ε is identically and independently distributed Gaussian noise. Inference with GPR consists of updating the prior distribution over $f(\mathbf{x})$ using input-output training data $(\mathbf{X}, \mathbf{y})$ to produce a posterior-predictive distribution, which is also Gaussian. By applying Bayes' theorem, the posterior-predictive distribution of function f_* for a given set of test points $\mathbf{X}_*$ and training data $(\mathbf{X}, \mathbf{y})$ is obtained by integrating over the training posterior $\mathbf{f}$ (also called the posterior over latent variables in the literature) [148],

$$\begin{aligned} p(f_*|\mathbf{X}_*, \mathbf{y}, \mathbf{X}) &= \int p(f_*|\mathbf{X}, \mathbf{X}_*, \mathbf{f}) p(\mathbf{f}|\mathbf{X}, \mathbf{y}) \, \mathrm{d}\mathbf{f} \\ &= \int p(f_*|\mathbf{X}, \mathbf{X}_*, \mathbf{f}) \frac{p(\mathbf{y}|\mathbf{f}, \mathbf{X}) p(\mathbf{f}|\mathbf{X})}{p(\mathbf{y}|\mathbf{X})} \, \mathrm{d}\mathbf{f}, \end{aligned} \tag{3.4}$$

which has a closed-form solution as long as the likelihood function $p(\mathbf{y}|\mathbf{f}, \mathbf{X})$ is normal (i.e., error ε is Gaussian). In this case, the predictive mean and covariance of f_* are given by

$$\mathbb{E}[f_*] = k(\mathbf{X}_*, \mathbf{X}) \left[k(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I} \right]^{-1} \mathbf{y}, \tag{3.5a}$$

$$\mathbb{V}[f_*] = k(\mathbf{X}_*, \mathbf{X}_*) - k(\mathbf{X}_*, \mathbf{X}) \left[k(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I} \right]^{-1} k(\mathbf{X}, \mathbf{X}_*), \tag{3.5b}$$

where $\mathbf{I}$ denotes the identity matrix. In other words, the predictive mean is a linear combination of the training observations $\mathbf{y}$, where the weighting is determined by the kernel function k. Equation (3.5b) shows how the training data conditions the variance—the first term on the RHS is the prior (co)variance, and the second denotes the change in variance from the training data. The posterior variance is a function of the distance between the test and training inputs. This is shown schematically in Fig. 3.1.

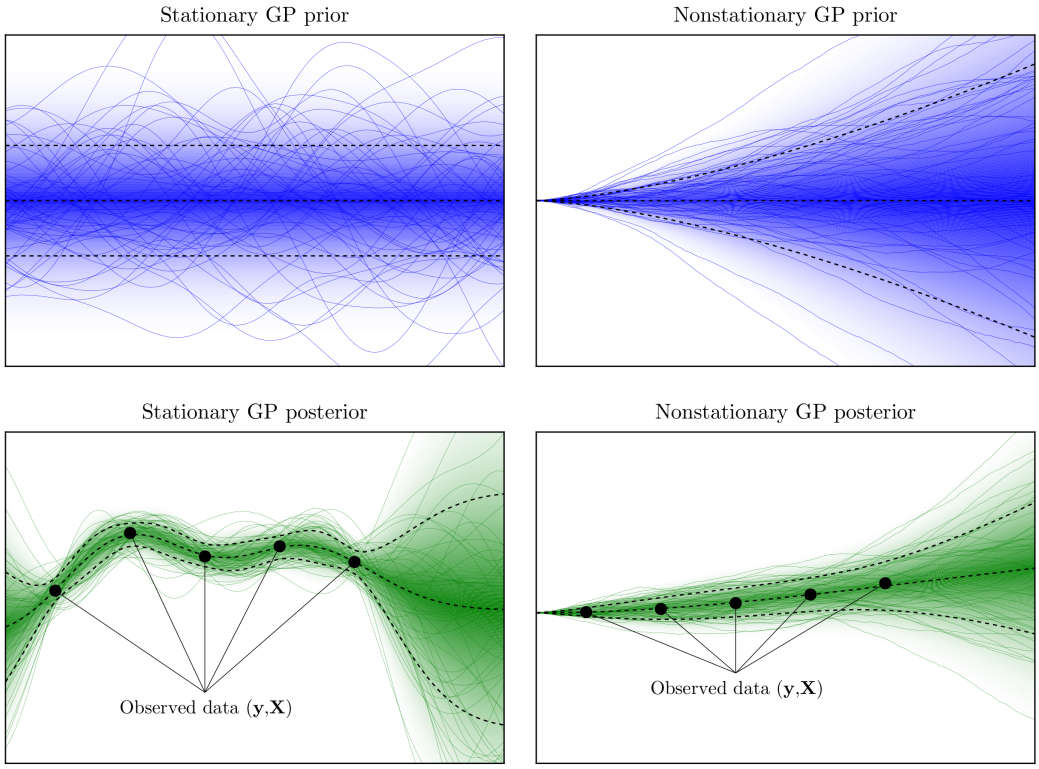


Fig. 3.1 Random draws from prior and posterior-predictive distributions of Gaussian processes with stationary and nonstationary kernels. In each case, the black dash lines indicate the mean and standard deviation of the distribution. The main difference between stationary and nonstationary processes is in extrapolating the posterior predictive. In the stationary case, the mean reverts to the prior mean, whereas in the nonstationary case it does not.

In general, kernel functions $k(.)$ may be categorised as either stationary or nonstationary. Stationary kernels are functions only of $\mathbf{x}-\mathbf{x}'$, that is, the difference between points in the D-dimensional input space. An example is the widely used Matérn family of functions given by

$$k(\mathbf{x},\mathbf{x}') = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{|\mathbf{x}-\mathbf{x}'|}{l} \right)^{\nu} K_\nu \left(\sqrt{2\nu} \frac{|\mathbf{x}-\mathbf{x}'|}{l} \right), \tag{3.6}$$

where σ_f is a scale hyperparameter controlling the prior variance, l the 'length scale' of the prior, Γ the gamma function, K_ν the Bessel function of the second kind and ν a hyperparameter controlling the smoothness of the process. The Matérn kernel is also isotropic, being only a function of the distance (the Euclidean norm) between the input points $|\mathbf{x}-\mathbf{x}'|$, making it

invariant to any rigid motion in the input space [148]. Stationary kernel functions are such that the associated posterior-predictive distribution reverts to the GP prior when predictions are made at test points $\mathbf{X}_*$ 'far' away from the training data $\mathbf{X}$ in terms of the characteristic length scale hyperparameter l of the kernel. In contrast to this, a nonstationary kernel function depends on the absolute values of the inputs; an example is the so-called Wiener velocity kernel [111], given by

$$k(x,x') = \sigma_f^2 \left(\frac{\min^3(x,x')}{3} + |x-x'| \frac{\min^2(x,x')}{2} \right), \tag{3.7}$$

which describes the covariance of an integrated Wiener process over scalar input x (often this is time). Nonstationary kernel functions behave differently when extrapolating—in the case of the Wiener velocity kernel, GPR gives a posterior-predictive distribution that extrapolates linearly along the last estimated trajectory in the training data (see Fig. 3.1), and the predictive variance continuous to increase as a function of distance from the training data.

In this work, stationary and nonstationary kernels are used in combination to describe circuit parameters over different input dimensions where some are naturally better modelled by one or the other depending on the input. In practice, the space of kernel functions is infinite, as any addition or multiplication of valid kernel functions is also a valid kernel function, returning a positive semidefinite covariance matrix.

3.1.2 Gaussian process classification

In addition to the regression problem where the target output $\mathbf{y}$ is continuous and unbound, Gaussian processes can be used in classification tasks where the output variable is categorical. In this work, GPs are used in binary classifiers to predict the probability that batteries will fail in the future. In this case, both inference and prediction become more complicated than for the regression case because the likelihood of observations is no longer Gaussian and the posterior-predictive distribution is a nonlinear function of the underlying GP. The predictive

distribution in the binary case (for single test input $\mathbf{x}_*$) is given by [148]

$$p(y_* = 1|\mathbf{X}, \mathbf{y}, \mathbf{x}_*) = \int \underbrace{\sigma(f_*)}_{\text{'link function'}} p(f_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*)\, \mathrm{d}f_*, \tag{3.8}$$

where $p(f_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*)$ is given by (3.4) and the link function $\sigma(f_*)$ is a sigmoidal function such as the logistic or Gaussian cumulative distribution function. As no closed form exists for (3.8) or the term $p(f_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*)$ (since the likelihood is not Gaussian), an approximation is required. First, the training posterior is approximated as a Gaussian using the Laplace approximation,

$$p(\mathbf{f}|\mathbf{X}, \mathbf{y}) \approx q(\mathbf{f}|\mathbf{X}, \mathbf{y}) \sim \mathcal{N}(\mathbf{f}|\hat{\mathbf{f}}, \Sigma), \tag{3.9}$$

where $\hat{\mathbf{f}}$ denotes the MAP estimate of the training posterior and Σ its approximate covariance. These may be found using Newton's method by initialising $\hat{\mathbf{f}} = 0$ and iterating

$$\hat{\mathbf{f}} = \hat{\mathbf{f}} - [\nabla\nabla\, \phi(\hat{\mathbf{f}})]^{-1} \nabla\phi(\hat{\mathbf{f}}), \tag{3.10}$$

where ϕ is the energy function of the training posterior, that is, $\phi(\mathbf{f}) = -\log p(\mathbf{y}|\mathbf{f}) - \log p(\mathbf{f}|\mathbf{X})$ and the ∇, $\nabla\nabla$ operators denote the gradient and Hessian of ϕ w.r.t. $\mathbf{f}$. Here, a logistic likelihood function is used, resulting in simple analytical expressions for $\nabla\phi(\hat{\mathbf{f}})$ and $\nabla\nabla\phi(\hat{\mathbf{f}})$. Once $\hat{\mathbf{f}}$ is found, Σ is the inverse of the Hessian $\nabla\nabla\, \phi(\hat{\mathbf{f}})$. Given the Gaussian approximation of $q(\mathbf{f}|\mathbf{X}, \mathbf{y})$, the predictive distribution (3.8) can then be written as

$$p(y_* = 1|\mathbf{X}, \mathbf{y}, \mathbf{x}_*) \approx \int \sigma(f_*) q(f_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*)\, \mathrm{d}f_*, \tag{3.11}$$

where $q(.)$ is Gaussian. In this work, the integral (3.11) was evaluated by approximating the logistic function $\sigma(f_*)$ using an expansion of error functions [149], so that the mean of the

RHS in (3.11) is given by [150]

$$\mathbb{E}[p(y_* = 1|\mathbf{X},\mathbf{y},\mathbf{x}_*)] \approx \sum_i \frac{1}{2}\beta_i \left[\operatorname{erf}\left(\gamma_i \sqrt{\frac{\alpha}{\alpha+\lambda_i^2}}\right) + 1\right], \tag{3.12}$$

where $\alpha = \frac{1}{2}\sigma_q^2$, $\gamma = \lambda\mu_q$, and β and λ are coefficients of the error function approximation of the logistic function. The GPC algorithm described above is implemented in the `scikit-learn` python package.

3.1.3 Estimating GP hyperparameters

The kernel functions describing GPs have hyperparameters that determine their prior and posterior distributions. As shown in section 3.1.1, the number and type of hyperparameters vary by kernel. However, the procedure to find optimal hyperparameters remains the same for both regression and classification (although additional approximations are required for the classification case, see the previous section). Bayes' theorem may be applied to find the posterior probability distribution of the hyperparameters given the training data,

$$p(\theta|\mathbf{X},\mathbf{y}) = \frac{p(\theta)\int p(\mathbf{y}|\theta,\mathbf{f},\mathbf{X})p(\mathbf{f}|\mathbf{X},\theta)\,\mathrm{d}\mathbf{f}}{p(\mathbf{y}|\mathbf{X})}, \tag{3.13}$$

where $p(\theta)$ is the prior distribution over the hyperparameters. Similarly to the case of finding parameter estimates in Chapter 2, the right hand side denominator is independent of θ, so using the negative log proportionality yields the energy function,

$$-\log p(\theta|\mathbf{X},\mathbf{y}) \propto \phi(\theta) = -\log p(\theta) - \log \int p(\mathbf{y}|\theta,\mathbf{f},\mathbf{X})p(\mathbf{f}|\mathbf{X},\theta)\,\mathrm{d}\mathbf{f}, \tag{3.14}$$

where the right-hand side for regression and classification is given by

$$\phi(\theta) = -\log p(\theta) + \begin{cases} \frac{1}{2}\left[\mathbf{y}^{\mathrm{T}}\left[k(\mathbf{X},\theta)+\sigma_n^2\mathbf{I}\right]^{-1}\mathbf{y} + \log|k(\mathbf{X},\theta)| + n\log 2\pi\right] \text{ , regression} \\ \frac{1}{2}\left[\hat{\mathbf{f}}^{\mathrm{T}}[k(\mathbf{X},\theta)]^{-1}\hat{\mathbf{f}} + \log\left|\mathbf{I}+\mathbf{W}^{\frac{1}{2}}k(\mathbf{X},\theta)\mathbf{W}^{\frac{1}{2}}\right|\right] + \log p(\mathbf{y}|\hat{\mathbf{f}}) \text{ , classification,} \end{cases} \tag{3.15}$$

where $k(\mathbf{X},\theta)$ denotes the kernel function (returning the $n \times n$ Gramian) parameterised by vector θ, $|.|$ denotes the matrix determinant, $\hat{\mathbf{f}}$ is the MAP estimate of the training posterior in the classification case, $\mathbf{W}$ the Hessian $\nabla\nabla\phi(\hat{\mathbf{f}})$ and n is the number of rows in the training set $\mathbf{y}$. Minimising the energy function $\phi(\theta)$ with respect to θ will yield MAP estimates of the hyperparameters. While MAP estimates are most commonly used, a more comprehensive Bayesian treatment of the hyperparameters would consider the full posterior $p(\theta|\mathbf{X},\mathbf{y})$; in this case computing the posterior-predictive distribution involves marginalising over the hyperparameter posterior,

$$p(f_*|\mathbf{X}_*,\mathbf{y},\mathbf{X}) = \int p(f_*|\mathbf{X}_*,\mathbf{y},\mathbf{X},\theta)p(\theta|\mathbf{X},\mathbf{y})\,\mathrm{d}\theta \tag{3.16}$$

The integral is intractable and may be approximated using a variety of techniques [151, 152]. In this work, however, it is assumed that there is 'enough data' and the hyperparameter posterior distribution has low variance. Also, even with more diffuse hyperparameter posteriors, the posterior-predictive distribution tends to only be substantially different from the type II MAP case if test points are 'far' away from the training set.

3.1.4 Computationally efficient implementation of GPR

The standard approach to fitting a GP to data for either regression or classification is to find the optimal hyperparameters, then use these and the data to compute the GP posterior and make predictions. Both tasks involve the inversion of the $n \times n$ Gramian matrix $\mathbf{K}$ returned by applying kernel function to training inputs (eqns. (3.5), (3.15)). The standard approach of applying the Cholesky decomposition [148] scales computationally as $\mathcal{O}(n^3)$ and can be

prohibitive for very large training sets. Multiple strategies exist to avoid this "big n" problem, such as sparse GP regression [153], reduced rank GP regression [154], approaches relying on special structures in the input data [155] or recursive methods [110]. Recently, it has also been shown that using a graphics processing unit to parallelise computations can yield very fast solutions to the posterior, NLML and its gradient [156].

Recursive methods are well suited for applying GP regression directly to battery telemetry data because they scale as $\mathcal{O}(n)$ in computational time and memory requirements over the time dimension of the data. Telemetry data usually consists of time series temperature, voltage and current measurements that can span several years with a sampling frequency often in the range of 1 Hz, so datasets can contain hundreds of millions of rows of data. An additional advantage of recursive methods in comparison to sparse methods is that they do not require sparsity in the time dimension, which can be important in time-series data when applying GP regression to understand changes in SOH—features such as knee points can make the positioning of inducing points more difficult in standard sparse methods.

Gaussian processes as solutions to stochastic differential equations

The principle of recursive solutions for GPR comes from the fact that Gaussian processes are solutions to linear time-invariant stochastic differential equations (LTI-SDEs) where the driving process is Gaussian noise [157, 110]. This means that the GP regression problem of the type

$$\mathbf{y} = f(\mathbf{x},t) + \varepsilon \ , \ \varepsilon \sim \mathcal{N}(0,\sigma_{\mathrm{n}}^2), \tag{3.17}$$

where f is a Gaussian process over spatial and temporal dimensions, $\mathbf{x}$ and t, may be expressed by a linear dynamic system where [111]

$$\frac{\partial \mathbf{f}(\mathbf{x},t)}{\partial t} = \mathcal{F}\mathbf{f}(\mathbf{x},t) + \mathbf{L}\omega(\mathbf{x},t) \tag{3.18a}$$

$$\mathbf{y}_t = \mathcal{H}\mathbf{f}(\mathbf{x},t) + \varepsilon(t), \tag{3.18b}$$

where the elements of vector $\mathbf{x}$ are the 'spatial' inputs (i.e., non-time dimensions) of the GP, and t is time. Note that the GP $f(\mathbf{x},t)$ in this context is described by the distribution over latent variables, $\mathbf{f}(\mathbf{x},t)$, a vector quantity. The linear transition operator $\mathcal{F}$, vector $\mathbf{L}$, observation operator $\mathcal{H}$, and the properties of the spatially resolved Gaussian noise $\omega(x,t)$ are all linked to the kernel function (and its hyperparameters) describing the GP in the standard batch fitting approach (3.5). For the stationary class of covariance functions (e.g., the Matérn family), the above dynamic system can be derived through the spectral properties of the kernel. Specifically, from the Wiener-Khinchine theorem, the Fourier transform of the kernel function yields the spectral density of the GP over time. By factorising the spectral density in the frequency domain, a stable state-space realisation (i.e. where the poles of the transfer function all lie in the negative half of the complex plane) may be retrieved [158, 66] that describes the GP. While this method does not apply to nonstationary processes, state-space representations for them may also be found through analysis of kernel basis functions [111, 159]. In this work, the kernel functions used separate over $\mathbf{x}$ and t, such that

$$k(\mathbf{x},\mathbf{x}',t,t') = k(\mathbf{x},\mathbf{x}')k(t,t'), \tag{3.19}$$

implying that the GP prior has zero covariance between the two sets of inputs $\mathbf{x}$ and t. As a result, the linear operator $\mathcal{F}$ in (3.18), becomes a matrix $\mathbf{F}$ defined by the hyperparameters of the kernel function.

Filtering distributions as solutions to LTI-SDEs

The standard recursive method used to numerically solve LTI-SDEs with a Gaussian forcing term and measurement noise is the Kalman filter [160], which returns the optimal solution for the marginal mean and covariance of latent state vectors $\mathbf{x}_t$ over time t conditioned on observations $\mathbf{y}_t$. This is equivalent to finding the posterior distribution of a GP [158] in the regression problem (3.3). To evaluate the GP posterior in practice, first a discrete representation of the function $f(\mathbf{x},t)$ over input domain $\mathbf{x}$ is required. To this end, assuming the input $\mathbf{x} \in \mathbb{R}^D$, a set of m points $\{\mathbf{x}_0, \mathbf{x}_1, ...\mathbf{x}_m\}$, is chosen to represent the value of the GP at each location. These then become the states in the linear dynamic system which are propagated through time, described by dynamics of (3.18). The temporal kernel $k(t,t')$ determines the order n of the Markov process describing the GP through time [158]. In other words, representing the system in companion form [161], the kernel function over the time dimension determines the order n of the highest time derivative in the state vector. Using the standard continuous-time relations of the Kalman-Bucy filter [162], the ordinary differential equations for the predictive mean and covariance of $\mathbf{x}_t$ are given by

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = \mathbf{F}\mathbf{x}_t \tag{3.20a}$$

$$\frac{\mathrm{d}\mathbf{P}_t}{\mathrm{d}t} = \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^{\mathrm{T}} + \mathbf{L}\mathbf{q}\mathbf{L}^{\mathrm{T}}, \tag{3.20b}$$

where the state vectors $\mathbf{x}_t \in \mathbb{R}^{(n*m)}$ and $\mathbf{L} \in \mathbb{R}^{(n*m)}$ are given by

$$\begin{aligned} \mathbf{x}_t &= \mathbf{1}_m \otimes \begin{bmatrix} \mathbf{x}_t & \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} & \frac{\mathrm{d}^2\mathbf{x}_t}{\mathrm{d}t^2} & \dots & \frac{\mathrm{d}^n\mathbf{x}_t}{\mathrm{d}t^n} \end{bmatrix}^{\mathrm{T}} \\ \mathbf{L} &= \mathbf{1}_m \otimes \begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix}^{\mathrm{T}}, \end{aligned} \tag{3.21}$$

where $\mathbf{1}_m$ denotes a vector of ones of size m and $\otimes$ is the Kronecker product. The Kronecker relation reflects the idea that the spatial process is propagated at each discrete point of $\mathbf{x}_t$ with

the same Markovian dynamics, which comes from the separability of the kernel function (3.19). The matrices $\mathbf{F}$ and the spectral density $\mathbf{q}$ are therefore given similarly by

$$\begin{aligned}\mathbf{F} &= \mathbf{I}_{\mathrm{m}} \otimes \mathbf{F}_0 \\ \mathbf{q} &= k(\mathbf{x}, \mathbf{x}') \otimes \mathbf{q}_0,\end{aligned} \tag{3.22}$$

where $\mathbf{I}_{\mathrm{m}}$ is the identity matrix size m and the repeated units $\mathbf{F}_0$ and $\mathbf{q}_0$ depend only on the structure and hyperparameters of $k(t,t')$ [158]. The initial state of the system reflects the GP prior. Therefore, a zero mean GP implies $\mathbf{x}_0 = \mathbf{0}$. On the other hand, the initial covariance depends on the type of kernel $k(t,t')$. For stationary kernels, the stability of the dynamic system implies that an asymptotic covariance $\mathbf{P}_\infty$ exists such that it satisfies the continuous-time Lyapunov equation,

$$\left.\frac{\mathrm{d}\mathbf{P}_{\mathrm{t}}}{\mathrm{d}t}\right|_{\mathbf{P}_{\mathrm{t}}=\mathbf{P}_\infty} = \mathbf{F}\mathbf{P}_\infty + \mathbf{P}_\infty\mathbf{F}^{\mathrm{T}} + \mathbf{L}\mathbf{q}\mathbf{L}^{\mathrm{T}} = \mathbf{0}, \tag{3.23}$$

so the initial covariance is set at $\mathbf{P}_\infty$, which is retrieved by solving (3.23). For nonstationary $k(t,t')$ however, $\mathbf{P}_\infty$ does not exist, and the initial covariance depends on starting time t. Therefore, the initial covariance is found by integrating (3.20b) from $0 \to t$ which, for the linear system of equations (3.18), has the solution

$$\mathbf{P}_{\mathrm{t}} = \exp(\mathbf{F}t)\mathbf{P}_0 \exp(\mathbf{F}t)^{\mathrm{T}} + \int_0^t \exp(\mathbf{F}t)\mathbf{L}\mathbf{q}\mathbf{L}^{\mathrm{T}} \exp(\mathbf{F}t)^{\mathrm{T}}\,\mathrm{d}t, \tag{3.24}$$

where $\mathbf{P}_0$, $\mathbf{L}$ and $\mathbf{q}$ depend on the kernel $k(t,t')$ [111]. For many of the nonstationary kernels, $\mathbf{F}$ is a nilpotent matrix, in which case evaluating (3.24) is trivial. Also, the use of the Kronecker relation in (3.21)-(3.22), which is associative, means that finding the initial covariance matrix for the overall system can be done for a single location in $\mathbf{x}$ for $k(t,t')$, where $\mathbf{P}_0 \in \mathbb{R}^{n\times n}$, and the overall spatially resolved initial covariance is then recovered by another Kronecker product,

$$\mathbf{P} = k(\mathbf{x}, \mathbf{x}') \otimes \mathbf{P}_0. \tag{3.25}$$

Algorithm 3.1 Standard discrete-time Kalman filter recursion to calculate the forward filtering distribution of a Gaussian process $p(\mathbf{f}(\mathbf{x},t)|y_{1:t})$ expressed as an LTI-SDE.

Initialisation
  $\mathbf{x}^+ = \mathbf{x}_0$
  $\mathbf{P}^+ = \mathbf{P}_0$
  $\phi = -\log p(\theta)$ ▷ ϕ is initialised at the (-)ve hyperprior log density
for $t = 1 \dots T$ **do**
  Propagation
    $\mathbf{x}_t^- = \mathbf{A}\mathbf{x}_{t-1}^+$ ▷ $\mathbf{A} = \exp(\mathbf{F}\Delta t)$
    $\mathbf{P}_t^- = \mathbf{A}\mathbf{P}_{t-1}^+\mathbf{A}^\mathrm{T} + \mathbf{Q}(\Delta t)$ ▷ $\mathbf{Q}(\Delta t) = \int_0^{\Delta t} \exp(\mathbf{F}t)\mathbf{L}\mathbf{q}\mathbf{L}^\mathrm{T}\exp(\mathbf{F}t)^\mathrm{T}\,\mathrm{d}t$
  Measurement and update
    $\hat{\mathbf{y}}_t = \mathbf{H}\mathbf{x}_t$ ▷ GP predictive mean (3.5a)
    $\mathbf{S}_t = \mathbf{H}\mathbf{P}_t^-\mathbf{H}^\mathrm{T} + \mathbf{R}$ ▷ $\mathbf{S}_t$ includes GP predictive variance (3.5b)
    $\mathbf{K}_t = \mathbf{P}_t^-\mathbf{H}^\mathrm{T}\mathbf{S}_t^{-1}$
    $\mathbf{x}_t^+ = \mathbf{x}_t^- + \mathbf{K}_t(\mathbf{y}_t - \hat{\mathbf{y}}_t)$
    $\mathbf{P}_t^+ = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_t^-(\mathbf{I} - \mathbf{K}_t\mathbf{H})^\mathrm{T} - \mathbf{K}_t\mathbf{R}\mathbf{K}_t^\mathrm{T}$ ▷ Joseph form covariance update
  Energy function update
    $\phi = \phi + \frac{1}{2}\log|2\pi\mathbf{S}_t| + \frac{1}{2}(\mathbf{y}_t - \hat{\mathbf{y}}_t)^\mathrm{T}\mathbf{S}_t^{-1}(\mathbf{y}_t - \hat{\mathbf{y}}_t)$
end for

Once the structure and initial mean and covariance of the state-space system representing the GP have been determined, the remaining task is to define the observation model. It is given by the predictive distribution over $\mathbf{x}_t$, described by the kernel function $k(\mathbf{x},\mathbf{x}')$. In other words, the linear operator $\mathcal{H}(\mathbf{x}_t)$ in (3.18) is given by (3.5a). If observations of $\mathbf{y}_t$ are at constant spatial locations over time, $\mathcal{H}(\mathbf{x}_t)$ is a matrix $\mathbf{H}$. This however, is not always the case as discussed in the detailed implementation (Section 3.2.3).

Given the state-space model, inference consists of applying the standard Kalman filter recursion over observations together with the Rauch-Tung-Striebel smoother (RTSS) [163]. The Kalman filter first gives the filtering distribution $p(\mathbf{f}(\mathbf{x},t)|\mathbf{y}_{1:t})$, along with the energy function $\phi(\theta)$, also calculated recursively (see Algorithm 3.1). This is used to estimate the log posterior probability of the kernel hyperparameter vector θ, which can be used to retrieve optimal hyperparameters either by using their MAP estimates or by approximating the complete

Algorithm 3.2 The RTSS calculating the marginal distribution of states $\mathbf{x}_t$ conditioned on all observations $\mathbf{y}_{1:T}$. It consists of a backward recursion over the forward filtering distribution returned by the Kalman filter. Matrices $\mathbf{A}$ and $\mathbf{Q}$ are the same as in the forward pass shown in Algorithm 3.1.

Initialisation ▷ Using mean and covariance at $t = T$ from the forward pass
  $\mathbf{x}_T^s = \mathbf{x}_T^+$
  $\mathbf{P}_T^s = \mathbf{P}_T^+$
for $t = T-1 \dots 1$ **do** ▷ Backward recursion
    $\mathbf{x}_{t+1}^- = \mathbf{A}_t \mathbf{x}_t$
    $\mathbf{P}_{t+1}^- = \mathbf{A}_t \mathbf{P}_t \mathbf{A}_t^T + \mathbf{Q}_t$
    $\mathbf{G}_t = \mathbf{P}_t^+ \mathbf{A}_t^T (\mathbf{P}_{t+1}^-)^{-1}$
    $\mathbf{x}_t^s = \mathbf{x}_t^+ + \mathbf{G}_t (\mathbf{x}_{t+1}^s - \mathbf{x}_{t+1}^-)$
    $\mathbf{P}_t^s = \mathbf{P}_t^+ + \mathbf{G}_t \left(\mathbf{P}_{t+1}^s - \mathbf{P}_{t+1}^-\right) \mathbf{G}_t^T$
end for

posterior $p(\theta|\mathbf{X},\mathbf{y})$. It is also possible to calculate the gradient $\partial\phi/\partial\theta$ in the same recursion (discussed later in 3.2.3 and Appendix B.2), which can be computationally more efficient than other methods of calculating the gradients when using gradient-based optimisation techniques.

Once optimal hyperparameters have been calculated, the remaining step is to calculate the GP posterior using those hyperparameters. Whereas the forward pass of the Kalman filter gives the filtering distribution using observations up to time t, that is, $p(\mathbf{f}(\mathbf{x},t)|\mathbf{y}_{1:t})$, the marginal distribution that is consistent with the GP posterior calculated in the batch approach (3.5) is the smoothing distribution, $p(\mathbf{f}(\mathbf{x},t)|\mathbf{y}_{1:T})$. This distribution is acausal, i.e. it estimates $\mathbf{f}(\mathbf{x},t)$ at each point in time using all available observations, $\mathbf{y}_{1:T}$. It is retrieved by the RTSS (see Table 3.2), which is applied as a backward recursion once the forward pass with the Kalman filter is complete. Although the RTSS only returns the marginal distributions of $\mathbf{x}_t$ over the time dimension, it is also possible to retrieve the full covariance matrix from batch mode GP regression [111], but in this work it is not necessary. Altogether, the recursive method is shown schematically in Fig. 3.2.

Predictions using the recursive framework are equally computationally efficient. Starting from the smoothing distribution, to make predictions over time, only the propagation of the mean and covariance of the state vector using (3.20) is required. For spatial predictions (i.e.

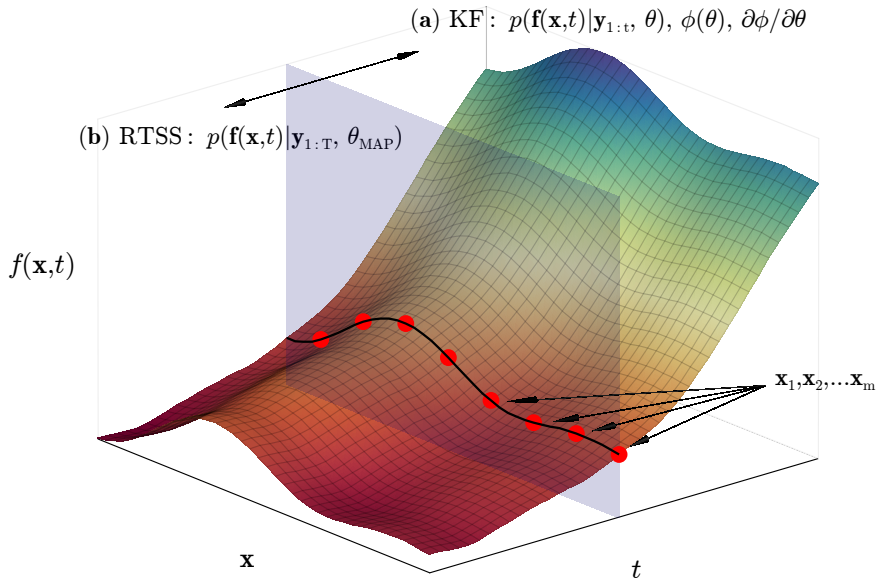


Fig. 3.2 Schematic representation of fitting a GP recursively, as a solution to a LTI-SDE. A set of points describing the value of a GP across the input space $\mathbf{x}$ is propagated through time and updated using the Kalman filter equations. The forward pass (step (a)) calculates the forward filtering distribution and the energy function. The forward pass is repeated many times to obtain estimates of hyperparameters. After this, the smoothing distribution is obtained by calculating both the forward and smoothing passes (a) + (b).

over the arbitrary $\mathbf{x}_*$), equations (3.5) can be applied to the state vector at any point in the smoothing/predictive distribution.

3.2 Parameterising equivalent circuit models with Gaussian process regression

As discussed earlier, using ECMs to perform SOH diagnosis in real-world systems results in noisy estimates, as ECM parameters are affected by operating conditions. Additionally, poorer sensor accuracy, numerical ill-conditioning and possible gaps in telemetry data make the task more difficult. The Bayesian framework brings many benefits for addressing these issues. By imposing an appropriately parameterised prior over the functions describing ECM

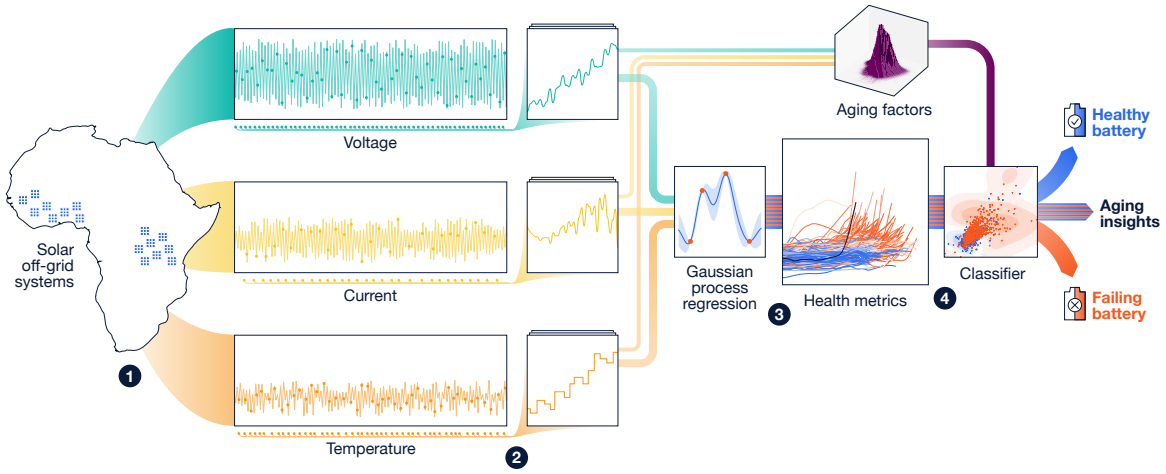


Fig. 3.3 Workflow schematic for the analysis of PV-connected VRLA batteries in sub-Saharan Africa [3]. Telemetry data are retrieved from batteries deployed in the field, which are sub-sampled to create a dataset for GPR. From this, stable health metrics are calculated and validated by using the health metrics as inputs into a binary GP classifier which outputs the probability that each battery has reached end of life. In addition, usage data over lifetime are used as extra inputs into the classifier to further condition the prediction, giving insight into factors driving ageing in the VRLA fleet.

parameters versus operating conditions and lifetime, stable estimates can be retrieved. Also, the estimates, in terms of a GP posterior/predictive distribution, come with the associated predictive uncertainty. In addition, using the functional prior allows for the extrapolation over both operating conditions and lifetime, effectively unifying diagnosis and prognosis.

Most of the following is from a previous publication [3], where the computationally efficient GPR solution was applied to estimate the parameters of a simple circuit model over the lifetime of 1027 PV-connected valve-regulated lead-acid (VRLA) batteries located in sub-Saharan Africa. It was shown that normalising circuit parameter estimates to constant operating conditions yields smoother estimates over lifetime that are consistent across the fleet. To validate the estimates, a GP classifier was used to identify batteries near to or at end-of-life. All field data, including validation data in the form of time-stamped repair information, was provided by BBOXX Ltd. In addition to validation of the health estimates, insight into the

ageing processes within the VRLA fleet was gained by augmenting the end-of-life classifier with metrics relating to the usage over lifetime, improving the performance of the classifier. The process workflow is shown in Fig. 3.3.

3.2.1 Physics-informed ECMs using Gaussian process regression

Battery model

Similarly to Li-ion chemistries, low-order physics-based models for lead-acid cells may be derived from porous electrode theory [48, 47], and some of these are on a par with equivalent circuits in terms of computational effort. However, the challenge of applying physics-based models to lead-acid is the same as it is with Li-ion, namely, the lack of reference electrodes and the requirement of prior knowledge of a large number of cell-specific parameters. However, insight from detailed physics-based models allow for the construction of a physics-informed combination of an equivalent circuit with GPs that can incorporate much of the underlying behaviour in a data-driven manner.

Similarly to the SPMe model for Li-ion cells in section 2.2, the voltage response to applied current in lead-acid cells can be formulated as the sum of the half-cell open-circuit potentials and a series of overpotentials due to the dynamic processes occurring inside the electrode sandwich [47, 48],

$$V_\mathrm{t} = U_\mathrm{p} - U_\mathrm{n} + \eta_\mathrm{k,p} - \eta_\mathrm{k,n} + \eta_\mathrm{c} + \eta_\mathrm{o}, \tag{3.26}$$

where U_i are the half-cell OCPs, $\eta_\mathrm{k,i}$ are the kinetic overpotentials on each electrode, η_c the concentration overpotential in the electrolyte and η_o the sum of ohmic overpotentials in the solid phase and electrolyte. The half-cell OCPs are experimentally determined functions of the sulfuric acid concentration on electrode surfaces, and overpotentials η are functions of the applied current. From asymptotic analysis in lead-acid systems, it has been shown that, at low C-rates, the overpotential η_c due to limits in ionic transport in the electrolyte is negligible

[47]. However, even at low C-rates the relationship between overpotential and applied current in lead-acid systems remains nontrivial, due to the kinetic overpotential $\eta_{k,i}$ being a complex function of applied current, SOC and temperature. The complexities arise mainly from the nucleation and dissolution of the discharge product, lead sulphate ($PbSO_4$), on the electrodes during discharge and charge respectively. During discharge from full SOC, the observed voltage overshoot—the so-called 'Coup-de-fouet' effect [164]—is a result of delayed nucleation of $PbSO_4$ on the surface of the electrodes. The effect is also observed with start of dissolution of $PbSO_4$ when charging from 0% SOC [165]. The reaction kinetics during charging near full SOC are likewise complicated, as the depletion of available $PbSO_4$ on the surface of the electrodes becomes a limiting factor and the kinetic overpotential increases substantially [166–168]. This in turn causes a strong increase in the rate of side reactions because hydrolysis of the aqueous electrolyte occurs, creating O_2 on the positive electrode and H_2 on the negative electrode. In sealed lead-acid systems (i.e. VRLA cells), the oxygen generated on the positive side will then recombine to form water at the negative electrode, lowering the total overpotential again [167]. Outside of first-principles physics-based models [166, 167], these processes have also been modelled with nonlinear circuit elements [169]. Bearing in mind that each of the physical processes mentioned above also has a temperature dependency, it is clear that the internal resistance observed at the terminals has a very complex dependency on operating conditions.

In addition to the variability of internal resistance to instantaneous operating conditions, outlined above, degradation over battery lifetime adds a further degree of freedom. Similarly to Li-ion systems, degradation is a result of multiple coupled mechanisms. These have been studied in detail in previous work [23] so here the main mechanisms are briefly summarised:

1. Loss of active material through corrosion and plate expansion:

   The positive plate construction in lead-acid systems consists of porous active material PbO_2 deposited on a lead-alloy grid. Corrosion of the positive grid at the Pb-PbO_2

interface is the most common form of degradation in lead-acid batteries [23]. It is favoured by high anodic potentials, so it readily occurs during the charging process as a side reaction [170]. The effect of corrosion is the loss of contact of the active material PbO_2 with the current collectors in the plane of the electrode plate. While not usually considered a problem in flooded lead-acid cells, it has also been argued that corrosion of components not sufficiently coated with electrolyte occurs on the negative electrode in sealed systems [23].

On the positive electrode, additional loss of active material occurs with cycling due to the differences in morphology between the solid reactant and product (PbO_2, $PbSO_4$) of the charge/discharge processes—cycling causes a spatial redistribution of PbO_2 over time on the positive electrode from the inside of the pores towards the surface [171]. This eventually leads to the electrical isolation of parts of the positive active material [172].

2. Sulfation and acid stratification:

The main charging/discharging reactions of healthy batteries are theoretically fully reversible. However, over extended time periods the discharge reaction product $PbSO_4$ can re-crystallise and become electrochemically passive [173], reducing the capacity by reducing the maximum acid concentration and increasing internal resistance. The extent of sulfation is determined by the length of time $PbSO_4$ crystals exist on the electrodes and is therefore a function of the time a lead-acid cell is kept in a partial state of charge. However, sulfation may also occur due to acid stratification. Acid stratification refers to the presence of a vertical concentration gradient in the electrolyte. The resulting diffusion potential between the top and bottom of the cell is counter-balanced by a vertical current within both electrodes, resulting in more $PbSO_4$ being formed at the bottom of the cells while $PbSO_4$ is converted back to the positive and negative active material at the top. As terminal voltage is a function of the average voltage along the length of the electrodes,

recharging may then cause incomplete charge towards the bottom, leaving $PbSO_4$ to re-crystallise [23].

3. Water loss:

   While not causing degradation directly, water loss from the electrolyte can accelerate other mechanisms and occurs through three different routes. Firstly, the corrosion reaction on the positive electrode during charging consumes water [23]. This mechanism occurs in both flooded and sealed valve-regulated systems. More significantly, hydrolysis occurs during the charging process, as O_2 is generated on the positive electrode and H_2 on the negative. In sealed lead-acid systems this is less of a problem as a recombination reaction occurs on the negative electrode with the oxygen generated on the positive, which regenerates (some of) the lost water. In sealed systems evaporation is also less of an issue due to the containment of gases within the cell. The effect of water loss is to increase the electrolyte acid concentration, which increases the rate of self-discharge and corrosion, which necessitates a higher float-charge current to maintain state of charge. If this is not properly managed, sulfation is also more likely to occur due to the additional time spent at a partial state of charge [23].

The extent to which degradation occurs through each of these mechanisms depends heavily on the battery operating conditions, especially temperature and (average) terminal voltage. Given this, a Thevenin circuit model with a series resistor that is a GP over multiple inputs may be used to describe the voltage response of a VRLA battery to applied current such that

$$\begin{aligned} V_\mathrm{t} &= V_0(z_\mathrm{t}) + R_0(\zeta_\mathrm{t}, u_\mathrm{t}, z_\mathrm{t}, T_\mathrm{t})u_\mathrm{t} + \varepsilon \ , \ \varepsilon \sim \mathcal{N}(0, \sigma^2_{\mathrm{n,t}}), \\ R_0 &\sim \mathcal{GP}\left(0, k(\mathbf{x}, \mathbf{x}')\right) \ , \ \mathbf{x} = [\zeta \ u \ z \ T] \end{aligned} \tag{3.27}$$

where $V_0(z_\mathrm{t})$ is the OCV as a function of SOC z_t, and $R_0(\zeta_\mathrm{t}, u_\mathrm{t}, z_\mathrm{t}, T_\mathrm{t})$ expresses the total internal resistance as a function of applied current u_t, SOC, temperature T_t and a chosen cumulative

stress factor over lifetime, ζ_t. In this chapter ζ_t is chosen to be calendar time, but other metrics such as cumulative charge throughput over lifetime could also be used. The model is intended to encapsulate all the physics of (3.26) in a low C-rate environment where diffusional overpotentials may be ignored. The error term ε is assumed zero-mean Gaussian with variance $\sigma^2_{n,t}$, which varies as a function of environmental factors.

The choice of internal resistance over capacity as a health metric was dictated by the available data. Most importantly, the average depth of discharge for the batteries considered in this study (see next section) was very low, with the nearly all usage falling in the 46-100% SOC range, making discharge capacity difficult to identify. Relatively sparse data sampling, resulting in inaccurate charge throughput estimation compounded this challenge. Alternatively, it is in principle possible to identify capacity loss from the behaviour of the overpotential during charging using equivalent circuit approaches [169]. However, the dynamic model is nonlinear with respect to capacity, especially near full SOC, necessitating very accurate SOC estimation. This was not possible with the data, making capacity estimation (where capacity would be modelled by a GP over time) too fragile to be applied to a large fleet of batteries.

In the following section the telemetry dataset which is used to fit the function R_0 is described, followed by the details of implementation.

3.2.2 Field dataset

BBOXX Ltd. operate approximately half a million home solar-battery systems in sub-Saharan Africa with varying chemistries and sizes. To apply GP regression to ECM parameter estimation for these batteries, a small subset of 1027 batteries was chosen. For validation, comparability between batteries was important, so carefully constructing the sub-sample for analysis was essential. For the GP classifier training data, the aim was to create a roughly 50/50 split of batteries that had reached end-of-life vs. healthy ones.

The 1027 PV-connected VRLA batteries came from the same manufacturer and had a nominal voltage of 12 V consisting internally of 6×2 V cells in series. The rated capacity was 20 Ah, and all of the batteries were connected to a 50 Wp solar panel. These systems were located in various countries in sub-Saharan Africa. Each battery was equipped with telemetry to measure current, voltage and temperature. Current sensors were calibrated with a 0.2% accurate reference before leaving the factory. Voltage and temperature sensors had systematic uncertainties of 3% and 2 °C, respectively. The steps taken to arrive at the subset of 1027 batteries were:

1. Lead-acid batteries from single manufacturer with nominal capacity 20 Ah were chosen.

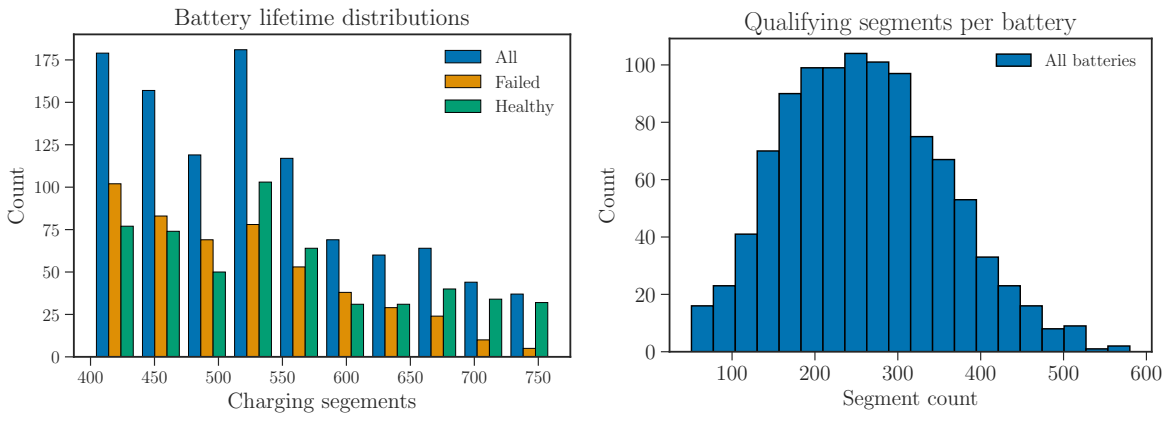
2. Lifetime was defined as the time since the 'activation date' designated by BBOXX Ltd. Batteries with a lifetime of over 400 days were chosen, with an activation date after 2018-09-01.

3. For those batteries that entered repair at some point after activation, end of life was defined as the date they entered repair. The dataset was filtered down so only batteries that either had no repairs or entered repair due to loss of capacity (as defined by BBOXX) were included. The failed set consisted of all the batteries that had been diagnosed with a loss of capacity when taken into a repair shop. To make up the total dataset with an approximate 52/48 split between healthy and failed cells, stratified sampling with respect to lifetime was used to generate a set of healthy batteries from the overall population with the same age distribution.

4. Any batteries that had large gaps in their telemetry data were removed (over 30 days).

5. Any batteries that had tampering alerts during operation were removed.

6. Telemetry data for each battery where temperature readings were clearly incorrect were removed.



(a) (b)

Fig. 3.4 (a) Age distribution of BBOXX batteries (b) Distribution of charging segment count in downsampled BBOXX dataset. The average count per battery is approximately 260.

7. Any batteries that did not have enough charging segments over their lifetime were removed. The method for calculating the number of charging segments available for each battery is described below. Batteries with fewer than 50 qualifying segments were removed. Additionally, batteries were omitted where the last qualifying segment was more than 30 days before the end of the dataset series.

8. Batteries that entered repair were labelled ‘faulty’, and their timeseries were truncated to only include data up to the repair date.

Using the steps outlined above, the set of 1027 batteries contained 491 faulty batteries and 536 healthy ones. The dataset consisted of 620 million rows of telemetry data, each row including a timestamp, voltage, current and temperature and the arrival date at the repair shop for each battery where applicable. Lifetimes from the activation date to the end of the data series range from 400-760 days, where the distribution of ages is shown in Fig. 3.4a.

The telemetry data collection algorithm used by BBOXX was designed to minimise data transfer. To do this, recordings of measurements in current, voltage and temperature are only triggered when sensors detect a change larger than a pre-specified threshold. In the absence of new measurements, there is also a scheduled measurement that operates at low frequency

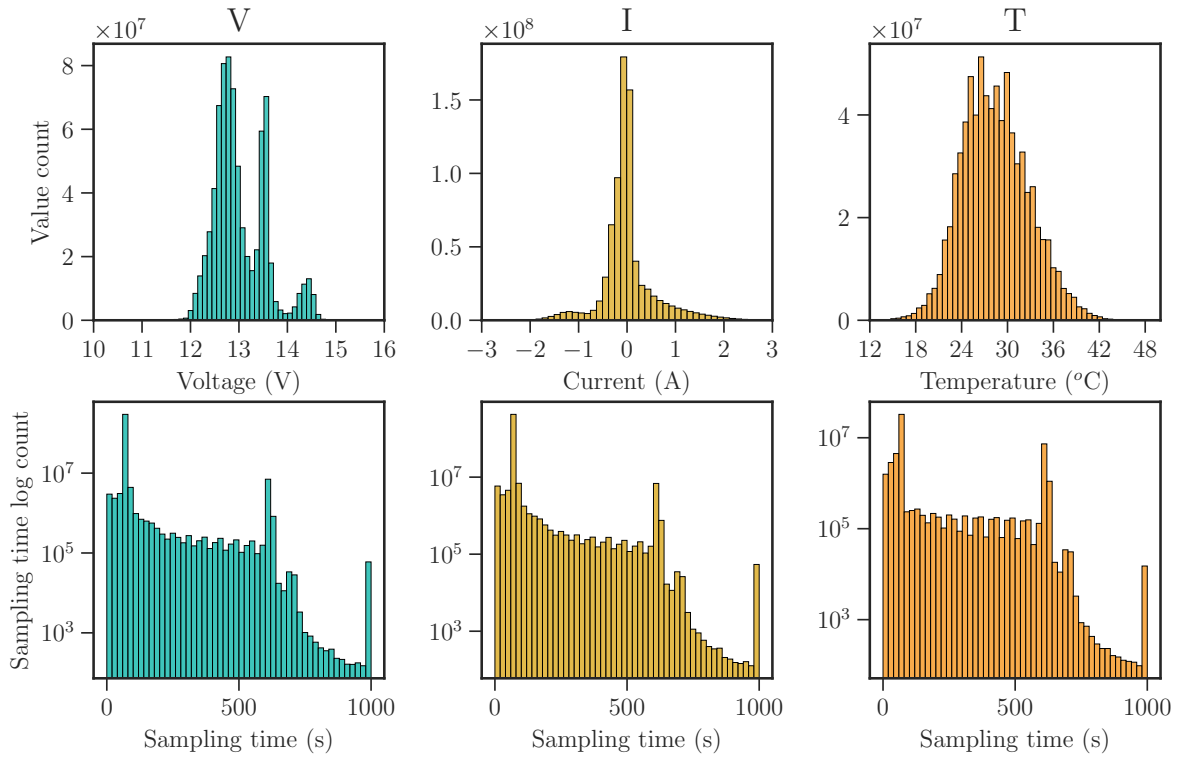


Fig. 3.5 Distributions of current, voltage and temperature and their sampling times. The mode of the measurement frequency is 1 min^{-1}. N.B. The bars at 1000 s represent the cumulative count of all sampling times over 1000 s.

that acts as a backstop. As a result, the telemetry was irregularly sampled. The distributions of telemetry measurements and their sampling times are shown in Fig. 3.5. The histograms of measurements reflect the diurnal load pattern experienced by the system shown in Fig. 3.6. Solar charging occurs during the day, and there is an upper voltage limit of approximately 14.5 V and once the current drops below a given threshold, float-charge starts at a voltage of roughly 13.5 V. Discharge during the night happens on average at much lower currents which are relatively constant.

Data downsampling and SOC estimation

While the recursive method to estimate the function R_0 could be applied to all telemetry data, it was easier and more efficient to focus on the charging segments only. This was for

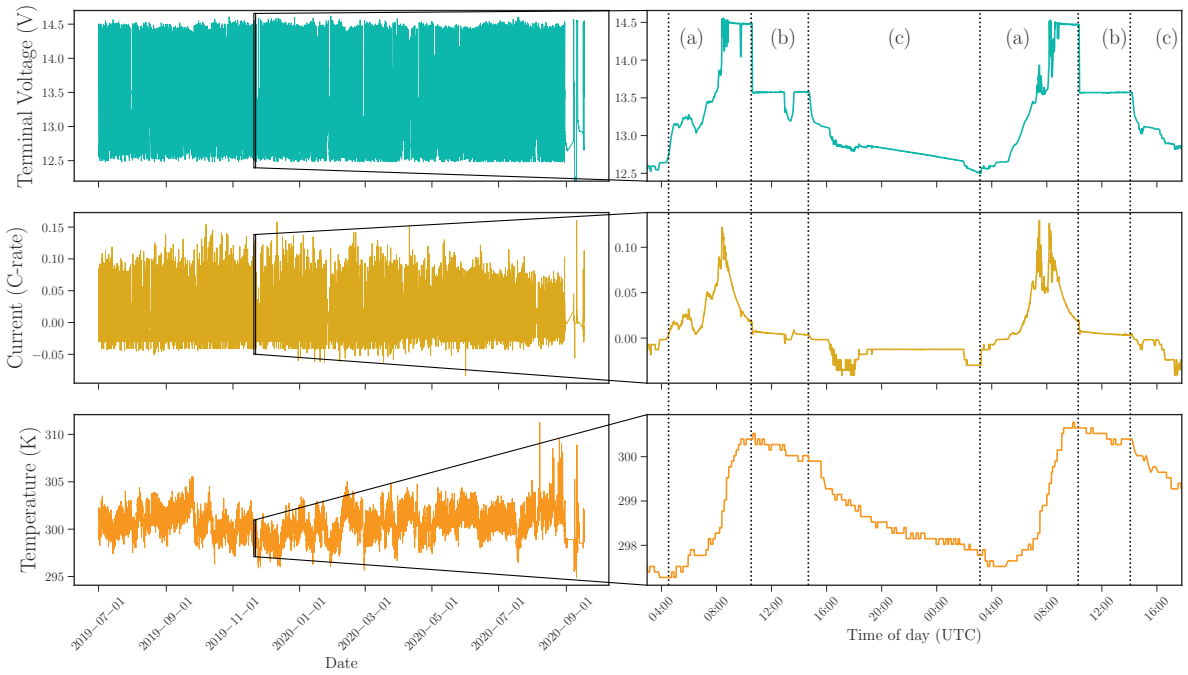


Fig. 3.6 Diurnal loading patterns experienced by the batteries consist of (a) solar charge, (b) float charge, (c) discharge at night time.

several reasons. Firstly, charging currents provide a substantially richer input signal into the model (3.27) because they are on average larger in amplitude and more varied than discharge currents due to variation in incident solar irradiation. Secondly, as there is a regular diurnal charge/discharge pattern over the data, it is not necessary to include both for SOH estimation as it was assumed that SOH is approximately constant within a 24-hour windows. The final reason was the estimation of SOC—because the telemetry did not include this, it had be to be inferred from the data. Discharge cycles tended to start directly from the end of floating charge periods (transition from (b)-(c) in Fig. 3.6) without a rest period, and therefore SOC estimation was difficult—the float charge periods preceding discharge vary in length of time and temperature and it could not be assumed that as soon as floating charge occurs the SOC was 100%. Transitioning from discharge to charge at the end of discharge in contrast often included a time period where the current was minimal, which allowed for a simple inversion of the OCV to retrieve an estimate of the SOC at the end of discharge.

Charging segment duration	> 6000 s
Starting voltage range	11.5 V-12.9 V
Starting current	< 0.1 A
max(Voltage in segment)	> 14 V
max(Time gap in recorded data)	< 610 s

Table 3.1 Qualifying charging segment conditions

Downsampling process

The downsampling of data consisted of choosing as many qualifying charging segments as possible over the lifetime of each battery. Qualifying charging segments were those that met the criteria in Table 3.1, and these conditions ensured that charging segments covered a reasonable state of charge range. Additionally, each charging segment was truncated to include voltages only up to 14 V due to increased uncertainty in estimating state of charge at higher voltages (see next section), because the magnitude of the side reactions increases exponentially with terminal voltage. After down-selecting appropriate charging segments, data were interpolated to a 1-minute time grid using piece-wise cubic hermite interpolation [174]. During charging, data points where u_t <0.2 A, or the estimated open circuit voltage was above the measured terminal voltage (which can sometimes occur due to inaccuracy of the state of charge estimate), were removed to improve numerical conditioning and ensure that resistance estimates were always positive. The effect of data downsampling was to reduce the amount of data from approximately 603,000 rows per battery to 38,000 rows over an average segment count of 260, giving a total dataset size of ~40 million rows. The distribution of number of charging segments for all batteries is shown in Fig. 3.4b. As explained above, batteries with fewer than 50 charging segments were not included.

SOC estimation

The average concentration of sulfuric acid in the electrolyte was used as a proxy for SOC. Normalising this to retrieve the conventionally defined SOC metric in the range of 0-100% requires

knowledge of the maximum acid concentration, which changes over lifetime. Therefore, the unnormalised absolute concentration was used. To infer acid concentration from measured data, the open circuit voltage of the battery was measured first in a laboratory using the galvanostatic intermittent technique (GITT) at 25 °C. From laboratory data, the electrolyte volume was also inferred by a least-squares fit, comparing the experimental OCV with that reported by Bode [175]. The details of the OCV parameterisation and electrolyte volume calculations can be found in Appendix B.1. These calculations were performed for a single example battery of the same specification as those in the field dataset.

Given the parameterised OCV curve from lab data, the SOC at the beginning of each charging segment was obtained by inverting the curve as the current at that point is very close to zero. State of charge (i.e. acid concentration) was then estimated for the rest of the charging segment in an open-loop manner, whereby Coulomb counting was applied with a lumped model for side reactions, namely hydrolysis on the positive electrode,

$$\frac{d\hat{c}_\mathrm{t}}{dt} = \frac{u_\mathrm{t} - I_{\mathrm{gas},0} e^{c_T(T_\mathrm{t}-T_0)+c_V(V_\mathrm{t}-V_{\mathrm{g},0})}}{FV_\mathrm{elec}}, \tag{3.28}$$

where u_t, T_t, V_t are the measured current, temperature and terminal voltage, F is Faraday constant, V_elec is the estimated electrolyte volume, and gassing current parameters $I_{\mathrm{gas},0}$, c_T, T_0, c_V, $V_{\mathrm{g},0}$ were from literature [176]. The substantial uncertainty in SOC estimation due to the variation in these parameters over lifetime was taken into consideration by projecting the input uncertainty in the acid concentration $\hat{c}$ to be measurement noise variance in GP regression, (i.e. $\sigma^2_{\mathrm{n,t}}$ in the model (3.27)).

Firstly, there was considerable uncertainty in the open circuit voltage function $V_0(\hat{c})$ due to hysteresis [177] as well as in the estimate of acid concentration $\hat{c}$, due to uncertainty in the parameters in equation 3.28. Additionally, experimental estimation of the OCV curve and electrolyte volume was done using a single battery, so variations in these across the population are another source of uncertainty. To account for this, a 10% uncertainty in $\mathrm{d}\hat{c}/\mathrm{d}t$ was assumed,

together with a 150 mV standard deviation caused by voltage measurement and open circuit voltage uncertainty, giving a total variance per charge segment as

$$\sigma^2_{\mathrm{n,t}} = 0.0225 + \mathrm{Var}(\hat{c}_\mathrm{t}) \left(\frac{dV_0}{d\hat{c}} \bigg|_{\hat{c}} \right)^2 \tag{3.29a}$$

$$\mathrm{Var}(\hat{c}_\mathrm{t}) = \sum_t 0.01 \Delta \hat{c}^2_\mathrm{t}, \tag{3.29b}$$

where $\Delta\hat{c}_\mathrm{t} = (\mathrm{d}c_\mathrm{t}/\mathrm{d}t)\,\Delta t$. In summary, GP regression included heteroskedastic noise $\sigma^2_{\mathrm{n,t}}$ which is pre-calculated rather than estimated as a hyperparameter in the fitting process. By fixing the noise variance in the fitting process, the hyperparameter estimation problem was significantly easier. Estimating heteroskedastic noise would have required a higher-dimensional hyperparameter vector, and make the (gradient) based optimisation process prone to numerical instability—by effectively fixing the diagonal of matrix $\mathbf{R}$ in step 11 of Algorithm 3.1 in each charging segment, numerical stability of the inversion in step 12, Algorithm 3.1 was no longer an issue as the matrix $\mathbf{S}_\mathrm{t}$ was practically guaranteed to be positive definite. Furthermore, it was found that solving the hyperparameter optimisation problem without fixing the noise variance often resulted in a solution whereby the GP would try to fit to the voltage very accurately (i.e. under-estimate the noise parameter), causing fluctuations in R_0 over time.

Input data normalisation

For the purposes of scaling hyperparameters, the operating point input data, consisting of applied current, temperature and estimated SOC, were normalised using population level estimates of their mean $\bar{x}$ and standard deviation σ_x, so that

$$X = \frac{x - \bar{x}}{\sigma_\mathrm{x}}\ ,\ x \in \{T_\mathrm{t}, u_\mathrm{t}, \hat{c}_\mathrm{t}\}. \tag{3.30}$$

The time input was normalised by taking the time since the activation date and dividing by a normalising factor of 400 days. This brings it approximately in the unit range, as the lifetime of the batteries range from 400-760 days.

3.2.3 Recursive GP regression to estimate internal resistance

The computationally efficient method of GPR (see 3.1.4) was applied to the data to estimate the function R_0 from the preprocessed data. Specifically, $R_0(\zeta_t, u_t, z_t, T_t)$ was modelled as a zero-mean GP, with prior covariance given by a composite kernel constructed from the sum of the kernel functions over the time dimension and that over the 'operating point' dimension. Denoting the overall input vector $\mathbf{x} = [\zeta\ u\ z\ T]$ and the subcomponent $\mathbf{x}_{\text{OP}} = [u\ z\ T]$, R_0 is then expressed as

$$R_0 \sim \mathcal{GP}\left(0, k_{\mathbf{x}}(\mathbf{x}, \mathbf{x}')\right)\ ,\ \text{where } k_{\mathbf{x}}(\mathbf{x}, \mathbf{x}') = k_{\zeta}(\zeta, \zeta') + k_{\mathbf{x}_{\text{OP}}}(\mathbf{x}_{\text{OP}}, \mathbf{x}'_{\text{OP}}). \tag{3.31}$$

Using the additive formulation of kernels implies that the evolution of R_0 over time is independent of operating point—that is, for all operating points $\mathbf{x}_{\text{OP}} = [u\ z\ T]$, degradation adds to the internal resistance equally. This 'decoupling' approximation was done to reduce the computational effort in fitting the GP. A multiplicative form could also have been used (see Chapter 4) which allows for more degrees of freedom in the evolution of circuit parameters. The beginning of life for the batteries in the BBOXX dataset was defined as the activation date, where $\zeta = 0$, implying that the degradation component was also equal to zero at BoL. This is shown schematically in Fig. 3.7.

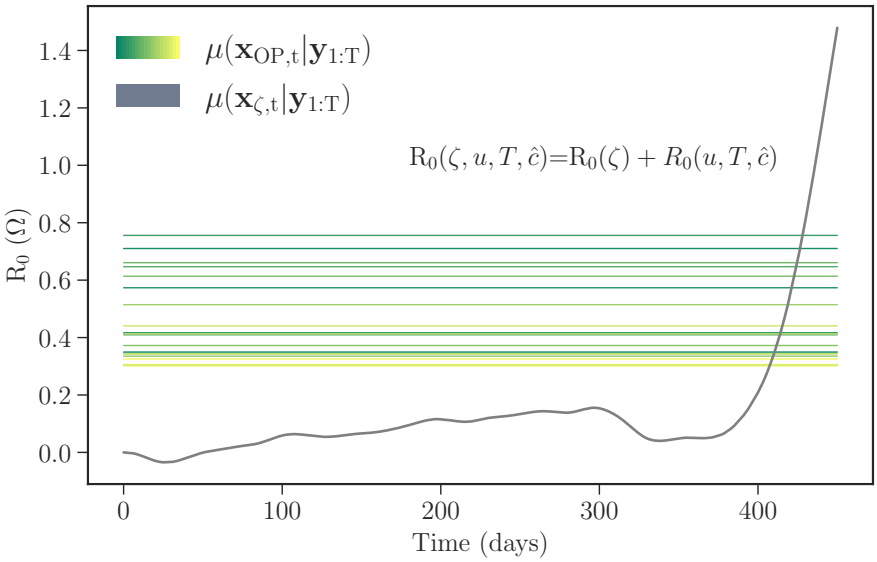


Fig. 3.7 Schematic of the behaviour of the additive GP describing R_0 over lifetime. The operating point dependency is constant over lifetime, shown here at discrete operating points $\mathbf{x}_{\mathrm{OP.t}}$, where each horizontal line represents R_0 at a specific operating point. GP extrapolation can then be used to calculate R_0 at an arbitrary operating point, which is a linear combination of these. The degradation component $\mathbf{x}_{\zeta,\mathrm{t}}$ starts at zero and is governed by the WV kernel.

The two kernel functions chosen for the GP prior (3.31) were the Wiener velocity (WV) kernel for ζ and the squared exponential (SE) kernel for $\mathbf{x}_{\mathrm{OP}}$,

$$k_\zeta(\zeta,\zeta') = \sigma_{\mathrm{f},0}^2\left(\frac{\min^3(\zeta,\zeta')}{3}+|\zeta-\zeta'|\frac{\min^2(\zeta,\zeta')}{2}\right) \tag{3.32a}$$

$$k_{\mathbf{x}_{\mathrm{OP}}}(\mathbf{x}_{\mathrm{OP}},\mathbf{x}'_{\mathrm{OP}}) = \sigma_{\mathrm{f},1}^2\exp\left(-\frac{1}{2}(\mathbf{x}_{\mathrm{OP}}-\mathbf{x}'_{\mathrm{OP}})^{\mathrm{T}}\Sigma^{-1}(\mathbf{x}_{\mathrm{OP}}-\mathbf{x}'_{\mathrm{OP}})\right), \tag{3.32b}$$

where the SE kernel (also known as the radial basis function kernel) is a special case of the Matérn family of kernels where the parameter $\nu=\infty$. It is one of the most widely used kernel functions in the GP regression. Here, it was used in in the automatic relevance detection (ARD) form, meaning that

$$\Sigma = \begin{bmatrix} l_{\mathrm{u}}^2 & 0 & 0 \\ 0 & l_{\mathrm{z}}^2 & 0 \\ 0 & 0 & l_{\mathrm{T}}^2 \end{bmatrix}, \tag{3.33}$$

simplifying the Mahalanobis distance inside the exponential in (3.32b) to give

$$k_{\mathbf{x}_{\mathrm{OP}}}(\mathbf{x}_{\mathrm{OP}}, \mathbf{x}'_{\mathrm{OP}}) = \sigma_{\mathrm{f},1}^2 \exp\left(- \sum_{x \in \mathbf{x}_{\mathrm{OP}}} \frac{(x - x')^2}{2 l_{\mathrm{x}}^2} \right). \tag{3.34}$$

Using ARD, the inverse of the length scale hyperparameters l provides an indication of which input in $\mathbf{x}_{\mathrm{OP}} = [u\ z\ T]$ carries the most relevance. The WV kernel on the other hand does not have a characteristic length scale hyperparameter. As $\zeta = 0$ at the beginning of life for the batteries, this implies that the 'degradation' part of the GP is zero at the start as well. The overall kernel may be fully parameterised by the hyperparameter vector $\theta_{\mathrm{h}} = [\sigma_{\mathrm{f,o}}\ \sigma_{\mathrm{f},1}\ l_{\mathrm{u}}\ l_{\mathrm{z}}\ l_{\mathrm{T}}]$. To fit the GP the recursive method previously described was used (section 3.1.4).

In order to construct the system of linear equations describing the dynamics of the GP over time (section 3.1.4), a discretised representation of $\mathbf{x}_{\mathrm{OP}} = [u\ z\ T]$ was needed. This was done by choosing 20 points in $\mathbf{x}_{\mathrm{OP}}$ by applying the k-means algorithm to the downsampled current, SOC and temperature data for each battery, which gave a set of points that represents the areas of high density of data in $\mathbf{x}_{\mathrm{OP}}$. The resulting linear dynamic system from the GP describing the summation of the two kernels was then given by the propagation of the mean and covariance of the overall state vector $\mathbf{x}_{\mathrm{t}}$, which is a concatenation of the state vectors from each kernel function,

$$\mathbf{x}_{\mathrm{t}} = \begin{bmatrix} \mathbf{x}_{\zeta,\mathrm{t}} & \mathbf{x}_{\mathrm{OP,t}} \end{bmatrix}^{\mathrm{T}}, \tag{3.35}$$

where $\mathbf{x}_{\zeta,\mathrm{t}} = [x_{\zeta,\mathrm{t}}\ \mathrm{d}x_{\zeta,\mathrm{t}}/\mathrm{d}t]^{\mathrm{T}}$ is due to the WV kernel and $\mathbf{x}_{\mathrm{OP,t}} \in \mathbb{R}^{20}$ is due to the SE kernel. In other words, the WV kernel gives a first-order Markov process and the SE kernel gives a dynamic system with no time dynamics because it describes a GP that is constant over time. The discrete time system describing the GP is then given by

$$\mathbf{x}_{\mathrm{t+1}} = \mathbf{A}_{\mathrm{t}} \mathbf{x}_{\mathrm{t}} \tag{3.36a}$$

$$\mathbf{P}^{-}_{\mathrm{t+1}} = \mathbf{A}_{\mathrm{t}} \mathbf{P}^{+}_{\mathrm{t}} \mathbf{A}^{\mathrm{T}}_{\mathrm{t}} + \mathbf{Q}_{\mathrm{t}}, \tag{3.36b}$$

where

$$\mathbf{A}_\mathrm{t} = \exp\left(\begin{bmatrix}\mathbf{F}_\mathrm{WV} & 0\\ 0 & \mathbf{F}_\mathrm{SE}\end{bmatrix}\Delta t\right), \mathbf{F}_\mathrm{WV} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}, \mathbf{F}_\mathrm{SE} = \mathbf{0}, \tag{3.37}$$

where $\mathbf{0}$ indicates a square matrix of zeros of size 20 and Δt denotes the (normalised) time step between charging segments. The matrix exponential to calculate $\mathbf{A}_\mathrm{t}$ was evaluated using a Taylor expansion, as the term inside the exponentiation is nilpotent. The process covariance $\mathbf{Q}_\mathrm{t}$ is also block diagonal, given by

$$\mathbf{Q}_\mathrm{t} = \begin{bmatrix}\mathbf{Q}_\mathrm{WV} & 0\\ 0 & \mathbf{Q}_\mathrm{SE}\end{bmatrix}, \mathbf{Q}_\mathrm{WV} = \sigma_{\mathrm{f},0}^2 \begin{bmatrix}\frac{1}{3}\Delta t^3 & \frac{1}{2}\Delta t^2\\ \frac{1}{2}t^2 & \Delta t\end{bmatrix}, \mathbf{Q}_\mathrm{SE} = \mathbf{0}, \tag{3.38}$$

which can be obtained by evaluating (3.24). As the prior for R_0 has zero mean, the state vector $\mathbf{x}_\mathrm{GP}$ is initialised at zero. The initial covariance matrix for the dynamic system is likewise block diagonal,

$$\mathbf{P}_0 = \begin{bmatrix}\mathbf{P}_{\mathrm{WV},0} & 0\\ 0 & \mathbf{P}_{\mathrm{SE},0}\end{bmatrix}, \tag{3.39}$$

initialised separately for the two subsystems. The integrated Wiener process is defined to have zero initial covariance, so $\mathbf{P}_{\mathrm{WV},0} = \mathbf{0}$. For the second GP over operating points, which has no time dependency, the initial covariance is given by $\mathbf{P}_{\mathrm{SE},0} = \sigma_{\mathrm{f},1}^2\mathbf{I}$, where $\mathbf{I}$ is the identity matrix of size 20 due to the size of $\mathbf{x}_{\mathrm{OP,t}}$.

Following the propagation of the GP in state space, the observation model gives the estimated internal resistance at all operating points in the data in a given charging segment expressed as a linear combination of elements within the state vector $\mathbf{x}_{\mathrm{GP,t}}$. Given n observations in a charging segment,the matrix $\mathbf{H}_\mathrm{t} \in \mathbb{R}^{n\times 22}$ is given by the concatenation of the two GPs,

$$\mathbf{H}_\mathrm{t} = \left[\begin{array}{c|c}\mathbf{H}_0 & \mathbf{H}_{1,\mathrm{t}}\end{array}\right] = \left[\begin{array}{cc|c}1 & 0 & k_{\mathbf{x}_\mathrm{OP}}(\mathbf{X}_{\mathrm{obs,t}}, \mathbf{X}_\mathrm{OP})\mathbf{K}^{-1}\\ \vdots & \vdots & \vdots\end{array}\right], \tag{3.40}$$

where $\mathbf{K} = k_{\mathbf{x}_{\mathrm{OP}}}(\mathbf{X}_{\mathrm{OP}}, \mathbf{X}_{\mathrm{OP}}) + \mathrm{diag}[\mathbf{P}^{-}_{\mathrm{SE,t}}]$, $\mathbf{X}_{\mathrm{obs,t}}$ is the $n \times 3$ input matrix containing the n rows of applied current, SOC and temperature for the charging segment at time t and $\mathbf{X}_{\mathrm{OP}}$ is the 20×3 matrix representing the coordinates of the vector $\mathbf{x}_{\mathrm{OP}}$ in terms of u, z, T. The leading two columns in each row come from incorporating the effect of the temporal GP (described by the WV kernel), where to make predictions only the current state is considered and the time derivative is ignored. The vector $\sigma^2_{\mathrm{GP,t}} \in \mathbb{R}^n$ of predictive variances for each point in the charging segment is calculated in a similar way by taking the diagonal from the standard GP relation,

$$\sigma^2_{\mathrm{GP,t}} = \mathrm{diag}\left[k_{\mathbf{x}_{\mathrm{OP}}}(\mathbf{X}_{\mathrm{obs,t}}, \mathbf{X}_{\mathrm{obs,t}}) - k_{\mathbf{x}_{\mathrm{OP}}}(\mathbf{X}_{\mathrm{obs,t}}, \mathbf{X}_{\mathrm{OP}})\mathbf{K}^{-1}k_{\mathbf{x}_{\mathrm{OP}}}(\mathbf{X}_{\mathrm{OP}}, \mathbf{X}_{\mathrm{obs,t}})\right], \tag{3.41}$$

The diagonal elements of the covariance of the state vector at time t, $\mathrm{diag}[\mathbf{P}^{-}_{\mathrm{SE,t}}]$, in $\mathbf{K}$ have the effect of reducing the weight of the points with greater uncertainty in the prediction of $R_{0,\mathrm{t}}$. Given the relations (3.40-3.41), the observation model may be formalised. Denoting the Gaussian random variable describing the GP at time t as $\mathbf{z}_{\mathrm{GP,t}}$, the observation model for $\mathbf{V}_{\mathrm{t}}$ is given by the following distributions (similarly to [178]),

$$\mathbf{z}_{\mathrm{t}} \sim \mathcal{N}(\mathbf{x}^{-}_{\mathrm{t}}, \mathbf{P}^{-}_{\mathrm{t}}) \tag{3.42a}$$

$$\mathbf{R}_{0,\mathrm{t}}|\mathbf{z}_{\mathrm{t}} \sim \mathcal{N}(\mathbf{H}_{\mathrm{t}}\mathbf{z}_{\mathrm{t}}, \sigma^2_{\mathrm{GP,t}}) \tag{3.42b}$$

$$\mathbf{V}_{\mathrm{t}}|\mathbf{R}_{0,\mathrm{t}} \sim \mathcal{N}(V_0(\hat{\mathbf{c}}_{\mathrm{t}}) + \mathbf{R}_{0,\mathrm{t}}\mathbf{u}_{\mathrm{t}}, \sigma^2_{\mathrm{n,t}}), \tag{3.42c}$$

which have the joint distribution

$$p\left(\begin{bmatrix}\mathbf{z}_{\mathrm{t}}\\ \mathbf{R}_{0,\mathrm{t}}\\ \mathbf{V}_{\mathrm{t}}\end{bmatrix}\right) \sim \mathcal{N}(\mathbf{m}', \mathbf{P}'), \tag{3.43a}$$

where

$$\mathbf{m}' = \begin{bmatrix} \mathbf{x}_t^- \\ \mathbf{H}_t\mathbf{x}_t \\ V_0(\hat{c}_t) + \mathbf{H}_t\mathbf{x}_t\mathbf{u}_t \end{bmatrix}, \tag{3.43b}$$

$$\mathbf{P}' = \begin{bmatrix} \mathbf{P}_t^- & \mathbf{P}_t^-\mathbf{H}_t^{\mathrm{T}} & \mathbf{P}_t^-\mathbf{H}_t I_{\mathrm{d,t}}^{\mathrm{T}} \\ \mathbf{H}_t\mathbf{P}_t^- & \mathbf{H}_t\mathbf{P}_t^-\mathbf{H}_t^{\mathrm{T}} + \sigma_{\mathrm{GP,t}}^2 & \left[\mathbf{H}_t\mathbf{P}_t^-\mathbf{H}_t^{\mathrm{T}} + \sigma_{\mathrm{GP,t}}^2\right]\mathbf{U}_{\mathrm{d,t}}^{\mathrm{T}} \\ \mathbf{U}_{\mathrm{d,t}}\mathbf{H}_t\mathbf{P}_t^- & \mathbf{U}_{\mathrm{d,t}}\left[\mathbf{H}_t\mathbf{P}_t^-\mathbf{H}_t^{\mathrm{T}} + \sigma_{\mathrm{GP,t}}^2\right] & \mathbf{U}_{\mathrm{d,t}}\left[\mathbf{H}_t\mathbf{P}_t^-\mathbf{H}_t^{\mathrm{T}} + \sigma_{\mathrm{GP,t}}^2\right]\mathbf{U}_{\mathrm{d,t}}^{\mathrm{T}} + \sigma_{\mathrm{n,t}}^2 \end{bmatrix}, \tag{3.43c}$$

so that the innovation covariance $\mathbf{S}_t$ of the Kalman filter is the last block element on the leading diagonal of $\mathbf{P}'$. The matrix $\mathbf{U}_{\mathrm{d,t}}$ is a diagonal matrix consisting of the current at each point of the charging segment, and the calculation of the vector $\sigma_{\mathrm{n,t}}^2$, which takes into account the variance due to uncertainty in $\mathbf{V}_0(\hat{\mathbf{c}}_t)$, is described in section 3.2.2. The propagation and observation models of the GP thus fully define the Kalman filter recursion to find the forward posterior distribution of the GP.

To retrieve estimates for the hyperparameter vector $\theta_{\mathrm{h}} = [\sigma_{\mathrm{f,o}}\ \sigma_{\mathrm{f,1}}\ l_{\mathrm{I}}\ l_{\mathrm{z}}\ l_{\mathrm{T}}]$, the energy function was recursively calculated as per Algorithm 3.1. In other words, for each pass through the data, pointwise estimates of the unnormalised negative log posterior probability given the input-output data, $\phi(\theta_{\mathrm{h}}) = -\log p(\theta_{\mathrm{h}}) - \log p(\mathbf{V}, \mathbf{x}|\theta_{\mathrm{h}})$ were calculated. The prior distribution over hyperparameters (i.e. the hyperprior), $p(\theta_{\mathrm{h}})$ was given by

$$p(\sigma_{\mathrm{f,0}}, \sigma_{\mathrm{f,1}}, l_{\mathrm{T}}, l_{\mathrm{u}}, l_{\hat{\mathrm{c}}}) = \prod_{m\in\{0,1\}} \chi(\sigma_{\mathrm{f,m}}, \mathrm{k}=1, \mathrm{s}=0.2) \prod_{x\in\{u,T,\hat{c}\}} \Gamma^{-1}(l_{\mathrm{x}}, \alpha=1, \beta=2), \tag{3.44}$$

where Γ^{-1} is the inverse Gamma distribution and χ is the chi distribution, which with $k=1$ is equal to the normal distribution truncated to the positive half-plane. The MAP estimates of hyperparameters were then found for each battery independently using the SciPy implementation of the limited memory Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm with box constraints [179], which is a gradient-based second-order optimisation algorithm (quasi-Newton

type). An alternative to using priors with strictly positive support (3.44) in conjunction with a constrained optimiser would have been to define priors over hyperparameters in the logarithmic domain, which potentially makes the optimisation problem easier. This type of approach is considered in Chapter 4, albeit with uniform priors.

To improve the stability and speed of calculating gradients, the gradient of the energy function was determined analytically by extending the recursive method outlined by Mbalawata et al. [69] to include terms for the observation model, equations (3.43). Differentiating the energy function with respect to θ_h and dropping the dependency on θ_h in notation for convenience,

$$\begin{aligned}\frac{\partial \phi}{\partial \theta_\mathrm{h}} = &- \frac{\partial \log p(\theta_\mathrm{h})}{\partial \theta_\mathrm{h}} + \frac{1}{2}\sum_n \mathrm{Tr}\left(\mathbf{S}_\mathrm{t}^{-1}\frac{\partial \mathbf{S}_\mathrm{t}}{\partial \theta_\mathrm{h}}\right) \\ &- \frac{1}{2}\sum_n \left(\mathbf{H}_\mathrm{t}\frac{\partial \mathbf{x}_\mathrm{t}^-}{\partial \theta_\mathrm{h}} + \frac{\partial \mathbf{H}_\mathrm{t}}{\partial \theta_\mathrm{h}}\mathbf{x}_\mathrm{t}^-\right)^\mathrm{T}\mathbf{S}_\mathrm{t}^{-1}\mathbf{e}_\mathrm{t} \\ &- \frac{1}{2}\sum_n \mathbf{e}_\mathrm{t}^\mathrm{T}\mathbf{S}_\mathrm{t}^{-1}\frac{\partial \mathbf{S}_\mathrm{t}}{\partial \theta_\mathrm{h}}\mathbf{S}_\mathrm{t}^{-1} \\ &- \frac{1}{2}\sum_n \mathbf{e}_\mathrm{t}^\mathrm{T}\mathbf{S}_\mathrm{t}^{-1}\left(\mathbf{H}_\mathrm{t}\frac{\partial \mathbf{x}_\mathrm{t}^-}{\partial \theta_\mathrm{h}} + \frac{\partial \mathbf{H}_\mathrm{t}}{\partial \theta_\mathrm{h}}\mathbf{x}_\mathrm{t}^-\right),\end{aligned} \tag{3.45}$$

where Tr denotes the matrix trace and

$$\frac{\partial \mathbf{S}_\mathrm{t}}{\partial \theta_\mathrm{h}} = \frac{\partial \mathbf{H}_\mathrm{t}}{\partial \theta_\mathrm{h}}\mathbf{P}_\mathrm{t}^-\mathbf{H}_\mathrm{t}^\mathrm{T} + \mathbf{H}_\mathrm{t}\frac{\partial \mathbf{P}_\mathrm{t}^-}{\partial \theta_\mathrm{h}}\mathbf{H}_\mathrm{t}^\mathrm{T} + \mathbf{H}_\mathrm{t}\mathbf{P}_\mathrm{t}^-\left(\frac{\partial \mathbf{H}_\mathrm{t}}{\partial \theta_\mathrm{h}}\right)^\mathrm{T} + \frac{\partial \sigma_{\mathrm{n,t}}^2}{\partial \theta_\mathrm{h}}. \tag{3.46}$$

The summation (3.45) can be done iteratively in the KF recursion and has to be done separately for each hyperparameter in θ_h. Details of this may be found in the Appendix B.2. In addition to an improvement in stability compared to numerical methods, the analytical gradient was also $\approx$30% faster to evaluate in this case.

3.2.4 Results

The method described in the previous section was applied to the fleet of 1027 VRLA batteries to first calculate MAP estimates for the kernel function hyperparameters, followed by calculating

the smoothing distribution. Distributed computing was used by creating an Apache Spark™ cluster running 30 cores on a virtualized Linux platform with 2×Intel(R) Xeon(R) Silver 4216 CPU @ 2.1 GHz physical processors. As a result, fitting hyperparameters to the 40 million rows of data took approximately 80 minutes.

Individual hyperparameter estimates

The distribution of hyperparameter MAP estimates for the fleet is shown in Fig. 3.8. Using the ARD kernel to describe the dependency on the instantaneous operating point it can be seen that in the median case the applied current is the input affecting R_0 the most, since its length scale is the shortest, whereas the dependency on temperature is the smallest (in the observed temperature range). Hyperparameters were estimated on normalised input data, so each length scale is relative to the observed range of each variable in the data. It should also be noted that, especially in the case of temperature and SOC, there are some outliers in hyperparameter estimates indicating that in those cases the GP hypers imply no relevance for those two inputs.

Assessing the effect of operating point on R_0

The magnitudes of the hyperparameters give some indication of the sensitivity of R_0 to operating conditions, but a better estimate of the effect of each input can be made by estimating the function at the population level. This was done by fitting another Gaussian process over R_0 estimates calculated for individual batteries. In other words, a population level function was estimated, given by

$$\mathbb{E}[p(R_0)] = k(\mathbf{x}^*, \mathbf{X}) \left[k(\mathbf{X}, \mathbf{X}) + \sigma_\mathrm{n}^2 \mathbf{I} + \sigma_\mathbf{X}^2\right]^{-1} \mu_\mathbf{X} \tag{3.47}$$

$$\mathbb{V}[p(R_0)] = k(\mathbf{x}^*, \mathbf{x}^*) - k(\mathbf{x}^*, \mathbf{X}) \left[k(\mathbf{X}, \mathbf{X}) + \sigma_\mathrm{n}^2 \mathbf{I} + \sigma_\mathbf{X}^2\right]^{-1} k(\mathbf{X}, \mathbf{x}^*), \tag{3.48}$$

where $\mathbf{X}$ is a 1027×3 matrix containing the mean current, SOC and temperature points of each battery, $\mu_\mathbf{X}$ the battery-wise mean R_0 estimates at locations $\mathbf{X}$, and $\sigma_\mathbf{X}^2$ the corresponding

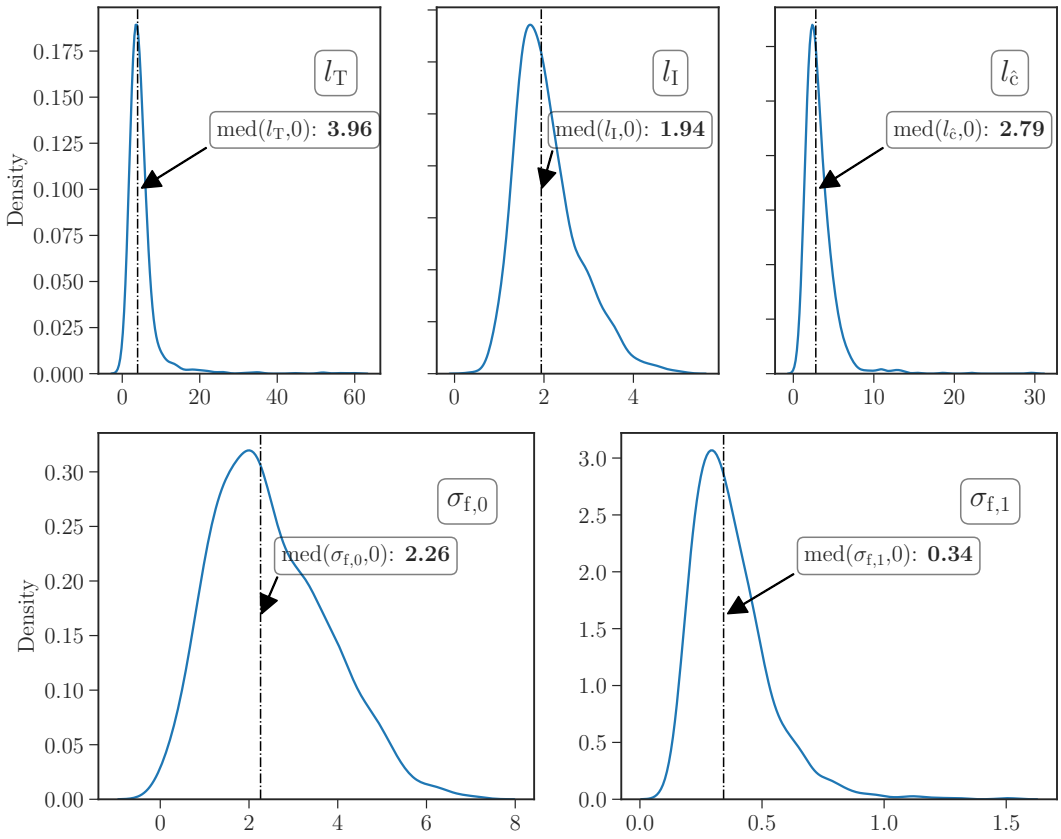


Fig. 3.8 Kernel density estimates of hyperparameter distributions across VRLA battery fleet, where MAP estimates of hyperparameters were calculated independently for each battery. Median values of MAP estimates across the fleet are indicated by the dashed vertical lines for each hyperparameter. For length scale l, shorter values imply more relevance of the input.

variance estimates. The kernel function used was again the SE kernel with ARD (3.34). A hyperprior similar to the individual battery case (3.44) was used, with an adjustment due to the lack of dependency on time, so that the hyperprior was given by

$$p(\sigma_f, \sigma_n, l_T, l_I, l_{\hat{c}}) = \chi(\sigma_n, k = 1, s = 0.1)\chi(\sigma_f, k = 1, s = 0.2) \prod_{x \in \{I,T,\hat{c}\}} \Gamma^{-1}(l_x, \alpha = 1, \beta = 2). \tag{3.49}$$

The posterior/predictive distributions are shown in Fig. 3.9, along with the distributions of the associated variables in the downsampled dataset. It is clear that the dependence on the applied current is the strongest, causing a total range in R_0 of 1.58 Ω between the 5th and 95th of the

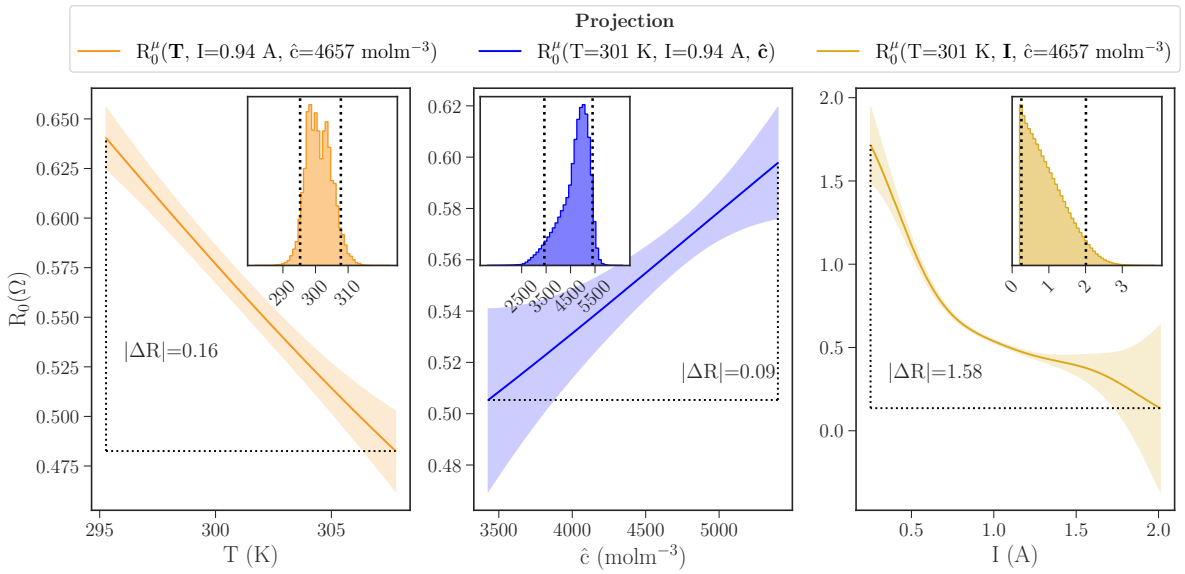


Fig. 3.9 Internal resistance varies significantly with operating conditions, as shown by projections of estimated R_0 as a function of temperature, applied current and SOC (acid concentration) between 5th and 95th percentiles of each. Shaded regions show +/- 2σ credible intervals. Insets show histograms of the independent variables for the down-selected dataset.

operating range. The resistance ranges for temperature and SOC variations are 0.16 Ω and 0.09 Ω respectively.

Standardised GP posterior distributions

To compare the health estimates between the different batteries in the VRLA fleet, it was necessary to take into account the strong dependence of R_0 on operating conditions. The average operating conditions vary between batteries and over lifetime. Therefore, to give a consistent health metric for each battery in the fleet, GPR was used to extrapolate estimates of R_0 to a constant operating point. This point was chosen to be the mean operating point over the entire fleet, where $T = 301$ K, $u = 0.94$ A and $\hat{c} = 4657\,\mathrm{mol\,m^{-3}}$. This extrapolation was performed simply by applying the standard predictive equations of the GP (3.5) using as inputs the mean and covariance estimates of the smoothed state vector $\mathbf{x}_{\mathrm{OP}}$ (which is constant over time; see Fig. 3.7). The effect of this normalisation is shown in Fig. 3.10. As expected,

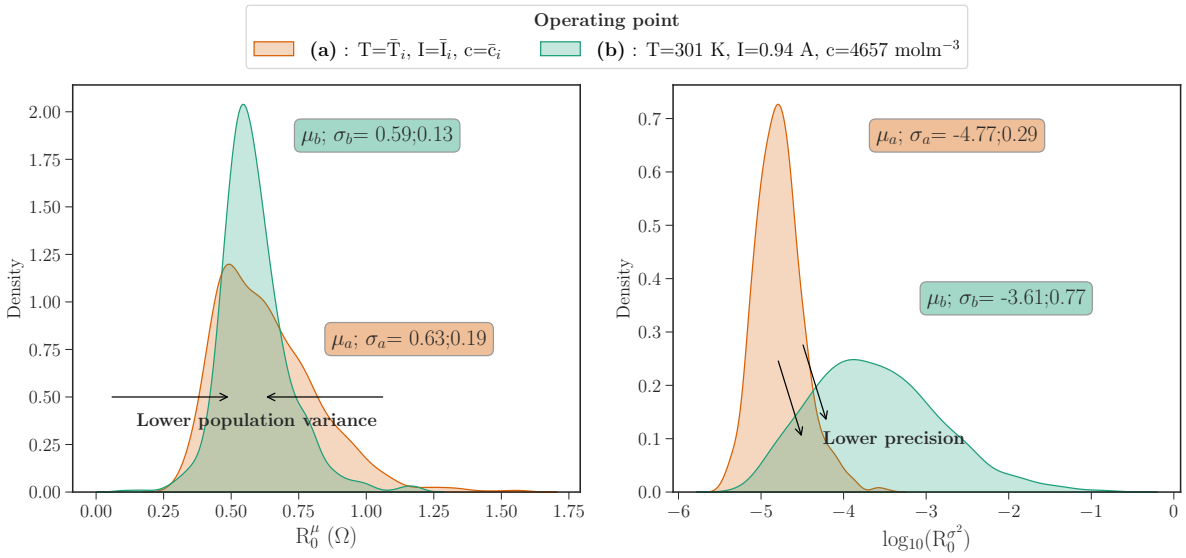


Fig. 3.10 There is a trade-off between using a single population-wide calibration point vs. a local calibration point per battery. Left-hand image shows lower variance in estimated mean of R_0 values with a single calibration point. However, in this case the uncertainty associated with each R_0 estimate increases, as right-hand image shows.

standardising the operating point reduced the variance in the mean estimates. The standard deviation in R_0 is 38% lower.

However, since GP was used to extrapolate away from the mean operating point of each individual battery, the corresponding precision of the estimates is lower (i.e. have higher variance), as the average predictive standard deviation increased from 0.09 Ω to 0.16 Ω.

Following standardisation, the smoothed and calibrated R_0 timelines at the chosen operating point for the entire fleet are shown in Fig. 3.11. From these, a vector-valued health metric was calculated, consisting of both the absolute value of R_0 and its local time derivative $\partial R_0/\partial t$. Both were easily retrieved from the state-space representation (3.35), as they constitute the state vector for the WV process.

These smoothed health trajectories may be compared against the benchmark case, where R_0 was estimated without dependency on the operating point and without imposing the WV kernel over its lifetime, Fig. 3.12. The timelines of this benchmark 'random walk' approach, common in the literature [31, 33], were estimated in the same KF framework as above. The

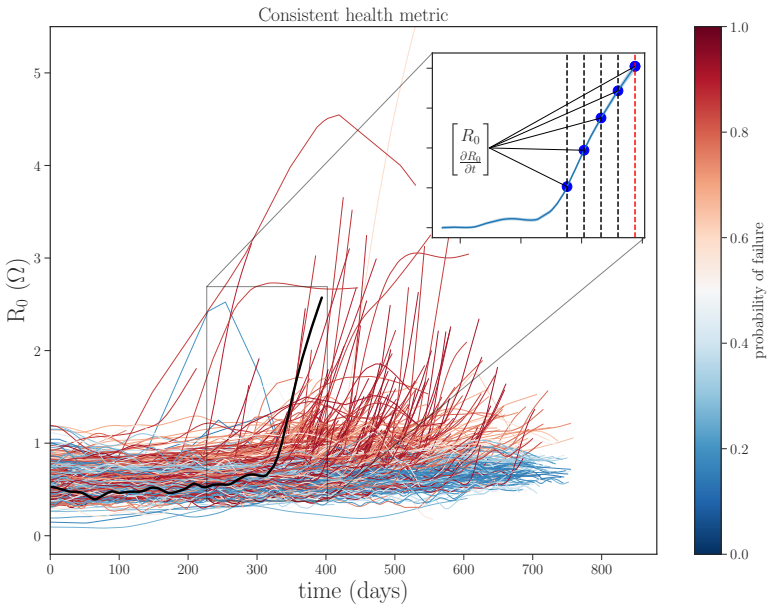


Fig. 3.11 R_0 timelines for all batteries in BBOXX dataset over their lifetimes. Inset: the overall health indicator than can be derived from the timeline includes both the absolute value of R_0 and its rate of change over time. The color of each indicates schematically which batteries are expected to be near to end of life.

absence of operating point dynamics and smoothing kernel function over time means that the discrete time transition matrix $A = 1$ and R_0 become scalars, and the only hyperparameter to be estimated is the discrete time (scalar) process noise Q, because the measurement noise was fixed similarly to the base case. Maximum-likelihood estimates of the process noise (i.e. using an noninformative, or uniform, hyperprior) were retrieved again using the L-BFGS-B algorithm. It is clear that the GPR estimation method using the additive kernel describing both the dependency of R_0 on time and operating point results in more stable estimates. In the next section, the GPR health metric is validated and its performance compared to the benchmark.

3.2.5 Validation and forecasting failure

Having created a stable health metric R_0 over the lifetime of each battery in the BBOXX dataset, validation had to be performed indirectly. This was due to the absence of reference performance tests during the lifetime of the batteries—such tests are routinely undertaken in

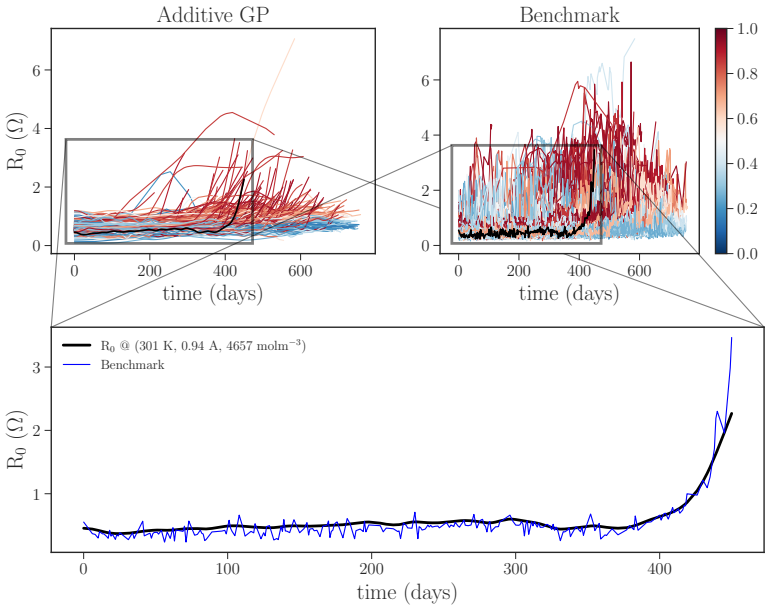


Fig. 3.12 Comparison of GPR estimates for R_0 to benchmark case: using GPR with kernel functions describing dependency on both operating point and time yield more stable estimates for over the lifetime of each battery.

laboratories but not in real-world operating conditions. Therefore, validation of the health metric is performed by correlating its value with the probability that a battery is at, or close to, its end-of-life point—as defined by independent checks from failed batteries in the field. End-of-life failure was defined as the point at which a given unit enters repair. BBOXX record repair dates and times in their database along with failure symptoms. There are many possible reasons for failure, but here the focus was on the cases where the maintenance team diagnosed a loss of capacity when a unit entered repair. Of the 1027 batteries in the dataset, 491 were diagnosed with loss of capacity at the end of the telemetry time series and 536 were healthy, i.e. customers had not reported a fault. Therefore, these repair data constituted a labelled binary outcome for batteries, which could be used to train a GP classifier, which took as inputs factors that affect the likelihood of the battery suffering from loss of capacity. It should be noted that the labelling itself is a significant source of uncertainty because customer-driven reporting of faults is subjective—the labelling therefore likely contains a non-negligible number of false positives and negatives.

In addition to validating the estimated health metric, the classification task was used to provide further insight into the ageing of batteries. This was done by including not only the health indicator as the input to the classifier, but several stress factors known to affect cell life [176] in lead-acid systems.

GP Classification of end-of-life failure

To predict end-of-life failure, many test cases were considered. First, the validity of the health metric calculated in the previous section was assessed by using it to predict failure at several stages towards end of life. This was done by first using the full time series for each battery, followed by successively removing 2, 4, 6 and 8 weeks of data from the end of each time series, thus determining to what extent the health metric could be used to forecast end of life beyond the end of the data. The benchmark (random walk case) was run also as a comparison in each case. In addition, an input matrix consisting solely of stress factors (as outlined below) was used in the same scenarios. Finally, an augmented input matrix consisting of both the health metric and the stress factors was used.

For each input data forecasting horizon, stratified 5-fold cross-validation (CV) was used in a nested manner. First, CV was used to split the 'master' dataset of 1027 batteries into training and test sets with a 48/52% failed/healthy split. Additionally, extra testing subsets within each test set were created that had different proportions of failed vs. healthy batteries. These were created by taking 10 random samples of 40% , 60% and 80% of the existing failed batteries within the test set while keeping the healthy set fixed, making the sub-test sets smaller in size. Altogether there were 3100 test cases, where

$$3100 = (4 \text{ input sets}) \times (5 \text{ CV sets}) \times (5 \text{ time horizons}) \times (31 \text{ test sets}).$$

Calculating stress factors for GP classifier

In addition to the resistance as a health metric, six cumulative stress factors that affect lifetime in VRLA systems were also included as inputs into the GP classifier to investigate performance improvements by including more information. Calendar age was defined as the time (in days) since the unit was marked as *activated* by BBOXX. Cumulative time at float charge was calculated by simply summing time increments conditional on the battery being in the float charge region, so that

$$\Sigma_S = \sum_t S(t)\Delta t \quad \text{where } S = \begin{cases} 1 & \text{if in float charge} \\ 0 & \text{otherwise} \end{cases}, \tag{3.50}$$

where the float charge region was defined as a period of more than 600 seconds where the voltage is between 13.4 V and 13.6 V. Total charge throughput was calculated by taking the absolute cumulative Coulomb count through lifetime, and the mean temperatures and voltages by using an expanding window from beginning of life. Finally, the discharge cycle count was calculated by adding up the discharge segments defined by any continuous discharge period greater than 600 s where $I > 0.05$ A.

To check for colinearity in the inputs, their cross-correlation was calculated, shown in Fig. 3.13. In most cases, correlations were below 40%. However, correlation within the health metric $[R_0 \; \partial R_0/\partial t]$ was high at 65%, implying the presence of an 'elbow point' [180] in the SOH estimate analogous to the 'knee point' often observed in capacity estimates [96], where there is a sudden acceleration of degradation. There was also a cross-correlation of over 60% between the mean voltage, total charge throughput and time spent in floating charge, which is expected, as the relatively high floating voltage of 13.5 V (comparing to the average) coincides with periods of very low current. Altogether, all the inputs added value to the forecasts because none of their cross-correlations were excessive.

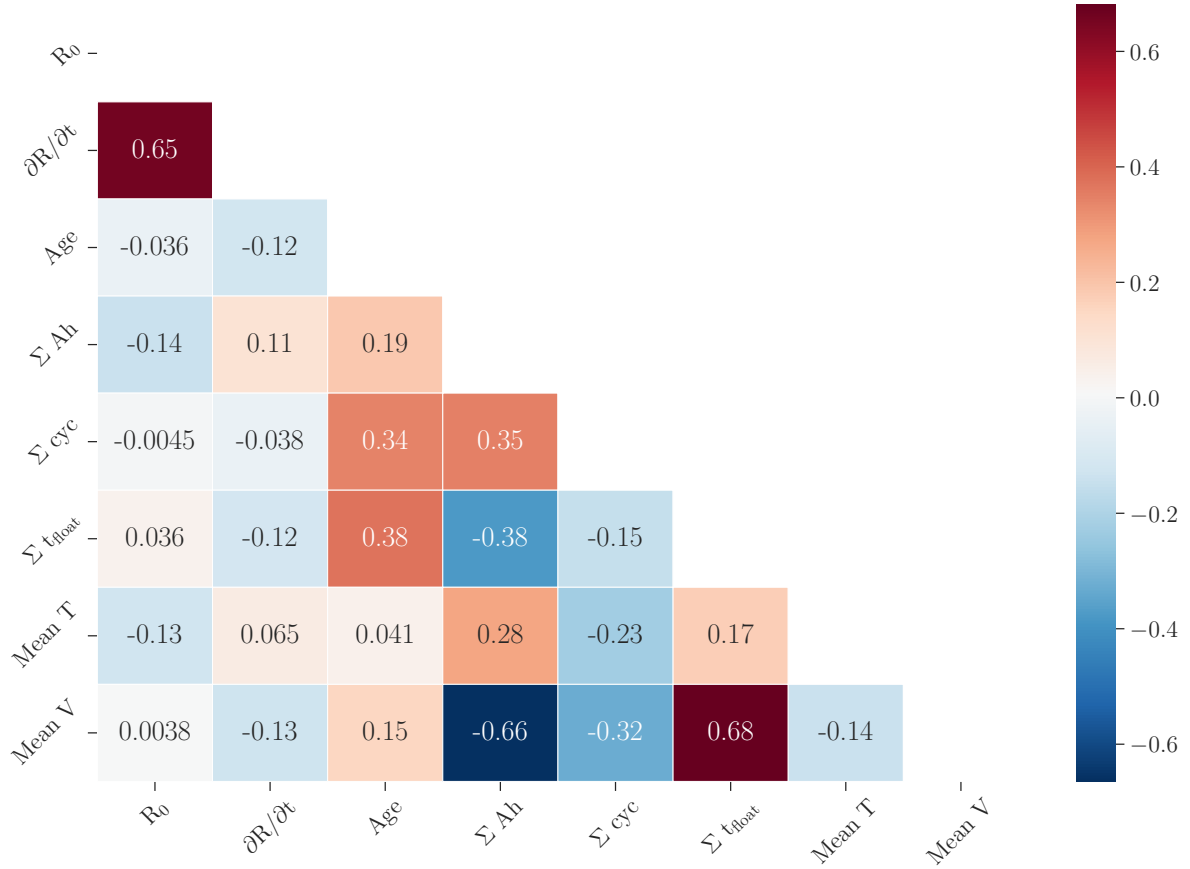


Fig. 3.13 Cross-correlation for classifier inputs used to forecast end-of-life failure.

Classification results

The various combinations of inputs and time horizons were tested using a standard GP classifier with the approximations outlined in Section 3.1.2, implemented in the `scikit-learn` package [150]. The chosen kernel function was the SE kernel with ARD using a uniform hyperprior. As before, using the ARD kernel gave an indication of the relative importance of the inputs in the classification task. The performance of the classifier was calculated in terms of the balanced accuracy, which is the average of its sensitivity and specificity,

$$\text{Balanced accuracy} = \frac{1}{2}(\text{Sensitivity+ Specifity}) = \frac{1}{2}\left(\frac{\text{TP}}{\text{TP+FN}} + \frac{\text{TN}}{\text{TN+FP}}\right), \tag{3.51}$$

where TP, FP, TN, FN are the true positive, false positive, true negative and false negative counts in the test set, respectively. Balanced accuracy is a performance metric that takes into account the number of real positives and negatives in the classification task.

Classifier performance aggregated over input case and forecast horizon is illustrated in Fig. 3.14 (breakdown in terms of true positives and negatives can be found in B.3). The bars indicate the $\pm 1\sigma$ range over all test cases. The best overall performance came from using

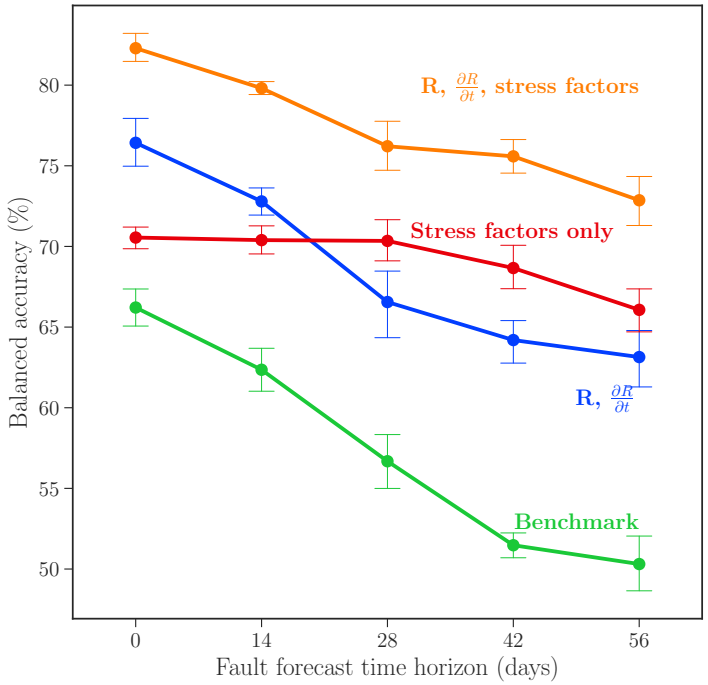


Fig. 3.14 Average performance of GP classifier for forecasting end-of-life failure for test cases, aggregated by input set and forecast horizon.

the augmented input dataset, that is the health metric combined with stress factors, giving an average balanced accuracy of 82% at the point of failure. This then drops to 73% when classifying faults 8 weeks ahead of time. The decrease in performance is likely due to the drop in predictive accuracy of the resistance and resistance gradient, which is evident from looking at the performance of the classifier using just the health metric on its own, where accuracy drops from 76-63% in the same time horizon. This is again related to the elbow point in internal resistance, highlighting the fact that resistance (and its time-gradient) are less informative

before the elbow. In comparison, using the stress factors alone only has less sensitivity of performance over the 8 week horizon, going from 71% to 66% balanced accuracy. For all cases, the benchmark method, where R_0 was assumed a random walk with no dependency on operating point, gives significantly poorer performance with a best case of 66% accuracy at point of failure dropping to 50% (equal to random classification) 8 weeks ahead of time.

In addition to classification performance, the length scales of the inputs in the augmented case give an indication of the relative importance of each input in distinguishing failure in the '0 week' case (i.e. at point of failure). Shown in Fig. 3.15, it is clear that the absolute value of R_0 is indeed the most important input, which is corroborated by the results in Fig. 3.14. Of the stress factors, average temperature and voltage are the most important. Age, cycle count, total charge throughput and total time spent in floating charge are the least important, although cross-correlations may still distort this result. Temperature is known to have a complex impact on lead-acid degradation, and the full dataset spans more than 20 °C range. Elevated temperatures, especially during charging, may improve lifetime due to improved solubility of lead sulphate [23], although they also increase electrode grid corrosion. Similarly, the relationship with mean voltage is complicated—a low average might indicate sulfation within the battery, but high voltages cause anodic corrosion [23] (see discussion on degradation mechanisms). While GP classification gives an indication of the importance of an input, it is difficult to determine whether high or low temperatures/voltages cause a higher probability of end-of-life failure as the cause could be a non-trivial combination of the two working in combination with other inputs.

Sensitivity to data availability

Data downsampled from raw telemetry data included current, voltage and temperature from all qualifying charging segments interpolated to a 1-minute time grid as specified in Section 3.2.2. To check the sensitivity of the method against poorer availability of data, two situations can

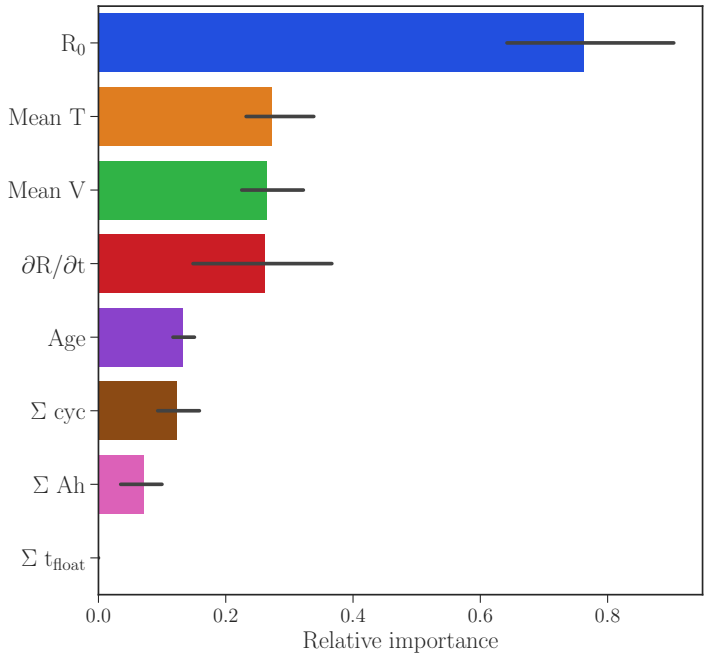


Fig. 3.15 Relative importance of ageing factors and health metric in forecasting failure in the '0 week', point of failure case. Grey bars indicate the $\pm 1\sigma$ in each training case.

be considered: (a) fewer qualifying segments available over the lifetime of the battery or (b) data within those segments are sparser. A sensitivity analysis was performed to account for both of these possibilities by specifying a lower interpolation frequency for the raw telemetry and choosing fewer charging segments from the available ones. The results of this analysis, shown in terms of the effect they have on classification performance, are shown in Table 3.2. Reducing the interpolation frequency did not have a substantial effect on performance—the largest effect was reducing the number of segments from an average of 260 per battery to 50, in which case the classification performance dropped by 2% in the augmented input case. Therefore, it is reasonable to conclude that the method is relatively insensitive to missing data.

3.2.6 Conclusions

In this chapter, a computationally efficient GPR framework was developed and used in conjunction with a simple electrical circuit model to calculate stable SOH metrics directly from

Test case	Data freq. (min)	Segments	Classifier mean balanced accuracy (%)			
			Benchmark	R, $\frac{\partial R}{\partial t}$	R, $\frac{\partial R}{\partial t}$, stress fac.	Stress fac.
baseline	1	∼260	57	69	77	69
100 segments	1	100	58	67	76	69
50 segments	1	50	57	66	75	69
2 minute	2	∼260	58	71	78	69
5 minute	5	∼260	58	70	78	70
10 minute	10	∼260	58	70	78	70

Table 3.2 Estimates of sensitivity of classifier performance to data availability, simulated by reducing interpolation frequency and/or number of qualifying charging segments.

telemetry data from VRLA batteries under real-world operating conditions in sub-Saharan Africa. The framework was applied to data from a total of 1027 batteries with lifetimes ranging from 400 to 760 days. It was shown how assuming an appropriate GP prior over the internal resistance can produce a corresponding posterior-predictive distribution describing the dependency of internal resistance on time as well as instantaneous operating conditions such as SOC, applied current and temperature. Compared to a benchmark case where internal resistance was assumed to be a random walk over lifetime and not dependent on operating conditions, it was shown how the GP method provides more stable estimates.

The internal resistance estimates were then validated against repairs data from the battery provider BBOXX by correlating the value of the health metric with the probability of end-of-life-type failure in the near future, obtaining a balanced accuracy of 76%. However, by adding extra information to the prediction—in the form of the cumulative stresses experienced by each battery over its lifetime—an accuracy of 82% was obtained when predicting whether a battery has failed. This accuracy remains 73% 8 weeks before failure. Considering that the labelling of faulty batteries was likely to cause a decrease in accuracy (due to it being customer driven), the accuracy is high, particularly compared to the benchmark case whereby an accuracy equal to random classification was obtained.

In general, this chapter shows that combining data-driven and model-driven methods provides a reasonable framework for the diagnosis and prognosis of battery health in real-world

operating scenarios. In addition, the results remain interpretable from the point of view of battery models based on physics. Additionally, because of the simplicity of the underlying model, the method could be equally applied to other chemistries (such as Li-ion) in varying configurations, from the cell to pack level. The only prerequisite is knowledge of the OCV curve, which in the case was obtained for a single battery under laboratory conditions.

Potential future work related to this chapter might involve building a more accurate predictive model of health, whereby population-level data could be used to give a prior expectation on the evolution of health given operating conditions (i.e. a non-zero mean for the GP), which can be adjusted with observations to again yield a predictive distribution. Ultimately, control algorithms could be made smarter and more health aware by incorporating this information.

The value of early prediction of failure in the case of BBOXX comes from timely management of battery replacement and optimal management of supply chains, which combine to increase the reliability and reduce the cost of providing home solar battery energy storage systems in remote locations. As no additional sensors are required on the devices themselves, the marginal cost of providing diagnosis and prognosis using this framework is small. Assuming that the cost savings involved can be passed on to end users, this type of diagnostic algorithm can contribute to an increasing adoption of home solar energy storage systems in the developing world, where there are still an estimated 800 million people living without electricity [11].

Chapter 4

State/parameter estimation with recursive Gaussian process regression

In the previous chapter, it was shown that computationally efficient recursive GP regression can be used at scale to calculate consistent health metrics for large fleets of batteries. Due to the characteristics of the available data, a very simple Thevenin circuit model was used, where the resistor was assumed to be a non-parametric function of lifetime and instantaneous operating conditions. In this chapter, the concept is extended and it is shown how a more complex electrothermal model may be similarly parameterised while simultaneously estimating states. Specifically, the electrical parameters of a first-order resistor-capacitor (RC) circuit model describing Li-ion cell electrical dynamics are retrieved as GP posteriors over lifetime and operating conditions, whilst also estimating state of charge, cell temperature and the voltage across the RC pair. The contents of this chapter are based on previous publications [2, 3].

In the previous chapter, a simplified Gaussian process structure was used to improve computational efficiency. This described the internal resistance of lead-acid batteries, decoupling the dependencies between degradation over time and instantaneous operating points. The aim of this chapter is to explore the case where this assumption is relaxed, in other words the operating point dependency may also change over time. The complexity of degradation in Li-ion cells is

such that it is reasonable to expect circuit parameters to evolve in a more complex manner than wholesale shifts of the entire function describing dependence on operating conditions. This is evident from simulation work relying on first principles [73], where physics-based models are coupled with degradation models and it can be seen that the latter affect the parameters of the battery over its lifetime. Degradation and this coupling are briefly discussed in the following to motivate the ECM + GP approach. Subsequently, a joint state/parameter estimation problem is formulated and simulated data are used to retrieve ground-truth functions for electrical parameters as GPs to illustrate the identifiability of the system. The algorithm is then applied to cell-level data collected under laboratory conditions [112], where validation is performed by comparing the algorithm's estimates of capacity and internal resistance with independent measurements of these over the lifetime of a cell.

4.1 Degradation in Li-ion cells

Similarly to lead-acid systems discussed in Chapter 3, degradation in Li-ion cells is complex and path-dependent, caused by many coupled mechanisms where the end result is a fade in both the discharge capacity and power a cell can deliver. There is a substantial body of literature on the analysis of Li-ion degradation mechanisms, their identification, and the evaluation of their relative effects over the useful lifetime of batteries [181].

Fig. 4.1 illustrates the interactions between stress factors, most common degradation mechanisms and degradation 'modes' (the initial symptoms of the mechanisms) in a Li-ion cell. The primary mechanisms include the following:

1. **Solid-electrolyte interphase (SEI) growth:** The SEI is a passivation layer (consisting of various compounds such as lithium carbonate) on the surface of the negative electrode formed by the reaction between the electrolyte and the negative electrode material. Following rapid formation at the beginning of life, it grows through the lifetime of the

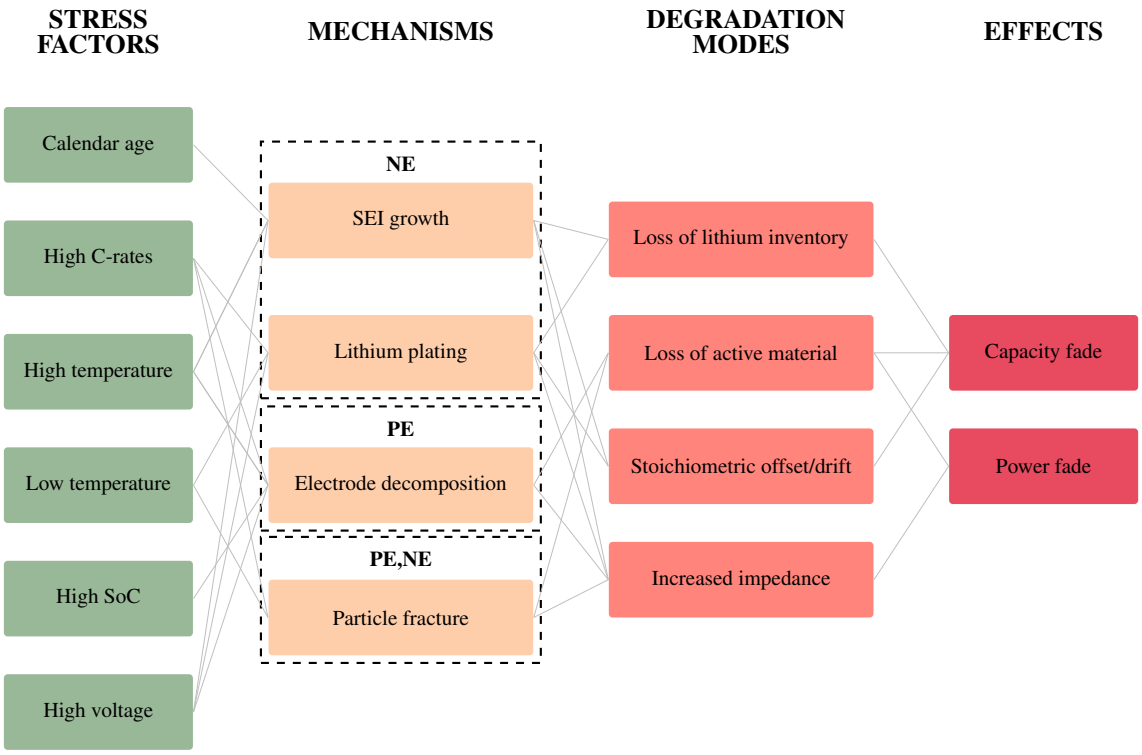


Fig. 4.1 Summary degradation mechanisms and modes in Li-ion cells [22, 181] and the stress factors associated with them. Different mechanisms on each electrode (denoted PE, NE) lead to multiple modes of degradation, which give rise to fade in cell capacity and its power capability. The connectors indicate causality.

battery, and is commonly associated with the 'calendar ageing' effect of Li-ion cells, resulting in loss of cycleable lithium inventory and capacity fade.

2. **Lithium plating:** A side reaction occurs on the negative electrode where instead of lithium intercalating into the negative electrode particles, it is plated as a metal on the surface. This occurs during the charging process and is exacerbated by high charging rates and/or low temperatures. Commonly associated with the cyclic ageing of cells.

3. **Particle fracture:** The breaking off of electrode particles on both electrodes is caused by volume changes during charge / discharge processes that result in mechanical stresses. This results in the loss of electrically active material. As with lithium plating, this degradation mechanism is associated with cycling of cells.

4. **Electrode structural decomposition:** Phase changes and decomposition due to the lack of electrochemical stability occurring (mainly) on the positive electrode can cause loss of active material and increased impedance due to poorer kinetics in the new phases formed [181].

These degradation mechanisms lead to four main symptoms or 'degradation modes', namely loss of lithium inventory (LLI) and loss of active material on each respective electrode ($\text{LAM}_{\text{NE/PE}}$). Briefly, LLI is the net effect of irreversible parasitic reactions, such as SEI growth and lithium plating, that consume usable lithium ions so that they cannot participate in further charge/discharge cycling; LAM in turn can be caused by particle cracking and/or loss of electrical coherence within the electrode due to the growth of surface layers. LLI and LAM also lead to the so-called stoichiometric offset, meaning that the state of lithiation of the positive electrode at full SOC is lower (i.e., it is at a higher potential) [22], which in turn can accelerate the degradation further. The rate of all degradation mechanisms is affected by operating and environmental conditions. Extremes in temperature, SOC and charging rates in particular may be associated with an increased degradation in cell performance [181]. It should be noted that Fig. 4.1 is intended only as an overview of Li-ion cell degradation—more detailed descriptions [181, 22] include broader sets of degradation mechanisms and address coupling between them.

A principled prediction of the effects of each degradation mechanism on cell performance over lifetime can be made by coupling degradation models with physics-based models such as the SPM [73]. The coupling is achieved by simultaneously modelling degradation side reactions and mechanical changes alongside the main battery dynamics. For example, in the SPM(e), the SEI reaction current can be added to the boundary conditions at the particle surfaces (2.12), and it can be assumed that the cumulative effect of SEI growth affects the volume fraction of negative active material by blocking pores [73]. Similar effects can be assigned to all degradation mechanisms that are specific to the electrodes and/or electrolyte.

Accurate physical modelling of degradation through these principled approaches is insightful and has been shown effective for digital twins and the optimisation of battery control algorithms [15, 182]. However, they remain open-loop methods, because the parameterisation of the coupled models suffers from poor identifiability not only of the battery model, but of the degradation model(s). As shown by Reniers et al. [73], many degradation models lead to very similar behaviour in battery capacity over lifetime, meaning that attributing observed capacity/power fade to each mechanism is not possible.

As an alternative to physics-based models for battery voltage response and degradation using porous electrode theory, the use of equivalent circuits with GPs to parameterise component values as functions offers a high degree of freedom that is parsimonious but preserves some physical meaning. The reduced accuracy of ECMs compared to PBROMs is mitigated by allowing circuit parameters to vary as functions of states. In this chapter, a first-order electrical RC-circuit is fully parameterised using the GP framework. By incorporating parameter variations, the resulting model is very similar to the so-called equivalent hydraulic model (EHM) [51, 52], a simplified physics-based model with a two-state representation that can be derived from the SPM. Given the similarity in dynamics and voltage response, the fundamental difference remaining between the EHM and the first-order GP + ECM RC circuit is that the latter uses a terminal-level (i.e. two electrode) open-circuit voltage function rather than decomposing it into individual contributions from negative and positive half-cells.

In addition to bridging the gap between ECMs and PBROMs, data-driven parameterisation with a high degree of freedom allows for a more insightful expression of degradation in model parameters. That is, by allowing the independent evolution of battery parameters as functions of states, changes can be linked to different degradation mechanisms to some extent. In this chapter, by using a multiplicative GP kernel over the input dimensions, rather than an additive one (as before), the surfaces describing circuit parameters are allowed to evolve in a more complex manner that describes degradation more accurately.

4.2 Parameterisation of an electrothermal model using a joint GP/state estimator

4.2.1 Low order electrothermal model

The electrical and thermal model a Li-ion cell is shown in Fig. 4.2. It consists of a first-order RC electrical circuit with a coupled one-dimensional lumped thermal model, where all four electrical parameters are considered GPs over state of charge and/or applied current, as well as lifetime, denoted ζ_t. The temperature dependence could also be added, but since the cell data used [112] in this study only has a range of 5 °C, this is not considered it here. The thermal model consists of heat generation due to the total overpotential (i.e. voltages across the series resistor and parallel RC pair) with convection to the ambient environment, making the assumption that heat conduction through the cell is fast (i.e. low Biot number), such that the cell internal temperature is relatively uniform.

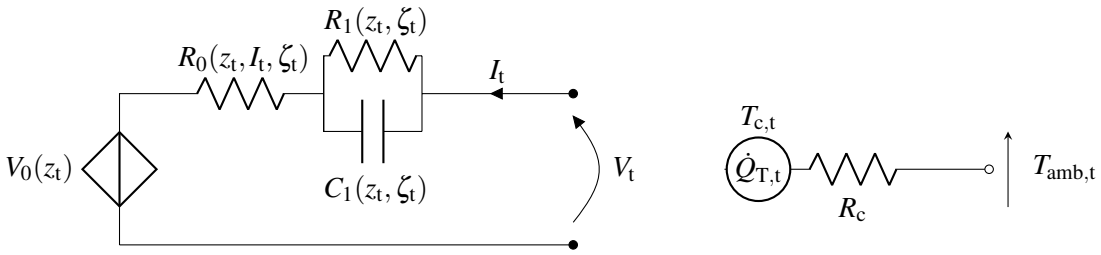


Fig. 4.2 Li-ion cell model: first-order RC circuit coupled with a one-dimensional thermal model. All electrical parameters are modelled as Gaussian processes.

The (nonlinear) continuous-time dynamics of the 3-state electrothermal model in Fig. 4.2 are given by a state-space representation

$$\begin{aligned} \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} &= f_B(\mathbf{x}_t, I_t, \theta) \\ \mathbf{y}_t &= h(\mathbf{x}_t, I_t, \theta), \end{aligned} \tag{4.1}$$

where the dynamics f_{B} for battery state vector $\mathbf{x}_{\mathrm{t}} = [z_{\mathrm{t}}\ V_{1,\mathrm{t}}\ T_{\mathrm{t}}]^{\mathrm{T}}$ are given by

$$\begin{aligned}\frac{\mathrm{d}z_{\mathrm{t}}}{\mathrm{d}t} &= I_{\mathrm{t}} Q^{-1}(\zeta_{\mathrm{t}}) \\ \frac{\mathrm{d}V_{1,\mathrm{t}}}{\mathrm{d}t} &= -\alpha(z_{\mathrm{t}},\zeta_{\mathrm{t}})V_{1,\mathrm{t}} + \beta(z_{\mathrm{t}},\zeta_{\mathrm{t}})I_{\mathrm{t}} \\ \frac{\mathrm{d}T_{\mathrm{c,t}}}{\mathrm{d}t}C_{\mathrm{c}} &= -\frac{T_{\mathrm{c,t}} - T_{\mathrm{amb,t}}}{R_{\mathrm{c}}} + V_{1,\mathrm{t}}I_{\mathrm{t}} + R_0(z_{\mathrm{t}},I_{\mathrm{t}},\zeta_{\mathrm{t}})I_{\mathrm{t}}^2,\end{aligned} \tag{4.2}$$

where z_{t} is the state of charge, I_{t} the applied current (positive for charging) and $Q^{-1}(\zeta_{\mathrm{t}})$ the inverse battery capacity as a function of lifetime ζ_{t}. Lifetime ζ_{t} can be measured by calendar age or by total charge throughput over lifetime. Voltage $V_{1,\mathrm{t}}$ is across the RC pair, and its time dynamics are controlled by the functions $\alpha(z_{\mathrm{t}},\zeta_{\mathrm{t}})$ and $\beta(z_{\mathrm{t}},\zeta_{\mathrm{t}})$. These are related to the circuit parameters, as $\alpha = 1/R_1C_1$ and $\beta = 1/C_1$. The thermal model is parameterised by its heat capacity C_{c} and thermal resistance R_{c}, which are considered known. Given these dynamics, the outputs $\mathbf{y}_{\mathrm{t}} = [V_{\mathrm{t}}\ T_{\mathrm{t}}]^{\mathrm{T}}$ are cell terminal voltage and temperature,

$$\begin{aligned}V_{\mathrm{t}} &= V_0(z_{\mathrm{t}}) + V_{1,\mathrm{t}} + R_0(z_{\mathrm{t}},I_{\mathrm{t}},\zeta_{\mathrm{t}})I_{\mathrm{t}} \\ T_{\mathrm{t}} &= T_{\mathrm{c,t}}\end{aligned} \tag{4.3}$$

The four functions $Q^{-1}(\zeta_{\mathrm{t}})$, $\alpha(z_{\mathrm{t}},\zeta_{\mathrm{t}})$, $\beta(z_{\mathrm{t}},\zeta_{\mathrm{t}})$ and $R_0(z_{\mathrm{t}},I_{\mathrm{t}},\zeta_{\mathrm{t}})$ are all assumed to be affine transformations of independent zero-mean Gaussian processes, so that

$$f \sim t_{\mathrm{f}}\left(\mathcal{GP}(0,k_{\mathrm{f}}(x,x'))\right)\ ,\ x = [z\ I\ \zeta]\ ,\ f \in \{Q^{-1},\alpha,\beta,R_0\}. \tag{4.4}$$

The affine transformation in each case is

$$t_{\mathrm{f}}(x) = c_{\mathrm{f}}(1+x), \tag{4.5}$$

where c_{f} is a constant. As each GP describing Q^{-1}, α, β and R_0 has a zero-mean, setting c_{f} effectively sets a nonzero prior mean for each circuit parameter—hence c_{t} should be chosen

so that it reflects the prior expectation of where the parameter lies. The reason for the transformation is to scale the system so that the GPs are in the unit range (making hyperparameter initialisation simpler) and to improve the numerical stability of the system dynamics (4.2) at the prior mean of the GP. As a Gaussian distribution remains Gaussian under arbitrary affine transformations, the four functions describing circuit parameters are also Gaussian processes.

4.2.2 Multiplicative GP kernel

In general, the kernel function k is constructed so that the 'extent of degradation' input, ζ_t (which for simplicity will considered to be cumulative charge throughput at time t in this work), is treated differently from the instantaneous operating condition inputs (z_t, I_t). Similarly to Chapter 3, a nonstationary kernel function describes each of the four Gaussian processes in the ζ_t dimension, which allows for better extrapolation than a stationary kernel which reverts back to the mean upon long-range extrapolation. The nonstationary kernel is again the Wiener velocity kernel,

$$k(\zeta,\zeta') = \sigma^2_{\zeta,\mathrm{s}}\left(\frac{\min^3(\zeta,\zeta')}{3} + |\zeta-\zeta'|\frac{\min^2(\zeta,\zeta')}{2}\right). \tag{4.6}$$

As before, the kernel describing the process over state of charge z_t and applied current I_t is the SE kernel with ARD,

$$k_{\mathrm{SE}}(\mathbf{x},\mathbf{x}') = \sigma^2_{\mathbf{x}}\exp\left(-\frac{1}{2}\sum_{x\in\mathbf{x}}\gamma_{\mathrm{x}}(x-x')^2\right),\ \mathbf{x}=[z\ I], \tag{4.7}$$

where γ_{x} are the inverse squared length scales over each dimension in $\mathbf{x}$.

Combining the kernels (4.6) and (4.7) is done by multiplying the two functions. In addition, a kernel describing short-term fluctuations over time in the parameters is added—this adds

equally to all points in $\mathbf{x}$, so that the overall kernel is

$$k_{\mathrm{f}} = \underbrace{k_{\mathrm{WV}}(\zeta,\zeta')k_{\mathrm{SE}}(\mathbf{x},\mathbf{x}')}_{\text{spatially resolved, smooth}} + \underbrace{k_{\mathrm{E}}(\zeta,\zeta')}_{\text{noise}}, \tag{4.8}$$

where k_{E} is the exponential kernel,

$$k_{\mathrm{E}}(\zeta,\zeta') = \sigma^2_{\zeta,\mathrm{r}} \exp\left(-\gamma_{\zeta,\mathrm{r}}|\zeta-\zeta'|\right), \tag{4.9}$$

where $\sigma_{\zeta,\mathrm{r}}$, $\gamma_{\zeta,\mathrm{r}}$ are the magnitude and inverse length scale of the noise process. Hence the evolution of parameters is decomposed to a longer-term smooth component and shorter-term fluctuations. When extrapolated, the short-term component decays quickly, so the extrapolation is smooth. Estimating the hyperparameter $\gamma_{\zeta,\mathrm{r}}$ from the data gives an estimate of the autocorrelation in the noise process. If $\gamma_{\zeta,\mathrm{r}}$ is high, then k_{E} effectively describes white noise over time.

4.3 Joint estimation of states and functional parameters

4.3.1 Discretisation and joint state vector

As seen in Chapter 3, to construct a finite-dimensional state-space representation of the GP described by kernel function (4.8), the input space $\mathbf{x}$ (i.e. in SOC and applied current) for each GP $f \in R_0, \alpha, \beta$ has to be discretised. To this end, n_{z} evenly spaced points over SOC (z) are chosen to represent the GP for α , β, which are only functions of SOC, and n_{zI} points for R_0, which is a function of both SOC and applied current. In other words, α, β and R_0 are each represented by a state vector, where each element corresponds to a point at a specific SOC and/or I. For R_0 the vector is effectively a stacked set of values at sampling points that represent a grid over discrete SOC and current values. Therefore the state vectors for the three GPs can

be written as

$$\begin{aligned}\mathbf{x}_{\alpha} &= \begin{bmatrix}\alpha_{z_1} & \alpha_{z_2} & \dots & \alpha_{z_{nz}}\end{bmatrix},\\ \mathbf{x}_{\beta} &= \begin{bmatrix}\beta_{z_1} & \beta_{z_2} & \dots & \beta_{z_{nz}}\end{bmatrix},\\ \mathbf{x}_{R_0} &= \begin{bmatrix}R_{0,(z_1,I_1)} & R_{0,(z_2,I_2)} & \dots & R_{0,(z_{nzI},I_{nzI})}\end{bmatrix}.\end{aligned} \tag{4.10}$$

A joint state vector may then be constructed to estimate the states of the GP and the battery simultaneously given the n_z and n_{zI} points in the input space for the respective GPs. Let the vector $\mathbf{x}_{\text{Batt,t}}$ denote the mean estimates of the battery states at time t,

$$\mathbf{x}_{\text{Batt,t}} = \begin{bmatrix}z_t & V_{1,t} & T_{c,t}\end{bmatrix}^{\text{T}}, \tag{4.11}$$

and the vector $\mathbf{x}_{\text{GP,s,t}}$ denote the state vector associated with the mean of each of the GPs describing the model parameters, so that

$$\mathbf{x}_{\text{GP,s,t}} = \begin{bmatrix}\mathbf{x}_{Q^{-1},t} & \mathbf{x}_{\alpha,t} & \mathbf{x}_{\beta,t} & \mathbf{x}_{R_0,t}\end{bmatrix}^{\text{T}}, \tag{4.12}$$

where $\mathbf{x}_{Q^{-1},t} \in \mathbb{R}^2$, $\mathbf{x}_{\alpha,t} \in \mathbb{R}^{2n_z}$, $\mathbf{x}_{\beta,t} \in \mathbb{R}^{2n_z}$, $\mathbf{x}_{R_0,t} \in \mathbb{R}^{2n_{zI}}$. The dimensionality of the state vectors is due to the number of points used for the discretisation of $\mathbf{x}$ in each case and the order of the Markov process due to the Wiener velocity kernel $k_{\text{WV}}(\zeta,\zeta')$. Specifically, the Wiener velocity kernel (4.6) has a dynamic representation [111] where

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}x\\ \frac{\mathrm{d}x}{\mathrm{d}t}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x\\ \frac{\mathrm{d}x}{\mathrm{d}t}\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}\omega(t), \tag{4.13}$$

where the spectral density of noise $\omega(t)$ is a function of $\sigma_{\zeta,\text{s}}$. This means that the kernel multiplication (4.8) results in each GP being represented by both the current state and its first-order time derivative at each spatial location in $\mathbf{x}$. This differs from the lead-acid case in Chapter 3, whereby the time derivative was considered common across all operating conditions.

The stationary exponential kernel (4.9), describing the short length scale noise process, has a single state representation, whereby

$$\frac{\mathrm{d}x}{\mathrm{d}t} = -\gamma_{\zeta,\mathrm{r}} x + \omega(t), \tag{4.14}$$

where the spectral density of the noise process $\omega(t)$ is a function of $\sigma_{\zeta,\mathrm{r}}$. In this case, the GP state vector for k_{E} is just

$$\mathbf{x}_{\mathrm{GP,r,t}} = \begin{bmatrix} x_{\mathrm{Q}^{-1},1,\mathrm{t}} & x_{\alpha,1,\mathrm{t}} & x_{\beta,1,\mathrm{t}} & x_{\mathrm{R}_0,1,\mathrm{t}} \end{bmatrix}. \tag{4.15}$$

The overall joint state/parameter system 'state' representation is then given by the concatenation of the battery and GP state vectors,

$$\mathbf{x}_{\mathrm{t}} = \begin{bmatrix} \mathbf{x}_{\mathrm{Batt,t}} \\ \mathbf{x}_{\mathrm{GP,s,t}} \\ \mathbf{x}_{\mathrm{GP,r,t}} \end{bmatrix}. \tag{4.16}$$

4.3.2 Initialisation and propagation

The joint system is nonlinear and may be propagated through time using an appropriate Bayesian filter. For computational speed, the extended Kalman filter is applied here. It is possible that other variants, such as the unscented Kalman filter or particle filter might provide more accurate results, but the EKF was considered adequate in initial tests using simulated data (Section 4.4). There are two timescales involved—the first is given by the sampling frequency of current, voltage and temperature data during cycling, which in this work is 1 Hz; the second is related to the extent of degradation, ζ, which covers the lifetime of the battery, and may be measured in cumulative charge throughput.

Initialisation

The initialisation of the two subsystems (i.e. for the battery states and GPs respectively) is consistent with the two timescales. A zero-mean GP is used to model the circuit parameters (nested inside the affine transformation (4.5)), which means that the initial mean estimates of the GPs are set so that $\mathbf{x}_{\mathrm{GP,t}} = \mathbf{0}$. The initial GP covariance matrix is block diagonal due to the assumption that all GPs are independent from each other, so that

$$\mathbf{P}_{\mathrm{GP},0} = \begin{bmatrix} \mathbf{P}_{\mathrm{Q}^{-1},\mathrm{WV},0} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{\alpha,\mathrm{WV},0} & \dots & \dots & \mathbf{0} \\ \vdots & \vdots & \mathbf{P}_{\beta,\mathrm{WV},0} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \mathbf{P}_{\mathrm{R}_0,\mathrm{WV},0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}_{\mathrm{E},0} \end{bmatrix}, \tag{4.17}$$

where the initial covariance for each 'smooth' (i.e. WV kernel) GP is given by the Kronecker relation

$$\mathbf{P}_{\mathrm{f,WV},0} = k_{\mathrm{f},\mathbf{x}}(\mathbf{U}_\mathrm{f}, \mathbf{U}_\mathrm{f}') \otimes \mathbf{P}_{\zeta_0,\mathrm{WV}}, \tag{4.18}$$

where $\mathrm{f} \in \{Q^{-1}, \alpha, \beta, R_0\}$ and $\mathbf{U}_\mathrm{f}$ are the coordinates of the discrete points chosen for each kernel function (i.e. the coordinates of (4.10)). This is a discrete representation of the initial covariance of the spatially resolved white noise process.

The initial covariance for the Wiener velocity process, $\mathbf{P}_{\zeta_0,\mathrm{WV}}$, is in this case determined by hyperparameters. In the standard formulation, the WV kernel (4.6) has zero covariance at $\zeta = 0$, resulting in the posterior estimate of the GP to be zero as well. However, the parameters of the circuit model are nonzero at the beginning of life. In the additive system used in Chapter 3, the WV process was chosen to model the degradation process only—in this case, initialising the process at zero at beginning of life is reasonable. However, this is not the case with the multiplicative kernel under discussion here, since circuit parameters have to be nonzero

at beginning of life. To reconcile this and to obtain a WV kernel with nonzero variance at beginning of life, the kernel may be 'truncated' by setting ζ_0 to a nonzero value. This is equivalent to shifting the starting point of the prior to the right in the nonstationary case in Fig. 3.1. To find the correct starting conditions in the state-space representation, first (3.24) is evaluated at ζ_0, giving

$$\mathbf{P}_{\zeta_0,\mathrm{WV}} = \exp(\mathbf{F}\zeta_0)\mathbf{P}_0\exp(\mathbf{F}\zeta_0)^{\mathrm{T}} + \int_0^{\zeta_0}\exp(\mathbf{F}t)\mathbf{L}\mathbf{q}\mathbf{L}^{\mathrm{T}}\exp(\mathbf{F}t)^{\mathrm{T}}\,\mathrm{d}t, \tag{4.19}$$

where the Wiener velocity kernel state-space system is described by (4.13), so that

$$\mathbf{F} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \ \mathbf{L} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ \mathbf{q} = \sigma^2_{\zeta,\mathrm{s}}, \ \mathbf{P}_0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \tag{4.20}$$

where the value of $\mathbf{P}_0$ (by definition) reflects the zero variance of the WV kernel at $\zeta = 0$, analogous to knowing the exact position and 'velocity' of the random process at the start of life. Substituting (4.20) into (4.19) then gives

$$\mathbf{P}_{\zeta_0,\mathrm{WV}} = \sigma^2_{\zeta,\mathrm{s}}\begin{bmatrix} \frac{1}{3}\zeta_0^3 & \frac{1}{2}\zeta_0^2 \\ \frac{1}{2}\zeta_0^2 & \zeta_0 \end{bmatrix}. \tag{4.21}$$

The value of ζ_0 must be found indirectly by considering the variance in circuit component values at beginning of life and then back-calculating the ζ_0 that is consistent with the WV process. The two-stage hyperparameter optimisation process (see section 4.3.4) first estimates the initial covariance at beginning of the GP describing the dependence on operating conditions (4.7) (i.e. the upper-left hand element in (4.21)), which can then be used to define the rest of the elements in $\mathbf{P}_{\zeta_0,\mathrm{WV}}$. Using this method, this gives the correct initial covariance of the

state-space system that describes the WV process at $\zeta > 0$. For the noise process,

$$\mathbf{P}_{\mathrm{E},0} = \sigma^2_{\zeta,\mathrm{r}}. \tag{4.22}$$

The GP mean and covariance describing the parameters only have to be initialised once for each battery. The battery states on the other hand have to be re-initialised whenever there is a gap in telemetry data. For parameter estimation, not all data are required because battery degradation is slow compared to the sampling frequency. Therefore, it is unnecessary to use all recorded data, and instead just a smaller number of specific charge/discharge cycles may be selected. In this work, the simulated and experimental datasets begin with a rest period, so the mean vector for battery states may be initialised as

$$\mathbf{x}_{\mathrm{Batt},0} = \begin{bmatrix} V_0^{-1}(V_\mathrm{t}) \\ 0 \\ T_{\mathrm{amb,t}} \end{bmatrix} \tag{4.23}$$

at the start of each cycle. The state covariance is initialised with fixed values, so that

$$\mathbf{P}_{\mathrm{Batt},0} = \begin{bmatrix} P_{\mathrm{z},0} & 0 & 0 \\ 0 & P_{\mathrm{V_1},0} & 0 \\ 0 & 0 & P_{\mathrm{T},0} \end{bmatrix}, \tag{4.24}$$

and the overall system $\mathbf{P}_0$ is formed by the block diagonal combination,

$$\mathbf{P}_0 = \begin{bmatrix} \mathbf{P}_{\mathrm{Batt},0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{\mathrm{GP},0} \end{bmatrix}. \tag{4.25}$$

Propagation in time

The mean and covariance of the joint system are propagated in discrete time by the extended Kalman filter [66] with additive noise,

$$\begin{aligned} \mathbf{x}_{\mathrm{t}}^{-} &= g(\mathbf{x}_{\mathrm{t}-1}, I_{\mathrm{t}-1}, T_{\mathrm{t}-1}) \\ \mathbf{P}_{\mathrm{t}}^{-} &= \mathbf{G}_{\mathrm{t}-1}\mathbf{P}_{\mathrm{t}-1}^{+}\mathbf{G}_{\mathrm{t}-1}^{\mathrm{T}} + \mathbf{Q}_{\mathrm{t}-1} + \lambda_{\mathrm{G,t}-1} \end{aligned} \tag{4.26}$$

where g describes both the system and parameter evolution dynamics, $\mathbf{G}_{\mathrm{t}}$ is the local Jacobian matrix of g at $\mathbf{x}_{\mathrm{t}}$, $\mathbf{Q}_{\mathrm{t}}$ the joint discrete time process covariance and $\lambda_{\mathrm{G,t}}$ an additional variance term arising from the posterior predictive variance of the GP (Appendix C.1, eqns. 3.43). In general, it was found that using (linearised) discrete time equations to propagate the system covariance proved to be more numerically stable than using the continuous time counterparts (3.20). In particular, while computing continuous time Jacobian matrices would have been simpler, using either forward Euler or 4^{th} order Runge-Kutta methods to propagate the covariance often resulted in a non-positive definite prior covariance $\mathbf{P}_{\mathrm{t}}^{-}$. This numerical instability would have made gradient-based hyperparameter optimisation very difficult.

Integrating over z_{t}

In order to evaluate the battery dynamics in (4.26), the values for α, β and R_0 must be calculated. Ordinarily, their predictive means and variances are given by (3.5). However, since the state of charge, z_{t}, is itself a Gaussian random variable, the predictive distributions must be marginalised (averaged) over the distribution z_{t} by performing the integral

$$p(f(z_{\mathrm{t}})) = \int p(f(z_{\mathrm{t}})|z_{\mathrm{t}})p(z_{\mathrm{t}})\,\mathrm{d}z_{\mathrm{t}} \tag{4.27}$$

at each time step, where $f(z_{\mathrm{t}})$ represents the predictive distribution of α, β and R_0. In other words, the average GP prediction across all possible values of the probability distribution of the

current z_t is calculated. For an arbitrary kernel function k, this integral is usually intractable and can be approximated using e.g. Taylor expansions [183]. However, for the SE kernel, (4.27) has an analytical solution given by Quiñonero-Candela et al. [184], where the mean and variance of $p(f(z_t))$ are (for scalar input z)

$$\begin{aligned}\mathbb{E}[f(z)] &= \mathbf{l}_{f,z}^T \delta \\ \mathbb{V}[f(z)] &= \sigma_{f,GP}^2(\mu_z) + \mathrm{Tr}\left((\mathbf{K}^{-1} - \mathbf{k}_{f,z,\mathbf{u}}\mathbf{k}_{f,z,\mathbf{u}}^T)\mathbf{L}\right) \\ &\quad + \mathrm{Tr}\left(\delta\delta^T(\mathbf{L}_{f,z} - \mathbf{l}_{f,z}\mathbf{l}_{f,z}^T)\right)\end{aligned} \tag{4.28}$$

where Tr denotes the matrix trace and $\mathbf{K} = \mathbf{K}_{\mathbf{f,uu}} + \mathrm{diag}(\mathbf{P}_{f,t})$ for each of the GPs that are functions of z_t. The parameter δ is given by $\mathbf{K}^{-1}\mathbf{x}_{f,t}$, where $\mathbf{x}_{f,t}$ is the state vector for the GP at time t, $\sigma_{GP}^2(\mu_z)$ is the standard GP predictive variance evaluated at the mean of z_t and $\mathbf{l}_z$, $\mathbf{L}_z$ are given by

$$\begin{aligned}\mathbf{l}_{f,z} &= \frac{\sigma_{f,\mathbf{x}}^2}{\sqrt{\gamma_{f,z}\sigma_z^2 + 1}} \exp\left(-\frac{1}{2(\gamma_{f,z}^{-1} + \sigma_z^2)}(\mu_z - \mathbf{u}_f)^2\right) \\ \mathbf{L}_{f,z} &= \frac{\mathbf{k}_{f,z,u}\mathbf{k}_{f,z,u}^T}{\sqrt{2\gamma_{f,z}\sigma_z^2 + 1}} \exp\left(\frac{2\gamma_{f,z}^2}{2\gamma_{f,z} + \sigma_z^2}(\mu_z - \overline{\mathbf{U}}_f)^2\right).\end{aligned} \tag{4.29}$$

Here, the vector $\mathbf{u}_f$ represents the coordinates of the discretisation points of z and the element of matrix $\overline{\mathbf{U}}_{f,ij} = \frac{1}{2}(\mathbf{u}_{f,i} + \mathbf{u}_{f,j})$. The variable $\gamma_{f,z}$ is the inverse square of the length scale and $\sigma_{f,z}^2$ the magnitude of GP f over the SOC input z, σ_z^2 is the variance of z. The vector $\mathbf{k}_{f,z,\mathbf{u}}$ is the kernel function k_{SE} evaluated at μ_z, i.e. the same as $\mathbf{l}_{f,z}$ where $\sigma_z^2 = 0$.

In summary, uncertainty in SOC, z_t, affects both the mean and the variance of the GP output, changing the weighting of the linear combination of points used to make a prediction and adding to the variance. As $\sigma_z^2 \to 0$, from equations (4.28, 4.29) the standard GP predictive equations are recovered (3.5).

State and covariance propagation

Given estimates of $\alpha(z_t)$, $\beta(z_t)$ and $R_0(z_t, I_t)$, discrete time propagation of battery dynamics is approximated using a zero-order hold on the applied current and linearising g with respect to battery states at time t. The propagation of the GP states $\mathbf{x}_{\mathrm{GP,t}}$ is independent of battery states and linear, where the state transition is given by

$$\mathbf{x}_{\mathrm{GP,t}} = \exp\left(\begin{bmatrix} \mathbf{F}_{\mathrm{WV}} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n_z} \otimes \mathbf{F}_{\mathrm{WV}} & \dots & \dots & \mathbf{0} \\ \vdots & \vdots & \mathbf{I}_{n_z} \otimes \mathbf{F}_{\mathrm{WV}} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \mathbf{I}_{n_{zI}} \otimes \mathbf{F}_{\mathrm{WV}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{\mathrm{E}} \end{bmatrix} \Delta\zeta_{t-1}\right) \mathbf{x}_{\mathrm{GP,t-1}}, \tag{4.30}$$

where $\mathbf{I}_n$ is the identity matrix size n, equal to the number of discrete 'spatial' points propagated through time for each GP and

$$\mathbf{F}_{\mathrm{WV}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \ \mathbf{F}_{\mathrm{E}} = -\gamma_{\zeta,\mathrm{r}}\mathbf{I}_4. \tag{4.31}$$

The variable $\Delta\zeta_t$ is the time step size in the GP 'degradation' timescale, which is larger than the time step of the system dynamics. Within each discharge cycle, this is assumed constant, therefore requiring only a single evaluation of the matrix exponential (4.30) for each discharge cycle. As $\mathbf{F}_{\mathrm{WV}}$ is nilpotent and the matrix inside the exponential in (4.30) is block diagonal, the exponentiation was done explicitly using a Taylor expansion up to the first order (as all higher order terms vanish). This is computationally efficient and was in fact necessary, since the built-in matrix exponential method in `julia` is (at the time of writing) not compatible with automatic differentiation, which was used for hyperparameter gradient calculations. To evaluate GPs α, β, and R_0 over battery lifetime, specific discharge cycles are selected from raw data at an appropriate rate (in this case, one in thirty) to improve computational efficiency—$\Delta\zeta$ is

therefore the cumulative charge throughput over weeks rather than seconds. The discrete time process noise covariance matrix $\mathbf{Q}_\mathrm{t}$ is block diagonal, where the values for the battery states are fixed and the values for the GP are determined by the kernel function hyperparameters. More specifically, the WV kernel and exponential kernels have discrete time process variances given as a function of the step $\Delta\zeta$,

$$\mathbf{Q}_\mathrm{WV}(\Delta\zeta) = \sigma^2_{\zeta,\mathrm{s}} \begin{bmatrix} \frac{1}{3}\Delta\zeta^3 & \frac{1}{2}\Delta\zeta^2 \\ \frac{1}{2}\Delta\zeta^2 & \Delta\zeta \end{bmatrix}, \; Q_\mathrm{E}(\Delta\zeta) = \sigma^2_{\zeta,\mathrm{r}}(1-\exp(-2\gamma_\mathrm{r}\Delta\zeta)), \tag{4.32}$$

which again combine with the battery state covariance giving

$$\mathbf{Q}_\mathrm{t}(\Delta\zeta) = \begin{bmatrix} \mathbf{Q}_\mathrm{Batt} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_\mathrm{GP}(\Delta\zeta) \end{bmatrix}, \text{ where } \mathbf{Q}_\mathrm{Batt} = \begin{bmatrix} q_\mathrm{z} & 0 & 0 \\ 0 & q_{\mathrm{V}_1} & 0 \\ 0 & 0 & q_{\mathrm{T_c}} \end{bmatrix}, \tag{4.33}$$

and the block diagonal GP process covariance is

$$\mathbf{Q}_\mathrm{GP}(\Delta\zeta) = \begin{bmatrix} \mathbf{Q}_{\mathrm{Q}^{-1},\mathrm{WV}}(\Delta\zeta) & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{\mathrm{k},\alpha,\mathrm{WV}}(\Delta\zeta) & \dots & \dots & \mathbf{0} \\ \vdots & \vdots & \mathbf{Q}_{\mathrm{k},\beta,\mathrm{WV}}(\Delta\zeta) & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \mathbf{Q}_{\mathrm{k},\mathrm{R}_0,\mathrm{WV}}(\Delta\zeta) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_\mathrm{E}(\Delta\zeta) \end{bmatrix}, \tag{4.34}$$

where $\mathbf{Q}_{\mathrm{f,WV}}(\Delta,\zeta)$, for $\mathrm{f} \in \{\alpha,\ \beta,\ R_0\}$ is resolved over the spatial input, that is,

$$\mathbf{Q}_{\mathrm{k,f,WW}}(\Delta\zeta) = k_{\mathrm{f,x}}(\mathbf{U}_\mathrm{f}, \mathbf{U}'_\mathrm{f}) \otimes \mathbf{Q}_{\mathrm{f,WV}}(\Delta\zeta), \tag{4.35}$$

where $\mathbf{U}_\mathrm{f}$ are the values of the coordinates of the discretisation points for each GP (4.10). The exponential process covariance $\mathbf{Q}_\mathrm{E}(\Delta\zeta) \in \mathbb{R}^{4\times4}$ is diagonal with elements given by (4.32).

4.3.3 Observation model

Following the propagation of the system by relations (4.26), the predicted voltage and temperature are given by (4.3), which involves re-evaluating $R_0(z_t, I_t)$ with the current estimate of the state vector $\mathbf{x}_{R_0,t}$. The predictive equation (4.3) is nonlinear in SOC z_t, due to the open-circuit potential V_0 and $R_0(z_t, I_t)$. Using the EKF, (4.3) is locally linearised to give the observation Jacobian $\mathbf{H}_t$. The dependency of R_0 on the SOC (z_t), which itself is a Gaussian random variable, can be calculated using the same methods as previously discussed (4.27). In addition, the uncertainty in R_0 from GP extrapolation to the current operating point (as per previous chapter) $\lambda_{H,t}$ is also incorporated into the predictive distribution for the output, which is

$$\begin{bmatrix} V_t \\ T_t \end{bmatrix} \sim \mathcal{N}(h(\mathbf{x}_t^-, I_t), \mathbf{S}_t), \tag{4.36}$$

where the covariance $\mathbf{S}_t$ of the output is given by

$$\mathbf{S}_t = \mathbf{H}_t \mathbf{P}_t^- \mathbf{H}_t^T + \mathbf{R} + \lambda_{H,t}, \tag{4.37}$$

where h is given by (4.3), $\mathbf{R}$ is the measurement noise covariance matrix that is estimated and the calculation $\lambda_{H,t}$ may be found in the appendix C.1.

Summarising all of the above, the full system recursion is given in Algorithm 4.1. This is in effect the same principle as used in Chapter 3, but with an added observer for battery states.

4.3.4 Hyperparameter optimisation and smoothed posterior

The model dynamics and outputs depend on the hyperparameters of the kernel functions. In addition, the noise parameters $\sigma_{n,V}$ and $\sigma_{n,T}$ for the output voltage and temperature need to be estimated. Similarly to the lead-acid case as described in Chapter 3, parameter estimates are found by using the Kalman filter which recursively updates the NLML in conjunction with a

Algorithm 4.1 Extended Kalman filter recursion with NLML calculation. Battery states are initialised at the beginning of every cycle and the GP is propagated over the lifetime of the battery. The NLML is calculated for all available data recursively.

Initialisation at $\zeta = \zeta_0$
  $\mathbf{x}_{\text{GP}}^{+} = \mathbf{x}_{\text{GP},0}^{+}$, $\mathbf{P}_{\text{GP}}^{+} = \mathbf{P}_{\text{GP},0}^{+}$
  $\phi_{\text{t}} = 0$
for $\mathcal{D}_{\text{s}} \in$ segments **do** ▷ loop over continuous discharge segments
    Initialisation at start of segment
    $\mathbf{x}_{\text{Batt}}^{+} = \mathbf{x}_{\text{Batt},0}$, $\mathbf{P}_{\text{Batt}}^{+} = \mathbf{P}_{\text{Batt},0}$ ▷ Battery state initialisation (4.23), (4.24)
    $\mathbf{x}_{\text{GP}}^{+} = \exp(\mathbf{F}\Delta\zeta)\mathbf{x}_{\text{GP}}^{+}$ ▷ $\Delta\zeta$ =distance between segments
    $\mathbf{P}_{\text{GP}}^{+} = \exp(\mathbf{F}\Delta\zeta)\mathbf{P}_{\text{GP}}^{+}\exp(\mathbf{F}\Delta\zeta)^{\text{T}} + \mathbf{Q}_{\text{GP}}(\Delta\zeta)$
    $\mathbf{x}^{+} = \begin{bmatrix}\mathbf{x}_{\text{Batt}}^{+} & \mathbf{x}_{\text{GP}}^{+}\end{bmatrix}^{\text{T}}$
    $\mathbf{P}^{+} = \oplus\left(\mathbf{P}_{\text{Batt}}^{+}, \mathbf{P}_{\text{GP}}^{+}\right)$
    for I_{t},V_{t},$\text{T}_{\text{t}} \in \mathcal{D}_{\text{s}}$ **do**
      Propagation
      $\mathbf{x}_{\text{t}}^{-} = g(\mathbf{x}_{\text{t}-1}, I_{\text{t}-1}, T_{\text{t}-1})$ ▷ Joint GP & battery dynamics
      $\mathbf{P}_{\text{t}}^{-} = \mathbf{G}_{\text{t}-1}\mathbf{P}_{\text{t}-1}^{+}\mathbf{G}_{\text{t}-1}^{\text{T}} + \mathbf{Q}_{\text{t}-1} + \lambda_{\text{G,t}-1}$
      Observation and update
      $\mathbf{e}_{\text{t}} = [V_{\text{t}}\ T_{\text{t}}]^{\text{T}} - h(\mathbf{x}_{\text{t}}, I_{\text{t}})$
      $\mathbf{S}_{\text{t}} = \mathbf{H}_{\text{t}}\mathbf{P}_{\text{t}}^{-}\mathbf{H}_{\text{t}}^{\text{T}} + \mathbf{R} + \lambda_{\text{H,t}}$
      $\mathbf{K}_{\text{t}} = \mathbf{P}_{\text{t}}^{-}\mathbf{H}_{\text{t}}^{\text{T}}\mathbf{S}_{\text{t}}^{-1}$
      $\mathbf{x}_{\text{t}}^{+} = \mathbf{x}_{\text{t}}^{-} + \mathbf{K}_{\text{t}}\mathbf{e}_{\text{t}}$
      $\mathbf{P}_{\text{t}}^{+} = (\mathbf{I} - \mathbf{K}_{\text{t}}\mathbf{H}_{\text{t}})^{\text{T}}\mathbf{P}_{\text{t}}^{-}(\mathbf{I} - \mathbf{K}_{\text{t}}\mathbf{H}_{\text{t}}) + \mathbf{K}_{\text{t}}\mathbf{R}\mathbf{K}_{\text{t}}^{\text{T}}$
      $\phi_{\text{t}} = \phi_{\text{t}} + \frac{1}{2}\mathbf{e}_{\text{t}}^{\text{T}}\mathbf{S}_{\text{t}}^{-1}\mathbf{e}_{\text{t}} + \frac{1}{2}\log|2\pi\mathbf{S}_{\text{t}}|$
    end for
end for

gradient-based optimiser. In this case, no prior is imposed on the hyperparameters—maximum likelihood estimates are used instead. Also, instead of finding analytical solutions to the NLML gradients $\partial\phi/\partial\theta$, they are calculated using automatic differentiation.

Once hyperparameter MLEs are calculated, the final step is to calculate a posterior distribution of latent states $\mathbf{x}_{\text{f,t}}$ consistent with batch mode GP regression. As in the previous chapter, this is done by calculating the smoothing distribution using the RTS smoother (Table 3.2). However, in contrast to the calculation of the forward distribution, which uses all I, V, T data in the discharge cycles, the smoothing distribution only needs to be calculated for the

Parameter	Description	Value	Units	GPR RMSE (%)
$\alpha(z)$	Inverse RC time constant	$0.015 - 0.09\,(0.5-z)^3$	s^{-1}	2.6
$\beta(z)$	Inverse RC capacitance	$0.002\left(1-(z-0.5)^2\right)$	F^{-1}	0.77
$R_0(z,I)$	Series resistor	$0.05\sinh^{-1}\left(\lvert I\rvert\right)/\lvert I\rvert + 0.04\,(z-1)^2$	Ω	0.98
Q^{-1}	Inverse cell capacity	1.2	Ah^{-1}	0.16
R_c	Thermal resistance	5.5	$\mathrm{K\,W}^{-1}$	Given
C_c	Heat capacity	15.7	$\mathrm{J\,K}^{-1}$	Given
V_0	Open-circuit potential	$3.64 + 0.55x - 0.72x^2 + 0.75x^3$	V	Given
$\sigma_{\mathrm{n,V}}$	Voltage meas. noise st. dev.	0.005	V	0.4†
$\sigma_{\mathrm{n,T}}$	Temp. meas. noise st. dev.	0.1	K	2.8†

Table 4.1 Simulation parameters and errors in estimates. The error in GPR estimates after a single discharge cycle is <3% for each circuit parameter (calculated as the average of the function error over the input domains). † these are a subset of the GP hyperparameters.

linear GP subsystem in the ζ timescale. The number of timesteps in this timescale equals the number of discharge cycles in the downsampled dataset.

4.4 Simulation with known ground-truth

To establish the identifiability of circuit parameters as functions of operating conditions using the joint state/parameter method, simulated data with known ground-truths for α, β and R_0 was used. To this end, the voltage and temperature responses from model (4.2) using the current profile from the US06 drive cycle [185] were used. The ground-truth functions α, β and R_0 were chosen arbitrarily and are shown in Table 4.1 alongside the other simulation parameters. To simulate measurement noise, zero-mean Gaussian noise was added to voltage and temperature measurements with standard deviations of 5 mV and 0.1 K respectively. The current profile, voltage and temperature responses and the internal states of the model (i.e. z and V_1) are shown in Fig. 4.3.

The joint GP/battery state estimator was applied to the simulated current, voltage and temperature data. Six evenly spaced points over the range of state of charge z in the simulation were chosen to describe the GPs α and β and similarly an evenly spaced 4×15 grid over

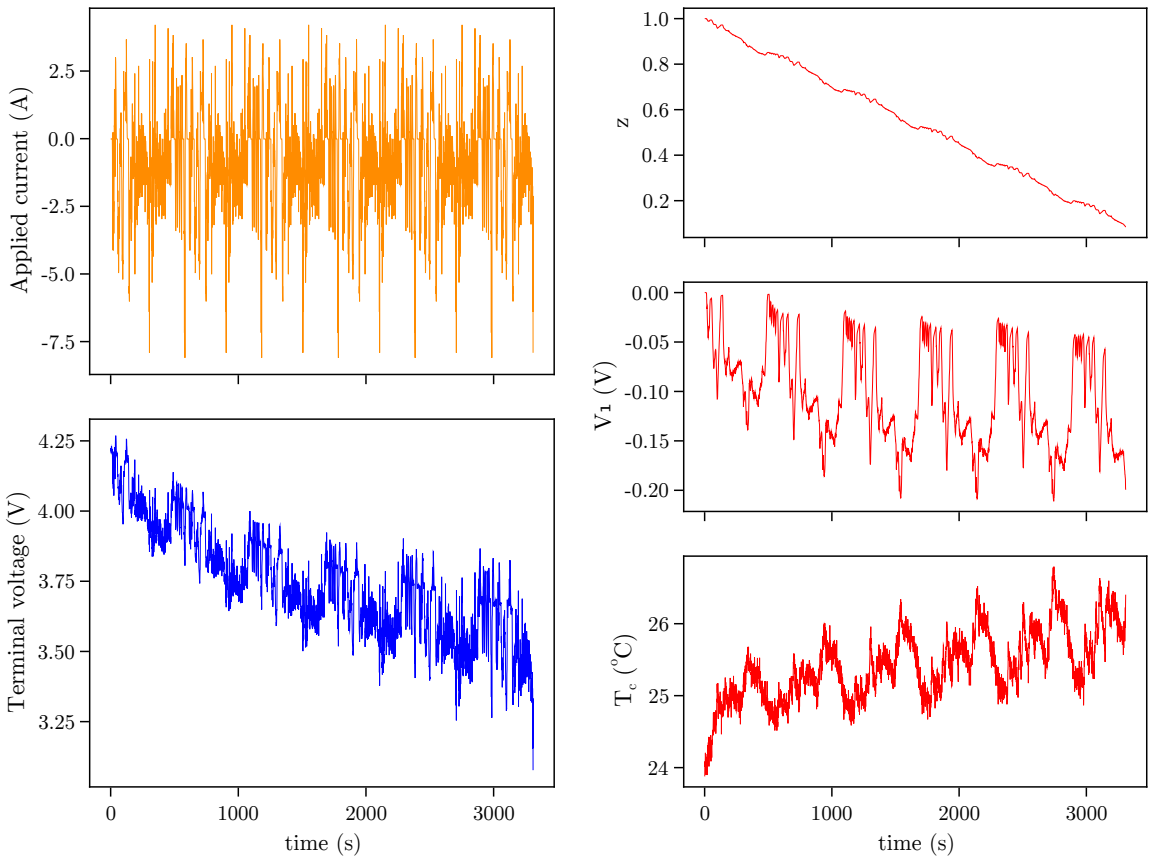


Fig. 4.3 Simulated input/output data for determining circuit parameter identifiability. The current profile consisted of 4 $\times$ repeats of the US06 drive cycle. Battery states and outputs are calculated simulating the circuit (4.2) with known ground-truth functions for α, β and R_0 (Table 4.1).

observed values of z and I was chosen for R_0. The constants in the transformation (4.5) for Q^{-1}, α, β and R_0 were set at $1.09\,\mathrm{Ah}^{-1}$, $0.01\,\mathrm{s}^{-1}$, $0.0007\,\mathrm{F}^{-1}$ and $0.04\,\Omega$ respectively. The GP hyperparameters, consisting of length scales for α, β and R_0, and the magnitudes for all four GPs as well as the noise parameters $\sigma_{\mathrm{n,V}}, \sigma_{\mathrm{n,T}}$ were estimated using the box-constrained BFGS algorithm implemented in the `Optim.jl` package in Julia [141], using forward-mode automatic differentiation to calculate the NLML gradients [142]. Positivity in parameters was imposed by running the optimiser in log-space and box constraints in the optimisation routine were imposed to guarantee numerical stability and differentiability of the NLML (Appendix C.2). The main purpose of the simulation was to establish the empirical identifiability of each

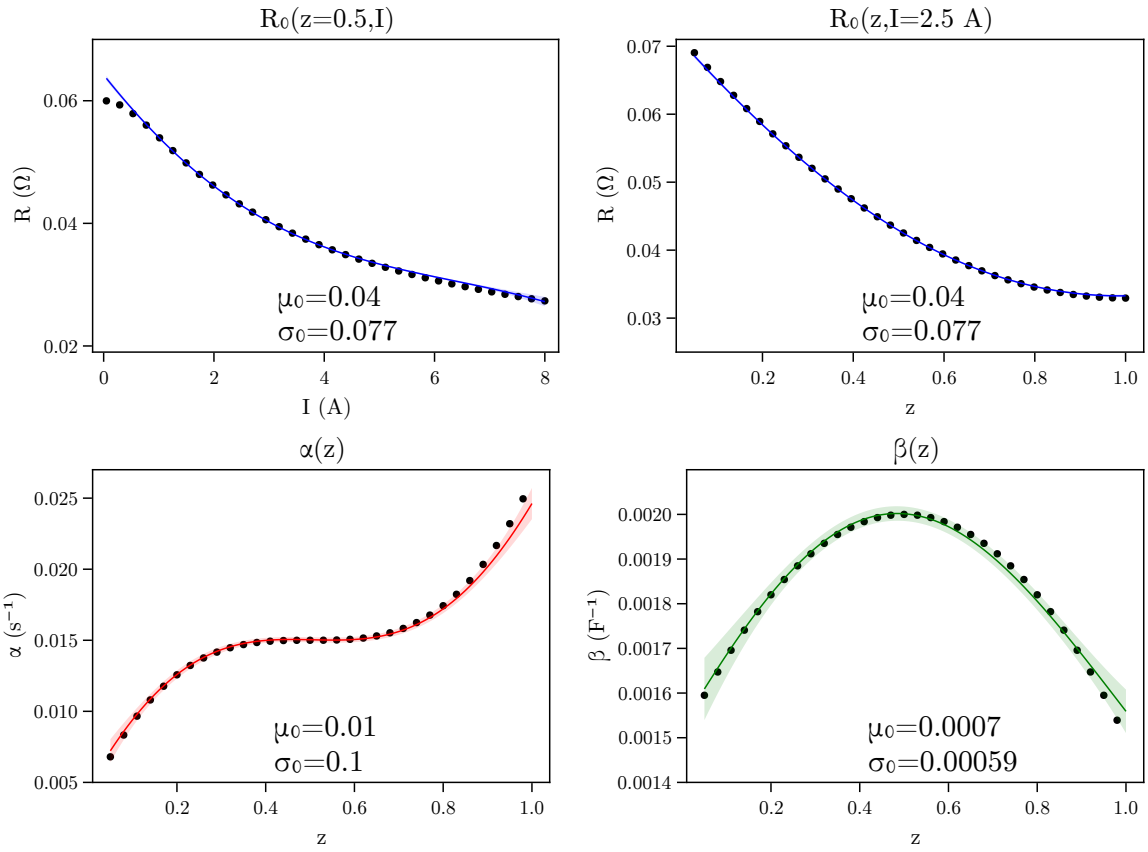


Fig. 4.4 Ground-truth functions for $\alpha(z)$, $\beta(z)$, $R_0(z,I)$, shown in black dots, together with their GP estimates showing the mean $\pm 2\sigma$, using 1 discharge cycle of input/output data. μ_0 and σ_0 in each show the prior mean and variance for each function, where the former is the constant c_f and the latter is an estimated hyperparameter of the GP.

of the functions over operating conditions, for which only a single discharge cycle needed to be considered.

The results of the estimation are shown in Fig. 4.4 and Table 4.1. In Fig. 4.4, projections through each function are shown. From these, it is clear that the GPR estimator accurately retrieves the ground-truth functions for the circuit parameters. However, a small loss of accuracy occurs for R_0 at low current I because the ground-truth function has an inflection point here. The hyperparameter MLE is such that the GP assumes a long length scale for $R_0(z,I)$ over both input dimensions, giving a predictive posterior that extrapolates to a higher value as $I \to 0$. This is partially due to the numerical ill-conditioning at very low applied current—i.e. a small

error in voltage and/or current causes a large change in estimated R_0. The GP in this case relies on extrapolating from regions of higher current where the function is more identifiable and smoother, with no turning points.

4.5 Results: Experimental data

4.5.1 Experimental data

The simulation work of the previous section shows that GP representations of known ground-truth functions for ECM parameters are readily retrieved from input-output data. Next, the GP estimator was implemented on real cell data using a cycling dataset generated during an experimental campaign by Jöst et al. [112] based on high-energy 18650 Li-ion cells (Samsung SDI INR18650-35E, NCA cathode / Gr+Si anode, nominal capacity 3450 mAh).

The cycle ageing was conducted at a temperature of 25 °C. The charging protocol consisted of charging at 0.3C to an upper voltage limit of 4.05 V, which was held until the current dropped below 0.02C. Discharging was conducted using a pre-recorded drive cycle profile with an average rate of 0.4C, with sections representing various types of urban and extra-urban driving. For the duration of the campaign, cells were actively air-cooled in a Binder MK240 thermal chamber. The battery cycler was a Digatron MCFT 20-5-60 ME with an accuracy of $\pm$ 20 mA/4 mV after calibration.

A checkup test procedure was conducted every 30 cycles and consisted of 0.3C discharge capacity test after CCCV full charge, followed by charge and discharge pulse tests at three SOC levels (80%, 50%, 20%) using three different pulse sizes (1.7 A, 3.4 A, 6.8 A). The pseudo open-circuit voltage was determined at beginning of life with a full discharge at 0.02C. To parameterise the thermal model of (4.2), the heat capacity of the cell (43.5 $\mathrm{J\,K^{-1}}$) was taken from literature [186]. The thermal resistance R_c was then determined from the thermal

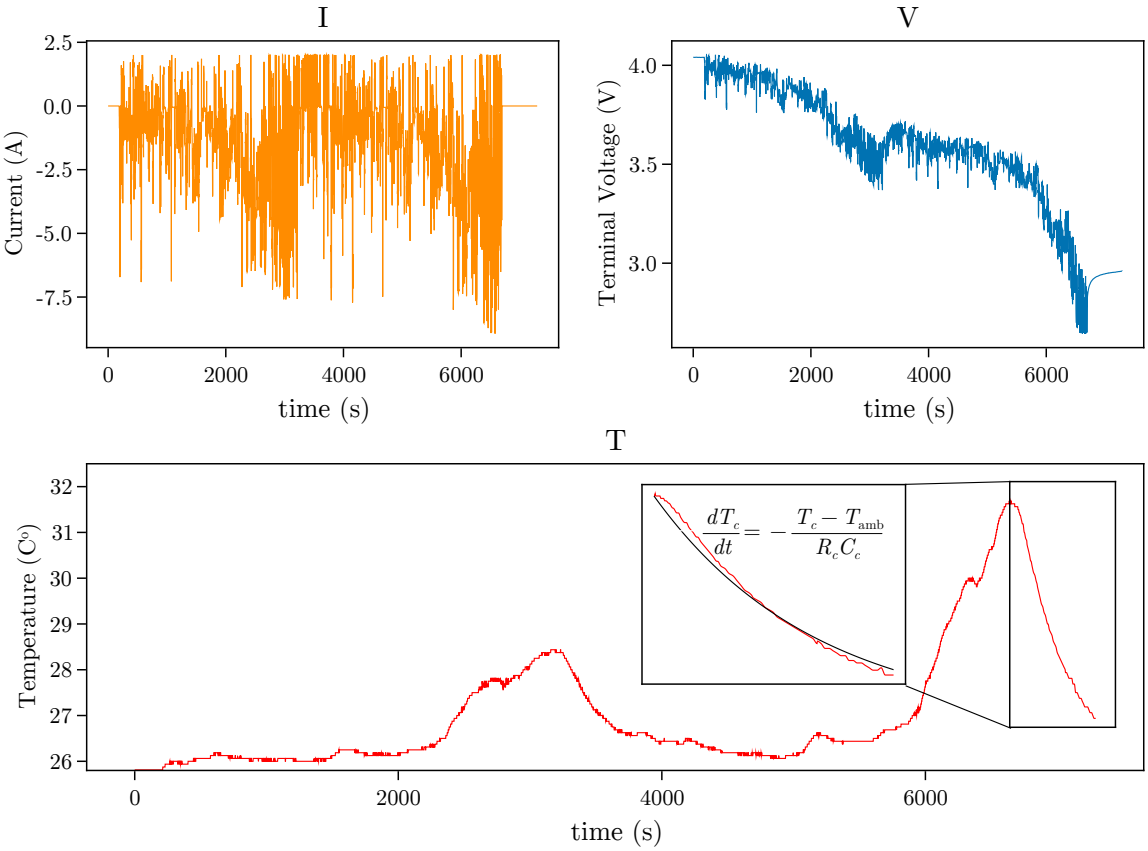


Fig. 4.5 Current, voltage and temperature profiles for the drive cycle used in experimental set-up. To parameterise the thermal model, a value from literature was used for C_c and a least squares fit was used to find an estimate for R_c from the thermal relaxation at the end of the drive cycle at beginning of life.

relaxation data following the first drive cycle at beginning of life using a least-squares fit, illustrated in Fig. 4.5.

4.5.2 Fitting functional parameters

The recursive framework outlined in section 4.3 was used to retrieve the electrical parameters in the model (4.2) for two example cells (cell numbers 009 and 015) in the dataset.

Data downsampling and validation

The dataset consisted of 30 repeated charge/discharge cycles between each checkup sequence. From each set of 30, a single discharge cycle was used for GP estimation of parameters. The discharge cycle chosen was the last one in the repeated set, immediately preceding the checkup sequence. This resulted in 27 and 28 discharge cycles for cells 009 and 015 respectively. Within each cycle, the data were interpolated to a frequency of 1 Hz using the piece-wise cubic hermite method [174].

For validation, the 0.3C discharge capacity test and internal resistance data calculated as the average (over charge/discharge pulses) $\Delta V/\Delta I$ from the first second of the relaxation after each 10 s pulse were used. The dataset was split into two sections for each cell, whereby the last 8 sets of cycles were left out to be used in an out-of-sample setting to assess the ability of the GP to forecast SOH evolution, giving an in-sample set of 19 and 20 discharge cycles for the two cells, respectively.

Hyperparameter estimation

The four GPs, Q^{-1}, α, β and R_0 each have a multiplicative kernel function of the type (4.8), where in the case of Q^{-1}, the function $k_{\mathrm{x}}(\mathbf{x},\mathbf{x}')$ is a constant as it has no dependency on z or I. In total, the hyperparameter vector θ_{h} controlling the properties of the 4 GPs over z, I and ζ contains 16 elements, where

$$\theta_{\mathrm{h}} = [\theta_{\mathbf{x}}\ \theta_{\zeta,\mathrm{s}}\ \theta_{\zeta,\mathrm{r}}\ \sigma_{\mathrm{n,V}}\ \sigma_{\mathrm{n,T}}]\ , \tag{4.38}$$

and the subcomponents are given by

$$\begin{aligned}\theta_{\mathbf{x}} &= [\sigma_{\mathrm{Q}^{-1}}\ \sigma_{\alpha,\mathbf{x}}\ \sigma_{\beta,\mathbf{x}}\ \sigma_{\mathrm{R}_0,\mathbf{x}}\ \gamma_{\alpha,\mathrm{z}}\ \gamma_{\beta,\mathrm{z}}\ \gamma_{\mathrm{R}_0,\mathrm{z}}\ \gamma_{\mathrm{R}_0,\mathrm{I}}]\ ,\\ \theta_{\zeta,\mathrm{s}} &= [\sigma_{\mathrm{Q}^{-1},\zeta,\mathrm{s}}\ \sigma_{\alpha,\zeta,\mathrm{s}}\ \sigma_{\beta,\zeta,\mathrm{s}}\ \sigma_{\mathrm{R}_0,\zeta,\mathrm{s}}]\ ,\\ \theta_{\zeta,\mathrm{r}} &= [\sigma_{\zeta,\mathrm{r}}\ \gamma_{\zeta,\mathrm{r}}]\ ,\end{aligned} \tag{4.39}$$

where $\theta_{\mathbf{x}}$, $\theta_{\zeta,\mathrm{s}}$, $\theta_{\zeta,\mathrm{r}}$ are the hyperparamaters the SE, WV and exponential kernels respectively, with measurement noise standard deviations $\sigma_{\mathrm{n,V}}$, $\sigma_{\mathrm{n,T}}$ for the terminal voltage and cell temperature respectively. It was assumed that both battery cells share the same set of hyperparameters and all estimation was done using the summed NLMLs from the two cells in each scenario. While substantial effort was put into computational efficiency of the forward calculation of the EKF estimator, it was found that the computational cost remained prohibitive to attempt to optimise the hyperparameter vector using the entire length of the dataset. Hyperparameter estimation scales nonlinearly with dimensionality of the vector and several steps were taken to reduce complexity to retrieve maximum likelihood estimates for the θ_{h}, as follows.

First, it was assumed that the length scale and magnitude parameters for the GPs describing α and β are shared, as they both relate to the behaviour of the RC pair over SOC, so that $\sigma_\alpha = \sigma_\beta$ and $\gamma_\alpha = \gamma_\beta$. Secondly, the WV magnitude parameters $\theta_{\zeta,\mathrm{s}}$ were grouped into two, so that $\theta_{\zeta,\mathrm{s}} = [\sigma_{0,\zeta,\mathrm{s}}\ \sigma_{1,\zeta,\mathrm{s}}]$, where $\sigma_{0,\zeta}$ was the WV kernel magnitude for Q^{-1} and $\sigma_{1,\zeta}$ the WV magnitude for α, β and R_0. The rationale for this grouping arises from assuming that the degradation process, in terms of *relative magnitude* (i.e. with respect to the value at beginning of life), is similar for α, β and R_0, which are dependent on operating conditions. The three steps above reduce the hyperparameter vector, so that $\theta_{\mathrm{h}} \in \mathbb{R}^{12}$. The process noise covariance

matrix for battery states was fixed so that

$$\mathbf{Q}_{\text{Batt}} = \begin{bmatrix} 10^{-12} & 0 & 0 \\ 0 & 10^{-6} & 0 \\ 0 & 0 & 10^{-4} \end{bmatrix}. \tag{4.40}$$

Finally, to use all available data (288,639 rows of current, voltage and temperature data in the in-sample set) to retrieve ML estimates of $\theta_{\text{h}} \in \mathbb{R}^{12}$ would still require substantial computational effort due to the high dimensionality of the optimisation problem. Therefore, to reduce computational effort, the hyperparameter estimation problem was split into two. As shown in the simulation case (Section 4.4), the dependency of each of the functions α, β and R_0 on battery states and operating conditions may be inferred from a single cycle. While this dependency may vary over the lifetime of the battery, it was assumed that the hyperparameters controlling the GPs over z and I remained constant. With this assumption, the first available cycle from the beginning of life was used to estimate the subset of hyperparameters $\theta_{\text{h},0} = [\theta_{\mathbf{x}}\ \sigma_{\text{n,V}}\ \sigma_{\text{n,T}}]$, because the NLML is independent of $\theta_{\zeta,\text{s}}$ and $\theta_{\zeta,\text{r}}$ over a single cycle. In addition to reducing the dimensionality of the estimation problem to $\theta_{\text{h}} \in \mathbb{R}^{8}$, this reduces the number of data rows required from 288,639 to 14,597. A multi-start process was used in the optimiser. First, the NLML using 1000 randomly chosen points for $\theta_{\text{h},0}$ was estimated. From these, 25 of the lowest NLML points were chosen as starting points and the same gradient-based optimization routine as in the simulation case was applied (box-constrained BFGS), where the final $\theta_{\text{h},0}$ chosen was that with the lowest overall NLML value. Following this process, $[\theta_{\zeta,\text{s}}\ \theta_{\zeta,\text{r}}]$ were estimated together with the MLE for $\theta_{\mathbf{h},\mathbf{0}}$ using the full in-sample dataset of 288,639 rows. In this case, the optimisation problem is 4-dimensional, so the number of iterations required is lower and only a single starting point, found by grid search, needs to be used. Again, the box-constrained BFGS algorithm was used.

$\sigma_{Q^{-1}}$	$\sigma_{\alpha,\beta,\mathbf{x}}$	$\sigma_{R_0,\mathbf{x}}$	$\gamma_{R_0,z}$	$\gamma_{R_0,I}$	$\gamma_{\alpha,\beta,\mathbf{x}}$	$\sigma_{n,V}$	$\sigma_{n,T}$	$\sigma_{0,\zeta,s}$	$\sigma_{1,\zeta,s}$	$\sigma_{\zeta,r}$	$\gamma_{\zeta,r}$
0.0023	9.0	0.59	14.0	4.5e-5	36.0	0.0045	0.013	0.062	0.37	0.05	480.0

Table 4.2 MAP estimates for hyperparameter vector θ_h. N.B. $\gamma = 1/2l^2$ in each case.

Estimation and validation of GP posterior

Given MLEs for hyperparameters θ_h, the smoothed posterior was found using the RTS smoother over lifetime in the in-sample set. With the smoothed estimates of $\mathbf{x}_{GP,t}$ and $\mathbf{P}_{GP,t}$, the GP estimates were extrapolated to the correct point in time over lifetime (from the cycling point to the checkup test), for validation purposes. At these points in the time axis, the GP predictive equations (3.5) were used to retrieve estimates of the functions over SOC z and I.

4.5.3 Results and discussion

Circuit parameter dependence on inputs

Fig. 4.6 shows the GP estimates for capacity, R_0, α and β over lifetime for cell #015 in the dataset, calculated as functions of full equivalent cycle count (FEC), defined as the total charge throughput divided by the nominal capacity. As the hyperparameters for the two fitted cells were calculated using a combined likelihood (Section 4.5.2), the results are very similar for both (equivalent plots for cell #009 may be found in Appendix C.3). The plots each show the long-term, smooth component of the GP (see eqn. 4.8), which provides the stable parameter estimates over lifetime. All equivalent plots including the noise kernel are in Appendix C.3. As shown in Table 4.2, the parameter $\gamma_{\zeta,r}$ implies a very short length scale for the noise process, implying very little auto-correlation in it. Other (inverse) length scale hyperparameters reflect the variability of each function over its inputs—R_0 has strong dependency on z (i.e. SOC) and no dependency on applied current. The RC pair has strong dependency on z.

These dependencies are shown in the right three plots of Fig. 4.6. Notably R_0 remains relatively constant down to approximately 10-15% SOC, where it starts increasing substantially.

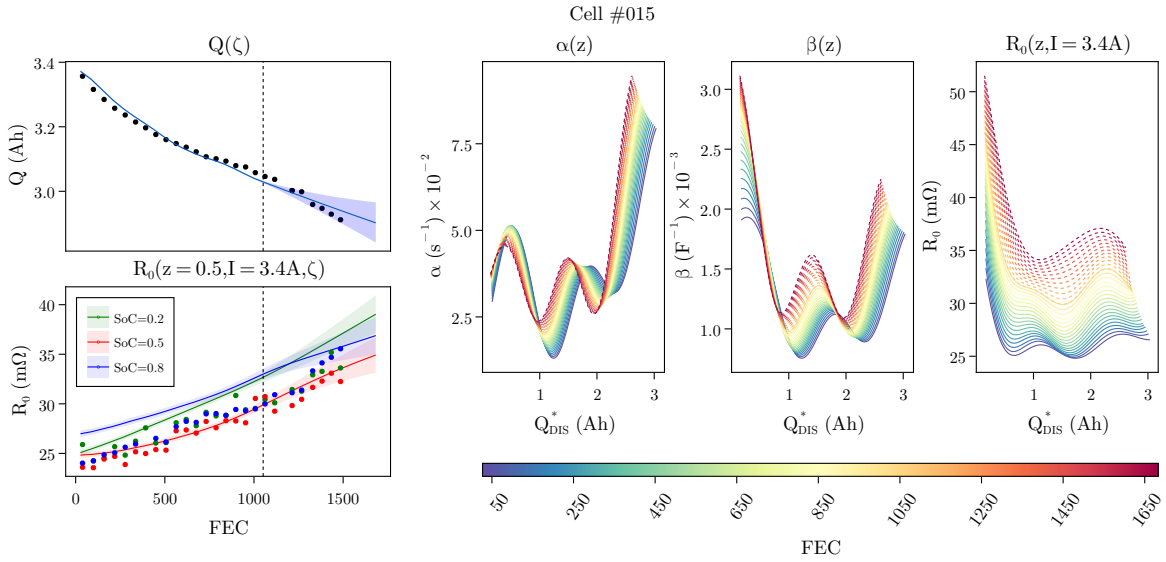


Fig. 4.6 GP posteriors for Q, R_0, α and β calculated for cell #015 in the dataset. The validation data for Q and R_0 indicate that the model accurately captures the evolution of state of health directly from operating data. The right three plots, showing α, β and R_0 as a function of discharge capacity in each case show the strong dependency of ECM parameters on SOC. The colours from blue -> red indicate indicate battery age, where age is defined as the full equivalent cycle count (FEC), which is the total charge throughput divided by the cell nominal capacity. The dashed lines in the three plots denote extrapolated points in time equivalent to the points in Q and R_0. The confidence bounds in these three are not displayed for clarity.

This is also reflected in the substantial increase in the rate of heat generated towards the end of the discharge cycle (Fig. 4.5). Each of the functions are shown in more detail in Fig. 4.7 with each dependency on state of charge and current, together with their 2σ credible intervals at the beginning of life (i.e. where $\zeta = \zeta_0$). As implied by the (inverse) length scale hyperparameter $\gamma_{R_0,I}$, R_0 has virtually no dependency on the applied current, with a 2σ range of less than 1 mΩ over the observed operating range, whereas the SOC dependency is estimated to have a range of approximately 8 mΩ at beginning of life. Similarly α and β have substantial ranges over SOC, with an estimated GP length scale that is much shorter over SOC. The short length scale is reflected in the rapid increase in predictive uncertainty in between the points at coordinates $\mathbf{U}$ that are propagated through time in state-space.

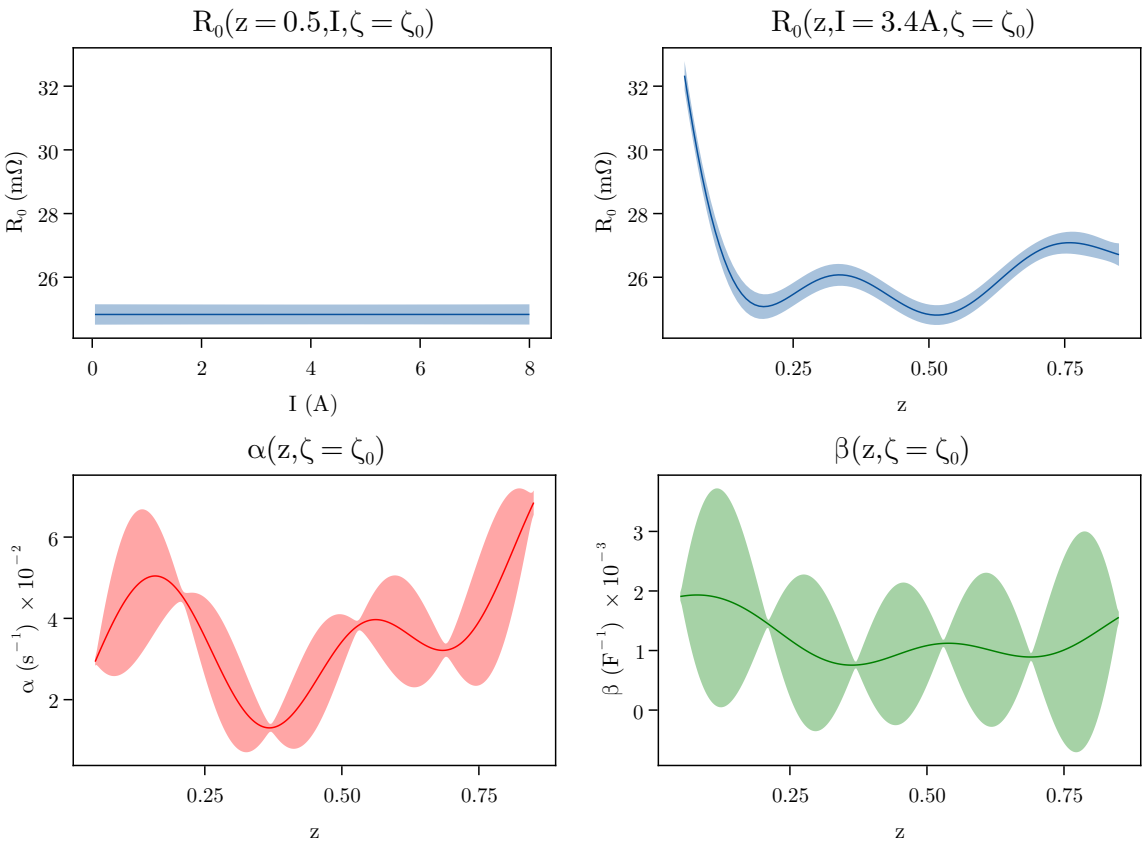


Fig. 4.7 ECM parameter function estimates for α, β and R_0 at the first checkup cycle. Shaded areas indicate $\pm 2\sigma$ credible intervals.

Table 4.3 summarizes the ranges of each circuit parameter over operating conditions at $\zeta = \zeta_0$, and the average change over lifetime. The conventional RC parameterisation in terms of R_1 and C_1 may be retrieved from α and β, and the SOC and lifetime dependencies are shown in Fig. 4.8. As R_1 and C_1 are nonlinear transformations of α and β, their posterior distributions are in fact non-Gaussian and are heavily skewed due to substantial probability mass near zero, especially in β. Therefore the values reported in Table 4.3 and Fig. 4.8 are the median values retrieved by sampling from $1/\alpha$, $1/\beta$ and β/α for R_1C_1, C_1 and R_1 respectively.

Cell	Case	Parameter					
		R_0 (mΩ)	α (s^{-1})	β (C^{-1})	R_1C_1 (s)	C_1 (F)	R_1 (mΩ)
#009	BoL range	8.0	0.064	0.001	62.0	880.0	0.043
	Δ Lifetime	12.0	0.006	0.0004	-1.5	-200.0	0.011
#015	BoL range	7.5	0.066	0.0012	64.0	910.0	0.042
	Δ Lifetime	11.0	0.0056	0.0004	-4.3	-200.0	0.0071

Table 4.3 Ranges of mean RC-circuit parameter estimates over operating conditions and average change over lifetime.

Validation

From Fig. 4.6, it is clear that both the interpolated and extrapolated errors in capacity and R_0 estimates from the GP are small and the means compare well to the values retrieved during checkup tests. The errors are shown in Table 4.4, with an average of 0.016 A h for capacity and 1.8 mΩ for internal resistance. However, the credible intervals imply that the GP is overconfident in its predictions. This is partially due to the intervals applying only to the WV process; the full credible intervals, including the noise kernel are broader (see Figs. C.2, C.3 in the Appendix), although they remain overconfident.

The error in R_0 is uneven across SOC, with the average for $z = 0.5$ being 1.1 mΩ, and 2.1 mΩ where $z = 0.2$ and $z = 0.8$. This is also clear in Fig. 4.6, where cases $z = 0.2$ and $z = 0.8$ over-estimate R_0. The over-estimation in these instances may be partially due to the properties of the SE kernel and the presence of a 'change point' [187] in the GP describing R_0—the rapid increase in R_0 at low SOC results in the length scale hyperparameter for R_0 being relatively short, which causes fluctuations in the estimate at higher SOC as well. In other words, R_0 would be modelled better by a GP that has a length scale that is a function over the input domain, such as the Gibbs kernel [188], discussed in Appendix C.4. This, however, introduces extra complexity to the model because the function determining the length scale as a function of the inputs would have to be specified.

Cell	Case	Q (Ah)	R_0(z, I = 3.4 A) (mΩ)		
			z=0.2	z=0.5	z=0.8
#009	Interpolated	0.018	2.0	0.92	2.7
	Extrapolated	0.014	2.2	0.98	1.8
#015	Interpolated	0.017	2.0	0.83	2.8
	Extrapolated	0.016	1.9	1.5	1.7

Table 4.4 RMSE values for capacity and internal resistance for the mean of the spatially resolved smooth GP vs. checkup tests, split by cell and GP interpolation/extrapolation cases.

In the case of α and β, the wide credible intervals between the discretisation points over SOC, consistent with the optimal hyperparameters for α and β (Table 4.2), with large magnitude and short length scale, imply that the process is highly uncertain—the wide credible intervals increase the predictive uncertainty of $V_{1,t}$, penalizing the error in the voltage less in the likelihood calculation. However, the magnitude of fluctuations in α and β over SOC mean that it is unlikely these are GP 'artefacts' similar to R_0. It is also the case that the wide credible intervals for α and β in Fig. 4.7 are partially due to the sparsity of the discretisation over the SOC domain for the two functions. With a denser set of points (more than the 6 points that were chosen), the estimate would show higher confidence. However, computational effort for the (E)KF scales as $\mathcal{O}(m^3)$ with the size of the state vector m [189].

Interpreting parameter dependencies on battery states and age

The series resistor R_0 in the circuit (Fig. 4.2) corresponds to the sum of ohmic and kinetic overpotentials, where the latter is typically assumed to follow Butler-Volmer kinetics (2.14d). However, R_0 is estimated to have no dependency on the applied current, whereas Butler-Volmer should show a negative relation. Therefore, R_0 is effectively a linear resistor, so these results justify the common approach to linearise Butler-Volmer kinetics [190], certainly in the case of the Li-ion cells used here. The nonlinear relationship of resistance with respect to SOC is to a degree expected, as exchange current density is a function of state of charge. However, the 'shape' of the dependency is not consistent with a symmetric charge transfer coefficient,

implicit in the expression for kinetic overpotential (2.14d), (2.14e). The profiles are nonetheless similar to those from experimental measurements with other Li-ion chemistries [191, 192]. Overall, the entire R_0 surface shifts upward as the cell ages, indicating a decrease in the reaction constant of either the anode or cathode. The degradation in capacity (top left Fig. 4.6) at early stages, measured during the first 20 checkup cycles, arguably shows a slowing trend, which would imply diffusion-limited SEI growth on the anode [73]. If this is the dominant ageing mechanism, the increase in R_0 is most likely due to the reaction current (i.e. the product of electrode surface area and exchange current density) on the anode decreasing. In the last few checkup cycles, a slightly increased rate of capacity degradation is observed, although this is not reflected as an acceleration in the increase in R_0. The upward shift in R_0 is not even across SOC as shown in Fig. 4.6; at the top of the SOC range, the curve shifts approximately 8 mΩ, whereas at the bottom end of the range, the shift is nearly 20 mΩ. The cells have a graphite anode with added silicon, where the latter often causes accelerated degradation of the electrode due to the large change in the volume of the silicon particles during charge and discharge, resulting in loss of active material [193]. As silicon participates most actively in the intercalation reactions at low SOC [194], it is consistent to see more substantial change in R_0 at low SOC due to mechanical degradation over cycling.

Interpreting changes in the RC parameters is more difficult. First, it is not clear which physical process the RC time constant can be attributed to. Couto et al. [52] show that using a low order Padé approximation of the PDEs describing solid-state diffusion in the electrodes results in a first-order system for the surface concentration at each electrode, where the time constant can be appropriately scaled to recover the diffusional time-scale. Using their method, the time constant represented by the GP α implies an average diffusional time-scale of $\approx$1000 s. However, as discussed in Chapter 2, it is not possible to firmly attribute this to either of the electrodes. If α does represent solid-state diffusion, the time-scale would be an average of the two electrodes, weighted according to half-cell OCP gradients.

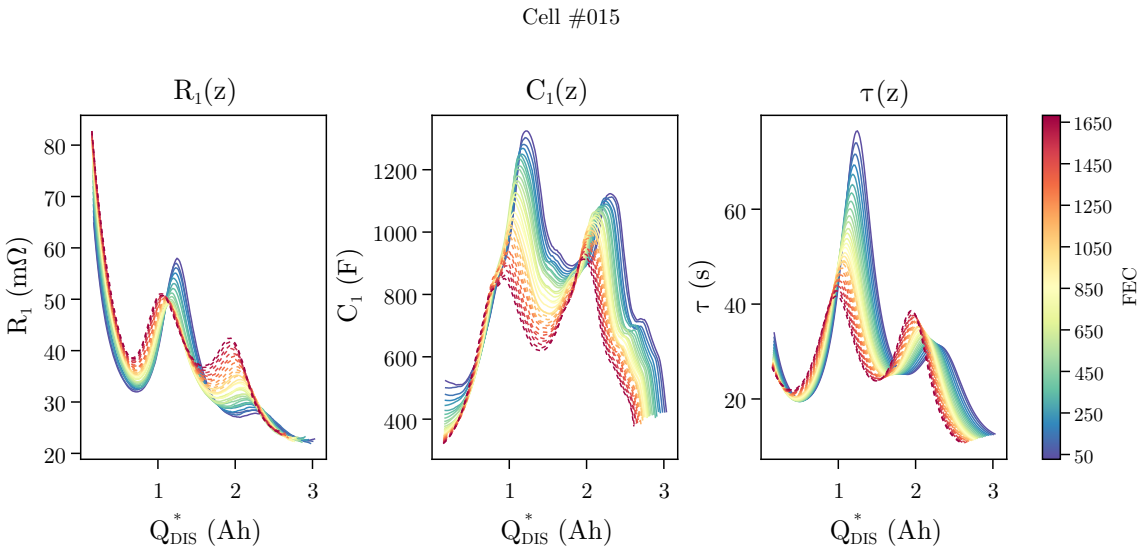


Fig. 4.8 Conventional RC-circuit parameterisation calculated from sampling from the ratio of GPs β/α, $1/\beta$ and $1/\alpha$ shown over lifetime. As the inverse of the GPs are no longer normally distributed and a strongly skewed, the plots here show the median of each of the inverses.

Another possible interpretation for α and β is the reaction relaxation time constant derived by Lin et al. [195], which describes the relaxation process of the inhomogeneity of SOC over the thickness of the electrode, originating from work by Newman and Tobias [196]. The time constant also shows very little drift, on average, over lifetime (Table 4.3), although at high SOC there is some increase matched by a decrease at lower SOC, as shown in Fig. 4.8. This is the net effect from R_1 increasing by approximately 25% on average and C_1 decreasing equally. Ultimately, the time constant from a single RC pair ECM is difficult to connect with physics-based models as it likely reflects an average of several processes.

4.6 Conclusions

In this chapter, GP regression was used to estimate the dependency of equivalent circuit parameters on battery states and operating conditions in a unified framework that estimates functional parameters and states simultaneously. This is an extension of the method proposed in

Chapter 3, showcasing the ability to parameterise more complicated models where the problem becomes nonlinear. The identifiability of the overall system was established via simulation, and the framework was then applied to parameterise a circuit model from experimental data. The results were validated against independent direct measurements of internal resistance and discharge capacity, showing a low error in both estimates.

While previous work [30, 31] has used similar joint state/parameter estimation approaches for battery models, incorporating GP regression into the process for parameters has multiple advantages. Firstly, using a GP kernel function to describe the evolution of functional parameters over lifetime gives a flexible method to extrapolate them into the future—existing literature has either used random walks or simple deterministic models for this purpose. Second, by incorporating the operating point dependency of each of the parameters into the framework, their estimates are more stable across lifetime in real-world scenarios where conditions vary. Imposing the GP prior also mitigates numerical ill-conditioning by acting as a regularisation mechanism in situations where parameters are not easily identifiable, e.g. when estimating resistance with very low currents. Furthermore, the Bayesian framework provides estimates of uncertainty that are a function of the amount of data in the training set in the vicinity of currently observed operating conditions and lifetime.

In contrast to physics-based models, the ECM framework requires only minimal previous knowledge of battery parameters, specifically the open-circuit voltage curve. In this study, the thermal model was parameterised using a heat capacity value from the literature. The thermal model, while not strictly necessary for the estimation of the electrical parameters, gives another constraint for the state dynamics and improves the identifiability of the system. Moreover, because of its simplicity, the framework is agnostic to chemistry and battery construction.

For future work, the applicability of this method could be shown in the case where only partial charging/discharging is observed over lifetime, which would more realistically reflect battery usage. In this case, the assumption of independence of the different functional parame-

ters can be relaxed. For example, by introducing a nonzero prior covariance between parameters in their evolution, readily identifiable parameters such as internal resistance could directly be related to remaining capacity, while taking into account the dependency of resistance on battery states and operating conditions.

Chapter 5

Conclusions

5.1 Contributions

The aim of this work was to determine whether it is possible to accurately estimate battery state of health from real-world operating data, with no access to direct measurements of either capacity or internal resistance. In answering this, the main contribution of this work is the implementation of Bayesian methods to achieve robust state of health estimation of battery systems. The approach combines existing data- and model-driven SOH estimation techniques, effectively posing battery health diagnosis and prognosis as a parameter estimation problem, but where battery model parameters are considered to be non-parametric functions of operating conditions, battery states and lifetime.

By using Gaussian process regression to augment basic equivalent circuit models, imposing functional dependencies on circuit parameters, the flexibility of ECMs is improved. Moreover, by (loosely) defining these dependencies through GPs, the battery model retains interpretability and comparability to physics-based approaches, while avoiding the complex parameterisation and identifiability issues related to the latter—discussed in Chapter 2. The GP framework simultaneously solves the model parameter identifiability problem, which is encapsulated in

the posterior-predictive distribution. The credible intervals in this distribution describe the practical identifiability of parameters at any point in the input space.

A computationally efficient GPR method was applied to a large fleet of PV-connected VRLA batteries in Chapter 3, where the series resistor of a simple Thevenin circuit was modelled as a GP over battery lifetime and operating conditions. Here it was shown that the method provides a SOH estimate that may be normalised across the population, so that like-for-like comparisons can be made between different batteries at consistent operating conditions. This method was validated by using the SOH metric to predict end-of-life failure using a Gaussian process classifier. In addition, it was shown how the performance of end-of-life prediction can be improved by considering additional inputs, such as cumulative stress factors known to affect battery life.

The versatility of the GP method was shown by applying the same technique to a more complex electrical and thermal model for a Li-ion cell in Chapter 4, where the usage profile was similar to that of a real-world electric vehicle. In this case, both battery states and the functions describing the model parameters were estimated simultaneously, showing how the approach works in an online setting. This application to a second battery type also shows how the approach is largely chemistry-agnostic. It could also be applied at the cell, module or pack level, where the precision of parameter estimates would be limited by the availability (or granularity) of telemetry data. In all cases, the set of prior data required is minimal, consisting only of the cell-level open-circuit voltage for a healthy cell. The thermal model in Chapter 4 also used parameters from the literature, but this model is not strictly necessary for the system, and was added as an extra dynamic state, improving the overall identifiability of the system. The validation of the circuit model in this case was done with measured DC internal resistance and constant-current discharge capacity, where in both cases the model was found to be accurate.

5.2 Future work

The work presented in this thesis prompts a number of future research questions. These centre on improving the robustness and accuracy of the existing methods, as well as using Bayesian techniques to extend the framework at both single battery and fleet levels.

5.2.1 Robustness of hyperparameter estimates and kernel selection

In both Chapters 3 and 4, MAP estimates for the hyperparameters were used. This was due to the computational effort that would have been required for a more thorough evaluation of the hyperparameter posteriors. This could have been done either by sampling techniques such as MCMC, or optimisation-based approximations such as variational inference. In this case, the credible intervals (shown, e.g., in Figs. 4.6, 4.7), calculated by marginalising over the hyperparameter posterior, would have been more accurate. In a similar vein, kernel functions were chosen *a priori*. The principled Bayesian approach in this respect would have been to do rigorous model selection based on marginal likelihood (i.e., the integral of the likelihood function over hyperparameter priors). This is also a very time-consuming task, which could be done through sampling methods targeted specifically to calculate the marginal likelihood [197].

5.2.2 Co-evolution of circuit parameters

The co-evolution of (i.e., identifying correlations between) ECM parameters over battery lifetime is an outstanding question. This is of interest because the data required for the identifiability of battery parameters such as capacity and resistance are different—in principle, internal resistance, although a function of SOC, temperature and applied current, is identifiable from a single, short current pulse. Capacity on the other hand requires a non-negligible depth of discharge to be accurately estimated. Therefore, estimating the function relating capacity and internal resistance to each other over battery lifetime is of interest, as this would enable

more accurate capacity estimates from much shorter discharge segments at varying operating conditions. In this work, the outputs of the GPs estimated in Chapter 4 were considered independent, that is, zero cross-covariance between them was assumed. This assumption could be relaxed by setting a prior covariance between, e.g., the series resistance and the capacity. The nonparametric approach is well suited to estimating the function between the two. There is some earlier work estimating it using parametric methods [35, 74], but it is likely that the functional form depends on the degradation mechanisms experienced by the cell.

5.2.3 Fleet level estimates as prior expectations

In Chapter 4, a GP with a WV kernel describing the ageing process was used to model circuit parameters. In reality, this model is an over-simplification and will not be able to extrapolate beyond the so-called 'knee' or 'elbow' points for capacity and internal resistance respectively. A Bayesian approach to this challenge of long-range prediction would be to build an expectation of the behaviour of state of health as a function of stress factors from an entire fleet of batteries. This would then serve as a prior mean function for SOH evolution of an individual battery. In earlier work [91], this has been shown using an (empirical) ageing model parameterised using data from a single cell as the mean function, rather than an expectation built at the fleet level based on real behaviour in the field. The prior mean would serve two purposes—first, it would give an estimate of circuit parameters in situations where identifiability is poor due to lack of data (e.g. if there are usage gaps or very short charge/discharge cycles). Second, it would also serve as a long-term predictor of SOH, in combination with the GP, where the latter would describe the battery specific variation around the population mean.

References

[1] Antti Aitio, Scott G. Marquis, Pedro Ascencio, and David Howey. Bayesian parameter estimation applied to the Li-ion battery single particle model with electrolyte dynamics. *IFAC-PapersOnLine*, 53(2):12497–12504, 2020.

[2] Antti Aitio and David Howey. Combining non-parametric and parametric models for stable and computationally efficient battery health estimation. *Proceedings of the ASME 2020 Dynamic Systems and Control Conference*, 2020.

[3] Antti Aitio and David A. Howey. Predicting battery end of life from solar off-grid system field data using machine learning. *Joule*, 5(12):3204–3220, 2021.

[4] Antti Aitio, Dominik Jöst, Dirk Uwe Sauer, and David A. Howey. Learning battery model parameter dynamics from data with recursive Gaussian process regression, 2022.

[5] Valentin Sulzer, Peyman Mohtat, Antti Aitio, Suhak Lee, Yen T. Yeh, Frank Steinbacher, Muhammad Umer Khan, Jang Woo Lee, Jason B. Siegel, Anna G. Stefanopoulou, and David A. Howey. The challenge and opportunity of battery lifetime prediction from field data. *Joule*, 5(8):1934–1955, 2021.

[6] Luis D. Couto, Dong Zhang, Antti Aitio, Scott Moura, and David Howey. Estimation of Parameter Probability Distributions for Lithium-Ion Battery String Models Using Bayesian Methods. *Proceedings of the ASME 2020 Dynamic Systems and Control Conference*, pages 1–10, 2020.

[7] Scott G. Marquis, Valentin Sulzer, Robert Timms, Colin P. Please, and S. Jon Chapman. An Asymptotic Derivation of a Single Particle Model with Electrolyte. *Journal of The Electrochemical Society*, 166(15):A3693–A3706, 2019.

[8] Błazej Miasojedow, Eric Moulines, and Matti Vihola. An adaptive parallel tempering algorithm. *Journal of Computational and Graphical Statistics*, 22(3):649–664, 2013.

[9] Dharik S. Mallapragada, Nestor A. Sepulveda, and Jesse D. Jenkins. Long-run system value of battery energy storage in future grids with increasing wind and solar generation. *Applied Energy*, 275(June):115390, 2020.

[10] INTERNATIONAL ENERGY AGENCY. Global EV Outlook 2021 - Accelerating ambitions despite the pandemic, 2021.

[11] ESMAP. Energy Sector Management Assistance Program. 2019. Mini Grids for Half a Billion People : Market Outlook and Handbook for Decision Makers. Technical report, World Bank, 2019.

[12] Micah S. Ziegler, Juhyun Song, and Jessika E. Trancik. Determinants of lithium-ion battery technology cost decline. *Energy & Environmental Science*, 14:6074–6098, 2021.

[13] U.S. Department of Energy. Energy Storage Grand Challenge Energy Storage Market Report 2020. *U.S. Department of Energy*, Technical(December):65, 2020.

[14] IRENA. Solar Pv in Africa: Costs and market. Technical Report September, IRENA, 2016.

[15] Jorn M. Reniers, Grietus Mulder, Sina Ober-Blöbaum, and David A. Howey. Improving optimal control of grid-connected lithium-ion batteries through more accurate battery and degradation modelling. *Journal of Power Sources*, 379(December 2017):91–102, 2018.

[16] Kristen A. Severson, Peter M. Attia, Norman Jin, Nicholas Perkins, Benben Jiang, Zi Yang, Michael H. Chen, Muratahan Aykol, Patrick K. Herring, Dimitrios Fraggedakis, Martin Z. Bazant, Stephen J. Harris, William C. Chueh, and Richard D. Braatz. Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy*, 4(5):383–391, 5 2019.

[17] Gonçalo dos Reis, Calum Strange, Mohit Yadav, and Shawn Li. Lithium-ion battery data and where to find it. *Energy and AI*, 5, 2021.

[18] Matthieu Dubarry, George Baure, and David Anseán. Perspective on State-of-Health Determination in Lithium-Ion Batteries. *Journal of Electrochemical Energy Conversion and Storage*, 17(4):1–8, 2020.

[19] Kieran Mc Carthy, Hemtej Gullapalli, Kevin M. Ryan, and Tadhg Kennedy. Review—Use of Impedance Spectroscopy for the Estimation of Li-ion Battery State of Charge, State of Health and Internal Temperature. *Journal of The Electrochemical Society*, 168(8):080517, 2021.

[20] Greg Davies, Kevin W. Knehr, Barry Van Tassell, Thomas Hodson, Shaurjo Biswas, Andrew G. Hsieh, and Daniel A. Steingart. State of Charge and State of Health Estimation Using Electrochemical Acoustic Time of Flight Analysis. *Journal of The Electrochemical Society*, 164(12):A2746–A2755, 2017.

[21] Peyman Mohtat, Suhak Lee, Jason B. Siegel, and Anna G. Stefanopoulou. Towards better estimability of electrode-specific state of health: Decoding the cell expansion. *Journal of Power Sources*, 427(January):101–111, 2019.

[22] Christoph R. Birkl, Matthew R. Roberts, Euan McTurk, Peter G. Bruce, and David A. Howey. Degradation diagnostics for lithium ion cells. *Journal of Power Sources*, 341:373–386, 2 2017.

[23] Paul Ruetschi. Aging mechanisms and service life of lead-acid batteries. *Journal of Power Sources*, 127(1-2):33–44, 2004.

[24] Trishna Raj, Andrew A. Wang, Charles W. Monroe, and David A. Howey. Investigation of Path-Dependent Degradation in Lithium-Ion Batteries. *Batteries & Supercaps*, pages 1–10, 9 2020.

[25] M. Berecibar, I. Gandiaga, I. Villarreal, N. Omar, J. Van Mierlo, and P. Van Den Bossche. Critical review of state of health estimation methods of Li-ion batteries for real applications, 2016.

[26] Rui Xiong, Linlin Li, and Jinpeng Tian. Towards a smarter battery management system: A critical review on battery state of health monitoring methods. *Journal of Power Sources*, 405(5):18–29, 2018.

[27] Yi Li, Kailong Liu, Aoife M. Foley, Alana Zülke, Maitane Berecibar, Elise Nanini-Maury, Joeri Van Mierlo, and Harry E. Hoster. Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review. *Renewable and Sustainable Energy Reviews*, 113:109254, 10 2019.

[28] Shida Jiang and Zhengxiang Song. A review on the state of health estimation methods of lead-acid batteries. *Journal of Power Sources*, 517(November 2021):230710, 2022.

[29] C. Cobelli and J. J. DiStefano. Parameter and structural identifiability concepts and ambiguities: a critical review and analysis. *The American journal of physiology*, 239(1):7–24, 1980.

[30] Gregory L. Plett. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation. *Journal of Power Sources*, 134(2):277–292, 8 2004.

[31] Gregory L Plett. Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 2: Simultaneous state and parameter estimation. *Journal of Power Sources*, 161:1369–1384, 2006.

[32] Il Song Kim. A technique for estimating the state of health of lithium batteries through a dual-sliding-mode observer. *IEEE Transactions on Power Electronics*, 25(4):1013–1022, 2010.

[33] Atsushi Baba and Shuichi Adachi. Simultaneous state of charge and parameter estimation of lithium-ion battery using log-normalized unscented Kalman Filter. In *2015 American Control Conference (ACC)*, pages 311–316. IEEE, 7 2015.

[34] Chao Hu, Byeng D. Youn, and Jaesik Chung. A multiscale framework with extended Kalman filter for lithium-ion battery SOC and capacity estimation. *Applied Energy*, 92:694–704, 2012.

[35] Andrew Chu, Anirudh Allam, Andrea Cordoba Arenas, Giorgio Rizzoni, and Simona Onori. Stochastic capacity loss and remaining useful life models for lithium-ion batteries in plug-in hybrid electric vehicles. *Journal of Power Sources*, 478(April):228991, 2020.

[36] Dong Zhang, Satadru Dey, Hector E. Perez, and Scott J. Moura. Real-time capacity estimation of lithium-ion batteries utilizing thermal dynamics. *IEEE Transactions on Control Systems Technology*, 28(3):992–1000, 2020.

[37] Seyed Mohammad Mahdi Alavi, Adam Mahdi, Stephen J Payne, and David A Howey. Identifiability of Generalized Randles Circuit Models. *IEEE Transactions on Control Systems Technology*, 25(6):2112–2120, 2017.

[38] William Ford. *Numerical Linear Algebra with Applications*. Elsevier Science & Technology, San Diego, 1st ed. edition, 2014.

[39] Alejandro F. Villaverde, Neil D. Evans, Michael J. Chappell, and Julio R. Banga. Input-dependent structural identifiability of nonlinear systems. *IEEE Control Systems Letters*, 3(2):272–277, 2019.

[40] Alejandro F. Villaverde. Observability and Structural Identifiability of Nonlinear Biological Systems. *Complexity*, 2019, 2019.

[41] Jamie Gomez, Ruben Nelson, Egwu E. Kalu, Mark H. Weatherspoon, and Jim P. Zheng. Equivalent circuit model parameters of a high-power Li-ion battery: Thermal and state of charge effects. *Journal of Power Sources*, 196(10):4826–4831, 5 2011.

[42] Jürgen Remmlinger, Michael Buchholz, Markus Meiler, Peter Bernreuter, and Klaus Dietmayer. State-of-health monitoring of lithium-ion batteries in electric vehicles by on-board internal resistance estimation. *Journal of Power Sources*, 196(12):5325–5331, 2011.

[43] Jürgen Remmlinger, Michael Buchholz, Thomas Soczka-Guth, and Klaus Dietmayer. On-board state-of-health monitoring of lithium-ion batteries using linear parameter-varying models. *Journal of Power Sources*, 239:689–695, 10 2013.

[44] John Newman and Karen E. Thomas-Alyea. *Electrochemical Systems*. Prentice-Hall international series in the physical and chemical engineering sciences. Wiley, Hoboken, N.J., 3rd editio edition, 1973.

[45] Marc Doyle, Thomas F. Fuller, and John Newman. Modeling of Galvanostatic Charge and Discharge of the Lithium/Polymer/Insertion Cell. *Journal of The Electrochemical Society*, 140(6):1526, 6 1993.

[46] Thomas F. Fuller, Marc Doyle, and John Newman. Simulation and Optimization of the Dual Lithium Ion Insertion Cell. *Journal of The Electrochemical Society*, 141(1):1, 1 1994.

[47] Valentin Sulzer, S. Jon Chapman, Colin P. Please, David A. Howey, and Charles W. Monroe. Faster Lead-Acid Battery Simulations from Porous-Electrode Theory: Part II. Asymptotic Analysis. *Journal of The Electrochemical Society*, 166(12):A2372–A2382, 7 2019.

[48] Valentin Sulzer, S. Jon Chapman, Colin P. Please, David A. Howey, and Charles W. Monroe. Faster Lead-Acid Battery Simulations from Porous-Electrode Theory: Part I. Physical Model. *Journal of The Electrochemical Society*, 166(12):A2363–A2371, 7 2019.

[49] A. M. Bizeray, S. Zhao, S. R. Duncan, and D. A. Howey. Lithium-ion battery thermal-electrochemical model-based state estimation using orthogonal collocation and a modified extended Kalman filter. *Journal of Power Sources*, 296:400–412, 2015.

[50] Ali Jokar, Barzin Rajabloo, Martin Désilets, and Marcel Lacroix. Review of simplified Pseudo-two-Dimensional models of lithium-ion batteries. *Journal of Power Sources*, 327:44–55, 2016.

[51] Julien Schorsch, Luis D. Couto, and Michel Kinnaert. SOC and SOH estimation for Li-ion battery based on an equivalent hydraulic model. Part II: SOH power fade estimation. *Proceedings of the American Control Conference*, 2016-July:4029–4034, 2016.

[52] Luis D. Couto, Julien Schorsch, Marco M. Nicotra, and Michel Kinnaert. SOC and SOH estimation for Li-ion battery based on an equivalent hydraulic model. Part I: SOC and surface concentration estimation. *Proceedings of the American Control Conference*, 2016-July:4029–4034, 2016.

[53] Valentin Sulzer, Scott G Marquis, Robert Timms, Martin Robinson, and S Jon Chapman. Python Battery Mathematical Modelling (PyBaMM). *ECSarXiv. February*, 7, 2020.

[54] Adrien M. Bizeray, Jin Ho Kim, Stephen R. Duncan, and David A. Howey. Identifiability and Parameter Estimation of the Single Particle Lithium-Ion Battery Model. *IEEE Transactions on Control Systems Technology*, 27(5):1862–1877, 2018.

[55] Dong Zhang, Luis D. Couto, and Scott J. Moura. Electrode-Level State Estimation in Lithium-Ion Batteries via Kalman Decomposition. *IEEE Control Systems Letters*, 5(5):1657–1662, 2021.

[56] Scott J. Moura, Nalin A. Chaturvedi, and Miroslav Krstic. Adaptive Partial Differential Equation Observer for Battery State-of-Charge/State-of-Health Estimation Via an Electrochemical Model. *Journal of Dynamic Systems, Measurement, and Control*, 136(1):011015, 10 2013.

[57] Madeleine Ecker, Thi Kim, Dung Tran, Philipp Dechent, and Stefan Käbitz. Parameterisation of a Physico-Chemical Model of a Lithium-Ion Battery Part I : Determination of Parameters. *Journal of the Electrochemical*, 162(9), 2015.

[58] Hao Tu, Scott Moura, Yebin Wang, and Huazhen Fang. Integrating Physics-Based Modeling with Machine Learning for Lithium-Ion Batteries. *arXiv preprint arXiv:2112.12979*, 2021.

[59] Jennifer Brucker, Wolfgang G. Bessler, and Rainer Gasper. Grey-box modelling of lithium-ion batteries using neural ordinary differential equations, 2021.

[60] Daniel J Tait, Ferran Brosa Planella, Theodoros Damoulas, and W Dhammika Widanage. Scalable Multitask Latent Force Models with Applications to Predicting Lithium-ion Concentration. *NeurIPS*, 2020.

[61] Gregory L Plett. Recursive approximate weighted total least squares estimation of battery cell total capacity. *Journal of Power Sources*, 196(4):2319–2331, 2011.

[62] Simon Schwunk, Nils Armbruster, Sebastian Straub, Johannes Kehl, and Matthias Vetter. Particle filter for state of charge and state of health estimation for lithium–iron phosphate batteries. *Journal of Power Sources*, 239:705–710, 10 2013.

[63] Jiani Du, Zhitao Liu, Youyi Wang, and Changyun Wen. An adaptive sliding mode observer for lithium-ion battery state of charge and state of health estimation in electric vehicles. *Control Engineering Practice*, 54:81–90, 9 2016.

[64] Hicham Chaoui, Navid Golbon, Imad Hmouz, Ridha Souissi, and Sofiene Tahar. Lyapunov-based adaptive state of charge and state of health estimation for lithium-ion batteries. *IEEE Transactions on Industrial Electronics*, 62(3):1610–1618, 2015.

[65] Pedro Ascencio, Kirk Smith, Charles W. Monroe, and David Howey. Adaptive observer for charge-state and crossover estimation in disproportionation redox flow batteries undergoing self-discharge. *Proceedings of the American Control Conference*, 2019-July:5452–5457, 2019.

[66] Simo Särkkä. *Bayesian filtering and smoothing*. Cambridge University Press, 2013.

[67] Shi Zhao, Stephen R. Duncan, and David A. Howey. Observability Analysis and State Estimation of Lithium-Ion Batteries in the Presence of Sensor Biases. *IEEE Transactions on Control Systems Technology*, 25(1):326–333, 1 2017.

[68] A. H. Mohamed and K. P. Schwarz. Adaptive Kalman filtering for INS/GPS. *Journal of Geodesy*, 73(4):193–203, 1999.

[69] Isambi S. Mbalawata, Simo Särkkä, and Heikki Haario. Parameter estimation in stochastic differential equations with Markov chain Monte Carlo and non-linear Kalman filtering. *Computational Statistics*, 28(3):1195–1223, 2013.

[70] I. Bloom, B. W. Cole, J. J. Sohn, S. A. Jones, E. G. Polzin, V. S. Battaglia, G. L. Henriksen, C. Motloch, R. Richardson, T. Unkelhaeuser, D. Ingersoll, and H. L. Case. An accelerated calendar and cycle life study of Li-ion cells. *Journal of Power Sources*, 101(2):238–247, 2001.

[71] M. Schimpe, M. E. von Kuepach, M. Naumann, H. C. Hesse, K. Smith, and A. Jossen. Comprehensive Modeling of Temperature-Dependent Degradation Mechanisms in Lithium Iron Phosphate Batteries. *Journal of The Electrochemical Society*, 165(2):A181–A193, 1 2018.

[72] Johannes Schmalstieg, Stefan Käbitz, Madeleine Ecker, and Dirk Uwe Sauer. A holistic aging model for Li(NiMnCo)O2 based 18650 lithium-ion batteries. *Journal of Power Sources*, 257:325–334, 2014.

[73] Jorn M. Reniers, Grietus Mulder, and David A. Howey. Review and Performance Comparison of Mechanical-Chemical Degradation Models for Lithium-Ion Batteries. *Journal of The Electrochemical Society*, 166(14):A3189–A3200, 2019.

[74] Bhaskar Saha, Kai Goebel, Scott Poll, and Jon Christophersen. Prognostics methods for battery health monitoring using a Bayesian framework. *IEEE Transactions on Instrumentation and Measurement*, 58(2):291–296, 2009.

[75] Wei He, Nicholas Williard, Michael Osterman, and Michael Pecht. Prognostics of lithium-ion batteries based on Dempster–Shafer theory and the Bayesian Monte Carlo method. *Journal of Power Sources*, 196(23):10314–10321, 12 2011.

[76] Arijit Guha and Amit Patra. State of Health Estimation of Lithium-Ion Batteries Using Capacity Fade and Internal Resistance Growth Models. *IEEE Transactions on Transportation Electrification*, 4(1):135–146, 2018.

[77] Dong Zhang, Satadru Dey, Hector E. Perez, and Scott J. Moura. Remaining useful life estimation of Lithium-ion batteries based on thermal dynamics. *Proceedings of the American Control Conference*, pages 4042–4047, 2017.

[78] Fangfang Yang, Dong Wang, Yinjiao Xing, and Kwok Leung Tsui. Prognostics of Li(NiMnCo)O2-based lithium-ion batteries using a novel battery degradation model. *Microelectronics Reliability*, 70:70–78, 2017.

[79] Weiming Xian, Bing Long, Min Li, and Houjun Wang. Prognostics of lithium-ion batteries based on the verhulst model, particle swarm optimization and particle filter. *IEEE Transactions on Instrumentation and Measurement*, 63(1):2–17, 2014.

[80] Arijit Guha, Amit Patra, and K. V. Vaisakh. Remaining useful life estimation of lithium-ion batteries based on the internal resistance growth model. In *2017 Indian Control Conference, ICC 2017 - Proceedings*, pages 33–38. IEEE, 2017.

[81] Daniel Zhang, Saurabh Mishra, Erik Brynjolfsson, John Etchemendy, Deep Ganguli, Barbara Grosz, Terah Lyons, James Manyika, Juan Carlos Niebles, Michael Sellitto, Yoav Shoham, Jack Clark, and Raymond Perrault. 2021 AI Index Report. Technical report, Stanford University, 2021.

[82] Hicham Chaoui and Chinemerem Christopher Ibe-Ekeocha. State of Charge and State of Health Estimation for Lithium Batteries Using Recurrent Neural Networks. *IEEE Transactions on Vehicular Technology*, 66(10):8773–8783, 2017.

[83] Hicham Chaoui, Chinemerem C. Ibe-Ekeocha, and Hamid Gualous. Aging prediction and state of charge estimation of a LiFePO4 battery using input time-delayed neural networks. *Electric Power Systems Research*, 146:189–197, 2017.

[84] Weihan Li, Neil Sengupta, Philipp Dechent, David Howey, Anuradha Annaswamy, and Dirk Uwe Sauer. Online capacity estimation of lithium-ion batteries with deep long short-term memory networks. *Journal of Power Sources*, 482:228863, 2021.

[85] Yunwei Zhang, Qiaochu Tang, Yao Zhang, Jiabin Wang, Ulrich Stimming, and Alpha A. Lee. Identifying degradation patterns of lithium ion batteries from impedance spectroscopy using machine learning. *Nature Communications*, 11(1):6–11, 2020.

[86] Verena Klass, Mårten Behm, and Göran Lindbergh. A support vector machine-based state-of-health estimation method for lithium-ion batteries under electric vehicle operation. *Journal of Power Sources*, 270:262–272, 2014.

[87] Robert R. Richardson, Christoph R. Birkl, Michael A. Osborne, and David A. Howey. Gaussian Process Regression for in Situ Capacity Estimation of Lithium-Ion Batteries. *IEEE Transactions on Industrial Informatics*, 15(1):127–138, 2019.

[88] Yi Li, Changfu Zou, Maitane Berecibar, Elise Nanini-Maury, Jonathan C.W. Chan, Peter van den Bossche, Joeri Van Mierlo, and Noshin Omar. Random forest regression for online capacity estimation of lithium-ion batteries. *Applied Energy*, 232(September):197–210, 2018.

[89] Zhenpo Wang, Jun Ma, and Lei Zhang. State-of-Health Estimation for Lithium-Ion Batteries Based on the Multi-Island Genetic Algorithm and the Gaussian Process Regression. *IEEE Access*, 5:21286–21295, 2017.

[90] Gae-won You, Sangdo Park, and Dukjin Oh. Real-time state-of-health estimation for electric vehicle batteries: A data-driven approach. *Applied Energy*, 176:92–103, 8 2016.

[91] Robert R. Richardson, Michael A. Osborne, and David A. Howey. Gaussian process regression for forecasting battery state of health. *Journal of Power Sources*, 357:209–219, 7 2017.

[92] Robert R. Richardson, Michael A. Osborne, and David A. Howey. Battery health prediction under generalized conditions using a Gaussian process transition model. *Journal of Energy Storage*, 23(November 2018):320–328, 2019.

[93] Datong Liu, Jingyue Pang, Jianbao Zhou, Yu Peng, and Michael Pecht. Prognostics for state of health estimation of lithium-ion batteries based on combination Gaussian process functional regression. *Microelectronics Reliability*, 53(6):832–839, 2013.

[94] Shan Yin, Jingyue Pang, Datong Liu, and Yu Peng. Remaining useful life prognostics for lithium-ion battery based on Gaussian processing regression combined with the empirical model. *PHM 2013 - Proceedings of the Annual Conference of the Prognostics and Health Management Society 2013*, pages 608–615, 2013.

[95] Jian Liu and Ziqiang Chen. Remaining useful life prediction of lithium-ion batteries based on health indicator and Gaussian process regression model. *IEEE Access*, 7:39474–39484, 2019.

[96] Samuel Greenbank and David Howey. Automated feature extraction and selection for data-driven models of rapid battery capacity fade and end of life. *IEEE Transactions on Industrial Informatics*, 3203(c):1–9, 2021.

[97] M. Lucu, E. Martinez-Laserna, I. Gandiaga, K. Liu, H. Camblong, W. D. Widanage, and J. Marco. Data-driven nonparametric Li-ion battery ageing model aiming at learning from real operation data – Part A: Storage operation. *Journal of Energy Storage*, 30(October 2019):101409, 2020.

[98] M. Lucu, E. Martinez-Laserna, I. Gandiaga, K. Liu, H. Camblong, W. D. Widanage, and J. Marco. Data-driven nonparametric Li-ion battery ageing model aiming at learning from real operation data - Part B: Cycling operation. *Journal of Energy Storage*, 30(October 2019):101410, 2020.

[99] Lingjun Song, Keyao Zhang, Tongyi Liang, Xuebing Han, and Yingjie Zhang. Intelligent state of health estimation for lithium-ion battery pack based on big data analysis. *Journal of Energy Storage*, 32(August):101836, 2020.

[100] Adnan Nuhic, Tarik Terzimehic, Thomas Soczka-Guth, Michael Buchholz, and Klaus Dietmayer. Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods. *Journal of Power Sources*, 239:680–688, 10 2013.

[101] Yongzhi Zhang, Rui Xiong, Hongwen He, and Michael G. Pecht. Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. *IEEE Transactions on Vehicular Technology*, 67(7):5695–5705, 2018.

[102] Benvolence Chinomona, Chunhui Chung, Lien-Kai Chang, Wei-Chih Su, and Mi-Ching Tsai. Long Short-Term Memory Approach to Estimate Battery Remaining Useful Life Using Partial Data. *IEEE Access*, 8:165419–165431, 2020.

[103] Paula Fermín-Cueto, Euan McTurk, Michael Allerhand, Encarni Medina-Lopez, Miguel F. Anjos, Joel Sylvester, and Gonçalo dos Reis. Identification and machine learning prediction of knee-point and knee-onset in capacity degradation curves of lithium-ion cells. *Energy and AI*, 1, 2020.

[104] Chaolong Zhang, Yigang He, Lifeng Yuan, and Sheng Xiang. Capacity Prognostics of Lithium-Ion Batteries using EMD Denoising and Multiple Kernel RVM. *IEEE Access*, 5:12061–12070, 2017.

[105] M. Lucu, E. Martinez-Laserna, I. Gandiaga, and H. Camblong. A critical review on self-adaptive Li-ion battery ageing models. *Journal of Power Sources*, 401(May):85–101, 2018.

[106] Laurent Valentin Jospin, Wray Buntine, Farid Boussaid, Hamid Laga, and Mohammed Bennamoun. Hands-on Bayesian Neural Networks – a Tutorial for Deep Learning Users. *IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE*, 17(2):29–48, 2022.

[107] Shanghai Electric Vehicle Public Data Collecting, Monitoring and Research Center, 2021.

[108] Qiushi Wang, Zhenpo Wang, Lei Zhang, Peng Liu, and Zhaosheng Zhang. A novel consistency evaluation method for series-connected battery systems based on real-world operation data. *IEEE Transactions on Transportation Electrification*, 7(2):1–1, 2020.

[109] Qian Huo, Zhikai Ma, Xiaoshun Zhao, Tao Zhang, and Yulong Zhang. Bayesian Network Based State-of-Health Estimation for Battery on Electric Vehicle Application and its Validation through Real-World Data. *IEEE Access*, 9:11328–11341, 2021.

[110] Simo Särkkä, Arno Solin, and Jouni Hartikainen. Spatiotemporal learning via infinite-dimensional bayesian filtering and smoothing: A look at gaussian process regression through kalman filtering. *IEEE Signal Processing Magazine*, 30(4):51–61, 2013.

[111] Arno Solin. *Stochastic Differential Equation Methods for Spatio-Temporal Gaussian Process Regression*. PhD thesis, Aalto University, 2016.

[112] Dominik Jöst, Florian Ringbeck, Alexander Blömeke, and Dirk Uwe Sauer. Timeseries data of a drive cycle aging test of 28 high energy NCA/C+Si round cells of type 18650, 2021.

[113] Arun K. Tangirala. Principles of system identification : theory and practice, 2015.

[114] Joel C. Forman, Scott J. Moura, Jeffrey L. Stein, and Hosam K. Fathy. Genetic identification and fisher identifiability analysis of the Doyle–Fuller–Newman model from experimental cycling of a LiFePO4 cell. *Journal of Power Sources*, 210:263–275, 7 2012.

[115] Xinfan Lin and Anna G. Stefanopoulou. Analytic Bound on Accuracy of Battery State and Parameter Estimation. *Journal of The Electrochemical Society*, 162(9):A1879–A1891, 2015.

[116] A W van der Vaart. *Asymptotic statistics*. Cambridge series on statistical and probabilistic mathematics ; 3. Cambridge University Press, Cambridge, 1998.

[117] R. Bellman and K. J. Åström. On structural identifiability. *Mathematical Biosciences*, 7(3-4):329–339, 1970.

[118] Ross Drummond and Stephen R. Duncan. Structural identifiability of a pseudo-2D Li-ion battery electrochemical model. *IFAC-PapersOnLine*, 53(2):12452–12458, 2020.

[119] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[120] Tom Gunter, Michael A Osborne, Roman Garnett, Philipp Hennig, and Stephen J Roberts. Sampling for Inference in Probabilistic Models with Fast Bayesian Quadrature. In Z Ghahramani, M Welling, C Cortes, N Lawrence, and K Q Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[121] Kevin P. Murphy. *Machine learning : a probabilistic perspective*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass., 2012.

[122] Michael Betancourt. A Conceptual Introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*, 2017.

[123] Steve Brooks. *Handbook for Markov chain Monte Carlo [electronic resource]*. Chapman & Hall/CRC handbooks of modern statistical methods. Taylor & Francis, Boca Raton, Fla., 2011.

[124] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[125] W. K. Hastings. Monte carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[126] Stuart Geman and Donald Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, 1984.

[127] Matti Vihola. Robust adaptive Metropolis algorithm with coerced acceptance rate. *Statistics and Computing*, 22(5):997–1008, 9 2012.

[128] Radford M. Neal. MCMC using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, pages 113–162, 2011.

[129] Matthew D Hoffman and Andrew Gelman. The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15:1593–1623, 2014.

[130] Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120(1 SPEC. ISS.):221–259, 2009.

[131] Stan Development Team. Stan Modeling Language Users Guide and Reference Manual 2.29, 2019.

[132] Kai Xu, Hong Ge, Will Tebbutt, Mohamed Tarek, Martin Trapp, and Zoubin Ghahramani. AdvancedHMC. jl: A robust, modular and efficient implementation of advanced HMC algorithms. In *Symposium on Advances in Approximate Bayesian Inference*, pages 1–10. PMLR, 2020.

[133] Michael Clerx, Martin Robinson, Ben Lambert, Chon Lok Lei, Sanmitra Ghosh, Gary R. Mirams, and David J. Gavaghan. Probabilistic Inference on Noisy Time Series (PINTS). *Journal of Open Research Software*, 7:1–6, 2019.

[134] Motohide Nishio and Aisaku Arakawa. Performance of Hamiltonian Monte Carlo and No-U-Turn Sampler for estimating genetic parameters and breeding values. *Genetics Selection Evolution*, 51(1):1–12, 2019.

[135] Marcelo A. da Silva, Eduardo S.B. de Oliveira, Alina A. von Davier, and Jorge L. Bazán. Estimating the DINA model parameters using the No-U-Turn Sampler. *Biometrical Journal*, 60(2):352–368, 2018.

[136] Lloyd N Trefethen. *Spectral methods in MATLAB*. Software, Environments, and Tools. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2000.

[137] Adrien Bizeray. *State and parameter estimation of physics-based lithium-ion battery models*. PhD thesis, Oxford University, 2016.

[138] Charles W Clenshaw and Alan R Curtis. A method for numerical integration on an automatic computer. *Numerische Mathematik*, 2(1):197–205, 1960.

[139] John Newman. FORTRAN Programs for the Simulation of Electrochemical Systems: Dualfoil, 2014.

[140] Scott J Moura. Doyle-Fuller-Newman Electrochemical Battery Model. Matlab-based programs, 2016.

[141] Patrick Kofod Mogensen and Asbjørn Nilsen Riseth. Optim: A mathematical optimization package for Julia. *Journal of Open Source Software*, 3(24):615, 2018.

[142] J Revels, M Lubin, and T Papamarkou. Forward-Mode Automatic Differentiation in Julia. *arXiv:1607.07892 [cs.MS]*, 2016.

[143] Sergey Oladyshkin and Wolfgang Nowak. The connection between Bayesian inference and information theory for model selection, information gain and experimental design. *Entropy*, 21(11), 2019.

[144] Charles J. Geyer. Practical markov chain monte carlo. *Statistical Science*, 7(4):473–483, 1992.

[145] Andrew Gelman and Donald B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Sciences*, 7(4):457–511, 1992.

[146] Karen E. Thomas and John Newman. Heats of mixing and of entropy in porous insertion electrodes. *Journal of Power Sources*, 119-121:844–849, 2003.

[147] V. J. Ovejas and A. Cuadras. Effects of cycling on lithium-ion battery hysteresis and overvoltage. *Scientific Reports*, 9(1):1–9, 2019.

[148] Carl Edward Rasmussen and Christopher K. I Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT, Cambridge, Mass. ; London, 2006.

[149] Christopher K.I. Williams and David Barber. Bayesian classification with gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351, 1998.

[150] F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, M Blondel, P Prettenhofer, R Weiss, V Dubourg, J Vanderplas, A Passos, D Cournapeau, M Brucher, M Perrot, and E Duchesnay. Scikit-learn: Machine Learning in {P}ython. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[151] Jarno Vanhatalo, Pasi Jylanki, and Aki Vehtari. Gaussian process regression with Student-t likelihood. *Advances in Neural Information Processing Systems 22 - Proceedings of the 2009 Conference*, pages 1910–1918, 2009.

[152] Håvard Rue, Sara Martino, and Nicolas Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 71(2):319–392, 2009.

[153] Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

[154] Arno Solin and Simo Särkkä. Hilbert space methods for reduced-rank Gaussian process regression. *Statistics and Computing*, 2019.

[155] Yunus Saatci. Scalable inference for structured Gaussian process models. *Dissertation*, 2011.

[156] Jacob R. Gardner, Geoff Pleiss, David Bindel, Kilian Q. Weinberger, and Andrew Gordon Wilson. GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration. *CoRR*, abs/1809.1, 2018.

[157] Finn Lindgren and Håvard Rue. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society. Series B*, 73(4):423–498, 2011.

[158] Jouni Hartikainen and Simo Särkkä. Kalman filtering and smoothing solutions to temporal Gaussian process regression models. In *Proceedings of the 2010 IEEE International Workshop on Machine Learning for Signal Processing, MLSP 2010*, pages 379–384, 2010.

[159] Simo Särkkä and Arno Solin. *Applied Stochastic Differential Equations*. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2019.

[160] Rudolph Emil Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[161] Mohinder S. Grewal and Angus P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. John Wiley & Sons Inc., Hoboken, NJ, fourth edi edition, 2014.

[162] R E Kalman and R S Bucy. New Results in Linear Filtering and Prediction Theory. *Journal of Basic Engineering*, 83(1):95–108, 3 1961.

[163] H. E. Rauch, F. Tung, and C. T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, 1965.

[164] P.E. Pascoe and A.H. Anbuky. VRLA battery capacity estimation using soft computing analysis of the coup de fouet region. In *Twenty-Second International Telecommunications Energy Conference*, pages 589–596, 2002.

[165] Moritz Huck, Julia Badeda, and Dirk Uwe Sauer. Modeling the crystal distribution of lead-sulfate in lead-acid batteries with 3D spatial resolution. In *Journal of Power Sources*, 2015.

[166] Dawn M. Bernardi* and Michael K. Carpenter. A Mathematical Model of the Oxygen-Recombination Lead-Acid Cell. *Journal of The Electrochemical Society*, 142(8):2631–2642, 1995.

[167] John Newman. Simulation of Recombinant Lead-Acid Batteries. *Journal of The Electrochemical Society*, 144(9):3081, 1997.

[168] W.B. Gu, G.Q. Wang, and C.Y. Wang. Modeling the overcharge process of VRLA batteries. *Journal of Power Sources*, 108(1-2):174–184, 6 2002.

[169] M. Thele, J. Schiffer, E. Karden, E. Surewaard, and D.U. Sauer. Modeling of the charge acceptance of lead-acid batteries. *Journal of Power Sources*, 168(1):31–39, 5 2007.

[170] D. A. J. Rand. *Valve-regulated lead-acid batteries*. Elsevier, 2004.

[171] A. F. Hollenkamp. When is capacity loss in lead/acid batteries 'premature'? *Journal of Power Sources*, 59(1-2):87–98, 3 1996.

[172] A. Winsel, E. Voss, and U. Hullmeine. The aggregate-of-spheres ('Kugelhaufen') model of the PbO2/PbSO4 Electrode. *Journal of Power Sources*, 30(1-4):209–226, 3 1990.

[173] Henry A. Catherino, Fred F. Feres, and Francisco Trinidad. Sulfation in lead-acid batteries. *Journal of Power Sources*, 129(1):113–120, 4 2004.

[174] R E Carlson and F N Fritsch. Monotone piecewise cubic interpolation. *SIAM J Numer. Anal.*, 17(2):238–246, 1980.

[175] Hans Bode. *Lead-acid batteries*. Wiley, 1977.

[176] Julia Schiffer, Dirk Uwe Sauer, Henrik Bindner, Tom Cronin, Per Lundsager, and Rudi Kaiser. Model prediction for ranking lead-acid batteries according to expected lifetime in renewable energy systems and autonomous power-supply systems. *Journal of Power Sources*, 168(1 SPEC. ISS.):66–78, 2007.

[177] M. García-Plaza, J. Eloy-García Carrasco, A. Peña-Asensio, J. Alonso-Martínez, and S. Arnaltes Gómez. Hysteresis effect influence on electrochemical battery modeling. *Electric Power Systems Research*, 152:27–35, 2017.

[178] S. Särkkä, A. Solin, and J. Hartikainen. Sparse Spatio-temporal Gaussian Processes with General Likelihoods. In *Proceedings of the 21th international conference on Artificial neural networks - Volume Part I*, pages 193–200, 2011.

[179] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

[180] Calum Strange, Shawn Li, Richard Gilchrist, and Gonçalo Dos Reis. Elbows of internal resistance rise curves in li-ion cells. *Energies*, 14(4):1–15, 2021.

[181] Jacqueline S. Edge, Simon O'Kane, Ryan Prosser, Niall D. Kirkaldy, Anisha N. Patel, Alastair Hales, Abir Ghosh, Weilong Ai, Jingyi Chen, Jiang Yang, Shen Li, Mei Chin Pang, Laura Bravo Diaz, Anna Tomaszewska, M. Waseem Marzook, Karthik N. Radhakrishnan, Huizhi Wang, Yatish Patel, Billy Wu, and Gregory J. Offer. Lithium ion battery degradation: what you need to know. *Physical Chemistry Chemical Physics*, 23(14):8200–8221, 2021.

[182] Jorn M. Reniers, Grietus Mulder, and David A. Howey. Unlocking extra value from grid batteries using advanced models. *Journal of Power Sources*, 487(December 2020):229355, 2021.

[183] Agathe Girard, Carl Edward Rasmussen, Joaquin Quiñonero Candela, and Roderick Murray-Smith. Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting. *Advances in Neural Information Processing Systems*, 15(December 2002):9–14, 2003.

[184] J Quinonero-Candela, Agathe Girard, and CE Rasmussen. Prediction at an Uncertain Input for Gaussian Processes and Relevance Vector Machines Application to Multiple-Step Ahead Time-Series Forecasting, 2003.

[185] Gregory L. Plett. *Battery Management Systems, Volume II: Equivalent-Circuit Methods*. Artech House Publishers, 2015.

[186] Marco Steinhardt, Elisabeth Irene Gillich, Alexander Rheinfeld, Ludwig Kraft, Markus Spielbauer, Oliver Bohlen, and Andreas Jossen. Low-effort determination of heat capacity and thermal conductivity for cylindrical 18650 and 21700 lithium-ion cells. *Journal of Energy Storage*, 42(August):103065, 2021.

[187] Michael Osborne. *Bayesian Gaussian Processes for Sequential Prediction, Optimisation and Quadrature*. PhD thesis, Oxford, 2010.

[188] Mark N Gibbs. *Bayesian Gaussian Processes for Regression and Classification*. PhD thesis, Cambridge, 1997.

[189] Nicholas Assimakis, Maria Adam, and Anargyros Douladiris. Information Filter and Kalman Filter Comparison : Selection of the Faster Filter. *International Journal of Information Engineering*, 2(1):1–5, 2012.

[190] Howie N. Chu, Sun Ung Kim, Saeed Khaleghi Rahimian, Jason B. Siegel, and Charles W. Monroe. Parameterization of prismatic lithium–iron–phosphate cells through a streamlined thermal/electrochemical model. *Journal of Power Sources*, 453(December 2019):227787, 2020.

[191] Qiaoyan Chen, Jiuchun Jiang, Haijun Ruan, and Caiping Zhang. Simply designed and universal sliding mode observer for the SOC estimation of lithium-ion batteries. *IET Power Electronics*, 10(6):697–705, 2017.

[192] Languang Lu, Xuebing Han, Jianqiu Li, Jianfeng Hua, and Minggao Ouyang. A review on the key issues for lithium-ion battery management in electric vehicles. *Journal of Power Sources*, 226:272–288, 2013.

[193] Niall Kirkaldy, Mohammad Amin Samieian, Gregory J Offer, Monica Marinescu, and Yatish Patel. Lithium-Ion Battery Degradation: Measuring Rapid Loss of Active Silicon in Silicon-Graphite Composite Electrodes. *ChemRxiv*, 2022.

[194] Weilong Ai, Niall Kirkaldy, Yang Jiang, Gregory Offer, Huizhi Wang, and Billy Wu. A composite electrode model for lithium-ion batteries with silicon/graphite negative electrodes. *Journal of Power Sources*, 527(December 2021):231142, 2022.

[195] Jie Lin, Howie N. Chu, David A. Howey, and Charles W. Monroe. Multiscale coupling of surface temperature with solid diffusion in large lithium-ion pouch cells. *Communications Engineering*, 1(1), 2022.

[196] John S. Newman and Charles W. Tobias. Theoretical Analysis of Current Distribution in Porous Electrodes. *Journal of The Electrochemical Society*, 109(12):1183, 1962.

[197] John Skilling. Nested sampling for general Bayesian computation. *Bayesian Analysis*, 1(4):833–860, 2006.

[198] T.R. Crompton. *Battery reference book*. Butterworth-Heinemann, second edition, 1995.

[199] Aaron Meurer, Christopher P Smith, Mateusz Paprocki, Ondrej Čertík, Sergey B Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E Granger, Richard P Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J Curry, Andy R Terrel, Štěpán Roučka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. SymPy: symbolic computing in Python. *PeerJ Computer Science*, 3:e103, 1 2017.

Appendix A

Trace plots for NUTS SPMe parameter estimates

The NUTS algorithm used in Chapter 2 creates a Markov chain in parameter space, where the stationary state is used to estimate the parameter posterior. To check for convergence of the chain in each of the estimation cases, the $\hat{R}$ statistic was calculated. This statistic is reflected in the similarity in the different posterior distributions generated by starting the sampling at different points in parameter space. These are shown for each of the estimation cases in Figs. A.1-A.3. Convergence in general is such that $\hat{R} < 1.01$, with the notable exception of the wide excursion case in Fig. A.3(d).

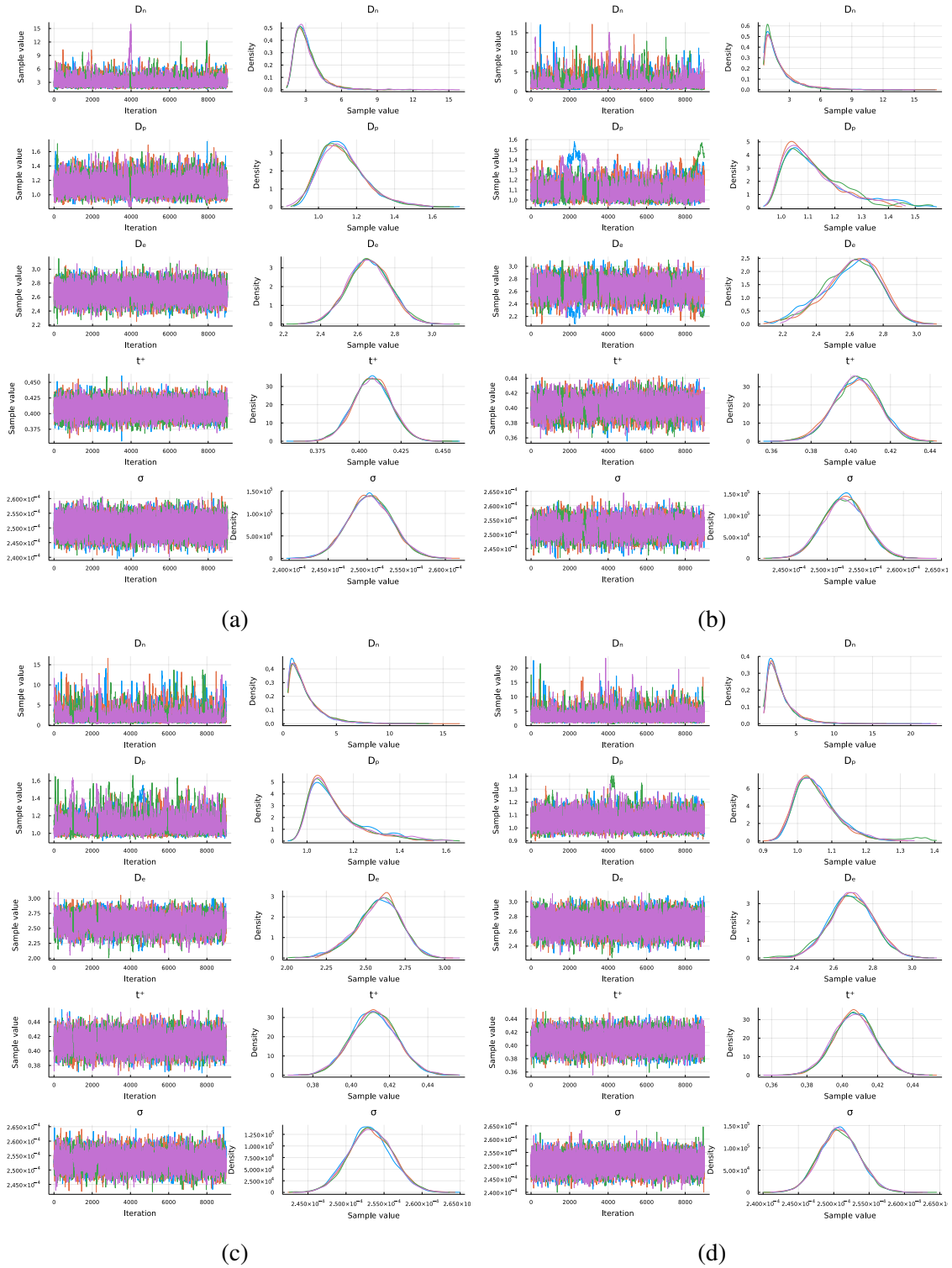


Fig. A.1 NUTS traces for SPMe parameters cases 1 - 4

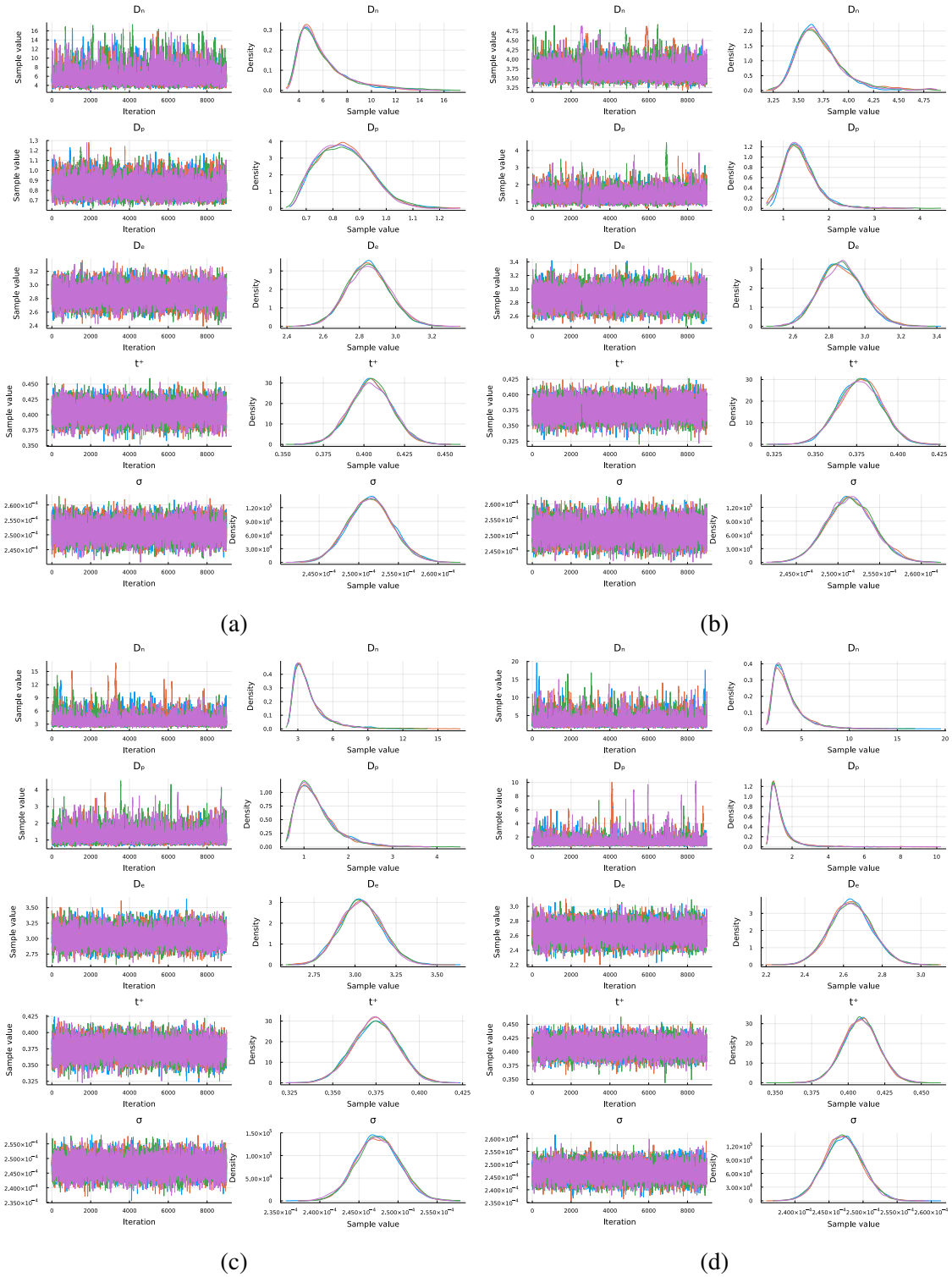


Fig. A.2 NUTS traces for SPMe parameters cases 5 - 8

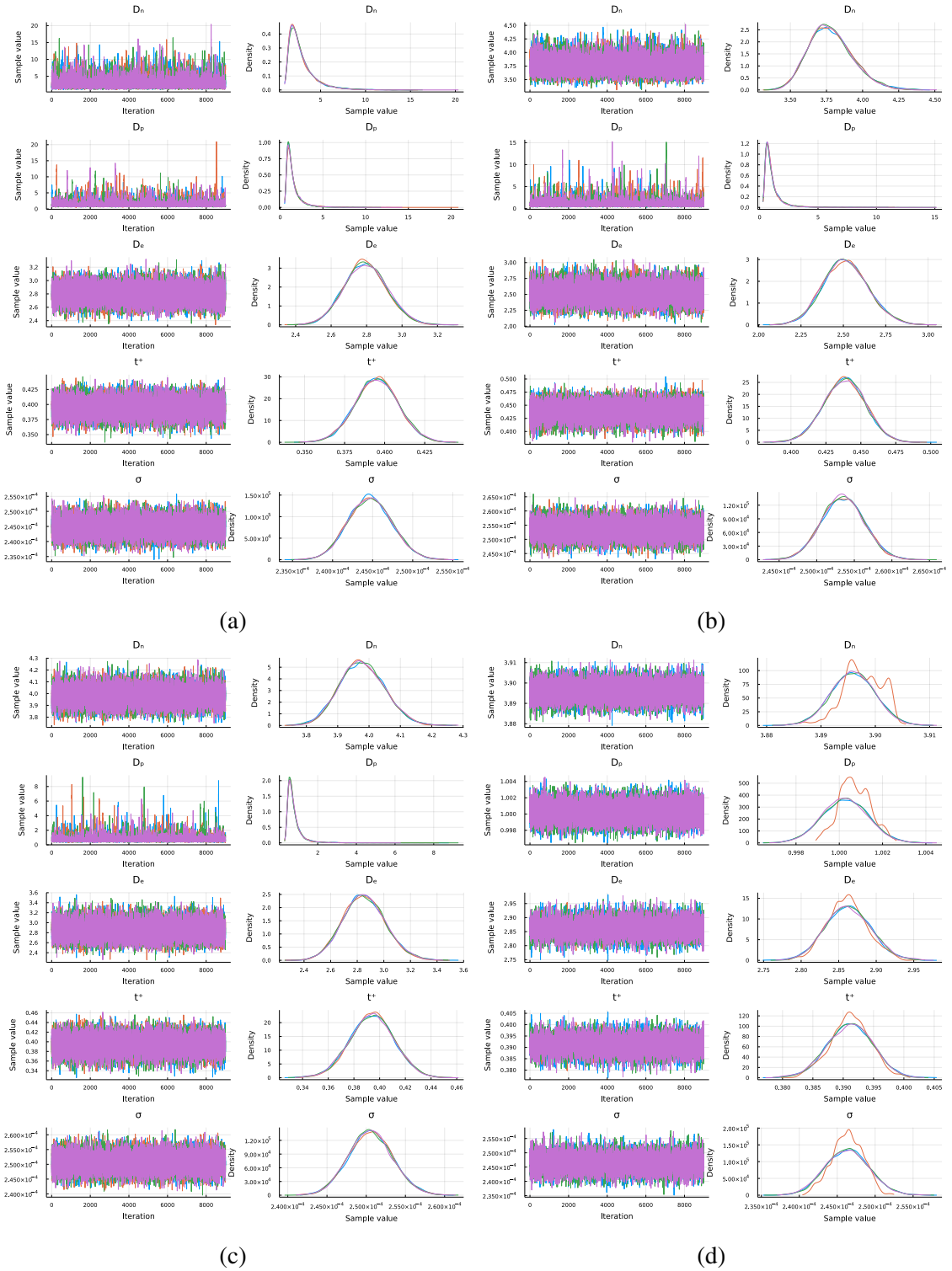


Fig. A.3 NUTS traces for SPMe parameters cases 9 - 11 and the wide excursion case.

Appendix B

Supplementary information for VRLA analysis

B.1 Parameterisation of OCV curve and electrolyte volume calculation for BBOXX VRLA battery

For the Coulomb counting approach for state of charge estimation, an estimate of electrolyte volume for the batteries in the dataset was required. This was obtained by a least squares fit of the OCV curve for lead acid cells from Bode [175] against experimentally measured lab OCV data. First, a GITT procedure was carried out with twenty 1 hour C/20 discharge segments, each with a 2 hour rest in-between them, at 25 °C, to retrieve pointwise estimates of the open circuit voltage as a function of capacity (see Fig. B.1).

To determine the electrolyte volume, the change in electrolyte concentration over discharge is given by

$$\Delta c_{HSO_4^-} = \frac{I\Delta t}{V_{\text{elec}}F},$$

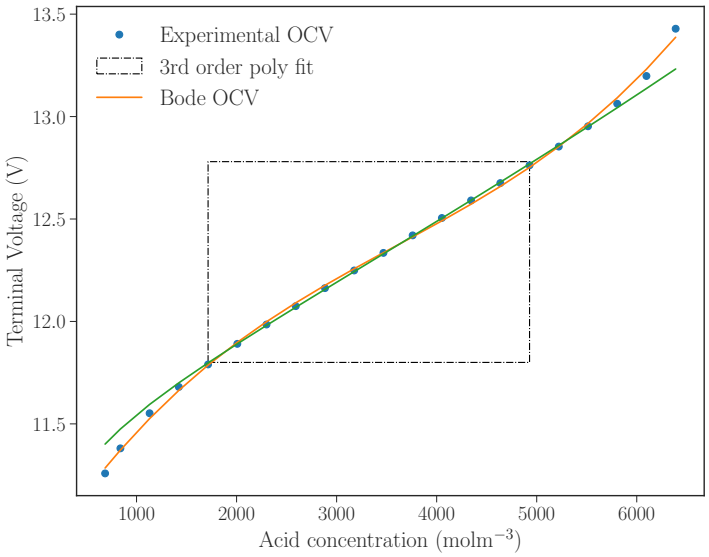


Fig. B.1 Experimentally determined point-wise estimates of OCV determined by GITT at 25 °C and a 3rd order polynomial fit vs. lead-acid OCV of Bode [175]. We matched the gradients by least squares over a range of acid concentrations in the range shown (central box).

where V_{elec} is the electrolyte volume and F is Faraday's constant. Bode expresses the OCV of a single cell as a function of acid molality m,

$$V_{\text{cell}}(m) = 1.922 + 0.148\log_{10}(m) + 0.064\log_{10}^2(m) + 0.074\log_{10}^3(m) + 0.034\log_{10}^4(m), \tag{B.1}$$

so we made the simple conversion from molality to molarity [48],

$$m = \frac{V_{\text{w}} c_{HSO_4^-}}{(1 - c_{HSO_4^-} V_{\text{e}}) M_{\text{w}}},$$

where $c_{HSO_4^-}$ is acid molarity, V_{w}, V_{e} and M_{w} the partial molar volumes of water and acid, and the molar mass of water respectively. (Note that 12 V lead-acid batteries actually contain six cells internally in series, V_{cell} is the voltage of one of these, so $V_0 = 6V_{\text{cell}}$.) Then the experimentally measured OCV curve (Fig. B.1), which was originally a function of Coulombs, was expressed as a function of acid concentration instead. To find the best-fit value of V_{elec}, first the Bode curve at a point in the middle on the range was inverted (illustrated in Fig. B.1), then

a least squares fit was applied over the range shown to match the gradient, which is inversely proportional to V_{elec}. The optimal value for V_{elec} was found to be 143 cm^3.

For further use in modelling, the experimental OCV obtained through GITT using was parameterised using a cubic polynomial as a function of estimated acid molarity. As the measured temperature range observed in the dataset was approximately 10 K and the sensitivity of the OCV to temperature is of the order of 0.1 mV K^{-1}/cell [198], its temperature dependency was ignored.

B.2 Hyperparameter gradients for VRLA internal resistance estimation

Deriving the recursive method to calculate the Jacobian vector $\partial\phi(\theta)/\partial\theta$ for improved stability and speed of GP hyperparameter optimization is from work by Mbalawata et al. [69]. However, the authors did not consider the dependency of the observation matrix $\mathbf{H}_\text{t}$ on hyperparameters, which is added here. From Section 3.2, the energy function may be differentiated with respect to the GP hyperparameters, giving

$$\begin{aligned}\frac{\partial\phi}{\partial\theta_\text{h}} = &-\frac{\partial\log p(\theta_\text{h})}{\partial\theta_\text{h}} + \frac{1}{2}\sum_n \text{Tr}\left(\mathbf{S}_\text{t}^{-1}\frac{\partial\mathbf{S}_\text{t}}{\partial\theta_\text{h}}\right)\\ &-\frac{1}{2}\sum_n\left(\mathbf{H}_\text{t}\frac{\partial\mathbf{x}_\text{t}^-}{\partial\theta_\text{h}} + \frac{\partial\mathbf{H}_\text{t}}{\partial\theta_\text{h}}\mathbf{x}_\text{t}^-\right)^\text{T}\mathbf{S}_\text{t}^{-1}\mathbf{e}_\text{t}\\ &-\frac{1}{2}\sum_n\mathbf{e}_\text{t}^\text{T}\mathbf{S}_\text{t}^{-1}\frac{\partial\mathbf{S}_\text{t}}{\partial\theta_\text{h}}\mathbf{S}_\text{t}^{-1}\\ &-\frac{1}{2}\sum_n\mathbf{e}_\text{t}^\text{T}\mathbf{S}_\text{t}^{-1}\left(\mathbf{H}_\text{t}\frac{\partial\mathbf{x}_\text{t}^-}{\partial\theta_\text{h}} + \frac{\partial\mathbf{H}_\text{t}}{\partial\theta_\text{h}}\mathbf{x}_\text{t}^-\right),\end{aligned} \tag{B.2}$$

where Tr denotes the matrix trace and

$$\frac{\partial\mathbf{S}_\text{t}}{\partial\theta_\text{h}} = \frac{\partial\mathbf{H}_\text{t}}{\partial\theta_\text{h}}\mathbf{P}_\text{t}^-\mathbf{H}_\text{t}^\text{T} + \mathbf{H}_\text{t}\frac{\partial\mathbf{P}_\text{t}^-}{\partial\theta_\text{h}}\mathbf{H}_\text{t}^\text{T} + \mathbf{H}_\text{t}\mathbf{P}_\text{t}^-\left(\frac{\partial\mathbf{H}_\text{t}}{\partial\theta_\text{h}}\right)^\text{T} + \frac{\partial\sigma_{\text{n,t}}^2}{\partial\theta_\text{h}}. \tag{B.3}$$

As all the matrices vary over charging segments, the evaluation of the summation in B.2 is done as a part of the KF recursion. In particular, the partials $\partial \mathbf{P}_t^- / \partial \theta_h$ and $\partial \mathbf{x}_t^- / \partial \theta_h$ have to be propagated along side the states $\mathbf{x}_t$ and covariances $\mathbf{P}_t^-$. The joint propagation of the partials and the states, in continuous time, is then given by the system of equations [69]

$$\frac{\mathrm{d}\mathbf{x}_t^-}{\mathrm{d}t} = \mathbf{F}\mathbf{x}_t^- \tag{B.4a}$$

$$\frac{\mathrm{d}\mathbf{z}_t^-}{\mathrm{d}t} = \mathbf{G}\mathbf{x}_t^- + \mathbf{F}\mathbf{z}_t^- \tag{B.4b}$$

$$\frac{\mathrm{d}\mathbf{P}_t^-}{\mathrm{d}t} = \mathbf{F}\mathbf{P}_t^- + \mathbf{P}_t^-\mathbf{F}^\mathrm{T} + \Sigma \tag{B.4c}$$

$$\frac{\mathrm{d}\mathbf{B}_t^-}{\mathrm{d}t} = \mathbf{G}\mathbf{P}_t^- + \mathbf{P}_t^-\mathbf{G}^\mathrm{T} + \mathbf{F}\mathbf{B}_t^- + \mathbf{B}_t^-\mathbf{F}^\mathrm{T} + W, \tag{B.4d}$$

where $\mathbf{z}_t^- = \partial \mathbf{x}_t^- / \partial \theta_h$, $\mathbf{B}_t^- = \partial \mathbf{P}_t^- / \partial \theta_h$, $\mathbf{G} = \partial \mathbf{F} / \partial \theta_h$, $\mathbf{W} = \partial \Sigma / \partial \theta_h$. Σ is equal to $\mathbf{L}\mathbf{q}\mathbf{L}^\mathrm{T}$ in the main text, where the spectral density $\mathbf{q}$ is a function of θ_h. For notational simplicity the dependency of $\mathbf{F}$ and Σ on θ_h is implicit. The first pair of equations are easily solved to yield give the discrete time updates, as

$$\begin{bmatrix} \mathbf{x}_{t+1}^- \\ \mathbf{z}_{t+1}^- \end{bmatrix} = \exp\left(\begin{bmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{G} & \mathbf{F} \end{bmatrix} \Delta t \right) \begin{bmatrix} \mathbf{x}_t^+ \\ \mathbf{z}_t^+ \end{bmatrix}, \tag{B.5}$$

where Δt is the time increment (in between charging segments in this case). The third and fourth equations are more complicated, and matrix fraction decomposition [69, 161] is used to solve the system, which gives

$$\begin{bmatrix} \mathbf{P}_{t+1}^- \\ \vdots \\ \mathbf{B}_{t+1}^- \\ \vdots \end{bmatrix} = \exp\left(\begin{bmatrix} \mathbf{F} & \Sigma & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{F}^\mathrm{T} & \mathbf{0} & \mathbf{0} \\ \mathbf{G} & \mathbf{W} & \mathbf{F} & \Sigma \\ \mathbf{0} & -\mathbf{G}^\mathrm{T} & \mathbf{0} & -\mathbf{F}^\mathrm{T} \end{bmatrix} \Delta t \right) \begin{bmatrix} \mathbf{P}_t^+ \\ \mathbf{I} \\ \mathbf{B}_t^+ \\ \mathbf{0} \end{bmatrix}. \tag{B.6}$$

Following the propagation, the update equations for the partial derivatives are then given by

$$\frac{\partial \mathbf{K}}{\partial \theta_{\mathrm{h}}} = \mathbf{B}_{\mathrm{t}}^{-}\mathbf{H}_{\mathrm{t}}^{\mathrm{T}}\mathbf{S}_{\mathrm{t}}^{-1} - \mathbf{P}_{\mathrm{t}}^{-}\mathbf{H}_{\mathrm{t}}^{\mathrm{T}}\mathbf{S}^{-1}\frac{\partial \mathbf{S}_{\mathrm{t}}}{\partial \theta_{\mathrm{h}}}\mathbf{S}^{-1} + \mathbf{P}_{\mathrm{t}}^{-}\frac{\partial \mathbf{H}_{\mathrm{t}}}{\partial \theta_{\mathrm{h}}}\mathbf{S}_{\mathrm{t}}^{-1} \tag{B.7a}$$

$$\mathbf{z}_{\mathrm{t}}^{+} = \mathbf{z}_{\mathrm{t}}^{-} + \frac{\partial \mathbf{K}}{\partial \theta_{\mathrm{h}}}(\mathbf{y}_{\mathrm{t}} - \mathbf{H}_{\mathrm{t}}\mathbf{x}_{\mathrm{t}}^{-}) - \mathbf{K}_{\mathrm{t}}\mathbf{H}_{\mathrm{t}}\mathbf{z}_{\mathrm{t}}^{-} - \mathbf{K}_{\mathrm{t}}\frac{\partial \mathbf{H}_{\mathrm{t}}}{\partial \theta_{\mathrm{h}}}\mathbf{x}_{\mathrm{t}}^{-} \tag{B.7b}$$

$$\mathbf{B}_{\mathrm{t}}^{+} = \mathbf{B}_{\mathrm{t}}^{-} - \frac{\partial \mathbf{K}}{\partial \theta_{\mathrm{h}}}\mathbf{S}_{\mathrm{t}}\mathbf{K}_{\mathrm{t}}^{\mathrm{T}} - \mathbf{K}_{\mathrm{t}}\frac{\partial \mathbf{S}_{\mathrm{t}}}{\partial \theta_{\mathrm{h}}}\mathbf{K}_{\mathrm{t}}^{\mathrm{T}} - \mathbf{K}_{\mathrm{t}}\mathbf{S}_{\mathrm{t}}\left(\frac{\partial \mathbf{K}}{\partial \theta_{\mathrm{h}}}\right)^{\mathrm{T}}, \tag{B.7c}$$

which completes the recursion. The propagation and update of the partial derivatives have to be done separately for each hyperparameter, meaning that (B.4)-(B.7) have to be evaluated for each hyperparameter over the forward recursion. A crucial factor determining the computational effort required to perform the recursion is the evaluation of the matrix exponential (B.6), which in the case of the additive kernel function (3.31) involves a square matrix of dimensionality 88×88. However, the matrix inside the exponential has a special structure as the state dynamic transition matrix $\mathbf{F}$ is a block diagonal combination of a nilpotent matrix and a matrix of zeros, and matrices $\mathbf{G}$,$\mathbf{W}$ are sparse. Therefore, there exists a closed form for the matrix exponential for all θ_{h} which avoids having to use the standard Taylor expansion to evaluate the matrix exponential. These explicit expressions were found using the `SymPy` package [199] in the `python` programming language. The second factor determining the speed of the analytical calculation vs. using numerical methods, such as finite differences, comes from the evaluation of the inverse of the innovation covariance matrix $\mathbf{S}_{\mathrm{t}}$ at every time step, which is computationally the most demanding part of the full KF recursion (and is required even if gradients were not calculated), as it has average dimensionality of 150×150—in the analytical method, this only has to be done once per time step to evaluate the entire Jacobian $\partial\phi/\partial\theta_{\mathrm{h}}$. As a result, the analytical estimates are approximately 30% faster and offer extra stability compared to numerical methods.

B.3 VRLA Classifier performance

The breakdown of true positives, true negatives, false positives and false negatives in each of the cross-validation test cases is given in table B.1. This is the breakdown in the 'master' training/test case, where the full 52/48 split of healthy/failed batteries is used.

Days before	Classifier input	TP	FP	FN	TN	Balanced Accuracy (%)
0	R, $\frac{\partial R}{\partial t}$	330	78	161	458	76
0	R, $\frac{\partial R}{\partial t}$, stress factors	389	78	102	458	82
0	Benchmark	244	95	247	441	66
0	Stress factors only	314	119	177	417	71
14	R, $\frac{\partial R}{\partial t}$	299	85	192	451	73
14	R, $\frac{\partial R}{\partial t}$, stress factors	378	95	113	441	80
14	Benchmark	203	91	288	445	62
14	Stress factors only	331	142	160	394	70
28	Stress factors only	349	161	142	375	71
28	Benchmark	175	123	316	413	56
28	R, $\frac{\partial R}{\partial t}$, stress factors	373	124	118	412	76
28	R, $\frac{\partial R}{\partial t}$	264	110	227	426	67
42	R, $\frac{\partial R}{\partial t}$	245	116	246	420	64
42	R, $\frac{\partial R}{\partial t}$, stress factors	373	132	118	404	76
42	Benchmark	123	119	368	417	51
42	Stress factors only	358	191	133	345	69
56	R, $\frac{\partial R}{\partial t}$	237	120	254	416	63
56	R, $\frac{\partial R}{\partial t}$, stress factors	363	151	128	385	73
56	Benchmark	107	117	384	419	50
56	Stress factors only	362	221	129	315	66

Table B.1 Total TP, FP, FN and TN count over CV cases in BBOXX classifier.

Appendix C

Supplementary information for joint GP/battery state estimator

C.1 Predictive variances in EKF recursion

The sparse matrices λ in eqns. (4.26), (4.37) arise from accounting for the extra uncertainties in state dynamics and predicted terminal voltage due to GP predictive variance. At each time-step, the GP predictive means and variances for α, β and R_0 are evaluated by eqns. (4.28,4.29), which depend on z_t and/or applied current.

The 'extra variance' from GP interpolation/extrapolation in the battery state priors and terminal voltage predictions are encapsulated in the λ terms in (4.26) and (4.37). They are the result of the discretisation of the GPs over inputs z_t and I_t, which affects the joint probability distributions relating to the state dynamics and output prediction [3, 110]. In the general case,

the conditional distributions are such that

$$\mathbf{x} \sim \mathcal{N}(\mathbf{m}, \mathbf{P}) \tag{C.1a}$$

$$\theta|\mathbf{x} \sim \mathcal{N}(\mathbf{H}\mathbf{x}, \Sigma_{\mathrm{GP}}) \tag{C.1b}$$

$$\mathbf{y}|\mathbf{x}, \theta \sim \mathcal{N}(g(\mathbf{x}, \theta), \Sigma_{\mathrm{n}}), \tag{C.1c}$$

where the battery model parameter $\theta \in \{\alpha, \beta, R_0\}$ is a linear combination of the relevant GP states, which is then used in the state transition and observation functions g, h to yield the predictive prior battery states. With the EKF, g is locally linearised, which gives the joint distributions of $\mathbf{x}$, θ and $\mathbf{y}$ as

$$\begin{bmatrix} \mathbf{x} \\ \theta \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}(\mathbf{m}^*, \mathbf{P}^*), \tag{C.2}$$

where

$$\mathbf{m}^* = \begin{bmatrix} \mathbf{m} \\ \mathbf{C}\mathbf{m} \\ \mathbf{g}(\mathbf{m}, \mathbf{C}\mathbf{m}) \end{bmatrix}, \tag{C.3}$$

and

$$\mathbf{P}^* = \begin{bmatrix} \mathbf{P} & \mathbf{P}\mathbf{C}^\mathbf{T} & \mathbf{P}\mathbf{C}^\mathbf{T}\mathbf{G}^\mathbf{T} \\ \mathbf{C}\mathbf{P} & \mathbf{C}\mathbf{P}\mathbf{C}^\mathbf{T} + \Sigma_{\mathrm{GP}} & [\mathbf{C}\mathbf{P}\mathbf{C}^\mathbf{T} + \Sigma_{\mathrm{GP}}]\mathbf{G}^\mathbf{T} \\ \mathbf{G}\mathbf{C}\mathbf{P}^\mathbf{T} & \mathbf{G}[\mathbf{C}\mathbf{P}\mathbf{C}^\mathbf{T} + \Sigma_{\mathrm{GP}}] & \mathbf{G}[\mathbf{C}\mathbf{P}\mathbf{C}^\mathbf{T} + \Sigma_{\mathrm{GP}}]\mathbf{G}^\mathbf{T} + \Sigma_{\mathrm{n}} \end{bmatrix}, \tag{C.4}$$

where $\mathbf{G}$ is the Jacobian of g. As the model parameters that are discretised over z_{t} and I_{t} include α, β and R_0, Σ_{GP} will be very sparse – it will only have nonzero terms for the state transitions of T and V_1 in (4.26) and the terminal voltage prediction in (4.37). To simplify the calculation, we refactor the bottom right-hand term in (C.4) to obtain the λ terms in (4.26,4.37), where the

Hyperparameter	Lower bound	upper bound
$\sigma_{\mathrm{Q}^{-1}}$	-8.0	0.0
σ_{α}	-3.0	3.0
σ_{β}	-3.0	3.0
σ_{R_0}	-3.0	3.0
$l_{\mathrm{R}_0,\mathrm{z}}$	$\log(\Delta\mathbf{u}_{\mathrm{R}_0,\mathrm{z}})$	3.0
$l_{\mathrm{R}_0,\mathrm{I}}$	$\log(\Delta\mathbf{u}_{\mathrm{R}_0,\mathrm{I}})$	3.0
l_{α}	$\log(\Delta\mathbf{u}_{\alpha})$	3.0
l_{β}	$\log(\Delta\mathbf{u}_{\beta})$	3.0
σ_{V}	-7.0	-1.5
σ_{T}	-7.0	-1.5

Table C.1 Hyperparameter constraints for simulated case in log terms. Δu_{f} denote the distance between discretized points in GP input space f.

entries for each are given by

$$\lambda_{\mathrm{G,t}}[2,2] = \sigma^2_{\mathrm{GP},\beta}\left(\frac{I_{\mathrm{t}}(1-\exp(-\Delta t\mu_{\alpha,\mathrm{t}}))}{\mu_{\alpha,\mathrm{t}}}\right)^2 + \sigma^2_{\mathrm{GP},\alpha}V^2_{1,\mathrm{t}}, \tag{C.5a}$$

$$\lambda_{\mathrm{G,t}}[3,3] = \sigma^2_{\mathrm{GP},\mathrm{R}_0}I_{\mathrm{t}}^4\left(1-\exp\left(-\frac{\Delta t}{R_{\mathrm{c}}C_{\mathrm{c}}}\right)\right)^2, \tag{C.5b}$$

$$\lambda_{\mathrm{H,t}}[1,1] = \sigma^2_{\mathrm{GP},\mathrm{R}_0}I_{\mathrm{t}}^2. \tag{C.5c}$$

C.2 Hyperparameter constraints

For both the simulated and real data cases, hyperparameter optimisation using the BFGS algorithm was performed in logarithmic space, so that the real values of hyperparameters were all constrained to be positive. The constraints for simulated and real data cases are shown in Tables C.1 and C.2, respectively. Note that the main text uses the inverse squared length scale parameter $\gamma_{\mathrm{f}} = 1/2l_{\mathrm{f}}^2$, which is mathematically more convenient. Also, the bounds for simulated and real data are different—in the live case, it was found that reducing the size of the search space for optimal hyperparameters reduced numerical errors.

Hyperparameter	Lower bound	upper bound
$\sigma_{\mathrm{Q}^{-1}}$	-7.0	-5.0
$\sigma_{\alpha,\beta}$	-3.0	5.0
σ_{R_0}	-3.0	5.0
$l_{\mathrm{R}_0,\mathrm{z}}$	$\log(\Delta\mathbf{u}_{\mathrm{R}_0,\mathrm{z}})$	5.0
$l_{\mathrm{R}_0,\mathrm{I}}$	$\log(\Delta\mathbf{u}_{\mathrm{R}_0,\mathrm{I}})$	5.0
$l_{\alpha,\beta}$	$\log(\Delta\mathbf{u}_{\alpha,\beta})$	3.0
σ_{V}	$\log(0.004)$	-3.0
σ_{T}	-5.0	-2.0
$\sigma_{0,\zeta,\mathrm{s}}$	-5.0	-1.0
$\sigma_{1,\zeta,\mathrm{s}}$	-5.0	-1.0
$\sigma_{\zeta,\mathrm{r}}$	-3.0	-1.0
$\gamma_{\zeta,\mathrm{r}}$	-5.0	-3.0

Table C.2 Hyperparameter constraints for real cell data in log terms. Δu_{f} denote the distance between discretized points in GP input space f.

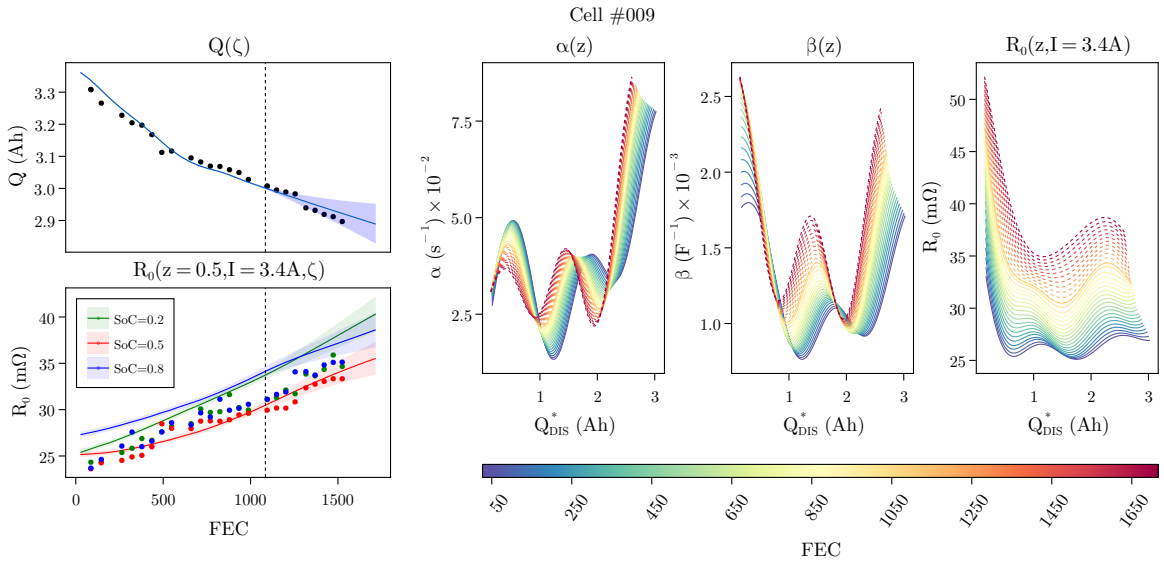
C.3 Results for second cell in dataset

As mentioned in the main text, the results for cell #009 are very similar to cell #015. The cells were aged identically and they shared hyperparameters. The equivalent illustrations for cell #009 are provided here.

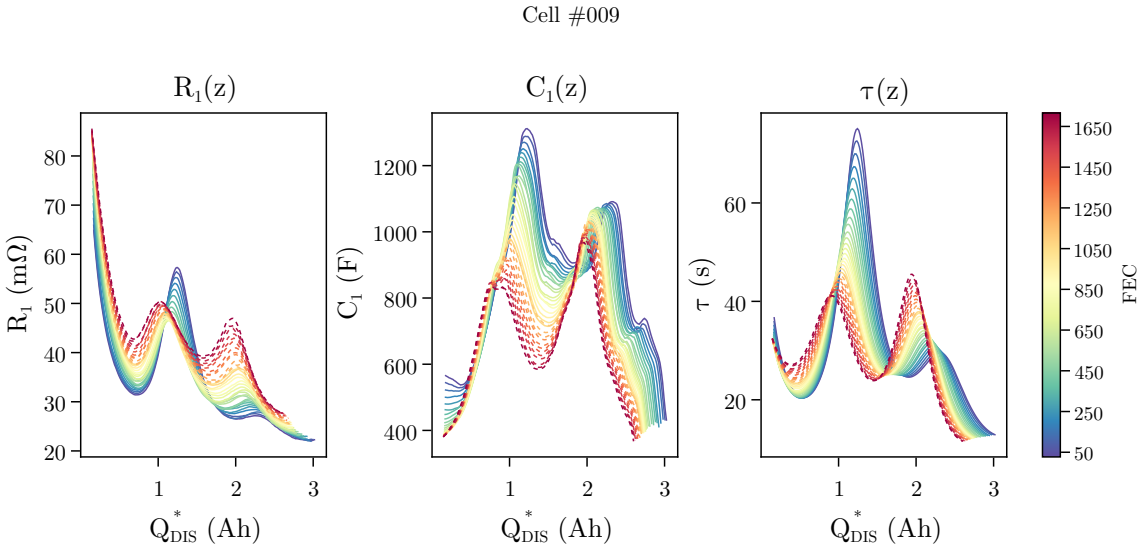
In addition, plots are provided to the overall GP, which include the noise kernel. The results are inline with those in the main text, with wider credible intervals.

C.4 Modelling $\mathbf{R_0}$ with the Gibbs kernel

In the main text, the function $R_0(I,z)$ was modelled using the SE kernel with ARD. However, it is possible that the SE kernel is not the best choice to model the dependence of R_0 on the state of charge z, as internal resistance remains roughly constant over a broad range and there is a sudden increase at very low SOC. This implies a GP length scale that varies over the input. To

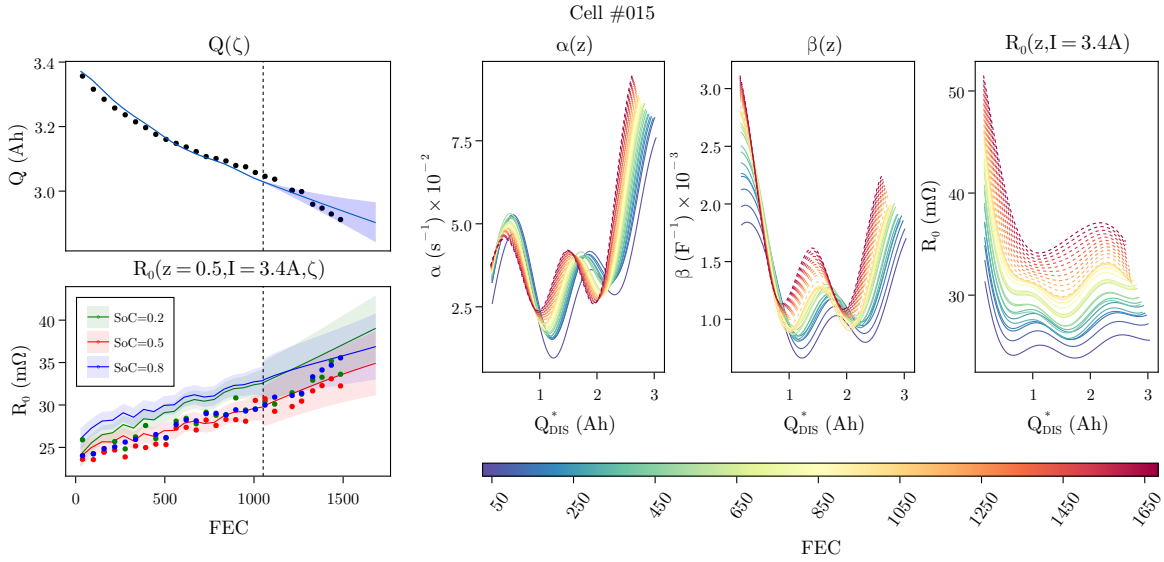


(a)

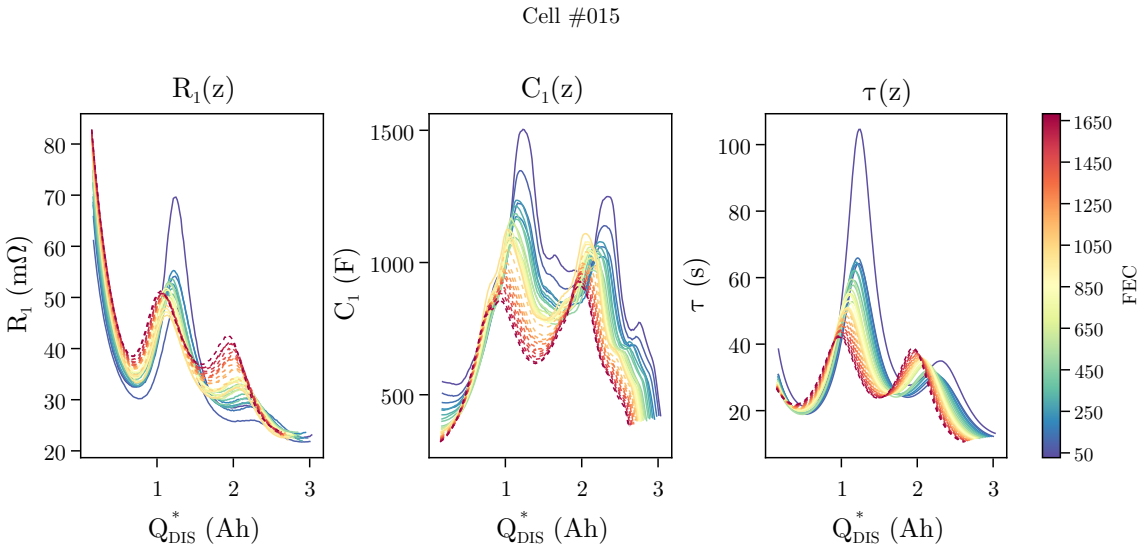


(b)

Fig. C.1 GP projections for cell #009 for the long-term smooth GP, see Fig. 4.6 in the main text.

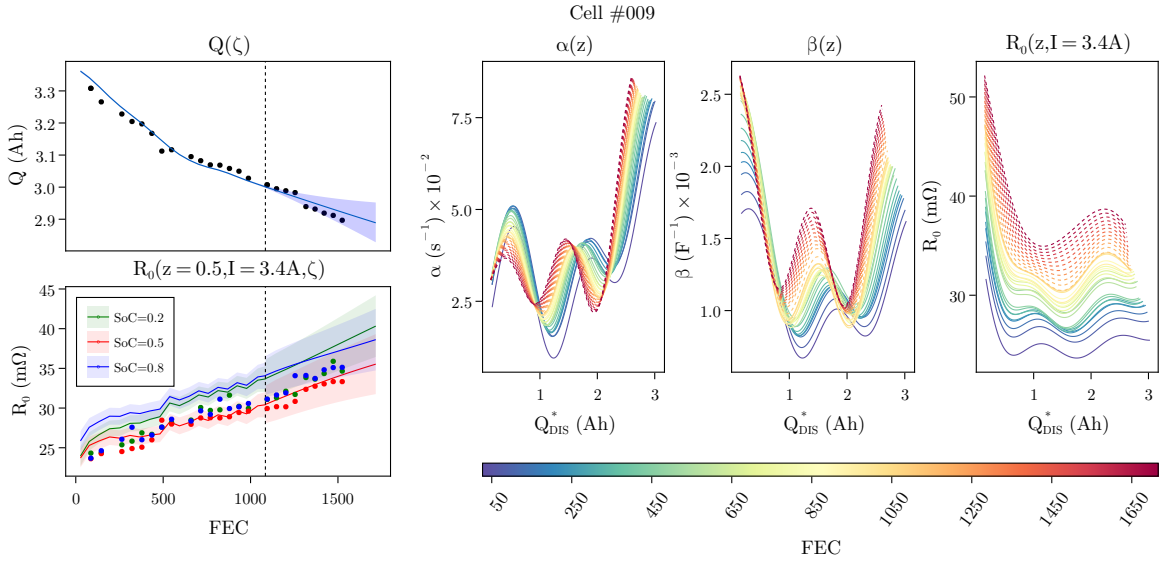


(a)

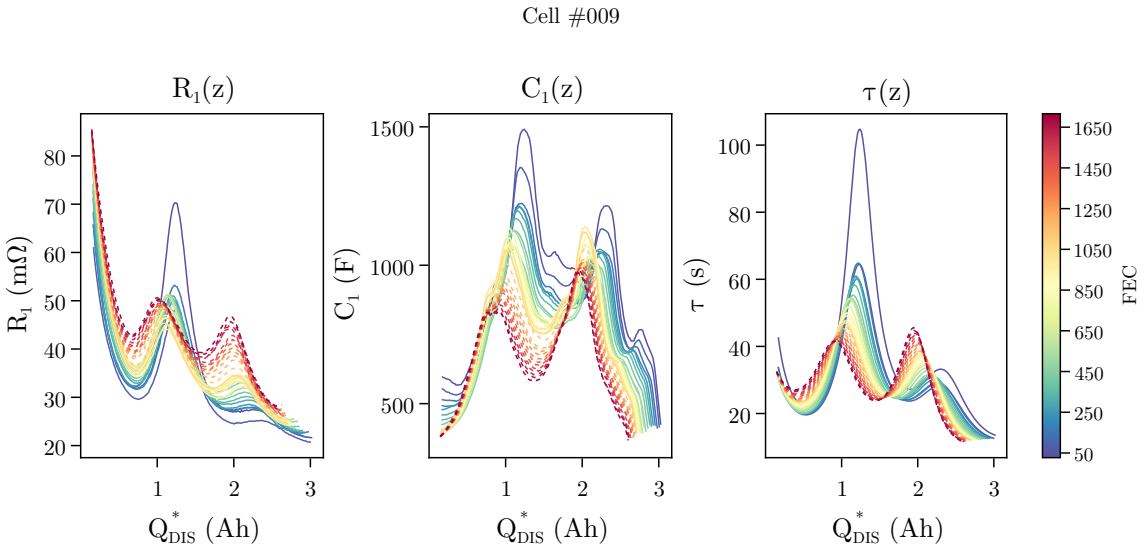


(b)

Fig. C.2 GP projections for cell #015 for the total GP, including the noise kernel, see Fig. 4.6 in the main text.



(a)



(b)

Fig. C.3 GP projections for cell #009 for the total GP, including the noise kernel, see Fig. 4.6 in the main text.

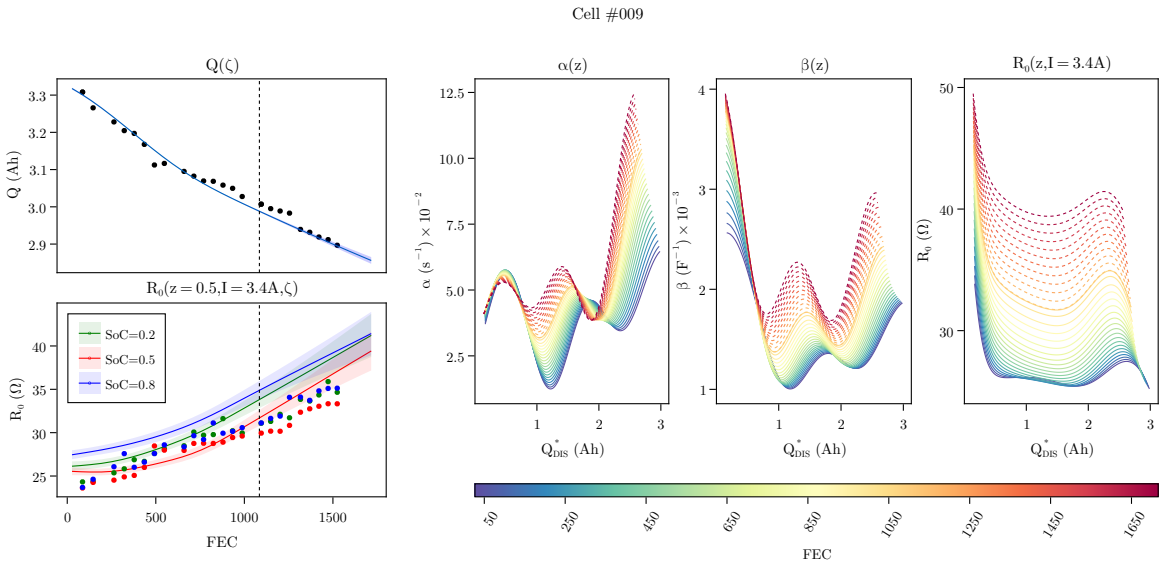
do this, an alternative kernel, the so-called Gibbs kernel [188],

$$k(x,x') = \frac{2l(x)l(x')}{l^2(x)+l^2(x')} \exp\left(\frac{-(x-x')^2}{l^2(x)+l^2(x')}\right), \tag{C.6}$$

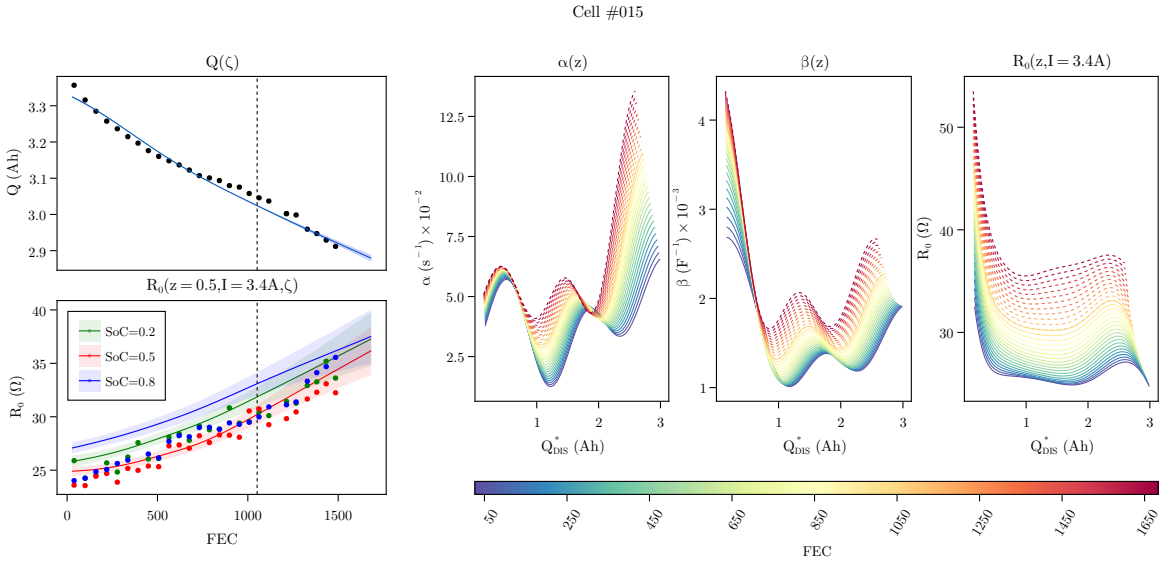
can be used. This is a non-stationary kernel, where the *a priori* defined function $l(.)$ defines the variation of length scale over the input. To compare with the SE kernel results, the joint GP estimator from the main text was also run with the Gibbs kernel modelling $R_0(z)$, keeping everything else the same. The length scale was described by the (scaled) logistic function,

$$l(x) = \frac{1}{1+e^{-10x}} \tag{C.7}$$

The results are shown in Fig. C.4. As is evident, comparing to Fig. 4.6, there is less fluctuation at higher SOC in R_0. However, the results are otherwise the same and the accuracy with respect to the experimental measurements is very similar to the base case.



(a)



(b)

Fig. C.4 Parameter estimation results for cells #009 and #015 using the Gibbs kernel to describe $R_0(z)$.