

Optimal Experimental Design for Large-Scale Inverse Problems via Multi-PDE-constrained Optimization

Andrea Petrocchi^{a,*}, Matthias K. Scharrer^b, Franz Pichler^b, Stefan Volkwein^a

^a*University of Konstanz, Universitaetsstrasse 10, 78464, Konstanz, Germany*

^b*Virtual Vehicle Research GmbH, Inffeldgasse 21a, 8010, Graz, Austria*

Abstract

Accurate parameter dependent electro-chemical numerical models for lithium-ion batteries are essential in industrial application. The exact parameters of each battery cell are unknown and a process of estimation is necessary to infer them. The parameter estimation generates an accurate model able to reproduce real cell data. The field of optimal input/experimental design deals with creating the experimental settings facilitating the estimation problem. Here we apply two different input design algorithms that aim at maximizing the observability of the true, unknown parameters: in the first algorithm, we design the applied current and the starting voltage. This lets the algorithm collect information on different states of charge, but requires long experimental times (60 000 s). In the second algorithm, we generate a continuous current, composed of concatenated optimal intervals. In this case, the experimental time is shorter (7000 s) and numerical experiments with virtual data give an even better accuracy results, but experiments with real battery data reveal that the accuracy could decrease hundredfold. As the design algorithms are built independent of the model, the same results and motivation are applicable to more complex battery cell models and, moreover, to other applications.

Keywords: lithium-ion batteries, parameter estimation, optimal input design

*Corresponding author

Email addresses: andrea.petrocchi@uni-konstanz.de (Andrea Petrocchi), Matthias.Scharrer@v2c2.at (Matthias K. Scharrer), franz.pichler@v2c2.at (Franz Pichler), Stefan.Volkwein@uni-konstanz.de (Stefan Volkwein)

1. Introduction

Mobile appliances and vehicles containing lithium-ion batteries have become ubiquitous over the recent decade. It is thus paramount to model and simulate lithium-ion cells efficiently and accurately. High accuracy in simulation requires the internal states of a cell to be known. These internal states may include a continuum of physical quantities, e.g., potentials, concentrations and temperature, i.e., a virtually infinite number of individual values. Very few quantities may actually be measured non-intrusively, e.g., cell potential and temperature on the outside.

Typical high-fidelity simulation models are based on the equations described by Doyle-Fuller-Newman (DFN) model in Newman and Tiedemann (1993); Newman and Thomas-Alyea (2004). The DFN model is a homogenized system of PDEs, that typically comprise of more than 50 parameters, a vast majority of which cannot be determined experimentally.

This model may be simplified further to a Single Particle Model (SPM) by neglecting spatially resolved effects, e.g., concentration differences among particles of the same electrode. Although it is known to result in less accuracy in general, it is still possible to achieve very high model accuracy by taking the SPM's limitations into account, e.g., currents below 1 C, i.e., the equivalent of a full discharge in more than 1 hour, as described in Santhanagopalan et al. (2006); Ning and Popov (2004).

The parameter estimation for lithium-ion battery models is usually extremely time consuming. One needs very long input currents and optimization time; for example, in Reddy et al. (2019), in order to approximate 44 battery parameters an input current longer than 200 000 s (approximately 55 hours) is used, and even the optimization method lasts for several hours. The estimation process is based on a fitting argument: chosen a certain input current, we observe the output voltage of the battery cell. The parameter estimation is then an optimization problem based on minimizing the discrepancy between the mathematical model and the real experimental response of the battery.

Treating non-convex optimization problems, it is possible to converge to a local minimum, possibly far from the true hidden parameter. For many optimization problems this can depend on the user-defined input, whose choice change drastically the results of the parameter estimation.

In the parameter estimation framework the input current is arbitrarily chosen and, as observed in many works (e.g., López et al. (2016)), the current impacts strongly on the identification of some of the parameters. For example, in Liu et al. (2016) an optimized battery cycling is linked to an improved parameter identifiability and therefore to robust battery health and experiment safety conditions.

The field of optimal design of experiment can help define, modify and improve the battery modeling framework. In the last decades, the literature concerning design of experiment has been extensive; we recommend some work about classic design of experiment in Goodwin and Payne (1977); Pázman (1986); Rafajłowicz (1986); Pukelsheim (2006); Atkinson et al. (2007); Pronzato and Pázman (2013). Different techniques of design of experiment have been used for building new battery models, or manufacturing optimized layouts in order to improve conditions such as ageing, thermal design and charging properties; see, e.g., Franceschini and Macchietto (2008); Román-Ramírez and Marco (2022).

The focus of this work focuses on the design of optimized inputs in the context of parameter estimation; see, e.g., Mehra (1974) for a survey on optimal input design for parameter estimation in dynamic systems. The optimal input design is strictly related to the optimal design of experiment, and the main purpose of such techniques in the framework of estimation is to improve the observability of the system.

It is well known that the many parameters in electrochemical lithium-ion battery models have different magnitude of observability – and hence of identifiability. Some parameters are for example more sensitive to extreme *states of charge*, and some are more sensitive to fast changes of applied current – the so-called *high frequency dynamics*. Standard methods of experimental design analyze the sensitivities of the parameters, namely measuring how the single parameters impact the observable quantities. In Schmidt et al. (2010) the parameters of a modified single-particle model are divided using a grouping algorithm and it is suggested that using experimental design methods it would be sensible to generate targeted data for the different groups.

In Park et al. (2018) an optimized excitation input for a parameter estimation problem is generated from a collection of predefined inputs. On the other hand, works that do not define a-priori a collection of inputs include Pozzi et al. (2018) and Pozzi et al. (2020), where the input design of experiment for parameter identification problem is analyzed for both an isothermal and not single particle models. In particular, in Pozzi et al. (2020) global

sensitivities are analyzed in a more rigorous approach, and a more precise parameter estimation is achieved.

In this work we start with an adaptive algorithm that, starting from some initial guesses for the parameter $\boldsymbol{\mu}^0$ and sequentially generate new input arrays $\mathbf{u}^{[n]}$ at each iteration $n > 0$. Each of these inputs is designed in order to maximize the observability of the current parameter, and with it we generate data and optimize the parameter $\boldsymbol{\mu}^n$; the inputs are chosen in order to improve or facilitate the parameter optimization and the convergence to the unknown underlying parameter $\boldsymbol{\mu}^*$, by collecting different pieces of information. Hence, the result of the input design algorithm is not intended to be a single “optimal” input, but rather a collection of inputs that collects the most information. In a later variation of the input design algorithm, we also try to optimize the experiment resources, such as minimizing the length of the excitation input. In this framework we defined optimized current sub-intervals, with ideas similar to the settings in Pozzi et al. (2018).

To quantify the information of a parameter $\boldsymbol{\mu}$ carried by the current experiment, an information matrix is used, defined by using the so-called sensitivities, which measures how the output is sensitive to change in the model parameters. This information matrix resembles the Fisher information matrix, a common used statistical object and known lower bound to the covariance matrix generated by the parameter estimation. Another interpretation of the Fisher information matrix is mentioned, among others, in Uciński (2005), where it is seen as an approximation of the Hessian of the least-squares problem given by the parameter estimation problem. Hence, improving the well-conditioning of the approximated information matrix can also help in making the cost function more convex around the unknown parameter – especially useful given the general non-convexity of the estimation problem.

In the standard battery cell parameter estimation applied currents are many hours long, in order to observe the battery as long as possible as collect enough information to infer the tens of parameters. Our goal is to create a collection of input current profiles much shorter than these, lasting just a few minutes, to infer 9 of the most observable parameters. One reason to treat the result of the design as a collection of inputs rather than a single “optimal” input is that in such a complex setting, we cannot expect to maximize the observability of all the hidden parameters with just one such short inputs, and thus we hope to capture the different information with different settings – or inputs.

2. Methods

This section has the following structure: we begin in Section 2.1 with the description of the numerical model. Then, we proceed to discuss the input design algorithm in Section 2.2, followed by the discretization of the design optimization and parameter estimation problems in Sections 2.3 and 2.4. In Section 2.5 we propose a modification of the design in order to improve a weakness of the previous framework and cut the experimental time. We consider the analysis of real battery data in Section 2.6 in order to study the quality of the generated optimal design.

2.1. The numerical model

The numerical model used in this work is based on the assumption that an electrode may be fully represented by one spherical symmetric particle – hence it is referred to as SPM. In each particle per electrode $i \in \{A, C\}$, with A and C marking cathode and anode, respectively, the scaled lithium concentration $\xi_i \in [0, 1]$ along the respective radius ($r \in [0, R_i]$) for any time $t \in [0, t_f]$ ($t_f > 0$) is governed by the system of equations

$$\dot{\xi}_i(t, r) - \frac{1}{r^2} \partial_r (r^2 D_i \partial_r \xi_i(t, r)) = 0, \quad (t, r) \in (0, t_f) \times (0, R_i), \quad (1)$$

where $D_i > 0$ is each particle's scaled diffusion constant. The spheres' symmetry is exploited by setting the natural center boundary condition at $r_i = 0$ to $\partial_r \xi_i(t, 0) = 0$. The boundary condition at the particles' outer border R_i is modeled by

$$-D_i \partial_r \xi_i(t, R_i) = \frac{1}{\rho_i c_{m,i}} j_i(\xi_i(t, R_i), u_i(t)), \quad t \in (0, t_f), \quad (2)$$

where $\rho_i c_{m,i}$ is the scaling constant of $\xi_i(t, r)$. The connection between mass transport and the electric domain is modeled in (2) by the Butler-Volmer expression

$$j_i(\xi_i(t, R_i), u_i(t)) = j_i^0(\xi_i(t, R_i)) \sinh\left(\frac{F}{R\vartheta}(u_i(t) - O_i(\xi_i(t, R_i)))\right), \quad (3)$$

where R is the universal gas constant and ϑ is the temperature. As the SPM formulation is expressed mainly by the open-circuit potentials $O_i(\xi_i(t, R_i))$ of the two electrodes, these are represented by a Redlich-Kister sum as derived

in Karthikeyan et al. (2008)

$$O_i(\xi_i(t, R_i)) = U_{0,i} + \frac{R\vartheta}{F} \ln \frac{1 - \xi_i(t, R_i)}{\xi_i(t, R_i)} + \frac{R\vartheta}{F} \sum_{k=0}^{n_i} A_{i,k} \left((2\xi_i(t, R_i) - 1)^{k+1} - \frac{2\xi_i(t, R_i)k(1 - \xi_i(t, R_i))}{(2\xi_i(t, R_i) - 1)^{k-1}} \right), \quad (4)$$

where $U_{0,i}$ is the open-circuit potential offset for each electrode. Throughout the rest of the paper, we fix $U_{0,A} = 0$ and use U_0 as a simplified substitute for $U_{0,C}$. The basic exchange current is modeled by

$$j_i^0(\xi_i(t, R_i)) = \exp \left(\log k_i + \frac{F}{R\vartheta} ((\xi_i(t, R_i) - .5)O_i(\xi_i(t, R_i)) - \int_0^{\xi_i(t, R_i)} O_i(x) dx) \right), \quad t \in (0, t_f), \quad (5)$$

where k_i is the proportional reaction rate factor. For implementation, $\tilde{k}_i := \log k_i$ is used. The electric boundary condition is applied by the current $i_{\text{cell}}(t)$

$$\frac{3m_i}{\rho_i R_i} F j_i(\xi_i(t, R_i), u_i(t)) = -i_{\text{cell}}(t), \quad (6)$$

where F is Faraday's constant. The resulting cell voltage v_{cell} is then defined as

$$u_C(t) + u_A(t) + i_{\text{cell}}(t)R_I = v_{\text{cell}}(t), \quad (7)$$

where R_I is the total inner resistance.

Our work concerns the estimation of nine parameters of this model, selected on the basis of their high impact on the model behavior – since, as mentioned for the cases of Schmidt et al. (2010); Park et al. (2018), different parameters are more or less identifiable.

The collection of considered parameters is depicted in Table 1, together with lower and upper bounds. In particular, the diffusion coefficients D_C, D_A appear in (1) as well as the initial state of charge of the anode ξ_A – since ξ_C can be recovered algebraically from a given cell voltage $v|_{t=0}$. The total inner resistance R_I appears in the voltage equation (7), while the masses m_C, m_A

Parameter	Meaning	Lower bound	Upper bound
D_C	Cathode diffusion coefficient	0.0001	0.01
D_A	Anode diffusion coefficient	0.0001	0.01
ξ_A	Anode initial state of charge	0.007	0.2
R_I	Inner resistance	0.003	0.07
m_C	Cathode active mass	0.01	0.052
m_A	Anode active mass	0.006	0.034
$\tilde{k}_C$	Cathode reaction rate	-20	-1
$\tilde{k}_A$	Anode reaction rate	-23	10
U_0	Cathode 0th RK parameter	3	4

Table 1: Unscaled parameter names and bounds.

are involved in the current equation (6). The reaction rate logarithms $\tilde{k}_C, \tilde{k}_A$ are relevant in the exchange current formula (5) and, finally, the first Redlich-Kister term U_0 is considered, while all the other model parameters are set to experimental acceptable standards.

As these values have different cardinalities, we apply the following scalings:

$$\begin{aligned}
\mu_1 &:= \log_{10} \left(\frac{D_C}{D_C^L} \right), & \mu_2 &:= \log_{10} \left(\frac{D_A}{D_A^L} \right), & \mu_3 &:= \frac{\xi_A}{(\xi_A^L + \xi_A^U)/2}, \\
\mu_4 &:= \frac{R_I}{(R_I^L + R_I^U)/2}, & \mu_5 &:= \frac{m_C}{(m_C^L + m_C^U)/2}, & \mu_6 &:= \frac{m_A}{(m_A^L + m_A^U)/2}, \\
\mu_7 &:= \tilde{k}_C - (\tilde{k}_C^L + \tilde{k}_C^U)/2, & \mu_8 &:= \tilde{k}_A - \tilde{k}_C, & \mu_9 &:= \frac{U_0}{(U_0^L + U_0^U)/2}.
\end{aligned}$$

Finally, we report the scaled parameter bounds in Table 2.

2.2. The input design algorithm

From now on, we use the battery model as a black box system that takes a model parameter vector $\boldsymbol{\mu} \in \mathbb{R}^d$ and a current density function $i : [0, t_f] \rightarrow \mathbb{R}$ input variables, whereas the voltage $v_{\boldsymbol{\mu}, i} : [0, t_f] \rightarrow \mathbb{R}$ is the output variable.

We use an adaptive algorithm that starts from an arbitrary parameter $\boldsymbol{\mu}^0$ and, for $n \geq 1$ (called *design iteration*), the algorithm iterates on three phases:

- In the first phase we generate a new current function $i^{[n]}$.

μ^L	μ^U
0	2
0	2
0.0676328502415459	1.93236714975845
0.0821917808219178	1.91780821917808
0.32258064516129	1.67741935483871
0.3	1.7
-9.5	9.5
-3	11
0.857142857142857	1.14285714285714

Table 2: Bounds for the scaled parameters

- In the second phase we read and store some data $w^{[n]}$ derived by using the current $i^{[n]}$.
- The last phase is the parameter optimization/estimation using previous data $\{w^{[1]}, \dots, w^{[n]}\}$, with which we obtain the parameter μ^n , approximation of the true hidden parameter μ^* .

There are different ways of ending this algorithm. One possible way is to stop the algorithm as soon as the last generated current function is sufficiently close to one of the already computed currents $i^{[k]}$, $k = 1, \dots, n-1$, in our collection. Therefore, we use the stopping criterion

$$\min_{i \in \{i^{[1]}, \dots, i^{[n-1]}\}} \|i^{[n]} - i\|_{L^2(0, t_f)} < \varepsilon \quad (8)$$

with a properly chosen tolerance $\varepsilon > 0$. However, a known limitation is that we have no guarantee of convergence. This fact can potentially generate an infinite number of current inputs. In the cases, where we have used this criterion, we have also set a maximum number of iterations as a safeguard. An alternative termination is to fix *a priori* a determinate number of inputs and forcing the algorithm to generate inputs sufficiently far from each other, by penalizing the closeness between them. We will later describe in detail this forced generation of inputs.

Next, we introduce a so-called symmetric information matrix in order to quantify the uncertainty produced by a certain input i . As in Atkinson et al. (2007), we define the information matrix $M \in \mathbb{R}^{d \times d}$ by using the sensitivity

variables, described in details in the next subsection. Then, the uncertainty given by an information matrix is defined as the inverse by applying the D-optimal design, or D-criterion (see, e.g., Goodwin and Payne (1977); Atkinson et al. (2007)), i.e.,

$$\phi(\mathbf{M}) := -\log(\det(\mathbf{M})),$$

which is equivalent to minimizing the generalized variance of the parameters – as discussed, e.g., in Atkinson et al. (2007). We can also define the uncertainty function by using the eigenvalues of $\mathbf{M}$ – especially convenient when the dimension d is high and evaluating the determinant gets computationally expensive. Indeed, if $\Lambda := \{\lambda_1, \dots, \lambda_d\}$ is the spectrum of $\mathbf{M}$, then

$$\phi(\mathbf{M}) := \log \left(\prod_{\lambda_j \in \Lambda} \frac{1}{\lambda_j} \right).$$

Observe that we can also interpret this criterion as minimization of the volume of the confidence ellipsoid for the parameters.

2.3. Approximation of the uncertainty measure

We fix the time step to $\delta = 0.1$ seconds, so that we have $K = 600$ time steps in an interval of 60s. Time steps $t_k := k\delta$ for $k = 0, \dots, K$ are discretized uniformly, and all current functions will be discretized as step function evaluated on such time steps, so that the current information is compressed into the *input array* $\mathbf{u}$ – or, simply, *input*. Given $n_u \in \mathbb{N}^+$ and $\tau_u := t_f/n_u$, the array

$$\mathbf{u} := [u_1, \dots, u_{n_u}, v_0] \in \mathbb{R}^{n_u+1} \tag{9}$$

corresponds to the current function

$$i_u(t) := \begin{cases} u_1 & \text{for } 0 \leq t < \tau_u, \\ u_2 & \text{for } \tau_u \leq t < 2\tau_u, \\ \vdots & \\ u_{n_u} & \text{for } (n_u - 1)\tau_u \leq t \leq t_f, \end{cases}$$

and the last component, v_0 , is the initial voltage of the battery, which is required by the numerical model. With $v_{\boldsymbol{\mu}, \mathbf{u}} : [0, t_f] \rightarrow \mathbb{R}$ we indicate the output voltage given the parameter $\boldsymbol{\mu}$ and input array $\mathbf{u}$ (hence given current

i_u and initial voltage v_0). We seldom use its discretization $v_{\mu,u}$ evaluated on the time steps $\{t_k\}_{k=0}^K$.

The sensitivity variables (or, simply, *sensitivities*) are usually evaluated by solving the so-called *sensitivity equations*. However, in a black-box model they can only be approximated by using finite differences. For $1 \leq j \leq d$, the j -th discretized sensitivity $s^j = s_{\mu,u}^j \in \mathbb{R}^{K+1}$ is defined as

$$s_{\mu,u}^j := \frac{v_{\mu+\nu e_j,u} - v_{\mu,u}}{\nu} \approx \frac{d}{d\mu_j} v_{\mu,u},$$

where e_j is the j -th Euclidean basis element, and $\nu := 10^{-3}$. Let us mention that the approximation of the sensitivities can be improved by using other (and more accurate) methods to evaluate the partial derivatives, such as automatic differentiation (see, e.g., Rall (1981)).

Then, the information matrix $M = M_{\mu}^u$, evaluated at some parameter μ and input array u , is defined as

$$(M)_{j,l} := \sum_{k=0}^{K-1} \frac{s_k^j s_k^l + s_{k+1}^j s_{k+1}^l}{2} (t_{k+1} - t_k), \quad 1 \leq j, l \leq d,$$

namely, as the approximation of the L^2 inner product $\langle s^j, s^l \rangle_{L^2(0,t_f)}$ via the trapezoidal rule, where the sensitivities are evaluated at the parameter μ and the current i_u .

As we want to maximize the total information coming from a collection of inputs, we modify the information matrix of each design iteration. In particular, we simply sum the information matrices, thus at design iteration n we define the *optimal design function*

$$\Phi(u) := -\log_{10} \left(\det \left(\sum_{m=1}^{n-1} M_{\mu^{n-1}}^{[m]} + M_{\mu^{n-1}}^u \right) \right) + \gamma \|u\|_2^2,$$

where $M^{[m]}$, $m = 1, \dots, n-1$, denotes the information matrix corresponding to input array $u^{[m]}$, $\gamma := 10^{-4}$ accompanies a regularization term and $\|\cdot\|_p$ indicates the ℓ^p -norm for $1 \leq p \leq \infty$.

Hence, to find a new input array we solve the optimization problem

$$\min_{u \in \mathbb{R}^{n_u+1}} \Phi(u) \quad \text{subject to} \quad u^{\min} \leq u \leq u^{\max}, \quad (10)$$

where $\mathbf{u}^{\min}$ and $\mathbf{u}^{\max}$ are the bounds in $\mathbb{R}^{n_u+1}$ chosen as

$$\begin{aligned} -8.8 &=: u_j^{\min} \leq u_j \leq u_j^{\max} := 8.8, \quad j = 1, \dots, n_u, \\ 3.3 &=: u_{n_u+1}^{\min} \leq v_0 \leq u_{n_u+1}^{\max} := 4.1. \end{aligned}$$

in our experiments, where the unit of current is A and the unit of the initial voltage is V.

The optimization problem (10) is solved numerically using the routine `fmin_l_bfgs_b` from the `scipy` Python library; see, e.g., Virtanen et al. (2020).

As mentioned earlier, when we want the algorithm to generate a fixed number of inputs, we need to make them distinct, since there is no information gain when the input generated is equal to one we already have. We therefore define an alternative (nonsmooth) optimization function that enforces the new inputs to be sufficiently far from each other

$$\hat{\Phi}(\mathbf{u}) := \Phi(\mathbf{u}) + \sum_{\substack{\mathbf{u}^{[n]} \text{ previous} \\ \text{controls}}} \frac{1}{1 + 100 \|\mathbf{u} - \mathbf{u}^{[n]}\|_{\infty}}. \quad (11)$$

The penalizing function, as well as the scalar 100 have been chosen in an intuitive way in order to guarantee that we stop our algorithm provided our current control is sufficiently far from the already computed controls, but also not to modify the initial optimization problem too strongly.

2.4. Parameter estimation

Assuming now that the input vector $\mathbf{u}$ (hence the current i) is fixed, we want to find the parameter vector $\boldsymbol{\mu}$ minimizing the discrepancy between our parametrized model and data coming from the real application. We assume, for the moment, that the data is given by the voltage corresponding to some unknown parameter $\boldsymbol{\mu}^*$, evaluated on the time array $\mathbf{t}$, i.e.,

$$w_{\mathbf{u}} = v_{\boldsymbol{\mu}^*, \mathbf{u}} : [0, t_f] \rightarrow \mathbb{R},$$

and its discretization is indicated as $\mathbf{w}_{\mathbf{u}} := \mathbf{v}_{\boldsymbol{\mu}^*, \mathbf{u}} \in \mathbb{R}^{K+1}$. This means that we consider noiseless virtual data.

For the numerical optimization we use a least squares method (from `scipy.optimize`, see Branch et al. (1999)) to fit $\mathbf{v}_{\boldsymbol{\mu}, \mathbf{u}}$ to $\mathbf{w}_{\mathbf{u}}$. In particular, we minimize the relative error $e_{\mathbf{u}}(\boldsymbol{\mu}) := (\mathbf{v}_{\boldsymbol{\mu}, \mathbf{u}} - \mathbf{w}_{\mathbf{u}})/\mathbf{w}_{\mathbf{u}}$ (where the division

is intended component-wise) by in particular solving

$$\min_{\boldsymbol{\mu} \in \mathcal{P}_{\text{ad}}} J(\boldsymbol{\mu}) := \frac{1}{2} \left\| \frac{\mathbf{v}_{\boldsymbol{\mu}, \mathbf{u}} - \mathbf{w}_{\mathbf{u}}}{\mathbf{w}_{\mathbf{u}}} \right\|_2^2, \quad (12)$$

where $\mathcal{P}_{\text{ad}} \subset \mathbb{R}^d$ is the closed, convex set of admissible parameters – in our case the box-constrained set defined by $\boldsymbol{\mu}^L \leq \boldsymbol{\mu} \leq \boldsymbol{\mu}^U$ in Table 2.

In the framework of multiple data, we can simply merge information by stacking the data vectors together. In particular, given $\mathbf{u}^{[1]}, \dots, \mathbf{u}^{[n]}$ input arrays generated by the algorithm, and $\mathbf{w}^{[1]}, \dots, \mathbf{w}^{[n]}$ associated data vectors (with $\mathbf{w}^{[n]} := \mathbf{w}_{\mathbf{u}^{[n]}} = \mathbf{v}_{\boldsymbol{\mu}^*, \mathbf{u}^{[n]}}$), then we want to fit $\mathbf{v}_{\boldsymbol{\mu}, \mathbf{u}^{[1]}}$ to $\mathbf{w}^{[1]}$ and so on. We concatenate all vectors of relative error into one vector $[\mathbf{e}_{\mathbf{u}^{[1]}}(\boldsymbol{\mu}), \dots, \mathbf{e}_{\mathbf{u}^{[n]}}(\boldsymbol{\mu})] \in \mathbb{R}^{n(K+1)}$ and we minimize its ℓ^2 -norm. The parameter generated at OID iteration n is then the optimizer $\boldsymbol{\mu}^n$. We mention that in order to optimize computing time, we evaluate voltage vectors in a parallel way for all different current inputs.

We later explore the effect of the input design on the parameter optimization problem by analyzing the Hessian matrix H of the least-squares function J defined in (12), evaluated in the optimum $\boldsymbol{\mu}^*$. In particular, we analyze the value

$$\beta := \lambda_{\max}(H) / \lambda_{\min}(H), \quad (13)$$

i.e., the conditioning number of the Hessian, where at iteration n this value will be indicated as β_n – as the Hessian matrix changes at each OID iteration n . When this value decreases, the smallest eigenvalue is growing because in this construction the biggest eigenvalue cannot decrease, hence indicating positivity of the Hessian and convexity of the optimization problem. Let us comment on the fact that we cannot use the term β in the definition of the optimal input design optimization problem, since this quantity depends on knowing the hidden parameter $\boldsymbol{\mu}^*$. Nonetheless, we hope to minimize this value indirectly.

2.5. Another design framework

Generating ten different short inputs sounds like a good idea if we want to look at the behavior of a battery cell in all different states of charge: starting from different initial voltages serves exactly this purpose. But there might be a problem in the actual use of these different current profiles, since we cannot evaluate all inputs at the same time in a parallel way – as we do in the

numerical model. On the contrary, before and after every input we need a phase of preparation to bring the battery cell at the right initial voltage, and a long phase of rest to let the internal electro-chemical dynamics stabilize. For this reason, even if ten different inputs of 60s each together sum up to 600s, actually setting up and running the experiment can take more than 60 000 s.

Hence, we have modified the design framework and structure in order to avoid this problem. The idea is to generate a sequence of new input intervals successively, rather than to use them as entirely different current profiles. As a consequence, we may not choose the initial voltage of each interval, because this will be given by the ending voltage of the previous interval. Rather, we may only choose the initial voltage, namely the starting voltage of the first interval; this is chosen as 3.9 V, equivalent to appr. 75 % SOC.

We also realize that we need a resting interval in order to capture the long-term dynamics. Hence, our input is translated into six “jumps” of a 120s high-frequency time interval followed by a 600s resting phase. An example of the first input with arbitrary jumps is plotted in Figure 1. This time we are not arbitrarily choosing the first input $u^{[1]} = (u_1^{[1]}, \dots, u_6^{[1]})$, but it is found in the first step of the OID algorithm. In the following iterations, we find input array $u^{[n]}$ by optimizing the uncertainty of the full-concatenated current input, i.e., we add the new input to the old one and hence consider only one longer input. See Figure 1 for a visual explanation.

2.6. Analysis on real battery data

While we evaluate these optimal inputs using virtual data $w_u = v_{\mu^*, u}$ in the algorithm, we want to check how beneficial they are in a real-world parameter estimation problem. To this end, we have run experiments on a real battery cell and collected the measurement data. The measured voltage data obtained through cycling two Molicel INR21700-P45B commercial cells is used in this work. For a given starting voltage, the cell is first fully charged at a constant current of 4.5 A and then, at reaching the top voltage of 4.2 V, switch to constant voltage mode until the magnitude of the current tapered down below 0.05 A. After a resting period of 600 s, the cell is discharged at 4.5 A to the given voltage followed again by a constant voltage step until 0.05 A is underrun. Then, the designed current profile is applied in a “current simulation” step. All tests were carried out using an Arbin LBT21084HC battery testing system and a Memmert incubator with Peltier cooling (model IPP600) for maintaining the temperature at $(25 \pm 2)^\circ\text{C}$ by forced air cooling.

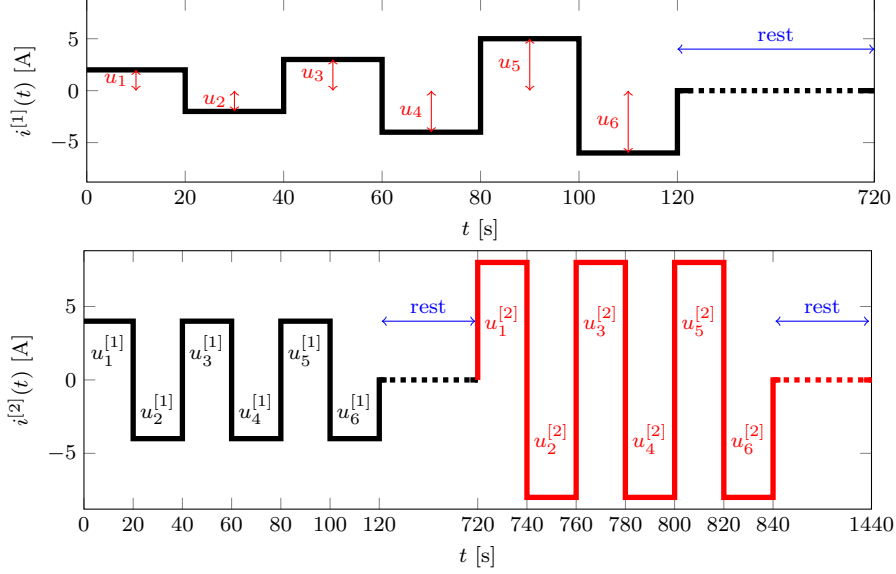


Figure 1: Visual arbitrary example of the concatenated-inputs setting. Above: initial current function, where the input variables define the 6 jumps and a successive rest phase is imposed. Below: once the first current function is fixed, the next input array $u^{[2]}$ is optimized by optimizing the concatenated current function $i^{[2]}$.

We want to use these real-world data to measure the effect of the optimal input design on the parameter estimation. This data also includes preparation steps at the beginning and the resting phase at the beginning, and depending on the test we will use information coming from the extra phases or not. Furthermore, the time stepping was chosen depending on the situation, varying between a minimum of 0.0002s (used in high-frequency dynamics) and a maximum of 1.0s (used in the resting phases).

3. Results and Discussion

3.1. First input design: collection of inputs

We have applied the input design algorithm with piecewise constant inputs consisting of 24 jumps, therefore each continuous sub-interval is 2.5s long and the input dimension is 25. We furthermore set the maximum number of inputs to 10 and arbitrarily fix the first input as

$$u^{[1]} := [-1, +1, -1, +1, \dots, -1, +1, 3.7] \in \mathbb{R}^{25}. \quad (14)$$

This corresponds to a quickly alternating input applied to a roughly mid-charge battery. The generated inputs are plotted in Figure 2.

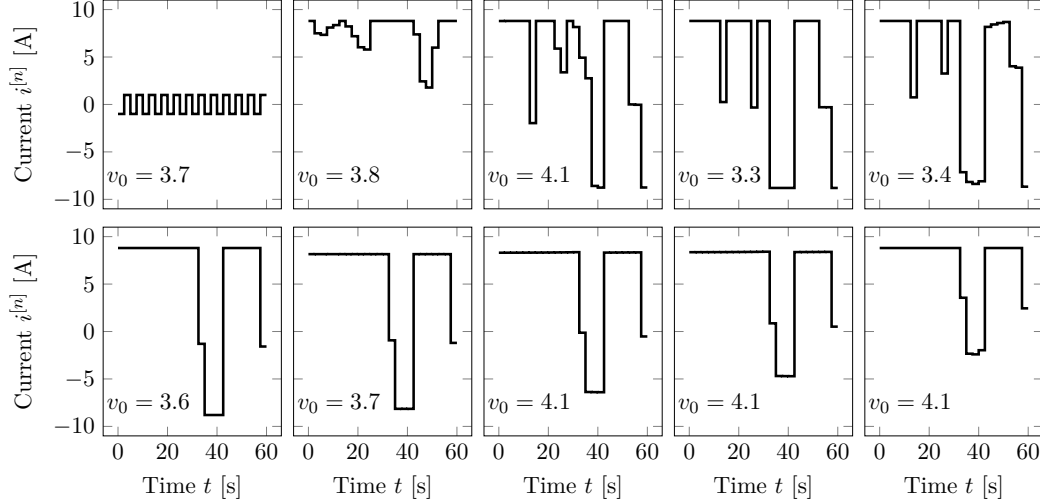


Figure 2: Input functions $i^{[n]}$, $n = 1, \dots, 10$ generated by the algorithm. In the lower left corner the initial voltages v_0 are indicated.

Notice how some of the inputs look very similar, but the penalization function introduced in (11) guarantees that we do not use the exact same input array more than once.

In order to analyze the effect of the input design algorithm, in Figure 3 we plot the change in different quantities, such as the design algorithm the uncertainty function $\hat{\Phi}$, the parameter optimization cost J , the relative error $e^{\text{rel}}(\boldsymbol{\mu}^n) := \|\boldsymbol{\mu}^* - \boldsymbol{\mu}^n\|_2 / \|\boldsymbol{\mu}^*\|_2$ and the sequence β_n defined in (13). In the same plot, in the lower area we plot a qualitative analysis of the input design, where we compare the results of 100 parameter optimizations with randomized initial parameter at the beginning and at the end of the algorithm, i.e., comparing the information before (in dark colored bars) and after (in light colored bars). Firstly, we can comment on the properties of the initial input $u^{[1]}$ defined in (14): while it looks like the error J is already low ($1.77 \cdot 10^{-10}$) at the initial parameter, the relative error of the parameter reads in fact 2.22. Namely, the initial input holds little information and it does not provide data such that the parameter optimization approximates $\boldsymbol{\mu}^*$. We can observe this in the bar plot, too: the blue bars do not approximate the parameter indicated with the red cross most of the time.

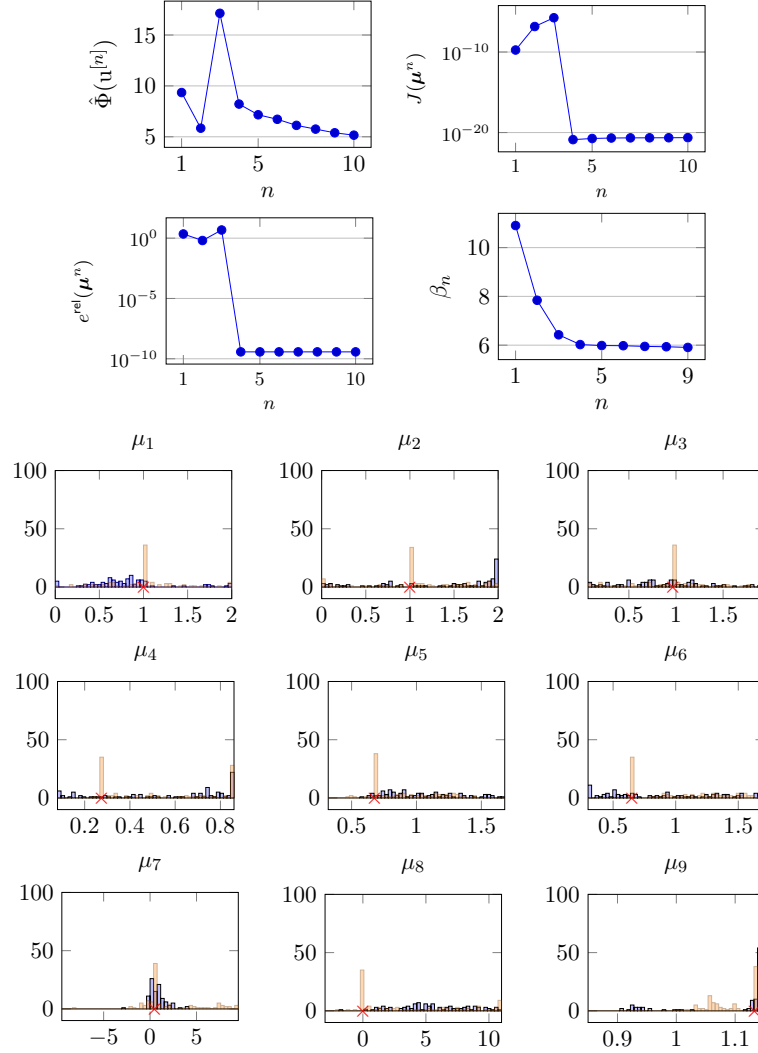


Figure 3: Top, blue: information corresponding to each design iteration n . First line: uncertainty function and cost evaluated in the optimal parameter. Second line: Euclidean norm of the error between the optimal parameter and the true parameter and value β_n given in (13). Bottom: qualitative statistical study for the input design algorithm. Dark bars (blue) show the results of 100 parameter estimations using only the input given by $u^{[1]}$. Light bars (orange) show the results using the optimal collection of inputs. The red crosses mark the hidden parameters.

During the design algorithm the uncertainty function, the parameter optimization cost and the relative error do not decrease monotonously. The OID algorithm converges to the true parameter after some preliminary steps, in particular reaching the neighborhood of μ^* in the fourth iteration. This could be explained by the behavior of the sequence β_n , which decreases monotonously, reaching a plateau after four iterations, when the algorithm finds the hidden parameter with a relative error of approximately $3.73 \cdot 10^{-10}$.

In the bar plots of Figure 3 we can observe that the input design improves the parameter estimation significantly: with a non-informative input (dark bars), the parameter estimation fails almost every time, while using the collection of selected inputs (light bars) the convergence to the hidden parameter (the red cross) improves greatly.

Although the converge to the optimal parameter does not happen always, the optimization results cluster around the optimal parameters. Overall, we can see a degree of bias in the bar plots of Figure 3 – namely, we can guess the hidden parameter in an easier way. However, this framework might find in its applicability some unexpected counter-effects, which we discuss in the next section.

3.2. Second input design: concatenated input

In this setting we generate 9 inputs, each lasting 120 s: therefore, concatenating the respective jumps and rest phases together we reach an “optimal” input lasting 6480 s, i.e. 108 minutes. This optimal current $i^{[9]}$ generated by the algorithm is plotted in Figure 4 together with the corresponding output voltage $v_{\mu^*, u^{[9]}}$ produced from the battery cell model using such input. Further information for each OID iteration are also plotted in Figure 4. In this case we can observe a non monotone decrease of the quantity β_n , but still a good convergence of the parameter optimization in later iterations, reaching a relative parameter error of $9.74 \cdot 10^{-12}$.

Similar to what we did previously, in the bottom part of Figure 4 we plot the result of 100 separate executions of the parameter optimization in order to show the estimation error for randomized initial parameters. We observe an overall better consistence and accuracy in the convergence to the true parameter; considering the fact that we the overall experiment time will be reduced at least 4 times, as shown in the next section, we regard this as a great improvement of the input design presented in the previous section.

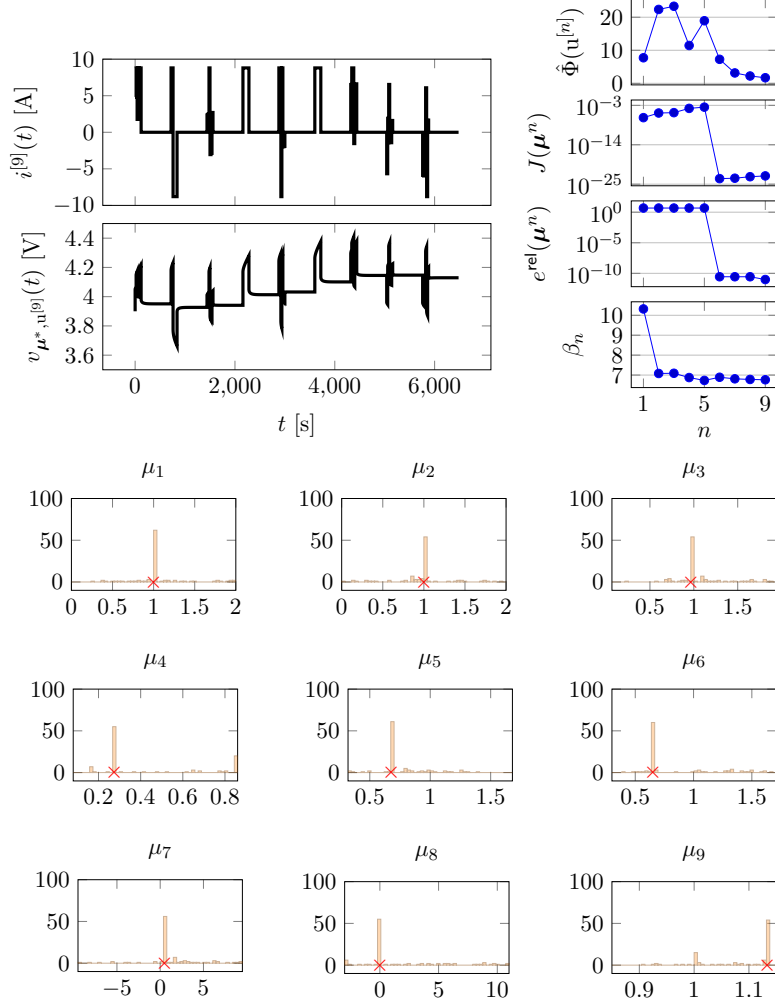


Figure 4: Top left, in black: optimal input generated by the algorithm and corresponding output voltage generated by the battery cell model. Top right, in blue: plots of different quantities at OID iteration n . Bottom bar plots: qualitative statistical results of the optimal parameter of 100 parameter estimations. The red crosses mark the true parameters μ^* .

3.3. Real battery data

We consider four different situations, depending on two different factors: whether we use the collection of inputs derived in Section 3.1 or the concatenated input derived in Section 3.2, and whether we consider the additional data provided by the preparation and resting phases, or not. These choices

are summarized in Table 3, and some of the results given in each test are summarized in Table 4.

	Cut data: only impulses	Full data: impulses + rest phases
Collection of inputs	Test 1	Test 2
Concatenated input	Test 3	Test 4

Table 3: Considered tests for the analysis of real data.

	TESTS			
	1	2	3	4
Experiment time (s)	600	63,362	6,480	14,199
Data points	20,350	83,160	216,002	223,727
Optimization time (s)	800	154	550	450

Table 4: First line: total input time t_f ; second line: number of data points; third line: average time for the parameter optimization.

In each of these situations we run 100 parameter optimizations in order to infer the unknown battery cell parameters. We analyze the quality of the estimation using a bar plot, and we try to infer the true parameter by proposing an optimal parameter. Since this is unknown, we finally perform a qualitative error analysis on the proposed parameters given in each test.

In particular, we will propose three parameters $\mu^{\text{opt},2}$, $\mu^{\text{opt},3}$ and $\mu^{\text{opt},4}$ inferred from test 2, 3 and 4, respectively. For each of these parameters we plot the output voltage produced by the numerical model $v_{\mu,u}(t)$, as well as the relative error given by the discrepancy with the real battery data $w(t)$, namely

$$e_{\mu^{\text{opt},k}}^{\text{rel},[n]}(t) := (w^{[n]}(t) - v_{\mu^{\text{opt},k},u}^{[n]}(t))/w^{[n]}(t) \quad \text{for } n = 2, 3, 4 \text{ and } k = 2, 3, 4.$$

The results are shown in the upcoming figures.

Test 1

By counting only the 10 one-minute intervals, we get a relatively small experiment time and number of data points. Although the optimization

is longer than in other cases, we cannot identify the unknown parameters with confidence. Even if we are using the same input collections as the results shown in Figure 3, the bar plots shown in Figure 5 look much worse. Therefore, in this case we conclude that this input design is not successful, as we derive no hint about where the hidden parameters might be.

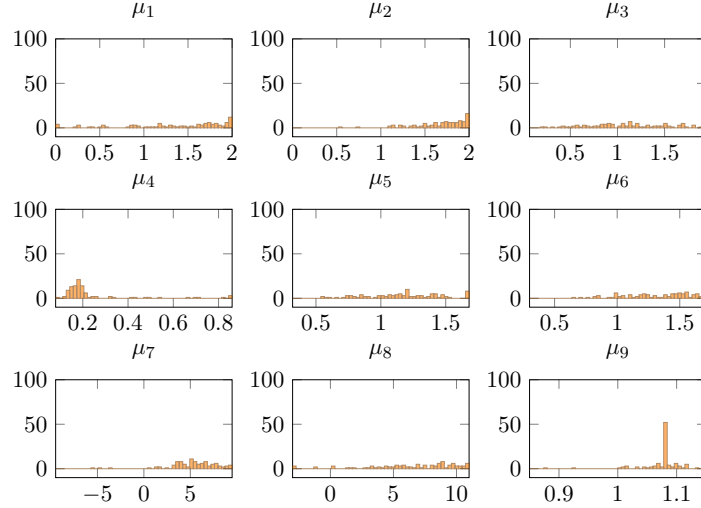


Figure 5: Results for Test 1. Bar plots of qualitative analysis of 100 parameter estimations starting from randomized initial guess.

Test 2

Since the production of the data necessary for Test 1 produces, by default, the data used in Test 2, this test is necessarily more informative and hence the estimation of the true parameter is more successful. However, the experiment lasts for over 17 hours, with an average optimization time of about two and a half minutes. As shown in Figure 6, we can identify a candidate $\boldsymbol{\mu}^{\text{opt},2}$, indicated with the blue mark.

The analysis of the model error given by the proposed parameter confirms the quality of such parameter: for the inputs $u^{[2]}$ and $u^{[3]}$ the relative error oscillates in the order of 10^{-3} , while for the input $u^{[4]}$ the order of the relative error is 10^{-2} . The numerical model tuned with such parameter is therefore an accurate simulator of the battery cell taken into consideration.

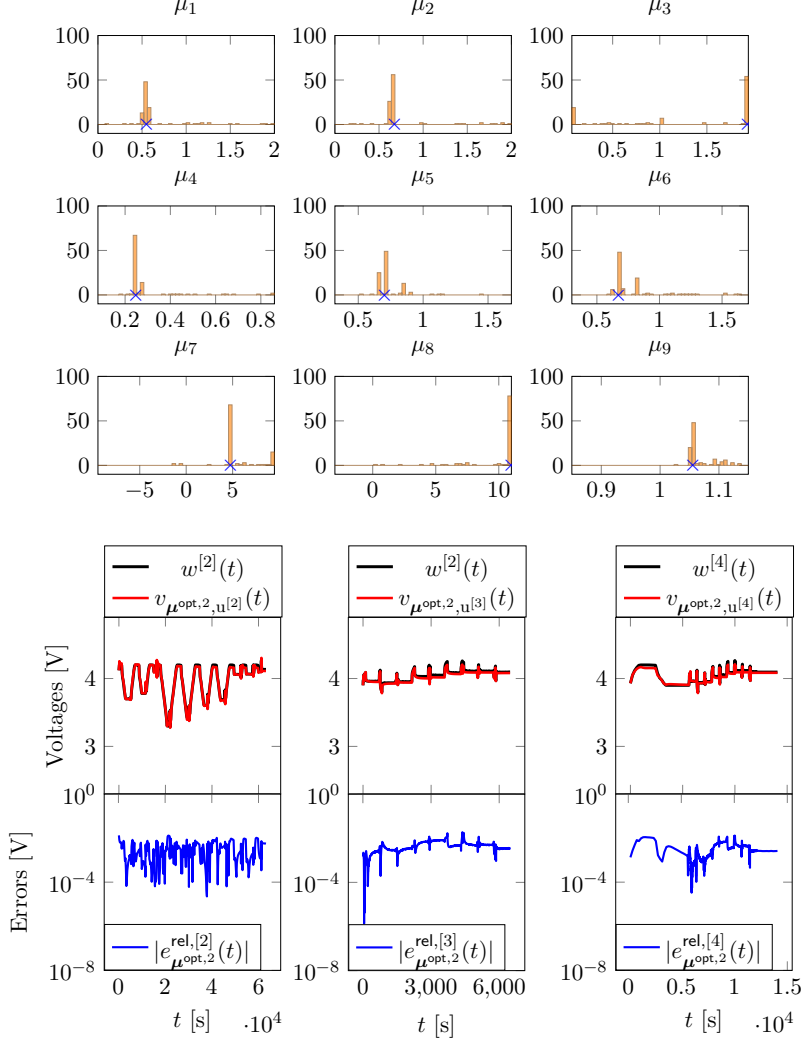


Figure 6: Results for Test 2. Top: bar plots of qualitative analysis of 100 parameter estimations starting from randomized initial guess; the blue cross finds the proposed parameter $\mu^{\text{opt},2}$; bottom: data, output and error generated by the proposed parameter.

Test 3

By construction, the time of the generated experiment is 108 minutes, and the optimization over 216,002 points takes in average several minutes. However, we can see that in this case the estimations gravitate around different local minima. With $\mu^{\text{opt},3}$ we have indicated the most common optimizer, indicated in Figure 7 with the blue mark.

This parameter produces accurate results for the data $w^{[3]}$, but accumulates a too high error in the higher voltage changes occurring in data observations $w^{[2]}$ and $w^{[4]}$, hence not capturing very well the full range of battery cell dynamics.

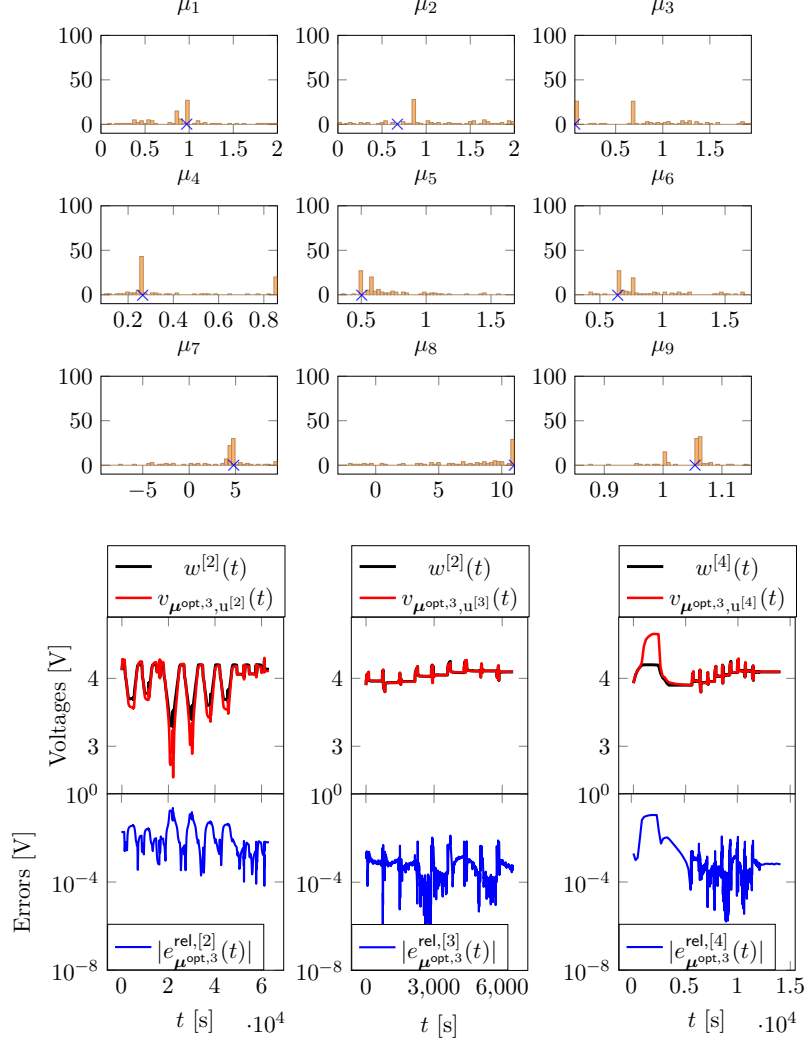


Figure 7: Results for Test 3. Top: bar plots of qualitative analysis of 100 parameter estimations starting from randomized initial guess; the blue cross finds the proposed parameter $\mu^{\text{opt},3}$; bottom: data, output and error generated by the proposed parameter.

Test 4

As mentioned already, Test 4 is obtained by using the data of Test 3 together with the measurements available in the preparation and rest phases. The candidate parameter is definitely more concentrated around a single parameter $\mu^{\text{opt},4}$, plotted in Figure 8 with the blue cross. Counter-intuitively, the optimization over a larger set of data produces in this case a faster convergence in the parameter optimization. We interpret this effect as a result of the well-posedness of the Hessian matrix and the convexity of the parameter optimization.

We can see the results in Figure 8. The proposed parameter $\mu^{\text{opt},4}$ holds good approximation of the 3 sets of data, comparable to the one given by parameter $\mu^{\text{opt},2}$. The construction introduced in Section 2.5 leads to major time cuts with the only impact being the optimization time, which averages to about 154 s for Test 2 and 450 s for Test 4, since the number of data points has almost tripled.

Comments on the accuracy of the optimizers

At a first glance, it seems that we cannot find a parameter to perfectly replicate the data. This might depend on a number of factors. The first justification comes from the fact that we are working with a simplified numerical model with a reduced number of active, unknown parameters, and therefore the approximation of a real battery cell data is indeed impossible. We can also imagine that adjusting a number of parameters which now are fixed – for example, the RK parameters – we can fit better to the data.

On the bright side, we can make considerations about the merit of the input design. It has in fact improved robustness the parameter estimation, because it has made it harder, i.e., it happens more rarely, to fall in other local optima. While it is possible that a better fitting parameter is hiding in another local optimum, it is highly unlikely.

It looks like the parameter inferred from Test 2, i.e., the longest experiment, is the one better approximating the data. Test 4, which is over 4 times shorter also produces a parameter that approximates well the real data dynamics. As usual, using more expensive experiments produces data of better quality, while using shorter experiments might save time by sacrificing some of the accuracy. We believe the main advantage of using the input design algorithm is the enforcement of the well-posedness of the Hessian matrix corresponding to the parameter optimization problem. The short optimization

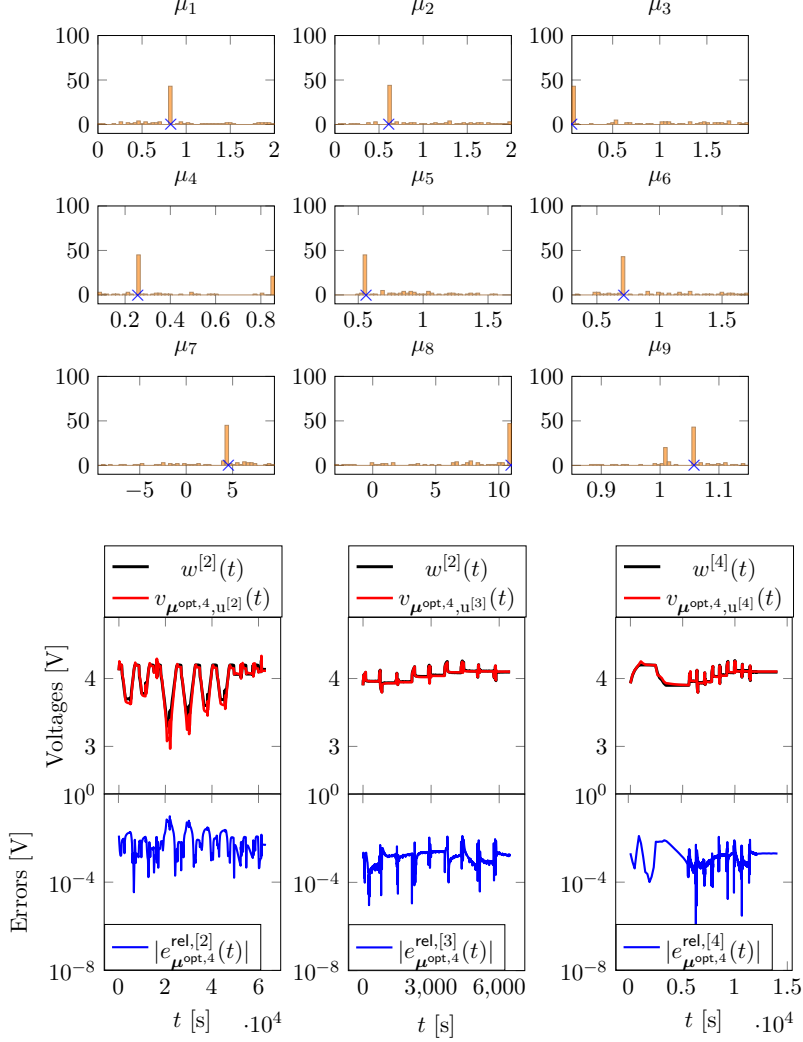


Figure 8: Results for Test 4. Top: bar plots of qualitative analysis of 100 parameter estimations starting from randomized initial guess; the blue cross finds the proposed parameter $\mu^{\text{opt},4}$; bottom: data, output and error generated by the proposed parameter.

times might be the best experimental benefit, as many optimization methods work very well with convex problems.

The input design might be even more effective when applied to more complex battery models, as it could be able to accurately represent the battery cell in all conditions: it can indeed happen that our single particle model fails at reproducing real battery cells in the settings chosen by the input design

where valuable information lies, for example, for very high or very low states of charge.

We can, of course, also argue that the parameter estimation has failed because the input design was badly set up. We have selected inputs with very restricted shapes and lengths in an arbitrary and intuitive way. For example, with longer current jumps, or longer pauses, or in general longer experiment, the input design could have found the optimal parameter reproducing the data. Of course, the possibilities of generalization are endless, and we encourage future works to obtain better results.

4. Conclusions

In this work we have applied an algorithm for the optimal input design for the parameter estimation of a lithium-ion battery model. After introducing the numerical model, we have described a general iterative input design algorithm, where inputs are chosen in order to maximize the observability of the parameters, quantified using an approximated Fisher information matrix. By discretizing the information matrix we then define the input design problem as a finite dimensional optimization problem.

At first, we generate a collection of different input functions in order to maximize the overall observation, i.e., the collective information. The data given by this collection of input is indeed effective for the parameter estimation, and we show the effect of the input design on the convexity of the optimization problem. However, in the lab experiments, the use of the optimal inputs results in extremely long experiment times, since before and after each input some extra time needs to be taken into account, either for a resting phase or for taking the battery to a certain state of charge.

For this reason, a second input design algorithm is used, where a continuous, longer input is obtained by concatenation of optimal sub-intervals. In this way, the problem of resting and state of charge preparation is avoided – or at least done only once – and the overall experiment is shorter.

We derive four tests to interpret the effect of the input design and the accuracy of the parameter estimation using real battery data. In three out of four tests the input design can indeed make the parameter estimation easier, in the way that it makes us aware of a common local optimum, which we propose as the approximator of the unknown parameter.

Although we infer different parameters with different discrepancy between the numerical output and the real battery voltage, we observe how the in-

put design was able to iteratively generate a current profile facilitating the parameter estimation, both in accuracy and in optimization time.

As mentioned, we believe that the advantage of the input design algorithms would be more evident in more complex battery cell models, where all the electro-chemical dynamics are reproduced and the data can be completely reproduced by the numerical model. In a wider scope, the techniques of optimal input/experimental design propose a goal-oriented way of defining experiment inputs and settings. Such designs might be built to guarantee several mathematical properties and rigor. The management of resources, such as time or energy, could be imposed a-priori in the design of the experiments in order to prevent waste.

Acknowledgments

The authors would like to acknowledge the financial support within the COMET K2 Competence Centers for Excellent Technologies from the Austrian Federal Ministry for Climate Action (BMK), the Austrian Federal Ministry for Labour and Economy (BMAW), the Province of Styria (Dept. 12) and the Styrian Business Promotion Agency (SFG). The Austrian Research Promotion Agency (FFG) has been authorised for the programme management. The authors further acknowledge support by the Deutsche Forschungsgemeinschaft for the project Localized Reduced Basis Methods for PDE-constrained Parameter Optimization under contract VO 1658/6-1.

Author contributions

Andrea Petrocchi: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing - Original Draft, Visualization. **Matthias K. Scharrer:** Software, Resources, Writing - Original Draft. **Franz Pichler:** Methodology, Software. **Stefan Volkwein:** Methodology, Writing - Review & Editing, Supervision, Project administration, Funding acquisition.

References

Atkinson, A., Donev, A., Tobias, R., 2007. Optimum experimental designs, with SAS. volume 34 of *Oxford Statistical Science Series*. Oxford University Press, Oxford. doi:10.1093/oso/9780199296590.001.0001.

- Branch, M., Coleman, T., Li, Y., 1999. A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal on Scientific Computing* 21, 1–23. doi:10.1137/S1064827595289108.
- Franceschini, G., Macchietto, S., 2008. Model-based design of experiments for parameter precision: State of the art. *Chemical Engineering Science* 63, 4846–4872.
- Goodwin, G., Payne, R., 1977. Dynamic system identification. volume 136 of *Mathematics in Science and Engineering*. Academic Press, Inc. [Harcourt Brace Jovanovich, Publishers], New York-London.
- Karthikeyan, D.K., Sikha, G., White, R.E., 2008. Thermodynamic model development for lithium intercalation electrodes. *Journal of Power Sources* 185, 1398 – 1407. URL: <http://www.sciencedirect.com/science/article/B6TH1-4T4XR9X-2/2/c0b51ae8a4bdea0067dbefc7ed39d7a2>, doi:DOI:10.1016/j.jpowsour.2008.07.077.
- Liu, J., Rothenberger, M., Mendoza, S., Mishra, P., Jung, Y.s., Fathy, H.K., 2016. Can an identifiability-optimizing test protocol improve the robustness of subsequent health-conscious lithium-ion battery control? an illustrative case study, in: 2016 American Control Conference (ACC), IEEE. pp. 6320–6325.
- López, C., Diana, C., Wozny, G., Flores-Tlacuahuac, A., Vasquez-Medrano, R., Zavala, V., 2016. A computational framework for identifiability and ill-conditioning analysis of lithium-ion battery models. *Industrial & Engineering Chemistry Research* 55, 3026–3042. doi:10.1021/acs.iecr.5b03910.
- Mehra, R., 1974. Optimal input signals for parameter estimation in dynamic systems—survey and new results. *IEEE transactions on automatic control* 19, 753–768.
- Newman, J., Thomas-Alyea, K.E., 2004. *Electrochemical Systems*. 3rd ed., John Wiley & Sons, Inc., Hoboken, New Jersey. doi:ISBN:978-0471477563.
- Newman, J., Tiedemann, W., 1993. Potential and current distribution in electrochemical cells. *Journal of The Electrochemical Society* 140, 1961–

1968. URL: <http://link.aip.org/link/?JES/140/1961/1>, doi:10.1149/1.2220746.
- Ning, G., Popov, B.N., 2004. Cycle life modeling of lithium-ion batteries. *Journal of The Electrochemical Society* 151, A1584–A1591. URL: <http://link.aip.org/link/?JES/151/A1584/1>, doi:10.1149/1.1787631.
- Park, S., Kato, D., Gima, Z., Klein, R., Moura, S., 2018. Optimal experimental design for parameterization of an electrochemical lithium-ion battery model. *Journal of The Electrochemical Society* 165, A1309.
- Pozzi, A., Ciaramella, G., Volkwein, S., Raimondo, D., 2018. Optimal design of experiments for a lithium-ion cell: parameters identification of an isothermal single particle model with electrolyte dynamics. *Industrial & Engineering Chemistry Research* 58, 1286–1299. doi:10.1021/acs.iecr.8b04580.
- Pozzi, A., Xie, X., Raimondo, D.M., Schenkendorf, R., 2020. Global sensitivity methods for design of experiments in lithium-ion battery context. *IFAC-PapersOnLine* 53, 7248–7255.
- Pronzato, L., Pázman, A., 2013. Design of experiments in nonlinear models. *Lecture notes in statistics* 212.
- Pukelsheim, F., 2006. Optimal design of experiments. SIAM.
- Pázman, A., 1986. Foundations of optimum experimental design. *Mathematics and its applications*, D. Reidel. URL: <https://cir.nii.ac.jp/crid/1130282269915354752>.
- Rafajłowicz, E., 1986. Optimum choice of moving sensor trajectories for distributed-parameter system identification. *International Journal of Control* 43, 1441–1451. doi:10.1080/00207178608933550.
- Rall, L., 1981. Automatic Differentiation: Techniques and Applications. volume 120 of *Lecture Notes in Computer Science*. Springer, Berlin. doi:10.1007/3-540-10861-0.
- Reddy, S., Scharrer, M., Pichler, F., Watzenig, D., Dulikravich, G., 2019. Accelerating parameter estimation in doyle–fuller–newman model for lithium-ion batteries. *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering* 38, 1533–1544.

- Román-Ramírez, L.A., Marco, J., 2022. Design of experiments applied to lithium-ion batteries: A literature review. *Applied Energy* 320, 119305.
- Santhanagopalan, S., Guo, Q., Ramadass, P., White, R.E., 2006. Review of models for predicting the cycling performance of lithium ion batteries. *Journal of Power Sources* 156, 620 – 628. URL: <http://www.sciencedirect.com/science/article/B6TH1-4GM4620-7/2/022854c2137ca57cdc1f45fb0d5542e2>, doi:DOI: 10.1016/j.jpowsour.2005.05.070.
- Schmidt, A.P., Bitzer, M., Imre, Á.W., Guzzella, L., 2010. Experiment-driven electrochemical modeling and systematic parameterization for a lithium-ion battery cell. *Journal of Power Sources* 195, 5071–5080.
- Uciński, D., 2005. Optimal measurement methods for distributed parameter system identification. *Systems and Control Series*, CRC Press, Boca Raton, FL.
- Virtanen, P., Gommers, R., Oliphant, T., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S., Brett, M., Wilson, J., Millman, K., Mayorov, N., Nelson, A., Jones, E., Kern, R., Larson, E., Carey, C., Polat, İ., Feng, Y., Moore, E., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E., Harris, C., Archibald, A., Ribeiro, A., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors, 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17, 261–272.