Science Arts & Métiers (SAM)
is an open access repository that collects the work of Arts et Métiers Institute of Technology researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: https://sam.ensam.eu
Handle ID: .http://hdl.handle.net/10985/20776

To cite this version :

Abel SANCARLOS, Morgan CAMERON, Andreas ABEL, Elias CUETO, Jean-Louis DUVAL, Francisco CHINESTA SORIA - From ROM of Electrochemistry to AI-Based Battery Digital and Hybrid Twin - Archives of Computational Methods in Engineering - Vol. 28, n°3, p.979-1015 - 2020

Any correspondence concerning this service should be sent to the repository Administrator : scienceouverte@ensam.eu

From ROM of Electrochemistry to AI-Based Battery Digital and Hybrid Twin

Abel Sancarlos[1,2,3] · Morgan Cameron[1] · Andreas Abel[1] · Elias Cueto[3] · Jean-Louis Duval[1] · Francisco Chinesta[2]

Abstract

Lithium-ion batteries are widely used in the automobile industry (electric vehicles and hybrid electric vehicles) due to their high energy and power density. However, this raises new safety and reliability challenges which require development of novel sophisticated Battery Management Systems (BMS). A BMS ensures the safe and reliable operation of a battery pack and to realize it a model must be solved. However, current BMSs are not adapted to the specifications of the automotive industry, as they are unable to give accurate results at real-time rates and during a wide operation range. For this reason, the main focus of this work is to develop a Hybrid Twin, as introduced in Chinesta et al. (Arch Comput Methods Eng (in press), 2018. so as to meet the requirements of the new generation of BMS. To achieve this, three reduced order model techniques are applied to the most commonly used physics-based models, each one for a different range of application. First, a POD model is used to greatly reduce the simulation time and the computational effort for the pseudo-2D model, while maintaining its accuracy. In this way, cell design, optimization of parameters, and simulation of battery packs can be done while saving time and computational resources. In addition, its real-time performance has been studied. Next, a regression model is constructed from data by using the sparse-Proper Generalized Decomposition (s-PGD). It is shown that it achieves real-time performance for the whole electric vehicle (EV) system with a battery pack. In addition, this regression model can be used in a BMS without issues because of the simple algebraic expression obtained. A simulation of the EV with the proposed approach is demonstrated using the system simulation tool SimulationX (*ESI ITI GmbH. Dresden, Germany*). Furthermore, the Digital Twin created using the s-PGD does not only allow for real-time simulations, but it can also adapt its predictions taking into consideration the real driving conditions and the real driving cycle to change the planning in real-time. Finally, a data-driven model based on the employment of Dynamic Mode Decomposition techniques is developed to extract an on-line model that corrects the gap between prediction and measurement, thus constructing the first (to our knowledge) hybrid twin of a Li-ion battery able to self-correct from data. In addition, thanks to this model, the above gap is corrected during the driving process, taking into consideration real-time restrictions.

1 Introduction

After nearly a century with the Internal Combustion Engine (ICE) dominating the personal transportation sector, it now appears that the Electric Vehicle (EV) is on the verge of becoming its successor. We have witnessed important movements and actions in this direction both at commercial and (inter-)national level in recent years. The former, through research and development of the market, the latter by changing legislation for these purposes. Needless to say, the electric battery sector is expected to become of utmost importance in the years to come.

One key issue in the EV is that its performance is closely tied to the performance of the battery. For this reason, it is fundamental to analyze this component of the car. Most EV have an on-board Battery Management System (BMS) that maintains safe and consistent operation of the battery module and optimize the performance of the battery system.

Francisco Chinesta
francisco.chinesta@ec-nantes.fr

Abel Sancarlos
Abel.SancarlosGonzalez@esi-group.com

[1] ESI Group, 3bis, Rue Saarinen, 94528 Rungis Cedex, France

[2] ESI Chair at PIMM Lab, ENSAM ParisTech, 151, bvd. de l'Hopital, 75013 Paris, France

[3] Aragon Institute of Engineering Research, Universidad de Zaragoza, Maria de Luna, s.n., 50018 Zaragoza, Spain

In order to infer micro-scale, unmeasurable states from the available, macro-scale measurements, a model must be solved for the BMS. In automotive applications, this model should accurately describe the battery behavior under the wide range of possible operating conditions encountered in practice.

Current BMS often use empirical or phenomenological models, such as the Equivalent-Circuit Models (ECM) for battery state estimation and control [22, 44]. These models have a relatively low computational cost, but they can only be used within the narrow operating conditions in which they have been parametrized. If the range of validity wants to be extended, a large amount of experimental data is required under a wide range of operating conditions. Furthermore, making predictions when degradation happens is challenging or simply impossible.

Alternatively, there exist different physics-based models describing the thermodynamics, reaction kinetics and transport within the cell [12, 13, 15]. These models are more accurate and also valid over a wide range of operating conditions. In addition, they can be directly coupled to degradation models. However, they are highly time-consuming to be of practical use in real-time applications due to the limited computational resources of an embedded BMS. A good summary of thermal-electrochemical models and ECM used in literature can be found in [54].

The physics-based model most commonly used is Newman's pseudo-2D (P2D) Model [13]. The P2D model can accurately describe lithium-ion battery time evolution responses over a wide operating range [14], and it is therefore an excellent starting point for the following generation of BMS [4].

Nevertheless, the P2D model needs a bigger computation effort than ECM does. For this reason, several attempts to obtain simplified models derived from the P2D model to guarantee its use in real-time applications can be found in the literature.

One of the most commonly used simplifications is known as the Single Particle Model (SPM) [11, 40, 41, 47]. It assumes that each electrode can be represented by a unique solid-phase particle and considers that the concentration in the electrolyte does not vary spatially or temporally. State estimation using the SPM and similar approximations often include the use of an Extended Kalman Filter (EKF) algorithm [47]—for which reduced-order versions exist, see [16]—, or a backstepping PDE state estimator [40].

In [11], an averaged electrochemical model similar to the SPM was used, by applying also the EKF, and in [41], an improvement of the SPM was made to include the dynamics of the concentration in the electrolyte. Nevertheless, these approaches are inherently limited because of the low current validity range of the SPM.

Other approaches include state estimation on reduced-order models derived from the P2D model [33–35, 50–52]. For example, in [52], the EKF is applied to a state space reduced-order model computed from the P2D model using a discrete-time realization algorithm [33–35]. However, as [4] remarks, the parameters of such reduced-order models may be difficult to interpret or have no direct physical meaning, which makes accounting for degradation effects difficult.

Most recent works have shown that the use of spectral numerical methods to discretize the P2D model produces a high model reduction without losing accuracy or physical significance of parameters [4]. However, the reduction depends on the choice of the basis. Therefore, if the maximum possible reduction is looked for, the best possible basis must be found (efficiency criteria). This is a goal which can be achieved with the POD approach proposed in this work.

For these reason, the main objective of this work is to develop a Hybrid Twin (HT), as introduced in [7], for lithium-ion batteries. In this way, a new generation of BMS for the automotive industry can be developed according to the expected necessities in this industrial application. The HT concept is based on Newman's P2D Model previously mentioned. This model is presented in detail in Sect. 2.

To that end, the first step is to achieve a simulation time for the model compatible with both the real-time feedback rates of the BMS and the available computer power in the car. To achieve this, we use first a classical Model Order Reduction (MOR) technique: the Proper Orthogonal Decomposition (POD) [8]. The analysis shows that the results of the POD model are indistinguishable, to the naked eye, to the original ones (without using the MOR).

In addition, POD results are compared with the software Lithium-ION SIMulation BAttery Toolbox (LIONSIMBA), see https://github.com/lionsimbatoolbox/LIONSIMBA. This is a reliable tool to compare our results because a validation of the software was done with respect to the COMSOL MultiPhysics commercial software (COMSOL, Inc., Stockholm, Sweden) and the Newman's DUALFOIL code [45].

The POD results and the methodology of the technique are detailed in Sect. 4. It is shown that by utilizing the proposed model, cell design, optimization of parameters and simulation of battery packs can be achieved while saving time and computational resources.

The second step is to integrate the developed model with all the other systems which constitute the EV under real-time constraints. The POD model may not be suitable or the best option to achieve this, so alternatives are explored.

Here we propose to use the sparse Proper Generalized Decomposition, s-PGD [25] to obtain an algebraic expression for the behavior of a battery cell according to different possible itineraries. We prove in this paper that we can model the behavior of the lithium-ion cell analyzed for such a wide range of itineraries using polynomials with the

proposed approach. Since our proposal transforms the problem of the cell model to the evaluation of a polynomial, there are no problems in coupling the battery model with the other systems in the EV. A detailed example of the accuracy and the range of use of the approach is discussed. The results and the explanation of the technique used in the regression (s-PGD) can be found in Sect. 5. To simulate the EV, the ESI system simulation software SimulationX (ESI ITI GmbH, Dresden, Germany) was used. The results of this study can be found in Sect. 6.

The Digital Twin created in the above section does not only allow real-time simulations, but can also adapt its predictions taking into consideration the real driving conditions and the real driving cycle to change the planning in real-time.

The next section (Sect. 7) is devoted to the data-driven modeling concept of the HT methodology. The main idea here is that biased deviations (between the model and the measurements) show hidden physics not considered in the model. These deviations therefore require a particular treatment, namely their on-line modeling by assimilating collected data. On the other hand, the unbiased deviation contribution is associated with noise and is addressed by using adequate filters.

For this reason, a data-driven model based upon the Dynamic Mode Decomposition [32] is developed to extract an on-line model to correct the gap between predictions and measurements. Thanks to the HT model, we can eliminate this gap during the driving process and take into consideration real-time restrictions. The results and the explanation of the technique used can be found in Sect. 7.

Finally, in Sect. 8 the general conclusions of the present work are shown.

2 Electrochemical Model: Newman's P2D Model

2.1 Newman's P2D Model

There exists a vast collection of models for the battery in the literature. Those related to the equivalent circuit of the battery are often simple enough to achieve a good computational time. However, they are typically limited to a narrow operational range. In addition, the battery models based on electrochemistry laws are generally preferred to the equivalent circuit or to other kinds of simplified models, because they also predict the physical cell limitations, which have a relevant effect in the automotive application, where the battery suffers very often the stress of very high transient loads. In addition, they are more accurate and have a wide operational range. The literature on electrochemical modeling of batteries is quite extensive, including both full order and simplified models.

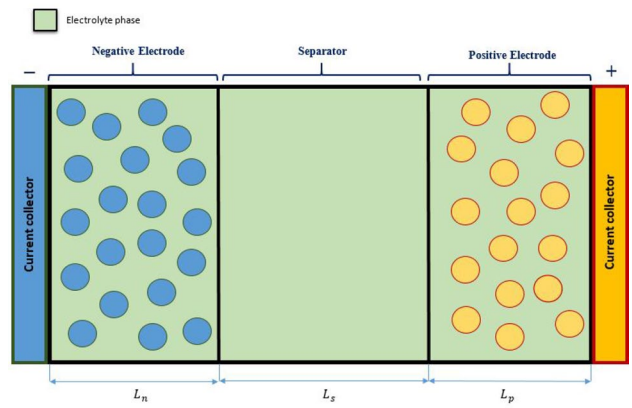

Fig. 1 Schematic of electrochemical cell model with coupled solid diffusion submodel

Newman's group developed a physics-based model using porous electrode and concentrated solution theories [13]. This model can accurately capture the battery response and has been widely used in the literature. However, there are problems to use the model in real-time applications or to achieve fast simulations and predictions because of the large computational effort needed to solve it. A method to reduce the computational resources required to solve the model is presented in Sect. 4.

The lithium-ion cell model shown in Fig. 1 consists of two porous electrodes composed of an active material (that can store lithium intercalated in the solid material) and a separator. The electrodes and the separator are soaked in an electrolyte that allows the transport of ions. During discharge, lithium stored in the anode is de-inserted from the active material and released as ions in the electrolyte. Driven by diffusion (concentration gradient) and migration (potential gradient), lithium ions travel through the separator to the cathode where they enter the lattice of the cathode active material.

The separator, while conductive to ions, is an electronic insulator, thus forcing electrons to follow an opposite path through an external circuit or load. On the other hand, the path of electrons and lithium-ions is reversed when the battery is charging (due to the fact that the physical reactions are made in the opposite sign).

In the domains of the problem, two phases are considered, the solid phase and the electrolyte phase, and they are treated as superimposed continua using porous electrode theory [4].

It is important to note that the x-dimension refers to the spatial dimension through the electrodes and the separator, according to Figs. 1, 2, and 3, and the r dimension refers to the depth in the particle where the diffusion of the lithium

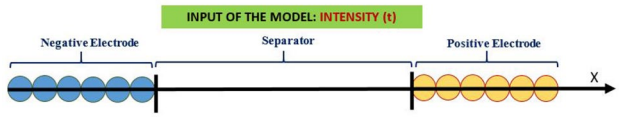

Fig. 2 Schematic of the P2D model

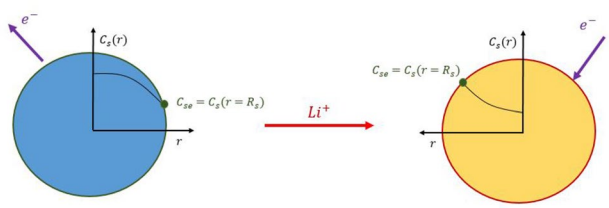

Fig. 3 Detail of the solid diffusion submodel. The path of electrons and the graphs of $c_s(r)$ showed are the ones for the discharging process

happens. The r-dimension of the spherical particles is called the pseudo second dimension. This is why the model is often referred to as Newman's P2D model in the literature.

The P2D model consists of a set of partial differential equations (PDEs) and algebraic constraints governing the evolution of lithium concentration and electric potential within the cell. The equations of the model describe the battery system with four quantities, i.e., concentration of lithium in solid and electrolyte phase, $c_s(r, x, t)$ and $c_e(x, t)$, respectively, and the volume-averaged electrical potential in the solid and electrolyte phase, $\phi_s(x, t)$ and $\phi_e(x, t)$. All the equations are coupled by the term $j^{\mathrm{Li}}(x, t)$ which is the reaction current resulting in production or consumption of Li.

The complete set of equations for the micro-macroscopic model is:

$$\frac{\partial c_s}{\partial t} = \frac{D_s}{r^2} \cdot \frac{\partial}{\partial r}\left(r^2 \frac{\partial c_s}{\partial r}\right), \tag{1}$$

$$\frac{\partial\left(\varepsilon_e \cdot c_e\right)}{\partial t} = \frac{\partial}{\partial x}\left(D_e^{\mathrm{eff}} \frac{\partial c_e}{\partial x}\right) + \frac{1 - t_+}{F} \cdot j^{\mathrm{Li}}, \tag{2}$$

$$j^{\mathrm{Li}} = \frac{\partial}{\partial x}\left(\sigma^{\mathrm{eff}} \frac{\partial \phi_s}{\partial x}\right), \tag{3}$$

$$j^{\mathrm{Li}} = -\frac{\partial}{\partial x}\left(K^{\mathrm{eff}} \frac{\partial \phi_e}{\partial x}\right) - \frac{\partial}{\partial x}\left(K_D^{\mathrm{eff}} \frac{\partial \ln c_e}{\partial x}\right), \tag{4}$$

$$j^{\mathrm{Li}}(x) = a_s \cdot i_0 \cdot \left[\exp\left(\frac{\alpha_a \cdot F}{R \cdot T} \cdot \eta\right) - \exp\left(-\frac{\alpha_c \cdot F}{R \cdot T} \cdot \eta\right)\right]. \tag{5}$$

These equations are supplemented with appropriate boundary conditions:

$$\left.\frac{\partial c_s}{\partial r}\right|_{r=R_s} = \frac{-j^{\mathrm{Li}}}{F \cdot a_s \cdot D_s}, \quad \left.\frac{\partial c_s}{\partial r}\right|_{r=0} = 0,$$
$$\left.\frac{\partial c_e}{\partial x}\right|_{x=0} = 0, \quad \left.\frac{\partial c_e}{\partial x}\right|_{x=L_t} = 0,$$
$$\sigma\varepsilon_s \left.\frac{\partial \phi_s}{\partial x}\right|_{x=0} = \sigma\varepsilon_s \left.\frac{\partial \phi_s}{\partial x}\right|_{x=L_t} = \frac{-I}{A}$$
$$\left.\frac{\partial \phi_s}{\partial x}\right|_{x=L_n} = \left.\frac{\partial \phi_s}{\partial x}\right|_{x=L_n+L_s} = 0,$$
$$\left.\frac{\partial \phi_e}{\partial x}\right|_{x=0} = 0, \quad \left.\frac{\partial \phi_e}{\partial x}\right|_{x=L_t} = 0$$

In addition, c_s and c_e are known at the initial time. The definition and the terminology used in the variables shown is summarized in Appendix 1.

Equation (1) enforces the lithium conservation in the solid phase. Similarly, Eq. (2) enforces the lithium conservation in the electrolyte phase. In contrast, Eq. (3) refers to the charge conservation in the solid phase. Similarly, Eq. (4) refers to the charge conservation in the electrolyte phase.

Finally, the Butler–Volmer Eq. (5) is used to couple the charge and species governing equations: The four unknowns c_s, c_e, ϕ_s and ϕ_e are coupled in the nonlinear definition of j^{Li}, resulting in a nonlinear problem. In addition, other variables which appear in the equations are defined empirically. These definitions are often nonlinear—such as, for example, the open circuit voltage U—thus adding more difficulties in the resolution.

The overpotential of Eq. (5) is obtained as:

$$\eta(x) = \phi_s(x) - \phi_e(x) - U(c_{se}(x)), \tag{6}$$

where $c_{se} = c_s(r = R_s, x)$ and therefore, the open circuit voltage U depends on the electrode.

In Eq. (5), exchange current density, i_0, is related to both solid surface and electrolyte concentrations according to

$$i_o = k_o \cdot c_e^{\alpha_a} \cdot \left(c_{s,\max} - c_{se}\right)^{\alpha_a} \cdot c_{se}^{\alpha_c}, \tag{7}$$

where k_0 is a kinetic rate constant that depends on the temperature. Its value is often obtained using the initial exchange current density and concentrations.

Finally, the cell potential is computed as:

$$V = \phi_s\left(x = L_t\right) - \phi_s(x = 0) - R_f \cdot I. \tag{8}$$

There are additional secondary equations to solve the model in the definition of some variables. The reader can find a detailed description of these equations in Appendix 2.

It is worth noting that there are as many equations (1) as particles in the x domain. Therefore, the discretization on the electrodes affects the number of equations to solve. This is

caused by the fact that usually there is one particle in each x node in the electrode domains.

If it is taken into account that the just-introduced model is one-dimensional in the cell space, the scheme of the model would be as depicted shown in Fig. 2 where the reader is reminded that there will be as many particles as nodes in x in the anode and cathode domain.

As previously stated, Eq. (1) is solved in the r-dimension of the spherical particles—the pseudo-second dimension—. For an in-depth analysis of this electrochemical model, we recommend the lecture of [20].

2.2 Definitions of the State of Charge (SoC)

We can find a lot of definitions of the SoC in the literature, and each one has its particular use and meaning. Fundamentally, the concept of SoC is related to the transport of lithium from one electrode to the other. A cell with a highly lithiated negative electrode and relatively delithiated positive electrode should correspond to a high SoC, since this implies that the cell has most of its stored charge.

The more extended definition is the bulk SoC. The bulk SoC counts the total lithium entering or exiting the electrode particles. Therefore, knowing the maximum and minimum concentration of lithium in each electrode (related to a correct operation of the cell), we can estimate the bulk SoC. The bulk SoC is computed using the following expression:

$$\mathrm{SoC} = \frac{\frac{c_{s,\mathrm{av}}}{c_{s,\max}} - \theta_{0\%}}{\theta_{100\%} - \theta_{0\%}}, \tag{9}$$

where $\theta_{100\%}$ and $\theta_{0\%}$ are the relation $\frac{c_{se}}{c_{s,\max}}$ on one electrode when the battery is fully charged and completely discharged respectively. $c_{s,\mathrm{av}}$ is the average value of the concentration in the electrode. It is computed as follows (for the anode):

$$c_{s,\mathrm{av}} = \frac{3}{L_n \cdot R_s^3} \int_0^{L_n} \int_0^{R_s} r^2 \cdot c_s(r,x) \cdot dr \cdot dx. \tag{10}$$

The first integral is used to compute the mean value of concentration of each particle and the second computes the mean value of all the particles along the electrode (since the lithium concentration varies spatially in the electrode).

It is important to note that the meaning of SoC depends on the choice of $c_{s,\mathrm{av}}$. For example, the voltage depends directly on the concentration at the particle surface and not on the mean value in the particles. Therefore, we can define the surface SoC redefining $c_{s,\mathrm{av}}$ as follows: The $c_{s,\mathrm{av}}$ is computed using (with respect to the negative electrode):

$$c_{s,\mathrm{av}} = \frac{1}{L_n} \int_{\Omega_n} c_{se}(x) \cdot dx. \tag{11}$$

where Ω_n refers to the domain of the anode.

Furthermore, the surface SoC can also be computed for each spatial position on the electrodes (therefore: $c_{s,\mathrm{av}} = c_s(x, r = R_s)$). This way, the surface SoC would be a function depending on time and space $\mathrm{SoC}(x,t)$. This is an interesting definition to see where depletion and saturation occur on the electrodes.

In addition, the surface SoC has implications in the available power that can be extracted at a given time, since the redox reactions are limited by the reactant concentrations at the particle surface [2]. The bulk and surface SoC converge to the same value at steady state (because at steady state the concentration profiles are stabilized), but it can be useful to track both metrics during dynamic current profiles.

For example, in a high discharge rate, there will be a significant difference between bulk and surface SoC. However the latter has the information about the quantity of lithium instantaneously available to move from one electrode to the other and the quantity of lithium that can be deposed in a given time on an electrode.

Despite the usefulness of Eq. (11), the profile of the surface concentration on the electrodes can have big variations and irregularities depending on the properties of the cell. In these cases, the mean value of Eq. (11) may not be suitable to approximate the dynamic behavior of the SoC in the electrodes.

The reason for this is that the critical points in the electrodes can reach the values of depletion or saturation much sooner than the average value of the electrode concentration. This can cause problems such as a sudden loss of power.

For this reason, control strategies normally try to avoid this [49]. In these cases, we may be interested in the surface SoC at the critical points.

In this case we redefine $c_{s,\mathrm{av}}$ as $c_{s,\mathrm{av}} = c_s(x = x_c, r = R_s)$, where R_s is the particle radius and x_c is the critical point on the electrode. In this way we can, for example, stop charging if the critical point in the electrode is close to a dangerous surface SoC[1]. In this work, we will refer to the above definition as critical SoC and to the definition of Eq. 11 as surface SoC.

Another important issue is the voltage limits of the cell. Usually the operation of the cell has to work between a given range of voltage. Trying to work outside this limits can have undesirable effects. The operational limits of the battery can come from a variety of sources: voltage limits, suitable concentration limits and limits to avoid depletion and saturation effects. In some way, the effects are related, for example the voltage depends on the surface concentration and their distribution on the electrodes as well as the current profile.

[1] The measure to control depletion and saturation on the electrodes is $0 < c_{se}(x)/c_{s,\max} < 1$. On the other hand, the measure to control depletion in the electrolyte is $c_e(x) > 0$ [49].

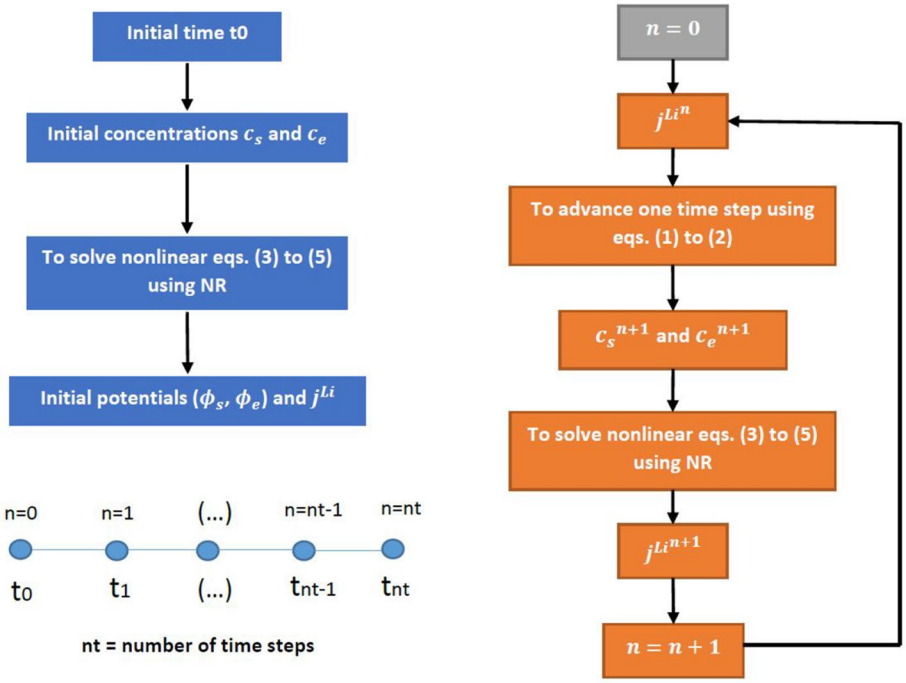

Fig. 4 Scheme to solve the electrochemical model

Finally, sometimes the cell voltage distributions are plotted as a function of the depth of discharge (DoD), a term which is used to describe how deeply the battery is discharged and it can be calculated as

$$\mathrm{DoD} = 1 - \mathrm{SoC}. \tag{12}$$

3 Solving Newman's P2D Model

We remind the reader that one of the goals of this work is to reduce, by means of a reduced-order model, the computational effort and the simulation time of the full-order model, FOM, explained in Sect. 2. Results of the reduced order model (ROM) will therefore be checked against the FOM. This section is devoted to explaining the method used to solve the model. In particular, a strategy to solve Eqs. (1), (2), (3), (4) and (5) is needed, because all equations are coupled and there is not a direct way to compute a solution.

Firstly, it can be noticed that the system of equations is nonlinear, so an algorithm for the solution of nonlinear systems is required. The Newton-Raphson method will be employed for the necessary linearization of these equations.

Secondly, it is noted that Eqs. (1) and (2) are the only ones which allow us to advance in the time domain. This can be accomplished in an implicit or explicit scheme. As a first approximation, and to avoid cumbersome expressions for the Jacobian matrix, only the spatial derivatives of c_s and c_e will be cast in an implicit scheme. The variable j^{Li} will be cast in an explicit way, allowing us to separate the nonlinear problem in Eqs. (3), (4) and (5). This scheme is sketched in Fig. 4.

The just-introduced approach can be further improved. For example, an additional iterative loop can be added (as shown in Fig. 5) to compute j^{Li} in an implicit way. This allows the augmentation of the size of the time steps without the introduction of any perceived instability.

The PDEs (2), (3) and (4) always contain a term of the form $\frac{\partial}{\partial x}\left(k\frac{\partial u}{\partial x}\right)$ where k is a discontinuos function. A finite element discretization is used for these equations, namely, a weak formulation of the problem, to avoid problems in the derivative of a discontinuos function. In contrast, Eq. (1) can be discretized using the FDM (Finite Difference Method), which is easier to implement than FEM.

To compute the value of the functions c_s and c_e at each time step, a Crank–Nicolson scheme is used to discretize Eq. (1) and an implicit Euler scheme for Eq. (2). The reason of using the implicit Euler scheme is its enhanced stability properties. Conversely, the Crank-Nicolson method is used to take advantage of the benefits of the implicit methods as well as an order of accuracy of $\mathcal{O}(\Delta t^2)$.

In the r-domain, two different step sizes are used to adapt the mesh to the shape of the solution. In addition, a finite difference formula of an order of accuracy of $\mathcal{O}(\Delta r^4)$ is used in the area where the function changes faster and abruptly. In the other areas, the classical formulas of an order of accuracy of $\mathcal{O}(\Delta r^2)$ are used.

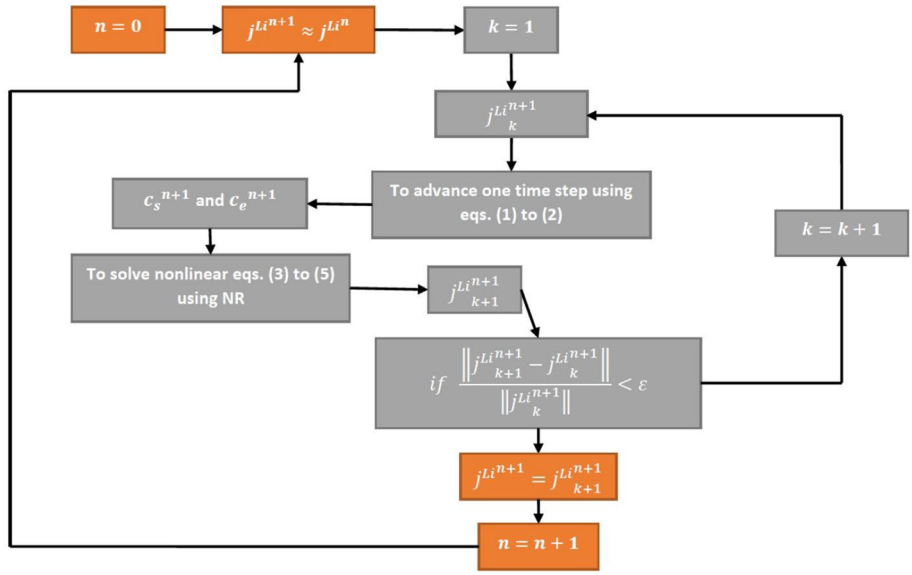

Fig. 5 Improved scheme to solve the electrochemical model

4 Proper Orthogonal Decomposition of the P2D Model

4.1 Reasons to Use the POD

In general, the solution of the model described in Sect. 2 requires a large computational effort in terms of both time and memory. Under such situations, classical methods are not able to provide satisfactory results under the stringent conditions imposed by real-time feedback rates.

Analyzing different batteries, we observed that the parameters of the model can influence significantly the stability and accuracy needed to solve the model and therefore also in the amount of computational resources. For example, some tests to simulate some seconds of battery life usually took times in the order of minutes to compute. Hence, the computational effort has to be reduced several orders of magnitude and to achieve that, Proper Orthogonal Decomposition is proposed [27, 38, 39, 43, 46].

In our simulations, it is observed that roughly 90% of the time is used in solving Eqs. (1). In addition, a big computational effort is needed to save and to compute the time evolution of the $c_s(r)$ functions at each particle. So it is clear that it is necessary to focus on this part of the model.

The problem in Eq. (1) is the large number of nodes along the r-domain needed to achieve an accurate solution without losing stability. This number of nodes has to be multiplied for the number of particles distributed along the x-domain. Therefore, one possible solution to the problem can be to reduce drastically the number of unknowns needed along r. For that purpose, a reduced basis is extracted to project all the possible functions $c_s(r)$. If the cardinality of the basis, $\mathtt{n}$, is small enough, the number of unknowns in Eq. (1) can be greatly reduced:

$$c_s(r) \approx \sum_{i=1}^{\mathtt{n}} \mathbf{F}_i(r) \cdot \alpha_i. \tag{13}$$

Now, if the reduced basis $\mathbf{F}_i(r)$ is known, we require only $\mathtt{n}$ coefficients α_i to determine the function c_s by Galerkin projection. It is important to keep in mind that in a Finite Element basis (for example) we may need hundreds of coefficients (or even more) to determine the function we are evaluating because of the refined mesh used.

Note that Eq. (13) can be expressed in a more compact form:

$$c_s(r) = \begin{bmatrix} \mathbf{F}_1(r) & \cdots & \mathbf{F}_{\mathtt{n}}(r) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_{\mathtt{n}} \end{bmatrix} = \mathbf{F} \cdot \boldsymbol{\alpha}. \tag{14}$$

To obtain the set of reduced basis, the following procedure is applied [8]. Let us consider $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_n] \in \mathbb{R}^{m \times n}$, a given matrix with rank $d \leq \min(m, n)$, where $\mathbf{y}_1, \dots, \mathbf{y}_n$ are column vectors. Further, let $\mathbf{Y} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$ be its singular value decomposition, where $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_m] \in \mathbb{R}^{m \times m}$, $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n] \in \mathbb{R}^{n \times n}$ are orthogonal matrices and the matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times n}$ has the form given by Eq. (16). Then for any $l = \{1, \dots, d\}$ the solution to following constrained optimization problem:

$$\begin{aligned} &\max_{\tilde{\mathbf{u}}_1, \dots, \tilde{\mathbf{u}}_l \in \mathbb{R}^m} \sum_{i=1}^{l} \sum_{j=1}^{n} \left| \langle \mathbf{y}_j, \tilde{\mathbf{u}}_i \rangle_{\mathbb{R}^m} \right|^2, \\ &\text{subject to } \langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j \rangle_{\mathbb{R}^m} = \boldsymbol{\delta}_{ij} \text{ for } 1 \leq i, j \leq l, \end{aligned} \tag{15}$$

with the Kronecker Delta tensor defined as

$$\boldsymbol{\delta}_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j, \end{cases}$$

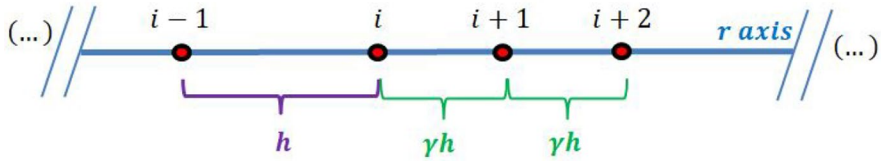

Fig. 6 Reference mesh to obtain the finite difference formula when the spatial step changes

is given by the singular vectors $\{\mathbf{u}_i\}_{i=1}^{l}$, i.e., by the first l columns of $\mathbf{U}$. Moreover, the maximum value of Eq. (15) is given by the sum of the first l singular values of the diagonal matrix $\mathbf{\Sigma}$,

$$\mathbf{\Sigma} = \begin{pmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{m\times n}, \qquad \mathbf{D} = \mathrm{diag}(\sigma_1, \dots, \sigma_d) \in \mathbb{R}^{d\times d}. \tag{16}$$

In Eq. (16), the diagonal entries are sorted in descending order.

The problem (15) consists in approximating all spatial coordinate vectors $\mathbf{y}_i$ of $\mathbf{Y}$ by a linear combination of normalized vectors as well as possible. The constraint of the problem serves to normalize the functions $\mathbf{u}_i$ and thus ensure the uniqueness of the solution. It also adds the property of orthonormality between the l functions which solves the problem.

The choice of l is usually based on heuristic considerations combined with observing the ratio between the modeled energy to the total energy contained in the system $\mathbf{Y}$, which is expressed by:

$$\zeta(l) = \frac{\sum_{i=1}^{l} \lambda_i}{\sum_{i=1}^{d} \lambda_i}. \tag{17}$$

Note also that $\sigma_i^2 = \lambda_i$.

Imagine that a set of experimental values or snapshots of the $c_s(r)$ functions are collected in a matrix $\mathbf{Y}$ during a battery simulation. We can use the SVD to extract a reduced basis $\{\mathbf{u}_i\}_{i=1}^{l}$, which better approximates the $c_s(r)$ functions. If the extracted basis is small enough, we will have a drastic reduction of unknowns.

If the snapshots reflect well all the possible shapes and degrees of freedom of the $c_s(r)$ functions, this reduced basis can be used to perform simulations different from the ones used to extract the reduced basis. The reason for this is that the $c_s(r)$ functions of other scenarios will be well approximated because their projection into the reduced basis is good enough to reproduce them. However, if there is a scenario which produces c_s functions which are not well reproduced by the reduced basis, the accuracy of the ROM will be severely affected.

4.2 POD of the Electrochemical Model

To use the POD methodology, the matrix $\mathbf{Y}$ which contains the snapshots has to be constructed (see Sect. 4.1). Each snapshot is a function $c_s(r)$ for a particle at a given time. However, our experience indicates that only the $c_s(r)$ functions in the particles placed at the boundaries of the electrodes are needed to construct a good reduced basis. In this way we can greatly reduce the size of the $\mathbf{Y}$ matrix to reproduce long-time dynamics. So, from Eq. (1):

$$\frac{\partial c_s}{\partial t} = \frac{D_s}{r^2} \cdot \frac{\partial}{\partial r}\left(r^2 \frac{\partial c_s}{\partial r}\right) = \frac{2 \cdot D_s}{r} \cdot \frac{\partial c_s}{\partial r} + D_s \cdot \frac{\partial^2 c_s}{\partial r^2}.$$

Using a Crank–Nicolson scheme:

$$\frac{c_s^{n+1} - c_s^n}{\Delta t} = \frac{1}{2} D_s \cdot \left(\frac{2}{r} \cdot \frac{\partial c_s^{n+1}}{\partial r} + \frac{\partial^2 c_s^{n+1}}{\partial r^2} + \frac{2}{r} \cdot \frac{\partial c_s^n}{\partial r} + \frac{\partial^2 c_s^n}{\partial r^2}\right). \tag{18}$$

By further discretizing Eq. (18) in space, we obtain the following algebraic expression:

$$\mathbf{L} \cdot \mathbf{c}_s^{n+1} = \mathbf{J} \cdot \mathbf{c}_s^n + \frac{1}{2}\left(\mathbf{b}^{n+1} + \mathbf{b}^n\right) = \mathbf{J} \cdot \mathbf{c}_s^n + \mathbf{p}(\mathbf{b}^{n+1}, \mathbf{b}^n). \tag{19}$$

where $\mathbf{p}$ is a vector which depends on the boundaries $(\mathbf{b}^{n+1}, \mathbf{b}^n)$, namely, the $j^{\mathrm{Li}}(x)$ in the n- and $n+1$-th time steps, according to the following finite difference discretization.

The matrices in Eq. (19) $(\mathbf{L}, \mathbf{J})$ are obtained by using a Finite Difference scheme of order $\mathcal{O}(\Delta r^4)$ between $R_s > r > 0.91 \cdot R_s$ and of order $\mathcal{O}(\Delta r^2)$ between $0.91 \cdot R_s > r > 0$ [31]. Two different Δr are used. A larger one (Δr_1) between $0.89 \cdot R_s > r > 0$ and a smaller one (Δr_2) between $R_s > r > 0.89 \cdot R_s$. The values of $\Delta r_1, \Delta r_2$ depend on the parameters of the cell analyzed.

To obtain the second-derivative formula when changing from Δr_1 to Δr_2, we look for $\frac{d^2 u_i}{dr^2}$ as in Fig. 6. In addition, the points $i-1, i, i+1$ and $i+2$ are the ones which will be used in the expression searched according to Fig. 6 where h is defined as Δr_1 and $\gamma \cdot h$ as Δr_2.

Given a smooth function $f : \mathbb{R} \to \mathbb{R}$, its Taylor series around a is

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(a)\,(x-a)^n}{n!}.$$

Therefore, according to the Taylor series, the following system of equations can be found:

$$\begin{pmatrix} u_{i-1} \\ u_{i+1} \\ u_{i+2} \end{pmatrix} = u_i \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + h \cdot \frac{du_i}{dr} \cdot \begin{pmatrix} -1 \\ \gamma \\ 2 \cdot \gamma \end{pmatrix} + h^2/2 \cdot \frac{du_i^2}{dr^2} \cdot \begin{pmatrix} 1 \\ \gamma^2 \\ 4 \cdot \gamma^2 \end{pmatrix} + h^3/6 \cdot \frac{du_i^3}{dr^3} \cdot \begin{pmatrix} -1 \\ \gamma^3 \\ 8 \cdot \gamma^3 \end{pmatrix} + h^4/24 \cdot \frac{du_i^4}{dr^4} \cdot \begin{pmatrix} 1 \\ \gamma^4 \\ 16 \cdot \gamma^4 \end{pmatrix} + \cdots$$

Now, we form a weighted sum of the equations with the coefficients 1, A, B:

$$\begin{aligned} u_{i-1} + A \cdot u_{i+1} + B \cdot u_{i+2} &= u_i \cdot (1 + A + B) \\ &+ h \cdot \frac{du_i}{dr} \cdot (2 \cdot B \cdot \gamma + A \cdot \gamma - 1) \\ &+ h^2/2 \cdot \frac{du_i^2}{dr^2} \cdot (4 \cdot B \cdot \gamma^2 + A \cdot \gamma^2 + 1) \\ &+ h^3/6 \cdot \frac{du_i^3}{dr^3} \cdot (8 \cdot B \cdot \gamma^3 + A \cdot \gamma^3 - 1) \quad + \quad \cdots \end{aligned}$$

removing the addends related to the third and first derivatives gives:

$$\begin{aligned} 2 \cdot B \cdot \gamma + A \cdot \gamma &= 1 \\ 8 \cdot B \cdot \gamma^3 + A \cdot \gamma^3 &= 1 \end{aligned}$$

Solving the system:

$$B = \frac{1 - \gamma^2}{6 \cdot \gamma^3}; \quad A = \frac{1 - 2 \cdot B \cdot \gamma}{\gamma}.$$

So, finally the sought expression is:

$$\frac{d^2 u_i}{dr^2} = 2 \cdot \frac{B \cdot u_{i+2} + A \cdot u_{i+1} - u_i \cdot (1 + A + B) + u_{i-1}}{h^2 \cdot (4 \cdot B \cdot \gamma^2 + A \cdot \gamma^2 + 1)} + \mathcal{O}(h^2). \tag{20}$$

Returning to Eq. (19), adding the terminology introduced in Eq. (14), we change Eq. (19) to introduce the reduced basis in the formulation of the problem as well as the new unknowns.

$$\mathbf{L} \cdot \mathbf{F} \cdot \boldsymbol{\alpha}^{n+1} = \mathbf{J} \cdot \mathbf{F} \cdot \boldsymbol{\alpha}^n + \mathbf{p}(\mathbf{b}^{n+1}, \mathbf{b}^n) \tag{21}$$

There are different strategies to solve the overdetermined system in Eq. (21). We choose Ordinary Least Squares. This method minimizes the sum of squared residuals, and leads to a closed-form expression for the estimated value of the unknown vector. Therefore, the solution can be written as:

$$\begin{aligned} \mathbf{L}_F &= \mathbf{L} \cdot F, \quad \mathbf{J}_F = \mathbf{J} \cdot F, \\ \mathbf{R} &= \mathbf{L}_F^\top \cdot \mathbf{L}_F, \quad \mathbf{M} = \mathbf{L}_F^\top \cdot \mathbf{J}_F, \\ \boldsymbol{\alpha}^{n+1} &= \mathbf{R}^{-1} \cdot \mathbf{L}_F^\top \cdot \left(\mathbf{J}_F \cdot \boldsymbol{\alpha}^n + \mathbf{p}(\mathbf{b}^{n+1}, \mathbf{b}^n)\right), \\ \boldsymbol{\alpha}^{n+1} &= \mathbf{R}^{-1} \cdot \left(\mathbf{M} \cdot \boldsymbol{\alpha}^n + \mathbf{L}_F^\top \cdot \mathbf{p}(\mathbf{b}^{n+1}, \mathbf{b}^n)\right). \end{aligned} \tag{22}$$

4.3 Results

To validate the robustness of the proposed scheme, two different batteries are studied, each one with different parameters. The parameters of the first cell are taken from [11, 48]. This is a 6 Ah cell, designed for its application to Hybrid Electric Vehicles (HEV). For this battery, a POD basis is extracted using a constant 12 A discharge current. To check the accuracy of the reduced model, a comparison between the POD model and the original, full-order model is done. Results of this comparison are detailed in Fig. 7, where a validation is done with a current profile according to the Freedom CAR test procedure, a U.S. Department of Energy program for the zero-emission vehicle and technology Research.

Other current profiles provided similar results, thus proving the validity of the approach. On the other hand, we achieve a reduction factor of 50 in the number of unknowns in this cell by using the POD basis. The simulation time and the computational effort to solve the problem is reduced by this factor approximately—note that roughly 90% of the time was used to solve Eq. (1).

The second cell parameters are taken from [53]. This type of cell is suitable for both HEV and EV applications, as we can notice in the types of simulations done in these references. The main parameters of this cell are detailed in Appendix 2.

For this second battery, a POD basis is extracted using a constant current discharge of 60 A. Figure 8 shows a comparison between the POD model and the original full order model. This figure shows the accuracy for the POD model to perform simulations with current profiles different from the one used to extract the basis. We remind ourselves that the basis is extracted with a constant current profile, while in the simulation of Fig. 8, the current used has a linear profile between different points. Despite the current differences, a good accuracy is achieved by the POD model as we can see in the comparison.

The reduced-order results are also compared to those provided by the software LIONSIMBA [53]. These results are shown in Fig. 9.

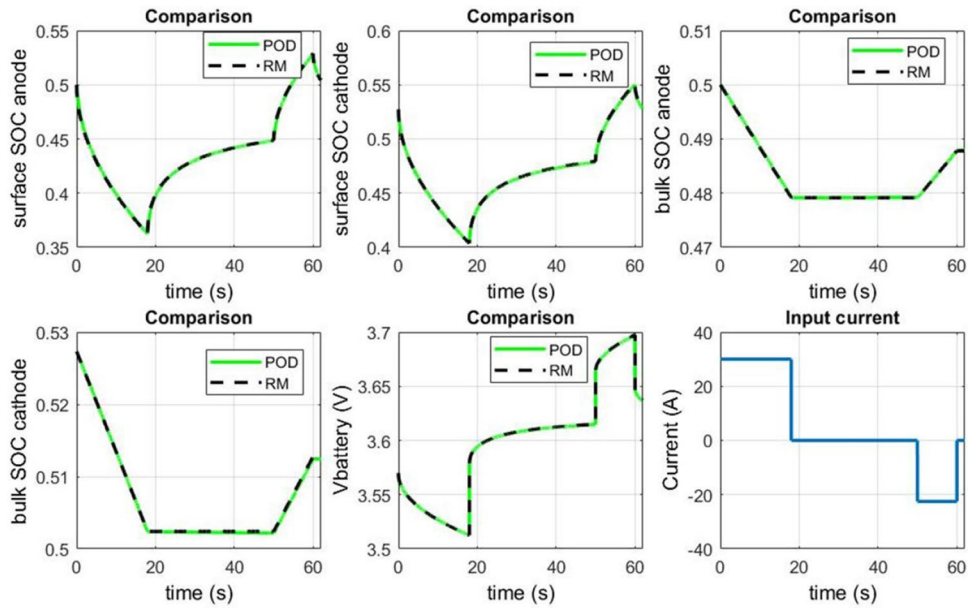

Fig. 7 Simulation of the reduced order model for the first cell. Comparison of the evolution in time between the POD model and the original, full-order electrochemical mode, of the different variables of the cell. The different SoC and cell voltages are compared and the input current used during the simulation is shown in the last picture. Legend: RM = Results of the full P2D model; POD = Results of the POD model

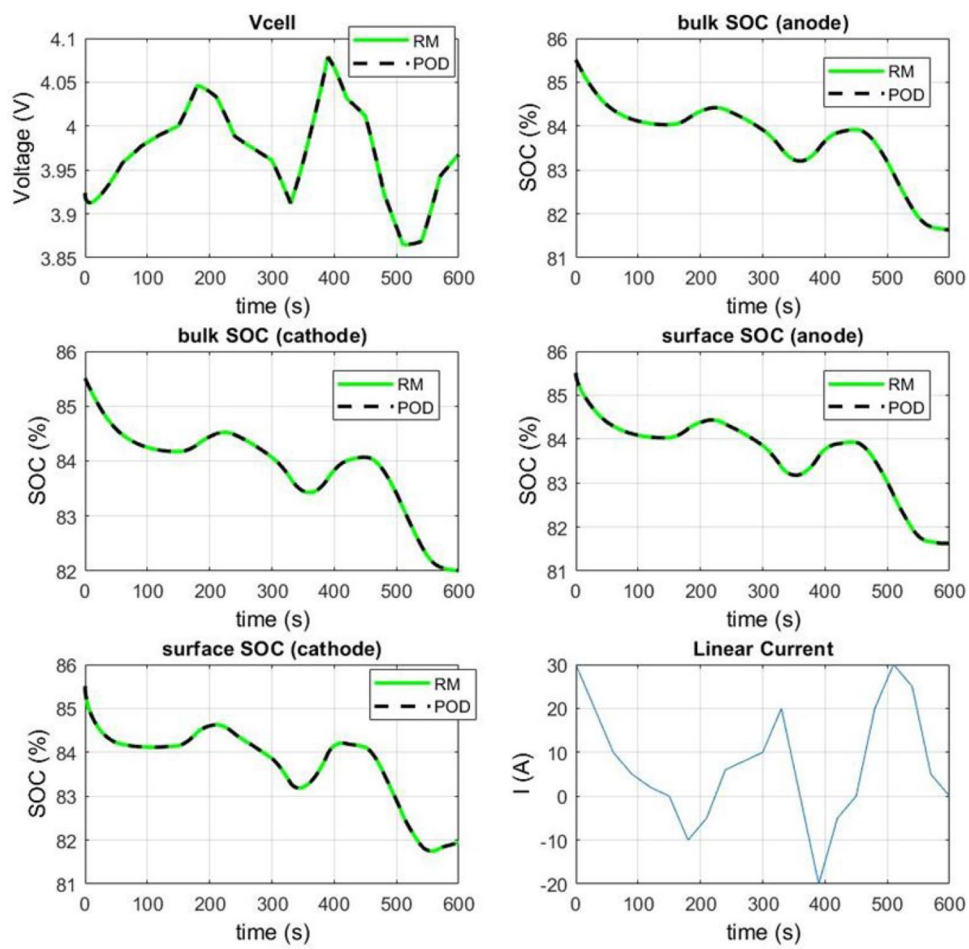

Fig. 8 Comparison of full- and reducer-order simulations of the second cell. The different SoC and the cell voltage are compared and the input current used during the simulation is shown in the last picture. Legend: RM = Results of the full P2D model; POD = Results of the POD model

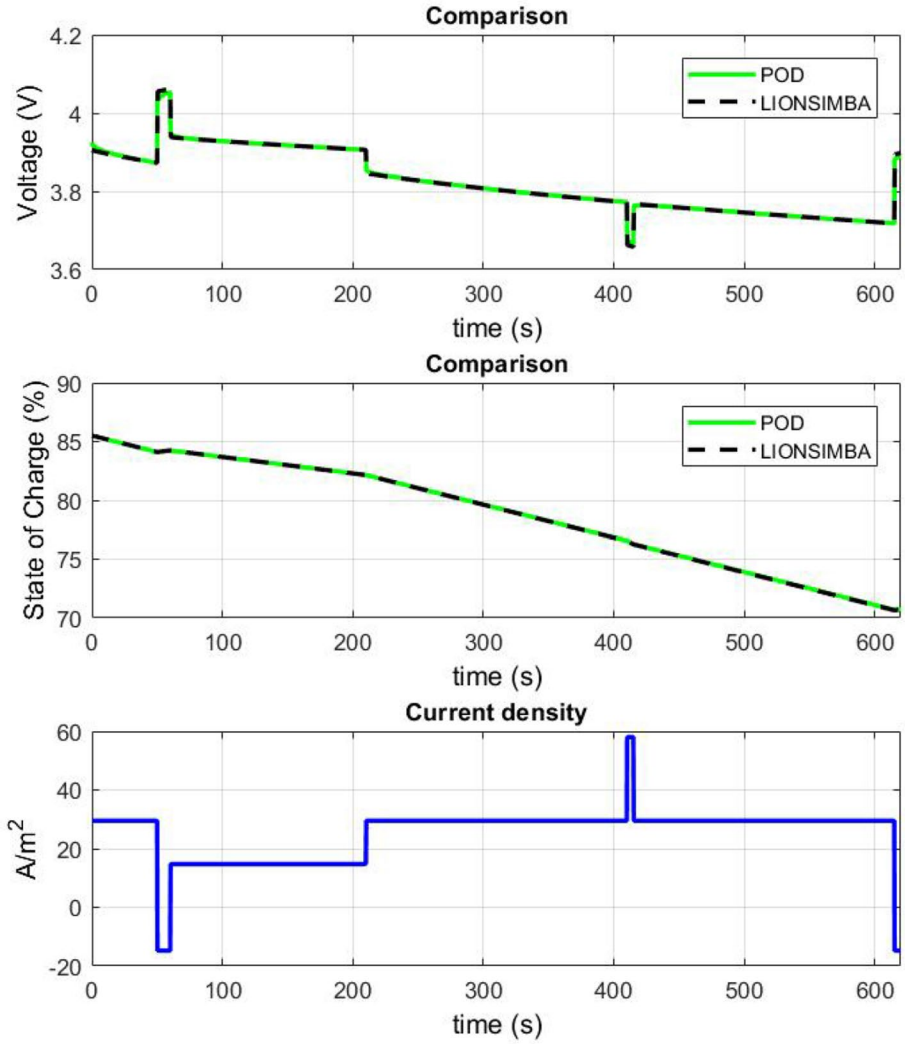

Fig. 9 Comparison of the reduced-order results with those provided by the software LIONSIMBA for the second cell

In addition to obtaining a good comparison in terms of results, we achieve a reduction factor of 40 in the number of unknowns in this cell using the POD basis. The simulation time and the computational effort to solve the problem is reduced by this factor approximately. Both cells are simulated in real-time (in other words, the time for a simulation of an interval t is less or equal than the elapsed time t). These results were tested on a laptop running an Intel Core Kabylake i7-7700HQ+HM175, with 8GB DDR4 RAM running at 2400MHz.

Based on the experience of our simulations, we conclude that the key issue to obtaining a good reduced basis is to collect the $c_s(r)$ functions from an off-line simulation of a complete discharge-charge cycle at constant current. As we show in Fig. 8, this basis is good enough to reproduce other current profiles with good accuracy.

Another important consideration is the value of the constant current used to extract the basis. Normally, a high value will produce more variations in the c_s functions, which will better enrich the reduced basis. However, in this case, the simulation will typically stop sooner due to, for example, reaching the cut-off voltage.

In these circumstances, the depth of the diffusion in the particles can be smaller because of the shorter simulation time. This provides a dilemma for selecting the appropriate value.

We remind the reader that the cut-off voltage is one of the limits imposed on the cell, as explained in Sect. 2.2, to ensure a suitable operation of the component. In this case, the cut-off voltage is referred to the limit assigned to the lowest value of the cell voltage.

5 PGD Regression: Learning the cell Behavior from Data

5.1 Introduction

In Sect. 4, a ROM is developed to allow fast simulations of the electrochemical model using less computational resources. As we saw, excellent agreement with existing, full-order models are obtained. However, the POD model is not necessarily the best option to achieve a full simulation of all the systems which constitute the EV or indeed to be used in a BMS. For this reason, we explored several alternatives, leading us to the approach presented in this section.

This method allows us to easily couple the battery model with the other subsystems of the vehicle. It can likewise be used in a BMS without issue because of the simple algebraic expression(s) obtained.

Furthermore, this PGD approach could be developed, in which batteries can be modeled from data. One motivation for the use of the latter approach is that, by working directly with data, no parameter fitting is necessary.

The motivation of such an approach is that by working directly with data, no parameter fitting will be necessary. This approach has been tested successfully in other branches of scientific computing, see, for instance, [5, 18, 26, 29, 30, 42], to name but a few. This methodology is specially convenient if it is expected that, for some reason, the model for the battery will change during its operating life, for instance.

As presented in Sect. 3, the response of the cell depends on the micro-scale, where many parameters are involved in the description of its dynamics. If a response surface-like approach is chosen under this setting, a large sampling effort is to be expected, that grows exponentially with the number of parameters. For this reason, a method able to circumvent the curse of dimensionality should be used [36]. Here, we employ the so-called sparse Proper Generalized Decomposition (s-PGD in what follows). In Sect. 5.2 an introduction to the s-PGD methodology is presented. The constructed model is explained in Sect. 5.3. Finally, results are presented in Sect. 5.4.

5.2 s-PGD Methodology

The reader unfamiliar with the standard Proper Generalized Decomposition (PGD) method for solving PDEs, is directed to review works such as [9, 10]. Here, let us consider an unknown function $f(s^1, \dots, s^{n_d})$, which depends on n_d different variables or dimensions s^k, $k = 1, \dots, n_d$. For the battery problem:

$$f : \Omega_1 \subset \mathbb{R}^{n_d} \to \mathbb{R}.$$

The s-PGD approach tries to approximate the function f using a separated (tensor) representation in each dimension. As in standard PGD procedures, it expresses the function f using a sum of products of one-dimensional functions along each dimension. Each sum is called a mode.

In the context of regression problems, the goal is to find a function $\tilde{f}$ which minimizes the distance to the sought function

$$\min \sum_{i=1}^{n_t} \left\| f(\mathbf{s}_i) - \tilde{f}(\mathbf{s}_i) \right\|, \tag{23}$$

and that takes the form

$$\tilde{f}(s^1, \dots, s^{n_d}) = \sum_{m=1}^{M} \prod_{k=1}^{n_d} \sigma_m^k(s^k), \tag{24}$$

where M is the number of modes and σ_m^k are the one-dimensional functions of the mode m and dimension k. n_t is the number of sampling points to perform the regression and $\mathbf{s}_i$ are the different vectors which contain the data points of the training set. $\|\cdot\|$ is the chosen norm to measure the distance between two points.

The other objective is that the function $\tilde{f}$ has to perform as well in the training set as in other possible sets.

This second goal is more difficult to achieve, yet is more important because this evaluates the predictive ability of the model $\tilde{f}$.

Achieving this is particularly difficult when confronted with a high-dimensional problem, which provides sparse data.

The s-PGD methodology is based on creating a function $\tilde{f}$ with the form of (24). Then, the functions $\{\sigma_m^k\}_{m=1}^M$ for each k are formed by a linear combination of a set of basis functions:

$$\sigma_m^k(s^k) = \sum_{j=1}^{D} N_j(s^k) \cdot \alpha_j = (\mathbf{N}_m^k)^\top \cdot \mathbf{a}_m^k, \tag{25}$$

where D represents the degrees of freedom of the chosen approximation. In addition, $\mathbf{N}_m^k$ is a column vector with the set of basis functions for the k dimension and the m-th mode and $\mathbf{a}_m^k$ is a column vector with the coefficients for the k dimension and the m-th mode. The important issue here is to know which set of basis functions are best suited for the problem at hand. For example, a Fourier basis or a polynomial basis can be selected.

The determination of the coefficients in each one-dimensional function for each mode $m = 1, \dots, M$ is done by employing a greedy algorithm such that, once the approximation up to order $M - 1$ is known, the new M-th order term is found using a non-linear solver (Picard, Newton, for instance):

$$\tilde{f}^M = \sum_{m=1}^{M-1} \prod_{k=1}^{n_d} \sigma_m^k(s^k) + \prod_{k=1}^{n_d} \sigma_M^k(s^k). \tag{26}$$

A standard choice is to select the same basis for each one of the modes:

$$\mathbf{N}_1^k = \mathbf{N}_2^k = \cdots = \mathbf{N}_M^k, \text{ for } k = 1, \ldots, n_d. \tag{27}$$

This choice may seem reasonable at first glance, however it may not be appropriate when dealing with non-structured sparse data.

It is known that the cardinality of the interpolation basis must not exceed the maximum rank provided by the training set. Indeed, this constraint, which provides an upper bound to build the interpolation basis, only guarantees that the minimization is satisfied by the training set, without saying anything of the other measured points. Hence, if there is not an abundance of sampling points in the training set, in the limit of low-data regime, high oscillations may appear out of these measured points. Usually, this is an undesirable effect because it affects the predictive ability of the constructed regression model.

In order to tackle this problem, the s-PGD uses the Modal Adaptivity Strategy (MAS) to take advantage of the greedy PGD algorithm. The idea is to minimize spurious oscillations out of the training set by starting the PGD algorithm looking for modes with relatively smooth bases. When it is observed that the residual decreases slowly or stagnates, a high-order basis can be used. By doing this, oscillations are reduced, since a higher-order basis will try to capture only what remains in the residual.

To deploy the strategy, an indicator and a stopping criteria have to be defined. Many different strategies can be envisaged. Here, we employ that defined in reference [25], where the methodology of the s-PGD is explained in more depth. Following that reference, the following norm is used for the PGD residual in the present work:

$$\mathcal{R}_T^M = \frac{1}{\sqrt{n_t}} \sqrt{\sum_{i=1}^{n_t} \left(f(\mathbf{s}_i) - \tilde{f}^M(\mathbf{s}_i)\right)^2} \tag{28}$$

where $\mathcal{R}_T^M$ is the residual of the PGD solution of M modes in the set T (training set) and $\tilde{f}^M$ is the PGD solution composed of M modes.

Then, for each $\tilde{f}^M$, we define

$$\Delta\mathcal{R}_T^M = \mathcal{R}_T^M - \mathcal{R}_T^{M-1} < \epsilon_r, \tag{29}$$

where ϵ_r is a tolerance defining the resilience of the s-PGD to increase the cardinality of the interpolation basis. The lower ϵ_r is, the more resilient the method is to increase the cardinality.

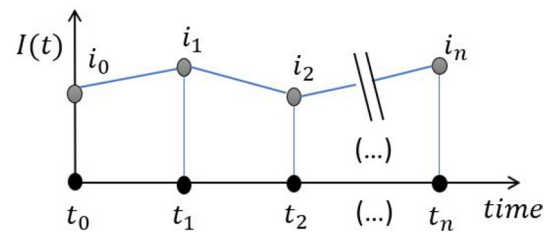

Fig. 10 Construction of a piece-wise linear intensity as a function of time

5.3 Proposed Model

This work is focused on the battery application in the EV. In this application, our main objective is to construct a regression model able to provide the response of the cell/battery for an arbitrary itinerary of the EV.

The itinerary depends on the initial State of Charge, the loading and the environmental conditions. Therefore, the goal is to build a mathematical model which give us functions f_1 and f_2 as follows:

$$\begin{aligned} \text{SoC}(t) &= f_1(\text{SoC}(t=0), \text{Load}, \text{Environment}) \\ V(t) &= f_2(\text{SoC}(t=0), \text{Load}, \text{Environment}). \end{aligned} \tag{30}$$

There are two immediate issues in developing the model in Eq. (30). The first is the high-dimensional problem of defining such an itinerary. The second is that the electrochemical model depends on a micro state. This implies that the dynamical evolution of the response depends on the time evolution of a large number of variables.

Considering the isothermal model of the cell and neglecting its time evolution, the load and the environment of an itinerary can be translated as the $I(t)$ demanded by the battery. In this way, the different itineraries can be modeled as different $I(t)$ for a given initial SoC.

It follows that a possible way to obtain a model by data regression is to parameterize the current instead. To this end, a piece-wise linear intensity is assumed and constructed as a function of the value of the current in some nodes as sketched Fig. 10. The problem is that this approach forces us to work in a high-dimensional space with potentially hundreds of parameters for a one-hour itinerary. This idea was therefore discarded.

Another possibility consists in parameterizing an interval of time as a function of a given initial condition and a final current at the end of the time interval. Once this interval is parameterized, the only thing we have to do to complete a simulation is to join the solutions of each interval.

The problem here is due to the micro-scale modeling of the cell. This means that we will need to follow the

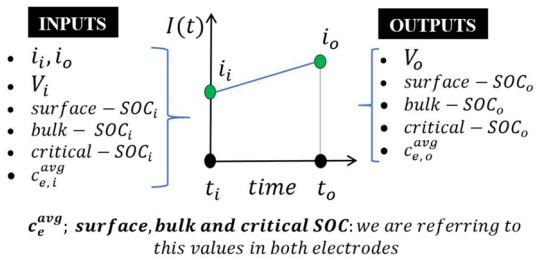

Fig. 11 Model of the parameterized interval. One regression is performed for each output. We will take for each output, the inputs which give us the best results. Here, subscript i refers to input values, while subscript o means output. Variable i refers to the current and the variable V refers to the voltage

micro-scale evolution of parameters such as $c_s(r, x, t)$ and $c_e(x, t)$ to compute and follow the time evolution of macroscopic parameters such as $V(t)$ and the $SoC(t)$. Therefore, this second idea produces hundreds of inputs or outputs on the modeled interval.

Nevertheless, we can consider analytical means to reduce the number of input and output parameters to make this second idea feasible. Sometimes, variables at the macro scale, which depend on a high number of parameters in the micro-scale, can be reduced to a simple expression. In this way, we use macroscopic parameters (surface SoC, bulk SoC and voltage) to reproduce the battery response, assuming for now that the system will be sufficiently described by these variables. This is not a random choice. The bulk SoC, the surface SoC and the voltage for a given intensity depend on the dynamics of the micro-scale parameters c_s and c_e.

We extend this approach by predicting the micro-scale parameters of interest using only the above-mentioned macro-scale variables. Concretely, depletion and saturation are predicted adding the critical SoC defined in Sect. 2.2. Furthermore, the average electrolyte concentration at each electrode, $c_{e,\text{avg}}$, is added into the s-PGD model. In this way, Li^+ depletion in the electrolyte is controlled.

To summarize, the s-PGD model retains only field variables relevant to estimation and control applications in the micro and macro scale. The great advantage of employing this approach is that the s-PGD model can predict the time evolution in all the scales observing only a few macro-scale variables. Proofs of this assertion are included in Sect. 5.4.

In Fig. 11, we sketch the proposed model. In this model, a time interval $\Delta t = t_o - t_i$ is parameterized as a function of given initial conditions at the beginning of the interval and the final current at the end of the interval.

A different PGD regression model will be constructed for each output. This model provides us with a forecast of the outputs Δt seconds in the future, taking the corresponding inputs into the regression. Note that it is not necessary to take all the inputs shown in Fig. 11 to perform the regression. For each output, we take the inputs which give the best results.

A polynomial basis is selected to use the s-PGD explained in Sect. 5.2 with the electrochemical model. The motivation for this choice is the requirement for a simple model that can be integrated in the SimulationX model of the EV.

Therefore, the expression of the output voltage as a function of all the relevant parameters will take the form

$$V_o = g(s^1, \dots, s^6) = \sum_{m=1}^{M} \prod_{k=1}^{6} \sigma_m^k(s^k) \tag{31}$$

where:

$s^1 = i_i, \quad s^2 = i_o,$

$s^3 =$ surface SoC (anode)

$s^4 =$ surface SoC (cathode), $\quad s^5 =$ bulk SoC (anode),

$s^6 =$ bulk SoC (cathode),

and σ_m^k are the one-dimensional functions constructed with different polynomial bases according to the Modal Adaptivity Strategy (MAS) explained in Section. 5.2.

5.4 Results

In this section, we analyze an example cell described in [53]. Its parameters are detailed in Appendix 2.

As already described in Sect. 5.3, polynomial bases are used for the one-dimensional functions of the s-PGD regression introduced in Sect. 5.2. The MAS is used, starting with low-degree approximations and finishing with higher-degree approximations.

The surrogate input to describe an itinerary, namely $I(t)$, is considered to be polygonal within each Δt. This assumption is made because, in principle, Δt is small enough to consider that the current has a linear profile between any two consecutive time instants.

We observe that the method chosen to extract and to select the data for the training set has a significant impact on the regression results. The regression is done performed on sparse data, because of the high dimensionality of the problem. For this reason, the sampling strategy is critical to ensuring the good predictive capabilities in the model.

A simple way to obtain a good training set is to consider the current nodes of the itinerary $I(t)$ (defined as in Fig. 10) as different parameters (dimensions $i_0, \dots, i_n$) of the problem. We remind ourselves that the real dimensions of the model proposed are the ones shown in Fig. 11, but for the sampling we choose to use $i_0, \dots, i_n$. The reason for this choice is that it is not possible to reconstruct the whole micro-scale state with the few dimensions which are chosen

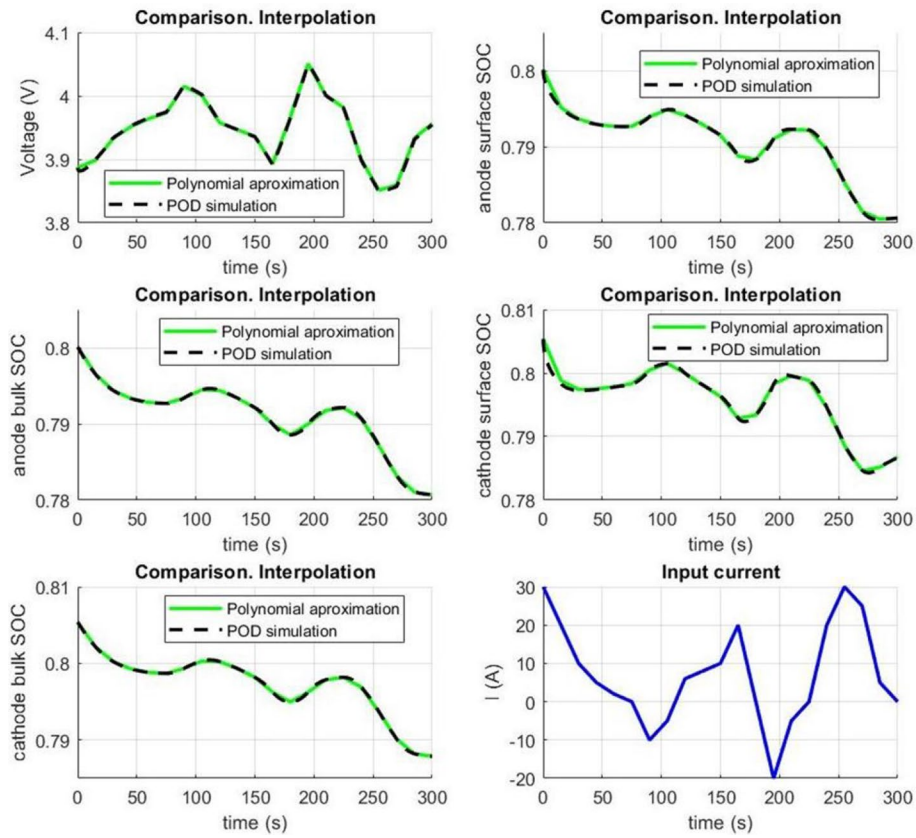

Fig. 12 Comparison of the results between the s-PGD model and the POD model in a 5 min itinerary. The s-PGD model is extracted based on itineraries of 5 min. $\Delta t = 15s$

in Fig. 11. The whole micro-scale state is needed to run the model to obtain the training set.

In order to obtain different $I(t)$ values, we use the Latin Hypercube Sampling (LHS) method in the dimensions $i_0, \ldots, i_n$. The sampling process described is repeated for different initial SoC of the cell. Then, when sampling is finished and the itineraries of the the sampling are simulated, the different results of the variables of interest (SoC, V, ...) are split into different intervals of size Δt to perform the regression via the s-PGD.

One way to improve the sampling described above is the following. The LHS which is used in this work (Matlab function `lhsdesign()`) produces normally oscillating $I(t)$ functions. It is convenient to enrich the training set by e.g. adding some $I(t)$ with constant current profiles or formed of 2-4 step functions. In this way, the regression is more adapted to current profiles representative of a constant discharging or charging of the battery and it can take into consideration this type of behavior that otherwise would be underestimated. Furthermore, in doing this, the training set is more adapted to simulations that attain greater depths of discharge (because, for example, some high constant current profiles are added in the training set).

In the results that follow, we obtained a regression using a sampling composed of 5-min itineraries ($\Delta t = 15$ s is used for the regression). These results are shown in Figs. 12, 13, 14 and 15. This regression turns out to be very accurate for itineraries of 5 min, where the relative error is always under 1% in all the tests.

This regression model can be used for itineraries longer than 5 min, but the relative error starts to increase up to 1%. In Figs. 14 and 15 we can see that the error can be acceptable for 10 min itineraries.

The inputs used for the regression of each output were:

$$
\begin{array}{lll}
V_o: & s^1 = i_o - i_i & s^2 = \frac{i_o + i_i}{2} \\
 & s^3 = \text{surface SoC (anode)} & s^4 = \text{bulk SoC (anode)} \\
 & s^5 = V_i & \\
\text{s-SoC}: & s^1 = i_i & s^2 = i_o \\
 & s^3 = \text{surface SoC} & s^4 = \text{bulk SoC} \\
 & s^5 = V_i & \\
\text{b-SoC}: & s^1 = i_i & s^2 = i_o \\
 & s^3 = \text{bulk SoC} &
\end{array}
$$

where the s-SoC and the b-SoC refer to the surface SoC and bulk SoC respectively.

The MAS explained in Sect. 5.2 is used to determine the higher-degree approximations used. The results to apply the MAS are:

1. For the surface and bulk SoC, only polynomial basis of degree one (thus, linear) are used.

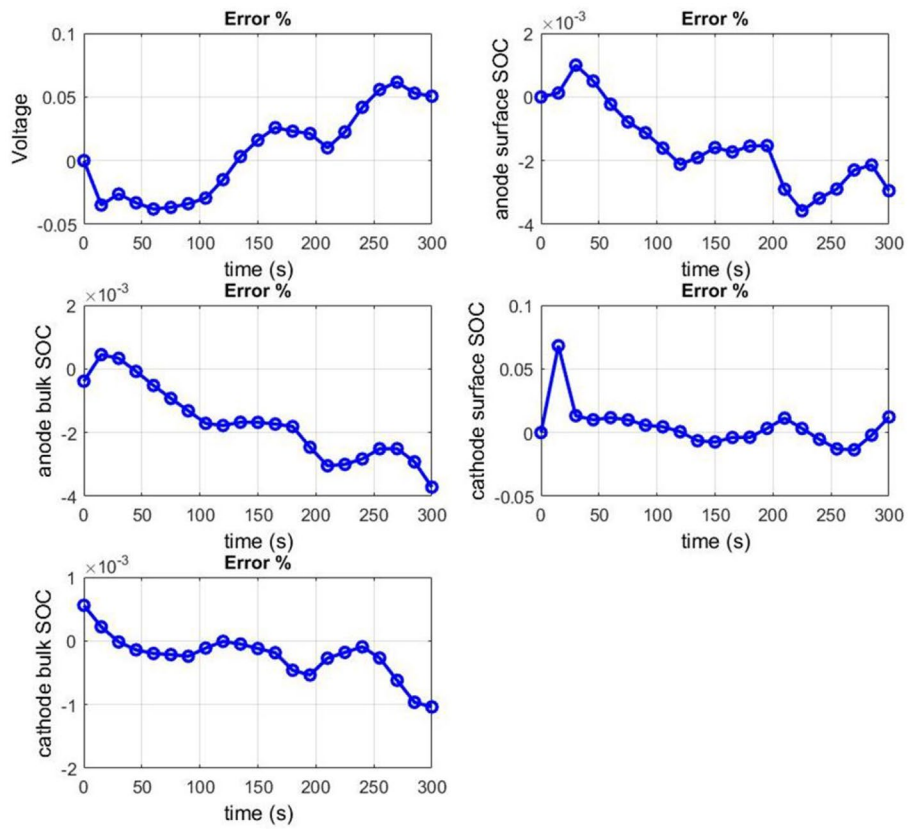

Fig. 13 Error between the s-PGD model and the POD model in a 5 min itinerary. The s-PGD model is extracted based on itineraries of 5 min. $\Delta t = 15s$

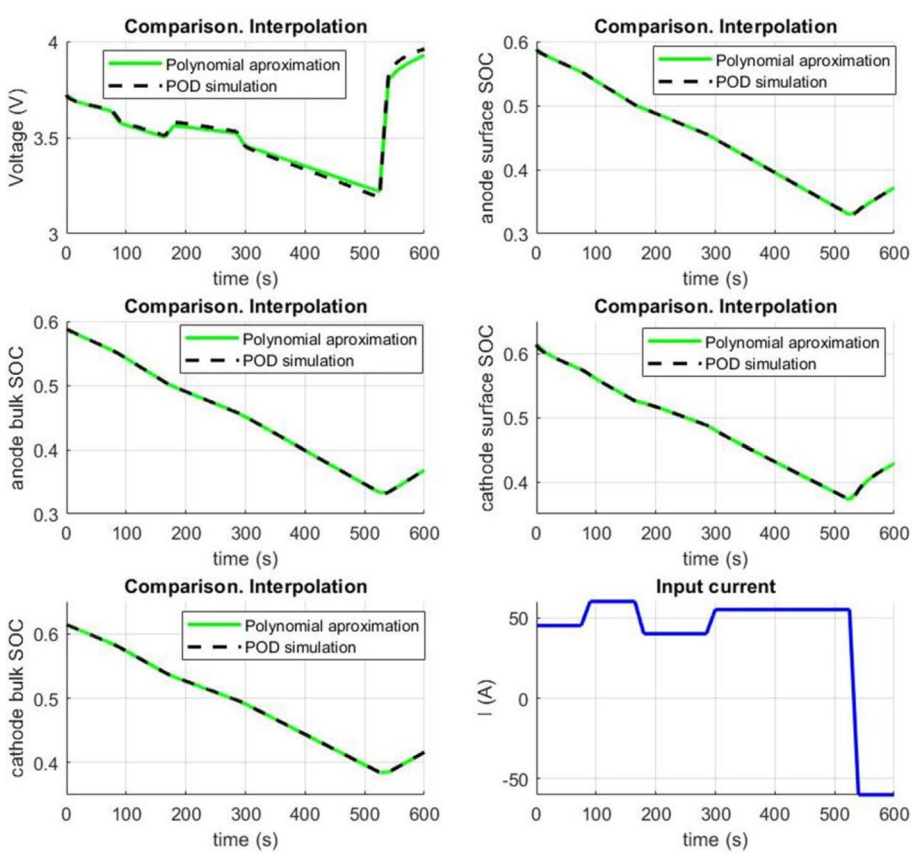

Fig. 14 Comparison of the results between the s-PGD model and the POD model in a 10 min itinerary. The s-PGD model is extracted based on itineraries of 5 min. $\Delta t = 15s$

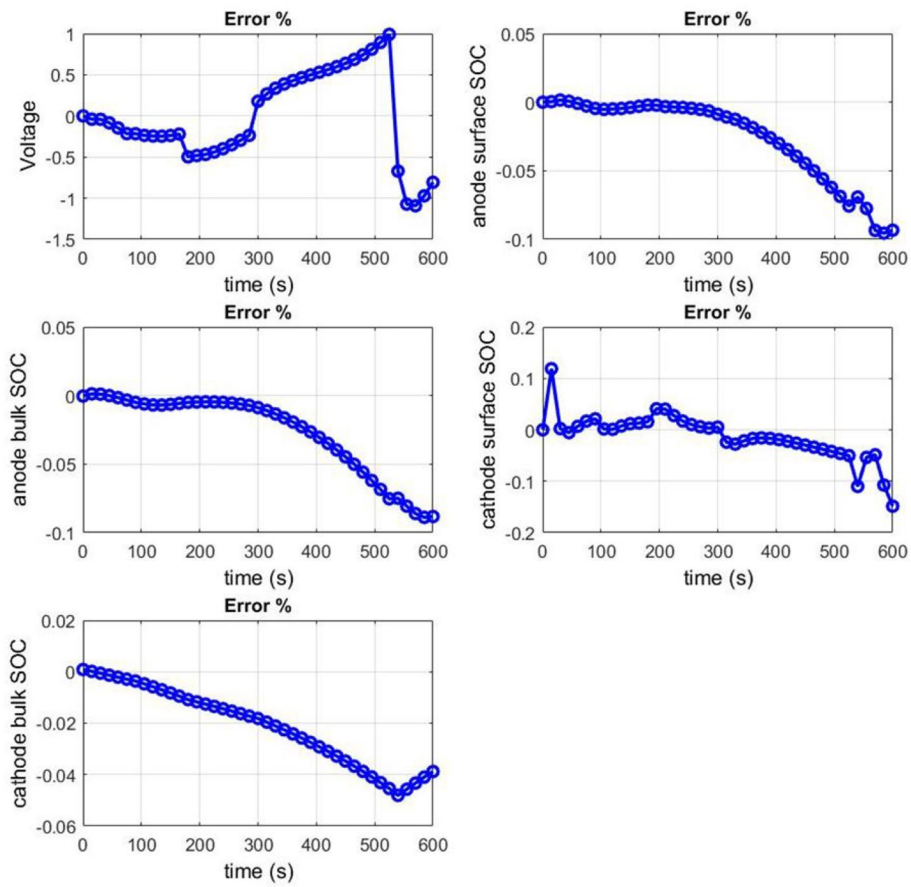

Fig. 15 Error between the s-PGD model and the POD model in a 10 min itinerary. The s-PGD model is extracted based on itineraries of 5 min. $\Delta t = 15s$

2. For the Voltage: a first group of modes is created with one-dimensional functions based on polynomial basis of degree one. Then, a second group of modes is created with one-dimensional functions based on polynomial bases of degree two.

This methodology can also be used with longer trajectories or smaller Δt. Consider, for instance, a regression obtained with a sampling composed of 10-min itineraries ($\Delta t = 5$ s is used for this regression). The results are shown in Figs. 16 and 17. This regression is very accurate to compute 10-min itineraries, where the relative error is under 1% in almost all the checking set of data points.

Trajectories longer than 10 min can also be computed. While the error will naturally increase, it can be acceptable if the difference between T_{sim} and T_{reg} is not too large. Note that T_{sim} is the duration of the simulated itinerary, while T_{reg} is the equivalent for the sampling itineraries for the PGD regression.

The results for the regression shown in Figs. 16 and 17 are

1. For the surface and bulk SoC, only polynomial bases of degree one are used.
2. For the voltage, a first group of modes is created with one-dimensional functions based on polynomial bases of degree one. Then, a second group of modes is created with one-dimensional functions based on polynomial bases of degree two. Finally, a third group of modes is created with one-dimensional functions based on polynomial bases of degree three.

To summarize, the results of using the s-PGD in the critical SoC and in the c_e^{avg} are shown according to Sect. 5.3 (where their importance was discussed). Here, $\Delta t = 2s$ is used. The values are with respect to the negative electrode.

Results are shown in Figs. 18 and 19. These regressions are intended to reproduce 10-min itineraries with an initial SoC of 85%. We can observe that we can reproduce the results with a high degree of accuracy. Inputs used for these regressions are the same as those used in the voltage regression with $\Delta t = 15s$ (but using c_e^{avg} or the critical SoC instead for the voltage input), while polynomials of degree two were employed in the approximation.

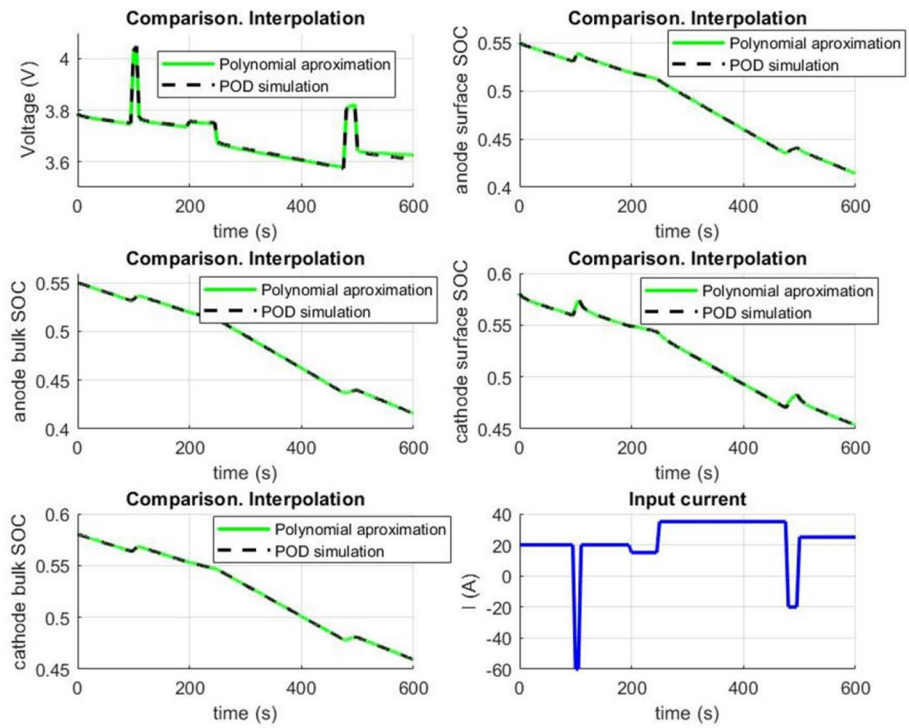

Fig. 16 Comparison of the results between the s-PGD model and the POD model in a 10 min itinerary. The s-PGD model is extracted based on itineraries of 10 min. $\Delta t = 5s$

6 Constructing a Digital Twin of the Electric Vehicle System

In this section, the s-PGD model developed in Sect. 5 (specifically the model with $\Delta t = 2$ s) is inserted in a HEV/EV model to simulate the whole vehicle under realistic operating conditions. To this end, the system simulation software *SimulationX* (ESI ITI GmbH, Dresden, Germany) is used. This is a tool in the field of multi-physics system simulation. It is based on the *Modelica* language, an object-oriented, declarative, multi-domain modeling language for component-oriented modeling of complex systems. See http://modelica.org for more information.

The neessary steps to insert the s-PGD model into a SimulationX model of the EV are summarized in Fig. 20.

Firstly, a battery package is created with all the elements needed for the battery model. This package is written using Modelica and encapsulates all elements needed to run the model such as types, classes and functions. In addition to describing equations in its native, equation-based form, Modelica provides a means of interfacing components to external C functions. A number of utility functions are created in C language such as functions for reading the input file containing the polynomial coefficients of the s-PGD solution. This file is generated by the program used to compute the s-PGD solution. By exploiting the Modelica external C interface, the regression information stored in the file can be tranparently loaded into the Modelica battery modeling components as a battery component that reproduces the s-PGD model in the full EV system simulation, Fig. 20.

Induction motors (for example the Tesla Model S) or synchronous motors (for example the Renault Zoe) are more widely used than the DC motors in the EV. However, for simplicity, and without loss of generality, we will model in this example the electric motor as a DC motor, which can be more often found in HEV applications.

A DC/DC converter is used to control the motor through the applied voltage. The model used in the converter is taken from [55]. Elements such as damping, inertias, springs, transmissions, rotational transformations and masses are used to model the mechanical part of the vehicle.

We consider a constant altitude during the itinerary, but note that it is straightforward to model different slopes where needed. The main forces acting against the movement of the car are the aerodynamic resistance, the rolling resistance and the grade resistance. Therefore, the tractive effort must be higher than these resistance forces to achieve the desired acceleration.

In Appendix 3 the interested reader can find the main parameters used in this example. An actual 10-min itinerary is used between Avenida de Valencia, E-46891 El Palomar and Plaza Barranc Marques, E-46890 Agullent, both in Spain. This itinerary is shown in Fig. 21.

The function $v(s)$, velocity as a function of position, is computed by assuming that the car is usually traveling at the maximum velocity permitted on that section of road. In Fig. 22, this velocity function is shown, while in Fig. 23

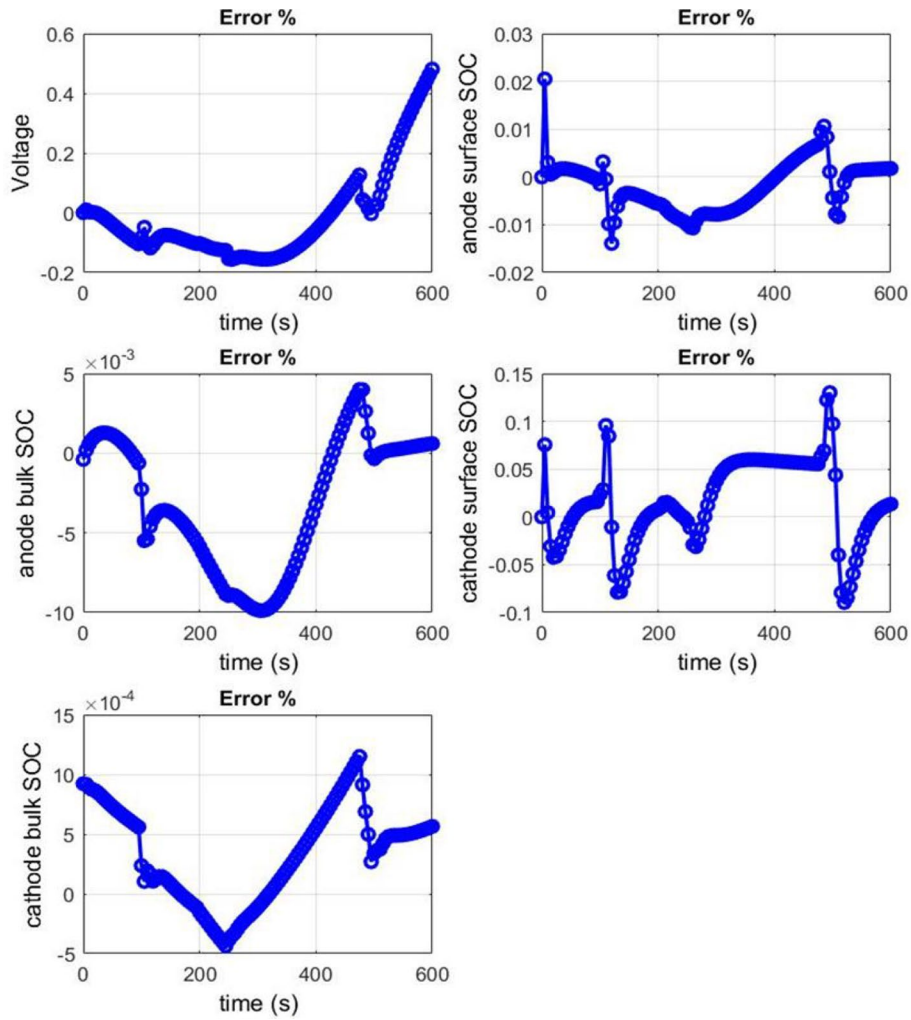

Fig. 17 Error between the s-PGD model and the POD model in a 10 min itinerary. The s-PGD model is extracted based on itineraries of 10 min. $\Delta t = 5s$

the velocity demanded by the driver as a function of time is shown.

A PI controller is used to ensure that the system follows the response $v(s)$. In our case, the controller sends a signal to the actuator of the control system as a function of the error between the sensed speed and the desired speed. The controller output can be expressed as:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau) d\tau,$$

where $e(t)$ is the error between the sensed speed and the desired speed and K_p and K_i are the proportional and integral gains respectively. In this way, the driver behavior is modeled by the PI controller which must be correctly parameterized to follow the desired speed response $v(s)$.

In Fig. 24, the SimulationX battery box model is shown. In Fig. 25, the SimulationX user parameter interface demonstrates how a user can select the number of (identical) cells are connected in series and in parallel. More than one type of cell can be used to create the battery by joining different battery boxes.

Furthermore, the possibility of adding an internal resistance to the model is created. In this way, the user can model linear voltage drops such as the contact resistance when the cells are connected through the collectors. In Fig. 26, we observe voltage variation in a cell due to the electrochemical model. In Fig. 27, we can observe the voltage variation in a cell due to the internal resistance (A resistance $R = 0.001\ \Omega$ is selected). In Fig. 28, a comparison of the bulk SoC and the surface SoC is shown. Finally, the critical SoC and the $c_{e,\text{avg}}$ in the negative electrode can be noticed from Fig. 29.

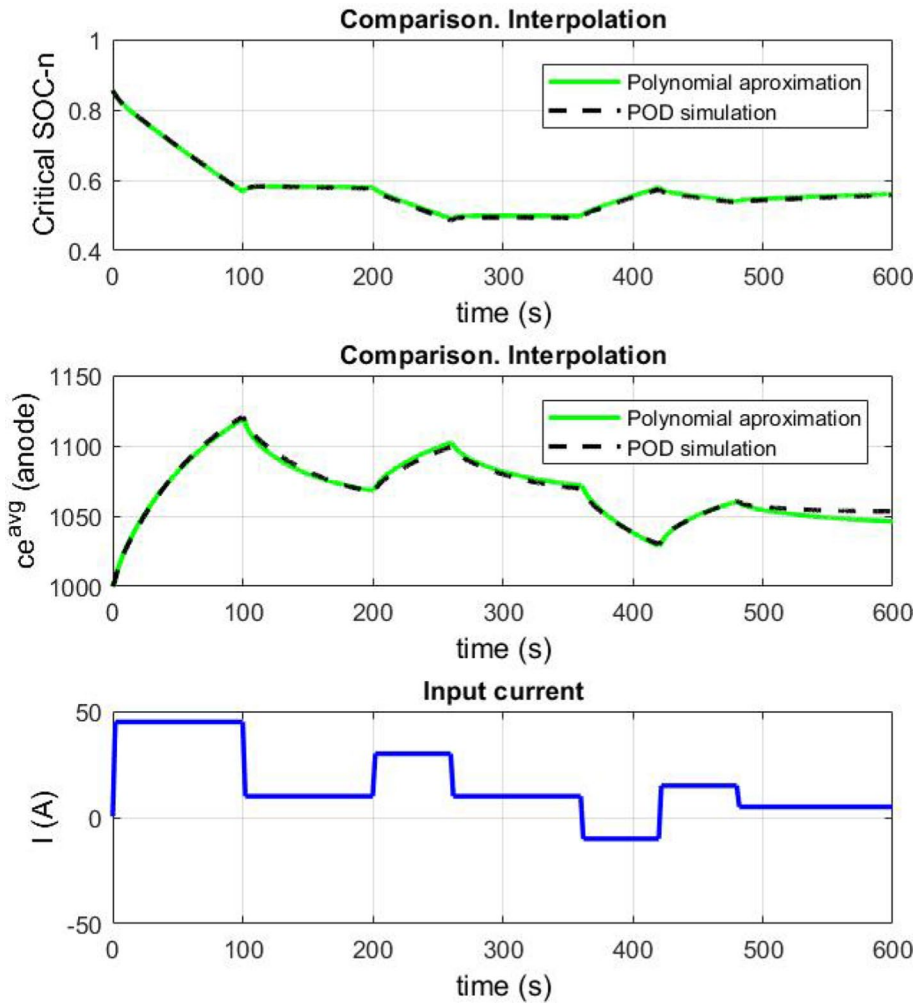

Fig. 18 Comparison of the results between the s-PGD model and the POD model in a 10-min itinerary (critical SoC and c_e^{avg}). The s-PGD model is extracted based on itineraries of 10 min. $\Delta t = 2s$

7 Constructing a Hybrid Twin

7.1 Introduction

In Sect. 5 we developed a battery model which can be simulated together with all the other systems within the EV under real-time feedback constraints. With the help of this model we obtained highly accurate results in the prediction of variables of interest as compared to the results of the original electrochemical model shown in Sect. 2.

In this section we go further. Our main objective is to develop a model for real-time applications capable to learn from data. In this way, systematic, biased deviations from the model can be attributed to deficiencies in the model. With the stream of experimental measurements the model will be corrected on-line and under real-time constraints, thus giving rise to the concept of the Hybrid Twin [7].

7.2 Hybrid Twins: The Big Picture

Each dynamical system can be characterized by the time evolution of its state vector, that is, the vector which contains the value of the variables needed to describe the system. The Hybrid Twin (HT) focuses on describing this evolution as well as possible, by taking into consideration that the physics-based model does not always fit the measurements exactly.

HTs are based on the idea that biased deviations (between the model and the measurements) show hidden physics—in other words, ignorance—and therefore, these deviations require a particular treatment. They will be modeled on-line by assimilating collected data. This correction on-the-fly is done using data-driven deviation models. On the other hand, the unbiased deviation contribution is associated to noise and it is addressed by using

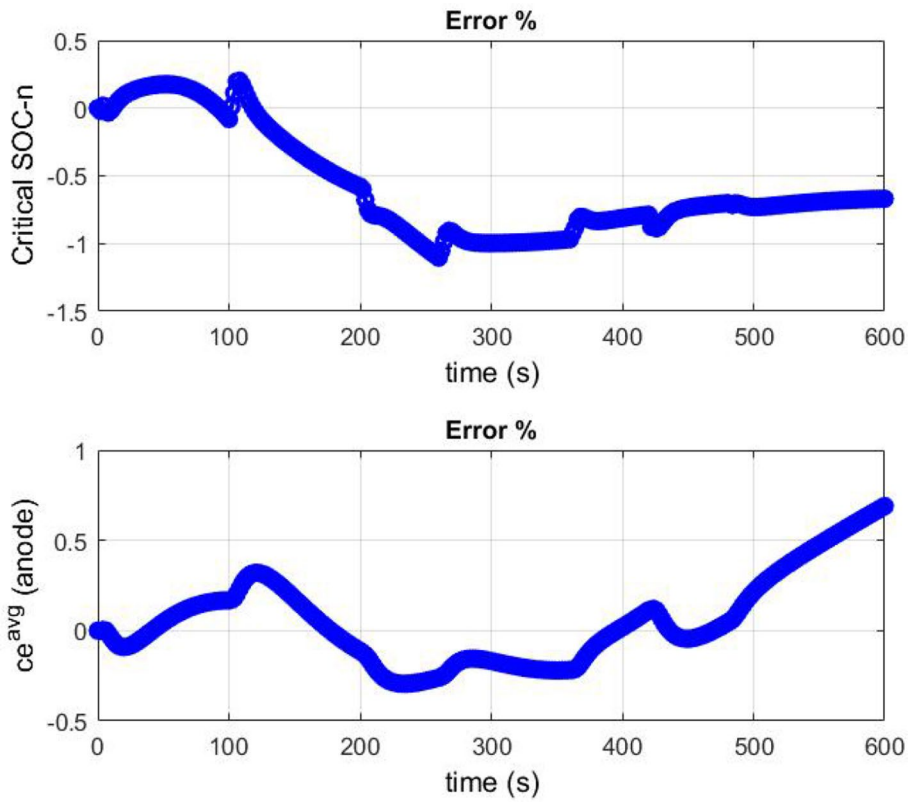

Fig. 19 Error between the s-PGD model and the POD model in a 10 min itinerary (critical SoC and c_e^{avg}). The s-PGD model is extracted based on itineraries of 10 min. $\Delta t = 2s$

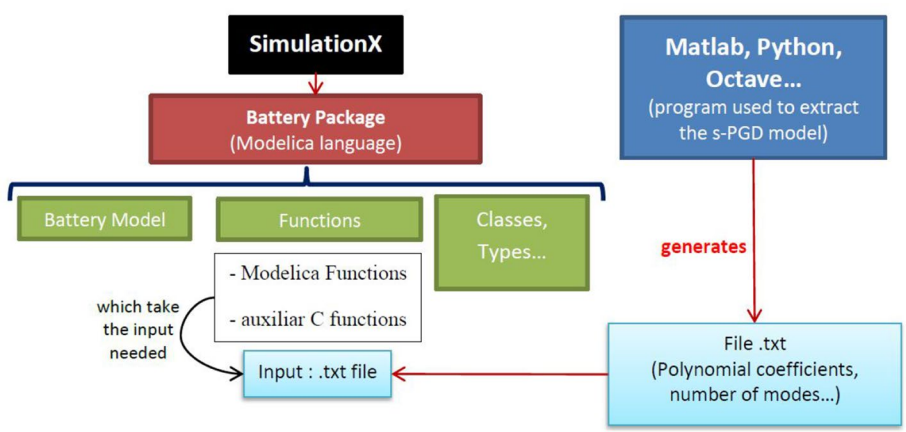

Fig. 20 Diagram showing the data flow for the s-PGD model when embedding in a SimulationX model

adequate filters. The interested reader can consult [17] or [42] for previous applications of this rationale.

In what follows the state vector at time t is denoted by $\mathbf{X}(t)$ and its time evolution by $\dot{\mathbf{X}}(t)$. A HT can thus be represented then by the following equation:

$$\dot{\mathbf{X}}(t) \approx \dot{\mathbf{X}}(t)^{\text{model}} + \dot{\mathbf{X}}(t)^{\text{correction}}. \tag{32}$$

In a HT framework [7] the dynamical evolution of the system will be represented as

$$\dot{\mathbf{X}}(t) = \mathbf{A}(\mathbf{X}, t; \boldsymbol{\mu}) + \mathbf{B}(\mathbf{X}, t) + \mathbf{C}(\mathbf{X}, t) + \mathbf{R}(t), \tag{33}$$

thus establishing that the rate of change of the state vector encompasses different contributions:

- The physics-based model $\mathbf{A}(\mathbf{X}, t; \boldsymbol{\mu})$, used to describe the dynamical system. Its contribution depends on a number of model parameters $\boldsymbol{\mu}$. This can be cast in the form of a ROM to ensure real-time feedback, namely, a PRM (Parametric Reduced Model) where the values of the involved parameters are identified from the assimilated-data.
- A data-based model correction $\mathbf{B}(\mathbf{X}, t)$, describing the gap between prediction and measurement.

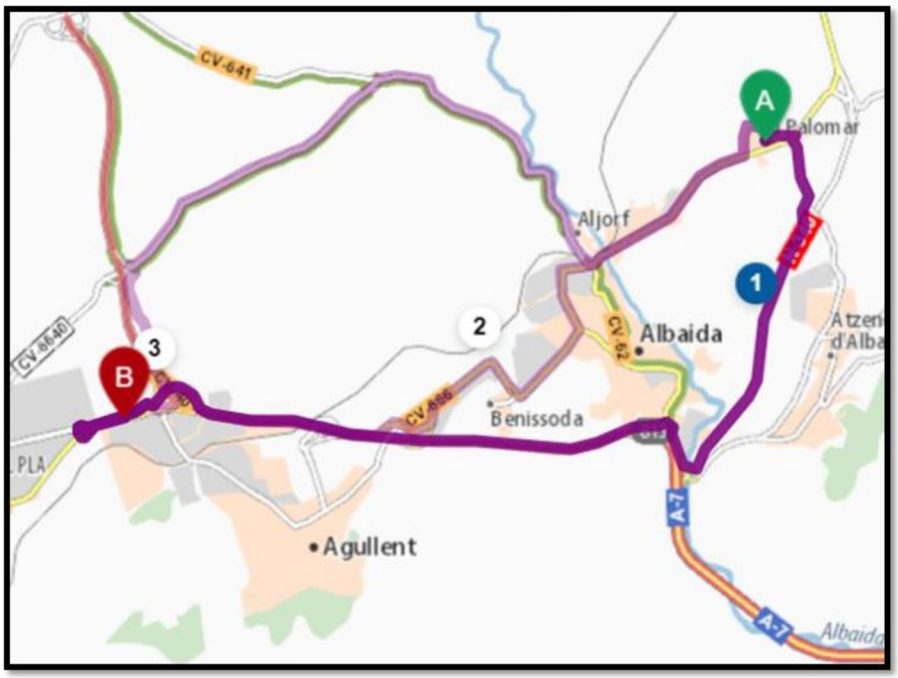

Fig. 21 Picture of the itinerary simulated by the SimulationX EV model. It corrsponds to a 10-min itinerary between Avenida de Valencia, E-46891 El Palomar and Plaza Barranc Marques, E-46890 Agullent, both in Spain

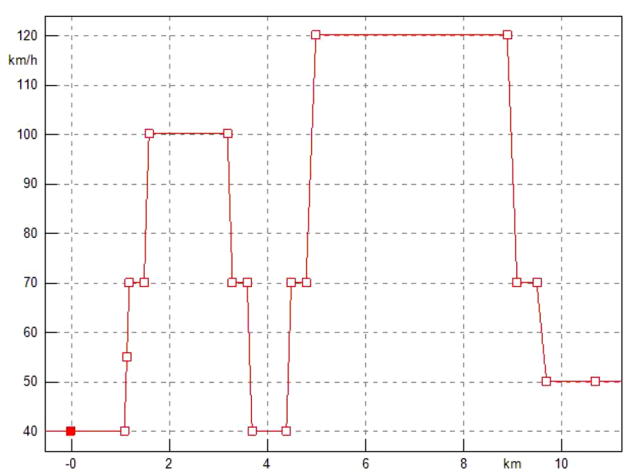

Fig. 22 Velocity (position) of the itinerary

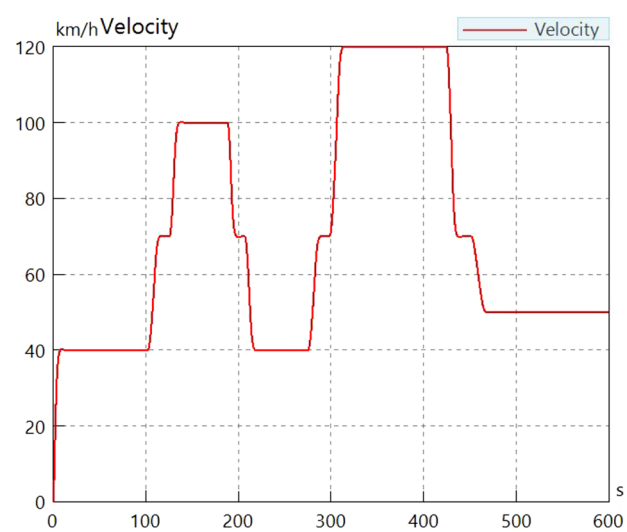

Fig. 23 Velocity response demanded by the driver

– External actions $\mathbf{C}(\mathbf{X}, t)$, introduced into the system dynamics in order to drive the model solution towards the desired target. It must be noted that the term $\mathbf{C}$ could depend on the model parameters $\boldsymbol{\mu}$ or it may not depend on $\mathbf{X}$.
– The unbiased noise $\mathbf{R}(t)$, that has been traditionally addressed using appropriate filters [7, 16]. This terms also includes external actions for which there is no possible prediction.

It is worth noting that the physics-based model $\mathbf{A}$ is calibrated on-line to adapt its output to the measurements, and the data-driven model correction $\mathbf{B}$ is constructed on-the-fly.

In the above definition, the adjective Hybrid comes from $\mathbf{B}(\mathbf{X}, t)$. The reason is that the model is composed of two contributions: one, $\mathbf{A}$, coming from a physics-based model—even if it is calibrated from data—and the other, $\mathbf{B}$, constructed from scratch from data. This double nature distinguishes digital twins from their hybrid counterparts.

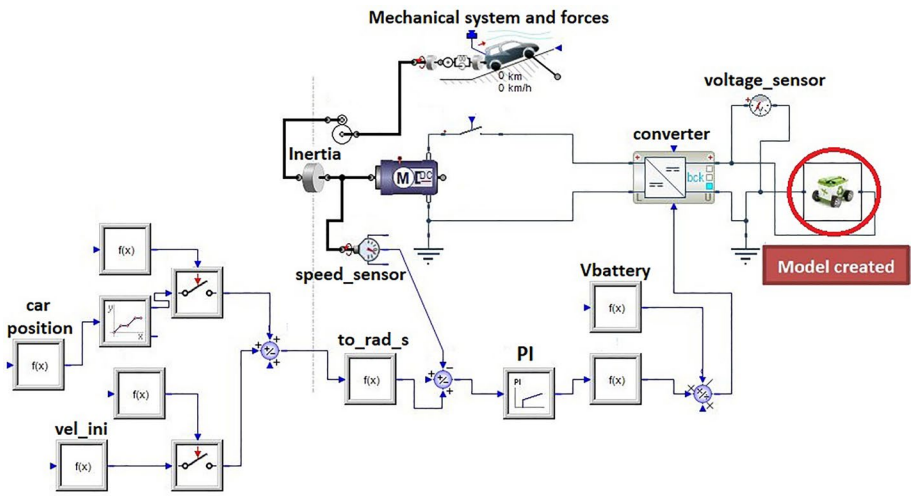

Fig. 24 Graphical view of the EV model created in SimulationX. In the red circle, the battery box created with the s-PGD solution is highlighted

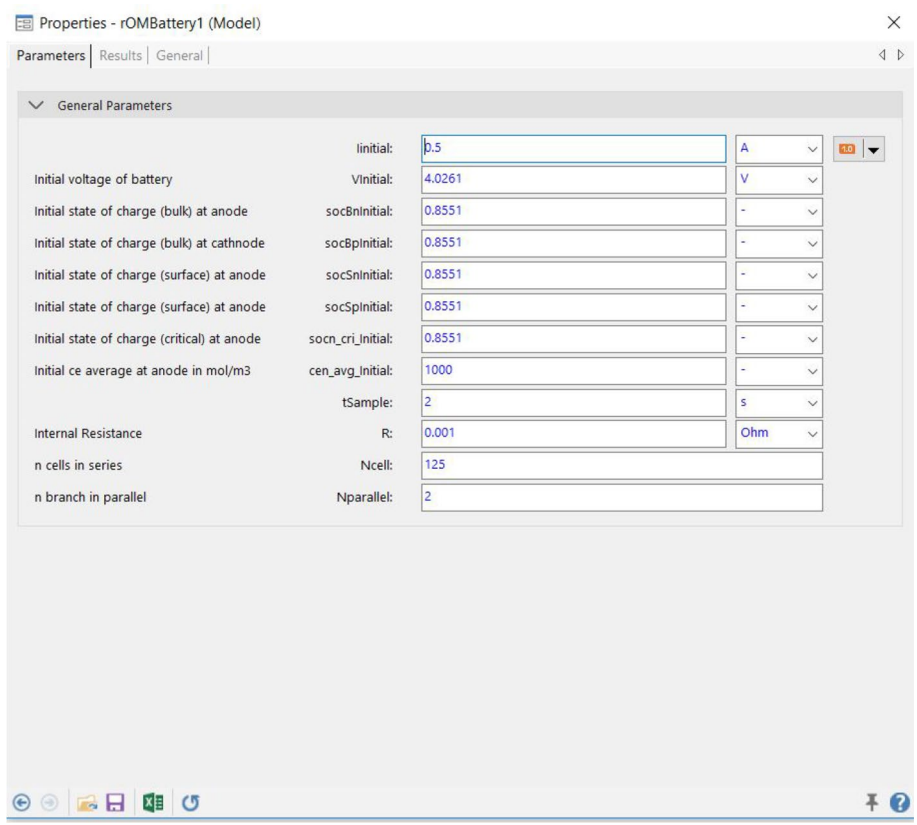

Fig. 25 Parameters of the SimulationX model created using the s-PGD solution

7.3 The Dictionary Model

The Dictionary Model starts with the supposition that we are dealing with a nonlinear dynamical system. So, we consider a nonlinear dynamical system:

$$\dot{\mathbf{X}}(t) = \mathbf{f}(\mathbf{U}, \mathbf{X}, t; \boldsymbol{\mu}), \tag{34}$$

where $\mathbf{X} \in \mathbb{R}^{n_x}$ is the vector representing the state of our dynamical system at time t, and $\boldsymbol{\mu}$ contains parameters of the system. $\mathbf{U} \in \mathbb{R}^{n_u}$ is a vector with the inputs of the system, and $\mathbf{f}$ represents the dynamics as a function of $\mathbf{U}$, $\mathbf{X}$, t and $\boldsymbol{\mu}$.

The discrete-time system is:

$$\mathbf{X}^{k+1} = \mathbf{f}(\mathbf{U}^k, \mathbf{X}^k, t_k; \boldsymbol{\mu}). \tag{35}$$

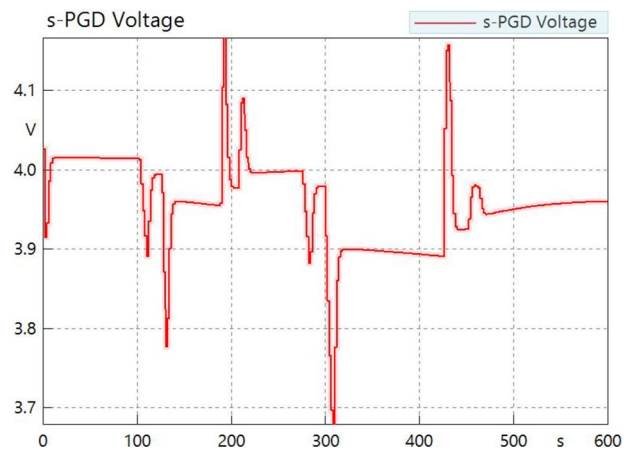

Fig. 26 Cell voltage predicted by the s-PGD model during the simulated itinerary

Fig. 27 Voltage drop produced by the internal resistance in a cell during the simulated. itinerary

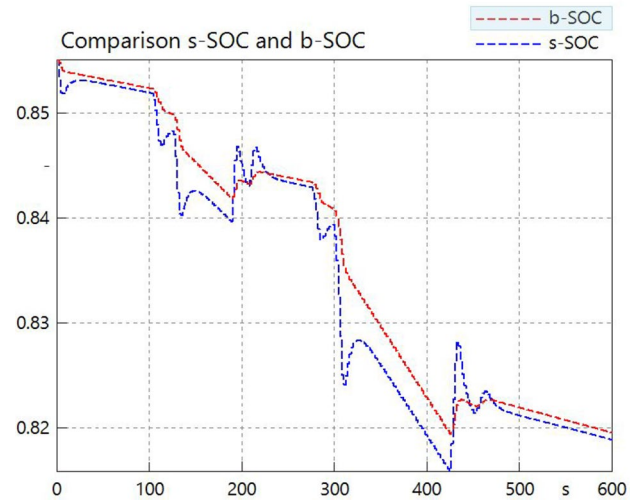

Fig. 28 Comparison between the bulk SoC and the surface SoC in the 10-min itinerary

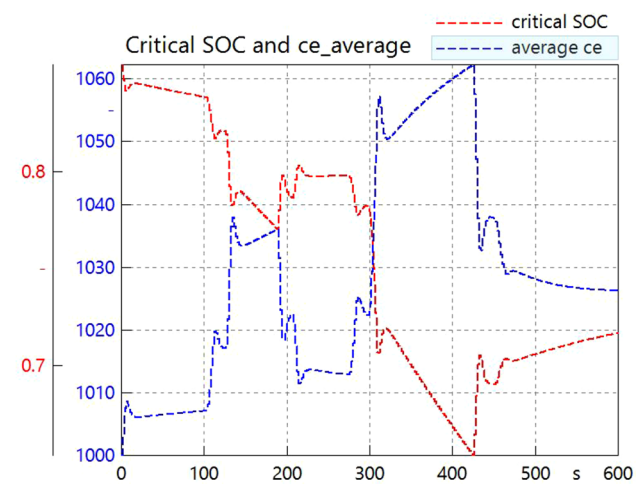

Fig. 29 Comparison between the critical SoC and the c_e^{avg} in the negative electrode during the simulated itinerary

If we assume that the system is linear,

$$\mathbf{X}^{k+1} = \mathbf{W}(\boldsymbol{\mu}) \cdot \mathbf{X}^k + \mathbf{V}(\boldsymbol{\mu}) \cdot \mathbf{U}^k, \tag{36}$$

where $\mathbf{X}^k \in \mathbb{R}^{n_x}$, $\mathbf{U}^k \in \mathbb{R}^{n_u}$, $\mathbf{W}(\boldsymbol{\mu}) \in \mathbb{R}^{n_x \times n_x}$ and $\mathbf{V}(\boldsymbol{\mu}) \in \mathbb{R}^{n_x \times n_u}$. The operator $\mathbf{W}(\boldsymbol{\mu})$ describes the dynamics of the unforced system while the operator $\mathbf{V}(\boldsymbol{\mu})$ characterizes the impact of the input $\mathbf{U}^k$ on the state $\mathbf{X}^{k+1}$.

The idea of the DM is to approximate the nonlinear behavior of Eq. (35) with locally linear approaches in terms of $\mathbf{X}^k$. To do this, the operators $\mathbf{W}$ and $\mathbf{V}$ of Eq. (36) must depend on $\mathbf{X}^k$. Therefore, the dynamical system described by the DM is:

$$\mathbf{X}^{k+1} = \mathbf{W}(\mathbf{X}^k, \boldsymbol{\mu}) \cdot \mathbf{X}^k + \mathbf{V}(\mathbf{X}^k, \boldsymbol{\mu}) \cdot \mathbf{U}^k. \tag{37}$$

This implies that there is a set of matrices $\mathcal{S}$ that must be obtained to describe the nonlinear behavior.

$$\mathcal{S} = \{\mathcal{Z}_1, \ldots, \mathcal{Z}_I\}; \ \mathcal{Z}_i = \{\mathbf{W}_i, \mathbf{V}_i\}. \tag{38}$$

The set $\mathcal{S}$ is the "dictionary", which allows us to reproduce a nonlinear model as a combination of local linear approaches.

It must be noted that if a pair of operators $\{\mathbf{W}_i, \mathbf{V}_i\}$ is needed for each $\mathbf{X}^k$, the method can be unfeasible because of the large amount of memory needed to store $\mathcal{S}$, and also because of the large size of the training set needed to obtain $\mathcal{S}$.

For this reason, this method will work well if each $\mathcal{Z}_i$ can be used in a wide range of state vectors $\mathbf{X}^k$. If this is not

the case, it is useful to look for a transformation $\mathbf{g}(\cdot)$, which transforms the evolution of the state vectors in a more linear behaviour $\mathbf{g}(\mathbf{X}^k)$. The Support Vector Machine (SVM) [6] and kernel methods [3, 19, 21] provide a number of tech-niques for constructing the feature space $\mathbf{g}(\mathbf{X}^k)$.

Here, a methodology based on the Dynamic Mode Decomposition (DMD) ideas [32] is proposed. The discrete equation for a local dynamical system must be rewritten as:

$$\mathbf{X}^{k+1} = \begin{bmatrix}\mathbf{W} & \mathbf{V}\end{bmatrix} \cdot \begin{bmatrix}\mathbf{X}^k \\ \mathbf{U}^k\end{bmatrix} = \mathbf{G} \cdot \mathbf{Y}^k. \tag{39}$$

Then, if we have n_s different snapshots for a given local linear system, we define the matrix $\mathbf{Y}_0 \in \mathbb{R}^{(n_x+n_u)\times(n_s-1)}$ and the matrix $\mathbf{X}_1 \in \mathbb{R}^{(n_x)\times(n_s-1)}$ as:

$$\mathbf{Y}_0 = \left[\mathbf{Y}^1, \mathbf{Y}^2, \ldots, \mathbf{Y}^{n_s-1}\right], \tag{40}$$

$$\mathbf{X}_1 = \left[\mathbf{X}^2, \mathbf{X}^3, \ldots, \mathbf{X}^{n_s}\right], \tag{41}$$

where each column corresponds to a snapshot. Therefore, the transition matrix $\mathbf{G}$ is learned to minimize the state reconstruction error:

$$\min \left\|\mathbf{X}_1 - \mathbf{G} \cdot \mathbf{Y}_0\right\|_F^2. \tag{42}$$

However, to guarantee stability, the Eq. (42) must be reformulated as:

$$\min \left\|\mathbf{X}_1 - \mathbf{G} \cdot \mathbf{Y}_0\right\|_F^2 \quad \text{s. t.} \quad \rho(\mathbf{W}) \leq 1 \tag{43}$$

where $\rho(\cdot)$ denotes the spectral radius.

Unfortunately, the feasible region constrained by $\rho(\mathbf{W}) \leq 1$ is nonconvex [23] and no general methodology exists to solve it.

We propose an option to deal with this problem if $\mathbf{W}$ is not too big. This is to change the restriction $\rho(\mathbf{W}) \leq 1$ to $\|\mathbf{W}\|_2 \leq 1$. Other approximations of other authors to this problem can be found in the literature [23, 24, 28].

A training set $\mathcal{T}$ composed of a collection of snapshots $\left\{\mathbf{Y}^1, \ldots, \mathbf{Y}^{N_f}\right\}$ of different simulations must be obtained. To decide the snapshots to obtain each local linear dynamical system, two methodologies are proposed.

The first one is to split each simulation time in intervals $(\mathcal{I}_1, \ldots, \mathcal{I}_N)$, where the linearization process is deemed good enough. Then, the collected snapshots $\left\{\mathbf{Y}^1, \ldots, \mathbf{Y}^{N_I}\right\}$ of each interval $\mathcal{I}_i$ are assigned to belong to the linear dynamical system obtained in their interval $\mathcal{I}_i$.

The second one is to use the k-means clustering [1, 37] algorithm to obtain the linear dynamical system of each cluster. In this second approach, given a set of snapshots $\left\{\mathbf{Y}^1, \ldots, \mathbf{Y}^{N_f}\right\}$, k-means clustering aims to partition the N_f observations into k ($k \leq N_f$) sets $\mathcal{K} = \left\{\mathcal{K}_1, \ldots, \mathcal{K}_k\right\}$ so as to minimize the within-cluster sum of squares:

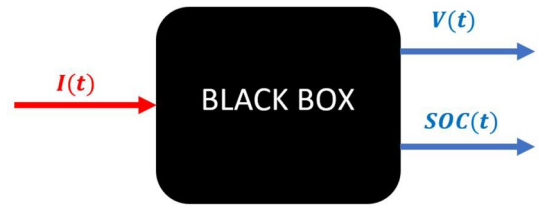

Fig. 30 Diagram of the approach used to apply the DM for the battery model

$$\underset{\mathcal{K}}{arg\,min} \sum_{i=1}^{k} \sum_{\mathbf{Y}\in\mathcal{K}_i} \left\|\mathbf{Y} - \boldsymbol{\mu}_i\right\|^2, \tag{44}$$

where $\boldsymbol{\mu}_i$ is the mean of points in $\mathcal{K}_i$.

Other techniques like the ones based on Hierarchical Clustering can be used. Here, techniques such as k-means clustering are preferrred because they can be applied to large volumes of data without difficulties.

7.4 Dictionary Model for the Electrochemical Model: Correcting the Gap

As we have introduced in Sect. 2, the electrochemical model is high-dimensional. If we use the ideas of the DM in the micro-scale, the matrices to work will be of an intractable size. For example, if a reduction of the dimension is not used, a state vector of thousands of dimensions can be found. In this case, the DM would be unfeasible. In addition, even if a drastic reduction of the dimension in the micro-scale could be applied, there is no guarantee that good results would be obtained because the system is highly nonlinear.

For these reasons, we must change the way we think about the electrochemical model to use the DM with success. We focus only on two variables of interest: The voltage and the bulk SoC.

The isothermal Newman's P2D model presented in this work give us the voltage and the bulk SoC as a function of both the micro-scale state of, and the current demanded by the battery, $I(t)$. However, the time evolution of the micro-scale is a consequence of $I(t)$. Therefore, the global overview would be that the Newman's P2D gives the evolution of the variables as a function of the initial micro-scale state of the battery and the current demanded by the battery $I(t)$.

It is worth noting that implies that the variable which dictates the time evolution of the whole system is $I(t)$ (for a fixed initial state). In this way, the model can be interpreted as a black box, as sketched in Fig. 30.

Therefore, a dynamical system, which has only voltage and bulk SoC as variables of interest, along with the input $I(t)$, is considered. Note that $I(t)$ carries two meanings: the time evolution of the current and the time evolution of the

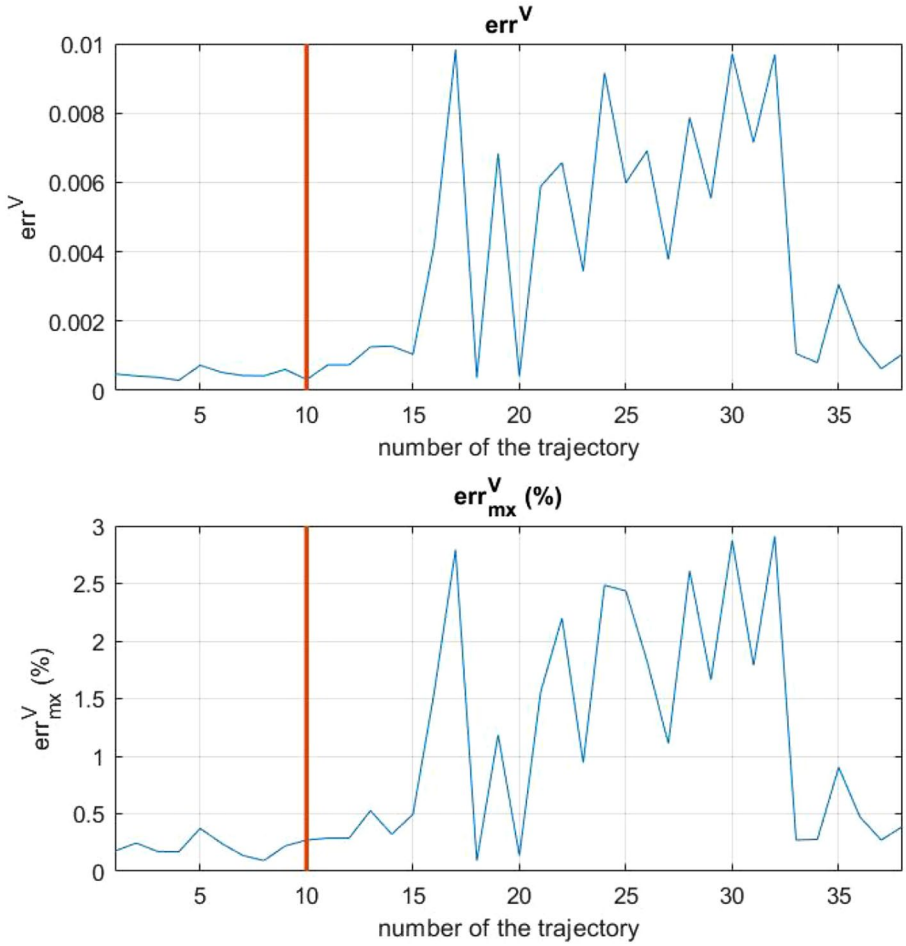

Fig. 31 Voltage error of using the model extracted through the DM for 38 trajectories (without any on-line correction). The first ten trajectories correspond to the training set. The transition between the training set and the not-trained trajectories is highlighted with a red line. (Color figure online)

micro scale. For this reason, the dynamical system that we are proposing must depend on the history. In this way, it takes into consideration how the micro scale is evolving, and how the response must change because of that.

Therefore, variables based on the time history of $I(t)$ must be created to measure this evolution of the micro-scale state. In addition, a transformation $\mathbf{g}(\cdot)$, introduced in Sect. 7.3, must be defined to linearize as much as possible the behavior of the system. The identified local linear systems have the form

$$\mathbf{Y} = \begin{bmatrix} V \\ \text{bulk SoC} \\ I \\ \frac{dI}{dt} \\ \int_0^t I\,dt \\ \int_0^t \int_0^t I\,dt\,dt \end{bmatrix} \longrightarrow \mathbf{X} = \begin{bmatrix} V \\ \text{bulk SoC} \end{bmatrix},$$

$$\mathbf{G} = \begin{bmatrix} g_{(1,1)} & g_{(1,2)} & g_{(1,3)} & g_{(1,4)} & g_{(1,5)} & g_{(1,6)} \\ g_{(2,1)} & g_{(2,2)} & g_{(2,3)} & g_{(2,4)} & g_{(2,5)} & g_{(2,6)} \end{bmatrix}.$$

As we have already seen in Sect. 7.3, the dictionary $\mathcal{S}$, which allows us to reproduce a nonlinear model as a combination of local linear approaches, is obtained in two phases.

The first stage is off-line during which a training set $\mathcal{T}$ composed of a collection of snapshots $\left\{\mathbf{Y}^1, \ldots, \mathbf{Y}^{N_f}\right\}$ of different simulations must be obtained. Then, in the on-line stage, the model is corrected, enriching the original dictionary. In this way, our model can be adapted so as to fit with unexpected experimental measurements.

Consider, for instance, a model trained to reproduce any possible 10-min trajectory from an initial SoC of 40-60%. If the model attempts to reproduce a trajectory of an initial SoC of 80% (not trained), the on-line correction must be able to reproduce well this new scenario. Or imagine that an attempt is made to employ model with a 20-min trajectory whereas it is trained only for 10-min journeys. Then, the on-line correction also needs to be able to reproduce well this new scenario. Gaps between reality and the electrochemical model should be corrected in the same way.

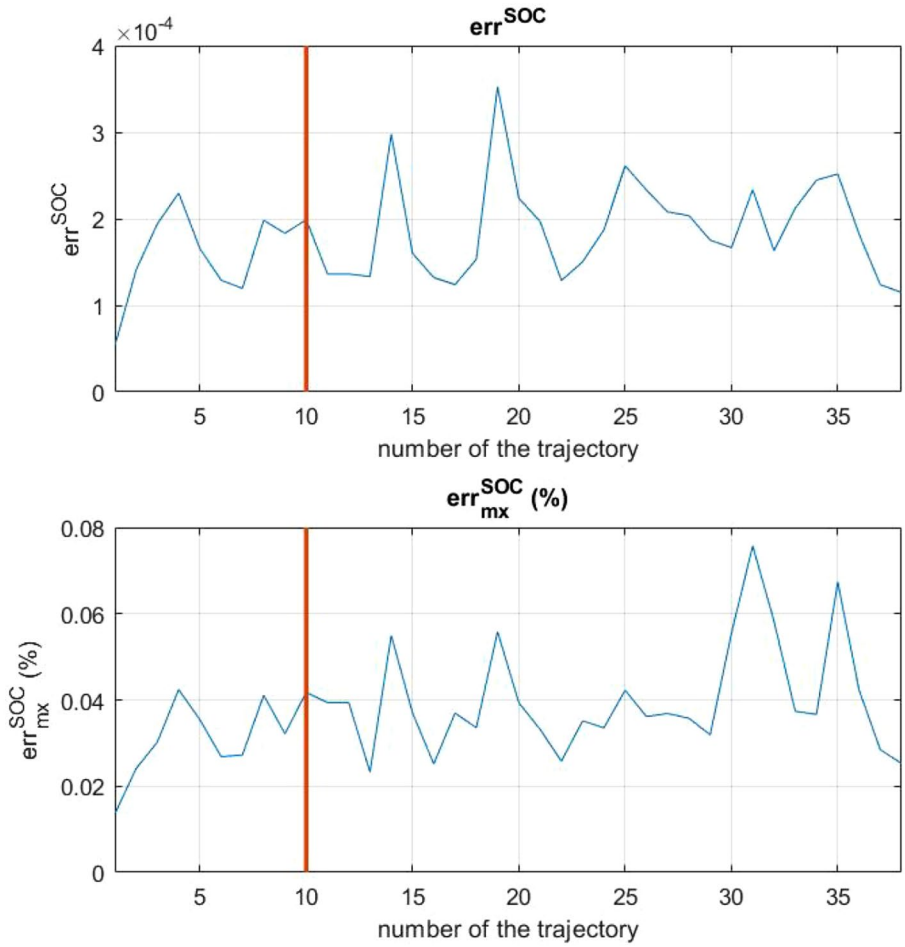

Fig. 32 SoC error of using the model extracted through the DM for 38 trajectories (without any on-line correction). The first ten trajectories correspond to the training set. The transition between the training set and the not-trained trajectories is highlighted with a red line. (Color figure online)

Noise is not considered in the results of Sect. 7.5. It could easily be, however, with the methodology already considered in Sect. 7.2.

7.5 Results

The systems to identify have the form presented in Sect. 7.4. In these, we expect our model to be able to reproduce an arbitrary 10-min trajectory with an initial SoC of 50%. The pseudo-experimental training set $\mathcal{T}$ is composed of a collection of snapshots obtained from the Newman's P2D electrochemical model shown in Sect. 2 using the POD methodology explained in Sect. 4. This training set $\mathcal{T}$ is obtained using ten different 10-min trajectories. In the comparisons that follow, the results of the DM are compared with the ones extracted from the Newman's P2D electrochemical model.

Two different error criteria are used. The first one is:

$$\mathrm{err}^{V}_{\max} = \max\left(100 \cdot \frac{|V_{DM}(t) - V_{\mathrm{P2D}}(t)|}{V_{\mathrm{P2D}}(t)}\right),$$
$$\mathrm{err}^{\mathrm{SoC}}_{\max} = \max\left(100 \cdot \frac{|\mathrm{SoC}_{DM}(t) - \mathrm{SoC}_{P2D}(t)|}{\mathrm{SoC}_{P2D}(t)}\right),$$

where V denotes voltage, SoC denotes the bulk state of charge, the subscript DM denotes the results obtained by the Dictionary Method and the subscript $P2D$ denotes the results obtained by Newman's P2D model.

The second error criteria used is given by

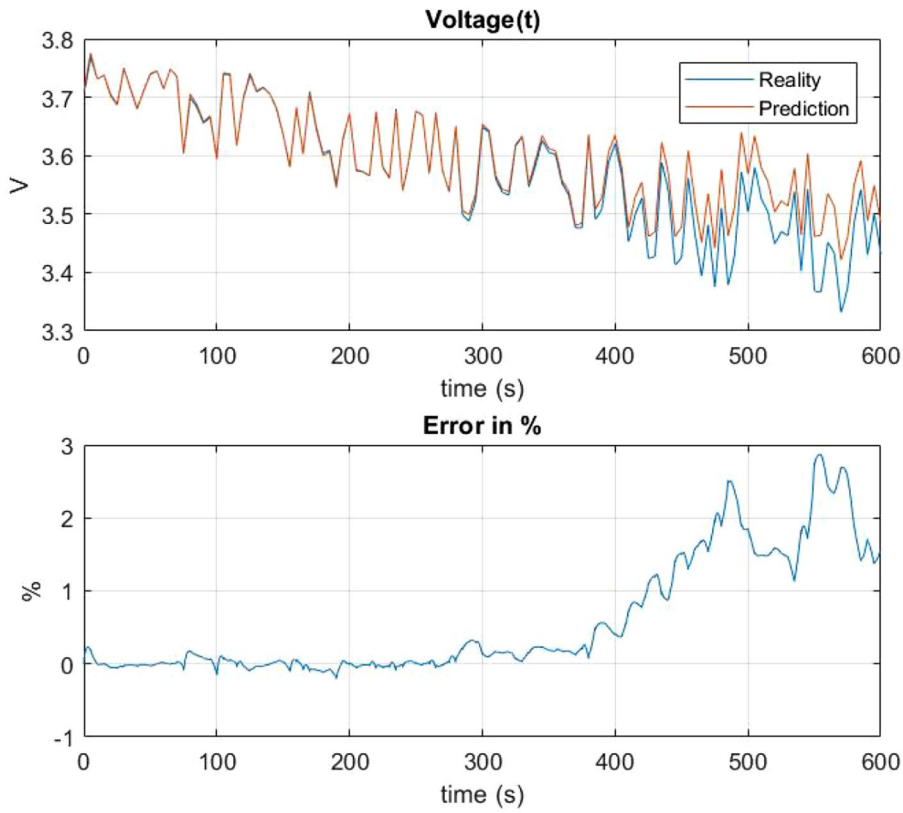

Fig. 33 Comparison of the results between the P2D model and the DM model for trajectory number 30. Here, the on-line correction term is not used

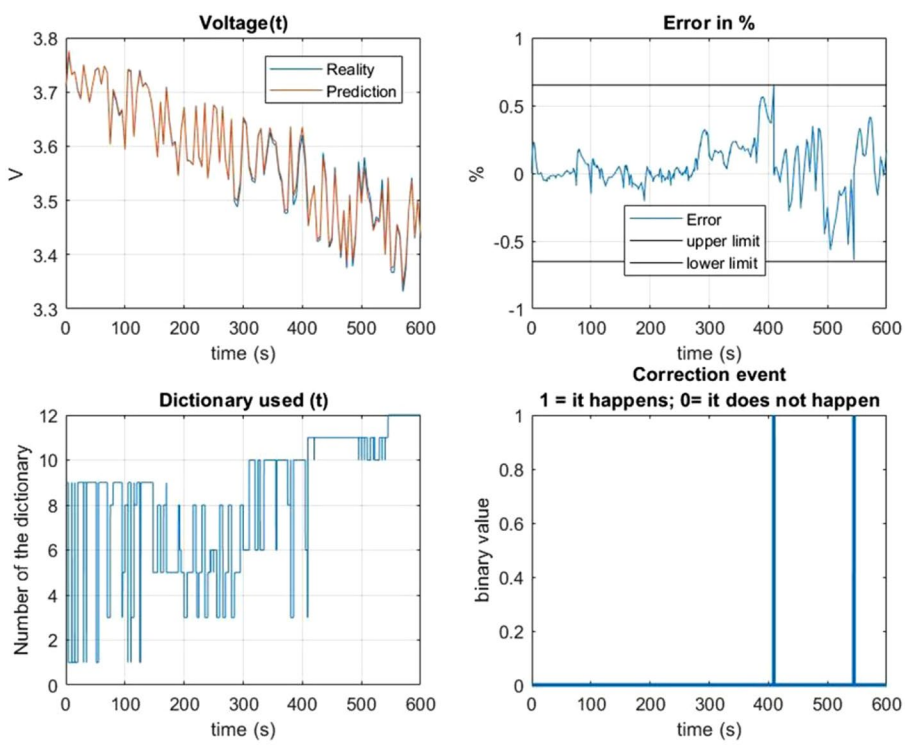

Fig. 34 Comparison of the results between the P2D model and the DM model for trajectory number 30. Here, the on-line correction is used. To update the on-line dictionary we impose the criteria to limit the maximum error to 0.65% for the voltage. The vertical lines in the correction event plot indicate when the correction event happens

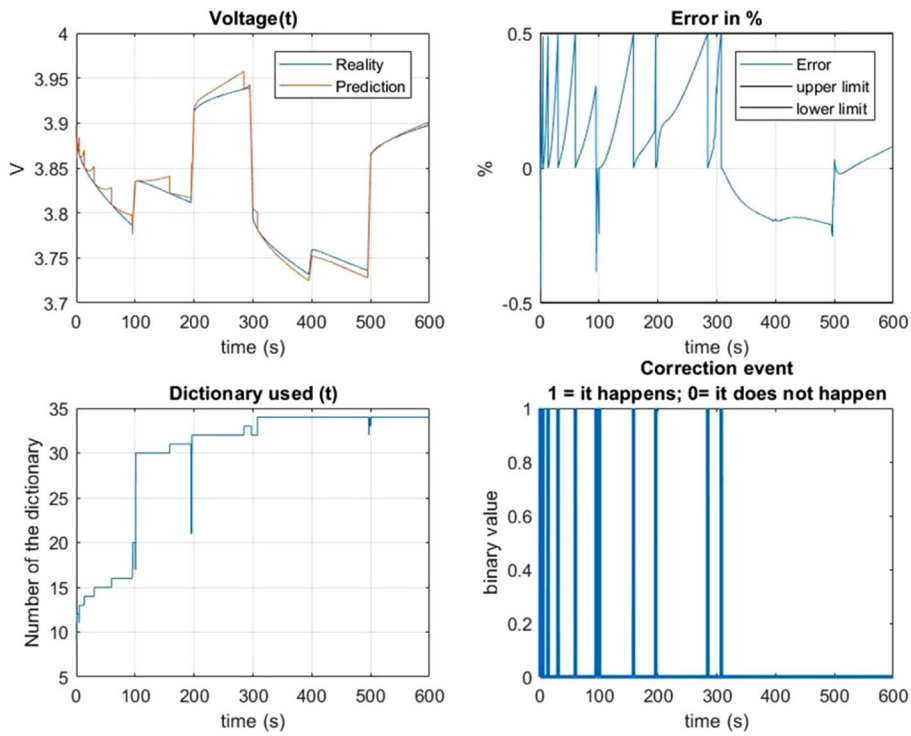

Fig. 35 The DM with on-line correction reacts in a region for which the model is not prepared. Results are shown for the voltage during a trajectory with an initial SoC of 85% (not initially considered in the training set). To update the on-line dictionary, we imposed the criteria to limit the maximum error to 0.5% for the voltage

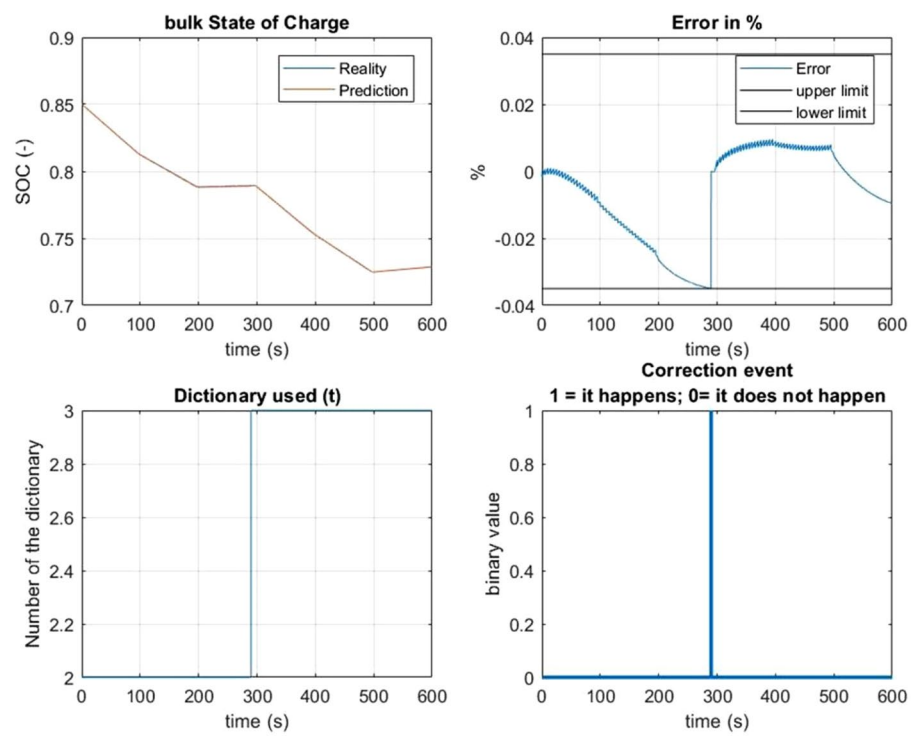

Fig. 36 The DM with on-line correction reacts to a region in which the model is not prepared. Here we can see the results for the SoC for a trajectory with an initial SoC of 85% (not trained for this scenario). To update the on-line dictionary we imposed the criteria to limit the maximum error to 0.035% for the SoC

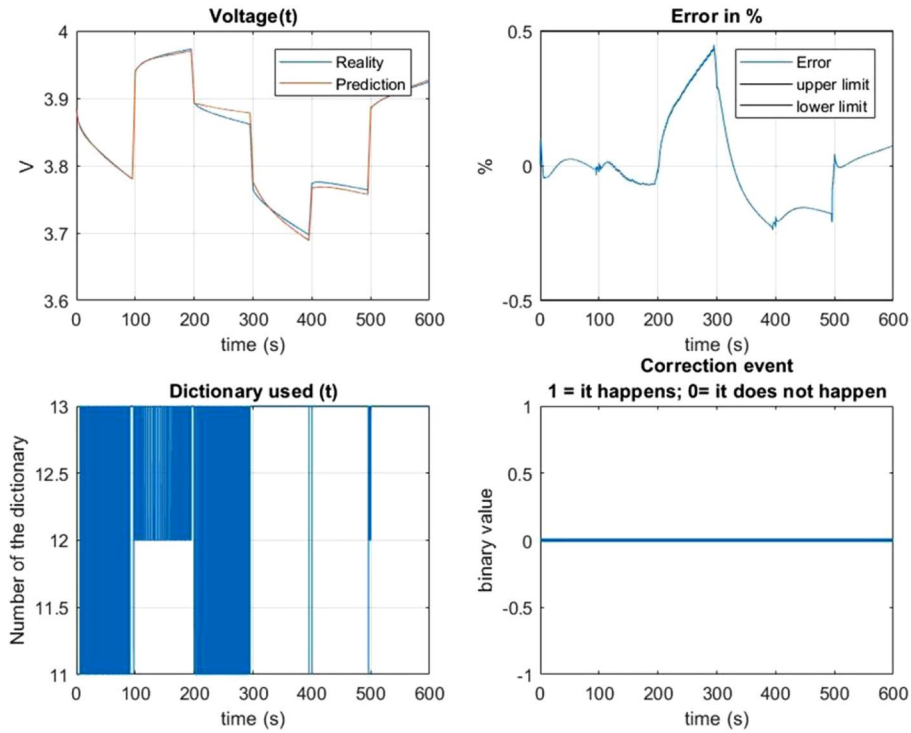

Fig. 37 The DM with on-line correction reacts to another trajectory with an initial SoC of 85%. We can see that no correction is needed for the voltage in this second trajectory due to the on-line learning of the first one (Fig. 35). To update the on-line dictionary we imposed the criteria to limit the maximum error to 0.5% for the voltage

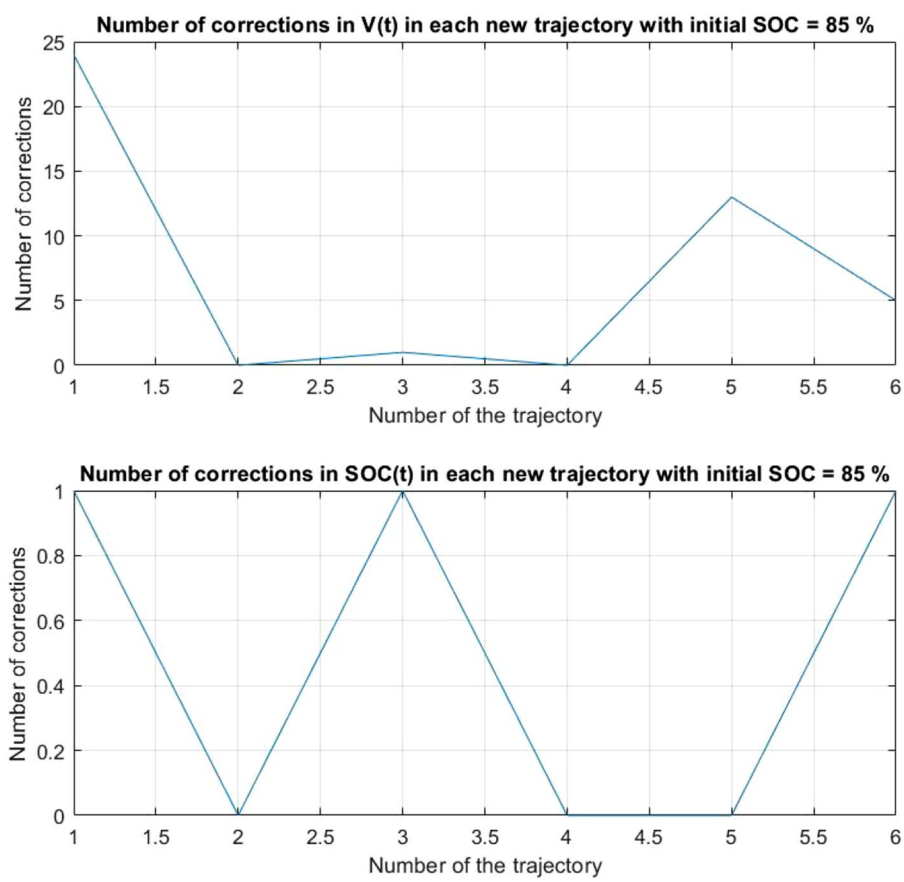

Fig. 38 Number of corrections needed for different trajectories with an initial SoC of 85% (for which the original model is not trained). To update the on-line dictionary we imposed the criteria to limit the maximum error to 0.035% for the SoC and to 0.5% for the voltage

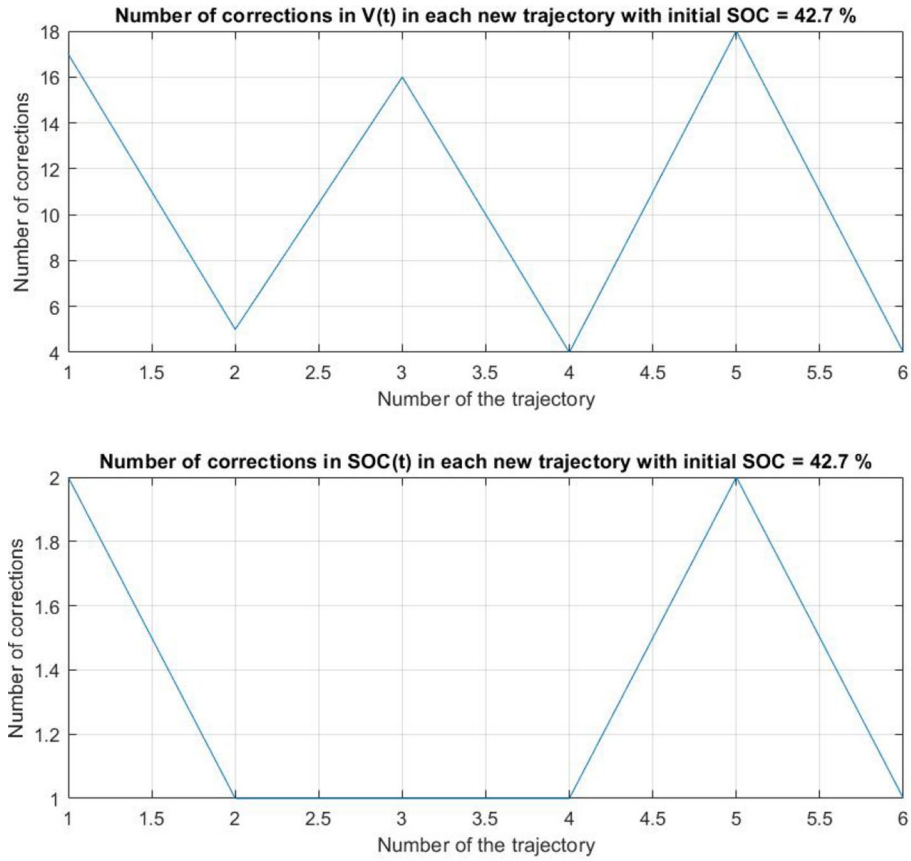

Fig. 39 Number of corrections needed for different trajectories with an initial SoC of 42.7% (for which the original model is not trained). To update the on-line dictionary we imposed the criteria to limit the maximum error to 0.035% for the SoC and to 0.5% for the voltage

$$\mathrm{err}^{V} = \frac{\left\| V_{DM}(t) - V_{P2D}(t) \right\|_2}{\left\| V_{P2D}(t) \right\|_2},$$
$$\mathrm{err}^{\mathrm{SoC}} = \frac{\left\| \mathrm{SoC}_{DM}(t) - \mathrm{SoC}_{P2D}(t) \right\|_2}{\left\| \mathrm{SoC}_{P2D}(t) \right\|_2}.$$

In Figs. 31 and 32, we can observe the error caused by the model extracted through the DM for 38 trajectories (without any on-line correction). The first ten trajectories correspond to the training set. The transition between the training set to the not-trained trajectories is highlighted with a red line. In these plots we can see that the voltage error is always lower than 3% and it can be lower than 1% in a lot of itineraries. Conversely, the SoC error is negligible (always lower than 0.08%).

In Fig. 33 and 34 we show the comparison of the DM with and without the on-line correction for a trajectory which we have chosen deliberately to be different to those in the training set. To update the on-line dictionary we impose the criteria to limit the maximum error to 0.65% for the voltage and to 0.05% for the SoC. In addition, the vertical lines in the correction event plot indicate when a correction event happens. We can notice in these figures that only two corrections are needed to maintain the error level below 0.65%.

In Fig.35, the on-line correction is shown to react within a region for which the model is not prepared. Here we can see the results for a trajectory with an initial SoC of 85%. We can see that a few corrections are needed at the beginning but later, the model attains good predictability abilities.

In Fig. 36, we show the SoC results for the above conditions. Note that only one correction is needed for the SoC, despite the fact that the model was not trained to work in these conditions.

In Fig. 37, we see how the on-line correction reacts to a different trajectory with an initial SoC of 85%. In this case, no correction is needed due to the on-line learning of the first one (see Fig. 35).

Finally, in Fig. 38 and 39, the number of corrections needed for six trajectories of an initial SoC of 85% and another six for an initial SoC of 42.7% are shown. As a result of these corrections, the error the voltage remains below 0.5%.

As an additional measure, the dictionary is enriched at each correction. With this addition, the number of corrections is reduced as the model learn how to react in these new scenarios. To update the on-line dictionary, we imposed the

criteria to limit the maximum error to 0.035% for the SoC and to 0.5% for the voltage.

8 General Conclusions

In Sect. 4 a ROM is developed using the POD technique. It can be observed that the results of the full model are reproduced with a high accuracy using the POD model. A large reduction in the computational time and resources needed to solve the electrochemical model is also achieved.

The saving in computational time and resources means that cell design, parameter fitting, and simulation of battery packs can be achieved under this rationale. The key to developing this model is the extraction of the basis functions using the results of off-line simulations. As we showed, with only an off-line simulation a wide operational range of the cell can be reproduced. To further enrich the basis, some off-line simulations close to a SoC of 100% and 0 % are recommended.

In Sect. 5, a data-driven ROM was developed using the s-PGD regression technique which can learn either from a physics-based model or experimental data. In Sect. 6, we showed that the developed model was used successfully in conjunction with the whole EV model to obtain fast results with the accuracy of the electrochemical model. We believe that this approach could be an excellent option and a good first step to develop the new generation of BMS for the automotive industry because of its accuracy and the low computational resources required. Furthermore, the Digital Twin created using the s-PGD does not only allow for real-time simulations, but can also adapt its predictions taking into consideration the real driving conditions and the real driving cycle to change the planning in real-time.

Indeed, it would be interesting to enrich the model with the thermal gradient or aging effects to further enhance its applicability. This would require a reformulation of the electrochemical model so that the phenomena were correctly described.

Finally, in Sect. 7, a model based on the hybrid twin paradigm is presented to correct the electrochemical model in real-time. To achieve this, a data-driven model based upon Dynamic Mode Decomposition is developed. Accurate results are found by using this data-driven model to reproduce the behavior of the cell. As we have shown, the data-driven model has excellent capabilities to adapt quickly and reliably to new scenarios (situations for which the model has not been trained).

As a future line of research, it would be interesting to see the capabilities of the data-driven model to predict the aging effects or to deal with the presence of noise (in this second case, we propose to create filters to eliminate the presence of noise in the measurements).

Acknowledgements This project has received funding from the Spanish Ministry of Economy and Competitiveness through Grants number DPI2017-85139-C2-1-R and DPI2015-72365-EXP, by the Regional Government of Aragon and the European Social Fund, research group T24 17R, as well as from the ESI Group International Chairs at University of Zaragoza and Arts et Métiers ParisTech.

Compliance with ethical standards

Conflicts of interest The authors declare that they have no conflict of interest.

Appendices

Nomenclature Electrochemical Model

See Table 1.

Table 1 Nomenclature used

Symbol	Definition	Unit
a_s	Active surface area per electrode unit volume	$\frac{m^2}{m^3}$
A	Electrode plate area	m^2
c	Concentration of lithium in a phase	$\frac{mol}{m^3}$
D	Diffusion coefficient of lithium species	$\frac{m^2}{s}$
F	Faraday's constant	$\frac{C}{mol}$
i_o	Exchange current density of an electrode reaction	$\frac{A}{m^2}$
I	Applied current	A
j^{Li}	Reaction current resulting in production or consumption of Li	$\frac{A}{m^3}$
L_n	Length of the anode	m
L_s	Length of the separator	m
L_p	Length of the cathode	m
L_t	Length of the cell	m
p	Bruggeman exponent	–
R	Universal gas constant	$\frac{J}{mol \cdot K}$
R_f	Film resistance on an electrode surface	$\Omega \cdot m^2$
R_s	Radius of solid active material particles	m
t_+^0	Transference number of lithium ion with respect to the vel. of solvent	–
T	Absolute temperature	K
U	Open-circuit potential of an electrode reaction	V
α_a	Anodic transfer coefficient for an electrode reaction	–
α_c	Cathodic transfer coefficient for an electrode reaction	–
η	Surface overpotential of an electrode reaction	V
K	Conductivity of an electrolyte	$\frac{S}{m}$
K_D	Diffusional conductivity of an electrolyte	$\frac{A}{m}$
ϕ	Volume-averaged electrical potential in a phase	V
σ	Conductivity of solid active materials in an electrode	$\frac{S}{m}$
c_{se}	Concentration of litium in the solid phase at solid/electrolyte interface	$\frac{mol}{m^3}$
k_o	kinetic rate constant	$\frac{m^{2.5}}{mol^{0.5} s}$
ε	Volume fraction or porosity of a phase	–
$\theta_{100\%}$	Stoichiometry at 100%	–
$\theta_{0\%}$	Stoichiometry at 0%	–
θ	$\frac{c_{se}}{c_{s,\max}}$	–
Symbol	Definition	Unit

The subscript s refers to the solid phase with the exception of L_s.

The subscript e refers to the electrolyte phase.

The subscript n refers to the anode.

The subscript p refers to the cathode.

The subscript "max" refers to maximum.

The superscript "eff" refers to effective.

The superscript "init" refers to initial

Cell Parameters and Additional Equations

See Table 2.

The units of the values shown in the table above are the same as those used in Table 1.

Additional equations:

$$D_e^{\mathrm{eff}} = \varepsilon_e^p \cdot D_e, \quad K^{\mathrm{eff}} = \varepsilon_e^p \cdot K, \quad K_D^{\mathrm{eff}} = \frac{2 \cdot R \cdot T \cdot K^{\mathrm{eff}}}{F} \cdot (t_+^0 - 1),$$
$$\sigma^{\mathrm{eff}} = \varepsilon_s \cdot \sigma, \quad a_s = \frac{3 \cdot \varepsilon_s}{R_s}.$$

Table 2 Cell parameters

Parameter	Cathode	Separator	Anode
c_e^{init}	1000	1000	1000
c_s^{init}	25,751	–	26,128
$c_{s,\max}$	51,554	–	30,555
D_e	7.5×10^{-10}	7.5×10^{-10}	7.5×10^{-10}
D_s	10^{-14}	–	3.9×10^{-14}
k_o	2.25196×10^{-6}	–	4.85416×10^{-6}
L	8×10^{-5}	2.5×10^{-5}	8.8×10^{-5}
R_s	2×10^{-6}	–	2×10^{-6}
ε_s	0.59	–	0.4824
ε_e	0.385	0.724	0.485
σ	100	–	100
a_s	885,000	–	723,600
p	4	4	4
F	96,485	96,485	96,485
R	8.314472	8.314472	8.314472
T	298.15	298.15	298.15
α_a	–	–	0.5
α_c	0.5	–	–
$\theta_{100\%}$	0.4955	–	0.8551
$\theta_{0\%}$	0.99174	–	0.01429
A	1	–	1
t_+^0	0.364	0.364	0.364
Parameter	Cathode	Separator	Anode

K, U_n and U_p are usually computed using empirical equations according to the simulated cell. The empirical equations for this cell are:

$$
\begin{aligned}
U_n(\theta_n) &= 0.7222 + 0.1387 \cdot \theta_n \\
&\quad + 0.029 \cdot \theta_n^{0.5} - \frac{0.0172}{\theta_n} + \frac{0.0019}{\theta_n^{1.5}} \\
&\quad + 0.2808 \cdot \exp(0.9 - 15 \cdot \theta_n) - 0.7984 \cdot \exp(0.4465 \cdot \theta_n - 0.4108), \\
U_p(\theta_p) &= \frac{-4.656 + 88.669 \cdot \theta_p^2 - 401.119 \cdot \theta_p^4 + 342.909 \cdot \theta_p^6 - 462.471 \cdot \theta_p^8 + 433.434 \cdot \theta_p^{10}}{-1 + 18.933 \cdot \theta_p^2 - 79.532 \cdot \theta_p^4 + 37.311 \cdot \theta_p^6 - 73.083 \cdot \theta_p^8 + 95.96 \cdot \theta_p^{10}}, \\
K(c_e) &= 4.1253 \times 10^{-2} + 5.007 \times 10^{-4} \cdot c_e - 4.7212 \times 10^{-7} \cdot c_e^2 \\
&\quad + 1.5094 \times 10^{-10} \cdot c_e^3 - 1.6018 \times 10^{-14} \cdot c_e^4.
\end{aligned}
$$

Parameters Used in SimulationX Model

See Table 3.

The parameters of the PI controller are assumed to have the following form:

$$G \cdot \left(1 + \frac{1}{Ti \cdot s}\right).$$

The parameters of the DC/DC converter are the default parameters in the SimulationX model described in [55]. The main forces considered for the linear movement of the vehicle are:

$$F_t = m \cdot a + R_{aero} + R_r + R_{grade}$$

where F_t is the tractive effort, and

Table 3 Main parameters

Parameter	Value	Parameter	Value
Vehicle mass m	1300kg	Initial displacement	0km
Initial velocity	0km/h	Driving shaft inertia	$0.9\ \mathrm{kg} \cdot \mathrm{m}^2$
Inertia side shafts	$0.5\ \mathrm{kg} \cdot \mathrm{m}^2$	Inertia Driving Wheels	$1.42\ \mathrm{kg} \cdot \mathrm{m}^2$
Wheel radius	300 mm	Axle transmission ratio	3
Air drag coefficient C_w	0.3	Rolling resistance Coef. C_r	0.01
Air density ρ	$1.293\ \mathrm{kg/m}^3$	Reference area A_{ref}	$1.5\ \mathrm{m}^2$
Hill gradient angle γ	0	PI controller G	1.39×10^{-3}
PI controller T_i	0.0011	Armature resistance R_a	$0.537\ \Omega$
Armature inductance L_a	1 mH	–	–
Parameter	Value	Parameter	Value

$$R_{\text{aero}} = \rho/2 \cdot A_{ref} \cdot C_w \cdot V^2,$$

represents the aerodynamic resistance. Likewise,

$$R_r = C_r \cdot m \cdot g \cdot \cos(\gamma),$$

represents the rolling resistance and, finally,

$$R_{\text{grade}} = m \cdot g \cdot \sin(\gamma),$$

is the grade resistance.

References

1. Arthur D, Vassilvitskii S (2007) K-means++: the advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms, SODA '07, pp. 1027–1035. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA. http://dl.acm.org/citation.cfm?id=1283383.1283494
2. Bartlett A (2015) Electrochemical model-based state of charge and state of health estimation of lithium-ion batteries. Ph.D. thesis, Ohio State University
3. Baudat G, Anouar F (2001) Kernel-based methods and function approximation. In: IJCNN'01. International joint conference on neural networks. Proceedings (Cat. No.01CH37222), vol 2, pp 1244–1249. https://doi.org/10.1109/IJCNN.2001.939539
4. Bizeray A, Zhao S, Duncan S, Howey D (2015) Lithium-ion battery thermal-electrochemical model-based state estimation using orthogonal collocation and a modified extended Kalman filter. J Power Sources 296:400–412. https://doi.org/10.1016/j.jpowsour.2015.07.019
5. Brunton SL, Proctor JL, Kutz JN (2016) Discovering governing equations from data by sparse identification of nonlinear dynamical systems. In: Proceedings of the national academy of sciences. https://doi.org/10.1073/pnas.1517384113. http://www.pnas.org/content/early/2016/03/23/1517384113.abstract
6. Burges CJ (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Discov 2(2):121–167. https://doi.org/10.1023/A:1009715923555
7. Chinesta F, Cueto E, Abisset-Chavanne E, Louis Duval J, El Khaldi F (2018) Virtual, digital and hybrid twins: a new paradigm in data-based engineering and engineered data. Arch Comput Methods Eng. https://doi.org/10.1007/s11831-018-9301-4
8. Chinesta F, Huerta A, Rozza G, Willcox K (2015) Encyclopedia of computational mechanics, chap. Model order reduction. Wiley, London
9. Chinesta F, Keunings R, Leygue A (2013) The proper generalized decomposition for advanced numerical simulations: a primer. Springer, Berlin Incorporated
10. Cueto E, Gonzlez D, Alfaro I (2016) Proper generalized decompositions: an introduction to computer implementation with Matlab, 1st edn. Springer, Berlin Incorporated
11. Domenico DD, Stefanopoulou A, Fiengo G (2009) PSM: lithium-ion battery state of charge (SOC) and critical surface charge (CSC) estimation using an electrochemical model-driven extended Kalman and filter. ASME J Dyn Syst Meas Control
12. Doyle M, Fuentes Y (2003) Computer simulations of a lithium-ion polymer battery and implications for higher capacity next-generation battery designs. J Electr Soc 150(6):A706–A713. https://doi.org/10.1149/1.1569478. http://jes.ecsdl.org/content/150/6/A706.abstract
13. Doyle M, Fuller TF, Newman J (1993) Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. J Electr Soc 140(6):1526–1533. https://doi.org/10.1149/1.2221597. http://jes.ecsdl.org/content/140/6/1526.abstract
14. Forman JC, Moura SJ, Stein JL, Fathy HK (2012) Genetic identification and fisher identifiability analysis of the doyle-fuller-newman model from experimental cycling of a lifepo4 cell. J Power Source 210(Complete):263–275. https://doi.org/10.1016/j.jpowsour.2012.03.009
15. Fuller TF, Doyle M, Newman J (1994) Simulation and optimization of the dual lithium ion insertion cell. J Electr Soc 141(1):1–10. https://doi.org/10.1149/1.2054684. http://jes.ecsdl.org/content/141/1/1.abstract
16. González D, Badías A, Alfaro I, Chinesta F, Cueto E (2017) Model order reduction for real-time data assimilation through extended Kalman filters. Comput Methods Appl Mech Eng 326(Supplement C):679–693. https://doi.org/10.1016/j.cma.2017.08.041
17. Gonzalez D, Chinesta F, Cueto E (2018) Learning corrections for hyperelastic models from data (submitted)
18. González D, Chinesta F, Cueto E (2018) Thermodynamically consistent data-driven computational mechanics. Contin Mech Thermodyn. https://doi.org/10.1007/s00161-018-0677-z
19. Ham J, Lee DD, Mika S, Schölkopf B (2004) A kernel view of the dimensionality reduction of manifolds. In: Proceedings of the twenty-first international conference on machine learning, ICML '04, pp 47. ACM, New York, NY, USA. https://doi.org/10.1145/1015330.1015417. http://doi.acm.org/10.1145/1015330.1015417

20. Hariharan KS, Tagade P, Ramachandran DS (2018) Mathematical modeling of lithium batteries—from electrochemical models to state estimator algorithms. Springer, London
21. Hofmann T, Schölkopf B, Smola AJ (2008) Kernel methods in machine learning. Ann Stat 36(3):1171–1220. http://www.jstor.org/stable/25464664
22. Hu X, Li S, Peng H (2012) A comparative study of equivalent circuit models for li-ion batteries. J Power Sources 198:359–367. https://doi.org/10.1016/j.jpowsour.2011.10.013
23. Huang W, Cao L, Sun F, Zhao D, Liu H, Yu S (2016) Learning stable linear dynamical systems with the weighted least square method. In: Proceedings of the twenty-fifth international joint conference on artificial intelligence, IJCAI'16, pp 1599–1605. AAAI Press. http://dl.acm.org/citation.cfm?id=3060832.3060844
24. Huang W, Harandi M, Zhang T, Fan L, Sun F, Huang J (2017) Efficient optimization for linear dynamical systems with applications to clustering and sparse coding. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in neural information processing systems vol 30, pp 3444–3454. Curran Associates, Inc. http://papers.nips.cc/paper/6936-efficient-optimization-for-linear-dynamical-systems-with-applications-to-clustering-and-sparse-coding.pdf
25. Ibáñez Pinillo R, Abisset-Chavanne E, Ammar A, González D, Cueto E, Huerta A, Louis Duval J, Chinesta F (2018) A multidimensional data-driven sparse identification technique: the sparse proper generalized decomposition. Complexity 2018:1–11. https://doi.org/10.1155/2018/5608286
26. Kaiser E, Kutz JN, Brunton SL (2018) Discovering conservation laws from data for control. arXiv preprint arXiv:1811.00961
27. Karhunen K (1946) Uber lineare methoden in der wahrscheinlichkeitsrechnung. Acad. Sci. Fennicae, ser. Al. Math. Phys, Ann, p 37
28. Kawashima H, Matsuyama T (2005) Hierarchical clustering of dynamical systems based on eigenvalue constraints. In: Singh S, Singh M, Apte C, Perner P (eds) Pattern recognition and data mining. Springer, Berlin, pp 229–238
29. Kevrekidis Y, Samaey G (2010) Equation-free modeling. Scholarpedia 5(9):4847
30. Kirchdoerfer T, Ortiz M (2016) Data-driven computational mechanics. Comput Methods Appl Mech Eng 304:81–101. https://doi.org/10.1016/j.cma.2016.02.001
31. Kutz JN (2013) Data-driven modeling & scientific computation: methods for complex systems & big data. Oxford University Press Inc, New York
32. Kutz JN, Brunton SL, Brunton BW, Proctor JL (2016) Dynamic mode decomposition: data-driven modeling of complex systems. SIAM-Society for Industrial and Applied Mathematics, Philadelphia
33. Lee LJ, Chemistruck A, Plett G (2012) Discrete-time realization of transcendental impedance functions, with application to modeling spherical solid diffusion. J Power Sources 206:367–377. https://doi.org/10.1016/j.jpowsour.2012.01.134
34. Lee LJ, Chemistruck A, Plett G (2012) One-dimensional physics-based reduced-order model of lithium-ion dynamics. J Power Sources 220:430–448. https://doi.org/10.1016/j.jpowsour.2012.07.075
35. Lee LJ, Aldrich LL, Stetzel DK, Plett G (2014) Extended operating range for reduced-order model of lithium-ion cells. J Power Sources 255:85–100. https://doi.org/10.1016/j.jpowsour.2013.12.134
36. Laughlin RB, Pines D (2000) The theory of everything. In; Proceedings of the national academy of sciences 97(1):28–31. https://doi.org/10.1073/pnas.97.1.28. http://www.pnas.org/content/97/1/28.abstract
37. Lloyd S (2006) Least squares quantization in PCM. IEEE Trans Inf Theor 28(2):129–137. https://doi.org/10.1109/TIT.1982.1056489
38. Loève MM (1963) Probability theory. The university series in higher mathematics, 3rd edn. Van Nostrand, Princeton
39. Meyer M, Matthies HG (2003) Efficient model reduction in nonlinear dynamics using the Karhunen–Loève expansion and dual-weighted-residual methods. Comput Mech 31(1–2):179–191. https://doi.org/10.1007/s00466-002-0404-1
40. Moura S, Chaturvedi N, Krstic M (2014) Adaptive pde observer for battery SOC/SOH estimation via an electrochemical model. ASME J Dyn Syst Meas Control 136:011015–011026
41. Moura SJ, Bribiesca Argomedo F, Klein R, Mirtabatabaei A, Krstic M (2017) Battery state estimation for a single particle model with electrolyte dynamics. IEEE Trans Control Syst Technol 25(2):453–468. https://doi.org/10.1109/TCST.2016.2571663. https://hal.archives-ouvertes.fr/hal-01478641
42. Moya B, Gonzalez D, Alfaro I, Chinesta F, Cueto E (2019) Learning slosh dynamics by means of data. Comput Mech. https://doi.org/10.1007/s00466-019-01705-3
43. Niroomandi S, Alfaro I, Cueto E, Chinesta F (2008) Real-time deformable models of non-linear tissues by model reduction techniques. Comput Methods Progr Biomed 91(3):223–231. https://doi.org/10.1016/j.cmpb.2008.04.008
44. Plett GL (2004) Extended Kalman filtering for battery management systems of lipb-based hev battery packs: part 2. Modeling and identification. J Power Sources 134(2):262–276. https://doi.org/10.1016/j.jpowsour.2004.02.032
45. Robinson LD, García RE (2015) Dualfoil.py: porous electrochemistry for rechargeable batteries. https://doi.org/10.21981/D3KP7TS5M. https://nanohub.org/resources/dualfoil
46. Rozza G, Huynh D, Patera A (2008) Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations—application to transport and continuum mechanics. Arch Comput Methods Eng 15(3):229–275
47. Santhanagopalan S, White RE (2006) Online estimation of the state of charge of a lithium ion cell. J Power Sources 161(2):1346–1355. https://doi.org/10.1016/j.jpowsour.2006.04.146
48. Smith K, Wang CY (2006) Solid-state diffusion limitations on pulse operation of a lithium ion cell for hybrid electric vehicles. J Power Sources 161(1):628–639. https://doi.org/10.1016/j.jpowsour.2006.03.050
49. Smith KA (2010) Electrochemical control of lithium-ion batteries [applications of control]. IEEE Control Syst Mag 30(2):18–25. https://doi.org/10.1109/MCS.2010.935882
50. Smith KA, Rahn C, Wang CY (2007) Control oriented ID electrochemical model of lithium ion battery. Energy Convers Manag 48:2565–2578
51. Smith KA, Rahn C, Wang CY (2008) Model-based electrochemical estimation of lithium-ion batteries, pp 714–719. https://doi.org/10.1109/CCA.2008.4629589
52. Stetzel DK, Aldrich LL, Trimboli M (2015) Plett G (2015) Electrochemical state and internal variables estimation using a reduced-order physics-based model of a lithium-ion cell and an extended Kalman filter. J Power Sour. https://doi.org/10.1016/j.jpowsour.2014.11.135
53. Torchio M, Magni L, Gopaluni B, Braatz R, Raimondo D (2016) Lionsimba: a Matlab framework based on a finite volume model suitable for li-ion battery design, simulation, and control. J Electr Soc 163:A1192–A1205. https://doi.org/10.1149/2.0291607jes

54. Wang Q, Jiang B, Li B, Yan Y (2016) A critical review of thermal management models and solutions of lithium-ion batteries for the development of pure electric vehicles. Renew Sustain Energy Rev 64:106–128. https://doi.org/10.1016/j.rser.2016.05.033
55. Winter M, Moser S, Schoenewolf S, Taube J, Herzog HG (2015) Average model of a synchronous half-bridge DC/DC converter considering losses and dynamics. In: Proceedings of the 11th international modelica conference

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.