Modeling and State Estimation for Lithium Sulfur Batteries as a Piecewise Affine System

Guillaume Goujard, Chitra Dangwal, Preet Gill, Dylan Kato, Scott J. Moura

***Abstract*—Lithium-sulfur (Li-S) is a promising battery chemistry for applications demanding high energy densities, such as electrified aircraft and heavy-duty trucks, among others. A critical challenge in modeling the Li-S chemistry lies in the use of differential algebraic (DAE) equations for representing the electrochemical dynamics. Due to their constrained and stiff nature, these equations are not conducive to real-time state estimation. In this study, we propose a novel approach to constrained state estimation for Li-S batteries by integrating a piecewise affine (PWA) model into a moving horizon estimation (MHE) framework. We begin by deriving the PWA model using a linear tree algorithm based on data obtained from simulations of a calibrated DAE model. We further leverage the unique structural advantages of the proposed PWA model to formulate a real-time state estimation algorithm grounded in a mixed-integer quadratic program. Overall, our initial findings, based on a single constant current trajectory, demonstrate that our approach offers an accurate and computationally efficient method for modeling and state estimation of Li-S batteries. The coupled PWA-MHE framework effectively captures the dynamics of the DAE system, even in the presence of high observational noise (20mV).**

***Index Terms*—state estimation, decision tree, control theory, machine learning**

I. Introduction

With growing interest in Li-S batteries, significant effort has been directed toward material and chemistry research. Such thrusts include better understanding of fundamental reactions, improvement of cell chemistry, and design of the cell electrolyte and electrodes [1]–[3]. While these efforts are critical for advancing Li-S technology, there is a need for more focus on modeling and control research to accelerate Li-S commercialization.

Battery models in literature range from high fidelity - computation intensive models to low fidelity - computation efficient models [4], [5]. Model selection largely depends on specific use case. First principle models provide insights about the system thermodynamics, kinetics, and transport behavior, but they suffer from drawbacks such as requiring prior knowledge of electro-thermochemistry, mathematical complexity, high computational costs, and model calibration difficulties. On the other hand, spatially averaged lumped models like equivalent circuit models (ECMs) or single particle model (SPM) are easier to simulate, calibrate and can be easily applied for battery management system (BMS) applications. In practice the tradeoffs between computational complexity, interpretability and accuracy are key factors when choosing the best model for a given problem.

Fotouhi et al. give an overview of the technological readiness of Li-S batteries from the standpoint of modeling and estimation [6] . Estimation techniques using ECMs for Li-S batteries are well studied in the literature. The key limitation of using ECMs is that they lack of information of electrochemical processes occurring inside the Li-S cell. Electrochemical models, on the other hand, give better insight into the governing physics inside the battery. Zero-Dimensional [4] models give insight into the reaction kinetics of the Li-S cell along with phenomena like precipitation and shuttle effect. 1D models [4] capture both the reaction kinetics and transports dynamics. Both 0D and 1D models capture the distinctive shape of a low discharge current voltage profile accurately in both "plateau" regimes. The modeling and state estimation for these electrochemical models are challenged by complicated reaction pathways, complex DAE system dynamics, weak local observability, and the lack of an estimator design tool for DAE. Some attempt has been made to circumvent some of these challenges while retaining the interpretability provided by electrochemical models. In [7], Xu reduced the 0D electrochemical DAE model to an ODE which was further used for state estimation using an unscented Kalman filter (UKF). A key challenge is that the resulting ODE model has a very stiff structure that can cause issues in its numerical implementation. In [8], state estimation for a 0D DAE model using an extended Kalman filter (EKF) is shown. Despite good performance of the estimator in this work, the measurement noise studied was very low compared to a real life setting, suggesting high measurement noise may pose an issue for EKF estimation.

To address the above challenges associated with standard DAE electrochemical models, we propose a PWA Li-S model that well-approximates the dynamics of a 0D DAE model. To this effect, a learning algorithm is developed in the same spirit as decision and classification tree methods [9]. This PWA framework simplifies the nonlinear DAE model equations into a set of disjoint linear systems. The key advantage is to retain the advantageous structure of linear systems while simultaneously capturing the evolution of electrochemical states. We show that this model can not only simulate states as accurately as the DAE models, but its structure makes it amenable to more advanced state estimation techniques like MHE [10] [11]. The most similar work might be [12], which uses a PWA approximation of a Newman-style lithium-ion battery model for the output equation for model predictive control.

The main contributions of this paper are summarized as follows:

- Development of a piece-wise affine learning algorithm that accurately approximates the 0D electrochemical DAE a Li-S battery cell.
- Constrained state estimation of Li-S battery using moving horizon estimation on our piece-wise affine model.

One of the limitations in the field of Li-S battery is the availability of experimental data. Most of the existing literature analyses the battery behavior in low current discharge conditions [4], [5], [7]. Due to this, we focus on modeling cell behavior during discharge.

The paper is organized as follows. In section II, we first introduce the DAE model taken from the literature to simulate the dynamics of a Li-S battery cell. After analyzing various related shortcomings of using this model for state estimation, we introduce PWA systems. In section III, a learning algorithm is then detailed to train a "PWATree" over a dynamical system dataset. The tree is then encoded into a moving horizon estimation framework. In section IV, we finally present results related to our use case: both the offline fit and online estimation with noise undergoing a full discharge cycle under constant current are presented.

II. DAE Model

A. Model

This section summarizes the Li-S model chosen as the baseline formulation for this study. We chose a zero-dimensional electrochemical model developed in our previous work [8] and originally adopted from [5]. The 0D model captures the reaction kinematics at the cathode while the transport dynamics are ignored. We chose the 0D electrochemical model as a baseline, since it captures the voltage dynamics and information about the chemical state of the cathode.

During the discharge process, Li-ions liberated from the anode move toward cathode, and reduce the sulfur species to different Li-polysulfide species via a series of complex electrochemical reactions. The zero-dimensional model in this study considers a 3-step electrochemical reaction given by :

$$\frac{3}{8}\mathrm{S}_8^0 + \mathrm{e}^- \longleftrightarrow \frac{1}{2}\mathrm{S}_6^{2-} \tag{1}$$

$$\mathrm{S}_6^{2-} + \mathrm{e}^- \longleftrightarrow \frac{3}{2}\mathrm{S}_4^{2-} \tag{2}$$

$$\frac{1}{6}\mathrm{S}_4^{2-} + \mathrm{e}^- \longleftrightarrow \frac{2}{3}\mathrm{S}^{2-} \downarrow \tag{3}$$

In the fully charged condition, the cathode is composed of elemental sulfur S_8^0. During discharge S_8^0 reduces to high order polysufides, which further reduce to low order sulfide $\mathrm{Li_2S}$, which further precipitates. The high order polysufides in a Li-S battery are soluble in the electrolyte. This causes a parasitic loss called as the 'shuttle effect', which is the movement of these soluble species back and forth between the electrodes. It is a key phenomenon to monitor in Li-S batteries.

The 0D electrochemical model takes the form of a differential algebraic equation (DAE) system (4).

$$\begin{aligned} \dot{x} &= f(x,z,u) \\ 0 &= g(x,z,u) \\ y &= h(x,z,u) \end{aligned} \tag{4}$$

where $x \in \mathbb{R}^5$, is the mass of sulfur species $[S_8^0,\ S_6^{2-},\ S_4^{2-},\ S^{2-}, S_p]$ involved in the reaction, z are the algebraic states that represent the currents involved in each electrochemical reaction step $[i_1,\ i_2,\ i_3]$ in (1)-(3). Finally, u is the input current and y is the output voltage. The differential function $f(x,z,u)$ is shown in (5).

$$f(x,z,u) = \begin{bmatrix} -\frac{3}{8}\frac{n_{\mathrm{S}8}M_{\mathrm{S}}}{n_eF}i_{H1} - k_s x_1 \\ \frac{1}{2}\frac{n_{\mathrm{S}6}M_{\mathrm{S}}}{n_eF}i_{H1} - \frac{n_{\mathrm{S}6}M_{\mathrm{S}}}{n_eF}i_{H2} - k_s x_2 \\ \frac{3}{2}\frac{n_{\mathrm{S}4}M_{\mathrm{S}}}{n_eF}i_{H2} - \frac{1}{6}\frac{n_{\mathrm{S}4}M_{\mathrm{S}}}{n_eF}i_L \\ \frac{2}{3}\frac{n_{\mathrm{S}}M_{\mathrm{S}}}{n_eF}i_L - k_p x_5(x_4 - S_*^{2-}) \\ k_p x_5(x_4 - S_*^{2-}) \end{bmatrix} \tag{5}$$

The algebraic constraints of the system, $0 = g(x,z,u)$, are summarized in the equations (6), (7), (9), (10) below. The currents $[i_1,\ i_2,\ i_3]$ associated with each electrochemical reaction (1)-(3) are modeled using Butler-Volmer kinetics:

$$\begin{aligned} i_j = -i_j^0 a_r \{ &\prod_i \left(\frac{x_i}{x_i^0}\right)^{s_{i,j}} e^{\frac{F}{2RT}\eta_j} \\ &- \prod_i \left(\frac{x_i}{x_i^0}\right)^{-s_{i,j}} e^{-\frac{F}{2RT}\eta_j} \end{aligned} \tag{6}$$

The sum of currents in each reaction i_j equals the total input current flowing through the battery:

$$I = \sum_j i_j \tag{7}$$

Each overpotential term η_j in (6) is related to the output voltage and the standard reduction potential for each reaction in (1)-(3).

$$h(x,z,u) = V(t) = E_j + \eta_j \tag{8}$$

The standard reduction potential is modeled using Nernst equation (9).

$$E_j = E_j^0 - \frac{RT}{n_jF}\sum_i s_{i,j} ln\left(\frac{x_i}{n_{S_i}M_{\mathrm{S}8}v}\right) \tag{9}$$

The active reaction area a_r in the cathode decreases as sulfur precipitate S_p increases, and is modeled as follows:

$$a_r = a_r^0(1 - \omega \cdot x_5)^\gamma \tag{10}$$

Further details about the DAE model and its parameters associated with (1)-(3) can be found in [13].

B. Challenges

1) Modeling: The DAE system and its reduced ODE form (4) from [7] are numerically challenging to solve. Instabilities arise when states approach zero or take complex values in equation (6). Therefore, initial conditions for low-order polysulfides $S_1 - S_4$ must be non-zero, even when starting fully charged with S_8 or S_6. The model is notably unstable when states $S_1 - S_4$ are near zero. Fig. 1 shows sulfur species evolution during constant current discharge. After 4000 seconds, S_8 is depleted, and lower-order sulfides emerge. Independent Gaussian noise ($\sigma = 10^{-6}$ gm) was added to the states. Despite its small spread, it considerably affects the voltage output plotted in red in the lower subplot. This shows the stiffness of the model output function [7],as small noise in state gets considerably amplified in the output. Model sensitivity to noise is here evident. However, the piecewise affine output function proposed in section (III) and plotted in blue mitigates this voltage noise by reducing the voltage output to its best affine approximation.

2) Estimation: Various algorithms like Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) are tempting options for Li-S battery state estimation. EKF is suitable for systems allowing local linear approximations, while UKF is better for nonlinear systems.

EKF requires solving algebraic equations for new state estimates, a computationally intensive step. Also as shown in Fig. 1, the output function is highly sensitive to state estimates, which makes EKF not ideal for this application. UKF, while more apt for nonlinear systems, struggles with ensuring non-negative sigma points, which is critical given the system's sensitivity to negative states for sulfur species. This limitation is also acknowledged in [7]. The paper addresses this fundamental challenge.

III. A PIECE-WISE AFFINE APPROXIMATION

Although the state dynamics presented in (4) are a general nonlinear system, they have a particular structure. For example, observe how the mass of S_8 in Fig. 1 decreases linearly from start to time 0.4e4s. More generally, (5) behaves linearly along certain regions of the state-space if the (i) shuttle effect and (ii) algebraic constraints are neglected. In this section, we introduce a data-driven tree-based PWA approximation to the nonlinear DAE system that leverages this particular structure. Then, the resulting system and its mathematical structure are exploited for state estimation. A PWA system defines both state update and observation output equations as PWA over a partition $\mathcal{M} = \{\mathbb{E}_i\}_{i \leq n_M}$ of the space.

$$\begin{aligned} x_{t+1} &= A_i x_t + B_i u_t + f_i \\ y_t &= C_i x_t + D_i u_t + g_i \\ &\forall (x_t, u_t) \in \mathbb{E}_i \end{aligned} \tag{11}$$

Note 1: f_i, g_i are constant offsets (different from functions f, g in (4)), thus why we call this system piecewise *affine*. *Note 2:* We wrote the following section for a controllable PWA system to keep the method general to accommodate varying current. However, in the results (section IV), we trained a PWATree over a constant current discharge profile. This boils down to learning a PWA autonomous system where B_i, D_i equal zero.

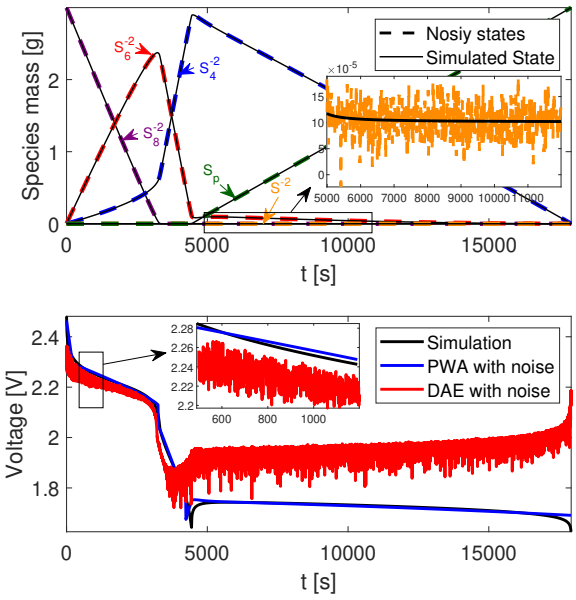

Fig. 1. Effect of small-magnitude (standard deviation of 10^{-6}) process noise on output voltage $h(x, u)$ for DAE vs PWA system. While noise is magnified in the DAE case, it is of the same order in the PWA case.

A. Notation

This section relates equivalences between linear algebra (dynamical systems) and machine learning (binary decision tree) concepts. It serves as reference to understand the relationship between a PWA system and PWATREE. The polyhedral (resp. discrete index) notation will be useful in the state estimation (resp. learning) section.

- *Dataset*: our dataset is sampled trajectories of associated states, controls, and outputs of the system (4). Denote it as $\mathcal{X} = (X, U, Y)$. Each element is a timeseries e.g. $X = (x_t)_{t \leq n}$. A state x_t (resp. control u_t) has dimension p (resp. m). For the 0D Li-S model, $p = 5$.
- *Split*: A split (or orthogonal hyperplane) is parameterized by a state (resp. control), an index j and a threshold τ_j. It separates the state and control space into two orthogonal half-spaces. Upper and lower half-spaces are distinguished by their direction $\alpha \in \{-1, 1\}$.

$$\Lambda_j = \left\{ \begin{bmatrix} x & u \end{bmatrix} \in \mathbb{R}^{p+m} \,\middle|\, \alpha \begin{bmatrix} x & u \end{bmatrix}_j \leq \tau_j \right\} \tag{12}$$

- *Rectangle*: A rectangle is a polytopic set defined as a non-empty intersection of orthogonal hyperplanes.

$$\mathbb{E} = \bigcap_{j=1}^{k} \Lambda_j \tag{13}$$

- $\mathcal{M} = \{\mathbb{E}_i\}_{i \leq n_M}$ partitions the space if for all (x, u), there exists a unique rectangle $\mathbb{E}_i$ where (x, u) be-

longs. Introducing the indicator binary variable $\delta^i = \mathbb{1}((x,u) \in \mathbb{E}_i)$, it is equivalent to:

$$\sum_{i=1}^{n_M} \delta_t^i = 1 \tag{14}$$

- The projection of dataset $\mathcal{X}$ to hyper-rectangles yield discrete index subsets: $R_i = \{t \in [n], (x_t, u_t) \in \mathbb{E}_i\}$.
- The parameter $\theta_i = (A_i, B_i, f_i, C_i, D_i, g_i)$ stores the linear system parameter of equation (11).
- *PWA System*: Finally, a PWA system is abstracted by the general parameter $\Theta = \{\mathbb{E}_i, \theta_i\}_{i \leq n_M}$ that contains the partitions and parameters.

Our objective is to find the best PWA system that minimizes a certain objective function, detailed in the next subsection.

B. Learning objective

Given a PWA system Θ, an initial condition x_0 and a timeseries of exogenous controls U, a trajectory of states and observation, a.k.a. dataset, $\hat{\mathcal{X}} = (\hat{X}, \hat{Y}, U)$ is simulated by simply *pushing forward* through (11). Hence we write: $\hat{\mathcal{X}} = \hat{\mathcal{X}}(\Theta, x_0, U)$. To facilitate SOC estimation, it is critical for our PWA system to be accurate at estimating the true nonlinear state trajectory. This is what the scoring function (15) measures. On the one hand, it minimizes the simulated state tracking error $x_t - \hat{x}_t$ while minimizing the voltage output error defined as $y_t - \hat{y}_t$.

$$\mathcal{S}(\mathcal{X}, \hat{\mathcal{X}}) = \sum_{t=0}^{T} ||x_t - \hat{x}_t||^2 + ||y_t - \hat{y}_t||^2 \tag{15}$$

Note: if our objective was to learn a decision tree, the scoring function (or splitting criterion) would simply be the training variance of the resulting tree (see [14]).

C. Learning algorithm

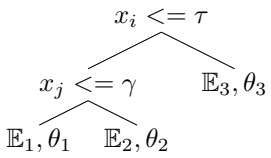

Fig. 2. Tree representation of a PWA system

Any PWA system defined over hyper-rectangles (13) can be encoded into a tree format, like illustrated in Fig. 2. This representation facilitates a tree-based learning algorithm where splits are selected to minimize a certain objective function (15). The parameters in each rectangle $\mathbb{E}_i$ can be estimated via a convex optimization sub-routine $\mathcal{O}$. Sub-routine is given by the mathematical program (16), which regresses the parameters θ of a linear system over a dataset $\mathcal{X}$. An L_1 penalty encourages sparsity in the estimated parameters. This sub-routine is a quadratic program and will be called to quickly evaluate the quality of a split which generates a new partition of the state-control space. The proposed method in Algorithm 1 follows the training algorithm of classification and regression trees. We recursively loop over *candidate* binary splits that partition the state-control space. To reduce computational burden, the evaluated splits are sampled from the quantiles of the distributions of each state. The single split which reduces the scoring function the most is selected. This generates a PWA system that yields the most faithful representation of the nonlinear system. The split is then enacted by recursively calling on the procedure for the left and right children $\mathbb{E}_l, \mathbb{E}_r$. Finally, Δ^{max} fixes the maximum depth of the tree while n_{min} makes sure that enough data is collected in the leaves to run the subroutine.

$$\begin{aligned} \mathcal{O}(x,y) = \min_{\theta} \quad & \sum_{t=1}^{N} ||w_t||^2 + \sum_{t=1}^{N} ||v_t||^2 + \lambda ||\theta||^1 \\ \text{subject to} \quad & x_{t+1} = Ax_t + Bu_t + f + w_t \\ & y_t = Cx_t + Du_t + g + v_t \\ & \theta = (A, B, f, C, D, g) \end{aligned} \tag{16}$$

where w_t, v_t represent the error between the (resp.) state and output trajectories in dataset $\mathcal{X}$ and trajectories predicted by the linear model parameterized by θ.

Algorithm 1: PWATREE

Input: Data $\mathcal{X} = \{(x_i, x_{i+1}, y_i)\}_{i \in I}$ current PWA Tree Θ_0, current depth Δ
Exogenous Data: initial dataset $\mathcal{X} = \{(x_t, x_{t+1}, y_t)\}$
Hyperparameters: $\{n_{min}, \Delta^{max}\}$, function $\mathcal{S}$, subroutine $\mathcal{O}$
Result: PWA system $\Theta = \{\mathbb{E}_i, \theta_i\}_{i \leq n_M}$
Compute simulation dataset $\hat{\mathcal{X}}$ from Θ_0
Compute initial score $v = \mathcal{S}(\mathcal{X}, \hat{\mathcal{X}})$
Set $v_{min} \leftarrow v$, SPLIT $\leftarrow$ FALSE
if $\Delta < \Delta^{max}$ *and* $|I| \geq n_{min}$ **then**
  for *Feature* j *and threshold* τ **do**
    Left child $R_l = \{i \in [n] : x_{i,j} < \tau\}$
    Compute affine parameters $\theta_l = \mathcal{O}(\mathcal{X}_{R_l})$
    Right child $R_r = \{i \in [n] : x_{i,j} > \tau\}$
    Compute affine parameters $\theta_r = \mathcal{O}(\mathcal{X}_{R_r})$
    Generate tentative PWATREE $\hat{\Theta} = \Theta_0 \cup \{(\mathbb{E}_l, \theta_l), (\mathbb{E}_r, \theta_r)\}$
    Compute simulation dataset $\hat{\mathcal{X}}$ from $\hat{\Theta}$
    Compute simulation score $v = \mathcal{S}(\mathcal{X}, \hat{\mathcal{X}})$
    if $v < v_{min}$ **then**
      Update $j^* \leftarrow j$, $\tau^* \leftarrow \tau$
      Update $v_{min} \leftarrow v$, SPLIT $\leftarrow$ TRUE Update $\Theta^* \leftarrow \hat{\Theta}$
    end
  end
end
if SPLIT *is* TRUE **then**
  $\Theta_l = \text{PWATREE}(\mathcal{X}_{R_l}, \Theta^*, \Delta + 1)$
  $\Theta_r = \text{PWATREE}(\mathcal{X}_{R_r}, \Theta^*, \Delta + 1)$
  $\Theta_0 \leftarrow \Theta_0 \cup \{(\mathbb{E}_l, \theta_l), (\mathbb{E}_r, \theta_r)\}$
end
return Θ_0

D. Moving horizon estimation

Moving horizon estimation (MHE) is an optimization technique that is employed to estimate the unknown system states given a state update, observation and potentially algebraic functions. It has been widely studied in diverse contexts [10] [11]. While this method has the disadvantage of being computationally expensive relative to state observers, it has particular advantages in our application. First, it imposes no assumptions on the noise process. Second, one can include linear constraints thus enabling constrained state estimation, a particularly helpful feature for the 0D Li-S model. The formulation of (17) introduces a baseline MHE formulation. Different flavors of MHE exist in the literature, but usually MHE presents 3 objectives to minimize. First, minimize the error between measured (y_τ) and modeled observations. Then, fit the state trajectory to the dynamical system model. Finally, fit the last known estimate (or initial condition) $\bar{x}_{t-h}$. The third term is an optional relaxation of the algebraic constraints $g(x) = 0$. Horizon parameter $h \in \mathbb{N}$ (not to be confused with function h in (4)) is the window size and can be tuned. The larger h is, the more accurate the estimation will be but at the expense of computation time. Note that the state estimates are made to belong to a polytopic set $\mathcal{C}$. Finally, some weights can be associated to each of those objectives according to prior knowledge of the system.

$$\begin{aligned}\text{argmin}_{\mathbf{x}\in\mathcal{C}} \sum_{\tau=t-h}^{t} &||y_\tau - h(\mathbf{x}_\tau)||_R^2 + ||\mathbf{x}_{\tau+1} - f(\mathbf{x}_\tau, u_\tau)||_Q^2 \\ &+ ||g(\mathbf{x}_{t-h})||_\xi^2 + ||\mathbf{x}_{t-h} - \bar{x}_{t-h}||_\Pi^2 \end{aligned} \tag{17}$$

The prospect of funneling a PWA system inside a MHE framework is appealing. MHE transforms to a Quadratic Program in the linear time invariant case. Other works [10] [15] have shown that a PWA-MHE can be formulated as a Mixed Integer Quadratic Program (MIQP). This formulation can be solved near-optimally by using the full extent of modern branch and bound solvers such as GUROBI.

E. PWA-MHE for Li-S battery

Replacing the nonlinear h and f functions from program (17) by our PWA system yields program (18). The algebraic constraints we consider are two-fold:

- As detailed in the notations, the sum of the indicators must equal one.
- Mass conservation is verified by constraining the sum of the normalized states to one. We later relax this constraint to allow for the Li-S shuttle effect.

$$\begin{aligned}\min_{\mathbf{x}} \quad & \sum_{\tau=t-h}^{t} ||\mathbf{v}_\tau||_R^2 + ||\mathbf{w}_\tau||_Q^2 + ||1 - 1^\top \mathbf{x}_\tau||_\xi^2 \\ & + ||\mathbf{x}_{t-h} - \bar{x}_{t-h}||_\Pi^2 \\ \text{subject to} \quad & \mathbf{x}_{\tau+1} = \sum_{i=1}^{n_M} \boldsymbol{\delta}_{\boldsymbol{\tau}}^{\boldsymbol{i}} \cdot \left(A_i \mathbf{x}_\tau + B_i u_\tau + f_i\right) + \mathbf{w}_\tau \\ & y_\tau = \sum_{i=1}^{n_M} \boldsymbol{\delta}_{\boldsymbol{\tau}}^{\boldsymbol{i}} \cdot \left(C_i \mathbf{x}_\tau + D_i u_\tau + g_i\right) + \mathbf{v}_\tau \end{aligned}$$

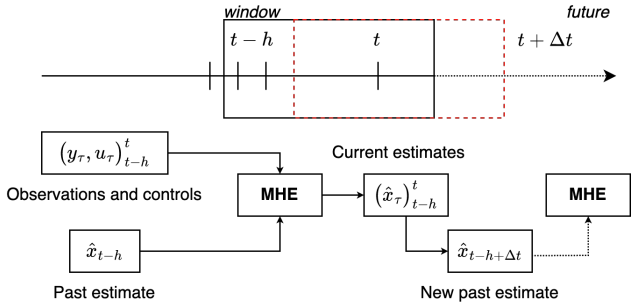

Fig. 3. Operational framework for moving horizon estimation: Two hyperparameters are the window length h and the update period Δt. The MHE algorithm estimates states based on the previous h observations, controls, and state estimate at time $t-h$. Finally, we reuse the first index of the next window in the next MHE calculation.

$$\sum_{i=1}^{n_M} \boldsymbol{\delta}_{\boldsymbol{\tau}}^{\boldsymbol{i}} = 1, \qquad \mathbf{x}_\tau \in [0,1] \tag{18}$$

This program can be reformulated into linear form by introducing new binary optimization variables and additional constraints. We refer the reader to these articles for more information [10] [15].

We finally illustrate the pipeline tying the estimation algorithm to the datasets of observations and controls in Fig 3.

IV. Results

The PWAtree and MHE algorithms are both tested on the same full discharge cycle at low discharge current. Our future research interest looks to expand this appproach to varying current. The dataset is simulated by the high-fidelity DAE model (4). Key statistics are summed up in Table I. The dynamics were previously graphed on Fig. 1. Additional Gaussian noise was added to the voltage output equation to mimic real-life conditions. Its magnitude (standard deviation of 20mV) is an order of magnitude greater than the usual distribution of errors of modern voltage sensors (order of 1mV). This magnitude was selected to showcase the robustness of the estimator design. Estimating states for this chemistry and profile is already a significant contribution, as previous papers have predominantly looked at estimating the states (i) in the high plateau region only where observability is high and (ii) for low observation noise to avoid infeasibility in the observation function (see Section II-B). In this section, we will first assess the offline fit of the PWATree to the data, then examine the estimation results.

TABLE I
Simulation Parameters

Parameter	Value
Constant current u (A)	1
Time to discharge (s)	$89,634$
Capacity (Ah)	24.9
Sub-sample Period (s)	500
Timesteps in simulation dataset	179
Standard deviation of observation noise (mV)	20
Initial state	$\begin{bmatrix}1 & 0 & 0 & 0 & 0\end{bmatrix}$

A. Piecewise Affine Fitting

We implemented Algorithm 1 on the aforementioned dataset. This ultimately returns a PWATREE of depth 2. Each iteration of the algorithm finds the best splits that minimizes the scoring function (15). Table II shows how the scoring function (analogous to training loss) decreases as we grow the tree depth, in terms of mean square state estimation error and scoring function (15). As expected, the tree yields better score as the tree grows in depth.

TABLE II
PWA SCORES WITH DEPTH/ITERATION OF ALGORITHM 1

Depth/Iteration	MSE State	MSE Output	Score
0	3.94	0.75	4.69
1	0.90	0.28	1.18
2	0.15	0.13	**0.28**

Fig. 4 depicts the final PWATREE. The fitting algorithm remarkably learns *interpretable* divisions in the dynamics. For example, the splits on S_8 and S_6 mark the depletion of those species which are key transitional steps in the dynamics of the chemical reactions. Fig. 5 highlights with vertical dotted lines the moments in time when the PWA system switches.

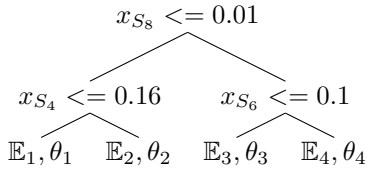

Fig. 4. PWAtree for Li-S battery constant current discharge

The state and output trajectories for the learned tree is presented Fig. 5. After three iterations of Algorithm 1, the PWA-trajectory is indistinguishable from the original nonlinear DAE model. Remarkably, even the characteristic lowest dip at time 20,000 sec is well approximated by a linear function in the states. This is due to a switch from $\mathbb{E}_2$ to $\mathbb{E}_1$ that allows for the output equation to change its form and in particular its intercept (precisely from $g_2 = 1.27$ to $g_1 = 1.16 \in \theta_1$ from Fig. 4). Thus, we see PWATREE accurately captures the dynamics of the states and the output function, and offers a compelling model approximation for state estimation.

B. Moving Horizon Estimation results

In this, we run a sequence of MHEs from a perturbed initial state and under noisy observations. Table III details the hyper-parameters used, Section III-D and Fig. 3 provide more detail on the timeline of this sequence. The value of the weights were hand-tuned to approximately equal the contribution of each objective function (10^{-4}) detailed in Program 17.

The results are presented in Fig. 6, including the noised voltage timeseries (lower subplot). Despite an offset initial condition, the MHE framework quickly converges within a small neighborhood of the actual values. This is explained

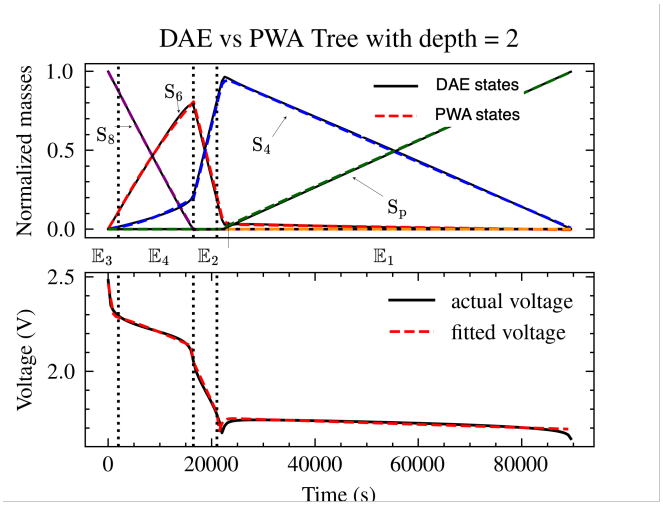

Fig. 5. Model to Model comparison for simulated dynamics for ground-truth DAE model (black solid) and optimal PWATREE (dashed color). The dynamics from the fitted tree closely follow those resulting from the nonlinear DAE model. The different splits where the PWA systems switches parameters are represented in dotted lines and the hyper-rectangles where the states lie are written between the subplots. The states leading to the transition from high to low plateau (time 20,000s) belong to the same PWA hyper-rectangle. This region is crucial for state estimation, as observability is lost in the low-plateau region, as analyzed in [7], [8].

TABLE III
MHE PARAMETERS

Parameter	Value
state constraint $\mathcal{C}$	$x_t \in (0,1)$, $\sum_i x_{i,t} = 1$
weights for output error R	1
state update error Q	10
past-estimate error Π	1
Length of window h	15
Update period Δt	5
Average of computation time (s)	3.4 (std 2.9)
Initial state	$[0.7 \quad 0.2 \quad 0.1 \quad 0 \quad 0]$

by the high voltage output measured at start. The only way for the MHE to match this high voltage input is to assign state S_8 to 1 which explains this fast convergence to the actual states. Another feature of the PWATREE approach is the ability to locate very precisely the switch time (dip time) of the transition from high to low plateau.

To showcase this feature, we run an instance of MHE over window A. of Fig. 6. We again start from an offset initial condition and and feed in 30 sampled observations from Time 15,000s to 30,000s. The internal states $\hat{x}_{t-h,t}$ are plotted along with the filtered observation $\hat{y}_t^{MHE}$ on Fig. 7. The states converges to their actual values thanks to the good identification of the dip at time 22,000s (dotted lines) vs actual dip time of 21,500s. For information, only state b. would be reported to the user after such a run.

After the dip, we rely on the accuracy of the linear approximation in the low plateau region to deliver accurate state estimates. Finally, it is important to highlight that the PWA-MHE can produce sulfur masses that are precisely zero (e.g., see the initial condition in Table III) yet constrained to be non-negative. This is not mathematically possible with the

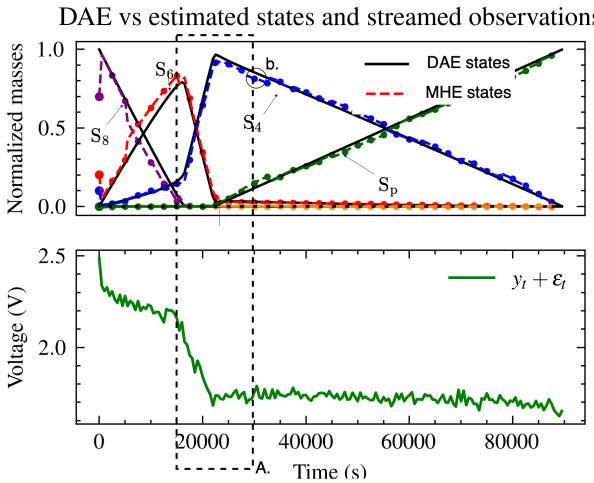

Fig. 6. State estimation results using PWA-MHE. The true states from the DAE model (in solid lines) are estimated by the algorithm despite the large measurement noise (standard deviation 20mV) and offset initial condition (first time step points). This figure is the result of running a sequence of MHE until the end while reporting the last estimate. In particular to estimate the circled state b., an MHE streamed a window A. of the past h-measurements.

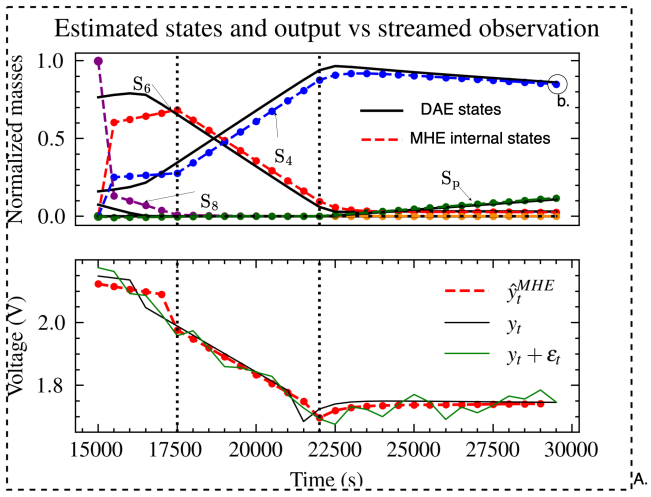

Fig. 7. One time-step MHE output. Despite a poor initialization (State S_8 concentrates all the mass), MHE is fitting the voltage observation (lower subplot in green) with the red curve over the window A. of past data. The state dynamics (upper subplot) are also estimated, such that their evolution minimizes the plant model error and deviation from initial condition. Finally, the last time step state b. is reported. Estimated states converge to true states.

original DAE model and a Kalman-based filter.

V. Conclusion and Future Work

In this article, we demonstrated how learning a PWA system through the use of a binary tree algorithm can yield an interpretable and tractable approximation to nonlinear dynamics. In the case of a Lithium Sulfur battery, a PWA system was learned with high accuracy. We further demonstrate the application of the PWATREE into a moving horizon estimation (MHE) algorithm. We use MHE to estimate states given noisy observations. This approach could be generalized to more challenging current profiles and other types of complex dynamical systems. Our future work will first involve generalizing the learning phase over a sequence of diverse charging and discharging profiles. This step is crucial to allow our estimation method to generalize to more challenging current profiles.

References

[1] C. D. Parke, L. Teo, D. T. Schwartz, and V. R. Subramanian, "Progress on continuum modeling of lithium–sulfur batteries," *Sustainable Energy & Fuels*, vol. 5, no. 23, pp. 5946–5966, 2021.

[2] S. Wang, S. Feng, J. Liang, Q. Su, F. Zhao, H. Song, M. Zheng, Q. Sun, Z. Song, X. Jia *et al.*, "Insight into mos2–mon heterostructure to accelerate polysulfide conversion toward high-energy-density lithium–sulfur batteries," *Advanced Energy Materials*, vol. 11, no. 11, p. 2003314, 2021.

[3] T.-Z. Hou, H.-J. Peng, J.-Q. Huang, Q. Zhang, and B. Li, "The formation of strong-couple interactions between nitrogen-doped graphene and sulfur/lithium (poly) sulfides in lithium-sulfur batteries," *2D Materials*, vol. 2, no. 1, p. 014011, 2015.

[4] K. Kumaresan, Y. Mikhaylik, and R. E. White, "A mathematical model for a lithium–sulfur cell," *Journal of the electrochemical society*, vol. 155, no. 8, p. A576, 2008.

[5] M. Marinescu, T. Zhang, and G. J. Offer, "A zero dimensional model of lithium–sulfur batteries during charge and discharge," *Physical Chemistry Chemical Physics*, vol. 18, no. 1, pp. 584–593, 2016.

[6] A. Fotouhi, D. J. Auger, L. O'Neill, T. Cleaver, and S. Walus, "Lithium-sulfur battery technology readiness and applications—a review," *Energies*, vol. 10, no. 12, p. 1937, 2017.

[7] C. Xu, T. Cleary, D. Wang, G. Li, C. Rahn, D. Wang, R. Rajamani, and H. K. Fathy, "Online state estimation for a physics-based lithium-sulfur battery model," *Journal of Power Sources*, vol. 489, p. 229495, 2021.

[8] Z. Huang, L. D. Couto, C. Dangwal, S. Xiao, W. Lv, D. Zhang, and S. J. Moura, "On electrochemical model-based state estimation for lithium-sulfur batteries," *ongoing review*.

[9] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, pp. 197–227, 2016.

[10] G. Ferrari-Trecate, D. Mignone, and M. Morari, "Moving horizon estimation for hybrid systems," *IEEE transactions on automatic control*, vol. 47, no. 10, pp. 1663–1676, 2002.

[11] F. Allgöwer, T. A. Badgwell, J. S. Qin, J. B. Rawlings, and S. J. Wright, "Nonlinear predictive control and moving horizon estimation—an introductory overview," *Advances in control: Highlights of ECC'99*, pp. 391–449, 1999.

[12] M. Torchio, L. Magni, R. D. Braatz, and D. M. Raimondo, "Design of piecewise affine and linear time-varying model predictive control strategies for advanced battery management systems," *Journal of The Electrochemical Society*, vol. 164, no. 4, p. A949, mar 2017. [Online]. Available: https://dx.doi.org/10.1149/2.0201706jes

[13] C. Dangwal, D. Kato, Z. Huang, A. Kandel, and S. Moura, "Global sensitivity analysis of 0-d lithium sulfur electrochemical model," in *IFAC World Congress 2023*, 2023.

[14] L. Breiman, J. Friedman, R. Olshen, and C. Stone, "Cart," *Classification and regression trees*, 1984.

[15] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.