IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 29, 2020, accepted October 9, 2020, date of publication October 26, 2020, date of current version November 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3033497

Fusion-Based Deterministic and Stochastic Parameters Estimation for a Lithium-Polymer Battery Model

ALI QAHTAN TAMEEMI, (Member, IEEE)
American University of Sharjah, Sharjah, UAE
e-mail: alijinto@gmail.com

ABSTRACT In this study, a vehicle localization technique was employed to determine the required quantities in the identification of battery models by considering the behavior of multiple batteries instead of data from a single battery. In previous studies, a plant (e.g., a battery, motor, super-capacitor, or fuel cell) was identified based on a single piece of data. However, such an approach is disadvantageous in that it neglects the effect of process and measurement noise and assumes that the parameters obtained using data from a single plant are identical for all plants of the same type. First, deterministic parameter estimation (DPE), particle swarm optimization (PSO), and teaching-learning-based optimization (TLBO) were initially applied to estimate the battery model parameters using data from a single battery. Second, a fusion-based approach was used to address the process and measurement noise problems through an adaptive unscented Kalman filter algorithm. With this approach, maximum likelihood estimation was employed to fuse multiple-battery data streams to enable the DPE, PSO, and TLBO to recalculate the model parameters based on filtered and fused quantities. A comparison between the experimental results and model outputs obtained using the aforementioned methods for parameter estimation indicated that the proposed multiple-battery approach enhances the accuracy of several identification methods. In contrast, it requires a high computational effort.

INDEX TERMS Universal adaptive stabilizer (UAS), particle swarm optimization (PSO), teaching-learning-based optimization (TLBO), unscented Kalman filter (UKF), maximum likelihood estimation (MLE).

I. INTRODUCTION

Lithium batteries are widely used in various fields. Accordingly, their price is anticipated to decrease over time as the market for this type of batteries further grows [1]. According to [2], lithium batteries are preferred over conventional batteries because they are lightweight, have a high energy density, and can operate within a wide temperature range. However, according to [3], over-discharging these batteries can result in thermal runaway. Therefore, to protect lithium batteries from damage [4], techniques for the accurate estimation of battery capacity are extensively discussed in the literature. In these techniques, battery models are useful tools. In [5], a Luenberger observer was employed to estimate model parameters and capacity in real time, in which the effect of noise was compensated using adaptive forgetting recursive total least squares. In [6], the battery capacity was accurately estimated in real time using an extended Kalman filter (EKF). In addition, the process and measurement noise covariances were identified using the maximum likelihood estimation (MLE). Note that the EKF requires model linearization by computing the Jacobian matrices, which degrades the ability of this technique to determine the optimal estimate [7]. As reported in [8], some studies assumed linearity of the battery models. In [9], the linearization problem was solved using an adaptive unscented Kalman filter (UKF) algorithm. In [10], a particle filter (PF) was applied to estimate the battery capacity. The battery model was utilized without linearization in the UKF and PF algorithms. Battery models have a key function in battery management systems (BMSs) to ensure safe and reliable performance, as reported in [11] and [12]. Unlike BMSs, a battery model is also required in algorithms for charging control [13]. In addition, different battery models perform important functions in power management algorithms to extend the battery lifetime. In the example presented in [14], a model predictive control was employed to enable fast super-capacitor discharge; however, the battery response was retarded.

The associate editor coordinating the review of this manuscript and approving it for publication was Yue Zhang.

Generally, a model is formulated via two main approaches, namely complex and simplistic approaches [15]. In a complex model, nonlinear differential equations are formulated from a physical perspective. In contrast, a simplistic model is expressed as a linear time-invariant representation. In the field of battery modeling, complex representations are preferred over simplistic approaches. The electrochemical models reported in [16] constituted the first type of battery model formed using numerous partial differential equations. The second type includes mathematical models whose main disadvantage is the high computational time required when used with PF algorithms, as reported in [17]. Finally, the third type comprises equivalent circuit models, which are considered simple in terms of model implementation. In [18], the Chen–Mora model is an example of an equivalent circuit representation. The present study aimed to estimate the parameters of this model accurately. In [19] and [20], the Chen–Mora model parameters for a lithium-ion battery pack were estimated using deterministic parameter estimation (DPE) in the form of a universal adaptive stabilizer (UAS) and a constrained optimization technique. The estimated parameters of a single battery were assumed to be identical for all batteries of the same type; however, this is not entirely the case. Moreover, greater parameter accuracy may be achieved when the noise effects and behavior of multiple batteries are considered.

In previous studies, single-battery data were frequently used to estimate the parameters of battery models. An example is described in [21], in which an adaptive observer was employed to estimate the electrochemical model parameters of a lithium-ion battery. Although this method requires large battery packs with a highly varying current profile, its main advantage is that it can reduce the computational time by splitting a complex model into four subsystems. In [22], the authors employed the forgetting factor, recursive least-squares estimation (RLSE), and the time-variant transfer function of the single RC-model to obtain the parameters of a battery. However, the RLSE algorithm used in that study neglected the effects of measurement noise on the results, whereas some forms of RLSE discussed in [7] considered the effects of noise, given that no measurement is perfect. In [23], a genetic algorithm was applied to determine the battery model parameters based on data from a single battery. Similarly, the EKF approach was proposed in [24] and [25] to estimate the battery capacity and model parameters. In this approach, the Jacobian evaluation remains the main problem. In [26], an artificial neural network was employed to model a lithium-polymer battery cell, in which the extremely large training data were the main difficulty. In [27], the battery model parameters were estimated in real time; however, the proposed technique required proper initialization for faster convergence.

Practically, identification tests of similar batteries yield parameter sets that are not identical but are relatively close. Accordingly, the following question arises: Which parameter set should be selected to run the model? In view of this question, a fusion technique is necessary to average the required quantities in the identification process. Signal fusion techniques are extensively applied in localization problems by averaging two or more data streams to delimit a vehicle. An EKF was applied to fuse an inertial measurement unit with a pipeline inspection gauge in [28]. In [29], the data obtained using two global positioning system (GPS) sensors were averaged using MLE with the assumption that the measurement noise covariances were known. In the present study, the MLE fusion technique discussed in [29] was employed to average the estimated terminal voltages according to their associated measurement noise covariances. Similarly, the estimated capacities yielded by the adaptive UKF algorithm were fused using the MLE approach. Based on the merged quantities, the parameters were recalculated. Therefore, such a method, which combines data from multiple batteries, is meaningful for modeling battery units.

The procedure developed in this study can be summarized as follows. First, the parameters of the open-circuit voltage equation were estimated using voltage relaxation tests and curve fitting. Subsequently, DPE, particle swarm optimization (PSO), and teaching-learning-based optimization (TLBO) were used to estimate the remaining model parameters from single-battery data. Second, an adaptive UKF was applied to estimate the battery model output, capacity, and dynamic process and measurement noise covariances for the data of each battery. Note that the initial process and measurement noise covariances were identified using an optimization routine to initiate the adaptive noise estimator equations in the UKF algorithm. Third, the estimated battery terminal voltages were fused according to their measurement noise covariances using the MLE method. In contrast, each capacity state estimated by the adaptive UKF was fused with related quantities from other battery data based on their associated process noise covariances using the MLE technique. Consequently, DPE, PSO, and TLBO were employed again to estimate the model parameters using the filtered and fused quantities instead of the directly measured quantities of a single battery. In a future study, the effects of temperature and aging on battery modeling will be considered. Nevertheless, even if these effects are neglected, battery modeling remains problematic.

To the best of the author's knowledge, no results similar to those presented in this paper have been previously reported. The remainder of this paper is organized as follows. Section II introduces the background of the Chen–Mora battery equivalent circuit model. Section III illustrates the standard deterministic and stochastic parameter estimation techniques. Section IV demonstrates the fusion-based parameter estimation approach. Section V presents the experimental results. Section VI concludes this study.

II. DISCRETE BATTERY MODEL

In this section, the background of the Chen–Mora equivalent circuit battery model [18], shown in Fig. 1, is discussed. The continuous form found in [19] and [20] is discretized using

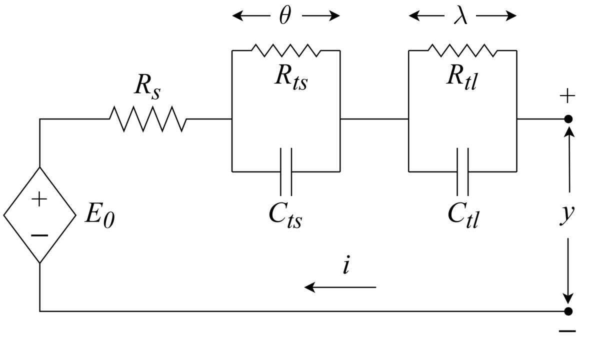

FIGURE 1. Chen–Mora equivalent circuit model.

the Euler method [30] because the UKF algorithm requires a discrete model. The resulting discrete model is expressed by Eqs.(1)–(4).

$$\rho_{k+1} = \rho_k - \frac{\Delta}{C_c} i_k, \quad C_c = 3600 C f_1 f_2 f_3 \tag{1}$$

In Eq.(1), the battery capacity $(\rho) \in [0, 1]$ is obtained using the Coulomb-counting technique described in [31], where the battery current and time step are denoted by i and Δ, respectively. The total battery storage is represented by C_c in Ampere-hours (A-hr). Moreover, the effects of temperature, charging/discharging cycles, and self-discharge are modeled by f_1, f_2, and f_3 in the [0, 1] range, respectively. The aforementioned quantities are assumed to be 1 under normal conditions.

$$\theta_{k+1} = \left(\frac{-\Delta}{R_{ts}(\rho_k) C_{ts}(\rho_k)} + 1\right) \theta_k + \frac{\Delta}{C_{ts}(\rho_k)} i_k \tag{2}$$

$$\lambda_{k+1} = \left(\frac{-\Delta}{R_{tl}(\rho_k) C_{tl}(\rho_k)} + 1\right) \lambda_k + \frac{\Delta}{C_{tl}(\rho_k)} i_k \tag{3}$$

$$y_k = E_0(\rho_k) - \theta_k - \lambda_k - R_s(\rho_k) i_k \tag{4}$$

In Eqs. (2) and (3), the voltages across $R_{ts}||C_{ts}$ and $R_{tl}||C_{tl}$ are represented by θ and λ, respectively. Both states are positive real quantities with the following initial conditions: $\theta_0 = 0$ and $\lambda_0 = 0$. Note that the model elements (i.e., R_{ts}, R_{tl}, C_{ts}, C_{tl}, and R_s) are described by Eqs.(6)–(10). As discussed in [20], each model element is a ρ-dependent quantity, and it has stable or unstable values when the calculated value of ρ is high or low, respectively. The model output (y) is described by Eq.(4), where the battery open-circuit voltage (E_0) and the internal resistance (R_s) are given by Eqs.(5) and (10), respectively. As reported in [32], the model parameters α_1–α_{21} are constants that satisfy $\alpha_{14} < \alpha_{17} < \alpha_{15} < \alpha_{13} < \alpha_{18} < \alpha_{16}$. The stability of this battery model was discussed in [19] and [33], where the stability was determined using two real thresholds, α_1–α_{21}, and ρ. Accordingly, the model is declared in stable, asymptotically stable, or unstable regions.

$$E_0(\rho_k) = -\alpha_1 e^{(-\alpha_2 \rho_k)} + \alpha_3 + \alpha_4 \rho_k - \alpha_5 \rho_k^2 + \alpha_6 \rho_k^3 \tag{5}$$

$$R_{ts}(\rho_k) = \alpha_7 e^{(-\alpha_8 \rho_k)} + \alpha_9 \tag{6}$$

$$R_{tl}(\rho_k) = \alpha_{10} e^{(-\alpha_{11} \rho_k)} + \alpha_{12} \tag{7}$$

$$C_{ts}(\rho_k) = -\alpha_{13} e^{(-\alpha_{14} \rho_k)} + \alpha_{15} \tag{8}$$

$$C_{tl}(\rho_k) = -\alpha_{16} e^{(-\alpha_{17} \rho_k)} + \alpha_{18} \tag{9}$$

$$R_s(\rho_k) = \alpha_{19} e^{(-\alpha_{20} \rho_k)} + \alpha_{21} \tag{10}$$

A battery model can be lumped into the simple representation given by Eq.(11), for which the process function, denoted by $f(\cdot)$, and measurement function, denoted by $h(\cdot)$, are described by the state and measurement equations, respectively. The process noise (w) and measurement noise (v) are assumed to be independent normal Gaussian distributions with zero mean and covariance, Q and R, respectively. Furthermore, x denotes the model state.

$$\left.\begin{aligned} x_{k+1} &= f(x_k, i_k) + w_k, \ w \sim \mathcal{N}(0, Q) \\ y_k &= h(x_k, i_k) + v_k, \quad\ \ v \sim \mathcal{N}(0, R) \end{aligned}\right\} \tag{11}$$

III. PARAMETER ESTIMATION METHODS

In this section, deterministic and stochastic approaches for battery model parameter estimation are briefly explained. A method is considered "deterministic" when it exhibits mathematical convergence analyses that demonstrate its ability to drive the error to zero as time approaches infinity. In contrast, a method is considered "stochastic" when it uses a set of empirical procedures to determine the minimum cost. Note that stochastic methods exhibit no mathematical convergence analyses; alternatively, they are proven via benchmark functions. Note also that the process and measurement noises are neglected within the standard structure of deterministic and stochastic methods, i.e., $w = 0$ and $v = 0$.

A. DETERMINISTIC PARAMETER ESTIMATION

The DPE technique was proposed in [19] and further adopted in [20]. This technique can be summarized as follows. First, open-circuit voltage parameters $(\hat{\alpha}_1$–$\hat{\alpha}_6)$ extracted from the voltage relaxation tests are required. In addition, the actual battery terminal voltage (y) and current (i) should be sampled at each time step until the battery is fully discharged. To determine the estimated battery model states and output, the UAS is used within the DPE structure. The error between the actual and estimated terminal voltages is used in an adaptation law to adjust the model parameters at each time step. Thus, the battery model parameters $(\hat{\alpha}_7$–$\hat{\alpha}_{21})$ are obtained according to the earlier steps. The method is conducted under a significantly low battery current. Unlike battery model identification, DPE was employed to estimate the model parameters of a fuel cell, permanent magnet DC motor, and super-capacitor in [34], [35], and [36], respectively.

$$E_{a,b}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(ka + b)} \tag{12}$$

The Mittag–Leffler (ML) function, in the form of the Nussbaum function in the UAS layout described in [19] and [20], was selected because it affords fast error handling between the actual battery terminal voltage and estimated output. The ML function depends on two positive real constants, a and b. According to [37], the ML function behaves as a Nussbaum function if $a \in (2, 3]$ and $b = 1$. Hence, the ML function is represented by Eq.(12), where $\Gamma(z + 1) = z\Gamma(z), z > 0$ is the standard gamma function. In [38], the Mittag–Leffler function was constructed in the MATLAB/Simulink environment.

$$\hat{R}_{ts}(\rho_k) = \hat{\alpha}_7(u_k)e^{(-\hat{\alpha}_8(u_k)\rho_k)} + \hat{\alpha}_9(u_k) \tag{13}$$
$$\hat{R}_{tl}(\rho_k) = \hat{\alpha}_{10}(u_k)e^{(-\hat{\alpha}_{11}(u_k)\rho_k)} + \hat{\alpha}_{12}(u_k) \tag{14}$$
$$\hat{C}_{ts}(\rho_k) = -\hat{\alpha}_{13}(u_k)e^{(-\hat{\alpha}_{14}(u_k)\rho_k)} + \hat{\alpha}_{15}(u_k) \tag{15}$$
$$\hat{C}_{tl}(\rho_k) = -\hat{\alpha}_{16}(u_k)e^{(-\hat{\alpha}_{17}(u_k)\rho_k)} + \hat{\alpha}_{18}(u_k) \tag{16}$$
$$\hat{R}_{s}(\rho_k) = -\hat{\alpha}_{19}(u_k)e^{(-\hat{\alpha}_{20}(u_k)\rho_k)} + \hat{\alpha}_{21}(u_k) \tag{17}$$

At each time step, equivalent circuit model elements (i.e., R_{ts}, R_{tl}, C_{ts}, C_{tl}, and R_s) are estimated using Eqs.(13)–(17) using the DPE technique. Each α_m parameter is a positive real value tuned via the discrete adaptation law described by Eq.(18), where $m \in [7 \cdots 21]$. The adaptation law produces a bounded solution using the upper and lower limits (i.e., α_{mu} and α_{ml}, respectively) around the steady-state value. In addition, the confidence levels of the lower and upper limits are represented by η_{xm} and η_{ym}, respectively.

$$\begin{aligned}\hat{\alpha}_m(u_{k+1}) = \hat{\alpha}_m(u_k) + (\tilde{y}_k^2 + \eta_{xm}(\alpha_{mu} - \hat{\alpha}_m(u_k))\\ +\eta_{ym}(\alpha_{ml} - \hat{\alpha}_m(u_k)))\Delta\end{aligned} \tag{18}$$

where $\tilde{y}$ denotes the error between the actual and estimated battery terminal voltages at time k, as expressed in Eq.(22). The proposed continuous observer reported in [19] is discretized using the Euler approach and introduced to observe the model states with the aid of an adaptive high-gain observer (u), as described by Eqs.(19) and (20), where $\hat{\theta}$ and $\hat{\lambda}$ are positive real values during the identification process.

$$\hat{\theta}_{k+1} = \left(\frac{-\Delta}{\hat{R}_{ts}(\rho_k)\hat{C}_{ts}(\rho_k)} + 1\right)\hat{\theta}_k + u_k\Delta \tag{19}$$
$$\hat{\lambda}_{k+1} = \left(\frac{-\Delta}{\hat{R}_{tl}(\rho_k)\hat{C}_{tl}(\rho_k)} + 1\right)\hat{\lambda}_k + u_k\Delta \tag{20}$$
$$\hat{y}_k = \hat{E}_0(\rho_k) - \hat{\theta}_k - \hat{\lambda}_k \tag{21}$$

By applying Eqs.(23) and (24), the adaptive high-gain observer receives the integrated value of the squared $\tilde{y}$ in the form of feedback to control the increase in gain. Note that the initial ϕ value is zero. Thus, according to Eq.(25), the output of the adaptive high-gain observer tunes the model states (i.e., $\hat{\theta}$ and $\hat{\lambda}$) to drive $\hat{y}$ to approximate the actual value of y. Accordingly, the model parameters $\hat{\alpha}_7$–$\hat{\alpha}_{21}$ are updated at each time step. The mean value of each parameter array is obtained as shown in Algorithm 1. For further information on the DPE initialization, the reader is referred to [19] and [20].

$$\tilde{y}_k = y_k - \hat{y}_k \tag{22}$$
$$\phi_{k+1} = \tilde{y}_k^2\Delta + \phi_k \tag{23}$$
$$N(\phi_k) = E_{a,b}(b\phi_k^a) \tag{24}$$
$$u_k = -N(\phi_k)\tilde{y}_k \tag{25}$$

Algorithm 1 Deterministic Parameter Estimation [19]

Requirements: Battery terminal voltage (y), capacity (ρ), and open-circuit voltage parameters ($\hat{\alpha}_1$–$\hat{\alpha}_6$).

Initialization: Initial parameters values $\hat{\alpha}_{m_0} > 0$ where $m \in [7 \cdots 21]$, upper bounds (α_{mu}), lower bounds (α_{ml}), and confidence levels (i.e., η_{xm} and η_{ym}).

Output: Estimated battery model parameters ($\hat{\alpha}_7$–$\hat{\alpha}_{21}$)

```
for $k = 1, 2, \cdots, N$ do                                  ▷ $N$: last data point.
    Read $y_k$ and $\rho_k$.                                   ▷ $\rho$ found based on Eq.(1).
    Find $\tilde{y}_k$ using Eq.(22).
    Evaluated each parameter value through Eq.(18).
    Determine the model elements from Eqs.(13)–(17).
    Estimate the states using Eqs.(19) and (20).
    Estimate the model output $\hat{y}_k$ based on Eq.(21).
    if $|\tilde{y}_k| \le \epsilon$ then                       ▷ $\epsilon$: a small positive real number.
        Store each estimated parameter value in its array.
    else
        Continue
    end if
end for
Determine the mean value of each parameter array to obtain the estimated model parameters ($\hat{\alpha}_7$–$\hat{\alpha}_{21}$).
```

B. STOCHASTIC PARAMETER ESTIMATION

As reported in [39], optimization techniques are applied to obtain the best possible solution under certain conditions. The main objective of such techniques is to either minimize the required cost or maximize the benefit. The majority of optimization problems in the literature are minimization problems. Some of the published methods do not require a cost function. This study was restricted to minimization techniques that satisfy a cost function. The cost of a practical problem can be expressed as a function of specific decision variables (i.e., model parameters). An optimization problem begins by defining a cost function structure to enable the optimization technique to tune the decision variables such that the overall cost is driven toward the global minimum value. The problem can be solved using either free-derivative or derivative-based approaches. Well-known meta-heuristic (i.e., free-derivative) methods such as PSO [40] and TLBO [41] were employed. In contrast, the derivative-based approach that assisted the MATLAB *fmincon* optimization routine used in [20] to estimate the model parameters of a single battery was avoided in this study because it requires a high computational effort compared with meta-heuristic methods.

$$X = \begin{pmatrix} \hat{\alpha}_{7,j=1} & \hat{\alpha}_{8,j=1} & \cdots & \hat{\alpha}_{21,j=1} \\ \hat{\alpha}_{7,j=2} & \hat{\alpha}_{8,j=2} & \cdots & \hat{\alpha}_{21,j=2} \\ \vdots & \vdots & \vdots & \vdots \\ \hat{\alpha}_{7,j=N_p} & \hat{\alpha}_{8,j=N_p} & \cdots & \hat{\alpha}_{21,j=N_p} \end{pmatrix} \tag{26}$$

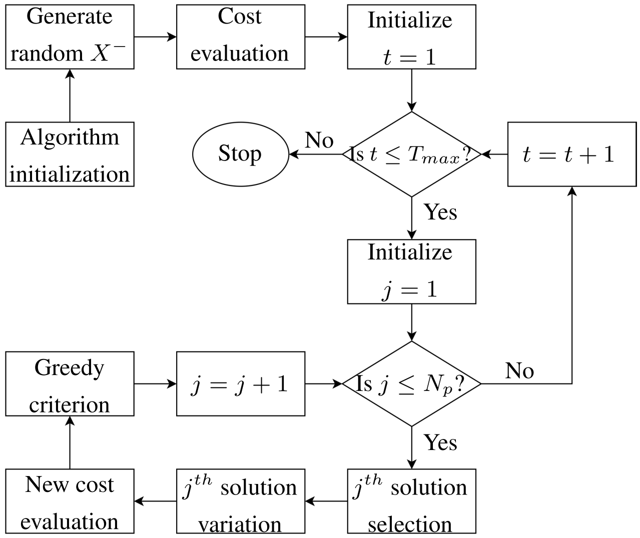

FIGURE 2. Generalized scheme for meta-heuristic techniques.

1) POPULATION AND COST FUNCTION EVALUATIONS

As shown in Fig. 2, predefined constants, namely the total number of iterations (T_{max}) and population size (N_p), are required by the user. The initial population in Eq.(26) is randomly generated between the specified upper and lower bounds for each decision variable. In addition, the costs of the initial population are determined by inputting each j^{th} solution along with the open-circuit voltage parameters and the recorded battery current into the model equations described by Eq.(11) to estimate the output ($\hat{y}_j$). To determine the cost of a particular solution, the mean squared error (MSE) expressed in Eq.(27) is evaluated by inputting the actual terminal voltage and battery model output of a particular solution.

After generating a random population and calculating its costs, a meta-heuristic algorithm selects each j^{th} solution that contains the decision variables, varies their values, computes the new cost of the varied solution, and accepts the varied solution if its cost is less than the previous cost before the variation. Note that the population before and after the update is denoted by X^- and X^+, respectively. As shown in Fig. 2, the process of solution selection, variation, cost evaluation, and greedy criterion are performed T_{max} times. Note also that a survivor strategy in the form of a greedy criterion is used to update a j^{th} solution of the population.

$$\min_{\tilde{y}} \mathbf{f} = \frac{1}{N} \sum_{k=1}^{N} (\tilde{y}_k)^2 \tag{27}$$

2) PARTICLE SWARM OPTIMIZATION

PSO, proposed in [40], is one of the popular free-derivative optimization approaches in the literature. The technique was used in [20] to estimate the Chen–Mora model parameters based on data from a single battery. Therefore, studying the accuracy of PSO in determining the parameters from data of a single battery and data from multiple batteries is a good opportunity. In the PSO algorithm, the population (X) is known as the "position". The change in position values per unit time is known as the velocity (V). In this study, physical terminologies (e.g., position and velocity) were different from their actual meanings to avoid confusion. Given that the PSO algorithm is still in its initialization phase, no historical information has been generated yet. Accordingly, the initial position (X^-) and velocity (V^-) values were generated within the domain of the decision variables, and the initial costs ($\mathbf{f}^-$) of individual solutions were evaluated as discussed in Section III-B1. Note that the size of X and V matrices is $N_p \times D$, where D is the total number of decision variables (i.e., $\hat{\alpha}_7$–$\hat{\alpha}_{21}$). Subsequently, the initial population is assigned as the personal best (p^{best}). In addition, one solution from the population is selected as the global best (g^{best}); this solution has the lowest cost. User predefined constants in the form of inertia (W) and acceleration coefficients (i.e., c_1 and c_2) are required to begin the algorithm.

To update the population (i.e., position), PSO utilizes the velocity expressed by Eq. (28), in which two sets of random vectors (i.e., r_1 and $r_2 \in [0, 1]$) with a size of $1 \times D$ are generated in each iteration in addition to the previous initialization. As a result, the velocity is updated according to Eq.(28). As illustrated in Algorithm 2, for T_{max} iterations, each j^{th}

Algorithm 2 Particle Swarm Optimization [40]

Requirements: Battery terminal voltage and current (i.e., y and i, respectively), open-circuit voltage parameters ($\hat{\alpha}_1$–$\hat{\alpha}_6$), cost function ($\mathbf{f}$), upper and lower bounds (i.e., α_{lb} and α_{ub}), population size (N_p), maximum number of iterations (T_{max}), and defined constants (i.e., W, c_1, and c_2).
Initialization: Initialize the random position (X^-) and velocity (V^-) within the bounds.
Output: Estimated battery model parameters ($\hat{\alpha}_7$–$\hat{\alpha}_{21}$).

```
Evaluate $\mathbf{f}^-$ values using $X^-$.
Assign $X^-$ as personal best ($p^{best}$).
Calculate the costs of $p^{best}$ and assign the solution with
  the lowest cost as the global best ($g^{best}$).
for $t = 1, 2, \cdots, T_{max}$ do
    for $j = 1, 2, \cdots, N_p$ do
        Update the velocity, $V_j^+$, using Eq.(28).
        Determine the new position, $X_j^+$, using Eq.(29).
        Bound $X_j^+$.
        Determine $\mathbf{f}_j^+$ value using Eq.(27).
        if $\mathbf{f}_j^+$ < personal best cost then
            Accept $X_j^+$ as personal best ($p_j^{best}$).
            if personal best cost < global best cost then
                Assign $p_j^{best}$ as $g^{best}$.
            end if
        end if
    end for
end for
Collect the estimated parameters ($\hat{\alpha}_7$–$\hat{\alpha}_{21}$) from $g^{best}$.
```

solution is varied using Eqs.(28) and (29), then bounded within the specified domain. Thereafter, the new cost of the j^{th} varied position solution, denoted by $\mathbf{f}_j^+$, is evaluated using the method discussed in Section III-B1. Following the greedy criterion, X_j^+ replaces X_j^- if $\mathbf{f}_j^+$ is less than $\mathbf{f}_j^-$. In addition, X_j^+ is assigned as personal best (p_j^{best}) and global best (g^{best}) if its cost is less than the stored personal best and global best costs from previous iterations, respectively. Finally, the battery model parameters ($\hat{\alpha}_7$–$\hat{\alpha}_{21}$) estimated via the PSO algorithm are collected from g^{best}.

$$V_j^+ = WV_j^- + c_1 r_1 \left(p_j^{best} - X_j^-\right) + c_2 r_2 \left(g^{best} - X_j^-\right) \quad (28)$$

$$X_j^+ = X_j^- + V_j^+ \quad (29)$$

3) TEACHING-LEARNING-BASED OPTIMIZATION

TLBO was recently introduced in [41] and further justified in [42]. The method mimics a classroom environment in which the process occurs in two phases: the teacher and learner phases. In the teacher phase, students learn from a teacher. In the learner phase, the students interact among themselves and attempt to increase their knowledge. In contrast with PSO, TLBO does not require predefined constants such as inertia and acceleration coefficients, which affect the PSO accuracy, as reported in [43]. However, TLBO requires other common predefined constants such as population size and maximum number of iterations. Accordingly, the algorithm begins by declaring the aforementioned constants and generating a random population within the domain of the decision variables, as discussed in Section III-B1. Thereafter, the cost (i.e., MSE) of each j^{th} generated solution is evaluated using Eq.(27), and the solution corresponding to the lowest cost is selected as the teacher (X^{best}). In the teacher phase, each j^{th} solution in X^- is modified according to Eq.(30), where T_f is a scalar teaching factor (i.e., either 1 or 2) during the update of all X^- solutions, r is a vector with a random number in the range [0, 1], and X^{mean} is a vector containing the mean of each decision variable. Note that the size of both r and X^{mean} is $1 \times D$. Furthermore, the cost function ($\mathbf{f}^+$) is evaluated using the new bounded solution (X_j^+). Subsequently, a greedy selection is performed to update the population solution if the cost of the new solution is less than the previous cost before the variation; otherwise, the population solution remains unchanged.

$$X_j^+ = X_j^- + r\left(X^{best} - T_f X^{mean}\right) \quad (30)$$

In the learner phase, initially, a partner solution (X_p) is randomly selected from X^-, and the solution produced by the teacher phase is further modified using Eq.(31), through which a selection is applied based on the cost of the partner and teacher solutions. Finally, the newly generated solution from the learner phase is bounded between the upper and lower limits, and a greedy search is used to update the population if required, as described in Algorithm 3. To consolidate, a population has N_p random bounded solutions; therefore, the first solution (i.e., $j = 1$) will undergo the teacher phase, and the resulting solution will be used within the learner phase. Similarly, the remaining individual solutions (i.e., $j = 2, \cdots, N_p$) are sequentially performed through the teacher and learner phases. When all the solutions are processed, a single iteration is accomplished using TLBO, and this iteration must be repeated T_{max} times. To date, battery model identification is performed according to data from a single battery. However, in the following section, multiple batteries are considered by fusing the required quantities for identification. The aim is to enhance the accuracy of the estimated parameters.

$$\left.\begin{aligned} X_j^{++} &= X_j^+ + r\left(X_j^+ - X_p\right), \quad if \quad \mathbf{f}^+ < \mathbf{f}_p \\ X_j^{++} &= X_j^+ - r\left(X_j^+ - X_p\right), \quad if \quad \mathbf{f}^+ \geq \mathbf{f}_p \end{aligned}\right\} \quad (31)$$

Algorithm 3 Teaching–Learning-Based Optimization [41]

Requirements: Battery terminal voltage and current (i.e., y and i, respectively), open-circuit voltage parameters ($\hat{\alpha}_1$–$\hat{\alpha}_6$), cost function ($\mathbf{f}$), upper and lower bounds (i.e., α_{lb} and α_{ub}), population size (N_p), and maximum number of iterations (T_{max}).
Initialization: Initialize the random position (X^-) within the bounds.
Output: Estimated battery model parameters ($\hat{\alpha}_7$–$\hat{\alpha}_{21}$).

```
Evaluate $\mathbf{f}^-$ using the $X^-$ solutions.
for $t = 1, 2, \cdots, T_{max}$ do
    for $j = 1, 2, \cdots, N_p$ do
        Choose $X^{best}$.
        Determine $X^{mean}$.
        Calculate $X_j^+$ from Eq.(30).
        Bound $X_j^+$ and evaluate its cost ($\mathbf{f}_j^+$).
        Accept $X_j^+$ if its cost is less than the cost of $X_j^-$.
        Choose any solution randomly ($T_p$).
        Determine $X_j^{++}$ as given in Eq.(31).
        Bound $X_j^{++}$ and evaluate its cost.
        Accept $X_j^{++}$ if its cost is less than $X_j^+$ cost.
    end for
end for
Collect the solution (i.e., $\hat{\alpha}_7$–$\hat{\alpha}_{21}$) from the population
    with the lowest cost.
```

IV. FUSION-BASED PARAMETER ESTIMATION

In this section, the effects of process and measurement noises are considered (i.e., $w \neq 0$ and $v \neq 0$). The uncertainty of a nonlinear system is handled using filtering approaches, such as the EKF, UKF, and PF. According to [7] and [44], the UKF exhibits a better performance than the EKF because it overcomes model linearization around a single point by testing the nonlinear model at several points (called sigma-points), which are inputted into the nonlinear model to obtain the transformed states. Subsequently, the desired states and covariances are utilized. Moreover, the UKF exhibits a better computational performance than PF algorithms; however, its output is expected to be less accurate. Therefore, the UKF was used in this study to estimate the model output and state based on the battery data and parameters obtained by the DPE technique. The dynamic process and measurement noise covariances were determined at each time step using the Sage–Husa approach discussed in [9] and [45].

To combine two or more sources of information, weighting factors are required. These factors allow more precise data to exert more influence on the outcome. Let us consider the problem of averaging two GPS sensors using MLE based on the associated measurement noise discussed in [29]. The goal is to find the most probable position that maximizes the likelihood function given by Eq.(53). Based on this, the estimated terminal voltages and battery capacities from the adaptive UKFs were averaged according to their measurement and process noise covariances, respectively. Thereafter, the averaged quantities were used for DPE, PSO, and TLBO. The mathematical justification of the estimated terminal voltage fusion is presented next; with appropriate changes, the same steps were followed to fuse the capacities.

A. ESTIMATION OF DYNAMIC PROCESS AND MEASUREMENT NOISE COVARIANCES

Inadequate process and measurement noise covariances cause filter divergence and insufficient fused quantities. To resolve this, various methods have been devised, including the Sage–Husa approach. This technique uses adaptive noise estimation equations within the Kalman filter, as described in [45]. It is also employed with the UKF algorithm, as reported in [9]. The ability to address nonlinear systems without linearization is one of the main advantages of the Sage–Husa approach. In contrast, other methods require model linearization when the system is nonlinear, which is similar to the scenarios reported in [46] and [47]. Next, an optimization technique is proposed to initiate the Sage–Husa approach using a reliable estimate of the initial covariances instead of using empirical quantities. Note that $\hat{Q}$ and $\hat{R}$ are the estimated process and measurement noise covariance matrices, respectively.

$$\begin{bmatrix} y_\tau \\ y_{\tau+1} \\ \vdots \end{bmatrix} \approx \begin{bmatrix} v_\tau \\ v_{\tau+1} \\ \vdots \end{bmatrix} + \underbrace{\begin{bmatrix} H_0F_0^{\tau-1} & \cdots & H_0 & 0 & \cdots & 0 \\ 0 & H_0F_0^{\tau-1} & \cdots & H_0 & \cdots & 0 \\ \vdots & \ddots & & & \vdots & \\ 0 & \cdots & 0 & H_0F_0^{\tau-1} & \cdots & H_0 \end{bmatrix}}_{\mathcal{O}} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \end{bmatrix} \tag{32}$$

1) ESTIMATION OF INITIAL PROCESS AND MEASUREMENT NOISE COVARIANCES

In this study, covariance values of process and measurement noises were obtained using a technique that uses the maximum likelihood and discrete disturbance state space of the battery model, as reported in [46]. Initially, the Jacobian values of the transition (F) and measurement (H) matrices from the battery model were determined by taking the partial derivatives of the process function, $(\partial f/\partial x)|_{x=x_0}$, and measurement function, $(\partial h/\partial x)|_{x=x_0}$, with respect to each state and were evaluated at the initial state values, x_0. Given several y measurements and system matrices (F_0 and H_0) that were evaluated at x_0, the unknown Q_0 and R_0 matrices can be obtained as follows. Let N denote the total number of samples, and assume that the battery model is operated in a stable region. Thereafter, select τ such that $\det(F_0^\tau) \leqslant \delta$ for all samples greater than or equal to τ, where δ is a small scalar threshold set to be less than 10^{-6}. This implies that the measurement values from time τ and beyond can be assumed to be independent of past noises. Accordingly, y can be approximated using Eq.(32); for further information, please refer to [46].

$$P = \mathcal{O} \begin{pmatrix} Q_0 & & \\ & \ddots & \\ & & Q_0 \end{pmatrix} \mathcal{O}^T + \begin{pmatrix} R_0 & & \\ & \ddots & \\ & & R_0 \end{pmatrix} \tag{33}$$

Based on Eq.(33), the covariance matrix (P) is obtained, and y is normally distributed with a zero mean. The maximum likelihood is calculated by taking the negative log-likelihood of the multivariate normal distribution vector (y) in Eq.(34), whose solution is subjected to $Q_0, R_0 \geqslant 0$. Note that the constrained MATLAB optimization routine *fmincon* is applied to solve the minimization problem expressed by Eq.(34).

$$\min_{Q_0, R_0} \mathbf{J} = \log(\det(P)) + y^T P^{-1} y \tag{34}$$

B. ADAPTIVE UKF ALGORITHM

Among the various techniques discussed above, this study employed the UKF algorithm because it is not burdened with the problem of linearization, and the necessary computational effort is reasonable. The standard UKF algorithm was extensively discussed in [7], although it utilizes an empirical process and measurement covariances. In [9], an adaptive UKF algorithm was employed to estimate the battery model states. From [7] and [9], this study combined the standard UKF algorithm and dynamic process and measurement noise covariance equations. The adaptive UKF algorithm proceeds as follows:

1) It is initialized by the expected value of the model state and covariance, as expressed in Eq.(35); this algorithm sets $n = 3$ because the model has three states.

$$\begin{aligned} \hat{x}_{k-1}^+ &= E(\hat{x}_0) \\ P_{k-1}^+ &= E\left[(\hat{x}_0 - \hat{x}_0^+)(\hat{x}_0 - \hat{x}_0^+)^T\right] \end{aligned} \tag{35}$$

2) The process and measurement noise covariances are initialized according to the procedure described in Section IV-A1.
3) A finite loop is used to iterate the UKF equations, starting from the initial time $(k - 1)$ to the end

of the data N.

$$\left.\begin{aligned} \hat{x}_{k-1}^{(j)} &= \hat{x}_{k-1}^{+} + \tilde{x}^{(j)}, & j &= 1, 2, \ldots, 2n \\ \tilde{x}^{(j)} &= \left(\sqrt{nP_{k-1}^{+}}\right)_j^T, & j &= 1, 2, \ldots, n \\ \tilde{x}^{(n+j)} &= -\left(\sqrt{nP_{k-1}^{+}}\right)_j^T, & j &= 1, 2, \ldots, n \end{aligned}\right\} \quad (36)$$

4) Using Eq.(36), the model states and covariances are propagated from time $(k-1)$ to k using sigma points, ensuring that $\sqrt{nP}$ does not exceed the matrix square root of nP calculated as $\left(\sqrt{nP}\right)^T\left(\sqrt{nP}\right) = nP$. However, Cholesky factorization is required to calculate the matrix square root. The procedure in [48] is used to evaluate $\sqrt{nP}$ at each time step; $\left(\sqrt{nP}\right)_j$ is the j^{th} row of the $\sqrt{nP}$ matrix. The reader is referred to [7], which provides a calculation example.

$$\hat{x}_k^{(j)} = f(\hat{x}_{k-1}^{(j)}, i_k) \quad (37)$$

5) Using Eq.(37), the obtained sigma points, denoted by $\hat{x}_{k-1}^{(j)}$, are inputted into $f(\cdot)$ equations in addition to the input i at time k to obtain the transformed states.

$$\hat{x}_k^{-} = \frac{1}{2n}\sum_{j=1}^{2n} \hat{x}_k^{(j)} + q_{k-1} \quad (38)$$

6) The mean of each transformed state is determined using Eq.(38).

$$P_k^{-} = \frac{1}{2n}\sum_{j=1}^{2n}\left(\hat{x}_k^{(j)} - \hat{x}_k^{-}\right)\left(\hat{x}_k^{(j)} - \hat{x}_k^{-}\right)^T + Q_{k-1} \quad (39)$$

7) A priori error covariance (P_k^{-}) is obtained using Eq.(39), in which the Q matrix is included to consider the process noise effect. At this stage, the update through the process equations, $f(\cdot)$, is accomplished.

$$\left.\begin{aligned} \hat{x}_{k}^{(j)} &= \hat{x}_{k}^{-} + \tilde{x}^{(j)}, & j &= 1, 2, \ldots, 2n \\ \tilde{x}^{(j)} &= \left(\sqrt{nP_{k}^{-}}\right)_j^T, & j &= 1, 2, \ldots, n \\ \tilde{x}^{(n+j)} &= -\left(\sqrt{nP_{k}^{-}}\right)_j^T, & j &= 1, 2, \ldots, n \end{aligned}\right\} \quad (40)$$

8) The new sigma points, $\hat{x}_k^{(j)}$, are derived using Eq.(40), in which the obtained a priori mean, $\hat{x}^{-}$, and covariance, P^{-}, are inserted.

$$\hat{y}_k^{(j)} = h(\hat{x}_k^{(j)}, i_k) \quad (41)$$

9) The predicted measurements, $\hat{y}_k^{(j)}$, are induced by inputting the new sigma points, $\hat{x}_k^{(j)}$, to the nonlinear measurement function, $h(\cdot)$, given by Eq.(41).

$$\hat{Y}_k = \frac{1}{2n}\sum_{j=1}^{2n} \hat{y}_k^{(j)} + r_k \quad (42)$$

10) Similarly, the mean of $\hat{y}_k^{(j)}$ in Eq.(42) is used to obtain the estimated model output $(\hat{Y})$ at time k.

$$P_y = \frac{1}{2n}\sum_{j=1}^{2n}\left(\hat{y}_k^{(j)} - \hat{Y}_k\right)\left(\hat{y}_k^{(j)} - \hat{Y}_k\right)^T + R_k \quad (43)$$

$$P_{xy} = \frac{1}{2n}\sum_{j=1}^{2n}\left(\hat{x}_k^{(j)} - \hat{x}_k^{-}\right)\left(\hat{y}_k^{(j)} - \hat{Y}_k\right)^T \quad (44)$$

11) The output error covariance (P_y) in Eq.(43) is obtained by considering the measurement noise covariance (R). The cross-error covariance (P_{xy}) in Eq.(44) is also evaluated.

$$K_k = P_{xy}P_y^{-1} \quad (45)$$

$$x_k^{+} = x_k^{-} + K_k(y_k - \hat{Y}_k) \quad (46)$$

$$P_k^{+} = P_k^{-} + K_k P_y K_k^T \quad (47)$$

12) At time k, the Kalman gain (K_k), a posteriori estimated state (x_k^{+}), and a posteriori error covariance (P_k^{+}) are determined using Eqs.(45)–(47), respectively.

$$\hat{q}_k = (1-c_k)\hat{q}_{k-1} + c_k(x_k^{+} - \frac{1}{2n}\sum_{j=1}^{2n}\hat{x}_k^{(j)}) \quad (48)$$

$$\hat{Q}_k = (1-c_k)\hat{Q}_{k-1} + c_k(K_k\tilde{Y}_k\tilde{Y}_k^T K_k^T + P_k^{+}) \quad (49)$$

$$\hat{r}_{k+1} = (1-c_k)\hat{r}_k + c_k(y_k - \frac{1}{2n}\sum_{j=1}^{2n}\hat{y}_k^{(j)}) \quad (50)$$

$$\hat{R}_{k+1} = (1-c_k)\hat{R}_k + c_k\tilde{Y}_k\tilde{Y}_k^T \quad (51)$$

13) In Eqs.(48)–(51), the Sage–Husa estimator is incorporated to update the $\hat{Q}$ and $\hat{R}$ matrices at each time step, where $c_k = (1-d)/(1-d^k)$, and d is the forgetting factor in the range $(0 < d < 1)$. Note that $\tilde{Y}$ is the error between y and $\hat{Y}$.

C. FUSION OF ESTIMATED TERMINAL VOLTAGES

The UKF approach is useful in addressing the effects of noise on the final model parameters. However, the method intrinsically has the disadvantage of obtaining battery model parameters based on the behavior of a single battery. This introduces a problem because batteries of the same type will not have an identical response even if they are operated under the same conditions. In this section, the maximum likelihood method is applied to obtain the average behavior of several batteries according to their associated noises. Thereafter, the model parameters are recalculated according to the fused quantities.

As discussed in [29], a measurement model, represented by Eq.(52), is required to relate the estimated battery terminal voltage $(\mathcal{Y})$ to a certain hidden state $(\mathcal{X})$ that the MLE method aims to estimate, where $\mathcal{Y} = \left(\hat{Y}_{B_1} \cdots \hat{Y}_{B_{tot}}\right)$ and $\mathcal{R} = diag\left(\hat{R}_{B_1} \cdots \hat{R}_{tot}\right)$. Both $\mathcal{Y}$ and $\mathcal{R}$ contain the estimated terminal voltages and measurement noise covariances of the used batteries, and the first and last batteries are denoted by

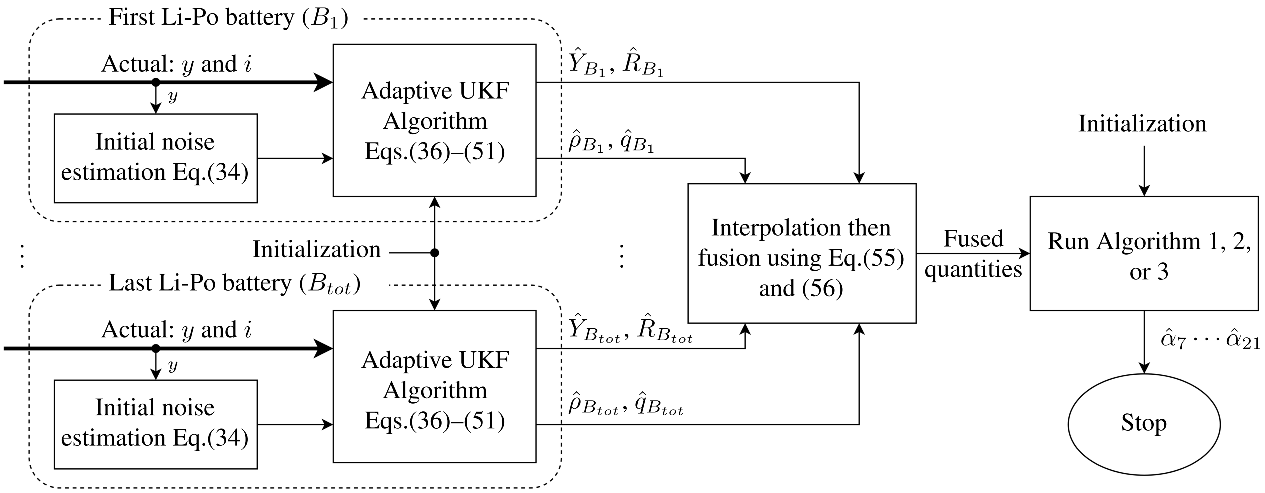
First Li-Po battery (B_1)
Actual: y and i
y
Initial noise estimation Eq.(34)
Adaptive UKF Algorithm Eqs.(36)–(51)
$\hat{Y}_{B_1}$, $\hat{R}_{B_1}$
$\hat{\rho}_{B_1}$, $\hat{q}_{B_1}$
Initialization
Last Li-Po battery (B_{tot})
Actual: y and i
y
Initial noise estimation Eq.(34)
Adaptive UKF Algorithm Eqs.(36)–(51)
$\hat{Y}_{B_{tot}}$, $\hat{R}_{B_{tot}}$
$\hat{\rho}_{B_{tot}}$, $\hat{q}_{B_{tot}}$
Interpolation then fusion using Eq.(55) and (56)
Fused quantities
Initialization
Run Algorithm 1, 2, or 3
$\hat{\alpha}_7 \cdots \hat{\alpha}_{21}$
Stop

FIGURE 3. General flowchart of the multiple-battery data approach.

B_1 and B_{tot}, respectively. Note that the subscript k is dropped for simplicity; however, $\mathcal{Y}$ and $\mathcal{R}$ are updated at each time step. The model given by Eq.(52) represents the relationship between data streams and the hidden state ($\mathcal{X}$), assuming that the measurements are unbiased. Moreover, the measurement noise (v) is assumed to be an independent normal Gaussian distribution with zero mean and covariance $\mathcal{R}$.

$$\left.\begin{array}{l}\mathcal{Y} = C\mathcal{X} + v \\ v \sim \mathcal{N}(0, \mathcal{R})\end{array}\right\} \tag{52}$$

According to [29], the probability distribution of $\mathcal{Y}$ is described by Eq.(53) on the condition that the measurement updates continue. It expresses the probability distribution of $\mathcal{Y}$ for the hidden state ($\mathcal{X}$) that the MLE method aims to determine at a specific value. Accordingly, the value of $\mathcal{X}$ that maximizes the likelihood function is obtained by applying the log-likelihood to Eq.(53) for convenience. The derivative with respect to $\mathcal{X}$ is also obtained from the $log(p(\mathcal{Y}|\mathcal{X}))$ function; subsequently, it is equated to zero. As a result, the value of $\mathcal{X}$ that maximizes the log-likelihood function is determined, and the fused quantity $\mathcal{X}$ is expressed by Eq.(54).

$$p(\mathcal{Y}|\mathcal{X}) = \frac{1}{\sqrt{(2\pi)^{tot}|\mathcal{R}|}} e^{(-\frac{1}{2}(\mathcal{Y}-C\mathcal{X})^T\mathcal{R}^{-1}(\mathcal{Y}-C\mathcal{X}))} \tag{53}$$

$$\mathcal{X} = (C^T\mathcal{R}^{-1}C)^{-1}C^T\mathcal{R}^{-1}\mathcal{Y} \tag{54}$$

For a detailed fusion form, Eq.(55) results from substituting $\mathcal{Y} = \left(\hat{Y}_{B_1} \cdots \hat{Y}_{tot}\right)$, $C = \left(I_{B1} \cdots I_{tot}\right)^T$, $\mathcal{R}^{-1} = diag\left(\hat{R}_{B_1}^{-1} \cdots \hat{R}_{tot}^{-1}\right)$, and $C^T\mathcal{R}^{-1} = \left(\hat{R}_{B_1}^{-1} \cdots \hat{R}_{tot}^{-1}\right)$ into Eq.(54). Each I is a 1×1 identity matrix, and the measurement noise covariance matrix ($\hat{R}$) of each battery also has a 1×1 size.

$$\mathcal{X} = (\hat{R}_{B_1}^{-1} + \cdots + \hat{R}_{tot}^{-1})^{-1}(\hat{R}_{B_1}^{-1}\hat{Y}_{B_1} + \cdots + \hat{R}_{B_{tot}}^{-1}\hat{Y}_{B_{tot}}) \tag{55}$$

Each estimated battery terminal voltage in Eq.(55) is divided by its corresponding measurement noise covariance at each time step. Thereafter, the resulting quantity is normalized by the sum of the measurement noise covariances. In other words, higher noise covariance values allow the estimated battery terminal voltage to contribute less to the final quantity ($\mathcal{X}$) and vice versa. Note that, with appropriate changes, the same procedures are applied to fuse the estimated capacities.

D. FUSION OF ESTIMATED CAPACITIES

Previously, the estimated terminal voltages of several batteries were fused using the maximum likelihood method to obtain the average quantity $\mathcal{X}$. Similarly, the estimated capacities are fused according to their associated noise covariances. To adapt the individual battery capacity state fusion, the estimated voltage fusion formula given by Eq.(55) is further modified and expressed as Eq.(56), where the fused battery capacity is denoted by $\bar{\rho}$. Moreover, the estimated capacity-state noise covariances from the first battery to the last battery are represented by $\hat{q}_{B1} \cdots \hat{q}_{B_{tot}}$, respectively.

$$\bar{\rho} = (\hat{q}_{B1}^{-1} + \cdots + \hat{q}_{B_{tot}}^{-1})^{-1}(\hat{q}_{B1}^{-1}\hat{\rho}_{B1} + \cdots + \hat{q}_{B_{tot}}^{-1}\hat{\rho}_{B_{tot}}) \tag{56}$$

E. PROPOSED APPROACH

1) ESTIMATION OF $\hat{\alpha}_1$–$\hat{\alpha}_6$

Voltage relaxation tests were applied for each battery, and the averaged open-circuit voltage (E_0) curve was determined. The resulting E_0 curves of all batteries were also averaged, that is, $(E_{0_{B_1}} + \cdots + E_{0_{Btot}})/tot$. Parameters $\hat{\alpha}_1$–$\hat{\alpha}_6$, which represented all the batteries, were obtained by shifting the averaged curve to the MATLAB curve fitting toolbox and determining the parameters that represented all batteries.

2) ESTIMATION OF $\hat{\alpha}_7$–$\hat{\alpha}_{21}$

To estimate $\hat{\alpha}_7$–$\hat{\alpha}_{21}$ for multiple batteries, several identical batteries (i.e., $B_1, \cdots, B_{tot}$) were continuously discharged with the same load. The initial noise estimation discussed in Section IV-A1 was applied to the data of each battery. As shown in Fig. 3, the actual terminal voltage, current, and initial noise covariances of all batteries were individu-

Algorithm 4 Proposed Fusion-Based Parameter Estimation Algorithm

Requirements: Actual terminal voltage and current of all batteries (i.e., for each battery, a single continuous discharge test, in addition to several pulse discharge tests.)

Initialization: Define the expected value of model states (x_0^+) and error covariance (P_0^+) according to Eq.(35), n sigma-points (n = number of states), initial parameters values $\hat{\alpha}_{m_0} > 0$ where $m \in [7 \cdots 21]$, upper bounds (α_{mu}), lower bounds (α_{ml}), and confidence levels (i.e., η_{xm} and η_{ym}).

Output: Estimated model parameters ($\hat{\alpha}_7$–$\hat{\alpha}_{21}$) based on data from multiple batteries.

```
for $B = 1, \cdots, tot$ do                    ▷ $tot$: total number of batteries, (i.e., $tot$=9).
    Obtain the open-circuit voltage parameters ($\hat{\alpha}_1$–$\hat{\alpha}_6$) of the $B^{th}$ battery according to Section V-A.
    Run DPE (i.e., Algorithm 1) to estimate $\hat{\alpha}_7$–$\hat{\alpha}_{21}$ of the $B^{th}$ battery.
    According to Section IV-A1, determine the initial process and measurement noise covariances by optimizing Eq.(34).
    for $k = 1, \cdots, N$ do                  ▷ $N$: last data point of the $B^{th}$ battery.
        Use the UKF algorithm (Steps 4–12) to estimate the model output and state using the parameters of a single battery.
        Update the process and measurement covariances based on Step 13.
    end for
    Store the estimated model output ($\hat{Y}_{B^{th}}$), measurement noise covariance ($\hat{R}_{B^{th}}$), battery capacity ($\hat{\rho}_{B^{th}}$), and process noise
covariance of the battery capacity state ($\hat{q}_{B^{th}}$).
end for
$time_{threshold} = \min\big(\text{last time point of the } B_1 \cdots \text{ last time point of the } B_{tot}\big)$
Set loop counter ($k$) to 1.
Initiate the unified interpolation time ($tq$) by zero.          ▷ $tq$: unified interpolation time array.
while $tq_k < (time_{threshold} - 1)$ do       ▷ Create the unified interpolation time array from time 0 to ($time_{threshold} - 1$)s.
    $tq_{k+1} = tq_k + \Delta$                 ▷ $\Delta$: time-step (i.e., $\Delta = 0.1$s).
    $k = k + 1$
end while
Interpolate $\hat{Y}_{B_1 \cdots B_{tot}}$, $\hat{\rho}_{B_1 \cdots B_{tot}}$, $\hat{q}_{B_1 \cdots B_{tot}}$, and $\hat{R}_{B_1 \cdots B_{tot}}$ with respect to $tq$ array.
Find the fused battery terminal voltage ($\mathcal{X}$) given by Eq.(55) based on the interpolated $\hat{Y}_{B_1 \cdots B_{tot}}$ and $\hat{R}_{B_1 \cdots B_{tot}}$.
Determine the fused capacity ($\bar{\rho}$) by inputting the interpolated $\hat{\rho}_{B_1 \cdots B_{tot}}$ and $\hat{q}_{B_1 \cdots B_{tot}}$ to Eq.(56).
Run DPE (Algorithm 1), PSO (Algorithm 2), and TLBO (Algorithm 3) based on the fused terminal voltage ($\mathcal{X}$), fused
capacity ($\bar{\rho}$), and open-circuit voltage parameters ($\hat{\alpha}_1$–$\hat{\alpha}_6$) of multiple batteries to estimate model parameters ($\hat{\alpha}_7$–$\hat{\alpha}_{21}$) for
each identification method.
```

ally inputted into the adaptive UKF algorithm (Section IV-B) to estimate the battery model state and output. Interpolation was used to generate equal data points because the samples of battery data differed slightly. This problem was solved by determining the lowest time space and generating the unified interpolation time (tq) with respect to a smaller time span, as described in Algorithm 4, to obtain an equal number of points for each array. Subsequently, all interpolated $\hat{Y}_{B_1}, \cdots, \hat{Y}_{B_{tot}}$ were fused according to their interpolated measurement noise covariances using the maximum likelihood approach (Section IV-C). Similarly, interpolated capacities were fused according to their associated noises, as expressed in Eq.(56). Subsequently, both fused battery terminal voltage ($\mathcal{X}$) and fused battery capacity ($\bar{\rho}$) were used for the identification algorithms. As described in Algorithms 2 and 3, the measured battery current was required; therefore, the mean value of the measured currents was inputted into the PSO and TLBO as well as $\mathcal{X}$ and $\bar{\rho}$ because the batteries were discharged using the same load.

V. EXPERIMENTAL RESULTS

In this section, the standard identification approach (i.e., DPE, PSO, and TLBO) and the proposed fusion approach are verified and compared with the experimental results. Note that both methods were implemented as offline algorithms by recording the terminal voltage and current of the batteries, and then individually processing the battery data using each algorithm. The sampling time (Δ) was set to 0.1s. The experimental setup is shown in Fig. 11. A power supply, a discharging board, a charger (SKYRC-IMAXB6MIN), nine batteries, and a laptop were used. This experiment was performed on an Intel Core i5-32230M at a 2.6-GHz processing speed. The power supply was employed to calibrate the voltage and current sensors; it also powered a cooling fan. In the discharging board, readings from an Arduino voltage sensor (0–25V) and LEM LA 25-NP current sensor were inputted into to a laptop via an Arduino board (Mega 2560). Thereafter, a MATLAB package [49] was used to read the sensors connected to the Arduino board in real time and store the data in the MATLAB environment. Nine DUPU lithium-polymer batteries

TABLE 1. Estimated parameters ($\hat{\alpha}_7$–$\hat{\alpha}_{21}$) of the data from a single battery and multiple batteries.

$\hat{\alpha}_m$	Data of a single battery			Data of multiple batteries		
	DPE	PSO	TLBO	DPE	PSO	TLBO
$\hat{\alpha}_7$	0.5524	0.7553	0.1714	0.5525	0.4688	0.6126
$\hat{\alpha}_8$	30.232	36.888	16.563	30.238	25.035	30.518
$\hat{\alpha}_9$	0.0553	0.0527	0.0224	0.0553	0.0306	0.1
$\hat{\alpha}_{10}$	5.548	4.5213	3.9926	5.5493	4.3467	1
$\hat{\alpha}_{11}$	150.01	158.49	100	150.01	180.22	101.24
$\hat{\alpha}_{12}$	0.0554	0.0471	0.0371	0.0554	0.0712	0.0445
$\hat{\alpha}_{13}$	756.02	749.78	512.22	756.18	771.63	500
$\hat{\alpha}_{14}$	16.54	14.933	2.4419	16.569	10.826	26.428
$\hat{\alpha}_{15}$	650.68	699.78	517.21	650.7	722.26	647.9
$\hat{\alpha}_{16}$	6048.4	6344.9	5539.6	6049.7	6984.6	6388.3
$\hat{\alpha}_{17}$	27.716	41.832	44.09	27.722	18.056	27.266
$\hat{\alpha}_{18}$	4005.6	4612.8	3065.7	4005.7	4503.9	4560.4
$\hat{\alpha}_{19}$	0.0564	0.0409	0.0976	1.0114	0.6472	0.1276
$\hat{\alpha}_{20}$	0.0564	0.0552	0.0162	1.0114	0.4947	0.7798
$\hat{\alpha}_{21}$	2.0053	1.8396	1.8139	1.0103	1.3736	1.6411

TABLE 2. Estimated open-circuit voltage (E_0) parameters.

Approach	$\hat{\alpha}_1$	$\hat{\alpha}_2$	$\hat{\alpha}_3$	$\hat{\alpha}_4$	$\hat{\alpha}_5$	$\hat{\alpha}_6$
Data of a single battery	1.4324	33.1452	7.2842	0.5486	0.2818	0.8007
Data of multiple batteries	1.5472	38.0636	7.2658	0.7565	0.6721	1.0325

(2S, 7.4V, and 1.3Ah) were discharged using two types of loads: a fixed-value power resistor and a variable resistor. The setup included seven individual fixed-value resistors, each with a 10-Ω load. The variable resistor (0–30Ω) was controlled via the servomotor to obtain a sine-wave load.

A. VOLTAGE RELAXATION TESTS

Voltage relaxation tests were conducted to obtain the E_0 equation parameters, i.e., $\hat{\alpha}_1$–$\hat{\alpha}_6$. The battery was discharged by repeated on/off cycles until the terminal voltage dropped below 6V. Each cycle comprised a 10-min discharging interval followed by a 10-min recovery interval activated by terminating the battery from the load. Each open-circuit voltage observation was sampled between the first and second periods before the battery was reconnected to the load. For parameter estimation of a single battery, four resistive loads (i.e., 40, 50, 60, and 70-Ω) were used to discharge the battery and obtain the E_0 parameters. The open-circuit voltages and their averaged curves are shown in Fig. 4 (a). Thereafter, the averaged curve and Eq.(5) were inputted into the MATLAB curve fitting toolbox to extract the E_0 parameters $\hat{\alpha}_1$–$\hat{\alpha}_6$ for a single battery. Similarly, in Fig. 4 (b), the E_0 parameters of multiple batteries required for the proposed fusion approach were obtained. The E_0 parameters for each scenario are listed in Table 2. The open-circuit voltage parameters were used within the identification methods to estimate the remaining model parameters.

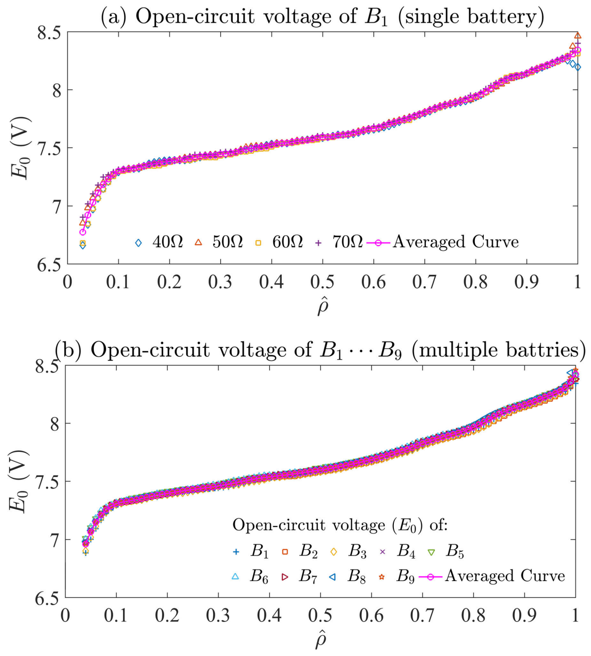

FIGURE 4. (a) Open-circuit voltage of a single battery (i.e., B_1). (b) Open-circuit voltage of multiple batteries (i.e., $B_1 \cdots B_9$).

B. SINGLE DATA APPROACH

The terminal voltage and current recorded from a single battery (i.e., B_1) as a result of 60-Ω continuous load discharge were inputted into the DPE, PSO, and TLBO to identify $\hat{\alpha}_7$–$\hat{\alpha}_{21}$. The first column in Table 1 shows the model parameters $\hat{\alpha}_7$–$\hat{\alpha}_{21}$ estimated using different methods. The DPE, PSO, and TLBO results for a single battery using the 60-Ω data are listed in the second, third, and fourth columns, respectively. In contrast, the DPE, PSO, and TLBO results obtained by fusing the data of the nine batteries ($tot = 9$) are listed in the fifth, sixth, and seventh columns, respectively. The confidence levels for the adaptation law were set to 10. The PSO predefined constants in the form of inertia (W) and acceleration coefficients (i.e., c_1 and c_2) were set to 0.1, 0.5, and 0.5, respectively. For both the PSO and TLOB algorithms, the population size (N_p) and total number of iterations (T_{max}) were set to 100. As shown in Fig. 10, the computational times of DPE, PSO, and TLBO based on the data from a single battery were 2.4537, 33.607, and 65.667 min, respectively. The accuracy of the battery model based on the estimated parameters is shown in Fig. 5 (a), where the output of the model is compared with the actual battery data. Both PSO and TLBO achieved better accuracy than DPE. However, while TLBO achieved the highest accuracy, it required a very long computational time because the cost function was involved twice per iteration.

C. MULTIPLE DATA APPROACH

The fusion approach aimed to enhance the accuracy of the identification techniques using continuous discharging data from the nine batteries (i.e., $B_1, \cdots, B_9$). First, voltage

TABLE 3. Statistics of terminal voltage discharging/charging error of batteries.

Data type	$\hat{\alpha}_1$–$\hat{\alpha}_{21}$	Method	Mean	Median	Mode	Standard Deviation
Discharging	Data of a single battery	DPE	0.0437	0.0479	-0.0293	0.0416
		PSO	0.0254	0.0279	-0.0293	0.0382
		TLBO	0.02472	0.0275	-0.0293	0.0383
	Data of multiple batteries	DPE	-0.0146	-0.009	-0.0603	0.0483
		PSO	0.0083	0.0109	-0.0603	0.0392
		TLBO	-0.0019	0.00043	-0.0603	0.0389
Charging	Data of a single battery	DPE	-0.0905	-0.0655	-0.7158	0.1144
		PSO	-0.0482	-0.028	-0.6667	0.1091
		TLBO	-0.0466	-0.0265	-0.6648	0.109
	Data of multiple batteries	DPE	-0.0024	0.0082	-0.5209	0.0985
		PSO	-0.0575	-0.0394	-0.6158	0.1005
		TLBO	-0.0347	-0.0191	-0.6152	0.1004

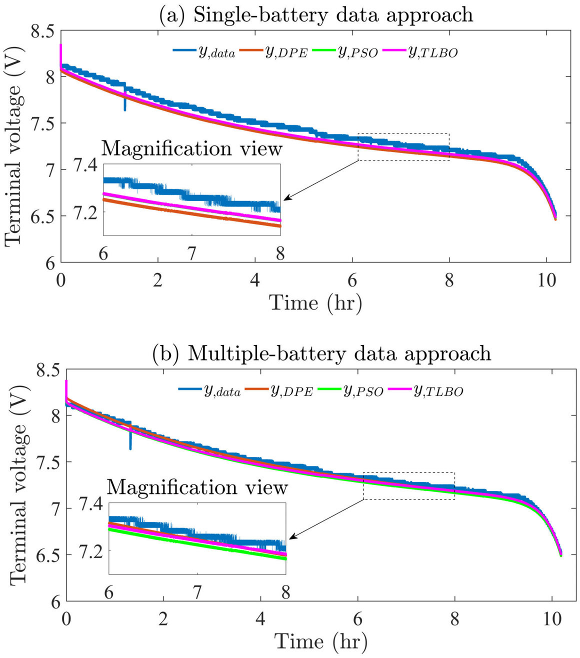

FIGURE 5. (a) Battery model output based on the single-battery data approach. (b) Battery model output based on the multiple-battery data approach.

relaxation tests along with the DPE algorithm were used to estimate each battery parameter ($\hat{\alpha}_1$–$\hat{\alpha}_{21}$). DPE was selected over PSO and TLBO to initiate the adaptive UKF because it has a low computational time. Accordingly, the estimated parameters of individual batteries were used to initiate the noise estimation and adaptive UKF algorithm. For each battery data, the initial process and measurement noise covariances were identified as explained in Section IV-A1, for which the Jacobians of the transition matrix (F) and measurement matrix (H) were determined according to the $f(\cdot)$ and $h(\cdot)$ equations. Thereafter, the $\mathcal{O}$ matrix was constructed according to Eq.(32). Furthermore, the initial state values were substituted as follows: $\rho_0 = 1$, $\theta_0 = 0$, and $\lambda_0 = 0$. The MATLAB optimization routine *fmincon* was utilized to determine the initial process and measurement noise values using the cost function expressed in Eq.(34). Note that the optimization routine was only applied at the beginning, whereas the adaptive noise estimator described in Eqs.(48)–(51) received the initial values and then updated the process and measurement noise covariances at each time step. The parameters $\hat{\alpha}_7$–$\hat{\alpha}_{21}$ from multiple batteries were obtained using the procedure described in Section IV-E2, as listed in Table 1. The computational time of the parameter estimation significantly increased, as shown in Fig. 10. However, the accuracy of the identification methods was significantly improved, as shown in Fig. 5 (b).

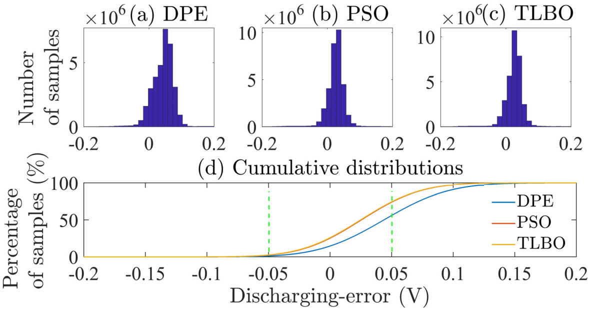

FIGURE 6. Model discharging-error histograms and cumulative distributions for the single-battery data approach. (a), (b), and (c) X- and Y-axes represent the discharging-error (V) and number of samples, respectively.

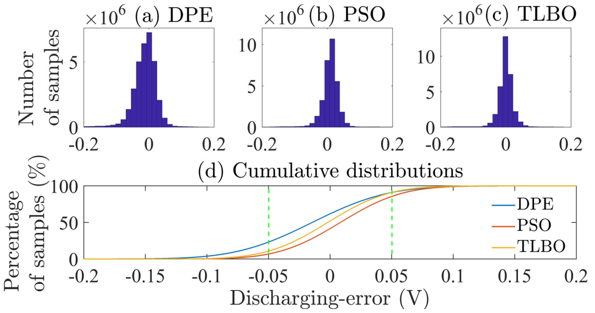

FIGURE 7. Model discharging-error histograms and cumulative distributions for the multiple-battery data approach. (a), (b), and (c) X- and Y-axes represent the discharging-error (V) and number of samples, respectively.

D. COMPARISON

The battery model parameters listed in Tables 1 and 2 were compared by determining the difference between the actual battery terminal voltage and the model output. Accordingly, a discrete Chen–Mora equivalent circuit model, represented by Eqs.(1)–(10), was constructed in the MATLAB environment, and the estimated parameters obtained by each method described above were inputted.

1) DISCHARGING COMPARISON

Adopting the study in [31], a single lithium-polymer battery cell was preferably discharged until it dropped to 3V. Accordingly, the battery discharging process was terminated when the terminal voltage dropped close to 6V because each battery employed in the study had two cells. During battery

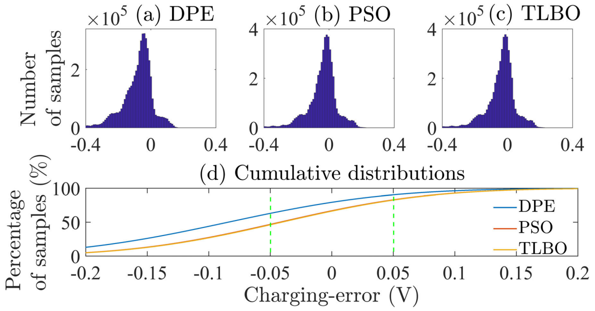

FIGURE 8. Model charging-error histograms and cumulative distributions for the single-battery data approach. (a), (b), and (c) X- and Y-axes represent the charging-error (V) and number of samples, respectively.

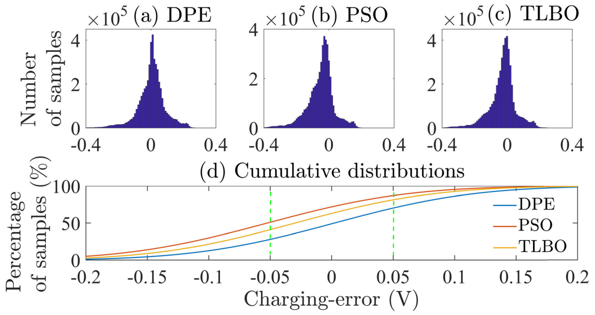

FIGURE 9. Model charging-error histograms and cumulative distributions for the multiple-battery data approach. (a), (b), and (c) X- and Y-axes represent the charging-error (V) and number of samples, respectively.

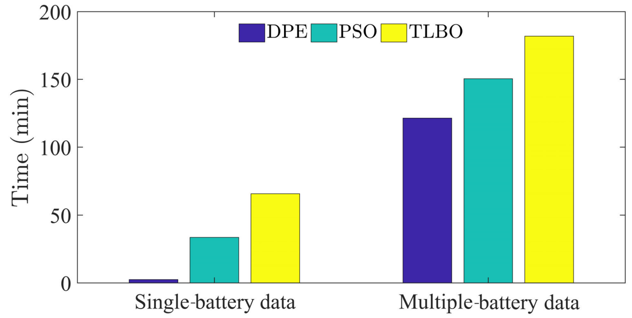

FIGURE 10. Computational time of different identification approaches.

discharging, the terminal voltage and current were sampled. Initially, the nine fully charged lithium-polymer batteries were individually discharged in three ways:

- Continuous discharge, 27 tests: nine lithium-polymer batteries were discharged with constant resistive loads (i.e., 40, 50, and 60-Ω).
- Pulse discharge (square-wave load), 36 tests: nine lithium-polymer batteries were discharged for 10-min (using 40, 50, 60, and 70-Ω constant resistive loads) followed by a 10-min recovery (no-load).
- Sine-wave discharge, 18 tests: the power potentiometer (0–30Ω) controlled by a servomotor was used to generate different sine-wave loads for all batteries.

Evidently, the model performance based on the fusion-based approaches was better than that of the normal approaches, as shown in the magnification views of Fig. 5 (a) and (b). For further analysis of each technique, several error arrays from all discharging tests were combined to form a large error array; the total number of samples for each full array was 33309653. Thereafter, statistical tools such as mean, median, mode, and standard deviation were applied to the full error array of each technique. The results are listed in Table 3. Based on Figs. V-C and 7 (a)–(c), the single-battery data approach exhibited small errors. In contrast, the multiple-battery data approach exhibited almost zero errors. Furthermore, the distribution of the multiple-battery data approach was more compact than that of the single-battery data approach according to the mean values and standard deviations listed in Table 3. According to the cumulative distributions shown in Fig. V-C (d), the model output error based on DPE, PSO, and TLBO using the single-battery data approach yielded 54.782%, 71.565%, and 71.967% of the discharging-error samples in the ±0.05-V range, respectively. In contrast, according to the cumulative distributions of the multiple-battery data approach shown in Fig. 7 (d), the model output errors based on DPE, PSO, and TLBO yielded 67.77%, 78.765%, and 80.13% of the discharging-error samples in the ±0.05-V range, respectively.

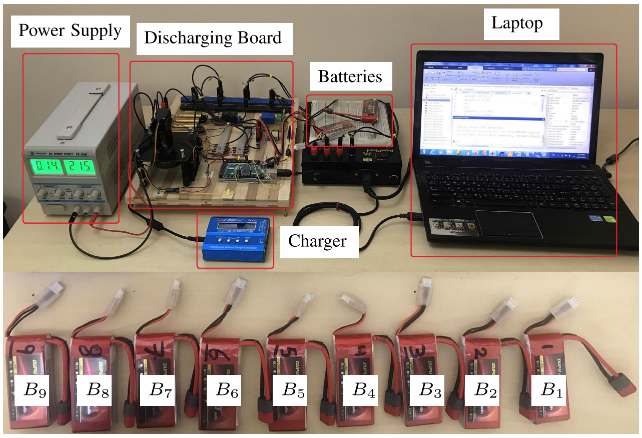

FIGURE 11. Experimental setup.

2) CHARGING COMPARISON

The estimated parameters were examined using charging profiles. Each battery was charged with 0.1, 0.2, and 0.3A using a charger (SKYRC-IMAXB6MIN), as shown in Fig. 11. The battery terminal voltage and current were also recorded, and the error arrays were assembled by determining the difference between the actual battery terminal voltage and the model output. Thus, the full charging-error array for each technique was constructed. The total number of charging-error samples for each method was 5229694. Similarly, the charging-error mean, median, mode, and standard deviation were calculated as listed in Table 3. The error histograms of the single- and multiple-battery data approaches are shown in Figs. 8 and 9, respectively. According to the mean values and standard deviations summarized in Table 3, the multiple-battery data approach was better distributed than the single-battery data approach. According to Fig. 8 (d), the single-battery data cumulative distributions of the DPE, PSO, and TLBO parameters were 27.33%, 36.24%, and 36.49% of the charging-error samples in the ±0.05-V range, respectively. However, Fig. 9 (d) shows the cumulative distributions of the multiple-battery data approach, where the DPE, PSO, and TLBO parameters were 42.34%, 36.09%, and 40.42% of the charging-error samples in the ±0.05-V range, respectively. Note that the charging profiles were significantly affected by the switching behavior of the DC/DC converter within the used charger.

VI. CONCLUSION

In this study, the advantage of estimating the lithium-polymer battery model parameters based on the consideration of data from multiple batteries instead of a single one was established. An adaptive UKF algorithm was individually applied to nine batteries, and the MLE fusion technique was employed to create fused quantities. Instead of inputting the quantities directly measured from a single battery into the identification algorithms, the fused quantities of multiple batteries were inputted into DPE, PSO, and TLBO for parameter estimation. The results indicated that fusing the data of only nine batteries resulted in a higher accuracy compared with ordinary methods. The comparison of the discharging/charging errors shows that the multiple-battery approach accomplished higher accuracy than the single-battery approach. The proposed method is therefore advantageous for battery unit modeling when several batteries are combined to power a load (e.g., an electric vehicle). In addition, an accurate model is useful for generating training data for pattern classification and neural network applications. In contrast, the computational time significantly increased owing to the involvement of adaptive UKF and MLE.

REFERENCES

[1] H. Melin, “The lithium-ion battery end-of-life market’a baseline study,” in *World Economic Forum: Cologny, Switzerland*, vol. 2018, pp. 1–11.

[2] T. B. Reddy, *Linden’s Handbook Batteries*, vol. 4. New York, NY, USA: McGraw-hill, 2011.

[3] L. H. Saw, A. A. O. Tay, and L. W. Zhang, “Thermal management of lithium-ion battery pack with liquid cooling,” in *Proc. 31st Thermal Meas., Model. Manage. Symp. (SEMI-THERM)*, Mar. 2015, pp. 298–302.

[4] V. Pop, H. J. Bergveld, D. Danilov, P. P. Regtien, and P. H. Notten, *Battery Management Systems: Accurate State-of-Charge Indication for Batterypowered Applications*, vol. 9. Springer, 2008.

[5] Z. Wei, J. Zhao, R. Xiong, G. Dong, J. Pou, and K. J. Tseng, “Online estimation of power capacity with noise effect attenuation for lithium-ion battery,” *IEEE Trans. Ind. Electron.*, vol. 66, no. 7, pp. 5724–5735, Jul. 2019.

[6] A. Wadi, M. F. Abdel-Hafez, and A. A. Hussein, “Mitigating the effect of noise uncertainty on the online state-of-charge estimation of li-ion battery cells,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8593–8600, Sep. 2019.

[7] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Hoboken, NJ, USA: Wiley, 2006.

[8] S. Abu-Sharkh and D. Doerffel, “Rapid test and non-linear model characterisation of solid-state lithium-ion batteries,” *J. Power Sources*, vol. 130, nos. 1–2, pp. 266–274, May 2004.

[9] S. Peng, C. Chen, H. Shi, and Z. Yao, “State of charge estimation of battery energy storage systems based on adaptive unscented Kalman filter with a noise statistics estimator,” *IEEE Access*, vol. 5, pp. 13202–13212, 2017.

[10] D. Jiani, W. Youyi, and W. Changyun, “Li-ion battery SOC estimation using particle filter based on an equivalent circuit model,” in *Proc. 10th IEEE Int. Conf. Control Autom. (ICCA)*, Jun. 2013, pp. 580–585.

[11] K. Liu, K. Li, Q. Peng, and C. Zhang, “A brief review on key technologies in the battery management system of electric vehicles,” *Frontiers Mech. Eng.*, vol. 14, no. 1, pp. 47–64, Mar. 2019.

[12] A. Obeid, U. Tariq, and S. Mukhopadhyay, “Supervised learning for early and accurate battery terminal voltage collapse detection,” *IET Circuits, Devices Syst.*, vol. 14, no. 3, pp. 347–356, May 2020.

[13] A. B. Khan, V.-L. Pham, T.-T. Nguyen, and W. Choi, “Multistage constant-current charging method for li-ion batteries,” in *Proc. IEEE Transp. Electrific. Conf. Expo, Asia–Pacific (ITEC Asia–Pacific)*, Jun. 2016, pp. 381–385.

[14] B. Hredzak, V. G. Agelidis, and M. Jang, “A model predictive control system for a hybrid battery-ultracapacitor power source,” *IEEE Trans. Power Electron.*, vol. 29, no. 3, pp. 1469–1479, Mar. 2014.

[15] P. Van Overschee and B. De Moor, *Subspace Identification for Linear Systems: Theory–Implementation–Applications*. Springer, 2012.

[16] S. Tang, Y. Wang, Z. Sahinoglu, T. Wada, S. Hara, and M. Krstic, “State-of-charge estimation for lithium-ion batteries via a coupled thermal-electrochemical model,” in *Proc. Amer. Control Conf. (ACC)*, Jul. 2015, pp. 5871–5877.

[17] D. A. Pola, H. F. Navarrete, M. E. Orchard, R. S. Rabie, M. A. Cerda, B. E. Olivares, J. F. Silva, P. A. Espinoza, and A. Perez, “Particle-filtering-based discharge time prognosis for lithium-ion batteries with a statistical characterization of use profiles,” *IEEE Trans. Rel.*, vol. 64, no. 2, pp. 710–720, Jun. 2015.

[18] M. Chen and G. A. Rincon-Mora, “Accurate electrical battery model capable of predicting runtime and I–V performance,” *IEEE Trans. Energy Convers.*, vol. 21, no. 2, pp. 504–511, Jun. 2006.

[19] D. Ali, S. Mukhopadhyay, H. Rehman, and A. Khurram, “UAS based li-ion battery model parameters estimation,” *Control Eng. Pract.*, vol. 66, pp. 126–145, Sep. 2017.

[20] H. M. Usman, S. Mukhopadhyay, and H. Rehman, “Universal adaptive stabilizer based optimization for li-ion battery model parameters estimation: An experimental study,” *IEEE Access*, vol. 6, pp. 49546–49562, 2018.

[21] D. W. Limoge and A. M. Annaswamy, “An adaptive observer design for real-time parameter estimation in lithium-ion batteries,” *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 2, pp. 505–520, Mar. 2020.

[22] X. Liu, Y. Jin, S. Zeng, X. Chen, Y. Feng, S. Liu, and H. Liu, “Power battery parameter online identification for electric vehicle using a decoupling multiple forgetting factors recursive least squares method,” *CSEE J. Power Energy Syst.*, vol. 6, no. 3, pp. 735–742, Sep. 2020.

[23] M. D. F. B. Binelo, A. T. Z. R. Sausen, P. S. Sausen, and M. O. Binelo, “Mathematical modeling and parameter estimation of battery lifetime using a combined electrical model and a genetic algorithm,” *TEMA (São Carlos)*, vol. 20, no. 1, p. 149, May 2019.

[24] G. L. Plett, “Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2. Modeling and identification,” *J. Power Sources*, vol. 134, no. 2, pp. 262–276, 2004.

[25] G. L. Plett, “Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation,” *J. Power Sources*, vol. 134, no. 2, pp. 277–292, 2004.

[26] S. Alawnah and A. Sagahyroon, “Modeling of smartphones’ power using neural networks,” *Eurasip J. Embedded Syst.*, vol. 2017, no. 1, p. 22, 2017.

[27] C. Wei, M. Benosman, and T. Kim, “Online parameter identification for state of power prediction of lithium-ion batteries in electric vehicles using extremum seeking,” *Int. J. Control, Autom. Syst.*, vol. 17, no. 11, pp. 2906–2916, Nov. 2019.

[28] W. M. F. Al-Masri, M. F. Abdel-Hafez, and M. A. Jaradat, “Inertial navigation system of pipeline inspection gauge,” *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 2, pp. 609–616, Mar. 2020.

[29] R. Shadmehr and S. Mussa-Ivaldi, *Biological Learning And Control: How the Brain Builds Representations, Predicts Events, and Makes Decisions*. Cambridge, MA, USA: MIT Press, 2012.

[30] H. Klee and R. Allen, *Simulation of Dynamic Systems With MATLAB and Simulink*. Boca Raton, FL, USA: CRC Press, 2016.

[31] P. Notten, H. Bergveld, and W. Kruijt, *Battery Management Systems: Design by Modeling*. Norwell, MA, USA: Kluwer, 2002.

[32] S. Mukhopadhyay and F. Zhang, “A high-gain adaptive observer for detecting li-ion battery terminal voltage collapse,” *Automatica*, vol. 50, no. 3, pp. 896–902, Mar. 2014.

[33] F. Zhang, Z. Shi, and S. Mukhopadhyay, "Robustness analysis for battery-supported cyber-physical systems," *ACM Trans. Embedded Comput. Syst.*, vol. 12, no. 3, pp. 1–27, Mar. 2013.

[34] A. Taieb, S. Mukhopadhyay, and A. Al-Othman, "Dynamic model of a proton-exchange membrane fuel cell using equivalent electrical circuit," in *Proc. Adv. Sci. Eng. Technol. Int. Conf. (ASET)*, Mar. 2019, pp. 1–4.

[35] H. M. Usman, S. Mukhopadhyay, and H. Rehman, "Permanent magnet DC motor parameters estimation via universal adaptive stabilization," *Control Eng. Pract.*, vol. 90, pp. 50–62, Sep. 2019.

[36] S. Mukhopadhyay, R. Dhaouadi, M. Takrouri, and R. Dogga, "Supercapacitor characterization using universal adaptive stabilization and optimization," *IEEE Open J. Ind. Electron. Soc.*, vol. 1, pp. 166–183, 2020.

[37] Y. Li and Y. Chen, "When is a mittag–leffler function a nussbaum function?" *Automatica*, vol. 45, no. 8, pp. 1957–1959, Aug. 2009.

[38] S. Mukhopadhyay. *Mittag-Leffler Function, M-File, CMEX DLL, and S-Function. File Exchange: MATLAB Central*. Accessed: Nov. 14, 2019. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/20731-mittag-leffl%er-function–m-file–cmex-dll–and-s-function?requestedDomain=true

[39] S. S. Rao, *Engineering Optimization: Theory and Practice*. Hoboken, NJ, USA: Wiley, 2019.

[40] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE ICNN*, vol. 4, Nov./Dec. 1995, pp. 1942–1948.

[41] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems," *Comput.-Aided Design*, vol. 43, no. 3, pp. 303–315, Mar. 2011.

[42] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: An optimization method for continuous non-linear large scale problems," *Inf. Sci.*, vol. 183, no. 1, pp. 1–15, Jan. 2012.

[43] R. V. Rao and V. Patel, "An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems," *Scientia Iranica*, vol. 20, no. 3, pp. 710–720, Dec. 2012.

[44] J. B. Rawlings and B. R. Bakshi, "Particle filtering and moving horizon estimation," *Comput. Chem. Eng.*, vol. 30, nos. 10–12, pp. 1529–1541, Sep. 2006.

[45] Q. Song and R. Liu, "Weighted adaptive filtering algorithm for carrier tracking of deep space signal," *Chin. J. Aeronaut.*, vol. 28, no. 4, pp. 1236–1244, Aug. 2015.

[46] M. A. Zagrobelny and J. B. Rawlings, "Identifying the uncertainty structure using maximum likelihood estimation," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2015, pp. 422–427.

[47] A. Wadi, W. Al-Masri, W. Siyam, M. F. Abdel-Hafez, and A. H. El-Hag, "Accurate estimation of partial discharge location using maximum likelihood," *IEEE sensors Lett.*, vol. 2, no. 4, pp. 1–4, 2018.

[48] J. D'Errico. *Finding the Nearest Positive Definite Matrix. File Exchange: MATLAB Central*. Accessed: Dec. 27, 2019. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/42885-nearestspd

[49] *MATLAB Support Package for Arduino Hardware. Mathworks*. Accessed: Nov. 11, 2019. [Online]. Available: https://www.mathworks.com/help/supportpkg/arduinoio/index.html

ALI QAHTAN TAMEEMI (Member, IEEE) received the B.Sc. degree in electrical engineering from Ajman University (AU), UAE, in 2014, and the M.Sc. degree in mechatronics engineering from the American University of Sharjah (AUS), UAE, in 2018. His research interests include the control of autonomous mobile robots, artificial neural networks, identification techniques, and detection algorithms.