Journal of Power Sources 196 (2011) 2319–2331

Contents lists available at ScienceDirect

Journal of Power Sources

journal homepage: www.elsevier.com/locate/jpowsour

Recursive approximate weighted total least squares estimation of battery cell total capacity

Gregory L. Plett*,[1]

Department of Electrical and Computer Engineering, University of Colorado at Colorado Springs, 1420 Austin Bluffs Parkway, Colorado Springs, CO 80918, United States

ARTICLE INFO

Article history:
Received 30 June 2010
Received in revised form 9 September 2010
Accepted 20 September 2010
Available online 29 September 2010

Keywords:
Battery cell total capacity estimation
Battery management system algorithms

ABSTRACT

Battery cell total capacity refers to the total amount of charge that can be extracted from a fully charged cell. Knowledge of the present total capacity value is important to being able to calculate the maximum energy storage capability of a battery pack, the remaining energy in a battery pack, and as an indicator of the battery's state of health. We show that traditional methods of estimating battery cell total capacity, which consider noises only in the accumulated ampere hour measurement, are biased. Battery cell total capacity must be estimated with knowledge of both the noises on the state of charge estimates and on the accumulated ampere hour measurements used to compute the total capacity estimate. We demonstrate how total least squares gives better results than traditional methods, and derive an approximate weighted total least squares algorithm that is suitable for implementation in an embedded battery management system.

1. Introduction

The total capacity of a battery cell is a value, usually expressed in ampere hours (Ah) or milliampere hours (mAh), that indicates the maximum electrical charge that the battery cell is capable of holding. New battery cells are manufactured with certain nominal total capacities, but as the cells age, their capacities generally decrease. Therefore, being able to accurately estimate the total capacity of a battery cell is important to being able to determine the health of that battery cell, the maximum energy that can be stored in that cell, and (with the additional knowledge of present state-of-charge), the present energy stored in the cell.

The state-of-charge (SOC) of a battery cell is a value between 0% and 100% that indicates the relative level of charge presently held by the battery cell. A state-of-charge of 100% corresponds to a "full" cell, while a state-of-charge of 0% corresponds to an "empty" cell. State-of-charge is sometimes referred to as "residual capacity" and is not to be confused with the battery cell total capacity. However, the two are related by the equation

$$z(t_2) = z(t_1) + \frac{1}{Q}\int_{t_1}^{t_2} \frac{\eta i(\tau)}{3600}\,d\tau. \tag{1}$$

where $z(t_2)$ is the battery cell SOC at time t_2, $z(t_1)$ is the battery cell SOC at time t_1, Q is the battery cell total capacity in ampere-hours, $i(t)$ is the battery cell current at time t in amperes, η is a unitless efficiency factor, which may take on different values depending on whether the current is positive or negative, and time is measured in seconds. The factor of 3600 converts seconds to hours. SOC itself is unitless. Note that in the convention used herein, discharge current is assumed to have negative sign and charge current is assumed to have positive sign. Note also that in this work we treat total capacity as an electrochemical property of the cell that is independent from both temperature and rate. (We define total capacity precisely in Section 2.)

Eq. (1) is the mathematical basis for most capacity estimation methods. We can rearrange its terms to get:

$$\underbrace{\int_{t_1}^{t_2} \frac{\eta i(\tau)}{3600}\,d\tau}_{y} = Q\underbrace{(z(t_2) - z(t_1))}_{x}, \tag{2}$$

where the obvious linear structure of $y = Qx$ becomes apparent. Using a regression technique, for example, one may compute estimates of Q. One needs only to find values for "x" and "y".

The problem with using standard (least squares) linear regression techniques is that both the integrated current value y and the difference between state-of-charge values x have sensor noise or estimation noise associated with them. The least squares linear regression problem is a solution to the equation $(y - \Delta y) = Qx$; that is, there is noise assumed on the measurements y, but not on the independent variable x. However, Eq. (2) is implicitly of the form

* Tel.: +1 719 255 3468; fax: +1 719 255 3589.
E-mail addresses: glp@eas.uccs.edu, gregory.plett@aevehicles.com.
[1] He is also consultant to American Electric Vehicles, 707 County Line Rd., P.O. Box 509, Palmer Lake, CO 80133, United States. Tel.: +1 719 488 1600x131; fax: +1 719 487 9485.

0378-7753/$ – see front matter
doi:10.1016/j.jpowsour.2010.09.048

$(y - \Delta y) = Q(x - \Delta x)$ since both the integrated current and SOC estimates have noise. That is, because estimates of SOC are generally imperfect, there will be noise on the x variable, and using standard least squares linear regression results in an inaccurate and biased estimate of battery cell total capacity.

The usual approach to counteract this problem is to try to ensure that the SOC estimates are as accurate as possible and then use standard least squares estimation anyway. For example, [1] puts constraints on how the capacity is estimated. It forces the cell current to be zero before the test begins (so that the cell is in an equilibrium state and the first SOC estimate is as accurate as possible) and similarly forces the cell current to be zero after the test ends (again, so that the cell is in an equilibrium state and the second SOC estimate is accurate). This procedure eliminates to a large extent (but not completely) the error in the x variable, and makes the regression reasonably accurate. A second example is [8], where battery current is periodically interrupted and a test performed to estimate the cell resistance so that the ongoing SOC estimates may be corrected for ohmic voltage losses. Again, the basic idea is to try to make x as noise-free as possible.

Both of these methods require a very structured and invasive setting for determining battery cell total capacity. The battery cell current must be controlled by the algorithm. Furthermore, neither correctly handles the residual noise in x: while they minimize the noise, they never totally eliminate it.

In this paper, we propose a method for determining battery cell total capacity in a non-invasive setting where the method does not impose constraints on the battery cell current, and where noise in the SOC estimates is correctly accounted for. That is, the algorithm optimally uses the noisy x and y data to compute a total capacity estimate that is unbiased by the estimation and measurement noises in x and y. This method is based on total least squares methods [2,4–7,15,16], rather than on ordinary least squares methods. Several variations are derived, with the final version able to determine an accurate estimated battery cell total capacity in a computationally efficient manner, without imposing constraints on the battery cell current.

It is not the intent of this paper to propose an SOC estimation algorithm. (For a survey of some available methods, see [9].) Indeed the total capacity estimation techniques proposed herein are agnostic – to a point – to the SOC estimation technique used to give the input SOC estimates that produce x. However, some SOC estimation methods may not be appropriate because they have a strong dependence on the total capacity estimate, leading to a circularity of dependencies. For example, Coulomb counting may not be used because it must have an accurate estimate of total capacity to give an accurate estimate of SOC. Using Coulomb counting with the methods presented herein will give a circular dependence that will not converge and will eventually go unstable. SOC estimation methods based only on voltage are appropriate although, arguably, their accuracy tends to be very poor, especially with cells having a very flat open circuit voltage characteristic (e.g., lithium-ion cells with an iron phosphate cathode chemistry, "LFP" cells). Our own preference is to use Kalman filter based methods, and particularly sigma-point Kalman filters (SPKF), which optimally combine voltage and current information and are very insensitive to errors in the value used for total capacity when making SOC estimates. This allows us to use the static constant Q_{nom} (the nominal capacity of the cell) within the SPKF when computing SOC estimates and still have accurate results. The SOC estimates produced by SPKF using the static value Q_{nom} are then used by the methods proposed herein to produce dynamic estimates $\hat{Q}$ of total capacity Q, which may be used for state-of-health determination. Typical SOC estimation errors using SPKF are on the order of about 1% for lithium-ion cells having manganese oxide cathode chemistries, "LMO" cells [13,14]. In unpublished work, we have also validated SPKF for LFP cells, where we find typical errors on the order of 3–5%, and without requiring reliance on a precise value of Q to do so.

The remainder of this article is organized as follows: Section 2 defines total capacity and some other important quantities. Section 3 introduces the weighted least squares and weighted total least squares methods, as well a method to evaluate the "goodness" of a total capacity estimation methodology overall, and a method to evaluate the dynamic uncertainty of the estimates produced by such a methodology. Section 4 outlines a simplified version of the weighted total least squares method that can be used in some scenarios, and is the motivation for the recursive approximate weighted total least squares method proposed in Section 5. Simulation results to demonstrate the features and limitations of the methods are presented in Section 6, and discussed in Section 7. Finally, Section 8 summarizes the article's main conclusions.

2. Defining total capacity

We now introduce some definitions in order to carefully define "total capacity," with the purpose of differentiating it from other terms having similar names.

Definition: A cell is *fully charged* when its open circuit voltage (OCV) reaches $v_h(T)$, a manufacturer specified voltage that may be a function of temperature T. (For example, typically $v_h(25\,^\circ\mathrm{C}) = 4.2\,\mathrm{V}$ for LMO cells, or $v_h(25\,^\circ\mathrm{C}) = 3.6\,\mathrm{V}$ for LFP cells.) A common method to bring a cell to a fully charged state is to execute a constant-current charge profile until the terminal voltage is equal to $v_h(T)$, followed by a constant-voltage profile until the charging current becomes infinitesimal. We define the state-of-charge (SOC) of a fully charged cell to be 100%.

Definition: A cell is *fully discharged* when its OCV reaches $v_l(T)$, a manufacturer specified voltage that may be a function of temperature T. (For example, typically $v_l(25\,^\circ\mathrm{C}) = 3.0\,\mathrm{V}$ for LMO cells, or $v_l(25\,^\circ\mathrm{C}) = 2.0\,\mathrm{V}$ for LFP cells.) A cell may be fully discharged by executing a constant-current discharge profile until its terminal voltage is equal to $v_l(T)$, followed by a constant-voltage profile until the discharge current becomes infinitesimal. We define the SOC of a fully discharged cell to be 0%.

Definition: The *total capacity*Q of a cell is the quantity of charge removed from a cell as it is brought from a fully charged state to a fully discharged state. While the SI unit for charge is Coulombs, it is more common in practice to use units of ampere hours (Ah) or milliampere hours (mAh) to measure the total capacity of a battery cell. The total capacity of a cell is not a fixed quantity: it generally decays slowly over time as the cell degrades.

Definition: The *discharge capacity*$Q_{[\mathrm{rate}]}$ of a cell is the quantity of charge removed from a cell as it is discharged at a constant rate from a fully charged state until its loaded terminal voltage reaches $v_l(T)$. Because the discharge capacity is determined based on loaded terminal voltage rather than open circuit voltage, it is strongly dependent on the cell's internal resistance, which itself is a function of rate and temperature. Hence, the discharge capacity of a cell is rate dependent and temperature dependent. Because of the resistive $I \times R$ drop, the discharge capacity is less than the total capacity unless the discharge rate is infinitesimal. Likewise, the SOC of the cell is nonzero when the terminal voltage reaches $v_l(T)$ at a non-infinitesimal rate. The discharge capacity of a cell at a particular rate and temperature is not a fixed quantity: it also generally decays slowly over time as the cell degrades.

Definition: The *nominal capacity*Q_{nom} of a cell is a manufacturer-specified quantity that is intended to be representative of the 1C-rate discharge capacity Q_{1C} of a particular manufactured lot of cells at room temperature, $25\,^\circ\mathrm{C}$. The nominal capacity is a constant value. Since the nominal capacity is representative of a lot of cells and the discharge capacity is representative of a single individual cell, $Q_{\text{nom}} \neq Q_{1C}$ in general, even at beginning of life. Also, since Q_{nom} is representative of a discharge capacity and not a total capacity, $Q_{\text{nom}} \neq Q$.

Total capacity is an intrinsic property of a cell's electrode materials, the volume of electrode active materials, and the design electrode state-of-charge ranges. (Note that an electrode's state-of-charge is a distinct quantity from the cell state-of-charge, although the two are related. The electrode state-of-charge is the value of v in Li_vC_6 in a graphite anode, the value of w in $Li_wMn_2O_4$ in an LMO cathode, or the value of w in Li_wFePO_4 in an LFP cathode, for example.) Total capacity is equal to the count of vacant positions in the cathode lattice structure when the cell is fully charged that would be filled with lithium ions when fully discharged. Equivalently, it is equal to the count of vacancies in the anode lattice structure when the cell is fully discharged that would be filled with lithium ions when fully charged. Changing temperature does not change this property, nor does changing the rate at which lithium moves between the anode and cathode.

The purpose of this paper is to propose an optimal method to estimate a cell's total capacity Q. We will assume that $\eta \approx 1$ at all values of current and temperature, which is reasonably accurate for lithium-ion type cells operated according to manufacturer specifications, where self discharge and side reactions causing SEI layer growth and lithium plating (and so forth) can be neglected. If the cell is operated outside of this range, then η must be specifically modeled. It is not the intention of this paper to propose a method to estimate the discharge capacity of a cell at a particular rate, although this can be computed from Q if a cell model is known.

3. Weighted least squares and weighted total least squares

3.1. Derivation of weighted ordinary least squares

Both ordinary least squares (OLS) and total least squares (TLS), as applied to battery cell total capacity estimation, seek to find a constant $\hat{Q}$ such that $y \approx \hat{Q}x$ using N-vectors of measured data $\mathbf{x}$ and $\mathbf{y}$. The i th element x_i in $\mathbf{x}$ and y_i in $\mathbf{y}$ correspond to data collected from a cell over an interval of time, where x_i is the estimated change in state-of-charge over that interval, and y_i is the accumulated ampere hours passing through the cell during that period. Specifically,

$$x_i = z(t_2) - z(t_1) \qquad \text{for time interval } i$$
$$y_i = \int_{t_1}^{t_2} \frac{\eta i(\tau)}{3600}\, d\tau.$$

The vectors $\mathbf{x}$ and $\mathbf{y}$ must be at least one sample long ($N \geq 1$), but multiple samples may be used to obtain better total capacity estimates.

The OLS approach assumes that there is no error on the x_i, and models the data as $\mathbf{y} = Q\mathbf{x} + \boldsymbol{\Delta}\mathbf{y}$, where $\boldsymbol{\Delta}\mathbf{y}$ is a vector of measurement errors, as depicted in Fig. 1(a). (The error bars on the data point are meant to illustrate the uncertainties, which are proportional to σ_{y_i}.) Here, we assume that $\boldsymbol{\Delta}\mathbf{y}$ comprises zero-mean Gaussian random variables, with known variances $\sigma_{y_i}^2$ (which are not necessarily equal to each other). OLS attempts to find an estimate $\hat{Q}$ of the true cell total capacity Q that minimizes the sum of squared errors Δy_i. We generalize that approach here slightly to allow for finding a $\hat{Q}$ that minimizes the sum of *weighted* squared errors, where the weighting takes into account the uncertainty of the measurement. That is, we desire to find a $\hat{Q}$ that minimizes the weighted least squares (WLS) merit function

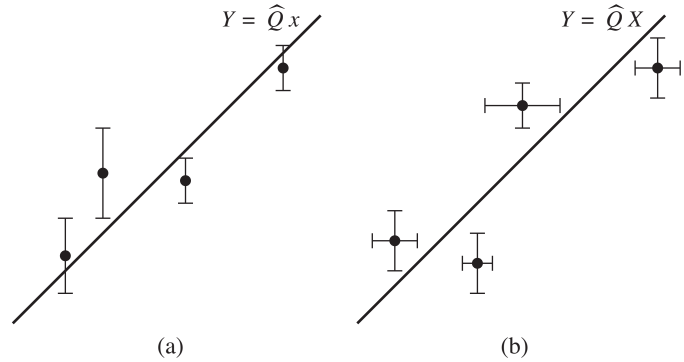

Fig. 1. Fitting a line to data having uncertainties: (a) WLS fitting to data with errors on y_i only and (b) WTLS fitting to data with errors on both x_i and y_i.

$$\chi^2_{\text{WLS}} = \sum_{i=1}^{N} \frac{(y_i - Y_i)^2}{\sigma_{y_i}^2} = \sum_{i=1}^{N} \frac{(y_i - \hat{Q}x_i)^2}{\sigma_{y_i}^2}. \tag{3}$$

In this equation, Y_i is a point on the line $Y_i = \hat{Q}x_i$ corresponding to the measured data pair (x_i, y_i), where y_i is assumed to have noise but x_i has no noise.

There are a number of approaches that may be taken to solve this problem, but one that will serve our purposes well is to differentiate Eq. (3) with respect to $\hat{Q}$ and solve for $\hat{Q}$ by setting the partial derivative to zero.

$$\frac{\partial \chi^2_{\text{WLS}}}{\partial \hat{Q}} = -2\sum_{i=1}^{N} \frac{x_i(y_i - \hat{Q}x_i)}{\sigma_{y_i^2}} = 0$$
$$\hat{Q}\sum_{i=1}^{N} \frac{x_i^2}{\sigma_{y_i}^2} = \sum_{i=1}^{N} \frac{x_i y_i}{\sigma_{y_i}^2}$$
$$\hat{Q} = \frac{\sum_{i=1}^{N} (x_i y_i/\sigma_{y_i}^2)}{\sum_{i=1}^{N} (x_i^2/\sigma_{y_i}^2)}.$$

If we define

$$c_{1,n} = \sum_{i=1}^{n} \frac{x_i^2}{\sigma_{y_i}^2}, \qquad \text{and} \qquad c_{2,n} = \sum_{i=1}^{n} \frac{x_i y_i}{\sigma_{y_i}^2},$$

then we can write $\hat{Q}_n = c_{2,n}/c_{1,n}$. The two quantities $c_{1,n}$ and $c_{2,n}$ may be computed recursively to minimize storage requirements and to even out computational requirements when updating $\hat{Q}_n$ when n gets large

$$c_{1,n} = c_{1,n-1} + \frac{x_n^2}{\sigma_{y_n}^2}$$
$$c_{2,n} = c_{2,n-1} + \frac{x_n y_n}{\sigma_{y_n}^2}.$$

The recursive approach requires an initial estimate of $c_{1,0}$ and $c_{2,0}$. One approach is to simply set $c_{1,0} = c_{2,0} = 0$. Alternately, we can recognize that a cell with nominal capacity Q_{nom} has that capacity over a state-of-charge range of 1.0. Therefore, we can initialize with a synthetic zeroth "measurement" where $x_0 = 1$ and $y_0 = Q_{\text{nom}}$. The value for $\sigma_{y_0}^2$ can be set to the manufacturing variance of the nominal capacity. That is, $c_{1,0} = 1/\sigma_{y_0}^2$ and $c_{2,0} = Q_{\text{nom}}/\sigma_{y_0}^2$.

This method may easily be adapted to allow fading memory of past measurements. We modify the WLS merit function to place more emphasis on recent measurements. We define the fading memory weighted least squares (FMWLS) merit function as

$$\chi^2_{\text{FMWLS}} = \sum_{i=1}^{N} \gamma^{N-i} \frac{(y_i - \hat{Q}x_i)^2}{\sigma^2_{y_i}}, \tag{4}$$

where the forgetting factor γ is in the range $0 \ll \gamma \le 1$. Then, the solution becomes

$$\hat{Q} = \frac{\sum_{i=1}^{N} \gamma^{N-i}(x_i y_i/\sigma^2_{y_i})}{\sum_{i=1}^{N} \gamma^{N-i}(x_i^2/\sigma^2_{y_i})}. \tag{5}$$

This solution may also easily be computed in a recursive manner. We keep track of the two running sums $\tilde{c}_{1,n} = \sum_{i=1}^{n} \gamma^{N-i} x_i^2/\sigma^2_{y_i}$ and $\tilde{c}_{2,n} = \sum_{i=1}^{n} \gamma^{N-i} x_i y_i/\sigma^2_{y_i}$. Then, $\hat{Q}_n = \tilde{c}_{2,n}/\tilde{c}_{1,n}$. When an additional data point becomes available, we update these quantities via

$$\tilde{c}_{1,n} = \gamma \tilde{c}_{1,n-1} + \frac{x_n^2}{\sigma^2_{y_n}}$$
$$\tilde{c}_{2,n} = \gamma \tilde{c}_{2,n-1} + \frac{x_n y_n}{\sigma^2_{y_n}}.$$

In summary, the WLS and FMWLS solutions have a number of nice properties:

1. They give a closed-form solution for $\hat{Q}$. No iteration or advanced algorithms are required–only simple multiplication, addition, and division.
2. The solutions can very easily be computed in a recursive manner.
3. Fading memory can easily be added to allow the estimate $\hat{Q}$ to place greater emphasis on more recent measurements than on earlier measurements, allowing adaptation of $\hat{Q}$ to adjust for true cell total capacity changes.

3.2. Derivation of weighted total least squares

The TLS approach assumes that there is error on both the x_i and y_i measurements, as depicted in Fig. 1(b), and models the data as $(\mathbf{y} - \mathbf{\Delta y}) = Q(\mathbf{x} - \mathbf{\Delta x})$. (The error bars on the data point are meant to illustrate the uncertainties in each dimension, which are proportional to σ_{x_i} and σ_{y_i}.) We assume that $\mathbf{\Delta x}$ comprises zero-mean Gaussian random variables, with known variances $\sigma^2_{x_i}$ and that $\mathbf{\Delta y}$ comprises zero-mean Gaussian random variables, with known variances $\sigma^2_{y_i}$, where $\sigma^2_{x_i}$ is not necessarily equal to or related to $\sigma^2_{y_i}$. TLS attempts to find an estimate $\hat{Q}$ of the true cell total capacity Q that minimizes the sum of squared errors Δx_i plus the sum of squared errors Δy_i. We generalize that approach here slightly to allow for finding a $\hat{Q}$ that minimizes the sum of *weighted* squared errors, where the weighting takes into account the uncertainty of the measurement. That is, we desire to find a $\hat{Q}$ that minimizes the weighted total least squares (WTLS) merit function

$$\chi^2_{\text{WTLS}} = \sum_{i=1}^{N} \frac{(x_i - X_i)^2}{\sigma^2_{x_i}} + \frac{(y_i - Y_i)^2}{\sigma^2_{y_i}}. \tag{6}$$

In this equation, X_i and Y_i are the points on the line $Y_i = \hat{Q}X_i$ corresponding to the noisy measured data pair (x_i, y_i). Since both x_i and y_i have noise, we must handle this optimization problem differently from the way we handled the WLS problem of Eq. (3). We use the approach of reference [6], where Lagrange multipliers λ_i are used to augment the merit function with the constraint that $Y_i = \hat{Q}X_i$. This yields

$$\chi^2_{\text{WTLS},a} = \sum_{i=1}^{N} \frac{(x_i - X_i)^2}{\sigma^2_{x_i}} + \frac{(y_i - Y_i)^2}{\sigma^2_{y_i}} - \lambda_i(Y_i - \hat{Q}X_i).$$

We set the partial derivatives $\partial\chi^2_{\text{WTLS},a}/\partial X_i = \partial\chi^2_{\text{WTLS},a}/\partial Y_i = \partial\chi^2_{\text{WTLS},a}/\partial\lambda_i = 0$. This gives intermediate result

$$X_i = \frac{x_i\sigma^2_{y_i} + \hat{Q}y_i\sigma^2_{x_i}}{\sigma^2_{y_i} + \hat{Q}^2\sigma^2_{x_i}}, \quad \text{and} \quad Y_i = \hat{Q}X_i.$$

With this result, we can re-write Eq. (6) in terms of known quantities as

$$\chi^2_{\text{WTLS}} = \sum_{i=1}^{N} \frac{(y_i - \hat{Q}x_i)^2}{\hat{Q}^2\sigma^2_{x_i} + \sigma^2_{y_i}}. \tag{7}$$

To find the value of $\hat{Q}$ that minimizes this merit function, we set the partial derivative $\partial\chi^2_{\text{WTLS}}/\partial\hat{Q} = 0$. That is,

$$\frac{\partial\chi^2_{\text{WTLS}}}{\partial\hat{Q}} = \sum_{i=1}^{N} \frac{2(\hat{Q}x_i - y_i)(\hat{Q}y_i\sigma^2_{x_i} + x_i\sigma^2_{y_i})}{(\hat{Q}^2\sigma^2_{x_i} + \sigma^2_{y_i})^2} = 0. \tag{8}$$

Unfortunately, this solution has none of the nice properties of the WLS solution. Namely,

1. There is no closed-form solution in the general case; a numerical method must be used instead to find $\hat{Q}$. One possibility is to perform a Newton–Raphson search for $\hat{Q}$ [15], where several iterations of the equation

   $$\hat{Q}_k = \hat{Q}_{k-1} - \frac{\partial\chi^2_{\text{WTLS}}/\partial\hat{Q}}{\partial^2\chi^2_{\text{WTLS}}/\partial\hat{Q}^2}$$

   are performed every time the data vectors **x** and **y** are updated with new data. The numerator of this update equation is the "Jacobian" of the original metric function, and is computed as Eq. (8). The denominator of this update equation is the "Hessian" of the original metric function, which can be found to be

   $$\frac{\partial^2\chi^2_{\text{WTLS}}}{\partial\hat{Q}^2} = 2\sum_{i=1}^{N} \frac{\sigma^4_{y_i}x_i^2 + \sigma^4_{x_i}(3\hat{Q}^2y_i^2 - 2\hat{Q}^3x_iy_i) - \sigma^2_{x_i}\sigma^2_{y_i}(3\hat{Q}^2x_i^2 - 6\hat{Q}x_iy_i + y_i^2)}{(\hat{Q}^2\sigma^2_{x_i} + \sigma^2_{y_i})^3}. \tag{9}$$

   The Newton–Raphson search can be initialized with a WLS estimate of $\hat{Q}$, and has the property that the number of significant figures in the solution doubles with each iteration of the update. In practice, we find that around four iterations produce double-precision results. Note that the metric function χ^2_{WTLS} is convex, so this iterative method is guaranteed to converge to the global solution [7].
2. There is no recursive update in the general case. This has storage implications and computational implications. To use WTLS, the entire vector **x** and **y** must be stored, which implies increasing storage as the number of measurements increase. Furthermore, the number of computations grows as N grows. This is not well suited for an embedded-system application that must run in real time with limited storage capabilities.
3. There is no fading memory recursive update (because there is no recursive update). A *non-recursive* fading memory merit function may be defined, however, as

   $$\chi^2_{\text{FMWTLS}} = \sum_{i=1}^{N} \gamma^{N-i} \frac{(y_i - \hat{Q}x_i)^2}{\hat{Q}^2\sigma^2_{x_i} + \sigma^2_{y_i}}. \tag{10}$$

The Jacobian of this merit function is

$$\frac{\partial \chi^2_{\text{FMWTLS}}}{\partial \hat{Q}} = 2\sum_{i=1}^{N} \gamma^{N-i} \frac{(\hat{Q}x_i - y_i)(\hat{Q}y_i\sigma^2_{x_i} + x_i\sigma^2_{y_i})}{(\hat{Q}^2\sigma^2_{x_i} + \sigma^2_{y_i})^2}. \tag{11}$$

The Hessian is

$$\frac{\partial^2 \chi^2_{\text{FMWTLS}}}{\partial \hat{Q}^2} = 2\sum_{i=1}^{N} \gamma^{N-i} \frac{\sigma^4_{y_i}x_i^2 + \sigma^4_{x_i}(3\hat{Q}^2y_i^2 - 2\hat{Q}^3x_iy_i) - \sigma^2_{x_i}\sigma^2_{y_i}(3\hat{Q}^2x_i^2 - 6\hat{Q}x_iy_i + y_i^2)}{(\hat{Q}^2\sigma^2_{x_i} + \sigma^2_{y_i})^3}. \tag{12}$$

A Newton–Raphson search may be used with this fading-memory cost function to find an estimate of Q.

In Section 4, we will address a special case of WTLS that gives a closed-form solution, with recursive update, and fading memory. In Section 5, we will give an approximate solution to the general WTLS problem that also has these nice properties. Before we do so, we first consider two important properties of both the WLS and WTLS solutions.

3.3. Evaluating the goodness of the model fit

When the measurement errors $\mathbf{\Delta x}$ and $\mathbf{\Delta y}$ are uncorrelated and Gaussian, the metric functions χ^2_{WLS} and χ^2_{WTLS} are chi-squared random variables. χ^2_{WLS} is a chi-squared random variable with $N-1$ degrees of freedom, because N data points y_i were used in its creation and one degree of freedom is lost when fitting $\hat{Q}$. χ^2_{WTLS} is a chi-squared random variable with $2N-1$ degrees of freedom, because N data points x_i and N additional data points y_i are used in its creation, and one degree of freedom is lost when fitting $\hat{Q}$. Knowledge of the distribution and the number of degrees of freedom can be used to determine, from the optimized values of the metric functions, whether the model fit is reliable; that is, whether the linear fit is a good fit to the data, and whether the optimized value of $\hat{Q}$ is a good estimate of the cell total capacity.

The incomplete gamma function $P(\chi^2 \mid \nu)$ is defined as the probability that the observed chi-square for a correct model should be less than a value χ^2 for degree of freedom ν. Its complement, $Q(\chi^2 \mid \nu) = 1 - P(\chi^2 \mid \nu)$, is the probability that the observed chi-square will exceed the value χ^2 by chance *even* for a correct model.[1] Therefore, to test for goodness of fit of a model, we must evaluate

$$Q(\chi^2|\nu) = \frac{1}{\Gamma(\nu/2)} \int_{\chi^2/2}^{\infty} e^{-t} t^{(\nu/2-1)}\, dt.$$

Methods for computing this function are built into many engineering analysis programs, and c-language code may be found in [15]. If the value obtained for $Q(\chi^2 \mid \nu)$ is small, then either the model is wrong and can be statistically rejected, or the variances $\sigma^2_{x_i}$ or $\sigma^2_{y_i}$ are poorly known, or the variances are not actually Gaussian. The third possibility is fairly common, but also generally benign if we are content to accept low values of $Q(\chi^2 \mid \nu)$ as representing a valid model [15]. It is not uncommon to accept models with $Q(\chi^2 \mid \nu) > 0.001$ and to reject them otherwise. We will see that when the hypothesized model is not a good fit to the data, the value of $Q(\chi^2 \mid \nu)$ becomes extremely small. However, when the hypothesized model is equal to the true model generating the data, even when $\hat{Q}$ is not precisely equal to Q, the value of $Q(\chi^2 \mid \nu)$ tends to be very close to unity. We will use this information later to show that the WLS model is not a good approach to total capacity estimation, whereas WTLS is much better.

[1] Note that the nomenclature $Q(\chi^2 \mid \nu)$ is standard for the (complementary) incomplete gamma function, and is not to be confused with the symbol used to denote true cell total capacity Q, or with the symbol used to denote the estimate of cell total capacity $\hat{Q}$.

3.4. Evaluating the confidence limits on the estimated total capacity $\hat{Q}$

When computing an estimate of cell total capacity $\hat{Q}$, it is also important to be able to specify the certainty of that estimate. Specifically, we would like to estimate the variance $\sigma^2_{\hat{Q}}$ of the total capacity estimate, with which we can compute confidence intervals such as three-sigma bounds $(\hat{Q} - 3\sigma_{\hat{Q}}, \hat{Q} + 3\sigma_{\hat{Q}})$ within which the true value of cell total capacity Q lies, with high certainty.

To derive confidence limits, we must re-cast the least-squares type optimization problem as a maximum-likelihood optimization problem. With the assumption that all errors are Gaussian, this is straightforward. If we form a vector $\mathbf{y}$ comprising elements y_i, and a vector $\mathbf{x}$ comprising corresponding elements x_i and a diagonal matrix $\mathbf{\Sigma_y}$ having corresponding diagonal elements $\sigma^2_{y_i}$, then minimizing χ^2_{WLS} is equivalent to maximizing

$$\begin{aligned} ML_{\text{WLS}} &= \frac{1}{(2\pi)^{N/2}|\mathbf{\Sigma_y}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{y} - \hat{Q}\mathbf{x})^T \mathbf{\Sigma_y}^{-1}(\mathbf{y} - \hat{Q}\mathbf{x})\right) \\ &= \frac{1}{(2\pi)^{N/2}|\mathbf{\Sigma_y}|^{1/2}} \exp\left(-\frac{1}{2}\chi^2_{\text{WLS}}\right), \end{aligned}$$

which is a maximum likelihood problem. (The constant to the left of the exponential causes the function to integrate to 1, yielding a valid probability density function.) Similarly, if we form a vector $\mathbf{d}$ concatenating $\mathbf{y}$ and $\mathbf{x}$, and a vector $\hat{\mathbf{d}}$ concatenating the corresponding elements Y_i and X_i, and a diagonal matrix $\mathbf{\Sigma_d}$ having diagonal elements $\sigma^2_{y_i}$ followed by $\sigma^2_{x_i}$, then minimizing χ^2_{WTLS} is equivalent to maximizing

$$\begin{aligned} ML_{\text{WTLS}} &= \frac{1}{(2\pi)^{N}|\mathbf{\Sigma_d}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{d} - \hat{\mathbf{d}})^T \mathbf{\Sigma_d}^{-1}(\mathbf{d} - \hat{\mathbf{d}})\right) \\ &= \frac{1}{(2\pi)^{N}|\mathbf{\Sigma_d}|^{1/2}} \exp\left(-\frac{1}{2}\chi^2_{\text{WTLS}}\right). \end{aligned}$$

The maximum-likelihood formulation makes it possible to determine confidence intervals on $\hat{Q}$. According to the Cramer–Rao theorem, the variance of $\hat{Q}$ (more precisely, the lower bound to the variance) is given by the negative inverse of the second derivative of the argument of the exponential function, evaluated at the $\hat{Q}$ that minimizes the least-squares cost function or maximizes the maximum-likelihood cost function [4]. Therefore,

$$\sigma^2_{\hat{Q}} \geq 2\left(\frac{\partial^2 \chi^2_{\text{WLS}}}{\partial \hat{Q}^2}\right)^{-1} \quad \text{for WLS}$$

$$\sigma^2_{\hat{Q}} \geq 2\left(\frac{\partial^2 \chi^2_{\text{WTLS}}}{\partial \hat{Q}^2}\right)^{-1} \quad \text{for WTLS.}$$

The second partial derivatives of the WTLS and FMWTLS metric functions were already derived in Eqs. (9) and (12). For WLS and FMWLS, we have

$$\frac{\partial^2 \chi^2_{\text{WLS}}}{\partial \hat{Q}^2} = 2\sum_{i=1}^{N} \frac{x_i^2}{\sigma^2_{y_i}} \quad \text{and} \quad \frac{\partial^2 \chi^2_{\text{FMWLS}}}{\partial \hat{Q}^2} = 2\sum_{i=1}^{N} \gamma^{N-i} \frac{x_i^2}{\sigma^2_{y_i}}, \tag{13}$$

which may be computed using the previously defined recursive parameters as

$$\frac{\partial^2 \chi^2_{\text{WLS}}}{\partial \hat{Q}^2} = 2c_{1,n} \quad \text{and} \quad \frac{\partial^2 \chi^2_{\text{FMWLS}}}{\partial \hat{Q}^2} = 2\tilde{c}_{1,n}. \tag{14}$$

4. Simplified method with proportional confidence on x_i and y_i

The general WTLS solution from Section 3.2 provides excellent results but is impractical to implement in an embedded system. Therefore, we search for cases that lead to simpler implementations. Here, we look at an exact solution when the uncertainties on the x_i and y_i data points are proportional to each other for all i, which leads to a simple solution that can easily be implemented in an embedded system. With insights from this solution we will next look at an approximate WTLS solution in Section 5 that also has nice implementation properties.

If $\sigma_{x_i} = k\sigma_{y_i}$, then the WTLS merit function of Eq. (6) reduces to a generalization of the standard TLS merit function

$$\chi^2_{\text{TLS}} = \sum_{i=1}^{N} \frac{(x_i - X_i)^2}{k^2\sigma^2_{y_i}} + \frac{(y_i - Y_i)^2}{\sigma^2_{y_i}} = \sum_{i=1}^{N} \frac{(y_i - \hat{Q}x_i)^2}{(\hat{Q}^2k^2+1)\sigma^2_{y_i}}. \quad (15)$$

Furthermore, the partial derivative of the WTLS merit function of Eq. (8) reduces to

$$\frac{\partial \chi^2_{\text{TLS}}}{\partial \hat{Q}} = 2\sum_{i=1}^{N} \frac{(\hat{Q}x_i - y_i)(\hat{Q}k^2y_i + x_i)}{(\hat{Q}^2k^2+1)^2\sigma^2_{y_i}}. \quad (16)$$

This equation may be solved for an exact solution to $\hat{Q}$, without requiring iteration to do so. We first collect terms

$$\begin{aligned}\frac{\partial \chi^2_{\text{TLS}}}{\partial \hat{Q}} &= 2\sum_{i=1}^{N} \frac{(\hat{Q}x_i - y_i)(\hat{Q}k^2y_i + x_i)}{(\hat{Q}^2k^2+1)^2\sigma^2_{y_i}} = 0 \\ &= \hat{Q}^2\sum_{i=1}^{N} k^2\frac{x_iy_i}{\sigma^2_{y_i}} + \hat{Q}\sum_{i=1}^{N}\frac{x_i^2 - k^2y_i^2}{\sigma^2_{y_i}} + \sum_{i=1}^{N}\frac{-x_iy_i}{\sigma^2_{y_i}} = 0\end{aligned}$$

$$\hat{Q} = \frac{-\left(\sum_{i=1}^{N}(x_i^2 - k^2y_i^2)/(\sigma^2_{y_i})\right) \pm \sqrt{\left(\sum_{i=1}^{N}(x_i^2 - k^2y_i^2)/(\sigma^2_{y_i})\right)^2 + 4k^2\left(\sum_{i=1}^{N}(x_iy_i)/(\sigma^2_{y_i})\right)^2}}{2\sum_{i=1}^{N}k^2(x_iy_i)/(\sigma^2_{y_i})}. \quad (17)$$

We simplify notation slightly by defining $c_{3,n} = \sum_{i=1}^{N} y_i^2/\sigma^2_{y_i}$. Then,

$$\hat{Q}_n = \frac{-(c_{1,n} - k^2c_{3,n}) \pm \sqrt{(c_{1,n} - k^2c_{3,n})^2 + 4k^2c_{2,n}^2}}{2k^2c_{2,n}}.$$

Which of the two roots to choose? We can show that this quadratic equation always has one positive root and one negative root. This can be proven by forming the Routh array, and performing the Routh test on its values [3]. The Routh array is:

$$\begin{array}{c|cc} \hat{Q}^2 & k^2c_{2,n} & -k^2c_{2,n} \\ \hat{Q}^1 & c_{1,n} - k^2c_{3,n} & 0 \\ \hat{Q}^0 & -k^2c_{2,n} & 0 \end{array}$$

The first column of the Routh array always has exactly one sign change, so there is one root of the polynomial in the right-half plane. The other root, therefore, must be in the left-half plane. By the fundamental theorem of algebra, because the coefficients $c_{1,n}, c_{2,n}$, and $c_{3,n}$ are real, the polynomial roots must either both be real or be complex conjugates. The fact that they are in different halves of the complex plane shows that they cannot be complex conjugates, and therefore must both be real. Therefore, we choose the largest root from the solution of the quadratic equation, which corresponds to the positive root. Recursive calculation is done via

$$\hat{Q}_n = \frac{-c_{1,n} + k^2c_{3,n} + \sqrt{(c_{1,n} - k^2c_{3,n})^2 + 4k^2c_{2,n}^2}}{2k^2c_{2,n}}, \quad (18)$$

where initialization is done by setting $x_0 = 1$ and $y_0 = Q_{\text{nom}}$. $\sigma^2_{y_i}$ is set to a representative value of the uncertainty of the total capacity. Therefore, $c_{3,0} = Q^2_{\text{nom}}/\sigma^2_{y_i}$, $c_{2,0} = Q_{\text{nom}}/\sigma^2_{y_i}$ and $c_{1,0} = 1/\sigma^2_{y_i}$, and

$$\begin{aligned} c_{1,n} &= c_{1,n-1} + \frac{x_n^2}{\sigma^2_{y_i}} \\ c_{2,n} &= c_{2,n-1} + \frac{x_ny_n}{\sigma^2_{y_i}} \\ c_{3,n} &= c_{3,n-1} + \frac{y_n^2}{\sigma^2_{y_i}}. \end{aligned}$$

The Hessian, which is required to compute the uncertainty of the estimate, may also be found in terms of the recursive parameters:

$$\frac{\partial^2 \chi^2_{\text{TLS}}}{\partial \hat{Q}^2} = \frac{(-4k^4c_2)\hat{Q}^3 + 6k^4c_3\hat{Q}^2 + (-6c_1 + 12c_2)k^2\hat{Q} + 2(c_1 - k^2c_3)}{(\hat{Q}^2k^2+1)^3}.$$

This can be used to predict error bounds on the estimate $\hat{Q}$. One-sigma bounds are computed as $\sqrt{2/(\partial^2\chi^2_{\text{TLS}}/\partial\hat{Q}^2)}$.

Fading memory may be easily incorporated. Recursive calculation is done via

$$\hat{Q}_n = \frac{-\tilde{c}_{1,n} + k^2\tilde{c}_{3,n} + \sqrt{(\tilde{c}_{1,n} - k^2\tilde{c}_{3,n})^2 + 4k^2\tilde{c}_{2,n}^2}}{2k^2\tilde{c}_{2,n}}, \quad (19)$$

where initialization is done by setting $x_0 = 1$ and $y_0 = Q_{\text{nom}}$. $\sigma^2_{y_i}$ is set to a representative value of the uncertainty of the total capacity. Therefore, $\tilde{c}_{3,0} = Q^2_{\text{nom}}/\sigma^2_{y_i}$, $\tilde{c}_{2,0} = Q_{\text{nom}}/\sigma^2_{y_i}$ and $\tilde{c}_{1,0} = 1/\sigma^2_{y_i}$, and

$$\begin{aligned} \tilde{c}_{1,n} &= \gamma\tilde{c}_{1,n-1} + \frac{x_n^2}{\sigma^2_{y_i}} \\ \tilde{c}_{2,n} &= \gamma\tilde{c}_{2,n-1} + \frac{x_ny_n}{\sigma^2_{y_i}} \\ \tilde{c}_{3,n} &= \gamma\tilde{c}_{3,n-1} + \frac{y_n^2}{\sigma^2_{y_i}}. \end{aligned}$$

After some straightforward manipulations, we can obtain the Hessian in terms of the recursive parameters $\tilde{c}_1$ through $\tilde{c}_3$:

$$\frac{\partial^2 \chi^2_{\text{FMTLS}}}{\partial \hat{Q}^2} = \frac{(-4k^4\tilde{c}_2)\hat{Q}^3 + 6k^4\tilde{c}_3\hat{Q}^2 + (-6\tilde{c}_1 + 12\tilde{c}_2)k^2\hat{Q} + 2(\tilde{c}_1 - k^2\tilde{c}_3)}{(\hat{Q}^2k^2+1)^3}.$$

In summary, this TLS solution shares the nice properties of the WLS solution:

1. It gives a closed-form solution for $\hat{Q}$. No iteration or advanced algorithms are required—only simple multiplication, addition, and division.

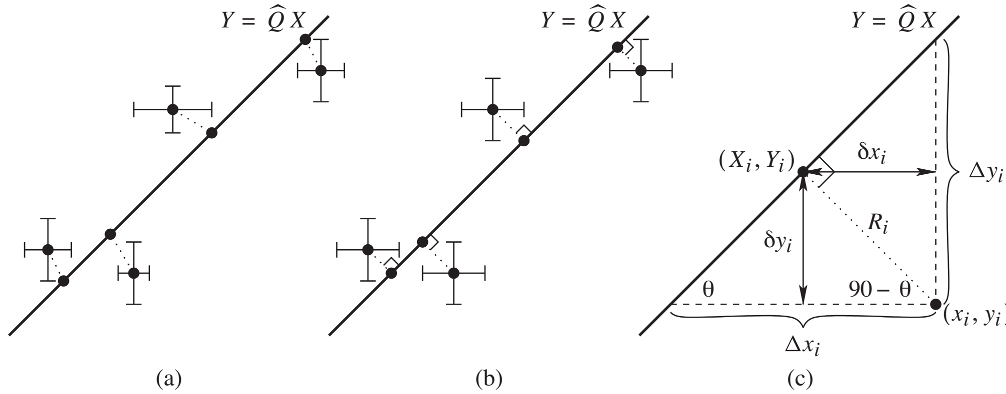

Fig. 2. Geometry of WTLS: (a) mapping between (x_i, y_i) and (X_i, Y_i) for WTLS with unequal confidence on x_i and y_i; (b) mapping between (x_i, y_i) and (X_i, Y_i) for WTLS with equal confidence on x_i and y_i; and (c) definitions for derivation of approximate WTLS.

2. The solution can be very easily computed in a recursive manner. We keep track of the three running sums $c_{1,n}$, $c_{2,n}$ and $c_{3,n}$. When an additional data point becomes available, we update the sums and compute an updated total capacity estimate.
3. Fading memory is easily added.

Unfortunately, this solution does not allow $\sigma_{x_i}^2$ and $\sigma_{y_i}^2$ to be arbitrary—they must be proportionally related by the scaling factor $\sigma_{x_i} = k\sigma_{y_i}$. The next section describes an approximation to TLS that allows an arbitrary relationship.

5. Approximate full solution with recursive formulation

We desire an approximate solution to the WTLS problem that allows $\sigma_{x_i}^2$ and $\sigma_{y_i}^2$ to be non-proportional, but which yields a recursive solution for feasible implementation in an embedded system. Fig. 2 shows the geometry of WTLS and motivates the approximate solution to be developed in this section.

Fig. 2(a) shows the relationship between data point (x_i, y_i) and its optimized map (X_i, Y_i) on the line $Y_i = \hat{Q}X_i$ when $\sigma_{x_i}^2$ and $\sigma_{y_i}^2$ are arbitrary. (The error bars on the data point are meant to illustrate the uncertainties in each dimension, which are proportional to σ_{x_i} and σ_{y_i}.) We see that the distance between x_i and X_i is not necessarily equal to the distance between y_i and Y_i. Indeed, if we know that the quality of the x_i measurement is better (poorer) than the quality of the y_i measurement, the distance to its map X_i on the line should be shorter (greater) than the distance from y_i to its map Y_i on the line.

Fig. 2(b) shows the relationship between data point (x_i, y_i) and its optimized map (X_i, Y_i) on the line $Y_i = \hat{Q}X_i$ when $\sigma_{x_i}^2$ and $\sigma_{y_i}^2$ are equal. In this case, the distance between x_i and X_i is equal to the distance between y_i and Y_i, and the line joining data point (x_i, y_i) and (X_i, Y_i) is perpendicular to the line $Y_i = \hat{Q}X_i$. (If σ_{x_i} and σ_{y_i} are not equal but proportional, the x- or y-axis may be scaled to yield transformed data points with equal variances, and hence the same idea applies.)

Fig. 2(c) illustrates the definitions that will be used to derive an approximate weighted total least squares (AWTLS) solution. As with the TLS solution, we enforce that the line joining data point (x_i, y_i) and (X_i, Y_i) be perpendicular to the line $Y_i = \hat{Q}X_i$. This will result in a solution that may be solved recursively. However, as with the WTLS solution, we weight the distance between x_i and X_i differently from the distance between y_i and Y_i in the optimization merit function. This will give a better total capacity estimate than TLS when the uncertainties on x_i and y_i are not proportional.

We define Δx_i be the x-distance between data point i and the line, and Δy_i be the y-distance between data point i and the line. The slope of the line is $\hat{Q} = \Delta y_i/\Delta x_i$ for all i. The angle of the line is $\theta = \tan^{-1}\hat{Q}$. The shortest distance between the line and a given data point is $R_i = \Delta y_i \cos\theta = \Delta y_i/\sqrt{1+\tan^2\theta} = \Delta y_i/\sqrt{1+\hat{Q}^2}$.

Let $\delta x_i = R_i \sin\theta$ and $\delta y_i = R_i \cos\theta$. These are the x- and y-components of the perpendicular distance between data point i and the fitting line. We then weigh our fitting merit function according to these variances. Therefore, we define the approximate weighted total least squares (AWTLS) merit function as

$$\chi^2_{\mathrm{AWTLS}} = \sum_{i=1}^{N} \frac{\delta x_i^2}{\sigma_{x_i}^2} + \frac{\delta y_i^2}{\sigma_{y_i}^2}.$$

Note that $\sin^2\theta = 1-\cos^2\theta = \hat{Q}^2/(1+\hat{Q}^2)$. Therefore,

$$\delta x_i^2 = \left(\frac{\Delta y_i^2}{1+\hat{Q}^2}\right)\left(\frac{\hat{Q}^2}{1+\hat{Q}^2}\right)$$
$$\delta y_i^2 = \left(\frac{\Delta y_i^2}{1+\hat{Q}^2}\right)\left(\frac{1}{1+\hat{Q}^2}\right).$$

Therefore, since $\Delta y_i = y_i - \hat{Q}x_i$,

$$\chi^2_{\mathrm{AWTLS}} = \sum_{i=1}^{N} \frac{(y_i-\hat{Q}x_i)^2}{(1+\hat{Q}^2)^2}\left(\frac{\hat{Q}^2}{\sigma_{x_i}^2} + \frac{1}{\sigma_{y_i}^2}\right). \tag{20}$$

To verify that AWTLS is an approximation to WTLS in at least some cases, we note that the two merit functions are equal when $\sigma_{x_i} = \sigma_{y_i}$ However, they are not equal when $\sigma_{x_i} = k\sigma_{y_i}$, but this will be corrected at the end of this section.

The Jacobian of the AWTLS merit function is

$$\frac{\partial \chi^2_{\mathrm{AWTLS}}}{\partial \hat{Q}} = \frac{2}{(\hat{Q}^2+1)^3}\sum_{i=1}^{N}\hat{Q}^4\left(\frac{x_iy_i}{\sigma_{x_i}^2}\right) + \hat{Q}^3\left(\frac{2x_i^2}{\sigma_{x_i}^2} - \frac{x_i^2}{\sigma_{y_i}^2} - \frac{y_i^2}{\sigma_{x_i}^2}\right) + \hat{Q}^2\left(\frac{3x_iy_i}{\sigma_{y_i}^2} - \frac{3x_iy_i}{\sigma_{x_i}^2}\right) + \hat{Q}\left(\frac{x_i^2-2y_i^2}{\sigma_{y_i}^2} + \frac{y_i^2}{\sigma_{x_i}^2}\right) + \left(\frac{-x_iy_i}{\sigma_{y_i}^2}\right). \tag{21}$$

This can be re-written in terms of recursively computed running summations

$$\frac{\partial \chi^2_{\mathrm{AWTLS}}}{\partial \hat{Q}} = \frac{2}{(\hat{Q}^2+1)^3}\left(c_5\hat{Q}^4 + (2c_4-c_1-c_6)\hat{Q}^3 + (3c_2-3c_5)\hat{Q}^2 + (c_1-2c_3+c_6)\hat{Q} - c_2\right).$$

where initialization is done by setting $x_0 = 1$ and $y_0 = Q_{\mathrm{nom}}$. $\sigma_{y_0}^2$ is set to a representative value of the uncertainty of the total capacity and $\sigma_{x_0}^2$ is set to a representative value of the uncertainty of a difference between two SOC estimates. Therefore, $c_{1,0} = 1/\sigma_{y_0}^2$, $c_{2,0} = Q_{nom}/\sigma_{y_0}^2$, $c_{3,0} = Q_{nom}^2/\sigma_{y_0}^2$, $c_{4,0} = 1/\sigma_{x_0}^2$, $c_{5,0} = Q_{nom}/\sigma_{x_0}^2$, $c_{6,0} = Q_{nom}^2/\sigma_{x_0}^2$, and

$$\begin{aligned}
c_{1,n} &= c_{1,n-1} + \frac{x_n^2}{\sigma_{y_n}^2}\\
c_{2,n} &= c_{2,n-1} + \frac{x_n y_n}{\sigma_{y_n}^2}\\
c_{3,n} &= c_{3,n-1} + \frac{y_n^2}{\sigma_{y_n}^2}\\
c_{4,n} &= c_{4,n-1} + \frac{x_n^2}{\sigma_{x_n}^2}\\
c_{5,n} &= c_{5,n-1} + \frac{x_n y_n}{\sigma_{x_n}^2}\\
c_{6,n} &= c_{6,n-1} + \frac{y_n^2}{\sigma_{x_n}^2}.
\end{aligned}$$

The roots of this quartic equation

$$c_5\hat{Q}^4 + (2c_4 - c_1 - c_6)\hat{Q}^3 + (3c_2 - 3c_5)\hat{Q}^2 + (c_1 - 2c_3 + c_6)\hat{Q} - c_2 = 0$$

are candidate solutions for $\hat{Q}$. They may be found using the Ferrari method, a closed-form solution that does not require iterative or eigenvalue analysis, so may be implemented readily in an embedded system [17]. However, of the four roots only one is the optimum, and we do not know any method to decide *a priori* which root to solve for. Indeed, in our experience, with some sets of data all roots are real, but with other sets of data some can be complex, and some can be negative. The only foolproof method we know to determine which root is the solution that minimizes χ^2_{AWTLS} is to evaluate the merit function χ^2_{AWTLS} at each of the four candidate solutions, and to retain the one that gives the lowest value. (This can be skipped for negative and complex roots, which are not possible solutions for battery cell capacity.) Computing the merit function may be very readily done if we rewrite it in terms of the running summations

$$\chi^2_{AWTLS} = \frac{1}{(\hat{Q}^2+1)^2}\left(c_4\hat{Q}^4 - 2c_5\hat{Q}^3 + (c_1+c_6)\hat{Q}^2 - 2c_2\hat{Q} + c_3\right).$$

When the assumptions made in deriving AWTLS are approximately true, the Hessian yields a good value for the error bounds on the total capacity estimate. After some straightforward but messy mathematics, we can find the Hessian to be

$$\begin{aligned}
\frac{\partial^2\chi^2_{AWTLS}}{\partial\hat{Q}^2} = \frac{2}{(\hat{Q}^2+1)^4}&(-2c_5\hat{Q}^5 + (3c_1 - 6c_4 + 3c_6)\hat{Q}^4\\
&+(-12c_2 + 16c_5)\hat{Q}^3 + (-8c_1 + 10c_3 + 6c_4 - 8c_6)\hat{Q}^2\\
&+(12c_2 - 6c_5)\hat{Q} + (c_1 - 2c_3 + c_6)).
\end{aligned} \tag{22}$$

Fading memory can be easily incorporated. The cost function is

$$\chi^2_{FMAWTLS} = \sum_{i=1}^{N}\gamma^{N-i}\frac{(y_i - \hat{Q}x_i)^2}{(1+\hat{Q}^2)^2}\left(\frac{\hat{Q}^2}{\sigma_{x_i}^2} + \frac{1}{\sigma_{y_i}^2}\right). \tag{23}$$

The Jacobian is

$$\frac{\partial\chi^2_{AWTLS}}{\partial\hat{Q}} = \frac{2}{(\hat{Q}^2+1)^3}\sum_{i=1}^{N}\gamma^{N-i}\left[\hat{Q}^4\left(\frac{x_iy_i}{\sigma_{x_i}^2}\right) + \hat{Q}^3\left(\frac{2x_i^2}{\sigma_{x_i}^2} - \frac{x_i^2}{\sigma_{y_i}^2} - \frac{y_i^2}{\sigma_{x_i}^2}\right) + \hat{Q}^2\left(\frac{3x_iy_i}{\sigma_{y_i}^2} - \frac{3x_iy_i}{\sigma_{x_i}^2}\right) + \hat{Q}\left(\frac{x_i^2 - 2y_i^2}{\sigma_{y_i}^2} + \frac{y_i^2}{\sigma_{x_i}^2}\right) + \left(\frac{-x_iy_i}{\sigma_{y_i}^2}\right)\right]. \tag{24}$$

This can be re-written in terms of recursively computed running summations

$$\begin{aligned}
\frac{\partial\chi^2_{FMAWTLS}}{\partial\hat{Q}} = \frac{2}{(\hat{Q}^2+1)^3}&\left(\tilde{c}_5\hat{Q}^4 + (-\tilde{c}_1 + 2\tilde{c}_4 - \tilde{c}_6)\hat{Q}^3\right.\\
&\left.+(3\tilde{c}_2 - 3\tilde{c}_5)\hat{Q}^2 + (\tilde{c}_1 - 2\tilde{c}_3 + \tilde{c}_6)\hat{Q} - \tilde{c}_2\right).
\end{aligned}$$

where initialization is done by setting $x_0 = 1$ and $y_0 = Q_{nom}$. $\sigma_{y_0}^2$ is set to a representative value of the uncertainty of the total capacity and $\sigma_{x_0}^2$ is set to a representative value of the uncertainty of a difference between two SOC estimates. Therefore, $\tilde{c}_{1,0} = 1/\sigma_{y_0}^2$, $\tilde{c}_{2,0} = Q_{nom}/\sigma_{y_0}^2$, $\tilde{c}_{3,0} = Q_{nom}^2/\sigma_{y_0}^2$, $\tilde{c}_{4,0} = 1/\sigma_{x_0}^2$, $\tilde{c}_{5,0} = Q_{nom}/\sigma_{x_0}^2$, $\tilde{c}_{6,0} = Q_{nom}^2/\sigma_{x_0}^2$, and

$$\begin{aligned}
\tilde{c}_{1,n} &= \frac{\gamma\tilde{c}_{1,n-1} + x_n^2}{\sigma_{y_n}^2}\\
\tilde{c}_{2,n} &= \frac{\gamma\tilde{c}_{2,n-1} + x_ny_n}{\sigma_{y_n}^2}\\
\tilde{c}_{3,n} &= \frac{\gamma\tilde{c}_{3,n-1} + y_n^2}{\sigma_{y_n}^2}\\
\tilde{c}_{4,n} &= \frac{\gamma\tilde{c}_{4,n-1} + x_n^2}{\sigma_{x_n}^2}\\
\tilde{c}_{5,n} &= \frac{\gamma\tilde{c}_{5,n-1} + x_ny_n}{\sigma_{x_n}^2}\\
\tilde{c}_{6,n} &= \frac{\gamma\tilde{c}_{6,n-1} + y_n^2}{\sigma_{x_n}^2}.
\end{aligned}$$

Again, this quartic equation may be solved using the Ferrari method, for example. The four candidate solutions must be checked against the merit function to determine which one is optimal. The merit function in terms of these variables is

$$\chi^2_{FMAWTLS} = \frac{1}{(\hat{Q}^2+1)^2}\left(\tilde{c}_4\hat{Q}^4 - 2\tilde{c}_5\hat{Q}^3 + (\tilde{c}_1+\tilde{c}_6)\hat{Q}^2 - 2\tilde{c}_2\hat{Q} + \tilde{c}_3\right).$$

The Hessian is

$$\begin{aligned}
\frac{\partial^2\chi^2_{FMAWTLS}}{\partial\hat{Q}^2} = \frac{2}{(\hat{Q}^2+1)^4}&(-2\tilde{c}_5\hat{Q}^5 + (3\tilde{c}_1 - 6\tilde{c}_4 + 3\tilde{c}_6)\hat{Q}^4\\
&+(-12\tilde{c}_2 + 16\tilde{c}_5)\hat{Q}^3 + (-8\tilde{c}_1 + 10\tilde{c}_3 + 6\tilde{c}_4 - 8\tilde{c}_6)\hat{Q}^2\\
&+(12\tilde{c}_2 - 6\tilde{c}_5)\hat{Q} + (\tilde{c}_1 - 2\tilde{c}_3 + \tilde{c}_6)).
\end{aligned} \tag{25}$$

Note that the AWTLS merit function in Eq. (20) does not equal the WTLS merit function in Eq. (7) when $\sigma_{y_i} = k\sigma_{x_i}$. This can be easily remedied. Define $\tilde{y}_i = ky_i$. Then $\sigma_{\tilde{y}_i} = \sigma_{x_i}$. Invoke the AWTLS or FMAWTLS methods to find total capacity estimate $\hat{Q}$ and Hessian H using input sequences comprised of the original $\mathbf{x}$ vector and the scaled $\tilde{\mathbf{y}}$ vector (i.e., $(x_i, \tilde{y}_i)$ with corresponding variances $(\sigma_{x_i}^2, k^2\sigma_{y_i}^2)$). The true slope estimate can be found as $\hat{Q}_{corrected} = \hat{Q}/k$ and the corrected Hessian can be found as $H_{corrected} = H/k^2$. This is the method used in the results section, where the proportionality constant is estimated as $k = \sigma_{x_1}/\sigma_{y_1}$. This scaling improves results even when σ_{y_i} and σ_{x_i} are not proportionally related, if k is chosen to give an "order of magnitude" proportionality or average proportionality between the uncertainties of x_i and y_i.

In summary, these AWTLS solutions share the nice properties of the WLS solution:

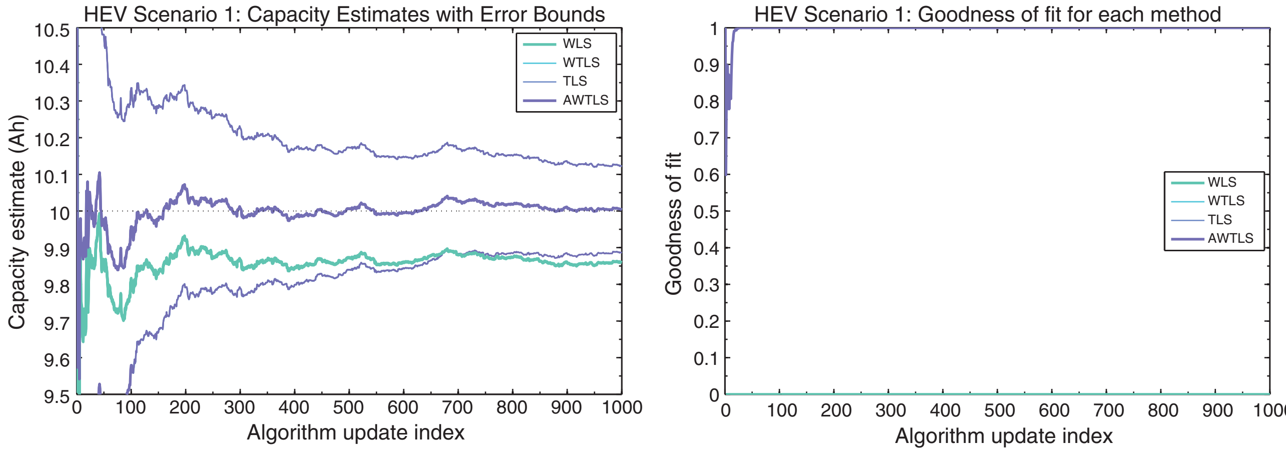

Fig. 3. Results from hybrid electric vehicle application simulation scenario 1.

1. They give a closed-form solution for $\hat{Q}$. No iteration or advanced algorithms are required—only simple multiplication, addition, and division.
2. The solution can be very easily computed in a recursive manner. We keep track of the six running sums $c_{1,n}$ through $c_{6,n}$. When an additional data point becomes available, we update the sums and compute an updated total capacity estimate.
3. Fading memory can easily be added to allow the estimate $\hat{Q}$ to place greater emphasis on more recent measurements than on earlier measurements, allowing adaptation of $\hat{Q}$ to adjust for true cell total capacity changes.
4. Furthermore, this method is superior to the TLS solution since it allows individual weighting on the x_i and y_i data points.

6. Simulations to test the methods

This section presents a number of usage scenarios to exercise the four different total capacity estimation methods and compare their performance. All scenarios use the fading-memory version of the four methods, but we omit the prefix "FM" for brevity. Unless otherwise mentioned, the fading memory forgetting factor $\gamma = 1.0$.

We assume that the individual SOC estimates that are input to these methods can be determined to an accuracy of $\sigma_z = 0.01$. This is being very generous, since the best method we are aware of, SPKF, achieves only around $\sigma_z = 0.01$ for LMO cells and $\sigma_z = 0.03$ for LFP cells in practice, when Q_{nom} is used instead of Q in the estimator. Other methods we have used, such as extended Kalman filtering or EKF [10–12], achieve around $\sigma_z = 0.02$ or higher for LMO cells in practice. (A nice advantage of both EKF and SPKF is that they give dynamic estimates of σ_z that ensure that the values of σ_{x_i} used in total capacity estimation are accurate.) However, despite limitations in present SOC estimation, we are confident that with better cell modeling, these methods can be improved in the future.

We use computer simulation rather than cell testing to validate the algorithms because it allows us to constrain a variety of factors that would be difficult to control in a real-time embedded system. These include: the efficacy and accuracy of the SOC estimation algorithms used to provide input to the total capacity estimation algorithms; the accuracy and precision of the raw sensor measurements used as input (including the challenges of bias errors, nonlinear errors, and random errors, for example); the repeatability of the experiment; and the fact that total capacity of a physical cell fades over time and the associated difficulty/impossibility of knowing the "true" value of total capacity with which to compare our results. Therefore, we choose to use synthetic data to isolate the performance of the total capacity estimation algorithms themselves, when all other factors are in some sense idealized. The x_i and y_i values are mathematically generated, as described in the individual subsections below. We are currently collecting long-term life test data from both LMO and LFP cells, which we intend to use in future publications to evaluate the effectiveness of these and other algorithms over the lifetime of a cell.

6.1. Hybrid electric vehicle application, scenario 1

The first sets of simulations that we present are for hybrid electric vehicle scenarios. From the perspective of total capacity estimation, these applications are characterized by the narrow window of SOC used. We assume that the vehicle uses a SOC range of 40–60%. Therefore, each time the total capacity estimate is updated, the true change in SOC can range from −0.2 to +0.2. We simulate this by choosing the true value of x_i to be a uniform random number selected between these limits.

The HEV application is also characterized by the fact that the battery pack is never fully charged to a precisely known SOC; therefore, each time the total capacity estimate is updated, two estimates of SOC are required to compute $x = z(t_2) - z(t_1)$. This gives an overall $\sigma_x^2 = 2\sigma_z^2 = 2(0.01)^2$. We simulate this by computing the "measured" value of x_i to be equal to the true value of x_i added to a zero-mean Gaussian random number having variance σ_x^2.

We compute the true value of y_i to be equal to the nominal capacity of the cell Q_{nom} multiplied by the true value of x_i. Noise on the y_i measurement is assumed to comprise accumulated quantization noises (for other noise factors, see section 7). For y_i computed by summing m_i measurements, taken at a 1 Hz rate, from a sensor having quantizer resolution q, the total noise is $\sigma_{y_i}^2 = q^2 m_i/(12 \times 3600^2)$. For HEV scenario 1, we assumed that the maximum range of the current sensor is $\pm 30Q_{nom}$ and that a 10 bit A2D is used to measure current. This leads to $q = 60/1024$. We chose $m_i = 300$ s for every measurement and a nominal capacity of $Q_{nom} = 10$ Ah. The recursive estimates were not initialized prior to the first data point being received (equivalently, the recursive parameters were initialized to zero).

Results from this scenario are presented in Fig. 3. The left frame shows estimates made using the four methods evolving over time as thick lines, and their three-sigma error bounds, computed using the Hessian method, as thin lines. We see that WTLS, TLS, and AWTLS give identical estimates and error bounds under this scenario, and converge to the neighborhood of the true total capacity. The WLS estimate is biased, and the error bounds are (incorrectly) so tight that they are indistinguishable from the estimate itself. The right frame shows the goodness of fit metric as applied to the

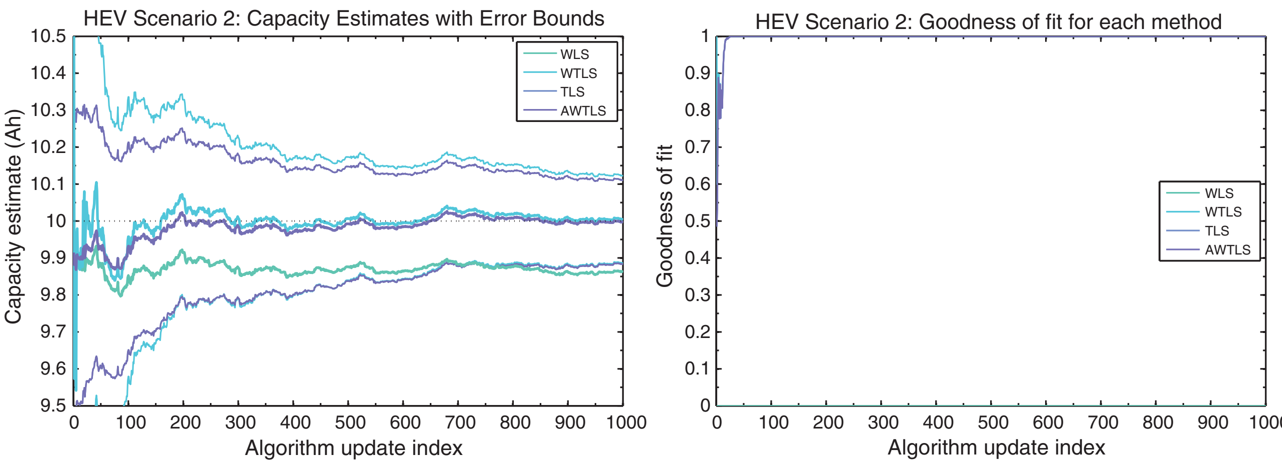

Fig. 4. Results from hybrid electric vehicle application simulation scenario 2.

four methods. Again, WTLS, TLS, and AWTLS give identical results, quickly converging to a value of 1.0 (i.e., the methods are confident that their estimate is reliable). The WLS method returns a vanishingly small value for goodness of fit, reflecting the fact that the method does not give a good value for its total capacity estimate and/or bounds.

6.2. Hybrid electric vehicle application, scenario 2

The second HEV scenario is identical to the first, except that the recursive methods are initialized with a total capacity estimate before further measurements are received. In this case, the methods were initialized with a "nominal" capacity estimate of 9.9 Ah (the true total capacity was still 10.0). Results are presented in Fig. 4. In this scenario, TLS and AWTLS give identical results for both their estimates, error bounds, and goodness of fit. WTLS cannot be calculated recursively, so the estimation cannot be initialized—its results are the same as for scenario 1. Once again, WLS is inferior to the other methods. TLS and AWTLS give the best results because of tighter error bounds while maintaining a high value for goodness of fit.

6.3. Hybrid electric vehicle application, scenario 3

In the third HEV scenario, we explore the ability of the algorithms to track a moving value of total capacity. This scenario is identical to HEV scenario 2, except that the true total capacity is changing with a slope of −0.001 Ah per measurement update, and a fading memory forgetting factor of $\gamma = 0.99$ is used for all methods. Results are presented in Fig. 5, where the true total capacity is drawn as a dotted black line. In this example, the WLS method *appears* to give good results, but its goodness of fit value is still vanishingly small because the error bounds are unreasonably tight, and almost never surround the true value of total capacity. WTLS, TLS, and AWTLS are also able to track the moving value of total capacity—TLS and AWTLS give the best results due to the ability to initialize with a reasonable initial value, yielding narrower error bounds.

6.4. Battery electric vehicle application, scenario 1

The next scenarios that we consider are typical of battery electric vehicle and plug-in hybrid electric vehicle operation. These are different from HEV application in several respects: the battery total capacity is larger, the relative rate of energy usage is lower, the range of SOC used by the vehicle is larger, and the BEV battery pack is sometimes fully charged to a known set point. In all cases, we consider a battery pack with total capacity of $Q_{nom} = 100$ Ah, and a maximum rate of $\pm 5Q_{nom}$. We again assume a 10-bit current sensor, which gives $q = 10/1024$ and $\sigma_{y_i}^2 = q^2 m_i/(12 \times 3600^2)$.

For the first BEV scenario we assume that the total capacity estimate is updated on a regular basis as the vehicle operates, with $m_i = 7200$ s. We assume that the battery SOC can change by ±40% in that interval, so the true value of x_i is chosen to be a uniform ran-

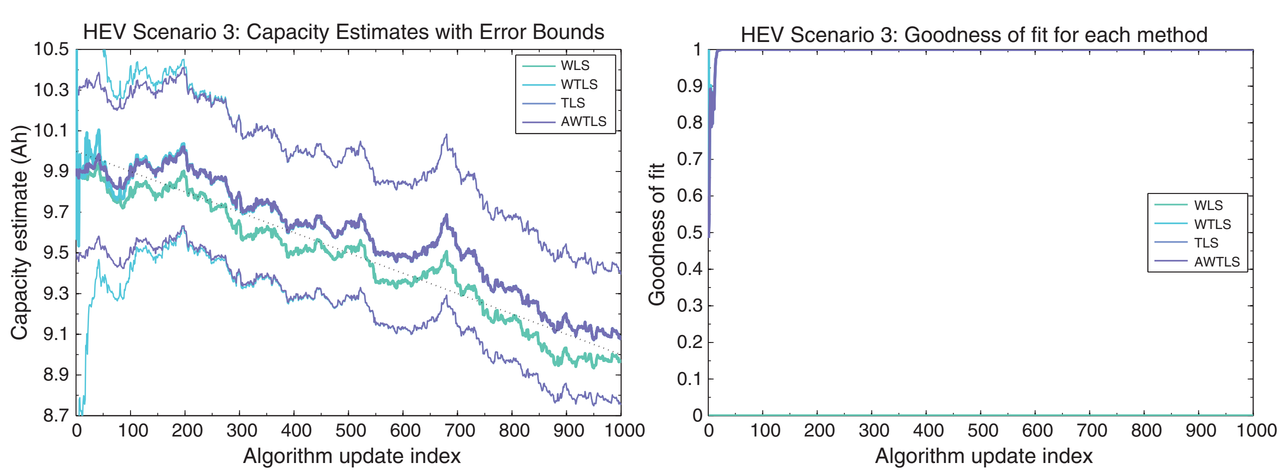

Fig. 5. Results from hybrid electric vehicle application simulation scenario 3.

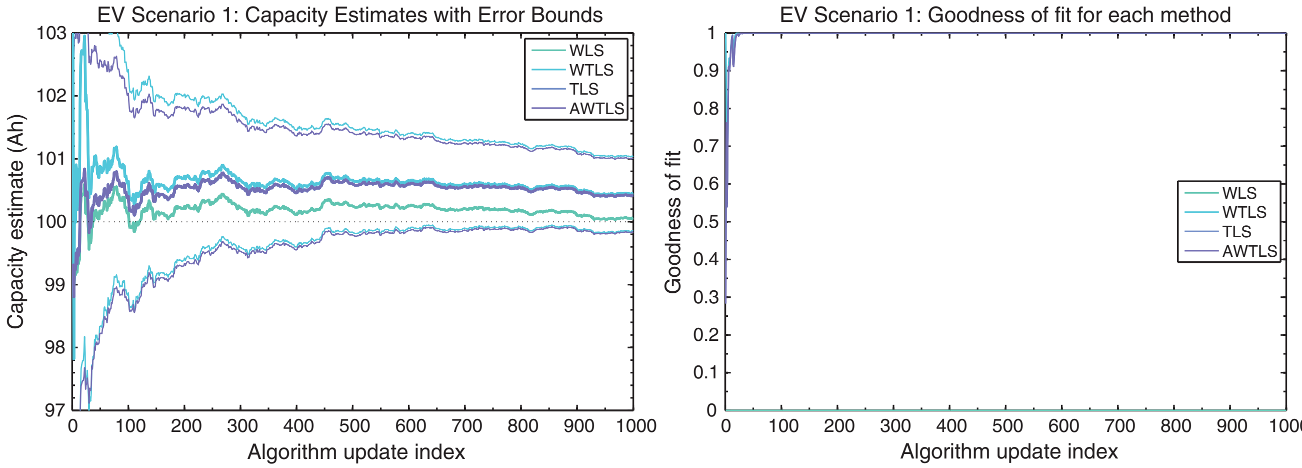

Fig. 6. Results from battery electric vehicle application simulation scenario 1.

dom variable between −0.4 and +0.4. Again, noise on x_i is Gaussian with variance $\sigma_{x_i}^2 = 2(0.01)^2$. The recursive methods are initialized with an initial total capacity estimate of 99 Ah.

Representative results of this scenario are presented in Fig. 6. These are very similar in most respects to the HEV scenario 2 results. Again, WLS fails because its error bounds are far too tight, leading to a vanishingly small goodness of fit. WTLS, TLS, and AWTLS all give good results, with TLS and AWTLS giving the best results due to lower error bounds because of the possibility of initialization.

6.5. *Battery electric vehicle application, scenario 2*

The asymptotic quality of the total capacity estimates is limited by the noise on the SOC estimation error. If this noise can be reduced, the total capacity estimates can become much more accurate. The BEV application allows a means to do this: whenever the battery pack is fully charged, we have a precisely known end-point SOC. Therefore, either $z(t_1)$ or $z(t_2)$ can be known "exactly" for every total capacity estimate update. This then allows us to use $\sigma_{x_i}^2 = \sigma_z^2 = (0.01)^2$.

The tradeoff is that we no longer have regular updates. Updates happen randomly, whenever the vehicle is charged. Therefore, m_i becomes a random variable. For this work, we treat m_i as a log-normal random variable with mode 0.5 h and standard deviation 0.6 h. This gives the probability density function of drive cycle durations that is shown in Fig. 7. (Other PDFs could easily be used: this one was chosen to give reasonable duration drive cycles that encompassed a variety of driving behaviors and distances.) Also, since a greater fraction of the battery pack would be used for an entire drive cycle than for a regular periodic update, we use an 80% range of SOC, so the true value of x_i is computed to be a uniform random number from −0.8 to +0.8.

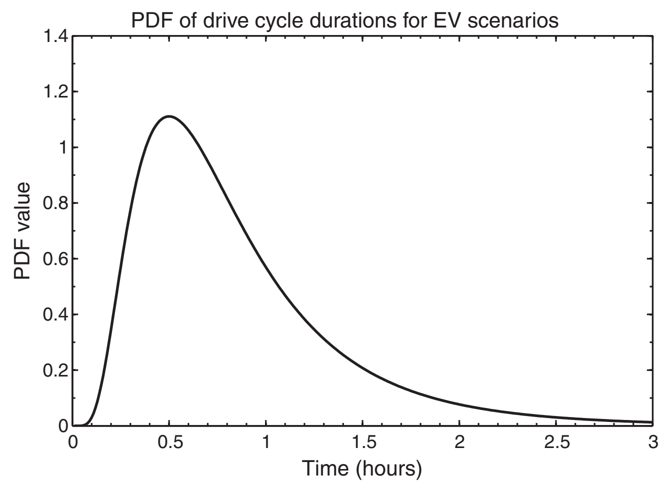

Fig. 7. PDF of drive cycle durations for EV scenarios 2 and 3.

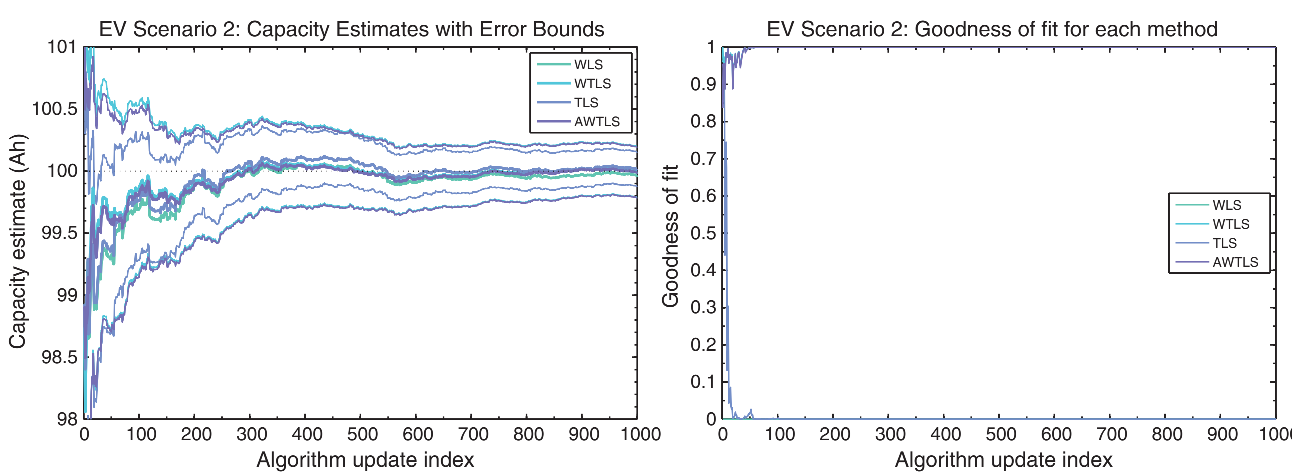

Fig. 8. Results from battery electric vehicle application simulation scenario 2.

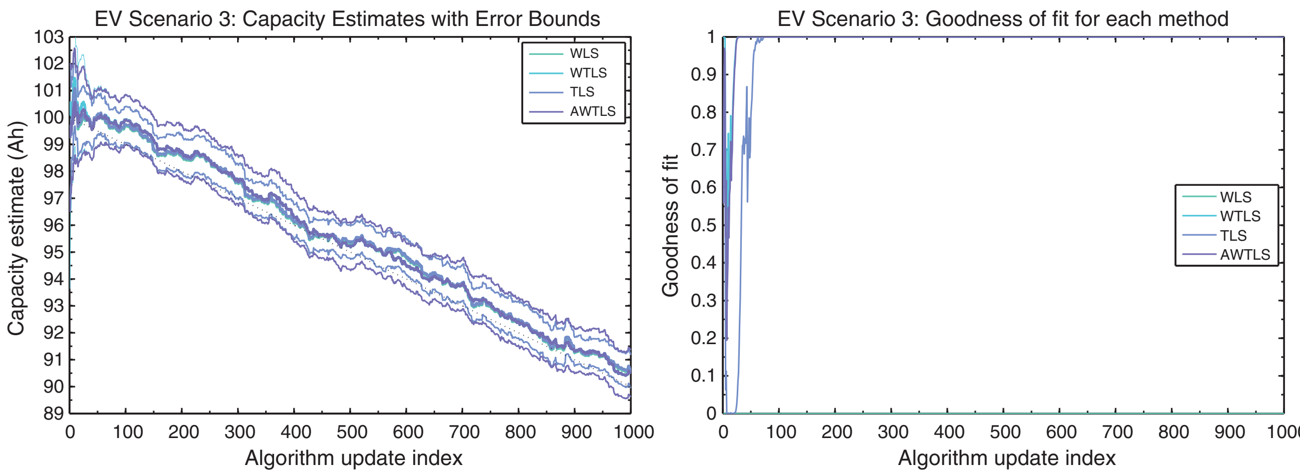

Fig. 9. Results from battery electric vehicle application simulation scenario 3.

Results from this scenario are presented in Fig. 8. WLS fails once again, but this time TLS also fails. TLS fails because $\sigma_{x_i} \neq k\sigma_{y_i}$ due to the variable-length drive cycles. The estimate given by TLS is actually quite reasonable, but the goodness of fit is very small. WTLS gives good results, and AWTLS gives the best results due to lower error bounds because of the ability to initialize the estimate.

Note that the asymptotic three-sigma error bounds drop from about $\pm 1\%$ to about $\pm 0.15\%$ of the total capacity due to having a lower value of σ_{x_i} and also due to the wider range in x_i.

6.6. *Battery electric vehicle application, scenario 3*

The final scenario we consider is identical to BEV scenario 2, except that we simulate a changing total capacity. The slope of the total capacity curve is chosen to be −0.01 Ah per measurement update, and $\gamma = 0.98$ was used. Representative results of this scenario are presented in Fig. 9. Once again, WLS fails and TLS is uncertain of its estimate for nearly 100 updates. However, TLS does recover and do quite well. The AWTLS method gives the best results.

7. Discussion

The simulation results have illustrated a few key properties of the four methods we have discussed to estimate total capacity:

- Noise on the SOC estimates must be considered in order to properly estimate battery total capacity. Least squares, weighted least squares, and other similar methods simply fail. They give biased estimates of total capacity, with unreliable error bounds. Methods related to total least squares, where noises on the SOC estimates are explicitly recognized and incorporated in the calculations, are required for reliable total capacity estimation.
- WTLS, in principle always gives the best results. However, we have seen that in practice that the TLS and AWTLS methods can give better results because they can be initialized with a nominal capacity estimate. Furthermore, since TLS and AWTLS give nice recursive solutions, one of them should always be used instead of WTLS.
- If the measurement update interval m_i is constant, and therefore $\sigma_{x_i} = k\sigma_{y_i}$ for all measurements, TLS and AWTLS give identical results. Therefore, the simpler TLS method is preferred. However, if $\sigma_{x_i} \neq k\sigma_{y_i}$, the AWTLS method gives better results than TLS, and sometimes TLS fails. This is particularly important for the BEV application where total capacity estimate updates are done when charging the battery, to yield a greatly improved total capacity estimate because of the reduction in σ_{x_i} due to knowing one SOC value exactly. AWTLS always gives results at least as good as the other methods.
- The noises that contribute to σ_{y_i} are assumed to be due to current sensor errors. In practice, these can include gain errors, bias errors, noise errors, and nonlinear errors. We have considered only the noise errors here. Gain errors and nonlinear errors will bias all of the methods; however, we believe that the biased value of the total capacity estimate will be consistent with the perceived capacity of the battery pack if the same current sensor is used to compute the battery pack total capacity estimate and to monitor pack operations. Bias error can be subtracted in a BEV setting if we can assume that the Coulombic efficiency of the cells is $\eta \approx 1$ by matching the discharged ampere hours from usage with the charged ampere hours.
- The output error bounds on the total capacity estimate, even with the optimum WTLS estimator, are larger than some might expect. This underscores the need for a method that predicts not only the estimate, but also dynamic error bounds on the estimate, as do the ones proposed in this article. Without dynamic error bounds, the user of the total capacity estimate has no idea how good or bad that estimate is. If the estimate is used to compute battery pack available energy, for example, the energy estimate may be overly optimistic or overly pessimistic, neither of which is acceptable.

8. Conclusions

In this paper, we have investigated several methods to estimate battery cell total capacity. We have shown that standard methods, which consider noises on only the accumulated ampere hours and not on the SOC estimates, are biased and give incorrect error bounds on their estimates. Weighted total least squares considers noise on both the accumulated ampere hours and on the SOC estimates used to estimate total capacity, and hence the total capacity estimates are far more robust. Weighted total least squares, however, does not allow a recursive solution, so is inappropriate for an embedded systems implementation. Standard total least squares can be implemented recursively, and under some scenarios, such as the HEV application, gives results that are as accurate as weighted total least squares. However, when the noises on the accumulated ampere hours and on the SOC estimates are not proportional, total least squares fails. To address this shortcoming, we have introduced an approximate weighted total least squares method that is recursive, and gives results that are indistinguishable from those from weighted total least squares in all settings and are at least as good as all methods considered. Furthermore, estimate error bounds and a goodness-of-fit metric may be readily computed.

Acknowledgments

This work was supported in part by American Electric Vehicles Inc. (AEV). The use of company facilities, and many enlightening discussions with Dr. Dan Rivers and others are gratefully acknowledged.

References

[1] E. Barsoukov, D.R. Poole, D.L. Freeman, Circuit and method for measurement of battery capacity fade. US Patent 6,892,148 (May 2005).
[2] G. Durando, G. Mana, Metrologia 39 (2002) 489–494.
[3] G.F. Franklin, J.D. Powell, A. Emami-Naeini, Feedback Control of Dynamic Systems, chapter 3, fifth edition, Pearson Prentice Hall, 2006.
[4] W. Gould, W. Sribney, Maximum Likelihood Estimation with STATA, chapter 1, Stata Press, 2006.
[5] S. Van Huffel, J. Vandewalle, Society for Industrial and Applied Mathematics (1991).
[6] M. Krystek, M. Anton, Measurement Science and Technology 18 (2007) 3438–3442.
[7] I. Markovsky, S. Van Huffel, Signal Processing 87 (10) (2007) 2283–2302, Special Section: Total Least Squares and Errors-in-Variables Modeling.
[8] T. Okada, Method of calculating rechargeable battery charge capacity. US Patent 5,789,924 (August 1998).
[9] S. Piller, M. Perrin, A. Jossen, Journal of Power Sources 96 (2001) 113–120.
[10] G.L. Plett, Journal of Power Sources 134 (August (2)) (2004) 252–261.
[11] G.L. Plett, Journal of Power Sources 134 (August (2)) (2004) 262–276.
[12] G.L. Plett, Journal of Power Sources 134 (August (2)) (2004) 277–292.
[13] G.L. Plett, Journal of Power Sources 152 (2) (2006) 1356–1368.
[14] G.L. Plett, Journal of Power Sources 152 (2) (2006) 1369–1384.
[15] W.H. Press, W.T. Vetterling, S.A. Teukolsky, B.P. Flannery, Numerical Recipes in C: The Art of Scientific Computing, chapter 9, second edition, Cambridge University Press, 1992.
[16] B.C. Reed, American Journal of Physics 57 (July (7)) (1989) 642–646.
[17] E.W. Weisstein, in: MathWorld—A Wolfram Web Resource, http://mathworld.wolfram.com/QuarticEquation.html.