Voltage Sensor Fault Detection in Li-ion Battery Energy Storage Systems

Namireddy Praveen Reddy#1, Yuxuan Cai*2, Roger Skjetne#3, Dimitrios Papageorgiou**4
#*Department of Marine Technology*, *Norwegian University of Science and Technology*
Trondheim, Norway
NAMIREDDY.P.REDDY@NTNU.NO
ROGER.SKJETNE@NTNU.NO
**Nordic Semiconductor ASA*
Oslo, Norway
YUXUAN.CAI@NORDICSEMI.NO
***Department of Electrical Engineering*, *Technical University of Denmark*
Kgs. Lyngby, Denmark
DIMPA@ELEKTRO.DTU.DK

***Abstract*—Safe and optimal operation of battery energy storage systems requires correct measurement of voltage, current, and temperature. Therefore, fast and correct detection of sensor faults is of great importance. In this paper, model-based and non-model-based voltage sensor fault detection methods are developed for a comprehensive comparison. The residual is generated from the difference of measured voltage and estimated voltage. In the model-based method, the voltage is estimated using an extended Kalman filter (EKF). In the non-model-based method, the voltage is predicted using a recurrent neural network (RNN) with long short-term memory (LSTM). For both methods, a scalar generalized likelihood ratio (GLR) detector is developed to detect changes in the sequence of residual signal data and compared with a systematically computed threshold. The parameters threshold (h) and window-size (M) used in the GLR detector, are computed based on the probability of false alarm ($\mathrm{P_f}$) and probability of correct detection ($\mathrm{P_d}$). The GLR detector demonstrates the ability to effectively detect the voltage sensor fault with a maximum delay of 500 ms for the model-based residual and 200 ms for the non-model-based method.**

***Index Terms*—Battery Energy Storage Systems, Sensor faults, Fault detection and diagnosis, Extended Kalman filter, Machine learning.**

I. Introduction

In battery energy storage systems (BESS), a battery management system (BMS) ensures safe and reliable operation by incorporating several functions such as data collection, state of charge (SOC) and state of health (SOH) estimation, cell balancing, charge/discharge control, and fault detection and diagnosis [1] [2]. To perform these functions, the BMS relies on data collected by the voltage, current, and temperature sensors installed on each cell of the battery pack. Typically, a battery pack consists of 100s of cells. Hence, the correspondingly large number of sensors installed implies a high probability of sensor faults. Such sensor faults can affect optimal operation of the BMS and may cause safety hazards. For instance, a voltage sensor fault can lead to overcharging or undercharging, which eventually may lead to an internal short circuit [3]. Furthermore, this can lead to wrong estimation of SOC and SOH, which not only define the safety margin of battery to avoid overcharge or overdischarge, but also help to make full utilization of battery capacity potential [4]. Therefore, it is important to detect the voltage sensor fault quickly and correctly.

The existing literature covers model-based and non model-based methods, such as machine learning for residual generation and sensor fault detection in several industrial applications [5], [6], [7], [8]. Structural analysis is used to find the residual and the statistical cumulative sum (CUSUM) test is applied to detect and isolate the current, voltage, and temperature sensor fault in [9]. A current sensor fault diagnostic method combining the particle swarm optimization-based residual generation and the statistical residual evaluation is proposed by [10]. Model-based residual generators that are sensitive to different faults are developed using structural analysis and EKF in [11]. A fault diagnostic scheme based on hybrid system and dual extended Kalman filter algorithm for sensor and relay faults in lithium-ion battery pack is presented in [12]. Neural network-based fault diagnosis is developed for detecting thermal runaway in [13] and [14]. A novel voltage prediction using long short-term memory(LSTM) recurrent neural network (RNN) together with a weather-vehicle-driver analysis is implemented for fault prognosis in [15].

Much of the existing research is lacking a comprehensive comparison of model-based and non-model-based methods for residual generation and fault detection and diagnosis. This paper attempts to address the existing research gap by developing a comprehensive comparison framework, where a model-based method (EKF) and a non-model-based method (LSTM-RNN) are implemented for residual generation. The residuals are evaluated by a scalar GLR detector, which is designed by systematically computing the threshold (h) and window size (M) based on the values of the probability of false alarm ($\mathrm{P_f}$) and probability of correct detection ($\mathrm{P_d}$).

This paper is organized into 6 sections. Section I is the introduction, covers the main motivation for this research, brief state-of-the-art literature, contributions compared with the existing literature, and outline. The lithium-ion battery model is presented in Section II. A model-based and non-model-based voltage estimation methods are covered in Section III. Design of the fault detection algorithm is explained in Section IV. Simulation results and observations are discussed in Section V. Conclusions are summarized and presented in Section-VI.

II. Lithium-ion Battery Model

A first order RC model, as shown in Fig. 1, is used for modeling the lithium-ion battery cells based on [16]. The model consists of an ideal voltage source $\mathrm{V_{oc}}$, a resistor $\mathrm{R_o}$, an RC branch (the polarization resistor $\mathrm{R_p}$ and the polarization capacitor $\mathrm{C_p}$ connected in parallel). In addition, $\mathrm{V_t}$ and $\mathrm{i_L}$ represent the terminal voltage and load current, respectively. The state-space model of the first order RC model in discrete-time domain can be represented as

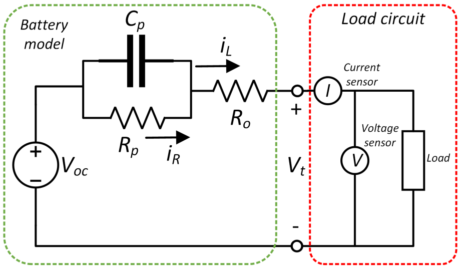

Fig. 1. Lithium-ion battery model.

$$\begin{bmatrix} \mathrm{z_{k+1}} \\ \mathrm{i_{R_p,k+1}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{A_{R_pC_p}} \end{bmatrix} \begin{bmatrix} \mathrm{z_k} \\ \mathrm{i_{R_p,k}} \end{bmatrix} + \begin{bmatrix} -\frac{\eta_k \Delta t}{Q} \\ 1 - \mathrm{A_{R_pC_p}} \end{bmatrix} \mathrm{i_k} \tag{1}$$

where $\mathrm{z_k}$ is the state of charge (SOC) at time step k, $\mathrm{i_{R_p,k}}$ is the current through the resistor $\mathrm{R_p}$ at time step k, and $\mathrm{A_{R_pC_p}} = \exp\left(\frac{-\Delta t}{\mathrm{R_pC_p}}\right)$. The current supplied by the battery to the load, $\mathrm{i_k}$, is input to the system. The output voltage $\mathrm{v_k}$ at time step k is expressed as

$$\mathrm{v_k} = \mathrm{OCV(z_k)} - \mathrm{R_p i_{R_p,k}} - \mathrm{R_o i_k}. \tag{2}$$

The parameters of a lithium-ion battery can be computed through the two tests: 1) an open circuit voltage (OCV) test, and 2) a dynamic test performed in a laboratory.

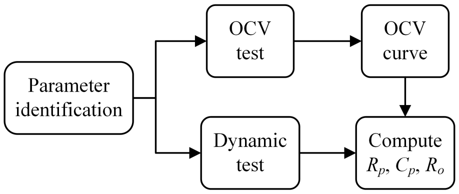

Fig. 2. Parameter identification.

A. Open Circuit Voltage (OCV) Test

In an OCV test, the battery is charged/discharged at a very low C-rate (C/20). The measured voltage can be considered approximately as the OCV, i.e., $\mathrm{v_k} \approx \mathrm{OCV(z_k)}$ since the current is very low. The coulombic efficiency is computed as

$$\eta = \frac{\text{total ampere-hours discharged at all instants}}{\text{total ampere-hours charged at all instants}}, \tag{3}$$

which is a function of temperature. Therefore, the result is only valid at a certain temperature. The corresponding SOC can be computed using the measured accumulated ampere-hours and computed coulombic efficiency. The OCV curve is an expression of how the OCV varies with the SOC as obtained by averaging the data during charging and discharging. The OCV curve is also valid at only a certain temperature. Fig. 3 shows the OCV curve obtained for the Li-ion battery considered in this work, at temperature 25^oC.

B. Dynamic Test

In the dynamic test, the cell is charged/discharged at a constant current (C/1) to excite the dynamic components. The voltage, current, temperature, ampere-hours charged, and ampere-hours discharged are recorded every second. With these data, the goal is to compute parameter values for $\mathrm{R_o}$ and $\mathrm{R_p}$. Accordingly, the residual between output measurement and OCV is expressed as

$$\tilde{\mathrm{v}}_\mathrm{k} = \mathrm{v_k} - \mathrm{OCV(z_k)} = -\mathrm{R_p i_{R_p,k}} - \mathrm{R_o i_k},$$

where $\mathrm{i_{R_p,k}}$ can be computed with subspace system identification, and $\mathrm{i_k}$ is considered a known input. $\mathrm{R_o}$ and $\mathrm{R_p}$ are then computed using a least-squares method.

C. Case-Study: Dataset used

In this work, the dataset from [17]-[18] is used. In [18], a 2.9 Ah Panasonic NCA 18650PF cell was tested in a thermal chamber under varying conditions. The cell specifications can be found in Table I.

TABLE I
Panasonic 18650PF cell specifications [18]-[19].

Parameter	Value
Nominal open circuit voltage	3.6 V
Nominal Capacity	Min. 2.75 Ah/Typ. 2.9 Ah
Min/max voltage	2.5 V/4.2 V
Minimum charging temperature	10 °C
Cycles to 80% capacity	500 (100% DOD, 25 °C)

The tests include charging/discharging cycles at 1C, at C/20, and a series of nine drive cycle tests. The drive cycles used one, or a mix of US06, HWFET, UDDS, LA92, and a custom neural network drive cycle. This dataset is suitable for system identification as it contains data of both OCV tests and dynamic tests. Moreover, this dataset has a large amount of data, which can be further used for machine learning-based estimation of SOC and voltage. Other available datasets either lack both tests for cell modeling or only have one or two drive cycles; far from enough for a machine learning algorithm.

The system identification is based on the enhanced self correcting (ESC) toolbox from [20]. The first step in the implementation is to identify the OCV relationship. With the OCV test data included in the dataset, we approximate the OCV curve as shown in Fig. 3. Using the OCV relationship and the dynamic test data, the parameters of the cell model are identified using least-square regression, resulting in: $\mathrm{R_o = 32m\Omega, R_p = 37.8m\Omega, C_p = 4.47F}, \eta_\mathrm{k} = 0.97$, $\mathrm{Q = 2.8Ah}$.

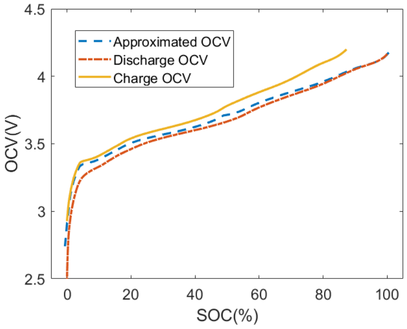

Fig. 3. Polarization curve: OCV vs SOC at $T = 25^oC$.

III. Voltage estimation

A. Model based Method: Extended Kalman Filter

The EKF from [21] is implemented in this work. The EKF is based on the following two assumptions:

- The expected value of a nonlinear function of the unknown state is assumed equal to the same nonlinear function evaluated at the expected value of the state.
- The EKF linearizes the nonlinear system using a truncated Taylor-series expansion around the current operating point. In Taylor-series expansion, higher-order terms are assumed negligible and discarded. For this reason, the EKF performs well for systems with mild nonlinearities.

A typical nonlinear system is represented by

$$\mathrm{x_{k+1}} = \mathrm{f(x_k, u_k, w_k)} \tag{4}$$

$$\mathrm{y_k} = \mathrm{h(x_k, u_k, v_k)}. \tag{5}$$

Here, $\mathrm{x_k, u_k}$, and $\mathrm{y_k}$ denote state vector, input vector and output vector, respectively. Process noise $\mathrm{w_k}$ and measurement noise $\mathrm{v_k}$ are assumed to be uncorrelated white Gaussian noise, with zero mean and covariance matrices having the properties:

$$\mathbb{E}\left[\mathrm{w_k w_j^T}\right] = \begin{cases} \Sigma_{\tilde{\mathrm{w}}}, & \mathrm{k = j} \\ 0, & k \neq j; \end{cases} \quad \mathbb{E}\left[\mathrm{v_k v_j^T}\right] = \begin{cases} \Sigma_{\tilde{\mathrm{v}}}, & \mathrm{k = j} \\ 0, & \mathrm{k \neq j}. \end{cases} \tag{6}$$

The EKF performs two steps: prediction and correction in each time step. Each of these two steps consists of three sub-steps as shown in Fig. 4. The inputs for the EKF are the mean ($\hat{x}_0 = \mu_0$) and error covariance of the initial state ($\Sigma_{\tilde{x}_0}$).

1) Prediction: There are three sub-steps to perform a priori estimation of the state, state-error covariance, and output. The state prediction is approximated using the first assumption of EKF, by

$$\mathrm{\hat{x}_k^-} \approx \mathrm{f}\left(\mathrm{\hat{x}_{k-1}^+, u_{k-1}, \bar{w}_{k-1}}\right). \tag{7}$$

The expected value of new state is approximated by assuming that it is reasonable to propagate $\mathrm{\hat{x}_{k-1}^+}$ and $\mathrm{\bar{w}_{k-1}^-}$ through the state equation. In the state-error covariance prediction step, $\mathrm{\tilde{x}_k^-}$ is approximated as

$$\begin{aligned} \mathrm{\tilde{x}_k^-} &= \mathrm{x_k - \hat{x}_k^-} \\ &= \mathrm{f\left(x_{k-1}, u_{k-1}, w_{k-1}\right) - f\left(\hat{x}_{k-1}^+, u_{k-1}, \bar{w}_{k-1}\right)} \\ &\approx \mathrm{\hat{A}_{k-1}\tilde{x}_{k-1}^+ + \hat{B}_{k-1}\tilde{w}_{k-1}}. \end{aligned} \tag{8}$$

The state-error co-variance can be computed by

$$\Sigma_{\mathrm{\tilde{x}_k}}^- = \mathrm{\hat{A}_{k-1}}\Sigma_{\mathrm{\tilde{x}_{k-1}}}^+ \mathrm{\hat{A}_{k-1}^T} + \mathrm{\hat{B}_{k-1}}\Sigma_{\tilde{\mathrm{w}}}\mathrm{\hat{B}_{k-1}^T}. \tag{9}$$

The system output is finally by

$$\mathrm{\hat{y}_k} \approx \mathrm{h}\left(\mathrm{\hat{x}_k^-, u_k, \bar{v}_k}\right), \tag{10}$$

and the expected value of the output is computed by assuming that it is reasonable to propagate the state prediction $\mathrm{\hat{x}_k^-}$ and the mean sensor noise $\mathrm{\bar{v}_k}$ through the output equation.

2) Correction: There are three sub-steps that are computing the Kalman gain, posteriori estimation of state, and state-error covariance. The output prediction error is first computed by

$$\begin{aligned} \mathrm{\tilde{y}_k} &= \mathrm{y_k - \hat{y}_k} \\ &= \mathrm{h\left(x_k, u_k, v_k\right) - h\left(\hat{x}_k^-, u_k, \bar{v}_k\right)} \\ &\approx \mathrm{\hat{C}_k\tilde{x}_k^- + \hat{D}_k\tilde{v}_k}, \end{aligned} \tag{11}$$

and the output-error covariance is computed as

$$\Sigma_{\mathrm{\tilde{y}_k}} \approx \mathrm{\hat{C}_k}\Sigma_{\mathrm{\tilde{x}_k}}^- \mathrm{\hat{C}_k^T} + \mathrm{\hat{D}_k}\Sigma_{\tilde{\mathrm{v}}}\mathrm{\hat{D}_k^T}. \tag{12}$$

The Kalman gain is computed as

$$\mathrm{L_k} = \Sigma_{\mathrm{\tilde{x}_k}}^- \mathrm{\hat{C}_k^T}\left[\mathrm{\hat{C}_k}\Sigma_{\mathrm{\tilde{x}_k}}^- \mathrm{\hat{C}_k^T} + \mathrm{\hat{D}_k}\Sigma_{\tilde{\mathrm{v}}}\mathrm{\hat{D}_k^T}\right]^{-1}. \tag{13}$$

The state estimate is computed by updating the state prediction using the estimator gain and $\mathrm{y_k - \hat{y}_k}$, that is,

$$\mathrm{\hat{x}_k^+} = \mathrm{\hat{x}_k^- + L_k\left(y_k - \hat{y}_k\right)}. \tag{14}$$

The state-error covariance is then computed by

$$\Sigma_{\mathrm{\tilde{x}_k}}^+ = \Sigma_{\mathrm{\tilde{x}_k}}^- - \mathrm{L_k}\Sigma_{\mathrm{\tilde{y}_k}}\mathrm{L_k^T}. \tag{15}$$

In this work, matrices $\mathrm{\hat{A}_k}$, $\mathrm{\hat{B}_k}$, $\mathrm{\hat{C}_k}$, $\mathrm{\hat{D}_k}$ for the Li-ion battery model (considered in this work) are computed analytically from the Jacobian of the system model 1-2. Since 1 is already linear, we only need to linearize the OCV curve in 2:

$$\mathrm{\hat{A}_k} = \mathrm{A} = \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{A_{RC}} \end{bmatrix} \tag{16}$$

$$\mathrm{\hat{B}_k} = \mathrm{B} = \begin{bmatrix} -\frac{\eta_\mathrm{k}\Delta \mathrm{t}}{\mathrm{Q}} \\ 1 - \mathrm{A_{RC}} \end{bmatrix} \tag{17}$$

$$\mathrm{\hat{C}_k} = \frac{\partial}{\partial \mathbf{x}_k}\mathbf{h}(\mathbf{x_k}, \mathbf{u_k}, \mathbf{v}_k) = \begin{bmatrix} \frac{\partial \mathrm{OCV(z_k)}}{\mathrm{z_k}} & -\mathrm{R_1} \end{bmatrix} \tag{18}$$

$$\mathrm{\hat{D}_k} = \frac{\partial}{\partial \mathbf{v}_k}\mathbf{h}(\mathbf{x_k}, \mathbf{u_k}, \mathbf{v}_k) = -\mathrm{R_0} \tag{19}$$

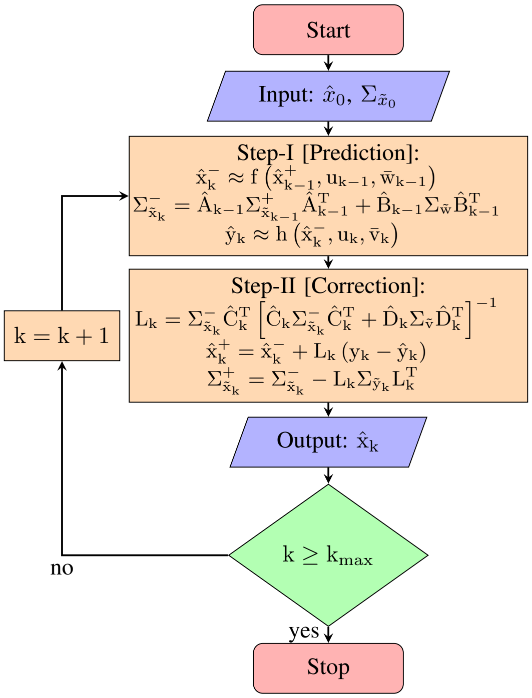
Start
Input: $\hat{x}_0$, $\Sigma_{\tilde{x}_0}$
Step-I [Prediction]:
$\hat{x}_k^- \approx f\left(\hat{x}_{k-1}^+, u_{k-1}, \bar{w}_{k-1}\right)$
$\Sigma_{\tilde{x}_k}^- = \hat{A}_{k-1}\Sigma_{\tilde{x}_{k-1}}^+\hat{A}_{k-1}^T + \hat{B}_{k-1}\Sigma_{\tilde{w}}\hat{B}_{k-1}^T$
$\hat{y}_k \approx h\left(\hat{x}_k^-, u_k, \bar{v}_k\right)$
Step-II [Correction]:
$L_k = \Sigma_{\tilde{x}_k}^- \hat{C}_k^T\left[\hat{C}_k\Sigma_{\tilde{x}_k}^-\hat{C}_k^T + \hat{D}_k\Sigma_{\tilde{v}}\hat{D}_k^T\right]^{-1}$
$\hat{x}_k^+ = \hat{x}_k^- + L_k\left(y_k - \hat{y}_k\right)$
$\Sigma_{\tilde{x}_k}^+ = \Sigma_{\tilde{x}_k}^- - L_k\Sigma_{\tilde{y}_k}L_k^T$
Output: $\hat{x}_k$
$k \geq k_{max}$
no
yes
$k = k + 1$
Stop

Fig. 4. Flowchart of Extended Kalman Filter.

B. Data-driven method: Machine Learning

Recurrent neural network (RNN) with long short term memory (LSTM) is used for voltage estimation due its ability to learn lengthy sequential data dependencies. LSTM-RNN used in this work, has three layers as shown in Fig. 5. Sequential data of current and SOC are the inputs to the LSTM-RNN and sequence of voltage is the output. Number of hidden layers used in this work are 120. Five drive cycles (US06, NN, and mixed cycles 2-4) are used as the training set, while one drive cycle (UDDS) is used as the validation set. LSTM-RNN is trained for a maximum of 100 epochs with a minibatch size of 256.

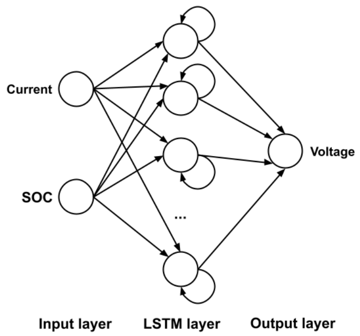

Fig. 5. LSTM-RNN topology for voltage prediction.

C. Validation

Fig. 6 and Fig. 7 show the measured ($\mathrm{V_m}$) and estimated voltages ($\hat{\mathrm{V}}_{\mathrm{EKF}}$ & $\hat{\mathrm{V}}_{\mathrm{LSTM-RNN}}$), as well as the error between them ($\mathrm{V_{err}}$), absolute $\mathrm{V_{err}}$, and absolute relative V_{err} for UDDS drive cycle. The $\mathrm{V_{err}}$ histogram plots of EKF and LSTM-RNN are shown in Fig. 8 and Fig. 9, respectively, which show that they are approximately Gaussian. Table. II shows key performance indicators such as mean average error (MAE), root square mean error (RSME), and maximum absolute relative error ($|\mathrm{V_{err}}|_{\mathrm{max}}[\%]$) for EKF and LSTM-RNN. It can be observed that the KPIs for both methods are within the error bounds of 5 %. Not only LSTM-RNN performs better than EKF in all the KPIs, but also LSTM-RNN requires only measured data for voltage estimation, whereas, EKF requires tedious laboratory experiments for system identification. Furthermore, LSTM-RNN has the potential to consider different factors such as temperature and aging when relevant and a large enough dataset is available.

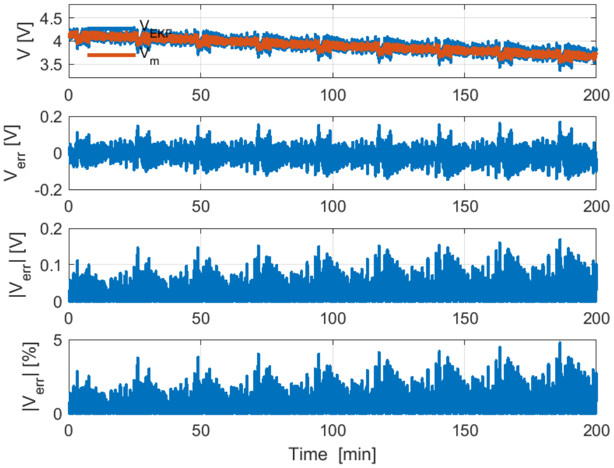

Fig. 6. Measured voltage ($\mathrm{V_m}$), estimated voltage using EKF ($\hat{\mathrm{V}}_{\mathrm{EKF}}$), voltage error ($\mathrm{V_{err}}$), absolute $\mathrm{V_{err}}$, and absolute relative $\mathrm{V_{err}}$.

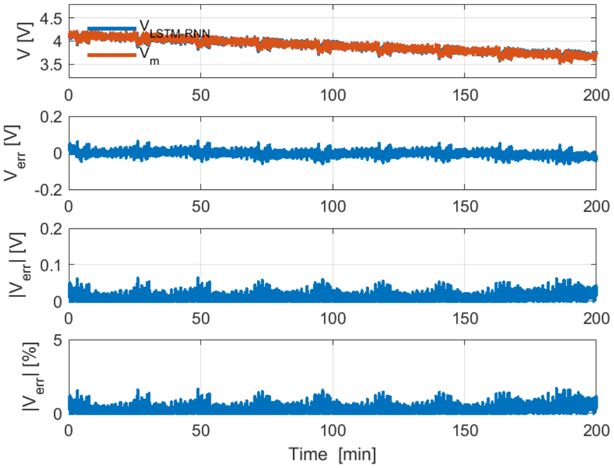

Fig. 7. Measured voltage ($\mathrm{V_m}$), estimated voltage using EKF ($\hat{\mathrm{V}}_{\mathrm{LSTM-RNN}}$), voltage error ($\mathrm{V_{err}}$), absolute $\mathrm{V_{err}}$, and absolute relative $\mathrm{V_{err}}$.

TABLE II
KEY PERFORMANCE INDICATORS FOR EKF AND LSTM-RNN.

Method	**MAE [%]**	**RSME [%]**	$\mathbf{\lvert V_{err}\rvert_{max}[\%]}$
EKF	0.58	0.82	4.78
LSTM-RNN	0.23	0.30	1.68

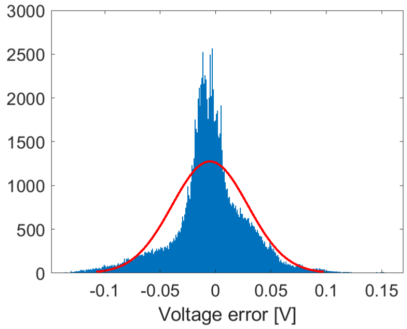

Fig. 8. Histogram of $\mathrm{V_{err}}$ in EKF.

IV. FAULT DETECTION & DIAGNOSIS

Fault detection of the voltage sensor fault in a lithium ion battery, is achieved by the analysis of a residual. In this work, the residual is generated from the error obtained by comparing the measured voltage and estimated voltage. As shown in the previous sections, the voltage is estimated using two approaches: 1) model-based EKF, and 2) data-driven LSTM-RNN.

The Generalised Likelihood Ratio (GLR) test is used for detecting an unknown change in the magnitude of the voltage residual. This change in mean corresponds to an additive fault in the voltage sensor. The scalar GLR decision function $\mathrm{g_M(k)}$ takes the form [22]

$$\mathrm{g_M(k)} = \frac{1}{2\sigma^2\mathrm{M}}\left[\sum_{\mathrm{i=k-M+1}}^{\mathrm{k}}(\mathrm{r(i)}-\mu_0)\right]^2 \tag{20}$$

where σ^2 and μ_0 denote the variance and the mean of the sequence of residual data, respectively. k and M are time-step and window size respectively. $\mathrm{r(i)}$ is the value of the residual at $\mathrm{i^{th}}$ time-step. Fig. 10 shows the flowchart of the fault detection scheme. The residual $(\mathrm{r_k})$ is obtained by the difference between the measured and estimated voltages. The sequence of the residual data is analyzed through the scalar GLR test, where the computed GLR decision function $\mathrm{g_M(k)}$ is compared with a threshold (h) to detect a potential fault.

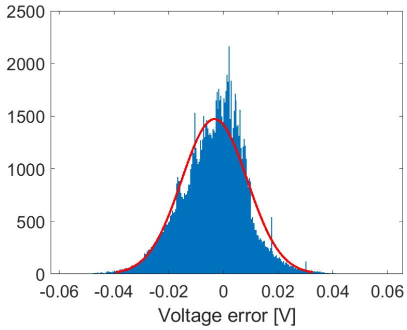

Fig. 9. Histogram of $\mathrm{V_{err}}$ in LSTM-RNN.

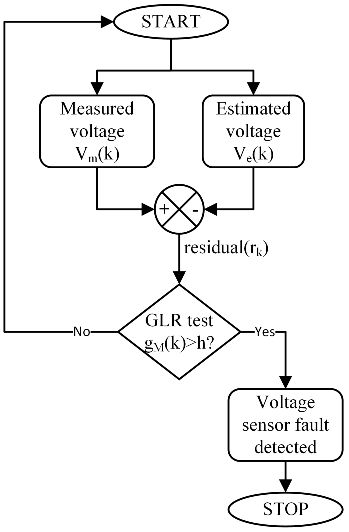
START
Measured voltage $V_m(k)$
Estimated voltage $V_e(k)$
+
-
residual(r_k)
GLR test $g_M(k)>h$?
No
Yes
Voltage sensor fault detected
STOP

Fig. 10. Flowchart of fault detection.

A. Design of GLR detector

In this subsection, a method for computing the threshold (h) and window size (M) based on the probability of false alarm $(\mathrm{P_f})$ and probability of correct detection $(\mathrm{P_d})$, are presented.

1) Calculation of threshold (h): The probability of false alarm $(\mathrm{P_f})$ can be computed based on [22], according to

$$\mathrm{P_f} = \int_{\mathrm{2h}}^{\infty}\mathrm{p_{\chi^2}(X;N)dX} \tag{21}$$

Where $\mathrm{p_{\chi^2}}$ is the probability density function of the chi-square distribution, given by

$$\mathrm{p_{\chi^2}(X;N)} = \begin{cases} \frac{1}{2^{N/2}\Gamma\left(\frac{N}{2}\right)}\mathrm{X}^{\frac{N}{2}-1}\mathrm{e}^{-\frac{X}{2}} & \text{if } \mathrm{X} \geq 0 \\ 0 & \text{if } \mathrm{X} < 0. \end{cases}$$

where, Γ is the gamma function. The threshold (h) can then be computed by solving eqn. 21.

2) Calculation of window size (M): The probability of correct detection $(\mathrm{P_d} = 1 - \mathrm{P_m})$ can be computed based on [22],

$$\mathrm{P_d} = 1 - \mathrm{P_m} = \int_{\mathrm{2h}}^{\infty}\mathrm{p_{\chi^2}(X;N;\lambda)dX} \tag{22}$$

where, $\mathrm{P_m}$ is the probability of missed detection. Here, $\mathrm{p_{\chi^2}(X;N;\lambda)}$ is the probability density function of the non-central chi-square distribution given by

$$p_{\chi^2}(X;N;\lambda) = \frac{1}{2}\left(\frac{X}{\lambda}\right)^{\frac{N}{2}-1}\mathrm{e}^{-\frac{X+\lambda}{2}}\iota_{\frac{N}{2}-1}\left(\sqrt{\lambda X}\right),$$

where λ is the non-centrality parameter given by

$$\lambda = \frac{\mathrm{M}(\mu_1-\mu_0)^2}{\sigma^2}.$$

The window size (M) can be computed by solving eqn. 22. Table. III shows the values of $\mathrm{P_f}$ and corresponding h, as well as $\mathrm{P_m}$ and corresponding M for EKF- and LSTM-RNN-based GLR detectors. It can be observed that h increases to reduce the probability of false alarm, whereas M increases to reduce the probability of missed detection. In this work, the simulation experiment is conducted for 200 minutes, where 10 samples of data is collected each second. Therefore, $\mathrm{P_f} = $ 1e-5 is considered, which gives the probability of approximately 1.2 false alarms in each UDDS drive cycle. The threshold computed by solving eqn. 21 is $\mathrm{h} = 11.51$ for both the methods. Hence, $\mathrm{P_m} = $ 1e-5 is considered, and the window size computed by solving eqn. 22 is $\mathrm{M} = 5$ and $\mathrm{M} = 4$ for EKS- and LSTM-RNN-based GLR detectors, respectively.

TABLE III
GLR-DETECTOR PARAMTERS: h & M.

Threshold (h)			Window-size (M)		
$\mathbf{P_f}$	EKF	LSTM-RNN	$\mathbf{P_m}$	EKF	LSTM-RNN
1e-2	4.6	4.6	1e-2	2	2
1e-3	6.9	6.9	1e-3	3	2
1e-4	9.2	9.2	1e-4	4	3
1e-5	11.51	11.51	1e-5	5	4
1e-6	13.82	13.82	1e-6	6	5

V. Results

In this section, the simulation results for both fault detection algorithms are presented for the UDDS drive cycle. In the considered dataset, the measured values have 10 samples in each second. The voltage sensor fault is modeled as an additive bias equal to 2%, 4%,...,20% of the measured voltage, each for 500 seconds duration (5000 samples). Fig. 11 and Fig. 12 show the voltage sensor fault, measured and estimated voltages, residual signal, $\mathrm{g_M}$, and alarm for EKF- and LSTM-RNN-based GLR detectors, respectively. The results show that both GLR algorithms are able to effectively detect all the bias faults considered. However, the EKF-based GLR detector detects with a maximum time delay of 500 ms, whereas, the LSTM-RNN-based GLR detector detects with a maximum delay of 200 ms. The LSTM-RNN has less time delay due to a lower window size (M). Moreover, when bias faults are ≥ 6 %, both algorithms have a window size of 1 and no time delay in detecting the fault. Furthermore, in EKF-based GLR, it can be observed that there are 46 false alarms when the voltage bias fault is 2 %, 4 %, and ≤ 6 %. This is due to the fact that the GLR detector is designed to detect the fault biases equal to 2 % and 4 %, whereas the maximum absolute error is 4.78 % (see Fig. 6 and Table. II). Therefore, the GLR detector detects all the instances where the voltage residual is more than 2 % and 4 %. This can be further proved by the fact that there are no false alarms when the GLR detector is designed to detect the bias faults ≥ 6 %. Similarly, two false alarms in LSTM-RNN based GLR detector can be explained.

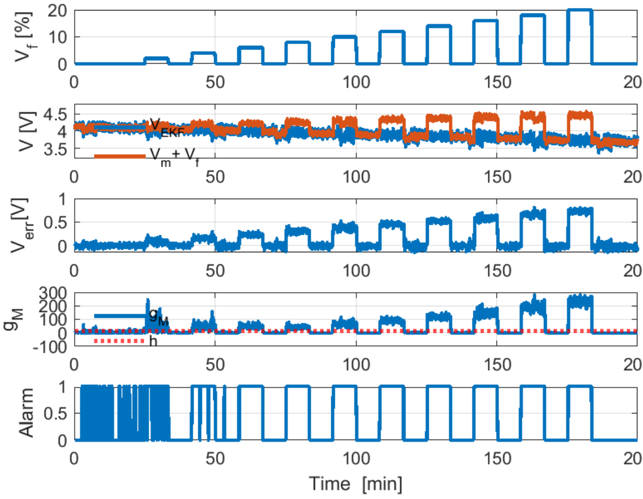

Fig. 11. Voltage sensor fault, measured and estimated voltages, residual signal, $\mathrm{g_M}$, and alarm for EKF-based GLR detector.

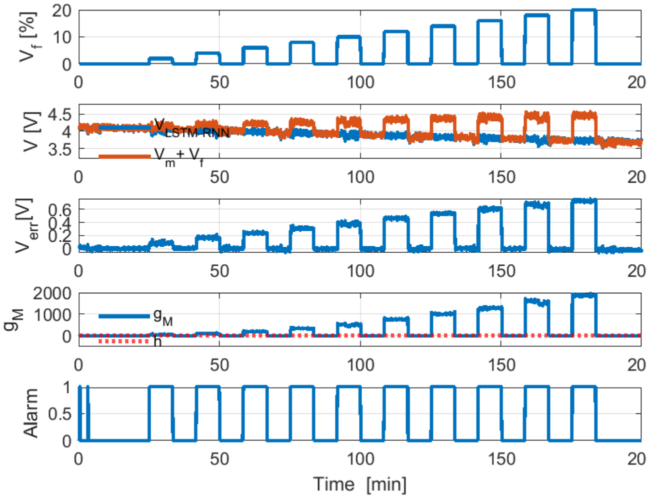

Fig. 12. Voltage sensor fault, measured and estimated voltages, residual signal, $\mathrm{g_M}$, and alarm for LSTM-RNN-based GLR detector.

VI. Conclusion

In this paper, a model-based method (EKF) and a non-model-based method (LSTM-RNN) are implemented for estimating the voltage. The residual is obtained by comparison of the measured and estimated voltages in both methods. In the simple residual analysis methods, the residual $\mathrm{r(k)}$ is typically compared with a predetermined threshold to detect the fault and raise the alarm. However, such methods often do not account for measurement noise, and the threshold is chosen empirically. The scalar GLR algorithm presented in this work provides a systematic approach to statistically compute the parameters, such as threshold and window size, and to detect sensor faults in a stochastic system. The results show that the

GLR detector is capable of efficiently detecting a sensor fault with a small time delay. The LSTM-RNN-based residual is faster than the EKF-based counterpart in detecting and clearing the voltage sensor fault.

References

[1] L. Lu, X. Han, J. Li, J. Hua, and M. Ouyang, "A review on the key issues for lithium-ion battery management in electric vehicles," *Journal of Power Sources*, vol. 226, pp. 272–288, Mar. 15, 2013.

[2] H. Rahimi-Eichi, U. Ojha, F. Baronti, and M.-Y. Chow, "Battery management system: An overview of its application in the smart grid and electric vehicles," *IEEE Industrial Electronics Magazine*, vol. 7, no. 2, pp. 4–16, Jun. 2013.

[3] R. Xiong, Q. Yu, W. Shen, C. Lin, and F. Sun, "A sensor fault diagnosis method for a lithium-ion battery pack in electric vehicles," *IEEE Transactions on Power Electronics*, vol. 34, no. 10, pp. 9709–9718, Oct. 2019.

[4] L. Ling and Y. Wei, "State-of-charge and state-of-health estimation for lithium-ion batteries based on dual fractional-order extended kalman filter and online parameter identification," *IEEE Access*, vol. 9, pp. 47 588–47 602, 2021.

[5] A. Sardashti and A. Ramezani, "Fault tolerant control of islanded AC microgrids under sensor and communication link faults using online recursive reduced-order estimation," *International Journal of Electrical Power & Energy Systems*, vol. 126, p. 106 578, Mar. 1, 2021.

[6] N. M. Dehkordi and S. Z. Moussavi, "Distributed resilient adaptive control of islanded microgrids under sensor/actuator faults," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2699–2708, May 2020, Conference Name: IEEE Transactions on Smart Grid.

[7] T. Mahmood, J. Li, Y. Pei, *et al.*, "An intelligent fault detection approach based on reinforcement learning system in wireless sensor network," *The Journal of Supercomputing*, Aug. 3, 2021.

[8] M.-K. Tran and M. Fowler, "Sensor fault detection and isolation for degrading lithium-ion batteries in electric vehicles using parameter estimation with recursive least squares," *Batteries*, vol. 6, no. 1, p. 1, Mar. 2020.

[9] Z. Liu, Q. Ahmed, J. Zhang, G. Rizzoni, and H. He, "Structural analysis based sensors fault detection and isolation of cylindrical lithium-ion batteries in automotive applications," *Control Engineering Practice*, vol. 52, pp. 46–58, Jul. 1, 2016.

[10] J. Hu, X. Bian, Z. Wei, J. Li, and H. He, "Residual statistics-based current sensor fault diagnosis for smart battery management," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 10, no. 2, pp. 2435–2444, Apr. 2022.

[11] K. Zhang, X. Hu, Y. Liu, X. Lin, and W. Liu, "Multi-fault detection and isolation for lithium-ion battery systems," *IEEE Transactions on Power Electronics*, vol. 37, no. 1, pp. 971–989, Jan. 2022.

[12] T. Lin, Z. Chen, C. Zheng, D. Huang, and S. Zhou, "Fault diagnosis of lithium-ion battery pack based on hybrid system and dual extended kalman filter algorithm," *IEEE Transactions on Transportation Electrification*, vol. 7, no. 1, pp. 26–36, Mar. 2021.

[13] D. Li, Z. Zhang, P. Liu, Z. Wang, and L. Zhang, "Battery fault diagnosis for electric vehicles based on voltage abnormality by combining the long short-term memory neural network and the equivalent circuit model," *IEEE Transactions on Power Electronics*, vol. 36, no. 2, pp. 1303–1315, Feb. 2021.

[14] O. Ojo, H. Lang, Y. Kim, X. Hu, B. Mu, and X. Lin, "A neural network based method for thermal fault detection in lithium-ion batteries," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 5, pp. 4068–4078, May 2021.

[15] J. Hong, Z. Wang, and Y. Yao, "Fault prognosis of battery system based on accurate voltage abnormity prognosis using long short-term memory neural networks," *Applied Energy*, vol. 251, p. 113 381, Oct. 1, 2019.

[16] R. Xiong, Q. Yu, L. Y. Wang, and C. Lin, "A novel method to obtain the open circuit voltage for the state of charge of lithium ion batteries in electric vehicles by using h infinity filter," *Applied Energy*, Transformative Innovations for a Sustainable Future – Part II, vol. 207, pp. 346–353, Dec. 1, 2017.

[17] P. Kollmeyer, "Panasonic 18650pf li-ion battery data," vol. 1, Jun. 21, 2018, Publisher: Mendeley Data.

[18] E. Chemali, P. J. Kollmeyer, M. Preindl, R. Ahmed, and A. Emadi, "Long short-term memory networks for accurate state-of-charge estimation of li-ion batteries," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 8, pp. 6730–6739, Aug. 2018, Conference Name: IEEE Transactions on Industrial Electronics.

[19] Y. Cai, "Condition monitoring and fault detection for marine battery energy storage systems," 2021.

[20] G. L. Plett, *Battery Management Systems: Volume 1, Battery Modeling*. Boston : London: Artech House Publishers, Sep. 30, 2015, 342 pp.

[21] Gregory L. Plett, *Battery Management Systems, Volume II: Equivalent-Circuit Methods*, 2nd edition. Boston: Artech House, Apr. 14, 2020, 328 pp.

[22] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*, Third. Springer, 2016.

Appendix A
The GLR algorithm

A. Proof of $\mathrm{g_M(k)}$ for the change of mean in a Gaussian sequence

Given: a Gaussian sequence $\mathrm{r(k)}$ with independent and identically distributed increments.
Prove: the decision function $\mathrm{g_M(k)}$ takes the form:

$$\mathrm{g_M(k)} = \frac{1}{2\sigma^2\mathrm{M}}\left[\sum_{\mathrm{i=k-M+1}}^{\mathrm{k}} (\mathrm{r(i)} - \mu_0)\right]^2$$

Proof:

For a Gaussian sequence, the cumulative sum of log-likelihood ratios is written as:

$$\mathrm{S_k}(\mu_1) = \sum_{\mathrm{i=j}}^{\mathrm{k}} \ln\frac{\mathrm{p}_{\theta_1}(\mathrm{r(i)})}{\mathrm{p}_{\theta_0}(\mathrm{r(i)})} = \frac{\mu_1 - \mu_0}{\sigma^2}\sum_{\mathrm{i=k-M+1}}^{\mathrm{k}}\left(\mathrm{r(i)} - \frac{\mu_1 + \mu_0}{2}\right) \tag{23}$$

To maximize $\mathrm{S_k}(\mu_1)$ with respect to μ_1, we need to find $\frac{\partial \mathrm{S_k}(\mu_1)}{\partial \mu_1} = 0$.

$$\begin{aligned}
\frac{\partial \mathrm{S_k}(\mu_1)}{\partial \mu_1} &= \frac{1}{\sigma^2}\sum_{\mathrm{i=k-M+1}}^{\mathrm{k}}\left(\mathrm{r(i)} - \frac{\mu_1+\mu_0}{2}\right) - \frac{\mu_1-\mu_0}{\sigma^2}\frac{\mathrm{M}}{2} = 0\\
\Rightarrow \sum_{\mathrm{i=k-M+1}}^{\mathrm{k}} \mathrm{r(i)} &- \sum_{\mathrm{i=k-M+1}}^{\mathrm{k}}\frac{\hat{\mu}_1+\mu_0}{2} - (\hat{\mu}_1-\mu_0)\frac{\mathrm{M}}{2} = 0\\
\Rightarrow \sum_{\mathrm{i=k-M+1}}^{\mathrm{k}} \mathrm{r(i)} &- (\hat{\mu}_1+\mu_0)\frac{\mathrm{M}}{2} - (\hat{\mu}_1-\mu_0)\frac{\mathrm{M}}{2} = 0\\
\Rightarrow \hat{\mu}_1 &= \frac{1}{\mathrm{M}}\sum_{\mathrm{i=k-M+1}}^{\mathrm{k}} \mathrm{r(i)}
\end{aligned}$$

Substitute $\hat{\mu}_1$ into 23, and yield

$$\mathrm{S_k}(\hat{\mu}_1) = \frac{1}{\sigma^2}\left(\frac{1}{\mathrm{M}}\sum_{\mathrm{i=k-M+1}}^{\mathrm{k}}\mathrm{r(i)} - \mu_0\right)\sum_{\mathrm{i=k-M+1}}^{\mathrm{k}}\left(\mathrm{r(i)} - \frac{1}{2}\left(\frac{1}{\mathrm{M}}\sum_{\mathrm{i=k-M+1}}^{\mathrm{k}}\mathrm{r(i)} + \mu_0\right)\right)$$

Note that

$$\sum_{\mathrm{i=k-M+1}}^{\mathrm{k}}\frac{1}{\mathrm{M}}\sum_{\mathrm{i=k-M+1}}^{\mathrm{k}}\mathrm{r(i)} = \sum_{\mathrm{i=k-M}}^{\mathrm{k}}\mathrm{r(i)},$$
$$\frac{1}{\mathrm{M}}\sum_{\mathrm{i=k-M+1}}^{\mathrm{k}}\mu_0 = \mu_0$$

Thus we have

$$\max_{\mu_1}\mathrm{S_k}(\mu_1) = \mathrm{S_k}(\hat{\mu}_1) = \frac{1}{2\sigma^2\mathrm{M}}\left(\sum_{\mathrm{k-M+1}}^{\mathrm{k}}(\mathrm{r(i)} - \mu_0)^2\right) \tag{24}$$

$$\mathrm{g_M(k)} = \max_{\mu_1}\mathrm{S_k}(\mu_1) = \mathrm{g_M(k)} = \frac{1}{2\sigma^2\mathrm{M}}\left[\sum_{\mathrm{i=k-M+1}}^{\mathrm{k}}(\mathrm{r(i)} - \mu_0)\right]^2 \tag{25}$$

Appendix B
GLR design methodology

The objective is to design the threshold h and window length M to meet a given probability $\mathrm{P_F}$ of false alarm, and a probability $\mathrm{P_D}$ of detection of a change in mean from μ_0 to μ_1.

Denote the log-likelihood taken over a window length M by $\mathrm{S_M}$,

$$\begin{aligned}\mathrm{S_M(k)} &= \sum_{\mathrm{i=k-M+1}}^{\mathrm{k}} \ln \frac{\mathrm{p}_{\theta_1}(\mathrm{r(i)})}{\mathrm{p}_{\theta_0}(\mathrm{r(i)})} \\ &= \frac{1}{\sqrt{\mathrm{M}}\sigma}\left[\sum_{\mathrm{i=k-M+1}}^{\mathrm{k}} (\mathrm{r(i)} - \mu_0)\right]\end{aligned}$$

Note that $\mathrm{S_M(k)} = 2\sqrt{\mathrm{g(k)}}$, and it has the following probability:

$$\begin{aligned}\mathrm{p}\left(\mathrm{S_M(k)}\right) &= \mathcal{N}(0,1) \text{ under } \mathcal{H}_0 \\ \mathrm{p}\left(\mathrm{S_M(k)}\right) &= \mathcal{N}\left(\frac{\sqrt{\mathrm{M}}\left(\mu_1 - \mu_0\right)}{\sigma}, 1\right) \text{ under } \mathcal{H}_1\end{aligned}$$

The probability law for $\mathrm{g_M(k)}$ is given by

$$\mathrm{p}\left(2\mathrm{g_M(k)}\right) = \chi_1^2 \text{ under } \mathcal{H}_0 \tag{26}$$

$$\mathrm{p}\left(2\mathrm{g_M(k)}\right) = \chi_1^2\left(\frac{\mathrm{M}\left(\mu_1 - \mu_0\right)^2}{\sigma^2}\right) \text{ under } \mathcal{H}_1 \tag{27}$$

where $\chi_1^2(\mathrm{x})$ denotes the chi-square distribution with one degree of freedom with the non-centrality parameter x.

To enforce the given probabilities of false and correct detection, they are given as:

$$\mathrm{P_F} = \mathrm{P}\left(\mathrm{g} > \mathrm{h} \mid \mathcal{H}_0\right) = \int_{\mathrm{h}}^{\infty} \mathrm{p}\left(\mathrm{g} \mid \mathcal{H}_0\right) \mathrm{dg} \tag{28}$$

$$\mathrm{P_D} = \mathrm{P}\left(\mathrm{g} \geq \mathrm{h} \mid \mathcal{H}_1\right) = \int_{\mathrm{h}}^{\infty} \mathrm{p}\left(\mathrm{g} \mid \mathcal{H}_1\right) \mathrm{dg} \tag{29}$$

where $\mathrm{p(2g} \mid \mathcal{H}_0)$ and $\mathrm{p(2g} \mid \mathcal{H}_1)$ denote the probability density function of the test function, 2g, conditioned on $\mathcal{H}_0$ and $\mathcal{H}_1$, respectively. Accounting to the previously determined probability density functions 26 and 27, yield:

$$\int_{\mathrm{2h}}^{\infty} \mathrm{p}_{\chi_1^2}(\mathrm{X})\mathrm{dX} = \alpha \tag{30}$$

$$\int_{\mathrm{2h}}^{\infty} \mathrm{p}_{\chi_1^2}\left(\mathrm{X}; \frac{\mathrm{M}\left(\mu_1 - \mu_0\right)^2}{\sigma^2}\right) \mathrm{dX} = \beta \tag{31}$$

If data are available for both $\mathcal{H}_0$ and $\mathcal{H}_1$ cases, the cumulative density functions $\mathrm{F(g_M} \mid \mathcal{H}_0; \mathrm{M}$ and $\mathrm{F(g_M} \mid \mathcal{H}_1; \mathrm{M}$ can be estimated from these data for different values of the window length M. From these cumulative density functions, the threshold h and window length M can be determined to achieve a required probability of false alarm α and a required probability of correct detection β. They can be expressed as:

$$\alpha = 1 - \mathrm{F}\left(\mathrm{h} \mid \mathcal{H}_0; \mathrm{M}\right) \Rightarrow \mathrm{h} = \mathrm{F}^{-1}(1-\alpha; \mathrm{M}) \tag{32}$$

$$\beta = 1 - \mathrm{F}\left(\mathrm{h} \mid \mathcal{H}_1; \mu_1, \mathrm{M}\right) \Rightarrow \mathrm{M} = \mathrm{F}^{-1}\left(1-\beta; \mu_1, \mathrm{h}\right) \tag{33}$$

Hence it is possible to determine a window size that provides a desired probability of detection.