IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 October 2023, accepted 15 November 2023, date of publication 21 November 2023,
date of current version 29 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3335598

RESEARCH ARTICLE

A Distribution Network State Estimation Method With Non-Gaussian Noise Based on Parallel Particle Filter

HAOTIAN MA, (Student Member, IEEE), WANXING SHENG, (Fellow, IEEE), AND KEYAN LIU, (Member, IEEE)
China Electric Power Research Institute Company Ltd., Beijing 100192, China

Corresponding author: Haotian Ma (t78945685153@163.com)

This work was supported in part by the State Grid Corporation Science and Technology Project (5400-202255154A-1-1-ZN, Key Technologies and Application of Digital Twin for Distribution Networks with High DG Penetration).

ABSTRACT The particle filter (PF) algorithm is a powerful method for tackling non-Gaussian noise interference in distribution network state measurement. However, this algorithm suffers from slow solving speed and lengthy calculation time. To overcome this, a state estimation method based on parallel particle filter (PPF) is proposed, which leverages the independent computation features of each particle in the PF model to improve computational efficiency. This study utilizes the parallel architecture of Compute Unified Device Architecture (CUDA) and General Purpose Graphics Processing Units (GPGPU) to establish a one-to-one correspondence between particles and computing threads. An improved rejecting-resampling method is introduced to solve the problem of low execution efficiency caused by unmerged access to GPGPU memory. In addition, according to the relationship between the particle number and estimation accuracy of state variable of the PPF, the optimal particle number suitable for parallel computation is solved. Ultimately, the simulation results indicate that the proposed method can be used to effectively filter the non-Gaussian-colored noises from the collected data, which meets the requirements of the distribution network state estimation for the accuracy and real-time performance.

INDEX TERMS Distribution networks, state estimation, parallel particle filter, non-Gaussian-colored noise.

I. INTRODUCTION

As the prevalence of distributed generation, electric vehicles, and other innovative equipment in distribution networks continues to grow, so too does the variability of their operational status, which exhibits pronounced spatio-temporal characteristics [1], [2], [3], [4]. The primary objective of distribution network state estimation is to ascertain the operational state of a distribution network in a manner that is both prompt and precise. However, the complex temporal and spatial peculiarities mean that secondary equipment in the distribution network and data acquisition devices can be subjected to fixed direction and feature interferences at different periods and regions [5], [6], [7], [8], [9]. The resulting interference in the collected data can be symbolized by a class of non-Gaussian-colored noises [10], [11], [12], [13], thus leading to inaccuracies in the outcomes of distribution network state estimation.

Traditional distribution network state estimation models typically handle Gaussian noise present in measurement data. Common methodologies in this regard encompass least square state estimation (LS-SE), Kalman filter state estimation (KF-SE), and particle filter state estimation (PF-SE) [14], [15], [16], [17], [18], [19]. Nonetheless, when these models are utilized for managing non-Gaussian noise, they exhibit reduced filtering accuracy and diminished computational efficiency.

Reference [20] introduces a Phasor-Assisted State Estimation (PASE) methodology that requires fewer phasor measurement units (PMUs) to achieve a specified estimation

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

error. However, a notable shortcoming of this methodology is its failure to address the impact of non-Gaussian-colored noise. In Reference [21], a distributed algorithm is proposed which employs a diffusion extended Kalman filter for the real-time estimation of power system oscillation parameters. Nevertheless, this methodology reveals inadequacies when applied to non-Gaussian measurements over extended time scales. Reference [22] formulates a robust Gaussian Mixture Unscented Kalman Filter (GM-UKF) and a compact batch-mode regression form to identify inaccurate values. However, this study does not tackle the non-linear measurement problem. Reference [23] proposes a state estimation methodology for non-linear measurement, which constructs a model utilizing the Taylor series of voltage, and subsequently applies the interval algorithm to solve the model. However, the derived results are solely applicable to the eradication of random errors. Reference [24] characterizes robust state estimation based on the PMU as a quadratic programming problem, addressing it in a decentralized fashion to accommodate autonomous operation modes among microgrids. Yet, this approach may not be apt for intricate distribution networks operating autonomously. Reference [25] presents a method rooted in stochastic matrix theory for data-driven matrix level measurement error cleansing. This approach employs WLS to construct a two-tiered state estimation scheme. However, the limitation of this method lies in its requirement for extensive data, rendering it applicable only to specific categories of noise. In [26], the particle filter method is harnessed for the dynamic estimation of the state of synchronous generators, taking into account the excitation and prime mover control systems. However, the constraint of this approach is its struggle to satisfy real-time demands.

Among the aforementioned methods, the LS-SE demonstrates commendable estimation outcomes when dealing with white noise, yet its filtering accuracy falters when confronted with colored noise. The KF-SE, grounded in the Gaussian model, encounters difficulties in addressing non-Gaussian noise issues. While the PF-SE exhibits versatility in handling any type of noise, it is disadvantaged by its relative computational sluggishness. Therefore, many scholars in other research fields have proposed state prediction or estimation models through algorithm fusion [27], [28], [29], [30]. Among these, exploring and utilizing the parallelism of algorithms is one of the effective methods to significantly enhance computational efficiency.

In recent years, advancements in chip technology have ushered in the application of General-Purpose Graphics Processing Units (GPGPU) in the realm of general computing [31]. The integration of a substantial number of computing units within the GPGPU makes it particularly well-suited to parallel computing technology. Reference [32] proposes a hierarchical fast parallel co-evolutionary immune particle swarm optimization algorithm based on GPGPU technology, and applied the algorithm to multiple parameters identification and temperature monitoring of permanent-magnet synchronous motor. In [33], the hybrid parallel-in-time-and-space (PiT+PiS) transient simulation on the CPU-GPU platform is presented to thoroughly exploit the parallelism from time and spatial perspectives. Reference [34] describes a time-domain power quality state estimation (PQSE) evaluation method for power system based on Kalman filter, which is implemented by parallel processing technology through GPGPU to reduce execution time. In [35], a fine-grained network decomposition method is proposed for large-scale electromagnetic transient simulation based on the parallel architecture of GPGPU. Leveraging the inherent parallel characteristics of the particle filter algorithm, the PPF algorithm can be developed utilizing parallel computing technology. This facilitates a rapid solution for distribution network state estimation in the presence of non-Gaussian-colored noise.

To sum up, this study presents the PPF algorithm based on GPGPU to the field of power distribution network state estimation, thereby establishing a parallel state estimation model. By thoroughly exploiting the particle parallelism in the Particle Filter, the model effectively accelerates state estimation in scenarios characterized by non-Gaussian-colored noises [36].

The rest of this paper is organized as follows: Section II sets forth a fundamental mathematical model for distribution network state estimation, with a focus on non-Gaussian-colored noises. Section III delves into the calculation process of the PPF and introduces a reject-resampling technique tailored for GPGPU parallel memory processing. Section IV outlines a method for determining the optimal particle number using PPF, accompanied by a detailed overview of the basic calculation process of PPF-SE. In Section V, case studies and simulation results are discussed, including a performance comparison, and ends with the conclusion in Section VI.

II. BASIC MODEL FOR DISTRIBUTION NETWORK STATE ESTIMATION

The data acquisition devices in the distribution network are vulnerable to external influences, which can result in signal distortion during the measurement of electrical variables. To mitigate this effect, it becomes imperative to estimate the current operational state of the distribution network by leveraging measurement information and system state models. This section introduces the non-Gaussian noise model and the distribution pattern of colored noise, and establishes a state estimation model for distribution network state estimation based on these noise characteristics. This model encompasses both the state equation and the measurement equation.

The basic model of state estimation can be expressed as

$$\begin{cases} X_k = f\left(X_{k-1}, \omega_k\right) \\ Z_k = h\left(X_k, v_k\right) \end{cases} \tag{1}$$

where X_k and Z_k are system state variables and measurement variables respectively. $f(.)$ and $h(.)$ are state transition equation and measurement equation respectively. ω_k is process noise and v_k is measurement noise.

The process noise ω_k signifies the disparity between the actual operational state and the ideal state, constituting an intrinsic aspect of the distribution network's operational characteristics that does not necessitate elimination. Conversely, the measurement noise v_k denotes the errors engendered by the information system during the acquisition and conversion of electrical quantities, impeding the accurate observation of the genuine state of the distribution network. Thus, the elimination of this form of noise becomes imperative.

In this research, we utilize non-Gaussian noise and colored noise to depict the stochastic properties that arise during the measurement process, as elucidated in the following description.

A. NON-GAUSSIAN NOISE AND COLORED NOISE

1) NON-GAUSSIAN NOISE

Laplace noise is a typical non-Gaussian noise, and its probability density function (PDF) is

$$p(x) = \frac{1}{\sqrt{2\sigma^2}} \exp\left(-\sqrt{\frac{2}{\sigma^2}}\left|x^2\right|\right) \tag{2}$$

where, σ^2 is the variance or power of noise. The Laplacian distribution can also be expressed in another form:

$$p(x|\mu, b) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right) \tag{3}$$

where b is the scale parameter, μ is the position parameter. Laplacian PDF has significant peaks and trails compared to Gauss PDF [37].

2) COLORED NOISE

Generally, the first order autocorrelation time series is used to generate colored noise, which can be expressed as

$$\Phi_t = \lambda \Phi_{t-1} + \gamma \varepsilon_t \tag{4}$$

where, Φ_t represents the t-th term of the autocorrelation sequence; λ is the autocorrelation coefficient; ε_t is the extended random component; γ is the random component correlation coefficient [38], [39].

When $-1 < \lambda < 0$, a blue noise sequence is generated, and when $0 < \lambda < 1$, a red noise sequence is generated. The colored noise generated by (4) has two important parameters: mean $E(\Phi)$ and variance $Var(\Phi)$. If $\Phi_0 = 0$, then the mean value $E(\Phi) = 0$, the variance $Var(\Phi)$ can be expressed as

$$Var(\Phi) = \frac{\gamma^2\left(T - \frac{(1+\lambda)^2}{1-\lambda^2} + \frac{2\lambda(1-\lambda)^T}{T(1-\lambda)^2}\right)}{(1-\lambda^2)(T-1)} \tag{5}$$

where T represents the length of the time series. $Var(\Phi)$ is related to both the parameters λ and γ.

B. STATE VARIABLES AND STATE EQUATIONS

1) STATE VARIABLES

In the context of distribution network state estimation, state variables refer to fundamental electrical parameters that can be employed to describe the operational state of the distribution network. In this research study, the state variables chosen are node voltage and phase angle. These two electrical variables are fundamental in the power system. When these variables are known, the current and power of any part of the system can be indirectly calculated, thus achieving the goal of fully understanding the state of the distribution network. It is important to note that analyzing the distribution network solely by selecting one phase as the transmission network is inadequate, considering the inherent operational characteristics of three-phase asymmetry. Thus, it becomes imperative to establish a three-phase asymmetry model specifically tailored for the distribution network. Typically, voltage amplitude and phase angle are selected as state variables, which can be expressed as

$$\begin{aligned} X &= \left[U_1^p, U_2^p, \cdots U_n^p, \theta_1^p, \theta_2^p, \cdots \theta_n^p\right] \\ U_i^p &= \left[U_i^a, U_i^b, U_i^c\right] \\ \theta_i^p &= \left[\theta_i^a, \theta_i^b, \theta_i^c\right] \end{aligned} \tag{6}$$

where, U_i^p represents node voltage amplitude state, θ_i^p represents the voltage phase angle state, i represents the i -th node within the network, and p represents the three phases of a, b, and c.

2) STATE EQUATIONS

State equation is established based on the changing rules of the system state variables, and is used to describe the transition relationship between the current state and the previous state. Ignoring the transient process generated by violent disturbance, the state changes during distribution network operation are described based on a quasi-steady-state model, which can be expressed as

$$X_k = f(X_{k-1}) + \omega_k \tag{7}$$

where X_k represents the state variable of the distribution network at time k.

Holt two-parameter exponential smoothing method is used to describe the dynamic model of state equation. Assume that the one-step predicted value of the system state at time k is $\tilde{X}_k$ and the estimated value is $\hat{X}_k$, then the predicted value at time $k+1$ referred to this method is

$$\begin{aligned} \tilde{X}_{k+1} &= s_k + b_k + \omega_k \\ s_k &= \alpha \hat{X}_k + (1-\alpha)\tilde{X}_k \\ b_k &= \beta(s_k - s_{k-1}) + (1-\beta) b_{k-1} \\ \omega_k &\sim N(0,1) \end{aligned} \tag{8}$$

where, s_k and b_k are intermediate variables, α and β are smooth parameters, $\alpha, \beta \in [0,1]$.

C. MEASUREMENT VARIABLES AND MEASUREMENT EQUATIONS

1) MEASUREMENT VARIABLES

The measured variable pertain to the electrical variables that can be collected during the operation of a distribution system.

According to the measuring device installed in the distribution network, the measurement collected mainly includes voltage amplitude, voltage phase Angle, node injected power, branch power flow and branch current, i.e

$$\begin{aligned}
Z &= \left[U_{ku}^{p}, \theta_{k\theta}^{p}, P_{kr}^{p}, Q_{kr}^{p}, P_{kl}^{p}, Q_{kl}^{p}, I_{kl}^{p}\right] \\
U_{ku}^{p} &= \left[U_{ku1}^{p}, U_{ku2}^{p}, \cdots, U_{kumu}^{p}\right],\ U_{kui}^{p} = \left[U_{kui}^{a}, U_{kui}^{b}, U_{kui}^{c}\right] \\
\theta_{k\theta}^{p} &= \left[\theta_{k\theta1}^{p}, \theta_{k\theta2}^{p}, \cdots, \theta_{k\theta m\theta}^{p}\right],\ \theta_{k\theta i}^{p} = \left[\theta_{k\theta i}^{a}, \theta_{k\theta i}^{b}, \theta_{k\theta i}^{c}\right] \\
P_{kr}^{p} &= \left[P_{kr1}^{p}, P_{kr2}^{p}, \cdots, P_{krmr}^{p}\right],\ P_{kri}^{p} = \left[P_{kri}^{a}, P_{kri}^{b}, P_{kri}^{c}\right] \\
Q_{kr}^{p} &= \left[Q_{kr1}^{p}, Q_{kr2}^{p}, \cdots, Q_{krmr}^{p}\right],\ Q_{kri}^{p} = \left[Q_{kri}^{a}, Q_{kri}^{b}, Q_{kri}^{c}\right] \\
P_{kl}^{p} &= \left[P_{kl1}^{p}, P_{kl2}^{p}, \cdots, P_{klml}^{p}\right], \\
P_{kli}^{p} &= \left[P_{kli}^{a}, P_{kli}^{b}, P_{kli}^{c}\right], li\,(li_h, li_e) \\
Q_{kl}^{p} &= \left[Q_{kl1}^{p}, Q_{kl2}^{p}, \cdots, Q_{klml}^{p}\right], \\
Q_{kli}^{p} &= \left[Q_{kli}^{a}, Q_{kli}^{b}, Q_{kli}^{c}\right], li\,(li_h, li_e) \\
I_{kl}^{p} &= \left[I_{kl1}^{p}, I_{kl2}^{p}, \cdots, I_{klml}^{p}\right], \\
I_{kli}^{p} &= \left[I_{kli}^{a}, I_{kli}^{b}, I_{kli}^{c}\right], li\,(li_h, li_e)
\end{aligned} \tag{9}$$

where, U_{ku}^{p} and $\theta_{k\theta}^{p}$ are the measurement of node voltage amplitude and phase angle; P_{kr}^{p} and Q_{kr}^{p} represent the measurement of injected active power and reactive power; P_{kl}^{p} and Q_{kl}^{p} represent the measurement of active and reactive power flowing through the branch; I_{kl}^{p} is the measurement of the current flowing through the branch; $li\,(li_h, li_e)$ indicates that the power flow direction of the branch li is li_h to li_e.

2) MEASUREMENT EQUATIONS

The measurement equation denotes the methodology of representing measured variables in relation to state variables, thereby elucidating the mapping approach from state variables to measured variables. According to the state variables and measurement variables of distribution network, the measurement equation can be established as

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix} = \begin{bmatrix} h_1\left(x_1, x_2, \cdots, x_n\right) \\ h_2\left(x_1, x_2, \cdots, x_n\right) \\ \vdots \\ h_m\left(x_1, x_2, \cdots, x_n\right) \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} \tag{10}$$

The association between the measured variables and state variables typically manifests as either a linear or nonlinear function. Specifically, the measurement of voltage and phase angle demonstrates a linear relationship with their respective state variables, while the measurement of active power, reactive power, and current exhibits a nonlinear relationship with the state variables. The measurement equations for voltage and phase angle are as follows:

$$\begin{aligned}
\widehat{U}_i^p &= U_i^p + v_{ui} \\
\widehat{\theta}_i^p &= \theta_i^p + v_{\theta i}
\end{aligned} \tag{11}$$

where, $\widehat{U}_i^p$ represents the voltage measurement variable of p-phase at node i; $\widehat{\theta}_i^p$ represents the phase angle measurement variable of p-phase at node i.

For active power $\widehat{P}_i^p$ and reactive power $\widehat{Q}_i^p$ injected in p phase of node i can be calculated using voltage magnitude, phase angle, and node admittance matrix. The specific measurement equations for active power and reactive power are as follows:

$$\begin{aligned}
\widehat{P}_i^p &= \sum_{j=1}^{J}\sum_{q=a}^{c} U_i^p U_j^q \left(G_{ij}^{pq}\cos\theta_{ij}^{pq} + B_{ij}^{pq}\sin\theta_{ij}^{pq}\right) + v_{Pi} \\
\widehat{Q}_i^p &= \sum_{j=1}^{J}\sum_{q=a}^{c} U_i^p U_j^q \left(G_{ij}^{pq}\sin\theta_{ij}^{pq} - B_{ij}^{pq}\cos\theta_{ij}^{pq}\right) + v_{Qi}
\end{aligned} \tag{12}$$

where, G_{ij}^{pq} and B_{ij}^{pq} are the real and imaginary parts of mutual admittance between p-phase of node i and q-phase of node j, respectively.

For the p-phase branch current flowing from node i to node j can be calculated using voltage phasors and the equivalent admittance of the circuit. Its measurement equation can be expressed as follows:

$$\widehat{I}_{ij}^p = \left| y_{i0}^p \dot{U}_i^p + \sum_{q=a}^{c} y_{ij}^{pq}\left(\dot{U}_i^p - \dot{U}_j^q\right)\right| + v_{Iij} \tag{13}$$

where, $\dot{U}_i^p$ represents the p-phase voltage phasor of node i, which can be derived by combining the amplitude and phase angle; y_{i0}^p is the p-phase ground branch admittance of node i; y_{ij}^{pq} is the branch admittance between p-phase of node i and q-phase of node j.

For the p-phase branch active power $\widehat{P}_{ij}^p$ and reactive power $\widehat{Q}_{ij}^p$ flowing from node i to node j can be determined through the utilization of voltage phasors and current phasors, yielding the following expressions:

$$\begin{aligned}
\tilde{S}_{ij}^p &= \dot{U}_i^p \dot{I}_{ij}^{p*} = \dot{U}_i^p \left(y_{i0}^p \dot{U}_i^p + \sum_{q=a}^{c} y_{ij}^{pq}\left(\dot{U}_i^p - \dot{U}_j^q\right)\right)^* \\
&= P_{ij}^p + jQ_{ij}^p \\
\widehat{P}_{ij}^p &= \mathrm{Re}\left(\tilde{S}_{ij}^p\right) \\
&= \mathrm{Re}\left(y_{i0}^{p*} U_i^{p2} + \sum_{q=a}^{c} y_{ij}^{pq*}\left(U_i^{p2} - \dot{U}_i^p \dot{U}_j^{q*}\right)\right) + v_{Pij} \\
\widehat{Q}_{ij}^p &= \mathrm{Im}\left(\tilde{S}_{ij}^p\right) \\
&= \mathrm{Im}\left(y_{i0}^{p*} U_i^{p2} + \sum_{q=a}^{c} y_{ij}^{pq*}\left(U_i^{p2} - \dot{U}_i^p \dot{U}_j^{q*}\right)\right) + v_{Qij}
\end{aligned} \tag{14}$$

where, $\tilde{S}_{ij}^p$ is the complex power of the branch, and ()* denotes the conjugate.

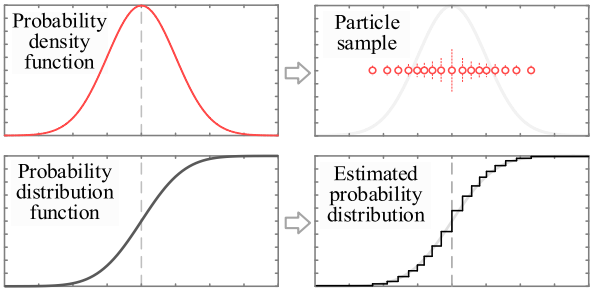

FIGURE 1. The basic concept of particle equivalence in PF.

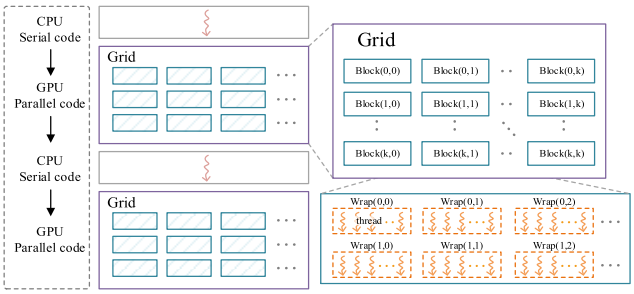

FIGURE 2. GPGPU multi-threads parallel computing mode.

III. PARALLEL PARTICLE FILTER ALGORITHM

A. PARTICLE FILTERING AND PARALLEL COMPUTING

Recursive Bayesian estimation is implemented via PF based on Monte Carlo simulation, which is suitable for any nonlinear system that can be described based on a state space model. This method is a computational model in which random numbers are used to solve mathematical or physical problems, whose idea is to approximate the posterior probability distribution function of desired variables using a large number of random sample points in a state space, as is shown in Fig. 1.

In Fig. 1, the circles represent discrete sampling points, also known as particles. Particles distributed according to a particular law can represent a certain probability distribution, thus indirectly transforming an integral problem into a summation problem of finite sample points.

For a PPF, the characteristics of independent operation among particles are taken into account, so as to combine the particle filtering algorithm with GPGPU, through which the solving speed of conventional particle filtering can be improved effectively.

CUDA is a unified computing device architecture geared towards General-Purpose computing on GPGPU parallel programming. It adopts a fine-grained parallel model based on the Single Instruction Multiple Thread (SIMT) paradigm. The fundamental programming model of CUDA is depicted in Fig.2.

Through CUDA, parallel computing tasks are executed by calling kernel functions, and threads are organized into thread blocks as well as thread grids to facilitate management. Threads are scheduled to execute through a thread bundle (warp), which consists of 32 consecutive threads, and the threads in each warp execute synchronously.

Each thread can access data from multiple memory spaces on the GPGPU. Memory is accessed in 32, 64 or 128 bytes and aligned against 32 bytes. When all threads in a warp access contiguous and aligned memory spaces, all access can be combined into a single memory read. If the memory addresses are not in the same memory segment, the access cannot be consolidated, which results in memory access performance degradation.

B. PARTICLE POSITION UPDATE AND WEIGHT CALCULATION

This section begins by elucidating the operating principle of particle filtering. Following this, it investigates its inherent parallelism and subsequently establishes a method for updating particle position and weight in a parallelizable manner. Assume that N independent and identically distributed particles $\{x_k^1, x_k^2, x_k^3, \cdots, x_k^N\}$ are randomly sampled from the posterior probability density, then the posterior probability density can be approximately expressed as:

$$\hat{p}\left(x_k|z_k\right) = \frac{1}{N}\sum_{i=1}^{N}\delta\left(x_k - x_k^l\right) \tag{15}$$

where, $\delta(\cdot)$ represents the Dirac-delta function. Based on this approximate expression, the conditional expectation of the function $f(x_k)$ can be expressed as

$$\begin{aligned}\hat{E}\left[f\left(x_k\right)\right] &= \int f\left(x_k\right) p\left(x_k|z_k\right) dx_k \\ &\approx \frac{1}{N}\sum_{i=1}^{N}\int f\left(x_k\right)\delta\left(x_k - x_k^l\right) dx_k \\ &= \frac{1}{N}\sum_{i=1}^{N} f\left(x_k^l\right)\end{aligned} \tag{16}$$

In (15) and (16), given that every particle is independent, the function values subsequent to each particle's passage through the nonlinear transfer function can be computed utilizing the principle of parallel computation. This function is dictated by the state transition function, which in turn signifies the prior probability density.

In practical problems, it is often difficult or even impossible to sample directly from the posterior probability density $p(x_k|z_k)$. Therefore, a reference distribution called importance probability density $q(x_k|z_k)$ is introduced, which should be known and easy to sample from.

$$\begin{aligned}E\left[f\left(x_k\right)\right] &= \int f\left(x_k\right)\frac{p\left(x_k|z_k\right)}{q\left(x_k|z_k\right)} q\left(x_k|z_k\right) dx_k \\ &= \int f\left(x_k\right)\frac{p\left(z_k|x_k\right) p\left(x_k\right)}{p\left(z_k\right) q\left(x_k|z_k\right)} q\left(x_k|z_k\right) dx_k \\ &= \int f\left(x_k\right)\frac{w\left(x_k\right)}{p\left(z_k\right)} q\left(x_k|z_k\right) dx_k\end{aligned} \tag{17}$$

where, w represents the weight value, i.e

$$w\left(x_k\right) = \frac{p\left(z_k|x_k\right) p\left(x_k\right)}{q\left(x_k|z_k\right)} \tag{18}$$

Since (17) is integral over x_k, and y_k can be treated as a constant, thus

$$E\left[f\left(x_k\right)\right] = \frac{\int f\left(x_k\right) w\left(x_k\right) q\left(x_k|z_k\right) dx_k}{p\left(z_k\right)} \tag{19}$$

And because the observation probability $p(z_k)$ can be expressed as

$$\begin{aligned} p\left(z_k\right) &= \int p\left(z_k|x_k\right) p\left(x_k\right) dx_k \\ &= \int \frac{p\left(z_k|x_k\right) p\left(x_k\right)}{q\left(x_k|z_k\right)} q\left(x_k|z_k\right) dx_k \\ &= \int w\left(x_k\right) q\left(x_k|z_k\right) dx_k \end{aligned} \tag{20}$$

Substituting (20) into (19):

$$E\left[f\left(x_k\right)\right] = \frac{\int f\left(x_k\right) w\left(x_k\right) q\left(x_k|z_k\right) dx_k}{\int w\left(x_k\right) q\left(x_k|z_k\right) dx_k} \tag{21}$$

To characterize the initial state of a system using Monte Carlo simulation principles, one might extract numerous particles that are independently and uniformly distributed from the importance probability density function. During such computations, each particle aligns with a single CUDA thread. Updates to individual particle positions and weights are then carried out within each respective thread. Essentially, every computational unit within the GPGPU is tasked with executing numerical computations of the nonlinear function, based on the initial values assigned to the particles.

If N independent particles $\{x_k^1, x_k^2, x_k^3, \cdots, x_k^N\}$ are extracted from the importance probability density $q(x_k|z_k)$, Equation (21) can be approximated as:

$$E\left[f\left(x_k\right)\right] \approx \frac{\frac{1}{N}\sum_{i=1}^{N} w\left(x_k^l\right) f\left(x_k^l\right)}{\frac{1}{N}\sum_{i=1}^{N} w\left(x_k^l\right)} = \sum_{i=1}^{N} w_k^l f\left(x_k^l\right) \tag{22}$$

where, w_k^l is defined as the sample particle weight:

$$w_k^l = \frac{w\left(x_k^l\right)}{\sum_{i=1}^{N} w\left(x_k^l\right)} \tag{23}$$

The value range of particle weight after normalization meets: $w_k^l \in [0, 1]$, and $\sum_{i=1}^{N} w_k^l = 1$.

When normalizing the weights of particles, a summation operation is required. Traditional summing operations are typically carried out in a serial manner, necessitating the back-and-forth interaction of data between the CPU storage end and the GPGPU storage end, resulting in a decrease in computational efficiency. In light of this, this study employs the method of parallel reduction when normalizing weights. In parallel reduction summation, data is partitioned into two distinct sets. The equivalent elements in each set are concurrently accumulated within each thread. Following this aggregation, the data is reorganized, and the parallel summation continues by utilizing the threads on the corresponding elements until the ultimate result is achieved. This process effectively optimizes parallel operations. The time complexity of parallel reduction is $O(\log n)$, where n is the number of elements to be reduced. This is significantly more efficient than the $O(n)$ time complexity of a simple serial reduction [40], [41].

In practical application, the prior transition probability density $p(x_k|x_{k-1})$ of the state is typically utilized as the importance probability density, i.e

$$q\left(x_k^l|x_{k-1}^l, z_k\right) = p\left(x_k^l|x_{k-1}^l\right) \tag{24}$$

At this time, particle sampling operations are carried out on CUDA threads based on the prior probability distribution to obtain initial particles. Alternatively, particle positions can be updated based on the probability density function, which is also performed on threads.

Then, compute the weight for each particle based on the nonlinear measurement equation, as follows

$$w_k^l = w_{k-1}^l p\left(z_k|x_k^l\right) \tag{25}$$

The particle weight is proportional to the probability density. Moreover, when normalizing the weights, they need to be calculated according to parallel reduction, which can improve parallel efficiency.

After the computation and normalization of particle weights, the filtered posterior probability density $p(x_k|z_k^l)$ can be approximated as follows

$$p\left(x_k|z_k^l\right) \approx \sum_{i=1}^{N} w_k^l \delta\left(x_k - x_k^l\right) \tag{26}$$

The corresponding particle weights are computed in accordance with (25), and the system state's posterior probability distribution is approximated through the weighted summation of particles. This process facilitates the recursive estimation of state expectation:

$$\hat{x}_k = \sum_{i=1}^{N} w_k^l x_k^l \tag{27}$$

The above weighted summation can still be processed using the CUDA reduction summation method on the GPGPU device end. Subsequently, the computed results are transferred to the CPU host end for further utilization.

Under the PPF, GPGPU threads are allocated in proportion to the number of particles, thus assigning the computation of each particle to a single thread, as illustrated in Fig. 3.

In practical iterative computations, after several iterations of particle propagation, only a handful of particles retain a substantial weight, rendering the weights of the remaining particles negligible. Consequently, the particle set fails to effectively represent the posterior probability distribution of the system state, leading to a decline or divergence in filtering

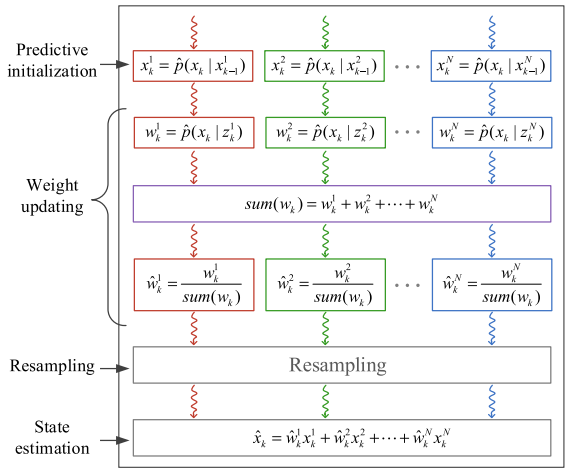

FIGURE 3. Particle parallel execution in PPF.

performance. This phenomenon is referred to as the 'particle degradation problem'.

However, in order to ensure that each particle within the PPF can be executed by an independent thread, it is necessary to enhance the parallelization of the resampling step. This issue is discussed in the following section.

C. PARTICLE RESAMPLING PROCESS

The particle resampling technique can solve the particle degradation problem to a certain extent. Resampling remaps the initial set of particles, that is, the particles with large weights are copied and output many times, while the particles with small weights are discarded. Under the condition of $p(x_k^c = x_k^l) = w_k^l$, a new equal weight supporting particle set $\{(x_k^c, w_k^c = 1/N)\}_{l=1}^N$ is generated, where N is the total number of particles.

During the resampling phase, particles need to be transferred from the device to the host for sequential allocation based on their indices, before being returned to the device. This procedure necessitates two instances of data interchange between the host and the device. In the context of GPGPU-based parallel computations, it is essential to minimize such data exchanges to augment memory access bandwidth. In light of this, this section outlines the fundamental procedure of resampling in particle filtering [42], [43]. Moreover, enhancements are incorporated into the resampling process by proposing a thread-level resampling approach, thereby optimizing it for GPGPU-based parallel computation.

The fundamental procedure for resampling can be outlined as follows: Generate sequences of random numbers uniformly distributed within the range of [0,1): $u_k \sim U[0, 1), k = 1, 2, \cdots, K$, and then compute the cumulative density function as follows

$$c_l = \left\{ \sum_{m=1}^{l} w_k^m \right\}, \quad l = 1, 2, \cdots, N \tag{28}$$

where, w_k^m is the normalized weight corresponding to the particle x_k^l before resampling at time k; According to the above equation, comparing the cumulative density function and random number, if the condition is satisfied, the particle x_k^l will be copied to the new index c as (29), then the particle resampling process can be realized

$$x_k^c = x_k \left(F^{-1}(u_k) \right) = x_k^l \ (c_{l-1} < u_l \leq c_l) \tag{29}$$

In the application, it is necessary to determine whether resampling is performed according to the specific situation. The effective sample size is used to measure the degradation degree of the particle set, and its estimated value is defined as:

$$\hat{N}_{eff} = \left[\sum_{i=1}^{N_{th}} \left(w_k^l \right)^2 \right]^{-1} \tag{30}$$

where, N_{th} is a preset particle number limit, which can be taken as $N_{th} = (2/3)^* N$ generally. Only when $\hat{N}_{eff} < N_{th}$ is satisfied can resampling be started, which can alleviate the problem of sample dilution caused by excessive resampling rapidly to a certain extent.

If the posterior probability density before resampling is (20), then the posterior probability density after resampling $p\left(x_k^c = x_k^l\right) = w_k^l$ can be expressed as:

$$\hat{p}(x_k | z_k) = \sum_{c=1}^{N} \frac{1}{N} \delta\left(x_k - x_k^c\right) = \sum_{l=1}^{N} \frac{N_l}{N} \delta\left(x_k - x_k^l\right) \tag{31}$$

where N_l represents the replication times of particle x_l in particle set $\{x_k^c\}_{c=1}^N$ during the resampling process.

The rejection resampling algorithm is an effective method to make the resampling process independent. It determines whether the particle weights need to be updated by setting certain eligibility criteria. In this way, interaction between particles across threads is unnecessary, thus effectively enhancing parallelism.

Under the assumption that the upper limit of the particle weight, denoted as sup w, is known, new particles can be procured using the rejection-resampling method. If a particle x_k^l satisfies the acceptance criteria, it will be accepted; if not, it is rejected and subsequently resampled until the acceptance criteria are met. By leveraging the rejection-resampling method, we circumvent the need for overall weight manipulation, thereby facilitating easier parallelization. The corresponding pseudocode for this method is provided as follows:

Algorithm 1 Rejecting-Resampling Algorithm

```
[{newx_k^l}_{l=1}^N] = RESAMPLE[{x_k^l, w_k^l}_{l=1}^N]
for l = 1 : N
    c = l
    u ~ U[0, 1]
    while u > w^l / sup w
        c ~ U[1, ..., N]
        u ~ U[0, 1]
    new x_k^l = x_k^c
endfor
```

where, new x is the new particle set after resampling; u is a random number that follows the uniform distribution between 0 and 1; c is a random integer that follows the uniform distribution between 1 and N; c is the index of the selected particle at time k, which is the original sample copied by particle l at time $k+1$.

The rejection-resampling approach can be executed through a singular CUDA kernel function, whereby each thread corresponds to an individual particle. However, when used to randomly select particles, rejection-resampling results in uncoalesced global memory access. For a large particle set, the efficiency of the PPF decreases as the amount of uncoalesced access increases. Given that uncoalesced memory access can be circumvented when threads within the same warp perform read/write operations on the same memory segment, an enhanced rejection-resampling algorithm is proposed. The pseudocode of this improved algorithm is provided as follows:

Algorithm 2 Improved Rejecting-Resampling Algorithm

$[\{newx_k^l\}_{l=1}^N] = \text{RESAMPLE}[\{x_k^l, w_k^l\}_{l=1}^N, Scount, Escount]$
kernel (thread $l = 1 : N$)
{ $c = 1$
  $s \sim U[1, \ldots, Scount]$
  while $u > w^l / \sup w$
    $c \sim U\,[(s\text{-}1)* Ecount+1, \ldots, s*Ecount]$
    $u \sim U\,[0, 1]$
  new $x_k^l = x_k^c$
}

where, s is a fixed continuous extent in the global memory. Let *Esize* be the byte size of a weight and *Ssize* be the byte size of an s section. *Scount* indicates the number of section s, then *Scount*=*Esize*×N/*Ssize*. *Ecount* is the number of weights in section s, then *Ecount*=*Ssize*/*Esize*. j is a random integer that is evenly distributed from the first element to the last element index in segment s; To facilitate thread and memory alignment, the number of particles is set to 2^k.

Let s represent the memory segment of the weight array $\{w_k^l\}_{l=1}^N$. Each warp randomly selects a segment, with the threads within this warp subsequently choosing the weight value in the corresponding segment. Based on the size of the segment, the weights in the global memory can be accessed through one or more coalesced operations, thereby enhancing the parallel efficiency of the algorithm.

IV. DISTRIBUTION NETWORK STATE ESTIMATION ALGORITHM BASED ON PPF

A. OPTIMAL NUMBER OF PPF PARTICLES

For the PPF algorithm, if the whole-process thread-level calculation is to be realized, the number of particles before and after resampling should be kept constant meanwhile the migration between particles and threads should not be performed. Therefore, it should be ensured that the particle number of the PPF is fixed in the filtering process. The number of particles will affect the estimation results of state variables, so it is necessary to analyze the influence of different particle numbers on the filtering effect.

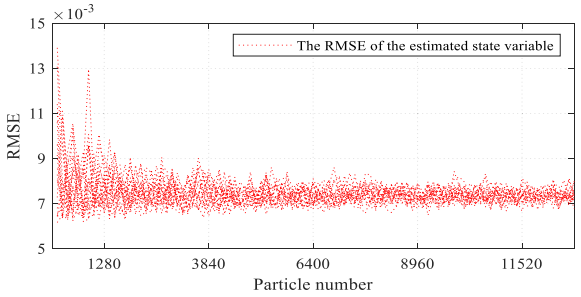

FIGURE 4. Relationship between state RMSE and particle number.

The indexes used to evaluate the performance of deterministic filtering include the root mean square error (RMSE), mean absolute error (MEAE) and maximum absolute error (MAAE). The most commonly used RMSE is selected as the evaluation index to verify the influence of particle number on filtering effect.

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{1}{M}\sum_{k=1}^{M}\left(\hat{X}_k - X_k\right)^2} \\ \text{MEAE} &= \frac{1}{M}\sum_{k=1}^{M}\left|\hat{X}_k - X_k\right| \\ \text{MAAE} &= \max_{k=1\cdots m}\left\{\hat{X}_k - X_k\right\} \end{aligned} \tag{32}$$

In addition, in the CUDA-based parallel framework, the number of threads should preferably be set to a multiple of 32, so the number of particles selected should also be a multiple of 32. Fig. 4 shows the RMSE changes of state variable under different particle numbers by PPF algorithm.

Fig 4 illustrates that as the number of particles increases, there is a corresponding enhancement in the filtering effect. Once the particle count reaches a certain value (in this host and device, it manifests as 6400 particles.), the filtering effect plateaus and becomes consistent, showing no further improvement with additional particles. Thus, when selecting the number of particles in the PPF the count should be no less than this certain value.

B. BASIC FLOW OF PARTICLE FILTERING

Based on the above discussion, the schematic diagram of particle sampling and propagation process of the PPF-SE algorithm are shown in Fig. 5. The figure elucidates the three primary stages of particle state sampling, encapsulating the observation weight updating and resampling processes inherent in the PPF algorithm.

Through parallelization decomposition, the CUDA-based PPF algorithm is realized by taking the idea of one-to-one correspondence between threads and particles. Through the algorithm, the same number of threads as that of particles are

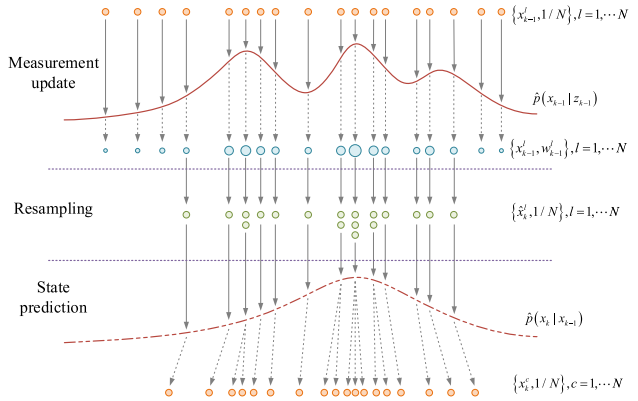

FIGURE 5. Three basic processes of PPF.

created and a corresponding memory space is allocated. Each thread is used to perform the calculation of particle position and weight. In addition, parallel reduction is used for extreme value solving and summation operation, so as to give full play to the parallel computing performance of GPGPU. The algorithm procedure of PPF-SE is as follows

Step 1: Input network parameters, state variables, measurement variables, establish the state equation and measurement equation.

Step 2: Initialization: Select particles $x_0^l \sim p(x_0)$, $l = 1,\ldots,N$, from the initial distribution $p(x_0)$, set the weight of particles as: $w_0^l = 1/N$, and assign them to GPGPU devices;

Step 3: Importance sampling: Sampling particle positions according to the probability transfer density function of the state equation, $x_k^l \sim p\left(x_k | x_{k-1}^l\right)$.

Step 4: Weight update: The likelihood function is established according to the measurement equation to calculate the corresponding weight of the particle: $w_k^l \sim p\left(y_k | x_k^l\right)$;

Step 5: Using CUDA parallel reduction function to normalize particle weights: $\hat{w}_k^l = w_k^l \Big/ \sum_{i=1}^{N} w_k^l$;

Step 6: Resampling: By the established Rejecting-Resampling method, N new particles are generated from particle set $\{x_k^l\}$ according to the importance weight of particles;

Step 7: State estimation: Calculate the system state based on the position and weight of particles, $\hat{x}_k = \sum_{i=1}^{N} \hat{w}_k^l x_k^l$.

Step 8: The model time is moved to the next moment, and the Steps 2 through 6 are repeated.

The flow chart of distribution network state estimation based on PPF is shown in Fig.6.

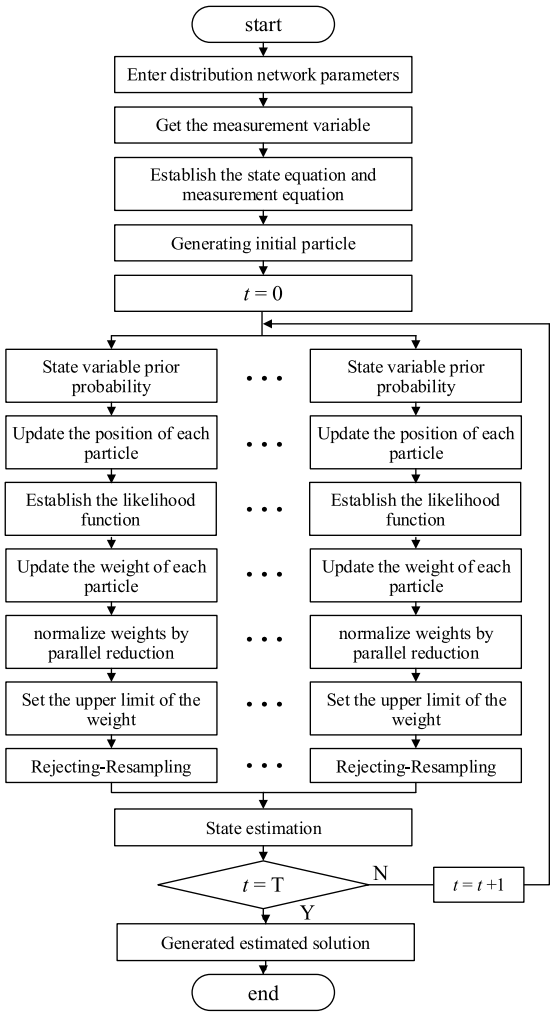

FIGURE 6. Flow chart of PPF-SE algorithm.

V. CASE STUDY

A. CASE DESCRIPTION

In this study, simulations are conducted using the IEEE 136-bus distribution network test system in Matpower [44], with the network topology depicted in Fig.7. The reason for choosing the Matpower 136-bus power distribution network test system is that it is a standardized simulation case, it is of moderate scale, and its data has a high degree of credibility. The test system encompasses 136 buses, 135 branches (excluding contact lines), and 6 switch paths (depicted as blue dotted lines in Fig.7). The system operates at a rated voltage of 13.8kV and a rated capacity of 100MVA. Simulations are performed on a platform equipped with an Intel Core i7 processor clocked at 4.2GHz, 16GB of memory, and running the Win10 operating system. The simulation software is Matlab R2022a + CUDA10.7.

Taking the A-phase voltage state of the system as the primary object of analysis. It is assumed that pseudo-measurements have been added to the voltage and phase angle at each node of the system. The original measured data for the state variables of voltage and phase angle are shown in Fig. 8, where (a) and (d) depict the distribution of the true voltage and phase angle, respectively. (b) and (e) illustrate the noise distribution of the two types of state variables under a Gaussian noise scenario, while (c) and (f) convey the noise distribution of the two types of states under a non-Gaussian noise scenario.

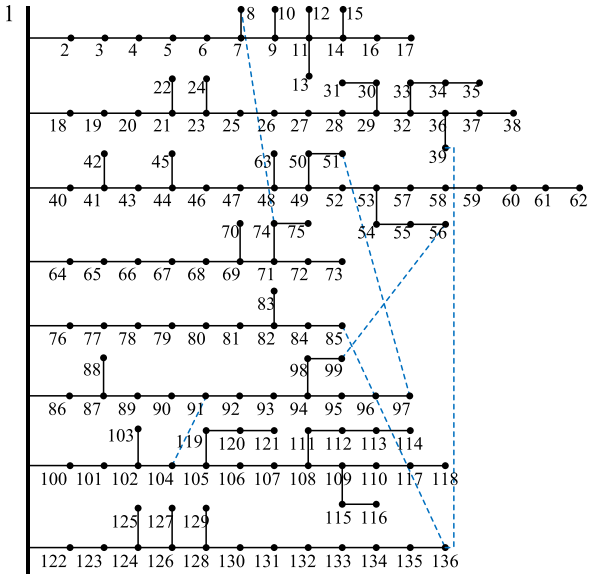

FIGURE 7. Topological structure of Matpower 136-bus test case.

As can be seen from Fig. 8, there is a clear presence of noise components in the measurement data, which can cause distortions during the voltage collection and application process. Moreover, Gaussian noise and non-Gaussian colored noise show distinctly different distribution patterns. The randomness of Gaussian noise is more uniform, whereas non-Gaussian colored noise exhibits a certain distribution pattern with varied randomness. Therefore, the purpose of this case study is to utilize the proposed PPF algorithm to filter out Gaussian noise and non-Gaussian colored noise, and to verify its effectiveness through comparison. To facilitate description and analysis, the A-phase voltage at the 52nd time is selected as the observation variable to verify the proposed method.

B. PERFORMANCE VERIFICATION OF PPF-SE ALGORITHM UNDER GAUSSIAN NOISE

When the measurement data of the distribution system is subjected to Gaussian noise interference, the proposed PPF algorithm is used to filter the raw voltage and phase data. In addition, the most commonly used WLS and KF algorithms are selected for comparison, and other superior filtering algorithms such as Bayesian Unscented Kalman Filter (BUKF) and Maximum Correntropy Criterion Extended Kalman Filter (MCCEKF) are chosen as performance references [45], [46], to demonstrate the effectiveness of the proposed algorithm in this paper.

The filtering results of various algorithms are shown in Fig. 9, where Fig. 9(a) displays the distribution of voltage state quantities and Fig. 9(b) displays the distribution of phase angle state variables. Additionally, a brief period is chosen from each state variable for magnification, forming an intuitive display of the comparative filtering performance, as shown in the subfigures in Fig.9(a) and (b).

TABLE 1. State estimation evaluation index under Gaussian noise.

	RMSE($\times 10^{-4}$)	MAAE($\times 10^{-4}$)	MEAE($\times 10^{-6}$)
Original	7.1607	23.4888	17.2712
WLS	4.9820	12.4759	9.1734
KF	3.7025	12.9359	9.5117
BUKF	2.0759	6.7587	4.9696
MCCEKF	2.1662	5.5822	4.1045
PPF	1.4713	3.2873	2.4171

As can be seen from Fig. 9, when the system's measured data contains Gaussian noise, several state estimation methods can be effectively used to filter state variables. Compared to the other four methods, the state estimation results obtained using the PPF algorithm can more stably fit the actual state. Moreover, the results obtained by the BUKF and MCCEKF algorithms are superior to those obtained by the WLS and KF algorithms. To analyze the estimation performance of the various methods more clearly and intuitively, we conducted a pre- and post-estimation error analysis under different algorithms, as shown in Fig. 10. Fig. 10 (a) and (c) represent the distribution of the per-node state variable error after filtering by several algorithms at time 52, while Fig. 10 (b) and (d) show the comparison of the error distribution after PPF filtering and the original error distribution over the entire duration.

It can be seen from Fig. 10 that the proposed PPF-SE algorithm has the best state estimation effect for the distribution network, whose overall error after estimation is comparatively stable, while larger errors at some times are caused by the prediction deviations in the importance sampling process.

In the case of measurements containing Gaussian noise, the WLS, KF, BUKF, MCCEKF, and PPF algorithms can all achieve a certain degree of state variable filtering effect. However, due to the voltage distribution characteristics of none spatial node in the distribution network are considered adequately, there is still a certain degree of fluctuation error. In order to quantitatively analyze the error, the evaluation indexes of voltage state variables are shown in Table 1.

As can be seen from Table 1, the RMSE, MEAE, and MAAE in the original measurement data are relatively large, indicating a significant error fluctuation in the measurement process. After filtering by various state estimation methods, these three indicators have significantly decreased, demonstrating the effectiveness of the WLS, KF, BUKF, MCCEKF, and PPF algorithms. The estimation performance indicator values obtained through WLS and KF are essentially the same, and the results obtained through the BUKF, MCCEKF, and PPF algorithms are significantly lower than the aforementioned two algorithms, with the PPF achieving the lowest indicator value. This suggests that among the five estimation methods, the PPF algorithm performs the best, while the WLS and KF have certain filtering effects, but are not as effective as the other three algorithms. In addition, all the methods applied some filtering effect on the Gaussian noise,

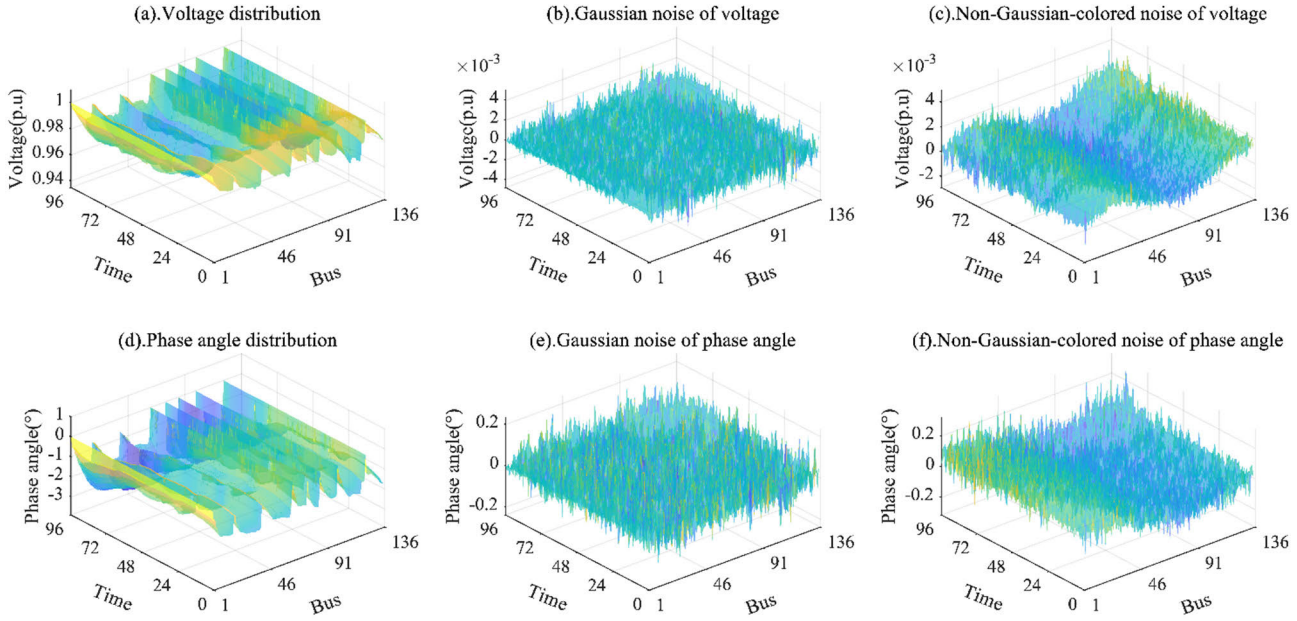

FIGURE 8. Distribution characteristics of initial state variables and two types of noise.

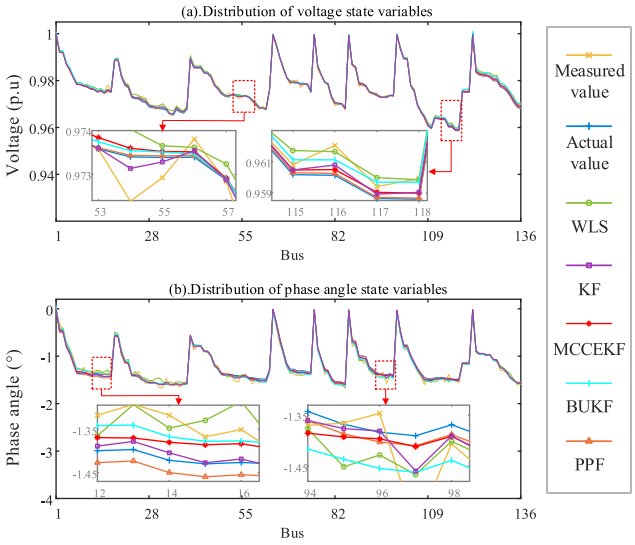

FIGURE 9. State estimation results under Gaussian noise.

suggesting that they all have a certain applicability under Gaussian noise.

C. PERFORMANCE VERIFICATION OF PPF-SE ALGORITHM UNDER NON-GAUSSIAN-COLORED NOISE

In the data measured by the power distribution network acquisition system, there are many non-Gaussian colored noise, which do not obviously follow a Gaussian distribution. State estimation methods based on WLS, KF, BUKF, MCCE, and PPF have different filtering effects on non-Gaussian colored noise, so these methods are used separately to process the voltage and phase angle state quantities with non-Gaussian noise. At this point, the results obtained by several state estimation methods are shown in Fig.11.

Fig.11(a) shows the distribution of voltage state variables, and Fig.11(b) shows the distribution of phase angle state variables. Similarly, a shorter period is selected in each state variable for magnified display to more intuitively present the filtering results for easy comparison, as shown in the subfigures.

From Fig. 11, it can be seen that, compared with the existence of Gaussian noise in the system, non-Gaussian colored noise will bring worse state estimation results, especially when using the WLS and KF algorithms. However, for BUKF and MCCEKF, they have certain processing capabilities for non-Gaussian colored noise, and the state estimation results are better. The PPF algorithm is not affected by non-Gaussian noise, and its filtering results for state variables are basically consistent with those in the presence of Gaussian noise. In order to analyze the distribution pattern of non-Gaussian colored noise and the filtering effect of various state estimation methods, the node voltage and phase angle noise distribution at time 52 are selected for verification, as shown in Fig. 12(a) and Fig. 12(c) respectively. In addition, Fig. 12(b) and Fig. 12(d) respectively display the comparison of the error distribution of the voltage state variables and phase angle state variables after PPF filtering throughout the entire time period with the original error distribution.

In Fig. 12, the voltage state variable of each node is affected by non-Gaussian colored noise, resulting in uneven distribution of voltage measurement errors. In the presence of non-Gaussian colored noise, the filtering effect of various state estimation algorithms is affected. WLS is unable to accurately implement effective estimation of state variables. KF, BUKF, and MCCEKF can filter this noise component

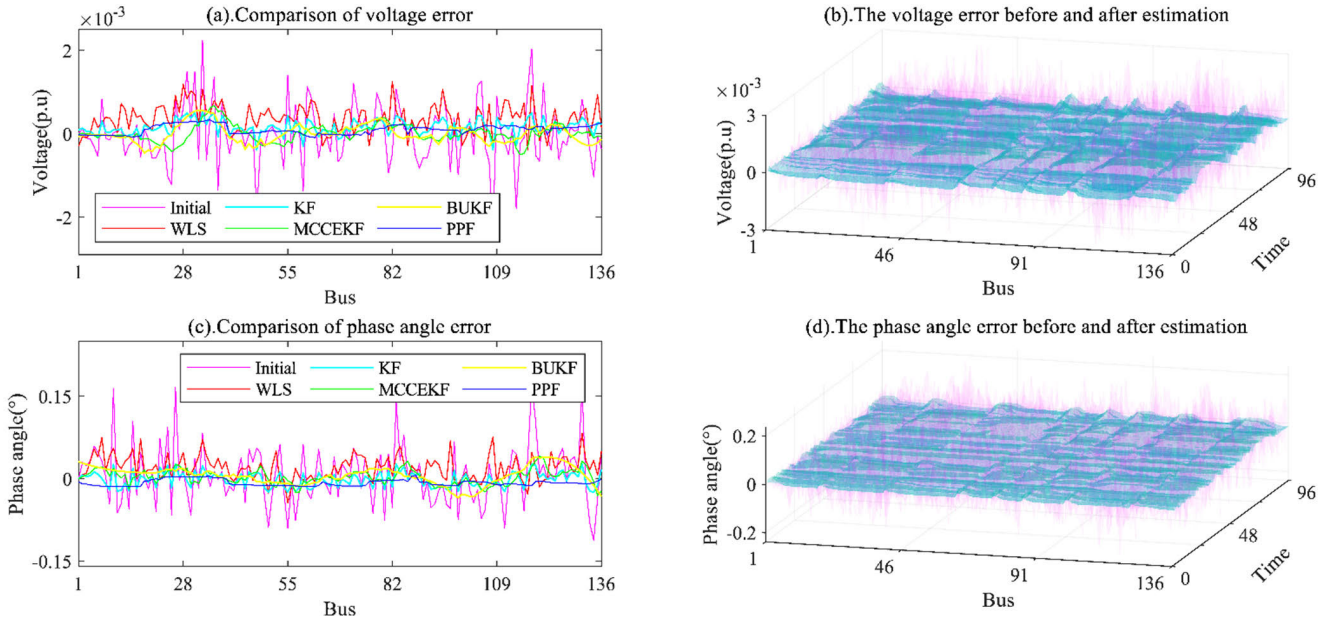

FIGURE 10. Noise distribution under Gaussian noise.

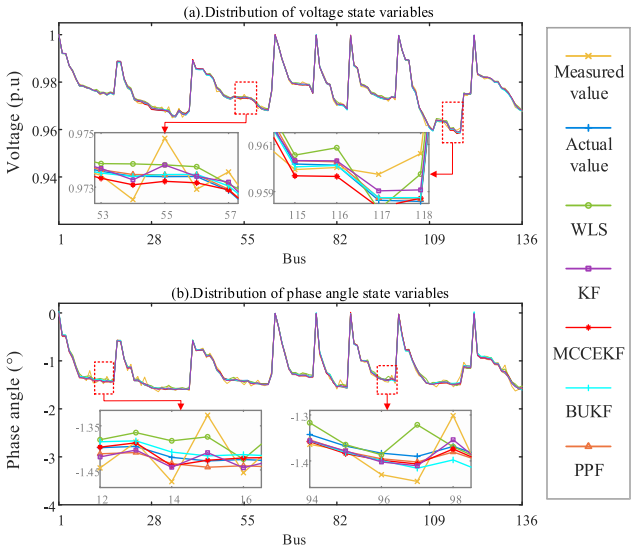

FIGURE 11. State estimation results under non-Gaussian-colored noise.

to a certain extent, but their effects are limited and their results still involve some non-Gaussian or colored properties. However, the PPF algorithm has achieved better state estimation results, which are essentially consistent with the filtering results under Gaussian noise. The filtering evaluation indicators in this situation are shown in Table 2.

As can be seen from Table 2, the filtering results obtained through WLS are relatively poor, indicating its fundamental lack of ability to process non-Gaussian colored noise. The state estimation evaluation indicators for MCCEKF are still relatively poor, which is due to the algorithm's ability to handle non-Gaussian noise but lack of ability to eliminate colored

TABLE 2. State estimation evaluation index under non-Gaussian noise.

	RMSE($\times 10^{-4}$)	MAAE($\times 10^{-4}$)	MEAE($\times 10^{-6}$)
Original	10.3914	27.7598	20.4116
WLS	9.8977	24.5602	18.0589
KF	3.9109	10.7019	7.8690
BUKF	3.6229	9.7272	7.1523
MCCEKF	7.2478	16.7241	11.6951
PPF-SE	1.4701	3.6846	2.7092

noise. The estimation results for the voltage state variables by KF and BUKF are essentially consistent, indicating a certain ability to filter non-Gaussian colored noise. However, among the various methods, PPF has the best state estimation effect, which demonstrates the effectiveness of the established state estimation method in dealing with non-Gaussian colored noise.

The Signal-to-Noise Ratio (SNR) is another metric used to measure the noise content in a signal, indicating the ratio of valid information to background noise within the signal. Typically, the higher the SNR, the better the signal quality, as this implies a higher proportion of valid information. The SNR of the original signal and the signal filtered using the five state estimation methods are calculated separately, as shown in Fig. 13. Specifically, Fig. 13(a) and (c) demonstrate the Gaussian SNR for voltage and phase angle, while Fig. 13(b) and (d) exhibit the non-Gaussian SNR for voltage and phase angle, respectively.

As can be seen from Fig.13, under Gaussian noise, whether for voltage state variables or phase angle state variables, the SNR enhancement effects of the BUKF, MCCEKF, and

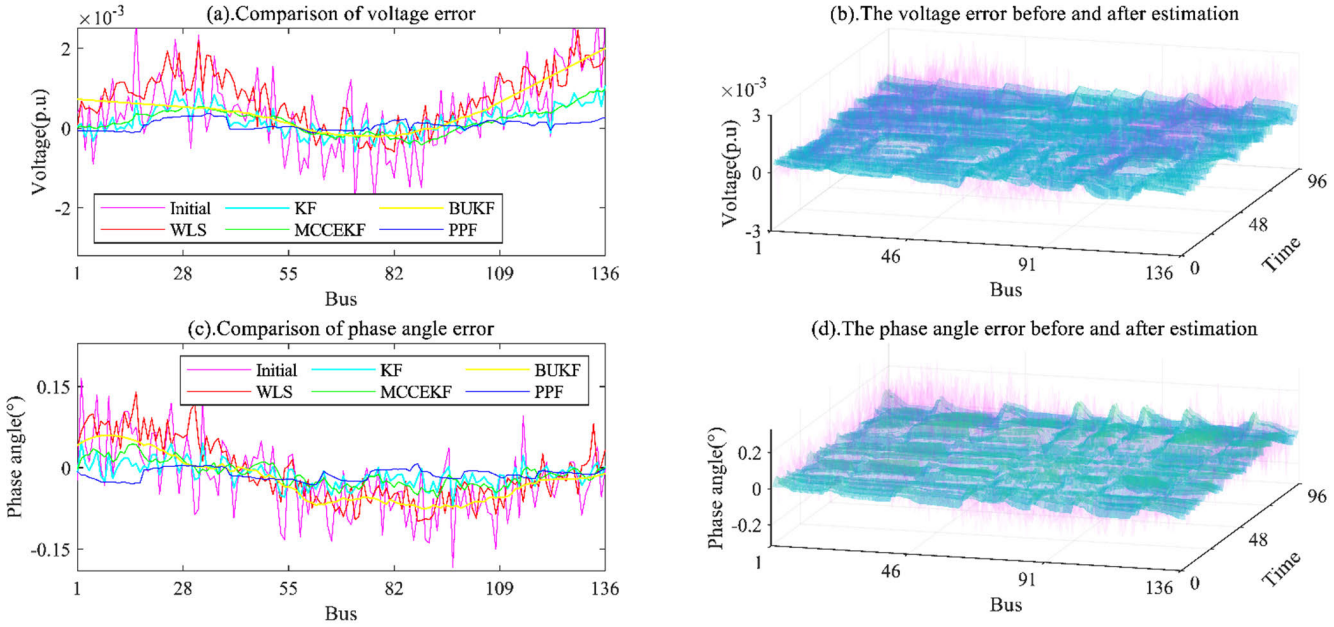

FIGURE 12. Noise distribution under non-Gaussian-colored noise.

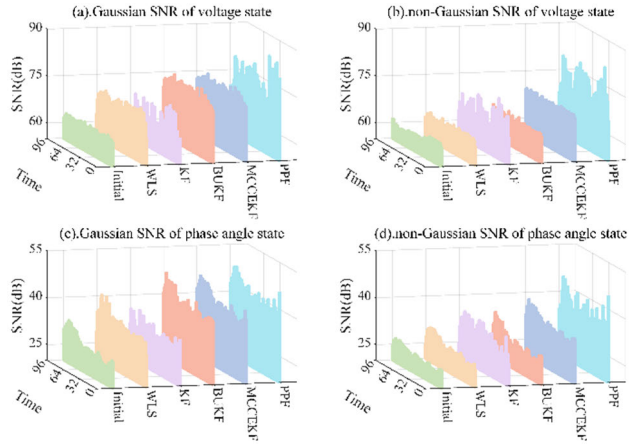

FIGURE 13. SNR under different estimation methods.

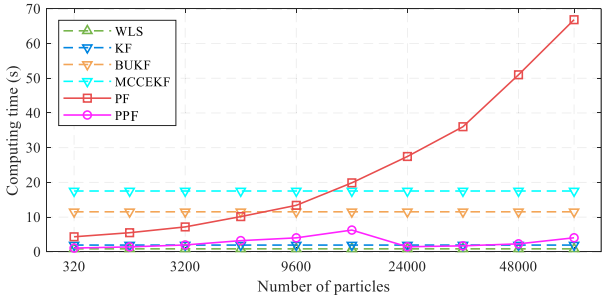

FIGURE 14. Computation time under different estimation methods.

PPF state estimation algorithms are all quite significant. This demonstrates that the three state estimation methods have roughly consistent filtering effects on Gaussian noise. However, in non-Gaussian noise, the BUKF algorithm is not adept at handling colored noise, resulting in a SNR lower than that of MCCEKF and PPF, with PPF achieving the highest SNR. This further exhibits PPF's capability in handling non-Gaussian colored noise.

D. PARALLEL PERFORMANCE ANALYSIS FOR PPF-SE ALGORITHM

Another advantage of using the PPF algorithm for estimating the state variables of the distribution network lies in its computation speed. This section primarily analyzes the computational complexity and execution time of various algorithms. The PPF algorithm fully leverages the parallel features of the GPGPU-CUDA architecture to enhance computational efficiency. The degree of parallelism is related to the number of particles chosen. The traditional PF algorithm would expend a significant amount of time processing each particle individually, while the PPF can handle all particles simultaneously. Therefore, by comparing the calculation time of the proposed PPF algorithm with various other algorithms, the improved efficiency of parallel computation can be observed, as shown in Fig.14.

As can be seen from Fig. 14, for the conventional PF method, the computing time consumed by state estimation increases with the increase of set-up particles, while for the proposed PPF algorithm, the calculation time is basically unchanged. When the number of particles is 48,000, the calculation time of PF is approximately 50.9s, while using the PPF algorithm, the consumption time is only 2.29s, through which the speed increases by about 22 times. Moreover,

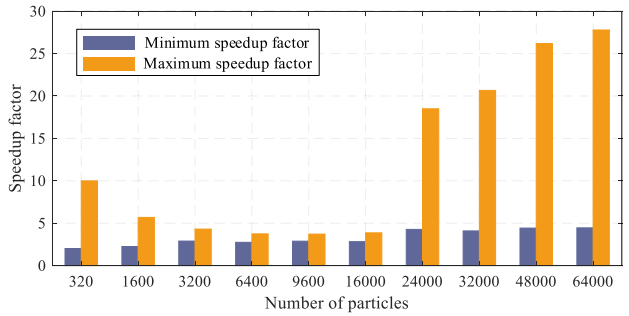

FIGURE 15. Parallel speedup factor with different particle numbers.

with the increase of particle number, the speed increase ratio is higher, which indicates that the application of parallel computing technology to the conventional PF algorithm has an obvious efficiency improvement effect. Furthermore, compared with the BUKF and MCCEKF algorithms, the computation time of the PPF algorithm is still lower, which demonstrates that the idea of parallelism effectively enhances the computational efficiency.

The speedup is an effective index to measure parallel efficiency. It is generally used to evaluate the performance improvement achieved by parallel computations compared to sequential ones. Fig. 15 shows the maximum and minimum parallel speedup factor of PPF-SE with different particle numbers.

In Fig. 15, the maximum speedup factor is the ratio at the longest serial computation time to the shortest parallel computation time, and the minimum speedup is the ratio at the shortest serial computation time to the longest parallel computation time. As can be seen from Fig. 15, when the number of particles is greater than 24,000, the speedup factor is significantly improved, which indicates that parallel computing technology is more suitable for large-scale computing. In addition, the speedup factor does not significantly improve when the number of particles is between 3200 and 16000. This is primarily due to the higher memory bandwidth utilization when the number of particles increases. In large-scale parallel computing, the utilization rate of memory bandwidth is usually higher than in small-scale parallel computing. When the number of threads increases, memory access conflicts can be reduced, thereby improving the utilization rate of memory bandwidth. On the other hand, when the number of threads exceeds a certain value, the overhead of task scheduling and context switching decreases, which also improves parallel efficiency. This indicates that there is still room for improvement and research value in the future.

VI. CONCLUSION

In this paper, a fast distribution system state estimation method based on PPF is proposed, through which the problem of non-Gaussian-colored noises in measured data can be effectively solved. Through this method, particle thread-level parallel computing based on CUDA-GPGPU can be realized while dealing with the characteristics that cannot be decomposed in the resampling process of the conventional PF algorithm. An IEEE 136-bus test system is used to validate the approach proposed in scenarios, where the measurements include non-Gaussian-colored noises. The conclusions drawn are as follows:

(1) By utilizing the GPGPU-CUDA architecture to parallelize the conventional PF state estimation algorithm, it is possible to effectively enhance the computational speed, compensating for the lengthy computation time of the PF algorithm. Moreover, as the number of particles increases, the parallel efficiency noticeably improves, although it is subject to the limitations of hardware conditions and the scale of state estimation computations.

(2) The established PPF state estimation algorithm can effectively deal with the non-Gaussian colored noise interference encountered during the measurement process in the distribution network. Compared with other methods, the results obtained under non-Gaussian noise are essentially consistent with those obtained under Gaussian noise, indicating its effectiveness in filtering out non-Gaussian colored noise.

Future work will further optimize the algorithm and expand its application in more complex and larger scale power systems.

REFERENCES

[1] X. Wang, "Power systems dynamic state estimation with the two-step fault tolerant extended Kalman filtering," *IEEE Access*, vol. 9, pp. 137211–137223, 2021.

[2] A. Abur and A. Gomez-Exposito, *Power System State Estimation: Theory and Implementation*. Boca Raton, FL, USA: CRC Press, 2004.

[3] H. Wu, Z. Xu, and M. Wang, "Unrolled spatiotemporal graph convolutional network for distribution system state estimation and forecasting," *IEEE Trans. Sustain. Energy*, vol. 14, no. 1, pp. 297–308, Jan. 2023.

[4] G. Tian, Y. Gu, Z. Yu, Q. Zhang, D. Shi, Q. Zhou, and Z. Wang, "Enhanced denoising autoencoder-aided bad data filtering for synchrophasor-based state estimation," *CSEE J. Power Energy Syst.*, vol. 8, no. 2, pp. 640–651, Mar. 2022.

[5] Y. Chen, Y. Yao, and Y. Zhang, "A robust state estimation method based on SOCP for integrated electricity-heat system," *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 810–820, Jan. 2021.

[6] M. Shafiei, G. Nourbakhsh, A. Arefi, G. Ledwich, and H. Pezeshki, "Single iteration conditional based DSSE considering spatial and temporal correlation," *Int. J. Electr. Power Energy Syst.*, vol. 107, pp. 644–655, May 2019.

[7] H. Karimipour and V. Dinavahi, "Robust massively parallel dynamic state estimation of power systems against cyber-attack," *IEEE Access*, vol. 6, pp. 2984–2995, 2018.

[8] Y. Zhang, J. Wang, and Z. Li, "Interval state estimation with uncertainty of distributed generation and line parameters in unbalanced distribution systems," *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 762–772, Jan. 2020.

[9] A. S. Zamzam, X. Fu, and N. D. Sidiropoulos, "Data-driven learning-based optimization for distribution system state estimation," *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 4796–4805, Nov. 2019.

[10] G. Wang, A. S. Zamzam, G. B. Giannakis, and N. D. Sidiropoulos, "Power system state estimation via feasible point pursuit: Algorithms and Cramér–Rao bound," *IEEE Trans. Signal Process.*, vol. 66, no. 6, pp. 1649–1658, Mar. 2018.

[11] S. Wang, J. Zhao, Z. Huang, and R. Diao, "Assessing Gaussian assumption of PMU measurement error using field data," *IEEE Trans. Power Del.*, vol. 33, no. 6, pp. 3233–3236, Dec. 2018.

[12] Y. Ju, W. Wu, F. Ge, K. Ma, Y. Lin, and L. Ye, "Fast decoupled state estimation for distribution networks considering branch ampere measurements," *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6338–6347, Nov. 2018.

[13] H. Long, Z. Wu, C. Fang, W. Gu, X. Wei, and H. Zhan, "Cyber-attack detection strategy based on distribution system state estimation," *J. Mod. Power Syst. Clean Energy*, vol. 8, no. 4, pp. 669–678, Jul. 2020.
[14] M. Mao, J. Xu, Z. Wu, Q. Hu, and X. Dou, "A multiarea state estimation for distribution networks under mixed measurement environment," *IEEE Trans. Ind. Informat.*, vol. 18, no. 6, pp. 3620–3629, Jun. 2022.
[15] X. Kong, X. Zhang, X. Zhang, C. Wang, H.-D. Chiang, and P. Li, "Adaptive dynamic state estimation of distribution network based on interacting multiple model," *IEEE Trans. Sustain. Energy*, vol. 13, no. 2, pp. 643–652, Apr. 2022.
[16] D. Cao, J. Zhao, W. Hu, Q. Liao, Q. Huang, and Z. Chen, "Topology change aware data-driven probabilistic distribution state estimation based on Gaussian process," *IEEE Trans. Smart Grid*, vol. 14, no. 2, pp. 1317–1320, Mar. 2023.
[17] K. R. Mestav, J. Luengo-Rozas, and L. Tong, "Bayesian state estimation for unobservable distribution systems via deep learning," *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 4910–4920, Nov. 2019.
[18] L. Zhang, G. Wang, and G. B. Giannakis, "Real-time power system state estimation and forecasting via deep unrolled neural networks," *IEEE Trans. Signal Process.*, vol. 67, no. 15, pp. 4069–4077, Aug. 2019.
[19] K. Dehghanpour, Z. Wang, J. Wang, Y. Yuan, and F. Bu, "A survey on state estimation techniques and challenges in smart distribution systems," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 2312–2322, Mar. 2019.
[20] C. Carquex, C. Rosenberg, and K. Bhattacharya, "State estimation in power distribution systems based on ensemble Kalman filtering," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6600–6610, Nov. 2018.
[21] Z. Yu, D. Shi, Z. Wang, Q. Zhang, J. Huang, and S. Pan, "Distributed estimation of oscillations in power systems: An extended Kalman filtering approach," *CSEE J. Power Energy Syst.*, vol. 5, no. 2, pp. 181–189, Jun. 2019.
[22] J. Zhao and L. Mili, "Robust unscented Kalman filter for power system dynamic state estimation with unknown noise statistics," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 1215–1224, Mar. 2019.
[23] Y. Zhang and J. Wang, "Towards highly efficient state estimation with nonlinear measurements in distribution systems," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 2471–2474, May 2020.
[24] C. Lin, W. Wu, and Y. Guo, "Decentralized robust state estimation of active distribution grids incorporating microgrids based on PMU measurements," *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 810–820, Jan. 2020.
[25] H. Yang, R. C. Qiu, L. Chu, T. Mi, X. Shi, and C. M. Liu, "Improving power system state estimation based on matrix-level cleaning," *IEEE Trans. Power Syst.*, vol. 35, no. 5, pp. 3529–3540, Sep. 2020.
[26] Y. Cui and R. Kavasseri, "A particle filter for dynamic state estimation in multi-machine systems with detailed models," *IEEE Trans. Power Syst.*, vol. 30, no. 6, pp. 3377–3385, Nov. 2015.
[27] C. Zhang, L. Luo, Z. Yang, S. Zhao, Y. He, X. Wang, and H. Wang, "Battery SOH estimation method based on gradual decreasing current, double correlation analysis and GRU," *Green Energy Intell. Transp.*, vol. 2, no. 5, Oct. 2023, Art. no. 100108.
[28] C. Zhang, S. Zhao, Z. Yang, and Y. Chen, "A reliable data-driven state-of-health estimation model for lithium-ion batteries in electric vehicles," *Frontiers Energy Res.*, vol. 10, Sep. 2022, Art. no. 1013800.
[29] C. Zhang, S. Zhao, and Y. He, "An integrated method of the future capacity and RUL prediction for lithium-ion battery pack," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 2601–2613, Mar. 2022.
[30] S. Zhao, C. Zhang, and Y. Wang, "Lithium-ion battery capacity and remaining useful life prediction using board learning system and long short-term memory neural network," *J. Energy Storage*, vol. 52, Aug. 2022, Art. no. 104901.
[31] G. Kan, T. Lei, K. Liang, J. Li, L. Ding, X. He, H. Yu, D. Zhang, D. Zuo, Z. Bao, M. Amo-Boateng, Y. Hu, and M. Zhang, "A multi-core CPU and many-core GPU based fast parallel shuffled complex evolution global optimization approach," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 2, pp. 332–344, Feb. 2017.
[32] Z.-H. Liu, X.-H. Li, L.-H. Wu, S.-W. Zhou, and K. Liu, "GPU-accelerated parallel coevolutionary algorithm for parameters identification and temperature monitoring in permanent magnet synchronous machines," *IEEE Trans. Ind. Informat.*, vol. 11, no. 5, pp. 1220–1230, Oct. 2015.
[33] T. Cheng, N. Lin, and V. Dinavahi, "Hybrid parallel-in-time-and-space transient stability simulation of large-scale AC/DC grids," *IEEE Trans. Power Syst.*, vol. 37, no. 6, pp. 4709–4719, Nov. 2022.
[34] R. Cisneros-Magaña, A. Medina, V. Dinavahi, and A. Ramos-Paz, "Time-domain power quality state estimation based on Kalman filter using parallel computing on graphics processing units," *IEEE Access*, vol. 6, pp. 21152–21163, 2018.
[35] Z. Zhou and V. Dinavahi, "Fine-grained network decomposition for massively parallel electromagnetic transient simulation of large power systems," *IEEE Power Energy Technol. Syst. J.*, vol. 4, no. 3, pp. 51–64, Sep. 2017.
[36] O. Brun, V. Teuliere, and J.-M. Garcia, "Parallel particle filtering," *J. Parallel Distrib. Comput.*, vol. 62, no. 7, pp. 1186–1202, Jul. 2002.
[37] T. Ahmad and N. Senroy, "An information theoretic approach to power-substation level dynamic state estimation with non-Gaussian noise," *IEEE Trans. Power Syst.*, vol. 35, no. 2, pp. 1642–1645, Mar. 2020.
[38] Y. Luo, J. Zhou, and W. Yang, "Distributed state estimation with colored noises," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 6, pp. 2807–2811, Jun. 2022.
[39] R. Tong and Z. Ye, "Supplementations to the higher order subspace algorithm for suppression of spatially colored noise," *IEEE Signal Process. Lett.*, vol. 24, no. 5, pp. 668–672, May 2017.
[40] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019.
[41] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.
[42] A. Daniyan, Y. Gong, and S. Lambotharan, "An improved resampling approach for particle filters in tracking," in *Proc. 22nd Int. Conf. Digit. Signal Process. (DSP)*, London, U.K., Aug. 2017, pp. 1–5.
[43] M. T. Rahman, M. Javad-Kalbasi, and S. Valaee, "Near-optimal resampling in particle filters using the Ising energy model," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Toronto, ON, Canada, Jun. 2021, pp. 5464–5468.
[44] (2023). *Matpower Home Page, A MATLAB Power System Simulation Package*. [Online]. Available: http://www.pserc.cornell.edu/matpower/
[45] Z. Liu, S.-C. Chan, H.-C. Wu, and J. Wu, "Bayesian unscented Kalman filter for state estimation of nonlinear and non-Gaussian systems," in *Proc. 24th EUSIPCO*, Budapest, Hungary, 2016, pp. 443–447.
[46] B. Xu and X. Wang, "Maximum correntropy extend Kalman filter for multiple AUVs cooperative localization," in *Proc. CCDC*, Nanchang, China, 2019, pp. 4670–4675.