

Dynamic State Estimation of a Synchronous Machine Using PMU Data: A Comparative Study

Ning Zhou, *Senior Member, IEEE*, Da Meng, *Member, IEEE*, Zhenyu Huang, *Senior Member, IEEE*,
and Greg Welch, *Member, IEEE*

Abstract—Accurate information about dynamic states is important for efficient control and operation of a power system. This paper compares the performance of four Bayesian-based filtering approaches in estimating dynamic states of a synchronous machine using phasor measurement unit data. The four methods are extended Kalman filter, unscented Kalman filter, ensemble Kalman filter, and particle filter. The statistical performance of each algorithm is compared using Monte Carlo methods and a two-area-four-machine test system. Under the statistical framework, robustness against measurement noise and process noise, sensitivity to sampling interval, and computation time are evaluated and compared for each approach. Based on the comparison, this paper makes some recommendations for the proper use of the methods.

Index Terms—Ensemble Kalman filter (EnKF), extended Kalman filter (EKF), particle filter (PF), phasor measurement unit (PMU), power system dynamics, state estimation, unscented Kalman filter (UKF).

I. INTRODUCTION

ELECTROMECHANICAL dynamic models are widely used to study transient and small signal stability problems in power systems. States are the minimum set of variables that can determine the status of a dynamic system [1]. A dynamic model with accurate states can faithfully reveal system responses and, therefore, be used to enhance system stability, and reliability of a power system.

The real-time states of electromechanical dynamics reflect the current status of a power system and can be used to coordinate controllers in a wide area. Traditionally, estimating

dynamic states (e.g., rotor speeds and angles) over a wide area was not possible because data from the supervisory control and data acquisition (SCADA) system is normally sampled every 2–4 s. This sampling rate is too low for the data to reveal electromechanical dynamic responses. In addition, SCADA data is not well synchronized. As a result, most controllers (e.g., power system stabilizers) for controlling the electromechanical dynamics only use local states to achieve their control objectives. Compatibility among controllers is only studied in planning models. Without global objectives and systematic coordination over wide areas, the influence of local controllers at the grid level may have adverse effects. For example, wide-area small-signal problems have been associated with “the introduction of high gain, and low time constant automatic voltage regulators” [2].

The dynamic states of wide-area power systems can be estimated in real time using data from phasor measurement units (PMU). The PMU data has a typical sampling rate of ten through 120 samples per second, is well synchronized with the global positioning system clock, and can continuously capture the dynamic response of a power system under normal and abnormal conditions. Huang *et al.* [3]–[5], Fan and Wehbe [6], and Farantatos *et al.* [25] revealed the benefits and potential use of PMU data with an extended Kalman filter (EKF) for online state and parameter estimation. Ghahremani and Kamwa [7], [8] used an EKF to simultaneously estimate the generator states and unknown inputs and an unscented Kalman filter (UKF) to estimate the dynamic states of a single-machine infinite bus system. Zhou *et al.* [9] proposed an ensemble Kalman filter (EnKF) method to simultaneously estimate the states and parameters. Zhou *et al.* [10] proposed an extended particle filter (PF) to estimate the dynamic states.

The previous work showed the value and feasibility of estimating dynamic states using PMU data with Bayesian-based filtering approaches. Yet, little work has been done on how to choose an appropriate method for various conditions encountered in actual applications. To better understand how to apply these methods for their optimal use, this paper systematically examines and compares four commonly used state estimation algorithms: EKF UKF, EnKF, and PF. First, this paper evaluates the statistical performance of the algorithms using Monte Carlo methods to reveal the average behavior of these algorithms under different instances of noise. In contrast Huang *et al.* [3], Fan and Wehbe [6],

Manuscript received December 17, 2013; revised May 20, 2014 and July 14, 2014; accepted July 22, 2014. Date of publication August 28, 2014; date of current version December 17, 2014. This work was supported in part by the U.S. Department of Energy (DOE) through its Advanced Grid Modeling Program and its Advanced Scientific Computing Research Applied Mathematics Program; and in part by the Pacific Northwest National Laboratory, operated by Battelle for DOE, under Contract DE-AC05-76RL01830. Paper no. TSG-00923-2013.

N. Zhou is with Department of Electrical and Computer Engineering, Binghamton University, Binghamton, NY 13902 USA (e-mail: zhenzhou@binghamton.edu).

D. Meng and Z. Huang are with Pacific Northwest National Laboratory, Richland, WA 99352 USA (e-mail: da.meng@pnl.gov; zhenyu.huang@pnl.gov).

G. Welch is with the University of Central Florida, Orlando, FL 32826 USA; and also with the University of North Carolina at Chapel Hill, Chapel Hill, NC 27599 USA (e-mail: welch@ucf.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSG.2014.2345698

and Ghahremani and Kamwa [7], [8] used only one single-noise instance to evaluate performance. Second, this paper tests how process noise may influence different algorithms by using a simple transient model to estimate states from simulated data using a more detailed sub-transient model with saturation effects. This setup introduces additional process noise and mimics the reality that a dynamic model is only a simplified description of a real system. Third, this paper evaluates how measurement sampling rates may influence the performance of the algorithms. Fan and Wehbe [6] and Ghahremani and Kamwa [7], [8] used a sampling rate much higher than 60 samples per second. On the other hand, Zhou *et al.* [9], [10] used a sampling rate of 25 samples per second. Huang *et al.* [3], [11] pointed out that lower sampling rate of the PMU measurement may degrade the EKF performance and proposed adding pseudo measurements through linear interpolation to increase the effective measurement sampling rate. The observation indicated that the sampling rate of measurement data may greatly affect the state estimation accuracy. To make a comparison under a realistic scenario, this paper assumes PMU measurement of 25 samples/second and uses interpolation to increase the effective sampling rate. Forth, in addition to adding noise to measurements, missing data and outliers are intentionally added to test the robustness of each algorithm when the actual measurement noise does not strictly follow the assumed noise model. All of these studies were carried out under a statistical framework using the Monte Carlo method.

This paper is organized as follows. Section II introduces the estimation model based on a fourth order generator models. Section III provides an overview of the EKF, UKF, EnKF, and PF. Section IV discusses the study approaches. Section V evaluates and compares the performance of the state estimation algorithms using a two-area-four-machine test system. In Section VI, the conclusion is discussed.

II. ESTIMATION MODELS

This section introduces estimation models that are used by all filtering algorithms for estimating states. Also, to apply the filtering method for discrete measurements, a modified Euler method is applied to discretize the continuous model. Note that in real-world applications, estimation models can be different from simulation models because an available model is often a simplified description of a real-world system. Considering this fact, to mimic real system dynamic responses, the estimation model can be intentionally set to be different from the simulation models in some studies. A simulation model is used to generate PMU data while an estimation model is used by a filtering method to estimate states. Readers may refer to [11] and [12] for simulation models used for mimicking a real system and generating simulation data.

A. Fourth Order Transient Model

This paper uses positive sequence model (1) to describe a synchronous machine in local d - q reference frame to estimate

the states

$$\begin{cases} \dot{\delta} = \omega_0 \Delta\omega & (1a) \\ \Delta\dot{\omega} = \frac{1}{2H} (T_m - T_e - K_D \Delta\omega) & (1b) \\ \dot{e}'_q = \frac{1}{T'_{d0}} (E_{fd} - e'_q - (x_d - x'_d) i_d) & (1c) \\ \dot{e}'_d = \frac{1}{T'_{q0}} (-e'_d + (x_q - x'_q) i_q). & (1d) \end{cases}$$

In (1), δ is the rotor angles in radian, by which the local q axis leads the global R axis; $\Delta\omega$ is rotor speed deviation; e'_d and e'_q are the transient voltages along local d and q axes; and i_d and i_q are the stator currents along local d and q axes. The parameter T_m is the mechanical torque; T_e is the electric air-gap torque; and E_{fd} is the internal field voltage. The parameter ω_0 is the rated value of the angular frequency; H is the inertia constant, and K_D is the damping factor. The parameters T'_{d0} and T'_{q0} are the open circuit time constants in the directions of the d and q axes, respectively; x_d and x_q are the synchronous reactance at d and q axes, respectively; and x'_d and x'_q are the transient reactance at d and q axes, respectively.

To facilitate the notation, (1) is transformed into a general state space model as given in (2) and (3)

$$\dot{x} = f_c(x, u) + w_c \quad (2a)$$

$$y = h_c(x, u) + v_c \quad (2b)$$

$$E[w_c w_c^T] = Q \quad (2c)$$

$$E[v_c v_c^T] = R \quad (2d)$$

$$x = [\delta \ \Delta\omega \ e'_q \ e'_d]^T \quad (3a)$$

$$u = [T_m \ E_{fd} \ i_R \ i_I]^T \quad (3b)$$

$$y = [e_R \ e_I]^T. \quad (3c)$$

In (2), x is the state vector, u is the input vector, and y is the output vector. Functions $f_c(*)$ and $h_c(*)$ are the state transition and output functions, respectively. The subscript “ c ” indicates the continuous-time model. The vectors w_c and v_c represent the process and output noise, respectively. They are modeled as Gaussian white noise whose covariance matrices are defined by (2c) and (2d), where $E[*]$ represents statistical expectation. In (3), e_R and e_I are the stator voltages along the R and I axes of the global reference frame. The symbol i_R and i_I are the stator currents along the R and I axes of the global reference frame.

To transform (1) into $f_c(*)$ in (2a), i_d , i_q , and T_e were written as functions of x and u using (4)

$$i_d = i_R \sin \delta - i_I \cos \delta \quad (4a)$$

$$i_q = i_I \sin \delta + i_R \cos \delta \quad (4b)$$

$$T_e \approx P_I = (e'_d + i_q x'_q) i_d + (e'_q - i_d x'_d) i_q. \quad (4c)$$

Similarly, to implement output function $h_c(*)$ in (2b), e_I and e_R were written as functions of x and u using (5). Note that i_d and i_q in (5) are the functions of i_R and i_I as in (4a) and (4b). The model defined by (1), (4), and (5) is then discretized and

used for estimating states

$$e_R = (e'_d + i_q x'_q) \sin \delta + (e'_q - i_d x'_d) \cos \delta \quad (5a)$$

$$e_I = (e'_q - i_d x'_d) \sin \delta - (e'_d + i_q x'_q) \cos \delta. \quad (5b)$$

B. Problem Formulation and Model Discretization

Assuming a sampling interval of Δt seconds, a real time dynamic state estimation can be formulated as a filtering problem as follows.

Given measurements $T_m(j\Delta t)$, $E_{fd}(j\Delta t)$, $I_l(j\Delta t)$, and $E_t(j\Delta t)$ for $j = 1, 2, \dots, k$, estimate the synchronous machine states $\delta(k\Delta t)$, $\Delta \omega(k\Delta t)$, $e'_q(k\Delta t)$ and $e'_d(k\Delta t)$. (Here, “j” is the time step index.)

Here, $E_t = e_R + je_I$ is the terminal voltage phasor and $I_l = i_R + ji_I$ is the terminal current phasor in the global reference frame. They are positive sequence variables and can be measured using PMUs.

To estimate states using the discrete measurements, the continuous time model in (2) was discretized into a discrete time model (6), where the subscript k indicates the time at $k\Delta t$

$$\begin{cases} x_k = f(x_{k-1}, u_{k-1}) + w_{k-1} & (6a) \\ y_k = h(x_k, u_k) + v_k. & (6b) \end{cases}$$

More specifically, the state transition function (2a) was discretized by applying the modified Euler method [12] using (7). In (7), \tilde{f}_c can be calculated by (8). When Δt is small enough, the discrete process noise w_{k-1} can be approximated by (9). Because the continuous time process noise w_c is defined by (2c), the mean of w_{k-1} is 0 and the covariance of w_{k-1} can be calculated as (10). Equation (10) indicates that the variance of process noise can be increased proportionally with the sampling interval Δt during the state estimation

$$x_k \approx x_{k-1} + \tilde{f}_c \Delta t + w_{k-1} \quad (7)$$

$$\begin{cases} \tilde{f}_c = (f_c(\tilde{x}_k, u_k) + f_c(x_{k-1}, u_k)) / 2 \\ \tilde{x}_k = x_{k-1} + \Delta t \cdot f_c(x_{k-1}, u_{k-1}) \end{cases} \quad (8)$$

$$w_{k-1} \approx \int_{(k-1)\Delta t}^{k\Delta t} w_c(\tau) d\tau \quad (9)$$

$$\begin{aligned} Q_d &\triangleq E(w_{k-1} w_{k-1}^T) \\ &= \int_{(k-1)\Delta t}^{k\Delta t} \int_{(k-1)\Delta t}^{k\Delta t} E[w_c(\tau_1) w_c^T(\tau_2)] d\tau_1 d\tau_2 \quad (10) \\ &= \int_{(k-1)\Delta t}^{k\Delta t} \int_{(k-1)\Delta t}^{k\Delta t} Q \delta(\tau_1 - \tau_2) d\tau_1 d\tau_2 = Q \Delta t. \end{aligned}$$

Measurement (2b) can be discretized into (11). Here, v_k is the discrete time output noise. Because the continuous time output noise v_c is defined by (2b), the mean value of v_k is 0. The covariance of v_k depends on how measurement instruments are setup. To simplify the study, this paper assumes no prefilter. Therefore, the covariance of the v_k may be computed using (12) [14]

$$y_k = h_c(x_k, u_k) + v_k \quad (11)$$

$$R_d \triangleq E(v_k v_k^T) = R. \quad (12)$$

III. REVIEW OF BAYESIAN-BASED FILTERING METHODS

The Kalman filter is the most widely used Bayesian-based method. It was named after Rudolf Kalman, who published his famous recursive method to estimate dynamic states [15]. Assuming Gaussian noise and a linear system, the Kalman filter provides minimum variance estimates of states through a recursive approach.

In addition to its original successful applications in linear systems, there are many publications adapting the Kalman filter to nonlinear systems. These nonlinear methods include but are not limited to EK, UKF, EnKF, and PF. One major difference among these nonlinear Kalman-filter methods is their approaches to propagating the mean and covariance of the dynamic states. The EKF [16], [17] linearizes the state space model using a first-order approximation. The mean and covariance of states are propagated using Jacobian matrices. The UKF [18] propagates the mean and covariance of states using a deterministic-sampling approach to pass the sigma points through the nonlinear system. The EnKF propagates the mean and covariance of states using a Monte Carlo sampling approach [19]. In the EnKF, the distribution of the states is represented by a collection of samples, referred to as ensembles. All the above Kalman filters assume the joint Gaussian distribution of both measurements and states, and use the Bayesian approach to derive the Kalman gain. In contrast, the PF [20] is a more general Bayesian approach, which does not rely on Gaussian noise assumption. Similar to the EnKF, the PF also uses the samples (also known as particles) to represent the probability distribution of random variables. Different from the EnKF, the PF directly corrects the states without assuming Gaussian distribution.

For a linear system with additive Gaussian noise, it is well known that the KF is an optimal estimator in the sense of obtaining minimum mean square error (MSE) estimate. In addition, the KF is a recursive estimator and can be implemented efficiently for many real time applications. Yet, for a nonlinear system, all available algorithms (such as EKF, UKF, EnKF, and PF) are only suboptimal estimators. Each algorithm has its advantages and disadvantages. The EKF is probably the most widely used estimation algorithm for nonlinear systems and is often considered a standard algorithm because of its high computational efficiency and high accuracy for quasi-linear systems [24]. Yet, when a system is highly nonlinear, the EKF tends to have poor estimation accuracy and even diverge. This is because first order Taylor approximation used in the EKF introduces too much error. In contrast, the UKF can achieve second or third order Taylor approximation for a nonlinear system. Therefore, the UKF tends to produce more accurate estimates than the EKF when system nonlinearity is severe and noise is additive. In addition, the UKF has the same order of the computational complexity as the EKF [18]. The EnKF was mainly motivated by the needs of solving a system with a large number of states. For a large system, the EnKF was shown to have overcome the unbounded error growth problem of the EKF and to require less computation time [19]. The applications of EKF, UKF, and EnKF are restricted by their additive Gaussian noise assumption. As a result, they are not suitable for analyzing probabilistic

distributions with multiple modes. In contrast, the PF is more general by not making the restrictive assumption and, therefore, is more applicable to highly nonlinear systems. It was shown that the PF can give more accurate estimates than the EKF for the two particular nonlinear models in [27]. Yet, the PF usually requires a very large number of samples to represent a probabilistic distribution and, therefore, is not suitable for a large system. Considering features of the EKF, UKF, EnKF, and PF, users often need to examine and test filtering algorithms to select appropriate algorithms according to the requirement of their specific applications.

Under a Bayesian framework, the implementations of these algorithms have a similar structure. After initialization, all the filtering algorithms assimilate one snap shot of data at every time step. For one snap shot of data, there are two steps: a prediction step and a correction step. In the prediction step, the mean and covariance of states at time step k are predicted based on the states at step $k-1$. In the correction step, the predicted mean and covariance are corrected based on new measurements obtained at time step k . The algorithms for implementing these filtering methods are detailed as follows.

A. EKF

The EKF linearizes the system at the current operating point using the Jacobian matrices as in (13)–(15), [16].

EKF Prediction

$$x_k^- = f(x_{k-1}, u_{k-1}) \quad k \geq 1 \quad (13a)$$

$$P_k^- = F_{k-1}P_{k-1}F_{k-1}^T + Q_d. \quad (13b)$$

EKF Correction

$$x_k = x_k^- + K_k \tilde{y}_k \quad (14a)$$

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_d)^{-1} \quad (14b)$$

$$\tilde{y}_k = z_k - h(x_k^-, u_k^-) \quad k \geq 1 \quad (14c)$$

$$P_k = (I - K_k H_k) P_k^- \quad (14d)$$

where x_k^- and P_k^- are known as the *a priori* mean and covariance (of the states), respectively. They are estimated from the data up to time step $k-1$. The symbols x_k and P_k are known as the *a posteriori* mean and covariance of the states respectively, which are derived by adding the information from z_k to x_k^- and P_k^- . The symbol K_k is the Kalman gain. The symbol \tilde{y}_k is the residual between estimate $h(x_{k-1}^-, V_{k-1})$ and measurement z_k . F_k and H_k are Jacobian matrices defined by (15). A perturbation approach is used to numerally derive the Jacobian matrices in this paper

$$F_{k-1} = \left. \frac{\partial f}{\partial x} \right|_{x_k^-} \quad (15a)$$

$$H_k = \left. \frac{\partial h}{\partial x} \right|_{x_k^-}. \quad (15b)$$

B. UKF

The UKF uses an unscented transform to pick a set of samples to represent the probability distribution of states and propagates these samples through the nonlinear functions f and h to reconstruct the mean and covariance. The UKF estimation method is summarized as (16) and (17) [18].

UKF Prediction

$$x_k^{j-} = f(x_{k-1}^j, u_{k-1}) \quad k \geq 1, i = 0, \dots, 2n \quad (16a)$$

$$x_{k-1}^0 = x_{k-1} \quad (16b)$$

$$x_{k-1}^i = x_{k-1} + \left(\sqrt{(n+\kappa)P_{k-1}} \right)_i, i = 1, \dots, n \quad (16c)$$

$$x_{k-1}^{i+n} = x_{k-1} - \left(\sqrt{(n+\kappa)P_{k-1}} \right)_i, i = 1, \dots, n \quad (16d)$$

$$x_k^- = \sum_{i=0}^{2n} W_i x_k^{i-} \quad (16e)$$

$$P_k^- = \sum_{i=0}^{2n} W_i (x_k^{i-} - x_k^-) (x_k^{i-} - x_k^-)^T + Q_d. \quad (16f)$$

UKF Correction

$$x_k = x_k^- + K_k \tilde{y}_k \quad (17a)$$

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_d)^{-1} \quad (17b)$$

$$H_k = \sum_{i=0}^{2n} W_i \tilde{y}_k \tilde{y}_k^T \quad (17c)$$

$$\tilde{y}_k = z_k - z_k^- \quad k \geq 1 \quad (17d)$$

$$z_k^- = \sum_{i=0}^{2n} W_i z_k^{i-} \quad (17e)$$

$$z_k^{i-} = h(x_k^{i-}, u_k) \quad k \geq 1, i = 0, \dots, 2n \quad (17f)$$

$$P_k = (I - K_k H_k) P_k^- \quad (17g)$$

where x_k^{i-} and W_i are $2n+1$ sigma points and their corresponding weights. κ is a scaling parameter that controls the positions of the sigma points.

C. EnKF

The EnKF uses samples (also known as ensembles) to represent and propagate the probability distributions of the states. By using a large number of samples, the probability density can be approximated with high accuracy. The EnKF can be summarized by (18) and (19) [19].

EnKF Prediction

$$x_k^{i-} = f(x_{k-1}^i, u_{k-1}) + w_{k-1}^i \quad k \geq 1, i = 1, \dots, n_{\text{enKF}} \quad (18a)$$

$$z_k^{i-} = h(x_k^{i-}, u_k) \quad k \geq 1, i = 1, \dots, n_{\text{enKF}} \quad (18b)$$

$$x_k^- = \frac{1}{n_{\text{enKF}}} \sum_{i=1}^{n_{\text{enKF}}} x_k^{i-} \quad (18c)$$

$$z_k^- = \frac{1}{n_{\text{enKF}}} \sum_{i=1}^{n_{\text{enKF}}} z_k^{i-}.$$

EnKF Correction

$$x_k^i = x_k^{i-} + K_k (z_k^i - z_k^{i-}) \quad (19a)$$

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_d)^{-1} \quad (19b)$$

$$P_k^- H_k^T = \frac{1}{n_{\text{enKF}}} \sum_{i=1}^{n_{\text{enKF}}} (x_k^{i-} - x_k^-) (z_k^{i-} - z_k^-)^T \quad (19c)$$

$$H_k P_k^- H_k^T = \frac{1}{n_{\text{enKF}}} \sum_{i=1}^{n_{\text{enKF}}} (z_k^{i-} - z_k^-) (z_k^{i-} - z_k^-)^T \quad (19d)$$

$$z_k^i = z_k + v_k^i \quad (19e)$$

where n_{enKF} is the total number of samples, which are used to represent the distribution. The variable w_{k-1}^i is a sample generated according to the Q_d to simulate process noise. The symbol x_k^i stands for the samples of *a posteriori* states. Note that using (19c) and (19d), the covariance matrix P_k does not need to be expressively calculated.

D. Basic PF

The PF can be applied to systems with Gaussian and other distributions. A basic PF approximates a probability distribution function by a set of weighted discrete samples, as shown in

$$p(x_{k-1}|u_{1:k-1}, y_{1:k-1}) = \sum_{i=1}^{n_{PF}} W_{k-1}^i \delta(x - x_{k-1}^i). \quad (20)$$

After processing several data snapshots, a PF often suffers from a degeneracy problem (i.e., the weight of only one particle tends to one while weights of all other particles tend to 0). To reduce the degeneracy problem, a resampling step is often added to redisperse the discrete samples by generating a new set of particles according to the discrete distribution of (20). To detect degeneracy, the effective sampling size N_{eff} is defined by (23). The basic PF process is described by the following equations [20].

PF Prediction

$$x_k^{i-} = f(x_{k-1}^i, u_{k-1}) + w_{k-1}^i. \quad (21)$$

PF Correction

$$\tilde{W}_k^i = W_{k-1}^i \cdot p_{v_k}(z_k | h(x_k^{i-}, u_k)) \quad (22a)$$

$$W_k^i = \frac{\tilde{W}_k^i}{\sum_{j=1}^{n_{PF}} \tilde{W}_k^j} \quad (22b)$$

$$x_k = \sum_{i=1}^{n_{PF}} x_k^{i-} W_k^i. \quad (22c)$$

PF Resampling if degeneracy is detected using

$$N_{\text{eff}} \approx \frac{1}{\sum_{i=1}^{n_{PF}} (W_k^i)^2}. \quad (23)$$

Here, \tilde{W}_k^i is prior weights of the i th state sample. $p_{v_k}(z_k | h(x_k^{i-}, u_k))$ is the likelihood of z_k given the prior states x_k^{i-} and inputs u_k . The likelihood function is determined by the measurement noise model (i.e., R_d). Symbol n_{PF} is the total number of samples that are used to represent the probability distribution of a state.

IV. STUDY APPROACHES

Because each nonlinear filtering algorithm often has its advantages and drawbacks, it is important to set up a specific case in a power system for comparison. To mimic the responses of a real system, the power system toolbox (PST) [22] was selected to generate simulation data. The two-area-four-machine test system shown in Fig. 1 (stored as d2asbeghp.m in PST) is used to generate the system dynamic responses to a three-phase fault. The fault is applied to bus 3

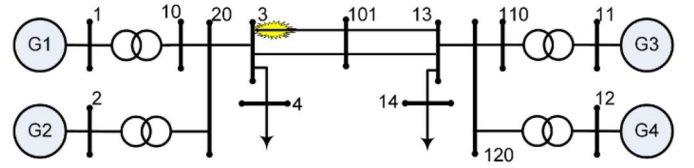


Fig. 1. Two-area four-machine system [22].

on the line between buses 3 and 101 at 6.1 s. The fault is cleared at 6.15 s at bus 3 and 6.20 s at bus 101. To capture the dynamics and reduce integration errors, the simulation time step is chosen to be 0.001 s.

Assume that PMUs are available at the generator bus 1 to measure the voltage and the current phasors. Filtering algorithms are set up to estimate the dynamic states of generator 1. To mimic the measurements from PMUs, the system responses are down-sampled to a rate of 25 samples per second. According to IEEE Standard C37.118-2005 [21], a certain percentage of total vector errors (TVE) are added to the system responses to mimic measurement noise.

Because of the randomness of the measurement and process noise, the performance of an algorithm needs to be evaluated under a statistical framework. To circumvent the difficulties of analytical solutions, the paper uses the Monte Carlo method to study the statistical accuracy. The Monte Carlo method generates various instances of random noise by randomly sampling noise's distribution. Through repeated samples and simulations, the probabilistic distribution of estimated states can be approximated by the resulting samples.

To compare the estimation accuracy of a filtering algorithm, MSE defined by (24) is used as a metric. To increase the dynamic range, $10 \log_{10}(\text{MSE})$ in dB is used in the studies of the following section. Here, the symbol $x_{k, \text{True}}$ represents the true state at k th time-step, while $\hat{x}_{k, m}$ is the corresponding estimated states in the m th Monte Carlo test case. The symbol, M , is the total number of Monte Carlo trials

$$\text{MSE}(\hat{x}_k) \triangleq \frac{1}{M} \sum_{m=1}^M (\hat{x}_{k, m} - x_{k, \text{True}})^2. \quad (24)$$

Note that M should be large enough to make estimated MSEs converge and small enough to avoid unnecessary computational burden [26]. To determine a proper M , $M = 10, 50, 100, 200, 400$ were tried. It was observed that the estimated MSEs did not change significantly after $M \geq 50$ for the studied problems. The observation indicated that the MSE estimation converged after 50 trials. Therefore, to leave some safety margins, the paper uses $M = 100$.

Computation time is another important factor to evaluate the performance of a filtering algorithm. For example, for a real time control application, it is often required that the estimation of current states must be finished before the next measurement arrives. In another word, a state estimation algorithm must be able to keep up with the measurement data flow. A computationally efficient algorithm uses less time to finish state estimation and is therefore, preferred because it lowers the computation costs. To compare the computational time, all the algorithms are implemented using MATLAB version 2013a and tested on a computer with a 3.4-GHz processor,

16 GB of memory, and a 64-bit operating system. The computation timer is started right before the prediction step of the first measurement and ended right after the correction step of the last measurement.

V. CASE STUDIES

In this section, dynamic simulation is carried out to compare the performance of the EKF, UKF, EnKF, and PF for the purpose of estimating dynamic states of a power system.

On the algorithm side, both EnKF and PF use samples to represent state probability distribution. To evaluate the influence of the sample number on MSEs, 200 and 2000 samples were used for testing these algorithms.

On the simulation side, four scenarios are set up to understand major influential factors.

- A simple scenario is set up as a benchmark for comparison. It is assumed that the system models and noise models are known. All input and output measurements are all available.
- A scenario with measurement interpolation is set up to evaluate how the interpolation method [3], [11] may influence the algorithms' performance. Therefore, all its setups mirror scenario (a). In addition, PMU measurements are interpolated.
- A realistic scenario considering model inadequacy is set up to understand how algorithms may perform in a more realistic scenario. In this setup, a more complex model is used to generate simulation data while a simpler model is used in filtering algorithm to estimate states. The setup is to mimic the more common reality where an available model is a simplified representation of a real system. In addition, higher levels of measurement noise are added to mimic noise introduced by potential transformers and current transformers.
- A realistic scenario with missing data and outliers is set up to understand the influence of missing data and outliers on the algorithm performance. This setup mirrors scenario (c). In addition, missing data and outliers are intentionally added to PMU measurement.

The four scenarios are described in detail in Section V-A–V-D. The algorithm estimation accuracy are compared using MSEs in dB in each subsection. The computational time is summarized in Section V-E.

A. Simple Scenario

The simulations are set up as follows.

- To simulate the generators' dynamic responses, a fourth-order transient model as shown in (1) is used. Governors and exciters are included. Sub-transient dynamics and field saturation effects are not modeled.
- For the PMU measurements, 1.0% measurement noise in TVE is added to the voltage and current phasors. T_m and E_{fd} are recorded with 1.0% measurement noise. The PMU sampling rate of 25 samples per second used for generating the measurement data is also used for estimation.
- For all the algorithms, the initial states are estimated by setting $\dot{x} = 0$ in (2) and then solving (2)

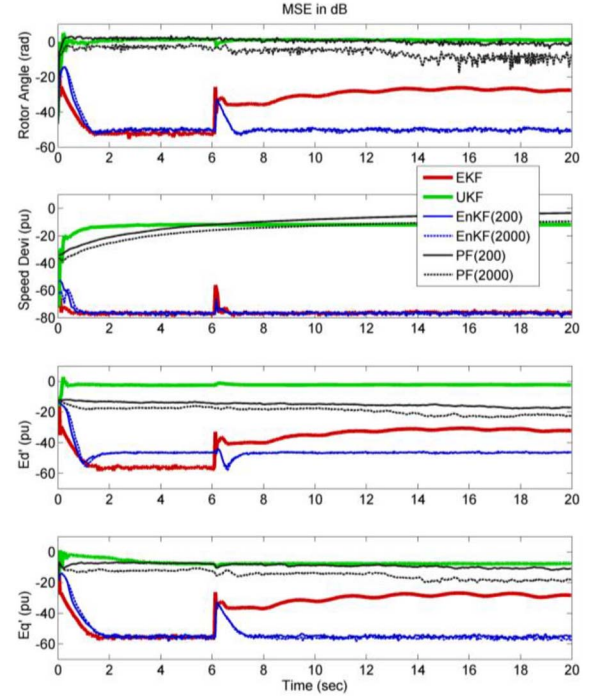


Fig. 2. Comparison of MSEs from the EKF, EnKF, PF, and UKF for 100 sets of Monte Carlo simulations of a simple scenario.

using the Gauss–Seidel method. To reflect uncertainty of the initial states, covariance P_0 is set to be ten times of the largest changes of the states, i.e., $P_0 = (10\max(\text{abs}(\text{diff}(x_{1:N}))))^2$. The output variance R_d is set to be 0.01^2 corresponding to the 1.0% of errors added, which is $\text{diag}([0.01, 0.01])^2$ for this paper. Q_d is set to be 1.0% the largest changes of the states, i.e., $Q_d = 1.0\% * \max(\text{abs}(\text{diff}(x_{1:N})))$, which is $\text{diag}([0.0474, 0.0042, 0.0289, 0.0137])^2$ for this paper.

The MSEs of the four states from EKF, UKF, EnKF, and PF are summarized in Fig. 2. It can be observed from Fig. 2 that the EnKF has the smallest MSE. Increasing the number of samples in EnKF does not significantly influence its estimation accuracy. In contrast, the MSEs of the PF noticeably decrease when the sample number is increased from 200 to 2000. UKF and PF have larger MSEs than the other methods.

Fig. 3 shows the EnKFs estimation results for all 100 sets of Monte Carlo simulation with $n_{pf} = 200$ samples. All the 100 EnKF estimates converge to the true states within 1.5 s. Note that to help the illustration, the true value of generator four rotor angle is used as the reference angle to generate the first plot of Fig. 3. For the PF with $n_{pf} = 2000$ samples, 80 sets out of the 100 PF estimates converged and the other 20 sets diverged. When the sample number of the PF is reduced from 2000 to 200, the number of converged sets decreases to 47. For the UKF, only 67 sets of estimates converged. For the EKF, all 100 estimates converged, but they have larger MSEs than the EnKF.

B. Scenario With Measurement Interpolation

The goal of this scenario is to evaluate how the interpolation method [3], [11] may influence the algorithms'

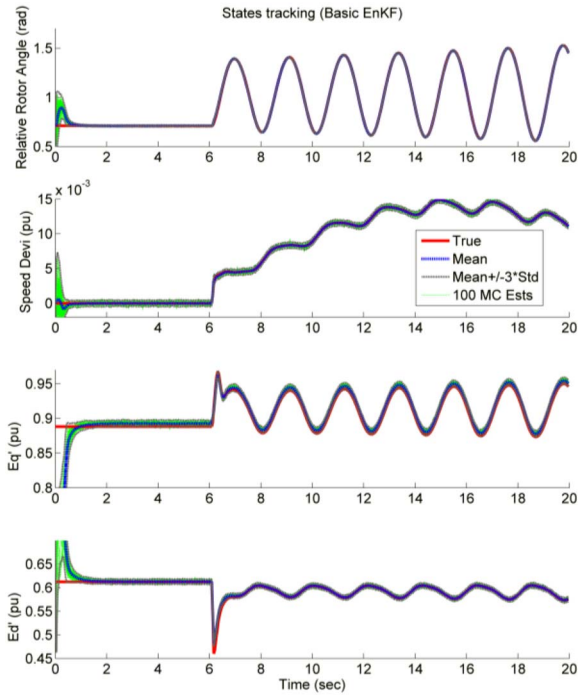


Fig. 3. Estimated states from the EnKF with 200 samples of 100 sets of Monte Carlo simulations for a simple scenario. (All 100 sets converge.)

performance. The interpolation method inserts the additional pseudo measurement points between two consecutive measurement samples. Introducing additional measurements increases the effective sampling rate, and reduces the linearization errors. In this scenario, the sampling rate is increased from 25 sample/s to 200 sample/s by adding seven additional pseudo measurements between every two measurement points through linear interpolation.

Note that because of the interpolation, the sampling time interval Δt in (7) is reduced from 40 ms to 5 ms. Following (10), the process noise covariance Q_d is reduced to 1/8 of that in scenario (a). The R_d remains the same as in (12). All the other setup remains the same as that in scenario (a).

The MSEs from the EKF, UKF, EnKF, and PF are compared in Fig. 4 between the cases with interpolation as in this scenario and the cases without interpolation as in scenario (a). It can be observed that the MSEs of the EKF, UKF, and PF are significantly reduced by the interpolation method. In comparison, changes of MSEs for the EnKF are less significant. In addition, with interpolation, the MSEs of the EKF, UKF, and EnKF are significantly smaller than those of the PF. With the interpolation method, all 100 sets of EKF, UKF, EnKF, and PF estimates converge.

C. Realistic Scenario Considering Model Inadequacy

In this scenario, the simulation model for generating measurement data is more complex than the estimation model used for estimating state. The goal is to mimic the more common reality where an available model is often a simplified representation of a real system. In addition, the measurement noise level is increased to include transformer noise. Finally, some

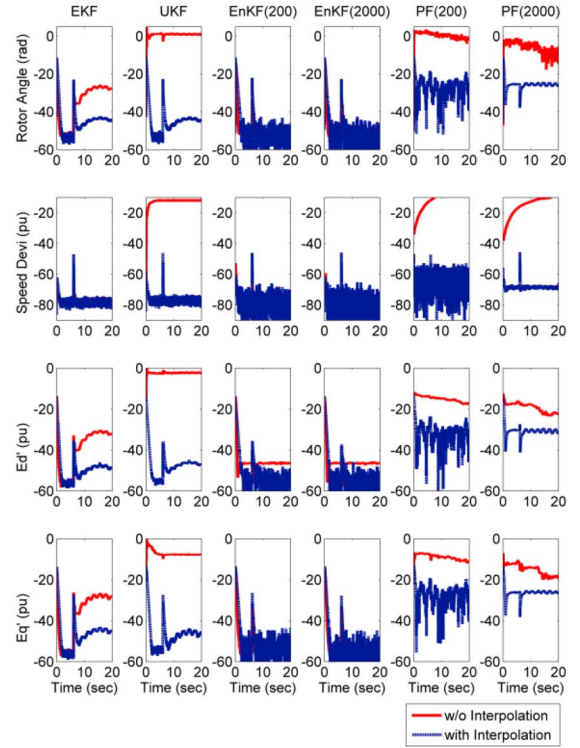


Fig. 4. Comparison of MSEs from the EKF, UKF, EnKF, and PF of 100 sets of Monte Carlo simulations for the scenarios with measurement interpolation and without interpolation. (Note: some of the curves for UKF and PF without interpolation are out of the charts. Refer to Fig. 2 for details.)

model inputs (e.g., T_m and E_{fd}) have to be estimated because they are not usually measured by PMUs.

- 1) To simulate generator responses, a sub-transient model is used. Field saturation effects are modeled by adding $S_1 = 0.0654$, $S_2 = 0.4786$ [13]. To increase oscillatory dynamics, the power system stabilizers are intentionally removed. By adding the sub-transient model and saturation effects, the simulation model is more complex than the estimation model shown in (1).
- 2) For the PMU measurements, 5% of measurement noise in TVE is added to the voltage and current phasors. Note that additional noise is added to include measurement noise introduced by current transformers and potential transformers. In addition, because PMUs may not always be available near a generator, it is assumed that E_{fd} and T_m are not measured in this scenario. E_{fd} is estimated as a special state using the state augmenting method proposed in [3]. Here, $E_{fd,k} = E_{fd,k-1} + w_{E_{fd},k-1}$ is used as the state transition model and initial value is set to be 1. T_m is estimated by low pass filtering P_{eff} [23]. Here, P_e is the real electric power and f is the system frequency. The low pass filter is a first order infinite impulse response filter with a time constant of 0.5 min. (Note that if the E_{fd} and/or T_m are measured, the measured values can replace the estimated values.)
- 3) For all the algorithms, the setup is same as that in scenario (b) (e.g., the data sampling rate is increased to 200 samples/s through linear interpolation), except that the output variance R_d is set to be $\text{diag}([0.05, 0.05])^2$ because the 5.0% measurement noise was added.

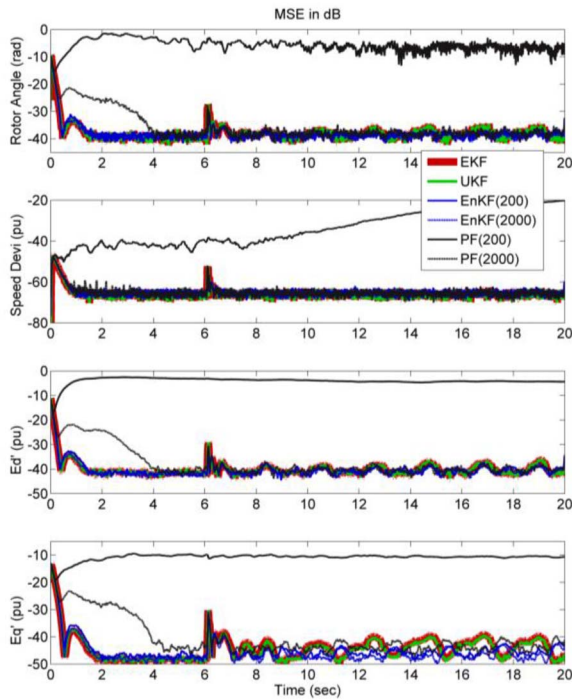


Fig. 5. Comparison of MSEs from the EKF, UKF, EnKF, and PF of 100 sets of Monte Carlo simulations for the realistic scenario.

The MSEs from the EKF, UKF, EnKF, and PF are summarized in Fig. 5. It can be observed that the EnKF, EKF, and UKF have similar MSEs. After approximately 4 s the MSEs of the PF with 2000 samples converge to similar levels. In contrast, the MSEs of the PF with 200 samples are persistently the largest among all the algorithms, indicating a performance degradation. All 100 estimates from EKF, UKF, and EnKF converge. All 100 estimates from PF with 2000 samples converge. In contrast, there are five sets of estimates that diverge for the PF using 200 samples.

Under this scenario, simulations are also performed with 1% and 3% measurement noise. It is observed that when the noise level is reduced, the MSEs get smaller.

D. Realistic Scenario With Missing Data and Outliers

The goal of this scenario is to evaluate all the algorithms when the PMU measurements are corrupted by missing data and outliers. Missing data may be caused by temporary communication failures, and is often identified via error detection schemes associated with communication protocols. Outliers are measurements with significantly large measurement errors that may be caused by extraordinary disturbances or temporary sensor failures. By checking the residuals, an outlier detector may be able to detect some outliers. Yet, because such an outlier detector needs to maintain a balance between correct and erroneous determinations (false positives and false negatives), it will not be able to detect all outliers. Undetectable outliers (false negatives) carry misleading information and presented a major challenge to state estimation algorithms.

To simulate missing data, all (simulated) measurement data between the 7th and 8th second is chosen to be missing and marked out. The missing data is then patched through linear

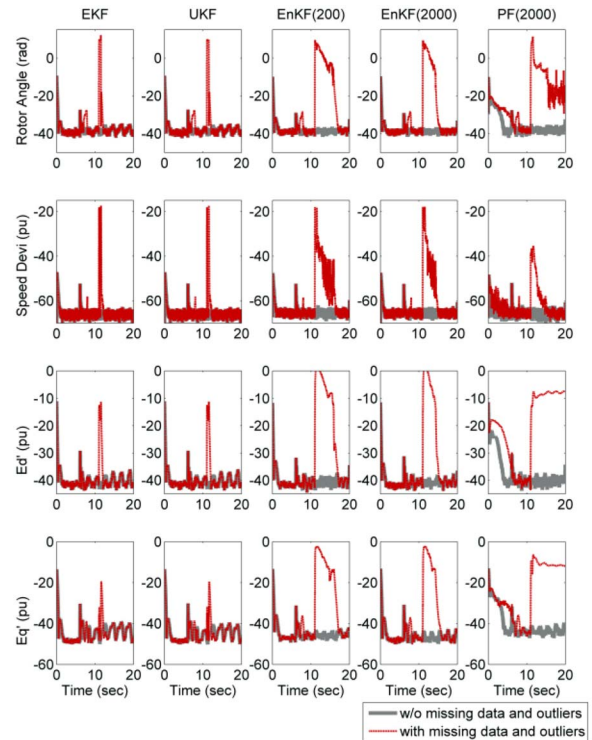


Fig. 6. Comparison of MSEs from the EKF, UKF, EnKF, and PF of 100 sets of Monte Carlo simulations for the realistic scenario with and without missing data and outliers.

interpolation. To mimic outliers, errors on the order of 12 times of the standard deviations of the voltage magnitudes and angles are added to the voltage phasor measurements between 11.0 and 11.5 s. These outliers are assumed to be undetected (false negatives) and are processed as normal data by all the filtering algorithms. The remaining setup is the same as that of scenario (c).

The MSEs of the EKF, UKF, EnKF, and PF are compared in Fig. 6 between the cases with missing data and outliers as in this scenario, and the cases without missing data and outliers as in scenario (c). It can be observed that the MSEs of all the algorithms are very similar for cases with and without missing data. This indicates that all algorithms are robust against missing data that last for 1 s.

On the other hand, the outliers at the 11th second cause significant increases in the MSEs of all the algorithms. After the outliers disappear at the 11.5 s, the MSEs of the EKF, UKF return to the original levels quickly, while it takes more time for the MSEs of the EnKF to return to the original levels. Note that EnKF (2000) converge faster than the EnKF (200) because of more samples used. For the PF (with 2000 samples), five out of 100 Monte Carlo trials diverged after the outliers occur. The observation indicates that the EKF and UKF are more robust to outliers compared to the EnKF and PF.

Fig. 7 shows that all 100 sets of EnKF estimates converge to the true states. Also, the estimates of the EKF and UKF for all 100 sets of data converge.

E. Computation Time

For the 20 s of simulated measurements, the computational time for different algorithms and scenario (a)–(c) are

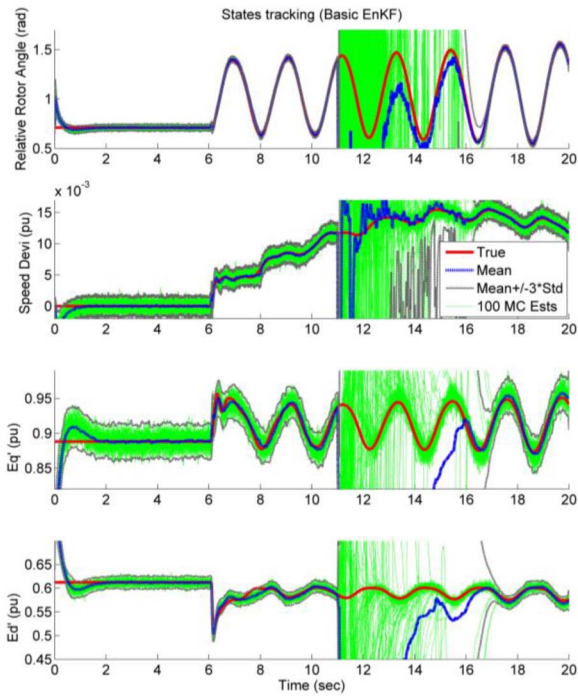


Fig. 7. Estimated states from the EnKF of 100 sets of Monte Carlo simulations for the realistic scenario with missing data and outliers. (All 100 sets converge.)

summarized in Fig. 8. Here, EnKF (200) and EnKF (2000) are for the EnKF algorithm with 200 and 2000 samples respectively. For scenario (d), the computation time is virtually same as that of scenario (c) and, therefore, not shown in the figure to be concise.

For each individual scenario, it can be observed that the EKF is the most efficient algorithm in computation time. The UKF is the second most efficient algorithm and its computation time is of the same order as the EKF. The EnKF (200) and the PF (200) use similar computation time. They both use about 8 ~ 10 times as much computation time as the EKF. The EnKF (2000) and the PF (2000) use the longest computation time. The EnKF (2000) and the PF (2000) have ten times as many samples as the EnKF (200) and the PF (200). As a result, they use almost ten times as much computation time as the EnKF (200) and PF (200). The observation indicates that the computation time of the EnKF and the PF is proportional to the number of samples used.

Comparison can also be made among different scenarios. Scenario (b) has eight times as many measurement points (by including seven additional interpolation points) as scenario (a). For the same algorithm, the computation time under scenario (b) is about eight times as long as that under scenario (a). Because algorithms under scenario (c) add an additional state for E_{fd} and need to estimate T_m , the computation time for the same algorithm is a little bit longer than it is under scenario (b).

To run an algorithm in real time, the computation time must be less than the measurement time duration, i.e., 20 s. It can be observed that EKF and UKF can run in real time under all the studied scenarios. EnKF (200) and PF (200) can only be run

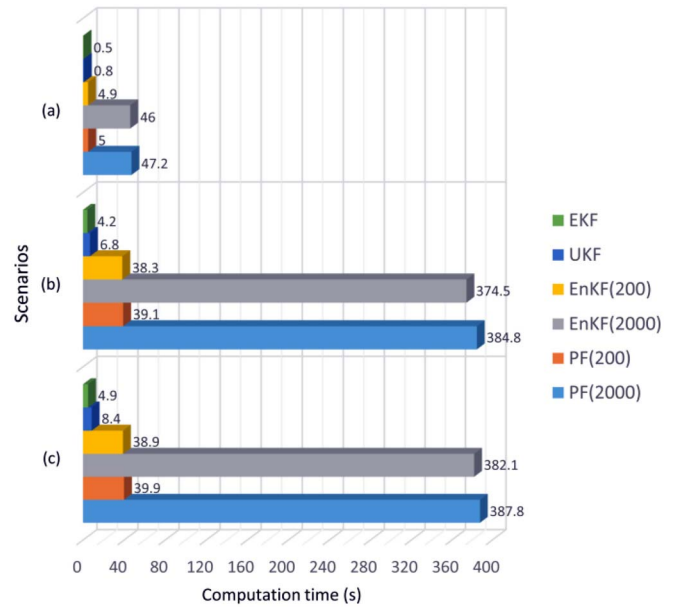


Fig. 8. Computation time of the EKF, UKF, EnKF, and PF under scenario (a)–(c).

TABLE I
PERFORMANCE COMPARISON AMONG THE EKF, UKF, ENKF, AND PF
(N/A: NOT APPLICABLE)

	EKF	UKF	EnKF	PF
Accuracy under scenario (a)	The 2 nd best with 0% diverged	33% diverged	The best with 0% diverged	20% diverged (PF 2000)
Efficacy of interpolation	High	High	Low	High
Number of samples needed	N/A	small	medium	large
Sensitivity to the missing data	Low	Low	Low	Low
Sensitivity to the outliers	Low	Low	Medium	High
Computation time	Shortest	Same order as EKF	longer than EKF	Same order as EnKF

in real time under scenario (a). EnKF (2000) and PF (2000) cannot run in real time under any studied scenarios.

VI. CONCLUSION

Accurate information about dynamic states is critical to efficient control of a power system, especially with the increasing complexity resulting from uncertainties and stochastic variations introduced by intermittent renewable energy sources,

responsive loads, mobile consumption of plug-in vehicles, and new market designs. Using a statistical framework, this paper compares the performance of an EKF, UKF, EnKF, and PF for the purpose of estimating dynamic states from real-time phasor measurements. To summarize the observations from the simulation using a two-area-four-machine test system, Table I is constructed for quick comparison. The following is shown.

- 1) The EnKF algorithms are more accurate than other algorithms when the typical PMU sampling rate is used for estimation.
- 2) Measurement interpolation methods can improve the estimation accuracy of the EKF, UKF, and PF at the expense of increasing computation time. The interpolation does not show significant influence on the estimation accuracy of the EnKF.
- 3) The PF needs more samples than the EnKF to achieve a reasonably good estimation accuracy. Increasing the number of samples can improve the estimation accuracy and convergence of the PF.
- 4) All of the algorithms are robust to missing data. The outliers cause some significant errors for all algorithms if the outliers are processed as normal data. The EKF and the UKF are more robust to the outliers than the EnKF and the PF. It takes longer time for an EnKF to regain accurate state tracking after being disrupted by the outliers. About 5% of trials in the PF diverged because of the outliers.
- 5) The EKF and the UKF are computationally more efficient than the EnKF and the PF for the studied problem.

The proposed algorithms are applied to the system responses to other faults with different durations and locations. The estimation accuracy and computation time are similar to the presented cases.

Based on the above observations, some recommendations can be made for selecting state estimation algorithms for the studied problem.

- 1) To reach a reasonably good accuracy, the PF needs a large number of samples. As a result, the PF is computationally expensive and is not suitable for estimating dynamic states in real time. However, it should be noted that there are still great potentials to further improve PF accuracy by increasing the effectiveness of samples and reduce computation time through parallel computation.
- 2) Using limited number of samples (i.e., 200), the EnKF can achieve a reasonably good estimation accuracy without relying on the measurement interpolation method. This observation indicates that by using the sample covariance, the EnKF are more stable than the EKF and UKF under lower sampling rate. According to [19], the computation time advantages of the EnKF over the EKF may become obvious when number of states is larger than the number of the EnKF samples. Because the studied problem involves only 4–5 states (much less than those in [19]), the computation time of EnKF is longer than that of the EKF and UKF. The EnKF (200) can still run in real time under scenario (a) but is more sensitive to the outliers than the EKF and UKF.

- 3) The EKF and UKF should be used with the measurement interpolation method. Adding pseudo measurements, the measurement interpolation method improves estimation accuracy of the UKF and EKF by reducing the system nonlinearity and process noise levels. Although the additional pseudo measurements increase computation burden, the EKF and UKF can still run in real time because of their high computational efficiency. In addition, the EKF and UKF reconverge to the true states more quickly than the EnKF after being disrupted by the outliers. Therefore, the EKF and UKF with the measurement interpolation method are recommended for the studied problems.

Ongoing and future work includes dynamic state estimation methods at system levels and sensitivity studies to determine how parameter errors may influence the state estimation, as well as efficient, accurate, and flexible methods for estimating the states in both real time and offline environments.

REFERENCES

- [1] P. DeRusso, R. Roy, C. Close, and A. A. Desrochers, *State Variables for Engineers*, 2nd ed. Hoboken, NJ, USA: Wiley, 1998.
- [2] M. A. Pai, D. P. Sen Gupta, and K. R. Padiyar, *Small Signal Analysis of Power Systems*. Harrow, U.K: Alpha Science International, 2004.
- [3] Z. Huang, P. Du, D. Kosterev, and B. Yang, "Application of extended Kalman filter techniques for dynamic model parameter calibration," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, Calgary, AB, Canada, Jul. 2009, pp. 1–8.
- [4] Z. Huang, K. Schneider, and J. Nieplocha, "Feasibility studies of applying Kalman filter techniques to power system dynamic state estimation," in *Proc. 8th Int. Power Eng. Conf. (IPEC)*, Singapore, Dec. 2007, pp. 376–382.
- [5] Z. Huang, P. Du, D. Kosterev, and S. Yang, "Generator dynamic model parameter calibration using phasor measurements at the point of connection," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1939–1949, May 2013.
- [6] L. Fan and Y. Wehbe, "Extended Kalman filtering based real-time dynamic state and parameter estimation using PMU data," *Elect. Power Syst. Res.*, vol. 103, pp. 168–177, Oct. 2013.
- [7] E. Ghahremani and I. Kamwa, "Dynamic state estimation in power system by applying the extended Kalman filter with unknown inputs to phasor measurements," *IEEE Trans. Power Syst.*, vol. 26, no. 4, pp. 2556–2566, Nov. 2011.
- [8] E. Ghahremani and I. Kamwa, "Online state estimation of a synchronous generator using unscented Kalman filter from phasor measurements units," *IEEE Trans. Energy Convers.*, vol. 26, no. 4, pp. 1099–1108, Dec. 2011.
- [9] N. Zhou, Z. Huang, Y. Li, and G. Welch, "Local sequential ensemble Kalman filter for simultaneously tracking states and parameters," in *Proc. IEEE North Amer. Power Symp. (NAPS)*, Champaign, IL, USA, Sep. 2012, pp. 1–6.
- [10] N. Zhou, D. Meng, and S. Lu, "Estimation of the dynamic states of synchronous machines using an extended particle filter," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4152–4161, Nov. 2013.
- [11] Y. Huang, G. V. Reklaitis, and V. Venkatasubramanian, "A heuristic extended Kalman filter based estimator for fault identification in a fluid catalytic cracking unit," *Ind. Eng. Chem. Res.*, vol. 42, no. 14, pp. 3361–3371, May 2003.
- [12] P. Kundur, *Power System Stability and Control*. New York, NY, USA: McGraw-Hill, 1994.
- [13] J. Chow and G. Rogers, *User Manual for Power System Toolbox*, Version 3.0, pp. 79–80, 1991–2008.
- [14] Wikipedia. (2013, Nov. 16). *Discretization* [Online]. Available: <http://en.wikipedia.org/wiki/Discretization>
- [15] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.

- [16] G. Welch and G. Bishop. (1995). "An introduction to the Kalman filter," Dept. Comput. Sci., Univ. North Carolina at Chapel Hill, Tech. Rep. TR 95-041 [Online]. Available: <http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html>
- [17] K. Shih and S. Huang, "Application of a robust algorithm for dynamic state estimation of a power system," *IEEE Trans. Power Syst.*, vol. 17, no. 1, pp. 141–147, Feb. 2002.
- [18] E. A. Wan and R. van der Merwe, "The unscented Kalman filter," in *Kalman Filtering and Neural Networks*, S. Haykin, Ed. Hoboken, NJ, USA: Wiley, 2001.
- [19] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics," *J. Geophys. Res.*, vol. 99, no. C5, pp. 143–162, May 1994.
- [20] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- [21] K. E. Martin *et al.*, "Exploring the IEEE Standard C37.118-2005 synchrophasors for power systems," *IEEE Trans. Power Del.*, vol. 23, no. 4, pp. 1805–1811, Oct. 2008.
- [22] J. H. Chow and K. W. Cheung, "A toolbox for power system dynamics and control engineering education and research," *IEEE Trans. Power Syst.*, vol. 7, no. 4, pp. 1559–1564, Nov. 1992.
- [23] P. Tripathy, S. C. Srivastava, and S. N. Singh, "A divide-by-difference filter based algorithm for estimation of generator rotor angle utilizing synchrophasor measurements," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 6, pp. 1562–1570, Jun. 2010.
- [24] J. J. Simon and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.
- [25] E. Farantatos, G. K. Stefopoulos, G. J. Cokkinides, and A. P. Meliopoulos, "PMU-based dynamic state estimation electric power systems," in *Proc. Power Energy Soc. Meeting (PES)*, Calgary, AB, Canada, Jul. 2009, pp. 1–8.
- [26] M. Fan, V. Vittal, G. T. Heydt, and R. Ayyanar, "Probabilistic power flow studies for transmission systems with photovoltaic generation using cumulants," *IEEE Trans. Power Syst.*, vol. 27, no. 4, pp. 2251–2261, Nov. 2012.
- [27] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F Radar Signal Process.*, vol. 140, no. 2, pp. 107–113, Apr. 1993.

Ning Zhou (S'01–M'05–SM'08) received the Ph.D. degree in electrical engineering with a minor in statistics from the University of Wyoming, Laramie, Wyoming, USA, in 2005.

He is an Assistant Professor with the Electrical and Computer Engineering Department, Binghamton University, Binghamton, NY, USA. He was with the Pacific Northwest National Laboratory, Richland, WA, USA, as a Power System Engineer from 2005 to 2013. His current research interests include power system dynamics and statistical signal processing.

Dr. Zhou is a Senior Member of the IEEE Power and Energy Society.

Da Meng (M'13) received the B.Eng., M.Eng., and Ph.D. degrees in electrical engineering from Xi'an Jiaotong University, Xi'an, China, the Chinese Academy of Sciences, Beijing, China, and Washington State University, Pullman, WA, USA, in 1997, 2000, and 2009, respectively.

He is currently with the Pacific Northwest National Laboratory, Richland, WA, USA, as a Power System Engineer. His current research interests include power system dynamics, power system voltage stability analysis, and integration of renewable resources.

Zhenyu Huang (M'01–SM'05) received the B.Eng. degree from the Huazhong University of Science and Technology, Wuhan, China, in 1994, and the Ph.D. degree from Tsinghua University, Beijing, China, in 1999.

From 1998 to 2002, he conducted extensive research at McGill University, Montreal, QC, Canada; the University of Alberta, Edmonton, AB, Canada; and the University of Hong Kong, Hong Kong. He is currently a Chief Engineer and a Team Lead at the Pacific Northwest National Laboratory, Richland, WA, USA. His current research interests include high performance computing, phasor technology, and power system stability and simulation.

Dr. Huang was the recipient of the 2009 IEEE Power and Energy Society (PES) Outstanding Young Engineer Award. He is the Co-Chair of the IEEE PES Task Force on High Performance Computing for Power Grid Analysis and Operation. He has led the PES Richland Chapter to win the 2007 IEEE PES Outstanding Small Chapter Award. He is a registered Professional Engineer in the state of Washington.

Greg Welch (M'99) received the B.S. degree (with highest distinction) in electrical technology from Purdue University, West Lafayette, IN, USA, in 1986, and the M.S. and the Ph.D. degrees in computer science from the University of North Carolina (UNC) Chapel Hill, Chapel Hill, NC, USA, in 1995 and 1997, respectively.

He was a Research Professor at UNC, and was with the Jet Propulsion Laboratory, NASA, Pasadena, CA, USA, and with the Defense Systems Division, Northrop Grumman, Herndon, VA, USA. He is currently the Florida Hospital Endowed Chair in Healthcare Simulation at the University of Central Florida, Orlando, FL, USA, with appointments at the College of Nursing, the College of Engineering and Computer Science, and the Institute for Simulation and Training. His current research interests include stochastic estimation, human-tracking systems, 3-D telepresence, projector-based graphics, and medical applications of computers.