

## Wrong transition and measurement models in power system state estimation

PIOTR KOZIERSKI<sup>1,3</sup>, MARCIN LIS<sup>2,4</sup>, DARIUSZ HORLA<sup>1</sup>

<sup>1</sup>*Faculty of Electrical Engineering, Institute of Control and Information Engineering*

<sup>2</sup>*Faculty of Electrical Engineering, Institute of Electrical Engineering and Electronics*

<sup>3</sup>*Faculty of Computing, Chair of Control and Systems Engineering*

*Division of Signal Processing and Electronic Systems*

<sup>1,2,3</sup>*Poznan University of Technology*

*Piotrowo 3a, 60-965 Poznań, Poland*

<sup>4</sup>*Spirvent sp. z o.o.*

*Polna 2C, 64-600 Oborniki, Poland*

*e-mail: piotr.kozierski@gmail.com, mail.dla.studenta@gmail.com*

(Received: 30.07.2015, revised: 03.04.2016)

**Abstract:** The influence of wrong information about transition and measurement models on estimation quality has been presented in the paper. Two methods of a particle filter, with and without the Population Monte Carlo modification, and also the extended and unscented Kalman filters methods have been compared. A small 5-bus power system has been used in simulations, which have been performed based on one data set, and this data set has been chosen from among 100 different – to draw the most general conclusions. Based on the obtained results it has been found that for the particle filter methods the implementation of the slightly higher standard deviation than the true value, usually increases the estimation quality. For the Kalman filters methods it has been concluded that optimal values of variances are equal to the true values.

**Key words:** particle filter, estimation quality, Population Monte Carlo

### 1. Introduction

Each measurement is associated with a measurement error. In many cases, it is assumed that the density distributions of measurement noises and internal (also called system or transition) noises are known. E.g. in the article [1] the extended Kalman filter method has been used to the power system dynamic state estimation (PSDSE) task. In this specific case estimation relates to the dynamics of power network, i.e. the state vector is composed of phase angles and angular velocities of generator rotors. The authors assumed that variances of both, the internal (transition) and measurement noises, are known.

In other papers related to PSDSE, such as [2-3], the knowledge about these distributions also has been assumed, however simultaneously it has been assumed that values of parameters

in state equations (in functions of state variables depending on the state variables from previous time step) are unknown.

But there are also articles, where noise densities are estimated together with the state variables. E.g. in [4] the author estimates not only the angle and velocity of Permanent Magnet Synchronous Motor (PMSM), but also the variances of measured currents.

In this article it has been decided to focus on cases with preset measurement and transition models, but with assumption that these models are incorrect. It has been checked how too small or too large variance impacts on the estimation quality.

A few different methods have been compared, i.e. a basic particle filter (Bootstrap Filter), the particle filter with the Population Monte Carlo modification, the extended Kalman filter (EKF) and the unscented Kalman filter (UKF). The description of the two latter methods can be found in [1, 3].

In the next Section the particle filter has been described. The Population Monte Carlo modification has been presented in the third Section. In the fourth Section an object, i.e. the power system has been presented. The next Section contains the methodology and the results obtained in simulations, which have been concluded in the last Section.

This paper is the extended version of the article presented during the national conference PES-10 (*Postepy w Elektrotechnice Stosowanej*), which will be printed in the *Prace Instytutu Elektrotechniki* journal.

## 2. Particle filter

The particle filter (PF) principle of the operation is based on the recursive Bayesian filter [5]

$$\underbrace{p(\mathbf{x}^{(k)} | \mathbf{Z}^{(k)})}_{\text{Posterior}} = \frac{\overbrace{p(\mathbf{z}^{(k)} | \mathbf{x}^{(k)})}^{\text{Likelihood}} \overbrace{p(\mathbf{x}^{(k)} | \mathbf{Z}^{(k-1)})}^{\text{Prior}}}{\underbrace{p(\mathbf{z}^{(k)} | \mathbf{Z}^{(k-1)})}_{\text{Evidence}}}, \quad (1)$$

where  $\mathbf{x}^{(k)}$  is the state vector at the  $k$ -th time step,  $\mathbf{z}^{(k)}$  is a vector of measurements, and  $\mathbf{Z}^{(k)}$  is the set of the measurement vectors from the beginning to the  $k$ -th time step

$$\mathbf{Z}^{(k)} = \{\mathbf{z}^{(1)} \quad \mathbf{z}^{(2)} \quad \dots \quad \mathbf{z}^{(k)}\}. \quad (2)$$

Additionally both vectors,  $\mathbf{x}^{(k)}$  and  $\mathbf{z}^{(k)}$ , should be considered as a realization of random variables  $\mathbf{X}$  and  $\mathbf{Z}$  with specified probability density functions (PDFs) [6]

$$\mathbf{x}^{(k)} \sim p(\mathbf{x}^{(k)} | \mathbf{x}^{(k-1)}), \quad (3)$$

$$\mathbf{z}^{(k)} \sim p(\mathbf{z}^{(k)} | \mathbf{x}^{(k)}), \quad (4)$$

where conditional PDF in (3) is the transition model, and the measurement model is presented in (4).

PF is a one of possible implementations of the Bayes filter, in which continuous posterior PDF has been replaced by a set of particles. It is assumed that each  $i$ -th particle in the  $k$ -th time step has some value  $\mathbf{x}^{i,(k)}$  (state vector) and weight  $q^{i,(k)}$ . For particles with a higher weight there is a higher probability that value  $\mathbf{x}^{i,(k)}$  is closer to the true state vector. Hence, the posterior PDF in PF is represented by

$$\hat{p}(\mathbf{x}^{(k)} | \mathbf{Z}^{(k)}) = \sum_{i=1}^N q^{i,(k)} \delta_D(\mathbf{x}^{(k)} - \mathbf{x}^{i,(k)}), \quad (5)$$

where  $N$  is the number of particles, and  $\delta_D$  is the Dirac delta.

If the number of particles  $N$  is high enough, the information contained in the set of particles is almost the same as in the continuous posterior PDF

$$p(\mathbf{x}^{(k)} | \mathbf{Z}^{(k)}) \stackrel{N \rightarrow \infty}{=} \hat{p}(\mathbf{x}^{(k)} | \mathbf{Z}^{(k)}). \quad (6)$$

The first PF was proposed by Gordon, Salmond and Smith [7] in 1993, and was called Bootstrap Filter. Its operation principle has been presented in Algorithm 1.

**Algorithm 1:** Bootstrap filter

- 1) Initialization. Draw  $N$  particles from the initial PDF  $\mathbf{x}^{i,(0)} \sim p(\mathbf{x}^{(0)})$ , set the initial time step  $k = 1$ .
- 2) Prediction. Draw  $N$  new particles from the transition model  $\mathbf{x}^{i,(k)} \sim p(\mathbf{x}^{(k)} | \mathbf{x}^{i,(k-1)})$ .
- 3) Update. Calculate particle weights based on the measurement model  $q^{i,(k)} \propto p(\mathbf{z}^{(k)} | \mathbf{x}^{i,(k)})$ .
- 4) Normalization. Normalize weights so that their sum is equal to 1.
- 5) Resampling. Draw  $N$  new particles based on the posterior PDF, obtained in previous steps.
- 6) Calculate the estimated value, increase the time step  $k = k + 1$ , go to the 2<sup>nd</sup> step.

The presented algorithm is easy to implement, but the estimated values are not optimal.

In a more general form the importance density (in 2<sup>nd</sup> step of the algorithm) may be any function dependent on both, the state vector from the previous time step  $\mathbf{x}^{i,(k-1)}$  and measurement vector  $\mathbf{z}^{(k)}$  (see Algorithm 2).

**Algorithm 2:** Generic particle filter

- 1) Initialization. Draw  $N$  particles from the initial PDF  $\mathbf{x}^{i,(0)} \sim p(\mathbf{x}^{(0)})$ , set initial values of particle weights  $q^{i,(0)} = 1/N$ , set the initial time step  $k = 1$ .
- 2) Prediction. Draw  $N$  new particles from the importance density  $\mathbf{x}^{i,(k)} \sim g(\mathbf{x}^{(k)} | \mathbf{x}^{i,(k-1)}, \mathbf{y}^{(k)})$ .
- 3) Update. Calculate particle weights based on the measurement model, the transition model and the importance density

$$q^{i,(k)} \propto q^{i,(k-1)} \frac{p(\mathbf{z}^{(k)} | \mathbf{x}^{i,(k)}) p(\mathbf{x}^{i,(k)} | \mathbf{x}^{i,(k-1)})}{g(\mathbf{x}^{i,(k)} | \mathbf{x}^{i,(k-1)}, \mathbf{z}^{(k)})}. \quad (7)$$

- 4) Normalization. Normalize weights so that their sum is equal to 1.
- 5) Check condition for resampling. If not met, go to the 7<sup>th</sup> step.

- 6) Resampling. Draw  $N$  new particles based on the posterior PDF, obtained in the previous steps; set new particle weights values (usually  $q^{i,(k)} = 1/N$ ).
- 7) Calculate the estimated value, increase the time step  $k = k + 1$ , go to the 2<sup>nd</sup> step.

In the presented above algorithm, the optimal choice of the importance density is [8]

$$g(\mathbf{x}^{(k)} | \mathbf{x}^{(k)}, \mathbf{z}^{(k)}) \propto p(\mathbf{z}^{(k)} | \mathbf{x}^{(k)}) p(\mathbf{x}^{(k)} | \mathbf{x}^{(k-1)}), \quad (8)$$

(see example in Fig. 1), however the analytical calculation is possible only for specific cases, and generally some other methods must be used (see the next Section).

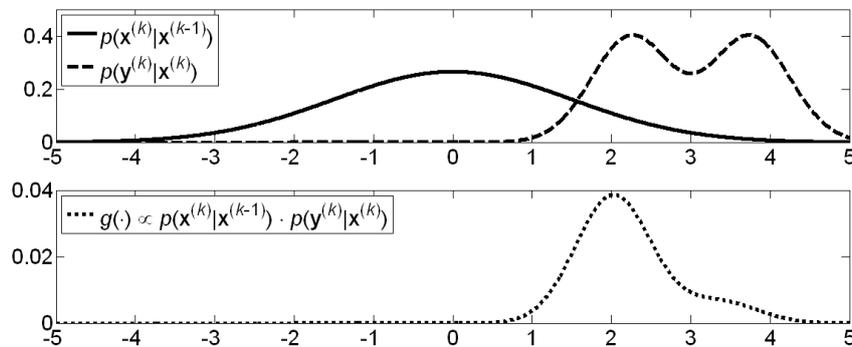


Fig. 1. The upper graph – exemplary transition and measurement models; the lower graph – multiplication of these two models and simultaneously optimal importance density for this case

The resampling step, which one can see in the both algorithms (second to the last step), can be compared to the survival of the fittest from the theory of natural selection [9]. In this step particles with low weights are deleted, and particles with high weights are copied (reproduced). There are many different kinds of resamplings, but the systematic resampling, which is one of the most used, has been chosen in this paper.

The operation principle of the systematic resampling is based on the assumption that all  $N$  particles are “drawn” using only one random number. This is possible because only the first value is drawn (uniformly distributed value from 0 to  $1/N$ ), and all others are increased by a multiple of  $1/N$ . If the particle’s weight is high enough (higher than  $2/N$ ), it is certain that it will be drawn to reproduce (the algorithm is more stable than multinomial resampling, in which every particle is selected from the whole range). In the Algorithm 3 one can find the pseudocode of the systematic resampling.

**Algorithm 3:** Systematic resampling

- 1) Initialization. Set value  $j = 1$  and sum of weights  $S = q^{1,(k)}$ , generate one random value  $u \sim \mathcal{U}[0, 1/N)$ .
- 2) For  $i = 1..N$  perform steps 3-5.
- 3) While  $S < u$  do:  $j = j + 1$  and  $S = S + q^{j,(k)}$ .
- 4) Choose particle  $\mathbf{x}^{j,(k)}$  for replication.

- 5) Calculate  $u = u + 1/N$ .
- 6) Set chosen particles as current set.
- 7) Set new particle weights  $q^{i(k)} = 1/N$ .

Description and comparison of many other resampling algorithms one can find in the article [10].

It should be also noted that the anti-zero bias (AZB) modification has been used. The first time it was proposed in [11]. AZB provides robustness to bad measurements data, but also the increase of the stability of the algorithm in specific cases.

Normally, the probability from measurement model  $p(\mathbf{z}^{(k)} | \mathbf{x}^{i(k)})$  is calculated as a product of all measurement probabilities

$$p(\mathbf{z}^{(k)} | \mathbf{x}^{i(k)}) \propto p(z_{(1)}^{(k)} | \mathbf{x}^{i(k)}) \cdot p(z_{(2)}^{(k)} | \mathbf{x}^{i(k)}) \cdot \dots \cdot p(z_{(m)}^{(k)} | \mathbf{x}^{i(k)}), \quad (9)$$

where  $m$  is the measurements number. It has been proposed to add small value  $b_0$ , i.e. AZB, to each component of the product (9), which can be written as

$$p(\mathbf{z}^{(k)} | \mathbf{x}^{i(k)}) \propto (p(z_{(1)}^{(k)} | \mathbf{x}^{i(k)}) + b_0) \cdot (p(z_{(2)}^{(k)} | \mathbf{x}^{i(k)}) + b_0) \cdot \dots \cdot (p(z_{(m)}^{(k)} | \mathbf{x}^{i(k)}) + b_0). \quad (10)$$

Thanks to this modification cases when all particle weights are equal to zero (and the algorithm must be terminated – division by zero – can not be performed) do not occur. The usage of AZB does not decrease the quality of estimation (even if the whole transition and measurement models are known), and additionally does not negatively affect the computation time. Deepened research on AZB has been already performed and the second version of the paper has been accepted (in *International Journal of Robust and Nonlinear Control*).

Readers interested in the particle filtering are referred to the literature [12-15].

### 3. Population Monte Carlo

As it has been written in the previous Section, the selection of the importance density to obtain drawn particles from an appropriate range is problematic. One of the solutions is the so-called Population Monte Carlo (PMC) approach, which assumes that particles are drawn in parts (populations). But each subsequent population can use information from the previous one (even in the same time step).

The description of the PMC presented in [16] is so general that based on this a lot of different modifications can be created. The importance density may be different for any particle and can depend on the time step  $k$  and the particle number  $i$ . The specific algorithm proposed in this paper consists in dividing the set of particles into  $P_N$  parts (populations), and the particles in each part are drawn based on the different importance density (the Gaussian distribution with parameters calculated based on the previous population).

The PMC algorithm has been described below, however one should remember that this algorithm corresponds to the second step of Algorithm 2.

**Algorithm 4:** Population Monte Carlo

- 1) For  $p = 1..P_N$  perform steps 2-11
- 2) If  $p = 1$  then (step 3)
- 3)  $\boldsymbol{\mu}_{[0]} = \mathbf{x}^{(k-1)}$ ,  $\boldsymbol{\sigma}_{[0]} = [\boldsymbol{\sigma}_U^T \quad \boldsymbol{\sigma}_\delta^T]^T$  – initial values of means and standard deviations.
- 4) Else (steps 5-7)
- 5)  $\boldsymbol{\mu}_{[p-1]} = \sum_{i=1}^{N/P_N} q_{[p-1]}^{i,(k)} \cdot \mathbf{x}_{[p-1]}^{i,(k)}$  – the calculation of the mean values, based on the particles from the previous population,
- 6)  $\boldsymbol{\Sigma}_{[p-1]} = \sum_{i=1}^{N/P_N} q_{[p-1]}^{i,(k)} \cdot (\mathbf{x}_{[p-1]}^{i,(k)} - \boldsymbol{\mu}_{[p-1]})^2$  – the calculation of the variance values, based on the particles from the previous population,
- 7)  $\boldsymbol{\sigma}_{[p-1]} = \sqrt{\boldsymbol{\Sigma}_{[p-1]}}$  – the calculation of the standard deviation values.
- 8) For  $i = 1..N/P_N$  perform steps 9-11
- 9)  $\mathbf{x}_{[p]}^{i,(k)} \sim \mathbf{g}(\mathbf{x}_{[p]}^{(k)} | \boldsymbol{\mu}_{[p-1]}, \boldsymbol{\sigma}_{[p-1]}^2) = N(\boldsymbol{\mu}_{[p-1]}, \boldsymbol{\sigma}_{[p-1]}^2)$  – drawing of the particle,
- 10)  $q_{[p]}^{i,(k)} = \frac{p(\mathbf{z}^{(k)} | \mathbf{x}_{[p]}^{i,(k)}) \cdot p(\mathbf{x}_{[p]}^{i,(k)} | \mathbf{x}^{i,(k-1)})}{\mathbf{g}(\mathbf{x}_{[p]}^{i,(k)} | \boldsymbol{\mu}_{[p-1]}, \boldsymbol{\sigma}_{[p-1]}^2)}$  – the calculation of the particle weight.
- 11) Remember a particle for the further calculations in the PF algorithm  
 $\mathbf{x}^{N \cdot (p-1) / P_N + i, (k)} = \mathbf{x}_{[p]}^{i, (k)}$ .

In the presented algorithm one can find the second power and the square root of vectors – this should be referred to the each vector element separately. Such notation has been used for the readability of the description. The exemplary result of Algorithm 4 (for the number of populations  $P_N = 5$  and the number of all particles  $N = 100$ ) has been presented in the Fig. 2. As one can see, using PMC the particles are drawn from the importance density, which is close to the optimal one.

In the performed simulations a one feature has been added to the Algorithm 4, after sixth step – the decrease of the variance has been limited in relation to the previous value (specifically, the value may be reduced 1.2 times in each subsequent population; the limitation has been added, because without this the variance decreases too much and the PMC gives the worse drawn particles).

It should be noted that computation time of the PF-PMC simulation is about two times longer than for the simple BF algorithm. Moreover calculations for every population must be performed sequentially and hence there are fewer opportunities to parallelization. One should also keep in mind that if the particles in the first population are drawn badly, probably all the remaining particles will not correspond to the optimal importance density.

The script (m-file) to draw graphs presented in the Fig. 2 one can find on the website [sites.google.com/site/piotrkozierski](http://sites.google.com/site/piotrkozierski) in the “Additional data”.

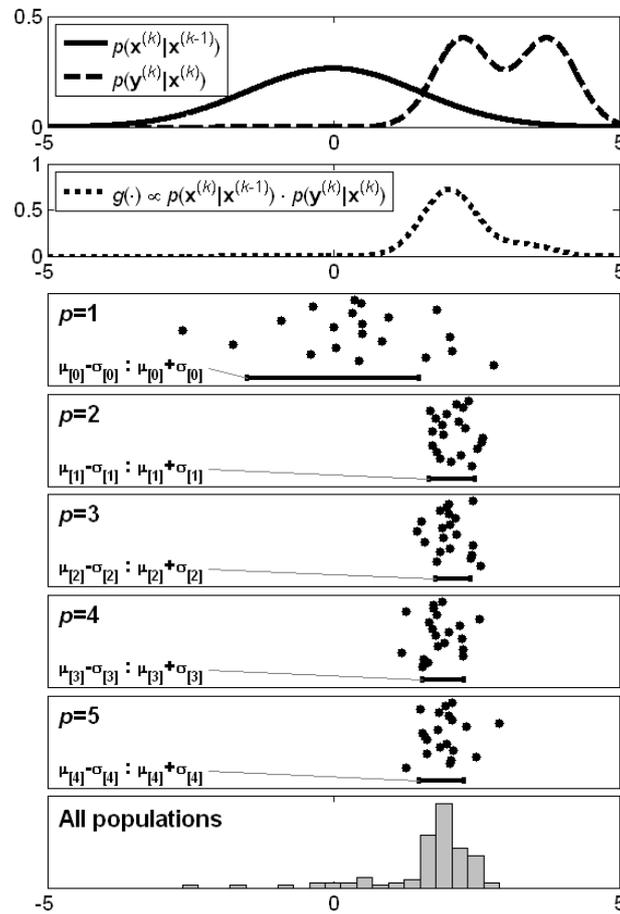


Fig. 2. The PMC principle of operation – example for  $P_n = 5$  and  $N = 100$ ; the transition and measurement models are visible in the first graph, and the optimal importance density is in the second graph; next 5 graphs contain drawn particles (dots) and the graphical representation of importance density for specific population  $p$  (the black line below the dots); the last graph contains the histogram of the drawn particles

#### 4. Power system

The power system is composed of  $B$  buses (nodes) and  $L$  branches (lines), and two buses can be connected by a branch. A  $\pi$  quadripole has been selected as a branch model (see Fig. 3), where  $\underline{y}'_{ij}/2$  is a half total line charging susceptance, and  $\underline{y}_{ij}$  is a line admittance

$$\underline{y}_{-ij} = \frac{1}{R_{ij} + sX_{ij}}, \quad (11)$$

where  $R_{ij}$  and  $X_{ij}$  are a resistance and a reactance respectively.

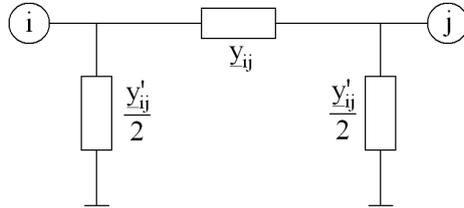


Fig. 3. Branch model in power system

The elements of the admittance matrix  $\underline{\mathbf{Y}}$  are calculated based on the expressions

$$\underline{Y}_{ii} = \sum_{\substack{j=1 \\ j \neq i}}^B \left( \frac{y'_{ij}}{2} + \underline{y}_{ij} \right), \quad (12)$$

$$\underline{Y}_{ij} = -\underline{y}_{ij} \quad i \neq j. \quad (13)$$

In exponential notation, the elements of admittance matrix can be written as

$$\underline{Y}_{ij} = Y_{ij} \cdot e^{j\mu_{ij}}, \quad (14)$$

where  $Y_{ij}$  and  $\mu_{ij}$  are used in equations (16-19).

A state of power system can be described as a set of voltage magnitudes  $U_i$  and phase angles  $\delta_i$  in all buses. However only a difference between two angles can be calculated. Therefore one bus must be reference (slack node) and the angle in such bus must be constant (usually equal to zero). Hence number of state variables is equal to  $2B-1$

$$\mathbf{x} = [x_1 \ \cdots \ x_{2B-1}]^T = [U_1 \ \cdots \ U_B \ \delta_2 \ \cdots \ \delta_B]^T. \quad (15)$$

It has been assumed that there are 5 different measurement types in the power system:

– active ( $P$ ) and reactive ( $Q$ ) power injections

$$P_i(\mathbf{U}, \boldsymbol{\delta}) = P_i = \sum_{j=1}^B U_i U_j Y_{ij} \cos(\delta_i - \delta_j - \mu_{ij}), \quad (16)$$

$$Q_i(\mathbf{U}, \boldsymbol{\delta}) = Q_i = \sum_{j=1}^B U_i U_j Y_{ij} \sin(\delta_i - \delta_j - \mu_{ij}), \quad (17)$$

– active and reactive power flows

$$P_{ij}(\mathbf{U}, \boldsymbol{\delta}) = P_{ij} = U_i^2 Y_{ij} \cos(-\mu_{ij}) - U_i U_j Y_{ij} \cos(\delta_i - \delta_j - \mu_{ij}), \quad (18)$$

$$Q_{ij}(\mathbf{U}, \boldsymbol{\delta}) = Q_{ij} = U_i^2 Y_{ij} \sin(-\mu_{ij}) - U_i U_j Y_{ij} \sin(\delta_i - \delta_j - \mu_{ij}) + U_i^2 \frac{y'_{ij}}{2}, \quad (19)$$

– nodal voltage magnitude  $U_i$

$$U_i(\mathbf{U}, \boldsymbol{\delta}) = U_i. \tag{20}$$

In the simulations the power system proposed in [17] has been used, which is composed of 5 buses and 6 branches (see Fig. 4). The designations have been explained in Fig. 5.

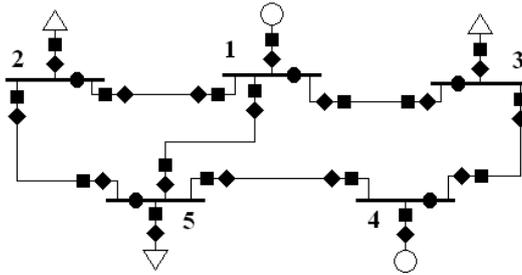


Fig. 4. The scheme of the power system used in simulations

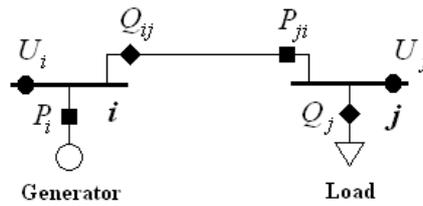


Fig. 5. The explanation of the designations in the Fig. 4

The assumption has been made, that the power system is the first order Markov Model. Hence, it can be written that

$$\mathbf{x}^{(k+1)} = \mathbf{f}_{MM}(\mathbf{x}^{(k)}, \mathbf{v}^{(k)}), \tag{21}$$

$$\mathbf{y}^{(k)} = \mathbf{g}_{MM}(\mathbf{x}^{(k)}, \mathbf{n}^{(k)}), \tag{22}$$

and more accurately

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{v}^{(k)}, \tag{23}$$

$$\mathbf{y}^{(k)} = \mathbf{h}(\mathbf{x}^{(k)}) + \mathbf{n}^{(k)}, \tag{24}$$

where  $\mathbf{h}$  is a vector of measurement functions (16-20) with length equal to  $m$  (number of measurements).

For more information about the power systems, books [17-20] are recommended.

## 5. Methodology and simulation results

Simulations have been performed using the power system presented in the Fig. 4. The estimation quality has been defined as an average value of the Root Mean Square Errors (RMSEs) of all state variables, which can be written as

$$\text{aRMSE} = \frac{1}{2B-1} \sum_{i=1}^{2B-1} \text{RMSE}_i, \quad (25)$$

$$\text{RMSE}_i = \sqrt{\text{MSE}_i}, \quad (26)$$

$$\text{MSE}_i = \frac{1}{M} \sum_{k=1}^M (\hat{x}_i^{(k)} - x_i^{(k)+})^2, \quad (27)$$

where  $M$  is the simulation length ( $M = 100$  in this case), the state variables  $x$  with plus and with hat are the true value and the estimated value respectively, and  $2B-1$  is the number of the state variables (in considered case – 9).

The results presented on the charts are mean values of aRMSE – simulations have been repeated from 100 times (for the number of the particles  $N = 100\ 000$ ) to 1000 and more times (for  $N \leq 1000$ ).

All simulations have been performed using the same signals and noises. The data set has been chosen from among 100 different in such a way that some initial simulations have been performed. Afterwards, the mean values (for the different models (transition and measurement) and the various values of  $m_{\text{ult}}$ ) have been calculated, and the case, for which the obtained results were the most similar, has been chosen for all next simulations. The  $m_{\text{ult}}$  value explains, how many times the modelled standard deviations of the models (transition and measurement) were greater than the true standard deviations.

The choice of good data set (a specific seed value of the random number generator) is very important, because in the paper prepared for the MMAR 2015 (Methods and Models in Automation and Robotics) conference, it has been observed that different seed value may completely change conclusions. Studies on the appropriate choice of data for calculations are currently being carried out.

4 methods have been used in calculations: Bootstrap filter, PF with PMC, EKF and UKF. The simulations have been performed for too high (10, 6, 3, 2, 1.5, 1.2 and 1.1 times), too small (also 10, 6, 3, 2, 1.5, 1.2 and 1.1 times) and correct values of the standard deviations, separately for the transition model and the measurement model.

The results have been presented in the Figs. 6-15. The values of aRMSE, which have been obtained for the object with an incorrect transition model, have been presented in the Figs. 6-9, whereas in the Figs. 10-13 one can see the results for a wrong measurement model. BF abbreviation means that Bootstrap filter method has been used, while PMC means that the Population Monte Carlo has been additionally used. The  $m_{\text{ult}}$  values have been added in the graphs as subscripts of the method abbreviations. Figs. 14-15 contain the results of the EKF and UKF methods, which have been compared only to the  $\text{BF}_1$  and  $\text{PMC}_1$  (for the readability). Additionally, the results of the EKF and UKF methods have been summarized in the Table 1.

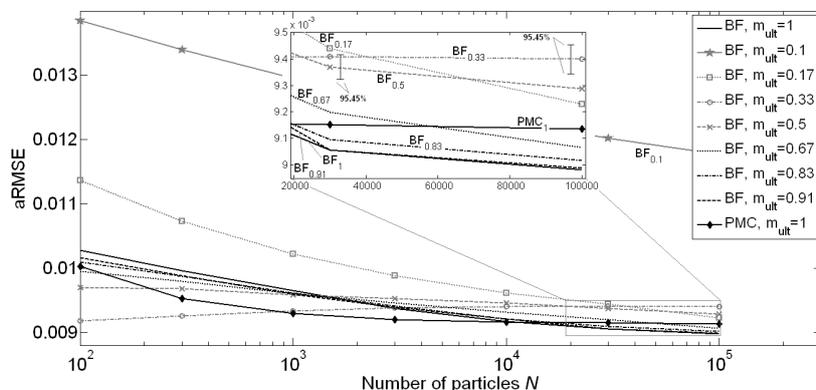


Fig. 6. Results of Bootstrap Filter method with an incorrect **transition model**,  $m_{ult} \leq 1$  (the standard deviation in the algorithm was  $m_{ult}$  times greater than the true value)

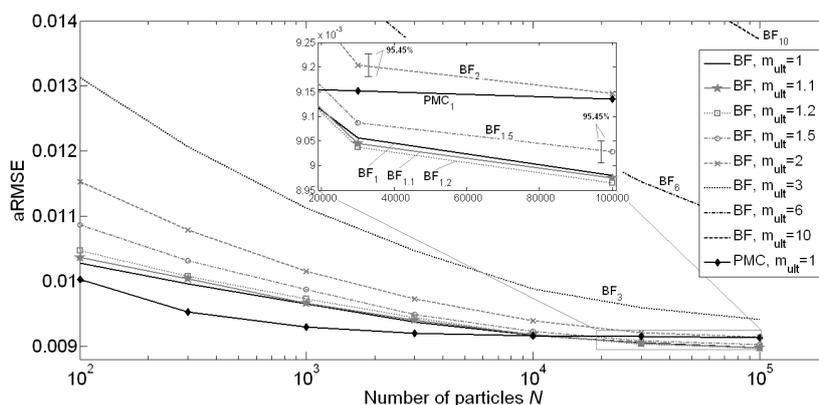


Fig. 7. Results of Bootstrap Filter method with an incorrect **transition model**,  $m_{ult} \geq 1$  (the standard deviation in the algorithm was  $m_{ult}$  times greater than the true value)

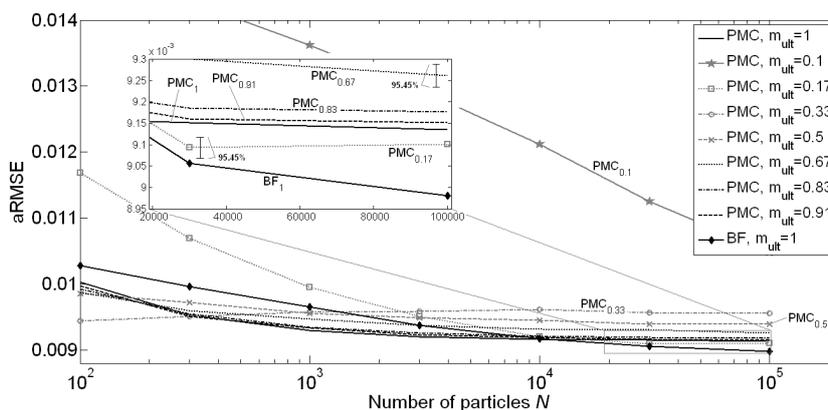


Fig. 8. Results of the PF-PMC method with an incorrect **transition model**,  $m_{ult} \leq 1$  (the standard deviation in the algorithm was  $m_{ult}$  times greater than the true value)

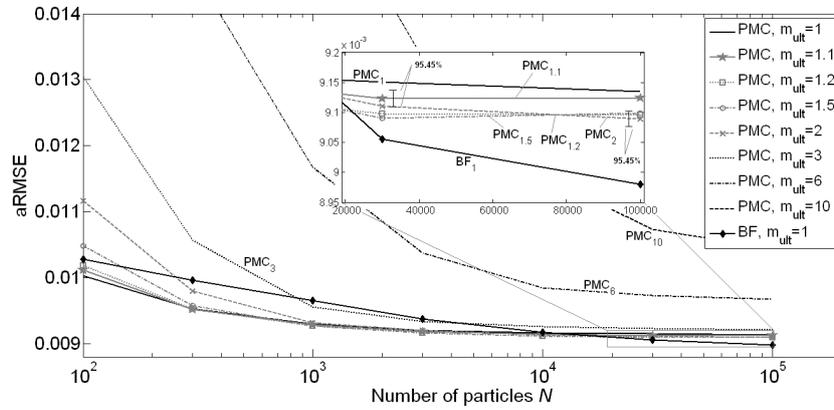


Fig. 9. Results of the PF-PMC method with an incorrect **transition model**,  $m_{ult} \geq 1$  (the standard deviation in the algorithm was  $m_{ult}$  times greater than the true value)

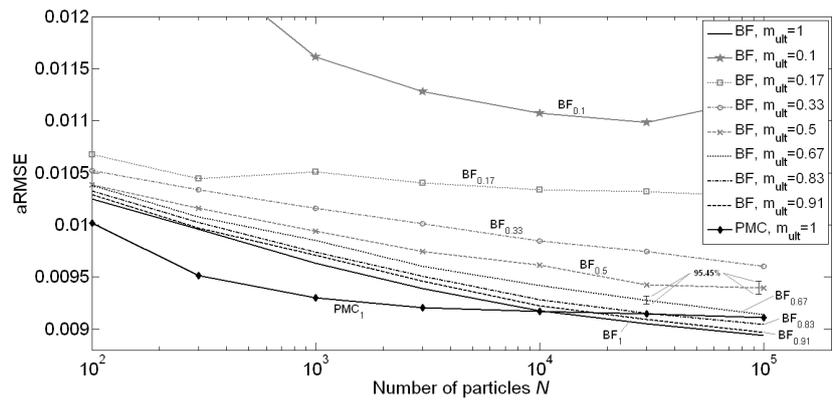


Fig. 10. Results of Bootstrap Filter method with an incorrect **measurement model**,  $m_{ult} \leq 1$  (the standard deviation in the algorithm was  $m_{ult}$  times greater than the true value)

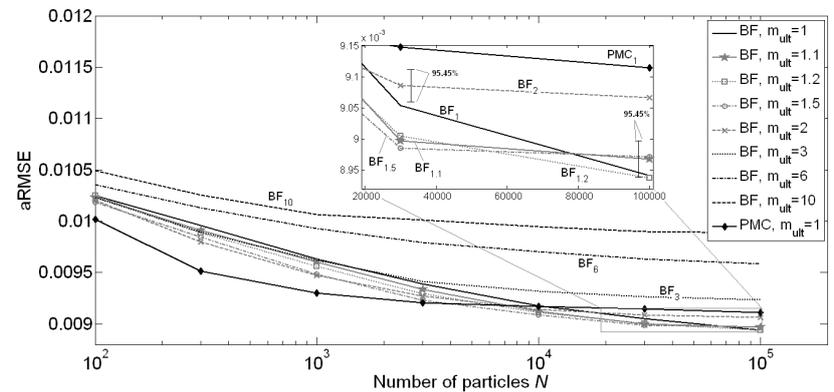


Fig. 11. Results of Bootstrap Filter method with an incorrect **measurement model**,  $m_{ult} \geq 1$  (the standard deviation in the algorithm was  $m_{ult}$  times greater than the true value)

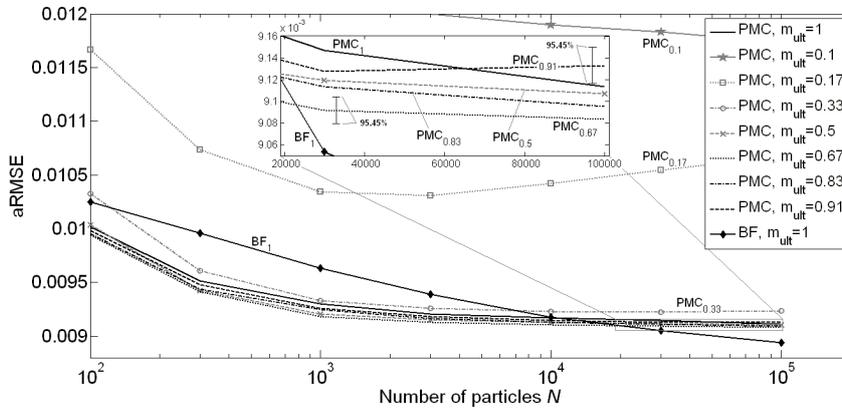


Fig. 12. Results of the PF-PMC method with an incorrect **measurement model**,  $m_{ult} \leq 1$  (the standard deviation in the algorithm was  $m_{ult}$  times greater than the true value)

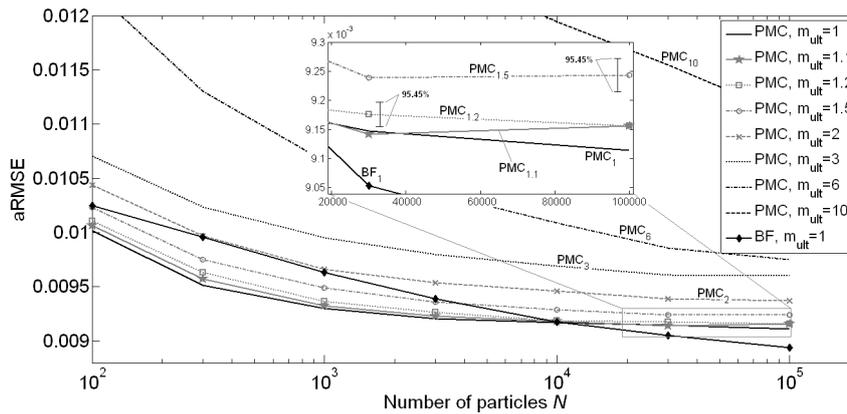


Fig. 13. Results of the PF-PMC method with an incorrect **measurement model**,  $m_{ult} \geq 1$  (the standard deviation in the algorithm was  $m_{ult}$  times greater than the true value)

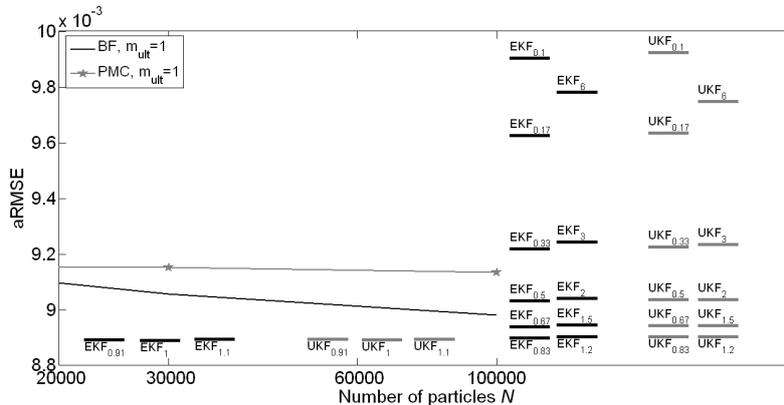


Fig. 14. Results of the EKF and UKF methods with an incorrect **transition model**

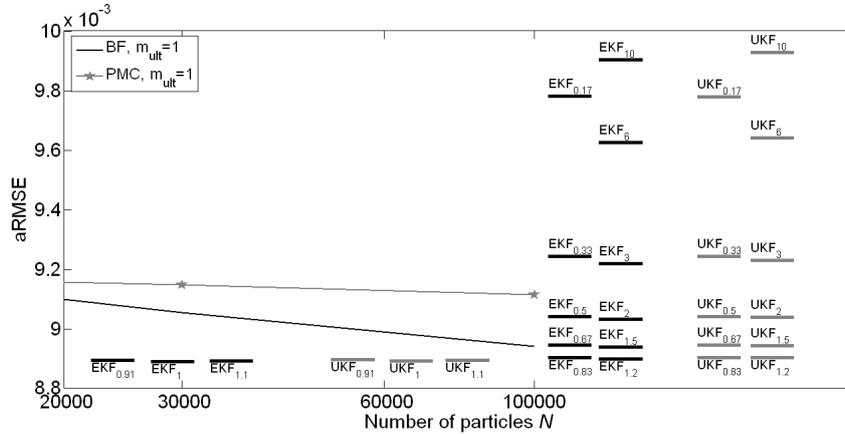


Fig. 15. Results of the EKF and UKF methods with an incorrect **measurement model**

Table 1. Results of the EKF and UKF methods (the same one can see in Figs. 14-15)

$m_{ult}$	aRMSE values			
	incorrect transition model		incorrect measurement model	
	EKF method	UKF method	EKF method	UKF method
0.1	0.0099049	0.0099241	0.010331	0.010328
0.17	0.0096256	0.0096359	0.0097824	0.0097806
0.33	0.0092198	0.0092269	0.0092446	0.0092436
0.5	0.0090315	0.0090379	0.0090415	0.0090409
0.67	0.0089393	0.0089443	0.0089456	0.0089457
0.83	0.0088995	0.008903	0.0089026	0.0089037
0.91	0.0088924	0.0088953	0.0088941	0.0088957
1	0.0088903	0.0088924	0.0088903	0.0088924
1.1	0.0088941	0.0088955	0.0088924	0.0088952
1.2	0.0089026	0.0089034	0.0088995	0.0089029
1.5	0.0089456	0.0089445	0.0089393	0.0089444
2	0.0090415	0.0090377	0.0090315	0.0090386
3	0.0092446	0.0092347	0.0092198	0.0092295
6	0.0097824	0.009748	0.0096256	0.0096414
10	0.010331	0.010279	0.0099049	0.009928

### 6. Conclusions

Based on the results presented in the previous Section, one can say that using the incorrect transition and measurement model of the PF methods can provide a slightly better estimation quality than the correct models.

In the BF algorithm the variance of the transition model can be slightly greater than the real one. Similar conclusions can be drawn for the PF with PMC. For the  $m_{ult}$  equal to 0.17 a quite good estimation has also been obtained (see Fig. 8), however after comparison with other results one can conclude that this is rather a particular case. For a smaller number of particles (in results of BF method) a better estimation has been obtained for variances (and standard deviations) smaller than the true value. It is because the changes of the state variables in the data set were consistent with the Gaussian distribution. Hence, it was better “not to move” than “to move too much”.

In the BF method, a wrong measurement model with a higher standard deviation than the real one provides the better estimation quality. It is especially visible for  $N < 10\,000$  (see Fig. 11). The different conclusion has been drawn only for PF-PMC with incorrect measurement model – better estimation quality can be obtained if the standard deviation in the algorithm is 1-2 times smaller than the true values.

Comparing the BF and PF-PMC methods, one can say that the better results can be obtained with the use of the BF method. It is because the PF-PMC method has a higher level of results saturation than the BF method, which has been presented in [21]. Moreover it has been confirmed that in high-dimensional plants the PF-PMC method provides better estimation quality for a small number of particles (in this case –  $N < 10\,000$ ). Based on this, one can say that PMC allows for a significant reduction of needed particles. One should keep in mind that simulations of PF-PMC are about 2 times longer than in the BF method, however the significant profit of using PMC is a much higher number of visible particles (about 1000).

Based on the results of EKF and UKF methods a few conclusions can be drawn. Firstly, the particle filter methods are very inefficient for a high-dimensional object. It is possible to obtain a better estimation quality than for Kalman filter methods, however a very large number of particles it would be needed.

Secondly, the obtained results are very symmetric (see Figs. 14-15) and in each case the best estimation quality is obtained for  $m_{ult} = 1$  (which is not typical). Hence one can say that the data set, which has been chosen, ensure to obtain the results, which reflect the reality. Therefore, there is a possibility that the best data set for the research can be chosen based on the Kalman filters results (the research on the choice of the simulation data set is currently performed).

Thirdly, the results of the extended and unscented Kalman filters are very similar, but in fact, UKF should operate better. It may be caused by too strong linearity in the used power grid – difference between phase angles are very small and  $\mu_{ij}$  values also (see Eq. (14)), hence nonlinear functions sine and cosine are not noticeable. Therefore in the future studies it is planned to change parameters of the power system proposed in [17].

Fourthly, it is intriguing that results of the EKF method are the same for incorrect transition and measurement models – results for one model and  $m_{ult}$  value are equal to the second model and  $1/m_{ult}$  value. Perhaps readers interested in the Kalman filters will want to explore this property.

The research on the comparison of the methods for the case in which standard deviations are estimated together with state variables is planned in the future. Another methods for the optimal choice of importance density  $g(\cdot)$  will also be considered.

## References

- [1] Huang Z., Schneider K., Nieplocha J., *Feasibility Studies of Applying Kalman Filter Techniques to Power System Dynamic State Estimation*, Proc. In Power Engineering Conference, IPEC 2007 376-382 (2007).
- [2] Shih K.R., Huang S.J., *Application of a Robust Algorithm for Dynamic State Estimation of a Power System*, Power Systems, IEEE Transactions on, DOI: 10.1109/59.982205, 17(1): 141-147 (2002).
- [3] Valverde G., Terzija V., *Unscented Kalman Filter for Power System Dynamic State Estimation*, IET Generation, Transmission & Distribution 5(1): 29-37 (2011).
- [4] Janiszewski D., *Particle Filter Approach for Permanent Magnet Synchronous Motor State Estimation*, Przegląd Elektrotechniczny 90(6): 56-60 (2014).
- [5] Candy J.V., *Bayesian Signal Processing*, WILEY, New Jersey, DOI: 10.1002/9780470430583, pp. 36-44 (2009).
- [6] Schön T.B., Wills A., Ninness B., *System Identification of Nonlinear State-space Models*, Automatica 47(1): 39-49 (2011).
- [7] Gordon N.J., Salmond D.J., Smith A.F.M., *Novel approach to nonlinear/non-Gaussian Bayesian state estimation*, IEE Proceedings-F 140(2): 107-113 (1993).
- [8] Šmídl V., Hofman R., *Adaptive Importance Sampling in Particle Filtering*, Proc. 16th Int. Conf. on Information Fusion (FUSION), pp. 9-16 (July 2013).
- [9] Thrun S., Burgard W., Fox D., *Probabilistic robotics*, MIT Press, Cambridge, 67-90 (2005).
- [10] Kozierski P., Lis M., Ziętkiewicz J., *Resampling in Particle Filtering – Comparison*, Studia z Automatyki i Informatyki 38: 35-64 (2013).
- [11] Kozierski P., Lis M., Owczarkowski A., Ziętkiewicz J., *Particle Filter in Power System State Estimation – Large Measurements Errors*, Proc. 16th Nat. Conf. on Advances in Applied Electrical Engineering (PES-9), pp. 157-160 (June 2014).
- [12] Arulampalam S., Maskell S., Gordon N., Clapp T., *A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking*, IEEE Proceedings on Signal Processing 50(2): 174-188 (2002).
- [13] Simon D., *Optimal State Estimation*, WILEY-INTERSCIENCE, New Jersey (2006).
- [14] Kozierski P., Lis M., *Particle Filter in Tracking Problem – Introduction*, Studia z Automatyki i Informatyki 37: 79-94 (2012). (in Polish)
- [15] Doucet A., Freitas N., Gordon N., *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, New York, DOI: 10.1007/978-1-4757-3437-9, pp. 225-246 (2001).
- [16] Cappe O., Guillin A., Marin J.M., Robert C.P., *Population Monte Carlo*, Journal of Computational and Graphical Statistics, DOI: 10.1198/106186004X12803, 13(4): 907-929 (2004).
- [17] Kremens Z., Sobierajski M., *The Analysis of Power Systems*, Wydawnictwa Naukowo-Techniczne, Warsaw, pp. 39-191 (1996). (in Polish)
- [18] Abur A., Exposito A.G., *Power System State Estimation: Theory and Implementation*, Marcel Dekker, Inc., pp. 17-49 (2004).
- [19] Andersson G., *Modelling and Analysis of Electric Power Systems*, EEH-Power Systems Laboratory, Swiss Federal Institute of Technology (ETH), Zürich (2008).
- [20] Milano F., *Power System Modelling and Scripting*, DOI: 10.1007/978-3-642-13669-6, Springer, London, Dordrecht, Heidelberg, New York (2010).
- [21] Kozierski P., *Population Monte Carlo and Adaptive Importance Sampling in Particle Filter*, Studia z Automatyki i Informatyki 39: 33-41 (2014).