STUDIA Z AUTOMATYKI I INFORMATYKI
TOM 42 – 2017

Jacek Michalski*, Piotr Kozierski†, Joanna Zietkiewicz‡

COMPARISON OF PARTICLE FILTER AND EXTENDED KALMAN PARTICLE FILTER

Keywords: state estimation, Extended Kalman Filter, Particle Filter, Extended Kalman Particle Filter

1. INTRODUCTION

All performed measurements (despite of units, such as temperature, water level or current in a line) are associated with some noise, and the correct state estimation allows to reduce errors. Kalman filter (KF) was proposed by Rudolf Kalman in 1960 [10], and its version for nonlinear systems, so called Extended Kalman Filter (EKF), 10 years later by Andrew Jazwinski [8].

The Particle Filter (PF) algorithm is much younger, because the first algorithm was proposed in 1993 [7]; however it is worth noting that already in the 1940s years Norbert Wiener suggested something like PF [18]. In 1980s and early 1990s Sequential Importance Sampling methods were studied, and by adding the resampling step to them the Sequential Importance Resampling (SIR – other name of PF) was created.

Since then there have been proposed many algorithms, which combine any version of KFs and PFs, e.g. Unscented Kalman Particle Filter [20, 15], Mixed Kalman Particle Filter [3] or Iterated Extended Kalman Particle Filter [14]. The authors decided to focus at the beginning on the Extended Kalman Particle Filter (EKPF), and the remaining ones will be studied in the future.

In Section 2, the formulation of the problem is presented. The article is focused on describing algorithms of state estimation: Extended Kalman Filter (Section 3), Particle Filter – Bootstrap Filter (Section 4) and Extended Kalman Particle Filter (Section 5). In Section 6, the examined objects are presented, and Section 7 contains results of the simulations, and conclusions are given in Section 8.

*Poznan University of Technology, Faculty of Electrical Engineering, Institute of Control, Robotics and Information Engineering, Division of Control and Robotics, e-mail: jacekinzynier@gmail.com

†Poznan University of Technology, Faculty of Electrical Engineering, Institute of Control, Robotics and Information Engineering, and also Poznan University of Technology, Faculty of Computing, Institute of Automation and Robotics, Division of Signal Processing and Electronic Systems, Piotrowo 3a Str, 60-965 Poznań, e-mail: piotr.kozierski@gmail.com

‡Poznan University of Technology, Faculty of Electrical Engineering, Institute of Control, Robotics and Information Engineering, Division of Control and Robotics, e-mail: joanna.zietkiewicz@put.poznan.pl

2. FORMULATION OF THE PROBLEM

The discrete system in state space is given by

$$\begin{cases} \underline{x}^{(k+1)} = \underline{f}(\underline{x}^{(k)}, \underline{u}^{(k)}; k) + \underline{v}^{(k)} \\ \underline{y}^{(k)} = \underline{h}(\underline{x}^{(k)}) + \underline{n}^{(k)} \end{cases} \tag{1}$$

where $\underline{x}^{(k)}$ is a state vector at k-th time step, $\underline{u}^{(k)}$ is an input vector, $\underline{y}^{(k)}$ is a measurement vector, $\underline{v}^{(k)}$ is a process noise vector, $\underline{n}^{(k)}$ is a measurement noise vector, $\underline{f}(\cdot)$ and $\underline{h}(\cdot)$ are transition and measurement nonlinear vector functions, respectively. The task is to reconstruct values of the state variables from the available measured outputs and known inputs of the system.

3. EXTENDED KALMAN FILTER

This method can be used for a nonlinear object (1) and probably is the most widely used estimation algorithm due to its implementation simplicity, satisfactory time of operation and relatively good estimation results (in general).

However, in special cases estimation results of EKF method may be poor. Such cases are plants, in which error propagation can not be well approximated by linear function, objects which have highly quantized measurements from sensors [9] and so called jump systems [13].

The EKF algorithm is composed of the two main steps:

- prediction step (time update):

$$\underline{\hat{x}}^{(k|k-1)} = f\left(\underline{\hat{x}}^{(k-1|k-1)}, \underline{u}^{(k-1)}\right), \tag{2}$$

$$\boldsymbol{P}^{(k|k-1)} = \boldsymbol{F}^{(k-1)} \boldsymbol{P}^{(k-1|k-1)} \boldsymbol{F}^{(k-1)^T} + \boldsymbol{Q}, \tag{3}$$

- filtration step (measurement update):

$$\boldsymbol{K}^{(k)} = \boldsymbol{P}^{(k|k-1)} \boldsymbol{H}^{(k)^T} \left(\boldsymbol{H}^{(k)} \boldsymbol{P}^{(k|k-1)} \boldsymbol{H}^{(k)^T} + \boldsymbol{R}\right)^{-1}, \tag{4}$$

$$\underline{\hat{x}}^{(k|k)} = \underline{\hat{x}}^{(k|k-1)} + \boldsymbol{K}^{(k)} \left(\underline{y}^{(k)} - \underline{h}(\underline{\hat{x}}^{(k|k-1)})\right), \tag{5}$$

$$\boldsymbol{P}^{(k|k)} = \left(\boldsymbol{I} - \boldsymbol{K}^{(k)} \boldsymbol{H}^{(k)}\right) \boldsymbol{P}^{(k|k-1)}, \tag{6}$$

where $\boldsymbol{F}^{(k-1)}$ is a Jacobian matrix calculated as $\boldsymbol{F}^{(k-1)} = \nabla \underline{f}_{k-1}\big|_{\underline{\hat{x}}^{(k-1|k-1)}}$, $\boldsymbol{H}^{(k)}$ is a Jacobian matrix calculated as $\boldsymbol{H}^{(k)} = \nabla \underline{h}_k\big|_{\underline{\hat{x}}^{(k|k-1)}}$. Superscripts $(k|k-1)$ denote values predicted at k-th time step with the use of the information from previous step and the transition model.

Despite the fact that the original Kalman filter did not apply Bayes' rule [10], it has been proven that Kalman's equations can be derived from Recursive Bayesian Filter (see (7) in the next section), under the assumption that models are linear-Gaussian. It is worth noting that there is a common misconception that Kalman filters can be used only to Gaussian plants, while there is nothing about distribution type, but only the mean and covariance [9].

More information about Kalman filtering can be found in books [18, 5].

4. PARTICLE FILTER

This method can be used for a nonlinear object (1) and also for plants with non-Gaussian transition or measurement models. The Particle Filter (PF) operation principle is based on the Recursive Bayesian Filter [2], given by

$$p\left(\underline{x}^{(k)}|\mathbf{Y}^{(k)}\right) = \frac{p\left(\underline{y}^{(k)}|\underline{x}^{(k)}\right) p\left(\underline{x}^{(k)}|\mathbf{Y}^{(k-1)}\right)}{p\left(\underline{y}^{(k)}|\mathbf{Y}^{(k-1)}\right)}, \tag{7}$$

where $\mathbf{Y}^{(k)}$ is a set of output measurements $\{\underline{y}^{(1)}, \ldots, \underline{y}^{(k)}\}$, $p\left(\underline{x}^{(k)}|\mathbf{Y}^{(k)}\right)$ is a posterior probability density function (PDF), $p\left(\underline{y}^{(k)}|\underline{x}^{(k)}\right)$ is a likelihood, $p\left(\underline{x}^{(k)}|\mathbf{Y}^{(k-1)}\right)$ is a prior PDF, and $p\left(\underline{y}^{(k)}|\mathbf{Y}^{(k-1)}\right)$ is an evidence

In the PF algorithm, posterior PDF is represented by a set of particles. Each particle consists of value $\boldsymbol{x}^i$ and weight q^i for $i = 1, 2, ..., N_p$, where N_p is the number of particles. Hence, the i-th particle can be written as a pair $\{\underline{x}^i,\ q^i\}$. When the particles number N_p is large enough, posterior PDF can be approximated by

$$p\left(\underline{x}^{(k)}|\mathbf{Y}^{(k)}\right) \overset{N_p\to\infty}{=} \hat{p}\left(\underline{x}^{(k)}|\mathbf{Y}^{(k)}\right) = \sum_{i=1}^{N_p} q^{i,(k)}\,\delta\left(\underline{x}^{(k)} - \underline{x}^{i,(k)}\right), \tag{8}$$

where $\delta(\cdot)$ is a Dirac's delta function.

The algorithm of Bootstrap Filter, which was proposed in 1993 by Gordon, Salmond and Smith [7], and is presented below.

Algorithm 1. Bootstrap Filter (BF)

1. Initialization. Draw initial values of the particles $\underline{x}^{i,(0)} \sim p(\underline{x}^{(0)})$, set time step $k := 1$.
2. Prediction. Draw N_p new particles from the transition model: $\underline{x}^{i,(k)} \sim p(\underline{x}^{(k)}|\underline{x}^{i,(k-1)})$.
3. Update. Calculate particles' weights from the measurement model

$$q^{i,(k)} = p(\underline{y}^{(k)}|\underline{x}^{i,(k)}). \tag{9}$$

4. Normalization. Scale values of the weights in such a way that their sum equals 1.
5. Resampling (systematic resampling has been used [12]).
6. End of the iteration. Calculate the estimate of the state vector at k-th time step, update the time step $k := k + 1$, go to Step 2.

In the most general form – Sequential Importance Resampling (SIR) – there are two main differences in comparison to BF, i.e. resampling does not have to be executed in every iteration (only when relatively low number of particles have significant weights values), and in the prediction step one can use any, so called importance function $\pi(\underline{x}^{(k)}|\underline{x}^{(k-1)}, \underline{y}^{(k)})$, from which particles will be drawn. Therefore, the expression for weights calculation in the update step of SIR algorithm is given by

$$q^{i,(k)} = q^{i,(k-1)}\frac{p(\underline{y}^{(k)}|\underline{x}^{i,(k)})p(\underline{x}^{(k)}|\underline{x}^{i,(k-1)})}{\pi(\underline{x}^{i,(k)}|\underline{x}^{i,(k-1)}, \underline{y}^{(k)})}. \tag{10}$$

The SIR approach is useful especially when the transition model is given by function, from which particles drawing is problematic.

The main disadvantage of PF usage is needed number of calculations, which grows exponentially with a number of state variables [19]. However, there is a number of advantages in favour of using particle filtering: the optimality in nonlinear and non-Gaussian plants (due to Bayesian based solution), the possibility of adaptation of performed calculations to used device (by changing the number of particles), the possibility of parallelize calculations on GPU [17] or FPGA [16, 19] (all calculations, except resampling, are performed independently for each particle).

For more information about particle filters references [1, 4] are suggested.

5. EXTENDED KALMAN PARTICLE FILTER

This is a modified PF algorithm, where particles in the prediction step are not drawn from the transition model, but from the PDF calculated using EKF method. Therefore, the weights calculation (11) is based on the importance function usage.

With the combination of EKF and PF algorithms, the meaningful estimation results should be obtained also for a very small particles number N_p, and with N_p increasing the estimation quality should improve. Except that strengths and weaknesses are similar to EKF and PF methods.

Algorithm 2. Extended Kalman Particle Filter (EKPF)

1. Initialization. Draw N_p particle values from initial PDF $\underline{x}^{i,(0)} \sim p(\underline{x}^{(0)})$, set initial values of covariance matrices $\boldsymbol{P}^{i,(0|0)} = \boldsymbol{Q}$, set the number of iteration $k = 1$.
2. Prediction. Draw N_p new particles $\underline{x}^{i,(k)} \sim g(\underline{x}^{(k)}|\underline{x}^{i,(k-1)}, \underline{y}^{(k)}) = \mathcal{N}(\underline{\hat{x}}^{i,(k|k)}, \boldsymbol{P}^{i,(k|k)})$, where $\underline{\hat{x}}^{i,(k|k)}$ and $\boldsymbol{P}^{i,(k|k)}$ are the values of estimated state variables and covariance matrices, respectively, calculated from (2)-(6) for each particle, and $\underline{\hat{x}}^{i,(k-1|k-1)}$ in (2) is equal to $\underline{x}^{i,(k-1)}$ after resampling (EKF calculations are performed N_p times at every time step).
3. Update. Calculate the weights of particles

$$q^{i,(k)} = \frac{p(\underline{y}^{(k)}|\underline{x}^{i,(k)})p(\underline{x}^{i,(k)}|\underline{x}^{i,(k-1)})}{g(\underline{x}^{i,(k)}|\underline{x}^{i,(k-1)}, \underline{y}^{(k)})}. \tag{11}$$

4. Normalization. Scale the weights in such a way that their sum equals 1.
5. Resampling (systematic resampling has been used [12]).
6. End of the iteration. Calculate the estimate of state vector, update the time step $k := k + 1$, go to Step 2.

6. EXAMINED OBJECTS

The systems *Ob*1 and *Ob*2 are plants without the input signal (autonomous), so change of the state is only due to the presence of a process noise. *Ob*1 is used very often in particle filters examination [1, 7]. According to [11] this object was firstly proposed in 1978 by Netto,

Gimeno and Mendes. *Ob*2 is the modification of *Ob*1 through simplification of measurement function and removal of the cosine term from transfer function. The *Ob*3 is multidimensional system with 3 inputs and 3 outputs (MIMO object).

- *Ob*1:

$$\begin{cases} x^{(k+1)} = 0.5x^{(k)} + \frac{25x^{(k)}}{1+x^{(k)^2}} + 8\,cos(1.2k) + v^{(k)} \\ y^{(k)} = \frac{x^{(k)^2}}{20} + n^{(k)} \end{cases},$$

$$\begin{aligned} v^{(k)} &\sim \mathcal{N}(0;\ 10), \\ n^{(k)} &\sim \mathcal{N}(0;\ 1), \\ x^{(0)} &= 0.1\,. \end{aligned}$$

- *Ob*2:

$$\begin{cases} x^{(k+1)} = 0.5x^{(k)} + \frac{25x^{(k)}}{1+x^{(k)^2}} + v^{(k)} \\ y^{(k)} = 2x^{(k)} + n^{(k)} \end{cases},$$

$$\begin{aligned} v^{(k)} &\sim \mathcal{N}(0;\ 10), \\ n^{(k)} &\sim \mathcal{N}(0;\ 10^2), \\ x^{(0)} &= 0.1\,. \end{aligned}$$

- *Ob*3:

$$\begin{cases} x_1^{(k+1)} = 0.5\left(x_1^{(k)^2}\right)^{\frac{1}{3}} + 0.3x_2^{(k)}x_3^{(k)} + 0.2u_1^{(k)} + v_1^{(k)} \\ x_2^{(k+1)} = 0.5\left(x_2^{(k)^2}\right)^{\frac{1}{3}} + 0.3x_3^{(k)}x_1^{(k)} + 0.2u_2^{(k)} + v_2^{(k)} \\ x_3^{(k+1)} = 0.5\left(x_3^{(k)^2}\right)^{\frac{1}{3}} + 0.3x_1^{(k)}x_2^{(k)} + 0.2u_3^{(k)} + v_3^{(k)} \\ y_1^{(k)} = 0.5\left(x_1^{(k)} + x_2^{(k)} + x_3^{(k)}\right) + n_1^{(k)} \\ y_2^{(k)} = 2\left(x_1^{(k)}\right)^2 + n_2^{(k)} \end{cases},$$

$$\begin{aligned} v_{1/2/3}^{(k)} &\sim \mathcal{N}(0;\ 0.1), \\ n_{1/2}^{(k)} &\sim \mathcal{N}(0;\ 0.1), \\ u_{1/2/3}^{(k)} &\sim \mathcal{U}[-1;1], \\ x^{(0)} &= [0.1,\ 0.1,\ 0.1]^T. \end{aligned}$$

7. RESULTS OF SIMULATIONS

Calculations were performed for every method and plant configurations. Each simulation was repeated 1000 times, and all signals in the system were different for each simulation. Simulations with PF and EKPF methods were performed with different numbers of particles.

Standard deviations were calculated based on the theory from [6], i.e. the variance of the mean value is m times smaller than the variance from m samples, for Gaussian PDF.

The quality index $aRMSE$ has been used, which is given by equations:

$$MSE_i = \frac{1}{M}\sum_{k=1}^{M}\left(\hat{x}_i^{(k)} - x_i^{+(k)}\right), \tag{12}$$

$$RMSE_i = \sqrt{MSE_i}, \tag{13}$$

$$aRMSE = \frac{1}{N_x}\sum_{i=1}^{N_x} RMSE_i\,, \tag{14}$$

where M is the length of the simulation, N_x is the number of state variables, $\hat{x}_i^{(k)}$ and $x_i^{+(k)}$ are estimated real values of the i-th state variable, respectively, at the k-th time step.

Quality indices of each methods for *Ob*1, *Ob*2 and *Ob*3 have been presented in Figures 1-3. Standard deviations with 95% probability round-trip (according to 68-95-99.7 rule) have been presented in the graphs.

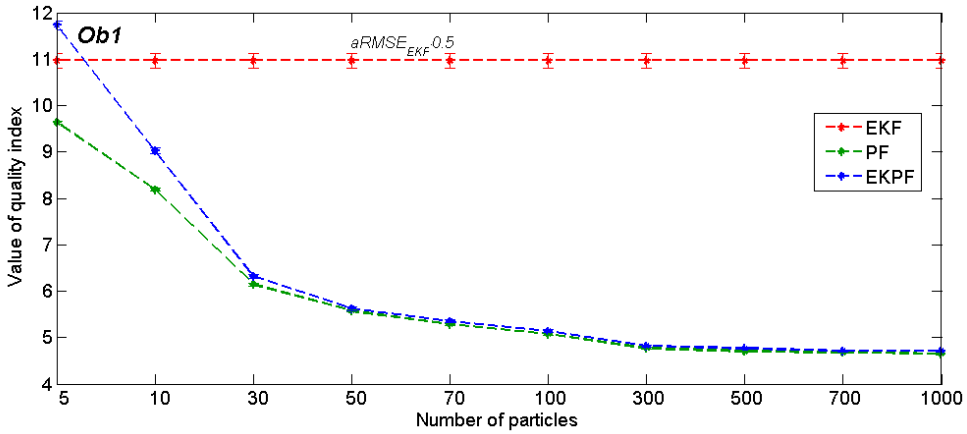

Fig. 1. Values of $aRMSE$ for *Ob*1; the result for EKF has been presented as two times smaller (for better readability)

8. Conclusions

Based on the simulation results, one can see that EKPF usually provides better estimation quality (for plants *Ob*2 and *Ob*3); however, in system *Ob*1 this relation is not met. It is caused by highly nonlinear transition and measurement models of this plant, and as it was mentioned, for such systems EKF methods can not be used. Additionally, KF methods have access to history only as the state vector from the previous time step (while PF methods, based on the Bayes rule, contain whole history) – this is insufficient for the plant with quadratic measurement function. Hence, EKF has the worst estimation quality, and EKPF method is

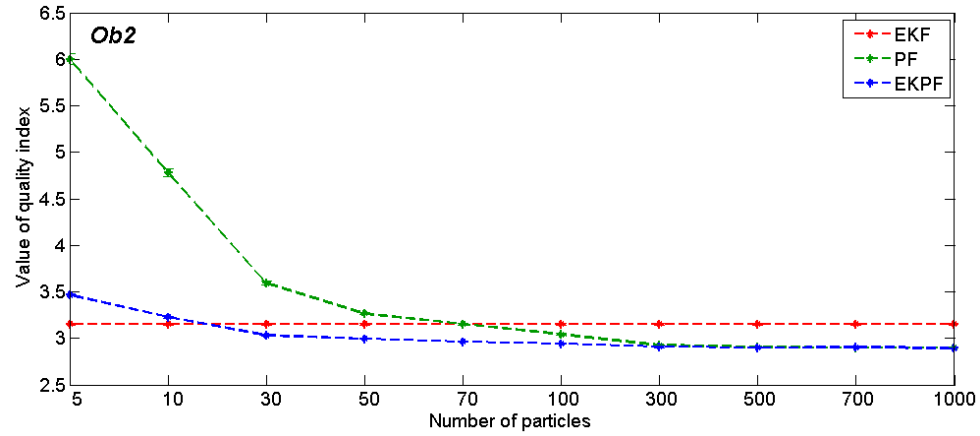

Fig. 2. Values of $aRMSE$ for $Ob2$

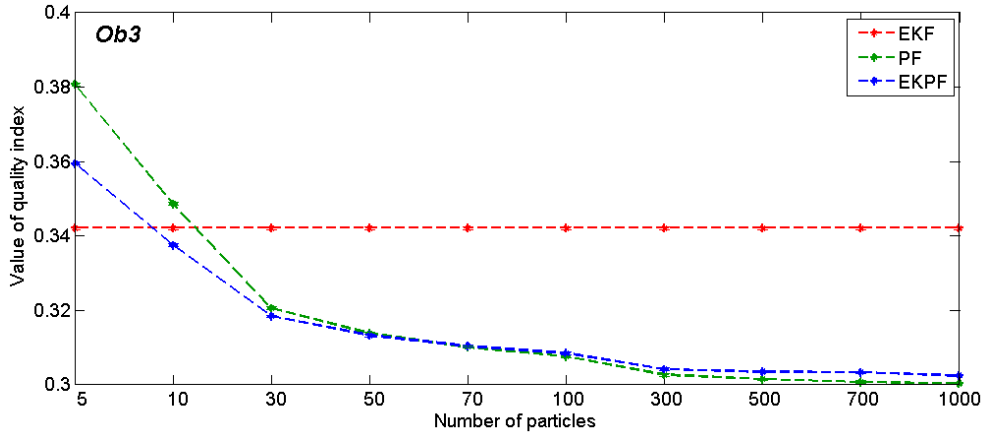

Fig. 3. Values of $aRMSE$ for $Ob3$

worse than PF (drawing particles, in the prediction step, from the PDF based on EKF mean and covariance deteriorates estimation quality).

For higher number of particles values of quality index is similar for PF and EKPF methods, which confirms that EKPF usage makes sense only for low particles number (because additional calculations extend computation time).

In the future, the authors plan to deepen studies related to hybrid filters, which combine PF and KF methods, such as Mixed Kalman Particle Filter or Iterated Extended Kalman Particle Filter, which have been mentioned in the introduction.

REFERENCES

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[2] J. V. Candy. Bayesian signal processing. *WILEY*, pages 36–44, 237–298, 2009.

[3] H. Chen, X. Liu, C. She, and C. Yao. Power system dynamic state estimation based on a new particle filter. *Procedia Environmental Sciences*, 11:655–661, 2011.

[4] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. In *handbook of Nonlinear Filtering 2009/12*, pages 656–704, 2012.

[5] S. H. (Ed.). *Kalman Filtering and Neural Networks*. Wiley, New York, 2001.

[6] A. Florek and P. Mazurkiewicz. *Sygnały i systemy dynamiczne*. Wydawnictwo Politechniki Poznańskiej, 2015.

[7] J. N. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings-F*, 140(2):107–113, 1993.

[8] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, Seabrook, 1970.

[9] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[10] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[11] G. Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1):1–25, 1996.

[12] P. Kozierski, M. Lis, and J. Zietkiewicz. Resampling in particle filtering – comparison. *Studia z Automatyki i Informatyki*, 38:35–64, 2013.

[13] F. Li, L. Wu, P. Shi, and C.-C. Lim. State estimation and sliding mode control for semi-markovian jump systems with mismatched uncertainties. *Automatica*, 51:385–393, 2015.

[14] L. Liang-Qun, J. Hong-Bing, and L. Jun-Hui. The iterated extended kalman particle filter. In *In Communications and Information Technology, ISCIT 2005*, pages 1213–1216, Beijing, China, October 2005.

[15] R. Merwe, A. Doucet, N. Freitas, and E. Wan. The unscented particle filter. *Technical Report CUED/F-INFENG/TR 380*, 2000.

[16] J. Mountney, I. Obeid, and D. Silage. Modular particle filtering fpga hardware architecture for brain machine interfaces. In *Conf Proc IEEE Eng Med Biol Soc.*, pages 4617–4620, Boston, Massachusetts USA, 2011.

[17] L. Murray. Gpu acceleration of the particle filter: the metropolis resampler. *arXiv preprint arXiv:1202.6163*, 2012.

[18] D. Simon. *Optimal State Estimation*. WILEY, New Jersey, 2006.

[19] S. Sutharsan, T. Kirubarajan, T. Lang, and M. McDonald. An optimization-based parallel particle filter for multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 48(2):1601–1618, 2012.

[20] E. A. Wan and R. V. D. Merwe. The unscented kalman filter for nonlinear estimation. *Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 153–158, 2000.

ABSTRACT

In this paper, three state estimation algorithms, namely: Extended Kalman Filter, Particle Filter (Bootstrap Filter) and Extended Kalman Particle Filter, have been presented. Particle Filter and Extended Kalman Particle Filter algorithms have been compared with a different number of particles and the results have been presented together with Extended Kalman Filter. Estimation quality has been checked

for three nonlinear objects (one- and multidimensional systems) and evaluated through the $aRMSE$ quality index value. Based on the obtained results it was concluded that Extended Kalman Particle Filter provide better estimation quality for low number of particles in comparison to simple particle filter. However it is not met for highly nonlinear system.

PORÓWNANIE ALGORYTMÓW FILTRU CZĄSTECZKOWEGO I ROZSZERZONEGO CZĄSTECZKOWEGO FILTRU KALMANA

STRESZCZENIE

W pracy zostały zaprezentowane trzy algorytmy estymacji – rozszerzony filtr Kalmana, filtr cząsteczkowy (algorytm Bootstrap) i rozszerzony cząsteczkowy filtr Kalmana. Algorytmy filtru cząsteczkowego i rozszerzonego cząsteczkowego filtru Kalmana zostały porównane dla różnej liczby cząsteczek, a wyniki zestawione z wynikami działania rozszerzonego filtru Kalmana. Jakość estymacji została sprawdzona dla trzech nieliniowych obiektów (systemy jedno- i wielowymiarowe) i oceniona za pomocą wskaźnika jakości $aRMSE$. Na podstawie otrzymanych wyników stwierdzono, że rozszerzony cząsteczkowy filtr Kalmana zapewnia lepszą jakość estymacji dla niewielkiej liczby cząsteczek w porównaniu do zwykłego filtru cząsteczkowego. Jednakże nie jest to spełnione dla silnie nieliniowego obiektu.

Received: 2017-11-14
Revised: 2017-12-16
Accepted: 2017-12-20