LIKELIHOOD PARTICLE FILTER AND ITS PROPOSED MODIFICATIONS

Keywords: state estimation, Hybrid Kalman Filter, Hybrid Kalman Particle Filter, Likelihood Particle Filter

1. INTRODUCTION

A very important branch of science in the noisy measurements environment is the state estimations of dynamical systems. It is used, for example, in mobile and flying robots (UAVs) [4, 9, 10, 15] or in the power systems [1, 6]. All measurements in technical (and nontechnical) problems are given with measurement errors, so state estimation is used to filter out this noises. Different estimation methods work better with different applications. It was proven in [2] that for specific object with quadratic measurement function Likelihood Particle Filter provides best estimated measurements fitting.

In this paper, the authors focused on unpopular methods: Hybrid Kalman Filter, Hybrid Kalman Particle Filter [13] and Likelihood Particle Filter [2]. Likelihood Particle Filter and its proposed modifications were compared in this article with other estimation methods. The algorithms applied in this article, namely: Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) and Bootstrap Particle Filter (BPF), were widely explained in [11, 12].

Section 2 contains formulation of the problem. Sections 3-5 present Hybrid Kalman Filter, Hybrid Kalman Particle Filter and Likelihood Particle Filter algorithms, respectively. In Section 6, the examined plants and modifications of Likelihood Particle Filter algorithms proposed by the authors are presented. Section 7 contains the results of simulations, and conclusions one can find in Section 8.

2. FORMULATION OF THE PROBLEM

The discrete system is given by state space equations

\[
\begin{align*}
    x^{(k+1)} &= f(x^{(k)}, u^{(k)}; k) + v^{(k)} \\
    y^{(k)} &= h(x^{(k)}) + n^{(k)}
\end{align*}
\]

\( ^a \text{Poznan University of Technology, Faculty of Electrical Engineering, Institute of Control, Robotics and Information Engineering, Division of Control and Robotics, e-mail: jacek.michalski95@wp.pl} \)

\( ^b \text{Poznan University of Technology, Faculty of Electrical Engineering, Institute of Control, Robotics and Information Engineering, and also Poznan University of Technology, Faculty of Computing, Institute of Automation and Robotics, Division of Signal Processing and Electronic Systems, Piotrowo 3a street, 60-965 Poznań, e-mail: piotr.kozierski@gmail.com} \)

\( ^c \text{Poznan University of Technology, Faculty of Electrical Engineering, Institute of Control, Robotics and Information Engineering, Division of Control and Robotics, e-mail: joanna.zietkiewicz@put.poznan.pl} \)

\( ^d \text{Poznan University of Technology, Faculty of Electrical Engineering, Institute of Control, Robotics and Information Engineering, Division of Control and Robotics, e-mail: wojciech.giernacki@put.poznan.pl} \)

© Authors and Poznańskie Towarzystwo Przyjaciół Nauk 2018, sait.cie.put.poznan.pl
where $\mathbf{x}^{(k)}$ is a state vector, $\mathbf{u}^{(k)}$ is an input vector, $\mathbf{y}^{(k)}$ is a measurement vector, $\mathbf{v}^{(k)}$ is a process noise vector, $\mathbf{n}^{(k)}$ is a measurement noise vector, $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ are vectors with transition and measurement nonlinear functions, respectively. The task is to reconstruct values of the state variables from the available measured outputs and known inputs of the system.

3. **Hybrid Kalman Filter**

This method was proposed in [13] and combines EKF and UKF algorithms, so Hybrid Kalman Filter (HKF) can improve better adjustment of estimates than EKF and UKF separately. Kalman filters are widely used in state estimation of dynamical linear and nonlinear objects, because these algorithms are simple and work fast. EKF algorithm uses linearization at a given point. In the UKF algorithm, the selection of sigma points and unscented transformation is used. More information about EKF and UKF algorithms one can find in [12]. HKF algorithm combines strengths and weakness of EKF and UKF methods.

**Algorithm 1: Hybrid Kalman Filter**

1. Initialization – according to UKF algorithm.
   
   $\hat{\mathbf{x}}^{(0|0)}: \mu^{(0|0)} = E[\hat{\mathbf{x}}^{(0|0)}], \quad \mathbf{P}^{(0|0)} = E[(\hat{\mathbf{x}}^{(0|0)} - \mu^{(0|0)})(\hat{\mathbf{x}}^{(0|0)} - \mu^{(0|0)})^T];$

   set step $k := 1$.

2. Calculate estimates $\hat{\mathbf{x}}^{(k|k)}_{\text{UKF}}$ and $\mathbf{P}^{(k|k)}_{\text{UKF}}$ using the UKF algorithm [12].

   $\left[\hat{\mathbf{x}}^{(k|k)}_{\text{UKF}}, \mathbf{P}^{(k|k)}_{\text{UKF}}\right] = \text{UKF}\left[\hat{\mathbf{x}}^{(k-1)}_{\text{UKF}}, \mathbf{P}^{(k-1)}_{\text{UKF}}, \mathbf{y}^{(k)}\right]$

   – the choice of the sigma points

   \begin{align*}
   \hat{\mathbf{x}}^{(k-1)}_0 &= \hat{\mathbf{x}}^{(k-1)}, \\
   \hat{\mathbf{x}}^{(k-1)}_i &= \hat{\mathbf{x}}^{(k-1)} + \left(\sqrt{\frac{N_x}{1-W_0}} \mathbf{P}^{(k-1)}\right)_{ii}, \quad i = 1, \ldots, N_x, \\
   \hat{\mathbf{x}}^{(k-1)}_{N_x+i} &= \hat{\mathbf{x}}^{(k-1)} - \left(\sqrt{\frac{N_x}{1-W_0}} \mathbf{P}^{(k-1)}\right)_{ii}, \quad i = 1, \ldots, N_x
   \end{align*}

   – prediction step (time update):

   \begin{align*}
   \hat{\mathbf{x}}^{(k|k-1)}_j &= f(\hat{\mathbf{x}}^{(k-1)}_j, \hat{\mathbf{x}}^{(k-1)}; k), \quad \hat{\mathbf{x}}^{(k|k-1)} = \sum_{j=0}^{2N_x} W_j \hat{\mathbf{x}}^{(k|k-1)}_j, \\
   \mathbf{P}^{(k|k-1)} &= \sum_{j=0}^{2N_x} W_j \left(\hat{\mathbf{x}}^{(k|k-1)}_j - \hat{\mathbf{x}}^{(k|k-1)}\right)\left(\hat{\mathbf{x}}^{(k|k-1)}_j - \hat{\mathbf{x}}^{(k|k-1)}\right)^T + Q
   \end{align*}
\[ z_{j}^{(k|k-1)} = b_{j}^{(k|k-1)}, \quad \tilde{z}_{j}^{(k|k-1)} = \sum_{j=0}^{2N} W_{j} z_{j}^{(k|k-1)}, \quad (7) \]

\[ P_{\tilde{y}}^{(k|k-1)} = \sum_{j=0}^{2N} W_{j} \left( y_{j}^{(k|k-1)} - \tilde{z}_{j}^{(k|k-1)} \right) \left( y_{j}^{(k|k-1)} - \tilde{z}_{j}^{(k|k-1)} \right)^{T} + R, \quad (8) \]

\[ P_{\tilde{y}}^{(k|k-1)} = \sum_{j=0}^{2N} W_{j} \left( y_{j}^{(k|k-1)} - \tilde{z}_{j}^{(k|k-1)} \right) \left( y_{j}^{(k|k-1)} - \tilde{z}_{j}^{(k|k-1)} \right)^{T}, \quad (9) \]

- filtration step (measurement update)

\[ K^{(k)} = P_{\tilde{y}}^{(k|k-1)} \left( P_{\tilde{y}}^{(k|k-1)} \right)^{-1}, \quad (10) \]

\[ \tilde{x}_{\text{UKF}}^{(k|k)} = \tilde{x}_{\text{UKF}}^{(k|k-1)} + K^{(k)} \left( y_{k}^{(k)} - \tilde{y}_{\text{UKF}}^{(k|k-1)} \right), \quad (11) \]

\[ P_{\text{UKF}}^{(k|k)} = P_{\text{UKF}}^{(k|k-1)} - K^{(k)} P_{\tilde{y}}^{(k|k-1)} K^{(k)^{T}}. \quad (12) \]

3. Calculate estimates \( \tilde{x}_{\text{EKF}}^{(k|k)} \) i \( P_{\text{EKF}}^{(k|k)} \) using the EKF algorithm [12], with the take results from Step 2 taken as input values to this algorithm

\[ \left[ \tilde{x}_{\text{EKF}}^{(k|k)}, P_{\text{EKF}}^{(k|k)} \right] = \text{EKF} \left[ \tilde{x}_{\text{UKF}}^{(k|k)}, P_{\text{UKF}}^{(k|k)}, y_{k}^{(k)} \right], \]

- prediction step (time update)

\[ \tilde{x}_{\text{EKF}}^{(k|k-1)} = a \left( \tilde{x}_{\text{EKF}}^{(k|k)}, y_{k}^{(k-1)}; k \right), \quad (13) \]

\[ P_{\text{EKF}}^{(k|k-1)} = F^{(k-1)} P_{\text{EKF}}^{(k|k)} F^{(k-1)^{T}} + Q, \quad (14) \]

- filtration step (measurement update)

\[ K^{(k)} = P_{\text{EKF}}^{(k|k-1)} H^{(k)^{T}} \left[ H^{(k)} P_{\text{EKF}}^{(k|k-1)} H^{(k)^{T}} + R \right]^{-1}, \quad (15) \]

\[ \tilde{x}_{\text{EKF}}^{(k|k)} = \tilde{x}_{\text{EKF}}^{(k|k-1)} + K^{(k)} \left( y_{k}^{(k)} - H \tilde{x}_{\text{EKF}}^{(k|k-1)} \right), \quad (16) \]

\[ P_{\text{EKF}}^{(k|k)} = \left[ I - K^{(k)} H \right] P_{\text{EKF}}^{(k|k-1)}. \quad (17) \]

4. Set \( \tilde{x}_{\text{EKF}}^{(k)} = \tilde{x}_{\text{EKF}}^{(k|k)} \) i \( P_{\text{EKF}}^{(k)} = P_{\text{EKF}}^{(k|k)} \), update the time step \( k := k + 1 \), go to Step 2.

4. Hybrid Kalman Particle Filter

This method was proposed in [13]. This is a modification of Particle Filter where particles in the prediction step are drawn not from the transition model, but from the PDF obtained by HKF method. Hence, the weight calculation in (18) is based on the importance function, in contrast to Bootstrap Particle Filter when weights are calculating from the measurement model only.
Algorithm 2: Hybrid Kalman Particle Filter (HKPF)

1. Initialization (for each particle like in HKF algorithm). Set time step \( k := 1 \).
   \[
   \tilde{x}^{(0)}_i, \mu^{(0)}_i = E[\tilde{x}^{(0)}], \\
P^{(0)} = E[(\tilde{x}^{(0)} - \mu^{(0)})(\tilde{x}^{(0)} - \mu^{(0)})^T].
   \]

2. Calculate estimates \(\tilde{x}^{i,(k)}_{\text{UKF}}, P^{i,(k)}_{\text{UKF}}\) and \(\tilde{x}^{i,(k)}_{\text{EKF}}, P^{i,(k)}_{\text{EKF}}\) using the UKF algorithm.
   \[
   \begin{bmatrix}
   \tilde{x}^{i,(k)}_{\text{UKF}} \\
P^{i,(k)}_{\text{UKF}}
   \end{bmatrix} = \text{UKF}\left[\begin{bmatrix}
   \tilde{x}^{i,(k-1)}_i, & P^{i,(k-1)}_i, & \tilde{y}^k
   \end{bmatrix}\right].
   \]

3. Calculate estimates \(\tilde{x}^{i,(k)}_{\text{EKF}}, P^{i,(k)}_{\text{EKF}}\) using EKF algorithm, with the take results from Step 2 taken as input values to this algorithm
   \[
   \begin{bmatrix}
   \tilde{x}^{i,(k)}_{\text{EKF}} \\
P^{i,(k)}_{\text{EKF}}
   \end{bmatrix} = \text{EKF}\left[\begin{bmatrix}
   \tilde{x}^{i,(k)}_{\text{UKF}}, & P^{i,(k)}_{\text{UKF}}, & \tilde{y}^k
   \end{bmatrix}\right].
   \]

4. Draw \(\tilde{x}^{i,(k)} \sim g(\tilde{x}^{i,(k)}|\tilde{x}^{i,(k-1)}, \tilde{y}^k) = N(\tilde{x}^{i,(k)}_{\text{EKF}}, P^{i,(k)}_{\text{EKF}})\).

5. Calculate particles weights according to equation
   \[
   q_{i,(k)} = \frac{p(y^k|x^{i,(k)})p(x^{i,(k)}|x^{i,(k-1)})}{g(x^{i,(k)}|x^{i,(k-1)}, \tilde{y}^k)}, \quad (18)
   \]

6. Normalization. Scale the weights in such a way that their sum equals 1.

7. Resampling (systematic resampling has been used [8]).

8. End of the iteration. Calculate estimate \(\tilde{x}^{i,(k)} = P^{i,(k)}_{\text{EKF}}\)
   go to step 2.

More information about particle filtering one can find in [2, 11, 12].

5. LIKELIHOOD PARTICLE FILTER

Likelihood Particle Filter algorithm was proposed in [2]. This is a modification of Particle Filter, where particles in the prediction step are drawn not from the transition model, but from the measurement model, and weights are calculated not based on the measurement model, but based on the transition model. This algorithm was prepared specially for specific object, with quadratic measurement function, i.e. Ob1 from Section 6.

In the LPF algorithm, the new variable \(g_{i,(k)}\) has been introduced. This is the square of state variable \(x_{i,(k)} = x_{i,(k)}^2\), so this variable have to be greater than zero.

Algorithm 3: Likelihood Particle Filter

1. Draw initial values of particles \(x_{i,(0)} \sim p(x_{(0)})\), set time step \( k = 1 \).

2. Draw \(N_{\mu}\) values of particles from measurement model \(x_{i,(k)} \sim p(y^k|x_{i,(k)}) \propto p(x_{i,(k)}|y^k)\).
   The draw for each particle should be repeated until \(g_{i,(k)} = x_{i,(k)}^2 \geq 0\)
3. For each particle draw \( w^i \sim U[0, 1] \). If \( w^i > 0.5 \) then \( z^{i,(k)} = \sqrt{z^{i,(k)}} \), otherwise \( z^{i,(k)} = -\sqrt{z^{i,(k)}} \).

4. Calculate the particles’ weights using the equation
\[
q^{i,(k)} = p(z^{i,(k)} | z^{i,(k-1)})z^{i,(k)}.
\]

5. Normalization. Scale the weights in such a way that their sum equals 1.

6. Resampling (systematic resampling has been used [8]).

7. End of the iteration. Calculate the estimate from \( k \)-th time step, update the step \( k := k + 1 \), and go to Step 2.

More information about LPF algorithms one can find in [13]. This method is useful when we need to have the best adjustment of object’s outputs, not state variables.

6. EXAMINED OBJECTS AND LPF’S MODIFICATIONS

Systems Ob1 and Ob2 are autonomous objects (without input signals). Ob1 is used very often in particle filters examination [2, 5]. According to [7], this object was firstly proposed in 1978 by Netto, Gimeno and Mendes. Ob2 is the modification of Ob1 through simplification of measurement function. Ob3 is MIMO system from [14]. It has 3 inputs and 2 outputs. Ob4 is a multidimensional linear system proposed by the authors. For the latter objects, the special modification of LPF algorithm has been shown.

- Ob1:
\[
\begin{align*}
    x^{(k+1)} &= 0.5x^{(k)} + \frac{25x^{(k)}}{1 + x^{(k)^2}} + 8\cos(1.2k) + v^{(k)} \\
    y^{(k)} &= x^{(k)^2} + n^{(k)},
\end{align*}
\]
\[
v^{(k)} \sim \mathcal{N}(0; 10),
\]
\[
n^{(k)} \sim \mathcal{N}(0; 1),
\]
\[
x^{(0)} = 0.1.
\]

- Ob2:
\[
\begin{align*}
    x^{(k+1)} &= 0.5x^{(k)} + \frac{25x^{(k)}}{1 + x^{(k)^2}} + v^{(k)} \\
    y^{(k)} &= 2x^{(k)} + n^{(k)},
\end{align*}
\]
\[
v^{(k)} \sim \mathcal{N}(0; 10),
\]
\[
n^{(k)} \sim \mathcal{N}(0; 10^2),
\]
\[
x^{(0)} = 0.1.
\]

- Ob3:
\[
\begin{align*}
    x_1^{(k+1)} &= 0.5 (x_1^{(k)^2})^{\frac{1}{2}} + 0.3x_2^{(k)}x_3^{(k)} + 0.2u_1^{(k)} + v_1^{(k)} \\
    x_2^{(k+1)} &= 0.5 (x_2^{(k)^2})^{\frac{1}{2}} + 0.3x_3^{(k)}x_1^{(k)} + 0.2u_2^{(k)} + v_2^{(k)} \\
    x_3^{(k+1)} &= 0.5 (x_3^{(k)^2})^{\frac{1}{2}} + 0.3x_1^{(k)}x_2^{(k)} + 0.2u_3^{(k)} + v_3^{(k)} \\
    y_1^{(k)} &= 0.5 (x_1^{(k)} + x_2^{(k)} + x_3^{(k)}) + n_1^{(k)} \\
\end{align*}
\]
\[
y_2^{(k)} = 2 (x_1^{(k)})^2 + n_2^{(k)} \]

© Authors and Poznańskie Towarzystwo Przyjaciół Nauk 2018, sait.cie.put.poznan.pl
\( v_{1/2/3}^{(k)} \sim \mathcal{N}(0; 0.1), \)
\( n_{1/2}^{(k)} \sim \mathcal{N}(0; 0.1), \)
\( u_{1/2/3}^{(k)} \sim \mathcal{U}[-1;1], \)
\( x^{(0)} = [0.1, 0.1, 0.1]^T. \)

- **Ob4**

\[
\begin{align*}
\begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ y_1^{(k)} \\ y_2^{(k)} \end{bmatrix} &= \begin{bmatrix} 0.705 & 0.095 \\ 0.195 & 0.605 \\ 0.800 & 0.005 \\ 0.005 & 0.800 \end{bmatrix} \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ y_1^{(k)} \\ y_2^{(k)} \end{bmatrix} + \begin{bmatrix} v_1^{(k)} \\ v_2^{(k)} \end{bmatrix},
\end{align*}
\]

\( v_{1/2}^{(k)} \sim \mathcal{N}(0; 4), \)
\( n_{1/2}^{(k)} \sim \mathcal{N}(0; 10), \)
\( z^{(0)} = [0, 0]^T. \)

Following the basic LPF method, the authors proposed its modifications for objects other than Ob1 for which this algorithm was developed. The algorithms obtained through these modifications work not only with objects with quadratic measurement functions but also for multivariable and unobservable objects.

**Algorithm 4: Likelihood Particle Filter for plant without square measurement function Ob2**

1. Draw initial values of particles \( x^{i,(0)} \sim p(x^{(0)}), \) set time step \( k = 1. \)
2. Draw \( N_p \) values of particles from measurement model \( x^{i,(k)} \sim \tilde{p}(x^{(k)}|z^{(k)}) \propto p(x^{(k)}|z^{(k)}). \)
3. Calculate the particles’ weights using the equation
\[
q^{i,(k)} = p(x^{i,(k)}|z^{i,(k-1)}) \tag{20}
\]
4. Normalization. Scale the weights in such a way that their sum equals 1.
5. Resampling (systematic resampling has been used [8]).
6. End of the iteration. Calculate the estimate from \( k \)-th time step, update time step \( k := k + 1, \) go to step 2.

**Algorithm 5: Likelihood Particle Filter for unobservable plant Ob3**

1. Draw initial values of particles \( x^{i,(0)} \sim p(x^{(0)}), \) set time step \( k = 1. \)
2. For \( y_2^{(k)} \) draw \( N_p \) values of particles from measurement model \( x^{i,(k)} \sim \tilde{p}(y_2^{(k)}|s_1^{(k)}) \propto p(s_1^{(k)}|y_2^{(k)}). \) A draw for each particle should be repeated until \( s_1^{(k)} = x_1^{(k+1)^2} \geq 0. \)
For \( y_1^{(k)} \) draw \( N_p \) particles from measurement model \( x^{i,(k)} = \begin{bmatrix} x_1^{(i,k)} \\ x_2^{(i,k)} \\ x_3^{(i,k)} \end{bmatrix}^T \sim p(x^{(k)}|z^{i,(k-1)}). \)

© Authors and Poznański Towarzystwo Przyjaciół Nauk 2018, sait.cie.put.poznan.pl
3. For each particle \( x_1 \) draw \( w_i \sim \mathcal{U}[0, 1] \). If \( w_i > 0.5 \) then \( x_1^{i,(k)} = \sqrt{s_1^{i,(k)}} \), otherwise \( x_1^{i,(k)} = -\sqrt{s_1^{i,(k)}} \).

4. Calculate the particles’ weights for \( x_1 \) using the equation
\[
q_i^{1,(k)} = p(x_1^{i,(k)} | x_1^{i,(k-1)}) q_i^{i,(k)}
\]
and for \( x_{2/3} \) from the measurement model
\[
q_i^{2,(k)} = p(y^{i,(k)} | x_2^{i,(k)})
\]

5. Normalization. Scale the weights in such a way that their sum equals 1.

6. Resampling (systematic resampling has been used [8]).

7. End of the iteration. Calculate the estimate from \( k\)-th time step, update time step \( k := k + 1 \), go to step 2.

For multivariable Ob4 LPF algorithm is the same like for Ob2, but measurement functions are not explicit functions of state variables, so it is necessary to determine the state variables in the functions of measured values:
\[
\begin{align*}
&\{ x_1^{(k)} = \lambda_1(y_1^{(k)}, y_2^{(k)}) \\
&x_2^{(k)} = \lambda_2(y_1^{(k)}, y_2^{(k)}) \\
&\{ y_1^{(k)} = 0.800x_1^{(k)} + 0.005x_2^{(k)} + n_1^{(k)} \\
&y_2^{(k)} = 0.005x_1^{(k)} + 0.800x_2^{(k)} + n_2^{(k)} \\
&\{ x_1^{(k)} = -0.078(y_2^{(k)} - n_2^{(k)}) + 1.2501(y_1^{(k)} - n_1^{(k)}) \\
&x_2^{(k)} = 1.2501(y_2^{(k)} - n_2^{(k)}) - 0.078(y_1^{(k)} - n_1^{(k)})
\end{align*}
\]

LPF modifications have been shown and examined for specific objects, but they can work also with another plants, in the case when they have the same properties.

7. RESULTS OF SIMULATIONS

Calculations were performed for every method and plant configurations. Each simulation was repeated 1.000 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 [3], i.e., the variance of the mean value is \( m \) times smaller than the variance from \( m \) samples, for Gaussian PDF.

The quality index \( \text{aRMSE} \) has been used, which is given by equations
\[
\begin{align*}
\text{MSE}_i &= \frac{1}{M} \sum_{k=1}^{M} (\hat{x}_i^{(k)} - x_i^{*(k)})^2, \quad (23) \\
\text{RMSE}_i &= \sqrt{\text{MSE}_i}, \quad (24) \\
\text{aRMSE} &= \frac{1}{N_x} \sum_{i=1}^{N_x} \text{RMSE}_i, \quad (25)
\end{align*}
\]
and the second quality index, which refers to measurements, is given by

\[
\epsilon_y = \frac{\sum_{k=1}^{M} \left( \frac{\sum_{i=1}^{N_x} |\hat{x}_i(k) - \hat{x}_i(k)|}{\sum_{i=1}^{N_y} |y_i(k) - y_i(k)|} \right)}{M}
\]

where \(M\) is the length of the simulation, \(N_x\) is the number of state variables, \(N_y\) is the number of measurements, \(\hat{x}_i(k)\), \(\hat{y}_i(k)\), and \(x_i^+(k)\), \(y_i^+(k)\) are estimated and real values, respectively, and \(y_i(k)\) is the noisy measurement, \(k\) marks the time step.

Quality indices of each methods for Ob1, Ob2, Ob3 and Ob4 have been presented in Figures 1-8. Standard deviations with 95% probability round-trip (according to 68-95-99.7 rule) have been presented in the graphs.

8. CONCLUSIONS

Based on the simulations results, one can see that estimation quality of each method depends on the object type choice. Ob1 is a strongly nonlinear plant and, as was proven in [11], for this object EKF algorithm works very poorly, hence HKF for Ob1 works bad too. HKF algorithm combines weakness of EKF and UKF, so for this objects its operation is unacceptable. Regarding the state variables match, above 30 particles, the best estimation quality provides BPF algorithm, and for small \(N_p\) number the UKF algorithm works the best. HKPF operation is worse than BPF, because EKF section for this system works very poorly. In measurement match (such choice of estimated state variables that the measure is the closest to the real one), the LPF algorithm is unacceptable, and even with few particles works better than any other algorithms with hundreds of particles.

Ob2 is the simplified version of Ob1, and for this plant the worst estimation quality can be observed when UKF algorithm is used. EKF works better for Ob2 than for Ob1, so for small number of particles the best estimation quality provides HKF algorithm, and HKPF works better than BPF and LPF’s modification. For large \(N_p\) the best results were observed for HKPF algorithm; \(\epsilon_y\) index (showing measurements fitting) gives very similar results here.

For Ob3 the best simulation results provide BPF and HKPF algorithms. In aRMSE these values are very similar, and for small \(N_p\) number better quality is provided by BPF algorithm. For \(\epsilon_y\) index, the best estimation quality can be observed for HKPF algorithm.

For multivariable Ob4, aRMSE and \(\epsilon_y\) results are very similar (except LPF algorithm). This is a linear plant, so the best estimation results provides KF algorithm (marked as EKF algorithm, this method for linear objects is optimal [12]). For small \(N_p\), UKF and HKF methods work better than BPF and HKPF, and along with the \(N_p\) increase the quality of BPF and HKPF (relative to BPF and HKPF) is getting better. Relatively poor estimation quality is provided by LPF modification.

As it has been proved in [11], the combination of Kalman Filter and Particle Filter algorithms works better than BPF for small number of particles. Increasing the number of particles, one can see that the quality of those algorithms becomes similar.

The proposed modification of LPF gives relatively good estimation quality, but only for one-dimensional plants. In the future authors plan to develop better modifications of this algorithm in such a way that the estimation quality will be even better than hitherto obtained.
Fig. 1. Values of $aRMSE$ for Ob1; the result for EKF has been presented as two times smaller (for better readability)

Fig. 2. Values of $\epsilon_y$ for Ob1; the result for EKF and HKF have been presented as two times smaller (for better readability)

REFERENCES


© Authors and Poznańskie Towarzystwo Przyjaciół Nauk 2018, sait.cie.put.poznan.pl
Fig. 3. Values of $aRMSE$ for Ob2

Fig. 4. Values of $\epsilon_y$ for Ob2


Fig. 5. Values of $aRMSE$ for $Ob3$

Fig. 6. Values of $\epsilon_y$ for $Ob3$


In this paper, three methods, namely: Hybrid Kalman Filter, Hybrid Kalman Particle Filter, and Likelihood Particle Filter for state estimation have been presented. These algorithms have been applied to three nonlinear objects and one linear object (one- and multivariable systems) and have been compared with Bootstrap Particle Filter. Moreover, authors proposed three modifications of Likelihood Particle Filter, intended for different types of objects. Operation of three particle Filter algorithms, namely Bootstrap Particle Filter, Hybrid Kalman Particle Filter and Likelihood Particle Filter, have been compared for a different number of particles, and the results have been presented together with Extended Kalman Filter, Unscented Kalman Filter and Hybrid Kalman Filter algorithm. It has been shown that Hybrid Kalman Particle Filter gives better results than Bootstrap Particle Filter for a low number of particles. Furthermore, Likelihood Particle Filter provides better than other examined methods out-
put match (through suitable choice of estimated state variables). For the linear object, Kalman Filter algorithm is the best.

WIARYGODNY FILTR CZĄSTECZKOWY I JEGO Modyfikacje

Jacek Michalski, Piotr Kozierski, Joanna Zietkiewicz, Wojciech Giernacki


Received: 2018-09-14
Accepted: 2018-11-05

© Authors and Poznańskie Towarzystwo Przyjaciół Nauk 2018, sait.cie.put.poznan.pl