New Grid for Particle Filtering of Multivariable Nonlinear Objects

Piotr Kozierski\textsuperscript{1,2), Jacek Michalski\textsuperscript{2), Talar Sadalla\textsuperscript{2), Wojciech Giernacki\textsuperscript{2)), Joanna Ziętkiewicz\textsuperscript{2), Szymon Drgas\textsuperscript{1)}}

Poznan University of Technology, Piotrowo street 3a, 60-965 Poznan, Poland\textsuperscript{1,2)}
Faculty of Computing, Institute of Automation and Robotics, Division of Signal Processing and Electronic Systems\textsuperscript{3)}
Faculty of Electrical Engineering, Institute of Control, Robotics and Information Engineering, Division of Control and Robotics\textsuperscript{2)}
Email: piotr.kozierski@gmail.com, jacek.michalski95@wp.pl

Abstract—In the paper a new grid (potentially linear, nonlinear and even semi-Markovian jump system) was presented. All transition and measurement functions were proposed. Moreover, the transition functions of two types were considered – dependent on one and many different state variables. Also 10 types of measurements were proposed for both nodal and branch cases. Based on the obtained results one can see, which measurement functions are “easy”, and which are “hard” for state estimation task.

I. INTRODUCTION

PARTICLE filter (PF) is potentially very good estimation method because is based on the optimal solution – Bayes filter. The biggest disadvantage of PFS is their need for computational power – number of calculations grows exponentially with a system variables number \([1]\). This is the reason why PF methods are usually used only for very small plants.

Some solutions to this problem are hybrid filters, e.g. Rao-Blackwellized PF (RBPF) \([2]-[3]\), or Marginalized PF \([4]\), in which state variables are divided into two groups – one group is estimated using PF method, and the second group – using Kalman Filter (KF) methods (linear, extended or unscented).

Another solution was proposed in \([5]\) – all state variables are divided into groups; however, the disadvantage of this method is loss of information contained in measurements, which uses state variables from two or more groups.

Dispersed Particle Filter (DPF) was proposed by the authors in one of the previous works \([6]\) – this method assumes that dependences between state variables and measurements are relatively sparse (transition and measurement models depend on relatively small number of different state variables). Unfortunately, previously used plants – the power systems – have number of state variables two times higher than number of nodes. Additionally, based on studies from \([7]\), the reasonable number of particles should be about 4 to the power of plant variables number. This caused reduction of plant dimension and simultaneously increase the number of nodes is needed.

For this reason the authors proposed a new grid for further research. In this network, one node is associated with one state variable – thanks to this, considerations on the network structure will be possible for systems with relatively small state vector length. Proposed grid is very general and one can model both linear and highly nonlinear models (for both transitions and measurements).

In the second section particle filter algorithm is described. In Section III the proposed grid is presented. Prepared simulations and obtained results are shown in the fourth section. The last section contains drawn conclusions.

II. PARTICLE FILTER

An object in state space can be written as
\[
\begin{align*}
\mathbf{x}^{(k)} &= f(\mathbf{x}^{(k-1)}, \mathbf{u}^{(k)}, k) + \mathbf{v}^{(k)} \\
\mathbf{z}^{(k)} &= h(\mathbf{x}^{(k)}, \mathbf{u}^{(k)}, k) + \mathbf{n}^{(k)}
\end{align*}
\]

where \(\mathbf{x}^{(k)}\) is a state vector, \(\mathbf{u}^{(k)}\) is input vector, \(\mathbf{z}^{(k)}\) is output vector, and vectors \(\mathbf{v}^{(k)}\) and \(\mathbf{n}^{(k)}\) are internal and measurement noises, respectively – all at \(k\)-th time step. The main task of particle filter is to estimate state vector based on the measurements and input signals.

The particle filters operation principle is based on the recursive Bayesian filtering \([8]\)
\[
\frac{p(\mathbf{x}^{(k)} \mid \mathbf{Z}^{(k)})}{p(\mathbf{x}^{(k)} \mid \mathbf{Z}^{(k-1)})} = \frac{p(\mathbf{z}^{(k)} \mid \mathbf{x}^{(k)}, \mathbf{Z}^{(k-1)})}{p(\mathbf{z}^{(k)} \mid \mathbf{Z}^{(k-1)})} \times \frac{p(\mathbf{x}^{(k)} \mid \mathbf{Z}^{(k-1)})}{p(\mathbf{x}^{(k)} \mid \mathbf{Z}^{(k-1)})}
\]

where \(\mathbf{Z}^{(k)}\) is a set of measurement vectors from the first to \(k\)-th time step, \(p(\mathbf{x}^{(k)} \mid \mathbf{Z}^{(k)})\) is a posterior Probability Density Function (PDF), \(p(\mathbf{x}^{(k)} \mid \mathbf{Z}^{(k-1)})\) is a prior PDF, \(p(\mathbf{z}^{(k)} \mid \mathbf{x}^{(k)}, \mathbf{Z}^{(k-1)})\) is a likelihood, and \(p(\mathbf{z}^{(k)} \mid \mathbf{Z}^{(k-1)})\) is an evidence.

The key idea in PF is to implement the posterior PDF as a set of particles. The \(i\)-th particle is represented by a pair \(\{\mathbf{x}^{(i,k)}, q^{(i,k)}\}\) – value (state vector) and weight. Higher weight increases probability that the value \(\mathbf{x}^{(i,k)}\) is close to the real state vector. If the number of particles, \(N\), is high enough, the information about the posterior PDF contained in particles set is the same as in continuous function.
The first particle filter was proposed in 1993 by Gordon, Salmond and Smith [9] and was called Bootstrap Filter. Operation principle of this PF is presented in Algorithm 1.

Algorithm 1 – Bootstrap Filter
1. Initialization. Draw $N$ initial values $x^{(0)}$ from initial PDF $p(x^{(0)})$, set time step $k=1$.
2. Prediction. Draw $N$ new particles from the transition model $x^{(k)} \sim p(x^{(k)} | x^{(k-1)})$.
3. Update. Compute the particle weights based on the measurement model $q^{(k)} = p(z^{(k)} | x^{(k)})$.
4. Normalization. Normalize weights so that their sum be equal to 1.
5. Resampling. Draw $N$ new particles using the posterior PDF obtained in previous steps (the chance that particle will be drawn is equal to normalized weight).
6. End of iteration. Calculate the estimated state vector, increase the time step $k=k+1$, go to the second step.

PF can be used for both non-Gaussian distributions and complex transition models. Example of such complex system model can be semi-Markovian jump system [10], in which the whole system model can switch itself (with some probability) into other structures (other equations), and also can go back to the previous “system states”.

For more information about PFs, references [11]-[15] are recommended.

III. PROPOSED NETWORK

The authors proposed grid, which on the one hand can be easily prepared, but on the other hand provides wide possibilities in creation of new plants. This is why there are many complex and maybe even illegible options presented below; however, the most common networks will be presented in a very simple way. Moreover, if one needs to add any dependence, which was not specified, still there is a possibility to present this on the scheme.

The proposed network is composed of nodes and lines (branches). Two different nodes can be connected by line. With every $i$-th node exactly one state variable $x_i$ is associated with. Nodes can be represented in two ways – by circles or by squares. Transition function $f_i$ depends on the shape of $i$-th node.

Filled figures are associated with measurements that when placed on lines (branches), refer to state variables that are associated to those lines (branch measurements), and when placed near the nodes (or on the nodal branches), refer to state variables in those specific nodes (nodal measurement).

It was assumed that networks are autonomous and thus any designation for input signal is not presented.

Expressions for different transition functions of the first type (circles) are presented below, and their connections with scheme designations are described in Table I.

<table>
<thead>
<tr>
<th>Eqn.</th>
<th>Designation of node</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3)</td>
<td>$x_i^{(k+1)} = \begin{cases} \alpha x_i^{(k)} + v_i^{(k)} &amp; \text{if } n \in Z \ \alpha' x_i^{(k)} + v_i^{(k)} &amp; \text{if } n \not\in Z \end{cases}$</td>
<td>$\alpha$ and $n$ should be given on the scheme. However, if one parameter is omitted, it is assumed that this value is equal to 1.</td>
</tr>
</tbody>
</table>
below, and their connections with scheme designations are described in Table II.

\[ f_{i,j}^{k} = \begin{cases} \left(x_{i}^{(k)}\right)^{m} \cdot \mu_{i,j} & m \in \mathbb{Z} \\ \left|x_{i}^{(k)}\right|^{m} \cdot \mu_{i,j} & m \notin \mathbb{Z} \end{cases} \]  

(7)

\[ f_{i,j}^{k} = \begin{cases} \sin\left(x_{i}^{(k)} - \sin\left(x_{i}^{(k)}\right)\right) \cdot \mu_{i,j} & m \in \mathbb{Z} \\ \sin\left|x_{i}^{(k)} - \sin\left|x_{i}^{(k)}\right|\right) \cdot \mu_{i,j} & m \notin \mathbb{Z} \end{cases} \]  

(8)

\[ f_{i,j}^{k} = \ln\left(0.001 + \left|x_{i}^{(k)}\right|^{m}\right) \cdot \mu_{i,j} \]  

(9)

Line functions \( f_{i,j} \) can be used in both, transition and measurement functions. Proposed types of transition functions, which use values of other state variables, are presented below, and their designations are described in Table III.

\[ x_{i}^{(k+1)} = \alpha \left(\sum_{j=1}^{N} f_{i,j}^{k} + v_{i}^{k}\right) + x_{i}^{(k)} \]  

(10)

\[ x_{i}^{(k+1)} = \alpha \left(\sum_{j=1}^{N} f_{i,j}^{k} + v_{i}^{k}\right) + \beta x_{i}^{(k)} + v_{i}^{k} \]  

(11)

\[ x_{i}^{(k+1)} = \alpha \left(\cos\left(x_{i}^{(k)} \cdot \beta k^{n}\right) + \sum_{j=1}^{N} \cos\left(f_{i,j}^{k} \cdot \beta k^{n}\right)\right) + v_{i}^{k} \]  

(12)

\[ x_{i}^{(k+1)} = \begin{cases} f_{i}\left(x_{i}^{(k)}, u_{i}^{(k)}, k\right) + v_{i}^{k} & \text{with prob. } p_{i} \\ - f_{i}\left(x_{i}^{(k)}, u_{i}^{(k)}, k\right) + v_{i}^{k} & \text{with prob. } (1 - p_{i}) \end{cases} \]  

(13)

The type of internal noise one can describe in a text or mark on the scheme. PDF type with parameters should be connected with specific node by dashed line. Examples have been presented in Fig. 1.

Measurements are marked by filled figures on the scheme. Measurement designations on branches have different meaning than designations associated with nodes. Moreover, one should keep in mind that measurement location matters (the first index indicates near which node the measurement is located), because measurement functions generally are not symmetric. Possible nodal measurements in \( i \)-th node \( (P_{i}, Q_{i}, R_{i}, S_{i}, T_{i}) \) are described by equations (14)-(18), whereas possible branch measurements between \( i \)-th (at this node measurement is placed) and \( j \)-th nodes \( (P_{ij}, Q_{ij}, R_{ij}, S_{ij}, T_{ij}) \) are described by equations (19)-(23). Designations of specific measurements are presented in Table IV. To use another measurement function one can simply use new filled figure on the scheme (function should be explained in text).

\[ P_{i}^{k} = z_{i}^{k} = \alpha \cdot x_{i}^{(k)} + n_{i}^{k} \]  

(14)

\[ Q_{i}^{k} = z_{i}^{k} = \alpha \cdot \sum_{j=1}^{N} x_{j}^{(k)} \cdot \mu_{i,j} + n_{i}^{k} \]  

(15)

\[ R_{i}^{k} = z_{i}^{k} = \alpha \cdot x_{i}^{(k)} + \sum_{j=1}^{N} f_{i,j}^{k} + n_{i}^{k} \]  

(16)
\[ S_{ij}^{(k)} = z_{ij}^{(k)} = \alpha \cdot \sum_{j=1}^{N_s} x_j^{(k)} \mu_{ij} \sin \left( x_i^{(k)} - x_j^{(k)} - \mu_{ij} \right) + n_o^{(k)} \]  
\[ T_{ij}^{(k)} = z_{ij}^{(k)} = \alpha \cdot x_i^{(k)} \]  
\[ P_{ij}^{(k)} = z_{ij}^{(k)} = \alpha \cdot \left( x_i^{(k)} - x_j^{(k)} + n_o^{(k)} \right) \]  
\[ Q_{ij}^{(k)} = z_{ij}^{(k)} = \alpha \cdot x_i^{(k)} - x_j^{(k)} + n_o^{(k)} \]  
\[ R_{ij}^{(k)} = z_{ij}^{(k)} = \alpha \cdot \left( x_i^{(k)} + x_j^{(k)} \sin \left( x_i^{(k)} + x_j^{(k)} \right), j \right) + n_o^{(k)} \]  
\[ S_{ij}^{(k)} + z_{ij}^{(k)} + \left( x_i^{(k)} + x_j^{(k)} \mu_{ij} \cos \left( x_i^{(k)} - x_j^{(k)} - \mu_{ij} \right) \right) + n_o^{(k)} \]  
\[ T_{ij}^{(k)} + z_{ij}^{(k)} = \alpha \cdot \frac{\max \left( x_i^{(k)} + x_j^{(k)} \mu_{ij} \cos \left( x_i^{(k)} - x_j^{(k)} - \mu_{ij} \right) \right)}{0.01 + \min \left( A; B \right)} + n_o^{(k)} \]  
\[ A = \log \left( 0.01 + \left| x_i^{(k)} x_j^{(k)} \right| \right); \quad B = -0.1 \left( x_i^{(k)} + x_j^{(k)} \right)^2 + 2 \]

The measurement noise should be marked in the same way as the internal noise – by description in text or using dashed line on the scheme. Examples of measurement noise designations are presented in Table IV.

\[ \text{Eqn.} \quad \text{Measur. designation} \quad \text{Explanation} \]
\[ (14), \quad \text{i} \quad \text{i} \quad \text{Nodal measurements should be placed near the node or should be connected with node by the dashed line. The second case is needed if one want to add information about measurement noise or about measurement scaling.} \]
\[ (15), \quad \text{f} \quad \text{i} \quad \text{Branch measurements should be placed on the specific line. One should keep in mind that measurements are different at both ends of the branch, so the position of sign should be unambiguous. If one want to add information about measurement noise or its scale, also a dashed line should be used.} \]
\[ (16), \quad \text{i} \quad \text{j} \quad \text{Measurement designations can be marked in one orientation or if one wish – can be rotated; however, one should use one approach.} \]
\[ (17), \quad \text{i} \quad \text{j} \quad \text{Nodal measurements should be placed on the specific line. One should use one approach.} \]
\[ (18), \quad \text{j} \quad \text{i} \quad \text{Nodal measurements should be placed near the node or should be connected with node by the dashed line. The second case is needed if one want to add information about measurement noise or about measurement scaling.} \]

The increase of aRMSE with higher number of particles for objects Ob401 and Ob402 that linearly changed state variables can be better tracked than ones changed nonlinearly. Moreover, the best estimation quality was obtained for linear object (linear transition model and linear measurement functions).

It is also visible (by comparison of results for R or T cases for all three objects) that complexity of measurements also impact on estimation quality (easier measurements provide better tracking).

The increase of aRMSE with higher number of particles for object Ob401 and cases Q and T is probably caused by higher chance that specific set of drawn values, which are
wrong, will match the measurements (index in one of dozens simulation had very high value). Simultaneously it is clearly visible that \( Q \) and \( T \) measurements are the hardest for estimation task. One can see also that for others measurement types even object with jump functions (Ob401) can be properly estimated.

It is also interesting that estimation quality for case \( Q \) in object Ob402 is rather weak for small number of particles \( N \), but is very good for high particles number, whereas for case \( Q \) in object Ob401 the results are worst of all examined cases and objects. This is probably caused by jump functions in Ob401.

Proposed network will be widely used by the authors for further research.

Fig. 4 Scheme of the object Ob403

REFERENCES


[8] Candy J. V., Bayesian Signal Processing, WILEY, New Jersey 2009, pp. 36-44. DOI: 10.1002/9780470470430583


