Proceedings of the Federated Conference on Computer Science and Information Systems pp. 1073–1077ISSN 2300-5963 ACSIS, Vol. 15 DOI: 10.15439/2018F308

New Grid for Particle Filtering of Multivariable Nonlinear Objects

Piotr Kozierski[1,2)], Jacek Michalski[2)], Talar Sadalla[2)], Wojciech Giernacki[2)], Joanna Ziętkiewicz[2)], Szymon Drgas[1)]
Poznan University of Technology, Piotrowo street 3a, 60-965 Poznan, Poland[1,2)]
Faculty of Computing, Institute of Automation and Robotics, Division of Signal Processing and Electronic Systems[1)]
Faculty of Electrical Engineering, Institute of Control, Robotics and Information Engineering, Division of Control and Robotics[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

$$\begin{cases} \mathbf{x}^{(k+1)} = \mathbf{f}\left(\mathbf{x}^{(k)}, \mathbf{u}^{(k)}, k\right) + \mathbf{v}^{(k)} \\ \mathbf{z}^{(k)} = \mathbf{h}\left(\mathbf{x}^{(k)}, \mathbf{u}^{(k)}, k\right) + \mathbf{n}^{(k)} \end{cases}, \tag{1}$$

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]

$$\overbrace{p\left(\mathbf{x}^{(k)} \mid \mathbf{Z}^{(k)}\right)}^{\text{posterior}} = \frac{\overbrace{p\left(\mathbf{z}^{(k)} \mid \mathbf{x}^{(k)}\right)}^{\text{likelihood}} \cdot \overbrace{p\left(\mathbf{x}^{(k)} \mid \mathbf{Z}^{(k-1)}\right)}^{\text{prior}}}{\underbrace{p\left(\mathbf{z}^{(k)} \mid \mathbf{Z}^{(k-1)}\right)}_{\text{evidence}}}, \tag{2}$$

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

This work was not supported by any organization

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 $\mathbf{x}^{i,(0)}$ from initial PDF $p(\mathbf{x}^{(0)})$, set 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. Compute the particle weights based on the measurement model $q^{i,(k)} = p(\mathbf{z}^{(k)}|\mathbf{x}^{i,(k)})$.
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.

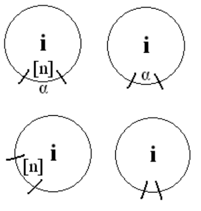
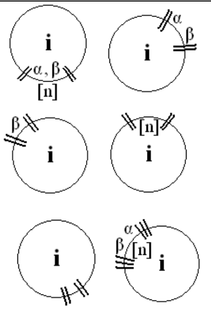
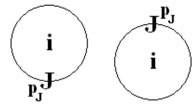
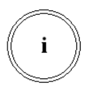
The proposed network is composed of nodes and lines (branches). Two different nodes can be connected by line. With every i-th node exactly one state variable x_i is associated with. Nodes can be represented in two ways – by circles or by squares. Transition function f_i depends on the shape of i-th node.

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 I.
Explanation of designations – nodes, part 1, for transition functions which are based only on i-th state variable

Eqn.	Designation of node	Explanation
(3)		α and n should be given on the scheme. However, if one parameter is omitted, it is assumed that this value is equal to 1. Also both parameters can be omitted (α=1 and n=1).
(4)-(5)	 i α, β [n] α β i β i [n] i i α β [n] i	When all three values are given, one must take into account that α should be written before β (above or on the left side of β). If any parameter (α, β or n) is not presented, it is assumed that it is equal to 1; however, one should keep in mind that β can be omitted only if α is also omitted. Designations for equation (5) are the same, but triple lines (through circles) should be used.
(6)		p_J should be written outside of the circle as J's sub- or super-script (also from the left). If p_J is omitted, it is assumed that p_J = 0.5.
-		To use another function (which must be explain in a text) a double circle should be used.

$$x_i^{(k+1)} = \begin{cases} \alpha\left(x_i^{(k)}\right)^n + v_i^{(k)} & n \in Z \\ \alpha\left|x_i^{(k)}\right|^n + v_i^{(k)} & n \notin Z \end{cases} \tag{3}$$

$$x_i^{(k+1)} = \begin{cases} \dfrac{\alpha\left(x_i^{(k)}\right)^n}{\beta + \left(x_i^{(k)}\right)^2} + v_i^{(k)} & n \in Z \\ \dfrac{\alpha\left|x_i^{(k)}\right|^n}{\beta + \left(x_i^{(k)}\right)^2} + v_i^{(k)} & n \notin Z \end{cases} \tag{4}$$

$$x_i^{(k+1)} = \begin{cases} \alpha\left(x_i^{(k)}\right)^n \cdot \cos(\beta k) + v_i^{(k)} & n \in Z \\ \alpha\left|x_i^{(k)}\right|^n \cdot \cos(\beta k) + v_i^{(k)} & n \notin Z \end{cases} \tag{5}$$

$$x_i^{(k+1)} = \begin{cases} f_i\left(x_i^{(k)}, \mathbf{u}^{(k)}, k\right) + v_i^{(k)} & \text{with prob. } p_J \\ -f_i\left(x_i^{(k)}, \mathbf{u}^{(k)}, k\right) + v_i^{(k)} & \text{with prob. } (1 - p_J) \end{cases} \tag{6}$$

For the second type of transition functions, connections between nodes matter. Branch between i-th and j-th nodes has a value $\mu_{i,j}$=$\mu_{j,i}$≠0, whereas if there is no connection between i-th and j-th nodes, branch value $\mu_{i,j}$=$\mu_{j,i}$=0. It is also assumed that $\mu_{i,i}$=1.

There are specified three types of lines, which differ in line functions $fl_{i,j}$ (where i is the number of node from which line function "was called"). These functions are written below, and their connections with scheme designations are described in Table II.

$$fl_{i,j}^{(k)} = \begin{cases} \left(x_j^{(k)}\right)^m \cdot \mu_{i,j} & m \in Z \\ \left|x_j^{(k)}\right|^m \cdot \mu_{i,j} & m \notin Z \end{cases} \tag{7}$$

$$fl_{i,j}^{(k)} = \begin{cases} \sin\left(\left(x_j^{(k)}\right)^m - x_i^{(k)}\right)\cdot \mu_{i,j} & m \in Z \\ \sin\left(\left|x_j^{(k)}\right|^m - x_i^{(k)}\right)\cdot \mu_{i,j} & m \notin Z \end{cases} \tag{8}$$

$$fl_{i,j}^{(k)} = \ln\left(0.001 + \left|x_j^{(k)}\right|^m\right)\cdot \mu_{i,j} \tag{9}$$

Line functions $fl_{i,j}$ 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.

$$x_i^{(k+1)} = \begin{cases} \alpha\left(x_i^{(k)}\right)^n \cdot \sum_{\substack{j=1\\j\neq i}}^{N_x} fl_{i,j}^{(k)} + v_i^{(k)} & n \in Z \\ \alpha\left|x_i^{(k)}\right|^n \cdot \sum_{\substack{j=1\\j\neq i}}^{N_x} fl_{i,j}^{(k)} + v_i^{(k)} & n \notin Z \end{cases} \tag{10}$$

$$x_i^{(k+1)} = \alpha \cdot \sum_{\substack{j=1\\j\neq i}}^{N_x} \frac{fl_{i,j}^{(k)}}{\beta + \left|fl_{i,j}^{(k)}\right|^n} + v_i^{(k)} \tag{11}$$

$$x_i^{(k+1)} = \alpha\left(\cos\left(x_i^{(k)}\cdot \beta k^n\right) + \sum_{\substack{j=1\\j\neq i}}^{N_x} \cos\left(fl_{i,j}^{(k)}\cdot \beta k^n\right)\right) + v_i^{(k)} \tag{12}$$

$$x_i^{(k+1)} = \begin{cases} f_i\left(\mathbf{x}^{(k)}, \mathbf{u}^{(k)}, k\right) + v_i^{(k)} & \text{with prob. } p_J \\ -f_i\left(\mathbf{x}^{(k)}, \mathbf{u}^{(k)}, k\right) + v_i^{(k)} & \text{with prob.} \left(1 - p_J\right) \end{cases} \tag{13}$$

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 ($P_{i,j}$, $Q_{i,j}$, $R_{i,j}$, $S_{i,j}$, $T_{i,j}$) 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).

TABLE II.
EXPLANATION OF DESIGNATIONS – LINES

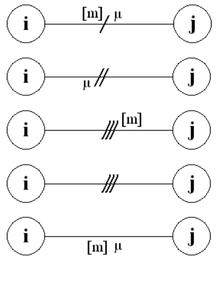
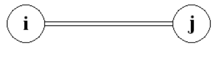
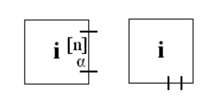
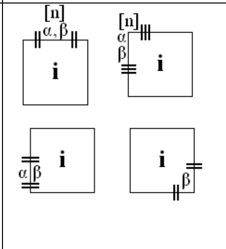
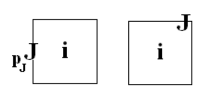
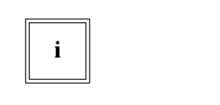
Eqn.	Line designation	Explanation
(7)-(9)		Values m and μ should be written on the scheme near to line center. Also information about branch type, in the form of diagonal lines (one for (7), two for (8) and three for (9)), should be presented there. If $\mu_{i,j}$=1, one can omit this value. If m=1 it also can be omitted. If branches in whole grid are only of first type, diagonal lines can be omitted.
-		For another line function (must be explain in text), double line should be used between nodes.

TABLE III.
EXPLANATION OF DESIGNATIONS – NODES, PART 2, FOR TRANSITION FUNCTIONS WHICH ARE BASED ALSO ON OTHER STATE VARIABLES

Eqn.	Designation of node	Explanation
(10)		n value should be written inside a square bracket. If α is omitted, it is assumed that α=1. If value n is omitted, it is assumed that n=1.
(11)-(12)	 [n] α, β i [n] α β i i α β i β	When all three values are given, one must take into account that α should be written before β (above or on the left side of β). If any parameter (α, β or n) is not presented, it is assumed that it is equal to 1; however, one should keep in mind that β can be omitted only if α is also omitted.
(13)		p_J should be written outside of the square as J's sub- or super-script (can be also from the left side). Omitted p_J means that p_J = 0.5.
-		To use another function (which must be explain in a text) double square should be used.

$$P_i^{(k)} = z_o^{(k)} = \begin{cases} \alpha \cdot \left(x_i^{(k)}\right)^m + n_o^{(k)} & m \in Z \\ \alpha \cdot \left|x_i^{(k)}\right|^m + n_o^{(k)} & m \notin Z \end{cases} \tag{14}$$

$$Q_i^{(k)} = z_o^{(k)} = \alpha \cdot \prod_{\substack{j=1\\ \mu_{i,j}\neq 0}}^{N_x} x_j^{(k)} \cdot \mu_{i,j} + n_o^{(k)} \tag{15}$$

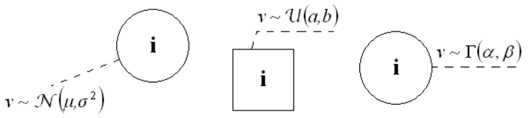

i
$v \sim \mathcal{N}(\mu,\sigma^2)$
$v \sim \mathcal{U}(a,b)$
i
i
$v \sim \Gamma(\alpha,\beta)$

Fig. 1 Examples of internal noise designation

$$R_i^{(k)} = z_o^{(k)} = \alpha \cdot x_i^{(k)} \cdot \sum_{\substack{j=1\\j\neq i}}^{N_x} fl_{i,j}^{(k)} + n_o^{(k)} \tag{16}$$

$$S_i^{(k)} = z_o^{(k)} = \\ = \alpha \cdot \sum_{j=1}^{N_x} x_i^{(k)} x_j^{(k)} \mu_{i,j} \sin\left(x_i^{(k)} - x_j^{(k)} - \mu_{i,j}\right) + n_o^{(k)} \quad (17)$$

$$T_i^{(k)} = z_o^{(k)} = \alpha \cdot x_i^{(k)} \cdot \\ \cdot \sum_{\substack{j=1 \\ j \neq i}}^{N_x} x_j^{(k)} \mu_{i,j} \cos\left(\frac{x_i^{(k)} - x_j^{(k)}}{\ln\left(1.01 + \left| fl_{i,j}^{(k)} \right|\right)} \right) + n_o^{(k)} \quad (18)$$

$$P_{i,j}^{(k)} = z_o^{(k)} = \alpha \cdot \left(x_i^{(k)} - x_j^{(k)}\right) + n_o^{(k)} \quad (19)$$

$$Q_{i,j}^{(k)} = z_o^{(k)} = \alpha \cdot x_i^{(k)} \cdot x_j^{(k)} + n_o^{(k)} \quad (20)$$

$$R_{i,j}^{(k)} = z_o^{(k)} = \alpha \cdot \\ \cdot \left(x_i^{(k)} + x_j^{(k)} \cdot \sin\left(\left(x_i^{(k)} + x_j^{(k)}\right) \cdot fl_{i,j}^{(k)}\right)\right) + n_o^{(k)} \quad (21)$$

$$S_{i,j}^{(k)} = z_o^{(k)} = \alpha \cdot \\ \cdot \left(x_i^{(k)2} - x_i^{(k)} x_j^{(k)} \mu_{i,j} \cos\left(x_i^{(k)} - x_j^{(k)} - \mu_{i,j}\right)\right) + n_o^{(k)} \quad (22)$$

$$T_{i,j}^{(k)} = z_o^{(k)} = \alpha \cdot \frac{\max\left\{x_i^{(k)} + x_j^{(k)} ; x_i^{(k)} \cdot x_j^{(k)}\right\}}{0.01 + \left|\min\{A ; B\}\right|} \cdot fl_{i,j}^{(k)} \\ A = \log_2\left(0.01 + \left|x_i^{(k)} x_j^{(k)}\right|\right); \quad B = -0.1\left(x_i^{(k)} + x_j^{(k)}\right)^2 + 2 \quad (23)$$

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.

IV. Simulation results

The simulations were performed for the networks presented in Fig. 2-4. Measurements were considered for 5 different cases – P, Q, R, S, and T. In each case all possible measurements of specific type were taken into account (10 branch measurements and 4 nodal measurements). However, one can see that differences between the second and third objects are only in line functions $fl_{i,j}$. hence only experiments for R and T cases were repeated for Ob403.

The average Root Mean Square Error was used to describe estimation quality and it can be written as

$$\text{aRMSE} = \frac{1}{N_x} \sum_{j=1}^{N_x} \text{RMSE}_j \,, \quad (24)$$

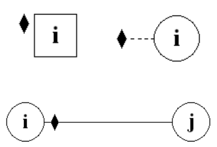
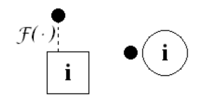
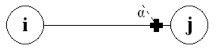
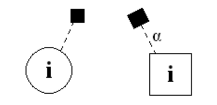
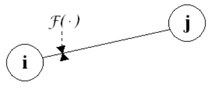
where N_x is the number of state variables, and

$$\text{RMSE}_j = \sqrt{\text{MSE}_j} = \sqrt{\frac{1}{M} \sum_{k=1}^{M} \left(x_j^{(k)} - x_j^{(k)+}\right)^2} \,, \quad (25)$$

where value with plus is the real value of state variable and M is the length of the simulation (set on the 1000 time steps in experiments).

All simulations were repeated 4000 times for Ob401, Q and T measurements, and 400 times for other cases. All results are presented in Fig. 5.

Table IV.
Explanation of designations – measurements (branch and nodal)

Eqn.	Measur. designation	Explanation
(14), (19)		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. 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. Measurement designations can be marked in one orientation or if one wish – can be rotated; however, one should use one approach.
(15), (20)		
(16), (21)		
(17), (22)		
(18), (23)		

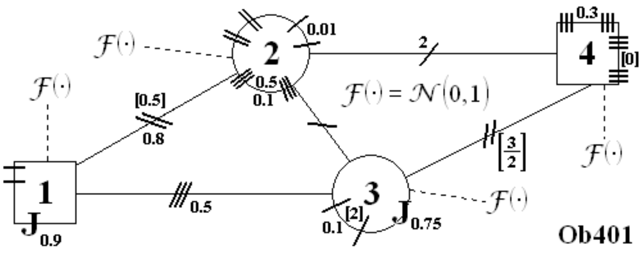


Fig. 2 Scheme of the object Ob401

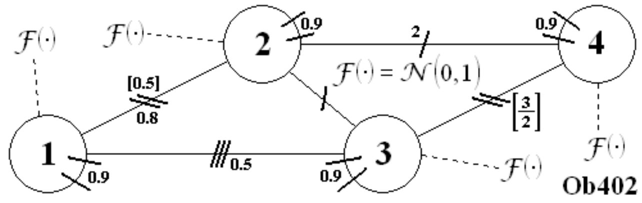

$\mathcal{F}(\cdot)$
2
0.9
2
0.9
4
[0.5]
0.8
$\mathcal{F}(\cdot) = \mathcal{N}(0,1)$
1
0.9
0.5
0.9
3
$\left[\frac{3}{2}\right]$
Ob402

Fig. 3 Scheme of the object Ob402

V. Conclusions

Based on the obtained results one can say (by comparison results of the same cases 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.

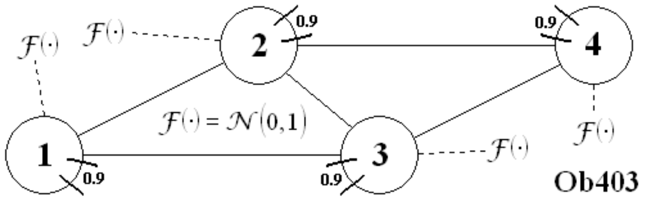


Fig. 4 Scheme of the object Ob403

References

[1] Sutharsan S., Kirubarajan T., Lang T., McDonald M., An Optimization-Based Parallel Particle Filter for Multitarget Tracking, IEEE Transactions on Aerospace and Electronic Systems, Vol. 48, No. 2, 4/2012, pp. 1601-1618. DOI: 10.1109/TAES.2012.6178081

[2] Doucet A., Freitas N., Murphy K., Russell S., Rao-Blackwellised particle filtering for dynamic Bayesian networks, Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence, 2000, pp. 176-183.

[3] Handeby G., Karlsson R., Gustafsson F., The Rao-Blackwellized Particle Filter: A Filter Bank Implementation, EURASIP Journal on Advances in Signal Processing, 2010, Article ID 724087, p. 10.

[4] Šmídl V., Hofman R., Marginalized Particle Filtering Framework for Tuning of Ensemble Filters, Monthly Weather Review, Vol. 139, No. 11, 2011, pp. 3589-3599.

[5] Djurić P. M., Lu T., Bugallo M. F., Multiple particle filtering, In 32nd IEEE ICASSP, April 2007, III pp. 1181-1184.

[6] Kozierski P., Lis M., Horla D., Level of Dispersion in Dispersed Particle Filter, in Methods and Models in Automation and Robotics (MMAR), 20th International Conference on, 2015, pp. 418-423.

[7] Kozierski P., Sadalla T., Owczarkowski A., Drgas S., Particle Filter in Multidimensional Systems, in Methods and Models in Automation and Robotics (MMAR), 21st International Conference on, 2016, pp. 806-810. DOI: 10.1109/MMAR.2016.7575240

[8] Candy J. V., Bayesian Signal Processing, WILEY, New Jersey 2009, pp. 36-44. DOI: 10.1002/9780470430583

[9] Gordon N. J., Salmond D. J., Smith A. F. M., Novel approach to nonlinear/non-Gaussian Bayesian state estimation, IEE Proceedings-F, Vol. 140, No. 2, 1993, pp. 107-113. DOI: 10.1049/ip-f-2.1993.0015

[10] Li F., Wu L., Shi P., Lim C. C., State Estimation and Sliding Mode Control for Semi-Markovian Jump Systems with Mismatched Uncertainties, Automatica, Vol. 51, 2015, pp. 385-393.

[11] Arulampalam S., Maskell S., Gordon N., Clapp T., A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking, IEEE Transactions on Signal Processing, Vol. 50, No. 2, 2002, pp. 174-188. DOI: 10.1109/78.978374

[12] Doucet A., Johansen A.M., A Tutorial on Particle Filtering and Smoothing: Fifteen years later, handbook of Nonlinear Filtering 2009/12, pp. 656-704.

[13] Imtiaz S. A., Roy K., Huang B., Shah S. L., Jampana P., Estimation of States of Nonlinear Systems using a Particle Filter, In IEEE International Conference on Industrial Technology, ICIT 2006, December, pp. 2432-2437.

[14] Kozierski P., Lis M., Ziętkiewicz J., Resampling in Particle Filtering – Comparison, Studia z Automatyki i Informatyki, Vol. 38, 2013, pp. 35-64.

[15] Doucet A., Freitas N., Gordon N., Sequential Monte Carlo Methods in Practice, Springer-Verlag, New York 2001, pp. 225-246. DOI: 10.1007/978-1-4757-3437-9

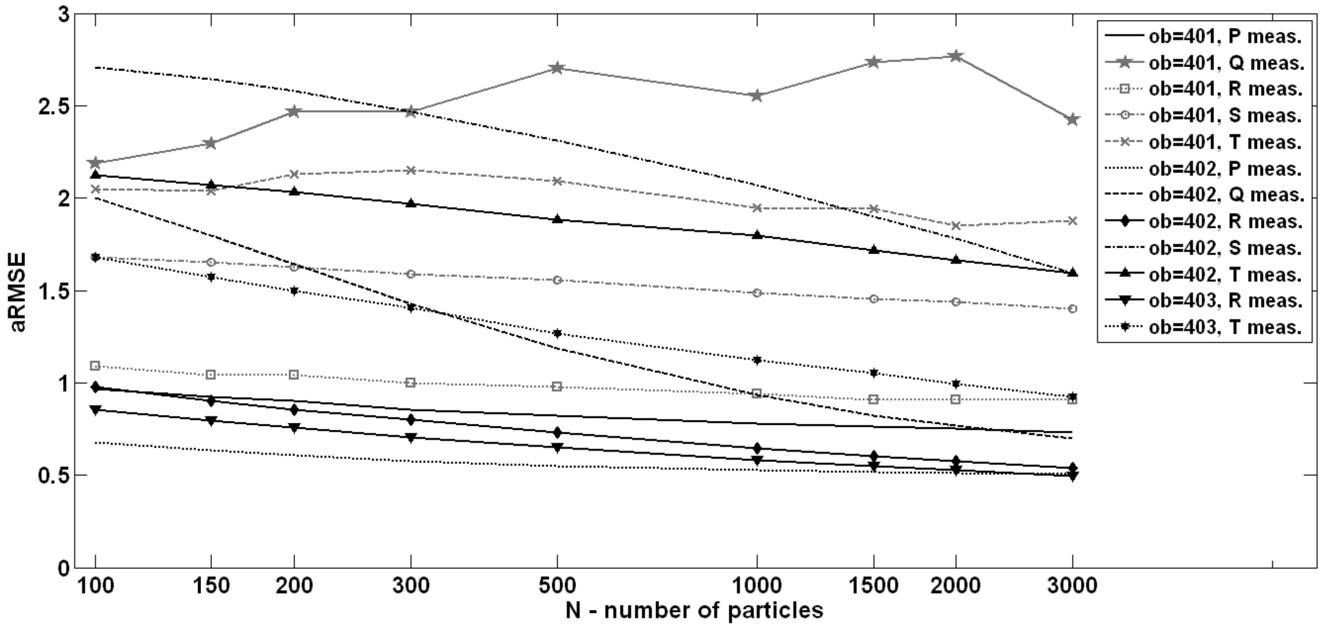


Fig. 5 Obtained results