Rao-Blackwellized Auxiliary Particle Filters for Mixed Linear/Nonlinear Gaussian models

Nordh, Jerker

Rao-Blackwellized Auxiliary Particle Filters for Mixed Linear/Nonlinear Gaussian models

Jerker Nordh

Department of Automatic Control

Lund University, Sweden

Email: jerker.nordh@control.lth.se

Abstract—The Auxiliary Particle Filter is a variant of the common particle filter which attempts to incorporate information from the next measurement to improve the proposal distribution in the update step. This paper studies how this can be done for Mixed Linear/Nonlinear Gaussian models, it builds on a previously suggested method and introduces two new variants which tries to improve the performance by using a more detailed approximation of the true probability density function when evaluating the so called first stage weights. These algorithms are compared for a couple of models to illustrate their strengths and weaknesses.

I. Introduction

The particle filter (PF) [1][2][3] has become one of the standard algorithms for nonlinear estimation and has proved its usefulness in a wide variety of applications. It is an application of the general concept of Sequential Importance Resampling (SIR)[4]. At each time the true probability density function is approximated by a number of point estimates, the so called particles. These are propagated forward in time using the system dynamics. For each measurement the weight of each particle is updated with the corresponding likelihood of the measurement. The last step is to resample the particles, this occurs either deterministically or only when some criteria is fulfilled. This criteria is typically the so called number of effective particles, which is a measure of how evenly the weights are distributed among the particles. The resampling is a process that creates a new set particles where all particles have equal weights, which is accomplished by drawing them with probability corresponding to their weight in the original set. This process is needed since otherwise eventually all the weights except one would go to zero. The resampling thus improves the estimate by focusing the particles to regions with higher probability.

The Auxiliary Particle Filter (APF)[5][6][7] attempts to improve this by resampling the particles at time t using the predicted likelihood of the measurement at time $t + 1$. If done properly this helps to focus the particles to areas where the measurement has a high likelihood. The problem is that it requires the evaluation of the probability density function $p(y_{t+1}|x_t)$. This is typically not available and is therefore often approximated by assuming that next state is the predicted mean state, i.e. $x_{t+1} = \bar{x}_{t+1|t}$ and the needed likelihood instead becomes $p(y_{t+1}|x_{t+1} = \bar{x}_{t+1|t})$.

This paper focuses on the special case of Mixed Linear/Nonlinear Gaussian (MLNLG) models, which is a special case of Rao-Blackwellied models. For an introduction to the Rao-Blackwellized Particle Filter see [8]. Rao-Blackwellized

models have the property that conditioned on the nonlinear states there exists a linear Gaussian substructure that can be optimally estimated using a Kalman filter. This reduces the dimensionality of the model that the particle filter should solve, thereby reducing the complexity of the estimation problem. The general form for MLNLG models is shown in (1), where the state vector has been split in two parts, $x = (\xi \quad z)^T$. Here ξ are the nonlinear states that are estimated by the particle filter, z are the states that conditioned on the trajectory $\xi_{1:t}$ are estimated using a Kalman filter.

$$
\begin{align}
\xi_{t+1} &= f_\xi(\xi_t) + A_\xi(\xi_t)z_t + v_\xi \tag{1a} \\
z_{t+1} &= f_z(\xi_t) + A_z(\xi_t)z_t + v_z \tag{1b} \\
y_{t+1} &= g(\xi_{t+1}) + C(\xi_{t+1})z_{t+1} + e \tag{1c} \\
\begin{pmatrix} v_\xi \\ v_z \end{pmatrix} &\sim N\left(0, \begin{pmatrix} Q_\xi(\xi_t) & Q_{\xi z}(\xi_t) \\ Q_{z\xi}(\xi_t) & Q_z(\xi_t) \end{pmatrix}\right) \tag{1d} \\
e &\sim N(0, R(\xi_t)) \tag{1e}
\end{align}
$$

In [9] an approximation is presented that can be used with the APF for this type of models, section II presents that algorithm and two variants proposed by the author of this paper. Section III compares the different algorithms by applying them to a number of examples to highlight their strengths and weaknesses. Finally section IV concludes the paper with some discussion of the trade-offs when choosing one of theses algorithms.

II. Algorithms

A. Auxiliary Particle Filter introduction

$$
\begin{align}
x_{t+1} &= f(x_t, v_t) \tag{2a} \\
y_t &= h(x_t, e_t) \tag{2b}
\end{align}
$$

Looking at a generic state-space model of the form in (2) and assuming we have a collection of weighted point estimates (particles) approximating the probability density function of $x_{t|t} \approx \sum_{i=1}^{N} w^{(i)}\delta(x_t - x_t^{(i)})$ the standard particle filter can be summarized in the following steps that are done for each time instant t.

1) (Resample; draw N new samples $\tilde{x}_t^{(i)}$ from the categorical distribution over $x_t^{(i)}$ with probabilities proportional to $w_t^{(i)}$. Set $\tilde{w}_t^{(i)} = \frac{1}{N}$.)

2) For all i; sample $v_t^{(i)}$ from the noise distribution
3) For all i; calculate $x_{t+1|t}^{(i)} = f(\tilde{x}_t^{(i)}, v_t^{(i)})$
4) For all i; calculate $w_{t+1}^{(i)} = \tilde{w}_t^{(i)} p(y_t|x_{t+1|t}^{(i)})$

The resampling step introduces variance in the estimate, but is necessary for the long term convergence of the filter. Step 1 is therefore typically not done at each time instant but only when a prespecified criteria on the weights are fulfilled.

The Auxiliary Particle Filter introduces an additional step to incorporate knowledge of the future measurement y_{t+1} before updating the point estimates $x_t^{(i)}$.

1) For all i; calculate $\tilde{w}_t^{(i)} = l^{(i)} w_t^{(i)}$, where $l^{(i)} = p(y_{t+1}|x_t^{(i)})$
2) (Resample; draw N new samples $\tilde{x}_t^{(i)}$ from the categorical distribution over $x_t^{(i)}$ with probabilities proportional to $l^{(i)} w_t^{(i)}$. Set $\tilde{w}_t = \frac{1}{N}$.)
3) For all i; sample from $v_t^{(i)}$ from the noise distribution
4) For all i; calculate $x_{t+1|t}^{(i)} = f(\tilde{x}_t^{(i)}, v_t^{(i)})$
5) For all i; calculate $w_{t+1}^{(i)} = \frac{\tilde{w}_t^{(i)}}{l^{(i)}} p(y_t|x_{t+1|t}^{(i)})$

The difference between the algorithms in the rest of section II is in how the first stage weights, $l^{(i)}$, are approximated for the specific class of model (1).

B. Algorithm 1

The first algorithm considered is the one presented in [9]. It computes $p(y_{t+1}|x_t)$ using the approximate model shown in (3). This approximation ignores the uncertainty in the measurement that is introduced through the uncertainty in ξ_{t+1}, thereby underestimating the total uncertainty in the measurement. Tbjs could lead to resampling of the particles even when the true particle weights would not indicate the need for resampling. Here P_{z_t} denotes the estimated covariance for the variable z_t. In (3e) the dependence on ξ_t has been suppressed to not clutter the notation.

$$
\begin{align}
p(y_{t+1}|x_t) &\sim N(\bar{y}_{t+1}, P_{y_{t+1}}) \tag{3a} \\
\bar{\xi}_{t+1} &= f_\xi(\xi_t) + A_\xi(\xi_t)\bar{z}_t \tag{3b} \\
\bar{z}_{t+1} &= f_z(\xi_t) + A_z(\xi_t)\bar{z}_t \tag{3c} \\
\bar{y}_{t+1} &= g(\bar{\xi}_{t+1}) + C(\bar{\xi}_{t+1})\bar{z}_{t+1} \tag{3d} \\
P_{y_{t+1}} &\approx C(A_z P_{z_t} A_z^T + Q)C^T + R \tag{3e}
\end{align}
$$

C. Algorithm 2

Our first proposed improvement to the algorithm in section II-B is to attempt to incorporate the uncertainty of ξ_{t+1} in our measurement by linearizing the measurement equation around the predicted mean. For the case when C and R has no dependence on ξ the pdf of the measurement can then be approximated in the same way, except for $P_{y_{t+1}}$ which instead is approximated according to (4), here J_g is Jacobian of g. The difference compared to model 1 lies in the additional terms for the covariance $P_{y_{t+1}}$ due propagating the uncertainty of ξ_{t+1} by using the Jacobian of g.

$$
\begin{align}
P_{y_{t+1}} &\approx (J_g A_\xi + C A_z) P_{z_t} (J_g A_\xi + C A_z)^T \\
&+ (J_g + C)Q(J_q + C)^T + R \tag{4}
\end{align}
$$

D. Algorithm 3

If the linearization scheme presented in Section II-C is to work well g must be close to linear in the region where ξ_{t+1} has high probability. Another commonly used approximation when working with Gaussian distributions and nonlinearities is the Unscented Transform used in e.g. the Unscented Kalman Filter[10]. The second proposed algorithms uses the UT to approximate the likelihood of the measurement. It propagates a set of so called Sigma-point through both the time update (2a) and the measurement equation (2b) and estimates the resulting mean and covariance of the points after the transformations. This is then used as the approximation for $\bar{y}_{t+1}$ and $P_{y_{t+1}}$. This avoids the need to linearize the model, and can thus handle discontinuities and other situations where linearization cannot be used or gives a poor approximation.

The Sigma-points, $x_\sigma^{(i)}$, were chosen according to (5), using twice the number of points as the combined dimension of the state and noise vectors (N_x). Σ_x is the combined covariance matrix. $(\sqrt{N_x \Sigma_x})_i$ is the i-th column of R, where $N_x \Sigma_x = RR^T$. All the points were equally weighted. This conforms to the method introduced in section III-A in [10].

$$
\begin{align}
\Sigma_x &= \text{diag}(P_{z_t}, Q, R) \tag{5a} \\
W^{(i)} &= (2N_x)^{-1}, \quad i \in 1..2N_x \tag{5b} \\
x_\sigma^{(i)} &= \bar{x} + (\sqrt{N_x \Sigma_x})_i, \quad i \in 1..N_x \tag{5c} \\
x_\sigma^{(i+N_x)} &= \bar{x} - (\sqrt{N_x \Sigma_x})_i, \quad i \in 1..N_x \tag{5d}
\end{align}
$$

III. RESULTS

To study the difference in performance of the three algorithms we will look at two models, the first one was introcuded in [11] and is an extension of a popular nonlinear model to include a linear Gaussian substructure. The second model was designed to highlight problems with the approximation used in Algorithm 1.

A. Model 1

$$
\begin{align}
\xi_{t+1} &= \frac{\xi_t}{1+\xi_t^2} \begin{pmatrix} 0 & 0.04 & 0.044 & 0.008 \end{pmatrix} z_t + \\
&+ 0.5\xi_t + 25\frac{\xi_t}{1+\xi_t^2} + 8\cos 1.2t + v_{\xi,t} \tag{6a} \\
z_{t+1} &= \begin{pmatrix} 3 & -1.691 & 0.849 & -0.3201 \\ 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \end{pmatrix} z_t + \\
&+ v_{z,t} \tag{6b} \\
y_t &= 0.05\xi_t^2 + e_t \tag{6c} \\
\xi_0 &= 0, \quad z_0 = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}^T \tag{6d} \\
R &= 0.1, \quad Q_\xi = 0.005 \tag{6e} \\
Q_z &= \text{diag}(0.01 \quad 0.01 \quad 0.01 \quad 0.01) \tag{6f}
\end{align}
$$

Notice the square in the measurement equation in (6c), depending on the magnitude of uncertainty in ξ_t a linearizion of this term might lead to a poor approximation. So we might expect that for this model the unscented transform based approximation might fare better.

The performance of the algorithms was tested by generating 25000 dataset from (6), all the algorithms were then tested on this collection for a number of different particle counts. The average RMSE values are presented in table I, they are also shown in Fig. 1. The relative RMSE of the algorithms compared to the standard particle filter is shown in Table II. As can be seen both algorithm 2 and 3 outperform Algorithm 1, most of the time Algorithm 3 also beats Algorithm 2. This is expected since they use a more detailed approximation of the true density of $p(y_{t+1}|x_t)$.

N	PF	Alg. 1	Alg. 2	Alg. 3
10	1.701	1.611	1.607	1.600
15	1.395	1.338	1.322	1.322
20	1.234	1.170	1.153	1.154
25	1.121	1.061	1.060	1.052
30	1.049	0.992	0.978	0.978
40	0.955	0.893	0.885	0.880
50	0.874	0.831	0.824	0.819
75	0.782	0.737	0.732	0.730
100	0.720	0.689	0.686	0.687

TABLE I.   AVERAGE RMSE FOR ξ OVER 25000 REALIZATIONS OF MODEL 1. AS CAN BE SEEN THE IMPROVED APPROXIMATIONS IN ALGORITHM 2 AND 3 LEAD TO SLIGHTLY BETTER PERFORMANCE, BUT FOR THIS MODEL THE DIFFERENCE IN PERFORMANCE OF THE ALGORITHMS IS SMALL

N	Alg. 1	Alg. 2	Alg. 3
10	94.7%	94.5%	94.1%
15	96.0%	94.8%	94.7%
20	94.8%	93.4%	93.5%
25	94.6%	94.5%	93.8%
30	94.5%	93.2%	93.2%
40	93.5%	92.7%	92.1%
50	95.0%	94.2%	93.7%
75	94.2%	93.5%	93.3%
100	95.7%	95.2%	95.4%

TABLE II.   RMSE COMPARED TO THE RMSE FOR THE REGULAR PARTICLE FILTER, FOR ξ OVER 25000 REALIZATIONS OF MODEL 1. AS CAN BE SEEN THE IMPROVED APPROXIMATIONS IN ALGORITHM 2 AND 3 LEAD TO SLIGHTLY BETTER PERFORMANCE, BUT FOR THIS MODEL THE DIFFERENCE IN PERFORMANCE OF THE ALGORITHMS IS SMALL

B. Model 2

$$\xi_{t+1} = 0.8\xi_t + a_t z_t + v_\xi \tag{7a}$$
$$z_{t+1} = 0.8z_t + v_z \tag{7b}$$
$$y_t = \xi_t^2 + a_t z_t + e_t \tag{7c}$$
$$v_\xi \sim N(0, 0.5a_t + 0.1(1 - a_t)) \tag{7d}$$
$$v_z \sim N(0, 0.1) \tag{7e}$$
$$R \sim N(0, 0.1) \tag{7f}$$
$$a_t = \begin{cases} 0, & t \text{ even} \\ 1, & \text{otherwise} \end{cases}$$

The model in (7) was constructed to exploit the weakness of neglecting the uncertainty introduced in the measurement
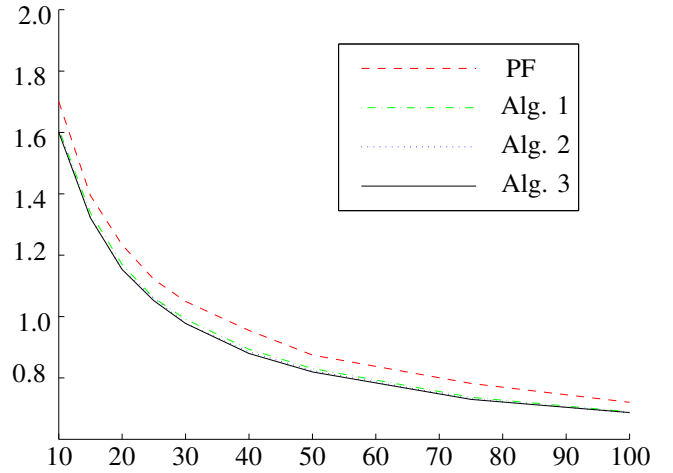

Fig. 1.   Average RMSE as a function of the number of particles when estimating model 1 with the three algorithms in this paper. The particle filter is included as a reference. Average over 25000 realizations. Data-points for 10, 15, 20, 25, 30, 40, 50, 75 and 100 particles

due to the uncertainty of ξ_{t+1}. In order for this to affect the estimate it is also important that the trajectory leading up to the current estimate is of importance. Since z_t depends on the full trajectory $\xi_{1:t}$ it captures this aspect. The estimate of z for different particles will also influence the future trajectory, this implies that even if two particles at time t are close to the same value of ξ the path they followed earlier will affect their future trajectories. This means that a premature resampling of the particles due to underestimating the uncertainty in the measurement can lead to errors in the following parts of the dataset.

To clearly demonstrate this issue a time-varying model was chosen were the z-state only affects every second measurement and every second state update. It turns out that this model is difficult to estimate, and when using few particles the estimates sometimes diverge with the esimated states approaching infinity. To not overshadow the RMSE of all the realization were the filters perform satisfactorily they are excluded, the number of excluded realizations are presented in Table III.

The model was also modified by moving the pole corresponding to the ξ-state to 0.85 and the one for the z-state to 0.9, this makes the estimation problem more difficult. The corresponding number of diverged estimates are shown in Table IV. It can seen that even though all three algorithms are more robust than the standard particle filter, Algorithm 1 clearly outperforms the rest when it comes to robustness.

The RMSE of the algorithms are shown in Fig. 2 for model (7) and Fig. 3 for the case with the poles moved to 0.85 and 0.9. Surprisingly the RMSE of Algorithm 1 increases as the number of particles increases. The author believes this is an effect of the particles being resampled using a too coarse approximation, thus introducing a modeling error. When the number of particles is increased the estimate of a particle filter normally converges to the true poster probability density, but in this case the estimate is converging to the true posterior for the incorrect model, leading to an incorrect estimate.
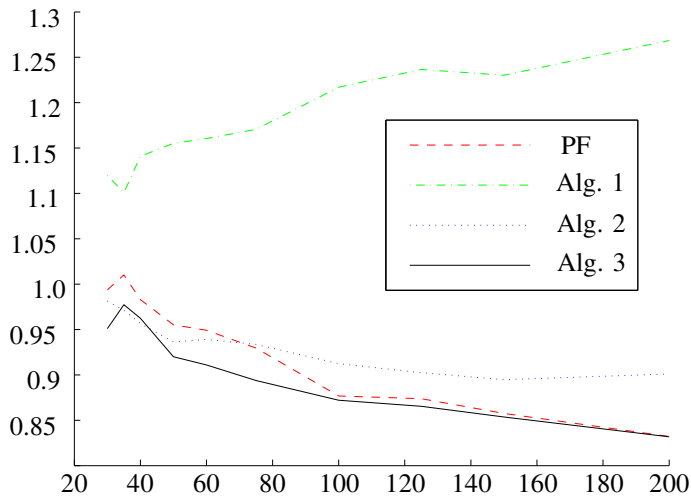
Fig. 2. Average RMSE as a function of the number of particles when estimating model 2 with the three algorithms in this paper. Data-points for 30, 35, 40, 50, 60, 75, 100, 125, 150 and 200 particles. The particle filter is included as a reference. Average over 1000 realizations, excluding those that diverged (RMSE $\geq$ 10000). Both algorithm 2 and 3 beat the PF for low particle counts, but only the algorithm 3 keeps up when the number of particles increases. Surprisingly algorithm 1's performance degrades as the number of particles increases.
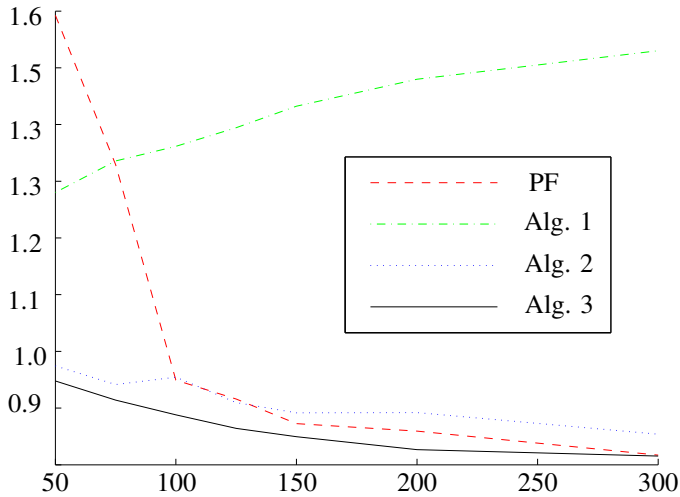

Fig. 3. Average RMSE for the three algorithms as a function of the number of particles when estimating the modified model 2 (poles in 0.85 and 0.9 instead). Data-points for 50, 75, 100, 125, 150, 200 and 300 particles. The particle filter is included as a reference. Average over 5000 realizations, except for the PF where 25000 realizations were used for 50, 75, 75, 100 and 125 particles due to the large variance of those estimates. The RMSE was calculated excluding those realizations that diverged (RMSE $\geq$ 10000). All filters beat the PF for low particle counts, but only algorithm 3 keeps up when the number of particles increases. Surprisingly the performance of algorithm 1 degrades as the number of particles increases.

N	PF	Alg. 1	Alg. 2	Alg. 3
25	3	0	0	2
50	2	0	0	0
75	1	0	0	0
100	0	0	0	0
125	0	0	0	0
150	0	0	0	0
200	0	0	0	0

TABLE III. THE NUMBER OF DIVERGED ESTIMATES FOR 1000 REALIZATIONS OF MODEL 2. A DIVERGED REALIZATION IS DEFINED AS ONE WHERE THE RMSE EXCEEDED 10000.

N	PF	Alg. 1	Alg. 2	Alg. 3
50	124	1	40	35
75	62	0	12	13
100	45	0	7	8
125	22	0	5	4
150	17	0	4	2
200	9	0	1	2
300	1	0	0	0

TABLE IV. THE NUMBER OF DIVERGED ESTIMATES FOR 5000 REALIZATIONS OF MODEL 2 WITH THE POLE FOR ξ MOVED TO 0.85 AND THE POLE FOR z IN 0.9 INSTEAD. A DIVERGED REALIZATION IS DEFINED AS ONE WHERE THE RMSE EXCEEDED 10000.

IV. Conclusion

The three algorithms compared in this paper are all variants of the general Auxiliary Particle Filter algorithm, but by choosing different approximations when evaluating the first stage weights (l) they have different performance characteristics.

As was especially evident when looking at model 2 all three auxiliary particle filters were more robust than the ordinary particle filter, particularly when using fewer particles. Algorithm 1 is especially noteworthy since it almost did not suffer at all from the problem with divergence that was noted for the other approximations when estimating model 2. However it does seem to have problems approximating the true posterior, as evident by the high RMSE that also was unexpectedly increasing when the number of particles were increased.

Algorithm 2 and 3 both beat the standard particle filter both in terms of average RMSE and robustness. Algorithm 3 has the best performance, but also requires the most computations due to the use of an Unscented Transform approximation for each particle and measurement.

In the end the choice of which algorithm to use has to be decided on a case by case basis depending on relative computational complexity for the different approximations for that particular model, so this article can not present any definitive advice for this choice. The linearization approach doesn't increase the computational effort nearly as much as using the Unscented Transform, but it doesn't always capture the true distribution with the needed accuracy. However, for models where the linearization works well this is likely the preferred approximation due to the low increase in computational effort needed.

When it is possible to increase the number of particles it could very well be most beneficial to simply use the standard particle filter with a higher number of particles, thus completely avoiding the need to evaluate $p(y_{t+1}|x_t)$. As always this is influenced by the specific model and how the uncertainty in the update step compares to the uncertainty in the measurement.

References

[1] N. Gordon, D. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *Radar and Signal Processing, IEE Proceedings F*, vol. 140, no. 2, pp. 107–113, 1993.

[2] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," *Handbook of Nonlinear Filtering*, vol. 12, pp. 656–704, 2009.

[3] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, feb 2002.

[4] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000.

[5] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American statistical association*, vol. 94, no. 446, pp. 590–599, 1999.

[6] ——, "Auxiliary variable based particle filters," in *Sequential Monte Carlo methods in practice*. Springer, 2001, pp. 273–293.

[7] A. M. Johansen and A. Doucet, "A note on auxiliary particle filters," *Statistics & Probability Letters*, vol. 78, no. 12, pp. 1498–1504, 2008.

[8] T. Schön, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *Signal Processing, IEEE Transactions on*, vol. 53, no. 7, pp. 2279–2289, 2005.

[9] C. Fritsche, T. Schon, and A. Klein, "The marginalized auxiliary particle filter," in *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2009 3rd IEEE International Workshop on*. IEEE, 2009, pp. 289–292.

[10] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[11] F. Lindsten and T. Schön, "Rao-blackwellized particle smoothers for mixed linear/nonlinear state-space models," Tech. Rep., 2011. [Online]. Available: http://user.it.uu.se/ thosc112/pubpdf/lindstens2011.pdf

[12] J. Nordh, "pyParticleEst." [Online]. Available: http://www.control.lth.se/Staff/JerkerNordh/pyparticleest.html