Marginalized Particle Filters for Mixed Linear/Nonlinear State-space Models

Thomas Schön, Fredrik Gustafsson, *Member, IEEE,* and Per-Johan Nordlund

***Abstract*— The particle filter offers a general numerical tool to approximate the posterior density function for the state in nonlinear and non-Gaussian filtering problems. While the particle filter is fairly easy to implement and tune, its main drawback is that it is quite computer intensive. However, due to faster computers this drawback can be overcome and as a result the particle filter has quickly become a popular tool in signal processing applications. The computational complexity increases quickly with the state dimension for the problem at hand. One remedy to this problem is a technique known as Rao-Blackwellization, where states appearing linearly in the dynamics are marginalized out. The result of this is that one Kalman filter is associated with each particle. Our main contribution in this article is to derive the details for the marginalized particle filter for a general nonlinear state-space model. We will also discuss some important special cases occurring in typical signal processing applications. The marginalized particle filter is applied to an integrated navigation system for aircraft. It is demonstrated that the complete high-dimensional system can be based on the particle filter using marginalization for all but three states. Excellent performance on real flight data is reported.**

***Index Terms*— State estimation, Particle filter, Kalman filter, Marginalization, Navigation systems, Nonlinear systems.**

I. Introduction

THE nonlinear non-Gaussian filtering problem we consider consists of recursively computing the posterior density of the state vector in a general discrete-time state-space model, given the observed measurements. A general formulation of such a model is provided in Model 1 below.

Model 1:

$$x_{t+1} = f(x_t, w_t), \tag{1a}$$
$$y_t = h(x_t, e_t), \tag{1b}$$

Here y_t is the measurement at time t, x_t is the state, w_t is the process noise, e_t is the measurement noise, and f, h are two arbitrary nonlinear functions. The two noise densities p_{w_t} and p_{e_t} are independent and are assumed to be known. □

The posterior density $p(x_t|Y_t)$, where $Y_t = \{y_i\}_{i=0}^{t}$, is given by the following general measurement recursion

$$p(x_t|Y_t) = \frac{p(y_t|x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})}, \tag{2a}$$
$$p(y_t|Y_{t-1}) = \int p(y_t|x_t)p(x_t|Y_{t-1})dx_t, \tag{2b}$$

and the following time recursion

$$p(x_{t+1}|Y_t) = \int p(x_{t+1}|x_t)p(x_t|Y_t)dx_t, \tag{2c}$$

initiated by $p(x_0|Y_{-1}) = p(x_0)$ [20]. For linear Gaussian models, the integrals can be solved analytically with a finite dimensional representation. This leads to the Kalman filter recursions, where the mean and the covariance matrix of the state are propagated [1]. Generally, no finite dimensional representation of the posterior density exists. Thus, several numerical approximations of the integrals (2) have been proposed. A recent important contribution is to use simulation based methods from mathematical statistics, sequential Monte Carlo methods, commonly referred to as particle filters [12], [11], [16].

An inherent problem with the particle filter is its high computational cost. Asymptotically as the number of particles tends to infinity we know that we get the optimal filter given by (2). However, in practice there is a trade-off between accuracy and the computational complexity. Another practically limiting factor is that the number of particles needed to get a decent approximation increases with the state dimension. This implies that for a large enough state dimension the particles filter can be ruled out. However, if there is a linear sub-structure in Model 1 this can be utilized in order to obtain better estimates and possibly reduce the computational demand. The basic idea is to partition the state vector as

$$x_t = \begin{bmatrix} x_t^l \\ x_t^n \end{bmatrix}, \tag{3}$$

where x_t^l denotes the state variable with conditionally linear dynamics and x_t^n denotes the nonlinear state variable [14], [31]. Using Bayes' theorem we can then marginalize out the linear state variables from (1) and estimate them using the Kalman filter [22], which is the optimal filter for this case. The nonlinear state variables are estimated using the particle filter. This technique is sometimes referred to as Rao-Blackwellization [14]. The idea is certainly not new, it has been around for quite some time, see e.g., [12], [7], [8], [2], [14], [30]. The contribution of this article is that we derive the details for a general nonlinear state-space model with a linear sub-structure. Models of this type are common and important

This work was supported by the competence center ISIS at Linköping University and the Swedish Research Council (VR).

T. Schön is with the Department of Electrical Engineering, Linköping University, Linköping, Sweden (e-mail: schon@isy.liu.se).
F. Gustafsson is with the Department of Electrical Engineering, Linköping University, Linköping, Sweden (e-mail: fredrik@isy.liu.se).
P-J. Nordlund is with the Department for Flight Data and Navigation, Saab Aerospace, Linköping, Sweden (e-mail: Per-Johan.Nordlund@saab.se).

in engineering applications, e.g., positioning, target tracking and collision avoidance [18], [4]. The marginalized particle filter has been successfully used in several applications, for instance in aircraft navigation [31], underwater navigation [24], communications [9], [35], nonlinear system identification [27], [33], and audio source separation [3].

Integrated navigation is used as a motivation and application example. In this military application, the primary position information is provided by a terrain positioning algorithm, where the particle filter has previously [5] proven to be powerful. The performance is close to the Cramér-Rao lower bound. However, it is poorly matched to the integration filter which among other things estimates offsets and drifts from the inertial sensors. The integration system is currently based on a Kalman filter and it requires Gaussian measurements, while the particle filter delivers a multi-modal measurement with possible positions. One idea would be to also base the integration system on the particle filter. However, the model is high-dimensional (27 states), and the particle filter cannot be applied without marginalization. Only three states appear nonlinearly in the model, so most of the states can be integrated out. We motivate that the marginalized particle filter can be implemented in real-time, and demonstrate its good performance for a scalable nine-dimensional sub-model of the integration system.

Section II explains the idea behind using marginalization in conjunction with general nonlinear state-space models. Three nested models are used, in order to make the presentation easy to follow and to facilitate the understanding. We comment on some important special cases and discuss some generalizations of the noise assumptions in Section III. Finally, the application example is given in Section IV and the conclusions are stated in Section V.

II. Variance Reduction by Marginalization

The variance of the estimates obtained from the standard particle filter can be decreased by exploiting linear substructures in the model. The corresponding variables are marginalized out and estimated using an optimal linear filter. This is the main idea behind the marginalized particle filter. The goal of this section is to explain how the marginalized particle filter works by using three nested models. The models are nested in the sense that the first model is included in the second, which in turn is included in the third. The reason for presenting it in this fashion is to facilitate the understanding, by incrementally extending the standard particle filter.

A. The Standard Particle Filter

The particle filter is used to get an approximation of the posterior density $p(x_t|Y_t)$ in Model 1. This approximation can then be used to obtain an estimate of some inference function, $g(\cdot)$, according to

$$I(g(x_t)) = E_{p(x_t|Y_t)}[g(x_t)] = \int g(x_t)p(x_t|Y_t)dx_t \tag{4}$$

In the following the particle filter, as it was introduced in [16], will be referred to as the standard particle filter. For a thorough introduction to the standard particle filter we refer to [11], [12]. The marginalized and the standard particle filter are closely related. The marginalized particle filter is given in Algorithm 1 and neglecting steps 4a and 4c results in the standard particle filter algorithm.

Algorithm 1: The marginalized particle filter

1) Initialization: For $i = 1, \dots, N$, initialize the particles, $x_{0|-1}^{n,(i)} \sim p_{x_0^n}(x_0^n)$ and set $\{x_{0|-1}^{l,(i)}, P_{0|-1}^{(i)}\} = \{\bar{x}_0^l, \bar{P}_0\}$.
2) For $i = 1, \dots, N$, evaluate the importance weights $q_t^{(i)} = p(y_t|X_t^{n,(i)}, Y_{t-1})$ and normalize
$$\tilde{q}_t^{(i)} = \frac{q_t^{(i)}}{\sum_{j=1}^N q_t^{(j)}}.$$
3) Particle filter measurement update: Resample N particles with replacement according to,
$$\text{Pr}(x_{t|t}^{n,(i)} = x_{t|t-1}^{n,(j)}) = \tilde{q}_t^{(j)}.$$
4) Particle filter time update and Kalman filter:
   a) Kalman filter measurement update:
      Model 2: (10),
      Model 3: (10),
      Model 4: (22).
   b) Particle filter time update: For $i = 1, \dots, N$, predict new particles according to
      $$x_{t+1|t}^{n,(i)} \sim p(x_{t+1|t}^n|X_t^{n,(i)}, Y_t).$$
   c) Kalman filter time update:
      Model 2: (11),
      Model 3: (17),
      Model 4: (23).
5) Set $t := t + 1$ and iterate from step 2.

B. Diagonal Model

Let us start our explanation of how the marginalized particle filter works by considering the following model,

Model 2:

$$x_{t+1}^n = f_t^n(x_t^n) \qquad\qquad + w_t^n, \tag{5a}$$
$$x_{t+1}^l = \qquad\qquad A_t^l(x_t^n)x_t^l + w_t^l, \tag{5b}$$
$$y_t = h_t(x_t^n) + C_t(x_t^n)x_t^l + e_t, \tag{5c}$$

The gaps in the equations above are placed there intentionally, in order to make the comparison to the general model (18) easier. We assume that the state noise is white and Gaussian distributed according to

$$w_t = \begin{bmatrix} w_t^l \\ w_t^n \end{bmatrix} \sim \mathcal{N}(0, Q_t), \quad Q_t = \begin{bmatrix} Q_t^l & 0 \\ 0 & Q_t^n \end{bmatrix}, \tag{6a}$$

and that the measurement noise is white and Gaussian distributed according to

$$e_t \sim \mathcal{N}(0, R_t). \tag{6b}$$

Furthermore x_0^l is Gaussian,

$$x_0^l \sim \mathcal{N}(\bar{x}_0, \bar{P}_0). \tag{6c}$$

The density of x_0^n can be arbitrary, but it is assumed known. The A^l and C matrices are arbitrary. □

Model 2 is called "diagonal model" due to the diagonal structure of the state equation (5a) – (5b). The aim of recursively estimating the posterior density $p(x_t|Y_t)$ can be accomplished using the standard particle filter. However, conditioned on the nonlinear state variable, x_t^n, there is a linear sub-structure in (5), given by (5b). This fact can be used to obtain better estimates of the linear states. Analytically marginalizing out the linear state variables from $p(x_t|Y_t)$ and using Bayes' theorem we obtain ($X_t^n = \{x_i^n\}_{i=0}^t$)

$$p(x_t^l, X_t^n|Y_t) = \underbrace{p(x_t^l|X_t^n, Y_t)}_{\text{Optimal KF}} \underbrace{p(X_t^n|Y_t)}_{\text{PF}}, \tag{7}$$

where $p(x_t^l|X_t^n, Y_t)$ is analytically tractable. It is given by the Kalman filter (KF), see Lemma 2.1 below for the details. Furthermore, $p(X_t^n|Y_t)$ can be estimated using the particle filter (PF). If the same number of particles are used in the standard particle filter and the marginalized particle filter the latter will, intuitively, provide better estimates. The reason for this is that the dimension of $p(x_t^n|Y_t)$ is smaller than the dimension of $p(x_t^l, x_t^n|Y_t)$, implying that the particles live in a smaller space. Let $\hat{I}_N^s(g(x_t))$ denote the estimate of (4) using the standard particle filter with N particles. When the marginalized particle filter is used the corresponding estimate is denoted by $\hat{I}_N^m(g(x_t))$. Under certain assumptions the following central limit theorem holds,

$$\sqrt{N}(\hat{I}_N^s(g(x_t)) - I(g(x_t))) \Longrightarrow \mathcal{N}(0, \sigma_s^2), \quad N \to \infty \tag{8a}$$

$$\sqrt{N}(\hat{I}_N^m(g(x_t)) - I(g(x_t))) \Longrightarrow \mathcal{N}(0, \sigma_m^2), \quad N \to \infty \tag{8b}$$

where $\sigma_s^2 \geq \sigma_m^2$. A formal proof of (8) is provided in [14], [13]. For the sake of notational brevity we suppress the dependence of x_t^n in A_t, C_t, and h_t below.

Lemma 2.1: Given Model 2, the conditional probability density functions for $x_{t|t}^l$ and $x_{t+1|t}^l$ are given by

$$p(x_t^l|X_t^n, Y_t) = \mathcal{N}(\hat{x}_{t|t}^l, P_{t|t}), \tag{9a}$$

$$p(x_{t+1}^l|X_{t+1}^n, Y_t) = \mathcal{N}(\hat{x}_{t+1|t}^l, P_{t+1|t}), \tag{9b}$$

where

$$\hat{x}_{t|t}^l = \hat{x}_{t|t-1}^l + K_t(y_t - h_t - C_t\hat{x}_{t|t-1}^l), \tag{10a}$$

$$P_{t|t} = P_{t|t-1} - K_t C_t P_{t|t-1}, \tag{10b}$$

$$S_t = C_t P_{t|t-1} C_t^T + R_t, \tag{10c}$$

$$K_t = P_{t|t-1} C_t^T S_t^{-1}, \tag{10d}$$

and

$$\hat{x}_{t+1|t}^l = A_t^l \hat{x}_{t|t}^l, \tag{11a}$$

$$P_{t+1|t} = A_t^l P_{t|t} (A_t^l)^T + Q_t^l. \tag{11b}$$

The recursions are initiated with $\hat{x}_{0|-1}^l = \bar{x}_0$ and $P_{0|-1} = \bar{P}_0$.

Proof: Straightforward application of the Kalman filter [22], [21]. ■

The second density, $p(X_t^n|Y_t)$, in (7) will be approximated using the standard particle filter. We can use Bayes' theorem and the Markov property inherent in the state-space model to write $p(X_t^n|Y_t)$ as

$$p(X_t^n|Y_t) = \frac{p(y_t|X_t^n, Y_{t-1})p(x_t^n|X_{t-1}^n, Y_{t-1})}{p(y_t|Y_{t-1})} p(X_{t-1}^n|Y_{t-1}), \tag{12}$$

where an approximation of $p(X_{t-1}^n|Y_{t-1})$ is provided by the previous iteration of the particle filter. In order to perform the update (12) we need analytical expressions for $p(y_t|X_t^n, Y_{t-1})$ and $p(x_t^n|X_{t-1}^n, Y_{t-1})$. They are provided by the following lemma.

Lemma 2.2: For Model 2 we have that

$$p(y_t|X_t^n, Y_{t-1}) = \mathcal{N}(h_t + C_t\hat{x}_{t|t-1}^l,\, C_t P_{t|t-1} C_t^T + R_t), \tag{13a}$$

$$p(x_{t+1}^n|X_t^n, Y_t) = \mathcal{N}(f_t^n,\, Q_t^n). \tag{13b}$$

Proof: Basic facts about conditionally linear models, see e.g., [19]. ■

We can now form the linear system (5b) – (5c) for each particle, $\{x_t^{n,(i)}\}_{i=1}^N$ and estimate the linear states using the Kalman filter. This requires one Kalman filter associated with each particle. The overall algorithm for estimating the states in Model 2 is given in Algorithm 1. From this algorithm it should be clear that the only difference from the standard particle filter is that the time update (prediction) stage has been changed. In the standard particle filter the prediction stage is given solely by step 4b in Algorithm 1. Step 4a is referred to as the *measurement update* in the Kalman filter [21]. Furthermore, we obtain the prediction of the nonlinear state variables, $\hat{x}_{t+1|t}^n$ in step 4b. According to (5a) the prediction of the nonlinear state variables do not contain any information about the linear state variables. This implies that $\hat{x}_{t+1|t}^n$ cannot be used to improve the quality of the estimates of the linear state variables. However, if we generalize Model 2 by imposing a dependence between the linear and the nonlinear state variables in (5a) we can use the prediction of the nonlinear state variables to improve the estimates of the linear state variables. In the subsequent section we will elaborate on how this affects the state estimation.

C. Triangular Model

Let us now extend Model 2 by including the term $A_t^n(x_t^n)x_t^l$ in the nonlinear state equation. This results in a "triangular model", defined below.

Model 3:

$$x_{t+1}^n = f_t^n(x_t^n) + A_t^n(x_t^n)x_t^l + w_t^n, \tag{14a}$$

$$x_{t+1}^l = A_t^l(x_t^n)x_t^l + w_t^l, \tag{14b}$$

$$y_t = h_t(x_t^n) + C_t(x_t^n)x_t^l + e_t, \tag{14c}$$

with the same assumptions as in Model 2. □

Now, from (14a) it is clear that $\hat{x}_{t+1|t}^n$ does indeed contain information about the linear state variables. This implies that there will be information about the linear state variable, x_t^l, in the prediction of the nonlinear state variable, $\hat{x}_{t+1|t}^n$. To understand how this affects the derivation let us assume that step 4b in Algorithm 1 has just been completed. This means that the predictions, $\hat{x}^n_{t+1|t}$, are available and the model can be written (the information in the measurement, y_t, has already been used in step 4a)

$$x^l_{t+1} = A^l_t x^l_t + w^l_t, \tag{15a}$$
$$z_t = A^n_t x^l_t + w^n_t, \tag{15b}$$

where

$$z_t = x^n_{t+1} - f^n_t. \tag{15c}$$

We can interpret z_t as a measurement and w^n_t as the corresponding measurement noise. Since (15) is a linear state-space model, with Gaussian noise the optimal state estimate is given by the Kalman filter, according to

$$\hat{x}^{l*}_{t|t} = \hat{x}^l_{t|t} + L_t(z_t - A^n_t \hat{x}^l_{t|t}), \tag{16a}$$
$$P^*_{t|t} = P_{t|t} - L_t N_t L^T_t, \tag{16b}$$
$$L_t = P_{t|t}(A^n_t)^T N^{-1}_t, \tag{16c}$$
$$N_t = A^n_t P_{t|t}(A^n_t)^T + Q^n_t. \tag{16d}$$

We have used "$*$" to distinguish this second measurement update from the first one. Furthermore, $\hat{x}^l_{t|t}$, and $P_{t|t}$ are given by (10a) and (10b) respectively. The final step is to merge this second measurement update with the time update to obtain the predicted states. This results in

$$\hat{x}^l_{t+1|t} = A^l_t \hat{x}^l_{t|t} + L_t(z_t - A^n_t \hat{x}^l_{t|t}), \tag{17a}$$
$$P_{t+1|t} = A^l_t P_{t|t}(A^l_t)^T + Q^l_t - L_t N_t L^T_t, \tag{17b}$$
$$L_t = A^l_t P_{t|t}(A^n_t)^T N^{-1}_t, \tag{17c}$$
$$N_t = A^n_t P_{t|t}(A^n_t)^T + Q^n_t. \tag{17d}$$

For a formal proof of this the reader is referred to Appendix I. To make Algorithm 1 valid for the more general Model 3 only the time update equation in the Kalman filter (11) has to be replaced by (17).

The second measurement update is called measurement update due to the fact that the mathematical structure is the same as a measurement update in the Kalman filter. However, strictly speaking it is not really a measurement update, since there does not exist any new measurement. It is better to think of this second update as a correction to the real measurement update, using the information in the prediction of the nonlinear state variables.

D. The General Case

In the previous two sections we have illustrated the mechanisms underlying the marginalized particle filter. We are now ready to apply the marginalized particle filter to the most general model.

Model 4:

$$x^n_{t+1} = f^n_t(x^n_t) + A^n_t(x^n_t)x^l_t + G^n_t(x^n_t)w^n_t, \tag{18a}$$
$$x^l_{t+1} = f^l_t(x^n_t) + A^l_t(x^n_t)x^l_t + G^l_t(x^n_t)w^l_t, \tag{18b}$$
$$y_t = h_t(x^n_t) + C_t(x^n_t)x^l_t + e_t, \tag{18c}$$

where we assume that the state noise is white and Gaussian distributed with

$$w_t = \begin{bmatrix} w^l_t \\ w^n_t \end{bmatrix} \sim \mathcal{N}(0, Q_t), \quad Q_t = \begin{bmatrix} Q^l_t & Q^{ln}_t \\ (Q^{ln}_t)^T & Q^n_t \end{bmatrix}, \tag{19a}$$

and that the measurement noise is white and Gaussian distributed according to

$$e_t \sim \mathcal{N}(0, R_t). \tag{19b}$$

Furthermore, x^l_0 is Gaussian,

$$x^l_0 \sim \mathcal{N}(\bar{x}_0, \bar{P}_0). \tag{19c}$$

The density of x^n_0 can be arbitrary, but it is assumed known. □

In certain cases some of the assumptions can be relaxed. This will be discussed in the subsequent section. Before we move on it is worthwhile to explain how models used in some applications of marginalization relate to Model 5. In [23] the marginalized particle filter was applied to underwater navigation using a model corresponding to (18), save the fact that $G^n_t = I, G^l_t = I, f^l_t = 0, A^n_t = 0$. In [18] a model corresponding to linear state equations and a nonlinear measurement equation is applied to various problems, such as car positioning, terrain navigation, and target tracking. Due to its relevance this model will be discussed in more detail in Section III. Another special case of Model 4 has been applied to problems in communication theory in [9], [35]. The model used there is linear. However, depending on an indicator variable the model changes. Hence, we can think of this indicator variable as the nonlinear state variable in Model 5. A good and detailed explanation of how to use the marginalized particle filter for this case can be found in [14]. They refer to the model as a jump Markov linear system.

Analogously to what has been done previously, the filtering distribution, $p(x_t|Y_t)$ is split using Bayes' theorem according to

$$p(x^l_t, X^n_t|Y_t) = p(x^l_t|X^n_t, Y_t)p(X^n_t|Y_t). \tag{20}$$

The linear state variables are estimated using the Kalman filter in a slightly more general setting than which was previously discussed. However, it is still the same three steps that are executed in order to estimate the linear state variables. The first step is a measurement update using the information available in y_t. The second step is a measurement update using the information available in $\hat{x}^n_{t+t|t}$ and finally there is a time update. The following theorem explains how the linear state variables are estimated.

Theorem 2.1: Using Model 4 the conditional probability density functions for x^l_t and x^l_{t+1} are given by

$$p(x^l_t|X^n_t, Y_t) = \mathcal{N}(\hat{x}^l_{t|t}, P_{t|t}), \tag{21a}$$
$$p(x^l_{t+1}|X^n_{t+1}, Y_t) = \mathcal{N}(\hat{x}^l_{t+1|t}, P_{t+1|t}), \tag{21b}$$

where

$$\hat{x}^l_{t|t} = \hat{x}^l_{t|t-1} + K_t(y_t - h_t - C_t\hat{x}^l_{t|t-1}), \tag{22a}$$
$$P_{t|t} = P_{t|t-1} - K_t M_t K_t^T, \tag{22b}$$
$$M_t = C_t P_{t|t-1} C_t^T + R_t, \tag{22c}$$
$$K_t = P_{t|t-1} C_t^T M_t^{-1}, \tag{22d}$$

and

$$\hat{x}^l_{t+1|t} = \bar{A}^l_t \hat{x}^l_{t|t} + G^l_t (Q^{ln}_t)^T (G^n_t Q^n_t)^{-1} z_t + f^l_t + L_t(z_t - A^n_t \hat{x}^l_{t|t}), \tag{23a}$$
$$P_{t+1|t} = \bar{A}^l_t P_{t|t} (\bar{A}^l_t)^T + G^l_t \bar{Q}^l_t (G^l_t)^T - L_t N_t L_t^T, \tag{23b}$$
$$N_t = A^n_t P_{t|t} (A^n_t)^T + G^n_t Q^n_t (G^n_t)^T, \tag{23c}$$
$$L_t = \bar{A}^l_t P_{t|t} (A^n_t)^T N_t^{-1}, \tag{23d}$$

where we have defined

$$z_t = x^n_{t+1} - f^n_t, \tag{24a}$$
$$\bar{A}^l_t = A^l_t - G^l_t (Q^{ln}_t)^T (G^n_t Q^n_t)^{-1} A^n_t, \tag{24b}$$
$$\bar{Q}^l_t = Q^l_t - (Q^{ln}_t)^T (Q^n_t)^{-1} Q^{ln}_t. \tag{24c}$$

Proof: See Appendix I. ■

It is worth noting that if the cross-covariance, Q^{ln}_t, between the two noise sources w^n_t and w^l_t is zero, then $\bar{A}^l_t = A^l_t$ and $\bar{Q}^l_t = Q^l_t$. We have now taken care of the first density, $p(x^l_t|X^n_t, Y_t)$, on the right hand side in (20). In order for the estimation to work we also have to consider the second density, $p(X^n_t|Y_t)$, in (20). This can be written as

$$p(X^n_t|Y_t) = \frac{p(y_t|X^n_t, Y_{t-1}) p(x^n_t|X^n_{t-1}, Y_{t-1})}{p(y_t|Y_{t-1})} p(X^n_{t-1}|Y_{t-1}), \tag{25}$$

where an approximation of $p(X^n_{t-1}|Y_{t-1})$ is provided by the previous iteration of the particle filter. The analytical expressions for $p(y_t|X^n_t, Y_{t-1})$ and $p(x^n_t|X^n_{t-1}, Y_{t-1})$ are provided by the following theorem.

Theorem 2.2: For Model 4 we have that

$$p(y_t|X^n_t, Y_{t-1}) = \mathcal{N}(h_t + C_t \hat{x}^l_{t|t-1},\ C_t P_{t|t-1} C_t^T + R_t), \tag{26a}$$
$$p(x^n_{t+1}|X^n_t, Y_t) = \mathcal{N}(f^n_t + A^n_t \hat{x}^l_{t|t},\ A^n_t P_{t|t} (A^n_t)^T + G^n_t Q^n_t (G^n_t)^T). \tag{26b}$$

Proof: Basic facts about conditionally linear models, see [19]. The details for this particular case can be found in [34]. ■

We have now derived all the details for estimating the states in Model 4. The complete algorithm is Algorithm 1. As pointed out before, the only difference between this algorithm and the standard particle filtering algorithm is that the prediction stage is different. If steps 4a and 4c are removed from Algorithm 1 we obtain the standard particle filter algorithm.

In this article we have used the most basic form of the particle filter. Several more refined variants exist, which in certain applications can give better performance. However, since the aim of this article is to communicate the idea of marginalization in a general linear/nonlinear state-space model we have used the standard particle filter. It is straightforward to adjust the algorithm given in this paper to accommodate e.g., the auxiliary particle filter [32] and the Gaussian particle filter [25], [26]. Several ideas are also given in the articles collected in [11].

The estimates as expected means of the linear state variables and their covariances are given by [31]

$$\hat{x}^l_{t|t} = \sum_{i=1}^{N} \tilde{q}^{(i)}_t \hat{x}^{l,(i)}_{t|t} \approx E_{p(x^l_t|Y_t)}\left[x^l_t\right], \tag{27a}$$
$$\hat{P}_{t|t} = \sum_{i=1}^{N} \tilde{q}^{(i)}_t \left(P^{(i)}_{t|t} + (\hat{x}^{l,(i)}_{t|t} - \hat{x}^l_{t|t})(\hat{x}^{l,(i)}_{t|t} - \hat{x}^l_{t|t})^T\right) \tag{27b}$$
$$\approx E_{p(x^l_t|Y_t)}\left[\left((x^l_t)^2 - E_{p(x^l_t|Y_t)}\left[(x^l_t)^2\right]\right)^2\right]. \tag{27c}$$

where $\tilde{q}^{(i)}_t$ are the normalized importance weights, provided by step 2 in Algorithm 1.

III. Important Special Cases and Extensions

Model 4 is quite general indeed and in most applications we use special cases of it. This fact, together with some extensions will be the topic of this section.

The special cases are just reductions of the general results presented in the previous section. However, they still deserve some attention in order to highlight some important mechanisms. It is worth mentioning that linear sub-structures can enter the model more implicitly as well. For instance via modeling colored noise and via sensor offsets and trends. These modeling issues are treated in several introductory texts on Kalman filtering, see e.g., Section 8.2.4 in [17]. In the subsequent section we discuss some noise modeling aspects. This is followed by a discussion of a model with linear state equations and a nonlinear measurement equation. Models of this type are important in applications such as positioning, navigation, collision avoidance and target tracking [18], [4].

A. Generalized Noise Assumptions

The Gaussian noise assumption can be relaxed in two special cases. First, if the measurement equation (18c) does not depend on the linear state variables, x^l_t, i.e., $C_t(x^n_t) = 0$, the measurement noise can be arbitrarily distributed. In this case (18c) does not contain any information about the linear state variables, and hence cannot be used in the Kalman filter. It is solely used in the particle filter part of the algorithm, which can handle all probability density functions.

Second, if $A^n_t(x^n_t) = 0$ in (18a), the nonlinear state equation will be independent of the linear states, and hence cannot be used in the Kalman filter. This means that the state noise, w^n_t, can be arbitrarily distributed.

The noise covariances can depend on the nonlinear state variables, i.e., $R_t = R_t(x^n_t)$ and $Q_t = Q_t(x^n_t)$. This is useful for instance in terrain navigation, where the nonlinear state variable includes information about the position. Using the horizontal position and a geographic information system (GIS) on-board the aircraft we can motivate noise covariances depending on the characteristics of the terrain at the current horizontal position. We will elaborate on this issue in Section IV.

B. An Important Model Class

A quite important special case of Model 4, is a model with linear state equations and a nonlinear measurement equation. In Model 5 below such a model is defined.

Model 5:

$$x^n_{t+1} = A^n_{n,t} x^n_t + A^n_{l,t} x^l_t + G^n_t w^n_t, \tag{28a}$$
$$x^l_{t+1} = A^l_{n,t} x^n_t + A^l_{l,t} x^l_t + G^l_t w^l_t, \tag{28b}$$
$$y_t = h_t(x^n_t) \qquad + e_t, \tag{28c}$$

with $w^n_t \sim \mathcal{N}(0, Q^n_t)$ and $w^l_t \sim \mathcal{N}(0, Q^l_t)$. The distribution for e_t can be arbitrary, but it is assumed known. □

The measurement equation (28c) does not contain any information about the linear state variable, x^l_t. Hence, as far as the Kalman filter is concerned (28c) cannot be used in estimating the linear states. Instead all information from the measurements enter the Kalman filter implicitly via the second measurement update using the nonlinear state equation (28a) and the prediction of the nonlinear state, $\hat{x}^n_{t+1|t}$, as a measurement. This means that in Algorithm 1, step 4a can be left out. In this case the second measurement update is much more than just a correction to the first measurement update. It is the only way in which the information in y_t enters the algorithm.

Model 5 is given special attention as several important state estimation problems can be modeled in this way. Examples include positioning, target tracking and collision avoidance [18]. For more information on practical matters concerning modeling issues, see e.g., [29], [28], [4], [31]. In the applications mentioned above the nonlinear state variable, x^n_t, usually corresponds to the position, whereas the linear state variable, x^l_t, corresponds to velocity, acceleration and bias terms.

If we compare Model 5 to Model 4 we see that the matrices A^n_t, A^l_t, G^n_t, and G^l_t are independent of x^n_t in Model 5, which implies that

$$P^{(i)}_{t|t} = P_{t|t} \qquad \forall\, i = 1, \ldots, N. \tag{29}$$

This follows from (23b) – (23d) in Theorem 2.1. According to (29) we only need one instead of N Riccati recursions, which leads to a substantial reduction in computational complexity. This is of course very important if we are to pursue a real-time implementation of the state estimation.

If we have almost linear dynamics in (18a) – (18b) we can linearize to get a model described by (28a) – (28b). Then we can use the extended Kalman filter instead of the Kalman filter. As is explained in [29], [28] it is common that the system model is almost linear, whereas the measurement model is severely nonlinear. In these cases use the particle filter for the severe nonlinearities and the extended Kalman filter for the mild nonlinearities.

IV. Integrated Aircraft Navigation

Briefly, the integrated navigation system in the Swedish fighter aircraft Gripen consists of an inertial navigation system (INS), a terrain-aided positioning (TAP) system and an integration filter. This filter fuses the information from INS with the information from TAP, see Fig. 1. For a more thorough description of this system the reader is referred to [31]. TAP is currently based on a point-mass filter as presented in [6], where it is also demonstrated that the performance is quite good, close to the Cramér-Rao lower bound. Field tests conducted by the Swedish air force have confirmed the good precision. Alternatives based on the extended Kalman filter have been investigated [5], but shown to be inferior particularly in the transient phase (the EKF requires the gradient of the terrain profile, which is unambiguous only very locally). The point-mass filter, as described in [6], is likely to be changed to a marginalized particle filter in the future for Gripen, see Fig. 2. TAP and INS are the primary sensors. Secondary sensors (GPS and so on) are used only when available and reliable. The current terrain-aided positioning filter has three states (horizontal position and heading), while the integrated navigation system estimates the accelerometer and gyroscope errors, and some other states. The integrated navigation is currently based on a Kalman filter with 27 states, taking INS and TAP as primary input signals.

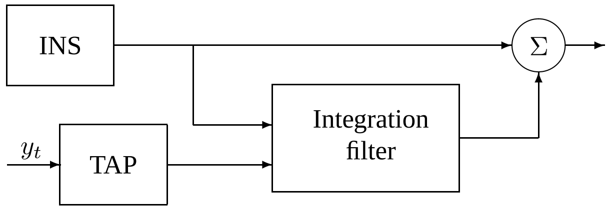

Fig. 1. The integrated navigation system consists of an inertial navigation system (INS), a terrain-aided positioning (TAP) system and an integration filter. The integration filter fuses the information from INS with the information from TAP, using a marginalized particle filter.

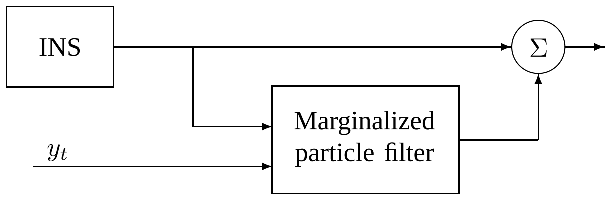

Fig. 2. Using the marginalized particle filter for navigation in Gripen. The terrain information is now incorporated directly in the marginalized particle filter. The radar altimeter delivers the hight measurement y_t.

The Kalman filter for integrated navigation requires Gaussian variables. However, TAP gives a multi-modal unsymmetric distribution in the Kalman filter measurement equation and it has to be approximated with a Gaussian distribution before used in the Kalman filter. This results in severe performance degradation in many cases, and is a common cause for filter divergence and system re-initialization.

The appealing new strategy is to merge the two state vectors into one, and solve integrated navigation and terrain-aided positioning in one filter. This filter should include all 27 states, which effectively would prevent application of the particle filter. However, the state equation is almost linear, and only three states enter the measurement equation nonlinearly, namely horizontal position and heading. Once linearization (and the use of EKF) is absolutely ruled out marginalization would be the only way to overcome the computational complexity.

A first step in this direction was taken in [18], where a six dimensional model was used for integrated navigation. In six dimensions, the particle filter is possible to use, but we can do better. As demonstrated in [18], 4000 particles in the marginalized filter outperforms 60000 particles in the standard particle filter.

The feasibility study presented here applies marginalization to a more realistic nine dimensional sub-model of the total integrated navigation system. Already here, the dimensionality has proven to be too large for the particle filter to be applied directly. The example contains all ingredients of the total system, and the principle is scalable to the full 27-dimensional state vector. The model can be simulated and evaluated in a controlled fashion, see [31] for more details. Here we present the results from field trials.

A. The Dynamic Model

In order to apply the marginalized particle filter to the navigation problem we need a dynamic model of the aircraft. We will only discuss the structure of this model. For details the reader is referred to [31] and the references therein. The errors in the states are estimated instead of the absolute states. The reason is that the dynamics of the errors are typically much slower than the dynamics of the absolute states. The model has the following structure

$$x^n_{t+1} = A^n_{n,t}x^n_t + A^n_{l,t}x^l_t + G^n_t w^n_t, \quad (30a)$$

$$x^l_{t+1} = A^l_{n,t}x^n_t + A^l_{l,t}x^l_t + G^l_t w^l_t, \quad (30b)$$

$$y_t = h\left(\begin{bmatrix} L_t \\ l_t \end{bmatrix} + x^n_t\right) + e_t. \quad (30c)$$

There are 7 linear states, and 2 nonlinear states. The linear states consist of 2 velocity states and 3 states for the aircraft in terms of heading, roll, and pitch. Finally there are 2 states for the accelerometer bias. The nonlinear states correspond to the error in the horizontal position, which is expressed in latitude, L_t, and longitude, l_t.

The total dimension of the state vector is thus 9, which is to large to be handled by the particle filter. The highly nonlinear nature of measurement equation (30c), due to the terrain elevation database, implies that we cannot use an extended Kalman filter. However, the model described by (30) clearly fits into the framework of the marginalized particle filter.

The measurement noise in (30c) deserves some special attention. The radar altimeter, which is used to measure the ground clearance, interprets any echo as the ground. This is a problem when flying over trees. The tree tops will be interpreted as the ground, with a false measurement as a result. One simple, but effective solution to this problem is to model the measurement noise as

$$p_{e_t}(\cdot) = \pi\mathcal{N}(m_1, \sigma_1) + (1 - \pi)\mathcal{N}(m_2, \sigma_2), \quad (31)$$

where π is the probability of obtaining an echo from the ground, and $(1-\pi)$ is the probability of obtaining an echo from the tree tops. The probability density function (31) is shown in Fig. 3. Experiments have shown that this, in spite of its simplicity, is a quite accurate model [10]. Furthermore, we can allow m_1, m_2, σ_1, σ_2, and π in (31) to depend on the current horizontal position, L_t, l_t. In this way we can infer information from the terrain data base on the measurement noise in the model. This allows us to model for instance whether we are flying over open water or over a forest.

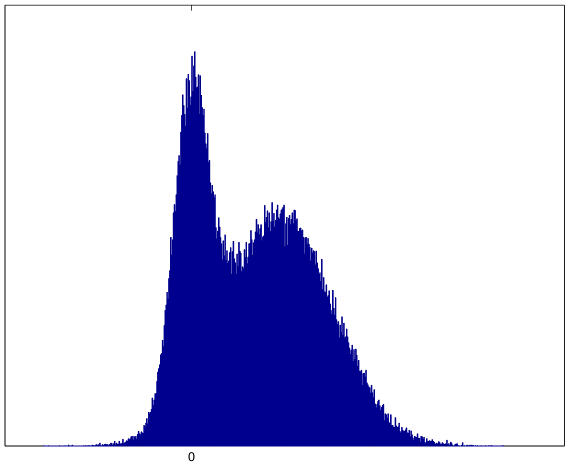

Fig. 3. A typical histogram of the error in the data from the radar altimeter. The first peak corresponds to the error in the ground reading and the second peak corresponds to the error in the readings from the tree tops.

B. Result

The flight that has been used is shown in Fig. 4. This is a fairly tough flight for the algorithm, in the sense that during some intervals data are missing, and sometimes the radar altimeter readings become unreliable. This happens at high altitudes and during sharp turns (large roll angle), respectively. In order to get a fair understanding of the algorithms performance, 100 Monte Carlo simulations with the same data have been performed, where only the noise realizations have been changed from one simulation to the other. Many parameters have to be chosen, but we only comment on the number of particles (see [15] for more details). In Fig. 5 we present a plot of the error in horizontal position as a function of time, for different number of particles. The true position is provided by the differential GPS (DGPS). From this figure it is obvious that the estimate improves as more particles are used. This is natural since the theory states that the densities are approximated better the more particles used. The difference in performance is mainly during the transient, where it can be motivated to use more particles. By increasing the number of particles the convergence time is significantly reduced and a better estimate is obtained. In our case this is true up to 5000 particles. Hence, we choose to use 5000 particles for this study.

In Fig. 6 the estimation error in the horizontal position is shown, together with the altitude profile of the aircraft and the ground elevation. The scale of all axes in the figures are between 0 and 1, due to secrecy reasons. During two intervals (illustrated in the upper plot in Fig. 6), when the aircraft is flying at a very high altitude the radar altimeter does not deliver any information. From the bottom plot in Fig. 6 we see that the estimation error increases during these intervals. However, when the measurements return the estimate

Fig. 4. The flight path used for testing the algorithm. The flight path is clockwise and the dark regions in the figure are open water.

converges again. Towards the end of the flight the estimation error increases, due to the sharp turns (see Fig. 4). There is not

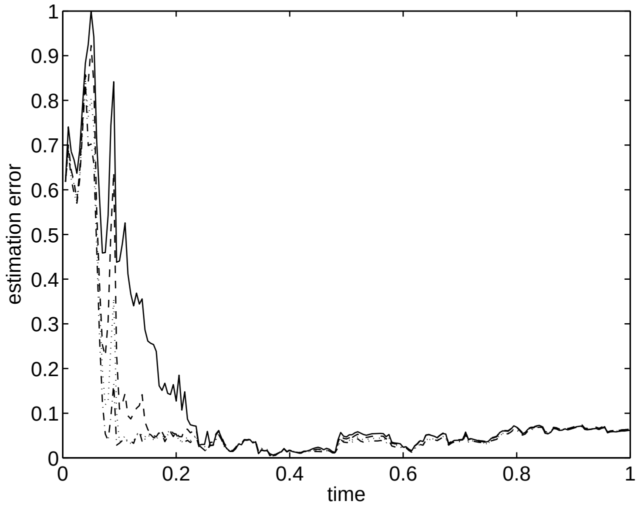

Fig. 5. The horizontal position error as a function of time for different numbers of particles. The solid line corresponds to 1200 particles, the dashed 2500 particles, the dotted 5000 particles, and the dash-dotted 10000 particles. We have used the marginalized particle filter given in Algorithm 1.

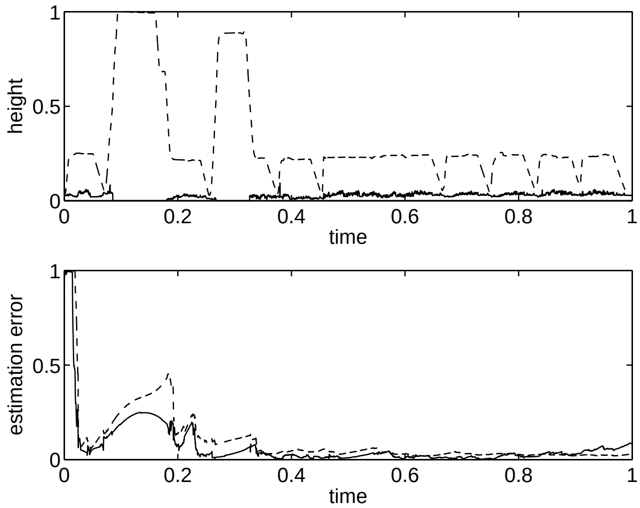

Fig. 6. In the top plot the altitude profile of the aircraft (dashed) and the ground elevation (solid) is shown. The bottom plot shows the horizontal estimation error (solid) and the corresponding standard deviation (dashed).

time enough for the algorithm to converge between these turns. The algorithm can be further improved, and in [15] several suggestions are given.

The conclusion from this study is that the marginalized particle filter performs well, and provides an interesting and powerful alternative to methods currently used in integrated aircraft navigation systems.

V. Conclusions

We have systematically applied marginalization techniques to general nonlinear and non-Gaussian state-space models, with linear sub-structures. This has been done in several steps, where each step implies a certain modification of the standard particle filter. The first step was to associate one Kalman filter with each particle. These Kalman filters were used to estimate the linear states. The second step was to use the prediction of the nonlinear state as an additional measurement. This was used to obtain better estimates of the linear state variables. Finally the complete details for the marginalized particle filter was derived for a general nonlinear and non-Gaussian state-space model. We have also described several important special cases. Conditions that imply that all the Kalman filters will obey the same Riccati recursion were given.

Finally, a terrain navigation application with real data from the Swedish fighter aircraft Gripen was presented. The particle filter is not a feasible algorithm for the full nine-state model since a huge number of particles would be needed. However, since only two states (the aircrafts horizontal position) appear nonlinearly in the measurement equation, a special case of the general marginalization algorithm can be applied. A very good result can be obtained with only 5000 particles, which is readily possible to implement in the computer currently used in the aircraft.

Appendix I
Proof for Theorem 2.1

The proof of (16) and (17) is provided as a special case of the proof below.

Proof: For the sake of notational brevity we will suppress the dependence on x_t^n in (18) in this proof. Let us start by writing (18) as

$$x_{t+1}^l = f_t^l + A_t^l x_t^l + G_t^l w_t^l, \tag{32a}$$
$$z_t^1 = A_t^n x_t^l + G_t^n w_t^n, \tag{32b}$$
$$z_t^2 = C_t x_t^l + e_t, \tag{32c}$$

where z_t^1 and z_t^2 are defined as

$$z_t^1 = x_{t+1}^n - f_t^n, \tag{32d}$$
$$z_t^2 = y_t - h_t, \tag{32e}$$

Inspection of the above equations gives that z_t^1 and z_t^2 can both be thought of as measurements, since mathematically (32b) and (32c) possess the structure of measurement equations. We have to take care of the fact that there is a cross-correlation between the two noise processes w_t^l and w_t^n, since $Q_t^{ln} \neq 0$ in (19a). We can use the Gram-Schmidt procedure to de-correlate the noise [17], [21]. Instead of w_t^l we can use

$$\begin{aligned}\bar{w}_t^l &= w_t^l - \mathrm{E}[w_t^l (w_t^n)^T](\mathrm{E}[w_t^n (w_t^n)^T])^{-1} w_t^n \\ &= w_t^l - Q_t^{ln}(Q_t^n)^{-1} w_t^n,\end{aligned} \tag{33}$$

resulting in $\mathrm{E}[\bar{w}_t^l (w_t^n)^T] = 0$ and

$$\bar{Q}_t^l = \mathrm{E}[\bar{w}_t^l (\bar{w}_t^l)^T] = Q_t^l - Q_t^{ln}(Q_t^n)^{-1} Q_t^{ln}. \tag{34}$$

We can now rewrite (32a) using (32b) and (33) according to (we assume that G_t^n is invertible. The case of a non-invertible G_t^n is treated in [5])

$$\begin{aligned}x_{t+1}^l &= A_t^l x_t^l + G_t^l[\bar{w}_t^l + Q_t^{ln}(Q_t^n)^{-1}(G_t^n)^{-1}(z_t^1 \\ &\quad - A_t^n x_t^l)] + f_t^l, \end{aligned}\tag{35}$$
$$= \bar{A}_t^l x_t^l + G_t^l \bar{w}_t^l + G_t^l Q_t^{ln}(G_t^n Q_t^n)^{-1} z_t^1 + f_t^l, \tag{36}$$

where

$$\bar{A}_t^l = A_t^l - G_t^l Q_t^{ln}(G_t^n Q_t^n)^{-1} A_t^n. \tag{37}$$

Our de-correlated system is

$$x_{t+1}^l = f_t^l + \bar{A}_t^l x_t^l + G_t^l Q_t^{ln}(G_t^n Q_t^n)^{-1} z_t^1 + G_t^l \bar{w}_t^l, \tag{38a}$$
$$z_t^1 = A_t^n x_t^l + G_t^n w_t^n, \tag{38b}$$
$$z_t^2 = C_t x_t^l + e_t, \tag{38c}$$

which is a linear system with Gaussian noise. Moreover, from (32d) and (32e) we have that Z_t^1 and Z_t^2 are known if X_{t+1}^n and Y_t are known. We are now ready to start the actual proof of the theorem, which will be done using induction. At time zero we have; $p(x_0^l|X_0^n, Y_{-1}) = p(x_0^l|x_0^n) = \mathcal{N}(\bar{x}_0^l, \bar{P}_0)$. Let us now assume that $p(x_t^l|X_t^n, Y_{t-1})$ is Gaussian at an arbitrary time, t.

The recursions are divided into three parts. First, we use the information available in the actual measurement, y_t, i.e., z_t^2. Once the measurement update has been performed we have the estimates, $\hat{x}_{t|t}^l$ and $P_{t|t}$. These can now be used to calculate the predictions of the nonlinear state, $\hat{x}_{t+1|t}^n$. These predictions will provide us with new information about the system. Second, we incorporate this new information by performing a second measurement update using the artificial measurement, z_t^1. Finally we do a time update using the result from the second step.

Part 1: Assume that both $p(x_t^l|X_t^n, Y_{t-1}) = \mathcal{N}(\hat{x}_{t|t-1}^l, P_{t|t-1})$ and z_t^2 are available. This means that we can compute

$$p(x_t^l|X_t^n, Y_t) = \frac{p(y_t|x_t^n, x_t^l) p(x_t^l|X_t^n, Y_{t-1})}{\int p(y_t|x_t^n, x_t^l) p(x_t^l|X_t^n, Y_{t-1}) dx_t^l}. \tag{39}$$

Using the fact that the measurement noise and thereby $p(y_t|x_t^n, x_t^l)$ is Gaussian and the Kalman filter [1] we have that $p(x_t^l|X_t^n, Y_t) = \mathcal{N}(\hat{x}_{t|t}^l, P_{t|t})$ where

$$\hat{x}_{t|t}^l = \hat{x}_{t|t-1}^l + K_t(z_t^2 - C_t \hat{x}_{t|t-1}^l), \tag{40a}$$
$$P_{t|t} = P_{t|t-1} - K_t M_t K_t^T, \tag{40b}$$
$$K_t = P_{t|t-1} C_t^T M_t^{-1}, \tag{40c}$$
$$M_t = C_t P_{t|t-1} C_t^T + R_t. \tag{40d}$$

Part 2: At this stage z_t^1 becomes available. Now using

$$p(x_t^l|X_{t+1}^n, Y_t) = \frac{p(x_{t+1}^n|x_t^n, x_t^l) p(x_t^l|X_t^n, Y_t)}{\int p(x_{t+1}^n|x_t^n, x_t^l) p(x_t^l|X_t^n, Y_t) dx_t^l} \tag{41}$$

analogously to part 1 we obtain $p(x_t^l|X_{t+1}^n, Y_t) = \mathcal{N}(\hat{x}_{t|t}^{l*}, P_{t|t}^*)$ where

$$\hat{x}_{t|t}^{l*} = \hat{x}_{t|t}^l + L_t(z_t^1 - A_t^n \hat{x}_{t|t}^l), \tag{42a}$$
$$P_{t|t}^* = P_{t|t} - L_t N_t^* L_t^T, \tag{42b}$$
$$L_t = P_{t|t}(A_t^n)^T (N_t^*)^{-1}, \tag{42c}$$
$$N_t^* = A_t^n P_{t|t}(A_t^n)^T + G_t^n Q_t^n (G_t^n)^T. \tag{42d}$$

Part 3: The final part is the time update, i.e., to compute

$$p(x_{t+1}^l|X_{t+1}^n, Y_t) = \int p(x_{t+1}^l|x_{t+1}^n, x_t^n, x_t^l) p(x_t^l|X_{t+1}^n, Y_t) dx_t^l \tag{43}$$

Since we have Gaussian state noise this corresponds to the time update handled by the Kalman filter. Hence, we have $p(x_{t+1}^l|X_{t+1}^n, Y_t) = \mathcal{N}(\hat{x}_{t+1|t}^l, P_{t+1|t})$ where

$$\begin{aligned}\hat{x}_{t+1|t}^l &= \bar{A}_t^l \hat{x}_{t|t}^l + G_t^l (Q_t^{ln})^T (G_t^n Q_t^n)^{-1} z_t^1 \\ &\quad + f_t^l + L_t(z_t^1 - A_t^n \hat{x}_{t|t}^l),\end{aligned} \tag{44a}$$
$$P_{t+1|t} = \bar{A}_t^l P_{t|t}(\bar{A}_t^l)^T + G_t^l \bar{Q}_t^l (G_t^l)^T - L_t N_t L_t^T, \tag{44b}$$
$$L_t = \bar{A}_t^l P_{t|t}(A_t^n)^T N_t^{-1}, \tag{44c}$$
$$N_t = A_t^n P_{t|t}(A_t^n)^T + G_t^n Q_t^n (G_t^n)^T. \tag{44d}$$

■

Acknowledgment

The authors would like to thank Petter Frykman for providing the plots in Section IV, which were generated as a part of his Master's thesis at Saab Aerospace in Linköping, Sweden. The authors would also like to thank the reviewers and the editor for their detailed and constructive comments on this paper.

REFERENCES

[1] B.D.O. Anderson and J.B. Moore. *Optimal Filtering*. Information and system science series. Prentice Hall, Englewood Cliffs, New Jersey, 1979.

[2] C. Andrieu and A. Doucet. Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society*, 64(4):827–836, 2002.

[3] C. Andrieu and S.J. Godsill. A particle filter for model based audio source separation. In *International Workshop on Independent Component Analysis and Blind Signal Separation (ICA 2000)*, Helsinki, Finland, June 2000.

[4] Y. Bar-Shalom and X-R. Li. *Estimation and Tracking: Principles, Techniques, and Software*. Artech House, 1993.

[5] N. Bergman. *Recursive Bayesian Estimation: Navigation and Tracking Applications*. PhD thesis, Linköping University, 1999. Dissertation No. 579.

[6] N. Bergman, L. Ljung, and F. Gustafsson. Terrain navigation using Bayesian statistics. *IEEE Control Systems Magazine*, 19(3):33–40, June 1999.

[7] G. Casella and C.P. Robert. Rao-blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

[8] R. Chen and J.S. Liu. Mixture Kalman filters. *Journal of the Royal Statistical Society*, 62(3):493–508, 2000.

[9] R. Chen, X. Wang, and J.S. Liu. Adaptive joint detection in flat-fading channels via mixture kalman filtering. *IEEE Transactions on Information Theory*, 46(6):2079–2094, 2000.

[10] C. Dahlgren. Non-linear black box modelling of jas 39 gripen's radar altimeter. LiTH-ISY-EX-1958, Automatic control and communications systems, Linköping university, Oct. 1998.

[11] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

[12] A. Doucet, S.J. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[13] A. Doucet, N. Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump Markov linear systems. Technical Report CUED/F-INFENG/TR 359, Signal Processing Group, Department of Engineering, University of Cambridge, Trupington street, CB2 1PZ Cambridge, 1999.

[14] A. Doucet, N. Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, 49(3):613–624, 2001.

[15] P. Frykman. Applied particle filters in integrated aircraft navigation. LiTH-ISY-EX-3406, Automatic control and communications systems, Linköping university, Apr. 2003.

[16] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. A novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113, 1993.

[17] F. Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons, 2000.

[18] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P-J. Nordlund. Particle filters for positioning, navigation and tracking. *IEEE Transactions on Signal Processing*, 50(2):425–437, Feb 2002.

[19] A.C. Harvey. *Forecasting, structural time series models and the Kalman filter*. Cambridge University Press, Cambridge, UK, 1989.

[20] A.H. Jazwinski. *Stochastic processes and filtering theory*. Mathematics in science and engineering. Academic Press, New York, 1970.

[21] T. Kailath, A.H. Sayed, and B. Hassibi. *Linear Estimation*. Information and System Sciences Series. Prentice Hall, Upper Saddle River, New Jersey, 2000.

[22] R. E. Kalman. A new approach to linear filtering and prediction problems. *Trans. AMSE, J. Basic Engineering*, 82:35–45, 1960.

[23] R. Karlsson. Particle filter and Cramér-Rao lower bound for underwater navigation. In *proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Hong Kong, Apr. 2003.

[24] R. Karlsson and F. Gustafsson. Particle filter for underwater navigation. In *Statistical Signal Processing Workshop (SSP'03)*, pages 509–512, St. Louis, USA, Sep. 2003.

[25] J.H. Kotecha and P.M. Djuric. Gaussian particle filtering. *IEEE Transactions on Signal Processing*, 51(10):2592–2601, 2003.

[26] J.H. Kotecha and P.M. Djuric. Gaussian sum particle filtering. *IEEE Transactions on Signal Processing*, 51(10):2602–2610, 2003.

[27] P. Li, R. Goodall, and V. Kadirkamanathan. Parameter estimation of railway vehicle dynamic model using Rao-Blackwellised particle filter. In *proceedings of the European Control Conference (ECC)*, Cambridge, UK, Sep. 2003.

[28] X.R. Li and V.P. Jilkov. A survey of maneuvering target tracking-part iii: Measurement models. In *proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, pages 423–446, San Diego, USA, Jul.–Aug. 2001.

[29] X.R. Li and V.P. Jilkov. Survey of maneuvering target tracking-part i: dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364, Oct. 2003.

[30] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. Springer, New York, USA, 2001.

[31] P-J. Nordlund. *Sequential Monte Carlo Filters and Integrated Navigation*. Licentiate thesis, Linköping university, 2002. Thesis No. 945.

[32] M.K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, June 1999.

[33] T. Schön and F. Gustafsson. Particle filters for system identification of state-space models linear in either parameters or states. In *proceedings of the 13th IFAC Symposium on System Identification*, pages 1287–1292, Rotterdam, The Netherlands, Sep. 2003.

[34] T. Schön. *On Computational Methods for Nonlinear Estimation*. Licentiate thesis, Linköping university, Oct. 2003. Thesis No. 1047.

[35] X. Wang, R. Chen, and D. Guo. Delayed-pilot sampling for mixture kalman filter with application in fading channels. *IEEE Transactions on Signal Processing*, 50(2):241–254, Feb. 2002.

Thomas Schön received the BSc degree in Business Administration and Economics in Feb. 2001, in Sep. 2001 he received the MSc degree in Applied Physics and Electrical Engineering and in Oct. 2003 he received the Licentiate of Engineering degree, all from Linköping University, Linköping, Sweden. He has since Dec. 2001 been pursuing the PhD degree at the division of automatic control and communication systems, Linköping University.

Fredrik Gustafsson received the MSc degree in Electrical Engineering in 1988 and the PhD degree in Automatic Control in 1992, both from Linköping University, Linköping, Sweden.

He is Professor of communication systems with the department of Electrical Engineering, Linköping University. His research is focused on statistical methods in signal processing, with applications to automotive, avionic, and communication systems.

Prof. Gustafsson is an associate editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING.

Per-Johan Nordlund received his MSc degree in Electrical Engineering in 1997 and his Licentiate degree of Engineering degree in automatic control in 2002, both from Linköping University, Linköping, Sweden. He is currently working as manager for the navigation department at Saab Aerospace, Linköping, Sweden.