The Marginalized Particle Filter in Practice

Thomas B. Schön, Rickard Karlsson and Fredrik Gustafsson
Division of Automatic Control
Department of Electrical Engineering
Linköping University, Sweden
SE–581 83 Linköping, Sweden
{schon, rickard, fredrik}@isy.liu.se

Abstract— The marginalized particle filter is a powerful combination of the particle filter and the Kalman filter, which can be used when the underlying model contains a linear sub-structure, subject to Gaussian noise. This paper will illustrate several positioning and target tracking applications, solved using the marginalized particle filter. Furthermore, we analyze several properties of practical importance, such as its computational complexity and how to cope with quantization effects.

1. Introduction

Many problems in for instance positioning and target tracking can be cast as nonlinear state estimation problems, where the uncertainty in the process model and/or in the measurement model may be non-Gaussian. Such a general model can be formulated according to

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

with state variable $x_t \in \mathbb{R}^m$, input signal u_t and measurements $\mathbb{Y}_t = \{y_i\}_{i=1}^t$, with known *probability density functions* (pdfs) for the process noise $p_w(w)$ and the measurement noise $p_e(e)$. Hence, traditional estimation methods based on the *Kalman filter* (KF) [25, 26], or linearized version thereof, do not always provide good performance. Over the past 40 years there has been several suggestions on how to tackle the problem of estimating the states in (1). An appealing solution is provided by the *particle filter* (PF) [11, 22, 37], which allows for a systematic treatment of both nonlinearities and non-Gaussian noise. However, due to the inherent computational complexity of the particle filter, real-time issues arise in many applications when the sampling rate is high. If the model includes a sub-structure with linear equations, subject to Gaussian noise, it is often possible to perform the estimation more efficiently. Here, this method is referred to as the *marginalized particle filter* (MPF), it is also known as the Rao-Blackwellized particle filter, see for instance [3, 4, 8, 11, 12, 40]. The MPF is a clever combination of the standard particle filter and the Kalman filter. It is a well known fact that in some cases it is possible to obtain better estimates, i.e., estimates with reduced variance, using the marginalized particle filter instead of using the standard particle filter [14].

The aim of this paper is to explain how the marginalized particle filter works in practice. We will try to achieve this by considering several applications where we have successfully applied the MPF. Since we cannot cover all the details in this paper references to more detailed descriptions are provided. Furthermore, the algorithm's computational complexity and the presence of quantization effects are analyzed, due to their importance in practical applications. To summarize, the analysis and applications covered are

Theory and analysis:
- *Background theory*
- *Complexity analysis*
- *Quantization effects*

Positioning applications:
- *Underwater terrain-aided positioning*
- *Aircraft terrain-aided positioning*
- *Automotive map-aided positioning*

Target tracking applications:
- *Automotive target tracking*
- *Bearings-only target tracking*
- *Radar target tracking*

There are certainly more applications of the marginalized particle filter reported in the literature. Just to mention a few, there are communication applications [9, 44], nonlinear system identification [10, 33, 39], GPS navigation [20] and audio source separation [4].

The paper is organized as follows. In Section 2, the background theory and MPF algorithm are briefly introduced. The algorithm performance, computational complexity and ability to handle quantization effects are analyzed in Section 3. In Section 4, the applications are introduced and the structure of the underlying models is reviewed. The positioning and target tracking application are described in more detail in Section 5 and Section 6, respectively. Finally, Section 7 provides a concluding discussion of some lessons learned in using the marginalized particle filter.

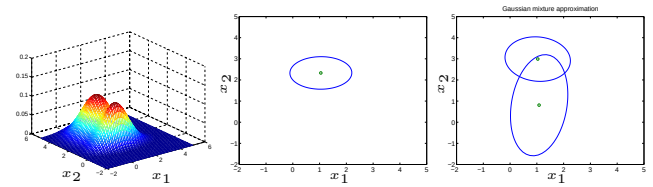
2. Marginalized Particle Filter

The aim of recursively estimating the filtering density $p(x_t|\mathbb{Y}_t)$ can be accomplished using the standard particle filter. However, if there is a linear sub-structure, subject to Gaussian noise, present in the model this can

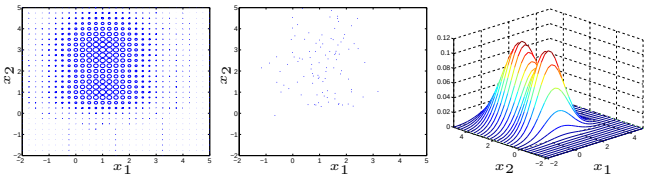
be exploited to obtain better estimates and possibly reduce the computational demand as well. This is the motivation underlying the marginalized particle filter.

Representation

The task of nonlinear filtering can be split into two parts: representation of the filtering probability density function and propagation of this density during the time and measurement update stages. Figure 1 illustrate different representations of the filtering density for a two-dimensional example. The *extended Kalman filter* (EKF) [2, 25], can be interpreted as using one Gaussian distribution for representation and the propagation is performed according to a linearized model. The *Gaussian sum filter*, [2, 41], extends the EKF to be able to represent multi-modal distributions, still with an approximate propagation.



(a) True pdf. (b) Gaussian approximation. (c) Gaussian sum approximation.



(d) Grid-based approximation. (e) Particle approximation. (f) PF posterior pdf (waterfall view).

Figure 1. True probability density function (pdf) and different approximate representations, in order of appearance, Gaussian, Gaussian sum, point-masses (grid-based approximation), particle samples and waterfall view that corresponds to the MPF.

Figure 1(d)–(f) illustrates numerical approaches where the exact nonlinear relations present in the model are used for propagation. The *point-mass filter* (grid-based approximation), [6] employ a regular grid, where the grid weight is proportional to the posterior. The *particle filter* (PF), [22], represents the posterior by a stochastic grid in form of a set of samples, where all particles (samples) have the same weight. Finally, the *marginalized particle filter* (MPF) uses a stochastic grid for some of the states, and Gaussian distributions for the rest. That is, the MPF can be interpreted as a particle representation for a subspace of the state dimension, where each particle has an associated Gaussian distribution for the remaining state dimensions. This is the *waterfall view* in Figure 1(f). It will be demonstrated that an exact nonlinear propagation is still possible if there is a linear sub-structure in the model. An important model class has the property that the (co-)variance is the same for all particles, which simplifies computations significantly.

Model

Consider a state vector x_t, which can be partitioned according to

$$x_t = \begin{pmatrix} x_t^l \\ x_t^n \end{pmatrix}, \tag{2}$$

where x_t^l denotes the linear states and x_t^n denotes the nonlinear states, in the dynamics and measurement relation. A rather general model with the properties discussed above is given by

$$x_{t+1}^n = f_t^n(x_t^n) + A_t^n(x_t^n)x_t^l + G_t^n(x_t^n)w_t^n, \tag{3a}$$
$$x_{t+1}^l = f_t^l(x_t^n) + A_t^l(x_t^n)x_t^l + G_t^l(x_t^n)w_t^l, \tag{3b}$$
$$y_t = h_t(x_t^n) + C_t(x_t^n)x_t^l + e_t, \tag{3c}$$

where the state noise is assumed white and Gaussian distributed with

$$w_t = \begin{pmatrix} w_t^l \\ w_t^n \end{pmatrix} \sim \mathcal{N}(0, Q_t),\ Q_t = \begin{pmatrix} Q_t^l & Q_t^{ln} \\ (Q_t^{ln})^T & Q_t^n \end{pmatrix}. \tag{3d}$$

The measurement noise is assumed white and Gaussian distributed according to

$$e_t \sim \mathcal{N}(0, R_t). \tag{3e}$$

Furthermore, x_0^l is Gaussian,

$$x_0^l \sim \mathcal{N}(\bar{x}_0, \bar{P}_0). \tag{3f}$$

Finally, the density of x_0^n can be arbitrary, but it is assumed known. More specifically, conditioned on the nonlinear state variables there is a linear sub-structure, subject to Gaussian noise available in (3), given by (3b).

Algorithm

Bayesian estimation methods, such as the particle filter, provide estimates of the filtering density function, $p(x_t|\mathbb{Y}_t)$. By employing the fact

$$p(x_t^l, \mathbb{X}_t^n|\mathbb{Y}_t) = p(x_t^l|\mathbb{X}_t^n, \mathbb{Y}_t)p(\mathbb{X}_t^n|\mathbb{Y}_t), \tag{4}$$

we can put the problem in a description suitable for the MPF framework, i.e., to analytically marginalize out the linear state variables from $p(x_t|\mathbb{Y}_t)$. Note that, $p(x_t^l|\mathbb{X}_t^n, \mathbb{Y}_t)$ is analytically tractable, since $\mathbb{X}_t^n$ is given. Hence, the underlying model is linear-Gaussian, and the pdf can be computed from the Kalman filter. Furthermore, an estimate of $p(\mathbb{X}_t^n|\mathbb{Y}_t)$ is provided by the particle filter. These two algorithms can then be combined into a single algorithm, the marginalized particle filter. Another name for this technique is the Rao-Blackwellized particle filter, and it has been known for quite some

time, see e.g., [3, 7, 8, 12, 14, 35, 40]. If the same numbers of particles are used in the standard particle filer and the marginalized particle filter, the latter will provide estimates of better or at least the same quality. Intuitively this makes sense, since the dimension of $p(x_t^n|\mathbb{Y}_t)$ is smaller than the dimension of $p(x_t|\mathbb{Y}_t)$, implying that the particles occupy a lower dimensional space. Furthermore, the optimal algorithm is used to estimate the linear state variables. For a detailed discussion regarding the improved accuracy of the estimates, see [13, 14].

The marginalized particle filter for estimating the states in a dynamic model in the form (3) is provided in Alg. 1. Since the focus of the present paper is on the practical aspects of Alg. 1, we will merely provide the intuition for this algorithm here. For a detailed derivation, see [40]. 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 4(b) in Alg. 1.

Let us now briefly discuss step 4 in Alg. 1. Step 4(a) is a standard Kalman filter measurement, update using the information available in the measurement y_t. Once this has been performed the new estimates of the linear states can be used to obtain a prediction of the nonlinear state $x_{t+1|t}^n$. This is performed in Step 4(b). Let us now consider model (3) conditioned on the nonlinear state variable. The conditioning implies that (3a) can be thought of as a measurement equation. This is used in step 4(c) together with a time update of the linear state estimates.

The estimates, as expected means, of the state variables and their covariances are given below.

$$\hat{x}_{t|t}^n = \sum_{i=1}^N \tilde{q}_t^{(i)} \hat{x}_{t|t}^{n,(i)}, \tag{9a}$$

$$\hat{P}_{t|t}^n = \sum_{i=1}^N \tilde{q}_t^{(i)} \left(\left(\hat{x}_{t|t}^{n,(i)} - \hat{x}_{t|t}^n \right) \left(\hat{x}_{t|t}^{n,(i)} - \hat{x}_{t|t}^n \right)^T \right), \tag{9b}$$

$$\hat{x}_{t|t}^l = \sum_{i=1}^N \tilde{q}_t^{(i)} \hat{x}_{t|t}^{l,(i)}, \tag{9c}$$

$$\hat{P}_{t|t}^l = \sum_{i=1}^N \tilde{q}_t^{(i)} \left(P_{t|t}^{(i)} + \left(\hat{x}_{t|t}^{l,(i)} - \hat{x}_{t|t}^l \right) \left(\hat{x}_{t|t}^{l,(i)} - \hat{x}_{t|t}^l \right)^T \right), \tag{9d}$$

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

3. Analysis

In this section, several properties important in the practical application of the marginalized particle filter are analyzed. First, the variance reduction inherent using the Rao-Blackwellization idea is explained. Second, the computational burden of MPF is analyzed in detail. Finally, quantization effects in the measurement relation are described.

Alg. 1 Marginalized Particle Filter (MPF)

1. Initialization: For $i = 1, \ldots, N$, 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. Particle filter measurement update: For $i = 1, \ldots, N$, evaluate the importance weights
$$q_t^{(i)} = p(y_t|\mathbb{X}_t^{n,(i)}, \mathbb{Y}_{t-1}), \tag{5}$$
and normalize $\tilde{q}_t^{(i)} = \frac{q_t^{(i)}}{\sum_{j=1}^N q_t^{(j)}}$.
3. Resample N particles with replacement,
$$\Pr(x_{t|t}^{n,(i)} = x_{t|t-1}^{n,(j)}) = \tilde{q}_t^{(j)}.$$
4. Particle filter time update and Kalman filter:
   (a) Kalman filter measurement update:
$$\hat{x}_{t|t}^l = \hat{x}_{t|t-1}^l + K_t(y_t - h_t - C_t \hat{x}_{t|t-1}^l), \tag{6a}$$
$$P_{t|t} = P_{t|t-1} - K_t M_t K_t^T, \tag{6b}$$
$$M_t = C_t P_{t|t-1} C_t^T + R_t, \tag{6c}$$
$$K_t = P_{t|t-1} C_t^T M_t^{-1}. \tag{6d}$$
   (b) Particle filter time update (prediction): For $i = 1, \ldots, N$, predict new particles,
$$x_{t+1|t}^{n,(i)} \sim p(x_{t+1|t}^n|\mathbb{X}_t^{n,(i)}, \mathbb{Y}_t).$$
   (c) Kalman filter time update:
$$\hat{x}_{t+1|t}^l = \bar{A}_t^l \hat{x}_{t|t}^l + G_t^l (Q_t^{ln})^T (G_t^n Q_t^n)^{-1} z_t + f_t^l + L_t(z_t - A_t^n \hat{x}_{t|t}^l), \tag{7a}$$
$$P_{t+1|t} = \bar{A}_t^l P_{t|t} (\bar{A}_t^l)^T + G_t^l \bar{Q}_t^l (G_t^l)^T - L_t N_t L_t^T, \tag{7b}$$
$$N_t = A_t^n P_{t|t} (A_t^n)^T + G_t^n Q_t^n (G_t^n)^T, \tag{7c}$$
$$L_t = \bar{A}_t^l P_{t|t} (A_t^n)^T N_t^{-1}, \tag{7d}$$
   where
$$z_t = x_{t+1}^n - f_t^n, \tag{8a}$$
$$\bar{A}_t^l = A_t^l - G_t^l (Q_t^{ln})^T (G_t^n Q_t^n)^{-1} A_t^n, \tag{8b}$$
$$\bar{Q}_t^l = Q_t^l - (Q_t^{ln})^T (Q_t^n)^{-1} Q_t^{ln}. \tag{8c}$$
5. Set $t := t + 1$ and iterate from step 2.

Variance Reduction

The variance of a function or estimator $g(U, V)$, depending on two random variables, U and V can be written as

$$\mathsf{Var}\left(g(U,V)\right) = \mathsf{Var}\left(\mathsf{E}\left(g(U,V)|V\right)\right) + \mathsf{E}\left(\mathsf{Var}\left(g(U,V)|V\right)\right), \tag{10}$$

where $\mathsf{E}(\cdot)$ is the expected value. Hence, in principle, the conditional inequality

$$\mathsf{Var}\left(\mathsf{E}\left(g(x_t^l, \mathbb{X}_t^n)|\mathbb{X}_t^n\right)\right) \leq \mathsf{Var}\left(g(x_t^l, \mathbb{X}_t^n)\right), \qquad (11)$$

can be employed. This is sometimes referred to as Rao-Blackwellization, see e.g., [38]. This is the basic part that improves performance using the marginalization idea. In the MPF setup, U and V are represented by the linear and nonlinear states.

Computational Complexity

In discussing the use of the MPF it is sometimes better to partition the state vector into one part that is estimated using the particle filter, $x_t^p \in \mathbb{R}^p$, and one part that is estimated using the Kalman filter, $x_t^k \in \mathbb{R}^k$. Obviously all the nonlinear states, x_t^n, are included in x_t^p. However, we could also choose to include some of the linear states there as well. Under the assumption of linear dynamics, this notation allows us to write (3) according to

$$x_{t+1}^p = A_t^p x_t^p + A_t^k x_t^k + w_t^p, \qquad w_t^p \in \mathcal{N}(0, Q_t^p), \quad (12a)$$
$$x_{t+1}^k = F_t^p x_t^p + F_t^k x_t^k + w_t^k, \qquad w_t^k \in \mathcal{N}(0, Q_t^k), \quad (12b)$$
$$y_t = h_t(x_t^p) + C_t x_t^k + e_t, \qquad e_t \in \mathcal{N}(0, R_t). \quad (12c)$$

First, the case $C_t = 0$ is discussed. For instance, the first instruction $P_{t|t}(A_t^k)^T$ corresponds to multiplying $P_{t|t} \in \mathbb{R}^{k\times k}$ with $(A_t^k)^T \in \mathbb{R}^{k\times p}$, which requires pk^2 multiplications and $(k-1)kp$ additions [21]. The total *equivalent flop (EF)*[1] complexity is derived in [32],

$$\mathcal{C}(p,k,N) \approx 4pk^2 + 8kp^2 + \frac{4}{3}p^3 + 5k^3 - 5kp + 2p^2 + (6kp + 4p^2 + 2k^2 + p - k + pc_3 + c_1 + c_2)N. \quad (13)$$

Here, the coefficient c_1 has been used for the calculation of the Gaussian likelihood, c_2 for the resampling and c_3 for the random number complexity. Note that, when $C_t = 0$ the same covariance matrix is used for all Kalman filters, which significantly reduce the computational complexity.

By requiring $\mathcal{C}(p+k, 0, N_{\text{PF}}) = \mathcal{C}(p, k, N(k))$, where N_{PF} corresponds to the number of particles used in the standard particle filter we can solve for $N(k)$. This gives the number of particles, $N(k)$, that can be used by the MPF in order to obtain the same computational complexity as if the standard particle filter had been used for all states. In Figure 2 the ratio $N(k)/N_{\text{PF}}$ is plotted for systems with $m = 3, \ldots, 9$ states. Hence, using Figure 2 it is possible to directly find out how much there is to gain in using the MPF from a computational complexity point of view. The figure also shows that the computational complexity is always reduced when the MPF can be used instead of the standard particle filter. Furthermore, as

[1] The EF complexity for an operation is defined as the number of flops that result in the same computational time as the operation.

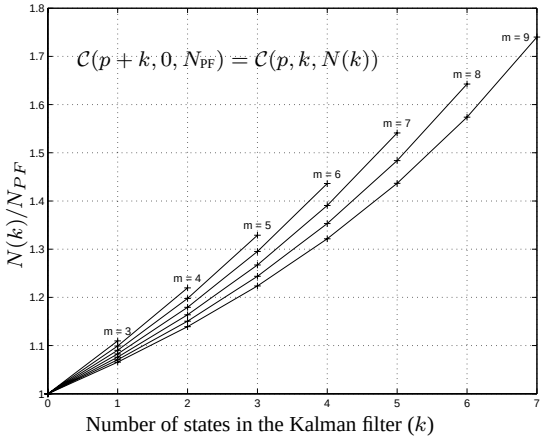


Figure 2. Ratio $N(k)/N_{\text{PF}}$ for systems with $m = 3, \ldots, 9$ states and $C_t = 0$, $n = 2$ is shown. It is apparent that the MPF can use more particles for a given computational complexity, when compared to the standard PF.

previously mentioned, the quality of the estimates will improve or remain the same when the MPF is used [14].

Second, if $C_t \neq 0$, the Riccati recursions have to be evaluated separately for each particle. This results in a significantly increased computational complexity. Hence, different covariance matrices have to be used for each Kalman filter, implying that (13) has to be modified. Approximately the complexity is given by [32],

$$\mathcal{C}(p,k,N) \approx (6kp + 4p^2 + 2k^2 + p - k + pc_3 + c_1 + c_2 + 4pk^2 + 8kp^2 + \frac{4}{3}p^3 + 5k^3 - 5kp + 2p^2 + k^3)N. \qquad (14)$$

In Figure 3 the ratio $N(k)/N_{PF}$ is plotted for systems with $m = 3, \ldots, 9$ states. For systems with few states the MPF is more efficient than the standard particle filter. However, for systems with more states, where most of

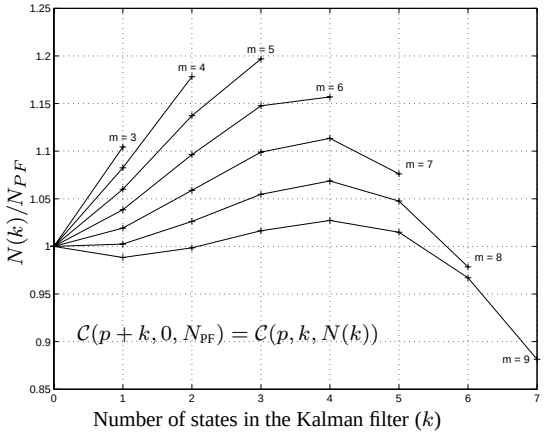


Figure 3. Ratio $N(k)/N_{\text{PF}}$ for systems with $m = 3, \ldots, 9$ states and $C_t \neq 0$, $n = 2$ is shown. For systems with high state dimension and many marginalized states the standard PF can use more particles than the MPF.

the states are marginalized the standard particle filter becomes more efficient than the MPF. This is due to the Riccati recursions mentioned above.

Quantization Effects

When implementing filters or estimators in hardware, the calculations can usually be performed with sufficient precision. However, the sensor or measurement relation may not always have sufficient resolution. This is referred as measurement quantization, and is a common problem in for instance telecommunication, where the channel bandwidth is limited. To be able to use limited communication resources, severe quantization may be needed. Also for large sensor networks applications, many very simple and cheap sensors with large quantization effects are used. Furthermore, many sensors or signal processing devices are naturally quantized, for instance range measurements in a pulsed radar or pixelized information from a vision system.

Here we will discuss quantization using a multi-level uniform quantization. Consider the problem of estimating x from the quantized measurements $y = \mathcal{Q}_m(x+e)$. The uniform quantization discussed here is implemented as the *midriser* quantizer, as described in [34]. If not saturated it is given as

$$\mathcal{Q}_m(z) = \Delta\left\lfloor \frac{z}{\Delta} \right\rfloor + \frac{\Delta}{2}. \tag{15}$$

Here, $\mathcal{Q}_m(\cdot)$ denotes the nonlinear quantization mapping with m levels, all with equal quantization height Δ. The $\lfloor\cdot\rfloor$ operator rounds downwards to the nearest integer. To keep a unified notation with the sign quantization $\mathcal{Q}_1(z) = \operatorname{sign}(z)$, the midriser convention will be used, so $y \in \{-m\Delta + \frac{\Delta}{2}, \ldots, (m-1)\Delta + \frac{\Delta}{2}\}$, with $\Delta = 2^{-b}$, using b bits, $2m = 2^b$ levels and $2^b - 1$ thresholds. The sign quantization corresponds to $b = 1$, $m = 1$ and $\Delta = 2$ in this notation.

In [29], this static problem is analyzed using the *maximum likelihood* (ML) estimator. The performance is also investigated using the Fisher information or *Cramér-Rao lower bound* (CRLB). The resulting likelihood function can also be used in the particle filter, allowing for a statistically correct treatment of measurement quantization effects in dynamic systems. If the model is in accordance with the requirement of the MPF algorithm, it is possible to handle the nonlinearity introduced by the quantization in the measurement equation in the MPF. In [29] different quantizers are studied. Below, only the simplest sign quantizer, $y_t = \mathcal{Q}_1(x_t + e_t), e_t \in \mathcal{N}(0, \sigma^2)$, is discussed. The probability function for y can be calculated using

$$\begin{aligned} p(y=-1|x) &= \operatorname{Prob}(x+e<0) = \operatorname{Prob}(e<-x) \\ &= \int_{-\infty}^{-x/\sigma} \frac{1}{\sqrt{2\pi}} \exp^{-\frac{t^2}{2}}\, dt \triangleq \Phi(-x/\sigma). \end{aligned} \tag{16}$$

Similarly,

$$p(y=+1|x) = \operatorname{Prob}(x+e\geq 0) = 1 - \Phi(-x/\sigma). \tag{17}$$

Hence, the discrete likelihood needed in the PF/MPF, in (5), can be written as

$$\begin{aligned} p(y|x) = \Phi(-x/\sigma)\,\delta(y+1) \\ + (1-\Phi(-x/\sigma))\,\delta(y-1), \end{aligned} \tag{18}$$

where

$$\delta(i) = \begin{cases} 1, & i=0, \\ 0, & i\neq 0. \end{cases} \tag{19}$$

The calculated likelihood can be used in the PF/MPF to incorporate the quantization effect in a statistically correct way. Similar for multi-level quantization. See [17] for an application of a particle smoother to handle the quantization problem.

Example 1 (Filtering – sign quantizer) *Consider the following scalar system with a sign quantizer*

$$\begin{aligned} x_{t+1} &= F_t x_t + w_t, \qquad x_0 = 0, \\ y_t &= \mathcal{Q}_1(x_t + e_t), \end{aligned}$$

where

$$F_t = 0.95, \quad \mathsf{Var}(w_t) = 0.10^2, \quad \mathsf{Var}(e_t) = 0.58^2.$$

In Figure 4 the RMSE for the KF and the PF are presented using 200 *Monte Carlo simulations. The measurement noise in the KF was adjusted in the filter to handle the quantization by adding an extra variance of* $\Delta^2/12$. *The PF used the correct sign quantized likelihood using* 1000 *particles. The theoretical Cramér-Rao lower bound is also given in Figure 4. For details, see [29].*

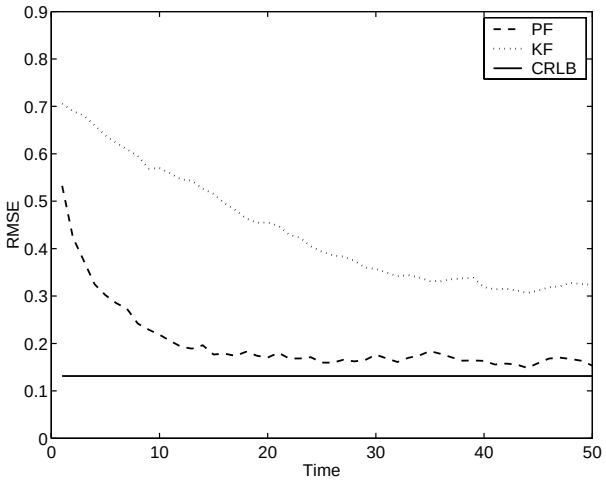


Figure 4. RMSE for the PF and KF for a linear Gaussian system with a sign quantizer in the measurement relation, compared to the Cramér-Rao lower bound .

Note that for the example presented only one state was used, hence no marginalization was applied. If the problem is formulated with linear-Gaussian dynamics and quantization in the measurement, these nonlinear states can be handled by the PF and the rest by the KF in the MPF framework.

4. Introducing the Applications

As discussed in the previous section, the different estimation methods handle nonlinearities in different ways. In the applications studied in this paper a framework consisting of linear-Gaussian system dynamics and nonlinear measurements is considered. Basically, two different areas are studied: GPS-free *positioning*, where the aim is to estimate the own platform's position and *target tracking*, where the state of an unknown, observed target is estimated from measurements. These applications also represent typical examples where sensor fusion techniques are important. The MPF provides an efficient way to incorporate both linear and nonlinear measurement relations. Both results from simulated data and experimental data are presented. More precisely, the studied applications are:

Positioning applications:

- *Underwater terrain-aided positioning:* Depth information from a geographical information system (GIS) database is used together with sonar depth measurements to improve positioning. A demonstrator system has been developed in co-operation with Saab Underwater Systems.
- *Aircraft terrain-aided positioning:* A height GIS database is used together with radar height measurements to improve the position, compared to only inertial navigation system (INS) measurements. A demonstrator system has been developed by Saab Aerospace.
- *Automotive map-aided positioning:* Utilizing wheel speed sensors from the ABS and information from a street-map database, car positioning independent of GPS is possible. This is available as a commercial product from NIRA Dynamics.

Target tracking applications:

- *Automotive target tracking:* Intelligent automotive systems require information about the host vehicle and its surroundings (lane geometry and the position of surrounding vehicles). Using vision and radar measurements, the corresponding estimation problem is addressed. A demonstrator vehicle has been developed in co-operation with Volvo Car Corporation.
- *Bearings-only target tracking:* When passive sensors, such as an infrared (IR) sensor are used, we can only measure the direction, bearing, to the unknown target. However, by appropriate maneuvering, the range and range rate can be estimated. This is studied in an air-to-sea application, i.e., an aircraft tracking a ship.
- *Radar target tracking:* A radar sensor measures at least range and direction (azimuth, elevation) to the target. In this particular application the computational aspects of the MPF are studied in detail.

The dynamic models employed in the applications all have a linear motion model and a nonlinear measurement model. By partitioning the state vector x_t into two parts, one for the linear state variables x_t^l and one for the nonlinear state variables x_t^n the model fits the MPF framework perfectly. For example, consider Cartesian position coordinates (X, Y, Z) and introduce the state vector $x_t = \begin{pmatrix} X_t & Y_t & Z_t & \dot{X}_t & \dot{Y}_t & \dot{Z}_t \end{pmatrix}^T$, with position and velocity states. In target tracking the relative distance between the target and the observation platform is often used as state. Furthermore, the first order derivatives of this distance, relative velocity, are included in the state vector. The resulting motion model is given by

$$x_{t+1} = F_t x_t + G_t w_t, \tag{20a}$$

where

$$F_t = \begin{pmatrix} I_3 & TI_3 \\ O_3 & I_3 \end{pmatrix}, G_t = \begin{pmatrix} \frac{T^2}{2} I_3 \\ TI_3 \end{pmatrix}, \tag{20b}$$

Here, I_3 denotes the 3×3 unity matrix and O_3 denotes the 3×3 null matrix. The measurement relation is in the sequel treated as a nonlinear relation of the state, subject to additive measurement noise,

$$y_t = h(x_t^n) + e_t. \tag{20c}$$

It can for instance represent range and bearing measurement from a radar, height or depth measurements for terrain navigation applications. In all these situations it is a function of the position states. For the example above, $x_t^n = \begin{pmatrix} X_t & Y_t & Z_t \end{pmatrix}^T$ and $x_t^l = \begin{pmatrix} \dot{X}_t & \dot{Y}_t & \dot{Z}_t \end{pmatrix}^T$. Another common state variable is the heading or course.

For a more thorough discussion regarding models for positioning, navigation, and tracking applications within the present setting we refer to [23]. Interesting to note is also that common phenomena such as bias or scale-factor estimation can often be introduced in the linear-Gaussian sub-system. Hence, the MPF provides an efficient way to handle such problems.

5. Positioning Applications

This section is concerned with position estimation, where information from geographical information systems is used together with different *distance measuring equipment* (DME). First, an underwater positioning method based on sonar depth measurements is presented. Second, the same idea is employed to solve the aircraft positioning problem using height measurements from a radar altimeter. Finally, the automotive positioning problem is briefly presented.

Underwater Terrain-aided Positioning

In this section we describe an *underwater* positioning method based DME information from sonar depth read-

ings and a comparison with a depth database to find the position of the host vessel. It is based on the preliminary studies in [27, 31], together with [28].

Using a sonar sensor and a *differential GPS* (DGPS), an underwater depth map was constructed, illustrated in Figure 5, together with the platform at depth $d_t = 0$ and with sonar depth measurements r_t. After the data

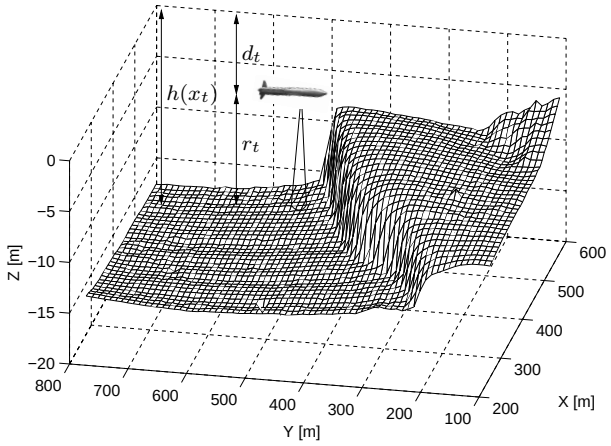


Figure 5. Underwater positioning using sonar depth measurements and a true terrain database. The sonar depth is d_t, and the sonar indicates the relative range to the sea floor r_t. The database gives $h(x_t)$.

for map generation was collected, an independent test run in the map region was performed, in order to collect measurements to test the PF/MPF map-aided positioning system. In [27] a coordinated turn model extended with bias terms was used. In order to apply the MPF a Taylor expansion was calculated, enabling for a model approximately in the correct form. The estimation performance reported for the MPF was similar to the PF, but to a much smaller computational burden. In order to fit the linear-Gaussian dynamics framework, we will only consider the model from [28]. The number of particles used initially was $N = 50000$, but quickly reduced to $N = 10000$, when the particle cloud had most of its particles clustered around the true position. The result is presented in Figure 6, where the parametric CRLB is calculated using an extended Kalman filter, evaluated around the true trajectory.

Aircraft Terrain-Aided Positioning

The Swedish fighter aircraft Gripen is equipped with an accurate radar altimeter as DME sensor and a terrain elevation database, similar to the discussion in the previous section. These measurements are used together with an *inertial navigation system* (INS) in order to solve the aircraft positioning problem. This problem has previously been studied, see e.g., [1, 6, 42]. The overall structure of the model used in this application is in the

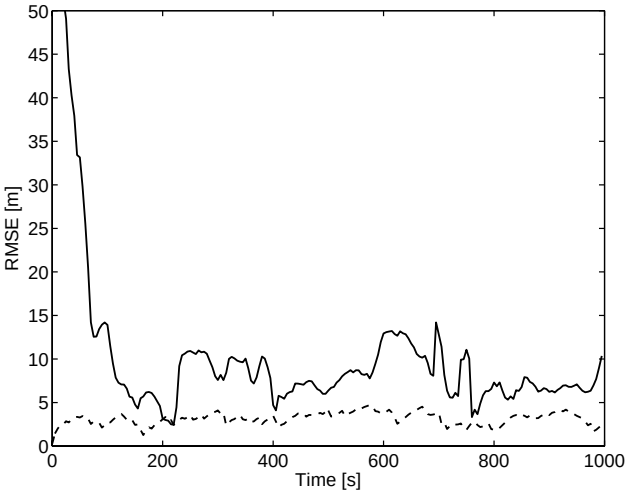


Figure 6. The position RMSE from the PF (solid line) using the experimental test data together with the parametric CRLB (dashed line) as the EKF solution around the true trajectory. The nominal speed is between 0.9–1.5 m/s. Note that only one experimental test run was available for the RMSE calculation.

form (12), with the following measurement equation,

$$y_t = h\left(\begin{pmatrix} X_t \\ Y_t \end{pmatrix} + x_t^n\right) + e_t \tag{21}$$

where X_t and Y_t denotes the error in latitude and longitude respectively. The feasibility study performed used a sub-model with 9 states. This sub-model contains all ingredients of the total system and the principle is scalable to the full model with 27 states. For details regarding the model we refer to [35] and the references therein.

The measurement equation (21) is highly nonlinear, due to the use of the terrain elevation database. This implies that the EKF cannot be used. Furthermore, the high dimension of the problem prevents the use of the particle filter. However, the model structure fits perfectly into the marginalized particle filter framework. This approach has been evaluated using authentic flight data with promising results, see Figure 7 where we provide a plot of the error in horizontal position for a different number of particles. From this plot it is clear that the main difference in performance is in the transient phase, in the stationary phase the performance is less sensitive to the number of particles used. Hence, the idea of using more particles in the transient phase suggests itself. This idea was used, for the same reason, in the previous section as well. For a more detailed account on these experiments, see [19, 35, 40].

Automotive Map-Aided Positioning

The idea is to use the information available from the wheel speed sensors together with digital map information to estimate the position of the car, without the need

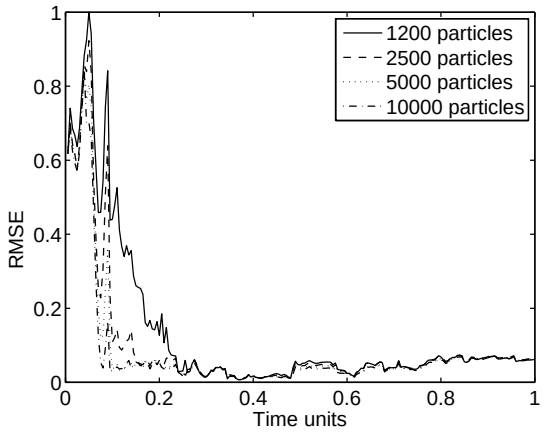


Figure 7. Horizontal position error as a function of time units for different numbers of particles. Note that the scale has been normalized for confidentiality reasons.

for GPS information. The resulting problem is nonlinear and fits into the framework provided by the particle filter and the marginalized particle filter. For further details on this approach, see e.g., [18, 24, 43].

6. Target Tracking Applications

In this section three target tracking applications are studied. First, an automotive target tracking problem is discussed. This is followed by a bearings-only estimation problem. Finally, a radar target tracking application highlight different computational aspects of the marginalized particle filter.

Automotive Target Tracking

This application deals with the problem of estimating the vehicle surroundings (lane geometry and the position of other vehicles), which is required by advanced automotive safety systems, such as adaptive cruise control, collision avoidance and lane guidance. For a thorough treatment of this application, see [16].

The main difference between tracking in automotive applications and tracking in other applications, such as air traffic control or naval tracking, is that in automotive tracking it can be assumed that the motion of the tracked objects, with a certain probability, is constrained to the road. In order to be able to use and benefit from this fact we make use of a curved coordinate system which is attached to and follows the road [15]. The measurements are provided by a vision system and a radar system. The vision system provides measurements of the road curvature, the yaw angle and the distance to the right and left lane markings. Furthermore, the radar provides range measurements to the surrounding vehicles. The final model, thoroughly derived in [15, 16] is in the form (12), which opens up for using the marginalized particle filter. The nonlinear part of the measurement equation for a given target i is

$$y_t = h(X_t^i, Y_t^i) + e_t, \tag{22}$$

where $h(\cdot)$ described the geometric transformation from a curved, road-aligned coordinate system to a Cartesian coordinate system, in which the measurements are registered. For details, see [16]. In evaluating the estimation performance we study the estimate of the road curvature. It is crucial to several automotive applications, such as adaptive cruise control systems, collision warning or any system that relies on assigning leading vehicles to the correct lane. For a leading vehicle 100 m in front of the host vehicle, a small curvature error of, say $0.5 \cdot 10^{-3}$ m^{-1} will result in an error of 2.5 m in the lateral direction [16]. This is enough to assign the leading vehicle to the wrong lane.

The data set used was collected in the northern parts of Sweden during winter. This implies that the visibility is low, which in turn implies that the measurements from the vision system definitely have to be supported by the radar measurements to obtain a solid overall estimate. In Figure 8 we provide the absolute curvature estimation error using the MPF and the EKF. Furthermore, the raw vision measurement of the curvature is also included. From Figure 8 it is clear that both filters improve the quality of the curvature estimate substantially. However, the performance of the MPF is only slightly better than the EKF. Hence, in this particular setting is might be hard to motivate using the MPF, due to its higher computational complexity. If we were to use more advanced measurement equations, such as those based on map information the MPF might be the only option.

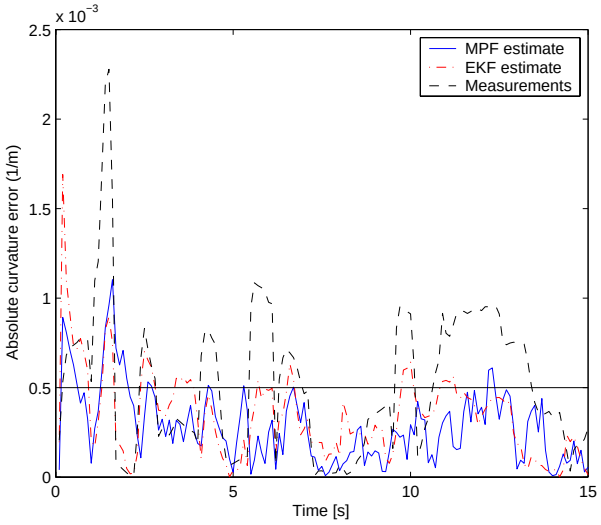


Figure 8. The absolute curvature error. Here, we have indicated the level $0.5 \cdot 10^{-3}$ m^{-1}, which is used as a motivating example in the text. For errors above this level, leading vehicles at a distance of 100 m are likely to be assigned to the wrong lane.

Bearings-Only Target Tracking

In this section, an air-to-sea bearings-only application is studied. Assume that the ship (target) and the aircraft (tracking platform) are described by the same type of linear dynamics as in Section 4 for the position and the velocity, save for the fact the relative quantities have been used as states. For bearings-only applications the measurement relation for the azimuth angle, φ, and elevation angle, θ, is given as

$$y_t = h(x_t) + e_t = \begin{pmatrix} \varphi_t \\ \theta_t \end{pmatrix} + e_t = \begin{pmatrix} \arctan\left(Y_t/X_t\right) \\ \arctan\left(\frac{-Z_t}{\sqrt{X_t^2+Y_t^2}}\right) \end{pmatrix} + e_t, \quad (23)$$

where X_t, Y_t, and Z_t denote the Cartesian components of the relative position.

In a simulation study the range estimation problem using an infrared (IR) sensor is considered. The PF and MPF are compared to a bank of EKFs, using the *range parameterized extended Kalman filter* (RPEKF) method, [5, 36]. The relative distance and the aircraft trajectory are illustrated in Figure 9. The target model used in the simulations assumes a small constant velocity. The terrain database has a resolution of 50 m. In Figure 9 the scenario is presented together with the marginal position densities in each direction, $p(X)$ and $p(Y)$, for time $t = 1$ s, using terrain constraints. In Figure 10 the position RMSE is presented for the PF and the MPF with and without the map constraints, and for the RPEKF. Obviously the incorporation of constraints improves the performance. The different particle filters have basically the same performance for this scenario. For details regarding the simulation study, the reader is referred to

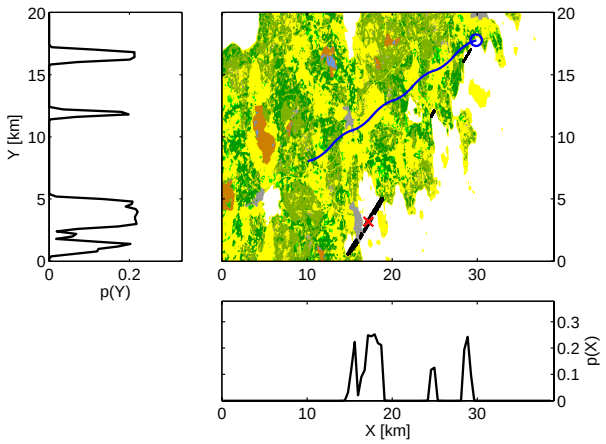


Figure 9. The position for the aircraft (o) and the target ship (X) together with the marginalized position pdf using the particle filter with terrain induced constraints at $t = 1$ s. The particle cloud and the future trajectory of the aircraft are also shown.

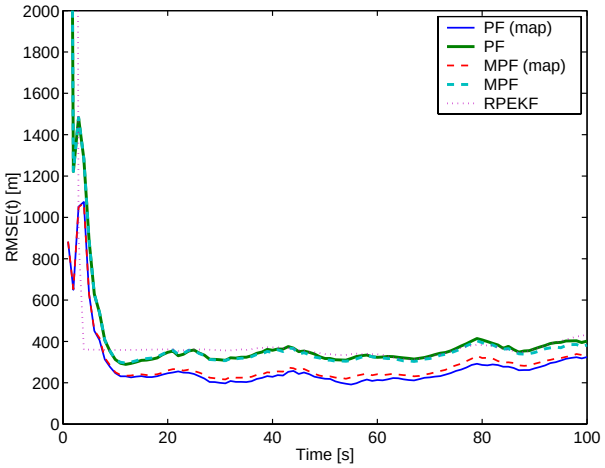


Figure 10. Position RMSE(t) for air-to-sea passive ranging using 100 Monte Carlo simulations.

[30], where similar bearings-only applications are described in detail, both for simulated data and for experimental data. For instance, experimental data from a passive sonar system on a torpedo is used for bearings-only tracking.

Radar Target Tracking

In this section, the radar target tracking application from [32] is highlighted. The general method for analyzing the computational complexity presented in [32] and briefly reviewed in Section 3, is illustrated using a common target tracking model. The problem of estimating the position and velocity of an aircraft is studied using the dynamics from Section 4, and the following measurement equation, which gives the range and azimuth from the radar system,

$$y_t = h(x_t) + e_t = \begin{pmatrix} \sqrt{X_t^2 + Y_t^2} \\ \arctan\left(Y_t/X_t\right) \end{pmatrix} + e_t, \quad (24)$$

where $\mathsf{Cov}(w) = \mathsf{diag}(1\ \ 1\ \ 1\ \ 1\ \ 0.01\ \ 0.01), \mathsf{Cov}(e) = \mathsf{diag}(100\ \ \ 10^{-6})$ and the state vector is $x_t = (X\ \ Y\ \ \dot{X}\ \ \dot{Y}\ \ \ddot{X}\ \ \ddot{Y})^T$, i.e., position, velocity and acceleration.

The model has two nonlinear state variables and four linear state variables. Two cases are now studied, the full PF, where all states are estimated using the PF and the completely marginalized PF, where all linear states are marginalized out and estimated using the KF. If we want to compare the two approaches under the assumption that they use the same computational resources, i.e., $\mathcal{C}(6, 0, N_{PF}) = \mathcal{C}(2, 4, N_{MPF})$, we obtain

$$N_{\text{PF}} = \underbrace{\left(1 - \frac{4c_3 + 56}{c_1 + c_2 + 6c_3 + 150}\right)}_{<1} N_{\text{MPF}}. \quad (25)$$

From (25) it is clear that for a given computational complexity more particles can be used in the MPF than in

the standard PF. This is verified experimentally in [32].

Using a constant computational complexity the number of particles that can be used is computed. The study is performed by first running the full PF and measure the time consumed by the algorithm. An Monte Carlo simulation, using $N = 2000$ particles, is performed in order to obtain a stable estimate of the time consumed by the algorithm. In Table 6 the number of particles (N), the total RMSE from 100 Monte Carlo simulations, and the simulation times are shown for the different marginalization cases. From Table 6 it is clear that the different

Table 1. Results from the simulation, using a constant computational complexity. If a certain state variable is estimated using the PF this is indicated with a P and if the KF is used this is indicated with a K.

	PPPPPP	**PP**KKPP	**PP**PPKK	**PP**KKKK
N	2000	2029	1974	2574
RMSE pos	7.10	5.81	5.76	5.60
RMSE vel	3.62	3.27	3.28	3.21
RMSE acc	0.52	0.47	0.45	0.44
Time	0.59	0.58	0.57	0.60

MPFs can use more particles for a given time, which is in perfect correspondence with the theoretical result given in (25).

Let us now discuss what happens if a constant velocity RMSE is used. First the velocity RMSE for the full PF is found using an Monte Carlo simulation. This value is then used as a target function in the search for the number of particles needed by the different MPFs. Table 2 clearly indicates that the MPF can obtain the

Table 2. Results using a constant velocity RMSE.

	PPPPPP	**PP**KKPP	**PP**PPKK	**PP**KKKK
N	2393	864	943	264
RMSE pos	7.07	6.98	7.12	7.27
RMSE vel	3.58	3.60	3.65	3.61
RMSE acc	0.50	0.51	0.49	0.48
Time	0.73	0.26	0.28	0.10

same RMSE using fewer particles. The result is that using full marginalization only requires 14% of the computational resources as compared to the standard PF in this example.

7. Concluding Discussion

In this paper several positioning and target tracking applications are solved using the marginalized particle filter. In the framework employed the dynamic motion models are linear, subject to Gaussian noise and the measurement models are nonlinear. This important special case of the general MPF allows for an efficient implementation.

The computational complexity of the MPF algorithm is thoroughly analyzed for a radar application, but because of the similarities in the studied models in the applications, these results are approximately valid for them as well. The radar application also illustrates another important property of the MPF, namely that the quality of the estimates is enhanced compared to the standard particle filter.

Another unifying feature among the various applications is that they all use measurements from various different sources, implying that we are indeed solving the *sensor fusion* problem using the MPF. Terrain-aided positioning problems are quite hard to handle using methods based on linearization, due to the fact that it is very hard to obtain a good linear description of the map database, used to form the measurement equations. Hence, the MPF is a very powerful tool for these applications. We saw that the computational complexity can be reduced substantially by decreasing the number of particles when the stationary phase is reached. This is a common idea, employed in all the applications, since more computational resources should be used in the transient phase.

Common for the measurement relation is that nonlinearities and non-Gaussian noise is handled in a statistically optimal way, by the particle filter. Particularly, if the measurement relation is subject to severe quantization this is important to handle. Quantization arises naturally in many applications, but typically in sensor networks where sensor fusion is applied based on information from a large number of very cheap sensors, this can be a major issue.

Acknowledgment

This work was supported by VINNOVA's Center of Excellence ISIS (Information Systems for Industrial Control and Supervision), and by the Swedish Research Council (VR).

References

[1] M. Ahlström and M. Calais. Bayesian terrain navigation with monte carlo method. Master's Thesis No LiTH-ISY-EX-3051, Department of Electrical Engineering, Linköping University, Sweden, Sept. 2000.

[2] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Information and system science series. Prentice Hall, Englewood Cliffs, NJ, USA, 1979.

[3] 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.

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

[5] S. Arulampalam and B. Ristic. Comparison of the particle filter with range-parameterized and modified polar EKFs for angle-only tracking. In *Proceedings of SPIE, Signal and Data Processing of Small Target*, pages 288–299, Orlando, FL, USA, 2000.

[6] N. Bergman. *Recursive Bayesian Estimation: Navigation and Tracking*

Applications. Dissertations No 579, Linköping Studies in Science and Technology, SE-581 83 Linköping, Sweden, May 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] M. J. Daly, J. P. Reilly, and M. R. Morelande. Rao-Blackwellised particle filtering for blind system identification. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, PA, USA, Mar. 2005.

[11] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, New York, USA, 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] A. Eidehall. *An Automotive Lane Guidance System*. Licentiate Thesis No 1122, Department of Electrical Engineering, Linköping University, Sweden, 2004.

[16] A. Eidehall, T. B. Schön, and F. Gustafsson. The marginalized particle filter for automotive tracking applications. In *Proceedings of the IEEE Intelligent Vehicle Symposium*, pages 369–374, Las Vegas, USA, June 2005.

[17] W. Fong and S. Godsill. Monte Carlo smoothing for non-linearly distorted signals. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 3997–4000, Salt Lake City, USA, May 2001.

[18] U. Forssell, P. Hall, S. Ahlqvist, and F. Gustafsson. Novel map-aided positioning system. In *Proceedings of FISITA*, Helsinki, Finland, 2002.

[19] P. Frykman. Applied particle filters in integrated aircraft navigation. Master's Thesis No LiTH-ISY-EX-3406, Department of Electrical Engineering, Linköping University, Sweden, Apr. 2003.

[20] A. Giremus and J. Y. Tourneret. Joint detection/estimation of multipath effects for the global positioning system. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 17–20, Philadelphia, PA, USA, Mar. 2005.

[21] G. H. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, third edition, 1996.

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

[23] 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.

[24] P. Hall. A Bayesian approach to map-aided vehicle positioning. Master's Thesis No LiTH-ISY-EX-3102, Department of Electrical Engineering, Linköping University, Sweden, Jan. 2000.

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

[26] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 82:35–45, 1960.

[27] R. Karlsson and F. Gustafsson. Particle filter for underwater navigation. In *Proceedings of the Statistical Signal Processing Workshop*, pages 509–512, St. Louis, MO, USA, Sept. 2003.

[28] R. Karlsson and F. Gustafsson. Bayesian surface and underwater navigation. Technical Report LiTH-ISY-R-2674, Department of Electrical Engineering, Jan. 2005. Submitted to IEEE Transactions on signal processing.

[29] R. Karlsson and F. Gustafsson. Particle filtering for quantized sensor information. In *Proceedings of the 13th European Signal Processing Conference*, Antalya, Turkey, Sept. 2005.

[30] R. Karlsson and F. Gustafsson. Recursive Bayesian estimation – bearings-only applications. *IEE Proceedings on Radar, Sonar, and Navigation. Special issue on target tracking: Algorithms and Applications*, 152(5):305–313, Oct. 2005.

[31] R. Karlsson, F. Gustafsson, and T. Karlsson. Particle filter and Cramér-Rao lower bound for underwater navigation. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong, Apr. 2003.

[32] R. Karlsson, T. Schön, and F. Gustafsson. Complexity analysis of the marginalized particle filter. *IEEE Transactions on Signal Processing*, 53(11):4408–4411, Nov. 2005.

[33] 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*, Cambridge, UK, Sept. 2003.

[34] S. P. Lipshitz, R. A. Wannamker, and J. Vanderkooy. Quantization and dither: A theoretical survey. *Journal of Audio Engineering Society*, 40(5):355–375, May 1992.

[35] P.-J. Nordlund. *Sequential Monte Carlo Filters and Integrated Navigation*. Licentiate Thesis No 945, Department of Electrical Engineering, Linköping University, Sweden, 2002.

[36] N. Peach. Bearings-only tracking using a set of range-parameterized extended Kalman filters. In *IEE Proceedings of Control Theory and Applications*, volume 142, pages 73–80, Jan. 1995.

[37] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: particle filters for tracking applications*. Artech House, London, UK, 2004.

[38] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer texts in statistics. Springer, New York, 1999.

[39] 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, Sept. 2003. Invited paper.

[40] T. Schön, F. Gustafsson, and P.-J. Nordlund. Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on Signal Processing*, 53(7):2279–2289, July 2005.

[41] H. W. Sorenson and D. L. Alspach. Recursive Bayesian estimation using Gaussian sum. *Automatica*, 7:465–479, 1971.

[42] M. Svensson. Implementation and validation of Bayesian terrain navigation. Master's Thesis No LiTH-ISY-EX-2039, Department of Electrical Engineering, Linköping University, Sweden, Apr. 1999.

[43] N. Svenzén. Real time implementation of map aided positioning using a Bayesian approach. Master's Thesis No LiTH-ISY-EX-3297-2002, Department of Electrical Engineering, Linköping University, Sweden, Dec. 2002.

[44] X. Wang, R. 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.