FAST PARTICLE FILTERS FOR MULTI-RATE SENSORS

Thomas B. Schön, David Törnqvist and Fredrik Gustafsson

Division of Automatic Control/Department of Electrical Engineering, Linköpings universitet
SE–581 83 Linköping, Sweden
phone: + (46) 13 28 13 73, fax: + (46) 13 28 26 22, email: {schon,tornqvist,fredrik}@isy.liu.se
web: www.control.isy.liu.se

ABSTRACT

Computational complexity is a major concern for practical use of the versatile particle filter (PF) for nonlinear filtering applications. Previous work to mitigate the inherent complexity includes the marginalized particle filter (MPF), with the fastSLAM algorithm as one important case. MPF utilizes a linear Gaussian sub-structure in the problem, where the Kalman filter (KF) can be applied. While this reduces the state dimension in the PF, the present work aims at reducing the sampling rate of the PF. The algorithm is derived for a class of models with linear Gaussian dynamic model and two multi-rate sensors, with different sampling rates, one slow with a nonlinear and/or non-Gaussian measurement relation and one fast with a linear Gaussian measurement relation. For this case, the KF is used to process the information from the fast sensor and the information from the slow sensor is processed using the PF. The problem formulation covers the important special case of fast dynamics and one slow sensor, which appears in many navigation and tracking problems.

1. INTRODUCTION

The nonlinear filtering problem deals with estimation of the states x_t, using the information in the measurements up to time t, $y_{1:t} \triangleq \{y_1, \cdots, y_t\}$, for a nonlinear dynamic system,

$$x_{t+1} = f_t(x_t, u_t, w_t), \tag{1a}$$

$$y_t = h_t(x_t, u_t, e_t). \tag{1b}$$

Here, u_t denotes a known input signal, e_t and w_t denote the stochastic measurement and process noise, respectively. Furthermore, the functions f and h contain the dynamic equations for the system and the measurement equations, respectively.

All information about the state that is present in the measurements is contained in the filtering probability density function (PDF) $p(x_t|y_{1:t})$. The particle filter [5, 3] provides an arbitrary good approximation to this PDF. However, it is a well-known fact from applications that the number of particles that are needed to get a good approximation of the true filtering PDF increases rapidly with the state dimension. Therefore, efficient implementations of the PF have to utilize some kind of structure inherent in the problem. An example of such a structure is when there is a linear Gaussian sub-structure available in (1). This is exploited in the marginalized particle filter (also referred to as the Rao-Blackwellized particle filter [3]) by the observation that conditioned on the nonlinear states it is a linear Gaussian system, which can be optimally estimated using the Kalman filter [9]. The resulting algorithm applies a PF to a low-dimensional part of the state vector, where each particle has an associated Kalman filter estimate of the remaining part of the state vector. For more information on the marginalized particle filter we refer to [13, 2].

In the present contribution we consider a different structure, which arises when there are multi-rate sensors (sensors providing measurements with different sampling times) available. It will be shown that if the inherent structure is exploited by the algorithm the quality of the estimates will be better at a lower computational cost, compared to the direct use of the PF. Algorithmically, there are several similarities between the present contribution and the MPF. We will study a class of filtering problems, where the model (1) has the following structure

$$x_{t+1} = A_t x_t + B_t u_t + G_t w_t, \qquad t = 0, 1, \ldots \tag{2a}$$

$$y_{1,t} = C_t x_t + D_t u_t + e_{1,t}, \qquad t = 1, 2, \ldots \tag{2b}$$

$$y_{2,t} = h(x_t, u_t, t) + e_{2,t}, \qquad t = r, 2r, \ldots. \tag{2c}$$

More specifically,

- The dynamic model is linear with Gaussian process noise $w_t \in \mathcal{N}(0, Q_t)$.
- The fast sensor $y_{1,t}$ is linear with Gaussian measurement noise $e_{1,t} \in \mathcal{N}(0, R_{1,t})$.
- The slow sensor provides measurements $y_{2,t}$ a factor r times slower than the fast sensor, and is a nonlinear function of the state and the input signal, with possibly non-Gaussian measurement noise $e_{2,t} \sim p_{e_{2,t}}(\cdot)$.

The typical application area we have in mind is navigation, where the fast sensor delivers measurements of the relative movements (inertial measurement unit or odometer) and the slow sensor provides measurements of absolute reference to given landmarks (camera, bearings-only or range sensors). This includes important specific applications, such as Simultaneous Localization and Mapping (SLAM) [15, 4] and navigation in sensor networks.

We also want to mention that the special case when the fast sensor is missing, given by (2a, 2c) alone, covers all the navigation and tracking applications surveyed in [6]. The presented algorithm then basically propagates the particles available after the measurement equation to a Gaussian mixture, which is resampled at the time of the next measurement.

2. NONLINEAR STATE FILTERING

The solution to the nonlinear state filtering problem is given by the filtering PDF and its conceptual form is given in the subsequent section. Furthermore, a popular approximation to this solution, the particle filter, is briefly introduced in Section 2.2.

2.1 Conceptual Solution

In discussing the conceptual solution to the nonlinear state estimation problem it is convenient to work either with the

PDF's or the distribution functions of the involved stochastic variables. In the present work we will work with PDF's. Hence, the first step is to describe model (1) using PDF's. This can be done according to,

$$x_{t+1} \sim p_t(x_{t+1}|x_t), \tag{3a}$$

$$y_t \sim p_t(y_t|x_t), \tag{3b}$$

where $\sim$ denotes distribution according to. Here, $p_t(x_{t+1}|x_t)$ and $p_t(y_t|x_t)$ are commonly referred to as the transition PDF and likelihood, respectively. Note that the deterministic input signal u_t is seen as a part of the model and the conditioning on the model is implicit in the PDF's. More specifically for the model under consideration in this work (2) we have the following transition PDF $p_t(x_{t+1}|x_t)$ and likelihood $p_t(y_t|x_t)$,

$$p_t(x_{t+1}|x_t) = p_{G_t w_t}(x_{t+1} - A_t x_t - B_t u_t), \tag{4a}$$

$$p_t(y_{1,t}|x_t) = p_{e_{1,t}}(y_t - C_t x_t - D_t u_t), \tag{4b}$$

$$p_t(y_{2,t}|x_t) = p_{e_{2,t}}(y_t - h(x_t, u_t, t)), \tag{4c}$$

The solution to the nonlinear state estimation problem using the model representation in (3) follows rather straightforwardly using Bayes' theorem and the Markov property, (see e.g., [8] for details). The resulting recursions for the filtering and prediction densities are given by

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{\int p(y_t|x_t)p(x_t|y_{1:t-1})dx_t}, \tag{5a}$$

$$p(x_{t+1}|y_{1:t}) = \int p(x_{t+1}|x_t)p(x_t|y_{1:t})dx_t. \tag{5b}$$

The reason for referring to this as a conceptual solution is that the multidimensional integrals in (5) typically do not allow for an analytical solution. However, there are a few special cases allowing for analytical solutions, such as the linear Gaussian case, when the solution is given by the Kalman filter [9]. For the more interesting nonlinear and/or non-Gaussian case we are forced to approximations of some kind and the particle filter provides a very interesting and powerful approximation, especially when the problem has an inherent structure which can be exploited.

2.2 Particle Filter

The main idea underlying the particle filter is to approximate the filtering PDF using a finite number of so called particles $\{x^{(i)}_{t|t}\}_{i=1}^N$ according to

$$\hat{p}_N(x_t|y_{1:t}) = \sum_{i=1}^{N} \gamma_t \delta(x_t - x^{(i)}_{t|t}), \tag{6}$$

where each particle $x^{(i)}_{t|t}$ has an importance weight γ_t associated to it. Note that $\delta(x_t - x^{(i)}_{t|t})$, denotes the delta-Dirac function located at $x^{(i)}_{t|t}$. The importance weight contains information about how probable the corresponding particle is. Due to the delta-Dirac form used in (6), a finite sum is obtained when this approximation is passed through an integral, i.e., the multidimensional integrals are simplified to finite sums, enabling an approximation to (5). All the details of the particle filter were independently discovered by [5, 10, 7]. However, the main ideas, save for the crucial resampling step have been around since the 1940's [11].

For an introduction to and derivation of the the particle filter we refer to e.g., [1, 3]. A rather basic form, used in this contribution, is given in Algorithm 1.

```
Algorithm 1 Particle filter
1. Initialize the particles, $\{x^{(i)}_{1|0}\}_{i=1}^N \sim p(x_0)$.
2. Compute the importance weights $\{\gamma_t^{(i)}\}_{i=1}^N$,
       $\gamma_t^{(i)} = p_t(y_t|x^{(i)}_{t|t-1}), \quad i = 1, \dots, N,$
   and normalize $\tilde{\gamma}_t^{(i)} = \gamma_t^{(i)} / \sum_{j=1}^N \gamma_t^{(j)}$
3. Draw N new particles with replacement, for each $i = 1, \dots, N$,
       $Pr(x^{(i)}_{t|t} = x^{(j)}_{t|t-1}) = \tilde{\gamma}_t^{(j)}, \quad j = 1, \dots, N.$
4. Predict the particles by drawing independent samples according to
       $x^{(i)}_{t+1|t} \sim p_t(x_{t+1}|x^{(i)}_{t|t}), \quad i = 1, \dots, N.$
5. Set $t := t + 1$ and iterate from step 2.
```

It is worth stressing that this is the basic form of the particle filter and that there are several embellishments available in the literature. However, for the contribution in this paper there is no reason to depart from the basic form and our ideas can of course be used together with most of the existing particle filters.

3. MULTI-RATE PARTICLE FILTER

Obviously one solution to the problem considered in this paper is simply to neglect the inherent structure and apply the particle filter directly. However, if we choose to make use of the inherent structure we can obtain better estimates at a lower computational cost. In the subsequent section we describe how this can be done and in Section 3.2 an efficient way of computing the estimates is given.

3.1 Algorithm

The algorithm will be explained using an induction type of reasoning. Let us start by assuming that the information in the slow, nonlinear measurement (2c) has just been used. The approximation of the filtering PDF is then given by (6). Now, during the time until the next slow measurement arrives, i.e., for the times $k = t+1, t+2, \dots, t+r$, model (2) is obviously reduced to

$$x_{k+1} = A_k x_k + B_k u_k + G_k w_k, \qquad w_k \sim \mathcal{N}(0, Q_k), \tag{7a}$$

$$y_{1,k} = C_k x_k + D_k u_k + e_{1,k}, \qquad e_{1,k} \sim \mathcal{N}(0, R_k), \tag{7b}$$

which is a linear Gaussian model. This implies, as is well known, that the solution to (5) is available in closed form via the Kalman filter [9]. Hence, there is no reason to use an approximate method such as the particle filter to compute the estimates for $p(x_k|y_{1,1:k}, y_{2,1:t}), k = t+1, \dots, t+r$, since the Kalman filter will produce the optimal estimate. Furthermore, this can be performed at a lower computational cost using the Kalman filter. The fact that $\hat{p}_N(x_t|y_{1:t})$, given by (6) is non-Gaussian prevents us from direct application of the Kalman filter. However, this can be efficiently handled using parallel Kalman filters, initiated with

$$\hat{x}^{(i)}_{k|k} = x^{(i)}_{k|k}, \qquad i = 1,\dots,N, \tag{8a}$$

$$P^{(i)}_{k|k} = 0, \qquad i = 1,\dots,N. \tag{8b}$$

Note that despite of (8b) the uncertainty is still present, since we run several Kalman filters in parallel. Hence, the uncertainty is inherent in the representation. Furthermore, (8b) is crucial since it implies that we can use the same covariance matrices $P_{k|k}, P_{k+1|k}$ and Kalman gains K_k for all particles and their computations can be performed off-line, once and for all, before the filter is employed. This is important, since it allows us to save computational resources for the online computations. The Kalman filtering computations can be performed using the standard recursions, the square-root implementation or any other form. Regardless of which form that is used the particle updates are in the following form

$$x^{(i)}_{t+r|t+r} = g(x^{(i)}_{t|t}, y_{1,t+1:t+r}, u_{t:t+r}), \qquad i = 1,\dots,N, \tag{9}$$

where g denotes a general function. More specifically, if the standard recursions for the Kalman filters are used it is straightforward to verify that the particle updates (9) are given by

$$x^{(i)}_{t+r|t+r} = Lx^{(i)}_{t|t} + \sum_{j=0}^{r} J_j u_{t+j} + \sum_{j=1}^{r} M_j y_{1,t+j}, \tag{10}$$

where $i = 1,\dots,N$ and

$$L = \prod_{j=r}^{1}(I - K_{t+j}C_{t+j})A_{t+j-1}, \tag{11a}$$

$$J_0 = \left(\prod_{i=r}^{2}(I - K_{t+i}C_{t+i})A_{t+i-1}\right)(I - K_{t+1}C_{t+i})B_t, \tag{11b}$$

$$J_j = \left(\prod_{i=r}^{j+2}(I - K_{t+i}C_{t+i})A_{t+i-1}\right)(I - K_{t+j+1}C_{t+j+1})\times (A_{t+j}K_{t+j}D_{t+j} + B_{t+j}), \quad j = 1,\dots,r-2, \tag{11c}$$

$$J_{r-1} = (I - K_{t+r}C_{t+r})(A_{t+j}K_{t+j}D_{t+j} + B_{t+j}) \tag{11d}$$

$$J_r = K_{t+r}, \tag{11e}$$

$$M_j = \left(\prod_{i=r}^{j+1}(I - K_{t+i}C_{t+i})A_{t+i-1}\right)K_{t+j}, \quad j = 1,\dots,r-1, \tag{11f}$$

$$M_r = K_{t+r}. \tag{11g}$$

The covariance matrices and the Kalman gain are given by

$$P_{k+1|k} = A_k P_{k|k} A_k^T + G_k Q_k G_k^T, \tag{12a}$$

$$K_k = P_{k|k-1}C_k^T(C_k P_{k|k-1} C_k^T + R_{1,k})^{-1}, \tag{12b}$$

$$P_{k|k} = P_{k|k-1} - K_k C_k P_{k|k-1}. \tag{12c}$$

Note that the term $\sum_{j=0}^{r} J_j u_{t+j} + \sum_{j=1}^{r} M_j y_{1,t+j}$ in (10) is the same for all particles, allowing us to save computations.

The final step is at time $t+r$, before the next measurement from the slow sensor delivers its next measurement $y_{2,t+r}$. In order to make use of this nonlinear measurement the result from the Kalman filters has to be assembled into a new set of particles. This is accomplished simply by drawing a set of particles from the current approximation of the filtering PDF $p(x_{t+r}|y_{1,1:t+r}, y_{2,1:t})$,

$$\hat{p}_N(x_{t+r}|y_{1,1:t+r}, y_{2,1:t}) = \sum_{i=1}^{N} \tilde{q}^{(i)}_{t+r}\mathcal{N}(x_{t+r}|x^{(i)}_{t+r|t+r}, P_{t+r|t+r}). \tag{13}$$

and the induction is complete. Note that the weights $q^{(i)}_{t+r}$ are given by (see [12] for details)

$$\tilde{q}^{(i)}_{t+r} = \frac{\prod_{k=0}^{r} p(y_{1,t+k}; \hat{y}^{(i)}_{1,t+k}, S_{t+k})}{\sum_{j=1}^{N}\prod_{k=0}^{r} p(y_{1,t+k}; \hat{y}^{(i)}_{1,t+k}, S_{t+k})}. \tag{14}$$

To sum up, we have now motivated Algorithm 2 below.

Algorithm 2 Multi-rate particle filter

```
1. Initialize the particles, $\{x^{(i)}_{1|0}\}_{i=1}^{N} \sim p(x_0)$.
2. Compute the importance weights $\{\gamma^{(i)}_t\}_{i=1}^{N}$

       $\gamma^{(i)}_t = p_t(y_{2,t}|x^{(i)}_{t|t-1}), \qquad i = 1,\dots,N,$    (15)

   and normalize $\tilde{\gamma}^{(i)}_t = \gamma^{(i)}_t / \sum_{j=1}^{N}\gamma^{(j)}_t$.
3. Draw N new particles with replacement, for each $i = 1,\dots,N$,

       $Pr(x^{(i)}_{t|t} = x^{(j)}_{t|t-1}) = \tilde{\gamma}^{(j)}_t, \qquad j = 1,\dots,N.$    (16)

4. Update the particles using the information in $\{y_{1,t+i}\}_{i=1}^{r}$ and $\{u_{t+i}\}_{i=1}^{r}$according to (10).
5. Draw N new particles,

       $x^{(i)}_{t+r|t+r} \sim \hat{p}_N(x_{t+r}|y_{1,1:t+r}, y_{2,1:t}), \quad i = 1,\dots,N,$

   where $\hat{p}_N(x_{t+r}|y_{1,1:t+r}, y_{2,1:t})$ is given by (13).
6. Set $t := t + r$ and iterate from step 2.
```

An obvious extension is the combined use of the marginalized particle filter and the ideas presented in this paper. The application of such an algorithm to the SLAM problem for a robot moving in 3D equipped with vision and inertial sensors could probably result in interesting results, see [14]. Note that this might require linearization of certain equations to get a model in the form (2).

3.2 Computing Estimates

The, in many respects, optimal estimate for the state at time t is the conditional expectation,

$$\hat{x}_{t|t} = \mathrm{E}(x_t|y_{1:t}). \tag{17}$$

Using the standard particle filter (Algorithm 1) this estimate is computed after step 2 and it is given by

$$\hat{x}_{t|t} = \sum_{i=1}^{N} \tilde{\gamma}_t x^{(i)}_{t|t}. \tag{18}$$

When the slow measurement $y_{2,t}$ has just been used in step 2 in the multi-rate particle filter (Algorithm 2) the conditional mean estimate (17) is of course given by (18) as well. The intermediate estimates, between two slow measurements can be computed in a similar fashion. However, that would be very computationally intensive. Using the fact that the order of linear operations can be interchanged will save a lot of computations here. To be more specific,

$$\hat{x}_{t+l|t+l} = \sum_{i=1}^{N} \tilde{q}_{t+l}^{(i)} x_{t+l|t+l}^{(i)} = \sum_{i=1}^{N} (L_l \tilde{q}_{t+l}^{(i)} x_{t|t}^{(i)} + M_l) \tag{19}$$

$$= L_l \sum_{i=1}^{N} \tilde{q}_{t+l}^{(i)} x_{t|t}^{(i)} + M_l, \tag{20}$$

where

$$L_l = \prod_{j=l}^{1} (I - K_{t+j} C_{t+j}) A_{t+j-1}, \tag{21}$$

Hence, the intermediate estimates can be computed at almost no cost at all.

4. SIMULATIONS

In this section, the multi-rate particle filter (Algorithm 2) will be compared to the standard particle filter (Algorithm 1). This is done using Monte Carlo simulations for an illustrative example.

4.1 Setup

Consider a vehicle moving in a two dimensional world. Information about this vehicle is obtained using two multi-rate sensors. One sensor measures the range to a fixed object at a known position (the origin in this case) at 1 Hz and the other sensor measures the velocity $(v_{x,t}, v_{y,t})$ at 10 Hz. For convenience we assume that the vehicle can be modelled using a constant velocity model according to,

$$\begin{pmatrix} p_{x,t+1} \\ p_{y,t+1} \\ v_{x,t+1} \\ v_{y,t+1} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{A} \begin{pmatrix} p_{x,t} \\ p_{y,t} \\ v_{x,t} \\ v_{y,t} \end{pmatrix} + \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}}_{G} \begin{pmatrix} w_{1,t} \\ w_{2,t} \end{pmatrix}, \tag{22a}$$

$$y_{1,t} = \underbrace{\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{C} \begin{pmatrix} p_{x,t} \\ p_{y,t} \\ v_{x,t} \\ v_{y,t} \end{pmatrix} + e_{1,t}, \tag{22b}$$

$$y_{2,t} = \underbrace{\sqrt{p_{x,t}^2 + p_{y,t}^2}}_{h(x_t)} + e_{2,t}, \tag{22c}$$

where $(p_{x,t}, p_{y,t})$ and $(v_{x,t}, v_{y,t})$ denote the position and velocity, respectively. Hence, the model is in the form (2) and the multi-rate particle filter given in Algorithm 2 can be applied. Model (22) is time-invariant and there are no input signals present. Hence, (10) – (11) can be simplified according to

$$x_{t+r|t+r}^{(i)} = L x_{t|t}^{(i)} + \sum_{j=1}^{r} M_j y_{1,t+j}, \tag{23}$$

where $i = 1, \dots, N$ and

$$L = \prod_{j=r}^{1} (I - K_{t+j} C) A, \tag{24a}$$

$$M_j = \left(\prod_{l=r}^{j+1} (I - K_{t+l} C) A \right) K_{t+j}, \quad j = 1, \dots, r-1, \tag{24b}$$

$$M_r = K_{t+r}. \tag{24c}$$

Furthermore, the sampling time is set to be $T_s = 1/10\,\text{s}$, the covariance for the process noise is set to be $\text{Cov}\left((w_{1,t}\ w_{2,t})^T\right) = 0.5 I_2$ and the covariance for the measurement noise is $\text{Cov}\left((e_{1,t}^T\ e_{2,t})^T\right) = I_3$. We have used 1000 Monte Carlo simulations in order to obtain reliable results.

4.2 Simulation Results

The root mean square errors (RMSE) over the Monte Carlo simulations for the position and velocity estimates are shown in Figure 1. The error in position estimates is lower for the multi-rate particle filter, whereas the velocity errors are almost identical. Note that the sawtooth pattern in the position error is due to a rapidly improved estimate every time an update on the range is available. The increasing error over time is due to incomplete observability. Measuring the range and velocity is not enough to estimate the position without drift.

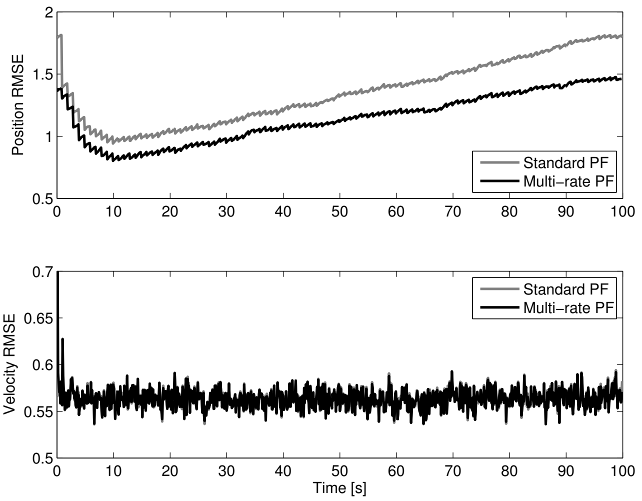

Figure 1: RMSE in position (top) and velocity (bottom) for 1000 Monte Carlo simulations, using 3000 particles.

In Figure 2 the RMSE for the position is provided as a function of the number of particles used in the filters. Here, we can see that the multi-rate particle filter performs better, but the difference becomes smaller as the number of particles increases. This is expected and in accordance to theory, since the particle filter converges to the optimal filter as the number of particles tends to infinity.

The computational time using the multi-rate particle filter is decreased, especially when a large number of particles are used. This is illustrated in Figure 3.

In studying particle filters it is always interesting to study the rate of divergence as a function of the number of particles used in the filter. First of all, let us define what is meant by divergence in this context. The filter is said to diverge

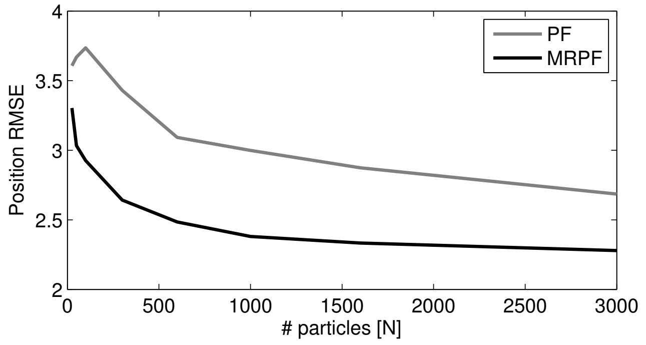

Figure 2: Position RMSE as a function of the number of particles, using 1000 Monte Carlo simulations.

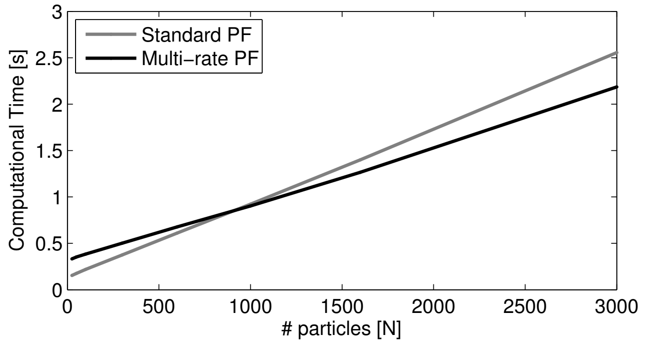

Figure 3: Illustration of the computational time as a function of the number of particles. The multi-rate particle filter requires fewer computations for large number of particles, compared to the standard particle filter.

whenever the sum of the importance weights fall below a certain problem dependent threshold $\lambda_t > 0$, i.e., when

$$\sum_{i=1}^{N} p(y_{2,t}|x_{t|t-1}^{(i)}) = \sum_{i=1}^{N} \gamma_t^{(i)} < \lambda_t. \qquad (25)$$

The motivation for this choice of divergence test is simply that it indicates when the particle cloud is too far from what is indicated by the present measurement. This is more likely to happen if there are fewer particles. Figure 4 shows the rate of divergence as a function of the number of particles. The performance of the multi-rate particle filter is significantly better for small N and slightly worse for large N. This makes sense since there is no resampling during the Kalman updates, effectively giving higher variance to the weights. To redeem this, resampling can be performed more often at the cost of computational speed.

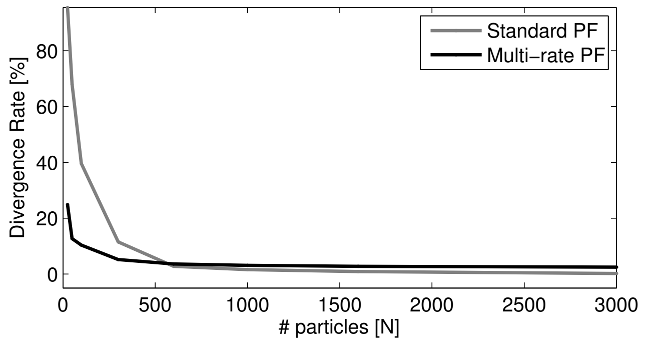

Figure 4: Rate of divergence as a function of the number of particles used in the filters.

5. CONCLUSION

We have proposed a new algorithm for state estimation, which can be used when the underlying dynamical model has a certain structure. The algorithm is based on the particle filter and exploits a linear Gaussian sub-structure. The resulting algorithm produces estimates of better quality at a lower computational cost, when compared to the standard particle filter. Finally, the structure commonly arises in sensor fusion applications, when there is one slow (nonlinear) and one fast (linear) sensor available.

REFERENCES

[1] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

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

[3] 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.

[4] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping (SLAM): Part I. *IEEE Robotics & Automation Magazine*, 13(2):99–110, June 2006.

[5] 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.

[6] 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, February 2002.

[7] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proceedings of the European Conference on Computer Vision*, volume 1, pages 343–356, Cambridge, UK, 1996.

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

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

[10] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

[11] N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

[12] U. Orguner. *Improved state estimation for jump Markov linear systems*. PhD thesis, Middle east technical university, Ankara, Turkey, 2006.

[13] 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.

[14] T. B. Schön, R. Karlsson, D. Törnqvist, and F. Gustafsson. A framework for simultaneous localization and mapping utilizing model structure. In *Proceedings of the 10th international conference on information fusion*, Québec, Canada, July 2007. Accepted for publication.

[15] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agents. The MIT Press, Cambridge, MA, USA, 2005.