

The Marginalized Particle Filter for Automotive Tracking Applications

Andreas Eidehall*[†] Thomas Schön[†] Fredrik Gustafsson[†]

*Vehicle Dynamics and Active Safety
Volvo Car Corporation
SE-405 31 Göteborg, Sweden
aeidehal@volvocars.com

[†]Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden
{eidehall,schon,fredrik}@isy.liu.se

Abstract— This paper deals with the problem of estimating the vehicle surroundings (lane geometry and the position of other vehicles), which is needed for intelligent automotive systems, such as adaptive cruise control, collision avoidance and lane guidance. This results in a nonlinear estimation problem. For automotive tracking systems, these problems are traditionally handled using the extended Kalman filter. In this paper we describe the application of the marginalized particle filter to this problem. Studies using both synthetic and authentic data shows that the marginalized particle filter can in fact give better performance than the extended Kalman filter. However, the computational load is higher.

Index Terms— automotive tracking, state estimation, extended Kalman filter, marginalized particle filter, marginalization

I. INTRODUCTION

Future intelligent automotive systems such as adaptive cruise control, collision avoidance or lane guidance will require detailed knowledge about the vehicle surroundings. In this paper, vehicle surroundings will refer to lane geometry and other vehicles. Typically, lane information is obtained from a vision system and other vehicles are detected using a radar.

The importance of integrating data from object tracking and road geometry tracking has quite recently been recognized [1], [5], [20]. The main idea is to try to improve the road geometry estimate by studying the motion of other vehicles and vice versa. For example, if a couple of tracked vehicles suddenly all start moving right, one of two things can have happened. The first is that they all started a lane change manoeuvre and the road remains straight. The other is that we are entering a curve and the vehicles are still following the center of their lanes. These possibilities can be treated within a Bayesian framework, together with the information from the lane tracker, to build a new estimator. In order to do this we need to use a nonlinear measurement equation based on the road geometry.

The most common approach to state estimation for nonlinear problems in general, and for this application in particular, is the Extended Kalman Filter (EKF). The EKF generally gives good performance for lane tracking and not much effort has been spent on studying other alternatives. In this work, we investigate and compare the performance of the EKF to the potentially more accurate particle filter [6], [7]. In particular,

we will study the Marginalized Particle Filter (MPF) [7], [8], [19]. The MPF is a subtle combination of the Kalman filter and the particle filter. It can be used for nonlinear estimation problems with a conditionally linear substructure, where the linear states are estimated using Kalman filters and the nonlinear states are estimated using the particle filter. There are two advantages with MPF compared to the standard particle filter. First, for the linear states, the Kalman filter provides the optimal solution. Second, the dimensionality problem associated with the standard particle filter is reduced, since the dimension of the state variables estimated using the standard particle filter is reduced using marginalization.

Several geometric models for combined road prediction and target tracking exists. We will use a model which assumes a road consisting of circle segments, not the traditionally used clothoid approximation. In [9], it was shown that this model can give better tracking performance. This model is presented in Section II. In order to solve the problem at hand, nonlinear estimation theory is needed. Hence, Section III is devoted to this and in particular the two algorithms EKF and MPF are discussed. These algorithms are then evaluated in Section IV. Finally the conclusions are given in Section V.

II. THE COMBINED LANE TRACKING AND OBJECT MODEL

The difference between tracking in automotive application 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, this section presents an appropriate dynamic model. The key idea, upon which this model is based, is the use of a curved coordinate system which is attached to and follows the road.

A. Dynamic motion model

The coordinates x and y denotes the position in the curved coordinate system, which is attached to the road according to Fig. 1. In these coordinates, the motion model for the other vehicles can be greatly simplified. For example, it allows us to use the equation $\dot{y}^i = 0$, which simply means that it is assumed that the other vehicles will follow their own lanes. In

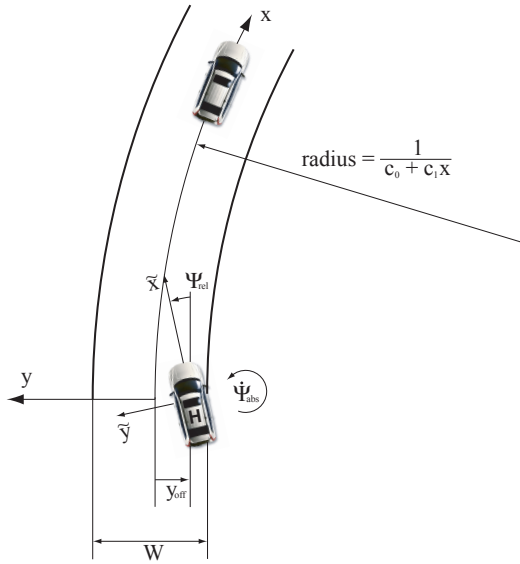


Fig. 1. The coordinate systems used in deriving the dynamic motion model. Here, (x, y) denotes the position in a curved coordinate system, which is attached to and follows the road. Furthermore, (\tilde{x}, \tilde{y}) denotes the position in a coordinate system, which is attached to the moving host vehicle.

the longitudinal direction we will use $\ddot{x}^i = -a \cos \Psi_{rel}$, where a is the measured acceleration of the host vehicle. Hence, we have the following motion model:

$$\dot{x}^i = v^i, \quad (1a)$$

$$\dot{v}^i = -a \cos \Psi_{rel}, \quad (1b)$$

$$\dot{y}^i = 0, \quad (1c)$$

where v^i is the longitudinal velocity of object i , i.e., the time derivative of x^i . For the road geometry parameters we first clarify that Ψ_{rel} is the angle between the host vehicle and the lane, whereas Ψ_{abs} is the angle to some fix reference. We can obtain a relationship between the two by differentiating Ψ_{rel} w.r.t. time,

$$\Psi_{rel} = \Psi_{abs} + \gamma \Rightarrow \quad (2a)$$

$$\dot{\Psi}_{rel} = \dot{\Psi}_{abs} + \dot{\gamma} = \dot{\Psi}_{abs} + \frac{v}{r} = \dot{\Psi}_{abs} + c_0 v, \quad (2b)$$

where r is the current road radius, v the velocity and γ denotes the angle between the lane and some fix reference. $\dot{\Psi}_{abs}$ can typically be measured with a yaw rate sensor. We also have

$$\dot{y}_{off} = \sin(\Psi_{rel})v \approx \Psi_{rel}v. \quad (3)$$

Using $\dot{W} = 0$ and $\dot{c}_1 = 0$ continuous-time motion equations for the host vehicle states can be written

$$\dot{W} = 0, \quad (4a)$$

$$\dot{y}_{off} = v\Psi_{rel}, \quad (4b)$$

$$\dot{\Psi}_{rel} = v c_0 + \dot{\Psi}_{abs}, \quad (4c)$$

$$\dot{c}_0 = v c_1, \quad (4d)$$

$$\dot{c}_1 = 0. \quad (4e)$$

The discrete-time dynamics is then given by assuming piecewise constant input signals, $[a, \dot{\Psi}_{abs}]$ [18]. Furthermore,

adding stochastic process noise, the discrete-time motion equations for the objects become

$$x_{t+1}^i = x_t^i + T_s v_t^i - a_t \cos \Psi_{rel,t} T_s^2 / 2 + w_{1,t}^i, \quad (5a)$$

$$v_{t+1}^i = v_t^i - a_t \cos \Psi_{rel,t} T_s + w_{2,t}^i, \quad (5b)$$

$$y_{t+1}^i = y_t^i + w_{3,t}^i, \quad (5c)$$

and for the host vehicle states

$$W_{t+1} = W_t + w_{4,t}, \quad (6a)$$

$$y_{off,t+1} = y_{off,t} + v T_s \Psi_{rel,t} + v^2 T_s^2 c_{0,t} / 2, \\ + v^3 T_s^3 c_{1,t} / 6 + v T_s^2 \dot{\Psi}_{abs,t} / 2 + w_{5,t}, \quad (6b)$$

$$\Psi_{rel,t+1} = \Psi_{rel,t} + v T_s c_{0,t} + v^2 T_s^2 c_{1,t} / 2, \\ + T_s \dot{\Psi}_{abs,t} + w_{6,t}, \quad (6c)$$

$$c_{0,t+1} = c_{0,t} + v T_s c_{1,t} + w_{7,t}, \quad (6d)$$

$$c_{1,t+1} = c_{1,t} + w_{8,t}. \quad (6e)$$

The variables $\{w_{i,t}\}_{i=1}^8$ are white, zero-mean Gaussian process noise, with covariance matrices Q_{host} and Q_{obj} for the host and object states, respectively.

B. Measurement model

The measurements for the host vehicle are Ψ_{rel}^m , c_0^m , L^m and R^m , where the two latter are the distances to the left and right lane marking, see Fig. 1. Superscript m denotes measured quantities. For the other vehicles we measure the position, \tilde{x}^m and \tilde{y}^m , which is expressed in the Cartesian coordinate system attached to the vehicle. These relate to the states as

$$L_t^m = W_t / 2 - y_{off,t} + e_{1,t}, \quad (7a)$$

$$R_t^m = -W_t / 2 - y_{off,t} + e_{2,t}, \quad (7b)$$

$$\Psi_{rel,t}^m = \Psi_{rel,t} + e_{3,t}, \quad (7c)$$

$$c_{0,t}^m = c_{0,t} + e_{4,t}, \quad (7d)$$

$$\begin{bmatrix} \tilde{x}_t^{i,m} \\ \tilde{y}_t^{i,m} \end{bmatrix} = T(x_t^i, y_t^i) + \begin{bmatrix} e_{5,t} \\ e_{6,t} \end{bmatrix}^i, \quad (7e)$$

where the variables $\{e_{i,t}\}_{i=1}^6$ denote white, zero-mean Gaussian measurement noise with covariance matrices R_{host} and R_{obj} for the host and object states, respectively. T is the geometric transformation from the (x, y) coordinates to the (\tilde{x}, \tilde{y}) coordinates and i is used to index the tracked objects. This transformation is given by [9]

$$T(x, y) = R(-\Psi_{rel}) \begin{bmatrix} (1 + c_0 y) \sin(c_0 x) \\ (1 + c_0 y) \cos(c_0 x) - 1 - c_0 y_{off} \end{bmatrix} \frac{1}{c_0},$$

where $R(-\Psi_{rel})$ is the rotation matrix

$$R(-\Psi_{rel}) = \begin{bmatrix} \cos(-\Psi_{rel}) & -\sin(-\Psi_{rel}) \\ \sin(-\Psi_{rel}) & \cos(-\Psi_{rel}) \end{bmatrix}. \quad (8)$$

III. NONLINEAR ESTIMATION

According to the previous section, the state-space model used in this application is nonlinear. Hence, we have to handle the problem of recursively estimating the state variable in a nonlinear state-space model,

$$x_{t+1} = f(x_t, u_t) + w_t, \quad (9a)$$

$$y_t = h(x_t) + e_t, \quad (9b)$$

where x_t denotes the state variable, u_t the input signal, w_t the process noise, y_t the measurements and e_t the measurement noise. The solution to this problem has been known for quite some time and it can be shown to be [12]

$$p(x_t|Y_t) = \frac{p(y_t|x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})}, \quad (10a)$$

$$p(x_{t+1}|Y_t) = \int p(x_{t+1}|x_t)p(x_t|Y_t)dx_t, \quad (10b)$$

where $Y_t = \{y_i\}_{i=0}^t$. Using the probability density function $p(x_t|Y_t)$, any state estimate can be formed, e.g., the commonly used least-mean-squares estimator $E(x_t|Y_t) = \int x_t p(x_t|Y_t)dx_t$. However, the problem is that the multi-dimensional integrals in (10) only permit a closed form solution in rather restrictive special cases. The most commonly used special case corresponds to that f and h in (9) are given by linear functions and w_t and e_t are independent Gaussian noise sequences. In this case (10) will reduce to the standard Kalman filter [14].

In the general case when there does not exist any closed form solution to (10) two different alternatives are discussed in the literature:

- 1) The nonlinear model (9) is approximated using a linear model, with Gaussian noise. The Kalman filter is then applied to this linearized model. This solution is referred to as the extended Kalman filter.
- 2) The optimal solution (10) is approximated using numerical methods, such as the particle filter.

Conceptually the second alternative provides the better solution. The reason is that it provides an approximation of the optimal solution to the original problem, rather than an optimal solution to an approximated problem.

The particle filter was first introduced by Gordon [10]. For a thorough introduction to the standard particle filter the reader is referred to [6], [7]. The particle filter is quite simple to implement and it is given in Algorithm 1 if steps 4a and 4c are neglected. If the model structure contains a conditionally linear substructure this can be utilized by the marginalized particle filter, which is discussed in Section III-B. Even though the (marginalized) particle filter provides an alternative which is conceptually superior to the EKF it should be kept in mind that the computational complexity of the particle filter can be quite substantial [15]. Furthermore, the quality delivered using the EKF is often sufficient. Still, it is interesting to solve the problem at hand using the (marginalized) particle filter, since it provides an alternative algorithm for solving the same problem. Furthermore, if more advanced measurement equations, such as the ones resulting from map information fusion, are considered the (marginalized) particle filter might be the only option. The reason is that these measurement equations simply cannot be handled within the Kalman filtering framework. However, within the particle filter framework the use of these highly nonlinear measurement equations is rather straightforward. The navigation problem for fighter aircraft, using terrain elevation maps, has been posed and solved using a marginalized particle filter [2], [17], [19].

In the subsequent sections it is explained how the EKF and the marginalized particle filter can be applied to the

automotive tracking problem studied in this paper, using the model described in Section II.

A. The Extended Kalman Filter

The extended Kalman filter has a long tradition in automotive applications. For details on the Kalman Filter and the extended Kalman filter, see e.g., [11], [13], [14], [16]. We will use a one-step ahead predictor based on the EKF. The equations are given below.

$$\hat{x}_{t+1|t} = A\hat{x}_{t|t-1} + AK_t(y_t - h(\hat{x}_{t|t-1})) + Bu_t, \quad (11a)$$

where the Kalman gain matrix K_t is given by,

$$C_t = \left. \frac{\partial h}{\partial x} \right|_{x=\hat{x}_{t|t-1}} \quad (11b)$$

$$K_t = P_{t|t-1}C_t^T(C_tP_{t|t-1}C_t^T + R)^{-1}, \quad (11c)$$

$$P_{t+1|t} = AP_{t|t-1}A^T + Q - AK_tC_tP_{t|t-1}A^T \quad (11d)$$

Here, Q and R are the combined process and measurement noise covariance, i.e.,

$$Q = \begin{bmatrix} Q_{host} & 0 \\ 0 & I_M \oplus Q_{obj} \end{bmatrix}, \quad R = \begin{bmatrix} R_{host} & 0 \\ 0 & I_M \oplus R_{obj} \end{bmatrix},$$

where M is the number of objects currently being tracked, I_M is an identity matrix of size M and \oplus is the Kronecker product. It is worth mentioning that in the state update (11a) a data association algorithm is used. The reader is referred to e.g., [3] for details regarding this matter.

B. The Marginalized Particle Filter

The dynamic model derived in Section II can be written on the following form

$$x_{t+1}^n = A_n^n x_t^n + A_l^n x_t^l + B^n(x_t^n)u_t + w_t^n, \quad (12a)$$

$$x_{t+1}^l = A_n^l x_t^n + A_l^l x_t^l + B^l(x_t^n)u_t + w_t^l, \quad (12b)$$

$$y_t = g(x_t^n) + C(x_t^n)x_t^l + e_t, \quad (12c)$$

where $w_t^n \sim \mathcal{N}(0, Q^n)$, $w_t^l \sim \mathcal{N}(0, Q^l)$, $e_t \sim \mathcal{N}(0, R)$, and w_t^n is independent of w_t^l . Furthermore, the state vector has been partitioned according to

$$x_t = \begin{bmatrix} x_t^l \\ x_t^n \end{bmatrix}, \quad (13)$$

where x_t^l denotes the state variables with conditionally linear dynamics and x_t^n denotes the state variables with nonlinear dynamics. The linear state variables can be marginalized out and estimated using the optimal Kalman filter, whereas the nonlinear state variables are estimated using the particle filter. This technique is sometimes referred to as Rao-Blackwellization [4]. The resulting algorithm will be referred to as the marginalized particle filter and it is thoroughly explained in e.g., [7], [8], [19]. It is well-known that the quality of the estimates will improve or remain unchanged when the MPF is used instead of the standard particle filter [8]. Furthermore, in some cases the computational complexity can be decreased using the MPF [15].

Applying the marginalized particle filter to (12) results in Algorithm 1. For a detailed derivation and discussion of this

TABLE I

THE TOTAL NUMBER OF STATES AND THE NUMBER OF NONLINEAR STATES
FOR M OBJECTS.

	M	1	2	3
Total	$3M + 5$	8	11	14
Nonlinear	$M + 2$	3	4	5

algorithm the reader is referred to [19].

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 $\tilde{q}_t^{(i)} = q_t^{(i)} / \sum_{j=1}^N q_t^{(j)}$.

3) Particle filter measurement update (resampling): 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,

$$\hat{x}_{t|t}^l = \hat{x}_{t|t-1}^l + K_t(y_t - g_t(x_t^n) - C_t \hat{x}_{t|t-1}^l),$$

$$P_{t|t} = P_{t|t-1} - K_t M_t K_t^T, \quad (14a)$$

$$M_t = C_t P_{t|t-1} C_t^T + R_t, \quad (14b)$$

$$K_t = P_{t|t-1} C_t^T M_t^{-1}. \quad (14c)$$

b) Particle filter time update (prediction): For $i = 1, \dots, N$, predict new particles,

$$x_{t+1|t}^{n,(i)} \sim p(x_{t+1}^n | X_t^{n,(i)}, Y_t).$$

c) Kalman filter time update

$$\hat{x}_{t+1|t}^l = A_t^l \hat{x}_{t|t}^l + A_t^l x_t^n + B^l(x_t^n) u_t + L_t(z_t - A_t^l \hat{x}_{t|t}^l), \quad (15a)$$

$$P_{t+1|t} = A_t^l P_{t|t} (A_t^l)^T + Q_t^l - L_t N_t L_t^T, \quad (15b)$$

$$N_t = A_t^l P_{t|t} (A_t^l)^T + Q_t^n, \quad (15c)$$

$$L_t = A_t^l P_{t|t} (A_t^l)^T N_t^{-1}, \quad (15d)$$

where $z_t = x_{t+1}^n - A_t^n x_t^n - B^n(x_t^n) u_t$.

5) Set $t := t + 1$ and iterate from step 2.

In order to be able to use Algorithm 1 we need to identify the structure (12) from the dynamic model (5) – (7). Here, x^n and x^l consist of the following states:

$$x_t^n = [\Psi_{rel,t} \quad c_{0,t} \quad x_t^1 \quad \dots \quad x_t^M]^T$$

$$x_t^l = [W_t \quad y_{off,t} \quad c_{1,t} \quad v_t^1 \quad y_t^1 \quad \dots \quad v_t^M \quad y_t^M]^T$$

where M denotes the number of tracked objects. Table I summarizes the number of states for different numbers of tracked objects. It can be seen that the motion dynamics (5)

and (6) fit the MPF structure (12) if we, for a single object, write

$$x_{t+1}^n = \underbrace{\begin{bmatrix} 1 & vT_s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{A_t^n} x_t^n + \underbrace{\begin{bmatrix} 0 & 0 & v^2 T_s^2 / 2 & 0 & 0 \\ 0 & 0 & v T_s & 0 & 0 \\ 0 & 0 & 0 & T_s & 0 \end{bmatrix}}_{A_t^l} x_t^l + \underbrace{\begin{bmatrix} T_s & 0 \\ 0 & 0 \\ 0 & \cos \Psi_{rel,t} T_s^2 / 2 \end{bmatrix}}_{B^n(x_t^n)} u_t + w_t^n \quad (16)$$

and

$$x_{t+1}^l = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ vT_s & v^2 T_s^2 / 2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{A_t^n} x_t^n + \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & v^3 T_s^3 / 6 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{A_t^l} x_t^l + \underbrace{\begin{bmatrix} 0 & 0 \\ vT_s^2 / 2 & 0 \\ 0 & 0 \\ 0 & \cos \Psi_{rel,t} T_s^2 / 2 \\ 0 & 0 \end{bmatrix}}_{B^l(x_t^n)} u_t + w_t^l. \quad (17)$$

Furthermore, the measurement model (7) conforms with this structure,

$$\begin{bmatrix} L_t^m \\ R_t^m \\ \Psi_{rel,t}^m \\ c_{0,t}^m \end{bmatrix} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ \Psi_{rel} \\ c_{0,t} \end{bmatrix}}_{g_{host}(x_t^n)} + \underbrace{\begin{bmatrix} -1/2 & 1 & 0 & 0 & 0 \\ 1/2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{C_{host}} x_t^l + e_{host,t}$$

and

$$\begin{bmatrix} \tilde{x}_t^m \\ \tilde{y}_t^m \end{bmatrix} = \underbrace{R(-\Psi_{rel,t})}_{g_{obj}(x_t^n)} \begin{bmatrix} \sin(c_0 x^i) \\ \cos(c_0 x^i) - 1 \end{bmatrix} + \underbrace{R(-\Psi_{rel,t})}_{C_{obj}(x_t^n)} \begin{bmatrix} 0 & 0 & 0 & 0 & \sin(c_0 x) \\ 0 & -1 & 0 & 0 & \cos(c_0 x) \end{bmatrix} x_t^l + e_{obj,t}.$$

Implementation of Algorithm 1 is now straightforward. Generalization of (16) and (17) to multiple objects is accomplished by extending the system matrices and the state vector accordingly.

IV. EVALUATION

In this section we will discuss how the two filters were compared. In order to evaluate a filter, the estimated states are typically compared to the true values, if available. Here, we have focused on the estimate of the curvature parameter c_0 , which is crucial to many automotive applications, such as adaptive cruise control systems, collision warning or any system that is dependent on assigning leading vehicles to

the correct lane. This is sometimes referred to as the *lane assignment problem* [9], [20].

In order to quantify the importance of the curvature parameter we approximate the \tilde{y} -part of (7e) using a second-order Taylor expansion around $\Psi_{rel} = y = y_{off} = 0$,

$$\tilde{y} = (\cos(c_0 x) - 1) \frac{1}{c_0} \approx \left(1 - \frac{(c_0 x)^2}{2} - 1\right) \frac{1}{c_0} = c_0 \frac{x^2}{2}. \quad (18)$$

This implies that for small changes in c_0 we have,

$$\Delta \tilde{y} \approx \frac{x^2}{2} \Delta c_0, \quad (19)$$

which means that for a leading vehicle 100 meters away from the host vehicle, a small error of, say $5 \cdot 10^{-4}$ (1/m) in c_0 will result in an error of 2.5 meters in \tilde{y} . This is enough to assign the leading vehicle to the wrong lane.

We have compared the filters using both synthetic and authentic data. The MPF has an extra design parameter compared to the EKF, which is the number of particle to use. The two data sets were both run, first through the EKF and then through the MPF using different numbers of particles.

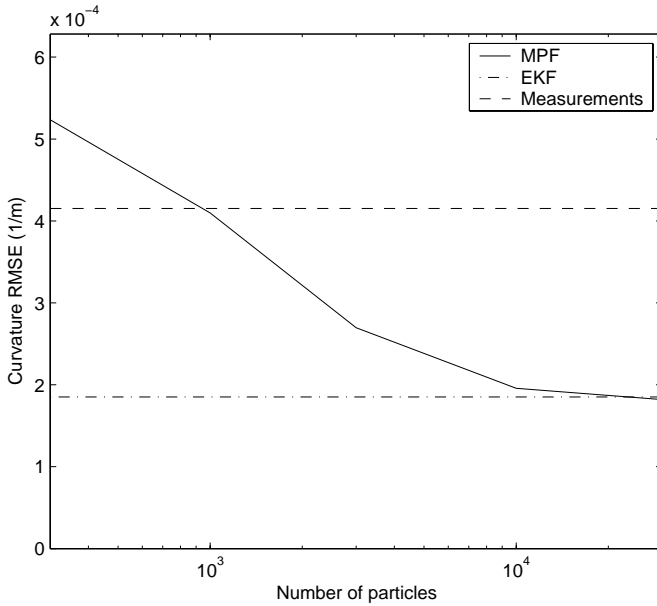


Fig. 2. The curvature RMSE for different numbers of particles, using synthetic data.

A. Synthetic data

For a simulation, the true values of all parameters are readily available. Furthermore, a simulation allow us to test any filter for specific scenarios or disturbance environments. We have tried to create a realistic environment, but at the same time tried to challenge the filters with fast changes in the road geometry. The results from the simulation are shown in Fig. 2. The plot shows the root mean square error (RMSE) of the curvature during the simulation using different numbers of particles in the MPF. In the same plot the EKF performance and the measured curvature, c_0^m , are shown.

Here, the MPF does not improve the estimation performance compared to the EKF. The explanation to this could be that the nonlinearities of the model are quite small and that the EKF is performing close to the Cramér-Rao lower bound [16].



Fig. 3. The data sequence was recorded during poor visibility conditions.

B. Authentic data

The authentic data set was recorded in the northern parts of Sweden during the winter. From Fig. 3 it is clear that the visibility of the lane markings is very low. Analogous to the simulation experiment, we have compared the EKF to the MPF

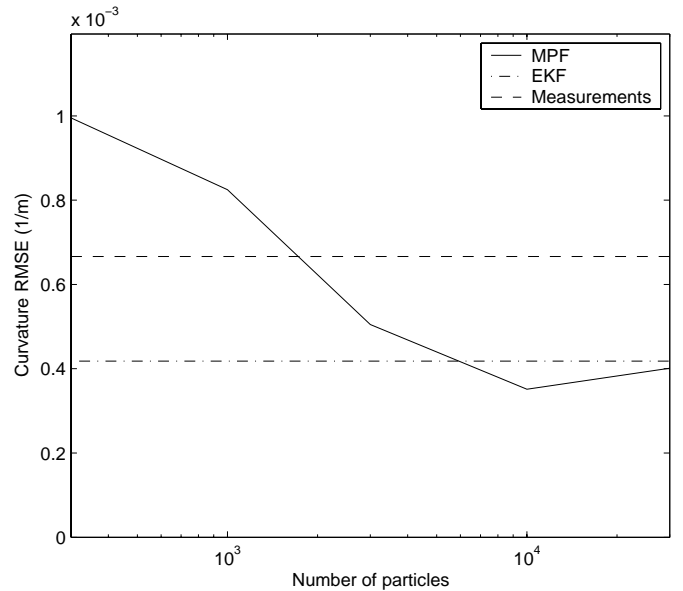


Fig. 4. The curvature RMSE for different numbers of particles, using authentic data.

using different numbers of particles and to the measured data. The true value of the curvature which was used to compute the magnitude of the errors was obtained from a detailed map. Fig. 4 show the results of the experiment. For reference, we have also plotted the curvature and the absolute curvature error over time in Fig. 5 and Fig. 6, respectively. As can be seen, the MPF outperforms the EKF during high particle number settings. Why the MPF is better here, but not for synthetic

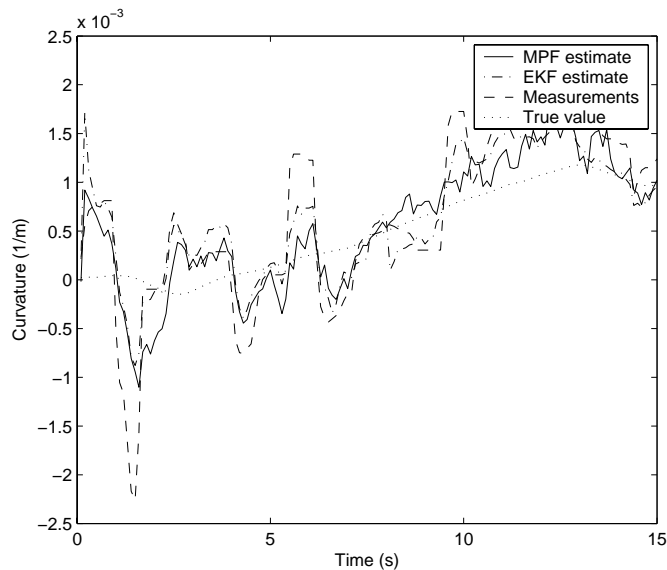


Fig. 5. The true curvature, the measured curvature and the estimates from the MPF and EKF using the authentic data.

data needs to be investigated further. The reason is probably that the MPF is more robust to model errors, since it does not depend on the derivatives of the measurement equations, or it is less sensitive to errors in other assumptions about noise and inputs used in the model. The fact that the curve is not monotonic in Fig. 4, i.e., the error is higher for the highest particle setting is due to the stochastic nature of the particle filter.

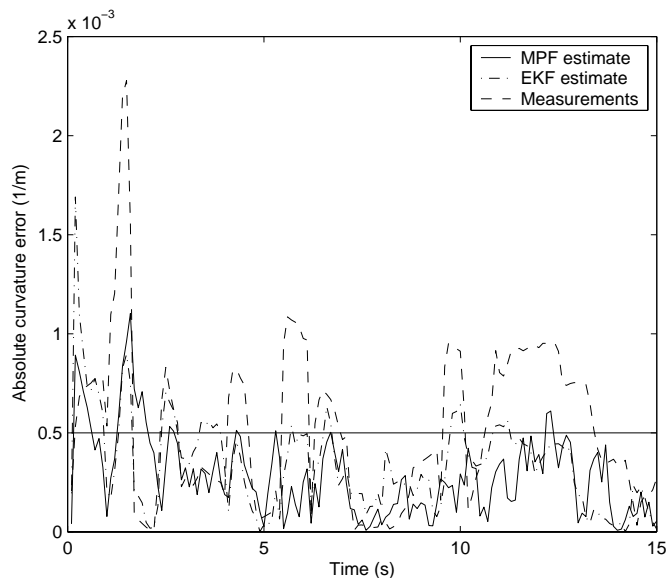


Fig. 6. The absolute curvature error. Here, the level $0.5 \cdot 10^{-3}$ (1/m) used as a motivating example in (19) has been indicated. For errors above this level, leading vehicles at a distance of 100 meters are likely to be assigned to the wrong lane.

V. CONCLUSIONS

We have shown that the marginalized particle filter can be implemented for automotive tracking and that it can give better performance than the traditionally used extended Kalman filter. Although the difference might not be big enough to motivate the extra computational cost today, future increases in computational power will allow its implementation. Furthermore, the marginalized particle filter might be the only choice if more advanced nonlinear measurement equations, such as map information fusion, are to be used.

REFERENCES

- [1] Automotive collision avoidance system field operational test. Technical report, National Highway Traffic Safety Administration, 2000.
- [2] N. Bergman, L. Ljung, and F. Gustafsson. Terrain navigation using Bayesian statistics. *IEEE Control Systems Magazine*, 19(3):33–40, June 1999.
- [3] S. S. Blackman. *Multiple-Target Tracking with Radar Applications*. Artech House, Inc., 1986.
- [4] G. Casella and C. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.
- [5] F. Dellaert and C. Thorpe. Robust car tracking using Kalman filtering and Bayesian templates. In *Proceedings of the SPIE conference on Intelligent Transportation Systems*, volume 3207, October 1997.
- [6] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.
- [7] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.
- [8] 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.
- [9] A. Eidehall and F. Gustafsson. Combined road prediction and target tracking in collision avoidance. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 619–624, Parma, Italy, June 2004.
- [10] N. Gordon, D. Salmond, and A. 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.
- [11] F. Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons, Ltd., 2000.
- [12] A. Jazwinski. *Stochastic processes and filtering theory*. Mathematics in science and engineering. Academic Press, New York, 1970.
- [13] T. Kailath, A. H. Sayed, and B. Hassibi. *Linear estimation*. Prentice Hall, 2000.
- [14] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering*, 82:35–45, 1960.
- [15] R. Karlsson, T. Schön, and F. Gustafsson. Complexity analysis of the marginalized particle filter. *Accepted for publication in IEEE Transactions on Signal Processing*, 2004.
- [16] S. M. Kay. *Fundamentals of statistical signal processing*. Prentice Hall, 1993.
- [17] P.-J. Nordlund. *Sequential Monte Carlo Filters and Integrated Navigation*. Licentiate thesis, Linköping university, 2002. Thesis No. 945.
- [18] W. Rugh. *Linear System Theory*. Information and system sciences series. Prentice Hal, Upper Saddle River, New Jersey, 2 edition, 1996.
- [19] T. Schön, F. Gustafsson, and P.-J. Nordlund. Marginalized particle filters for mixed linear/nonlinear state-space models. *Accepted for publication in IEEE Transactions on Signal Processing*, 2004.
- [20] Z. Zomotor and U. Franke. Sensor fusion for improved vision based lane recognition and object tracking with range-finders. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, November 1997.