Particle Filters for Positioning, Navigation and Tracking

Fredrik Gustafsson
Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, Per-Johan Nordlund

Division of Communication Systems
Department of Electrical Engineering
Linköpings universitet, SE-581 83 Linköping, Sweden
WWW: http://www.comsys.isy.liu.se
Email: fredrik@isy.liu.se

5th February 2001

REGLERTEKNIK
AUTOMATIC CONTROL
LINKÖPING

Report No.: LiTH-ISY-R-2333
Submitted to IEEE Trans. on Signal Processing

Abstract

A framework for positioning, navigation and tracking problems using particle filters (recursive Monte Carlo methods) is developed. A general algorithm which is parsimonious with the particle dimension is presented. Automotive and airborn applications illustrate numerically the advantage over classical Kalman filter based algorithms. Here the use of non-linear models and non-Gaussian noise is the main explanation for the improvement in accuracy.

More specifically, we describe how the technique of map matching is used to match an aircraft's elevation profile to a digital elevation map, and a car's horizontal driven path to a street map. In both cases, real-time implementations are available, and tests have shown that the accuracy is comparable to satellite navigation (as GPS), but with higher integrity. Based on simulations, we also argue how the particle filter can be used for positioning based on cellular phone measurements, for integrated navigation in aircraft, and for target tracking in aircraft and cars. Finally, the particle filter enables a promising solution to the combined task of navigation and tracking, which is illustrated both on airborne hunting and collision avoidance in cars.

Keywords: Monte Carlo methods, Statistical signal processing, Bayesian estimation, Particle filters, Applications, Positioning, Navigation, Tracking, GPS, Integrated navigation, Collision avoidance, Divergence tests

Particle Filters for Positioning, Navigation and Tracking

Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell,
Jonas Jansson, Rickard Karlsson, Per-Johan Nordlund,
Department of Electrical Engineering, Linköping University
SE-581 83 Linköping, Sweden
Email: fredrik@isy.liu.se

Submitted to IEEE Transactions on Signal Processing.
Special issue on Monte Carlo methods for statistical signal processing.
http://www-sigproc.eng.cam.ac.uk/~sjg/ieee/Special_Issue.html

Abstract— **A framework for positioning, navigation and tracking problems using particle filters (recursive Monte Carlo methods) is developed. A general algorithm which is parsimonious with the particle dimension is presented. Automotive and airborn applications illustrate numerically the advantage over classical Kalman filter based algorithms. Here the use of non-linear models and non-Gaussian noise is the main explanation for the improvement in accuracy.**

More specifically, we describe how the technique of map matching is used to match an aircraft's elevation profile to a digital elevation map, and a car's horizontal driven path to a street map. In both cases, real-time implementations are available, and tests have shown that the accuracy is comparable to satellite navigation (as GPS), but with higher integrity. Based on simulations, we also argue how the particle filter can be used for positioning based on cellular phone measurements, for integrated navigation in aircraft, and for target tracking in aircraft and cars. Finally, the particle filter enables a promising solution to the combined task of navigation and tracking, which is illustrated both on airborne hunting and collision avoidance in cars.

I. Introduction

RECURSIVE IMPLEMENTATIONS of Monte Carlo based statistical signal processing [14] are known as *particle filters*, see [10], [8]. These may be a serious alternative for real-time applications classically approached by model-based Kalman filter techniques [22], [18]. The more non-linear model, or the more non-Gaussian noise, the more potential particle filters have, especially in applications where computational power is rather cheap and the sampling rate slow. The research has since the paper [15] steadily intensified and is now boosting, where the first article collections and monographs are soon coming out. Besides, the first simulation based applications of particle filtering have appeared [1], [19].

The paper describes a general framework for a number of applications, where we have implemented the particle filter. The problem areas are

- *Positioning,* where one's own position is to be estimated. This is a filtering problem rather than a static estimation problem, when an inertial navigation system is used to provide measurements of movement.
- *Navigation,* where besides the position also velocity, attitude and heading, acceleration and angular rates are included in the filtering problem.
- *Target tracking,* where another object's position is to be estimated based on measurements of relative range and angles to one's own position.

These problems are related in that they can be described by quite similar state space models. Traditional methods are based on linearized models and Gaussian noise approximations so that the Kalman filter can be applied [2]. Research is focused on how different state coordinates or multiple models can be used to limit the approximations. In contrast to this, the particle filter approximates the optimal solution numerically based on a physical model, rather than applying an optimal filter to an approximate model. The applications we will describe are:

- *Car positioning by map matching.* A digital road map is used to constrain the possible positions, where a dead-reckoning of wheel speeds is the main external input to the algorithm. By matching the driven path to a road map, a vague initial position (order of km's) can be improved to a meter accuracy. This principle can be used as a supplement to, or even replacement to, GPS (global positioning system).
- *Car positioning by Radio Frequency (RF) measurements.* The digital road map above can be replaced by, or supplemented by, measurements from a terrestrial wireless communications system. For handover (to transfer a connection from one base station to another) operation, the mobile stations (MS) monitor the received signal powers from a multitude of base stations, and report regularly to the network. These measurements provide a power map which can be used in a similar manner as above. Mobile stations in a near future will moreover provide the possibility of monitoring the traveled distance of the radio signals from a number of base stations [12]. Such measurements can also be utilized in the same manner as with the power measurements.
- *Aircraft positioning by map matching* or *terrain naviga-*

The affiliations of the last five authors are SaabTech Systems, NIRA Dynamics, Volvo Car Corporation, Saab Dynamics and Saab Gripen, respectively. All authors are, or have been, affiliated with the Department of Electrical Engineering, Linköping University.

tion. A GIS contains, among other information, terrain elevation. The aircraft is equipped with sensors such that the terrain elevation can be measured. By map matching, the position can be deducted [4].

- *Integrated navigation.* The aircraft's Inertial Navigation System (INS) uses dead-reckoning to compute navigation and flight data, *i.e.* position, velocity, attitude and heading. The INS is regarded as the main sensor for navitation and flight data due to being autonomous and having high reliability [7]. However, small offsets cause drift and its output has to be stabilized. Here, terrain navigation is used today.
- *Target tracking.* A classical problem in signal processing literature, where radar or IR measures relative angle, and for radar also relative range and range rate, to the object [3]. For the case of bearings only measuring IR sensor, either the state dynamics or measurement equation is very non-linear depending on the choice of state coordinates, so here the particle filter is particularly promising.
- *Combined navigation and tracking.* Because the target tracking measurements are relative to one's own platform, positioning is an important sub-problem. Since the sensor introduces a cross-coupling between the problems, a unified treatment is tempting.
- *Car collision avoidance* is very similar to the target tracking problem, here we are interested in predicting the own car's and other objects' future position, see [28]. Based on the prediction, collision avoidance actions such as warning, braking and steering are undertaken when a collision is likely to happen. In order to have enough time to warn the driver the prediction horizon needs to be quite long. Therefore, utilizing knowledge about road geometry and infrastructure becomes important. One way to improve the prediction of possible maneouvres, is to use information in a digital map. Thus, this is a specific project including all aspects of the problems listed above.

The outline is as follows. We will start with a general framework of models covering all of our applications in Section II. Then, a general algorithm is presented covering all applications, where special attention is paid on practical problems as divergence test, initialization and real-time requirements. Each application in the list above is devoted its own section, and conclusions and open questions of general interest are discussed in VIII.

II. Models

Central for all navigation and tracking applications is the motion model to which various kind of model based filters can be applied. Models which are linear in the state dynamics and non-linear in the measurements are considered:

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

Motion models (1a) are further discussed in Section II-A. These are to a large extent similar in all applications. Instead, the difference between the applications lies in the availability of measurements. Section II-B provides an extensive list of possible measurement equations (1b).

A. Motion Models

The signals of primary interest in navigation and tracking applications are related to position, velocity and acceleration as summarized in Table I. Depending on whether

Object	Position	Velocity	Acceleration
Own	$p^{(1)}$	$v^{(1)}$	$a^{(1)}$, $\delta a^{(1)}$ acc. bias
Other	$p^{(2)}$	$v^{(2)}$	-

TABLE I

Interesting signals in navigation and tracking applications. Index (1) and (2) indicates signals related to one's own and another platform respectively. All quantities can belong to either one, two or three-dimensional spaces, depending on the application.

the signals are measureable or not, they may be components of either the state vector x_t or the input signal u_t. The ambition here is to discuss models through which the applications are naturally related. In specific applications, however, other parameterizations might provide better understanding design variables and algorithm tuning.

In positioning and navigation applications the signals related to the own platform are of interest. If the velocity $v_t^{(1)}$ is assumed measureable (and thus part of the input signal), the state dynamics can be modeled as

$$p_{t+1}^{(1)} = \underbrace{p_t^{(1)}}_{x_t} + \underbrace{T_s v_t^{(1)}}_{B_u u_t} + \underbrace{T_s w_t^{(1)}}_{B_w w_t} \tag{2a}$$

In several navigation applications, such as airborne, measurements of the acceleration are used instead of velocity. These are typically biased, and the true acceleration can be expressed as

$$a_{\text{true,t}}^{(1)} = a_t^{(1)} + \delta a_t^{(1)},$$

where $a_t^{(1)}$ is the measured acceleration and $\delta a_t^{(1)}$ is the bias. The position is extracted by dead-reckoning of the measured acceleration, and therefore the presence of acceleration bias is critical. The natural thing to do is to include the bias in the state vector, and the measured acceleration in the input signal. The resulting motion model is

$$\begin{pmatrix} p_{t+1}^{(1)} \\ v_{t+1}^{(1)} \\ \delta a_{t+1}^{(1)} \end{pmatrix} = \underbrace{\begin{pmatrix} I & T_s \cdot I & T_s^2/2 \cdot I \\ 0 & I & T_s \cdot I \\ 0 & 0 & I \end{pmatrix}}_{A^{(1)}} \begin{pmatrix} p_t^{(1)} \\ v_t^{(1)} \\ \delta a_t^{(1)} \end{pmatrix} + \underbrace{\begin{pmatrix} T_s^2/2 \cdot I \\ T_s \cdot I \\ 0 \end{pmatrix}}_{B_u^{(1)}} a_t^{(1)} + \underbrace{\begin{pmatrix} T_s^3/6 \cdot I \\ T_s^2/2 \cdot I \\ T_s \cdot I \end{pmatrix}}_{B_w^{(1)}} w_t^{(1)} \tag{2b}$$

Analogously, a bias in any other measured signal (*e.g.* a bias in the velocity in Equation (2a)) can be considered by incorporating it in the state equation.

This far, the focus has been on the own platform. In a simple model of the movements of the other platform, the assumption is that its velocity $v^{(2)}$ is subject to an unknown acceleration. This yields

$$\begin{pmatrix} p_{t+1}^{(2)} \\ v_{t+1}^{(2)} \end{pmatrix} = \underbrace{\begin{pmatrix} I & T_s \cdot I \\ 0 & I \end{pmatrix}}_{A^{(2)}} \begin{pmatrix} p_t^{(2)} \\ v_t^{(2)} \end{pmatrix} + \underbrace{\begin{pmatrix} T_s^2/2 \cdot I \\ T_s \cdot I \end{pmatrix}}_{B_w^{(2)}} w_t^{(2)} \quad (2c)$$

In the target tracking literature, a popular choice of motion model is given by the "coordinated turn"-model [3].

When considering tracking of another platform, while moving the own platform, joint navigation and target tracking can be employed. Essentially, the total motion model comprises the motion models (2b) and (2c):

$$\begin{pmatrix} p_{t+1}^{(1)} \\ v_{t+1}^{(1)} \\ \delta a_{t+1}^{(1)} \\ p_{t+1}^{(2)} \\ v_{t+1}^{(2)} \end{pmatrix} = \begin{pmatrix} A^{(1)} & 0 \\ 0 & A^{(2)} \end{pmatrix} \begin{pmatrix} p_t^{(1)} \\ v_t^{(1)} \\ \delta a_t^{(1)} \\ p_t^{(2)} \\ v_t^{(2)} \end{pmatrix} + \begin{pmatrix} B_u^{(1)} \\ 0 \end{pmatrix} a_t^{(1)} + \begin{pmatrix} B_w^{(1)} & 0 \\ 0 & B_w^{(2)} \end{pmatrix} \begin{pmatrix} w_t^{(1)} \\ w_t^{(2)} \end{pmatrix} \quad (2d)$$

B. Measurement Equations

The main difference between the considered applications is the measurements available. Basically, the measurements are related to the positions of one's own platform $p^{(1)}$ and of the other object $p^{(2)}$. Therefore, the measurement equations can be categorized as depending on $p^{(1)}$ only, or depending on both $p^{(1)}$ and $p^{(2)}$:

$$y_t^{(1)} = h^{(1)}(p_t^{(1)}) + e_t^{(1)} \quad (3a)$$

$$y_t^{(2)} = h^{(2)}(p_t^{(1)}, p_t^{(2)}) + e_t^{(2)}, \quad (3b)$$

where the measurement noise contributions $e_t^{(1)}$ and $e_t^{(2)}$ are characterized by their distributions. If not explicitely mentioned, a Gaussian distribution is used.

In the studied applications, measurements from at least one of the categories above are available. It is important to note, that any combination of the sensors are possible. The presented applications are just a few examples.

B.1 Measurements of Relative Distance

As always, any position has to be related to a coordinate system and a reference position. Several types of sensors (*e.g.* GPS, RF) basically measure the distance relative to that reference point. One possibility is distance measurements of the own position relative to points of known positions p_i, $i = 1, \dots, M$, which yields M measurement equations with

$$h_{a,i}^{(1)}(p_t^{(1)}) = \left| p_i - p_t^{(1)} \right|, \; i = 1, \dots, M \quad (3c)$$

This is also applicable when the position of another object is related to one's own position (*e.g.* radar, sonar, ultrasound):

$$h_b^{(2)}(p_t^{(1)}, p_t^{(2)}) = \left| p_t^{(2)} - p_t^{(1)} \right| \quad (3d)$$

Some sensors do not measure the relative distance explicitly, but rather a quantity related to the same. One example is sensors that measure the received radio signal power transmitted from a known position p_i. This received power typically decays as $\sim K_1/r^{\alpha}$, $\alpha \in [2,5]$, where K_1 and α are depending on the radio environment, antenna characteristics, terrain etc. In a logarithmic scale, the measurements are given by

$$h_{c,i}^{(1)}(p_t^{(1)}) = K - \alpha \log_{10} \left| p_i - p_t^{(1)} \right|, \;\; i = 1, \dots, M \quad (3e)$$

where $K = \log_{10} K_1$ [20]. Analogously, we can consider the situation when we focus on the power or intensity transmitted or reflected from an object and received at one's own position. The measurement is thus modeled by

$$h_d^{(2)}(p_t^{(1)}, p_t^{(2)}) = K - \alpha \log_{10} \left| p_t^{(1)} - p_t^{(2)} \right| \quad (3f)$$

B.2 Measurements of Relative Angle

Similarly, the sensors can measure the relative angle between two positions (*e.g.* radar, IR, sonar, ultrasound). Given points of known positions p_i, $i = 1, \dots, M$, the relative angle measurements can be described by

$$h_{e,i}^{(1)}(p_t^{(1)}) = \text{angle} \left\{ p_i, p_t^{(1)} \right\}, \; i = 1, \dots, M \quad (3g)$$

When relating the angle of an object to one's own position, we have

$$h_f^{(2)}(p_t^{(1)}, p_t^{(2)}) = \text{angle} \left\{ p_t^{(1)}, p_t^{(2)} \right\} \quad (3h)$$

B.3 Measurements of Relative Velocity

Some sensors (*e.g.* radar) typically measure the Doppler shift of signal frequencies to estimate the magnitude of the relative velocity. This is essentially only applicable when relating the velocity of an object to one's own velocity. The measurements are categorized by

$$h_{g,i}^{(2)}(v_t^{(1)}, v_t^{(2)}) = \left| v_t^{(2)} - v_t^{(1)} \right| \quad (3i)$$

B.4 Map Related Measurements

Figure 1 illustrates how ground altitude is computed from radar measurements of height over ground and barometric measurements from which altitude is computed. The measured terrain height together with relative movement from the INS build up a height profile as illustrated in Figure 2, and the task is to find the current position and orientation of this profile on the map.

The measurement in terrain navigation is the measured ground height, and $h_h(p^{(1)})$ is the height at point $p^{(1)}$ according to the Geographical Information System (GIS).

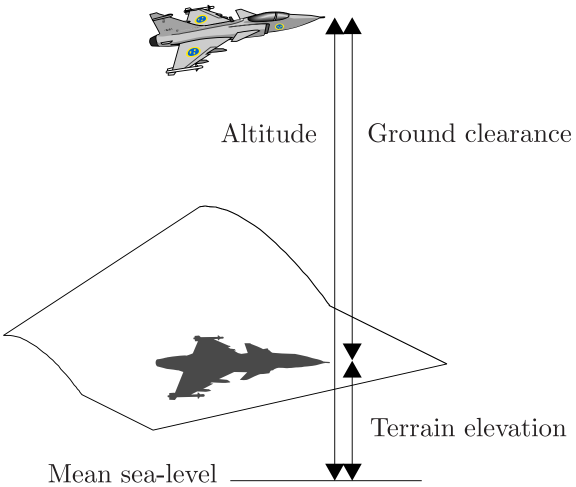


Fig. 1

Aircraft measures absolute altitude and height over ground from which ground altitude y is computed.

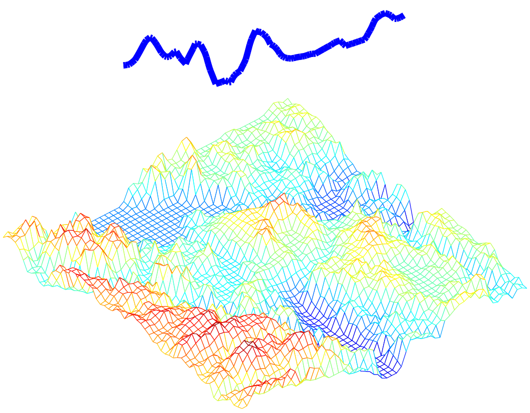


Fig. 2

Measured terrain elevation y together with measured velocity can be seen as the profile above the terrain elevation map $h(p^{(1)})$.

Much effort has been spent on modeling the measurement error $e_t^{(1)}$ in a realistic way. It has turned out that a Gaussian mixture with two modes works well. One mode has zero mean, and the other a positive mean which corresponds to radar echos from the tree tops. The ground type in GIS can be used to switch the mean and variances in the Gaussian mixture. For instance, over sea there is only one mode with a small variance.

For map matching in the car positioning case, there is no real measurement. Instead, $h_j^{(1)}(p_t^{(1)})$ denotes the distance to the nearest road, and the measurement

$$y_t^{(1)} = h_j^{(1)}(p_t^{(1)}) + e_t^{(1)}$$

should therefore be equal to zero. A simple and relevant noise model is white and zero mean Gaussian noise.

C. Applications

The applications discussed briefly in Section I are explored in further detail in the sequel. Typical state vectors, input signals and available (non-linear) sensor information are summarized in Table II. Motivations and more elaborative discussions regarding the applications and appropriate models are found in Sections IV, V, VI and VII.

III. The Particle Filter

A. Recursive Bayesian Estimation

Consider systems that are described by the generic state space model (1). The optimal Bayesian filter in this case is given below. For further details, consult [4].

We assume independent noise with probability densities p_{e_t} and p_{w_t}. Denote the observations at time t

$$Y_t = \{y_0, \dots, y_t\}.$$

The Bayesian solution to compute the posterior distribution, $p(x_t|Y_t)$, of the state vector, given past observations, is given by [4]

$$\begin{aligned} p(x_{t+1}|Y_t) &= \int p(x_{t+1}|x_t)p(x_t|Y_t)\,dx_t \\ &= \int p_{w_t}(B^{\dagger}(x_{t+1} - Ax_t - B_u u_t)p(x_t|Y_t)\,dx_t \end{aligned} \tag{4a}$$

$$\begin{aligned} p(x_t|Y_t) &= \frac{p(y_t|x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})} \\ &= \frac{p_{e_t}(y_t - h(x_t))p(x_t|Y_{t-1})}{c_t} \end{aligned} \tag{4b}$$

$$\hat{x}_t^{\text{MMS}} = \int x_t p(x_t|Y_t) dx_t \tag{4c}$$

$$P_t^{\text{MMS}} = \int (x_t - \hat{x}_t^{\text{MMS}})^2 p(x_t|Y_t) dx_t \tag{4d}$$

where $\dagger$ denotes the Moore-Penrose pseudo-inverse, c_t a normalization constant, and $\hat{x}_t^{\text{MMS}}$ the minimum mean square (MMS) estimate.

If the noise distributions are independent, white and zero mean Gaussian with $\text{E}e_t^2 = R$, $\text{E}w_t^2 = Q$ and the measurement equation is linear in the state, *i.e.* $h(x_t) = Cx_t$, the optimal solution is given by the Kalman estimator [22]. Table III summarizes the time and measurement update equations for the Kalman estimator.

B. Particle Filter Implementation

A numerical approximation to (4) is given in the following algorithm.

Algorithm 1: **Particle Filter**

1. *Initialization:* Generate $x_0^i \sim p_{x_0}, i = 1, \dots, N$. Each sample of the state vector is referred to as a *particle*.
2. *Measurement update:* Update the likelihood:

$$p^i := p^i p(y_t|x_t^i) = p^i p_{e_t}(y_t - h(x_t^i)).$$

Application	State vector	Input	Measurement equations
Car positioning	$p_t^{(1)}$	$v_t^{(1)}$	Road map $h_j(p_t^{(1)})$, possibly GPS or base station distances $h_{a,i}^{(1)}(p_t^{(1)})$, base station powers $h_{c,i}^{(1)}(p_t^{(1)})$
Aircraft positioning	$p_t^{(1)}$	$a_t^{(1)}$	Altitude map $h_j(p_t^{(1)})$, GPS or other reference beacons $h_{a,i}^{(1)}(p_t^{(1)})$
Navigation in aircraft	$p^{(1)}, v^{(1)}, \delta a_t^{(1)}$	$a_t^{(1)}$	Altitude map $h_j(p_t^{(1)})$, GPS or other reference beacons $h_{a,i}^{(1)}(p_t^{(1)})$
Tracking	$p_t^{(2)}, v_t^{(2)}$		distance $h_b^{(2)}(p_t^{(1)}, p_t^{(2)})$, bearing $h_f^{(2)}(p_t^{(1)}, p_t^{(2)})$, doppler $h_g^{(2)}(p_t^{(1)}, p_t^{(2)})$, intensity $h_d^{(2)}(p_t^{(1)}, p_t^{(2)})$
Navigation and tracking in aircraft	$p_t^{(1)}, v_t^{(1)}, \delta a_t^{(1)}, p_t^{(2)}, v_t^{(2)}$	$a_t^{(1)}$	Altitude map $h_j(p_t^{(1)})$, GPS or other reference beacons $h_{a,i}^{(1)}(p_t^{(1)})$ distance $h_b^{(2)}(p_t^{(1)}, p_t^{(2)})$, bearing $h_f^{(2)}(p_t^{(1)}, p_t^{(2)})$, doppler $h_g^{(2)}(p_t^{(1)}, p_t^{(2)})$, intensity $h_d^{(2)}(p_t^{(1)}, p_t^{(2)})$
Navigation and tracking in cars	$p_t^{(1)}, v_t^{(1)}, \delta a_t^{(1)}, p_t^{(2)}, v_t^{(2)}$	$a_t^{(1)}$	Road map $h_j(p_t^{(1)})$, possibly GPS or base station distances $h_{a,i}^{(1)}(p_t^{(1)})$, base station powers $h_{c,i}^{(1)}(p_t^{(1)})$ distance $h_b^{(2)}(p_t^{(1)}, p_t^{(2)})$, bearing $h_f^{(2)}(p_t^{(1)}, p_t^{(2)})$, doppler $h_g^{(2)}(p_t^{(1)}, p_t^{(2)})$, intensity $h_d^{(2)}(p_t^{(1)}, p_t^{(2)})$

TABLE II

List of considered applications with respective state vector (cf. Table I), input signal and sensor information.

Normalize to $p^i = p^i / \sum_i p^i$. As an approximation to (4c), take

$$\hat{x}_t \approx \sum_{i=1}^{N} p^i \hat{x}_t^i,$$

3. *Resampling:*

(a) *Bayesian bootstrap.* Take N samples with replacement from the set $\{x_t^i\}_{i=1}^N$ where the probability to take sample i is p^i. Let $p^i = 1/N$. This step is also called *Sampling Importance Resampling (SIR).*

(b) *Importance sampling.* Only resample as above when the effective number of samples is less than a threshold,

$$N_{\text{eff}} = \frac{1}{\sum_i (p^i)}^2 < N_h.$$

Here $1 \leq N_{\text{eff}} \leq N$, where the upper bound is attained when all particles have the same weight, and the lower bound when all probability mass is at one particle. The threshold can be chosen as $N_h = 2N/3$.

4. Prediction: $x_{t+1}^i = Ax_t^i + B_u u_t + B_w w_t^i$

5. Let $t := t + 1$ and iterate to item 2.

The key point with resampling is to prevent high concentration of probability mass at a few particles. Without this step, some p^i will converge to 1 and the filter would brake down to a pure simulation. The resampling can be efficiently implemented using a classical algorithm for sampling N ordered independent identically distributed variables [27], [4].

It can be shown analytically, that under some conditions the estimation error is bounded by g_t/N. The function g_t grows with time, but does not depend on the dimension of the state space. That is, in theory we can expect the same good performance for high order state vectors. This is one of the key reasons for the success of the particle filter compared to other numerical approaches such as the point mass filter [4] and filter banks [18]. The computational steps are compared to the Kalman filter in Table III.

C. Sample Impoverishment

When the particle filter is used in practice, we often wish to minimize the number of particles to reduce the computational burden. For many applications using recursive Monte Carlo methods depletion or sample impoverishment may occur, i.e. the effective number of samples is reduced. This means that the particle cloud will not reflect the true density. Several different methods are proposed in the literature to reduce this problem.

By introducing an additional noise to the samples the depletion problem can be reduced. This technique is called *jittering* in [13], but a similar approach was introduced in [15] under the name *roughening*. In [11] the depletion problem is handled by introducing an additional Markov Chain Monte Carlo (MCMC) step to separate the samples.

In [15] the so-called *prior editing* method is discussed. The estimation problem is delayed one time-step, so that the likelihood can be evaluated at the next time step. The idea is to reject particles with sufficiently small likelihood values, since they are not likely to be resampled. The update step is repeated until a feasible likelihood value is received. The roughening method could also be applied before the update step is invoked. The *auxiliary particle filter* [26] is constructed in such a way that we will simulate from particles associated with large predictive like-

Algorithm	Kalman filter	Particle filter
Time update	$x := Ax$ $P := APA^T + BQB^T$	$p^i := Ap^i + Bw^i$
Measurement update	$x := x + K(y - Cx)$ $P := P - KC^T P$ $K = PC(C^T PC + R)^{-1}$	$p^i := p^i p_e(y - h(x^i))$

TABLE III
COMPARISON OF KF AND PF: MAIN COMPUTATIONAL STEPS.

lihoods directly. A two stage resampling may be used by this method.

D. Rao-Blackwellization

Despite the theoretical independence of accuracy on the particle dimension, it is well-known that the number of particles needs to be quite high for high-dimensional systems, see for instance Section VI for an illustration. To be able to use a small N, and also to reduce the risk of divergence, a procedure known as Rao-Blackwellization [9] can be applied. The idea is to use the Kalman filter for the part of the state space model that is linear, and the particle filter for the other part. All of the motion models given in Section II can actually be re-written in the form

$$\begin{pmatrix} x_{t+1}^{pf} \\ x_{t+1}^{kf} \end{pmatrix} = \begin{pmatrix} I & A^{pf} \\ 0 & A^{kf} \end{pmatrix} \begin{pmatrix} x_t^{pf} \\ x_t^{kf} \end{pmatrix} + \begin{pmatrix} B_u^{pf} \\ B_u^{kf} \end{pmatrix} u_t + \begin{pmatrix} B_w^{pf} \\ B_w^{kf} \end{pmatrix} w_t \tag{5a}$$

$$y_t = h(x_t^{pf}) + e_t, \tag{5b}$$

where x_t^{pf} (pf short for particle filter) and x_t^{kf} (kf short for Kalman filter) is a partition of the state vector with w_t Gaussian. Also $p(x_0^{kf})$ needs to be Gaussian, but $p(e_t)$ and $p(x_0^{pf})$ are arbitrarily given distributions. As the indeces indicate, the Kalman filter will be applied to one part and the particle filter for the other part of the state vector.

For a derviation of the algorithm, suppose first that the particle filter part of the state vector is known. That is, the sequence $X_t^{pf} = \{x_0^{pf}, \ldots, x_t^{pf}\}$ is known. We can, temporally, consider $z_t = x_{t+1}^{pf} - x_t^{pf}$ as the measurement. The state space model is here

$$x_{t+1}^{kf} = A^{kf} x_t^{kf} + B_u^{kf} u_t + B_w^{kf} w_t$$
$$z_t = A^{pf} x_t^{kf} + B_u^{pf} u_t + B_w^{pf} w_t.$$

Since this model is linear and Gaussian, the optimal solution is provided by the Kalman filter. We then know that $p(x_t^{kf}|X_t^{pf})$ is Gaussian, so

$$x_t^{kf}|X_t^{pf} = x_t^{kf}|Z_{t-1} \in N(\hat{x}_{t|t-1}^{kf}, P_{t|t-1}^{kf}),$$

where $\hat{x}_{t|t-1}^{kf}$ and $P_{t|t-1}^{kf}$ are given by the Kalman filter equations adjusted for correlated noise [22],

$$\begin{aligned} K_t &= P_{t|t-1}^{kf} A^{pf} (A^{pf} P_{t|t-1}^{kf} A^{pf} + B_w^{pf} Q_t (B_w^{pf})^T)^{-1} \\ \hat{x}_{t+1|t}^{kf} &= \overline{A}^{kf} (\hat{x}_{t|t-1}^{kf} + K_t(z_t - A^{pf} \hat{x}_{t|t-1}^{kf})) + \\ &\quad B_u^{kf} u_t + B_w^{kf} (B_w^{pf})^\dagger z_t \\ P_{t+1|t}^{kf} &= \overline{A}^{kf} (P_{t|t-1}^{kf} - K_t A^{pf} P_{t|t-1}^{kf})(\overline{A}^{kf})^T, \end{aligned}$$

where $\overline{A}^{kf} = A^{kf} - B_w^{kf}(B_w^{pf})^\dagger A^{pf}$ († denotes the Moore-Penrose pseudo-inverse).

Now, to compute $p(x_t|Y_t) = p(x_t^{pf}, x_t^{kf}|Y_t)$, note that

$$p(X_t^{pf}, x_t^{kf}|Y_t) = p(x_t^{kf}|X_t^{pf}) p(X_t^{pf}|Y_t),$$

where $Y_t = \{y_0, \ldots, y_t\}$. We only have to compute $p(X_t^{pf}|Y_t)$. Repeated use of Bayes' rule gives

$$p(X_t^{pf}|Y_t) = \frac{p(y_t|x_t^{pf}) p(x_t^{pf}|X_{t-1}^{pf})}{p(y_t|Y_t)} p(X_{t-1}^{pf}|Y_{t-1}).$$

We have a non-linear and non-Gaussian measurement equation, so to solve the measurement update, the particle filter will be used to approximate this distribution.

The particle predictions $p(x_{t+1}^{pf}|X_t^{pf})$ are given by

$$x_{t+1}^{pf}|X_t^{pf} = x_t^{pf} + A^{pf} x_t^{kf}|X_t^{pf} + B_u^{pf} u_t + B_w^{pf} w_t,$$

so $p(x_{t+1}^{pf}|X_t^{pf})$ is given by

$$N(x_t^{pf} + A^{pf} \hat{x}_{t|t-1}^{kf} + B_u^{pf} u_t, A^{pf} P_{t|t-1}^{kf} (A^{pf})^T + B_w^{pf} Q_t (B_w^{pf})^T).$$

The result is a particle filter with N particles estimating x_t^{pf}. For each particle, one Kalman filter estimates $\{x_t^{kf,i}; i = 1, .., N\}$. Note that P_t^{kf} and the Kalman gain K_t is the same for all particles, implying a very efficient implementation of the N parallel Kalman filters, where the P and K updates in Table III are done only once per time step.

The estimate of the particle filter part is computed in the normal way, and for the Kalman filter part x_t^{kf} we can take the MMS estimate (4c)

$$\hat{x}_t^{kf} \approx \sum_{i=1}^{N} w_t^i E_{p(x_t^{kf}|X_t^{pf,i})}[x_t^{kf}] = \sum_{i=1}^{N} w_t^i \hat{x}_{t|t-1}^{kf,i},$$

with covariance (4d)

$$P_t^{kf} \approx P_{t|t-1}^{kf} + \sum_{i=1}^{N} w_t^i (\hat{x}_{t|t-1}^{kf,i} - \sum_{i=1}^{N} w_t^i \hat{x}_{t|t-1}^{kf,i})^2.$$

where the weights are defined by $w_t^i = p(y_t|x_t^{pf,i})$. For details on the derivation see [25].

IV. Car Positioning

Wheel speed sensors in ABS are available as standard components in the test car (Volvo V40). From this, yaw rate and speed information are computed as described in [17]. Therefore, the velocity vector $v_t^{(1)}$ is considered available as an input signal, and the motion model in (2a) with measurement equation given by (3a) is thus appropriate. The initial position is either marked by the driver or given from a different system, *e.g.* a terrestrial wireless communications system, where crude position information is available today [12] or GPS. The initial area should cover an area not extending more than a couple of kilometers to limit the number of particles to a realizable number. With infinite memory and computation time, no initialization would be necessary.

The car positioning with map matching has been implemented in a car and the particle filter runs in real-time with sampling frequency 2 Hz on a modern laptop with a hand-coded digital road map. This corresponds to a measurement equation specified by $h_j^{(1)}(p_t^{(1)})$ in Section II-B.4.

Figure 3 shows a sequence of images of the particle cloud on a flight image of the local area. The driven path consists of a number of 90 degrees turns. Initially, the particles are spread uniformly over all admissible positions, that is, on the roads, covering an area of about one square kilometer. After the first turns, a few clouds are left. After 4–5 turns, the filter essentially has converged. One can note that the state evolution on the straight path extends the cloud along the road to take into account unprecise velocity information. Details of the implementation are found in [16], [19].

GPS is used as a reference positioning system. It provides reliable position estimates in rural areas, but is hampered in non line-of-sight situations and when the signals are attenuated by foliage etc. After convergance, the map matching particle filter is seen equal to, or even slightly better, than GPS in terms of performance, see Figure 4. However, in test drives along forests, close to high buildings and tunnels, the GPS performance deteriorates quickly. Furthermore, the GPS has a convergence time of about 45 seconds when turned on, not shown in Figure 4.

For comparison, the particle filter using map matching and filters based on measurements from a fictive terrestrial wireless communications system are applied to data from a simulation setup mimicking the real case above. The area is essentially covered by one macro cell, but yet another base station is assumed within measurable distance.

The base stations in a terrestrial wireless communications system act as beacons by transmitting pilot signals of known power. The mobile station monitors the M (in GSM

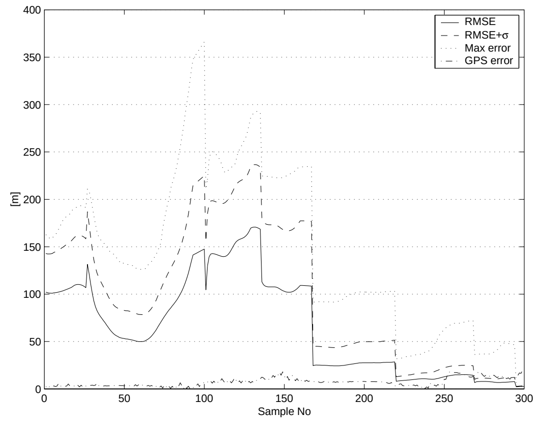


Fig. 4

Car positioning: RMSE for particle filter and GPS, respectively.

(Global System for Mobile Communications), $M = 5$) strongest signals, and reports regularly (or event-driven) the list to the network. Based on these lists, the network centrally transfers connections from one base station to another (handover) when the mobile is moving during the service session. According to the empirical model by Okumura-Hata [20], this provides M measurement equations as in (3e), one for each available base station (in this simulation, $M = 2$), and $p_e(e) \in \mathrm{N}(0, \sigma_e^2)$, where $\sigma_e = 6$ dB. Similar measurements, but with a different motion model (the velocity is unknown) are used in [21]. Point-mass implementation of estimators based on RF measurements is also discussed in [6].

To provide more accurate positioning via RF measurements, future mobile stations will be able to estimate the traveled distance of radio signals from a multitude of base stations. In the ideal case, the signals have traveled without reflections to the mobile station (line-of-sight situation), and the estimates describe the distance to the base stations. The M (M is typically 1-3) measurement equations can thus be modeled by (3c), and they represents a rather ideal situation. Moreover, the noise is modeled as $p_e(e) \in \mathrm{N}(0, \sigma_e^2)$, where $\sigma_e = 3$ dB. The received power measurements discussed above are available today, but are of worse accuracy due to unmodeled power variations.

A third alternative is to simply integrate the relative movements provided by the ABS (dead-reckoning). Monte-Carlo simulations based on these different approaches are summarized in Figure 5. It is interesting to note that map matching provides a position accuracy of roughly the same accuracy as with accurate distance measurements (which would almost never be the case in a real situation), without relying on external signals. Furthermore, integrating the ABS signals directly yields an increasing error over time.

V. Terrain Elevation Matching

The air fighter JAS 39 Gripen is equipped with an accurate radar altimeter and a digital map. This corresponds to

Fig. 3
Car positioning: Sequence of illustrations of particle clouds (white dots) plotted on a flight image for visualization. The centre point '+' shows the true position and 'x' the estimate.

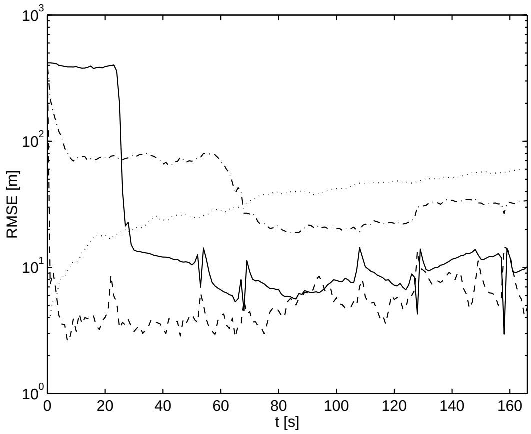


Fig. 5
RF positioning: Monte-Carlo performance over time in the simulated scenario. The map matching (solid) needs some 25 seconds to converge, but after this burn-in time, the algorithm provides RMSE=8.7 m. This is almost as good as with ideal distance measurements to two base stations (dashed) with RMSE=7,0 m. For comparison, power measurements (dash-dotted) yield RMSE=36 m, and dead-reckoning (dotted) a steadily increasing error with RMSE=50 m.

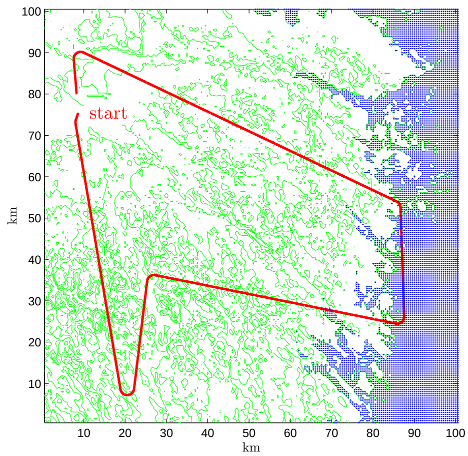


Fig. 6
Terrain navigation: Test track over a part of south-eastern Sweden.

the measurement equation characterized by $h_h(p^{(1)})$ in Section II-B.4. The velocity vector is obtained by integrating the acceleration provided by the inertial navigation system. Since $v_t^{(1)}$ is available as an input signal, the motion model in (2a) is appropriate.

The particle filter has been applied to a number of flight tests on the new fighter JAS 39 Gripen, and Figure 6 shows the path in one of them. In these tests, differential GPS (DGPS) is taken as the true position, and the resulting position error is shown in Figure 7. The accuracy beats the first generation commercial system, and comes down to the value of the point mass filter described in [5]. Since the point mass filter satisfies the Cramer-Rao lower bound, there is no better filter. The advantage of the particle filter over the point mass filter is firstly a much less complex algorithm occupying only some 30 lines of code (Ada), and secondly the possibility to extend the functionality by including other parameters such as barometric height offset in the state vector (that is, increasing the particle dimension). Saab has decided to implement the particle filter in all Gripen in parallel with the first generation system.

VI. Integrated Navigation Systems

As a simplified study to illustrate the Rao-Blackwellization procedure, a one-dimensional navigation model is used according to (2b) and the measurement of position is taken from the terrain navigation algorithm according to Section II-B.4. The simulation parameters are

$$T_s = 1, \quad w_t^{(1)} \sim \mathrm{N}(0, 0.0001^2), \quad \mathrm{e}_{\mathrm{t}}^{(1)} \sim \mathrm{N}(0, 6^2).$$

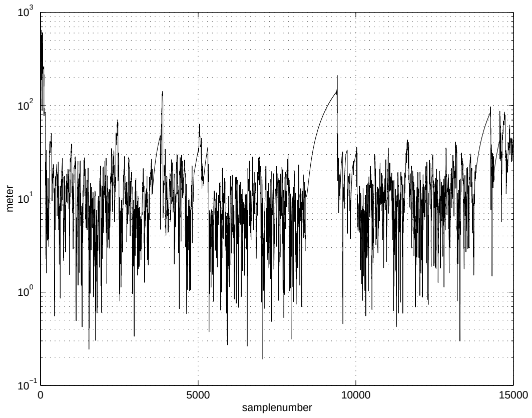


Fig. 7
TERRAIN NAVIGATION: ESTIMATION ERROR RELATIVE A GPS REFERENCE, AS A FUNCTION OF SAMPLE NUMBER. NOTE THE GROWTH IN ERROR OVER OPEN WATER.

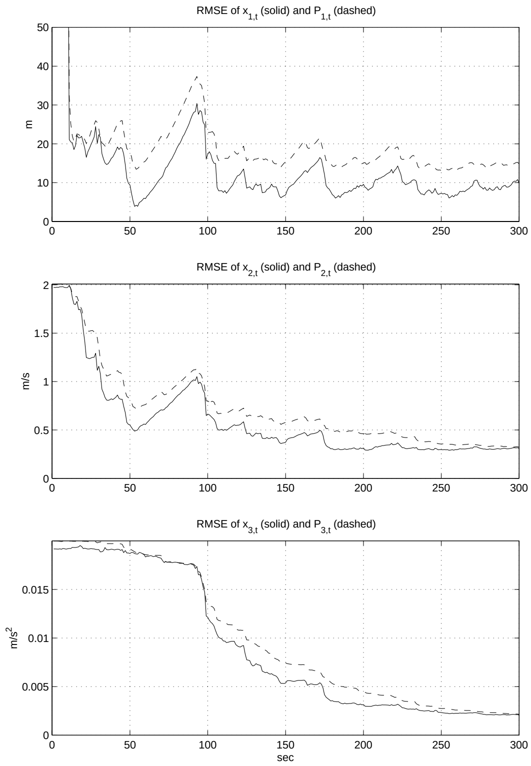


Fig. 8
INTEGRATED NAVIGATION: PARTICLE FILTER WITH 20000 PARTICLES AND 100 MONTE CARLO SIMULATION. (TWO FILTERS DIVERGED.)

It should be noted that the one-dimensional navigation model is valid only when the earth is modeled as flat. In this case the one-dimensional model can easily be extended to the two-dimensional case. However, as soon as one accounts for the curvature of the earth the model becomes more complicated, see [7]. In practice, there also exist gyro sensor errors, which further complicate the problem.

Figure 8 shows the result when using $N = 20000$ particles. The performance deteriorates when this number is decreased, in particular the transient requires many particles. The basic problem is high dimensionality and small process noise, but following the Rao-Blackwellisation procedure we partition the state vector and rewrite the motion model according to (5), with

$$x_t^{pf} = p_t^{(1)}, \quad x_t^{kf} = \begin{pmatrix} v_t^{(1)} \\ \delta a_t^{(1)} \end{pmatrix}.$$

The result of this combined Kalman and particle filter with $N = 1000$ (< 20000) particles and 100 Monte Carlo runs is shown in Figure 9. No filter diverged.

VII. TARGET TRACKING

Similar target tracking algorithms can be used for different applications, such as air traffic control (ATC) or collision avoidance. Often linear models such as (1) can be used, but nonlinear state equations may occur. For instance, when the tracking object is moving in straight paths or on circular segments, different variations of the so-called coordinated turn model [3] can be utilized. For maneuvering targets, multiple models are used to enhance tracking performance. The *Interacting Multiple Model* (IMM) [3] is one classical multiple model algorithm based on the interaction of several extended Kalman filters [2]. Hard constraints on system states, such as velocity and acceleration boundaries or obstacles from the terrain may introduce nonlinearities in many applications, which could degrade performance if not handled by the tracking filter. Two different applications will be presented in more detail below. Both uses realistic measurements (3g), modeling the radar loob in the angle noise distribution, and (3c), with uniform range noise distribution.

A. *Air Traffic Control (ATC)*

In [23], a simple nearly coordinated turn model [3] was used for an ATC radar application.

In the simulation study presented in Table VII-A, two different simulation based methods are compared to the state-of-the-art IMM method. The particle filter algorithms tested are the Bayesian bootstrap method (3a) and APF [26]. The particle filters are here extended to the multiple model case, where target maneuvers are according to a Markov chain. Three different turn assumptions were made (right/left turn or straight flying) in the simulations presented. The true path projected in the horizontal plane is viewed in Figure 10. It was generated with a true turn rate value chosen as an intermediate value of the turn rate used in the multiple model conditioning, thus allowing the IMM to mix between models, and the particle filter process noise to perform the turn interaction. The incorporation of hard constraints on the velocity is also straightforward for the particle filter case. The radar sensor used in the application measures range, azimuth and elevation at a rather low update rate, to emulate a track-while-scan (TWS) be-

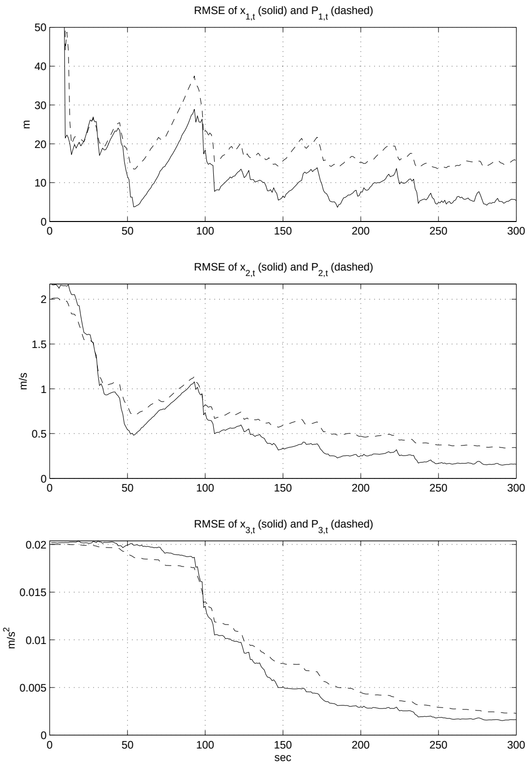


Fig. 9

INTEGRATED NAVIGATION: COMBINED KALMAN AND PARTICLE FILTER WITH $N = 1000$ (< 20000) PARTICLES AND 100 MONTE CARLO RUNS.

	APF	Bootstrap	IMM-3	Measurements
RMSE	34.03	40.84	42.20	63.96

TABLE IV

TARGET TRACKING: RMSE COMPARISON FOR ATC MONTE CARLO SIMULATIONS.

havior. In Table VII-A the IMM method is compared to the particle filters and measurements only, viewing the position RMSE for 100 Monte Carlo simulations. For the Bayesian bootstrap case, two simulations diverged. Depending on the choice of process noise, the slight difference between the IMM and the Bayesian bootstrap may change. The marginalized density is also shown in Figure 11 together with the particle cloud.

B. Collision Avoidance

The coordinated turn model can be used for collision avoidance to track the car position and predict future positions. The goal of the prediction in this case is not necessarily to get an as good point estimate as possible. Instead, we are interested in the whole distribution of possible maneuvers. Figure 12 shows a simulation where the collision is still avoidable. This would not be obvious from just looking at the point estimate.

The main contributions to the process noise come from

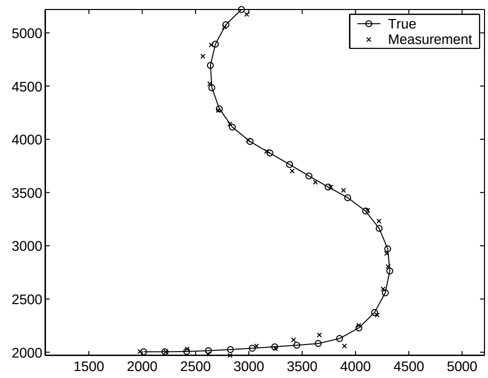


Fig. 10

TARGET TRACKING: RMSE FROM 100 MONTE CARLO SIMULATIONS, 800 PARTICLES

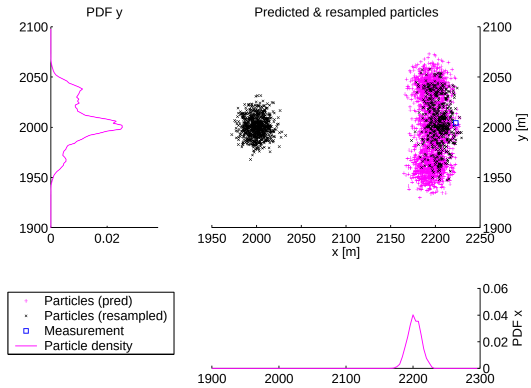


Fig. 11

TARGET TRACKING: PARTICLE CLOUD AND DENSITY

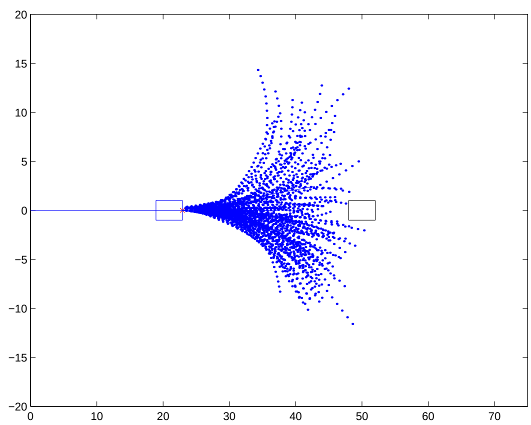


Fig. 12

COLLISION AVOIDANCE: THE LEFT RECTANGLE IS THE OWN CAR, WHICH IS APPROACHING RAPIDLY THE RIGHT RECTANGLE. THE TRAJECTORIES INDICATE 31-STEP AHEAD PREDICTION USING 100 PARTICLES. THERE ARE STILL POSSIBLE TRAJECTORIES AVOIDING COLLISION, OF WHICH THE DRIVER WILL MOST PROBABLY CHOOSE ONE. THUS, NO ACTIVE CONTROL IS NEEDED AT THIS STAGE.

the driver's action via steering wheel, gas and brake. A lot of effort has to be spent on how to choose the process noise so that it corresponds to the drivers behavior and the physical limitations of the car. The vehicle and driver behavior changes significantly for different speeds of the vehicle. Thus, in order to get a good prediction with this model, it is necessary to let the process noise w_t change with different speeds. It is also important in this application to incorporate knowledge about the environment to improve the prediction. For example, it is likely that the car will travel on the road and if there are some hard boundaries like rails or other stationary objects these are hard constraints on the car's movement.

VIII. Conclusions and Discussion

We have given a general framework for positioning and navigation applications based on a flexible state space model and a particle filter. Five applications illustrate its use in practice. Evaluations in real-time, off-line on real data and in simulation environments show a clear improvement in performance compared to existing Kalman filter based solutions, where the new challenge is to find non-linear relations, state constraints and non-Gaussian sensor models that provide the most information about the position. Thus, *modeling* is the most essential step in this approach, compared to the various *implementations* of the Kalman filter found in this context (linearization issues, choice of state co-ordinates, filter banks, Gaussian sum filters, etc.).

General conclusions from the implementations are as follows: A choice of state coordinates making the state equation linear is beneficial for computation time and opens up the possibility for Rao-Blackwellization. This procedure enables a significant decrease in the particle state dimension. The evaluation of the likelihood one step ahead before resampling (APF, prior editing) is, together with adding extra state noise (jittering, roughening), crucial for avoiding divergence, and implies that the number of particles can be decreased further. Our implementations run in real-time (2Hz), even in Matlab, and have some 2000 particles.

Open questions for further research and development are listed below:

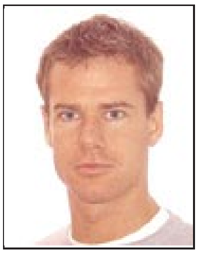
- *Divergence tests.* It is essential to have a reliable way to detect divergence and to restart the filter (for the latter, see the transient below). For car positioning, the number of resamplings in the prior editing step turned out to be a very good indicator of divergence. Another idea, used in the terrain navigation implementation where the sampling rate is higher than necessary, is to split up the measurements to a filter bank, so that particle filter number i, $i = 1, 2, \dots, n$ gets every n'th sample. The result of these n particle filters are approximately independent and voting can be used to restart each filter. This has turned out to be an efficient way to remove the outliers in data.
- *Transient improvement.* The time it takes until the estimate accuracy comes down to the stationary value (the Cramer-Rao bound) depends on the number of particles. Given limited computational time, it may be advantageous to increase the number of particles N after a restart and discard samples in such a way that $N \cdot f_s$ is constant.
- Since the particle filter has shown good improvement over linearization approaches, it is tempting to try even more accurate non-linear models. In particular, the flight dynamics of one's own vehicle is known and indeed used in model-based control, but is very rare in navigation applications, see [24] for one attempt in this direction. In that study, it seems that the computational burden and linearization errors imply no gain in total performance. As a possible improvement, the particle filter may take full advantage of a more accurate model, where parts of the non-linear dynamics from driver/pilot inputs are incorporated.

Acknowledgment

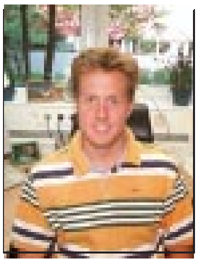
The competence centre ISIS at Linköping University has brought all of the authors together and provides funding for Rickard and Per-Johan. We are very grateful to Christophe Andrieu and Arnoud Doucet for our fruitful discussions on the theoretical subjects. We want to acknowledge our gratitude to the master students Magnus Ahlström, Marcus Calais, who have implemented the terrain navigation filter, and Peter Hall, who implemented the car positioning filter, and the supporting companies SAAB Dynamics and NIRA Automotive, respectively.

Fredrik Gustafsson is professor in Communication Systems at the Department of Electrical Engineering at Linköping University. He received the M.S. degree in electrical engineering in 1988 and the Ph.D. degree in automatic control in 1992, both from Linköping University, Sweden. His research is focused on statistical methods in signal processing, with applications to automotive, avionic and communication systems. He is an associate editor of IEEE Transactions of Signal Processing.

Fredrik Gunnarsson is a research associate at Communications Systems, Department of Electrical Engineering, Linköping University. He received the M.Sc. degree in applied physics and electrical engineering in 1996 and the Ph.D. degree in electrical engineering in 2000, both from Linköping University, Sweden. His research interests include control and signal processing in terrestrial wireless communications systems and automotive applications.



Niclas Bergman is with SaabTech Systems. He received his M.Sc. degree in applied physics and electrical engineering in 1995 and the Ph.D. degree in electrical engineering in 1999, both from Linköping University, Sweden. He is currently working with research and development in the areas of target tracking and navigation, and responsible for the coordination of data fusion activities within the Saab group.



Urban Forssell Urban Forssell is president and CEO of NIRA Dynamics AB. The company focuses on advanced signal processing and control in vehicles. He received his M.Sc. degree in applied physics and electrical engineering in 1995 and the Ph.D. degree in automatic control in 1999, both from Linköping University, Sweden.

Jonas Jansson is employed at Volvo Car Corporation with developing a collision avoidance system. He is since 1999 spending 50% of his time as a PhD student at Linköping University. His current research interests focus on particle filter implementation of navigation and tracking systems.

Rickard Karlsson has worked at SAAB Dynamics with target tracking and sensor fusion since 1997. He is since 1999 spending half his time as a PhD student at Linköping University. His current research interests focus on particle filter implementation of target tracking algorithms with radar and/or IR sensors.

Per-Johan Nordlund has worked at SAAB Gripen with developing a new version of the navigation system for the fighter JAS 39 Gripen for several years. He is since 1999 spending 75% of his time as a PhD student at Linköping University. His current research interests focus around particle filter implementation of integrated navigation systems with particular attention to complexity aspects and fault detection.

References

[1] M. Ahlström and M. Calais. Bayesian terrain navigation with Monte Carlo method. Master Thesis LiTH-ISY-EX-3051, Dept of Elec. Eng. Linköping University, S-581 83 Linköping, Sweden, 2000. In Swedish.

[2] B.D.O. Anderson and J.B. Moore. *Optimal filtering*. Prentice Hall, Englewood Cliffs, NJ., 1979.

[3] Y. Bar-Shalom and X.R. Li. *Estimation and tracking: principles, techniques, and software*. Artech House, 1993.

[4] N. Bergman. *Recursive Bayesian Estimation: Navigation and Tracking Applications*. Dissertation nr. 579, Linköping University, Sweden, 1999.

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

[6] J. Blom, F. Gunnarsson, and F. Gustafsson. Estimation in cellular radio systems. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing.*, Phoenix, AZ, USA., March 1999.

[7] K.R. Britting. *Inertial Navigation Systems Analysis*. Wiley - Interscience, 1971.

[8] J.F.G. de Freitas, A. Doucet, and N.J. Gordon, editors. *Sequential Monte Carlo methods in practice*. Springer–Verlag, 2000.

[9] A. Doucet and C. Andrieu. Particle filtering for partially observed Gaussian state space models. Technical Report CUED/F-INFENG/TR393, Department of Engineering, University of Cambridge CB2 1PZ Cambridge, September 2000.

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

[11] A. Doucet, N.J. 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, May 2000.

[12] C. Drane, M. Macnaughtan, and C. Scott. Positioning GSM telephones. *IEEE Communications Magazine*, 36(4), 1998.

[13] P. Fearnhead. *Sequential Monte Carlo methods in filter theory*. PhD thesis, University of Oxford, 1998.

[14] W. Gilks, S. Richardson, and D. Spiegelhalter. *Markov Chain Monte Carlo in practice*. Chapman & Hall, 1996.

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

[16] F. Gustafsson. Car positioning system. Swedish patent application nr, 2000.

[17] F. Gustafsson, S. Ahlqvist, U. Forssell, and N. Persson. Sensor fusion for accurate computation of yaw rate and absolute velocity. In *Submitted to SAE 2001, Detroit.*, 2001.

[18] Fredrik Gustafsson. *Adaptive filtering and change detection*. John Wiley & Sons, Ltd, 2000.

[19] P. Hall. A bayesian approach to map-aided vehicle positioning. Master Thesis LiTH-ISY-EX-3104, Dept of Elec. Eng. Linköping University, S-581 83 Linköping, Sweden, 2001. In Swedish.

[20] M. Hata. Empirical formula for propagation loss in land mobile radio services. *IEEE Transactions on Vehicular Technology*, 29(3), 1980.

[21] H. Jwa, Kim S., S. Cho, and Chun J. Position tracking of mobiles in a cellular radio network using the constrained bootstrap filter. In *Proc. National Aerospace Electronics Conference*, Dayton, OH, USA, October 2000.

[22] T. Kailath, A.H. Sayed, and B. Hassibi. *Linear estimation*. Information and System Sciences. Prentice-Hall, Upper Saddle Riber, New Jersey, 2000.

[23] Rickard Karlsson and Niclas Bergman. Auxiliary particle filters for tracking a maneuvering target. In *Proc. of the 39th IEEE Conference on Decision and Control*, Sydney, Australia, Dec 2000.

[24] M. Koifman and I.Y. Bar-Itzhack. Inertial navigation system aided by aircraft dynamics. *IEEE Transactions on Control Systems Technology*, 7(4):487–493, 1999.

[25] P-J. Nordlund. Recursive state estimation of nonlinear systems with applications to integrated navigation. Technical Report LITH-ISY-R-2321, Department of Electrical Engineering, Linköping University, SE-58183 Linköping, Sweden, 2000.

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

[27] B.D. Ripley. *Stochastic Simulation*. John Wiley, 1988.
[28] Stephen Rohr, Richard Lind, Robert Myers, William Bauson, Walter Kosiak, and Huan Yen. An integrated approach to automotive saftey systems. *Automotive engineering international*, September 2000.