

Marginalized Particle Filter for Accurate and Reliable Terrain-Aided Navigation

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

Abstract—This paper details an approach to the integration of INS (Inertial Navigation System) and TAP (Terrain-Aided Positioning). The solution is characterized by a joint design of INS and TAP, meaning that the highly nonlinear TAP is not designed separately but jointly with the INS using one and the same filter.

The applied filter extends the theory of the MPF (Marginalized Particle Filter) given by [1]. The key idea with MPF is to estimate the nonlinear part using the particle filter and the part which is linear, conditionally upon the nonlinear part, is estimated using the Kalman filter. The extension lies in the possibility to deal with a third multi-modal part, where the discrete mode variable is also estimated jointly with the linear and nonlinear parts. Conditionally upon the mode and the nonlinear part, the resulting subsystem is linear and estimated using the Kalman filter.

Given the nonlinear motion equations which the INS uses to compute navigation data, the INS equations must be linearized for the MPF to work. A set of linearized equations is derived and the linearization errors are shown to be insignificant with respect to the final result. Simulations are performed and the result indicates near-optimal accuracy when compared to the Cramer-Rao lower bound.

Index Terms—Terrain-aided navigation, particle filter, Kalman filter, marginalized.

I. INTRODUCTION

ACCURATE AND RELIABLE navigation systems have been identified as a critical enabling technology for enhanced aircraft capabilities in the coming 10-20 years. One reason is the foreseen increased use of unmanned aerial vehicles (UAVs). Following the introduction of UAVs the requirements on the navigation system (cost, size and performance) is strengthened, and no stand-alone navigation sensor is capable of meeting them all. The solution is to blend the output from two or more navigation sensors to achieve an overall good enough accuracy and reliability.

Due to its reliability and short-term accuracy, even for flight conditions involving substantial maneuvering, inertial navigation systems (INS) are usually regarded as the primary source of navigation data. The major drawback with inertial navigation is that initialization and sensor errors cause computed quantities to drift. To stabilize the drift and ensure long-term accuracy the inertial navigation system is integrated with one or more aiding sources. The standard aiding source today is the global positioning system (GPS), see e.g. [2],

[3], [4]. Although satellite navigation is seeing a widespread use, problems with the GPS such as reception limitation and interference increase the relevance of other aiding navigation sensors. One example is terrain-referenced navigation, or terrain-aided positioning (TAP). The principle is to measure terrain variations along the flight path and compare it to a database with stored terrain elevation for given positions. Although TAP does not suffer from the limitations applicable for GPS, there are other criteria which must be met. The distance to the ground needs to be within the operating range of the radar altimeter, you need a terrain elevation map over the area of interest and last but not least you need terrain variation along the flight path (which is not always the case e.g. when flying over water). Nevertheless, often the drawbacks of TAP are easier to accept than those for GPS, and the idea of using the terrain height [5], [6] or landmarks [7], [8] for positioning purposes has been around for quite some time now.

The challenge with TAP is to deal with its highly nonlinear, non-analytical characteristics. When facing a nonlinear estimation problem, a standard tool among practitioners is to apply the extended Kalman filter (EKF). Due to TAPs multi-modal character, corresponding to a measured terrain profile matching several profiles in the database, the EKF often fails. Better performance is obtained using grid-based methods, e.g. the point-mass filter [6], where the probability is discretized over the state space. This is possible due to the low dimensionality of TAP (either two or three dimensions if considering altitude besides horizontal position).

Traditionally, integrating TAP with INS has been performed using separate filters, one for TAP estimating position and another for estimating INS quantities using position from TAP as input [9]. Here, we use state-of-the-art joint design, meaning that we blend TAP and INS tightly in one and the same filter, see Figure 1.

Using this tight fusion technique means that we need to solve a nonlinear, high-dimensional problem. Here high-dimensional means that we have to consider not only position but also INS computed quantities such as velocity, attitude and heading. This rules out grid-based methods which, due to the computational load increasing exponentially with the dimension, are tractable only up to three dimensions. Simulation-based methods, such as the particle filter (PF) [10], have the promising feature of theoretically being independent of dimensionality. Simulation results indicate however that this is not the case in practice, although less dependent compared to the grid-based methods. Moreover, based on analysis and simulations [11] we know that a high performance INS with position error typically in the range of one nautical mile per hour (1.825 km/hr) is not very well suited for the particle filter.

Manuscript received January 17, 2007; revised October 23, 2007 and April 7, 2008. This work was supported by ISIS, NFFP and Saab Aerosystems.

P.-J. Nordlund is with the Department of Decision Support and Autonomy, Saab Aerosystems, Linköping, Sweden (e-mail: Per-Johan.Nordlund@saab.se).

F. Gustafsson is with the Department of Electrical Engineering, Linköping University, Linköping, Sweden (e-mail: fredrik@isy.liu.se).

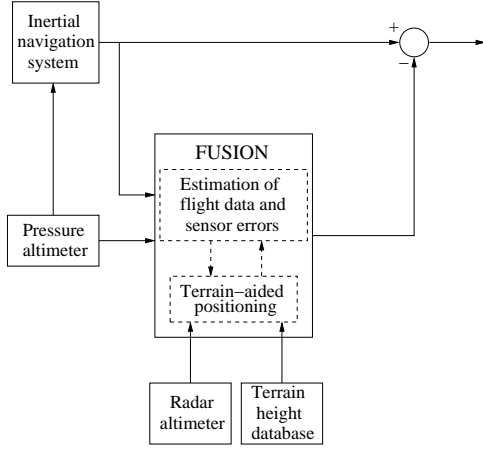


Fig. 1. Chosen configuration for blended INS/TAP. The pressure altimeter is used to stabilize INS vertical channel.

This has to do with the process noise being so small, making the particles cluster in state space and thereby increasing the discretization error of the particle filter. For the stand-alone particle filter to work on the blended INS/TAP described here the number of particles needed for filter convergence is simply too large to be computationally tractable [11].

However, we derive a set of linearized equations for the INS errors and we show that the linearization errors are small. Particularly this is true when the errors are kept small either using a high performance INS or by feeding the error back to the INS. Still the problem is highly nonlinear due to TAP, but now with a conditionally linear sub-structure corresponding to the additional INS related quantities. This means that conditionally upon position the INS related quantities can be estimated using the extended Kalman filter while position is estimated using the particle filter. The combined Kalman/particle filter is also known as the marginalized particle filter (MPF) [1] or the Rao-Blackwellized particle filter (RBPF) [12].

For the system to be able to provide accurate estimates of position we need an accurate estimate of altitude. One way forward and the one detailed in this paper is to use measurements from a radar altimeter (RA). The ground clearance measurements from the RA are however subject to a mode dependent error characteristic. The measurement error reflects e.g. whether there are a lot of trees on the ground or not. Conditionally upon the mode and nonlinear horizontal position, altitude is straightforwardly estimated by the Kalman filter. The MPF from [1] is here extended to account for this multi-modal character of the terrain elevation measurements. The extension consists of including estimation of the discrete RA measurement error mode as a third part in the joint filter design. The use of an airborne laser scanner (ALS) to measure ground clearance [13] is an interesting alternative to the radar altimeter. The accuracy and possibility to filter measurements which originate from tree reflections should yield significantly better estimation accuracy. On the other hand, the ALS has problem penetrating fog, rain and clouds which limits its applicability.

II. OUTLINE

This paper begins by a derivation of INS error dynamics in Section III. The INS nonlinear equations, for the sake of completeness given in Appendix I, are linearized and the resulting linear INS error equations are shown to accurately describe the aircraft dynamics. The non-analytical and highly nonlinear terrain-aided positioning system is introduced in Section IV. The derivation of the extended marginalized particle filter (MPF) is given in Section V. The derivation consists of three lemmas, where each lemma provides result on how to estimate the linear, multi-modal and nonlinear parts respectively. The details on the algorithm for the blended INS/TAP system is given in VI where we also analyze convergence properties of the filter. The algorithm is tested in a simulation study as described in VII. Finally conclusions are elaborated on in Section VIII.

III. INS ERROR DYNAMICS

Collect all navigation variables, i.e. latitude L , longitude l , altitude h , velocity in north v_n , east v_e and down v_d directions and attitude and heading represented by a transformation matrix from body to navigation frame C_b^n , in a state vector

$$z = [L \ l \ h \ v_n \ v_e \ v_d \ \text{vec}(C_b^n)]^T. \quad (1)$$

The input variables, i.e. accelerations f^b and angular rates ω^b , are collected in the vector

$$w = [f_x^b \ f_y^b \ f_z^b \ \omega_x^b \ \omega_y^b \ \omega_z^b]^T. \quad (2)$$

For details regarding the navigation and input variables see Appendix I. The state dynamics, given by (76), (78) and (84) in Appendix I, can compactly be written according to

$$\dot{z} = f(z, w). \quad (3)$$

Denote the corresponding INS state and input vectors by z^{ins} and w^{ins} . Due to initialization and sensor errors the state vector computed by the INS will differ from the true state vector. Define the INS state and sensor errors according to

$$x = z - z^{\text{ins}}, \quad u = w - w^{\text{ins}}. \quad (4)$$

Combining (3)–(4) we can write the error dynamics as

$$\dot{x} = \dot{z} - \dot{z}^{\text{ins}} = f(z, w) - f(z - x, w - u). \quad (5)$$

The goal is to provide a set of linearized equations describing the INS error dynamics,

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) + \Delta, \quad (6)$$

such that Δ representing the linearization error is small. Below we derive x , u , A and B in (6) such that

$$\frac{|\Delta_i|}{|[Ax]_i|} < 0.01, \quad i = 1, \dots, n, \quad \forall x \in \text{Table I} \quad (7)$$

where n is the number of states in x . We will show that x and u according to

$$x = [\tilde{L} \ \tilde{l} \ \tilde{h} \ \tilde{v}_n \ \tilde{v}_e \ \gamma_n \ \gamma_e \ \gamma_d \ b_x^a \ b_y^a]^T, \quad (8)$$

$$u = [u^h \ u_x^a \ u_y^a \ u_x^\gamma \ u_y^\gamma \ u_z^\gamma \ u_x^b \ u_y^b]^T. \quad (9)$$

and A and B according to Appendix II are adequate. The vector x in (8) is extended with two states for accelerometer biases b_x^a and b_y^a compared to (4). Moreover, u in (9) consists of white noise components including process noise for the accelerometer biases u_x^b and u_y^b . Note that there is no accelerometer bias or noise along body frame z -axis. Instead white noise u^h enters the equation for altitude error \tilde{h} directly. The reason for this is that the INS is supported by a pressure altimeter, see Figure 1, which compensates for any drift in the vertical channel.

TABLE I
RANGES ON NAVIGATION DATA.

$ L \leq 70$ deg	$ \tilde{L} \leq 2000/r_0$ rad
$h \leq 5000$ m	$ \tilde{h} \leq 100$ m
$ v_{\text{tot}} \leq 200$ m/s	$ \tilde{v}_{\text{tot}} \leq 2$ m/s
$ \gamma_{n,e} \leq 5 \cdot 10^{-4}$ rad	$ \gamma_d \leq 1 \cdot 10^{-3}$ rad
$ f_{x,y,z}^b \leq 20$ m/s ²	$ b_{x,y}^a \leq 5 \cdot 10^{-3}$ m/s ²

From (76) and (5) the expressions for the latitude and longitude errors become

$$\begin{aligned}\dot{\tilde{L}} &= \frac{v_n}{r_L(L) + h} - \frac{v_n - \tilde{v}_n}{r_L(L - \tilde{L}) + h - \tilde{h}}, \\ \dot{\tilde{l}} &= \frac{v_e \cos^{-1} L}{r_l(L) + h} - \frac{(v_e - \tilde{v}_e) \cos^{-1}(L - \tilde{L})}{r_l(L - \tilde{L}) + h - \tilde{h}}.\end{aligned}\quad (10)$$

Apply Taylor expansion on (10) around h , ε^2 and L and we can rewrite the equations according to

$$\begin{aligned}\dot{\tilde{L}} &= \frac{1}{r_0} \tilde{v}_n + \Delta_{\tilde{L}}, \\ \dot{\tilde{l}} &= \frac{v_e \sin L}{r_0 \cos^2 L} \tilde{L} + \frac{1}{r_0 \cos L} \tilde{v}_e + \Delta_{\tilde{l}}.\end{aligned}\quad (11)$$

Inserting values on the errors involved from Table I, the magnitude on $\Delta_{\tilde{L}}$ and $\Delta_{\tilde{l}}$ in (11) is

$$\begin{aligned}\frac{|\Delta_{\tilde{L}}|}{|[Ax]_{\tilde{L}}|} &< \varepsilon^2 + \frac{v_n \tilde{h}}{\tilde{v}_n r_0} < 9 \cdot 10^{-3}, \\ \frac{|\Delta_{\tilde{l}}|}{|[Ax]_{\tilde{l}}|} &< \frac{\tilde{v}_e \varepsilon^2 + v_e \tilde{h}/r_0}{v_e \tilde{L} \tan L + \tilde{v}_e} < 8 \cdot 10^{-3}.\end{aligned}\quad (12)$$

From (76) we see that the equation for the altitude error \tilde{h} becomes

$$\dot{\tilde{h}} = -\tilde{v}_d. \quad (13)$$

The INS is unstable in the vertical channel with a time constant of approximately 10 minutes [14]. For an operational INS the vertical channel must therefore be stabilized. Typically this is done using a pressure sensor. A simple and reasonable assumption is that the pressure sensor error drifts according to a random walk process, where the driving noise is described by the u^h -component in (9). Moreover, we assume that the stabilization works through an altitude filter such that the INS altitude error follows the pressure altitude error, resulting in the equation

$$\dot{\tilde{h}} \approx u^h. \quad (14)$$

Applying (5) on (78) gives the velocity error equation

$$\begin{aligned}\dot{\tilde{v}}^n &= \tilde{C}_b^n f^b + (C_b^n - \tilde{C}_b^n) \tilde{a}^b - (\tilde{\Omega}_{en}^n + 2\tilde{\Omega}_{ie}^n) v^n - \\ &\quad (\Omega_{en}^n - \tilde{\Omega}_{en}^n + 2(\Omega_{ie}^n - \tilde{\Omega}_{ie}^n)) \tilde{v}^n + \tilde{g}^n,\end{aligned}\quad (15)$$

with $f^{b,\text{ins}} = f^b - \tilde{a}^b$ and $\tilde{g}^n \approx [0 \ 0 \ \tilde{g}_d]^T$. Taylor expansion on $\tilde{\omega}_{en}^n$ using (80) and (5) around ε^2 , h and L and on $\tilde{\omega}_{ie}^n$ around L provides

$$\begin{aligned}\tilde{\omega}_{en}^n + 2\tilde{\omega}_{ie}^n &= \hat{\omega}_{en}^n + 2\hat{\omega}_{ie}^n + \Delta_{\delta\omega}, \\ \hat{\omega}_{en}^n + 2\hat{\omega}_{ie}^n &= \begin{bmatrix} \frac{\tilde{v}_e}{r_0} - 2\omega_{ie} \tilde{L} \sin L \\ -\frac{\tilde{v}_n}{r_0} \\ -\frac{v_e \tilde{L}}{r_0 \cos^2 L} - \frac{\tilde{v}_e \tan L}{r_0} - 2\omega_{ie} \tilde{L} \cos L \end{bmatrix}.\end{aligned}\quad (16)$$

We can also simplify $\omega_{en}^n - \tilde{\omega}_{en}^n + 2(\omega_{ie}^n - \tilde{\omega}_{ie}^n)$ in (15) according to

$$\begin{aligned}\omega_{en}^n - \tilde{\omega}_{en}^n + 2(\omega_{ie}^n - \tilde{\omega}_{ie}^n) &= \hat{\omega}_{en}^n + 2\omega_{ie}^n + \Delta_{\omega}, \\ \hat{\omega}_{en}^n + 2\omega_{ie}^n &= \begin{bmatrix} \frac{v_e}{r_0} + 2\omega_{ie} \cos L \\ -\frac{v_n}{r_0} \\ -\frac{v_e \tan L}{r_0} - 2\omega_{ie} \sin L \end{bmatrix}.\end{aligned}\quad (17)$$

Moreover, define a small-angle transformation $\gamma^n = [\gamma_n \ \gamma_e \ \gamma_d]^T$, in skew-symmetric matrix form denoted by Γ^n , through

$$\begin{aligned}\tilde{C}_b^n &= C_b^n - C_b^{n,\text{ins}} = C_b^n - C_n^{n,\text{ins}} C_b^n \\ &= (I - C_n^{n,\text{ins}}) C_b^n = \Gamma^n C_b^n + \Delta_{\tilde{C}_b^n}.\end{aligned}\quad (18)$$

The small-angle transformation describes a rotation of the navigation frame computed by the INS relative to the true navigation frame. The rest term $\Delta_{\tilde{C}_b^n}$ consists of second and higher order terms of γ^n which are obtained after Taylor expansion of $I - C_n^{n,\text{ins}}$. Applying the approximations on the velocity error equation in (15), together with $f^n = C_b^n f^b$, $\Gamma^n f^n = -F^n \gamma^n$ and $\Omega^n v^n = -V^n \omega^n$, yields

$$\begin{aligned}\dot{\tilde{v}}^n &= -F^n \gamma^n + C_b^n \tilde{a}^b + \tilde{V}^n (\hat{\omega}_{en}^n + 2\omega_{ie}^n) + \\ &\quad V^n (\hat{\omega}_{en}^n + 2\hat{\omega}_{ie}^n) + \tilde{g}^n + \Delta_{\tilde{v}}^n.\end{aligned}\quad (19)$$

The error introduced in \tilde{v}_n and \tilde{v}_e when going from (15) to (19) is upper limited by

$$\begin{aligned}\|\Delta_{\tilde{v}_{n,e}}\|_{\infty} &\leq \|\Delta_{\tilde{C}_b^n} f^b\|_{\infty} + \|\Gamma^n \tilde{a}^b\|_{\infty} \\ &\quad + \|\tilde{V}^n \Delta_{\omega}\|_{\infty} + \|V^n \Delta_{\tilde{\omega}}\|_{\infty}.\end{aligned}\quad (20)$$

Using (16) and (17) we have

$$\begin{aligned}\|\Delta_{\tilde{C}_b^n} f^b\|_{\infty} &\leq 2\gamma_d^2 \max(f_x^b, f_y^b, f_z^b) \\ \|\Gamma^n \tilde{a}^b\|_{\infty} &\leq 2\gamma_d \max(b_x^a, b_y^a, b_z^a), \\ \|\tilde{V}^n \Delta_{\omega}\|_{\infty} &\leq \tilde{v} \left(\frac{\tilde{v} \tan L}{r_0} + \frac{\varepsilon^2 v L \tan L}{r_0} + 2\omega_{ie} \tilde{L} \right), \\ \|V^n \Delta_{\tilde{\omega}}\|_{\infty} &\leq v \left(\frac{\tilde{v} \varepsilon^2 \tan L}{r_0} + \frac{v \tilde{h} L \tan L}{r_0^2} + 2\omega_{ie} \tilde{L} \right),\end{aligned}\quad (21)$$

with $v = \max(v_n, v_e)$. Using values from Table I we obtain

$$\begin{aligned}\frac{\|\Delta_{\tilde{v}_{n,e}}\|_{\infty}}{|[Ax]_{\tilde{v}_{n,e}}|} &\leq \frac{(40 + 10 + 4 + 2) \cdot 10^{-6}}{(10000 + 5000 + 300 + 100) \cdot 10^{-6}} \\ &\leq 4 \cdot 10^{-3}.\end{aligned}\quad (22)$$

Finally, using (84) and (18) it is straightforward to show [14] that the linearized equation for γ^n is given by

$$\dot{\gamma}^n = C_b^n \tilde{\omega}_{ib}^b - \tilde{\omega}_{in}^n - \Omega_{in}^n \gamma^n + \Delta_{\gamma^n}, \quad (23)$$

$$\|\Delta_{\gamma^n}\|_\infty \leq 2\gamma_d^2 \left(\frac{v \tan L}{r_0} + \omega_{ie} \right),$$

where $v = \max(v_n, v_e)$ and $\tilde{v} = \max(\tilde{v}_n, \tilde{v}_e)$. The rest term Δ_{γ^n} consists of second and higher order terms of γ^n . The error is therefore upper limited by

$$\frac{\|\Delta_{\gamma^n}\|_\infty}{|[Ax]_{\Delta_{\gamma^n}}|} \leq \frac{0.4 \cdot 10^{-9}}{(1000 + 500 + 100) \cdot 10^{-9}} < 3 \cdot 10^{-4}. \quad (24)$$

The error characteristics for the accelerometers and rate gyros are in general involved, see e.g. [15]. The easiest, but for the application adequate, way to model the accelerometer errors is to use a slowly varying offset and white noise. Normally there also exist offsets in the rate gyros, but these are for the application here considered small and therefore neglected. Note however, the algorithm is readily modifiable to include the influence of gyro drift in cases where the gyro offset is larger or the time interval is longer. The accelerometer offset, or bias, can with good accuracy be modelled as a first order Gauss-Markov process,

$$\dot{b}^a = -\frac{1}{\tau} b^a + u^b \approx u^b. \quad (25)$$

The last approximation is valid because the time constant τ is usually rather large. We will here incorporate accelerometer biases acting only in the x - and y -directions in body frame. This is an implication from the assumption that the INS altitude error follows the pressure altitude, meaning that any z -accelerometer bias is compensated for through the stabilization of the vertical channel. This is true for situations where roll and pitch angles are close to zero. During a turn, climb and/or dive the altitude filter is effected by x - and/or y -accelerometer biases. Here we assume turns, climbs and/or dives are rare such that we can neglect this effect.

To be able to apply our discrete time filter (6) has to be discretized

$$x_{t+1} = F_t x_t + G_t u_t. \quad (26)$$

For a small sampling period T_s the Euler approximation provides

$$\begin{aligned} F_t &= I + T_s A_t, \\ G_t &= T_s \left(I + \frac{T_s}{2} A_t \right) B_t, \\ \mathbb{E}[u_t u_t^T] &= Q_t / T_s, \end{aligned} \quad (27)$$

where $A(t) = A_t$, $B(t) = B_t$ and $\mathbb{E}[u(t)u^T(t)] = Q(t) = Q_t$ are considered constant during the sampling period.

IV. TERRAIN-AIDED POSITIONING

The idea behind terrain-aided positioning is to use the terrain height profile, obtained by projecting the path of the aircraft onto the ground. The INS computed altitude provides a measurement of altitude above mean-sea level. At the same time the ground clearance, i.e. the distance between the aircraft and the ground, is measured using a radar altimeter.

The difference between these two measurements provides a measurement on the terrain height at the location where the measurement was performed. A number of such measurements build up a measured terrain height profile. The aircraft carries a terrain elevation database where the terrain height is stored as a function of sampled horizontal position. The measured terrain height profile is compared with all possible profiles obtained from the database. The database profile which resembles the measured profile the most is selected, and thereby determines the aircraft's position.

The equation for terrain-aided positioning is

$$y_t = h_t^{\text{ins}} - h_t^{\text{ra}} = h(L_t, l_t) - \tilde{h}_t + e_t, \quad (28)$$

where y_t is measured terrain height and $h(\cdot)$ is the terrain height given by the database as a function of horizontal position, i.e. latitude and longitude. The term \tilde{h}_t is the INS altitude error. Moreover, e_t is the measurement noise, having a probability density which here is given by

$$p(e_t) = \sum_{\lambda_t=1}^2 \Pr(\lambda_t) \mathcal{N}(m_t^{(\lambda_t)}, R_t^{(\lambda_t)}), \quad (29)$$

i.e. a Gaussian mixture with two modes. The first mode ($\lambda_t = 1$) represents the case where the radar altimeter beam hits the ground and thereby reflecting the true ground clearance. The second mode ($\lambda_t = 2$) models the case where the beam hits a tree top, giving a measurement of the ground clearance which is too small. The probability for each of the two events is $\Pr(\lambda_t = 1)$ and $\Pr(\lambda_t = 2)$ respectively. The radar altimeter is a pulsed system operating at 4.3 GHz which makes it sensitive to reflections from e.g. trees. Together with a wide beam lobe (≈ 50 deg) the radar altimeter normally measures the closest distance to the ground or any obstacle, even during moderate roll and pitch angles. To eliminate roll and pitch dependent errors the measurements from the radar altimeter are not used when roll or pitch angle is larger than 25 deg.

A simple way of avoiding the mode dependent error characteristics is to approximate the probability density in (29) with a single Gaussian. However, it is shown in [16] that the gain when taking advantage of multi-modal characteristics can be significant. Simulations show that e.g. horizontal position is estimated with approximately 70% better accuracy.

V. THE MARGINALIZED PARTICLE FILTER

The main idea of the particle filter is to discretize the posterior probability density for the state x_t according to

$$p(x_t | Y_t) \approx \sum_{i=1}^N \bar{w}_t^{(i)} \delta_{x_t^{(i)}}(x_t), \quad (30)$$

where δ is the delta-Dirac function and $Y_t = \{y_0, \dots, y_t\}$ is the stacked vector of measurements. The weights $\bar{w}_t^{(i)}$, where $\sum_{i=1}^N \bar{w}_t^{(i)} = 1$, together with the particles $x_t^{(i)}$ are such that they together yield a set of samples approximately drawn from the posterior probability density. Theoretically we can solve almost any estimation problem using the particle filter, as long as the number of particles N is high enough.

In many cases the underlying motion model has structures which can be exploited for the purpose of decreasing N and thereby the computational load. Consider a state-space model which can be written on the form

$$x_{t+1}^n = f_t^n(x_t^n) + F_t^n x_t^1 + G_t^n u_t^n, \quad (31a)$$

$$x_{t+1}^d = f_t^d(x_t^n) + F_t^d x_t^d + G_t^d u_t^d, \quad (31b)$$

$$x_{t+1}^l = f_t^l(x_t^n) + F_t^l x_t^1 + G_t^l u_t^l, \quad (31c)$$

$$y_t = h_t(x_t^n) + H_t x_t^d + e_t(\lambda_t), \quad (31d)$$

where $x_t = [(x_t^n)^T \ (x_t^d)^T \ (x_t^l)^T]^T$. The superscripts n, d and l denote which part of the state vector has a nonlinear, discrete and linear structure respectively. Note that spaces in (31) are used to emphasize what parts of the state vector are affected by other parts and the measurements. This is important for the results derived below. Assume that the process noise is Gaussian distributed according to

$$u_t = \begin{bmatrix} u_t^n \\ u_t^d \\ u_t^l \end{bmatrix} \sim \mathcal{N}(0, Q_t), \quad Q_t = \begin{bmatrix} Q_t^n & 0 & 0 \\ 0 & Q_t^d & 0 \\ 0 & 0 & Q_t^l \end{bmatrix}. \quad (32)$$

See [1] on how to deal with a mutually correlated process noise. Also assume that x_0^d and x_0^l are Gaussian distributed i.e.

$$x_0^d \sim \mathcal{N}(0, P_0^d), \quad x_0^l \sim \mathcal{N}(0, P_0^l). \quad (33)$$

The measurement noise e_t is a sum of M Gaussians according to

$$e_t \sim \sum_{\lambda_t=1}^M \Pr(\lambda_t) \mathcal{N}(m_t^{(\lambda_t)}, R_t^{(\lambda_t)}), \quad (34)$$

with mode transition probabilities

$$\pi_{\lambda_{t-1}}^{\lambda_t} = \Pr(\lambda_t | \lambda_{t-1}), \quad \lambda_t, \lambda_{t-1} = 1, \dots, M. \quad (35)$$

The aim is to recursively estimate the probability density function (pdf) for x_t given all available measurements Y_t . The pdf is then used to compute an estimate of x_t , here the mean value, and the corresponding covariance of the estimate. The direct approach is to apply the particle filter. However, for the class of systems described by (31) there exists a more efficient way. Consider the probability density $p(x_t^1, x_t^d, X_t^n | Y_t)$, where $X_t^n = \{x_0^n, \dots, x_t^n\}$ is the stacked vector of state history. This pdf can be factorized using Bayes' rule according to

$$\begin{aligned} p(x_t^1, x_t^d, X_t^n | Y_t) &= \\ p(x_t^1 | x_t^d, X_t^n, Y_t) p(x_t^d | X_t^n, Y_t) p(X_t^n | Y_t) &= \\ p(x_t^1 | X_t^n, Y_t) p(x_t^d | X_t^n, Y_t) p(X_t^n | Y_t). \end{aligned} \quad (36)$$

Assume for now that we have an estimate based on the particle filter of $p(X_t^n | Y_t)$ according to

$$p(X_t^n | Y_t) \approx \sum_{i=1}^N \bar{w}_t^{(i)} \delta_{X_t^{n,(i)}}(X_t^n). \quad (37)$$

From (37) we have the probability density

$$p(x_t^n | Y_t) \approx \sum_{i=1}^N \bar{w}_t^{(i)} \delta_{x_t^{n,(i)}}(x_t^n), \quad (38)$$

by extracting x_t^n from X_t^n . Moreover, combining (36) and (37) we have estimates of

$$\begin{aligned} p(x_t^1 | Y_t) &= \int p(x_t^1 | X_t^n, Y_t) p(X_t^n | Y_t) dX_t^n \\ &\approx \sum_{i=1}^N \bar{w}_t^{(i)} p(x_t^1 | X_t^{n,(i)}, Y_t), \\ p(x_t^d | Y_t) &= \int p(x_t^d | X_t^n, Y_t) p(X_t^n | Y_t) dX_t^n \\ &\approx \sum_{i=1}^N \bar{w}_t^{(i)} p(x_t^d | X_t^{n,(i)}, Y_t). \end{aligned} \quad (39)$$

In the forthcoming three Lemmas we derive expressions for how to recursively compute $p(X_t^n | Y_t)$, $p(x_t^1 | X_t^n, Y_t)$ and $p(x_t^d | X_t^n, Y_t)$, i.e.

Lemma 1: $p(x_t^1 | X_t^n, Y_t)$ by the Kalman filter,

Lemma 2: $p(x_t^d | X_t^n, Y_t)$ by a bank of M^{t+1} Kalman filters,

Lemma 3: $p(X_t^n | Y_t)$ by the particle filter.

The recursions are such that we do not need knowledge of the state history X_t^n but only x_{t-1}^n and x_t^n . The derived expressions are then used together with (38) and (39) to obtain estimates of the posterior pdfs of x_t^1 , x_t^d and x_t^n .

Lemma 1 (Conditionally linear single Gaussian) For the state-space model (31), with the assumptions according to (32)–(33), we have that

$$p(x_t^1 | X_t^n, Y_t) = \mathcal{N}(\hat{x}_{t|t}^1, P_{t|t}^1), \quad (40a)$$

$$p(x_{t+1}^1 | X_{t+1}^n, Y_t) = \mathcal{N}(\hat{x}_{t+1|t}^1, P_{t+1|t}^1), \quad (40b)$$

where

$$\hat{x}_{t|t}^1 = \hat{x}_{t|t-1}^1, \quad P_{t|t}^1 = P_{t|t-1}^1, \quad (41)$$

and

$$\begin{aligned} \hat{x}_{t+1|t}^1 &= (F_{p,t}^1 - K_{p,t} F_{p,t}^n) \hat{x}_{t|t}^1 + \\ &\quad K_{p,t} (x_{t+1}^n - f_t^n(x_t^n)) + f_t^1(x_t^n), \\ P_{t+1|t}^1 &= F_{p,t}^1 P_{t|t}^1 (F_{p,t}^1)^T + G_{p,t}^1 Q_t^1 (G_{p,t}^1)^T - K_{p,t} S_{p,t} K_{p,t}^T, \\ K_{p,t} &= F_{p,t}^1 P_{t|t}^1 (F_{p,t}^1)^T S_{p,t}^{-1}, \\ S_{p,t} &= G_{p,t}^1 Q_t^1 (G_{p,t}^1)^T + F_{p,t}^1 P_{t|t}^1 (F_{p,t}^1)^T. \end{aligned} \quad (42)$$

Proof: Conditionally upon X_t^n , x_t^1 is independent of Y_t and thereby unaffected by the multi-modal noise e_t given by (34). The result then follows immediately from [1]. ■

In practice the above means that we can estimate $p(x_t^1 | Y_t)$ as the weighted sum of N Kalman filters applied to each sequence of $\{X_t^{n,(i)}\}_{i=1}^N$.

Lemma 2 (Conditionally linear multi-modal Gaussian)

For the state-space model (31), with the assumptions according to (32)–(35) and $\Lambda_t = \{\lambda_0, \dots, \lambda_t\}$, we have that

$$p(x_t^d | X_t^n, Y_t) = \sum_{\Lambda_t} \bar{\alpha}_t^{(\Lambda_t)} \mathcal{N}(\hat{x}_{t|t}^{d,(\Lambda_t)}, P_{t|t}^{d,(\Lambda_t)}), \quad (43a)$$

$$p(x_{t+1}^d | X_t^n, Y_t) = \sum_{\Lambda_t} \bar{\alpha}_t^{(\Lambda_t)} \mathcal{N}(\hat{x}_{t+1|t}^{d,(\Lambda_t)}, P_{t+1|t}^{d,(\Lambda_t)}), \quad (43b)$$

where

$$\hat{x}_{t|t}^{d,(\Lambda_t)} = \hat{x}_{t|t-1}^{d,(\Lambda_{t-1})} + P_{t|t-1}^{d,(\Lambda_{t-1})} H_t^T (S_{f,t}^{(\Lambda_t)})^{-1} (y_t - \hat{y}_{t|t-1}^{(\Lambda_t)}), \quad (44a)$$

$$P_{t|t}^{d,(\Lambda_t)} = P_{t|t-1}^{d,(\Lambda_{t-1})} - P_{t|t-1}^{d,(\Lambda_{t-1})} H_t^T (S_{f,t}^{(\Lambda_t)})^{-1} H_t (P_{t|t-1}^{d,(\Lambda_{t-1})})^T, \quad (44b)$$

$$\hat{y}_{t|t-1}^{(\Lambda_t)} = h_t(x_t^n) - H_t \hat{x}_{t|t-1}^{d,(\Lambda_{t-1})} - m_t^{(\lambda_t)}, \quad (44c)$$

$$S_{f,t}^{(\Lambda_t)} = R_t^{(\lambda_t)} + H_t P_{t|t-1}^{d,(\Lambda_{t-1})} H_t^T, \quad (44d)$$

$$\begin{aligned} \hat{x}_{t+1|t}^{d,(\Lambda_t)} &= f_t^d(x_t^n) + F_t^d \hat{x}_{t|t}^{d,(\Lambda_t)}, \\ P_{t+1|t}^{d,(\Lambda_t)} &= F_t^d P_{t|t}^{d,(\Lambda_t)} (F_t^d)^T + G_t^d Q_t^d (G_t^d)^T, \end{aligned} \quad (45)$$

and

$$\begin{aligned} \alpha_t^{(\Lambda_t)} &= \mathcal{N}(\hat{y}_{t|t-1}^{(\Lambda_t)}, S_{f,t}^{(\Lambda_t)}) \pi_{\lambda_{t-1}}^{\lambda_t} \bar{\alpha}_{t-1}^{(\Lambda_{t-1})}, \\ \bar{\alpha}_t^{(\Lambda_t)} &= \frac{\alpha_t^{(\Lambda_t)}}{\sum_{\Lambda_t} \alpha_t^{(\Lambda_t)}} \end{aligned} \quad (46)$$

Proof: The probability $p(x_t^d | X_t^n, Y_t)$ can be written according to

$$p(x_t^d | X_t^n, Y_t) = \sum_{\Lambda_t} p(x_t^d | X_t^n, Y_t, \Lambda_t) \Pr(\Lambda_t | X_t^n, Y_t). \quad (47)$$

Conditionally upon λ_t , e_t is a single Gaussian. Together with $\bar{\alpha}_t^{(\Lambda_t)} = \Pr(\Lambda_t | X_t^n, Y_t)$ the result for (43)–(45) follows from [1]. Using Bayes' rule repeatedly and the principle of induction, the probability $\Pr(\Lambda_t | X_t^n, Y_t)$ is recursively given by

$$\begin{aligned} \Pr(\Lambda_t | X_t^n, Y_t) &= \frac{p(y_t | X_t^n, Y_{t-1}, \Lambda_t) \Pr(\Lambda_t | X_{t-1}^n, Y_{t-1})}{p(y_t | X_t^n, Y_{t-1})} = \\ &= \frac{p(y_t | X_t^n, Y_{t-1}, \Lambda_t) \Pr(\lambda_t | \lambda_{t-1}) \Pr(\Lambda_{t-1} | X_{t-1}^n, Y_{t-1})}{\sum_{\Lambda_t} p(y_t | X_t^n, Y_{t-1}, \Lambda_t) \Pr(\lambda_t | \lambda_{t-1}) \Pr(\Lambda_{t-1} | X_{t-1}^n, Y_{t-1})} = \\ &= \frac{p(y_t | X_t^n, Y_{t-1}, \Lambda_t) \Pr(\lambda_t | \lambda_{t-1}) \bar{\alpha}_{t-1}^{(\Lambda_{t-1})}}{\sum_{\Lambda_t} p(y_t | X_t^n, Y_{t-1}, \Lambda_t) \Pr(\lambda_t | \lambda_{t-1}) \bar{\alpha}_{t-1}^{(\Lambda_{t-1})}}. \end{aligned} \quad (48)$$

From [1] we know that $p(y_t | X_t^n, Y_{t-1}, \Lambda_t) = \mathcal{N}(\hat{y}_{t|t-1}^{(\Lambda_t)}, S_{f,t}^{(\Lambda_t)})$. Together with the mode transition probability $\pi_{\lambda_{t-1}}^{\lambda_t} = \Pr(\lambda_t | \lambda_{t-1})$ the formulas in (46) follows. ■

In practice this means that we have to apply one Kalman filter for each sequence of particles and each sequence of modes. The number of possible mode sequences increases exponentially with time and must somehow be limited. One way is to include the estimate of the mode sequence in the particle filter, which automatically limits the number such that only the most probable mode sequences survive. Another way is to merge mode sequences which are identical from $t - L$ up to and including t , so as to keep the number constant ($= M^L$), using e.g. the generalized pseudo-Bayesian (GPB) or interacting multiple model (IMM) filter [17], [18], [19].

Lemma 3 (Gaussian distributed likelihood and prior)

The probability $p(X_t^n | Y_t)$ is recursively given by

$$p(X_t^n | Y_t) = \frac{p(y_t | X_t^n, Y_{t-1}) p(x_t^n | X_{t-1}^n, Y_{t-1})}{p(y_t | Y_{t-1})} p(X_{t-1}^n | Y_{t-1}). \quad (49)$$

For the state-space model (31), with the assumptions according to (32)–(35), we have that

$$p(y_t | X_t^n, Y_{t-1}) = \sum_{\Lambda_t} \alpha_t^{(\Lambda_t)}, \quad (50a)$$

$$p(x_{t+1}^n | X_t^n, Y_t) = \mathcal{N}(\hat{x}_{t+1|t}^n, P_{t+1|t}^n), \quad (50b)$$

where

$$\begin{aligned} \hat{x}_{t+1|t}^n &= f_t^n(x_t^n) + F_t^n \hat{x}_{t|t}^1, \\ P_{t+1|t}^n &= F_t^n P_{t|t}^1 (F_t^n)^T + G_t^n Q_t^n (G_t^n)^T, \end{aligned} \quad (51)$$

with $\alpha_t^{(\Lambda_t)}$ given by (46).

Proof: Expression (49) is given by repeated use of Bayes' rule. For $p(x_{t+1}^n | X_t^n, Y_t)$ see [1]. For $p(y_t | X_t^n, Y_{t-1})$ rewrite it according to

$$\begin{aligned} p(y_t | X_t^n, Y_{t-1}) &= \sum_{\Lambda_t} p(y_t | X_t^n, Y_{t-1}, \Lambda_t) \Pr(\Lambda_t | X_{t-1}^n, Y_{t-1}) = \\ &= \sum_{\Lambda_t} \mathcal{N}(\hat{y}_{t|t-1}^{(\Lambda_t)}, S_{f,t}^{(\Lambda_t)}) \pi_{\lambda_{t-1}}^{\lambda_t} \bar{\alpha}_{t-1}^{(\Lambda_{t-1})} = \\ &= \sum_{\Lambda_t} \alpha_t^{(\Lambda_t)}, \end{aligned} \quad (52)$$

where the last step is given by (46). ■

For the particle filter algorithm, we can choose to use $p(x_t^n | X_{t-1}^{n,(i)}, Y_{t-1})$ to update the samples, i.e.

$$x_t^{n,(i)} \sim p(x_t^n | X_{t-1}^{n,(i)}, Y_{t-1}), \quad (53)$$

knowing that this is a Gaussian density and thereby easy to sample from. The weights are then calculated according to

$$\begin{aligned} w_t^{(i)} &= \bar{w}_{t-1}^{(i)} p(y_t^n | X_{t-1}^{n,(i)}, Y_{t-1}) = \bar{w}_{t-1}^{(i)} \sum_{\Lambda_t} \alpha_t^{(i, \Lambda_t)}, \\ \bar{w}_t^{(i)} &= \frac{w_t^{(i)}}{\sum_k w_t^{(k)}}, \end{aligned} \quad (54)$$

which together with (53) yield (38).

A very important special case of (31) is when the matrices F_t^n , G_t^n , F_t^d , G_t^d , F_t^1 , G_t^1 and H_t are independent of x_t^n . In this case one can deduce from Lemma 1 and 2 that the Kalman filter covariance matrices are

$$P_{t|t}^{1,(i)} = P_{t|t}^1 \text{ and } P_{t|t}^{d,(i, \Lambda_t)} = P_{t|t}^{d,(\Lambda_t)}, \quad (55)$$

i.e. independent of $X_t^{n,(i)}$, and at each time t we only have to update it once for x_t^1 and Λ_t times (each mode sequence) for $x_t^{d,(\Lambda_t)}$. This implies that, for a given number of samples N , the computational load for the marginalized particle filter is approximately the same as for the stand-alone particle filter. In this case, given that the number of samples needed for MPF is significantly lower than for the particle filter, the gain with respect to computational load can be substantial.

VI. BLENDED INS/TAP USING MPF

A. The applied algorithm

For the purpose of applying the marginalized particle filter we separate the position and altitude states from the others in (8) according to

$$\begin{aligned} x_t^n &= [\tilde{L}_t \quad \tilde{l}_t]^T, \\ x_t^d &= \tilde{h}_t, \\ x_t^l &= [\tilde{v}_{n,t} \quad \tilde{v}_{e,t} \quad \gamma_{n,t} \quad \gamma_{e,t} \quad \gamma_{d,t} \quad b_{x,t}^a \quad b_{y,t}^a]^T. \end{aligned} \quad (56)$$

Using the system equations derived in Sections III and IV, the discrete state propagation and measurement equations become

$$\begin{bmatrix} x_{t+1}^n \\ x_{t+1}^d \\ x_{t+1}^l \end{bmatrix} = F_t \begin{bmatrix} x_t^n \\ x_t^d \\ x_t^l \end{bmatrix} + G_t \begin{bmatrix} u_t^d \\ u_t^l \end{bmatrix}. \quad (57a)$$

$$y_t = h \left(\begin{bmatrix} L_t^{\text{ins}} \\ l_t^{\text{ins}} \end{bmatrix} + x_t^n \right) - x_t^d + e_t, \quad (57b)$$

where $h(\cdot)$ in (57b) is the terrain database height with input arguments latitude $L_t = L_t^{\text{ins}} + \tilde{L}_t$ and longitude $l_t = l_t^{\text{ins}} + \tilde{l}_t$ and

$$\begin{aligned} F_t &= I + T_s A_t = \begin{bmatrix} F_{n,t}^n & 0_{2 \times 1} & F_{l,t}^n \\ 0_{1 \times 2} & 1 & 0_{1 \times 7} \\ F_{n,t}^l & 0_{1 \times 7} & F_{l,t}^l \end{bmatrix}, \\ G_t &= T_s \left(I + \frac{T_s}{2} A_t \right) B_t = \begin{bmatrix} 0_{2 \times 1} & G_t^n \\ T_s \left(1 + \frac{T_s}{2} \right) & 0_{1 \times 7} \\ 0_{7 \times 1} & G_t^l \end{bmatrix}, \\ u_t^d &= u_t^h, \\ u_t^l &= [u_{x,t}^a \quad u_{y,t}^a \quad u_{x,t}^\gamma \quad u_{y,t}^\gamma \quad u_{z,t}^\gamma \quad u_{x,t}^b \quad u_{y,t}^b]^T. \end{aligned} \quad (58)$$

Note that the state propagation model in (57a) is linear as opposed to the more general nonlinear model used in Section V.

To only have to compute one covariance matrix P_t^l , the matrices $F_{l,t}^l$, $F_{l,t}^n$, G_t^n and G_t^l must all be independent of x_t^n . This is achieved by not compensating INS computed quantities with estimated errors before entering F_t , i.e. $x_t = x_t^{\text{ins}} + \tilde{x}_t \approx x_t^{\text{ins}}$. An alternative is to compensate using the MPF estimates, meaning that we use the same compensation for all $i = 1, \dots, N$. The second alternative should be better if the INS errors are large, but for simplicity the first alternative is chosen here.

For the altitude error $x_t^d = \tilde{h}_t$ we choose to estimate it using the GPB filter. This means that we use two Kalman filters, each one conditioned on one of the modes in (29). For each time t the number of modes is always two. The recursions are then

given by

$$\begin{aligned} \hat{x}_{t|t}^{d,(i,\lambda_t)} &= \hat{x}_{t|t-1}^{d,(i)} + P_{t|t-1}^{d,(i)} H_t^T (S_{f,t}^{(i,\lambda_t)})^{-1} (y_t - \hat{y}_{t|t-1}^{(i,\lambda_t)}), \\ P_{t|t}^{d,(i,\lambda_t)} &= P_{t|t-1}^{d,(i)} - P_{t|t-1}^{d,(i)} H_t^T (S_{f,t}^{(i,\lambda_t)})^{-1} H_t (P_{t|t-1}^{d,(i)})^T, \\ \alpha_t^{(i,\lambda_t)} &= \mathcal{N}(\hat{y}_{t|t-1}^{(i,\lambda_t)}, S_{f,t}^{(i,\lambda_t)}) \sum_{\lambda_{t-1}=1}^2 \pi_{\lambda_{t-1}}^{\lambda_t} \bar{\alpha}_{t-1}^{(\lambda_{t-1})}, \\ \hat{y}_{t|t-1}^{(i,\lambda_t)} &= h(x_t^{n,(i)} - H_t \hat{x}_{t|t-1}^{d,(i)} - m_t^{(\lambda_t)}), \\ S_{f,t}^{(i,\lambda_t)} &= R_t^{(\lambda_t)} + H_t P_{t|t-1}^{d,(i)} H_t^T. \end{aligned} \quad (59)$$

To keep the number of mode sequences constant the result from the two Kalman filters are merged, using $\bar{\alpha}_t^{(i,\lambda_t)} = \alpha_t^{(i,\lambda_t)} / \sum_{\lambda_t=1}^2 \alpha_t^{(i,\lambda_t)}$, according to

$$\begin{aligned} \hat{x}_{t|t}^{d,(i)} &= \sum_{\lambda_t=1}^2 \bar{\alpha}_t^{(i,\lambda_t)} \hat{x}_{t|t}^{d,(i,\lambda_t)}, \\ P_{t|t}^{d,(i)} &= \sum_{\lambda_t=1}^2 \bar{\alpha}_t^{(i,\lambda_t)} (P_{t|t}^{d,(i,\lambda_t)} + (\hat{x}_{t|t}^{d,(i,\lambda_t)} - \hat{x}_{t|t}^{d,(i)})^2). \end{aligned} \quad (60)$$

We will add some artificial process noise u_t^{add} for the latitude and longitude error states, to deal with particle filter discretization errors and to further decrease the number of needed particles. This will change the state propagation equation for horizontal position in (57a) to

$$x_{t+1}^n = F_{n,t}^n x_t^n + F_{l,t}^n x_t^l + \underbrace{G_t^n u_t^l + u_t^{\text{add}}}_{u_t^n}, \quad (61)$$

which should be compared to the propagation equation for x_t^l , i.e.

$$x_{t+1}^l = F_{n,t}^l x_t^n + F_{l,t}^l x_t^l + G_t^l u_t^l. \quad (62)$$

The process noises u_t^n and u_t^l are mutually correlated. On the other hand, here $Q_t^{\text{add}} \gg Q_t^l$ which yields $u_t^n \approx u_t^{\text{add}}$ and the correlation is therefore neglected.

A summary of the applied algorithm is given in Algorithm 1.

Algorithm 1 The MPF for blended INS/TAP

1) Initialization:

For $i = 1, \dots, N$, sample $x_0^{n,(i)} \sim p(x_0^n)$, and set

$$\begin{aligned} \{\hat{x}_{0|-1}^{1,(i)}, P_{0|-1}^1\} &= \{0, P_0^1\}, \\ \{\hat{x}_{0|-1}^{d,(i)}, P_{0|-1}^{d,(i)}\} &= \{0, P_0^{d,(i)}\}, \\ \{\bar{\alpha}_{-1}^{(i,1)}, \bar{\alpha}_{-1}^{(i,2)}\} &= \{\Pr(\lambda_t = 1), \Pr(\lambda_t = 2)\}. \end{aligned}$$

2) GPB filter measurement update:

For $i = 1, \dots, N$ and $\lambda_t = 1, 2$, compute

$$\begin{aligned} \{\hat{x}_{t|t}^{d,(i,\lambda_t)}, P_{t|t}^{d,(i,\lambda_t)}, \alpha_t^{(i,\lambda_t)}\} &\text{ using (59),} \\ \bar{\alpha}_t^{(i,\lambda_t)} &= \alpha_t^{(i,\lambda_t)} / (\alpha_t^{(i,1)} + \alpha_t^{(i,2)}), \\ \{\hat{x}_{t|t}^{d,(i)}, P_{t|t}^{d,(i)}\} &\text{ using (60).} \end{aligned}$$

3) *Particle filter measurement update:*

For each $i = 1, \dots, N$, update

$$w_t^{(i)} = \bar{w}_{t-1}^{(i)} \sum_{\lambda_t=1}^2 \alpha^{(i, \lambda_t)}, \quad \bar{w}_t^{(i)} = w_t^{(i)} / \sum_i w_t^{(i)}.$$

4) *Resampling:*

Resample N times with replacement according to

$$\Pr(x_t^{(i)} = x_t^{(k)}) = \bar{w}_t^{(i)}.$$

5) *Kalman filter measurement update:*

For each $i = 1, \dots, N$, set

$$\hat{x}_{t|t}^{1,(i)} = \hat{x}_{t|t-1}^{1,(i)}, \quad P_{t|t}^1 = P_{t|t-1}^1.$$

6) *GPB filter time update:*

For each $i = 1, \dots, N$, compute

$$\begin{aligned} \hat{x}_{t+1|t}^{d,(i)} &= \hat{x}_{t|t}^{d,(i)}, \\ P_{t+1|t}^{d,(i)} &= P_{t|t}^{d,(i)} + T_s(1 + T_s/2)^2 Q_t^d. \end{aligned}$$

7) *Particle filter time update:*

For $i = 1, \dots, N$, sample

$$x_{t+1}^{n,(i)} \sim p(x_{t+1}^n | X_t^{n,(i)}, Y_t) \text{ using (50b).}$$

8) *Kalman filter time update:*

For each $i = 1, \dots, N$, compute

$$\{\hat{x}_{t+1|t}^{1,(i)}, P_{t+1|t}^{1,(i)}\} \text{ according to (42).}$$

B. Convergence Analysis of Algorithm 1

For the estimation of x_t^n there are to the authors knowledge not many results which can be used for convergence analysis. The results that do exist e.g. [20] are unfortunately rather conservative. Simulations indicate however that given a large enough number of samples the estimate of x_t^n does converge.

We can on the other hand analyze the behaviour of x_t^1 and x_t^d . Below we show that the estimates of both x_t^1 and x_t^d always converge, although they likely converge to something wrong if the estimate of x_t^n diverges. Rewrite the model for x_t^1 according to

$$\begin{aligned} x_{t+1}^1 &= F_{1,t}^1 x_t^1 + F_{n,t}^1 x_t^n + G_t^1 u_t^1 \\ z_t^1 &= x_{t+1}^1 - F_{n,t}^1 x_t^n = F_{1,t}^1 x_t^1 + u_t^n, \end{aligned} \quad (63)$$

The term $F_{n,t}^1 x_t^n$ in (63) can be regarded as a known input signal.

Suppose first that the aircraft is traveling without any turns, at constant speed, at the same altitude, and that the path is located around 60 degrees latitude. In this case, the eigenvalues of $F_{1,t}^1$ all lie on or slightly outside the unit-circle. For the Riccati recursion to converge a necessary condition is that the system is detectable [21]. For detectability in this case we need full observability. To investigate the observability we can use the observability matrix

$$O(t, t+k) = \begin{bmatrix} H_t \\ \vdots \\ H_{t+k-1} F_{t+k-2} \cdots F_t \end{bmatrix}, \quad (64)$$

where in our case $H_t = F_{1,t}^n$ and $F_t = F_{1,t}^1$. We know from [22] that if $\text{rank}(O(t, t+k)) = \dim(x_t)$, then the system is observable. It is straightforward to verify, under the flight conditions stated above, we actually have full observability after only four steps, i.e. $k = 4$. On the other hand, most of

the singular values of $O(t, t+k)$ are very small, and this is true for larger values on k as well.

To clarify, we can simplify the system equations further by discarding those elements which are insignificant during shorter periods of time, say one or two minutes. For these short periods of time we can neglect that the earth rotates and that the surface of the earth is curved, i.e.

$$\omega_{ie}^n \approx 0_{3 \times 1}, \quad \omega_{en}^n \approx 0_{3 \times 1}. \quad (65)$$

The simplification above means that the state transition matrix will look like

$$F_{1,t}^1 = I + T_s \begin{bmatrix} 0_{2 \times 2} & 0 & f_d & -f_e & c_{b,11}^n & c_{b,12}^n \\ -f_d & 0 & f_n & c_{b,21}^n & c_{b,22}^n & \\ 0_{5 \times 2} & & 0_{5 \times 5} & & & \end{bmatrix}. \quad (66)$$

Moreover, although the term f^n in (19) can be regarded as a known input signal (at least $f^{n,\text{ins}}$ is known), it is convenient for the analysis to rewrite it as

$$f^n = \dot{v}^n + (\Omega_{en}^n + 2\Omega_{ie}^n)v^n - g^n \approx \begin{bmatrix} \dot{v}_n \\ \dot{v}_e \\ \dot{v}_d - g_d \end{bmatrix}. \quad (67)$$

From the simplified system matrix $F_{1,t}^1$ above and the expression for the specific force f^n we can draw two conclusions. First of all, if there is no horizontal acceleration, i.e. $\dot{v}_n = \dot{v}_e = 0$, γ_d will not be observable. This is easily seen from (66), because in this case $f_n \approx f_e \approx 0$ and γ_d will thereby not have any influence on v_n or v_e , hence unobservable. Secondly, flying along a straight path means that only the sum of $-f_d \gamma_n$ and b_e^a , and $f_d \gamma_e$ and b_n^a , where

$$\begin{aligned} b_n^a &= c_{b,11}^n b_x^a + c_{b,12}^n b_y^a, \\ b_e^a &= c_{b,21}^n b_x^a + c_{b,22}^n b_y^a, \end{aligned} \quad (68)$$

are observable. We need a change in C_b^n to be able estimate the individual components in the two sums.

The detectability criteria is only a necessary condition for the Riccati recursion to converge. A necessary and sufficient condition is to also require that the system is unit-circle controllable [21]. Here, it is straightforward to verify, by inspection of G_t^k , that the system is actually controllable, and thereby also unit-circle controllable.

The same reasoning as for x_t^1 applies to x_t^d . Rewrite the model for x_t^d according to

$$\begin{aligned} x_{t+1}^d &= x_t^d + G_t^d u_t^d \\ z_t^d &= y_t - h \left(\begin{bmatrix} L_t^{\text{ins}} \\ j_t^{\text{ins}} \end{bmatrix} + x_t^n \right) = -x_t^d + e_t(\lambda_t). \end{aligned} \quad (69)$$

It is obvious that the model is both observable and controllable thereby providing sufficient conditions for the Riccati equation to converge. The mode variable λ_t could possibly cause the estimate to converge to something wrong, but simulations show that this is highly unlikely.

VII. SIMULATION RESULTS

In this section we apply the marginalized particle filter according to Algorithm 1 on simulated inertial navigation data. Terrain elevation data is taken from a commercial database, which contains terrain elevation at discrete points separated with 50 metres in both north and east directions. Elevation data at intermediate points is computed using bilinear interpolation. The flight trajectory projected onto the ground is depicted in Figure 2.

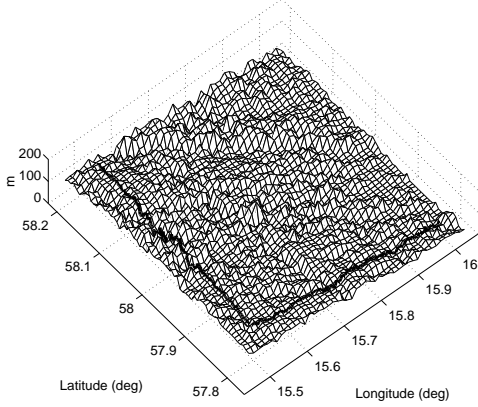


Fig. 2. Terrain elevation profile along the flight trajectory.

As can be deduced from Figure 2 the flight trajectory makes a turn after about half of the distance. The main reason for this turn is to make γ_n and γ_e distinguishable from b_x^a and b_y^a . The measurements are assumed unavailable during the turn, to imitate the fact that the radar altimeter provides poor ground clearance measurements when the absolute value of the bank angle $|\phi|$ is large. The bank angle during the turn is 60 deg. Moreover, to make γ_d observable, the speed along the path changes from time to time according to Figure 3. Note that the turn and speed changes are used to make attitude, heading and accelerometer biases observable. Position and velocity errors are observable without accelerations meaning that the algorithm does not require accelerations for accurate position and velocity estimates.

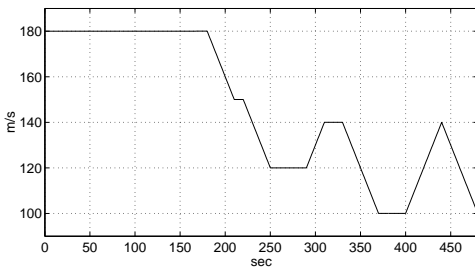


Fig. 3. Ground speed along the flight trajectory.

To simulate INS data we have used the truth flight profile given by Figure 2 and 3 and worked backwards through the nonlinear motion model given in Appendix I. Sensor errors according to Table II have then been added to the exact sensor measurements obtained from the backward propagation. Finally the sensor measurements, now with errors added, are

run through the nonlinear motion model to yield as close to authentic INS data as possible. Note that INS initial alignment is not simulated but initial errors given by Table II are used to initialize the INS computations. Simulated terrain elevation

TABLE II
SIMULATED INS ERRORS.

$p(x_0^n)$	$\mathcal{U}(-\frac{1000\sqrt{3}}{r_0}, \frac{1000\sqrt{3}}{r_0}) \cdot \mathcal{U}(-\frac{1000\sqrt{3}}{r_0 \cos L_0}, \frac{1000\sqrt{3}}{r_0 \cos L_0})$
$p(x_0^d)$	$\mathcal{N}(0, 50)$
$p(x_0^1)$	$\mathcal{N}(0, \text{diag}(1, 1, \frac{0.05\pi}{180}, \frac{0.05\pi}{180}, \frac{0.1\pi}{180}, 10^{-3} \cdot 5_{1 \times 2}))$
$p(u_t^d)$	$\mathcal{N}(0, 0.2)$
$p(u_t^1)$	$\mathcal{N}(0, \text{diag}(10^{-4}, 10^{-4}, 10^{-6} \cdot 1_{1 \times 5}))$

measurements have been created along the flight path by adding a random error defined by

$$p(e_t) = 3/4 \cdot \mathcal{N}(0, 3^2) + 1/4 \cdot \mathcal{N}(12, 6^2), \quad (70)$$

to the true terrain elevation. We have assumed mode transition probabilities for the measurement noise from (35) according to

$$\begin{bmatrix} \pi_1^1 & \pi_2^1 \\ \pi_1^2 & \pi_2^2 \end{bmatrix} = \begin{bmatrix} 3/4 & 3/4 \\ 1/4 & 1/4 \end{bmatrix}. \quad (71)$$

Note that these particular parameter values are not authentic but gives an adequate example on the distribution of the radar altimeter measurement error over dense forest. In practice the values are found empirically by comparing measurements from GPS, radar altimeter and terrain height database over different types of terrain.

For the marginalized filter we used sampling period $T_s = 1$ sec and 12000 particles ($N = 12000$). No significant improvement was obtained using more than 12000 particles. For the additional process noise we chose

$$u_t^{\text{add}} \sim \mathcal{N}(0, 2 \cdot 10^{-3} P_{t|t-1}^{\text{n,MPF}}), \quad (72)$$

applied to x_t^n according to (61). Deterministic resampling [10] were performed if $\sum_i 1/(\bar{w}_t^{(i)})^2 < 2N/3$ and at least five filter iterations have past since the last resampling.

The result (RMSE_t and $\sqrt{P_t^{\text{MEAN}}}$) based on 100 Monte Carlo simulations is depicted in Figure 4 for position ($s_L = \tilde{L}r_0$ and $s_l = \tilde{l}r_0 \cos L$) and altitude errors, Figure 5 for velocity and acceleration errors, and Figure 6 for platform orientation errors respectively. The RMSE_t and P_t^{MEAN} are computed according to

$$\text{RMSE}_t = \left(\frac{1}{100} \sum_{m=1}^{100} \|\hat{x}_{t|t-1}^{\text{MPF},(m)} - x_t^{\text{true},(m)}\|_2^2 \right)^{1/2}, \quad (73a)$$

$$P_t^{\text{MEAN}} = \frac{1}{100} \sum_{m=1}^{100} \text{tr}(P_{t|t-1}^{\text{MPF},(m)}), \quad (73b)$$

m representing the m :th Monte Carlo simulation.

The RMSE_t for horizontal position decreases to about 30 m after 40 sec, and the stationary error level lies around 20 m. The RMSE_t for the altitude converges to its stationary value, approximately 1 m, after 30 sec. The RMSE_t for horizontal velocity drops below 0.2 m/s, and the INS horizontal platform orientation error (γ_n and γ_e) drop below $0.02/\sqrt{2} \approx 0.015$ deg

for each of the two errors. The RMSE_t for b_x^a and b_x^a approach $3 \cdot 10^{-3}/\sqrt{2} \approx 2 \cdot 10^{-3} \text{ m/s}^2$, and for γ_d it approaches 0.075 deg. Note also the distinct increase of the position and velocity error and uncertainty during the turn (between time 150 to 175 sec). This is due to that no terrain elevation measurements are used during this period of time.

In the same figures the corresponding Cramer-Rao posterior prediction bounds are shown. The bound is computed according to [14]

$$\begin{aligned} P_0^{\text{CR}} &= P_0, \\ P_{t+1}^{\text{CR}} &= F_t P_t^{\text{CR}} (I - (I + \mathcal{R}_t^{-1} P_t^{\text{CR}})^{-1} \mathcal{R}_t^{-1} P_t^{\text{CR}}) F_t^T \\ &\quad + G_t Q_t G_t^T, \end{aligned} \quad (74)$$

where F_t and G_t are taken from (58), Q_t and P_0 from Table II and \mathcal{R}_t^{-1} is given by

$$\begin{aligned} \mathcal{R}_t^{-1} &= \mathbb{E}_{p(e_t)} \left[\left(\frac{d}{de_t} \log p(e_t) \right)^2 \right] \cdot \\ &\quad \mathbb{E}_{p(x_t^n)} \begin{bmatrix} \frac{\partial}{\partial x_t^n} h(x_t^n) \\ -1 \\ 0_{7 \times 1} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x_t^n} h(x_t^n) \\ -1 \\ 0_{7 \times 1} \end{bmatrix}^T. \end{aligned} \quad (75)$$

\mathcal{R}_t^{-1} is computed by evaluating the expectations in (75) using a large number of samples from $p(e_t)$ according to (70) and $p(x_0^n) \approx p(x_t^n)$ according to Table II. As can be seen from the figures, the RMSE_t of the filter estimates are all slightly larger than the corresponding P_t^{CR} , but the difference is small indicating that the applied filter is close to being optimal with respect to RMSE_t , at least after filter convergence.

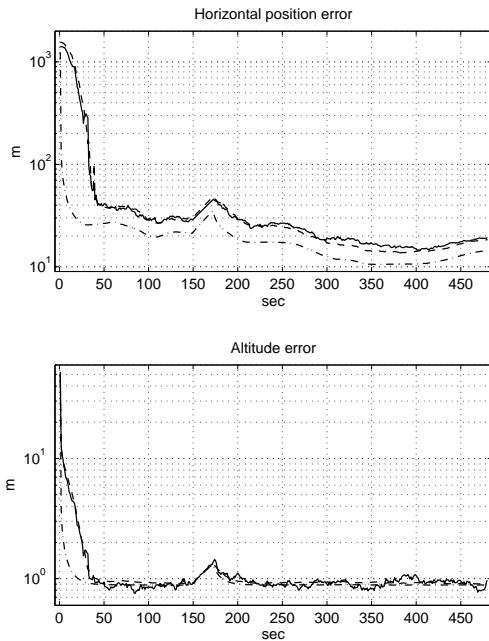


Fig. 4. RMSE_t (solid line), $\sqrt{P_t^{\text{MEAN}}}$ (dashed line) and $\sqrt{P_t^{\text{CR}}}$ (dash-dotted line) for horizontal position $\sqrt{s_L^2 + s_l^2}$ and altitude \hat{h} .

VIII. CONCLUSIONS

In this paper we have extended the MPF from [1] to account for a multi-modal measurement noise. The extended filter has

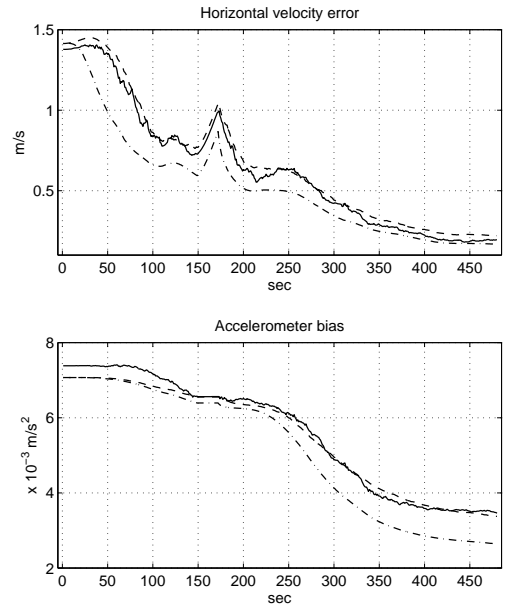


Fig. 5. RMSE_t (solid line), $\sqrt{P_t^{\text{MEAN}}}$ (dashed line) and $\sqrt{P_t^{\text{CR}}}$ (dash-dotted line) for horizontal velocity $\sqrt{\hat{v}_n^2 + \hat{v}_e^2}$ and accelerometer bias $\sqrt{(b_x^a)^2 + (b_y^a)^2}$.

been applied on a tightly blended INS/TAP navigation system. We have shown that by concentrating on the inertial navigation errors, we can linearize the state transition equations without introducing any significant errors. The MPF takes advantage of the linearized structure, and estimates it using relatively fast Kalman filters. The highly nonlinear terrain-aided positioning system only depends on position, meaning that we can focus the computer intensive particle filter on the position part of the state vector only. Compared to applying a stand-alone particle filter we can decrease the number of samples substantially and thereby making the applied MPF computationally tractable.

Simulations have been performed on simulated inertial navigation data, using a commercial terrain elevation database to simulate the terrain-aided positioning system. The simulation result is compared with the Cramer-Rao lower bound. The comparison shows that we obtain nearly optimal accuracy, at least after filter convergence. The deviation between the lower bound and the simulation result partly depends on the fact that the particle filter still only provides an approximate solution particularly due to discretization errors. Another possible contributing factor to the deviation could be that the Cramer-Rao bound is not a tight bound in this case. There could very well exist other bounds that are tighter, see e.g. [23].

APPENDIX I

Based on measured accelerations and angular rates in three dimensions the INS computes position, velocity, attitude and heading. The computations are based on an accurate nonlinear motion model describing the kinematics of the system. These equations will not be derived here, for detailed derivations see e.g. [24], [25], [26]. To be able to characterize the INS mathematically we will need a number of coordinate frames given by:

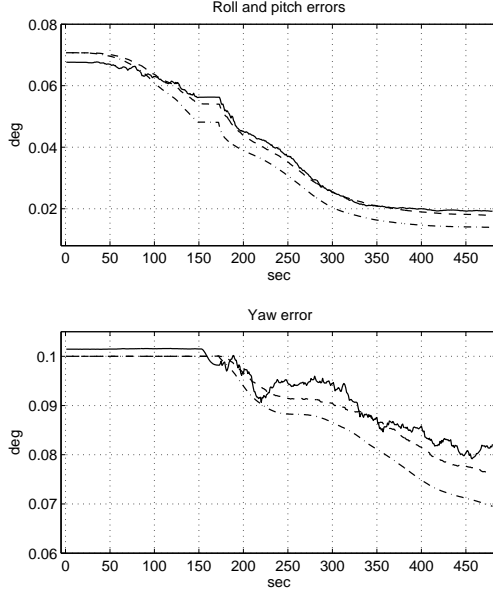


Fig. 6. RMSE_t (solid line), $\sqrt{P_t^{\text{MEAN}}}$ (dashed line) and $\sqrt{P_t^{\text{CR}}}$ (dash-dotted line) for roll and pitch errors $\sqrt{\gamma_n^2 + \gamma_e^2}$ and yaw error γ_d .

- i Inertial frame, fixed in the inertial space. For navigation periods shorter than days this frame can be approximated with an earth centered non-rotating frame.
- e Earth-centered frame, fixed to the earth, i.e it rotates with the earth.
- n Navigation frame, with its center attached to the aircraft. The x , y and z -axis are aligned with north, east and the ellipsoid normal respectively. The velocity e.g. is denoted by $v^n = [v_n, v_e, v_d]^T$.
- b Body frame, attached to the aircraft, thereby always translating and rotating with the aircraft. The x , y and z -axis points through the nose, right wing and belly respectively. The acceleration e.g. is denoted by $a^b = [a_x, a_y, a_z]^T$.

The horizontal position is usually given as two angles, latitude and longitude. Latitude refers to the angle between the normal to the reference ellipsoid and the equatorial plane, and will be denoted by L . Longitude is the angle between the same normal and a plane intersecting the Greenwich meridian, and will be denoted by l . The reference ellipsoid is defined by the World Geodetic System 1984 (WGS84), see [27] or Table III for numerical values. The equations for latitude, longitude and altitude are

$$\begin{aligned} \dot{L} &= \frac{v_n}{r_L + h}, \\ \dot{l} &= \frac{v_e}{(r_l + h) \cos L}, \\ \dot{h} &= -v_d. \end{aligned} \quad (76)$$

In (76), the two radii of curvature are given by

$$r_L = \frac{r_0(1 - \varepsilon^2)}{(1 - \varepsilon^2 \sin^2 L)^{3/2}}, \quad r_l = \frac{r_0}{(1 - \varepsilon^2 \sin^2 L)^{1/2}}, \quad (77)$$

TABLE III
PARAMETERS FOR WGS84.

Parameter	Notation	Numerical value
Semimajor axis	r_0	$6.378137 \cdot 10^6$ m
Reciprocal of flattening	$1/f$	298.2572
First eccentricity	ε	0.08181919
Angular velocity	ω_{ie}	$7.292115 \cdot 10^{-5}$ rad/sec
Gravity at equator	g_0	9.780325 m/s ²
Gravity formula constant	k	0.001931853
Gravity formula constant	m	0.003449787

where the constant ε is the earth's first eccentricity, see Table III.

The velocity of the aircraft relative to the earth, expressed in the navigation (n) frame and denoted by $v^n = [v_n \ v_e \ v_d]^T$, is given as the solution to the differential equation

$$\dot{v}^n = C_b^n f^b - (\Omega_{en}^n + 2\Omega_{ie}^n) v^n + g^n. \quad (78)$$

The vector f^b is the acceleration sensed by the accelerometers (specific force vector) and C_b^n is a transformation matrix from body frame to navigation frame. The matrices Ω_{en}^n and Ω_{ie}^n represent the rotation of the navigation frame relative to earth and earth relative to inertial frame respectively, both expressed in the n-frame. The rotation described by Ω_{en}^n arises when travelling over the curved surface of the earth. The matrices Ω_{en}^n and Ω_{ie}^n are both the skew-symmetric matrix

$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (79)$$

of the corresponding vector $\omega = [\omega_x \ \omega_y \ \omega_z]^T$. In vector form the rotations are given by

$$\omega_{ie}^n = \begin{bmatrix} \cos L \\ 0 \\ -\sin L \end{bmatrix} \omega_{ie}, \quad \omega_{en}^n = \begin{bmatrix} \frac{v_e}{r_l + h} \\ -\frac{v_n}{r_L + h} \\ -\frac{v_e \tan L}{r_l + h} \end{bmatrix}, \quad (80)$$

where ω_{ie} is a scalar representing the angular velocity of the earth. For a numerical value see Table III.

The acceleration f^b includes the effect of the gravity vector, g^n , which represents the sum of the earth's gravitation, G^n , and the centripetal acceleration due to the rotation of the earth, i.e.

$$g^n = G^n - (\Omega_{ie}^n)^2 r^n, \quad (81)$$

where r^n is the position vector of the aircraft measured from the centre of the earth. The WGS84 ellipsoid is defined in such a way that the angle between g^n and the normal to the ellipsoid is minimized. The deflection of the vertical, i.e. the remaining error angle between the ellipsoid's normal and the gravity vector, is usually less than $5 \mu\text{rad}$. Therefore, without introducing any significant errors, the gravity is approximately

given by [27]

$$\begin{aligned} g^n &\approx [0 \quad 0 \quad g_d]^T, \\ g_d &= g_0 \frac{1 + k \sin^2 L}{(1 - \varepsilon^2 \sin^2 L)^{1/2}} \cdot \\ &\quad \left(1 - \frac{2h}{r_0} (1 + f + m - 2f \sin^2 L) + \frac{3h^2}{r_0^2} \right). \end{aligned} \quad (82)$$

For numerical values on g_0 , ε , f , k , m and r_0 see Table III.

The attitude and heading of the aircraft are often represented by an orthogonal matrix C_b^n , $(C_b^n)^T C_b^n = I$, relating a vector in the body frame to a vector in the navigation frame. This matrix is referred to as a direction cosine matrix (DCM), and the coupling to the attitude and heading of the aircraft is

$$C_b^n = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix}. \quad (83)$$

In (83), ϕ , θ and ψ are the roll, pitch and heading angles, and s and c are short for sin and cos respectively. The corresponding matrix differential equation for C_b^n is given by

$$\dot{C}_b^n = C_b^n \Omega_{ib}^b - \Omega_{in}^n C_b^n, \quad (84)$$

where the skew-symmetric matrices Ω_{ib}^b and Ω_{in}^n are again given by their vector counterparts ω_{ib}^b and ω_{in}^n according to (79). The vector ω_{ib}^b corresponds to the angular rates exhibited by the body frame, expressed in the body frame, i.e. the angular rates sensed by the rate gyros. Moreover, $\omega_{in}^n = \omega_{ie}^n + \omega_{en}^n$.

APPENDIX II

$A(t)$ is given by

$$A(t) = \begin{bmatrix} A_n^n(t) & 0_{2 \times 1} & A_1^n(t) \\ 0_{1 \times 2} & 0 & 0_{1 \times 7} \\ A_n^1(t) & 0_{7 \times 1} & A_1^1(t) \end{bmatrix}$$

where

$$\begin{aligned} A_n^n(t) &= \begin{bmatrix} 0 & 0 \\ \frac{v_e \sin L}{r_0 \cos^2 L} & 0 \end{bmatrix}, \\ A_1^n(t) &= \begin{bmatrix} \frac{1}{r_0} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{r_0 \cos L} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \\ A_n^1(t) &= \begin{bmatrix} d_L^{v_n} & d_L^{v_e} & d_L^{\gamma_n} & 0 & d_L^{\gamma_d} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T, \\ A_1^1(t) &= \begin{bmatrix} d_{v_n}^{v_n} & d_{v_e}^{v_n} & 0 & f_z & -f_e & c_{b,11}^n & c_{b,12}^n \\ d_{v_e}^{v_e} & d_{v_e}^{v_e} & -f_z & 0 & f_n & c_{b,21}^n & c_{b,22}^n \\ 0 & -\frac{1}{r_0} & 0 & d_{\gamma_n}^{\gamma_n} & d_{\gamma_d}^{\gamma_n} & 0 & 0 \\ \frac{1}{r_0} & 0 & d_{\gamma_n}^{\gamma_e} & 0 & d_{\gamma_d}^{\gamma_e} & 0 & 0 \\ 0 & \frac{\tan L}{r_0} & d_{\gamma_n}^{\gamma_d} & d_{\gamma_d}^{\gamma_d} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \end{aligned}$$

and

$$\begin{aligned} d_L^{v_n} &= -\frac{v_e^2}{r_0 \cos^2 L} + 2\omega_{ie} v_e \cos L \\ d_L^{v_e} &= \frac{v_n v_e}{r_0 \cos^2 L} - 2\omega_{ie} v_d \sin L + 2\omega_{ie} v_n \cos L \\ d_L^{\gamma_n} &= \omega_{ie} \sin L \\ d_L^{\gamma_d} &= \frac{v_e}{r_0 \cos^2 L} + \omega_{ie} \cos L \\ d_{v_n}^{v_n} &= \frac{v_d}{r_0} \\ d_{v_e}^{v_n} &= -\frac{2v_e \tan L}{r_0} - 2\omega_{ie} \sin L \\ d_{v_n}^{v_e} &= \frac{v_e \tan L}{r_0} + 2\omega_{ie} \sin L \\ d_{v_e}^{v_e} &= \frac{v_n \tan L}{r_0} + \frac{v_d}{r_0} \\ d_{\gamma_n}^{\gamma_e} &= -\omega_{ie} \sin L + \frac{v_e \tan L}{r_0}, \quad d_{\gamma_e}^{\gamma_n} = -d_{\gamma_n}^{\gamma_e} \\ d_{\gamma_n}^{\gamma_d} &= \frac{v_n}{r_0}, \quad d_{\gamma_d}^{\gamma_n} = -d_{\gamma_n}^{\gamma_d} \\ d_{\gamma_e}^{\gamma_d} &= \omega_{ie} \cos L + \frac{v_e}{r_0}, \quad d_{\gamma_d}^{\gamma_e} = -d_{\gamma_e}^{\gamma_d}. \end{aligned}$$

$B(t)$ is given by

$$B(t) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_{b,11}^n & c_{b,12}^n & 0 & 0 & 0 & 0 & 0 \\ 0 & c_{b,21}^n & c_{b,22}^n & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{b,11}^n & c_{b,12}^n & c_{b,13}^n & 0 & 0 \\ 0 & 0 & 0 & c_{b,21}^n & c_{b,22}^n & c_{b,23}^n & 0 & 0 \\ 0 & 0 & 0 & c_{b,31}^n & c_{b,32}^n & c_{b,33}^n & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

REFERENCES

- [1] T. Schön, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2279–2289, July 2005.
- [2] R. G. Brown and P. Y. C. Hwang, *Introduction to random signals and applied Kalman filtering*, 3rd ed. John Wiley & Sons, 1992.
- [3] H. Carvalho, P. del Moral, A. Monin, and G. Salut, "Optimal nonlinear filtering in GPS/INS integration," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 3, pp. 835–849, July 1997.
- [4] B. Azimi-Sadjadi and P. S. Krishnaprasad, "Approximate nonlinear filtering and its application in navigation," *Automatica*, vol. 41, no. 6, June 2005.
- [5] A. J. Henley, "Terrain aided navigation - current status, techniques for flat terrain and reference data requirements," in *Proceedings of IEEE Position, Location and Navigation Symposium*, 1990.
- [6] N. Bergman, L. Ljung, and F. Gustafsson, "Terrain navigation using Bayesian statistics," *IEEE Control Systems Magazine*, vol. 19, no. 3, pp. 33–40, June 1999.
- [7] J. Dezert, "Improvement of strapdown inertial navigation using pdfaf," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 3, July 1999.
- [8] J. Kim and S. Sukkarieh, "Autonomous airborne navigation in unknown terrain environments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 3, July 2004.
- [9] P. Pucar and J. Palmqvist, "Saab NINS/NILS - an autonomous landing system for Gripen," in *Proceedings of the Position Location and Navigation Symposium, IEEE 2000*, 2000.
- [10] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo methods in practice*. Springer-Verlag, 2001.

- [11] P.-J. Nordlund and F. Gustafsson, "Sequential Monte Carlo filtering techniques applied to integrated navigation systems," in *Proceedings of the 2001 American Control Conference*, vol. 6, 2001, pp. 4375–4380.
- [12] C. Andrieu and A. Doucet, "Particle filtering for partially observed Gaussian state space models," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 64, no. 4, 2002.
- [13] J. Campbell, M. Uijt de Hagg, and F. van Graas, "Terrain-referenced positioning using airborne laser scanner," *Navigation: Journal of The Institute of Navigation*, vol. 52, no. 4, 2005.
- [14] P.-J. Nordlund, "Sequential Monte Carlo filters and integrated navigation," Licentiate thesis 945, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 2002.
- [15] A. Lawrence, *Modern Inertial Technology: Navigation, Guidance and Control*, 2nd ed. New York, USA: Springer-Verlag, 1998.
- [16] G. Hendebay and F. Gustafsson, "Fundamental filtering limitations in linear non-Gaussian systems," in *Proceedings of the 16th IFAC World Congress*, 2005.
- [17] F. Gustafsson, *Adaptive Filtering and Change Detection*, 1st ed. John Wiley & Sons Inc, 2000.
- [18] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 780–783, Aug 1988.
- [19] P.-J. Nordlund and F. Gustafsson, "Recursive estimation of three-dimensional aircraft position using terrain-aided positioning," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, 2002, pp. 1121–1124.
- [20] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Transactions on Signal Processing*, vol. 50, no. 3, March 2002.
- [21] T. Kailath, A. Sayed, and B. Hassibi, *Linear estimation*. Prentice Hall, 2001.
- [22] W. J. Rugh, *Linear System Theory*, 2nd ed. Upper Saddle River, N.J. USA: Prentice Hall, 1996.
- [23] A. J. Weiss and E. Weinstein, "A lower bound on the mean-square error in random parameter estimation," *IEEE Transactions on information theory*, vol. 31, no. 5, pp. 680–682, 1985.
- [24] K. R. Britting, *Inertial Navigation Systems Analysis*. Wiley - Interscience, 1971.
- [25] P. G. Savage, *Strapdown Inertial Navigation - Lecture Notes*, Strapdown Associates Inc., Plymouth, MN, USA, Feb 1990.
- [26] J. L. Farrell, *Integrated aircraft navigation*. Academic Press, 1976.
- [27] "Department of Defense World Geodetic System 1984," National Imagery and Mapping Agency, Tech. Rep. NIMA TR8350.2, 3rd Ed, July 1997.



Per-Johan Nordlund received his MSc degree in Electrical Engineering in 1997 and his Licentiate of Engineering degree in automatic control in 2002, both from Linköping University, Linköping, Sweden. Between 2002 and 2006 he worked as manager for the navigation department at Saab Aerosystems, Linköping, Sweden. Since Feb. 2006 he is pursuing the PhD degree at division of automatic control, Linköping University.



Fredrik Gustafsson received his MSc degree in Electrical Engineering in 1988 and his PhD degree in Automatic Control in 1992, both from Linköping University, Linköping, Sweden. He is Professor of communication systems within the department of Electrical Engineering, Linköping University. His research is focused on statistical methods in signal processing, with applications to automotive, avionics, and communication systems. Prof. Fredrik Gustafsson is an associate editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING.