

## MEMS INS/GPS Data Fusion using Particle Filter

V. Bistrovs, A. Kluga

Department of Transport Electronics and Telematics, Riga Technical University,

Lomonosova iela 1, V korpuss, LV-1019, Riga, Latvia, e-mail: bistrov@inbox.lv, ansis.kluga@rtu.lv

**crossref** <http://dx.doi.org/10.5755/j01.eee.112.6.450>

### Introduction

Targeting low cost solution for land vehicles, Micro–Electro–Mechanical Systems (MEMS) based inertial sensors are used. These sensors suffer from complex stochastic error characteristics that are difficult to model. Kalman filter (KF) has limited capabilities in providing accurate estimation of such system parameters, because KF is restricted to use only Gaussian linear models for these sensors' stochastic errors. EKF makes linearization of non–linear model, and after solve problem optimally by KF. This first order linearization in EKF introduces additional errors and difficulty in estimating process [1, 2]. It is becoming important to include elements of nonlinearity in order to model accurately the underlying dynamics of inertial system. To solve the problem of nonlinear filtering, the particle filter (PF) was proposed. It was first introduced by Gordon et al. (1993). PF exploits numerical representation techniques for approximating the filtering probability density function (PDF) of inherently nonlinear non–Gaussian systems. Using these methods, the obtained estimates can be set arbitrarily close to the optimal solution (in the Bayesian sense) at the expense of computational complexity [2, 3].

### Integration strategy and sensor error characteristics

In this paper total state system and measurement model and loosely coupled integration is used for GPS and IMU systems. The position from GPS navigation solution is input as a measurement to the integration PF filter. In a loosely coupled architecture, there is a stand-alone GPS navigation solution available in addition to the integrated solution.

MEMS sensors are cost effective with small dimensions, but its performance is seriously affected by errors such as biases, scale factor variations, drifts and sensor noise.

The measurement  $\gamma$  provided by accelerometer may be expressed in terms of the applied acceleration, acting along its sensitive axis ( $a_x$ ), bias ( $b$ ) and scale factor ( $S$ ) by the equation[3]

$$\gamma = a_x + b_{const} + b_{random} + (S_{const} + S_{random})a_x + \eta, \quad (1)$$

here  $b_{const}$  and  $b_{random}$  are deterministic and random part of accelerometer bias ( $m/s^2$ ),  $S_{const}$  and  $S_{random}$  are deterministic and random part of accelerometer scale factor,  $\eta$ – accelerometer sensor noise . Ideally, when no input is applied to the sensor, the output signal received from the sensor should read 0. However, this is not the case, and an offset called the bias exists. An uncompensated accelerometer bias introduces an error proportional to time ( $t$ ) in the velocity and proportional to  $t^2$  in the position. Moreover every time the sensor switched on, a slightly different bias or scale factor value is observed for low–cost MEMS sensors. As these errors cannot be easily separated from the sensor data, these are generally modelled as stochastic processes. The stochastic model for predicting accelerometer bias was developed basing on 1<sup>st</sup> order Gauss–Markov model. Proposed sensor bias error model is defined by the following stochastic difference equation

$$b_{k+1} = \beta \cdot b_k + \frac{1}{b_k} w_k, \quad (2)$$

here  $\beta = [0.9...1]$  is user defined value for predicted bias mean value control ,  $w_k$  – samples from white noise process. This difference equation was used at prediction step for PF.

### System and measurement model for GPS/IMU system

A state propagation model is used for a Particle Filter to predict acceleration, velocity and position referenced in the body frame, accelerometer bias and scale factor

The 1 D acceleration dynamic model is used [4] with the state vector  $\mathbf{x}_k$ , which comprises 5 states

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, w_{k-1}), \quad (3)$$

here  $f(.)$  represent system linear model,  $w$ – process noise which gathers any mismodeling effect or disturbances in the state characterization. Subscript  $k$  denotes the discrete time index

$$\mathbf{x}_k = [d_x \ v_x \ a_x \ b \ S], \quad (4)$$

here  $d_x$  – position of the vehicle along x axis in the body frame,  $v_x$  – velocity,  $a_x$  – acceleration,  $b$  – accelerometer bias,  $S$  – accelerometer scale factor.

The relation between the measurements  $\mathbf{z}_k$  and the states is modelled by following nonlinear function

$$\mathbf{z}_k = h(\mathbf{x}_k, u_k), \quad (5)$$

here  $h(\cdot)$  is a known non-linear function and  $u$  is measurement noise

$$\mathbf{z}_k = [\gamma \ d_{x\_GPS}], \quad (6)$$

here  $\gamma$  – accelerometer output signal,  $d_{x\_GPS}$  – estimation of vehicle position along x axis in the body frame based on GPS measurements. The accelerometer output signal is modelled by nonlinear measurement equation (1). Both process and measurement noise are assumed white, with known statistics and mutually independent. The initial a priori PDF  $p(\mathbf{x}_0)$  of the state vector is assumed to be known.

### Particle filter algorithm formulation

Particle filters (PF) are capable of handling highly nonlinear models with any kind of noise distribution. The particle filter is a special version of the Bayesian filter, and is based on sequential Monte Carlo (SMC) sampling.

The goal is to compute filtered estimates of  $\mathbf{x}_k$  taking into account all available measurement up to time  $k$ ,  $\mathbf{z}_{1:k}$ . From a Bayesian point of view, the solution is to recursively obtain the a posterior probability density function  $p(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})$  of states at time  $k$  given all available measurements. The PDF are constantly changing shape when receiving indirect measurements. PF represents the posterior probability density function  $p(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})$  by a set of random samples with associated weight as follows

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i), \quad (7)$$

here  $\delta(x)$  is Dirac delta function,  $w_k^i$  is associated weight with  $\mathbf{x}_k^i$ ,  $N$  is the particle number,  $\mathbf{z}_{1:k}$  denotes the measurements accumulated up to  $k$ . The weights' values are always positive  $w_k^i > 0$ , and sum of weights is equal to 1

$$\sum_{i=1}^N w_k^i = 1. \quad (8)$$

As the number of samples  $N$  becomes very large, this Monte Carlo (MC) characterization becomes an equivalent representation to the usual functional description of the posterior PDF under weak assumptions, according to the Strong Law of Large Numbers [3].

For the MC method, independent particles are required from posterior density function. In general, it is not possible to draw samples directly from posterior  $p(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})$ . Instead, the samples are drawn from a simpler distribution called the importance PDF  $q(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})$ .

Importance density function is similar to desired posterior distribution, but the weights of these generated samples need to be adjusted to represent the target function as closely as possible. [3]. If the samples  $\mathbf{x}_{0:k}^i$  were drawn from an importance density  $q(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})$  then the weights in (7) are defined by (9)

$$w_k^i \propto \frac{p(\mathbf{x}_{0:k}^i | \mathbf{z}_{1:k})}{q(\mathbf{x}_{0:k}^i | \mathbf{z}_{1:k})}, \quad (9)$$

here  $q(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})$  – importance density function.

Taking in account that conditional probability distribution of future states, given the present state and all past states, depends only upon the present state and not on any past states and the measurements are assumed to be conditionally independent given the states, the following recursive general equation for weights updating can be obtained [5]

$$\tilde{w}_k^i = w_{k-1}^i \cdot \frac{p(\mathbf{z}_k | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_{1:k})}, \quad (10)$$

here  $\tilde{w}_k^i$  – unnormalised weights,  $p(\mathbf{z}_k | \mathbf{x}_k^i)$  – likelihood function,  $p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)$  – transition prior distribution. It's necessary to make normalisation of weights in (10), in order sum of all weights will be equal to 1

$$w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^N \tilde{w}_k^i}. \quad (11)$$

The choice of importance density function  $q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_{1:k})$  is one of the most critical issues in the design of a particle filter. The reason of this is that the samples are drawn from the proposed distribution, and the proposed distribution is used to evaluate importance weights. The most popular suboptimal choice is to use the conditional prior of the state vector as the proposed distribution for importance density function

$$q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_{1:k}) = p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i). \quad (12)$$

Substituting (12) in (10) the weight's update equation is

$$\tilde{w}_k^i = w_{k-1}^i \cdot p(\mathbf{z}_k | \mathbf{x}_k^i), \quad (13)$$

here  $p(\mathbf{z}_k | \mathbf{x}_k^i)$  – likelihood function. In this case, the weights depend only on the likelihood function.

The particle filter has a problem that it degenerates quickly over time. In practical terms this means that after a certain number of recursive steps, most particles will have negligible. Degeneracy can be reduced by using a resampling step [5].

Resampling is a scheme to eliminate particles with small weights and to concentrate and replace on particles with large weights. In this paper systematic resample is used. Systematic resampling [5] is preferred resample scheme since it is simple to implement, has computational

complexity  $O(N)$  and minimizes the variance of importance weights variation. Although resampling helps to overcome the degeneracy phenomenon, it unfortunately introduces other problems. Resampling causes the samples that have high importance weights to be statistically selected many times, thus the algorithm suffers from the loss of diversity. This occurs especially in the case of small process noise since the particles at the same place cannot separate enough. It is also referred as sample impoverishment.

It is also mentioned in [5] that using the particle filter is not appropriate when the process noise is zero. In order to reduce effect of sample impoverishment, additional noise is added in the state model.

The summary of implemented particle algorithm for navigation data fusion is presented below:

- 1) Initialisation. The particle filter is initialized by drawing samples  $\mathbf{x}_0^i, i=1, \dots, N$  from the prior density function  $p(\mathbf{x}_0)$  and set weights  $w_0^i$  to  $1/N$ ,  $N$  is the number of particles.
- 2) Prediction. Particles at time step  $k-1$   $\mathbf{x}_{k-1}^i$  are passed through system model (3) to obtain the predicted particles at time step  $k$ .
- 3) Update. In the measurement update the new measurements are used to assign a probability, represented by the normalized importance weight, to each particle. The weights are calculated using (11) and (13). The normalized importance weights and the corresponding particles constitute an approximation of posterior distribution  $p(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})$ .
- 4) Systematic resampling. The resampling step will then return particles which are equally probable. Move to stage 2.

## Implementation and Results

The MTi-G – a GPS aided MEMS based inertial measurement unit was used in experiments. The MTi-G was rigidly installed inside moving vehicle. The separated raw data from accelerometer and GPS unit was used for combined processing (fusion). As the result of sensor data processing by PF, we obtain estimations of vehicle acceleration, velocity and position along x-axis in the body frame. Also some extra parameters such as bias and scale factor of accelerometer can be estimated. The algorithm is developed in such way, that during GPS outages it continues to estimate position and velocity of moving vehicle. This was implemented by estimating of two set of normalized importance weights according (13), basing on GPS measurements and IMU (accelerometer) measurements. In case of lack of GPS signal, algorithm uses set of normalized importance weights based on accelerometer measurements for system state estimation. The nonlinear measurement model  $h(\mathbf{x}_k, \mathbf{u})$  is directly used in PF, what is impossible for Kalman filter, as measurement model for KF should be linear (that is not always the case for real systems) or linearised. Also particles from uniform PDFs' for bias and scale factor were used for PF initialisation.

The evolution of approximated posterior probability density function according to (7) at different time steps is

shown in the Fig. 1. The numbers of particles  $N=100$ . From Fig. 1 it can be noticed, that approximated posterior PDF has not Gaussian form.

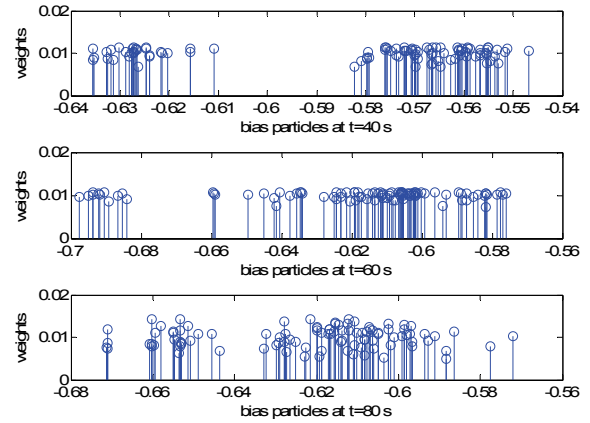


Fig. 1. Evolution of posterior PDF at  $t=40, 60, 80$  s.  $N=100$

The estimation of acceleration signal, accelerometer bias and scale factor using PF is shown in the Fig. 2.

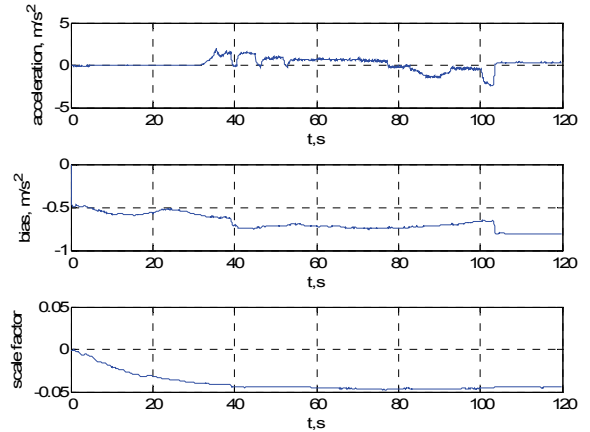


Fig. 2. IMU state parameters estimation using navigation data fusion with PF

Position estimation along x-axis (direction of the vehicle movement) in the body frame versus time  $t$  is shown in the Fig. 3. This estimation was made using combined processing of IMU and GPS data with PF. Also, only GPS based estimation of position is presented in the Fig. 3.

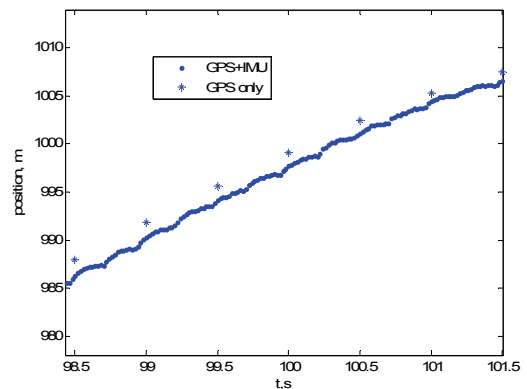


Fig. 3. Vehicle position estimation along x-axis versus time

The result of sensor data processing using PF during GPS signal outages is represented in the Fig. 4.

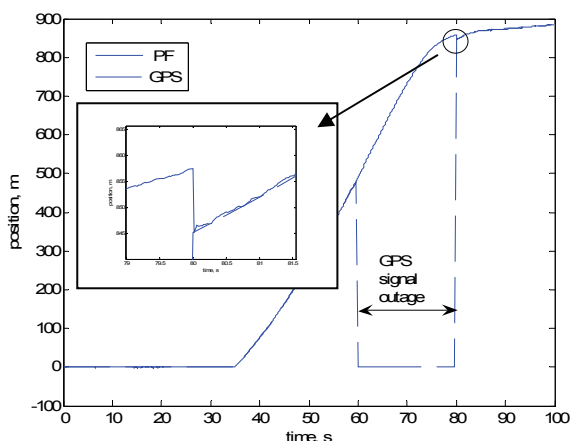


Fig. 4. PF result of position estimation with 20 s of GPS signal outage

The calculated position estimation drift during 20 s of GPS signal outage is 12 m. PF immediately corrects the position of the vehicle, when GPS measurements become available (enlarged fragment of Fig. 4).

## Conclusions

Presented PF algorithm for accelerometer and GPS data fusion has following advantages over KF:

- no need for nonlinear measurement model linearization, this is especially attractive for real low cost navigation system parameters estimation.
  - no restriction for using non Gaussian PDF for estimating system states,
  - implementing and debugging the PF is not a major issue comparing with EKF.
- Also, presented fusion algorithm continues to have reliable navigation solution during small GPS signal outages.

## References

1. **Yafei Ren, Xizhen Ke.** Particle Filter Data Fusion Enhancement for MEMS-IMU/GPS // IIM, 2010. – Vol. 2. – P. 417–421.
2. **Francois S, Pang S. K., Carmi A., Godsill S.** On MCMC-Based Particle Methods for Bayesian Filtering: Application to Multitarget tracking // 2009 3<sup>rd</sup> IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive processing, 2009. – P. 360–363.
3. **Aggarwal P., Syed Z., Noureldin A., El-Sheimy N.** MEMS-Based Integrated Navigation. – Artech House, 2010. – 197 p.
4. **Bistrovs V.** Analyse of Kalman algorithm for Different Movement Modes of Land Mobile // Electronics and Electrical Engineering. – Kaunas: Technologija, 2008. – No. 6 (86). – P.89–92.
5. **Arulampalam Sanjeev M., Maskell S., Gordon N., Clapp T.** A Tutorial on particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking // IEEE Transactions on signal Processing, 2002. – Vol. 50. – No.2. – P. 174–188.

Received 2011 02 27

**V. Bistrovs, A. Kluga. MEMS INS/GPS Data Fusion using Particle Filter // Electronics and Electrical Engineering. – Kaunas: Technologija, 2011. – No. 6(112). – P. 77–80.**

Coupling GPS with Micro-Electro-Mechanical Systems (MEMS) Inertial Navigation Systems (INS) is a challenging way of improving land vehicle navigation performance. MEMS inertial sensors suffer from complex stochastic errors, which are difficult to compensate and model using conventional Kalman Filter, as it provides an effective solution to the linear Gaussian filtering problem. However where there is nonlinearity, either in the model specification or observation process, other methods are required. Particle filtering techniques are good candidates to solve the corresponding nonlinear estimation problem associated to MEMS INS/GPS hybridization. Here nonlinear models for accelerometer were successfully implemented using particle filter (PF). The performance of the resulting algorithm was illustrated through experimental results. Also the position estimation results using PF during GPS signal outages were presented. Ill. 4, bibl. 5 (in English; abstracts in English and Lithuanian).

**V. Bistrovs, A. Kluga. MEMS INS/GPS duomenų sintezė taikant dalelių filtrą // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2011. – Nr. 6(112). – P. 77–80.**

Sausumos transporto priemonių navigacijos sistemų našumą galima padidinti taikant mikroelektromechanines sistemas ir inercines navigacijos sistemas. Atsitiktines klaidas, turinčias įtakos mikroelektromechaninių sistemų jutikliams, sunku kompensuoti naudojant tradicinius Kalmano filtrus. Esant dideliame netiesiškumui, būtina taikyti kitą metodą, pavyzdžiui, dalelių filtrą. Dalelių filtras buvo pritaikytas akcelerometruose naudojant netiesinius modelius. Il. 4, bibl. 5 (anglų kalba; santraukos anglų ir lietuvių k.).