DOCTORAL THESIS Statement

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Radioelectronics

Ing. Ondřej Jakubov

Low Complex PVT Estimation Using Factor Graphs

Ph.D. Programme: Electrical Engineering and Information Technology
Branch of Study: Radioelectronics

Doctoral Thesis Statement for Obtaining the Academic Title of "Doctor",
Abrreviated to Ph.D.

Prague, June 2013

The doctoral thesis was produced in combined manner

Ph.D. study at the department of Radioelectronics of the Faculty of Electrical Engineering of the CTU in Prague

Candidate:	Ing. Ondřej Jakubov Nottingham Scientific Limited Loxley House, Tottle Road, Nottingham, United Kingdom, NG21RT
Supervisor:	Doc. Dr. Ing. Pavel Kovář Department of Radioelectronics Faculty of Electrical Engineering of the CTU in Prague Technická 2, 166 27 Prague 6
Supervisor-Specialist:	Ing. Petr Kačmařík, Ph.D. Department of Radioelectronics Faculty of Electrical Engineering of the CTU in Prague Technická 2, 166 27 Prague 6
Opponents:	_____ _____ _____ _____

The doctoral thesis statement was distributed on _____:

The defence of the doctoral thesis will be held on _____at _____ a.m./p.m. before the Board for the Defence of the Doctoral Thesis in the branch of study 2601V010 Radioelectronics in the meeting room No. _____ of the Faculty of Electrical Engineering of the CTU in Prague.

Those interested may get acquainted with the doctoral thesis concerned at the Dean Office of the Faculty of Electrical Engineering of the CTU in Prague, at the Department for Science and Research, Technická 2, Praha 6.

_____

Chairman of the Board for the Defence of the Doctoral Thesis
in the branch of study 2601V010 Radioelectronics
Faculty of Electrical Engineering of the CTU in Prague
Technická 2, 166 27 Prague 6

Contents

1 Current Situation of the Studied Problem

Measurement Data Increase The *global navigation satellite systems* (GNSS) are being developed by many countries nowadays. The constantly increasing number of visible space vehicles (SV) and other radio beacons (RB), such as pseudolites and cellular mobile stations, requires computational power increase of the digital signal processors [1,3–6] generating the estimates of the user position, velocity, and time (PVT) from the digitized signal samples.

Estimator Decomposition A *conventional receiver architecture* (Fig. 1 left) decomposes the estimator into several *local tracking channels*, each for a tracked SV, and the *navigation processor*. The local tracking channel provides an estimate of the clock biased user-to-SV range, named as *pseudorange* ρ, and its first derivative *pseudorange rate* $\dot{\rho}$, which become observables for the navigation processor estimating the final PVT. The navigation processor operates at lower rate than the local tracking channels, 0.1s or 1s typically. This suboptiomal solution has been shown to attain the posterior Cramér-Rao lower bound (PCRLB) when the SVs are seen at high power without high dynamics, interference, and multipath effects [15]. If the PVT estimates or its predictions are utilized to control the feed-back systems in the local tracking channels, the tracking sensitivity and precision at high dynamics, signal blockage and interference can be significantly improved (sensitivity by 7 dB for GPS L1 blocked SV) [11,12]. This is known as *vector tracking architecture* (Fig. 1 left, red). The higher the rate of the navigation processor the faster it can feed the tracking channels with up-to-date information and hence sustain dynamics. The approach when the PVT is estimated directly from the sampled signal, *direct positioning architecture* (Fig. 1 right), offer higher multipath mitigation ability [14,15], however is intractable in real time with current technology and is not a subject of study.

Local Tracking Channel A functional block diagram of the local tracking channel of a conventional architecture is depicted in Fig. 1 (up). Formation of pseudorange ρ_i and pseudorange rate $\dot{\rho}_i$ to ith SV is decomposed into the following steps: given the initial estimates of the code delay and carrier frequency shift, synchronization to the code (DLL) and carrier (FLL, PLL) is maintained. Carrier-to-noise power ratio is estimated ($\hat{C/N_0}$), the bit boundary is found and the lock detect status is monitored. Within the bit stream, frames are detected, demodulated, deinterleaved and decoded. A counter of the elapsed number of bits is reset ($N_b = 0$) and system time of frame transmission t_F is stored. The satellite orbit and time information are stored. At user time t_U derived from the local oscillator, the actual number of elapsed bits N_b and its fractional part ΔN_b are sensed, the satellite transmission time is estimated as $\hat{t}_{S,i} = t_F + (N_b + \Delta N_b)T_b$ where T_b is the nominal bit period. The pseudorange is formed as $\rho_i = c\,(t_U - t_{S,i})$ where c is the speed of light. The pseudorange rate is formed as $\dot{\rho}_i = -c\hat{f}_{s,i}/f_c$ where $\hat{f}_{s,i}$ is the frequency shift estimate from the carrier tracking loop at time t_U, f_c is the carrier frequency. Satellite position $\mathbf{x}_S$, and velocity $\mathbf{v}_S$ at estimated transmission time $t_{S,i}$ are calculated.

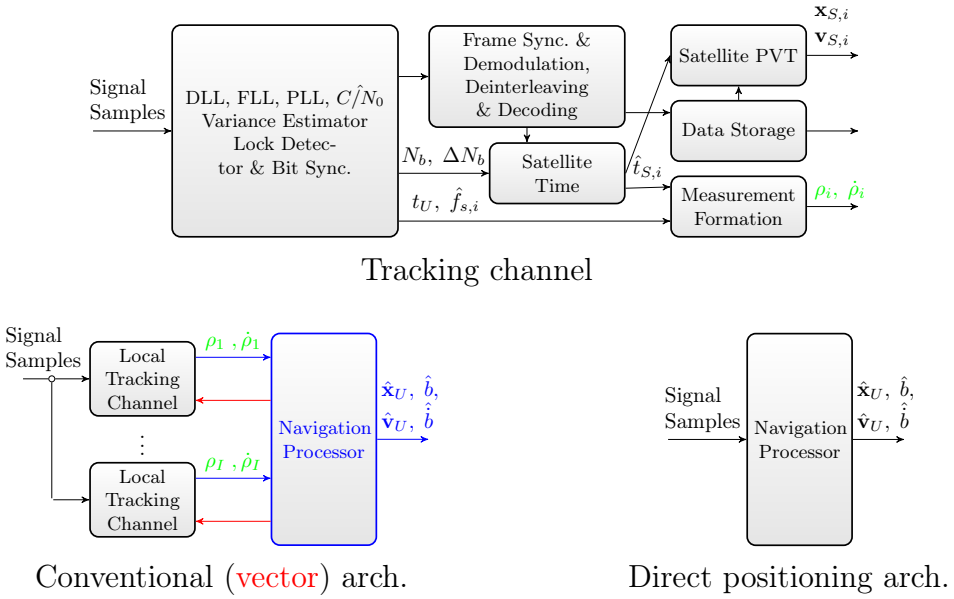

Figure 1: GNSS receiver architectures, conventional (low left blue), vector tracking (low left blue and red), direct positioning (low right), tracking channel (up).

Navigation Processor The navigation processor estimates the user position $\mathbf{x}_U$, user-to-satellite clock bias b, user velocity $\mathbf{v}_U$, user-to-satellite clock drift $\dot{b}$ based on measurement equations derived from the geometry of the given problem and the noise model of the pseudorange $w_{\rho,i}$ and pseudorange rate $w_{\dot{\rho},i}$

$$\rho_i = \|\mathbf{x}_U - \mathbf{x}_{S,i}\| + b + w_{\rho,i} \tag{1}$$

$$\dot{\rho}_i = -\mathbf{1}_i^T \cdot (\mathbf{v}_U - \mathbf{v}_{S,i}) + \dot{b} + w_{\dot{\rho},i} \tag{2}$$

where $\mathbf{1}_i = (\mathbf{x}_U - \mathbf{x}_{S,i}) / \|\mathbf{x}_U - \mathbf{x}_{S,i}\|$. PVT filters additionally utilize the history of the observation and user motion model.

Navigation Processor - Current Methods The current PVT estimation/filtering algorithms involve matrix manipulations which complexity grows significantly with the increasing number of measurements [2,16]. These algorithms, including *least squares* (LS), *weighted least squares* (WLS), *extended Kalman Filter* (EKF), strictly rely on first order Taylor linearization of the pseudorange measurument model 1. If distances to narrow RBs are measured or the user maneuvers quickly, the geometry changes rapidly with respect to the time step of PVT estimation and the basic assumptions of the model simplification are violated. The algorithms such as *unscented Kalman Filter* (UKF), and *particle filter* (PF) can model the nonlinearity based on the representation of the probability density function (PDF) by a finite number of samples [7–10]. These methods are mostly of quadratic or linear dependency on the number of visible RBs even though they work on vector data, but their performance is strictly corrupted if a relatively large number of representative samples are not selected.

2 Aims of the Doctoral Thesis

The aim of the thesis was to find an algorithm which can facilitate the requirements on the navigation processor by any means, while preserving the properties such as accuracy and convergence in comparison with the existing methods WLS, EKF, for a high number of tracked SVs. By any means, we mean

- reduced number of floating point operations (flops)
- avoidance of matrix inversion
- straightforward parallel implementation

which would enable engineers to

- employ a simplier microprocessor or just a microcontroller in their embedded system
- with less code and data memory
- or with complex processor increase the rate of this processor and hence foster the dynamic performance of the vector tracking architecture.

3 Working Methods

Study After the study of the GNSS receiver architectures, existing statistical estimation methods and their adoption to GNSS, future demand on a low complex PVT estimation algorithm had been identified. Inspired by a simplified 2-D snapshot localization method based on iterative message passing in a cycle factor graph involving only scalar operations [19], attention was turned to the factor graph theory and the sum-product algorithm.

Algorithm Development Chen's algorithm [19] was then generalized for a GNSS measurement model, including extension of other parameters to estimate - z coordinate to get 3-D user position $\mathbf{x}_U$, clock bias b_U, clock drift $\dot{b}_U$, user velocity $\mathbf{v}_U$, and redefinition of the observables - pseudoranges $\{\rho_i\}_{i=1}^{I}$ and pseudorange rates $\{\dot{\rho}_i\}_{i=1}^{I}$. Symbol I denotes the number of visible SVs. Further generalization for Gaussian PDF representation and first order Taylor approximation of the measurement model was found. A similarity to the factor graph model of the existing WLS algorithm was noticed and a unified procedure to derive such scalar models from the vector ones was identified. This procedure was applied to derive a scalar alternative to the EKF which we briefly describe in Subsection 4.1.

Simulations Monte Carlo simulations were carried out to verify the convergence and measure the accuracy of the proposed method. The adopted model assumed Gaussian pseudoranges and pseudorange rates uncorrelated over satellite channels without any biases. The user was moving in a circle with low and moderate dynamics. The results are further described in Subsection 4.2. A case study simulation was conducted to verify that the proposed algorithm is able to operate in the vector tracking architecture and offer similiar benefits as the existing EKF. The correlation outputs were modelled using semi-analytic methods [17] employing author's simulator *GNSSTracker* [18]. Atmospheric biases, oscillator phase noise, finite signal bandwidth were added. The pseudoranges and pseudorange rates were correlated over satellite channels due to an incorporated orbit motion model, see Subsection 4.3.

Receiver Development During the PhD study, the author actively contributed to open source/open hardware GNSS receiver project *the Witch Navigator* (WNav). The author developed the data decoders and PVT modules for GPS L1 C/A, GLONASS L1/L2, RF front end control, and did others such as web, logos, templates, promotion.

Experiment The WNav receiver was used as a testing platform for a case study experiment of the proposed method in a conventional receiver architecture. The signal was generated with high-fidelity GPS L1 Spirent simulator GSS6560, see Subsection 4.4.

4 Results

In this section, we first briefly describe the proposed algorithm along with the EKF, since these are compared in the simulations and experiments, see Subsection 4.1. The description is given for a generalized system. The reader may obtain the final algorithm by substituting models from Section 1. Monte Carlo analysis of the convergence and accuracy is given in Subsection 4.2, a case study on the vector tracking tracking architecture in Subsection 4.3, and the experimental results with WNav receiver in Subsection 4.4.

4.1 Proposed Algorithm

System Model Assume that $\boldsymbol{\theta}_n$ is a random vector parameter to be estimated with Gauss-Markov state-space model

$$\boldsymbol{\theta}_n = \mathbf{a}_n\left(\boldsymbol{\theta}_{n-1}\right) + \mathbf{B}\mathbf{u}_n \tag{3}$$

where $\mathbf{a}_n$ is a p-dimensional function, $\mathbf{u}_n$ is an r-dimensional vector being a Gaussian random variable uncorrelated over time, named as *driving noise*, with zero mean and covariance matrix $\mathbf{Q}_n$

$$\mathrm{E}\left[\mathbf{u}_{n+m}\mathbf{u}_n^T\right] = \begin{cases}\mathbf{0} & n \neq m \\ \mathbf{Q}_n & n = m\end{cases} \tag{4}$$

and $\mathbf{B}_n$ is a $p \times r$ matrix. Assume the initial value of the parameter is Gaussian $\boldsymbol{\theta}_{-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{\theta}}, \mathbf{C}_{\boldsymbol{\theta}}\right)$ and independent of $\mathbf{u}_n$ for $n \geq 0$. If $\mathbf{x}_n$ is an $M \times 1$ observation vector expressed by the following additive Gaussian noise model at time n, forming the *measurement equation*,

$$\mathbf{x}_n = \mathbf{h}_n\left(\boldsymbol{\theta}_n\right) + \mathbf{w}_n \tag{5}$$

where $\mathbf{h}_n$ denotes an M-dimensional function, $\mathbf{w}_n$ is $M \times 1$ zero mean Gaussian observation vector, $\mathbf{w}_n \sim \mathcal{N}\left(\mathbf{0}, \mathbf{C}_n\right)$, uncorrelated over time with covariance matrix $\mathbf{C}_n$

$$\mathrm{E}\left[\mathbf{w}_{n+m}\mathbf{w}_n^T\right] = \begin{cases}\mathbf{0} & n \neq m \\ \mathbf{C}_n & n = m.\end{cases} \tag{6}$$

Extended Kalman Filter The sequential suboptimal MMSE estimator of $\boldsymbol{\theta}_n$ based on first order Taylor linearization, *named as extended Kalman filter*, can be summarized by the following recursion:
Prediction:

$$\tilde{\boldsymbol{\theta}}_n = \mathbf{a}_n\left(\hat{\boldsymbol{\theta}}_{n-1}\right). \tag{7}$$

Minimum Prediction MSE Matrix:

$$\tilde{\mathbf{M}}_n = \mathbf{A}_n\mathbf{M}_n\mathbf{A}_n^T + \mathbf{B}_n\mathbf{Q}_n\mathbf{B}_n^T. \tag{8}$$

Kalman Gain Matrix:

$$\mathbf{K}_n = \tilde{\mathbf{M}}_n\mathbf{H}_n^T\left(\mathbf{H}_n\tilde{\mathbf{M}}_n\mathbf{H}_n^T + \mathbf{C}_n\right)^{-1}. \tag{9}$$

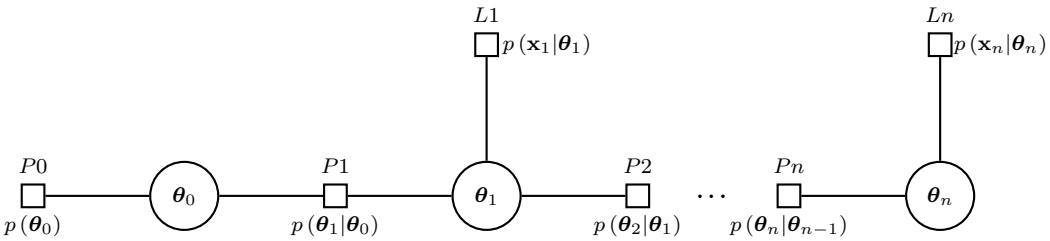

Figure 2: Factor graph for Bayesian filtering

Correction:

$$\hat{\boldsymbol{\theta}}_n = \tilde{\boldsymbol{\theta}}_n + \mathbf{K}_n \left(\mathbf{x}_n - \mathbf{h}_n \left(\tilde{\boldsymbol{\theta}}_n \right) \right). \tag{10}$$

Minimum MSE Matrix:

$$\mathbf{M}_n = \left(\mathbf{I} - \mathbf{K}_n \mathbf{H}_n \right) \tilde{\mathbf{M}}_n. \tag{11}$$

The recursion is initialized with $\hat{\boldsymbol{\theta}}_{-1} = \boldsymbol{\mu}_{\boldsymbol{\theta}}$, $\tilde{\mathbf{M}}_{-1} = \mathbf{C}_{\boldsymbol{\theta}}$. In (8), (9), (11), we substitute

$$\mathbf{A}_n = \left. \frac{\partial \mathbf{a}_n(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right|_{\hat{\boldsymbol{\theta}}_{n-1}} \tag{12}$$

$$\mathbf{H}_n = \left. \frac{\partial \mathbf{h}_n(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right|_{\tilde{\boldsymbol{\theta}}_n}. \tag{13}$$

Scalar Extended Kalman Filter - Approach The scalar EKF can be derived from a general factor graph representing Bayesian filter based on the following equality

$$p(\boldsymbol{\theta}_n|\mathbf{x}_{1:n}) \propto p(\mathbf{x}_n|\boldsymbol{\theta}_n) \int p(\boldsymbol{\theta}_n|\boldsymbol{\theta}_{n-1})\, p(\boldsymbol{\theta}_{n-1}|\mathbf{x}_{1:n-1})\, \mathrm{d}\boldsymbol{\theta}_{n-1} \tag{14}$$

which is depicted in Figure 2. Symbol $\propto$ denotes equality up to a scaling factor. The state-space PDF $p(\boldsymbol{\theta}_n|\boldsymbol{\theta}_{n-1})$ and the likelihood PDF $p(\mathbf{x}_n|\boldsymbol{\theta}_n)$ are then decomposed into a product of the dirac-delta conditional PDF and Gaussian PDF of the measurement noise $\mathbf{w}_n$ and driving noise $\mathbf{u}_n$, respectively. The dirac-delta function of vector arguments is further developed into a product of dirac-delta functions each for a single element for the modeled vector. Similarly, the multivariate Gaussian PDF is decomposed into a product of scalar Gaussian PDF. This can be applied if and only if the covariance matrix is diagonal. If this assumption does not hold, the Gaussian vector is decomposed according to SVD decomposition [16] to yeild a product of a correlating matrix and a Gaussian vector with diagonal covariance matrix. The multivariate PDF hence again decomposes into a product of dirac-delta PDFs and now scalar Gaussian PDFs. The resulting factor graph for the state-space model is depicted in Figure 3 left, for the measurement model in Figure Figure 3 right.

Scalar Kalman Filter - Algorithm The update rules for the factor and variable nodes are depicted in Figure 4. The *message passing algorithm* starts at variable nodes $\theta_{0,1},\ldots,$

$\theta_{0,p}$ where the messages sent to factor nodes $P_{1,1},\ldots,\ P_{1,p}$ comprise the mean and variance of the [1]initial distribution $\lambda_{\theta_{0,i}\to P_{1,j}}(\theta_{0,i}) = \left\{\mu_{0,i},\ \sigma_{0,i}^2\right\}$ for $i,\ j \in \{1,\ldots,\ p\}$ where $\mu_{0,i} = [\boldsymbol{\mu}_0]_i$ and $\sigma_{0,i}^2 = [\mathbf{C}_0]_{i,i}$. The messages from variable nodes $u_{0,i}$ are also sent to the factor nodes $P_{1,1},\ldots,\ P_{1,p}$ $\lambda_{u_{1,i}\to P_{1,j}}(u_{1,i}) = \left\{0,\ \sigma_{u,i}^2\right\}$ and updates are calculated according to Figure 4. The branches to variable nodes $\theta_{1,1},\ldots,\theta_{1,p}$ are disconnected and the resulting messages are sent back towards the variable nodes $\theta_{0,1},\ldots,\ \theta_{0,p}$ and the iterations start. When the last iteration, the branches to the variable nodes $\theta_{1,1},\ldots,\theta_{1,p}$ are connected, the messages are sent to them. The observed values are sent to the likelihood factor nodes $\lambda_{x_{n,i}\to L_{n,i}}(x_{n,i}) = \{x_{n,i},\ 0\}$ for $i \in \{1,\ldots,\ M\}$ and so do the noise variable nodes $\lambda_{w_{n,i}\to L_{n,i}}(w_{n,i}) = \{0,\ \sigma_i^2\}$. Next, the messages from the previous state-space factor nodes $P_{1,1},\ldots,\ P_{1,p}$ are sent through variable nodes $\theta_{1,1},\ldots,\theta_{1,p}$ to the likelihood factor nodes $L_{1,1},\ldots,L_{1,p}$ and the updates are therein calculated. The results are then sent back to variable nodes $\theta_{1,1},\ldots,\theta_{1,p}$ and then updated taking the unchanged messages from factor nodes $P_{0,1},\ldots,P_{0,p}$. The branches to the future factor nodes $P_{2,1},\ldots,\ P_{2,p}$ remain disconnected. The updated messages are sent towards factor nodes $L_{1,1},\ldots,L_{1,p}$ and iterations are started in this way. The iterations are finished after the last updates at variable nodes $\theta_{1,1},\ldots,\theta_{1,p}$ for which the future variable nodes $P_{2,1},\ldots,\ P_{2,p}$ are now connected. The messages sent to these factor nodes represent the beliefs and the means can be used to obtain the approximated MMSE estimates. And so continuous the algorithm sequentially.

Scalar Extended Kalman Filter - Algorithm The EKF can be approximated on the scalar FG as follows. Suppose that

$$\mathbf{a}(\boldsymbol{\theta}_n) = \begin{bmatrix} a_1(\theta_{n,1},\ldots,\ \theta_{n,p}) \\ \vdots \\ a_p(\theta_{n,1},\ldots,\ \theta_{n,p}) \end{bmatrix} \tag{15}$$

and

$$\mathbf{h}(\boldsymbol{\theta}_n) = \begin{bmatrix} h_1(\theta_{n,1},\ldots,\ \theta_{n,p}) \\ \vdots \\ h_M(\theta_{n,1},\ldots,\ \theta_{n,p}) \end{bmatrix}. \tag{16}$$

Let us next assume that $h_{i,j} = [\mathbf{H}]_{i,j}$ and $a_{i,j} = [\mathbf{A}]_{i,j}$. The conditional PDFs $p(\boldsymbol{\theta}_n|\boldsymbol{\theta}_{n-1},\ \mathbf{u}_n)$, $p(\mathbf{x}_n|\boldsymbol{\theta}_n,\ \mathbf{w}_n)$ both can be approximated for the EKF as Gaussian

$$p(\boldsymbol{\theta}_n|\boldsymbol{\theta}_{n-1},\ \mathbf{u}_n) = \prod_{i=1}^{p} \delta\left(\theta_{n,i} - a_i(\theta_{n-1,1},\ldots,\ \theta_{n-1,p}) - \sum_{j=1}^{r} b_{i,j}u_j\right) \tag{17}$$

$$p(\mathbf{x}_n|\boldsymbol{\theta}_n,\ \mathbf{w}_n) = \prod_{i=1}^{M} \delta(x_{n,i} - h_i(\theta_{n,1},\ldots,\ \theta_{n,p}) - w_i). \tag{18}$$

[1]The initial cross-correlation between the parameter is not assumed. To regard it, one can resort to SVD decomposition.

The update rules of the scalar variable node remain unchanged compared to the KF's. However, the mean of the factor node update will change and the variance will remain unchanged. Assume the situation in Figure 4(b). Next, assume the PDF (=factor function) $p_F\left(\theta|\theta_1, \ldots, \theta_K\right)$ will have the following form

$$\begin{aligned} p_F\left(\theta|\theta_1, \ldots, \theta_K\right) &= \delta\left(a\theta - f\left(\theta_1, \ldots, \theta_K\right)\right) && (19) \\ &\approx \delta\left(a\left(\theta - \breve{\theta}\right) - \sum_{k=1}^{K} a_k\left(\theta_k - \breve{\theta}_k\right)\right) && (20) \end{aligned}$$

where f is K-dimensional function and $\breve{\theta}$, $\breve{\theta}_1, \ldots, \breve{\theta}_K$ are constant linearizing points such that

$$\breve{\theta} = f\left(\breve{\theta}_1, \ldots, \breve{\theta}_K\right). \tag{21}$$

The updated mean will have the following form

$$\mu = \breve{\theta} + \left(\sum_{k=1}^{K} a_k\left(\mu_k - \breve{\theta}_k\right)\right)/a. \tag{22}$$

In the scalar EKF, the linearization will generally take place in both state-space and likelihood factor nodes. The linearizing points for the state-space model nodes will be the estimated parameters from the previous time $\hat{\theta}_{n-1,i}$ where $i \in \{1, \ldots, p\}$. The linearizing points for the likelihood nodes will be the means of the output messages from the state-space model nodes. This complies with the vector EKF where predictions are used to linearize the nonlinear observation function $\mathbf{h}$.

The complexity of the scalar EKF is slightly higher than that of the scalar KF due to the necessity of evaluation of the Jacobians $\mathbf{A}$, $\mathbf{H}$ if they depend on time. The number of flops might be increased if functions $\mathbf{a}$, $\mathbf{h}$ are complicated. The linearizing point can be updated after every iteration to improve the convergence which requires the evaluations of the Jacobians.

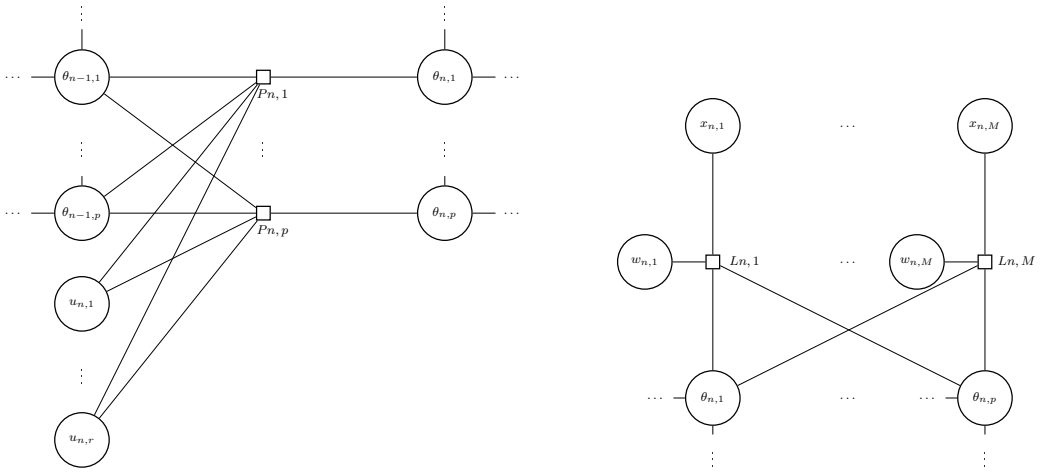

Figure 3: FG for state-space model (left) and likelihood function (right) of a scalar KF

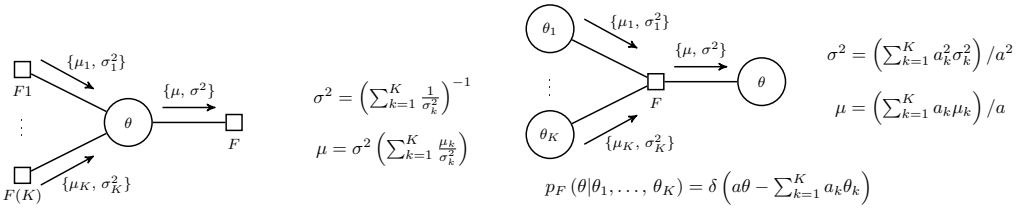

Figure 4: Update rules for scalar vertices of KF - (a) variable node, (b) factor node

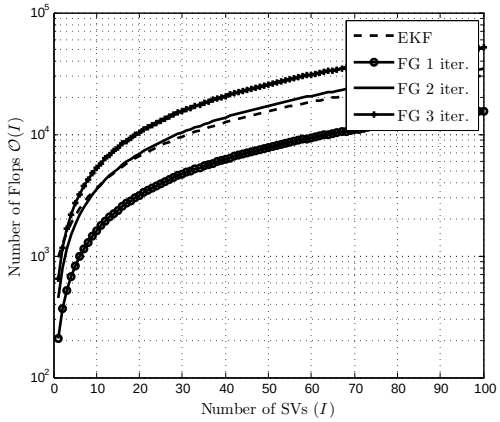

Figure 5: Complexity comparison of the PVT filtering algorithms. Symbol $\mathcal{O}(I)$ denotes the number of floating point operations (flops) depending on the number of SVs denoted as I

4.2 Convergence, Accuracy and Complexity

Complexity The complexity of the algorithm is quadratic in the number of states and linear in the number of observations. This is not true for the vector KF where the complexity is cubic in states and cubic or quadratic in observations depending on the form of the KF. However, the sum-product algorithm in the scalar case does not yield truly the MMSE estimates of the parameter $\boldsymbol{\theta}_n$, since the cycles are present in the graph. The complexity then will be a multiple of the number of iterations, and the accuracy with convergence will have to be investigated for every single implementation of the scalar Kalman filter. Figure 5 plots the number of flops for the proposed algorithm until first three iterations and the EKF in case of adoption to the investigated GNSS PVT model. The number of flops is comparable for both methods. The advantage of the scalar Kalman filter is that the update rules operate only scalars and simple arithmetic operations unlike the vector case. The algorithm is fully distributed. The updates are calculated in separate processing units each for a tracked satellite. The method can hence be implemented in hardware logic utilizing simple parallel entities.

Convergence The convergence histograms of the EKF and the FG-based scalar iterative EKF algorithms are depicted in Figure 4.2 (up) for randomly distributed 4-8 SVs (left), 8-64 (right) under low user dynamics. The filtering run for 1000s and was repeated 1000 times. For the number of visible SVs equal sixteen and more, no divergence has been observed in 1000 repetitions. This fact substantiates us to employ the iterative filtering for large data vectors. Luckily, we see that the convergence is relatively fast and can be reached within few iterations. High dynamics scenarios were investigated in the thesis. It is shown that the convergence is observed if no modelling to high dynamics is introduced, otherwise the proposed method diverges. If the modeled dynamics is acceptably low at high dynamics, iterations generally improve convergence probability.

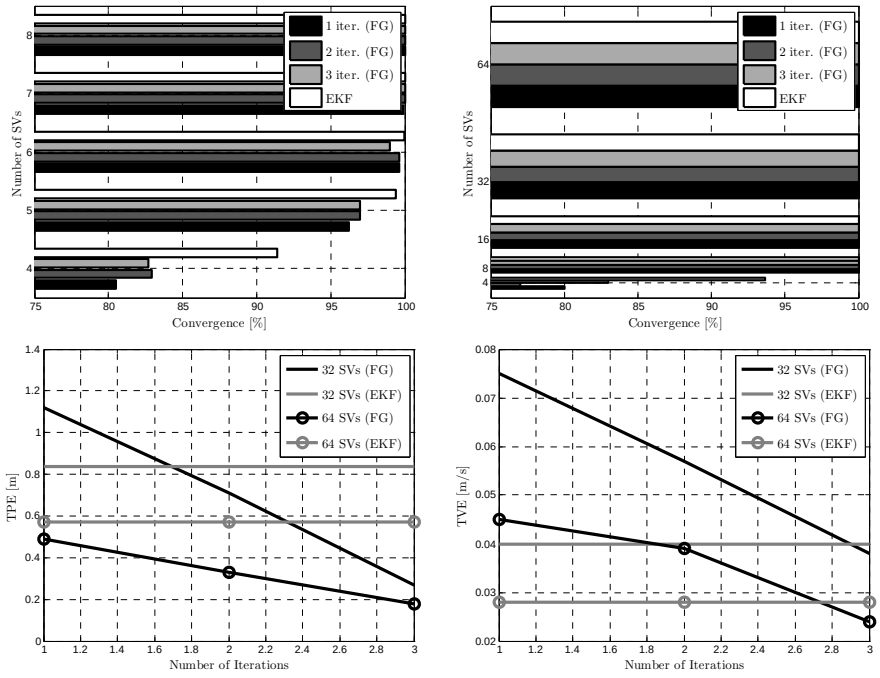

Figure 6: Convergence histograms (up), accuracy (down, position error - low left, velocity error - low right) - comparison of the proposed algorithm with the EKF

Accuracy In Figure 7 (low), we illustrate the accuracy of the algorithms. The position error is in the left column, whereas the the velocity is in the right column. The same scenario as in the previous case is assumed. The thesis further documents simulations of moderate dynamics and various motion models. It is observed that if we increase acceleration, the FG-based filter outperforms the EKF in accuracy for larger number of SVs or for more than one iteration. In this case, the performance is improved over iterations due to the fact that a linearizing point is calculated for each iteration. The linearization fails for a large position difference between the predicted (also linearizing) value and the true value. Here, we iteratively shift towards the true value with iterations, whereas the EKF does this only once at an epoch. If we model the higher dynamics and remain stable, the EKF performs better and the iteration do not improve the accuracy, but the difference in accuracy between the two filters decays with larger number of the SVs.

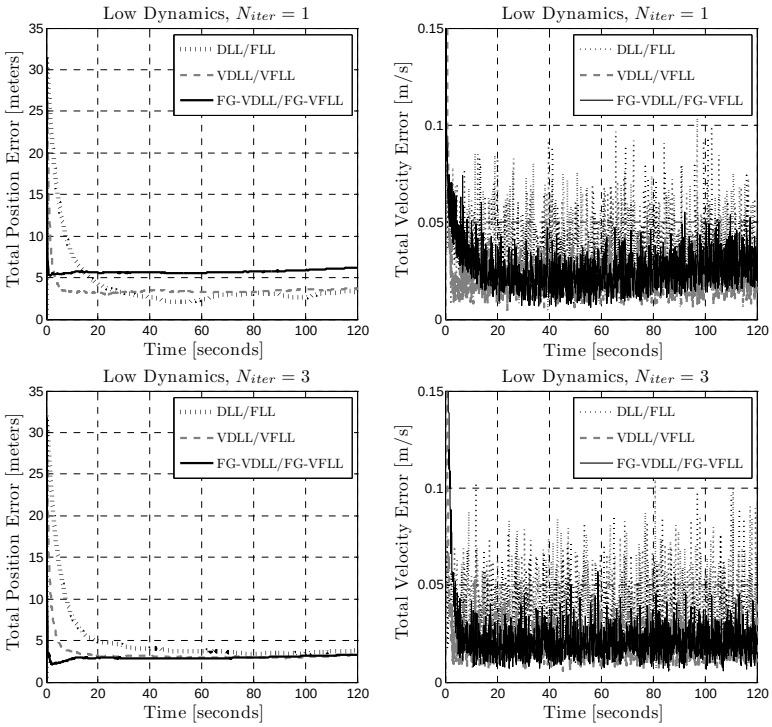

Figure 7: Vector tracking - simulation results - low dynamics. (up left) position error $N_{\text{Iter.}} = 1$, (up right) velocity error $N_{\text{Iter.}} = 1$, (low left) position error $N_{\text{Iter.}} = 3$, (low right) velocity error $N_{\text{Iter.}} = 3$.

4.3 Simulation - Vector Tracking Architecture

The simulation results documenting the open-sky scenario are in Figure 7 for low user dynamics. Results of the moderate dynamics are illustrated in the thesis. Position and velocity errors are plotted for a single iteration ($N_{\text{Iter.}} = 1$) and for three iterations ($N_{\text{Iter.}} = 3$). The proposed FG-based algorithm incorporated to the vector tracking architecture (FG-VDLL/FG-VFLL) is compared with the EKF of the scalar tracking architecture (DLL/FLL) and with the EKF of the vector tracking architecture (VDLL/VFLL). All algorithms adopt identical motion model. Second order FLL aids first order DLL in the scalar tracking loops. It is clear that a single iteration on the FG results in comparable position and velocity filtering errors as for the EKF vector tracking loop in an open-sky scenario at low dynamics. An increased number of iterations slightly improves the performance at low dynamics. At moderate dynamics, both VDLL/VFLL and FG-VDLL/FG-VFLL perform similarly for $N_{\text{Iter.}} = 3$. For $N_{\text{Iter.}} = 1$ conclusions about precision are difficult to make. In Figure 8, we sweep C/N_0 for all the tracked SVs and observe stability. Moderate dynamics and a single iteration are considered. It is apparent from the figures that the FG-VDLL/FG-VFLL loop looses lock at approximately 2 dB lower C/N_0 than the VDLL/VFLL loop. Fig. 8(low left) depicts the C/N_0 estimate errors of the FG-VDLL/FG-VFLL channels.

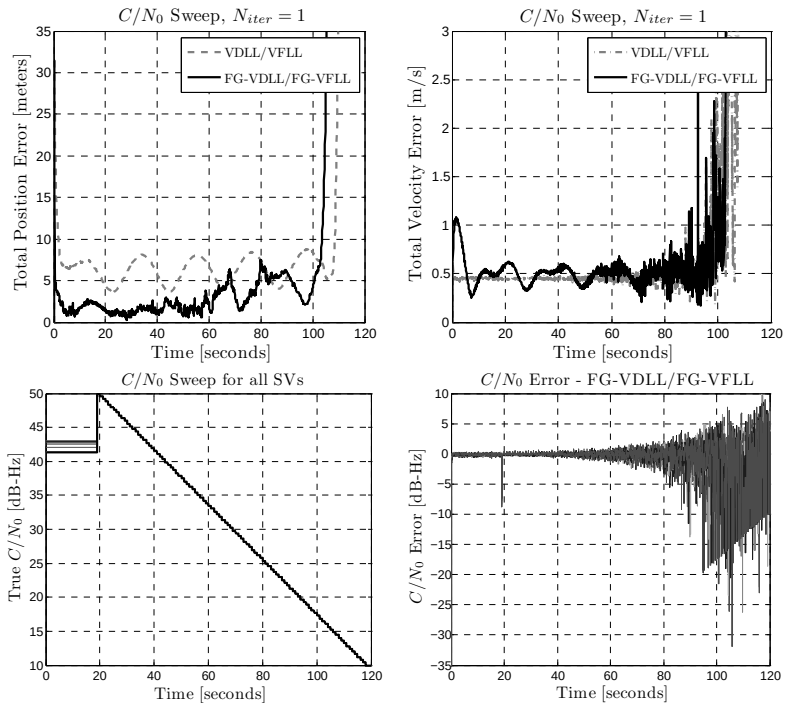

Figure 8: Vector tracking - simulation results - C/N_0 sweep**.** (up left) position error at C/N_0 sweep, (up right) velocity error at C/N_0 sweep, (low left) swept C/N_0 of all visible SVs, (low right) C/N_0 estimate error for all visible SVs.

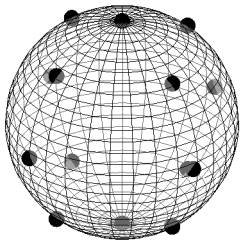

Figure 9: User positions on the Earth - static experiment (14 positions)

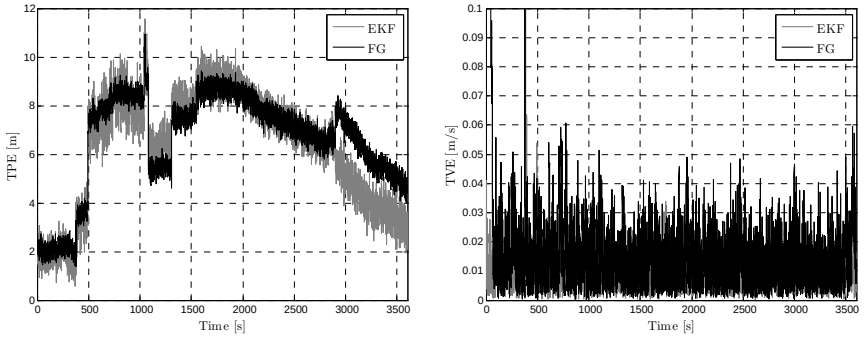

Figure 10: Experimental results - accuracy. Position error (left), velocity errors (right) - averaged errors of static scenarios.

4.4 Experiment - Conventional Architecture

Methodology We assume that the user moves in a circle with constant circular orbit speed in an open-sky scenario. Firstly, we place the user to 14 static points on the Earth, see Figure 9, at a given time and investigate the filtering performance. Secondly, two dynamic scenarios ($a = 1\,\mathrm{m/s^2}$, $a = 10\,\mathrm{m/s^2}$) with radius 10 km, 1 km and velocity $100\,\mathrm{m/s}$ are supposed. The number of visible SVs has always been 11. The measurements were taken for 1 hour. The navigation update time was $T_N = 0.1$ s. Second order DLL, PLL were used with equivalent loop noise bandwidth 0.5 Hz, 30.0 Hz, respectively. The driving noise std. was always 0.01 m/s which mismatches the motion model, but still filters out. The FG-based filter ran with three iterations.

Results In Figure 10, we plot the results obtained by the WNav receiver. The left column depicts the averaged position error and the right column depicts avaraged velocity error of the static user. The accuracy of both filters is comparable. However, convergence was observed only in 10/14 cases. The reason is as discussed in Section 4.2 - low number of the SVs causes divergence. In thesis, it is shown that with acceleration $a = 10$ m/s^2 the FG-based filter remains stable and its accuracy is comparable to the EKF.

Use Case	Benefits	How
CPU's offload to HW logic	Low-end CPU	HDL directly, C-to-HDL, C-to-RTL
	SW/HW tradeoffs	
	Higher number of SVs to track	
	Higher stability at high dynamics	Faster PVT updates in a vector tracking architecture
	Avoiding inertial sensors	
CPU's implementation	Smaller code and data size	No matrix library in use
	Lower requirements on the OS	

Table 1: Use cases and benefits of the proposed algorithm

5 Conclusions

In the study, we introduced a novel PVT estimation/filtering method which

- requires simple parallel processing entities, each for a tracked satellite
- requires simple arithmetic operations such as sum, subtraction, multiplication, division

which enables engineers to offload the CPU into the hardware logic (HW) and hence:

- employ a low-end CPU and tradeoff the HW logic with the CPU
- track a higher number of SVs in the HW logic
- by faster updates of the PVT estimates and predictions, improve the stability of the vector tracking architecture and in some applications eliminate the need of inertial sensors.

If the algorithm is implemented in the CPU, the following advantages have been identified:

- inclusion of the matrix library can be avoided resulting in lower code size
- by storing only the actual mean and variance, the requirements on the data memory reduce.

By extensive Monte Carlo simulations, we proved that the algorithm converges if the number of the tracked SVs is larger than 16. For low number of SVs, the requirements on the navigation processor are low, anyway. The accuracy is comparable to the EKF for three and less iterations for low user dynamics. If the dynamics increases, it should not be modeled by the proposed filter, rather the number of the iterations should be increased. In this case, the EKF can be outperformed in accuracy, in price of the complexity increase. We showed by a case study, that the 7 dB tracking sensitivity benefit of the vector tracking architecture is lowered to 5dB. By experiment with WNav receiver developed at CTU in Prague, co-developed by the author, we demonstrated that a comparable precision to the EKF can be achieved. The use cases and benefits of the proposed algorithm are summurised in Tab. 1.

List of Literature Used in the Thesis Statement

[1] T. Pany, *Navigation Signal Processing for GNSS Software Receiver - GNSS Technology and Applications*. 1st ed. Norwood: Artech House, 2010.

[2] E. D. Kaplan and C. J. Hegarty, *Understanding GPS: Principles and Applications*. 2nd ed. Norwood, MA: Artech House, Inc., 2006.

[3] N. C. Shivaramaiah and A. G. Dempster, *Global Navigation Satellite Systems - Signal, Theory and Applications: Chapter 2 Baseband Hardware Design in Modernised GNSS Receivers*, 1st ed. IntechOpen, 2012.

[4] C. Stober et al., "ipexSR: A real-time multi-frequency software GNSS receiver," in *Proceedings of ELMAR*, Zadar, Croatia, 2010, pp. 407 – 416.

[5] H. Afzal and G. Lachapelle, "Design methodology for a dual frequency configurable GPS receiver," in *Proceedings of GNSS10*, Portland, OR, 2010, pp. 1 – 9.

[6] O. Jakubov, P. Kovář, P. Kačmařík, F. Vejražka, "The Witch Navigator - a low-cost GNSS software receiver for advanced processing techniques," *Radioengineering*, 2010, vol. 19, no. 4, pp. 536 - 543.

[7] E. Wan and R. V. der Merwe, "The unscented Kalman filter for nonlinear estimation," in *IEEE Symposium on Adaptive Systems for Signal Processing*, Communications and Control, 2000, pp. 604 – 611.

[8] T. Bo, C. Pingyuan, and C. Yangzhou, "Sigma-point Kalman filters for GPS based position estimation," in *Proceedings of Fifth International Conference on Information, Communications and Signal Processing*, Bangkok, 2005, pp. 213 – 217.

[9] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transaction on Signal Processing*, 2002, vol. 50, no. 2, pp. 174 - 188.

[10] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karls- son, and P. J. Nordlund, "Particle filters for positioning, navigation, and track- ing," *IEEE Transactions on Signal Processing*, 2002, vol. 50, no. 2, pp. 425 – 437.

[11] J. Spilker, "Vector delay lock loop processing of radiolocation transmitter signals," U.S. Patent 5 398 034, March 14, 1995. [Online]. Available: http://www.freepatentsonline.com/5398034.html.

[12] B. Pany, T. Eissfeller, "Use of vector delay lock loop receiver for GNSS signal power analysis in bad signal conditions," in *Proceedings of IEEE/ION PLANS 2006*, San Diego, CA, 2006, pp. 893 – 903.

[13] D. Benson, "Interference benefits of a vector delay lock loop (VDLL) GPS receiver," in *Proceedings of 63rd Annual Meeting of the Institute of Navigation*, Cambridge, Massachusetts, 2007, pp. 1 - 8.

[14] P. Closas, C. Fernandez-Prades, D. Bernal, and J. A. Fernandez-Rubio, "Bayesian direct position estimation," in *Proceedings of ION GNSS 21st International Technical Meeting of the Satellite Division*, Savannah, GA, September 2008, pp. 183 – 190.

[15] P. Closas, "Bayesian signal processing techniques for GNSS receivers: from multipath mitigation to positioning," Ph.D. dissertation, Universitat Politecnica de Catalunya, Barcelona, 2009.

[16] S. M. Kay, Fundamentals of Statistical Signal Processing: Volume I: Estimation Theory, 1st ed. New Jersey: Prentice-Hall, 1993.

[17] D. Borio, P. B. Anantharamu, and G. Lachapelle, "SATLSim: a semi-analytic framework for fast GNSS tracking loop simulations," *GPS Solutions*, May 2011, pp. 1–5.

[18] O. Jakubov, P. Kacmarik, P. Kovar, and F. Vejrazka, "Universality and realistic extensions to the semi-analytic simulation principle in GNSS signal processing," *Radioengineering*, 2012, vol. 21, no. 2, pp. 647 – 658.

[19] J. C. Chen, C. S. Maa, and Y. C. Wang, "Mobile position location using factor graphs," *IEEE Communications Letters*, 2003, vol. 7, no. 9, pp. 431 – 433.

List of Candidate's Works Relating to the Doctoral Thesis

Impacted Journals

[A1] O. Jakubov, P. Kovář, P. Kačmařík, F. Vejražka, "The Witch Navigator - a low-cost GNSS software receiver for advanced processing techniques," *Radioengineering*, 2010, vol. 19, no. 4, pp. 536 - 543. Contribution 25%.

[A2] O. Jakubov, P. Kovář, P. Kačmařík, F. Vejražka, "Distributed extended Kalman filter for position, velocity, time estimation in satellite navigation receivers," *Radioengineering*, submitted May 2013, pp. 1 - 15. Contribution 75%.

Papers Excerpted in Web of Science

None

Papers in Reviewed Journals

None

Patents

None

Utility Models

None

Other Publications

[A3] O. Jakubov, P. Kovář, "Signal processing algorithms in GPS, Galileo, and GLONASS integrated receiver," PhD thesis proposal, in *Proceedings of GNSS PhD Summit 2011*, Munich, Germany, November 2011, pp. 1 - 9. Contribution 50%.

List of Other Candidate's Works

Impacted Journals

[B1] O. Jakubov, P. Kačmařík, P. Kovář, F. Vejražka, "Universality and realistic extensions to the semi-analytic simulation principle in GNSS signal processing," *Radioengineering*, 2012. vol. 21, no. 2, pp. 536 - 543. Contribution 75%.

Papers Excerpted in Web of Science

None

Papers in Reviewed Journals

None

Patents

None

Utility Models

[B2] P. Kovář, P. Kačmařík, O. Jakubov, F. Vejražka, "The Witch Navigator HW 1.1", JC – Computer hardware and software, 9.12.2011. Contribution 10%.

[B3] P. Kačmařík, O. Jakubov, P. Kovář, F. Vejražka, "WNav GPS L1 C/A – GPS software receiver for civil signal in L1 band", JW – Navigation, communications, detection a countermeasures, 9.12.2011. Contribution 40%.

[B4] P. Kovář, P. Kačmařík, O. Jakubov, F. Vejražka, "Universal GNSS correlator IP", JC – Computer hardware and software, 9.12.2011. Contribution 10%.

Other Publications

[B5] P. Kovář, P. Kačmařík, O. Jakubov, F. Vejražka, "The implementation of dual frequency GLONASS receiver to the Witch Navigator," in *Proceedings of 18th Saint Petersburg International Conference on Integrated Navigation Systems*, St. Petersburg, Russia, May/June 2011, pp. 348 - 354. Contribution 25%.

[B6] O. Jakubov, P. Kačmařík, P. Kovář, F. Vejražka, "Reduced-complexity GNSS software simulator based on correlator output signal modeling," in *Proceedings of 18th Saint Petersburg International Conference on Integrated Navigation Systems*, St. Petersburg, Russia, May/June 2011, pp. 378 - 380. Contribution 75%.

[B7] P. Kačmařík, P. Kovář, O. Jakubov, F. Vejražka, "The Witch Navigator - a software GNSS receiver built on real-time Linux," in *Proceedings of 13th Real-Time Linux Workshop*, 2011, Prague, Czech republic, October 2011, pp. 17 - 28. Contribution 25%.

[B8] P. Kovář, P. Kačmařík, O. Jakubov, F. Vejražka, "The Witch Navigator - a software GNSS receiver for education and research," in *Proceedings of IGNSS 2011 Conference & Exhibition*, Sydney, Australia, November 2011, pp. 1 - 11. Contribution 25%.

[B9] P. Kovář, P. Kačmařík, O. Jakubov, F. Vejražka, "Witch Navigator - a low cost software receiver for education and research," in *Proceedings of European Navigation Conference ENC2011*, London, United Kingdom, November 2011, pp. 1 - 10. Contribution 25%.

[B10] P. Kovář, P. Kačmařík, O. Jakubov, P. Roule, F. Vejražka, "Two-channel method for the Processing of wide-band GNSS signals, its implementation and verification," in *Proceedings of IEEE/PLANS 2012*, San Diego, CA (USA), April 2012, pp. 1 - 10. Contribution 10%.

[B11] P. Roule, O. Jakubov, P. Kovář, P. Kačmařík, F. Vejražka, "GNSS signal processing in GPU", in *Proceedings of European Navigation Conference ENC2012*, Gdańsk, Poland, April 2012, pp. 1 - 10. Contribution 10%.

Response and Reviews

Citations

None

Awards

December 2011 - Paper [A3]
"Among three best doctoral thesis proposals"
GNSS PhD Summit 2011, University of FAF, Munich, Germany

Others

WNav receiver delivery to other institues - Paper [A1]:

Project TRANSMIT - Training Research and Applications Network to Support the Mitigation of Ionospheric Threats, *Melania Susi Melania.Susi@nottingham.ac.uk, Nottingham Geospatial Institute, United Kingdom*

Takuji Ebinuma, ebinuma@nsat.t.u-tokyo.ac.jp, Department of Aeronautics and Astronautics, University of Tokyo, Japan

Summary

The constantly increasing number of visible GNSS space vehicles (SV) and other radio beacons (RB), such as pseudolites and cellular mobile stations, challenges not only the design of a receiver's digital signal processor providing standard outputs of pseudoranges, pseudorange rates and navigation data, but also the design of the position-velocity-time (PVT) fusion algorithm handling large vector data, various coordinate systems and time references.

Receiver architectures, statistical estimation methods and their adoption to GNSS have been studied. Future demand on a low complex PVT estimation/filtering algorithm had been identified. The operations of the PVT estimation/filtering algorithms used in practice involve matrix manipulations whose complexity grows significantly with the increasing number of measurements. These algorithms, including least squares (LS), weighted least squares (WLS), extended Kalman Filter (EKF), strictly rely on first order Taylor linearization of the pseudorange measurement model. The linearization attains the (posterior) Cramér-Rao lower bound for distant SVs and low user dynamics. If distances to narrow RBs are measured or the user maneuvers quickly, the geometry changes rapidly with respect to the time step of PVT estimation and the basic assumptions of the model simplification are violated.

Inspired by Chen's algorithm (2003) of a simplified 2-D snapshot localization method based on iterative message passing in a cycle factor graph involving only scalar operations, factor graph theory and the sum-product algorithm have been studied. A generalization of Chen's algorithm was first found for a one-shot GNSS PVT estimation, second extended for a user motion model and observation history, thus becoming a distributed equivalent to the EKF involving no more vector and matrix operations.

By extensive Monte Carlo simulations, it was proved that the algorithm converges if the number of the tracked SVs is larger than 16. Further it was shown that the accuracy is comparable to the EKF for three and more iterations for low user dynamics. If the dynamics increases, an increase in the iterations reduces the error and can outperform the EKF. By a case study, it was shown that the 7 dB tracking sensitivity benefit of the vector tracking architecture is lowered to 5dB. By experiment with WNav receiver developed at CTU in Prague, co-developed by the author, it was demonstrated that a comparable precision to the EKF can be achieved.

The algorithm can be implemented in hardware logic and hence offload the CPU of the navigation processor. In the HW logic, the PVT estimates and predictions can be produced with higher rate and foster the performance of the vector tracking architecture at high dynamics. If the algorithm is implemented in the CPU, matrix library need not be included, thus reducing the code size. Due to the nature of the iterative process, only actual means and variances need to be stored which reduces the requirements on the data memory.

Résumé

Narůstající počet viditelných GNSS družic and jiných radiomajáků, jako např. pseudolitů a základnových stanic mobilných operátorů, se stávájí výzvou pro návrh nejen digitálního signálového procesoru přijímače poskytující standardní výstupy pseudovzdáleností a jejich derivací, ale také výzvou pro návrh sjednocujícího algoritmu odhadu polohy, rychlosti a času (PVT), jenž pracuje s rozsáhlými vektory, souřadnými soustavami a časovými referencemi.

Architektury přijímačů, stastické metody odhadu parametrů a jejich aplikace v družicové navigaci byly studovány. Operace odhadu a filtrace PVT použité v komerčních přijímačích využívají maticových operací, jejichž složitost výrazně narůstá s počtem sledovaných družic. Tyto algoritmy, mezi které řadíme nejmenší čtverce (LS), váhované nejmenší čtverce (WLS) a rozšířený Kalmánův filtr (EKF), spoléhají na linearizaci modelu měření pseudovzdálenosti Taylorovým rozvojem prvního řádu. Tato linearizace produkuje odhady s rozptylem na hranici Cramér-Raovy dolní meze za podmíky vzdálených družic a nízké dynamiky uživatele. Pro krátké vzdálenosti radiomajáků či pro rychle manévrujícího uživatele se geometrie mění rychle ve vztahu k časovému kroku odhadu PVT a základní předpoklady linearizující model neplatí.

Po inspiraci Chenovým algoritmem (2003) zjednošující 2-D lokalizaci na základě iterativního algoritmu předávání zpráv nad faktorovým grafem obsahující smyčky, byla podrobně studována teorie faktorových grafů a sum-product algoritmu. Nejprve byl zobecněn Chenův algoritmus pro odhad PVT v GNSS, poté byl rozšířen pro pohybový model uživatele a historii pozorování, čímž se stal distribuovaným ekvivalentem rozšířeného Kalmánova filtru bez vektorových a maticových operací.

Rozsáhlými Monte Carlo simulacemi bylo dokázáno, že algoritmus konverguje pro více než 16 sledovaných družic. Dále bylo ukázáno, že přesnost je srovnatelná s EKF pro tři a více iterací za nízké dynamiky. Při zvýšené dynamice, zvýšení počtu iterací sníží chybu odhadu a v extrémních případech vykazuje vyšší přesnost než EKF. Případovou studijí bylo ukázáno, že 7 dB zlepšení citlivosti udržení synchronizace vektorové architektury přijímače se snížilo navrženou metodou na 5 dB. Experimentem s WNav přijímačem vyvinutým na ČVUT v Praze, spoluvyvinutým autorem, bylo demonstrováno, že srovnatelná přesnost s EKF může být dosažena.

Algoritmus je vhodný pro implementaci v logických obvodech a může tak odlehčit CPU navigačního procesoru. V logických obvodech mohou být odhady a predicke produkovány s vyšší frekvencí a posílit tak ještě výkonnost vektorové architektury za vysoké dynamiky. Při implementaci algoritmu v CPU, maticová knihovna nemusí být použita, což vede ke snížení nároků na paměť kódu. Vzhledem k iterativní povaze algoritmu, nutnosti uchovávat pouze střední hodnoty a rozptyly, nároky na datovou paměť jsou rovněž sníženy.