PARTICLE FILTER ALGORITHMS AND EXPERIMENTS FOR HIGH SENSITIVITY GNSS RECEIVERS

J. Dampf [(1)**], N. Witternigg [(2)], M. Schwinzerl [(2)], R. Lesjak [(2)], M. Schönhuber [(2)], G. Obertaxer [(2)], T. Pany [(3)*]

[(1)] Trimble Terrasat GmbH
Haringstraße 19, 85635 Höhenkirchen-Siegertsbrunn, Germany
Email: Juergen_Dampf@trimble.com

[(2)] JOANNEUM RESEARCH GmbH
Steyrergasse 17, 8010 Graz, Austria
Email: Norbert.Witternigg@joanneum.at

[(3)] Universität der Bundeswehr München
Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany
Email: Thomas.Pany@unibw.de

ABSTRACT

Conventional Global Navigation Satellite System (GNSS) receivers are typically designed to operate in open sky conditions and follow the key drivers power consumption, size, and accuracy. In the civil domain, especially with the era of positioning in mobile phones and car navigation systems, the need of high-sensitive and robust GNSS receivers has grown, as they are typically operated in more challenging environments within urban areas, under canopy or even indoors. The past answers to the market needs have been vector tracking (VT) concepts and sensor fusion with focus on GNSS and Inertial Navigation Systems (INS). This paper shows initial results of the novel GNSS only positioning method 'Direct Position Estimation (DPE)' using a Bayesian Particle Filtering (PF) approach for estimating the users position, implemented in a software based GNSS receiver.

1. INTRODUCTION

Today's mass market GNSS receivers are typically designed and optimized for dedicated applications. Mass marked receivers are driven by power consumption and size, while for example geodetic receivers need to achieve very high accuracies down to the centimetre level by usage of differential corrections and carrier phase measurements of the GNSS signals. Geodetic receivers are designed to be operated under open sky conditions, depending on very vulnerable carrier phase observations, while other receivers are designed to achieve higher availability and robustness in applications like car navigation systems and mobile phones. Such GNSS receivers are more likely operated in challenging environments and typically depend more on code based positioning techniques. In the latter case of GNSS receivers, higher sensitivity, integrity and robustness play a more dominant role, as GNSS signals can be blocked, weakened, or influenced by environmental factors like multipath. Not only environmental sources cause a degradation of the signal, also a trade-off is made regarding the grade of hardware components to fulfil the application needs. As an example, mobile phones use very cheap low grade GNSS antennas causing loss of signal strength. Long integration times are applied in order to collect more signal energy over time and achieve the user requirements in sensitivity and robustness of reception. Vector tracking and sensor fusion with focus on GNSS and INS have been implemented and became standard.

But all the afore mentioned GNSS receiver concepts and todays mass market approaches share the same philosophy of a 2-step positioning approach [1], where the first step refers to synchronisation of the GNSS signal parameters code-phase τ and Doppler frequency f_d independently for each tracked GNSS signal, while the second step refers to the positioning (trilateration) of the signal. The complex baseband model of the received signal $x(t)$, considering only line-of-sight signals, is a superposition of M signals and zero-mean additive white Gaussian noise (AWGN) $n(t)$

$$x(t) = \sum_{i=1}^{M} a_i c_i(t-\tau_i)\, d_i(t-\tau_i) - \exp(j2\pi f_{d,i} t) + n(t) \tag{1}$$

where a refers to the signal amplitude, c to the Direct Sequence Spread Spectrum (DSSS) Pseudo Random Noise (PRN) code sequence and $d \in \{-1,1\}$ to the navigation data signal containing for example the satellite orbit parameters. In case of data free pilot signals, the navigation data bit can be considered as $d = 1$. In order to track the signal parameters of interest, tracking loops continuously try to align an internally generated replica signal, which is based on the estimates $\hat{\tau}$ and $\widehat{f_d}$ from the previous integration epoch, thus resulting in a two-dimensional state estima-

* formerly IGASPIN GmbH, Reininghausstraße 13a, 8020 Graz, Austria
** PhD at TU Graz, Institute of Geodesy, Steyrergasse 30, 8010 Graz, Austria and (*)

tion problem [1]. Based on these measurements the second step estimates the users' navigation solution in terms of Position, Velocity and Time (PVT). Typically, this is done by means of a Least Squares (LSQ) adjustment, elevation and/or signal strength (C/N0) dependent filtering using weighted LSQ (WLSQ) or Kalman-Filtering (KF). The KFs typically operate in the PVT domain and can be updated with LSQ adjusted position and velocity estimates in a loosely coupled concept or with pseudorange and Doppler observations in the tightly coupled concept. Closing the GNSS internal signal tracking loops via the navigation solution leads to an ultra-tightly or deeply coupled system as shown in Figure 1 with a dash-dotted line.

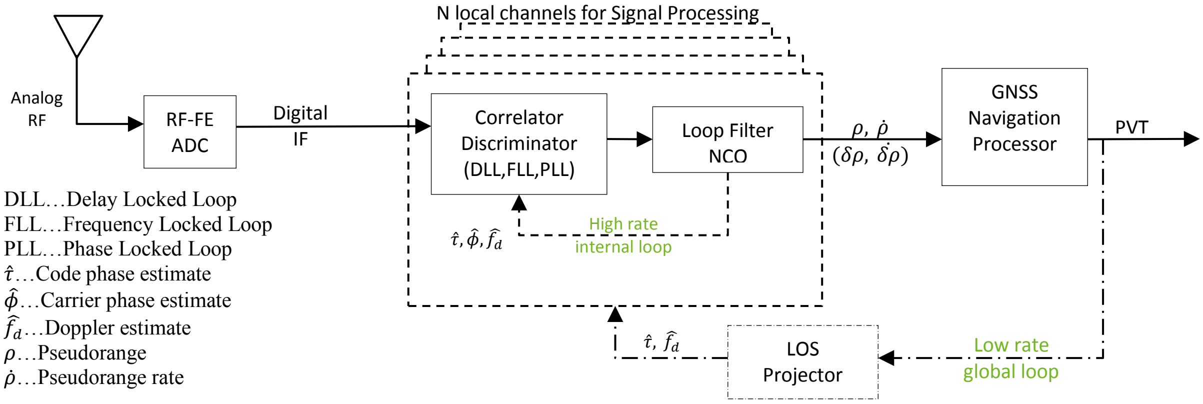
N local channels for Signal Processing
Analog RF
RF-FE ADC
Digital IF
Correlator Discriminator (DLL,FLL,PLL)
Loop Filter NCO
ρ, $\dot{\rho}$
($\delta\rho$, $\delta\dot{\rho}$)
GNSS Navigation Processor
PVT
$\hat{\tau}, \hat{\phi}, \hat{f}_d$
High rate internal loop
$\hat{\tau}$, $\hat{f}_d$
LOS Projector
Low rate global loop
DLL…Delay Locked Loop
FLL…Frequency Locked Loop
PLL…Phase Locked Loop
$\hat{\tau}$…Code phase estimate
$\hat{\phi}$…Carrier phase estimate
$\hat{f}_d$…Doppler estimate
ρ…Pseudorange
$\dot{\rho}$…Pseudorange rate

Figure 1. Conventional GNSS receiver tracking architecture (solid) and vector tracking (dash-dot.), redrawn from [7]

$$\begin{bmatrix}\hat{\tau}_i \\ \hat{f}_{d,i}\end{bmatrix} = \begin{bmatrix}\hat{H}_{\rho,i} \\ \hat{H}_{\dot{\rho},i}\end{bmatrix} \hat{X}_N \qquad (2)$$

Direct Position Estimation (DPE) as a novel and promising GNSS positioning concept leaves the common 2-step approach and instead performs a single PVT estimation step directly from the signal samples. Starting with the PVT solution the intention here is to calculate a correlation weight from a set of GNSS signals dedicated to one single PVT solution. This is achieved by projecting the PVT solution into the Line-of-Sight (LOS) direction of each signal. Based on the projected code-phase $\hat{\tau}$ and Doppler $\hat{f}_d$ a replica signal for each GNSS signal is generated. The resulting complex valued correlation values are connected via the geometric relationship of the PVT solution and can be accumulated, thereby generating a correlation value for the fed back PVT solution. Based on the fact, that $\hat{\tau}$ and $\hat{f}_d$ depend on eight states of the PVT (three states for the position, three states for the velocity, and two states for receiver clock), we end up in an eight-dimensional estimation problem [1].

2. GNSS DPE-RECEIVER ARCHITECTURE

As DPE leaves the domain of classical tracking architectures, the new concept was integrated into the highly flexible and performant software based GNSS receiver SX3. In order to keep the implementation effort low, it was tried to reuse existing functionality of the software receiver. The software receiver already implements advanced positioning methods like vector tracking, which already closes the loop of PVT feedback to the internal tracking loops, one major aspect of DPE. This paper describes the implementation of a non-parametric Bayesian filter with focus on a particle filter for the PVT estimation, shortly named BDPE: Each particle of the particle filter represents a PVT state. In comparison to vector tracking, where only the best PVT estimate is fed back for replica signal generation, for BDPE all particle states would have to be fed back to generate PRN replicas for each individual state. When working with an eight-dimensional optimization problem, a large number of particles (possible user states) is inevitable, ending up in a large number of replica signals to be generated and correlated. As this direct correlation approach would be computational expensive, an indirect and more efficient approach was considered as shown in Figure 2. The GNSS receiver was configured as a GPS L1 vector tracking receiver, while the PVT estimate from the particle filter is used as feedback. The PVT (state) estimate $\hat{X}_N$ is projected into LOS by using (2) in order to observe the code-phase and Doppler estimate, which are used as input to the replica signal generation within each signal tracking loop i. $\hat{H}_{\rho,i}$ and $\hat{H}_{\dot{\rho},i}$ are design (geometry) matrices obtained from the satellite ephemeris and navigation solution.

Additionally a FFT based Multi-Correlator (MC), as shown in [2], was set up, which uses post-correlation values as input to the FFT and allows also for long integration times (non-coherent in case of unknown data-bits and coherent in case of pilot signals or data signals where the navigation data bit has been wiped off). The MC maps are centred around the fed back PVT estimate and span over a defined code-phase and Doppler range. The obtained synthetic multi-correlator results contain the well-known triangular shaped correlation function in case of GPS L1 C/A for each tracked channel N, as symbolically shown in Figure 2. The idea of DPE is to correlate against a replica signal which is a superposition of all signals to be tracked. To achieve this in post-correlation, the corresponding correlation values at the distinct code-phase and Doppler position of the MC map are interpolated and accumulated. The FFT based approach introduces a finite resolution and finite span of code-phase and Doppler values. Both parameters directly influence the processing complexity and must be chosen carefully to achieve real-time performance with DPE. The interpolation of choice is a simple bilinear interpolation between the MC grid points. Alternatively, a sinc-interpolation would allow a more accurate interpolation as it better describes the sinc-characteristics of the Doppler, but with the cost of higher computational complexity. The interpolated correlation values are accumulated to a particle weight and are used as input to the update step of the PF. Practical aspects of DPE receivers are also discussed in [8].

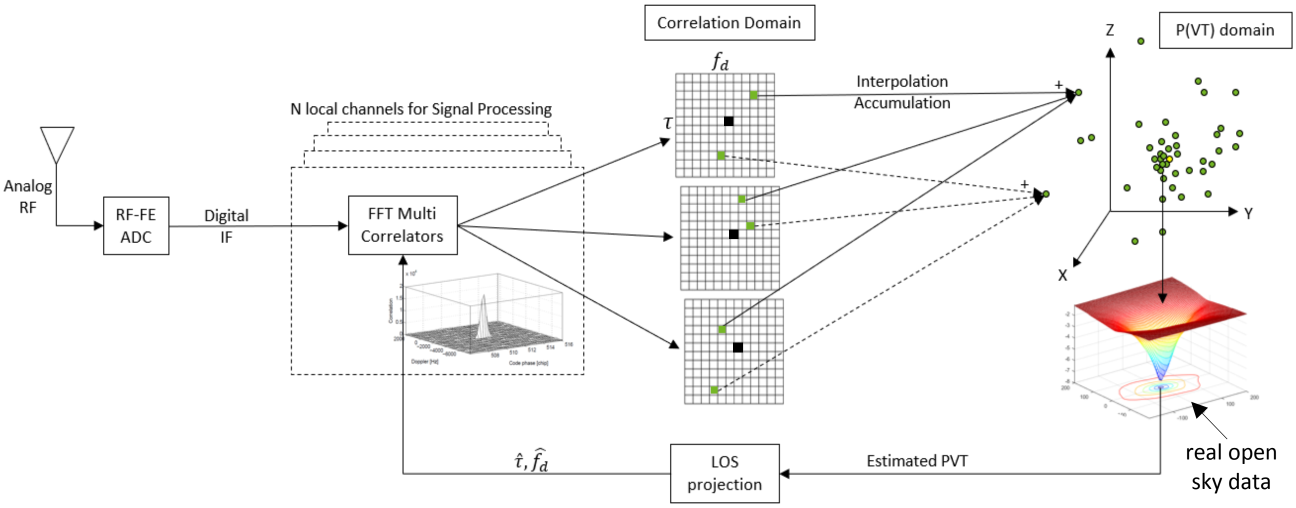

Figure 2. BDPE receiver architecture showing the fed back PVT estimate from the particle filter, LOS projection, multi-correlation by the individual tracking channels and interpolation /accumulation to achieve a particle weight

3. PARTICLE FILTER

The implemented PF operates on inter-independent particles (i.e. eight-dimensional state vectors) which represent an estimate (or "guess") of the system's overall state in the PVT space. For each particle, an associated scalar weight encodes a quality/likelihood estimation which is also used to calculate a weighted, centre-of-mass (CoM) mean over all particles $\langle p \rangle_w$ (and the associated weighted variations/standard deviation σ_w) in an ensemble, which can be interpreted as $\widehat{X_N}$.

After bootstrapping the PF operation by (stochastically varying and) distributing a "cloud" of particles around an estimated initial PVT state, the particles in each filtering step are first subject to operations that shape the cloud (i.e. resampling, propagation and filtering) before the weights of the particles are updated according to the results of the LOS projection. Thus a new CoM PVT estimate $\langle p \rangle_w$is fed back to the GNSS receiver as an updated estimated result of the user state.

FILTERING

The direct availability of a scalar weight for each particle allows the straight-forward implementation of filtering schemes to incorporate external information and known constraints which the system has to obey. If maps or sensor readings are available for the identification of forbidden (or at least unlikely) areas in the position or velocity space, then particles located in such areas can be marked by reducing their weights. In a pedestrian scenario a sidewalk would for example be a very likely area, but for bicycle and automotive scenarios increasingly less so. Other examples for this approach include walls and obstacles obtained from building maps, floor plans or proximity sensors. Transformations between different coordinate systems and reference frames allow the application of filter constraints but may significantly contribute to the costs of a PF implementation, both in terms of complexity and runtime/performance budgets.

INITIALIZATION

The particle filter needs an initial PVT state, where a set of particles (possible user states) is drawn from. The implemented filter supports normal, uniform, or equidistant distributed particles. This initial and approximate PVT state can be obtained from different sources, for example from the mobile phone network. The presented implementation uses an initial PVT solution obtained from a conventional Single Point Position (SPP) approach running in parallel to the PF solution.

PROPAGATION

PF processing steps correspond to discrete points in time (i.e. epochs). Hence measures to propagate the particle cloud along a trajectory in the PVT space between these points in time are needed. In the PF implementation at hand, a simple first-order model in the position domain and in the clock error, i.e. $\vec{x}[t+\Delta t] \approx \vec{x}[t] + \Delta t \cdot \vec{v}[t]$ and $ClkErr[t+\Delta t] \approx ClkErr[t] + \Delta t \cdot ClkDrift[t]$ is used for propagating between two subsequent filtering steps separated by Δt. Each particle is propagated using its "own" values for velocity and clock drift, therefore allowing the particle cloud to change its shape depending on the distribution of the internal state variable. The application of a limited stochastic variation to each particle in form of a process noise contribution (and the standard deviation describing the spread) has to reflect the connection between the state coordinates imposed by the propagation model. This is also crucial with respect to the assumption that the particle cloud would propagate, at least in good approximation, towards the result of the CoM PVT state at time t+Δt if the same translation is applied using the weighted mean velocity and clock drift as parameters, i.e. $\langle p[t+\Delta t]\rangle_w \approx \langle p\rangle_w[t+\Delta t]$.

RESAMPLING

The processes described so far can contribute to a "dilation" of the particle cloud, e.g. by reducing the number of particles providing support for $\langle p\rangle_w$ due to filtering or by reducing the sampling density via phase-space volume increases due to the propagation, both giving rise to the need for a "resampling" step. During the resampling a new particle cloud is generated which should ideally offer a higher concentration of particles near regions of the PVT space associated with higher weights. In the PF implementation discussed in this paper, this has been realized using an importance sampling scheme where each of the N particles is assigned an (arbitrary but unique) index $i \in [0, N]$ so that for any p_i a quantity $S_i = \sum_{j=0}^{i} w_j$ can be calculated from all the weights of particles with indices $j < i$.

Note that S_i monotonically increases (regardless of the chosen indexing scheme) and is limited by the interval $[0, S_N]$. By generating a random number y from a uniform distribution on this interval and finding the particle with the first index k so that $S_k > y$, we have a much higher chance of hitting a particle which contributes a large weight to the ensemble. This is because the "jumps" in S_i correspond to the increase in weight by p_i. Selecting M existing particles from the particle cloud in this fashion and generating copies of them by applying some statistical variation allows generation of a new particle cloud with approximately the same CoM configuration but smaller variation around the regions with higher weight. Note that resampling also allows the change of the size of the particle cloud as N=M is not a requirement for this scheme. Particle filter concepts are described in [6].

LOS PROJECTION AND UPDATE OF WEIGHTS

After applying propagation, filtering and potentially resampling, new GNSS information is taken into account. This is done in turn for each GNSS channel measurement by calculating the LOS between the user state and the satellite (S) and then by finding the code-phase and Doppler $\tau_i^{(S)}$ and $f_{d,i}^{(S)}$ for each individual particle p_i by projecting the vector difference between the user state and the particle onto the LOS. $\tau_i^{(S)}$ and $f_{d,i}^{(S)}$ are then used to evaluate the complex-valued multi-correlator map corresponding to the currently evaluated satellite.

In the currently studied implementation of the PF, a temporary weight $\widetilde{w_\iota}$ is calculated for p_i by summing up the $I^{(S)} \coloneqq I^{(S)}\left(\tau_i^{(S)}, f_{d,i}^{(S)}\right)$ and $Q^{(S)} \coloneqq Q^{(S)}\left(\tau_i^{(S)}, f_{d,i}^{(S)}\right)$, respectively in-phase and quadrature components of the multi-correlator maps over all satellites, i.e. $\widetilde{w_\iota} = \sum_{(S)}\left(I^{(S)^2} + Q^{(S)^2}\right)$, leading up to an updated weight by means of $w_i[t+\Delta t] = u(\widetilde{w_\iota}, w_i[t])$. Different implementations for the mapping u have been studied, including taking $\widetilde{w_\iota}$ directly and dampening the new weight by multiplying it with an exponentially suppressed previous weight. Using the new weights, an updated estimated user state can be calculated and the resulting $\langle p\rangle_w$ can be fed back to the GNSS receiver for the next iteration.

PERFORMANCE & SCALABILITY IMPLICATIONS

The time budget for a single PF iteration is for real-world applications constrained by the timing limits of the GNSS receiver. As with other Monte Carlo based approaches, the number of samples N (i.e. particles) should however be large enough to have confidence in the statistical significance of the derived results.

For the steps outlined above, algorithms and approaches have been chosen that scale with $\mathcal{O}(N)$ or (in the case of resampling which requires a search over the interval $[0, S_N]$) at worst $\mathcal{O}\left(N \cdot log(N)\right)$. With N typically in the ballpark of 10^4 to 10^5 particles and while using off-the-shelf consumer-grade computing systems, the scaling behaviour and run-time costs have been proven to be compatible with such a soft real-time scenario.

Furthermore, the inter-independentness of the particles (in combination for example the choice of finding $\widetilde{w_\iota}$ via a dependency-free sum) allows straight forward parallelization of these numerically intensive portions of the PF in both single-instruction-multiple-data (SIMD) and shared-memory/multi-threaded fashions. This is especially relevant with respect to the prospect of commercially feasible implementations of even more complex PFs using the increasingly available low-cost, low-power, multi-core and super-scalar System-On-A-Chip (SOC) architectures.

4. TEST SETUP

The DPE / PF is a software only approach and was implemented as a user library in the SX-3 GNSS receiver. The SX-3 of IFEN GmbH is a complete receiver package including single or dual radio frequency (RF) input that allows API access to all receiver engines as well as sensor data and assistance. The receiver experiments were performed from recorded data of several test runs with both fixed receiver positions and test drives with a vehicle in different environments.

A test track of about 30 km length was chosen to contain sections of varying environmental ‘difficulty’ for the software, from open sky scenarios to highway, suburban and urban scenarios. Street canyons with multipath reception were observed as well as bridges with underpasses and a passage in a tunnel.

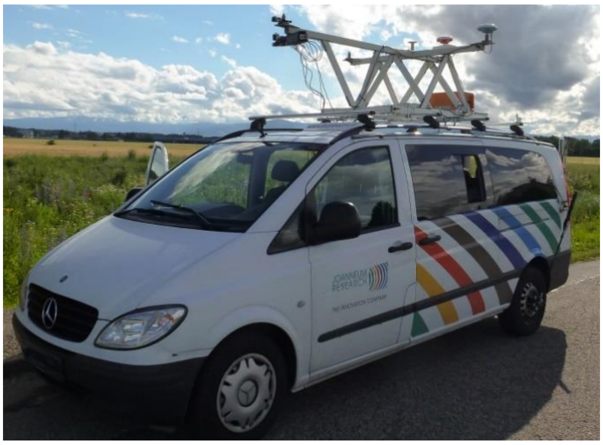

Figure 3. Measurement van with antenna setup on the roof and high-grade IMU (image: JR)

The measurement setup for the test receiver consisted mainly of a roof mounted geodetic multi-frequency antenna (NavXperience 3G+C) and the RF frontend of the SX-3 receiver for the recording of the data. For the reference measurements, a geodetic JAVAD Sigma receiver was used to measure code and carrier phase measurements in combination with a navigation grade iMAR iNav RQH inertial measurement unit. The JAVAD Sigma receiver and the SX3 RF frontend were connected to the same antenna using a passive antenna splitter. As GNSS base station, the IGS station GRAZ (LEICA GRX1200+GNSS) was used. The data was post-processed using Waypoint Inertial Explorer.

The test drive was performed in the scope of the project iRTK with a completely different scientific topic where ultra-tight coupling RTK was investigated using the IFEN SX3 software receiver in combination with two MEMS IMUs from Xsens. Figure 3 shows the measurement van with the roof mounted antennas. The red antenna was used for this investigation, the iMAR IMU was mounted below the orange box to avoid direct sun. The second antenna and the wheel sensor were used for a different investigation.

5. RESULTS

The herein presented results correspond to a dataset gathered with the previously described test setup in Graz, as shown by the yellow trajectory in Figure 4. The BDPE results have been compared to a standard SPP solution, which is based on elevation dependent WLSQ, and an Extended Kalman Filtered (EKF) vector tracking solution, noted in the legends as VT-EKF.

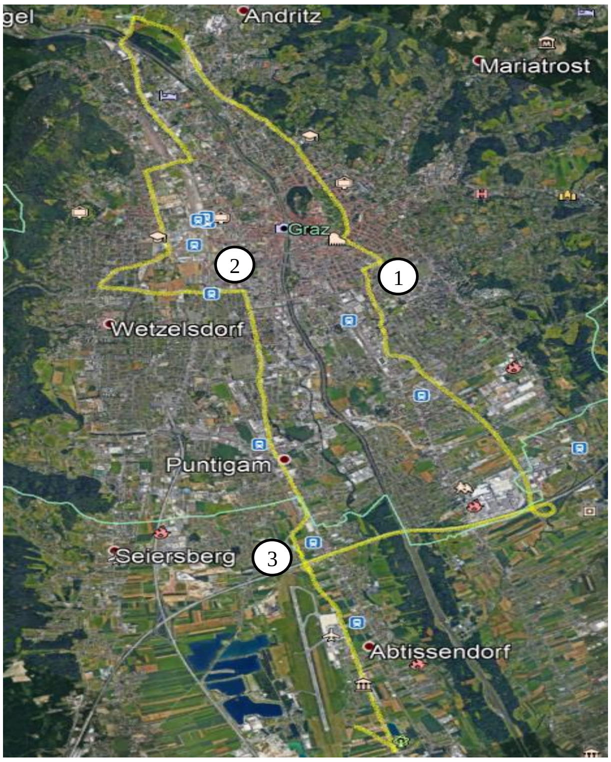

Figure 4. Vehicle track from the south through the city Graz to the north and back, including open sky, highways, sub urban, urban (canyons), bridges and tunnels. Numbers refer to Figure 6, Figure 8 and Figure 10. © 2017 Google, Image Landsat / Copernicus

Three significant sequences of one dataset (locations marked with a white circle 1-3 and refer to the related figures) have been analysed and the BDPE solution was compared to the high-grade reference solution, to a SPP and an advanced vector tracking solution, as shown in Table 1. Some PF parameters for these first results are based on an empirical evaluation. For the results shown only GPS L1 C/A has been evaluated. All solutions use the same coherent integration time of $T_{coh} = 20ms$. In parallel to the drive a reference station in Graz recorded the navigation data bits for later investigation with longer coherent integration times by doing da data wipe off on data signals. The reported values in Table 1 refer to the whole track through the city. For sake of simplicity the cross-track (XTrack) error to the reference trajectory has been evaluated, which is the shortest geometric distance to the reference. It can be observed, that SPP as well as VT encounters a lower availability compared to the BDPE solution. 'Availability' in Table 1 refers to the number of available navigation solutions in the dataset. VT navigation solutions with a cross-track error >= 50 m are considered as invalid (not available) and thus VT also may not achieve 100% availability. This constraint is introduced to derive meaningful statistical values for VT in case of a diverging filter. In case of SPP, a navigation solution is not available if less than four valid satellites are available. BDPE benefits from the fact that it correlates against a set of available signals, even if the LOS signal is currently blocked. This leads to a higher availability of solutions, if the true PVT state is covered by the particle cloud and due to the higher sensitivity.

Table 1. Results of the complete run

Solution	T_{coh}	1σ-XTrack Error (<50m Err)	avail. (<50m Err)
SPP	20 ms	3.62 m (σ|98,1% SPP)	98,1 %
VT-EKF	20 ms	1.39 m (σ|98,7% VT)	98,7 %
BDPE	20 ms	1.96 m (σ|100% DPE)	100 %

In open sky conditions the BDPE solution shows a similar performance as the VT solution, while both are less noisy compared to SPP due to the filtering attribute. The BDPE results are outlined in three different scenarios:

- Multipath caused by urban canyon
- Signal weakening and outages caused by bridges
- Signal blockage within a tunnel

URBAN CANYON

The urban canyon focuses on a sequence with buildings of about six floors on both sides of the street, as shown in Figure 6. The investigated road is the Steyrergasse in Graz driving from West to East. With the current parametrization and implementation, it can be clearly seen, that the BDPE solution is still vulnerable in case of multipath, even if the effect is reduced compared to the SPP solution.

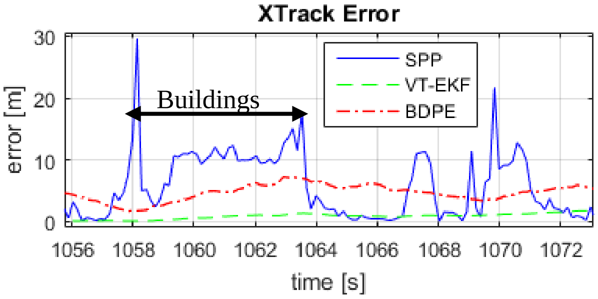

Figure 5. Cross-track error of the urban canyon.

It was shown with a theoretical analysis and simulations in [3], that DPE is similar vulnerable to short multipath with a relative multipath delay up to 0.4 chips (~120m for GPS C/A on L1). It is assumed that this applies to the presented multipath scenario and explains the deviation in Figure 5.

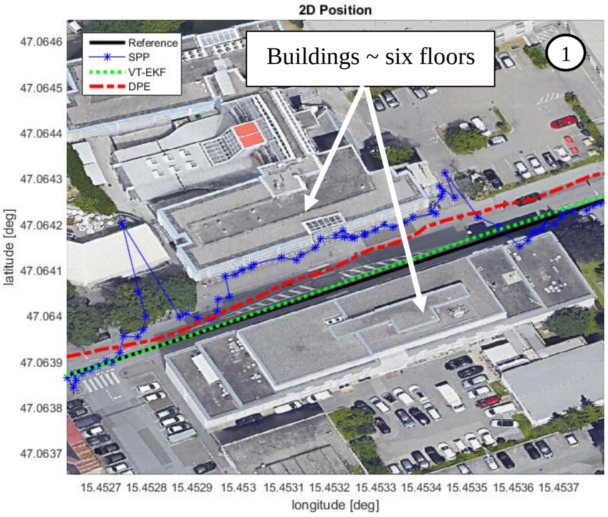

Figure 6. Results within an urban canyon (blue=SPP; green=VT; red=BDPE). © 2017 Google, Image Landsat / Copernicus

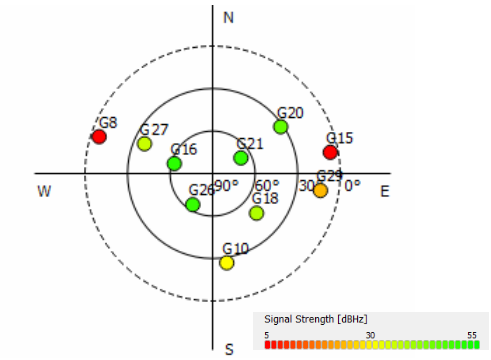

Figure 7. Satellite const. GPS L1 C/A in urban canyon

BRIDGES

The higher sensitivity and filtering attribute of the BDPE leads to an improved behaviour under bridges, where signals are weakened or completely blocked. Figure 8 shows a scenario with two railway bridges and short outages of GNSS signals.

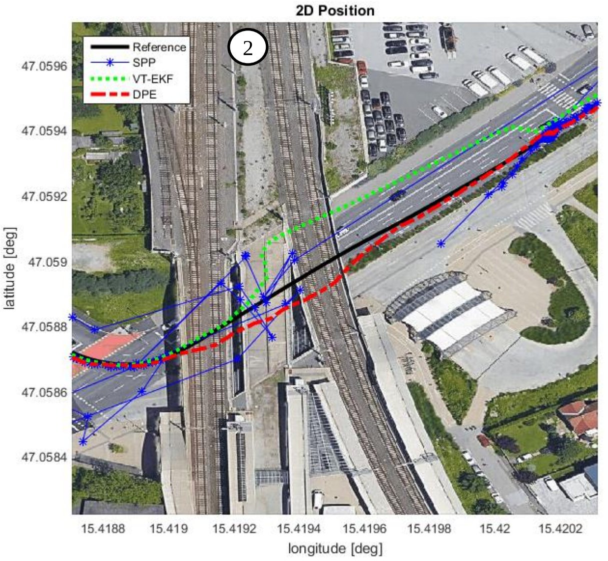

Figure 8. Railway bridges at Don Bosco, Graz, Austria driving to the North-East (blue=SPP; green=VT; red=BDPE). © 2017 Google, Image Landsat / Copernicus

As expected in such a situation less SPP solutions are available or are connected with large errors as many signals lose lock and cannot contribute to the solution. The SPP solution is unfiltered and thus varies significantly, compared to the other solutions. Vector tracking delivers a continuous solution, but it seems that the tracking loops start to diverge from second 3223 till 3226 and re-lock on the signals from that point on. In comparison the BDPE does not tend to diverge and converges quickly after second 3225. While the BDPE filter propagates underneath the bridge, it still accounts for signal contributions from all satellites. In such a situation the process noise of the filter must be chosen in a way to represent the true uncertainty, because a 're-lock' on the PVT state can only be established if the particle cloud still covers the true state.

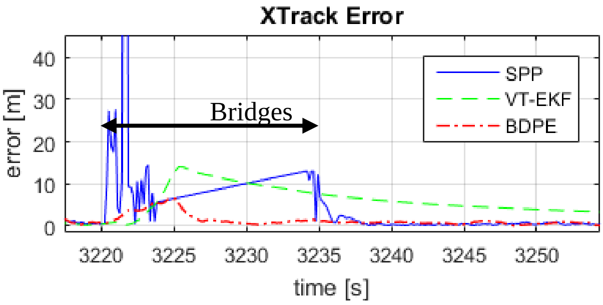

Figure 9. Cross-track error of under railway bridges

TUNNEL

Based on the fact that DPE should allow for higher sensitivity, a tunnel was considered in the measurement run. The passed tunnel is not deep under the earth, nevertheless it is assumed that only multipath components enter at both sides of the tunnel and contribute to the solution. Figure 10 shows the behaviour of the different solutions when entering a tunnel, while the yellow box highlights the tunnel area. From both plots it can be seen, that SPP delivers only a few estimates that are off by more than 50 meters. Within the shown time interval in Figure 12, SPP reported eight to nine satellites in nearly open sky conditions before and after the tunnel but found no solution in the tunnel. BDPE reported ten tracked satellites. One satellite had an elevation < 5 deg which causes a weak and noisy signal. This signal was not available for SPP, was sometimes available for the vector tracking solution, and was always available in BDPE. All solutions use the same elevation cut-off angle of 1 deg.

Figure 11 and Figure 12 show that VT has tracked some signals at least several meters into the tunnel. Figure 11 shows that the vector tracking solution is updated in a meaningful way until second 3900, while at this time in Figure 12 more than eight signals have been reported in track by VT. The number of tracked satellites has to be observed with caution, as they are considered to be tracked until the C/N0 drops below a defined threshold. The C/N0 estimation is typically averaged and thus the response to the unlocked state can be delayed. Nevertheless, the VT loops have been kicked out of their working region and the solution diverges. In comparison, the BDPE still correlates against all supported signals and thus reports the full number of available satellites. The results show that the cross-track error of BDPE implementation remains comparably small, but a close look at the exit shows an overshoot in the along-track direction, which results from the propagation of the filter through the tunnel.

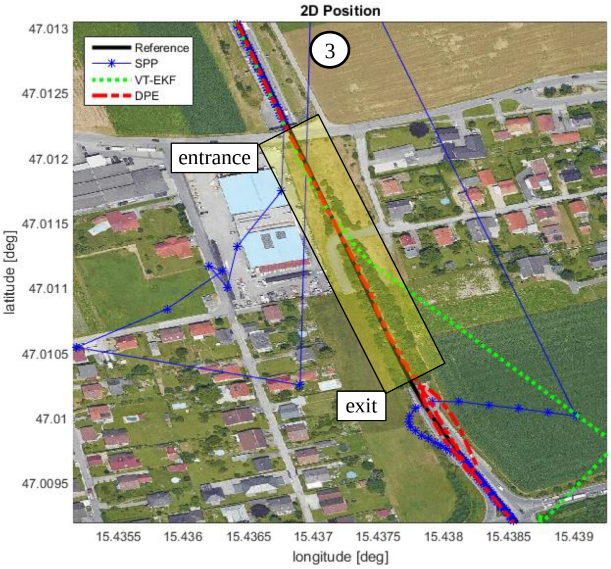

Figure 10. Tunnel in Feldkirchen near Graz, Austria driving to the South (blue=SPP; green=VT; red=BDPE). © 2017 Google, Image Landsat / Copernicus

After exiting the tunnel the SPP and the BDPE solution converge quickly, which indicates a proper selection of the process noise parameters of the BDPE filter to cover the true PVT state at time of exit. The shown vector tracking setup does not account for resetting the tracking loops and solution, and thus does not recover at the tunnels exit.

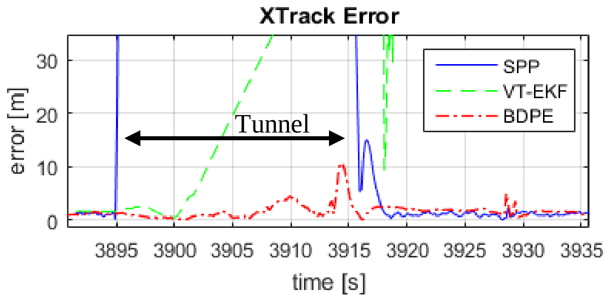

Figure 11. Cross-track error through the tunnel

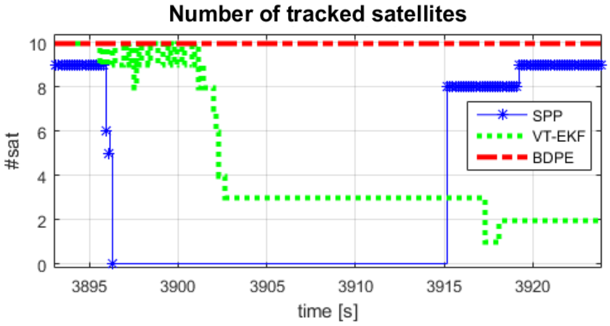

Figure 12. No. of tracked satellites through the tunnel

6. CONCLUSION

This paper discusses a first implementation of BDPE into a commercial software based GNSS receiver without discussing in detail the tools to achieve a real-time behaviour, as many optimizations in terms of parallelization still can be done. Initial results with an empirically parametrized particle filter have been presented on a dynamic reference data set through the city of Graz with focus on challenging environments. The results compare BDPE both to a conventional WLSQ epoch based SPP as well as to an advanced Kalman filtered vector tracking solution. It should be mentioned, that both methods (SPP and vector tracking) are on higher quality level in terms of implementation and parametrization, as these methods have been developed and improved over years. BDPE shows in multipath prone urban canyons an improvement compared to SPP, but cannot compete against a vector tracking solution which applies outlier detection methods. BDPE shows the most benefit in areas where high sensitivity is needed. Two examples of passing under bridges and through a tunnel already give an indication of the capability of this novel method. In future work the processing performance will further be optimized by a better utilization of parallelization and the algorithms will be extended to other GNSS systems and frequencies to fully exploit the potential of this method in challenging GNSS scenarios, for example under canopy or indoors.

7. ACKNOWLEDGEMENTS

Thanks go to Prof. M. Wieser, Institute of Geodesy at the Technical University Graz (TUG) for supporting the PhD and having a very cooperative work together with the institute. Thanks go also to IFEN which contributes by supporting the project with the SX3 software based receiver and IGASPIN as project partner, with support of representative datasets and technical knowledge. Acknowledgements should also go to P. Closas for his fundamental work concerning DPE. The project was funded by the Austrian Research Promotion Agency and executed by JOANNEUM RESEARCH under contract No 847994.

8. REFERENCES

1. Closas, P. (2009). Bayesian Signal Processing Techniques for GNSS Receivers: from multipath mitigation to positioning. Universitat Politècnica de Catalunya, Spain

2. Stöber, C., Kneißl, F., Eissfeller, B., Pany, T. (2011). Analysis and Verification of Synthetic Multi-correlators. Proc. 24th ION GNSS, Portland, OR, pp. 2060-2069

3. Closas, P., Fernandez-Prades, C., and Fernandez-Rubio, J.A. (2009). Cramér Rao Bound Analysis of Positioning Approaches in GNSS Receivers. IEEE Trans. on Signal Processing 57, 3775–3786

4. Axelrad, P. et al. (2011). Detection and Direct Positioning Using Multiple GNSS Satellites. Navigation: Journal of The Institute of Navigation Vol. 58, No. 4.

5. Perez-Fontan F., et al., (2002), Blockage and Multipath Modeling for the Multi-Satellite Navigation Channel in Urban Areas. Dept. Tecnologías de las Comunicaciones, E.T.S.E.T, Vigo, Spain.

6. Arulampalam, S., et al., (2002), A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. IEEE Trans. on Signal Processing, 1053-587X, pp174-188

7. Won., J.-H., Eissfeller, B., Pany, T. (2011). Implementation, Test and Validation of a Vector-Tracking-Loop with the ipex Software Receiver. Proc. 24th ION GNSS, Portland

8. Closas, P., Gusi-Amigó, A. (2017). Direct Position Estimation of GNSS Receivers. IEEE Signal Processing Magazine, 1053-5888

9. Bhamidipati, S., Ng, Y., Gao, G. X. (2017). Multi-Receiver GPS-Based Direct Time Estimation. Inside GNSS, Jan/Feb 2017, pp43-4