PARTICLE FILTER ALGORITHMS AND EXPERIMENTS FOR HIGH SENSITIVITY GNSS RECEIVERS

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

[(1)] JOANNEUM RESEARCH GmbH
Steyrergasse 17, 8010 Graz, Austria
Email: norbert.witternigg@joanneum.at

[(2)] Trimble Terrasat GmbH
Haringstraße 19, 85635 Höhenkirchen-Siegertsbrunn, Germany

[(3)] Universität der Bundeswehr München
Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany

ABSTRACT

The paper describes the development and first real-world test results of a Particle Filter (PF) for a software based GNSS receiver. The filter implements a non-parametric Bayesian filter using the Direct Position Estimation Principle (DPE), thereby eliminating the conventional tracking loops for the purpose of higher sensitivity and robustness of reception. The 'particles' of the filter represent system states with an eight-dimensional approach for 3D-position, 3D-velocity, clock error and clock drift. The Measurements were performed in suburban and urban environments in Graz/ Austria and results were compared to a standard Single Point Positioning (SPP) and advanced Vector Tracking (VT) solution.

1. INTRODUCTION. Particle filtering is an innovative algorithm for the calculation of position and time in high sensitivity GNSS receivers and applies to challenging scenarios involving weak Radio Frequency (RF) signals, obstacles, blocking, multi-path and other constraints. The individual satellites of a GNSS may be affected differently - thus signals from some satellites may arrive in much better quality at the receiver than those from others. Apart from the higher sensitivity, a particle filter should be capable of providing a probability density function of the user position as a realistic estimate for the positioning process.

2. GNSS DPE-RECEIVER. Conventional and todays mass market GNSS receivers acquire and track the satellite signals to obtain the signal parameters code-phase τ and Doppler frequency f_d, then follow a 2-step positioning approach (Closas, 2009-1) where the first step refers to the synchronization of the code phase and Doppler frequency using conventional tracking loops while the second step refers to the positioning (trilateration) of the users' navigation solution in terms of Position, Velocity and Time (PVT). This is done independently for each GNSS signal typically 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). There the KFs typically operate in the PVT domain and are updated with LSQ adjusted position and velocity estimates in a loosely coupled concept or with pseudorange and Doppler observations in the tightly coupled concept (Dampf et al., 2017). The Direct Position Estimation (DPE) concept leaves this 2-step approach and instead performs the PVT estimation step directly from the signal samples; resp. from the multi-correlator maps. The multi-correlator maps are calculated as the cross-correlation function between the user controlled reference signal and the received satellite signals. The cross-correlation function is calculated as a two-dimensional grid of code phase and Doppler offsets (Stöber et al., 2011). This is performed

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

in parallel for all acquired satellites and is repeated for all available systems of a multi GNSS setup. The results of the navigation solution allow for new estimates of code phase $\hat{\tau}$ and Doppler $\hat{f}_d$ which are fed back in a low rate global tracking loop as shown in *Figure 1*. The multi-correlator maps are centred on the fed back PVT estimate and span over a defined code phase and Doppler range. The method performs collectively on all inputs, i.e. all generated correlation maps are interpolated and accumulated for the navigation solution. The interpolation method of choice is a bilinear or sinc interpolation between the multi-correlator grid points. Within the PF the interpolated correlation values are accumulated to form a particle weight for each state variable and are input to the next processing step. The process requires an (initial) approximate knowledge of the user position as well as knowledge of the satellite ephemeris. No independent tracking or channel processing is performed and all correlation results contribute to the updated PVT estimate, with the key advantages of the method being:

- All available correlations and therefor all available signal energy is accumulated.
- Only correlation with the Line-of-Sight (LOS) signal will be performed.

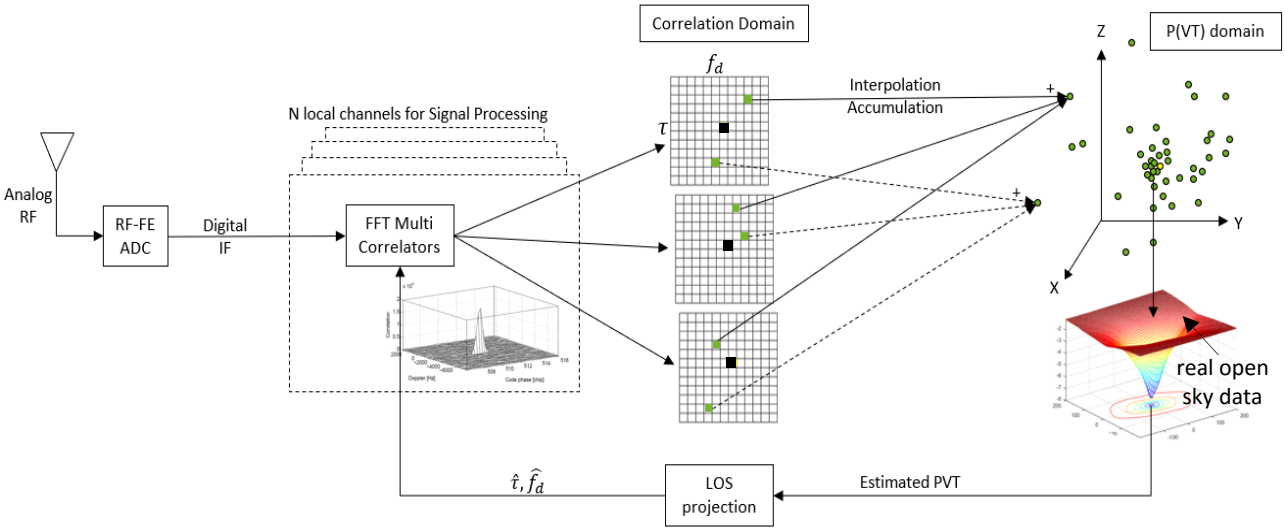

Figure 1: Block diagram of the receiver architecture showing the feedback of the PVT estimate from the particle filter (Dampf et al., 2017, with elements of Won et al., 2011)

3. PARTICLE FILTER. The particle filter implements a quality or likelihood estimator where a large number of 'particles' represent inter-independent samples of the system state in the PVT space. It is a non-parametric Bayesian filter with a particle filter for the PVT estimation (shortly named BDPE). The particles have n degrees of freedom, e.g. eight independent states for 3D position, 3D velocity, clock error and clock drift. Each particle is assigned an associated scalar weight that is used to calculate a weighted, Centre-of-Mass (CoM) over all particles $\langle p \rangle_w$ plus an associated weighted standard deviation σ_w in an ensemble which can together be interpreted as the probability-weighted expectation value of the system state. *Figure 2* shows an example of the processing steps of the PF. The sequence of these processing steps may however slightly vary with different implementations.

3.1. *Initialization.* The particle filter needs an initial guess of the PVT state and an initial uncertainty of the states to populate the particle distribution. The filter supports different distributions of the particle states like Gaussian, uniform and equidistant distribution. Typically a Gaussian distribution has been used. The initial approximate PVT state can be ob-

tained from different sources as for example from the mobile phone network. The presented implementation uses an initial PVT solution

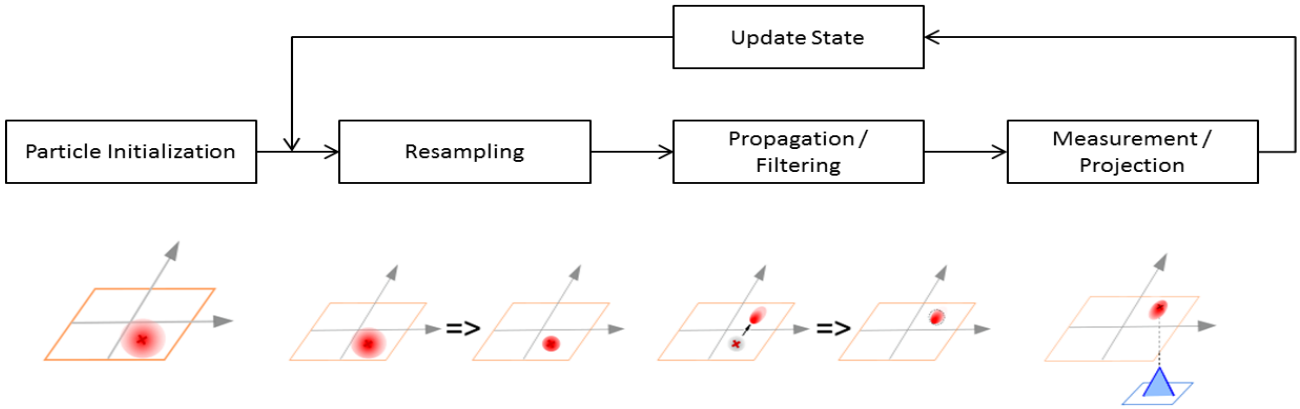

Figure 2: Block diagram of the particle filter showing a symbolic representation of the development stages of the particle cloud

obtained from a conventional SPP approach running in parallel to the PF solution.

3.2. *Resampling.* A resampling step is required if there are no more enough particles available providing support for $\langle p \rangle_w$ due to 'dilatation' of the particle cloud, due to filtering or due to a reduction of the particle cloud during the propagation step. During 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 our implementation we used 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 'steps' in S_i correspond to the increase in weight by p_i (*Figure 3*). Selecting M existing particles from the particle cloud in this fashion and generating copies of them by applying some statistical variation allows generating a new particle cloud with approximately the same CoM configuration but smaller variation around the

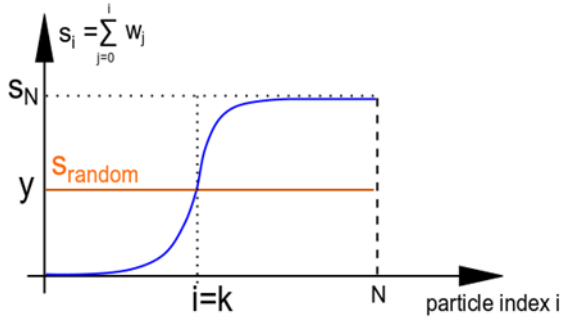

Figure 3: Cumulative distribution of the particle weights with random selection y from a sum S in the interval [0, S_N]

regions with higher weight. Note that resampling also allows changing of the size of the particle cloud as $(N = M)$ is not a requirement.

3.3. *Propagation/Filtering.* The propagation step describes the time evolution of the particle cloud. PF processing steps correspond to discrete points in time (i.e. epochs or shorter intervals). Hence measures to propagate the particle cloud along a trajectory in the PVT space between these points in time are needed. We used a simple linear Galilean transformation between two states in the position do-

main and for the clock error, i.e.

$$\vec{x}_{t+\Delta t} \approx \vec{x}_t + \Delta t \cdot \vec{v}_t \quad (1)$$

$$ClkErr \approx ClkErr_{\,t} + \Delta t \cdot ClkDrift_{\,t} \quad (2)$$

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 $t + \Delta t$ if the same translation is applied using the weighted mean velocity and clock drift as parameters, i.e.

$$\langle p[t + \Delta t] \rangle_w \approx \langle p \rangle_w [t + \Delta t] \quad (3)$$

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.

3.4. *Measurement/Projection.* 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 $\tau_i^{(S)}$ and Doppler $f_{d,i}^{(S)}$ for each individual particle p_i by projecting the vector difference between the User State and the Particle State onto the LOS. $\tau_i^{(S)}$ and $f_{d,i}^{(S)}$ are then used to adjust the complex-valued multi-correlator map corresponding to the currently evaluated satellite. The contributions from different satellites are combined additively.

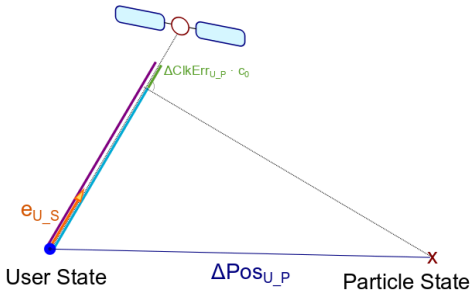

Figure 4: LOS projection in the code phase plane

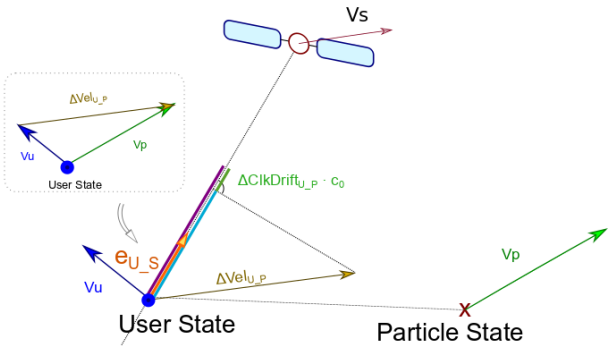

Figure 5: LOS projection in the Doppler plane

Figures 4 and *5* show the principle of the LOS projection with the relations

$$\Delta\tau = < e_{U\_S}, \Delta Pos_{U\_P} > + \Delta clkErr_{U\_P} . c_0 \quad (4)$$

$$\Delta f_d . \lambda = < e_{U\_S}, \Delta \mathrm{Vel}_{U\_S} > + \Delta clkDrift_{U\_P} . c_0 \quad (5)$$

3.5. *Update State.* In our implementation a temporary weight $\widetilde{w_i}$ is calculated for p_i by summing up in-phase and quadrature compo-

nents of the multi-correlator maps over all satellites, i.e.

$$\widetilde{w_i} = \sum_{(S)} (I^{(S)^2} + Q^{(S)^2}) \quad (6)$$

leading to an updated weight of

$$w_i[t + \Delta t] = u(\widetilde{w_i}, w_i[t]) \quad (7)$$

Different implementations for the mapping u have been studied, e.g. taking $\widetilde{w_i}$ 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.

4. IMPLEMENTATION. The time budget for a single PF iteration and real-world applications is 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 the order of (N), or, in the case of resampling which requires a search over the interval [0, S_N], at worst scale with the order of $(N.\log(N))$. Typical values of N during our tests were chosen between 10^4 to 10^5 particles. While using off-the-shelf consumer-grade computing equipment, the scaling behaviour and run-time costs have been proven to be compatible with such a soft real-time scenario.

5. SOFTWARE SIMULATION. The quality of the generated PVT solution from the DPE process is influenced by environmental parameters like for example the number of visible/usable GNSS signals, the quality (i.e. SNR) of these signals, the presence or absence of multi-path, blocking, or fading impairments on the signal, etc. The PF, on the other hand, offers a range of configuration and implementation parameters that directly influence its performance and, ideally, can be used to tune and optimize the fidelity of the PVT estimation. Examples for such parameters are, among others, the statistical properties of the process noise distributions, parametrizations and thresholds for triggering the resampling process, the functional form and the internal parameters used in the mapping $u(\widetilde{w_i}, w_i[t])$ from equation (7) or the maps and constraints used during the filtering steps. The difficulty of studying the complex relationships between these two distinct groups of parameters, especially with a view towards the deployment of the DPE during experimental field trials, motivated the development of a software simulation testbed for qualitative assessments. It uses a set of artificially generated multi-correlator maps, corresponding qualitatively to a mocked/ generated satellite constellation to influence the PF in a scripted way (*Figure 6*).

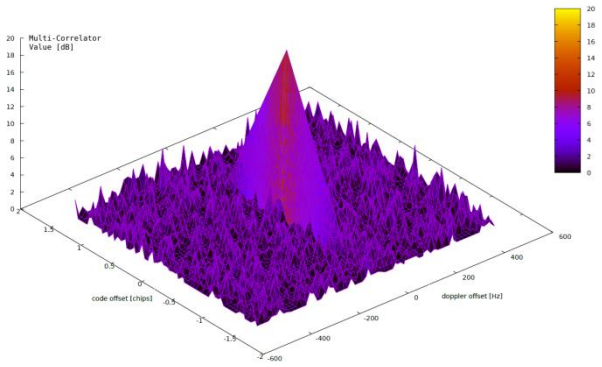

Figure 6: Synthetic correlator map to be used in the simulator

Effects like fading, blocking or multi-path contributions can be applied on these synthetic multi-correlator maps, allowing the study of the filters response under such effects within a controlled environment. Since in the context of the software simulator no actual position estimation is performed, a trajectory for the eight-dimensional PVT space describing the "true" user state has to be provided for each simulation as well.

6. 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 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 real-world test runs with both fixed receiver positions and test drives with a vehicle in different environments.

Figure 7: Measurement van with antenna setup on the roof and high-grade IMU

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 7* 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.

7. RESULTS. The presented results correspond to a dataset gathered with the previously described setup in Graz, as shown with the yellow trajectory in *Figure 8*. The test track of about 30 km length was chosen to

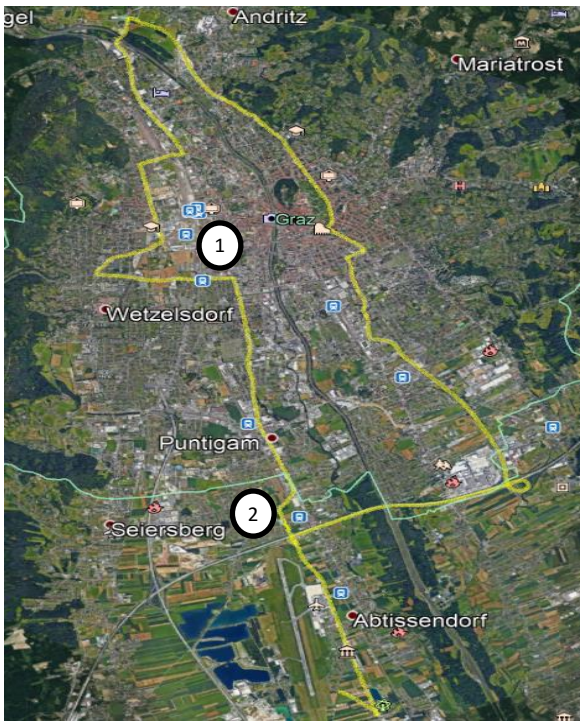

Figure 8:Vehicle track through the city of Graz/Austria. Map image © 2017 Google, Landsat / Copernicus

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. The BDPE results have been compared to a standard SPP solution, which is based on WLSQ, and an Extended Kalman Filtered (EKF) vector tracking solution, noted in the legends as VT-EKF. The comparison with the vector tracking solution was possible because the SX-3 receiver already supports this method. Vector tracking is an advanced 'state of the art' positioning method where the internal receiver tracking loops are closed via the navigation solution.

This makes the method more sensitive and robust to signal outages, because the tracking loops do not work independently from each other. As a simple example it can be explained in a way that the tracking loops of blocked signals are assisted by the navigation solution and by the tracking loops of the remaining strong signals. Two significant sequences of one dataset (locations marked with a white circle 1 and 2 and refer to the related figures) have been analysed in more detail and the BDPE solution was compared for the whole run to the high-grade reference solution, to a SPP and to an advanced vector tracking solution, as shown in *Table 1*. 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 using 'data wipe off' on the GNSS signal. 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

Table 1: Results from the full trajectory

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 %

dataset. VT navigation solutions with a cross-track error >= 50 m are considered 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 deemed available if less than four valid satellites are seen. BDPE benefits from the fact that it correlates against a set of available signals, even if one LOS signal is currently blocked. This leads to a higher availability and sensitivity if the true PVT state is covered by the particle cloud. 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 particle filter behaviour was investigated in more detail in two significant environments:

- Signal weakening and outages under bridges
- Signal blockage within a tunnel

7.1 *Bridges*. The higher sensitivity and filtering attribute of BDPE leads to an improved behaviour under bridges, where signals are weakened or completely blocked. *Figure 9* shows a scenario with two railway bridges and short outages of the GNSS signals.

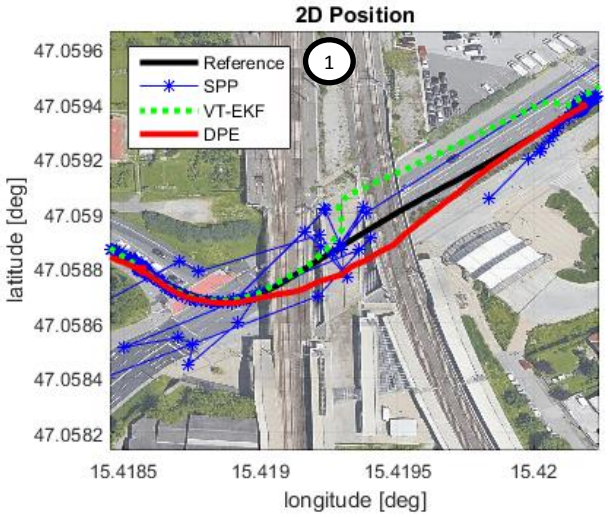

Figure 9: Railway bridges at Don Bosco, Graz, Austria driving to the North-East (blue=SPP; green=VT; red=BDPE). Map image © 2017 Google, Landsat/ Copernicus

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 when compared to the other solutions. Vector tracking delivers a continuous solution, but it seems that the tracking loops start to diverge under

the bridges and relock on the signals afterwards. The BDPE results also show an offset but relock more quickly after the bridges. While the BDPE filter propagates underneath the bridge, it still accounts for signal contributions from all satellites. *Figure 10* shows the behaviour of the weighted particle cloud at three evaluated positions during a pass underneath the bridge.

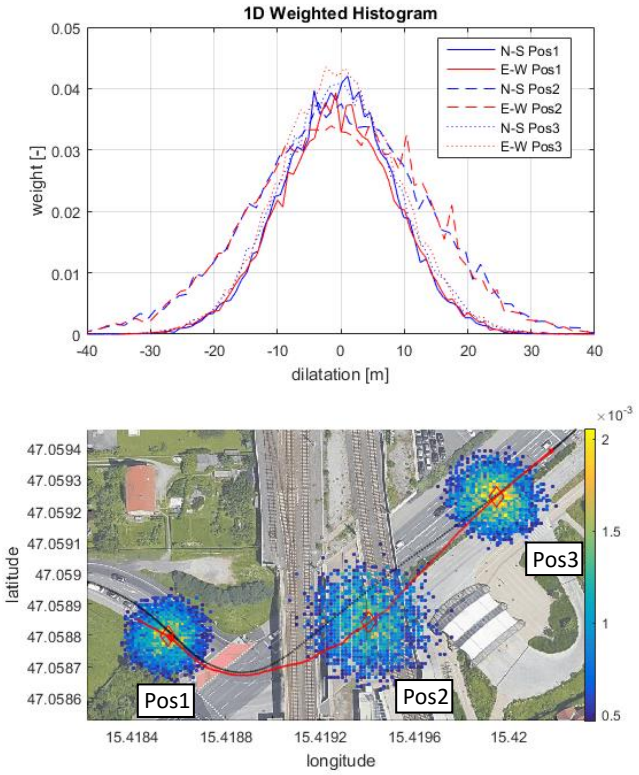

Figure 10: Particle cloud behaviour when passing the bridge, evaluated at three positions. Yellow dots indicate high and blue dots low correlation with the LOS GNSS signal. The reference trajectory is shown in black and BDPE in red. Map image © 2017 Google, Landsat / Copernicus

The upper plot represents the accumulated 1D-histogram of the weighted particles over north-south (N-S) and east-west (E-W) directions, while zero refers to the centre of mass. The lower plot shows the particle clouds at these three positions in the local coordinate frame. Each particle, which equals to a dot in the figure, has a particle weight derived from correlating a replica signal generated with the particle PVT state against the gathered GNSS signal. A high weight (highly probable position) of the particle is shown in yellow and a low weight (less probable position) particle state is shown in blue. The lower plot additionally shows the reference trajectory in black and the BDPE trajectory in red. Pos 1 and Pos 3 show nearly open sky conditions. If the GNSS signals are blocked, the particle variance is growing with the process noise, which also means that the uncertainty of the state estimate grows. This can be seen in both plots by the widened probability function in north-south and east-west directions as well as by the wider distribution of the particles at Pos 2.

7.2 *Tunnel*. The tunnel area is considered a very challenging environment, where only multipath components or very weak direct LOS signals contribute to the navigation solution. *Figure 11* shows the behaviour of the SPP, Vector Tracking and BDPE solutions.

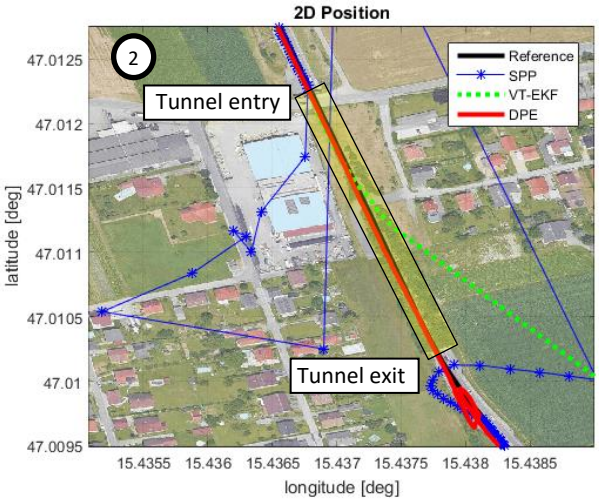

Figure 11: Tunnel in Feldkirchen near Graz, Austria driving to the South (blue=SPP; green=VT; red=BDPE). Map image © 2017 Google, Landsat / Copernicus

When driving into the tunnel, very inaccurate and later no SPP solutions at all were found because not enough GNSS signals could be tracked. The VT solution achieved a better performance and delivered a valid navigation solution at least several meters into the tunnel. After approximately 70 m into the tunnel, also the VT solution started to diverge because of the lack of available GNSS observations, as shown by the number of tracked satellites in

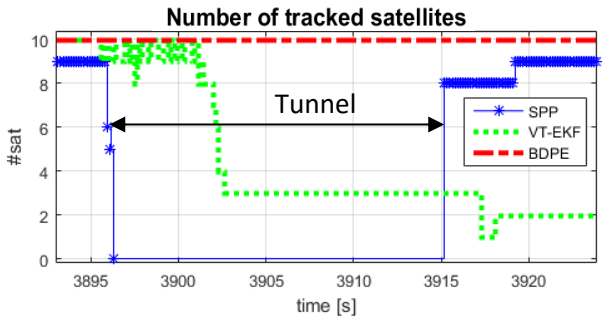

Figure 12: Number of tracked satellites in the tunnel

Figure 12. It is expected that the BDPE solution performs better in such a difficult signal environment as it does not depend on single channel signal tracking and correlates against all GNSS signals that are known from the ephemeris data and particle state. Accordingly, BDPE reports ten tracked satellites in *Figure 12*. As described above, the particle filter propagates the particles by a first order model and increases the particle variance based on the process noise parameters. This behaviour is reasonable because in case of no signal, the uncertainty of the navigation solution grows with time and the filter tries to cover this uncertainty with a lager variance of particles. But this comes along with the side effect of having a lower particle density around the true state when keeping the number of particles constant. All solutions use the same elevation cut-off angle of 1 deg. Note that one satellite had an elevation < 5 deg which caused 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. The lower plot in *Figure 13* shows the particle clouds on five positions (Pos 1 – Pos 5) along the trajectory through the tunnel, the upper plot shows a one-dimensional weighted histogram over north-south (N-S) and east-west (E-W) directions in the local frame at (Pos 1 – Pos 3). Pos 1 is located about 20 m in front of the tunnel entry in near open sky conditions, Pos 2 is about 20 m into the tunnel, Pos 3 about 150 m into the tunnel, Pos 4 is located 20 m after the tunnel exit and Pos 5 finally experiences near open sky conditions again. From Pos 1 to Pos 3 it can be clearly seen that the position uncertainty and thus the particle variance grows with no or weakened signals over time. Pos 2 and Pos 3 are very interesting points as they are located within the tunnel. It can be observed that the probability functions in north-south and east-west directions still describe a normal (like) distribution with an increased variance compared to the open sky solution.

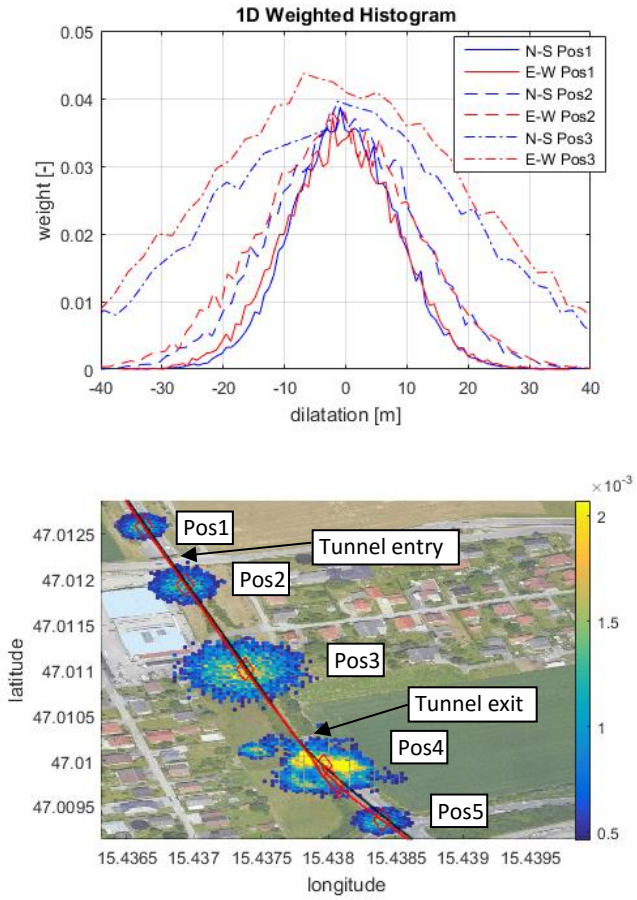

Figure 13: Particle cloud behaviour when driving through a tunnel, evaluated at five positions. Yellow dots indicate high and blue dots low correlation with the LOS GNSS signal. The reference trajectory is shown in black and BDPE in red. Map image © 2017 Google, Landsat / Copernicus

As the particle filter uses history information about its states, the near normal distribution vanishes over time, even if there is only noise in the gathered GNSS signal. Pos 4 shows the particle cloud at the tunnels exit. It can be seen from the red trajectory in the lower plot of *Figure 13* and *Figure 14*, that the true position was contained in the particle cloud, but over-

shoots slightly in the along-track direction. The behaviour at Pos 4 is shown in more detail in *Figure 14*. The lower plot shows at Pos 4, that the particle cloud degenerated through the tunnel and two dominant local particle concentrations can be observed. The upper plot in *Figure 14* clearly identifies these two concentrations, which could correspond to the old

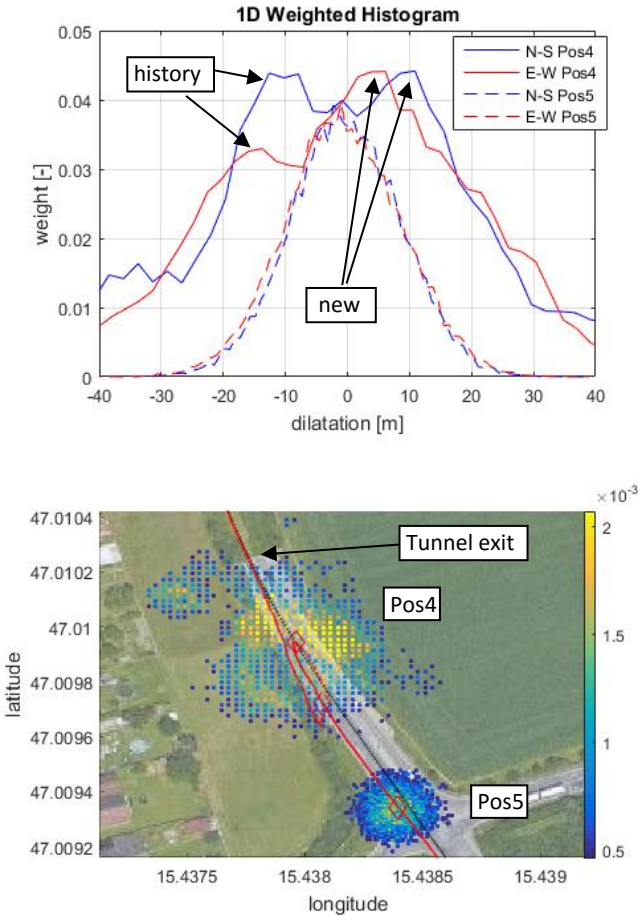

Figure 14: Particle cloud behaviour at the tunnels exit. Yellow dots indicate high and blue dots indicate low correlation with the LOS signal. The reference trajectory is shown in black and BDPE in red. Map image © 2017 Google, Landsat / Copernicus

through the tunnel propagated state, which is off by several meters after the tunnel, and the new state, a newly formed normal distribution from the strong GNSS signals after the tunnel exit with an offset to the old PVT state. From Pos 4 to Pos 5 it can be seen, that the particle filter relocks on the true PVT state very quickly. It is assumed, that BDPE accumulated signal energy through the tunnel, even if none or only a very weak LOS signal was available. The BDPE navigation solution shows especially from Pos 2 to Pos 4 a higher robustness compared to the SPP and VT solution. After exiting the tunnel the SPP and the BDPE solutions converge quickly, which indicates a proper selection of the process noise parameters of the BDPE filter to cover the true PVT state at the time of the tunnel exit. The compared vector tracking setup does not account for resetting of the tracking loops and thus does not recover in this situation.

8. CONCLUSIONS. The paper discusses the development and implementation of a Particle Filter working after the BDPE principle. First tests have been performed in simulated and real world environments with the filter embedded in a commercial software GNSS receiver (SX-3). The tests used recorded data from a test drive through the city of Graz in Austria and the results were compared to both a conventional WLSQ epoch based SPP receiver architecture as well as to an advanced Kalman filtered vector tracking solution. From theory and first results the Particle Filter shows the most benefit in areas where high sensitivity and robustness of reception is required. The examples from the test drive in Graz support this assumption although the current implementation only uses one GNSS signal (GPS L1/CA) and is therefore not yet exploiting the significant advantages of multi-frequency and/or multi-GNSS. The inclusion of input from sensors and maps to the filtering process is another promising step to be looked at in the near future.

9. ACKNOWLEDGEMENTS. The current paper bases on work also described in (Dampf et al., 2017). Acknowledgements should go to IFEN GmbH for providing the SX3 Software Receiver for the software development, to IGASPIN GmbH for their support with representative datasets and technical knowledge as well as to P. Closas for his fundamental work.

10. FINANCIAL SUPPORT. The project was funded by the Austrian Research Promotion Agency and executed by JOANNEUM RESEARCH under contract No 847994.

11. REFERENCES.

Arulampalam, S., et al. (2002), A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. IEEE Trans. on Signal Processing, 1053-587X, 174-188

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

Closas, P. (2009-1). Bayesian Signal Processing Techniques for GNSS, Thesis, Universitat Politècnica de Catalunya, Spain

Closas, P., Fernandez-Prades, C., and Fernandez-Rubio, J.A. (2009-2). Cramér Rao Bound Analysis of Positioning Approaches in GNSS Receivers. IEEE Trans. on Signal Processing 57, 3775–3786

Closas, P., Gusi-Amigó, A. (2017). Direct Position Estimation of GNSS Receivers. IEEE Signal Processing Magazine, 1053-5888, 72-84

Dampf, J., Witternigg, N., Schwinzerl, M., Lesjak, R., Schönhuber, M., Obertaxer, G. and Pany, T. (2017). Particle Filter Algorithms and Experiments for High Sensitivity GNSS Receivers. Proceedings of the 6th International Colloquium on Scientific and Fundamental Aspects of GNSS / Galileo. Valencia, Spain

Perez-Fontan, F. and Enjamio-Cabado, C. (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.

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

Won, J.-H., Eissfeller, B., Schmitz-Pfeiffer, A., Floch, J.-J. and Colzi, E. (2011). Implementation, Test and Validation of a Vector-Tracking-Loop with the ipex Software Receiver. Proc. 24th ION GNSS, Portland