Real Time Results of Vector Delay Lock Loop in a Light Urban Scenario

Katrin Dietmayer, Florian Kunzi, Dr. Fabio Garzia,
Matthias Overbeck and Dr. Wolfgang Felber
Fraunhofer Institute for Integrated Circuits IIS, Nuremberg, Germany

***Abstract*—The need of a robust navigation solution increased during the last decade. As an example, for autonomous driving it is essential to have a stable satellite signal tracking even in difficult environments such as urban canyons. The paper describes a real-time vector delay lock loop (VDLL) approach implemented directly on the GNSS receiver hardware to achieve a deep integration in the signal processing. This approach uses the code correlation values fetched directly from the receiver hardware to provide an estimated position, velocity and time (PVT) solution using an extended Kalman filter (EKF). The solution is then used to calculate new steering code values for all channels at the same time, which are sent back to the hardware to steer the code numerically controlled oscillators (NCOs). The approach is tested in a real-world light urban scenario, where different signal degradations occur including multipath, signal shadowing or reflections. The VDLL real-time implementation is analyzed and compared to a reference system. A detailed description of the scenario and the used GNSS receiver hardware is shown.**

***Index Terms*—vector tracking (VT), VDLL, tracking loops**

I. Introduction

Developments in autonomous driving have continued to advance over the last years. A significant requirement for this technology is to gain a precise, robust, and continuous information about the vehicle's current position. The use of GNSS technology could play an important role in this. But to meet the necessary requirements for the position, velocity, and time (PVT), the quality of the PVT, expressed in terms of accuracy, availability, and robustness has to increase. Urban environments in particular make reliable GNSS positioning difficult. The GNSS signal reception and thus the PVT solution in such scenarios is highly affected from signal degradations like shadowing, multipath or interference caused by high buildings, infrastructure, trees or other road participants. Standalone GNSS receivers with scalar tracking approaches are strongly affected by such disturbances and can only provide limited methods to mitigate them.

After successfully acquiring a satellite signal, a conventional GNSS receiver tracks each signal independently. Such receivers usually have two main parts, the individual signal processing unit, which performs the signal acquisition and tracking loops, and a common navigation processing unit providing a navigation solution. The tracking loops are used to synchronize a locally-generated replica with the received satellite signal. This replica has to be adjusted continuously due to the overall movement. A correlation between both provides correlation values which are used to estimate the divergence between the replica and the received signal. After filtering, new code and carrier parameters are calculated and used to steer the code and carrier numerically controlled oscillator (NCO) to obtain a stable reference.

The reception of reflected or diffracted signals in addition to the line of sight reception leads to a distortion of the ideal correlation function and thus to a deterioration of the code and carrier estimations. As a result, the pseudorange and Doppler measurements are degraded which leads to an inaccurate navigation solution. At worst, the lock of the tracking loop is lost and the satellite signal has to be reacquired. The total or partial line of sight signal blockade, in addition to non line of sight reception caused by the urban environment, bias the pseudorange measurements too. That is because of the ”extra” path the signal has traveled compared to the theoretical ”direct” path. This leads to the same effects, an inaccurate navigation solution. Shadowing effects or blocked signals can cause degraded pseudorange and Doppler measurements which again lead to an inaccurate navigation solution. A complete signal tracking loss of the blocked satellite is most likely. Without enough satellites in tracking the navigation processor could be unable to compute a navigation solution at all.

To gain a more robust navigation solution, Vector Tracking is one promising approach, see for example [2]. The main difference between a standard approach and the vector tracking is that all channels are tracked together and in relation to each other [3]. Hence, the signal processing unit isn't independent for each satellite. The navigation processing unit is replaced with an extended Kalman filter (EKF). Both units are now interconnected and not individual anymore.

This paper extends our previous research work related to the real-time implementation of the vector delay lock loop (VDLL) in simulated [4] and real-world scenarios in preferable conditions [5], where the integration of the VDLL on the GNSS hardware receiver was only investigated for tests without or only light multipath or other signal degradations. The focus was set to evaluate the performance of the real-time VDLL and the integration on the hardware.

This paper shows the result of a VDLL in light urban environments compared to the standard approach where scalar tracking loops are used. To compare the results, an additional reference system was added to the test setup. To examine how the VDLL reacts in a light urban scenario, a test path in an area with e.g. multi-story buildings and trees was chosen. This is shown and analyzed to investigate which critical points

978-1-7281-0244-3/20/$31.00 ©2020 IEEE

occurred and where the GNSS signals were subjected to degradations. The used hardware where the VDLL approach is implemented is described. To evaluate the whole test setup the GNSS signals were recorded, analyzed and replayed. This ensures the profitable use of the signals as the tests can be repeated and hence analyzed in detail with the different approaches and settings.

II. Vector delay lock loop and receiver design

A. Vector delay lock loop design

The vector tracking (VT) is widely explained in literature, e.g. [3]. As a first step, the VDLL, one variant of the VT, was chosen to be implemented and tested [4]. This approach replaces the code tracking loops of the scalar tracking approach, where each satellite is tracked independently. This means, the carrier tracking loops are still closed by scalar loop filters and the code loops are closed by calculating the code phase of the tracked satellites in relation with all other tracked satellites, [6].

Each time new correlation values are available, the code phase residuals of all channels are calculated. A normalized early-minus-late-envelope discriminator [7] is used to calculate the difference between measured and predicted code delays:

$$\frac{1}{2}\frac{\sqrt{I_E^2+Q_E^2}-\sqrt{I_L^2+Q_L^2}}{\sqrt{I_E^2+Q_E^2}+\sqrt{I_L^2+Q_L^2}}$$

where I_E represents the in-phase early, I_L the in-phase late, Q_E the quadrature early and Q_L the quadrature late correlator outputs. Besides this, the pseudorange-rate is calculated for each channel and used together with the code phase residuals. The pseudorange-rate residuals are then calculated by the EKF, using the predicted receiver and satellite velocities. Fig. 1 shows a top-level diagram of the VDLL approach.

For the VDLL implementation, an EKF is used to calculate a navigation solution at each iteration k. A detailed explanation of the filter steps can be widely found in the literature, e.g. [3] or [8]. The observed EKF state vector x_k at the iteration k includes the position $p_k := (p_x, p_y, p_z)$, velocity $\dot{p_k}$, clock bias b_k and drift d_k:

$$x_k = (p_k, \dot{p_k}, b_k, d_k)^T$$

The receiver and satellite dynamics are computed in Earth-Centered Earth-Fixed (ECEF) coordinates.

The EKF is designed with a constant velocity model. This means, the receiver velocity is considered constant in each axis through each update step. To cope with velocity changes caused by un-tracked acceleration, a noise model is used at each iteration. Additionally, the receiver clock frequency drift f causes a clock drift error and the receiver's clock phase drift ϕ an error on the clock offset. These are assumed to be white Gaussian noise. The system noise covariance matrix Q of the used EKF shows the noise changes in the system with respect to time [9], [4]. It is crucial to have an accurate tuning so that the EKF states convergence towards the true state vector. Together with the single sided power spectral density (PSD) matrix S_a, which includes the velocity error variance terms along the ECEF frame axes, the oscillator's phase noise PSD S_ϕ and frequency noise PSD S_f, Q is computed as a diagonal matrix with δt as the updating interval:

$$Q = \begin{pmatrix} \frac{1}{3}S_a\delta t^3 & \frac{1}{2}S_a\delta t^2 & 0_{3\times1} & 0_{3\times1} \\ \frac{1}{2}S_a\delta t^2 & S_a\delta t & 0_{3\times1} & 0_{3\times1} \\ 0_{1\times3} & 0_{1\times3} & S_\phi t + \frac{1}{3}S_f\delta t^3 & \frac{1}{2}S_f\delta t^2 \\ 0_{1\times3} & 0_{1\times3} & \frac{1}{2}S_f\delta t^2 & S_f\delta t \end{pmatrix}$$

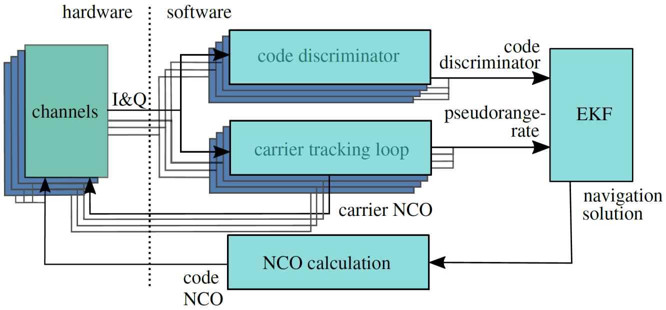

Fig. 1. Vector delay lock loop top level diagram.

Besides the system noise, the EKF also defines a measurement noise matrix R. This reflects the noise of the measurements, for the VDLL implementation the code phase residual noise and the pseudorange-rate noise. There are different ways to model this, e.g. as a constant diagonal matrix. As the VDLL is directly implemented on the receiver hardware, a model based on the known dynamics is chosen. Taking the carrier-to-noise density ratio (C/N_0) for each satellite, the EKF is fine-tuned [3], this is also possible with only a coarse estimation [8], based on the signal's prompt correlation values from the hardware. Even in case of a weak signal or a complete signal blockade for a short time the tracking will continue, hence a power estimation is available and used to tune the EKF measurement noise matrix. If multipath or other interference occurs, this will also contribute to the C/N_0 of the satellite signal line of sight [10]. Satellites with a low C/N_0 do not significantly affect the navigation solution estimation. Only signals with a high C/N_0 have an influence on the system estimation. Additionally, a weighting based on the elevation of each satellite is used [11]. Satellites with a low elevation are often affected by signal degradations and consequently have a low influence on the overall navigation solution. Using the receiver in real applications e.g. multipath, ionospheric or tropospheric disturbances affect the received signals. These effects could also be taken into account when tuning the measurement noise matrix to compensate such signal degradations. The matrix entries of R for each satellite $j \in 1\ldots\mathrm{N}$ are denoted with σ_ρ^j and $\sigma_{\dot{\rho}}^j$ for the covariances of pseudorange ρ and pseudorange-rate $\dot{\rho}$ [3]. Including the elevation θ^j of each satellite j, the weighted entries of R are:

$$\sigma^j_{\theta,\rho} = \frac{1}{\sin^2(1.125 \cdot (\theta^j - (10 \cdot \pi/180)))} \cdot \sigma^j_{\rho}$$

$$\sigma^j_{\theta,\dot{\rho}} = \frac{1}{\sin^2(1.125 \cdot (\theta^j - (10 \cdot \pi/180)))} \cdot \sigma^j_{\dot{\rho}}$$

where a 10° shift is used as all satellite below 10° are automatically excluded from the VDLL approach. Hence the measurement noise covariance matrix is:

$$R = \begin{pmatrix} \sigma^1_{\theta,\rho} & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \sigma^2_{\theta,\rho} & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \sigma^{\mathrm{N}}_{\theta,\rho} & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \sigma^1_{\theta,\dot{\rho}} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \sigma^2_{\theta,\dot{\rho}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & \sigma^{\mathrm{N}}_{\theta,\dot{\rho}} \end{pmatrix}$$

In each iteration, the code loops are then closed through the navigation solution estimated by the EKF [2]. Therefore, the pseudorange of each satellite is calculated using the estimated receiver position and the corresponding satellite position. The new code NCO values are then sent back as a vector to the receiver hardware to steer the hardware NCOs. As already mentioned, the carrier loops are already closed individually for each satellite through a standard scalar tracking loop.

B. Receiver design

The whole VDLL application is implemented on the so called GOOSE© receiver, Fig. 2. This receiver is based on a Xilinx Kintex 7 field-programmable gate array (FPGA) which is connected to an external processor using a PCIe interface [12]. The FPGA receives digital samples from a customized tri-band radio frequency frontend. The acquisition module with 60 channels is implemented on a re-configurable device, which can also be controlled from the external processor. From the GOOSE© receiver hardware the complex correlation values are fetched and the previously implemented scalar code tracking loops are replaced by the interlaced VDLL algorithm, while the carrier loops are used as before [13]. GOOSE© provides e.g. the GPS L1 C/A signal correlation values with a coherent 20ms integration time. This means each EKF iteration step needs to be finished within this period in order to close the loops in time.Otherwise, the system could include an unnecessary tracking error or get unstable and affect the whole tracking performance. Over the same PCIe interface, the new carrier NCO values are sent back directly after they have been provided and calculated through the scalar loop. The code NCO are sent back after a successful EKF step through the same interface as it provides the possibility to steer all hardware channels at once [14]. Additionally, the decoded or loaded ephemeris information is available to calculate the satellite position. As the EKF has to be initialized with a valid PVT the GOOSE© receiver is designed to first use the scalar tracking loops after a successful satellite acquisition. If at least four satellites are tracked, a first PVT solution is calculated and used for the EKF initialization. It should be mentioned that the GOOSE© receiver supports a clock steering option, but as this is not considered at the EKF design, this feature is disabled for the VDLL.

Fig. 2. Photo of GOOSE© receiver used for record, playback and VDLL.

III. Test Setup

To test the performance and the limits of the VDLL in situations with different signal degradations a light urban scenario test was done. Therefore, a track near the department of Fraunhofer IIS in Nuremberg, Germany, was chosen. For this test, the GPS L1 C/A signals were used and recorded with the GOOSE© receiver during the test drive. The environment includes e.g. medium high buildings, road signs, trees and other road users. The green line in Fig. 3 shows the ideal trajectory of the recorded test drive taken from the used reference system. Although having a varying velocity, the constant velocity EKF design fits for this scenario, as the chosen update rate of 50Hz is able to handle the un-tracked acceleration through the covariance matrices well enough. A 3G+C OEM mobile antenna from navXperience was mounted

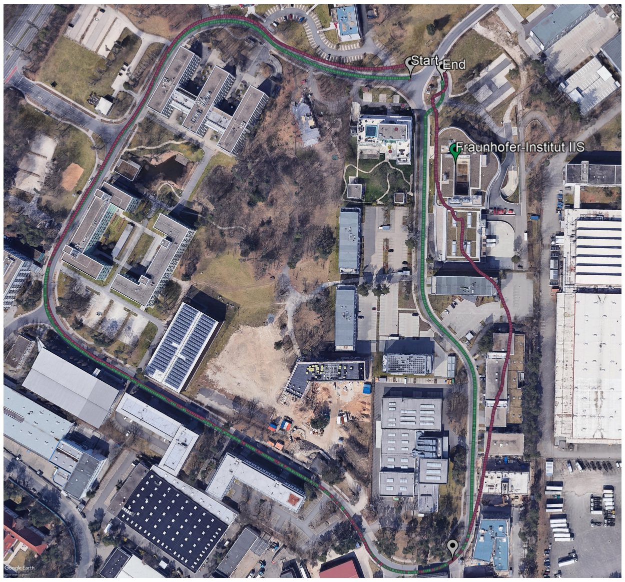

Fig. 3. The ideal trajectory from a reference system in green and the VDLL trajectory in orange of the test drive near the Fraunhofer IIS department, picture from Google Earth Pro.

on top of the vehicle and connected to the GOOSE© receiver. Besides the antenna a fish eye camera was mounted to be able to visually identify possible obstacles after the test. The recorded scenario was replayed using a Multi-GNSS Simulation & Test Environment (MGSE©) signal replay unit, which was connected to the same GOOSE© receiver. This means, the scenario is repeatable with the same conditions and can be tested with the VDLL approach and standard scalar tracking loops.

A. Automotive Dynamic Motion Analyzer

Since the used real-world scenario has some challenging parts, a precise system for comparison of the navigation solution had to be selected, which is why the Automotive Dynamic Motion Analyzer (ADMA) was used as a reference system [15]. Two different outputs are provided: A loosely coupled GPS/ INS solution from an integrated navigation system (INS) and a standalone GPS RTK navigation solution. The ADMA was mounted inside the test vehicle and connected to the same antenna as the GOOSE© receiver. The VDLL approach was compared with the ADMA solution.

B. Test track and conditions

Along the test track several critical points were passed, which impact not only the visibility of the satellites. During some parts, the buildings were near the road and hence blocked the visibility from one side, or were like an urban canyon. Several multipath events or diffractions occur besides short time signal blockage. Figure 4 shows the four GPS L1 C/A signals which were tracked with the VDLL.

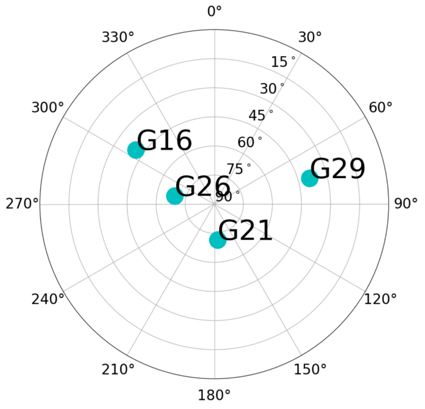

Fig. 4. Skyplot of the recorded and tracked GPS L1 C/A signals.

IV. Test Results

Fig. 5 shows the position solutions of the VDLL of the GOOSE© receiver, the GPS/ INS and standalone GPS solution from the ADMA and a scalar tracking solution from the same GOOSE© receiver, with the same carrier loop settings as for the VDLL implementation. All position solutions are aligned through the timestamp. The GPS/ INS solution is used as the reference position and velocity. The VDLL implementation requires a first PVT solution to initialize the EKF. Therefore, the GOOSE© receiver starts with a scalar tracking (ST) approach and hands over to the VDLL after it provides a first navigation solution. The VDLL starts with a higher overall uncertainty in the EKF and needs a short duration to reach a stable state [11]. During these few seconds the PVT solution shows a high offset compared to the reference, which can be seen in the following figures at the beginning of the test drive. As the scalar tracking of the GOOSE© has also only four satellites, the high offsets and long periods without a PVT solution are caused by the harsh environment. Each time a satellite lock is lost, the signal has to be reacquired, but with less then four satellites there is no possibility to provide a continuous PVT solution. In Fig. 3 and Fig. 5 a large deviation of the position solution of the VDLL from the ADMA reference after two thirds of the test can be seen. As a consequence, the detailed test track evaluation is divided into two main parts. During part 'A', from start to the highlighted point in Fig. 3 or Fig. 5, two critical parts are examined in detail, see Section IV-A. Part 'B', from the end of part 'A' to the end of the test, the reason for the position deviation is examined, see Section IV-B.

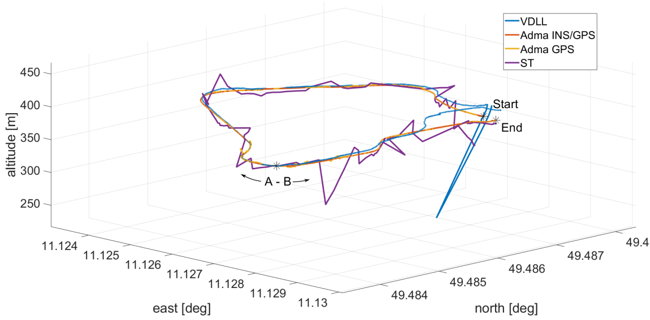

Fig. 5. 3D position solution comparison between the VDLL, the GPS/ INS and GPS standalone ADMA reference.

Fig. 6 shows the estimated C/N_0 from VDLL over time into run of the complete test drive. The variations are caused by different effects, like shadowing, multipath or else from e.g. nearby buildings and trees. The signals from the satellites G16 and G29 are more frequently influenced by the environment then G21 and G26 due to their lower elevation, see Fig. 4.

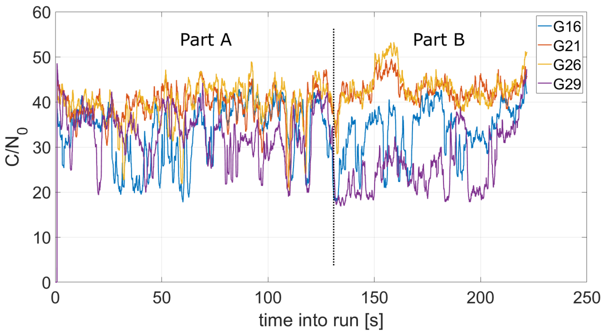

Fig. 6. Estimated C/N_0 measurements calculated by the VDLL.

Fig. 7 shows the calculated absolute velocity of the VDLL compared to the GPS/ INS ADMA. With the constant velocity model of the EKF every acceleration is modeled as noise. Nevertheless, this is estimated well enough during part ’A’. The standard deviation of 14.32 m/s between the absolute velocity of the VDLL compared to the reference GPS/ INS velocity is high due to part ’B’ and the overall noise due to the signal degradation.

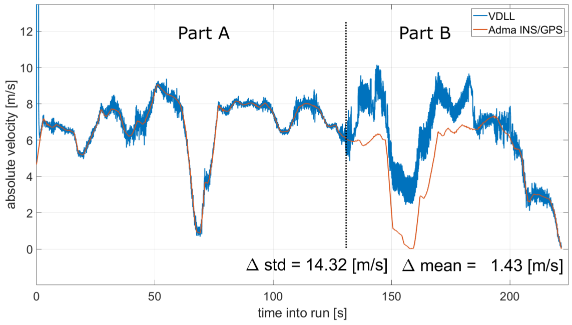

Fig. 7. Velocity calculated by the VDLL during the test drive.

A. Part A

During part ’A’ of the test drive the EKF provides an adequate PVT solution. All four satellites are in a stable vector code and scalar carrier tracking. During this part the standard deviation in up is 7.30 m, north 3.67 m and east 0.99 m, see Fig. 8. The position solution converges during the test duration, but needs longer due to the signal degradation from the environment. Using only four GPS L1 C/A signals in the VDLL, the results are good and a PVT solution is provided throughout the complete test time.

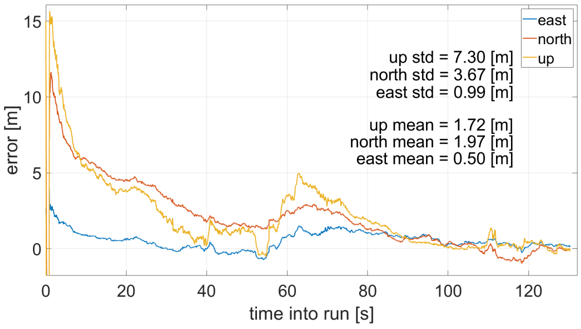

Fig. 8. Position Error of part A, VDLL compared to the GPS/ INS ADMA reference.

For a better understanding of the processes, the part after the first curve is examined in more detail. Fig. 9 shows the position solution of the used systems and the environment at this point. A multi-story building on the left side of the road and high trees on the right. The trees disturbed the direct line of sight reception of satellite G16. There is also the possibility that non line of sight signals are received from the building on the opposite. The C/N_0, calculated through the VDLL, drops significantly, see Fig. 10. Even the GPS reference solution of the ADMA shows deviations in the position during this part, see yellow line in Fig. 9. For short incidents, the signal reception of G16 is too poor to grant a good scalar carrier tracking. Therefore, a fixed threshold of the signal strength causes that the carrier tracking loop is bridged by avoiding to steer the hardware with new carrier Doppler values. Otherwise it could happen, that the scalar tracking loop is lost, and with less than four satellites in track no PVT is provided. This effect is shown in Fig. 11, where the carrier Doppler is marked in some examples. On the other hand, the VDLL is still able to provide adequate code Doppler values during this part. At the end of this short section, after 42 s into run, the C/N_0 of satellite G29 also drops, where G16 rises again. Here the building on the left side of the street ends and there are no more trees on the right side. Hence, the line of sight reception of G29 is poor as the satellite has low elevation and is probably hidden by the building corner.

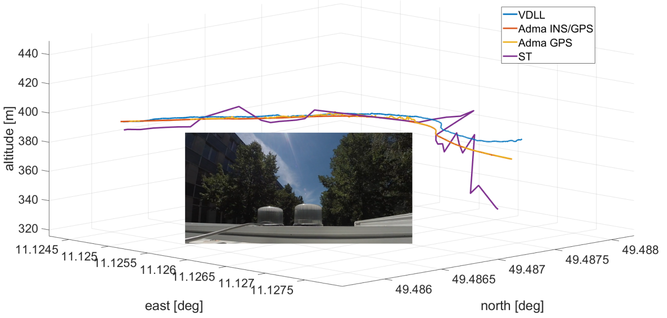

Fig. 9. 3D position comparison during the section after the first bend.

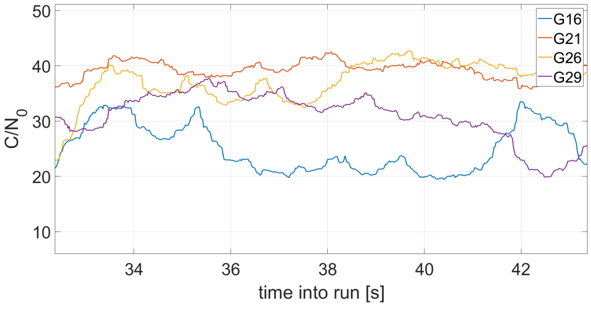

Fig. 10. The C/N_0 of satellite G16 drops significantly passing by high trees.

B. Part B

The second part of the test drive shows a large PVT offset. Fig. 12 shows the environment at the start of part ’B’. On the left side high trees are located and on the right side a multi-story building very close to the road. At the start of part ’B’ all satellites show more noise, see Fig. 13. Satellites G16 and G29 have a constantly low C/N_0 and the signal strength of satellite G29 is below the signal strength threshold. Hence, the carrier tracking loop is not used to calculate new carrier Doppler values. Past the building, the threshold of the signal strength

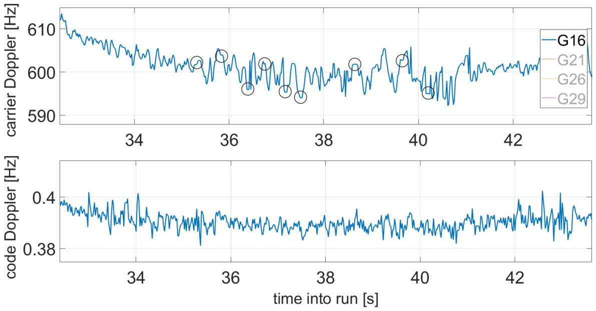

Fig. 11. The code Doppler calculated from VDLL and the carrier Doppler from scalar tracking loop for satellite G16. The carrier loop incidents are highlighted with black circles.

is passed and the scalar carrier tracking loop should take over again. But there is still a challenging environment and due to different signal degradations the scalar carrier tracking loop is broken, see the rectangle in Fig. 14. This causes a lot of noise in the EKF as the pseudorange-rate measurements from the scalar carrier tracking loop is used as an input. Since the PVT estimation is based on all satellite measurements and due to the bad prerequisite of only tracking four satellites, the position drifts away. Normally, a carrier tracking loop indicator is used to determine if the lock is lost. Due to the harsh conditions and the bridging of the scalar tracking loop, this indicator would often signalize a loss of lock and stop the corresponding channel. To get a better understanding of the VDLL behavior and to avoid stopping channels during short signal blockage, this loop indicator is omitted in the tests and no channels are stopped. This is only an option for research. Future work includes further testing and research to handle and mitigate such incidents.

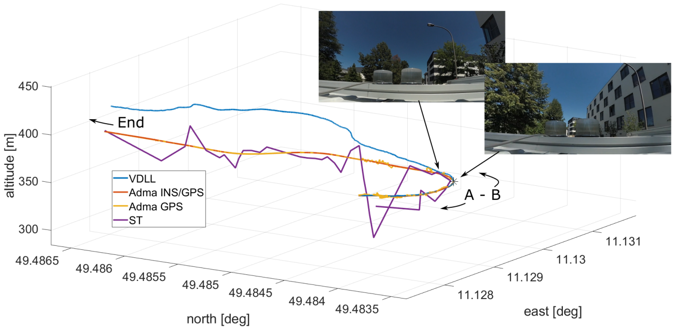

Fig. 12. 3D position comparison at the beginning of part ’B’.

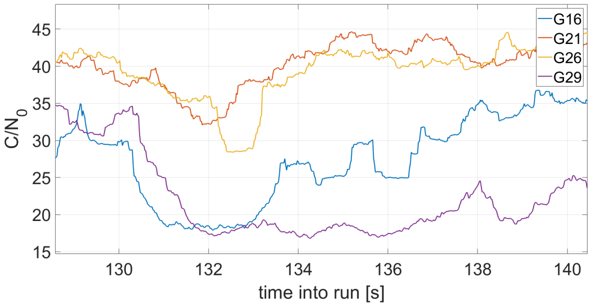

Fig. 13. The C/N_0 of satellite G16 drops significantly passing by high trees.

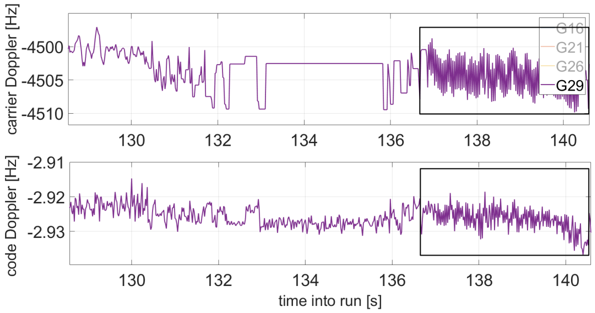

Fig. 14. The code Doppler calculated from VDLL and the carrier Doppler from scalar tracking loop for satellite G29. The carrier loop is disturbed.

V. Conclusion and Future Work

The paper presents the results of a real-time implementation of a VDLL in a light urban scenario. The tests have shown good results with the recorded and replayed real-world test scenario. Short signal disturbances due to high trees are handled well and for all signals a stable tracking was reached. Using a simple EKF constant velocity model with a proper noise estimation leads to a stable code tracking. The probability of multipath reception by passing by multi-story buildings is very high, but the VDLL kept a stable tracking. The scalar carrier tracking loop seems to be a bottle neck in harsh conditions. When the scalar carrier tracking loop is not stopped but needs to, the EKF gets unstable and affects the complete system. Further research and tests have to be done to improve the scalar carrier tracking loop performance and cooperation with the VDLL. Also, an adequate indicator for the scalar and vector tracking loops to stop channels if necessary has to be implemented. But this indicator has to be set not too sensitive, a good balance has to be found. Otherwise, channels which are only blocked for a short time will also be stopped. In this test, stopping the channel from satellite G29 would have required to stop the VDLL as with three satellites the PVT solution would be even worse. The algorithm would need to hand over to a standard scalar code tracking loop and wait until four or more satellites are in tracking again.

Future work includes not only a method to indicate the loss of tracking lock and stop the corresponding channels but also to omit the scalar tracking loop by including them to the VT.

A vector delay and frequency lock loop is a promising solution to gain an even more robust tracking. However, including more GNSS systems, like the European Galileo is another task. This would provide better signal coverage and improve the overall PVT robustness and accuracy.

References

[1] E. D. Kaplan and C. J. Hegarty, “Understanding GPS: Princiles and Applications, 2nd Ed.,” Artech House, 2006.

[2] E. Shytermeja, A. Garcia-Pena and O. Julien, “Performance Comparison of a proposed Vector Tracking architecture versus the Scalar configuration for a L1/E1 GPS/Galileo receiver,“ ENC2016, European Navigation Conference, May 2016, Helsinki, Finland. pp.ISBN: 9781479989164.

[3] P. D. Groves, “Principles of GNSS Inertial and Multisensor Integation navigation systems, 2nd Ed.” Artech House, 2013.

[4] K. Dietmayer, M. Saad, C. Strobel, F. Garzia, M. Overbeck and W. Felber, “Real time implementation of Vector Delay Lock Loop on a GNSS receiver hardware with an open software interface,“ ENC2019, European Navigation Conference, April 2019, Warsaw, Poland.

[5] K. Dietmayer, F. Kunzi, C. Strobel, M. Saad, F. Garzia, M. Overbeck, W. Felber, “Results of a real time implementation of vector delay lock loop in a dynamic field test,“ Proceedings of the 32nd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2019), Miami, Florida, September 16-20, 2019, pp. 3759 - 3767.

[6] M. Lashley, D. Bevly and M. Perovello, “Vector Delay Lock Loops,” in Inside GNSS, September 2012.

[7] Y. Yang, J. Zhou and O. Loffeld, “GPS Receiver Tracking Loop Design based on a Kalman Filtering Approach,” Proceedings ELMAR-2012, pp. 121-124, 2012.

[8] F. M. G. Sousa and F. D. Nunes, “Performance Analysis of a VDFLL GNSS Receiver Architecture under Ionospheric Scintillation and Multipath Conditions,“ Proceedings of the 2014 IEEE/ION Position, Location and Navigation Symposium PLANS, Monterey, CA, May 2014, pp. 602-611.

[9] S. Zhao and D. Akos, “An Open Source GPS/GNSS Vector Tracking Loop - Implementation, Filter tuning and Results,” Proceedings of the 2011 International Technical Meeting of the Institute of Navigation, San Diego, CA, January 2011, pp. 1293-1305.

[10] T. Pany and B. Eissfeller, “Use of a Vector Delay Lock Loop Receiver for GNSS Signal Power Analysis in Bad Signal Conditions,“ Proceedings of the IEEE/ION Position Location and Navigation Symposium PLANS 2006, San Diego, CA, April 2006, pp. 893-903.

[11] K. Dietmayer, F. Kunzi, C. Strobel, M. Saad, F. Garzia, M. Overbeck and W. Felber, “Solution for satellite rise and set in a vector delay lock loop on GNSS receiver hardware,” Proceedings of International Conference on Localization and GNSS, June 2019.

[12] M. Overbeck, F. Garzia, A. Popugaev, O. Kurz, F. Förster, W. Felber, A. Ayaz, S. Ko, and B. Eissfeller, “GOOSE – GNSS Receiver with an Open Software Interface,” Proceedings of the 28th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2015), Tampa, Florida, September 2015, pp. 3662-3670.

[13] F. Garzia, C. Strobel, M. Overbeck, N. Kumari, S. Joshi, F. Förster and W. Felber, “A Multi-Frequency Multi-Constellation GNSS Development Platform with an Open Interface,” in: IEEE Xplore 2016, DOI 978-1-4799-8915-7.

[14] M. Overbeck, F. Garzia, C. Strobel, C. Nickel, M. Saad, D. Meister and W. Felber, “GNSS-Receiver with Open Interface for Deeply Coupling and Vector Tracking,” Proceedings of the 29th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2016), Portland, Oregon, September 2016, pp. 1222-1229.

[15] Automotive Dynamic Motion Analyzer. Available online: https://www.genesys-offenburg.de/en/home/ (accessed on 03 February 2020).