GPS Solutions (2022) 26:125
https://doi.org/10.1007/s10291-022-01305-4

ORIGINAL ARTICLE

Improving GNSS baseband using an RTK-position-aided code tracking algorithm

Yiran Luo[1] · Li-Ta Hsu[2] · Zhetao Zhang[3] · Naser El-Sheimy[1]

Received: 16 December 2021 / Accepted: 19 July 2022 / Published online: 13 August 2022

Abstract

In an urbanized environment, a superior baseband algorithm fundamentally enhances global navigation satellite system (GNSS) receivers by resisting interferences such as reflection leading to multipath signals and abnormal dynamics caused by user's motion or non-line-of-sight (NLOS) signals. A carrier aiding algorithm is frequently adopted to remove most dynamic errors on local code signals. Depending on this idea, all-channel spatial information based on the user's velocity is also exploited by existing vector tracking techniques to aid code tracking. As known, a carrier phase is one order of magnitude less noisy and more capable of mitigating multipath than a code phase. The traditional carrier frequency aiding improves the precision of the code phase estimation by suppressing random noise, but it is powerless in eliminating the biased error. To solve this issue, a real-time kinematic (RTK) solution, which is the product of the carrier phase, is proposed to remove the absolute biased error in the vector delay/frequency lock loop (VDFLL). Thus, the GNSS baseband can be optimized with both the carrier frequency and the absolute carrier phase aiding. Two stationary real-world experiments are conducted in open-sky and light-urban situations to verify the proposed GNSS software-defined radio (SDR) using collected GPS L1 C/A intermediate frequency (IF) data. The testing results demonstrate that the proposed SDR improves the pseudorange quality and positioning accuracy by a maximum of 48.5% and 42.5%, respectively.

Keywords GNSS baseband · Navigation · Float RTK positioning · Vector delay/frequency/phase lock loop · Multipath mitigation

Introduction

Accurate navigation in urbanized environments is highly dependent on global navigation satellite systems (GNSSs). The initial guess from GNSS devices largely affects the positioning performance for the multi-sensor-based navigation system, such as smartphones or wearable sensors (Lachapelle et al. 2018). Carrier-based measurements outperform code-based measurements in terms of both Doppler frequency and phase resolution due to the property of GNSS signals. By exploiting this characteristic, the carrier aiding algorithms are commonly used in current GNSS receiver designs (Kaplan and Hegarty 2017; Yang et al. 2017). So, most of the dynamic uncertainties caused by the Doppler effect are removed from the pseudorange measurements.

To optimize the code tracking further, designers have enhanced the GNSS receivers with vector tracking techniques (Lashley et al. 2009; Lashley and Bevly 2013; Xu et al. 2020). On the one hand, the vector delay lock loop (VDLL) takes advantage of all-channel pseudorange

✉ Yiran Luo
yiran.luo@ucalgary.ca

✉ Li-Ta Hsu
lt.hsu@polyu.edu.hk

Zhetao Zhang
ztzhang@hhu.edu.cn

Naser El-Sheimy
elsheimy@ucalgary.ca

[1] Department of Geomatics Engineering, University of Calgary, Calgary T2N1N4, Canada

[2] Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong SAR, China

[3] School of Earth Sciences and Engineering, Hohai University, Nanjing 211100, China

rates to compensate for the dynamic stress of code errors (Spilker Jr 1996). It has had an accelerated navigation performance in high-dynamic and weak situations, respectively (Ren and Petovello 2017; Chen and Gao 2019). On the other hand, code tracking in a vector delay/frequency lock loop (VDFLL) can be assisted by the user's velocity (Lashley et al. 2009). For example, the velocity compensates for the carrier Doppler, which aids the code estimation to reduce the tracking error. Also, it has been verified to be efficient in improving the high-precision positioning performance in a harsh environment (Luo et al. 2019a, 2019b, 2019c). These days, this technique is still under hot debate. For example, it has been upgraded to a collaborative-navigation version (Watts and Martin 2020) and has been enhanced to perform better in multipath mitigation (Hsu et al. 2015).

The pseudorange measurements with low accuracy break down the reliability of an RTK GNSS receiver. To solve this issue, in this research, we focus on reducing the bias of the baseband code error using the predicted pseudorange from the carrier-phase-based RTK position. Some research has solved this problem with the phasor based on a one-second super-correlation approach (Faragher et al. 2018, 2019; Groves et al. 2020), but it requires a long integration. As discussed earlier, the carrier aiding in the baseband reduces the code error in harsh situations. However, these carrier frequency aiding algorithms primarily process the random code error other than the code bias. Thus, we aim to eliminate the code bias error with an absolute carrier phase aiding.

The traditional carrier aiding algorithms are illustrated in Fig. 1 (top and middle). For the conventional tracking (CT) receiver, the carrier Doppler directly aids the code tracking; for the traditional VDFLL receiver, the carrier Doppler is first compensated by the predicted user velocity; after that, the code Doppler is aided by the compensated carrier Doppler (Lashley et al. 2009; Lashley 2009).

The architecture of the proposed algorithm is summarized in Fig. 1 (bottom), where the ordinary single point positioning (SPP) estimator is replaced by the RTK engine, which is used to predict a more accurate absolute code phase to remove the bias in the code tracking error. The RTK engines leverage the measurements from both the rover/user and base/reference stations. In contrast to the carrier frequency aiding, this architecture is based on a carrier phase aiding for code tracking. More details will be discussed in the subsequent sections.

In summary, in Fig. 1, the traditional architectures in the top and middle panels only explore the strength of a carrier Doppler to optimize the code frequency. In contrast, the absolute carrier phase is not exploited. Therefore, this research solves this problem by deeply integrating the RTK positioning results into the GNSS baseband, i.e., the code tracking loop or the delay lock loop (DLL).

As known, the carrier phase measurement is naturally superior to the pseudorange measurement in mitigating multipath interference, suppressing random noise, and computing a high-accuracy position. Thus, this work aims to improve the quality of pseudorange measurements in the GNSS baseband in an urban area with the aid of carrier phase measurements predicted from the float RTK position. In this research, the carrier aiding for code signal processing will be extended by utilizing the strength of both carrier Doppler frequency and carrier phase in a vector GNSS receiver. The main contributions are stated:

1. To develop a VDFLL GNSS software-defined radio (SDR) receiver where the RTK algorithms are deeply integrated into the GNSS baseband, we build an SDR platform to compute RTK positions starting from the intermediate frequency (IF) data;
2. To remove the bias in the code tracking error with the platform built in Contribution 1, we improve the GNSS baseband using the float RTK position to predict a submeter- or decimeter-level geometry distance accounting for an accurate code bias estimation in real-time processing;
3. To form an adequately accurate pseudorange prediction for Contribution 2, we use the information in the base station to estimate the user clock bias error and compute the systematic error with the between-receiver single-difference algorithm;
4. In order to verify the proposed algorithms in Contributions 2 and 3, we conduct two stationary real-world experiments in the open-sky and light-urban areas based on the SDR platform developed in Contribution 1.

We first investigate the proposed DLL aided by the float RTK position where the code loop architecture, computations of GNSS measurements, float RTK algorithm, and work process of the proposed RTK-position-aided vector tracking are introduced; the next section introduces a hardware-in-the-loop (HIL) simulation to generate a reference of the absolute code phase to be compared with the results estimated from the proposed algorithm; the subsequent section demonstrates the measurement and positioning qualities of the proposed algorithm by conducting real-world experiments; the last section concludes this work.

Design of the RTK-position-aided VDFPLL baseband

The GNSS baseband processor deals with the incoming IF signal and outputs GNSS measurements, i.e., pseudoranges, carrier phases, and Doppler frequencies. Thus, the baseband algorithms determine the quality of the measurements,

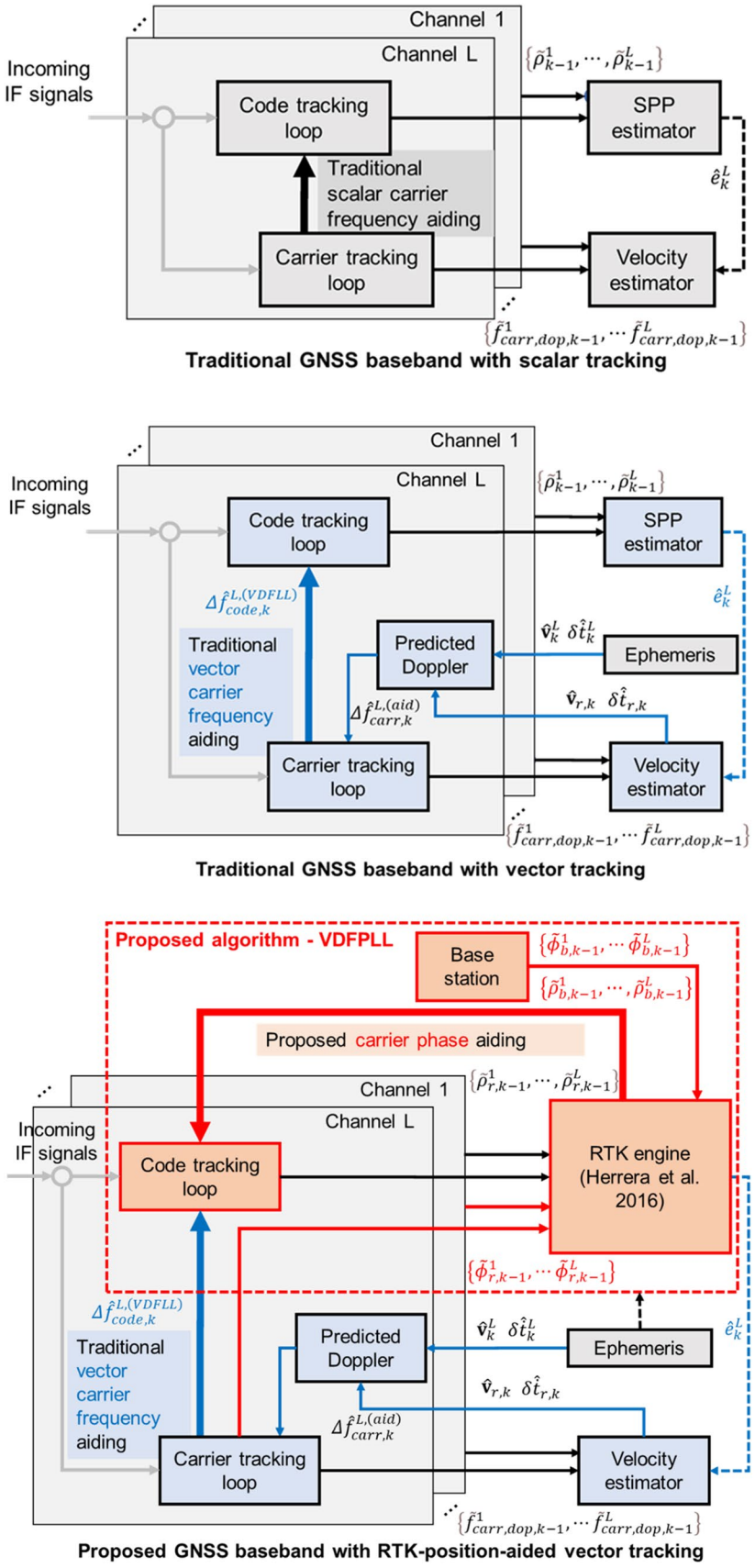


Fig. 1 Architectures of the traditional scalar-tracking (top), the traditional vector-tracking (middle), and the proposed GNSS receivers (bottom) using the carrier aiding. The blue lines correspond to the carrier frequency aiding from the VDFLL algorithm; the red lines correspond to the proposed absolute carrier phase aiding. The two traditional architectures only take advantage of carrier frequency but not the absolute carrier phase, while the proposed architecture exploits both

which, again, is one of the core indicators of the navigating performance toward a GNSS receiver design. This section introduces the proposed RTK-position-aided baseband DLL architecture and the related algorithms.

Architecture of the proposed baseband code tracking

The general architecture of the proposed vector GNSS baseband aided by the float RTK solutions is given in Fig. 1 (bottom), while Fig. 2 further illustrates the details of the proposed code tracking process:

At first, the GNSS baseband algorithm is initialized by the traditional scalar tracking algorithms and outputs initial pseudoranges, carrier phases, and Doppler frequencies. Then, the initial Doppler frequencies are input to the velocity estimator, and the initial pseudoranges and carrier phases are input to the RTK engine, as shown in Fig. 1 (bottom).

Next, the incoming IF signal is simultaneously processed by the carrier and code tracking loop at the current epoch with the proposed methods after the initialization is completed at the previous epoch. The work process and the related algorithms will be elaborated subsequently.

Computation of Doppler frequency and carrier phase measurements in the carrier tracking loop

As discussed earlier, in Fig. 2, after the initialization with the scalar tracking loop (STL), integrating and dumpling (I&D) the incoming IF signal gives the in-phase (I) and quadrature (Q) baseband signals, discriminating the prompt I/Q components with a Costas loop discriminator produces the carrier phase error, and, then, a loop filter smooths the discriminated carrier error.

Next, using the filtered carrier phase error and with the aiding of the velocity, the Doppler measurement is computed as

$$\tilde{f}^{i}_{\mathrm{carr,dop},k} = \Delta\hat{f}^{i,(\mathrm{aid})}_{\mathrm{carr},k} + \frac{1}{T_{\mathrm{coh}}}\Delta\hat{\varphi}^{i+}_{r,\mathrm{carr},k} \tag{1}$$

with

$$\Delta\hat{f}^{i,(\mathrm{aid})}_{\mathrm{carr},k} = \frac{f_r}{c}\begin{bmatrix} \hat{\mathbf{v}}_{r,k} - \hat{\mathbf{v}}^{i}_{k} \\ \left(\delta\hat{t}_{r,k} - \delta\hat{t}^{i}_{k}\right)c \end{bmatrix}^{T} \cdot \left[\hat{\mathbf{e}}^{i}_{k}, 1\right] \tag{2}$$

$$\text{Projection of LOS geometry} \triangleq \left[\hat{\mathbf{e}}^{i}_{k}, 1\right] \tag{3}$$

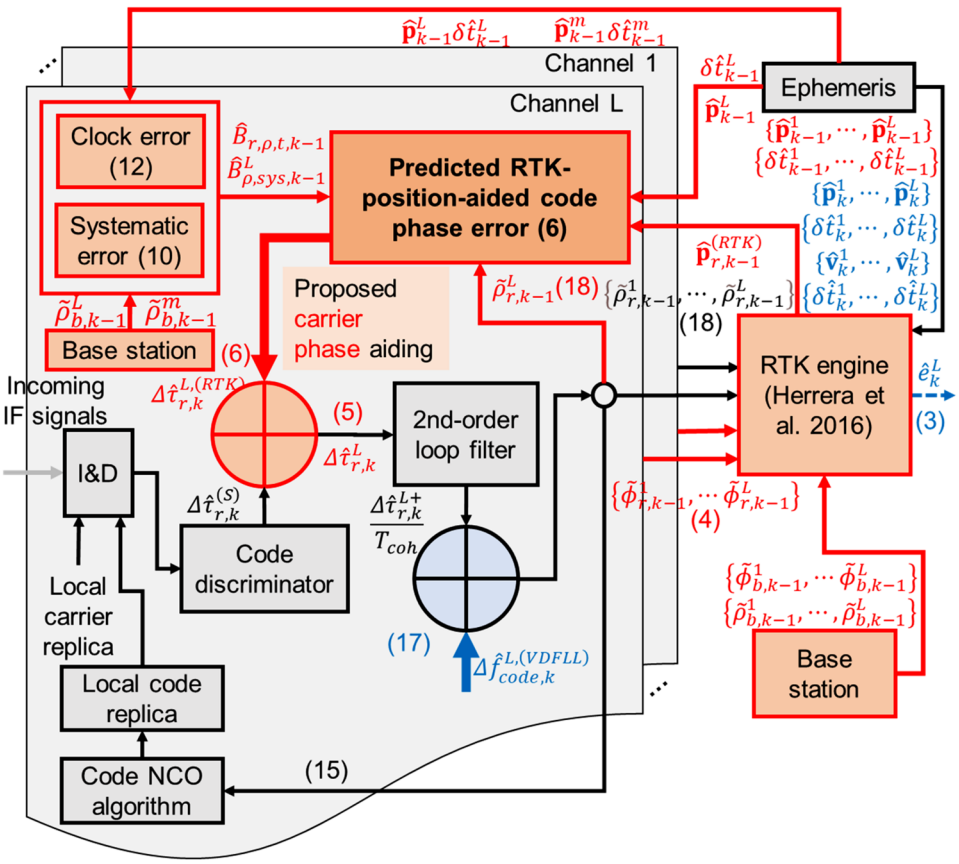


Fig. 2 Architecture of the proposed code tracking loop. Red colors correspond to the main contribution of the absolute carrier phase aiding in this architecture

where the subscript k is the epoch index; $\Delta\hat{\varphi}^{i+}_{r,\mathrm{carr},k}$ is the estimated carrier phase error passing through a first-order loop filter of which its input is the output of a Costas loop discriminator for the prompt I/Q correlations (Kaplan and Hegarty 2017); the subscript r corresponds to the rover station and i denotes the satellite pseudo-random noise (PRN) number; $(\cdot)$ denotes the inner product operator; $\hat{\mathbf{v}}_{r,k}$ is the predicted user's velocity vector using the subset of the instantaneous measured carrier Doppler frequencies, i.e., $\left\{\tilde{f}^{1}_{\mathrm{carr,dop},k-1},\cdots,\tilde{f}^{L}_{\mathrm{carr,dop},k-1}\right\}$ where the superscript L denotes the index number of the available channels; $\hat{\mathbf{v}}^{i}_{k}$ is the satellite velocity vector predicted from the broadcast ephemeris; $\delta\hat{\dot{t}}_{r,k}$ is the predicted local clock drift error; $\delta\hat{\dot{t}}^{i}_{k}$ is the predicted satellite clock drift error from the broadcast ephemeris; f_r is the radio frequency; c is the speed of light; $\hat{\mathbf{e}}^{i}_{k}$ is the unit cosine vector.

When the Doppler measurement is estimated, the carrier phase measurement is also obtained by integrating the Doppler and accumulating the Doppler integral as

$$\tilde{\phi}_k = \tilde{\phi}_{k-1} + \hat{f}_{\mathrm{carr,dop},k} T_{\mathrm{coh}} \tag{4}$$

where T_{coh} is the coherent integration time; $\hat{f}_{\mathrm{carr,dop},k}$ is the predicted carrier Doppler frequency at the k-th epoch from the measured carrier Doppler frequency at the $k-1$ th epoch, i.e., $\hat{f}_{\mathrm{carr,dop},k} = \tilde{f}_{\mathrm{carr,dop},k-1}$; it is worth explaining that the instantaneous Doppler frequency used for the numerically controlled oscillator (NCO) algorithm is directly the Doppler measurement without averaging and smoothing process in the proposed SDR, as it responds to the prompt user's velocity prediction in the VDFLL quickly.

Computation of pseudorange measurements in the code tracking loop

The Doppler and carrier phase measurements are the productions of the carrier tracking loop; correspondingly, the pseudorange is measured by the code tracking loop that is simultaneously working with the carrier tracking.

As shown in Fig. 2, after the I&D, a code phase error $\Delta\hat{\tau}^{i,(S)}_{r,k}$ is estimated by an early-minus-late-envelope code discriminator (Kaplan and Hegarty 2017). Then, the discriminated code error is compensated as follows:

$$\Delta\hat{\tau}^{i}_{r,k} = \Delta\hat{\tau}^{i,(S)}_{r,k} + \Delta\hat{\tau}^{i,(RTK)}_{r,k} \tag{5}$$

with

$$\Delta\hat{\tau}^{i,(RTK)}_{r,k} = \frac{f_c}{c}\left(\tilde{\rho}^{i}_{r,k-1} - \hat{\rho}^{i,(RTK)}_{r,k-1}\right) \tag{6}$$

$$\hat{\rho}^{i,(RTK)}_{r,k-1} = \hat{r}^{i,(RTK)}_{r,k-1} + \hat{B}_{r,\rho,t,k-1} + \hat{B}^{i}_{\rho,sys,k-1} \tag{7}$$

$$\hat{r}^{i,(RTK)}_{r,k-1} = \left\|\hat{\mathbf{p}}^{i}_{k-1} - \hat{\mathbf{p}}^{(RTK)}_{r,k-1}\right\| \tag{8}$$

where $\Delta\hat{\tau}^{i,(RTK)}_{r,k}$ is the compensating code phase error; $\tilde{\rho}^{i}_{r,k-1}$ and $\hat{\rho}^{i,(RTK)}_{r,k-1}$ are the measured (from the code loop filter) and predicted pseudoranges (based on the estimated RTK position) at the same epoch; $\hat{r}^{i,(RTK)}_{r,k-1}$ is the predicted geometry distance; $\hat{\mathbf{p}}^{i}_{k-1}$ is the satellite three-dimension (3D) position estimated from the ephemeris; $\hat{\mathbf{p}}^{(RTK)}_{r,k-1}$ is the estimated float RTK position; in (7), $\hat{B}_{r,\rho,t,k-1}$ and $\hat{B}^{i}_{\rho,sys,k-1}$ are the estimated clock bias error and the systematic error, e.g., the atmospheric delay errors and ephemeris errors; how to compute the two values will be stated after.

It is worth adding that different from the traditional VDLL/VDFLL, where the aided code phase error is from the difference between code phases predicted at the current epoch and measured at the previous epoch (Spilker Jr 1996; Lashley 2009; Zhao and Akos 2011; Hsu et al. 2015; Xu and Hsu 2019), the proposed RTK-position-aided code phase error (6) is estimated from the difference between the current-epoch predicted code phase and the current-epoch measured code phase which is seldom mentioned in the previous works; the former and the latter are a relative and an absolute aiding approach, respectively.

To derive the estimators of the clock bias error and the systematic error, we first give the models of the pseudoranges by

$$\rho^{i}_{x,k-1} \triangleq r^{i}_{x,k-1} + \left(B^{i}_{\rho,sys,k-1} + B_{x,\rho,t,k-1}\right) + \varepsilon^{i}_{x,\rho,k-1} \tag{9}$$

where subscripts $x \in \{b, r\}$, b and r correspond to base and rover stations, respectively; $r^{i}_{x,k-1}$, $B^{i}_{\rho,sys,k-1}$, $B_{x,\rho,t,k-1}$, and $\varepsilon^{i}_{x,\rho,k-1}$ are the real geometry distance, systematic error, clock bias error, and the random noise, respectively.

Based on (9), the systematic error is estimated by

$$\hat{B}^{i}_{\rho,sys,k-1} = \tilde{\rho}^{i}_{b,k-1} - \hat{r}^{i}_{b,k-1} - B_{b,\rho,t,k-1} \tag{10}$$

with

$$\hat{r}^{i}_{b,k-1} = \left\|\hat{\mathbf{p}}^{i}_{k-1} - \hat{\mathbf{p}}_{b,k-1}\right\| \tag{11}$$

It is worth noting that the systematic error estimator (10) is a biased estimator affected by the base-station clock bias error $B_{b,\rho,t,k-1}$. Fortunately, the between-receiver single-difference algorithm can remove such biased error and will be stated subsequently.

One needs to use the between-receiver single-difference algorithm to estimate an accurate user's receiver clock bias error. Finally, the estimator for the user clock bias is proposed by

$$\hat{B}_{r,\rho,t,k-1} = \tilde{\rho}^{m}_{r,k-1} - \tilde{\rho}^{m}_{b,k-1} - \hat{r}^{m}_{r,k-1} + \hat{r}^{m}_{b,k-1} + B_{b,\rho,t,k-1} \tag{12}$$

with

$$\hat{r}^{m}_{r,k-1} = \left\| \hat{\mathbf{p}}^{m}_{k-1} - \hat{\mathbf{p}}_{r,k-1} \right\| \tag{13}$$

where superscript m denotes the PRN number of the master satellite, excluding the multipath and non-line-of-sight (NLOS) effect.

Similar to (10), it can also be found that (12) is a biased estimator caused by the base station clock error. However, combining (10) and (12) with a summation operation gives

$$\begin{aligned} \hat{B}^{i}_{\rho,sys,k-1} + \hat{B}_{r,\rho,t,k-1} = \tilde{\rho}^{i}_{b,k-1} - \hat{r}^{i}_{b,k-1} + \tilde{\rho}^{m}_{r,k-1} \\ - \tilde{\rho}^{m}_{b,k-1} - \hat{r}^{m}_{r,k-1} + \hat{r}^{m}_{b,k-1} \end{aligned} \tag{14}$$

where the common biased error $B_{b,\rho,t,k-1}$ with inverse signs is well-eliminated such that an unbiased pseudorange prediction $\hat{\rho}^{i,(RTK)}_{r,k-1}$, see (7), is finally obtained.

Previous contents stated the computation of the predicted pseudorange (7) for the code phase error (6). Then, how to obtain the measured pseudorange of (6) will be introduced.

At the beginning of tracking, still, the initially measured pseudorange is produced by the conventional STL. Once the initialization is done, a compensating code error $\Delta\hat{\tau}^{i,(RTK)}_{r,k}$ can be predicted from the previous epoch. Then, the compensated discriminated code phase error $\Delta\hat{\tau}^{i}_{r,k}$ from (5) is filtered by a second-order loop filter algorithm, and its filtered version is $\Delta\hat{\tau}^{i+}_{r,k}$. After that, such filtered error is used to form the code frequency in the NCO algorithm as follows:

$$\hat{f}^{i}_{\text{code},k+1} = f_c + \hat{f}^{i}_{\text{code,dop},k+1} \tag{15}$$

with

$$\hat{f}^{i}_{\text{code,dop},k+1} = \Delta\hat{f}^{i,(\text{VDFLL})}_{\text{code},k} + \frac{1}{T_{\text{coh}}}\Delta\hat{\tau}^{i+}_{r,k} \tag{16}$$

$$\Delta\hat{f}^{i,(\text{VDFLL})}_{\text{code},k} = -\frac{f_c}{f_r}\tilde{f}^{i}_{\text{carr,dop},k} \tag{17}$$

where f_c is the code rate; $\Delta\hat{f}^{i,(\text{VDFLL})}_{\text{code},k}$ is the compensating frequency with the VDFLL algorithm.

At last, the measured pseudorange is computed as

$$\tilde{\rho}^{i}_{r,k} = c\left(\hat{t}_{r,k} - \frac{1}{f_c}\left(\hat{\tau}^{i}_{\text{code},k} + \hat{f}^{i}_{\text{code},k}T_{\text{coh}}\right)\right) \tag{18}$$

where $\hat{\tau}^{i}_{\text{code},k}$ is the remaining code phase in chips, and it is extracted from the measured week seconds, the frames, the bits, the code cycles within a one-bit interval, the integer code chips within a single code cycle, as well as the initial fractional or absolute code phase of the k th epoch within one sampling interval at the local time $\hat{t}_{r,k}$ (Dierendonck 1996). For example, if the IF sampling rate is 10.23 MHz, as the code rate of GPS L1 C/A signals is 1.023 MHz, then its absolute code phase should not exceed 0.1 chips. Therefore, $\left(\hat{\tau}^{i}_{\text{code},k} + \hat{f}^{i}_{\text{code},k}T_{\text{coh}}\right)$ in the measured pseudorange (18) that contains the next-epoch real absolute code phase information can be used to predict the absolute code phase (6) at the next epoch.

Float RTK system models using pseudorange and carrier phase measurements

At last, to predict a sub-meter- or decimeter-level absolute code phase error, the float RTK algorithm is proposed to compute the user position. This part will discuss how to build the float RTK model using the measured pseudoranges and carrier phases which are obtained from the previously mentioned tracking loops.

Measurement vector

First, the double-difference (DD) measurements are computed as

$$\nabla\Delta\tilde{\rho}^{(i,m)}_{rb,k} = \tilde{\rho}^{i}_{r,k} - \tilde{\rho}^{m}_{r,k} - \tilde{\rho}^{i}_{b,k} + \tilde{\rho}^{m}_{b,k} \tag{19}$$

$$\nabla\Delta\tilde{\phi}^{(i,m)}_{rb,k} = \tilde{\phi}^{i}_{r,k} - \tilde{\phi}^{m}_{r,k} - \tilde{\phi}^{i}_{b,k} + \tilde{\phi}^{m}_{b,k} \tag{20}$$

where $\nabla\Delta\tilde{\rho}^{(i,m)}_{rb,k}$ and $\nabla\Delta\tilde{\phi}^{(i,m)}_{rb,k}$ are the DD pseudorange and carrier phase measurements, respectively.

Next, assuming that the master satellite is tracked at channel 1, the measurement vector is given by

$$\delta\mathbf{z}_k = \begin{bmatrix} \nabla\Delta\tilde{\rho}^{(2,l_m)}_{rb,k} - \nabla\Delta\hat{r}^{(2,l_m)}_{rb,k} \\ \vdots \\ \nabla\Delta\tilde{\rho}^{(L,l_m)}_{rb,k} - \nabla\Delta\hat{r}^{(L,l_m)}_{rb,k} \\ \lambda\nabla\Delta\tilde{\phi}^{(2,l_m)}_{rb,k} - \nabla\Delta\hat{r}^{(2,l_m)}_{rb,k} \\ \vdots \\ \lambda\nabla\Delta\tilde{\phi}^{(L,l_m)}_{rb,k} - \nabla\Delta\hat{r}^{(L,l_m)}_{rb,k} \end{bmatrix} \tag{21}$$

where the predicted DD geometry distance satisfies $\nabla\Delta\hat{r}^{(l_i,l_m)}_{rb,k-1} = \hat{r}^{l_i}_{r,k} - \hat{r}^{l_m}_{r,k} - \hat{r}^{l_i}_{b,k} + \hat{r}^{l_m}_{b,k}$ based on (11) and (13); L denotes the total tracking channel number of the rover receiver and the superscripts $\{1, 2, \cdots, L\}$ denote the respective channel indexes; superscript l_m denotes the index of the channel with the master satellite from the base station; λ is the carrier wavelength.

State error vector

The state error vector of the float RTK algorithm is given by

$$\delta \mathbf{x}_k = \begin{bmatrix} \delta \mathbf{p}_{r,k} \\ \delta \nabla \Delta N_k^{(2,l_m)} \\ \vdots \\ \delta \nabla \Delta N_k^{(L,l_m)} \end{bmatrix}$$

where $\delta \mathbf{p}_{r,k}$ is the state vector of the 3D positioning error, $\delta \nabla \Delta N_k^{(l_i,l_m)}$ is the DD integer ambiguity error, and $l_i \in \{1, 2, \cdots, L\}$ that is the channel index of the rover station.

Design matrix

Then, the design matrix for the DD measurements is

$$\mathbf{H} = \begin{bmatrix} \mathbf{e}_k^2 - \mathbf{e}_k^1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{e}_k^L - \mathbf{e}_k^1 & 0 & \cdots & 0 \\ \mathbf{e}_k^2 - \mathbf{e}_k^1 & \lambda & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{e}_k^L - \mathbf{e}_k^1 & 0 & \cdots & \lambda \end{bmatrix}$$

where, again, $\hat{\mathbf{e}}_k^{l_i}$ is a function of the predicted distance using the 3D position prediction $\hat{\mathbf{p}}_{r,k}^{'(RTK)}$, and it is computed as

$$\hat{r}_{r,k}^{i,(\mathrm{RTK})} = \left\| \hat{p}_k^i - \hat{p}_{r,k}^{'(\mathrm{RTK})} \right\| \tag{22}$$

with

$$\hat{\mathbf{p}}_{r,k}^{'(\mathrm{RTK})} = \hat{\mathbf{p}}_{r,k-1}^{(\mathrm{RTK})} + \hat{\mathbf{v}}_{r,k-1} T_{\mathrm{coh}} \tag{23}$$

where $\hat{\mathbf{p}}_{r,k-1}^{(RTK)}$ is the 3D position directly estimated from the float RTK algorithm at the $k-1$ th epoch.

Next, the measurement noise matrix is given by

$$\mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & (0.01)^2 \end{bmatrix} \otimes \mathbf{Q}$$

with

$$\mathbf{Q} = [\mathbf{q}]_{1:1} \cdot \left(\mathbf{1}_{L-1} \cdot \mathbf{1}_{L-1}^T \right) + \mathrm{diag}\left([\mathbf{q}]_{2:L} \right)$$

$$\mathbf{q} = \left[\left(w_r^1\right)^2 + \left(w_b^1\right)^2, \cdots, \left(w_r^L\right)^2 + \left(w_b^L\right)^2 \right]$$

where $\otimes$ is the Kronecker product operator; $[\cdot]_{m:n}$ denotes a sub-vector composed of the elements from m to n; diag$(\cdot)$ is the diagonal matrix operator; $\mathbf{1}_m$ is the column vector of ones with a dimension of m; $\mathbf{Q}$ is the cofactor matrix; $\left(w_r^l\right)^2$ and $\left(w_b^l\right)^2$ are the weights of the measurements which are computed by the Eq. (12) of (Herrera et al. 2016).

Finally, the extended Kalman filter (EKF) algorithm is used in this work to obtain the float RTK positioning estimation $\hat{\mathbf{p}}_{r,k}^{(RTK)}$. More detailed content can refer to the open-source software package, goGPS v0.4.3 (Herrera et al. 2016). Again, the $\hat{\mathbf{p}}_{r,k}^{(RTK)}$ will be used to compensate for the next-epoch tracking code phase error as described by (5). Thus, a closed loop is formed.

Work process of the proposed VDFPLL

We use the coherent integration time of 5 ms in the proposed RTK-position-aided SDR. Thus, the data rate in the baseband of the rover receiver is 200 Hz. However, the reference receiver from the base station only provides measurements with no higher than a 1-Hz data rate.

The code frequency and phase are jointly aided by the carrier Doppler from the user's velocity and the carrier phase from the RTK position in the proposed RTK-aided tracking architecture, so we nominate it as a vector delay/frequency/phase lock loop (VDFPLL). Then, how the proposed VDFPLL is integrated into the receiver baseband is illustrated in Fig. 3 (bottom), and the diagram of the traditional VDFLL is also plotted in Fig. 3 (top) as a comparison.

Figure 3 (bottom) shows that the proposed tracking strategy repeats every second, where the VDFPLL and the VDFLL are implemented once and four times a cycle, respectively. The VDFPLL is the one based on the proposed RTK-position-aided algorithm, while the VDFLL is the traditional vector tracking one where the projection (3) is computed from the least-square SPP in the SDR. The standard STL takes up the other tracking intervals except for the VDFPLL and VDFLL ones. It is worth mentioning that the choice of the VDFPLL updating rate is due to the reference-rover exchange rate (1 Hz), and the VDFLL one is a design choice that refers to the authors' previous works (Luo et al. 2019a, 2019c).

In the VDFPLL, the SDR uses both the base and rover measurements to compute the RTK position and the proposed predicted pseudorange. More specifically, the user position is computed with the known high-accuracy base station position and the estimated baseline. The differencing measurements between the base and rover stations improve the baseline estimating performance and offer much more accurate estimations of the clock bias and systematic errors. The details of the message exchanges between the reference and rover receivers are stated as follows:

Step 1: the SDR extracts the respective pseudorange and carrier phase measurements by (18) and (4) at the same local time $\hat{t}_{r,k-1}$ when the reference receiver outputs the measurements at the time $\hat{t}_{b,k-1}$. Thus, there is no latency between the two station measurements;

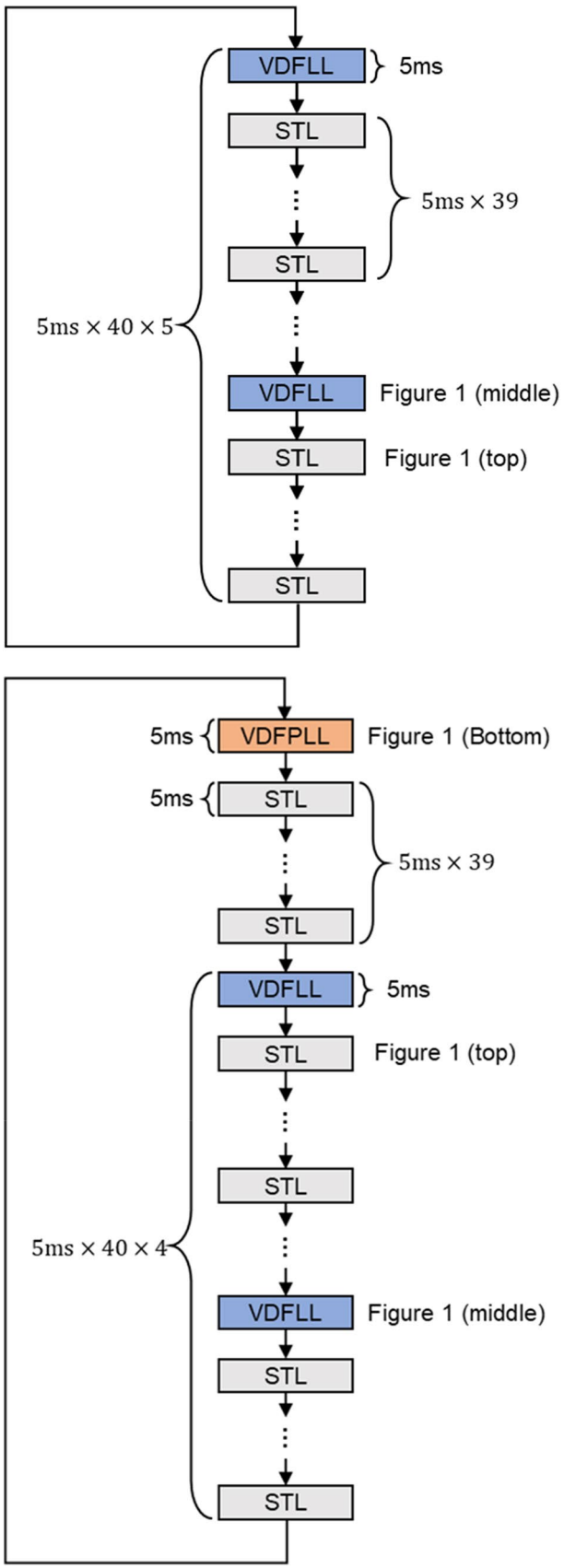


Fig. 3 Tracking processes of the traditional VDFLL (top) and proposed VDFPLL SDRs (bottom) in this work, where the latter repeats each second. More details about the loop filter settings can refer to Table 1. Each block corresponds to a coherent integration interval

Step 2: the RTK engine computes the float RTK position from all the available channels with (19), (20), and (21);
Step 3: the SDR computes the predicted geometry distance $\hat{r}_{r,k-1}^{i,(RTK)}$ by (8);
Step 4: the SDR computes the predicted summation of the systematic error and clock bias error $\left(\hat{B}^{i}_{\rho,sys,k-1} + \hat{B}_{r,\rho,t,k-1}\right)$ by (14);
Step 5: the SDR computes the predicted pseudorange $\hat{\rho}_{r,k-1}^{i,(RTK)}$ by (7);
Step 6: the SDR computes the code phase error by (6) to compensate for the discriminated code error by (5);
Step 7: the SDR updates the code NCO algorithm and generates the locally replicated signal. Then, the algorithm returns to *Step 1* for the next-interval VDFPLL.

Hardware-in-the-loop simulation test

In this section, a HIL testing is conducted to produce the reference of the code phase at the beginning of each updating interval to show the strength of the proposed VDFPLL in reducing the absolute code phase error in the baseband processing.

The reference code phase is generated with the HIL simulation SDR tracking the real-world GPS IF data. A diagram that illustrates the HIL testing process is given in Fig. 4.

On the one hand, in Fig. 4, the HIL simulation SDR starts from receiving the IF signal collected in the real world. Next, such test SDR that simulates reality generates the reference pseudorange through the user's true 3D position as well as the clock bias error, systematic error, and multipath error. Differencing the measured pseudorange and the reference pseudorange gives a code phase error. Then, the error compensates for the discriminating output after the test SDR implements the traditional I&D and code discriminating algorithms. It is worth emphasizing that the measured and reference pseudoranges are formed within the same local time epoch. In this case, the absolute code phase error for the incoming signal and the local replica are time-synchronized between the proposed and test SDRs.

On the other hand, in the test SDR, once the code phase error passes through a tracking loop filter, a code frequency that is used for the NCO algorithm will be obtained. The reference Doppler frequency projected from the user's true 3D velocity will compensate for such code frequency. At last, the code phase reference that reflects the user's true position is obtained.

Table 1 Parameter settings for different SDRs

SDR Type	Diagram	Loop Filter Mode	PLL		DLL	
			Order	Bandwidth	Order	Bandwidth
RTK-VDFPLL (Proposed)	Figure 3 (bottom)	VDFPLL	1st	4 Hz		
		VDFLL	1st	4 Hz		
		STL	3rd	5 Hz		
SPP-VDFPLL	Figure 3 (bottom)	VDFPLL	1st	4 Hz		
		VDFLL	1st	4 Hz		
		STL	3rd	5 Hz	2nd	0.4 Hz
RTK-VDFLL	Figure 3 (top)	VDFLL	1st	4 Hz		
		STL	3rd	5 Hz		
SPP-VDFLL	Figure 3 (top)	VDFLL	1st	4 Hz		
		STL	3rd	5 Hz		
CT	Figure 1 (top)	STL	3rd	18 Hz		

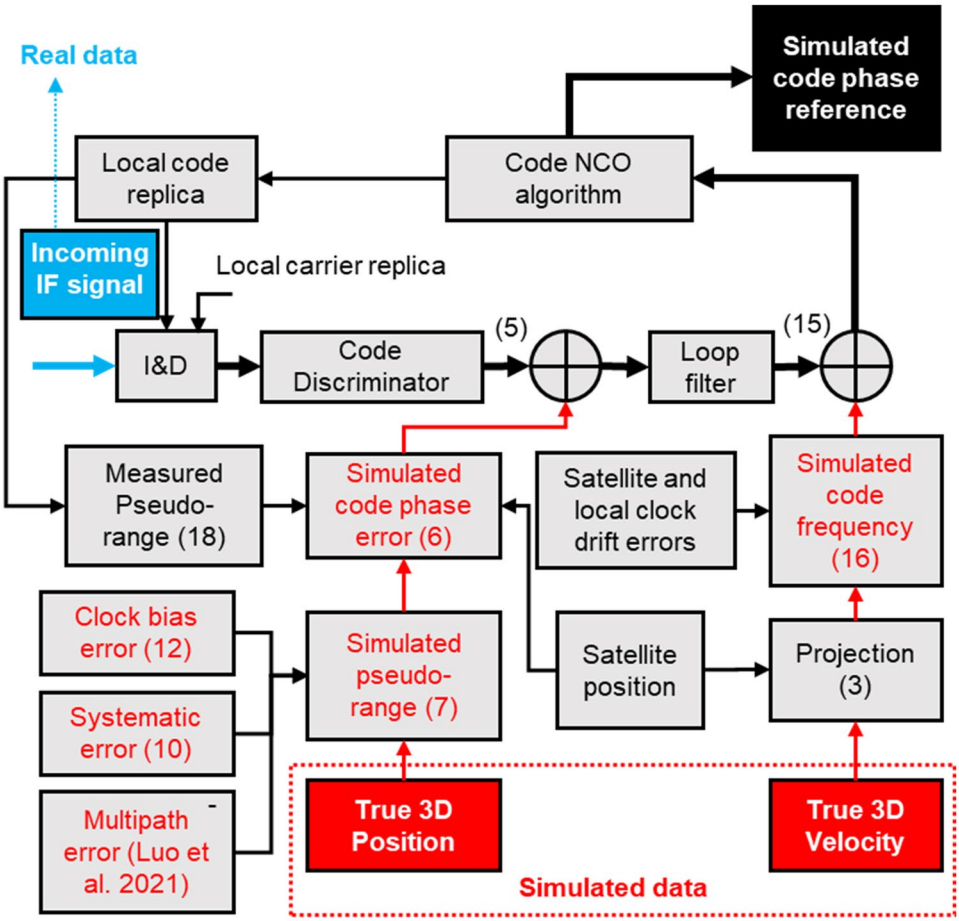


Fig. 4 Diagram of the HIL simulation SDR to generate the code phase reference in a single channel. The incoming GPS IF signal is collected in the real world, and the SDR processing the IF data with the true 3D position and velocity takes place as though the real-world receiver is being used

Again, it should be pointed out that the simulated code phase error (6) and the simulated code frequency (16) are computed from the known truths instead of the RTK results. More details about the HIL simulation SDR design can refer to Luo et al. (2021).

The static GPS L1 C/A IF data are collected in an open-sky area in the city of Calgary to be processed with the SDR platform. Parameter settings of the vector GNSS SDR algorithm are provided: the T_{coh} is set as 5 ms. The early-late correlator spacing is 0.4 chips. The tracking process is introduced in Fig. 3. This tracking strategy is selected

based on empirical results which can refer to the authors' previous works (Luo et al. 2019a, 2019c). Thus, it will not be discussed in detail.

The sky plot for this experiment is shown in Fig. 5. The IF data of incoming GPS L1 C/A signals are collected with the Fraunhofer front-end, the sampling rate is 10.125 MHz, and the quantization type is complex int8. It is worth emphasizing some received satellite signals could also be interfered with the signal reflection of far-away buildings or ground.

Different traditional GPS SDRs will be compared with the proposed SDR in the simulations and the experiments in the subsequent section. Therefore, how the loop filters and parameter settings apply to these SDRs is listed in Table 1 where "CT" corresponds to the SDR with the conventional STL (Kaplan and Hegarty 2017), "RTK-VDFLL" represents the traditional VDFLL based on the RTK positioning projection (Lashley et al. 2009), and "RTK-VDFPLL" is the proposed SDR, "SPP-VDFLL" represents the traditional VDFLL algorithm based on the SPP projection, and "SPP-VDFPLL" is the SDR using the SPP solution to predict the pseudorange (see (7)) in the VDFPLL.

The proposed RTK-position-aided VDFPLL algorithm is conducted in the experiment to generate the code phase $\hat{\tau}^i_{\text{code},k}$ that contains the absolute code phase information as mentioned in (18). At the same time, the CT and the VDFLL algorithm are also experimented with as comparisons. Finally, the simulation puts out the code phase reference $\tau^i_{\text{code},k}$ based on the user's true position and velocity.

Aiming to show how the proposed algorithm improves the code error accuracy in the baseband processing, the absolute code phase errors $\delta\tau^i_{\text{code},k}$, that is computed by $\hat{\tau}^i_{\text{code},k} - \tau^i_{\text{code},k}$ is plotted in Fig. 6 (Luo et al. 2021).

According to Fig. 6, the traditional VDFLL performs better than the CT in removing the code phase bias if the channel does not contain only a single LOS signal. However, some bias errors still remain. The proposed VDFPLL eliminates the entire code phase errors as the results curve fluctuates more closely to the zeros.

The code delay error curve estimated from the channel PRN21 less affected by the multipath effect is also plotted in Fig. 7 as a comparison. The results show that the code delay error difference (around 3 m) between the proposed VDFPLL and the traditional two methods is much smaller than the ones of PRN10 (around 10 and 17 m). It means that the proposed VDFPLL improves the code phase accuracy more readily in a multipath channel than in a LOS channel.

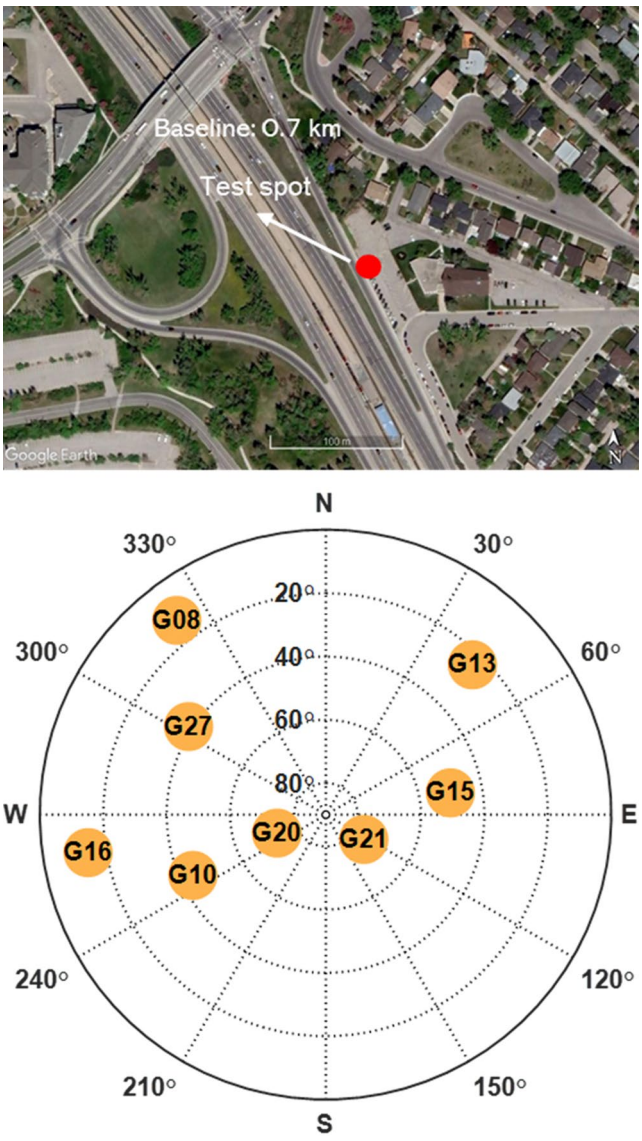


Fig. 5 Test spot (top) and sky plot (bottom) of the open-sky static experimental area in Calgary

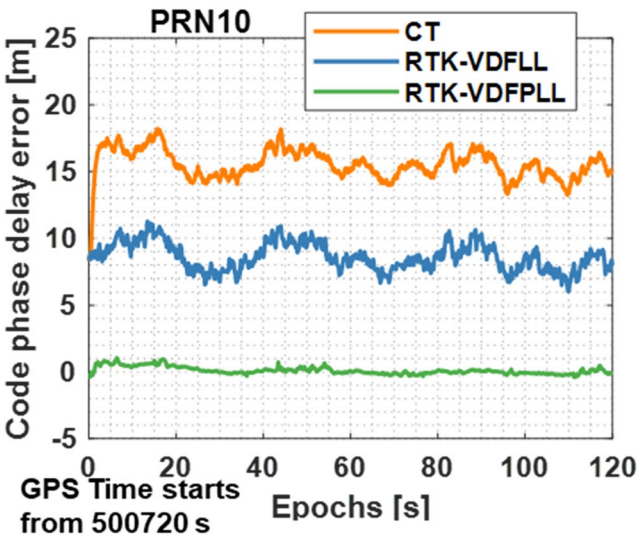


Fig. 6 HIL simulation test and experimental results of the absolute code phase errors for PRN10 with multipath interference

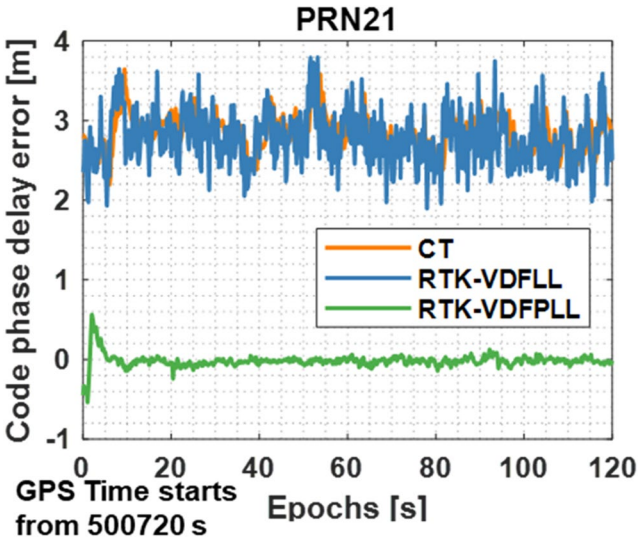


Fig. 7 HIL simulation test and experimental results of the absolute code phase errors for PRN21 without multipath interference

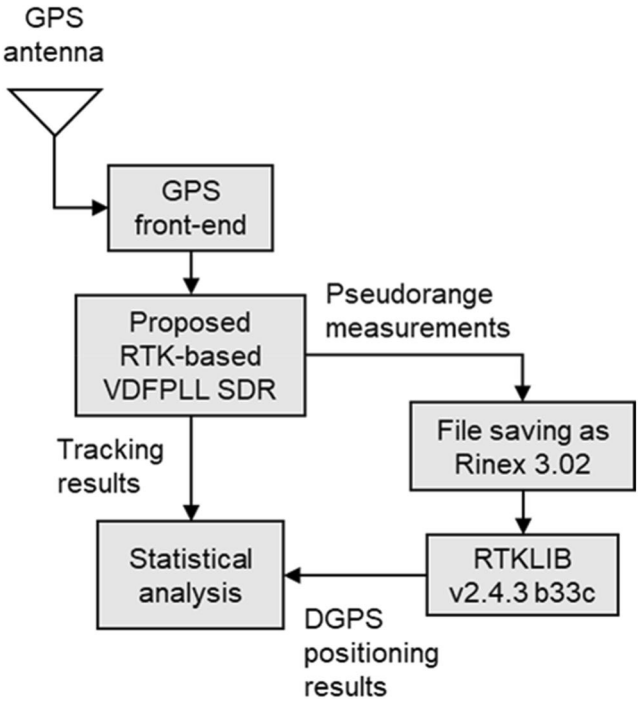


Fig. 8 Assessment process for the pseudorange measurement quality from the proposed RTK-based VDFPLL SDR

In summary, since the reference is a mapping of the true position in the baseband, the proposed algorithm has a more positive effect on the baseband signal accounting for a more accurate position. The measurement and positioning qualities are experimented with to verify this point of view subsequently.

Experiment results and discussions

Two static experiments are conducted in open-sky and light-urban areas to verify the proposed RTK-position-aided VDFPLL GPS SDR. The GPS L1 C/A IF data are collected in an open-sky area in the city of Calgary, while the data for the latter situation are collected in a light-urban area of Hong Kong.

This research focuses on improving the baseband code tracking, so the proposed SDR's pseudorange quality will be validated. It will be verified with both the tracking and differential GPS (DGPS) positioning results. The assessment methods are illustrated in Fig. 8.

Open-sky area

The open-sky situation has been introduced in the previous section. The sky plot for the testing area and the visible satellites are also shown in Fig. 5.

Before discussing the measurement performance, we would like to indicate the different modes for the proposed float-RTK-based VDFPLL SDR operations. After the code phase and Doppler of all the available satellites are acquired, the proposed receiver starts from the standard scalar tracking. Then, after the bit and frame synchronization, the receiver starts to output velocity estimation. In this case, the standard VDFLL mode is first activated. Next, after 8 s of the frame synchronization, the receiver begins to compute the float RTK position at the GPS time 500714 s. So, the proposed VDFPLL mode starts to work at this moment. The curves of the float RTK positioning errors computed by the proposed receiver are plotted in Fig. 9. The results show that the receiver takes time to converge the float RTK solutions. Although some data that are not fully converged over the processing time for the statistical analysis about the pseudorange performance, it will not affect the conclusion of this work.

The code-minus-carrier (CMC) error is commonly used to assess the GNSS measurement performance in multipath mitigation (Hwang et al. 1999). Then, the curves of the CMC errors for the visible PRN10 are plotted in Fig. 10 (top). The results show that the CT is significantly affected by multipath interference. While the SPP- and RTK-VDFLL algorithms perform very closely, their results are better than the CT ones. The overall CMC errors of the proposed RTK-VDFPLL are slightly smaller than the two VDFLL algorithms. As discussed in the previous section, the results are consistent with the HIL simulation results.

It is worth noting that the CMC error curve of the proposed RTK-VDFPLL overlaps for some epochs in the PRN10 channel. The reason is that the pseudorange

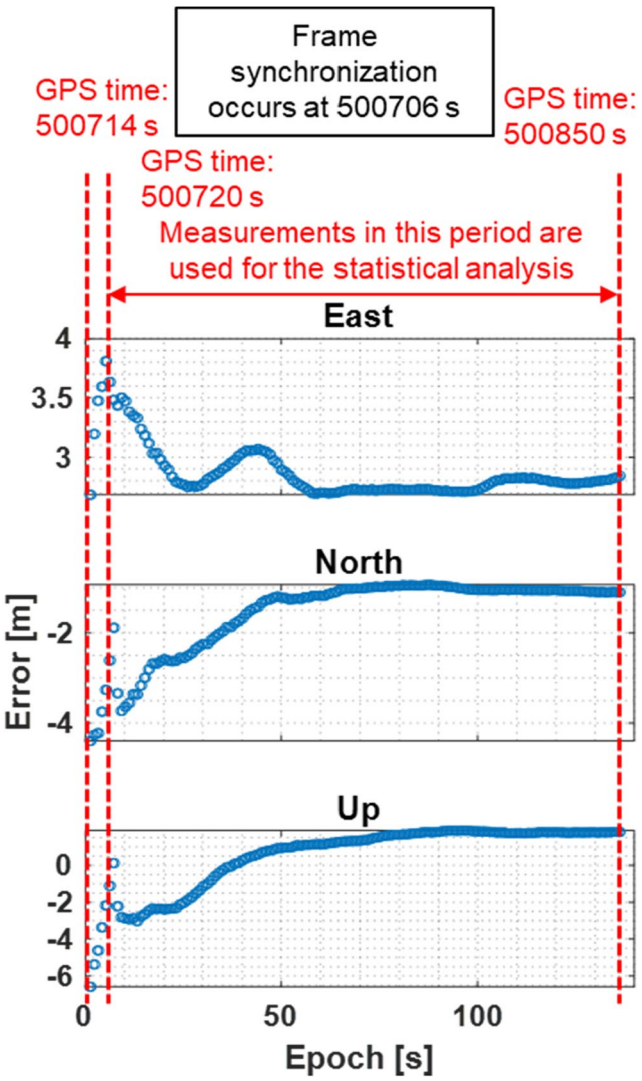


Fig. 9 Curves of the float RTK positioning errors in the open sky computed by the proposed RTK-based VDFPLL SDR (real-time positioning results). The converged positioning errors do not have a sub-meter level due to the multipath interference

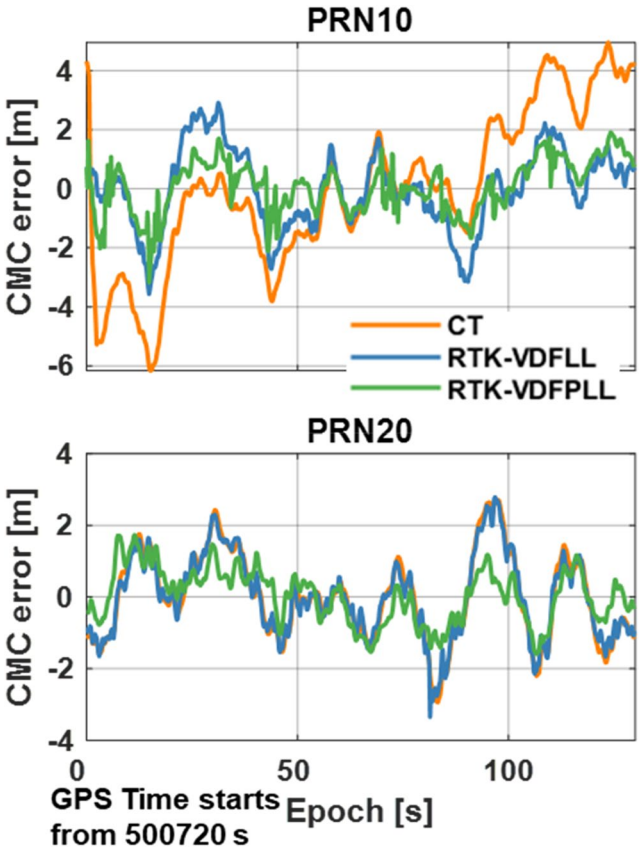


Fig. 10 Curves of CMC error estimations. The multipath mitigation performance of the proposed RTK-based VDFPLL algorithm is better than the traditional algorithms in both severe and slight interference cases

measurement is constantly interfered with by the multipath effect, which is not well mitigated in the code discriminator. Again, the pseudorange prediction cannot sufficiently match the change of the multipath signal, so the code phase delay error computed by the difference between the pseudorange measurement and prediction causes a severe shift in the discriminated code phase error leading to the inconstancy.

If the multipath interference is not severe, the CMC error can still be suppressed, and the curve will also be smooth, as shown in Fig. 10 (bottom), where the results of the PRN20 satellite with a higher elevation angle are provided.

To clearly show the error distribution, we also illustrate the cumulative distribution function (CDF) curves of the CMC errors in Fig. 11. The result proves that the proposed algorithm can even elevate the GNSS SDR in an open-sky area. In other words, the upper bound of the navigation performance has been further lifted. This result is meaningful to the design of the next-generation GNSS-based navigation system.

Furthermore, in order to verify the presented VDFPLL GNSS SDR and assess its pseudorange measurements from the positioning results, an open-source package, i.e., RTKLIB v2.4.3 b33c (Takasu and Yasuda 2009), is used to calculate the DGPS results from the pseudorange measurements produced by the proposed receiver.

Figure 12 (top) shows that the horizontal positioning estimations of the RTK-VDFPLL are the least noisy among all tested algorithms. Besides, its averaging value is much closer to the true value. Therefore, the proposed RTK-VDFPLL approach has both the highest precision and accuracy.

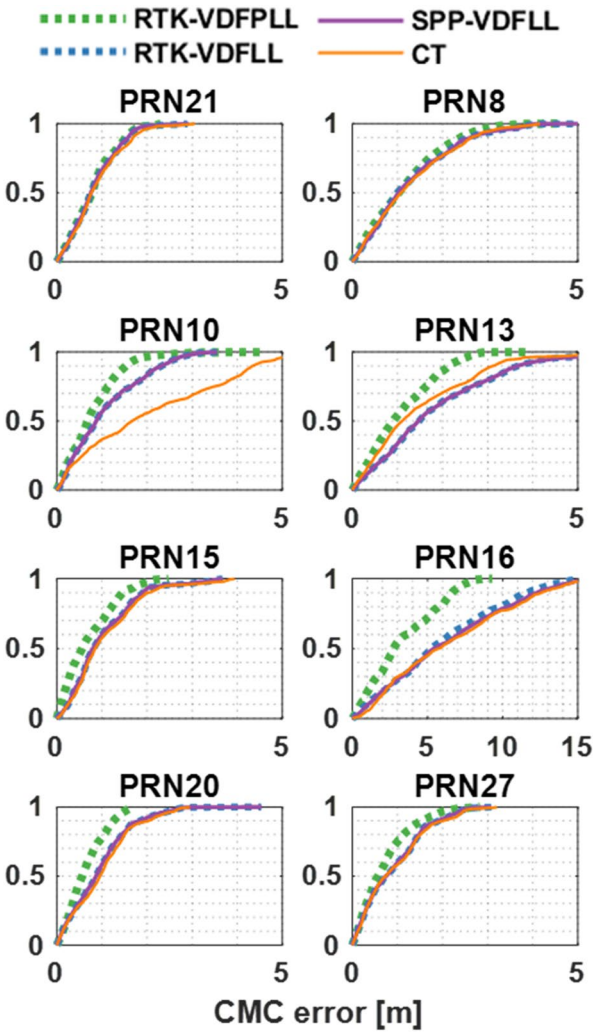


Fig. 11 CDF curves of the CMC errors in the open sky. The proposed RTK-based VDFPLL SDR has the highest performance, and the maximum improvement is 48.5% (within 95% errors of PRN 16)

The CDF curves of the 3D positioning root-mean-squared errors (RMSEs) are also plotted in Fig. 12 (bottom) to show the overall positioning performance. The two VDFLL performances exceed the CT only when the 68% errors are considered; otherwise, there is no improvement if it is out of this range. The 3D positioning accuracy always shows the best performance for the proposed algorithm, and its maximum improvement is 42.5% where the 100% error is compared to the SPP-VDFLL algorithm.

Light-urban area

Figure 13 shows the test spot in Hong Kong and the sky plot of visible satellites. The deep-color area in this plot reflects

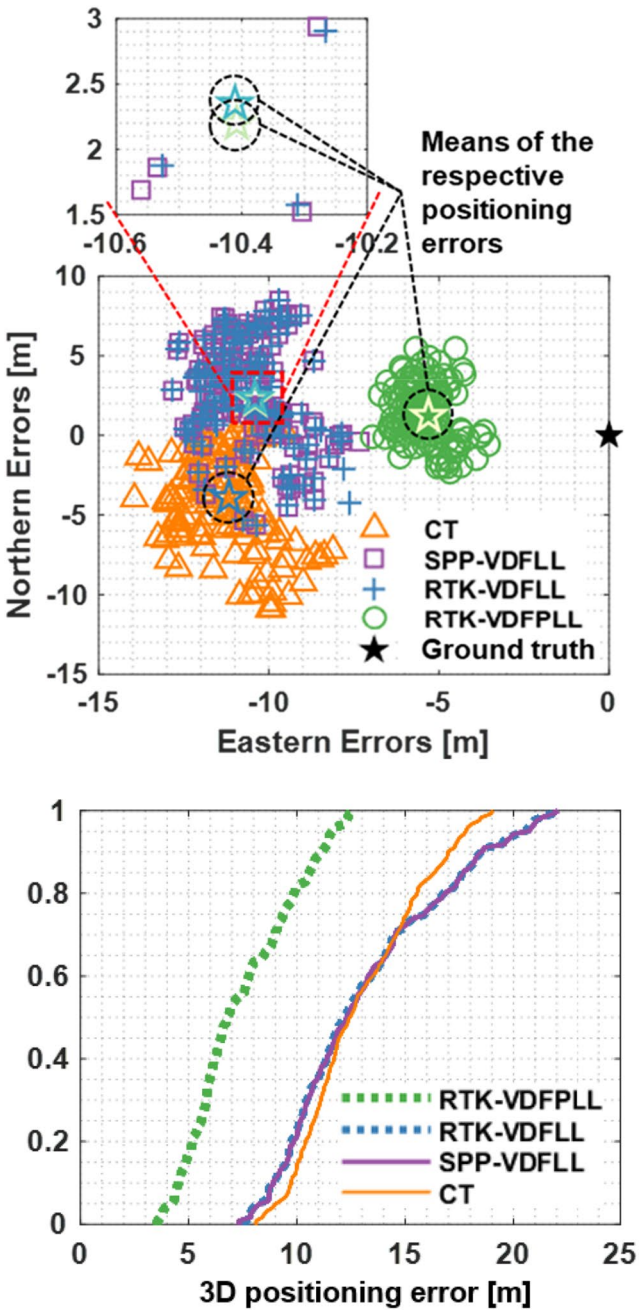
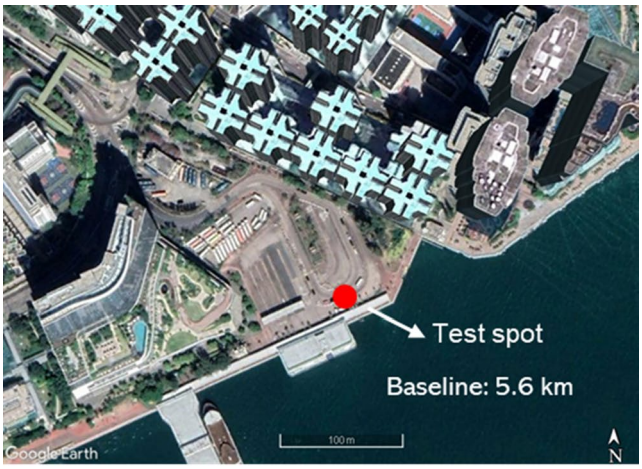


Fig. 12 Horizontal plot of the DGPS positioning errors (top) and CDF curves of the DGPS 3D positioning RMSEs (bottom) (post-processing positioning results from RTKLIB)

the mask of the surrounding buildings with respect to the GNSS antenna. It can be inferred that the NLOS signals are not received. The GPS L1 C/A IF data are collected with LabSat3 Wideband, where the sampling rate is 58 MHz, and the quantization type is complex 2 bits.



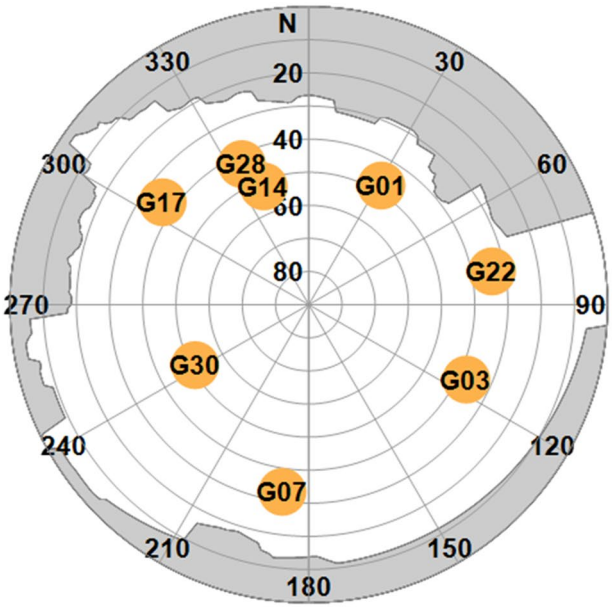


Fig. 13 Sky plot and its sky mask of the urban experiment area in Hong Kong

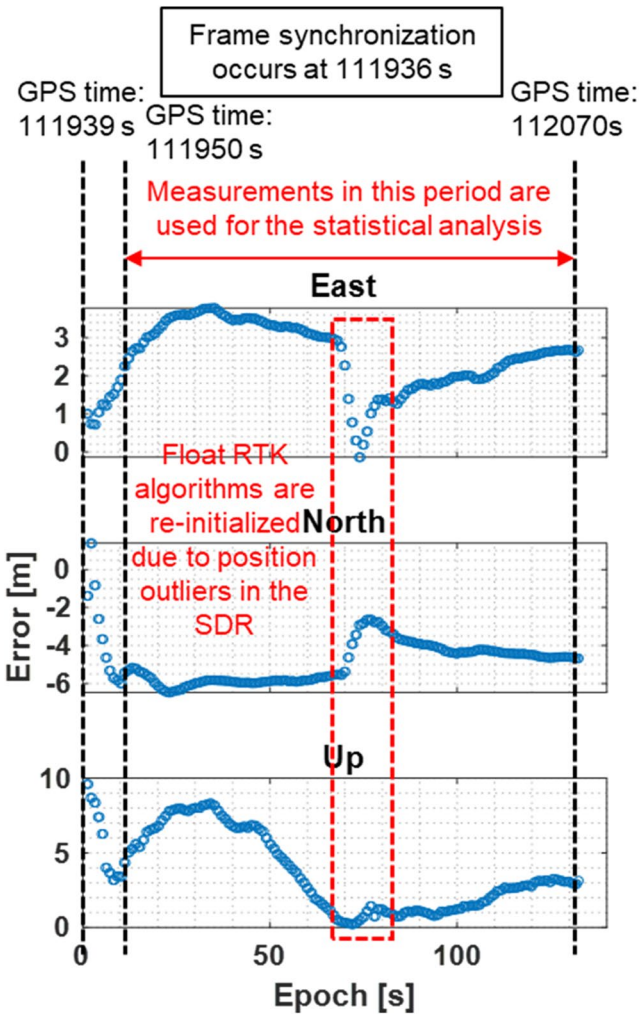


Fig. 14 Curves of the float RTK positioning errors in the light urban area computed by the proposed RTK-based VDFPLL SDR (real-time positioning results). The results are worse than the open-sky data, but the proposed SDR also raises the pseudorange measurement quality

The operational modes of the proposed RTK-based VDFPLL SDR are the same as those used for the open-sky data signal processing. The frame synchronization and the standard VDFLL occur at the GPS time 111936 s. The proposed VDFPLL mode starts at 3 s after. The float RTK position errors computed from the proposed SDR are plotted in Fig. 14. It shows that the float RTK positioning is more challenging to converge in the urban area than in the open sky. Even if the float RTK results are not as good as enough in urban environments, it still improves the SDR performance with the proposed VDFPLL algorithm. More details will be discussed subsequently. It is worth mentioning that since the positioning algorithms are not well designed in the SDR, the proposed VDFPLL SDR performance will be improved further once the SDR RTK algorithm is optimized. We will complete it in our future research.

The CDF curves of the CMC errors are given in Fig. 15. The RTK-VDFPLL algorithm still performs best in this more challenging environment. However, the SPP cannot be used in the VDFPLL architecture as its related measurement

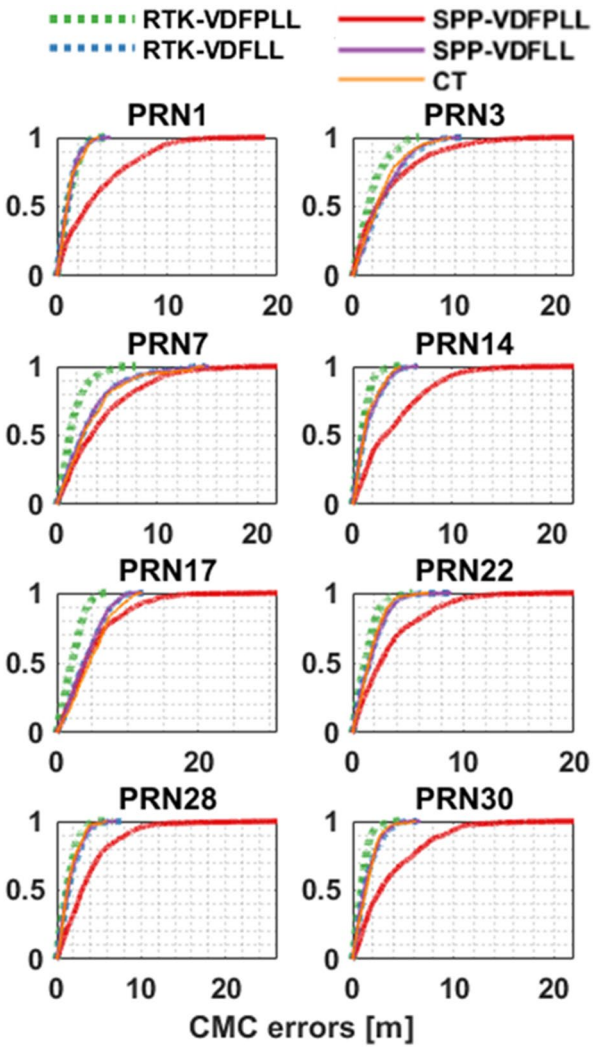


Fig. 15 CDF curves of the CMC RMSEs in the urban area. The proposed RTK-based VDFPLL SDR has the highest performance in all the tracking channels

quality is highly affected by the inaccurate positioning estimation.

Furthermore, the horizontal and the vertical positioning performance produced by the DGPS algorithm from the RTKLIB package is illustrated in Fig. 16. It shows that the two VDFLL algorithms embrace very similar positioning accuracy to the CT one. On the other hand, both VDFPLLs produce less noisy solutions, but more outliers are obtained from the SPP-VDFPLL algorithm due to the wildly inaccurate position estimations. Finally, the RTK-VDFPLL is undoubtedly the most precise and reliable among all the testing algorithms.

It shows that the horizontal positioning means of the proposed RTK-VDFPLL perform worse than the ordinary VDFLL algorithms. To further explore the reason, the positioning accuracies are computed and listed in Table 2. As a result, some results can be found:

First, the horizontal results from the two VDFPLL algorithms are both more biased than the VDFLL and CT algorithms, while the vertical results of the VDFPLL are significantly improved. For example, compared with the traditional RTK-VDFLL, the proposed RTK-VDFPLL has improved the mean, standard deviation (STD), and RMSE of the vertical positioning results by 98.2%, 45.5%, and 50.9%, respectively. Because the signal reflection affects the horizontal RTK position, the multipath effect is less likely to affect the local signals projected in the vertical direction. The proposed VDFPLL can naturally produce the measurements, removing the multipath error in this direction. For example, the much smaller means of the RTK/SPP-VDFPLL in the up direction, i.e., -0.122 m and 0.595 m, can verify such discussion.

Second, still referring to the positioning means, it shows that the SPP- and RTK-VDFPLL results have more significant changes on the means while the two VDFLL algorithms do not when compared with the CT results. Eventually, it proves that the proposed VDFPLL functions to remove the code phase bias, as stated in the contributions (second) in the introductory section.

Third, the STDs of the proposed RTK-VDFPLL in all three directions are smaller than the other algorithms. For instance, compared to the traditional RTK-VDFLL, the RTK-VDFPLL has raised the accuracy by 39.8%, 45.0%, and 45.5% in eastern, northern, and vertical directions, respectively. It means the proposed algorithm can also suppress the random noise of the code-based measurement, even if the RTK solution is more biased due to the multipath interference. In other words, the proposed RTK-VDFPLL improves the precision of the GNSS baseband in urban areas.

Conclusions

This work proposes and achieves a vector GPS L1C/A SDR based on the vector delay/frequency/phase lock loop (VDFPLL) by deeply integrating the float RTK position into the GPS baseband code tracking. Besides, the user's clock bias and systematic errors are precisely estimated using base station information. Furthermore, the static GPS IF data are collected in the real world, and the experiments

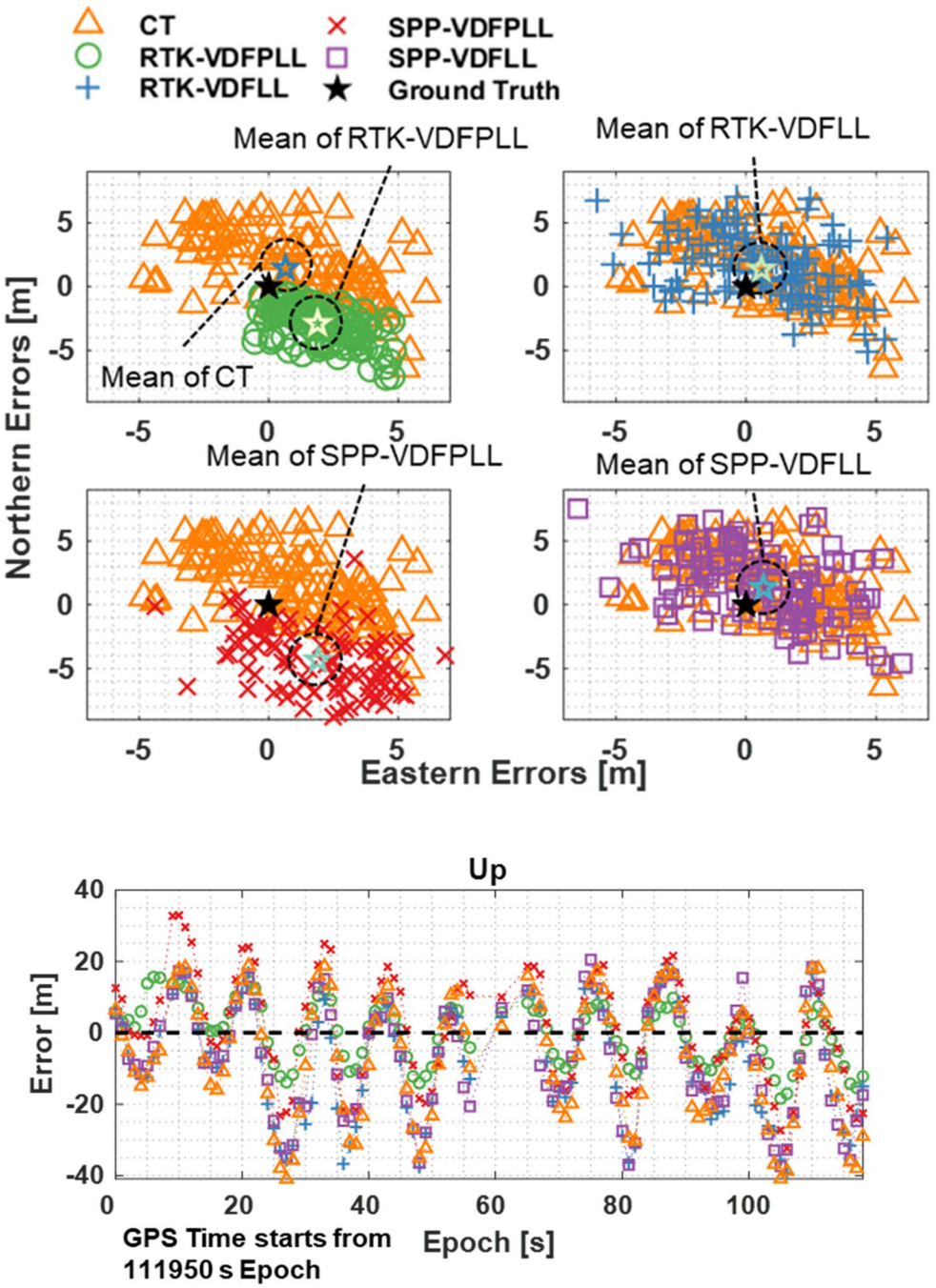


Fig. 16 Horizontal (top) and vertical (bottom) DGNSS positioning errors (post-processing positioning results from RTKLIB). The proposed RTK-based VDFPLL SDR provides the pseudorange measurements that produce the most precise and stable positioning results

are implemented in both open-sky and light-urban areas to verify the proposed algorithm. Finally, by comparing the proposed algorithm with the traditional ones, we draw some conclusions:

1. The proposed RTK-position-aided VDFPLL has increased all the visible pseudorange measurement quality by a maximum of 48.5% and its 3D positioning accuracy by a maximum of 42.5% in an open-sky area;
2. Both the biased and random error accuracies of the vertical positioning results have been significantly improved with the proposed RTK-position-aided VDFPLL SDR by 98.2% and 45.5% in an urban area;

Table 2 Performance of the DGPS positioning (post-processing positioning results from RTKLIB) with different algorithms

SDR	Direction	Mean (2 σ) [m]	STD (2 σ) [m]	RMSE (2 σ) [m]
RTK-VDFPLL	East	1.888	1.295	2.286
	North	−2.893	1.270	3.157
	Up	−0.122	7.855	7.818
SPP-VDFPLL	East	1.901	2.145	2.859
	North	−4.324	2.888	5.192
	Up	0.595	13.611	13.559
RTK-VDFLL	East	0.590	2.151	2.220
	North	1.356	2.309	2.669
	Up	−6.914	14.423	15.933
SPP-VDFLL	East	0.670	2.246	2.334
	North	1.353	2.449	2.788
	Up	-5.694	14.375	15.398
CT	East	0.646	2.472	2.543
	North	1.452	2.445	2.834
	Up	-5.536	15.835	16.703

3. The urban pseudorange-based positioning precision has been improved with the RTK-position-aided VDFPLL in the three directions by 39.8%, 45.0%, and 45.5%, respectively.

Acknowledgements This research has been supported by the funding of Prof. Naser El-Sheimy from NSERC CREATE and Canada Research Chairs programs. This work was also sponsored in part by the National Natural Science Foundation of China (42004014) and partly by the Natural Science Foundation of Jiangsu Province (BK20200530).

Data Availability The GPS IF datasets analyzed during the current study are available from the corresponding author on reasonable request.

References

Chen S, Gao Y (2019) Improvement of carrier phase tracking in high dynamics conditions using an adaptive joint vector tracking architecture. GPS Solut 23(1):15. https://doi.org/10.1007/s10291-018-0806-y

Dierendonck AJ Van (1996) GPS Receivers. In: Parkinson BW, Spilker Jr JJ, Axelrad P, Enge P (eds) Global Positioning System: Theory and Applications, Volume 1. American Institute of Aeronautics and Astronautics, Inc., Washington D.C.

Faragher R, Couronneau N, Powe M, Esteves P, Crockett M, Martin H, Ziglioli E, Higgins C, Buckle D, Zigliol E, Higgins C, Buckle D (2018) Supercorrelation: Enhancing the Accuracy and Sensitivity of Consumer GNSS Receivers with a DSP Upgrade. In: Proc. ION GNSS 2018, Institute of Navigation, Miami, Florida, USA, Sep 24–28. pp 357–375

Faragher R, Powe M, Esteves P, Couronneau N, Crockett M, Martin H, Ziglioli E, Higgins C (2019) Supercorrelation as a Service: S-GNSS Upgrades for Smartdevices. In: Proc. ION GNSS 2019, Institute of Navigation, Miami, Florida, USA, Sep 16–20. pp 256–274

Groves PD, Zhong Q, Faragher R, Esteves P (2020) Combining Inertially-aided Extended Coherent Integration (Supercorrelation) with 3D-Mapping-Aided GNSS. In: Proc. ION GNSS 2020, Institute of Navigation, Sep 22–25. pp 2327–2346

Herrera AM, Suhandri HF, Realini E, Reguzzoni M, de Lacy MC (2016) goGPS: open-source MATLAB software. GPS Solut 20(3):595–603. https://doi.org/10.1007/s10291-015-0469-x

Hsu LT, Jan SS, Groves PD, Kubo N (2015) Multipath mitigation and NLOS detection using vector tracking in urban environments. GPS Solut 19(2):249–262. https://doi.org/10.1007/s10291-014-0384-6

Hwang PY, McGraw GA, Bader JR (1999) Enhanced differential GPS carrier-smoothed code processing using dual-frequency measurements. Navig J Inst Navig 46(2):127–137. https://doi.org/10.1002/j.2161-4296.1999.tb02401.x

Kaplan ED, Hegarty C (2017) Understanding GPS. Principles and applications, 3rd edn. Artech house

Lachapelle G, Gratton P, Horrelt J, Lemieux E, Broumandan A (2018) Evaluation of a low cost hand held unit with gnss raw data capability and comparison with an android smartphone. Sensors 18(12):4185. https://doi.org/10.3390/s18124185

Lashley M (2009) Modelling and performance analysis of GPS vector tracking algorithms. Auburn University

Lashley M, Bevly DM (2013) Performance comparison of deep integration and tight coupling. Navig J Inst Navig 60(3):159–178. https://doi.org/10.1002/navi.43

Lashley M, Bevly DM, Hung JY (2009) Performance analysis of vector tracking algorithms for weak GPS signals in high dynamics. IEEE J Sel Top Signal Process 3(4):661–673. https://doi.org/10.1109/JSTSP.2009.2023341

Luo Y, Hsu L-T, Xiang Y, Xu B, Yu C (2021) An absolute-position-aided code discriminator towards gnss receivers for multipath mitigation. In: Proc. ION GNSS 2021, Institute of Navigation, St. Louis, Missouri, USA, Sep 20–24. pp 3772–3782

Luo Y, Li J, Yu C, Lyu Z, Yue Z, El-Sheimy N (2019a) A GNSS software-defined receiver with vector tracking techniques for land vehicle navigation. In: Proc. ION 2019a Pacific PNT Meeting, Honolulu, Hawaii, USA, April 8–11. pp 713–727

Luo Y, Li J, Yu C, Wang J, El-Sheimy N (2019b) Performance of Carrier-Based High-Precision Positioning Aided by a Vector Tracking Technique for Land Vehicle Navigation. In: The 11th International Conference on Mobile Mapping Technology (MMT 2019b),Shenzhen, China. Shenzhen, China, pp 283–290

Luo Y, Li J, Yu C, Xu B, Li Y, Hsu L-T, El-Sheimy N (2019c) Research on time-correlated errors using allan variance in a kalman filter applicable to vector-tracking-based gnss software-defined receiver for autonomous ground vehicle navigation. Remote Sens 11(9):1026. https://doi.org/10.3390/rs11091026

Ren T, Petovello MG (2017) A stand-alone approach for high-sensitivity GNSS receivers in signal-challenged environment. IEEE Trans Aerosp Electron Syst 53(5):2438–2448. https://doi.org/10.1109/TAES.2017.2699539

Spilker Jr JJ (1996) Fundamentals of signal tracking theory. In: Parkinson BW, Spilker Jr JJ, Axelrad P, Enge P (eds) Global Positioning System: Theory And Applications, Volume 1. American Institute of Aeronautics and Astronautics, Inc., Washington, DC

Takasu T, Yasuda A (2009) Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB. In: Proceedings of the International symposium on GPS/GNSS, Jeju, Korea. pp 4–6

Watts TM, Martin SM (2020) Cooperative Navigation with the Vector Delay/Frequency Lock Loop. In: Proc. ION GNSS 2020, Institute of Navigation, Sep 22–25. pp 2803–2821

Xu B, Hsu L-T (2019) Open-source MATLAB code for GPS vector tracking on a software-defined receiver. GPS Solut 23(2):46. https://doi.org/10.1007/s10291-019-0839-x

Xu B, Jia Q, Hsu L-T (2020) Vector tracking loop-based GNSS NLOS detection and correction: algorithm design and performance analysis. IEEE Trans Instrum Meas 69(7):4604–4619. https://doi.org/10.1109/TIM.2019.2950578

Yang C, Pany T, Soloviev A (2017) An Implementation of variable IF tracking loop (VITAL) and initial test results. Navigation 64(4):515–533. https://doi.org/10.1002/navi.212

Zhao S, Akos D (2011) An open source GPS/GNSS vector tracking loop - Implementation, filter tuning, and results. In: Proc. ION ITM 2011, Institute of Navigation, San Diego, CA, USA, Jan 24 - 26. pp 1293–1305

Yiran Luo is a Postdoctoral Associate at University of Calgary. She received joint Ph.D. degrees from the Department of Geomatics Engineering, University of Calgary, and the School of Information and Electronics, Beijing Institute of Technology. She was a Postdoctoral Fellow with the Department of AAE, Hong Kong Polytechnic University. Her research interests include advanced GNSS receiver design, super-resolution GNSS signal processing, and urban/indoor navigation.

Li-Ta Hsu received a B.S. and Ph.D. degree in aeronautics and astronautics from National Cheng Kung University, Taiwan, in 2007 and 2013, respectively. He is currently an associate professor with the Aeronautical and Aviation Engineering Department, Hong Kong Polytechnic University, and served as a postdoctoral researcher in the Institute of Industrial Science at the University of Tokyo, Japan. In 2012, he was a visiting scholar in University College London, U.K. He is an Associate Fellow of RIN. His research interests include GNSS positioning in challenging environments and localization for pedestrians, autonomous vehicles, and unmanned aerial vehicles.

Zhetao Zhang is an Associate Professor at the Hohai University. He obtained his Ph.D. with distinction from Tongji University. His current research focuses on the precise positioning and navigation with GNSS and multi-source sensors under complex conditions. He was a Postdoctoral Fellow and visiting Ph.D. student at the Hong Kong Polytechnic University and the University of Calgary, respectively.

Naser El-Sheimy is a Professor at University of Calgary. He is the Founder and the CEO of PPI and Founder of METI. He is a fellow of the Canadian Academy of Engineering and the U.S. ION. He is also the Tier-I Canada Research Chair in Geomatics Multi-Sensor Systems. His research expertise includes geomatics multi-sensor systems, GPS/INS integration, and mobile mapping systems.