QUALITY ANALYSIS OF SMARTPHONE GNSS OBSERVATIONS AND IMPACT ON PRECISE POSITIONING

F Zangenehnejad [a] *, Y Gao [a]

[a] Department of Geomatics Engineering, Schulich School of Engineering, University of Calgary,

Calgary, AB, Canada; (farzaneh.zangenehnej, ygao)@ucalgary.ca

KEY WORDS: Smartphone positioning, Precise point positioning (PPP), Smartphone GNSS observations, Smartphone GNSS logging Apps, GnssLogger App, Geo++ RINEX logger App, UofC CSV2RINEX tool

ABSTRACT:

The use of low-cost and ultra-low-cost receivers such as smartphones and wearables has been raising these days due to their affordability and widespread availability. Along with the release of Android Version 7 in 2016, the raw GNSS measurements became accessible through the location API, which includes the GNSSMeasurement class and GNSSClock class. However, the users are required to extract the typical GNSS observations, such as pseudorange, carrier-phase, and Doppler observations, from the raw data coming from these two classes. It should be mentioned some quality concerns may arise during the conversion process from the raw GNSS measurements to the typical GNSS measurements. These quality concerns would subsequently affect GNSS data processing such as cycle slip detection, code smoothing and ultimately positioning performance. In this paper, we first analyse the quality of GNSS observations logged from smartphones. Furthermore, we address potential issues that can arise during the generation of GNSS observations. We also introduce our developed in-house software, UofC CSV2RINEX, which is designed to convert CSV files into RINEX files.

1. INTRODCTION

Precise point positioning (PPP) has become a powerful technique to support accurate positioning with the global navigation satellite system (GNSS) observations from a single receiver. Unlike the relative positioning method, PPP doesn't require a reference station with known precise coordinates, which significantly reduces operational costs and complexity (Zumberge, 1997; Kouba and Héroux, 2001). It can therefore be employed in a wide range of applications, including scientific and civilian applications. PPP can provide centimeter to decimeter-level positioning accuracy in both static and kinematic modes. Such accuracy levels can be generally obtained using the high-end geodetic receivers. However, in recent years, the use of low-cost and ultra-low-cost receivers has been increasing due to their wide use in mass-market applications demanding precise position information. Many latest smartphones have also been equipped with advanced carrier-phase tracking measurement technology to support precise position determination. Improving smartphone positioning is currently a hot research topic, but it requires developing advanced positioning algorithms to overcome the inherent large noise in GNSS observations from the smartphones. The final accuracy obtainable for smartphone positioning significantly depends on the quality of smartphone GNSS observations. Therefore, a comprehensive analysis of the smartphone GNSS observations is necessary. These observations include pseudorange, carrier-phase, Doppler shift, and carrier-to noise density ratio (C/N0). In 2016, Google released an open-source application called GnssLogger App that logs raw measurements from the android.location application programming interface (API) of GnssClock and GnssMeasurement classes. However, the new API does not provide the typical GNSS observations (i.e., pseudorange, carrier-phase, and Doppler observations) directly to the users. Users must generate GPS time, pseudorange, carrier-phase, and Doppler observations themselves using the raw measurements from the android.location. It is available at https://play.google.com/store/apps/details?id=com.google.android.apps.location.gps.gnsslogger&hl=en&gl=US

At first, this app only provided the android.location API raw measurements in CSV format including all types of location and sensor data such as GNSS and other sensor data. However, in the updated versions, the GNSS observations can be directly saved in Receiver INdependent Exchange (RINEX) format as well. It is also capable of logging sensor data such as accelerometer, gyroscope and magnetometer data. In 2017, the Geo++ GmbH Company launched Geo++ RINEX Logger App, an open-source application that generates the GNSS pseudorange, carrier-phase, and Doppler observations in the RINEX format (Geo++ GmbH 2018). GnssLogger App and Geo++ RINEX Logger are not the only Apps; there are other Apps as well, such as rinexON app, GalileoPVT app, G-RitZ logger and GNSS/IMU Android Logger (Zangenehnejad and Gao, 2021). In this paper, our focus is on evaluating the performance of the GnssLogger App and Geo++ RINEX Logger.

While generating the carrier-phase and Doppler observations is relatively easy, the process of generating pseudorange observations can still be challenging. There have been concerns about the quality of GNSS observations generated by these apps, ranging from biases to observation inconsistency. These issues can affect GNSS data processing, cycle slip detection, code smoothing, and ultimately, positioning performance.

The paper is structured as follows. How to convert raw Android location-related measurements into typical GNSS observations,

* Corresponding author

including pseudorange, carrier-phase, and Doppler, is explained in the subsequent section. Following that, a detailed description of the employed mathematical model, specifically the uncombined precise point positioning (UPPP) model, is explained. In the numerical results section, an assessment is conducted to evaluate the quality of generated GNSS observations from various smart devices using different loggers. This section thoroughly investigates the inconsistency between pseudorange, carrier-phase, and Doppler measurements reported in Zangenehnejad et al. (2022). Additionally, the presence of carrier-phase observations without temporal changes and their possible causes are addressed. Lastly, the positioning performance of three RINEX files (generated by GnssLogger, Geo++ RINEX logger, and UofC CSV2RINEX convertor) is investigated using GNSS observations from the Xiaomi Mi8 in kinematic mode. The paper concludes by drawing final conclusions in the last section.

2. GNSS OBSERVATION GENERATION FROM RAW MEASUREMENT OF ANDROID SMARTPHONES

The pseudorange, carrier-phase, and Doppler observations, which are typical GNSS observations, must be generated using the raw measurements obtained from the Android smartphones. How to convert the raw measurements logged through the android.location API to the GNSS observations has been provided in details in the white paper published by the European Global Navigation Satellite Systems Agency, GSA (2018). Table 1 presents the generation formula of pseudorange, carrier-phase, and Doppler observations from the raw measurements of Android smartphones. In this table, t_{Rx} is the received time (measurement time) in nanosecond, t_{Tx} is the received GNSS satellite time at the measurement time in nanosecond reported in the CSV file (one of the variables in the GNSSMeasurement class) and c is the speed of light. *AccumulatedDeltaRangeMeters* is the accumulated delta range (ADR) since the last channel reset which is one of the variables from GNSSMeasurement class within the Android API package "location" and λ denotes the signal's wavelength. *PseudorangeRatemetersperSecond* is the pseudorange rate at the timestamp in m/s and can be found as one of the variables in GNSSMeasurement class. All of these variables can be found either in the GNSSMeasurement class or in the GNSSClock class. For more information about these parameters, the reader can refer to European GSA (2018), Zangenehnejad and Gao (2021) and Zangenehnejad et al., (2023).

Pseudorange (meter)	$P = (t_{Rx} - t_{Tx}) \times 10^{-9} \times c$
Carrier-phase (cycle)	$\varphi = AccumulatedDeltaRangeMeters/\lambda$
Doppler (Hz)	$doppler = -PseudorangeRatemetersperSecond/\lambda$

Table 1. GNSS observation generation from raw measurement of Android smartphones

The generation of carrier-phase and Doppler observations is relatively straightforward, requiring only the following parameters: *AccumulatedDeltaRangeMeters*, λ (wavelength), and *PseudorangeRateMetersPerSecond*. However, the process of generating pseudorange observations can still be challenging. Each logging app implements its own GNSS observation conversion algorithm with different parameter settings, thresholds, and float computing accuracies. These variations can impact the quality of the observations saved into the RINEX file. As part of this contribution, we have developed our in-house converter, UofC CSV2RINEX, written in C++. This converter enables the conversion of a CSV file into a RINEX file. The converter can be accessed at the following link https://github.com/FarzanehZangeneh/csv2rinex (Zangenehnejad et al., 2023).

3. UNCOMBINED PRECISE POINT POSITIONING (UPPP)

The PPP functional model used in this contribution is the dual-frequency uncombined PPP model (SF-PPP). The undifferenced GNSS code and carrier phase observations for the satellite s and the receiver r on frequency j are as follows (Teunissen, and Kleusberg 1998)

$$\begin{aligned} E(P^s_{r,j}) &= \rho^s_r + T^s_r + cdt_r - cdt^s + \gamma_j I^s_{r,1} + b_{r,j} - b^s_{,j} \\ E(\Phi^s_{r,j}) &= \rho^s_r + T^s_r + cdt_r - cdt^s - \gamma_j I^s_{r,1} + \lambda_j N^s_{r,j} + B_{r,j} - B^s_{,j} \end{aligned} \quad (1)$$

where E is the mathematical expectation operator, P_j and Φ_j denote the pseudorange and carrier phase observations on the frequency j in meters, ρ is the geometric range between satellite and receiver as a function of the satellite and the receiver coordinates, T is the tropospheric delay (m) which can be spilt into dry and wet parts, dt_r and dt^s are the receiver and satellite clock errors (s), respectively, $I^s_{r,1}$ is the first-order slant ionospheric delay on frequency L1 (m), $\gamma_j = f_1^2 / f_j^2$ is the frequency-dependent multiplier factor (in the case of L1 frequency $\gamma_j = 1$), fj is the corresponding frequency, λ_j is the corresponding carrier phase wavelength (m), $N^s_{r,j}$ denotes the integer carrier phase ambiguity term in cycle, $b_{r,j}$ and $B_{r,j}$ denote the frequency-dependent receiver pseudorange and carrier phase hardware delays (biases), respectively, and $b^s_{,j}$ and $B^s_{,j}$ are the frequency-dependent satellite pseudorange and carrier phase hardware delays (biases), respectively.

The precise satellite clock errors provided by International GNSS Service (IGS) are based on the ionosphere-free (IF) linear combination of code observations on L1 and L2 frequencies, i.e., P1 and P2, as follows (Kouba and Héroux, 2001):

$$cdt^{s,IF} = cdt^s + b^s_{IF(1,2)} \quad (2)$$

where $b^s_{IF(1,2)} = \alpha^{L1,L2}_{IF} b^s_1 + \beta^{L1,L2}_{IF} b^s_2$ is the satellite ionosphere-free code bias in which b^s_1 and b^s_2 are the satellite pseudorange hardware delays for P1 and P2, respectively. The coefficients $\alpha^{L1,L2}_{IF}$ and $\beta^{L1,L2}_{IF}$ are also of the form

$$\alpha_{IF}^{L1,L2} = f_1^2 / (f_1^2 - f_2^2) \quad and \quad \beta_{IF}^{L1,L2} = -f_2^2 / (f_1^2 - f_2^2) \quad (3)$$

The uncombined PPP model for L1 and L5 frequencies can then be rewritten as follows:

$$\begin{aligned}
&E(P_{r,1}^s) = \rho_r^s + T_r^s + (cdt_r + b_{r,1}) - cdt^{s,IF} + I_{r,1}^s + (b_{IF(1,2)}^s - b_{,1}^s) \\
&E(\Phi_{r,1}^s) = \rho_r^s + T_r^s + (cdt_r + b_{r,1}) - cdt^{s,IF} - I_{r,1}^s + [\lambda_1 N_{r,1}^s - b_{r,1} + B_{r,1} - B_{,1}^s + b_{IF(1,2)}^s] \\
&E(P_{r,3}^s) = \rho_r^s + T_r^s + (cdt_r + b_{r,1}) - cdt^{s,IF} + \gamma_3 I_{r,1}^s + (b_{IF(1,2)}^s - b_{,1}^s) \\
&E(\Phi_{r,3}^s) = \rho_r^s + T_r^s + (cdt_r + b_{r,1}) - cdt^{s,IF} - \gamma_3 I_{r,1}^s + [\lambda_3 N_{r,3}^s - b_{r,1} + B_{r,3} - B_{,3}^s + b_{IF(1,2)}^s]
\end{aligned} \quad (4)$$

By introducing

$$c\tilde{dt}_r = cdt_r + b_{r,1}, \qquad \lambda_1 \tilde{N}_{r,1}^s = \lambda_1 N_{r,1}^s + B_{r,1} - B_{,1}^s + b_{IF(1,2)}^s - b_{r,1}$$

and $\lambda_3 \tilde{N}_{r,3}^s = \lambda_3 N_{r,3}^s + B_{r,3} - B_{,3}^s + b_{IF(1,2)}^s - b_{r,1}$, we have

$$\begin{aligned}
&E(P_{r,1}^s - \frac{1}{\gamma_2 - 1} DCB_{1,2}^s) = \rho_r^s + T_r^s + c\tilde{dt}_r - cdt^{s,IF} + I_{r,1}^s \\
&E(\Phi_{r,1}^s) = \rho_r^s + T_r^s + c\tilde{dt}_r - cdt^{s,IF} - I_{r,1}^s + \lambda_1 \tilde{N}_{r,1}^s \\
&E(P_{r,3}^s - \frac{\gamma_2}{\gamma_2 - 1} DCB_{1,2}^s + \frac{1}{\gamma_2 - 1} DCB_{2,3}^s) = \rho_r^s + T_r^s + c\tilde{dt}_r - cdt^{s,IF} + \gamma_3 I_{r,1}^s - DCB_{1,3}^r \\
&E(\Phi_{r,3}^s) = \rho_r^s + T_r^s + c\tilde{dt}_r - cdt^{s,IF} - \gamma_3 I_{r,1}^s + \lambda_3 \tilde{N}_{r,3}^s
\end{aligned} \quad (5)$$

where $DCB_{1,3}^r = b_{r,1} - b_{r,3}$, $b_{IF(1,2)}^s - b_{,1}^s = \frac{1}{\gamma_2 - 1} DCB_{1,2}^s$,

$b_{IF(1,2)}^s - b_{,3}^s = \frac{\gamma_2}{\gamma_2 - 1} DCB_{1,2}^s - \frac{1}{\gamma_2 - 1} DCB_{2,3}^s$ with

$DCB_{1,2}^s = b_{,1}^s - b_{,2}^s$, $DCB_{2,3}^s = b_{,2}^s - b_{,3}^s$ which are the satellite differential code biases (DCB) available from the IGS. The unknowns here are the receiver position, the receiver clock error $c\tilde{dt}_r$, the real-valued carrier-phase ambiguity terms $\lambda_1 \tilde{N}_{r,1}^s$ and $\lambda_3 \tilde{N}_{r,3}^s$, the slant ionospheric delay $I_{r,1}^s$, the tropospheric delay and $DCB_{1,3}^r$. The ionospheric delay can be also modeled by the global ionospheric maps (GIM) or the empirical models, i.e., ionosphere-corrected.

4. RESULTS AND DISCUSSION

This section consists of three sections as follows:

- Smartphone GNSS observations quality assessment
- Analysis of GNSS observations from different logging Apps
    a. Inconsistency between pseudorange, carrier-phase and Doppler observations
    b. Carrier-phase observations with no change over time
- UPPP positioning accuracy analysis

4.1 Smartphone GNSS observations quality assessment

The quality of smartphone GNSS observations plays an important role in high-precision smartphone positioning. In this section, we first assess the characteristics of GNSS observations from several GNSS smartphones in terms of their C/N0 records and carrier-phase continuity. The C/N0 referred to the ratio of the carrier power and the noise power per unit bandwidth. It can be regarded as a powerful indicator of the GNSS signal strength in the sense that a larger C/N0 indicates a stronger signal while a lower C/N0 shows a weaker signal.

Observations of two GNSS smartphones namely Xiaomi Mi8 and Samsung S20 are used, both equipped with the Broadcom chipset. These devices are dual-frequency smartphones supporting L5/E5a frequencies for GPS and Galileo, respectively. The Xiaomi Mi8 and Samsung S20 were put on the top of the geodetic pillars with known coordinates on the rooftop of the Civil Engineering building, University of Calgary, Calgary, Canada. The dataset was collected on November 23th, 2022 with a sampling interval of 1 sec for about 1:30 hours.

Figure 1 provides the C/N0 measurements for GPS on the first frequency for two devices. The GPS C/N0 ranges from 7-46 and 17-46 dB-Hz for Xiaomi Mi8 and Samsung Ultra S20, respectively. Despite collecting data in the same environment and at the same time, the two smartphones reveals varying performance in terms of their C/N0 records and number of visible satellites. They also showed that C/N0 value of GPS raw observations of the smartphones is 10dB-Hz lower than the C/N0 values obtained from a geodetic-quality antenna and receiver. One can also see that the GNSS measurements of the smartphones have rapid changes/variations over such short time duration. A similar phenomenon has been reported in the study conducted by Li and Geng (2019).

In addition to the signal strength indicated by the C/N0 values, the continuity (availability) of the GNSS observations is also of importance. Figure 2 provides the carrier-phase continuity for GPS for the two devices. In this figure, the red dots denote the epochs in which the carrier-phase observations were missing while the code observations were still observed. The figure clearly demonstrate another aspect of the lower quality of GNSS smartphone observations compared to high-end geodetic receivers.

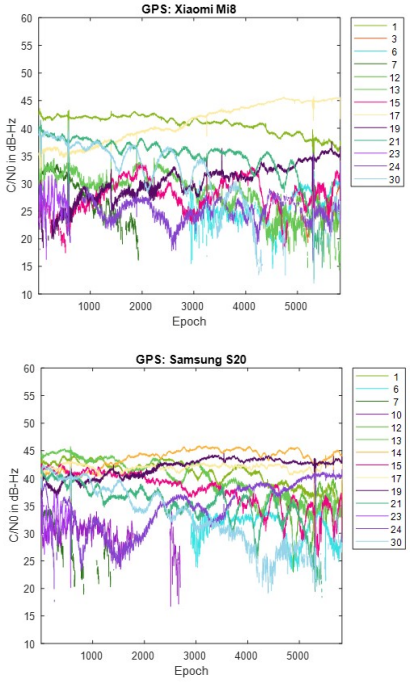

Figure 1. C/N0 measurements for GPS L1 signal for two devices

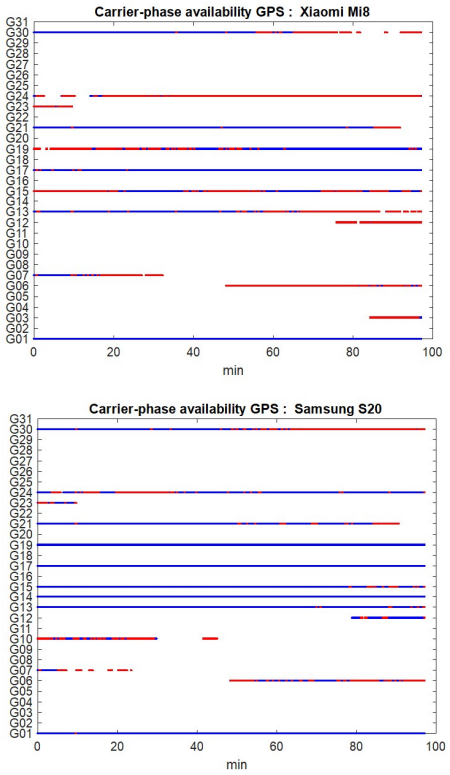

Figure 2. GPS carrier-phase continuity

4.2 Smartphone Analysis of GNSS observations from different logging Apps

This section is divided into two subsections. The first subsection highlights the inconsistency between the pseudorange, carrier-phase, and Doppler observations. The second subsection demonstrates the presence of carrier-phase observations with no changes over time.

4.2.1 Inconsistency between pseudorange, carrier-phase and Doppler observations: GnssLogger App and Geo++ RINEX App both are capable of providing GNSS observables in RINEX format. Different logging apps have different performance which affects the positioning results as well. In this section, we first investigate outputs of the three RINEX files, (1) RINEX file saved by GNSSLogger App, (2) RINEX file logged by Geo++ RINEX Logger App and (3) RINEX file generated by our convertor toolbox, UofC CSV2RINEX. The performance of the three RINEX files was investigated using the same dataset as in the previous section (Xiaomi Mi8 and Samsung S20 dataset). In the subsequent plots, GPS PRN 01 were chosen for presenting the results. This PRN was selected due to their better availability and continuity throughout the observation period (refer to Figure 2). It is worth noting that similar results were observed for other PRNs and other constellations as well. For better visualization, the first 900 epochs (15 minutes) were used to generate the figures. Table 2 also presents a concise summary of the experiment.

Devices	Xiaomi Mi8 and Samsung S20
PRN	PRN 01 (GPS)
Mode	Static
App logger	Geo++ RINEX logger (v2.1.6), GnssLogger (v3.0.5.6)
Date	23 November, 2022
Duration	~ 1 hr 30 min
Sampling interval	1 sec

Table 2. GNSS data information

To evaluate the quality of the smartphone GNSS observations from various RINEX files (RINEX from the Geo++ RINEX logger, GnssLogger and our convertor UofC2RNX), the following three quality indicators are used:

1- The first indicator involves calculating the first-order differentiation of GNSS pseudorange and carrier-phase observations by taking the difference between adjacent observations epochs and dividing by the sampling interval T. These differences are then compared to the Doppler observations. The first-order differences of pseudorange and carrier-phase observations are expected to follow the same trend as the Doppler observations.
2- The second indicator is the geometry-free combination (code-minus-phase or CMP), which cancels out the geometric part of the measurement, including the geometric range, receiver and satellite clock, and tropospheric delay, and leaves ambiguity, ionosphere term, multipath, and noise. This combination is useful for detecting the possible cycle-slips in the carrier-phase observations, as a cycle-slip appears as a jump in the CMP plot.
3- The third indicator involves obtaining the predicted carrier-phase using the discrete Doppler measurements. The carrier-phase predicted error is then calculated as the difference between the original and predicted carrier-phase observations ($\hat{\Phi}^{s}_{r,j}$).

Indicator	Formula
First-order differentiation of pseudorange and phase versus Doppler observations	$\begin{cases} \text{diff}(P^s_{r,j})/T \\ \text{diff}(\Phi^s_{r,j})/T \\ -\lambda_j D^s_{r,j} \end{cases}$
Geometry-free (Code minus phase: CMP)	$P^s_{r,j} - \Phi^s_{r,j}$
Carrier-phase predicted error	$\hat{\Phi}^s_{r,j} - \Phi^s_{r,j}$

Table 3. Different indictors used to analyze raw GNSS observations (Zangenehnejad et al., 2023)

Figure 3 shows the first-order differentiation of GPS pseudorange and carrier-phase observations as well as the Doppler observations for PRN 01 on the L1 frequency from the three RINEX files for the Xiaomi Mi8. To have a better view, the difference between Doppler observations and the first-order differentiation of the pseudorange (i.e., $-\lambda_j D^s_{r,j} - diff(P^s_{r,j})/T$) and the difference between Doppler observations and the first-order differentiation of the carrier-phase observations, (i.e., $-\lambda_j D^s_{r,j} - diff(\Phi^s_{r,j})/T$) are also depicted in the right panel of this figure. In some graphs, the red

line cannot be seen at this zoom setting since it is under the green one (Doppler).

A few observations can be highlighted from the figure. (1) The Doppler observations of the three RINEX files are the same. As mentioned, the Doppler shift is obtained as *doppler* = -*PseudorangeRatemetersperSecond*/λ (see Table 1) showing that generating the Doppler observation is straightforward and without any complications. (2) Shown in the top panel of Figure 3 is related to the GnssLogger RINEX file. As observed, there is an offset between the Doppler and the pseudorange observations. Such an offset can be caused during the pseudorange generation from the raw measurements in the Android API "location"-related classes. An offset is probably applied to the pseudorange observations. Applying such offset will not affect the solution as it can be lumped into the receiver clock bias and the real-valued ambiguities. (3) Shown in the middle panel of Figure 3 is related to the Geo++ RINEX logger output. Similar offset can be observed here not only for the pseudorange but also for the carrier-phase observations. The carrier-phase observations follow the pseudorange observations behavior in terms of the anomalies, spikes and jumps. This shows what happened to the pseudorange observations during their generation procedure happens to the carrier-phase observations as well. (4) Shown in the bottom panel of Figure 3 is related to the converted RINEX file from our developed convertor following the equations in GSA (2018). Unlike the other two RINEX files, there is no offset between the pseudorange and carrier-phase and Doppler observations and the first-order differences of pseudorange and carrier-phase observations follow the same trend of the Doppler observations.

The Doppler observations can be employed for cycle slip detection and/or code smoothing. Considering the possible biases and anomalies in the data, the Doppler observations must be carefully analyzed before use. The other two indicators, the CMP combination and the carrier-phase predicted error, are then utilized to further investigate the effect of possible biases and anomalies in the data. Figure 4 represents the CMP and the carrier-phase predicted error for GPS PRNs 01 from the three RINEX files for the Xiaomi Mi8 on the L1 frequency. The main purpose of this plot is to evaluate how the possible anomalies/jumps or offsets affects the CMP and the carrier-phase predicted error. The top, middle and bottom panels of Figure 4 include the CMP plot computed by using the GnssLogger RINEX, the Geo++ RINEX logger output and the converted RINEX, respectively. There are two important points about this figure that needed to be expressed. (1) First, let's look at the plot of the CMP values obtained from the GnssLogger RINEX (the top panel). The CMP does not include the geometric part while it includes the carrier-phase ambiguity, twice the ionospheric error, pseudorange and carrier-phase noise and multipath. Therefore, such behavior is not expected from the CMP values for the GnssLogger RINEX file, as it must be a constant value with a reasonable noise level as long as there is no cycle-slip in the data. It shows that the pseudorange and carrier-phase observations are not consistent (i.e., they are divergent, see the smaller panel in Figure 4 (top-left) in which the pseudorange and carrier-phase observations shifted to zero to have a better view). This indicates that the CMP combination cannot be employed to detect the possible cycle-slips in this case. We should note that this slop is the same for all PRNs. Therefore, in the case of using single-difference between satellites or double-difference observations, this issue will not affect the positioning results as it will be removed through the differencing procedure. (2) As can also be observed in this figure, the carrier-phase predicted error obtained from the Geo++ RINEX logger, depicted in the middle panel, does not have an expected behavior for any reason. This also indicates that the Doppler observations cannot be employed here to detect the possible cycle-slips in the data. It is not clear to us why these plots are like that, as these Apps have not disclosed their internal algorithms. The same plots for the Samsung S20 are given in Figures 5 and 6. They support the similar conclusion as before.

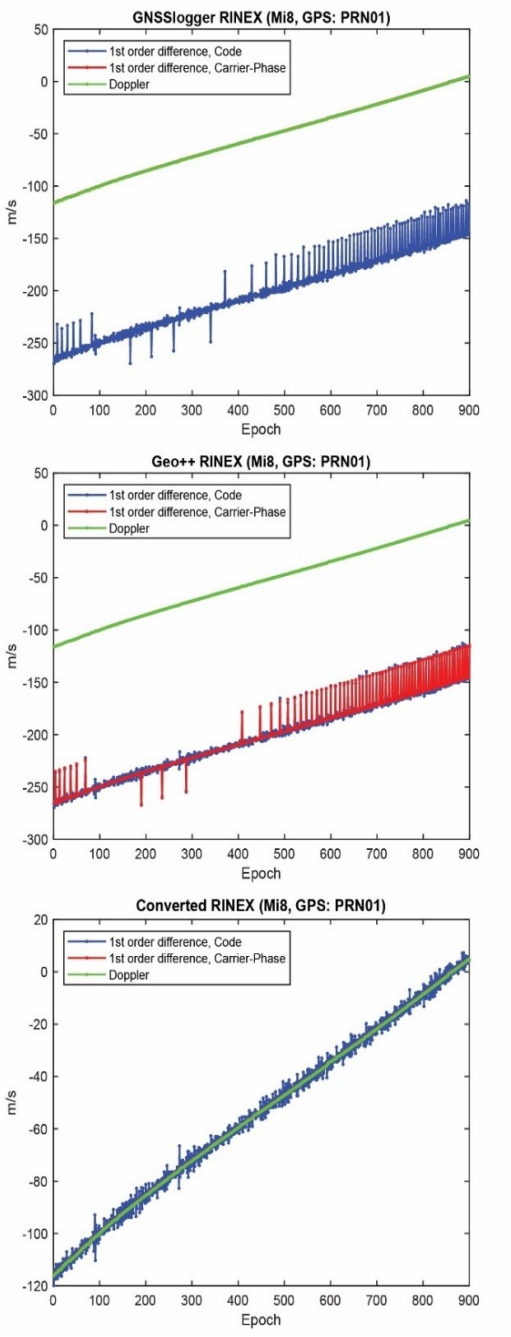

Figure 3. First-order differentiation of GPS pseudorange and carrier-phase observations as well as Doppler observations on L1 frequency for PRN 01 for Xiaomi Mi8 (Zangenehejad et al. 2023)

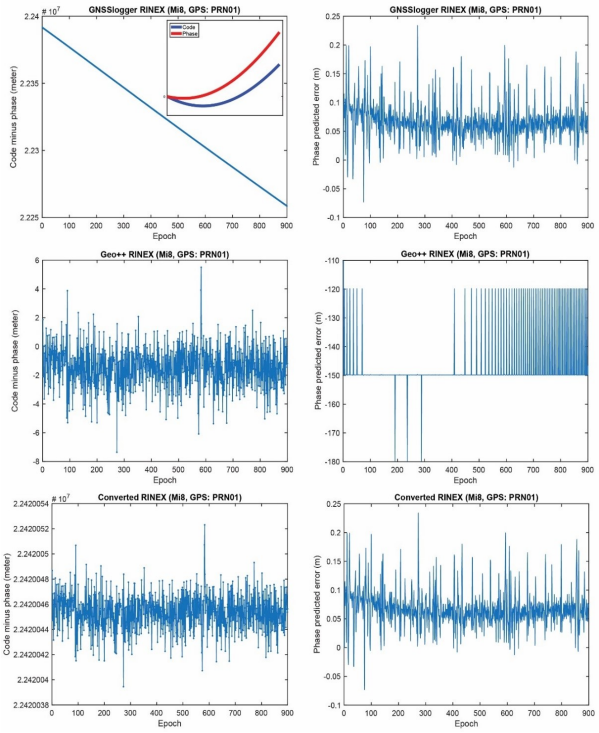

Figure 4. CMP and carrier-phase predicted error for GPS PRNs 01 for Xiaomi Mi8 (Zangenehnejad et al. 2023)

Finally, Table 4 provides a summary of consistency or inconsistency between different GNSS observations in the three RINEX files. The highlighted cells in this table indicate the CMP combination cannot be implemented in the cycle-slip detection procedure while using the RINEX by GNSSLogger App. In addition, the Doppler observations cannot be employed for the cycle-slip detection while using Geo++ RINEX logger output.

Combination	GnssLogger	Geo++ RINEX logger	UofC CSV2RINEX
Code & Phase	No (Attention required!)	Yes	Yes
Code & Doppler	No	No	Yes
Phase & Doppler	Yes	No (Attention required!)	Yes

Table 4 Consistency between different GNSS observations in three RINEX files

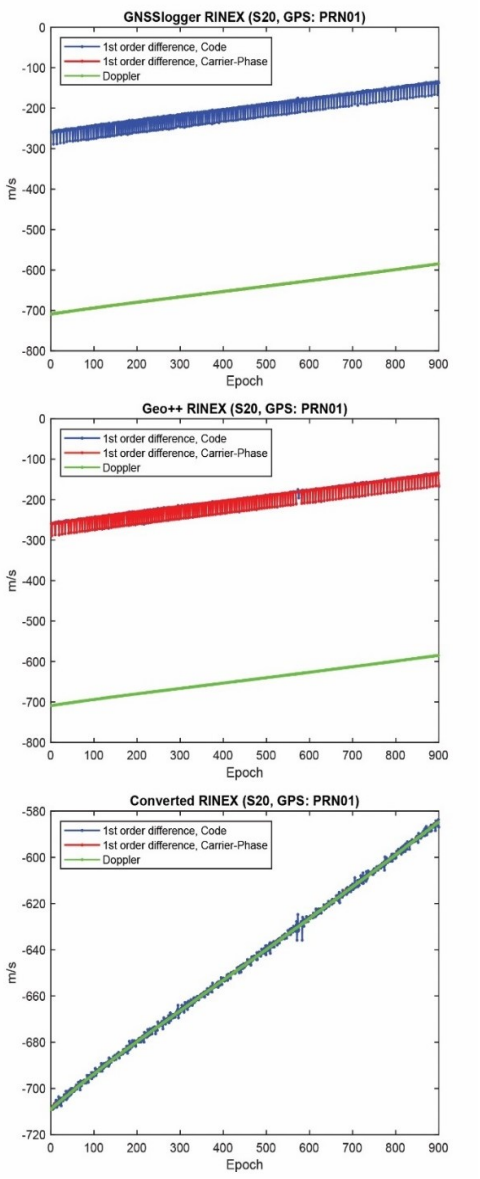

Figure 5. First-order differentiation of GPS pseudorange and carrier-phase observations as well as Doppler observations on L1 frequency for PRN 01 for Samsung S20 (Zangenehnejad et al. 2023)

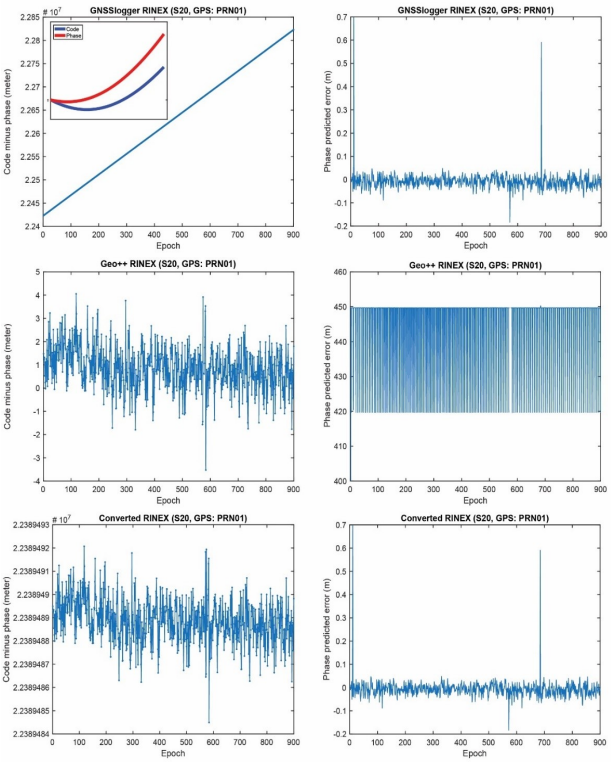

Figure 6. CMP carrier-phase predicted error for GPS PRNs 01 for Samsung S20 (Zangenehnejad et al. 2023)

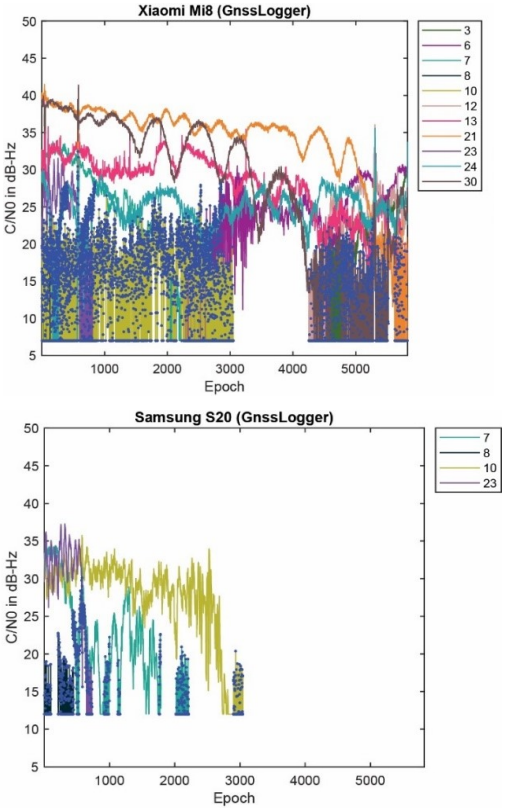

Figure 7. C/N0 records of GPS PRNs with problem in carrier-phase observations logs (Blue dotes: epochs of carrier-phase observations with no change over time)

4.1.2. Carrier-phase observations with no change over time: Another problem is also observed in the RINEX files saved by the GnssLogger App which is the existence of some carrier-phase observations with no changes over time. Such phenomenon was observed for both Xiaomi Mi8 and Samsung S20. Figure 7 provides the C/N0 records for the mentioned GPS PRNs along with the epochs in which the carrier-phase observations have no changes over time for the Xiaomi Mi8 and Samsung S20, depicted in the left and right panels, respectively. They are shown with the blue dots. The number of such satellites is more for the Xiaomi Mi8 compared to the Samsung S20. Those PRNs mostly belong to the lower C/N0 values. A C/N0 mask is usually set to 15-25 dB-Hz, however, there are still some of those epochs with the C/N0 larger than the threshold. It therefore needs further attention than just masking the lower C/N0. As given in Table 1, the carrier-phase observation can be obtained from *AccumulatedDeltaRangeMeters* variable (abbreviated as ADR) from the GNSSMeasurement class. Checking validity of the carrier measurements by using the *AccumulatedDeltaRangeState* variable is crucial. By looking at the ADR logs from the Xiaomi Mi8 and Samsung S20, it is revealed that these questionable epochs are mainly related to the invalid ADR. However, it was not observed in the Geo++ RINEX file meaning that it properly handled the invalid ADR by excluding them. In the converted RINEX, such an issue was not seen as well.

Concisely, the results confirmed the importance of evaluating the logging Apps before employing them since these logging Apps are the basics of any smartphone positioning algorithm development. In the next subsection, the UPPP positioning accuracy obtained from the three RINEX files are assessed in the post-processed mode within a kinematic experiment.

4.3 UPPP positioning accuracy analysis

In this section, we present the results of a kinematic test conducted on April 22nd, 2022, and lasted for approximately one hour. The test was conducted in a mostly open-sky environment with overpasses, specifically in Calgary, Alberta, Canada (Figure 8).

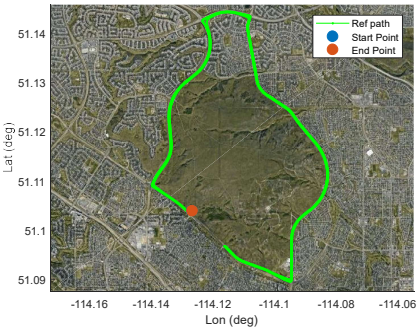

Figure 8. Kinematic experiment done on April 22, 2022

The kinematic experiment involved three geodetic receivers (two U-blox F9P and one Septentrio AsteRx-m2). The phone were placed on the vehicle roof. The offsets between all units were measured and applied prior to comparison. The reference trajectory of the vehicle during the kinematic experiment was obtained by the RTK fixed solutions from the three geodetic receivers as the rover receivers. A geodetic receiver on a geodetic pillar (with true position) on the rooftop of the Civil Engineering building, University of Calgary, was also selected as the base receiver. Figure 9 provides the positioning errors for Xiaomi Mi8 using the two RINEX files (RINEX by GEO++ RINEX logger and our converted RINEX) in the post-processed mode. The results of the RINEX file from the GnssLogger App was not provided since the obtained accuracy was at single point positioning (SPP) accuracy level due to the frequent cycle slip detected. It should be noted that the RMS values provided in this figure have been computed using all epochs. The results confirm the better performance of the converted RINEX in terms of East, North and Up RMS values.

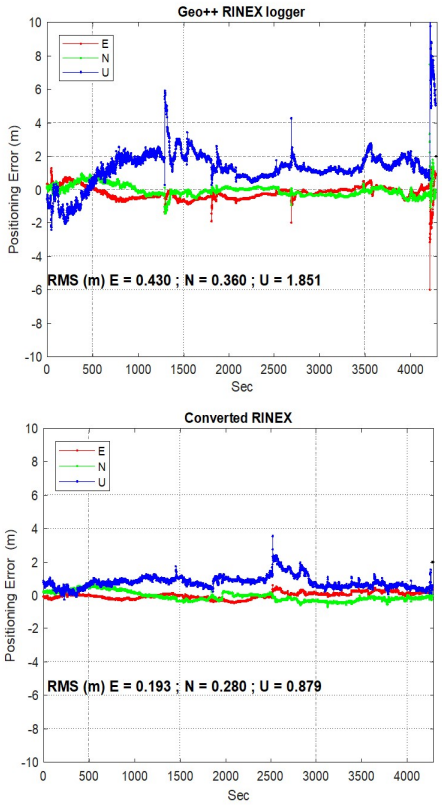

Figure 9. Horizontal positioning errors (Zangenehnejad et al. 2023)

Finally, it is important to note that while there are several open-source apps available for generating typical GNSS observations from the Android location API and saving them in the RINEX format, we must remain careful regarding the generation of GNSS observations. As demonstrated, there are potential issues that can arise in the generated observations, emphasizing the need for careful examination and quality assessment.

5. SUMMARY AND CONCLUSION

Two popular apps, GnssLogger and Geo++ RINEX logger, were evaluated along with a new converter called UofC2RNX. Quality indicators were employed to assess the consistency and accuracy of the observations, including comparing first-order differentiations of pseudorange and carrier-phase with Doppler observations, using the geometry-free combination to detect cycle-slips, and calculating the predicted carrier-phase error. The conclusions drawn from our study are as follows:

(1) Inconsistency was observed between the generated pseudorange, carrier-phase, and Doppler observations in the RINEX outputs of the GnssLogger and Geo++ RINEX Logger apps.

(2) Some carrier-phase observations showed no changes over time in the RINEX files saved by the GnssLogger App. These observations were mostly associated with lower C/N0 values and invalid ADR states.

(3) The UofC CSV2RINEX output demonstrated improved positioning accuracy compared to both the Geo++ RINEX Logger and GnssLogger outputs.

It is recommended to collaborate with the developers of these apps in future endeavors to understand and assess the models and algorithms employed for generating GNSS observations.

REFERENCES

European Global Navigation Satellite Systems Agency, GSA 2018. Using GNSS Raw Measurements on Android Devices-Towards Better Location Performance in Mass Market Applications (white paper). Luxembourg: Publications Office of the European Union, Available at https://www.gsa.europa.eu/system/files/reports/gnss_raw_measurement_web_0.pdf

Geo++ GmbH, 2018. Logging of GNSS Raw Data on Android. Geo++, Available from: http://www.geopp.de/logging-of-gnss-raw-data-on-android.

Kouba, J., Héroux, P., 2001. Precise Point Positioning using IGS Orbit and Clock Products. *GPS Solutions*, 5(2), pp. 12-28

Li, G., Geng, J., 2019. Characteristics of raw multi-GNSS measurement error from Google Android smart devices. *GPS Solutions*, 23(3), pp. 1-16.

Teunissen, P.J.G., Kleusberg, A., 1998. *GPS for Geodesy* (2nd ed.), Berlin-Heidelberg-New York: Springer-Verlag.

Zangenehnejad, F., Gao, Y., 2021. GNSS Smartphones Positioning: Advances, Challenges, Opportunities, and Future Perspectives. *Satellite Navigation*, 2.1 (2021), pp. 1-23.

Zangenehnejad, F., Jiang, Y., Gao, Y., 2022. Improving Smartphone PPP and RTK Performance Using Time-Differenced Carrier Phase Observations. In Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022) (pp. 2287-2300).

Zangenehnejad, F., Jiang, Y., Gao, Y., 2023. GNSS Observation Generation from Smartphone Android Location

API: Performance of Existing Apps, Issues and Improvement. *Sensors*, 23(2), 777.

Zumberge, J.F., Heflin, M.B., Jefferson, D.C., Watkins, M.M. (1997). Precise point positioning for the efficient and robust analysis of GPS data from large networks., *Journal of Geophysical Research,* 102(B3), pp. 5005-5018.