Analyzing Android GNSS Raw Measurements Flags Detection Mechanisms for Collaborative Positioning in Urban Environment

Thomas VERHEYDE
TéSA Research Laboratory
Toulouse, FRANCE
thomas.verheyde@recherche.enac.fr

Antoine BLAIS
ENAC Research Laboratory
Toulouse, FRANCE
antoine.blais@enac.fr

Christophe MACABIAU
ENAC Research Laboratory
Toulouse, FRANCE
christophe.macabiau@enac.fr

François-Xavier MARMET
Centre National d'Études Spatiales
Toulouse, FRANCE
francois-xavier.marmet@cnes.fr

***Abstract*—The release of Android GNSS raw measurements, in late 2016, unlocked the access of smartphones' technologies for advanced positioning applications. Recently, smartphones' GNSS capabilities were optimized with the release of multi-constellation and multi-frequency GNSS chipsets. In the last few years, several papers studied the use of Android raw data measurements for developing advanced positioning techniques such as Precise Point Positioning (PPP) or Real-Time Kinematic (RTK), and quantified those measurements compare to high-end commercial receivers. However, characterizing different smartphone models and chipset manufacturers in urban environment remains an unaddressed challenge. In this paper, a thorough data analysis will be conducted based on a data collection campaign that took place in Toulouse city center. Collaborative scenarios have been put in place while navigating in deep urban canyons. Two vehicles were used for this experiment protocol, equipped with high-end GNSS receivers for reference purposes, while seven smartphones were tested. Android algorithms reliability of both the multipath and cycle slip flags were investigated and evaluated as potential performance parameters. Our study suggests that their processing may differ from one brand to another, making their use as truthful quality indicators for collaborative positioning yet open to debate.**

***Index Terms*—Android Raw Measurements, Cycle Slip Flag, Multipath Flag, Collaborative Positioning**

I. Introduction

In May 2016, Google announced that GNSS raw data measurements will be available on Android smartphone devices via their latest Android Application Programming Interface (API) called Android Nougat (7.0). This innovation allowed developers and the scientific community to obtain access to GNSS measurements from embedded smartphones receiver. Code, phase, Doppler and C/N0 data can now be retrieved from Android's mass market receivers. Following this milestone, mobile chipset manufacturers started to develop innovative technology, including the newly announced Broadcom *BCM 47765* dual-frequency, multi-constellation chipset [1].

Commercial opportunities rose from this technological achievement, that led mobile manufacturers to compete in order to obtain the world's most precise smartphone. Multiple phone companies joined the race, releasing dozens modern smartphones, equipped with various chipset models, that are multi-frequency and multi-constellation ready. Those technological progress potentially unlocked the access of a wide crowdsourced and connected network of embedded smartphone receivers.

In the last few years, several research studies explored the possible implementation of advanced GNSS processing techniques (e.g. PPP, RTK) on Android mass market device [2] [3]. Positioning performances were then compared to low-cost and high-end commercial receivers. Most of those works focused on one phone model in optimal conditions without characterizing other smartphones and chipset brands.

Android based positioning is most of the time performed in constrained environments around urban areas. The main challenge associated with positioning in an urban environment, is signal degradation caused by disruptive multipath and Non-Line of Sight (NLOS) signals reception. Apprehending those difficulties is even more challenging while using embedded smartphones' linearly polarized antennas. Antenna design architecture limitations make them unoptimized for acquiring multi-frequency GNSS signals.

On the other hand, the Android positioning API provides detection mechanisms in the form of *flags* in order to detect multipath and cycle slip occurrences. However, their detection algorithms are unknown and coping with those flags could become ambiguous.

To overcome those issues, a thorough study has been conducted during a data collection campaign in Toulouse city center. In the interest of developing a global smartphone qualification method, an analysis was made on seven smartphones in a constrained environment. An assessment of Android flags

978-1-7281-6455-7/20/$31.00

will be presented as an introduction to smartphone performance parameters. In future work, the identified performance parameters will be used and exchanged in a collaborative smartphone network. This shared data would help network's users to qualitatively and quantitatively assess their smartphone's performances.
This article will be articulated in three main sections. First, the data collection campaign will be presented. Then, a detailed investigation on multipath and cycle slip flag algorithms is conducted. Finally, this paper will be concluded by a discussion on how to integrate performance parameters in a collaborative smartphone network.

II. Data Collection Campaign

Our data collection campaign took place in August 2019, in Toulouse city center. The goal of this campaign was to accurately depict urban conditions encountered by Android users. A fleet of two vehicles was used along a specific trajectory as shown in figure 1. Collaborative scenarios were established along the way. This data collection campaign lasted for 2 hours and 10 minutes.

A. Experimentation Protocols

The two vehicles were both equipped with high-end GNSS Commercial-Off-The-Shelf (COTS) equipments, a NovAtel SPAN receiver coupled with a high-grade IMU unit, a Septentrio PolaRX5 and a Ublox F9P for reference purposes. Table I lists all smartphones analyzed during this data campaign. Each mobile was securely placed on their assigned car's rooftop for the entire duration of the data collection. All smartphones were multi-constellation (Galileo, GPS, GLONASS & Beidou compatible) and multi-frequency (L1/L5 & E1/E5a) except for the Google Pixel 3 that was single-frequency. The smartphone selection process was decided in function of their brand and model as shown in table I and were running Android Pie 9.0. GNSS Raw data measurements were recorded by each device in a *.log* format using the GNSSlogger application. Finally, it was noticed that the Google Pixel 3 and the Xiaomi Mi 9 did not record any phase measurements data for the entire campaign whereas the other Android phones recorded them correctly.

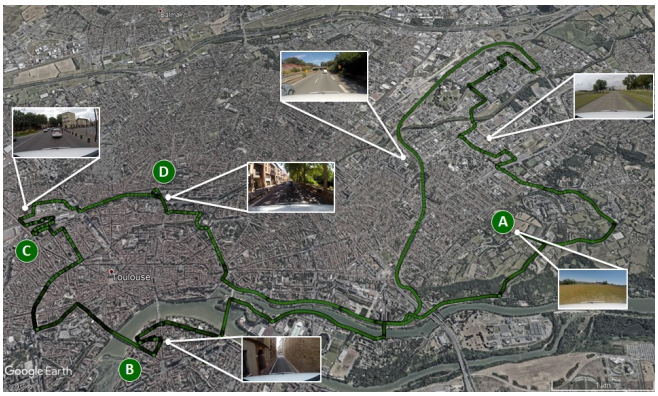

Fig. 1. Data Collection Campaign Vehicles' Trajectory.

TABLE I
Android Smartphones Analyzed

Car ID Number	Smartphones		
	Brand	***Model***	***Chipset***
1	Xiaomi	Mi 8	Broadcom BCM 47755
1	Xiaomi	Mi 9	Qualcomm Snapdragon 855
1	Google	Pixel 3	Qualcomm Snapdragon 845
1	Honor	View 20	HiSilicon Kirin 980
2	Huawei	Mate 20X	HiSilicon Kirin 980
2	Xiaomi	Mi 8	Broadcom BCM 47755
2	Honor	View 20	HiSilicon Kirin 980

B. Collaborative Scenarios

Throughout the collection campaign, collaborative scenarios were implemented. Figure 1 shows the different cooperative scenarios created. Each scenario is represented by a letter and a picture, taken by our on-board camera, illustrating the environment condition. The first one, labeled **A** in figure 1, represents a test case in nominal conditions (open-sky). Scenario **B** illustrates Android-based positioning in a deep urban environment. The third test case (**C**) defines a collaborative event between two users with one being in good reception condition and the other, on the contrary, positioned in constrained environment. This scenario has been achieved by setting one car on the last level (in open-sky conditions) of a five stories parking garage while the second car was roaming around the streets of Toulouse. Finally, the last scenario named **D**, is a static case in urban condition around Canal du Midi. Every collaborative scenario lasted between fifteen and twenty minutes. Outside of those specific test cases, the two cars were strictly following each other throughout the data collection process.

III. Analyzing Android Multipath and Cycle Slip Flag Algorithms

Smartphones' embedded GNSS receiver architecture is mostly similar to COTS GNSS receivers, from capturing the signal to estimating its position. Android allowed their users to have access to raw data measurements outputted by the baseband signal processing unit of the chipset receiver. Raw data measurements range from the most basic parameters (i.e code, phase, Doppler, C/N0) to more complex ones (i.e Automatic Gain Control (AGC), signals states and indicators). A GNSS raw measurement task force group, created by the European GNSS Agency (GSA), wrote a white paper [4] explaining in details Android's location data service. Among these complex measurements we find the '*Multipath Indicator*' and the '*AccumulatedDeltaRangeState*' parameters. Few information are released by Android and/or by chipset manufacturers concerning their computation algorithms. In a time where smartphone GNSS receiver's technology rapidly advances, it became crucial to understand and evaluate those flags reliability in order to better characterize smartphones' positioning performances.

A. Android Flags Detection Process

Android raw data measurements are obtained through the use of the '*Android.location*' API [5]. Within this API, a public class called *GnssMeasurement* contains GNSS data supposedly coming directly from the embedded chipset. This class is divided into two data groups. The first one, called '*Public methods*', regroups all GNSS raw data measurements. The second one, named '*Constant*', gathers information about received signals characteristics. Within this second group, we find a '*Multipath Indicator*' and an '*AccumulatedDeltaRangeState*' that provide multipath and cycle slip flags detection mechanism to the user.

1) Multipath Indicator: The Android multipath indicator state flag can take three different values. If the flag takes the value of **1**, a multipath interference has been detected for that measurement. On the other hand, when the indicator is set to the value **2** it signifies that multipath was not detected. Moreover, the indicator can also take the value of **0**, meaning that the presence or absence of multipath is unknown.
In our study, the multipath detection mechanism is simply activated when the indicator shows a value of **1**. It has to be noted that only Honor View 20 smartphones reported signals being unaffected by multipath (i.e *Multipath Indicator* $= 2$).

2) Accumulated Delta Range State: Android phase measurement characterization is based on the combination value of six state indicators. Each indicator corresponds to a constant value, and the overall addition of those states is prompted to the user by the '*AccumulatedDeltaRangeState*' parameter. Those states are listed below:

- *ADR_STATE_CYCLE_SLIP*: value = 4
- *ADR_STATE_HALF_CYCLE_REPORTED*: value = 16
- *ADR_STATE_HALF_CYCLE_RESOLVED*: value = 8
- *ADR_STATE_RESET*: value = 2
- *ADR_STATE_UNKNOWN*: value = 0
- *ADR_STATE_VALID*: value = 1

Processing cycle slip flag detection is done by identifying *ADR_STATE_CYCLE_SLIP* and *ADR_STATE_RESET* constants presence in the final *AccumulatedDeltaRangeState* value. Thus, to detect Android cycle slip, we set values that '*AccumulatedDeltaRangeState*' could take ($Valid\_State = [1, 8, 16, 9, 17, 24, 25]$). If the current '*AccumulatedDeltaRangeState*' indicator value falls out of our selection we then flag our current measurement to be impacted by a cycle slip.
Even though multipath and cycle slip detection mechanisms are provided by the Android API, no information is yet to be found about how the chipset computation process is made and how they are transferred to the Android *GnssMeasurement* class. We will now show the flags repeatability and efficiency in an urban environment.

B. Preliminary Measurements Analysis in an Urban Canyon

Android based positioning in urban conditions was expected to be difficult due to possible signal degradations and the use of an inefficient smartphone antenna. Nevertheless, it was seen

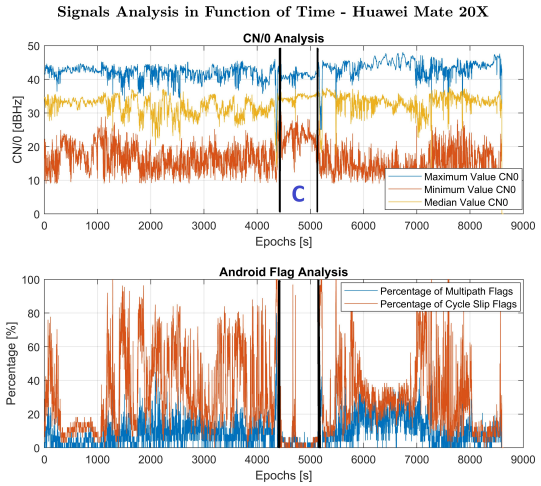

Fig. 2. Signal Analysis for Huawei Mate 20X - Car ID n°2

that each smartphone tracked more than 30 signals (considering all frequencies and all constellations simultaneously) per epoch during our entire data collection. All our tested chipset brands (Broadcom, Qualcomm and Kirin) achieved the same tracking performance. Although, it has been noted that both Honor View 20s under-performed compared to others units.
Due to the rapid evolution of the user-to-satellite propagation channel, we observed fast varying C/N0 values. The top graph of Figure 2 illustrates those C/N0 fast fluctuations observed over time. For each smartphone, the minimum, median and maximum C/N0 value has been computed in function of each individual received signal (considering all frequencies and all constellations simultaneously) for every epoch. For the Huawei Mate 20X, the median value of C/N0 range between 30 and 35 dBHz. The bottom graph of figure 2 shows the percentage of signals where a multipath and/or a cycle slip detection has been recorded in function of time. The percentage computation was obtained by dividing the number of flags detected by the total number of received signals for that specific epoch. The first observation made here is that cycle slip seems to be often detected by the receiver whereas multipath detections remain less frequent.
The mark, labeled **C**, on figure 2 highlights the third collaborative scenario. During this time, the second car was parked on the last floor of a parking garage in open-sky reception condition. C/N0 values of all signals improved, while both cycle slip and multipath flag detection decreased as expected. On the other hand, uncorrelated situations between C/N0 and flags detection have been observed during multiple occasions. This situation can be observed on figure 2 between epoch 400 and 1000, where median signal strength remains constant during that time period whereas flags activation numbers suddenly decrease.
Similar analysis has been performed for all tested smartphones. As stated before, the Google Pixel 3 and the Xiaomi Mi 9 did not record any phase measurements data. It is then

safe to state that cycle slip detection is not possible for those devices. Moreover, no multipath flags have been raised during our data campaign by either phone. This evidence suggests that the '*Multipath Indicator*' algorithm is not a naive linear correlation of C/N0 variation but exploits the phase measurement to detect multipath.
Independently of those phones, it appeared that both Xiaomi Mi 8 and the Huawei Mate 20X exhibit similar behaviors. However, both Honor View 20 (equipped with the same chipset as the Huawei Mate 20X) generated fewer cycle slip flags.

C. Correlating Flags Detection Mechanisms

In order to confirm the previously stated hypotheses, a detailed analysis of multipath and cycle slip flags has been conducted. Multiple basic GNSS measurements have been tested through a series of correlation events. While processing the preliminary analysis of our data samples, we stated that multipath and cycle flag detection algorithms were not solely linearly correlated to C/N0. To validate this hypothesis, flags distributions in function of C/N0 and elevation were analyzed. Figure 3 represents the cycle slip flag detection distribution in function of C/N0. Histograms and cumulative density functions (cdf) are drawn here.
Cycle slip detection distribution seems to be quite uniformly distributed over C/N0 values. Even though our tested smartphones are not equipped with the same chipset component, they tend to have similar detection behaviors (increased detection activity between 13 and 16 dBHz, before peaking around the C/N0 value of 22 dBHz). However, Honor View 20s did not detected as many cycle slip flags (i.e section III-B) as other devices and their distribution are surprisingly shifted toward high C/N0 value (around 35dBHz).
Multipath flag detection distribution in function of C/N0 have similar characteristics has the one observed in the case of cycle slip flag distribution. Clear peaks are located at 16 dBHz, 25 dBHz and 33 dBHz. While the peak around 16dBHz most likely corresponds to low-elevation satellites, the two other peaks can not be fully explained by physical phenomenon in

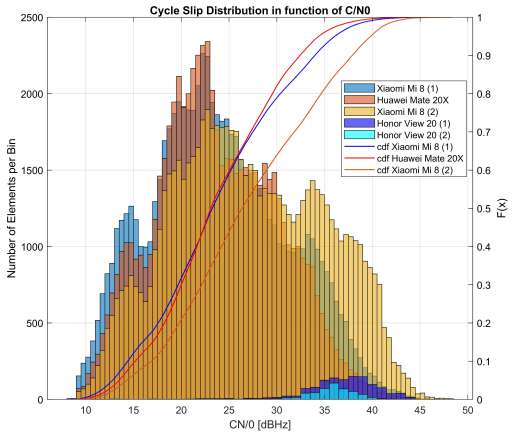

Fig. 3. Cycle Slip Flags Detection Distribution in Function of C/N0

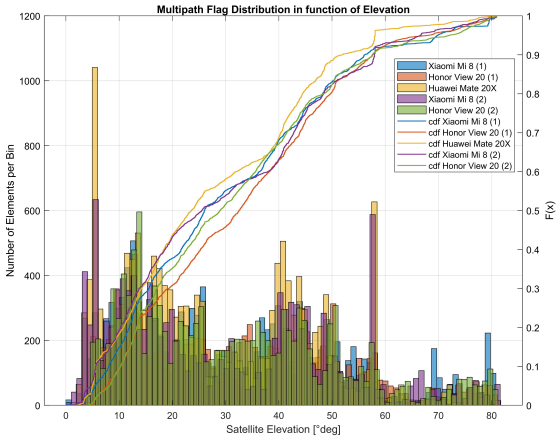

Fig. 4. Multipath Flag Distribution in Function of Elevation

the propagation channel. Once again, the distribution behaviors of multipath flags are similar from phone to phone (including Honor View 20s). A simple interpolation of C/N0 does not describe the Android multipath detection mechanism.
The distribution of multipath and cycle slip flags have also been studied in function of satellite elevation. Figure 4 shows the distribution of multipath flag detection in function of satellite elevation. From this graph, it is clear that no direct correlation between multipath detection technique and satellite elevation can be established. Moreover, smartphones tend to follow a similar distribution trend that could indicate that the multipath detection algorithm might not be directly outputted by the chipset itself.
Overall, Android flag detection systems are not naively only interpolated from C/N0 and satellite elevation value of the current signals. Detection mechanisms might be as complex as the one found in modern COTS GNSS receivers. All smartphone brands and models shown similar distribution patterns making us believe that those estimation algorithms may be using the same detection techniques at the chipset level and/or that these flags might be computed at a common low-level Android layer.

D. Android Detection Algorithm Efficiency

The lack of information regarding Android multipath and cycle slip detection mechanisms breed doubts about flags integrity and efficiency. We demonstrated that implemented detection algorithms are not based on a simple linear combination of basics GNSS parameters. Here, we studied flag detection mechanisms in function of the signal frequency and constellation. Signals captured by smartphones can be of different natures and some of them should be more robust to signal degradation. The following tables show the percentage of multipath and cycle slip flag activations over the number of received signals in function of signal frequency and constellation. For this study case, we will focus on comparing the L1/E1 to the L5/E5 frequency and thus on a comparison

TABLE II
FLAGS DETECTION PERCENTAGE IN FUNCTION OF FREQUENCY

Smartphones	L1 Signals		L5 Signals	
	Multipath	*Cycle Slip*	*Multipath*	*Cycle Slip*
Mi 8(1)	6.66	34.5	5.59	14.6
Mi 9	0	0	0	0
Pixel 3	0	0	0	0
View 20(1)	11.9	0.45	0	0.02
Mate 20X	8.62	31.3	6.93	15.2
Mi 8(2)	6.35	31.3	5.73	20.6
View 20(2)	11.6	0.88	0	0.01

TABLE III
FLAGS DETECTION PERCENTAGE IN FUNCTION OF CONSTELLATION

Smartphones	GPS L1		Galileo E1	
	Multipath	*Cycle Slip*	*Multipath*	*Cycle Slip*
Mi 8(1)	6.83	29.8	5.55	23.5
Mi 9	0	0	0	0
Pixel 3	0	0	0	0
View 20(1)	13.3	0.72	1.82	0.01
Mate 20X	10.9	29.6	10.53	28.30
Mi 8(2)	7.81	27.6	3.52	33.74
View 20(2)	14.4	1.60	1.61	0.01

between GPS and Galileo. Table II depicts the flag detection percentage in function of both frequencies for every multi-constellation tested smartphone. L5 received signals are known to be more robust to multipath error. Their larger bandwidth narrows the autocorrelation peak, thus reducing multipath impact. However, results in table II show that the percentage of multipath flag detection on L1 and L5 is similar for both Xiaomi Mi 8s and the Huawei Mate 20X. On the other hand, Honor' smartphones did not flag any multipath event for L5 received signals (over 46,000 L5 signals samples were collected within 2.5 hours of data collection considering all constellation received signals).

Cycle slip detection represents more than 30 percent of incoming signals. This statistic is shared by once again both Xiaomi Mi 8s and the Huawei phones. Thereafter, for L5 triggered cycle slip flag, the activation percentage is reduced by half.

Table III represents the percentage of flag detection for GPS L1 and Galileo E1. As a reminder, the CBOC signal modulation of Galileo E1 signals makes it more robust to multipath degradations. Both Honor View 20s exhibit a significant decrease of their multipath flag detection for Galileo E1, as expected. Nevertheless, other smartphones showed minor changes between both types of signals in terms of multipath detection percentage. Cycle slip detection seems to not be affected by neither the frequency nor the signal modulation type. Moreover, it has been noted that none of the smartphones detected cycle slips for Galileo E5 received signals. More than 23,000 E5 samples were collected that day for all devices. In order to get a sense on those indicators efficiency, the Code-Minus-Carrier (CMC) of the two highest satellites for each smartphone has been computed. Figure 5 displays satellite reception conditions (C/N0 and elevation) for Galileo PRN 12 and GPS PRN 27 on both frequencies for the analyzed Android device in dense urban canyons. Both satellites have been selected because they were at the highest elevation angle and visible during an extensive part of our data collection campaign. Similar results were obtained by Rokobun [6] in static and open-sky condition.

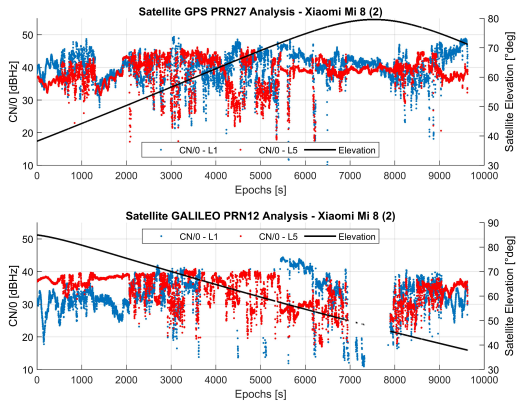

Fig. 5. Highest Galileo and GPS Satellite Analysis for Xiaomi Mi 8 (2)

The CMCs are then computed to visualize potential large cycle slips and multipath degradation. Pseudorange and phase measurements model are described by equation (1). Equation (2) models the difference between the code and phase measurements for a satellite SV at epoch i. Both ϵ_{Phase} and $\epsilon^{\phi}_{Multipath}$ terms have been neglected since $\epsilon_{Code} >> \epsilon_{Phase}$ and $\epsilon^{\rho}_{Multipath} >> \epsilon^{\phi}_{Multipath}$.

$$\rho_i^{SV} = r + c(t_{rx} - t_{tx}) + \epsilon_{Iono} + \epsilon_{Tropo} + \epsilon_{code} + \epsilon^{\rho}_{Multipath} \quad (1)$$

$$\phi_i^{SV} = r + c(t_{rx} - t_{tx}) - \epsilon_{Iono} + \epsilon_{Tropo} + N\lambda + \epsilon_{Phase} + \epsilon^{\phi}_{Multipath}$$

$$CMC_i = \rho_i^{SV} - \phi_i^{SV} = 2\epsilon_{Iono} - N\lambda + \epsilon_{code} + \epsilon^{\rho}_{Multipath} \quad (2)$$

$$CMC_i(detrended) = \epsilon_{code} + \epsilon^{\rho}_{Multipath} \quad (3)$$

where:

- r = User-satellite range
- $c(t_{rx} - t_{tx})$ = Receiver minus satellite clock offset
- ϵ_{Iono} = Ionospheric error
- ϵ_{Tropo} = Tropospheric error
- ϵ_{Code} = Delay Lock Loop (DLL) Jitter
- ϵ_{Phase} = Phase Lock Loop (PLL) Jitter
- $N\lambda$ = Ambiguity integer number
- $\epsilon_{Multipath}$ = Multipath error

Equation (3) presents the detrended CMC model where the influence of the ionospheric error and the ambiguity terms have been removed. The ambiguity term $N\lambda$ has been fixed by computing a sliding CMC mean for continuous observation segments and rounding to the nearest integer. Thus, satellite

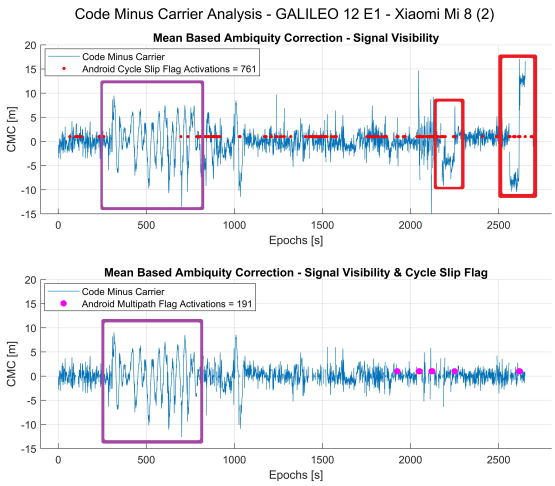

Fig. 6. CMC Analysis of Multipath and Cycle Slip Occurrences

continuous tracking segments being shorts, our mean ambiguity fixing computation also corrects for the ionospheric term (ϵ_{Iono}) since ionospheric error is a slow varying component. The remaining parameters of equation (3) are then multipath errors plus white Gaussian error noise.

Figure 6 exhibits CMC computations for Xiaomi Mi 8, mounted on the second car (car ID 2, c.f Table I). On this graph, the top plot represents the CMC evolution in time, while applying the sliding mean fixing method on segments where the satellite was physically visible by the receiver. This implies that cycle slip should still be visible on that plot (e.g. red boxes on figure 6), and red dots show where a cycle slip flag activation has been made by Android. Thereafter, the bottom plot illustrates the computed CMC values still corrected by the sliding mean fixing method per segments. However this time, segments were said to be continuous if the satellite was physically visible *and* if the Android flag algorithm did not detect any cycle slip. In this case, visible cycle slips remaining on the figure would mean that Android failed to correctly detect cycle slips. Theoretically at this stage, cycle slips should have been removed, leaving multipath and noise characteristic behaviors on our CMC plot. Purple dots illustrate Android multipath flag detection.

The presented graph was computed for the Xiaomi Mi 8 on Galileo PRN 12 - E1 signals. On the top figure 6, cycle slip flags have been activated 761 times over 8350 seconds. The overall flag activation seems to be over proportionate and too strict to detect real occurrences. However, the few cycle slips that happened during our data collection seem to have been successfully detected by Android. Cycle slips occurrences, shown by red boxes, can be identified on figure 6 top plot whereas they do not appear on the bottom graph. Figure 6 also highlights the use of multipath flags. Firstly, multipath flag algorithm only detected 191 occurrences. This number is supposedly underestimating the reality of our deep urban environment data collection. Moreover, a significant multipath event is visible on both top and bottom graph.

A typical multipath oscillation can be seen (depicted by purple boxes) and not being detected at any moment by the Android algorithm. This phenomenon was often seen for other satellites and other smartphones, implying that the multipath indicator is not triggered by a simple threshold on CMC. It can be stated that the multipath Android detection method is not efficient, whereas cycle slip detection is more robust.

IV. Conclusions

This paper presented Android multipath and cycle slip detection mechanisms. Their computation processes seem to not be exclusively based on a naive interpolation of C/N0 or satellite elevation parameters. Moreover, the similarities observed between smartphone brands make us believe that detection algorithms might be computed at a low-level Android layer. Multipath flags tend to be inconsistent whereas cycle slip flags were proven to be coherent despite their high false alarm activation frequency. Previous studies conducted in static scenarios [7] [8] concluded that collaborative smartphone positioning could be achieved and that GNSS measurements can be modeled. However, Android multipath and cycle slip indicators might not be used as reference parameters to qualitatively assess smartphone positioning performance. Therefore, future efforts will be put into characterizing qualitatively smartphone measurements in order to establish a collaborative network using Android devices.

V. Acknowledgment

This study has been conducted as a part of a PhD research project on *Precise cooperative positioning of low-cost mobiles in an urban environment*. The authors would like to acknowledge Thalès Alénia Space and CNES for sponsoring this work and ENAC for providing access to research equipment.

References

[1] M.D. Castillo, "Broadcom Introduces Second Generation Dual-Frequency GNSS," *https://www.broadcom.com/blog/broadcom-introduces-second-generation-dual-frequency-gnss*, 2019. [Online]

[2] U. Robustelli, V. Baiocchi, and G. Pugliano, "Assessment of Dual Frequency GNSS Observations from a Xiaomi Mi 8 Android Smartphone and Positioning Performance Analysis," *Green Communications in Smart City*, 2019.

[3] B. Chen, G. Chengfa, L. Yongsheng, and S. Puyu, "Real-time Precise Point Positioning with a Xiaomi Mi 8 Android Smartphone," *Remote Sensors, Control, and Telemetry.*, 2019.

[4] GNSS Raw Measurements Task Force - GSA, "Using GNSS Raw Measurements on Android Device," *White Paper - European GNSS Agency (GSA)*, 2017.

[5] Google, "Guides to Raw GNSS Measurements - gnssmeasurement," *https://developer.android.com/reference/android/location/GnssMeasurement*, 2017. [Online]

[6] M. Garcia and X. Banqué, "Smartphone Dual-Frequency GNSS Data Preprocessing for High Accuracy Applications," *Presentation Android Raw Measurement Task Force*, 2019.

[7] V. Lehtola, S. Söderholm, M. Koivisto, and L. Montloin, "Exploring GNSS Crowdsourcing Feasibility: Combinations of Measurements for Modeling Smartphone and Higher-end GNSS Receiver Performance," *Sensors*, vol. 19, no. 13, pp. 1–17, 7 2019.

[8] T. Verheyde, A. Blais, C. Macabiau, and F.-X. Marmet, "Statictical Analysis of Android GNSS Raw Data Measurements in an Urban Environment for Smartphone Collaborative Positioning Methods," *Presentation at INC 2019*, 2019.