Time-Synchronized GNSS/IMU Data Logging from Android Smartphone and its Influence on the Positioning Accuracy

Himanshu Sharma, Mohamed Bochkati, Thomas Pany
Institute of Space Technology and Space Applications (ISTA), Space Systems Research Center (FZ-Space)
Universität der Bundeswehr München

BIOGRAPHY

Himanshu Sharma is a research associate at the Institute of Space Technology and Space Applications (ISTA). He received his Bachelors in Technology (B. Tech) degree in the field of Electronics and Communication from maharishi Dayanand University, Rohtak, India. He accomplished his Masters in Science (M.Sc.) in the field of Communications and Signal Processing from Technical University of Ilmenau, Thuringia, Germany. His research interests are precise positioning, Signal Processing and Image processing. Currently, He is involved in the project focused on RTK positioning using the mobile phone GNSS Raw Data

Bochkati Mohamed is a research associate at the Institute of Space Technology and Space Applications (ISTA) at the Universität der Bundeswehr, München. His research focuses on Deeply Coupled GNSS/INS integration and novel calibration techniques for MEMS-IMUs. Before he joined the University of the Bundeswehr, he was a research associate at the Institute of Geodesy (ife) at the Leibniz University Hannover, where his research activities include quantum inertial navigation systems and GNSS receiver clock modeling. He holds a Bachelor's and Master's degree in Geodesy and Geoinformation from Technical University Munich.

Pany Thomas is with the Universität der Bundeswehr München at Space Systems Research Center (FZ-Space) where he leads the satellite navigation unit LRT 9.2 of the Institute of Space Technology and Space Applications (ISTA). He teaches navigation focusing on GNSS, sensors fusion and aerospace applications. Within LRT 9.2 a good dozen of full-time researchers investigate GNSS system and signal design, GNSS transceivers and high-integrity multi-sensor navigation (inertial, LiDAR) and is also developing a modular UAV-based GNSS test bed. ISTA also develops the MuSNAT GNSS software receiver and recently focuses on smartphone positioning and GNSS/5G integration. He has a PhD from the Graz University of Technology (sub auspiciis) and worked in the GNSS industry for seven years. He authored around 200 publications including one monography and received five best presentation awards from the US Institute of Navigation. Thomas Pany also organizes the Munich Satellite Navigation Summit.

ABSTRACT

Since the advancement in Smartphone GNSS positioning and the availability of GNSS observation via Android API, much of the focus have been given to bring down the Smartphone GNSS positioning accuracy from decimeter level to the centimeter level. But, considering many limitations such as Smartphone GNSS antenna, multipath and cycle slips, bringing down the accuracy to centimeter level is not an easy task. Many research work in this domain has demonstrated that the use of embedded sensors (IMU) inside the Smartphone can definitely ease the burden on GNSS. Researchers have already successfully processed the GNSS/IMU data collected from the Smartphone and highlighted some interesting results. But, when dealing with the GNSS/IMU processing, time-synchronization between GNSS and IMU data plays a vital role. Specially, in devices like Smartphone, when the GNSS and IMU data are not driven by the common clock, it becomes more important to study and analyze the effect of time-synchronizing between these two data sets. This paper gives reader an in-depth explanation about the need of time-synchronizing and the complete implementation overview. This paper will start with section I, giving an introduction to the current state of Smartphone GNSS positioning and discussing some limitations. In section II, we give an overview about our in-house GNSS/IMU logger developed at the institute. The complete time-synchronizing methodology has been implemented inside this Android Logger. In Section III, we will discuss the core of our research regarding implementation of time-synchronizing technique and their relevant equations. Further in section IV and section V, we will discuss some measurement setup and their corresponding results to validate the implementation. Finally, section VI, will focus on the conclusion and achievement related to the work.

I. INTRODUCTION

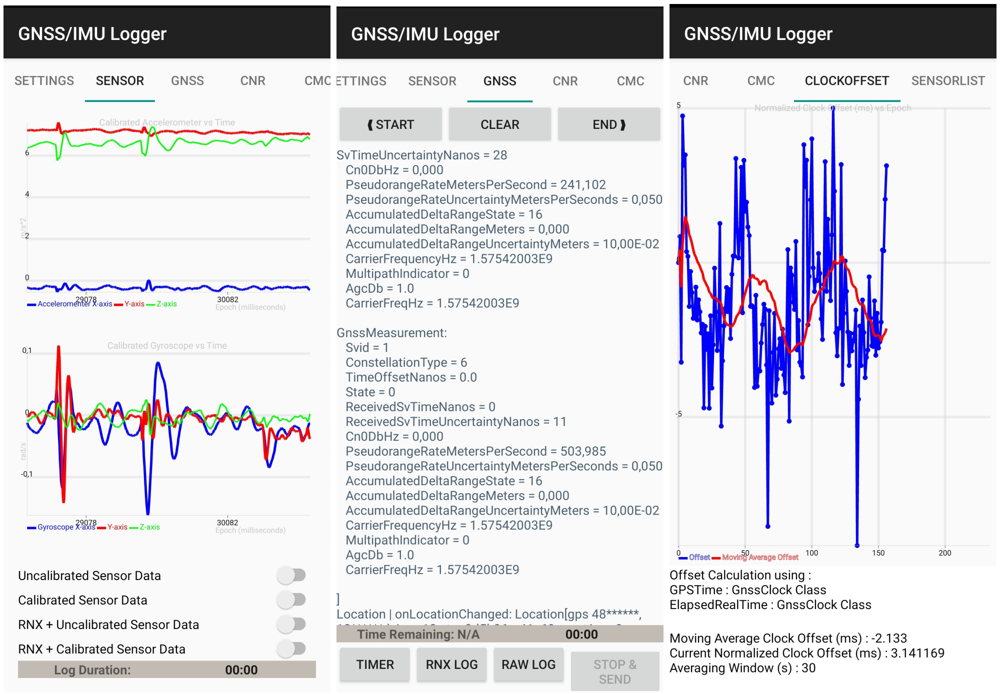
Today, the vast majority of GNSS receivers are installed in Smartphones with 1.5 billion devices produced every year. Most of these newly produced phones make GNSS raw measurements available to the applications, a feature supported by the Android operation system since 2017. This led to numerous new Smartphone applications and 1000+ research papers focusing on GNSS positioning with mobile phones [1]. Despite the latest innovation in the past years in the domain of GNSS carrier-phase positioning, there are still key limiting factors which need to be addressed before full scale use of Smartphones in high precision applications. Cycle-slip, a jump in the carrier-phase pseudorange due to the multipath is one of the prominent factors. Since, the other source of cycle-slip such as duty-cycle can be controlled in the recent smartphones. But the multipath still cause a severe degradation which needs to be addressed. An increasing number of these phones is supporting dual-frequency measurements on the L1/E1 and L5/E5a bands. The use of an additional frequency (L5/E5a) with higher chipping rate (10 times to that of L1) results in a narrower correlation peak, making the measurements more precise and eliminating some of the multipath distortions. While these developments pave the way to transfer high precision positioning technology from expensive professional devices to mass-market Smartphones, there is still the major hurdle of successful carrier-phase positioning (i.e., ambiguity fixing) to overcome before reliable decimeter- or centimeter-level positioning is achieved with phones. The availability of raw GNSS measurements from the Smartphone does not guarantee the feasibility of successful RTK (Real Time Kinematics) positioning. Due to limited access to the GNSS chip hardware, it is difficult to evaluate the base-band processing performance of the GNSS chip. Instead, we can only analyze the observation data of the Smartphone. In the series of measurement campaign performed at UniBw M, it was quite evident that the Smartphone measurement quality under standard environmental conditions is in-sufficient for RTK positioning with cm-level accuracy [2] [3] [4] [5] [6]. The re-transmission setup used in [4] showcase the effect of multipath and its impact on positioning accuracy. Having a good quality antenna, the GNSS-chip inside the Smartphone could provide high quality measurement data with ambiguity fixing [6]. In another set of experiment, a choke ring was used to mitigate the multipath. A comparison between two similar Smartphone (Xiaomi Mi 8) with/without choke-ring platform also indicate that the major limitation, to achieve cm-level positioning accuracy comes of the Smartphone GNSS antenna and its multipath mitigating in-capabilities. With the success of the choke-ring experiments in both static and dynamic scenarios, an accuracy of the positioning solution is reached that is sufficient to localize the antenna phase center (APC) in the frame of the Smartphones. Low-cost mass-market GNSS antennas in the Smartphones are subject to low gain and poor multipath suppression. Mobile devices utilize an omni-directional linearly or elliptically polarized antenna due to the unknown orientation of the Smartphone in use. This type of antenna has advantages in terms of received signal strength and the number of received signals [7], but also makes the antenna very sensitive to the multipath (MP) effects. The latter limitation is generally accepted since the design drivers of Smartphone antennas are mainly cost and signal availability and not the observation data quality. Furthermore, not only the antenna itself but also other components of the phone, like the screen of the device and other transmitting antenna (Wi-Fi, Bluetooth), affect the Smartphone antenna [8], leading to the reception pattern irregularities. Considering the type of antenna, the repeatability of the calibration is considered good enough to apply the corrections in a positioning algorithm [5]. The use of choke-ring platform and re-transmission setup has a limitation when it comes to a practical application. The existing GNSS algorithms can only provide float solutions with the GNSS measurements from the Smartphone. To enhance both position accuracy and availability with the existing hardware, it needs an external aiding from additional sensor such interial measurement unit (IMU). The usability and quality analysis of embedded sensor (accelerometer and gyroscope) in aiding GNSS measurements from Smartphones has been extensively discussed in this contribution [9].

II. GNSS/IMU LOGGER

GNSS/IMU logger is an Android application developed by the Universität der Bundeswehr München (UniBw M) [10]. The logger is explicitly developed for research purposes and is available on the Android (Google) Play Store. The user manual available within the logger, gives a complete description about the use of tool and list all the GNSS software where the data logged with the application has been processed and tested successfully. The institute also provide a MATLAB tool which can be used to convert logged sensor data to Inertial Explorer (**.imr*) and MuSNAT [11] compatible format. Few of the screenshots from Android application is shown below in figure 1.

Detailed list of features provided by our GNSS/IMU logger are as follows:

- RINEX 3.03 with dual-frequency and multi-constellation GNSS observation, i.e. GPS (L1 and L5), Galileo (E1 and E5a), GLONASS, Beidou
- GNSS raw measurements
- Both calibrated and uncalibrated IMU raw signal (accelerometer, gyroscope) and magnetometer. The uncalibrated IMU data can be used for the laboratory calibration. If a calibration is not possible, the user can switch to the calibrated data, which contains correction that are applied internally during the collection procedure.
- UTC time Synchronized GNSS / IMU data to be directly used for fusion purposes



Uncalibrated Sensor Data ○
Calibrated Sensor Data ○
RNX + Uncalibrated Sensor Data ○
RNX + Calibrated Sensor Data ○
Log Duration: 00:00

```
SvTimeUncertaintyNanos = 28
  Cn0DbHz = 0,000
  PseudorangeRateMetersPerSecond = 241,102
  PseudorangeRateUncertaintyMetersPerSeconds = 0,050
  AccumulatedDeltaRangeState = 16
  AccumulatedDeltaRangeMeters = 0,000
  AccumulatedDeltaRangeUncertaintyMeters = 10,00E-02
  CarrierFrequencyHz = 1.57542003E9
  MultipathIndicator = 0
  AgcDb = 1.0
  CarrierFreqHz = 1.57542003E9

GnssMeasurement:
  Svid = 1
  ConstellationType = 6
  TimeOffsetNanos = 0.0
  State = 0
  ReceivedSvTimeNanos = 0
  ReceivedSvTimeUncertaintyNanos = 11
  Cn0DbHz = 0,000
  PseudorangeRateMetersPerSecond = 503,985
  PseudorangeRateUncertaintyMetersPerSeconds = 0,050
  AccumulatedDeltaRangeState = 16
  AccumulatedDeltaRangeMeters = 0,000
  AccumulatedDeltaRangeUncertaintyMeters = 10,00E-02
  CarrierFrequencyHz = 1.57542003E9
  MultipathIndicator = 0
  AgcDb = 1.0
  CarrierFreqHz = 1.57542003E9

]
Location | onLocationChanged: Location[gps 48******,
```

Time Remaining: N/A 00:00

Offset Calculation using :
GPSTime : GnssClock Class
ElapsedRealTime : GnssClock Class

Moving Average Clock Offset (ms) : -2.133
Current Normalized Clock Offset (ms) : 3.141169
Averaging Window (s) : 30

Figure 1: Screenshots of GNSS/IMU logger app (version: v2.1.0.1)

- Other GNSS performance parameter indicators such as, Code-Minus-Carrier (CMC), Signal-to-Noise-Ratio (CN0), clock offset.

The GPS Time synchronization feature discussed in this paper has been completely implemented inside the latest version of GNSS/IMU logger (v2.1.0.1) and is available for download.

III. TIME SYNCHRONIZATION MECHANISM

One of the first problem encountered when investigating the GNSS/INS fused (e.g. Loosely Coupled) positioning solution with Smartphone sensors (GNSS/IMU) or any low cost commercial off-the-shelf components (COTS) is the time synchronization. The GNSS measurements from the smartphone are `GnssClock` driven and GPS time tagged. Whereas, the `SystemClock` is used to timestamp the IMU measurement. Android 7 does not provide the GNSS time directly, but the internal hardware clock and the bias to the true GPS time (in nanoseconds) is provided if the receiver has estimated the GNSS reference time. When the receiver has estimated the GPS time, it can be computed as:

$$localestimateofGPStime = TimeNanos - (FullBiasNanos + BiasNanos) \tag{1}$$

where, *FullBiasNanos* is the bias between the receiver clock and the GPS time in nanoseconds and *BiasNanos* is in sub-nanoseconds. The Sensor Logger fragment inside the logger used the equation below to log UTC time stamped IMU data.

$$timeInMillis = System.currentTimeMillis + (event.timestamp - SystemClock.elapsedRealtimeNanos) \tag{2}$$

where, *System.currentTimeMillis* is the standard "wall" clock (time and date) expressing milliseconds since the epoch, event.timestamp is the time in nanoseconds at which the event happened. For a given sensor, each new sensor event should be monotonically increasing and *SystemClock.elapsedRealtimeNanos* returns nanoseconds since boot, including time spent in sleep.

Although, the *SystemClock* get synchronized to the UTC time via network provider, one can often encounters a small timing error between the sampling instances of the GNSS and IMU measurements. System time is affected by users, broadcast, and network, so the time may jump backward or forward unpredictably. As a result, the synchronization error between GNSS and

IMU data using Android system time can reach hundreds of milliseconds, and the error may even exceed 1 s with the increase of time [12]. The GNSS/IMU logger is capable of logging RINEX and sensor data simultaneously. However, both these logging modules are driven by separate clock classes. The goal is to synchronize system clock GPS time calculated using *GnssClock* class.

The GPS time calculated in the RINEX logger fragment is passed to the sensor logger fragment in the Android App. The offset between GPS time and UTC time is then calculated using the equation 3. This offset value is then used to timestamp the IMU data in GPS time frame using equation 4. The simple block diagram in Fig. 2 to showcase the difference between unsynchronized and synchronized GNSS/IMU logging. For an unsynchronized logging, *gnssclock* class is used to used to timestamp the RINEX data. Similarly, the IMU data is logged using the UTC timestamped using the *SystemClock*. With the time synchronized GNSS/IMU logging, the GPS-time update is passed to IMU logging module every time the GNSS measurement are available from the API.

$$SysgpssOffsetnano = localestimateofGPStime - SystemClock.elapsedRealtimeNanos \tag{3}$$

$$gpstimemilli = (SysgpssOffsetnano + event.timestamp)/1000000L \tag{4}$$

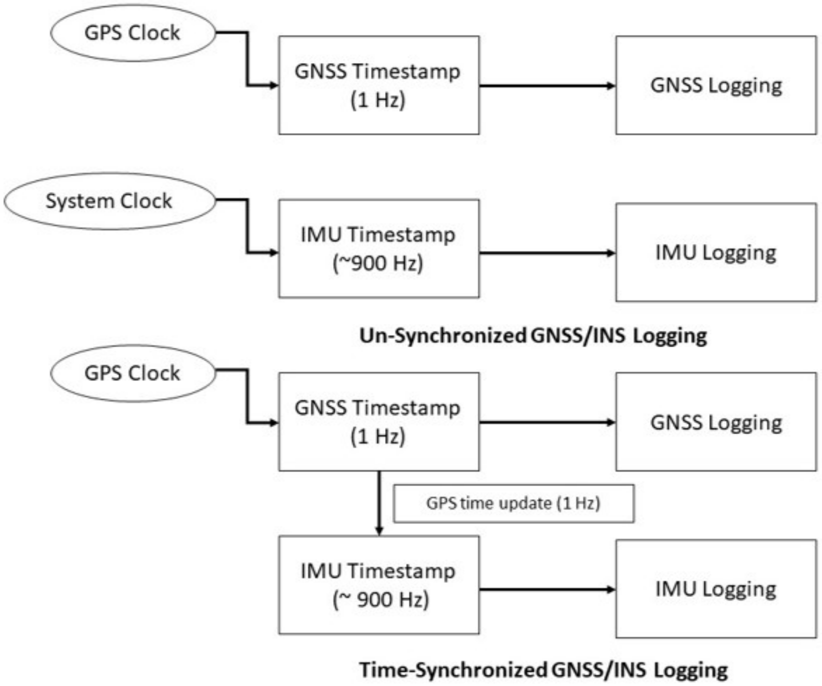


Figure 2: Unsynchronized GNSS/IMU and synchronized GNSS/IMU logging

A practical implementation to visualize the need of time synchronization was also done during the implementation. Despite of 18 leap seconds to synchronize UTC to GPS, a visible jitters can be seen in the figure 3. To damp this jitters from GPS-UTC, a moving average filter with window size 30 sec was implemented. This means that the offset average of 30 epochs was passed to sensor logger module to timestamp sensor data.

The left block of RINEX logger uses *systemclock* and *gnssclock* class to calculate the offset (Fig. 4 as explained above. The moving average offset block calculate the offset between GPS minus UTC time and pass it to sensor logger block on the right. The calculated offset and sensor event is used to GPS time stamp the IMU data. On the other hand *elpasedrealtime* and *SensorEvent* from *systemclock* is used to UTC timestamp the IMU data. Both timestamp are stored in sensor log file for valid comparison which will be done in the later part. One more point to be noted is that the elapsed *realtimenano* from *gnssclock* is more precise in comparison to *elapsedrealtimenano* from systemclock but is not available below Android 29 and not all the smartphones.

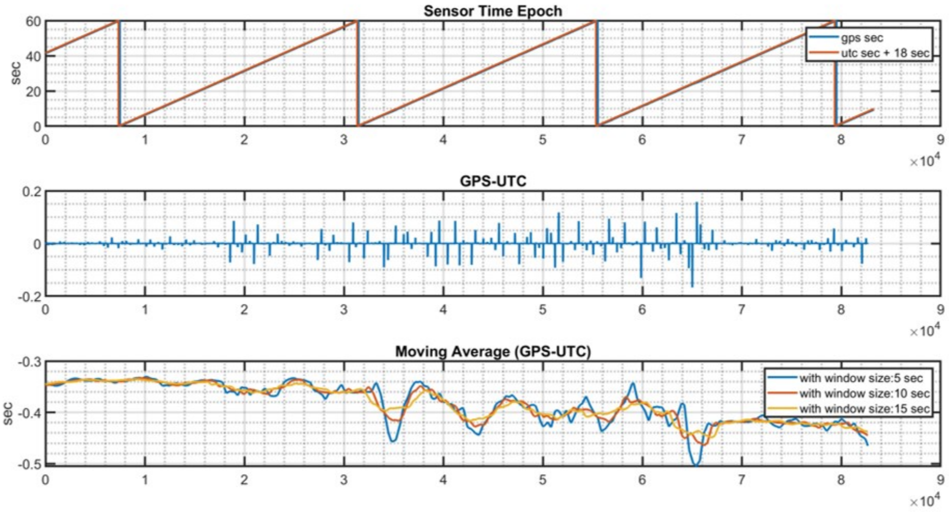
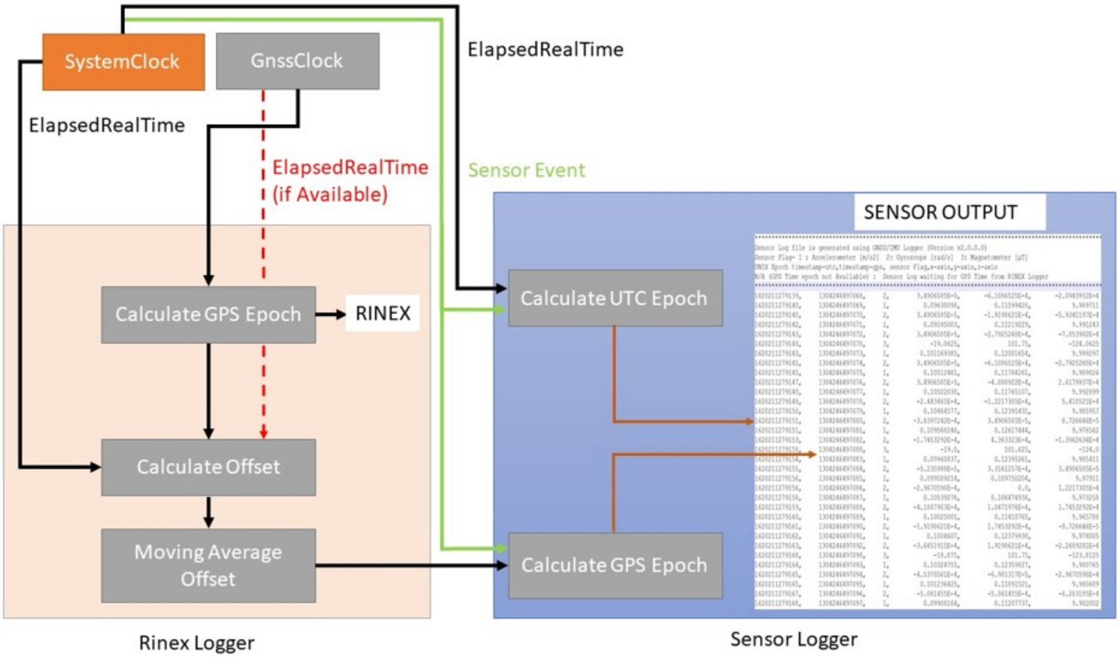


Figure 3: GPS-UTC time offset calculated with Xiaomi Mi 8



```
Sensor Log file is generated using GNSS/IMU Logger (Version v2.0.0.0)
Sensor Flag= 1 : Accelerometer [m/s2]  2: Gyroscope [rad/s]  3: Magnetometer [µT]
UNIX Epoch timestamp-utc,timestamp-gps, sensor Flag,x-axis,y-axis,z-axis
N/A (GPS Time epoch not Available) :  Sensor Log waiting for GPS Time from RINEX Logger
```

Figure 4: Synchronization mechanism implemented

IV. MEASUREMENT CAMPAIGN

In the first campaign a re-transmission setup was placed inside the car. The receiving patch antenna was placed on the top of the vehicle and and transmitting antenna was placed inside the vehicle near to the smartphone as shown in figure 5. The GNSS signal received from the antenna on the roof top is is amplified before being re-transmitted by antenna inside the car. The data was logged inside the UniBw M campus. The amplification was however limited to L1 frequency only. The data was processed with NovAtel *Inertial Explorer* (version 8.70) using the settings as stated in Tab. 1.

In another set of measurement campaign, the Smartphone is now placed on the roof top of a measurement van. In order to mitigate the multipath, the smartphone is placed on the choke-ring platform mounted on the roof of measurement van as depicted in Fig. 6.

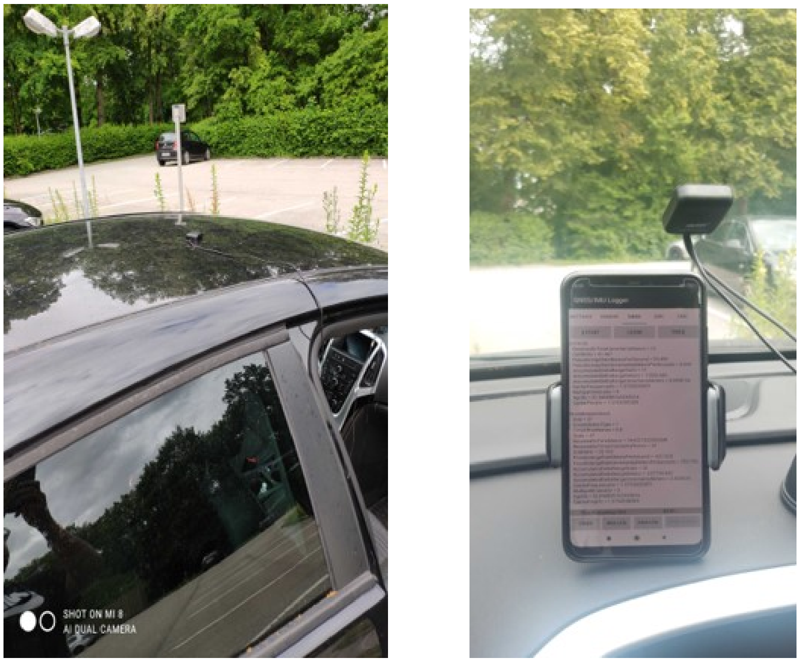


Figure 5: Measurement setup with Re-Transmission antenna

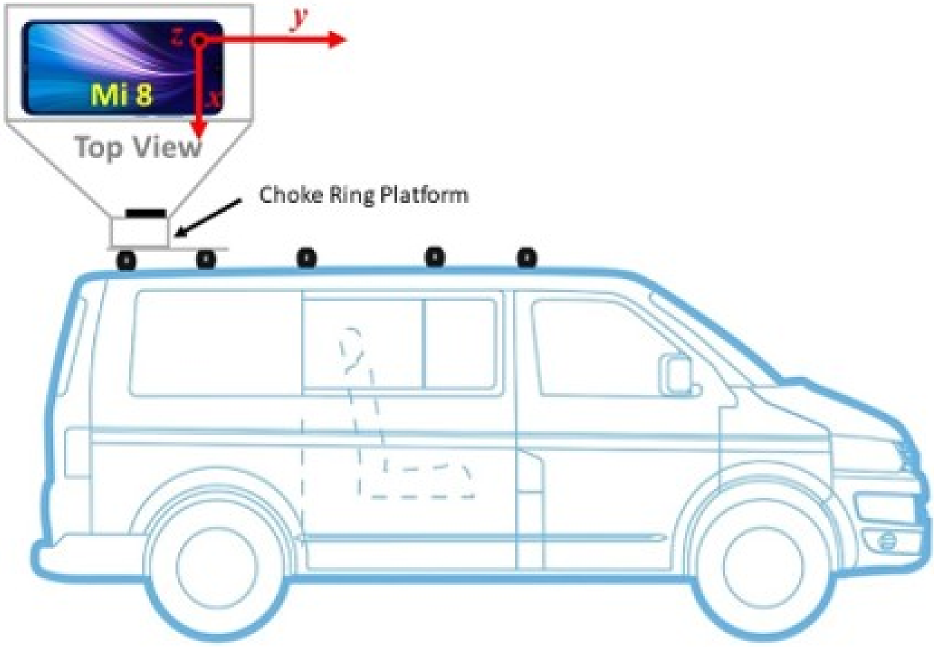


Figure 6: Measurement setup with Choke-Ring [9]

V. RESULTS AND ANALYSIS

In the first processing step with IE, no IMU data was used i.e. GNSS only solution was computed. Due to the re-transmission setup, good quality of GNSS signal was received and the ambiguity fixing worked well in majority of the trajectory. However, during the drive, GNSS signal interruptions happen as shown in the figure 7.

The ambiguity fixing status in plot 8 , lots of data gaps are visible between epochs 138000 and 138200. In addition to missing

GNSS/IMU Processing	Loosely Coupled
GNSS Processing	Differential
Constellation used	GPS, GALILEO
IMU Profile	Ground vehicle, MEMS

Table 1: Employed settings in *Inertial Explorer*

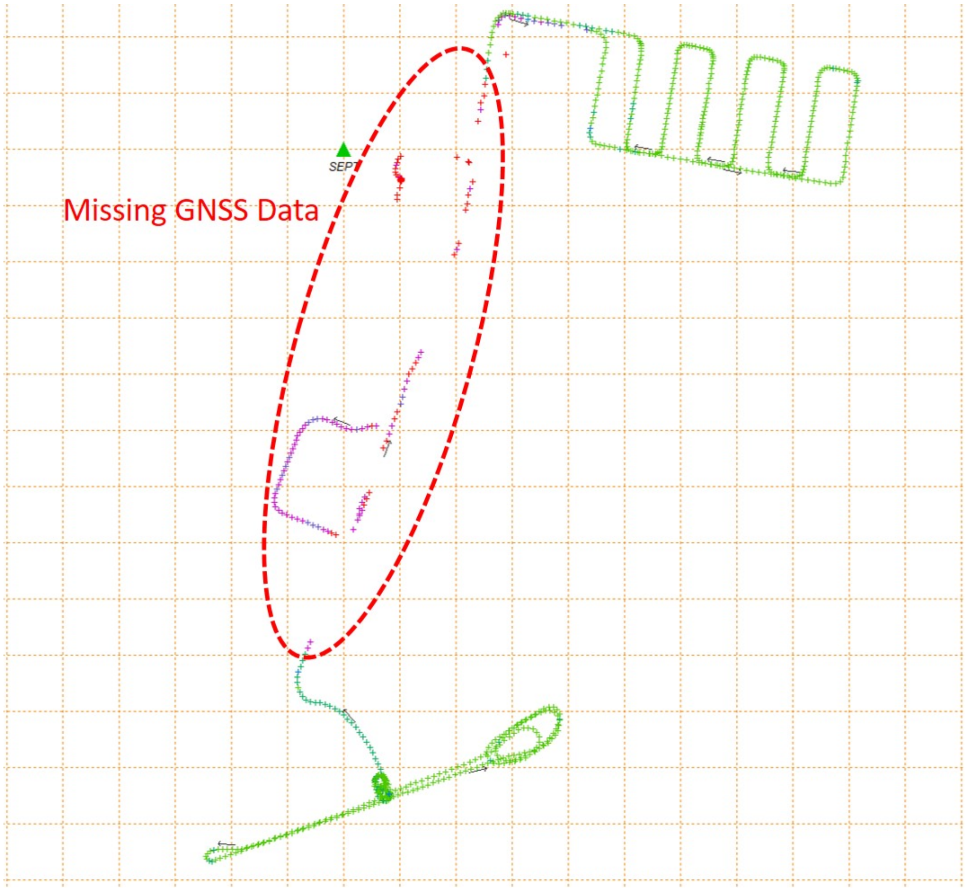


Figure 7: GNSS Only Trajectory with Re-transmission Setup

GNSS data, the ambiguity status for this duration is also float due to dense trees and multipath source around. In the second run, a loosely-coupling GNSS/IMU processing was done. The integration of IMU data shows a significant improvement in the processing results. The GNSS/IMU trajectory plot in 9 and Ambiguity status plot in 10 indicated the data gaps created due to missing GNSS data was filled by IMU data. Furthermore, the IMU signal has a positive impact on the number ambiguities fixed as shown in the figure 10.

However, it must be noted that in standard Loosely Coupled GNSS/INS processing, the ambiguity fixing should be unaffected as the LC uses positioning solution (Tight Coupling uses observation data) where the ambiguity state is determined before the IMU data is added. Such behaviour in the result is not explained in details as there is only limited information available about the processing engine from Inertial Explorer.

The second data set logged with choke-ring setup is now processed with the IE. To analyze the affect of time synchronization, two processing options were chosen. In the first processing, GNSS/IMU data was processed with UTC (unsynchronized) timestamp IMU data. In the second processing option, GNSS/IMU processing with GPS (synchronized) timestamp IMU data was done. The output trajectory and heading values for both processing options were exported and analyzed using MATLAB.

Plot 11 shows 3 heading signals, one from the COG (coarse over ground) indicated in blue, second heading in red from GPS time synchronized IMU data, third heading in orange from from UTC time (non synchronized) IMU data. For the ease of

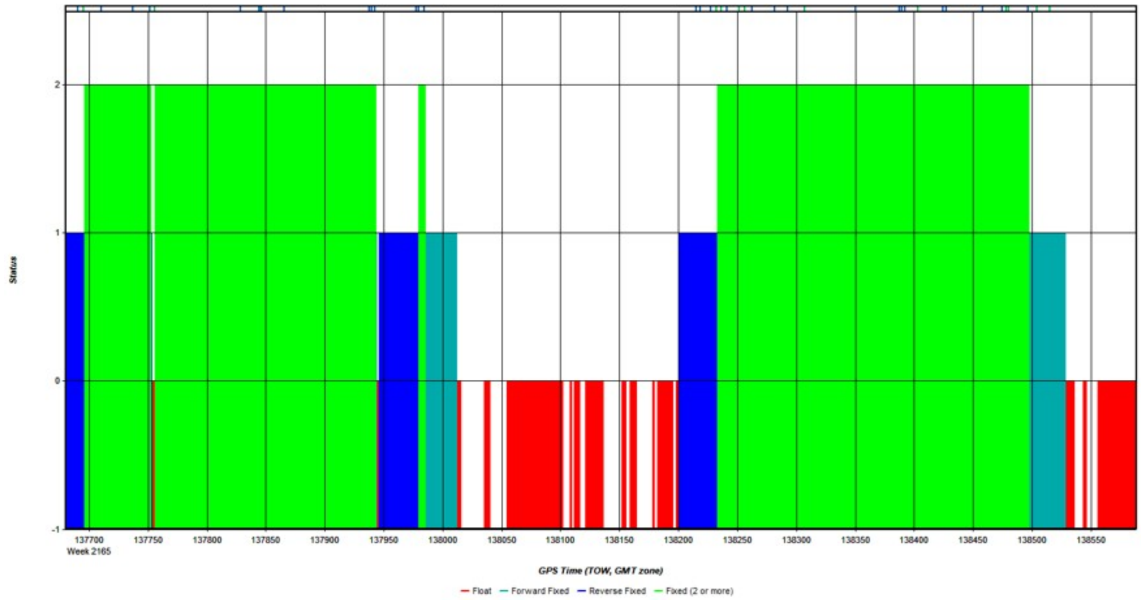


Figure 8: GNSS Only Ambiguity Status with Re-transmission Setup

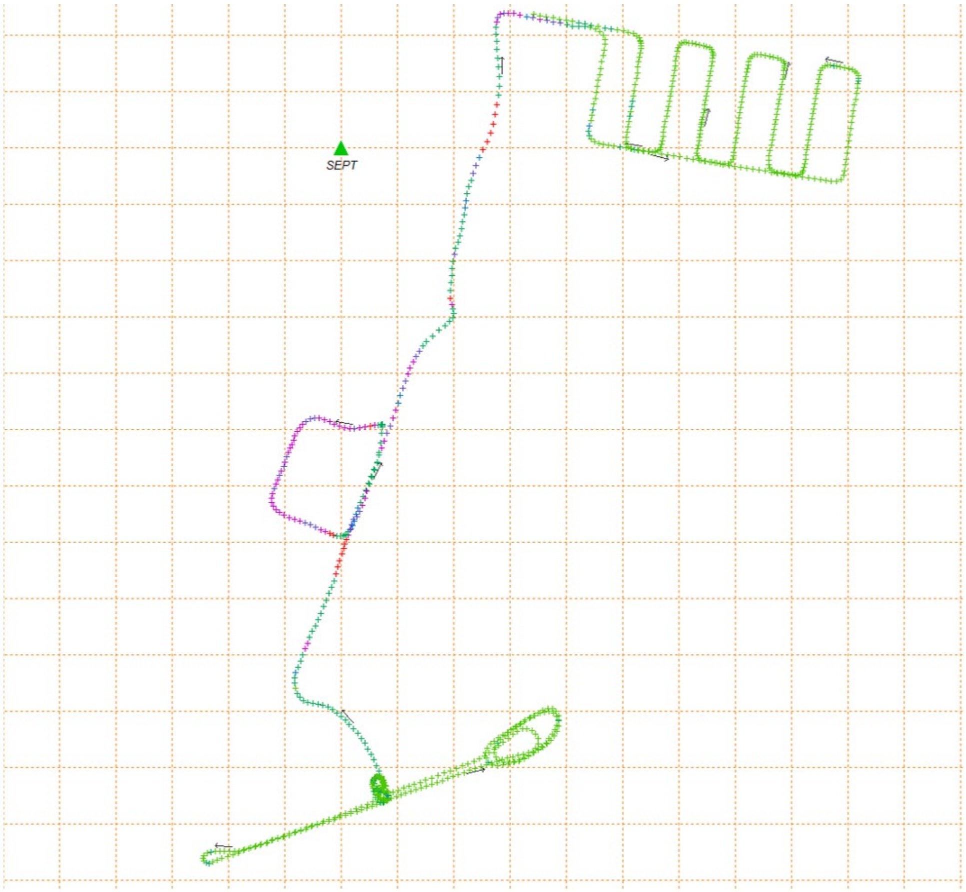


Figure 9: GNSS/IMU Trajectory with Re-transmission Setup

understanding two sector have been shown to indicate the influence of time synchronization on the heading.

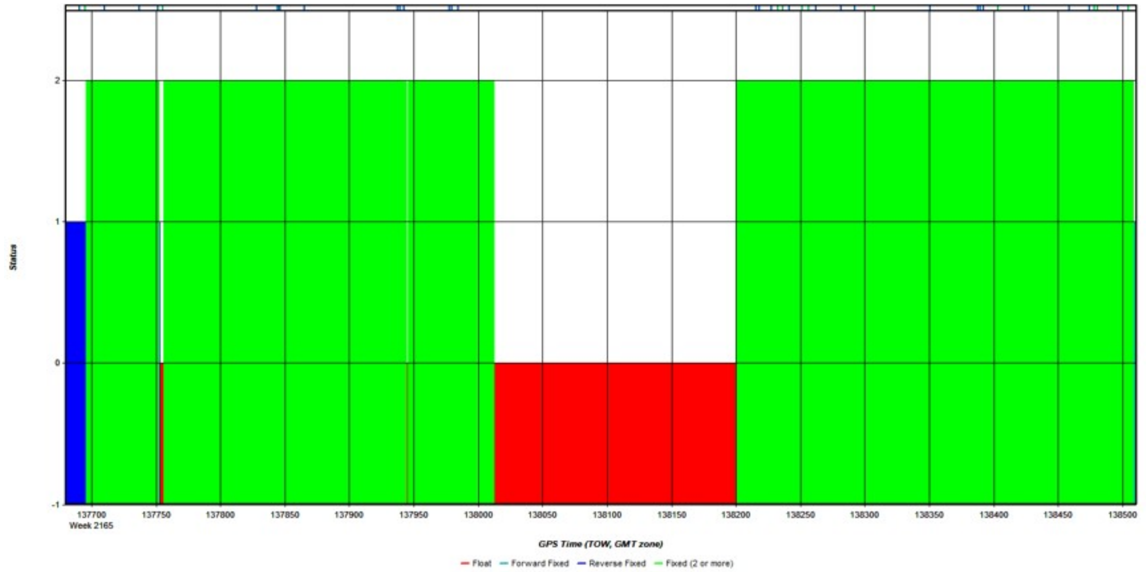


Figure 10: GNSS/IMU Ambiguity Status with Re-transmission Setup

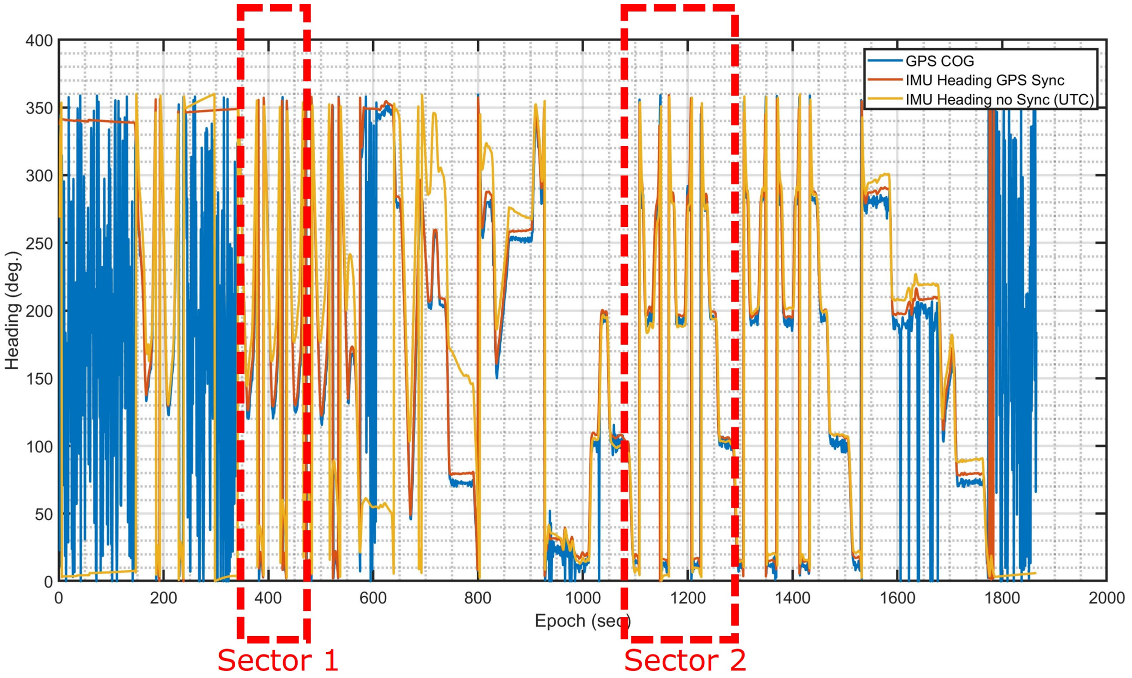


Figure 11: Heading Comparison with GNSS/IMU Processing (Synchronized and Un-Synchronized IMU data)

Sector 1 in the figure 12 taken from the initial motion of Fig. 8 to calibrate the sensors. In part one it is clearly visible that the heading from GPS sync IMU data is more aligned to heading value from COG. For any time epoch in P1, the difference in COG – non sync IMU heading in much larger as compared to the difference in COG–GPS sync IMU heading. In another part of trajectory P2, the GPS sync IMU heading still follows the COG value closely. However, in this section the non-GPS sync IMU heading is also good. A similar behaviour as P1 can been seen in part P3 and P4. Now moving onto the sector 2 in the figure 13 of the trajectory, it is quite clear that the GPS sync heading either outperform the UTC heading or performs similar through out the trajectory.

Now to show a clear difference a plot for IMU heading minus COG was plotted in Fig. 14 for both UTC and GPS IMU heading.

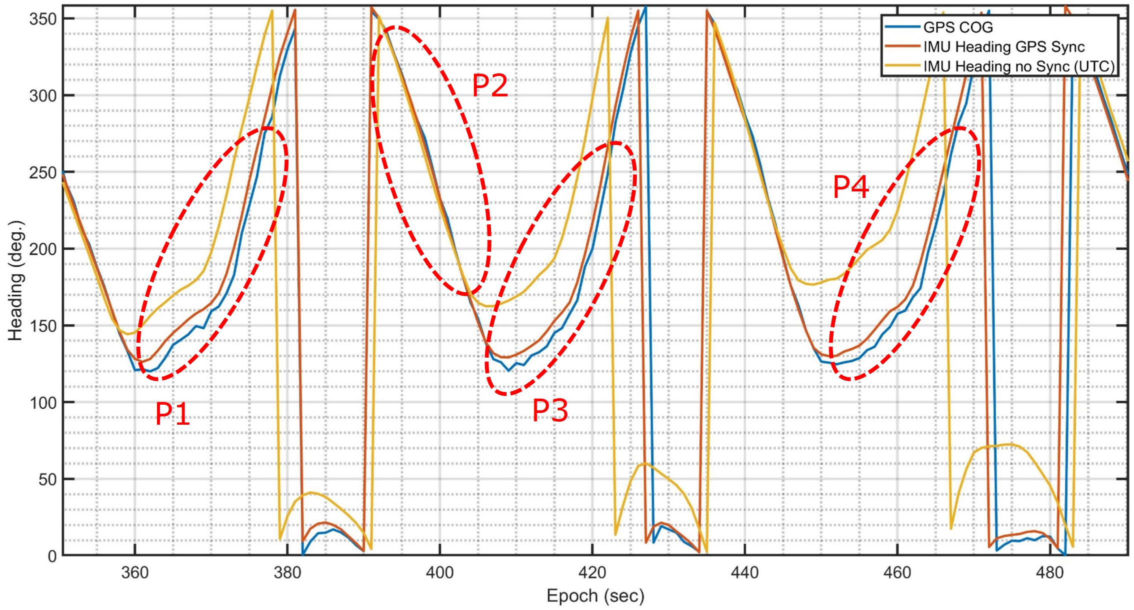


Figure 12: Heading comparison (sector 1) with GNSS/IMU processing (synchronized and unynchronized IMU data)

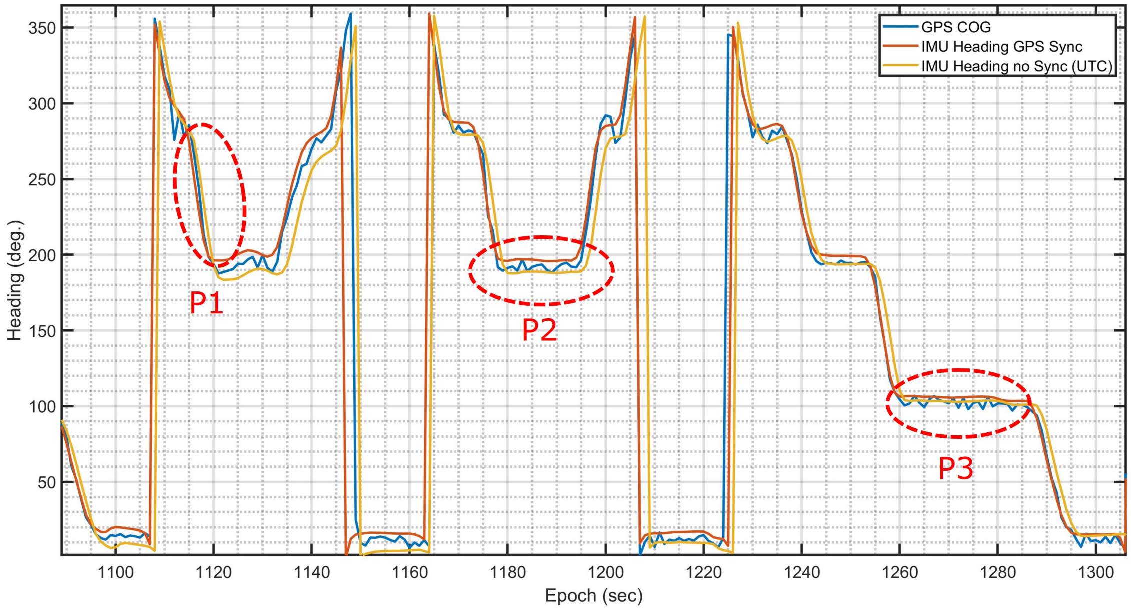


Figure 13: Heading comparison (sector 2) with GNSS/IMU processing (synchronized and unsynchronized IMU data)

As can been seen, the IMU heading minus COG value for GPS sync IMU data is much smaller and smoother in comparison to the UTC heading minus COG. This plots clearly indicated the positive influence of time synchronization the IMU data logged from the smartphone.

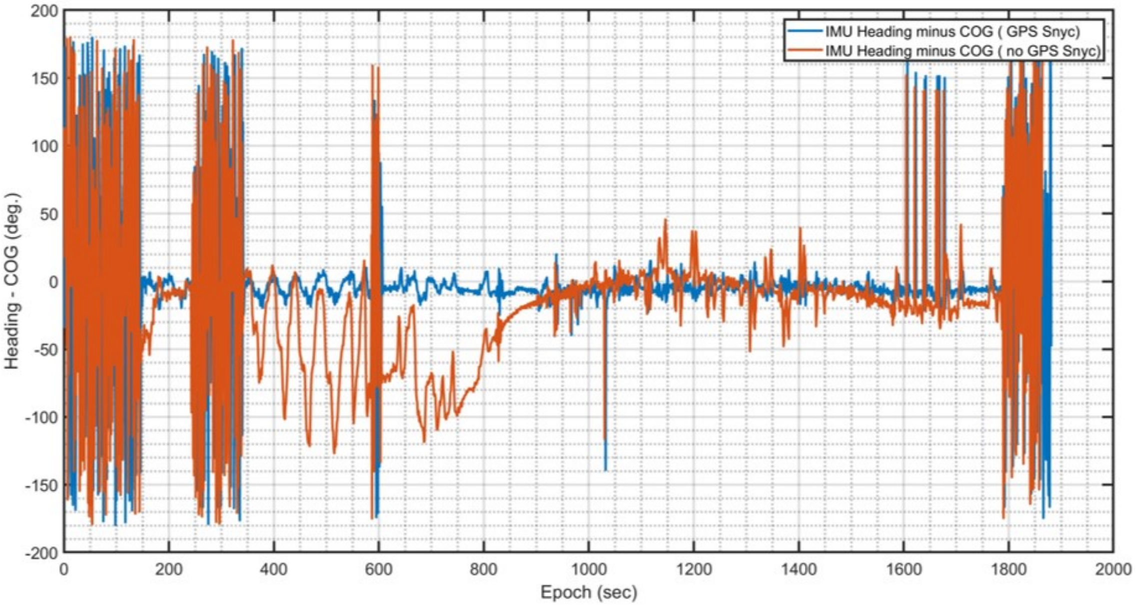


Figure 14: COG minus heading for synchronized and unynchronized GNSS/IMU processing

VI. CONCLUSION AND FUTURE WORK

The GNSS/IMU processing (LC) has been successfully demonstrated and highlights the advantage of using IMU data in the highly dense environment when GNSS data is not always available. A re-transmission setup when used with dynamic scenario has shown excellent positioning results with GNSS only processing by fixing large number of ambiguities. The time-synchronization mechanism has demonstrated that the GNSS and IMU logged with the Smartphone produces a significant improvement in the heading information when compared with un-synchronized data. Though, the *SystemClock* used for time-stamping the IMU data get synchronized to UTC time via network, internet or some other source. But, due to cheap clock and the un-availability of synchronization source, the UTC time tends to drift away from GPS time (excluding lead second correction). In addition to this drift a jitter of the range of 0.2 seconds due to clock jump can been seen with Xiaomi MI8. The implementation of moving average filter (window size of 30 seconds) proved to be useful techniques for mitigating this jitter. However, it must be noted that in addition to this time-synchronization approach, the availability of *ElapsedRealTimeNano* from *GnssClock* class is very likely to minimize the jitter due to fact that *GnssClock* is expected to be more precise than *SystemClock*. But, the *GnssClock.ElapsedRealTimeNano* parameter is not available before Android API 29 and is also manufacturer specific. The Smartphone Xiaomi MI8 used in this paper does not supports the availability of *ElapsedRealTimeNano* from *GnssClock* class. In addition to these findings, there are still some open points which need to be addressed in the future work.

- The comparison of GPS Synchronized trajectory with Reference trajectory.
- The use of IMU data in tight coupling (TC) when raw code and carrier phase measurements are used instead of position values in GNSS/IMU Kalman filter.
- Impact of further optimization of IMU profile chosen for processing.
- Offline Calibration of Smartphone IMU using sophisticated techniques such as turn-table and their impact on positioning accuracy when added with GPS time-synchronization.

REFERENCES

[1] G. M. Fu, M. Khider, and F. van Diggelen, “Android Raw GNSS Measurement Datasets for Precise Positioning,” in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, 2020, pp. 1925–1937.

[2] H. Sharma, A. nd Schütz, and T. Pany, “Preliminary analysis of the RTK positioning using android GNSS raw measurements and application feasibility for the trajectory mapping using UAVs,” in *Proceedings of the 31st International Technical*

Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2018). Institute of Navigation, oct 2018.

[3] H. Sharma, A. Schütz, and T. Pany, "Qualitative analysis of smartphone GNSS raw measurements and effect of duty cycling on the RTK positioning," in *NAVITEC*, Noordwijk, The Netherlands, 2018.

[4] H. Sharma, M. Bochkati, G. Kestel, and T. Pany, "Researchers achieve 1–2 cm accuracy with commercial smartphone," *Inside GNSS*, 2019.

[5] F. Darugna, J. B. Wübbena, G. Wübbena, M. Schmitz, S. Schön, and A. Warneke, "Impact of robot antenna calibration on dual-frequency smartphone-based high-accuracy positioning: a case study using the Huawei Mate20X," *GPS Solutions*, vol. 25, no. 1, nov 2020.

[6] H. Sharma, A. Schütz, and T. Pany, "Emulating the smartphone GNSS receiver to understand and analyze the anomalies in RTK positioning using GNSS raw measurements," in *Proceedings of the 2019 International Technical Meeting of The Institute of Navigation*. Institute of Navigation, feb 2019.

[7] V. Pathak, S. Thornwall, M. Krier, S. Rowson, G. Poilasne, and L. Desclos, "Mobile handset system performance comparison of a linearly polarized GPS internal antenna with a circularly polarized antenna," in *IEEE antennas and propagation society international symposium. Digest. Held in conjunction with: USNC/CNC/URSI North American Radio Sci. Meeting (Cat. No. 03CH37450)*, vol. 3. IEEE, 2003, pp. 666–669.

[8] B. Xiao, H. Wong, B. Wang, and K. L. Yeung, "Effect of the screen to metal-frame smartphone antennas," in *2019 International Workshop on Antenna Technology (iWAT)*. IEEE, 2019, pp. 29–32.

[9] M. Bochkati, H. Sharma, C. A. Lichtenberger, and T. Pany, "Demonstration of Fused RTK (Fixed) + Inertial Positioning Using Android Smartphone Sensors Only," in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2020, pp. 1140–1154.

[10] ISTA-UniBw M, "GNSS/IMU Logger [Mobile app]," 2021. [Online]. Available: https://play.google.com/store/apps/details?id=com.ista.android.apps.location.gps.gnsslogger&hl=de&gl=US

[11] T. Pany, D. Dötterböck, H. Gomez-Martinez, M. S. Hammed, F. Hörkner, T. Kraus, D. Maier, D. Sánchez-Morales, A. Schütz, P. Klima *et al.*, "The Multi-Sensor Navigation Analysis Tool (MuSNAT)–Architecture, LiDAR, GPU/CPU GNSS Signal Processing," in *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, 2019, pp. 4087–4115.

[12] M. Benndorf and T. Haenselmann, "Time synchronization on android devices for mobile construction assessment," in *The Tenth International Conference on Sensor Technologies and Applications. Thinkmind*, 2016.