Budownictwo i Architektura 18(3) 2019, 51-62 **DOI: 10.35784/bud-arch.738**

The problem of GNSS positioning with measurements recorded using Android mobile devices

Bogdan Skorupa

Department of Integrated Geodesy and Cartography;
Faculty of Mining Surveying and Environmental Engineering;
AGH University of Science and Technology
e–mail: bskorupa@agh.edu.pl; ORCID: 0000-0002-2073-1957

Abstract: The current work presents the issue of determining the position of the observer using measurements registered with GNSS (Global Navigation Satellite System) receivers that Android mobile devices are equipped with. The discussed questions concern using GNSS measurement data, which have been made available in the Android system since version 7.0. The present paper has the character of a review. It demonstrates how measurement data can be obtained via Application Programming Interface. Moreover, it discusses the available software that can be for registering measurements and their initial analysis. Subsequently, it reviews scientific works concerning the problem of positioning with the use of smartphones. Special emphasis was placed on tests consisting in an analysis of phase observations registered using dual-frequency receivers. The summary of the article presents the prospects for using mobile devices in precise point positioning. It also points out the limitations to achieving high accuracy and reliability of such measurements.

Keywords: GNSS, Android, positioning

1. Introduction

In August 2016, Google launched Android version 7.0 Nougat. Beginning with this version of the system, users of mobile devices equipped with a satellite navigation module gained access to GNSS (Global Navigation Satellite System) observations. Until then, positioning using mobile devices was based on the location calculated by the internal software of the GNSS chipset. The release of GNSS measurements makes it possible to use proprietary calculation algorithms adapted to the limitations of the equipment and to field conditions of positioning. It also provides an opportunity to analyse the acquired GNSS observations outside the recording device, using external software [1], [2].

In 2018, smartphones equipped with dual-frequency GNSS receivers appeared on the mobile devices market. They enable recording of code and phase measurements as well as ephemeris data transmitted by satellites. The availability of two frequencies allows for increasing the accuracy and reliability of positioning by reducing multipath signal error and eliminating the influence of ionospheric refraction [3]. This opens the way to the application of measurement and calculation methods previously reserved for high-precision geodetic methods.

It is anticipated that devices equipped with dual-frequency GNSS chipsets will be used for positioning with an accuracy of single decimetres or even centimetres. This will make it possible to use them in various types of services related to the functioning of the smart city: precise navigation, autonomous vehicles [4] or support for emergency services operations [5].

2. Access to GNSS observations from the programming interface

In Android, access to data recorded by the built-in sensors is via the Application Programming Interface. Launching a new version of the operating system entails the publication of the appropriate version of the API. Until the release of Android version 7.0 Nougat, access to navigation data was via the android.location module containing the GpsSatellite, GPSStatus and Location classes. These classes were used to obtain information about the satellites used in the navigation solution, such as azimuth or horizontal height, as well as calculated values of the location and speed of the GNSS receiver. Starting from API 24, the following classes were added to the android.location module: GNSSClock, GNSSNavigationMessage and GNSSMeasurements [6] (Fig. 1).

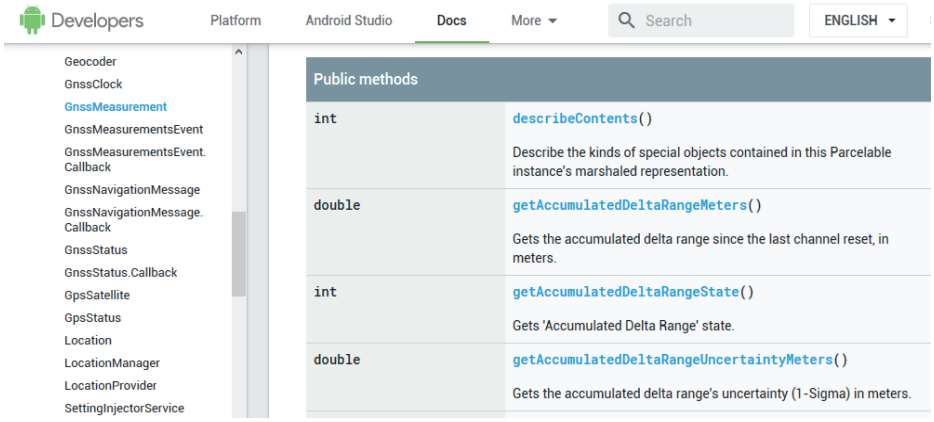


Fig. 1. Android API 24 online documentation (Source: https://developer.android.com/guide/topics/sensors/gnss)

Carrier phase measurements and Doppler measurements are available directly by calling the getAccumulatedDeltaRangeMeters() and getPseudorangeRateMetersPerSecond() methods, which belong to the GNSSMeasurements class. For pseudorange, its value should be calculated using intermediate data. The pseudorange equation shall be as follows [8]:

$$\rho = (t_r - t_s) \cdot c \tag{1}$$

where c is the speed of light in vacuum, t_s is the time the satellite generated the signal, while t_r is the time the signal was received by the receiver. The getReceivedSvTimeNanos() method of the GNSSMeasurements class gets the epoch of signal generation t_s in the form of the total number of nanoseconds [6]. This number is expressed in the time system appropriate for a given satellite constellation. The t_s value is calculated from the beginning of the current week for all systems except GLONASS (Globalnaya Navigatsionnaya Sputnikovaya Sistema), which refers to the beginning of the day. To calculate the measurement time t_r, the getTimeNanos() and getFullBiasNanos() methods of the GNSSClock class should be used. The first one gets the clock value t_0, expressing the time interval that has passed since the GNSS receiver has started working. The second one, in turn, gets the value of the difference Δt_0 between t_0 and the GPS (Global Positioning System) time calculated since the first GPS epoch (6.01.1980). The values of t_0 i Δt_0 are expressed in whole nanoseconds. To include fractional nanoseconds, dt_0 and δt_0 corrections must be included, returned

by GNSSMeasurements.getTimeffsetNanos() and GNSSClock.getBiasNanos() methods respectively. Assuming that the reference time for the GNSS receiver is the time of the GPS system, the equation is as follows:

$$t_{GPS} = (t_0 + dt_0) - (\Delta t_0 + \delta t_0) \tag{2}$$

The equation (2) expresses the measurement epoch as the number of nanoseconds that have elapsed since the first epoch of the GNSS receiver reference time. Subsequently, the measurement epoch is related to the beginning of the week or the beginning of the day, taking into account the differences between the time system in which the GNSS receiver works and the time system of the observed satellite. Thus, the measurement epoch expressed in seconds, related to the beginning of the week of GPS or Galileo time will be calculated with the following formula:

$$t_r = \frac{t_{GPS}}{10^9} \bmod 604800 \tag{3}$$

for the Beidou system

$$t_r = \left(\frac{t_{GPS}}{10^9} \bmod 604800 \right) + 14^s \tag{4}$$

while for the GLONASS system

$$t_r = \left(\frac{t_{GPS}}{10^9} \bmod 86400 \right) + 10800^s + l_s \tag{5}$$

In equation (5), l^s denotes a leap second. Since 2018 $l^s = 16^s$.

An important limitation in the use of observations obtained using mobile devices is the problem of the so-called "duty cycle." It is a way to save energy. It involves turning satellite tracking by the GNSS chipset on and off for 0.2^s and 0.8^s, respectively. This leads to the so-called "cycle slips", making it difficult or impossible to resolve cycle ambiguity in GNSS phase measurements [5]. Starting from Android version 9.0, it is possible to disable the "duty cycle" from the API level [7].

3. Software for obtaining measurement data

Currently, there are a number of applications that enable the acquisition of observational data from mobile devices. However, not all devices equipped with the appropriate Android version make it possible to obtain GNSS measurement data. The type of data also varies depending on the device model (Table 1).

Table 1 Availability of GNSS measurement data for selected smartphone models. (Source: https://developer.android.com/guide/topics/sensors/gnss)

Device model	Android system version	Navigation message	Receiver clock reading	Phase measurement	Availability of L5	Supported satellite systems
Xiaomi Mi8	8.1	yes	yes	yes	yes	GPS GLONASS Galileo Beidou QZSS

Device model	Android system version	Navigation message	Receiver clock reading	Phase measurement	Availability of L5	Supported satellite systems
Huawei P10	7.0	yes	yes	yes	no	GPS GLONASS Galileo Beidou QZSS
Samsung Galaxy S10	9.0	no	yes	no	no	GPS GLONASS Galileo
LG G7 ThinQ	8.0	no	yes	no	no	GPS GLONASS

Google provides the GNSSLogger application for recording observational data and software for analysing the acquired data called GNSS Analysis app. These tools are available with the source code via the GitHub repository, as part of the GPS Measurement Tools [9] project. GNSSLogger application (Fig. 2) saves code and phase observations to a text file in the format specified in the project documentation [9]. Currently, the application does not provide navigation data. The software also allows user to make basic analyses directly on the smartphone screen, such as:

- calculation of the location in real time using the method of least squares,
- visualisation of the difference between the location calculated by the application and that determined directly by the internal software of the GNSS chipset, along with its presentation on Google maps,
- GNSS signal strength chart,
- residual plots, related to the location indicated by the user, or the average location determined by the GNSS chipset.

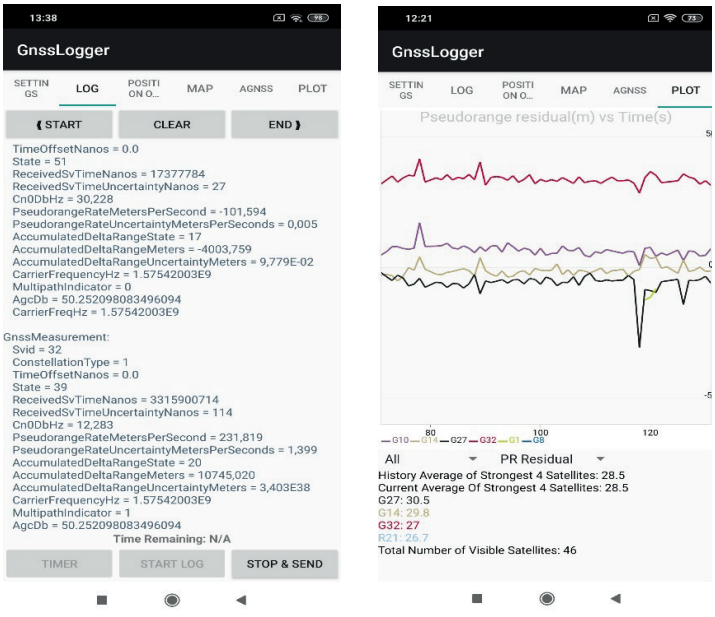


Fig. 2. Screenshots of the GNSSLogger software (Source: own study)

Observation data collected using the GNSSLogger application can be further analysed using the GNSS Analysis Tools set run on a desktop computer equipped with the Windows, Linux or Mac OS operating system. They are a set of MATLAB scripts, which can be executed in the MATLAB Runtime environment, available free of charge from MathWorks. GNSS Analysis Tools allows for interactive control of software operation parameters. The settings are: selection of the analysed satellites, setting the start and end time of the analysis, enabling/disabling modelling of ionospheric and tropospheric refraction, and choosing the method of determining the reference position used to calculate clock errors and the residuals for individual satellites. The software creates a number of plots for selected parameters (Fig. 3), including i.e.:

- the time plot of C/N0 (Carrier to Noise ratio) of all available satellites,
- zenithal projection showing the distribution of satellites above the horizon of the test point,
- pseudorange values as a function of time,
- graph of location components calculated with the method of least squares, related to the average location or the location defined by the user,
- pseudorange residual plot.

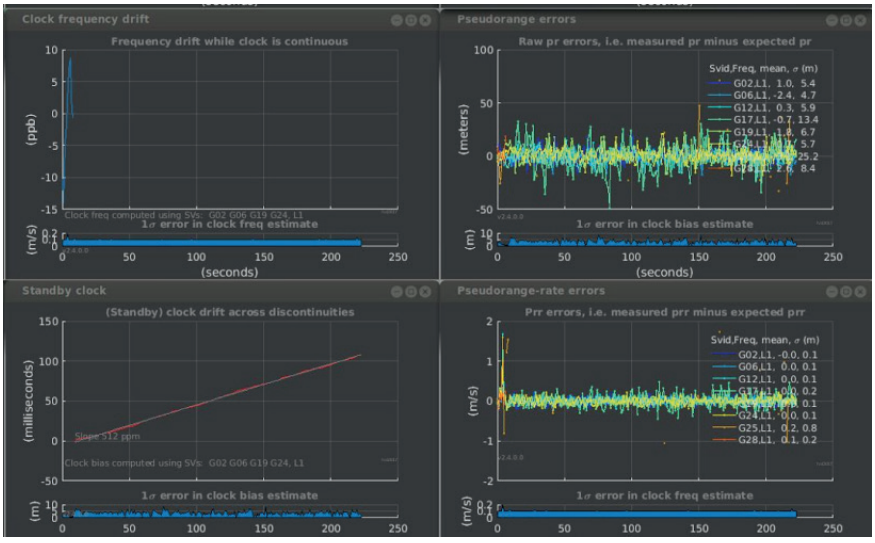
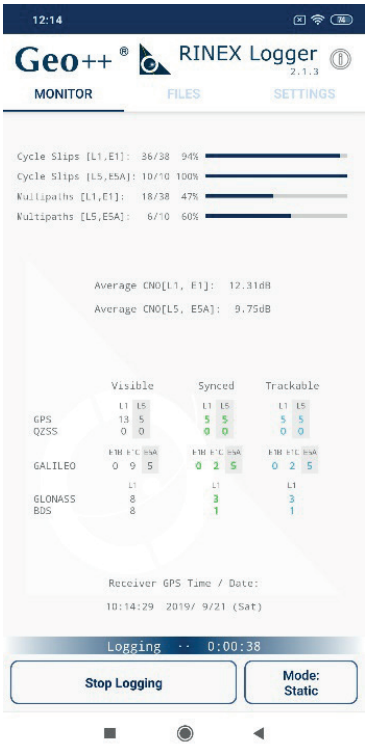


Fig. 3. Screenshots of GNSS Analysis Tools (Source: [10])

An interesting option offered by the application is a comparative analysis based on log files registered by GNSSLogger. Details on the installation and use of GNSS Analysis Tools software can be found in the manual [10]. A certain deficiency of GNSS Analysis Tools is the inability to save measurement data in the commonly used RINEX (Receiver Independent Exchange Format) data format [11]. This conversion is possible with Python scripts, provided by Rokubun in the GitHub repository [12]. These tools enable the conversion of GNSSLogger logs to the RINEX v. 3.0 format. Saving data in the RINEX format allows users to analyse observations with commonly used, free programmes such as RTKLib [13] and GLab [14], or commercial software [15].

Direct recording of measurement data in RINEX format is enabled by Geo++ RINEX Logger [16] and RinexOn applications from NSL (Nottingham Scientific Limited) [17]. Both programmes are available free of charge on Google Play. The application offered by Geo++ (Fig. 4a) enables recording of measurement data of GPS, GLONASS, Galileo, Beidou and QZSS (Quasi-Zenith Satellite System) systems. Supported frequencies are L1, L5, E1B, E1C, E5A. Users can choose to save data in RINEX 2.1 or 3.03 format. The data is saved in the device's internal memory in the Geopp_Rinex_logger folder, according to the RINEX file naming convention. The user has the option of setting information entries in the heading of the observation set. This includes, but is not limited to, the name of the test point, receiver and antenna type, and their numbers. One-hour sessions can be saved in individual files or appended to the recently created RINEX file. The programme does not output files with navigation data. This option is provided by RinexOn (Fig. 4b). This software creates navigation and observation files (GPS, GLONASS and Galileo) in the RINEX 3.03 format. RinexOn also has many more additional functions, such as: visualisation of the position of satellites, signal strength graphic, visualisation of the current position on Google Maps.

a)



b)

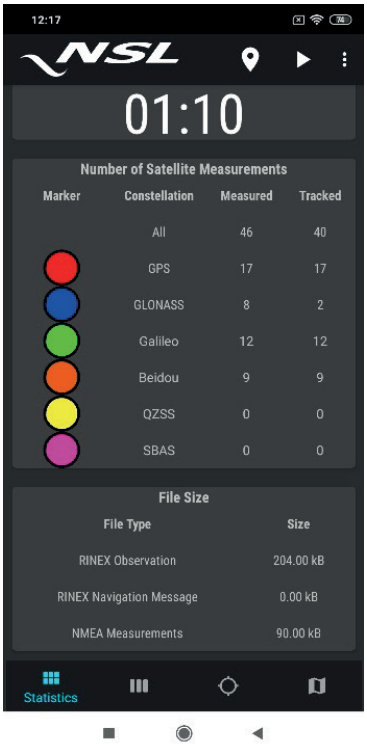


Fig. 4. Screenshots of a) Geo++ RINEX Logger, b) RinexOn application (Source: own study)

Manufacturers of Geo++ RINEX Logger and RinexOn do not provide source codes. An interesting programme which is available together with the source code and detailed documentation is GNSS Compare (Fig. 5). It was created by young programmers cooperating as Galfins Team [18]. The application won the Galileo Smartphone App Challenge, organised by ESA (European Space Agency) in 2017.

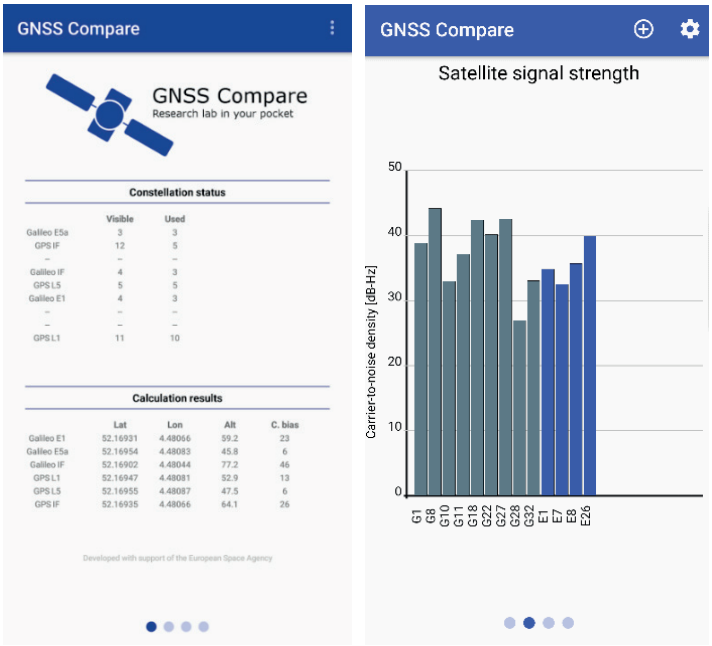


Fig. 5. Screenshots of the GNSS Compare application (Source: [18])

GNSS Compare was created for engineers and developers of navigation software as a tool for testing algorithms and hardware solutions. The position of the receiver can be calculated based on GPS, Galileo or GPS + Galileo code observations. The least squares or extended Kalman filter algorithms are available for calculating location. The programme does not support RINEX format, but outputs observation data in its own text format. The graphical interface provides visualisation of GNSS signal strength and location of the device on Google Maps. In addition, the programme presents the differences between the location determined by the GNSS chipset and that calculated on the basis of GNSS measurement data. External libraries, GoGPS [19], EJML (Efficient Java Matrix Library) [20] and GNSS Logger [9], were used in developing the software.

4. Review of published experimental works

Since the release of API 24, numerous works have been published containing descriptions of measurement tests and calculation experiments. They are intended to determine the degree of usability of the available GNSS measurement data. The use of phase methods, which were previously reserved for professional satellite receivers, is particularly interesting.

The work [5] describes a measurement experiment carried out using a Xiaomi Mi8 smartphone equipped with a Broadcom BCM47755 chipset that enables measurement on two frequencies: L1 and L5 for GPS and E1 and E5a for Galileo. The measurements were carried out at two points, one of which provided good measurement conditions, while the other was surrounded by tall buildings. The observations were recorded in a one-hour session, with a one-second recording interval, using RinexOn. Measurements were analysed using RTKLib software, in relation to the nearest permanent station. Static sessions were analysed only for the L1 frequency of the GPS system because the reference station did not record L5 and E5a

measurements. The coordinates of the point with good measurement conditions were determined with an RMS (Root Mean Square) error of 1.02 m for horizontal components and 0.5 m for height. 1.95 m and 7.82 m respectively were obtained for the point situated in a high-rise area.

Publication [21] describes test measurements carried out with the static and rapid static method using the Xiaomi Mi8 smartphone. Rapid static measurement was performed at 8 test points, the coordinates of which were previously determined with RTK (Real Time Kinematic) method, using a geodetic receiver. At each test point, 10 two-minute sessions were performed with a one--second recording interval. Geo++ Rinex Logger software was used to record the measurements. The measurement results were analysed with the RTKLib programme in relation to the WROC permanent station, located at a distance of about 60 m. Only a small number (approx. 10%) of recorded sessions contained a sufficient number of code and phase measurements to enable the calculation of point coordinates. The average differences in the point coordinates determined in relation to the reference value ranged from 0.2 – 4.0 m for the north component, 0.3 – 2.9 m for the east component and 0.3 – 16.5 m for the height. No solution was obtained for two test points. A 4-hour static session was also performed at one of the test points. Unfortunately, the Mi8 device did not register any phase measurements. Therefore, 4 static sessions were performed at the same point using a Huawei P10 smartphone, equipped with a Broadcom 4774 single frequency chipset. Differences in the point coordinates in relation to the reference location were 0.3 – 1.0 m for the north component and 0.5 – 5.0 m for the east component, while they did not exceed 4.0 m for height. The authors note that despite the clear horizon at the test point, mobile devices tracked 30 – 50% fewer satellites than the receiver at the nearby WROC permanent station.

The publication [7] presents tests using the PPP (Precise Point Positioning) method [3]. The experiment used a Xiaomi Mi8 smartphone, located on the roof of a building, in good observation conditions. The coordinates of the test point were determined by a geodetic receiver, based on a 12-hour static session. Observations were recorded using Geo++ RINEX Logger, with a one-second recording interval, during a 12-hour measurement session. The measurements were analysed using RTKLib software, relying on a solution based on the extended Kalman filter. The accuracy of determining a location in static PPP measurements compared to the results obtained with professional GNSS receivers is presented in Table 2.

Table 2. RMS error of location components determined using different GNSS receivers (Source: [22])

Source of measurement data	East component [cm]	North component [cm]	Height component [cm]
Smartphone with a dual-frequency receiver	21.8	4.1	11.0
Professional single-frequency receiver	14.6	25.8	27.1
Professional dual-frequency receiver	0.2	0.1	0.5

In the case of analysing measurements at one frequency (L1 and E1), no solution convergence was obtained. Basing on the publication [22], the authors determine that the solution convergence is obtained if the accuracy of the 3D location reaches 1 m and is maintained for a minimum of 20 consecutive measurement epochs. A measurement test in the PPP kinematic mode was also carried out. The test consisted in measuring 350 path points around a sports field. When measuring with a smartphone on two frequencies, horizontal deviations from the theoretical path exceeded 20 metres, while the deviations for measurement on one frequency were 3-5 meters, rarely exceeding 10 m.

5. Summary

Access to code and phase observations offers new perspectives for the use of mobile devices in precise positioning. The possibility of using phase measurements on two frequencies is particularly important. However, the currently obtained accuracy differs significantly from what professional geodetic-class receivers offer. In addition, the reliability of measurements leaves much to be desired, especially when they are taken at a point surrounded by field obstacles. Smartphone GNSS antennas use linear polarisation, showing susceptibility to multipath effects [5], [7]. In addition, the characteristics of phase centre variation of antennas used in mobile devices have not yet been determined. This is important when analysing differential measurements taken using different types of GNSS antennas. For this reason, it would be interesting to conduct differential phase measurements, recording them with identical devices. In this case, the effect of phase centre variation should be significantly reduced. As the authors of the quoted papers [7] [23] observe, measurements using smartphones are characterised by frequent occurrence of the so called "cycle slips." This problem has been the subject of numerous publications describing effective methods of detection and repair of "cycle slips" [24], [25] [26]. It should be borne in mind, however, that the use of advanced methods of detection and repair of cycle slips can lead to a significant increase in calculation time. Therefore, there is a need to create new algorithms to capture this problem, especially at the stage of an initial analysis of GNSS data. Intensive research work carried out in many scientific centres will probably lead to a significant improvement in the methods of positioning with smartphones. This will allow for a certain range of measurement tasks that previously required the use of expensive professional GNSS receivers to be taken over by smartphones. In addition, it is worth paying attention to the alternative possibilities of using smartphones, partly resulting from their additional equipment, enabling communication via Internet connections. The paper [27] indicates the possibility of using two-frequency phase measurements recorded by smartphones to monitor the phenomenon of ionospheric refraction. This is one of the basic tasks of the GNSS permanent station network. Determination of refractive corrections is the core of algorithms for calculating differential corrections in RTN (Real Time Network) measurement systems made available to users [28].

In the coming years, we can expect rapid development of algorithms and measuring techniques adapted to the specifics of mobile devices. This will probably be due to the widespread availability of smartphones and great interest from scientific and research centres as well as potential users of such solutions.

6. Acknowledgements

This publication was prepared under the research subsidy no. 16.16.150.545.

References

[1] Realini E. et al., "Precise GNSS Positioning Using Smart Devices", *Sensors,* vol. 17 (2017), . https://doi.org/10.3390/s17102434

[2] Lachapelle G. et al., "Evaluation of a Low Cost Hand Held Unit with GNSS Raw Data Capability and Comparison with an Android Smartphone", *Sensors,* vol. 18 (2018). https://doi.org/10.3390/s18124185

[3] Hofmann-Wellenhof B. et al., *GNSS – Global Navigation Satellite Systems: GPS, GLONASS, Galileo & more,* Vienna: Springer-Verlag, 2008.

[4] Maciuk K., "The application of GNSS system in logistics", *Budownictwo i Architektura,* vol. 17 (2018), pp. 181-188. https://doi.org/10.24358/Bud-Arch_18_173_13

[5] Robustelli U., Baiocchi V., Pugliano G., "Assessment of Dual Frequency GNSS Observations from a Xiaomi Mi 8 Android Smartphone and Positioning Performance Analysis", *Electronics,* vol. 8 (2019). https://doi.org/10.3390/electronics8010091

[6] *White Paper on using GNSS Raw Measurements on Android devices*. Prague, Czech Republic: European GNSS Agency, 2017. https://doi.org/10.2878/449581

[7] Wu Q. et al., "Precise Point Positioning Using Dual-Frequency GNSS Observations on Smartphone", *Sensors,* vol. 19 (2019). https://doi.org/10.3390/s19092189

[8] Seeber G. *Satellite Geodesy.* Berlin, New York: Walter De Gruyter, 2003.

[9] "GPS Measurement Tools". Available: https://github.com/google/gps-measurement-tools [Accessed: 11 September 2019]

[10] *GNSS Analysis Tools Installation Instructions and User Manual v. 2.6.3.0*, 18.09.2018. Available: https://github.com/google/gps-measurement-tools/releases/download/V2.6.3.0 [Accessed: 11 September 2019]

[11] "The Receiver Independent Exchange Format". Available: ftp://igs.org/pub/data/format/rinex303.pdf [Accessed: 11 September 2019]

[12] "Rokubun". Available: https://github.com/rokubun/android_rinex. [Accessed: 13 September 2019]

[13] Wiśniewski B., Bruniecki K., Moszyński M., "Evaluation of RTKLIB's Positioning Accuracy Using low-cost GNSS Receiver and ASG-EUPOS", *International Journal of Marine Navigation and Safety of Sea Transportation*, vol. 7 (2013), pp. 79-85. http://dx.doi.org/10.12716/1001.07.01.10

[14] Hernandez-Pajeras M. et al., "The ESA/UPC GNSS-Lab Tool (gLAB): An advanced multipurpose package for GNSS data processing", in 5th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing, Noordwijk 2010, 2010. http://dx.doi.org/10.1109/NAVITEC.2010.5708032

[15] Khaled M., Abdel M., "Comparison of GPS Commercial Software Packages to Processing Static Baseline up to 30 km", *ARPN Journal of Engineering and Applied Sciences,* vol. 10 (2015), pp. 10640-10650.

[16] "Geo++ RINEX Logger". Available: http://www.geopp.de/logging-of-gnss-raw-data-on-android. [Accessed: 11 September 2019]

[17] "Nottingham Scientific Limited". Available: https://www.flamingognss.com/rinexon [Accessed: 17 September 2019]

[18] "Galfins Team". Available: https://gnss-compare.readthedocs.io/en/latest/team.html [Accessed: 17 September 2019]

[19] "GoGPS Project". Available: http://www.gogps-project.org [Accessed: 11 September 2019]

[20] "Efficient Java Matrix Library". Available: https://ejml.org [Accessed: 11 September 2019]

[21] Wielgocka N., Hadaś T., „Czy to już możliwe?" *Geodeta* , vol. 4 (2019), pp. 8-12.

[22] Li P., Zhang X., "Integrating GPS and GLONASS to accelerate convergence and initialization times of precise point positioning", *GPS Solution,* vol. 18 (2014), pp. 461-471. https://doi.org/10.1007/s10291-013-0345-5

[23] Gogoi N. et al., "A Controlled-Environment Quality Assessment of Android GNSS Raw Measurements", *Electronics,* vol. 8 (2019). https://doi.org/10.3390/electronics8010005

[24] Xu G., *GPS Theory, Algorithms and Applications*. Berlin Heidelberg New York: Springer, 2007.

[25] Liu Z., "A new automated cycle slip detection and repair method for a single dual-frequency GPS receiver", *Journal of Geodesy*, vol. 85 (2011), pp. 171-183. https://doi.org/10.1007/s00190-010-0426-y

[26] De Lacy M. C. et al., "The Bayesian detection of discontinuities in a polynomial regression and its application to the cycle-slip problem", *Journal of Geodesy* vol. 82 (2008), pp. 527–542. https://doi.org/10.1007/s00190-007-0203-8

[27] Pankratius V. et al., "Mobile crowd sensing in space weather monitoring: the mahali project", *IEEE Communications Magazine*, vol. 52 (2014), pp. 22-28. https://doi.org/10.1109/MCOM.2014.6871665

[28] Wei E. et al., "VRS Virtual Observations Generation Algorithm", *Journal of Global Positioning System,* vol. 5 (2006), pp. 76-81. https://doi.org/10.5081/jgps.5.1.76