

A multi-GNSS software-defined receiver: design, implementation, and performance benefits

Stefan Söderholm¹ · Mohammad Zahidul H. Bhuiyan¹ · Sarang Thombre¹ · Laura Ruotsalainen¹ · Heidi Kuusniemi¹

Received: 19 May 2015 / Accepted: 28 April 2016 / Published online: 18 May 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract Global navigation satellite systems (GNSSs) have been experiencing a rapid growth in recent years with the inclusion of Galileo and BeiDou navigation satellite systems. The existing GPS and GLONASS systems are also being modernized to better serve the current challenging applications under harsh signal conditions. Therefore, the research and development of GNSS receivers have been experiencing a new upsurge in view of multi-GNSS constellations. In this article, a multi-GNSS receiver design is presented in various processing stages for three different GNSS systems, namely, GPS, Galileo, and the Chinese BeiDou navigation satellite system (BDS). The developed multi-GNSS software-defined receiver performance is analyzed with real static data and utilizing a hardware signal simulator. The performance analysis is carried out for each individual system, and it is then compared against each possible multi-GNSS combination. The true multi-GNSS benefits are also highlighted via an urban scenario test carried out with the hardware signal simulator. In open sky tests, the horizontal 50 % error is approximately 3 m for GPS only, 1.8 to 2.8 m for combinations of any two systems, and 1.4 m when using GPS, Galileo, and BDS satellites. The vertical 50 % error reduces from 4.6 to 3.9 when using all the three systems compared to GPS only. In severe urban canyons, the position error for GPS only can be more than ten times larger, and the solution availability can be less than half of the availability for a multi-GNSS solution.

Keywords Multi-GNSS · BeiDou · Galileo · Software receiver · SDR · Performance analysis · Receiver architecture

1 Introduction

The United States Department of Defense (DoD) Navigation System using Timing And Ranging (NAVSTAR) global positioning system (GPS) [1] was declared fully operational in 1995 and has since then evolved to being the de facto standard for satellite navigation systems. GLObalnaja NAVigatsionnaja Sputnikovaja Sistema (GLONASS) was developed in parallel with GPS, but was allowed to deteriorate drastically. Today, its value can hardly be overestimated since it offers an almost complete constellation of modern satellites, and it is also truly global. Unfortunately, still today, it only offers frequency division multiple access (FDMA) modulated signals [2], and thus, a relatively large bandwidth is required to receive all the signals. The European Galileo system is currently in its initial operation capability (IOC) phase with 12 satellites in orbit. The last two satellites were launched in December 2015, and two additional satellites are planned to be launched in May 2016. Two of the satellites that have been launched are on wrong orbits [3]. Though initially planned to be available already in 2010 [4], initial Galileo services are scheduled now to begin within the next year, and Galileo will become a truly global system by the end of this decade [5]. The BDS [6] consists of a mixed space constellation that has, when fully operational, five geostationary Earth orbit (GEO), twenty seven medium Earth orbit (MEO), and three inclined geo-synchronous satellite orbit (IGSO) satellites. The ground tracks of all BDS satellites are shown in Fig. 1. The GEO satellites are operating in orbit at an altitude of 35,786 km and positioned at 58.75° E, 80° E, 110.5° E, 140° E, and 160° E, respectively. The MEO

✉ Stefan Söderholm
stefan.soderholm@nls.fi

¹ Department of Navigation and Positioning, Finnish Geospatial Research Institute, National Land Survey, Geodeetinrinne 2, FIN-02430 Kirkkonummi, Finland

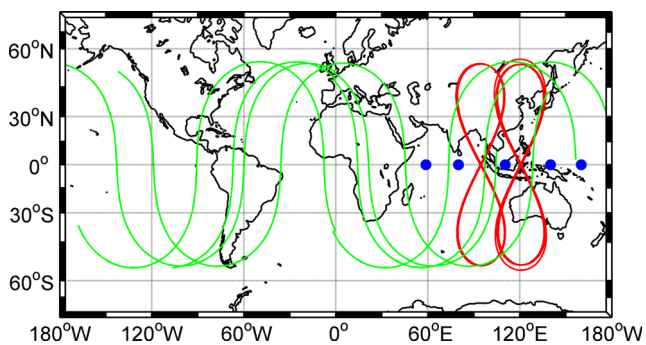


Fig. 1 Ground track of BDS satellites. GEO = blue dots, IGSO = red, MEO = green

satellites are operating in orbit at an altitude of 21,528 km with an inclination of 55° to the equatorial plane, whereas the IGSO satellites are operating in orbit at an altitude of 35,786 km with an inclination of 55° to the equatorial plane.

As of March 2015, the BDS has five GEO, five IGSO, and four MEO satellites [7]. These satellites broadcast navigation signals and messages within three frequency bands (termed as B1 in 1561.098 MHz, B2 in 1207.14 MHz, and B3 in 1268.52 MHz) using code division multiple access modulation (CDMA). As of now, the interface control document (ICD) was released only for B1 and B2 frequencies [6]. Considering the civilian CDMA only-modulated signals on the L1 band offered by three different systems, i.e., GPS, Galileo, and BDS, there is currently a total of 50 satellites orbiting around the world [8]. The minimum number of visible satellites in any place on earth during a 24-h period in February 2015 from these three systems is shown in Fig. 2.

Other regional systems like quasi zenith satellite system (QZSS), Indian regional navigation satellite system (IRNSS), GPS aided augmented navigation (GAGAN), and space-based augmentation system (SBAS) in general offer only a few additional satellites, and the satellites are often targeted to improve performance for well-defined areas rather than globally.

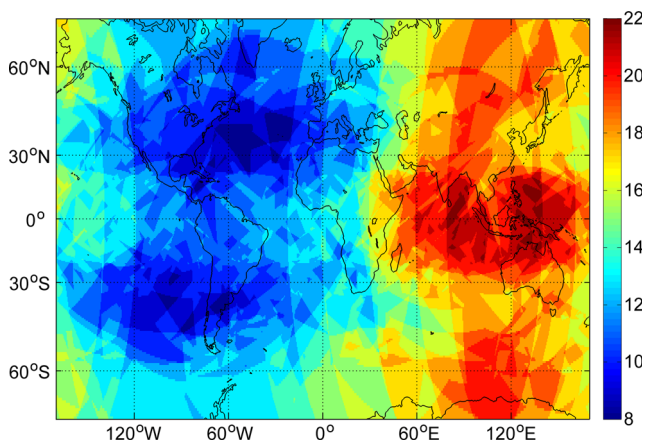


Fig. 2 Example of minimum number of GPS + Galileo + BDS satellites combined visible on Earth during a 24-h period in February 2015

The development of modern software-defined receivers (SDR) for GNSS signals can be considered to have initiated with the dissertation work by Akos [9] at the University of Colorado. He presented the design and the architecture of a GPS/Galileo SDR receiver in Matlab with test results for GPS signals. Later, this receiver was converted into a real-time receiver implemented in C, the gpSrx [10, 11]. The Matlab version of the receiver was later documented as a book edited by Borre et al. [12]. The documented receiver was capable of performing all steps from signal acquisition to navigation utilizing GPS and Galileo signals. One of the co-authors of Dennis Akos [10, 11] later founded a company named NordNav Technologies, that developed the R30, a commercial 24 channel real-time SDR for L1 band capable of receiving GPS and Galileo E1 signals [13, 14].

The group at the University of Calgary also developed their own SDR that was first presented in 2004 [15]. At that time, it supported only GPS signals and utilized a front end called the GPS signal tap made by Accord Inc. The Institute of Geodesy and Navigation at the University FAF Munich was also one of the forerunners of GNSS SDR development with its own SDR named as ipexSR [16], which is a real-time SDR for a personal computer (PC) platform running in Windows operating system (OS). The receiver was capable of receiving three GPS frequencies in L1, L2, and L5 bands in addition to the signals broadcasted from Galileo's test satellites GIOVE A and GIOVE B. Several groups also presented their contribution on the development of multi-frequency SDRs for GPS and Galileo during the last decade [17–21].

Later at the end of the last decade when the GLONASS signals again provided a reliable complementary system to GPS, several research groups around the world started working on integrating GPS and GLONASS. A group in Italy [22] presented a solution with the universal software radio peripheral (USRP) front end where they sampled wide bandwidth data and divided the data into separate channels for GPS and GLONASS using two down converters. They presented a performance analysis for combined GPS and GLONASS observations. In [23], Ferreira presented a GPS/Galileo/GLONASS SDR with a major emphasis on the configurability of the receiver. The focus of the paper was on the developed hardware designed for sampling the data from the front end and streaming it over the Ethernet to the computer where the signal processing was done in software. GPS + Galileo signal compatibility was shown, but no GLONASS signals were acquired successfully. Also, the presented architecture was not designed to simultaneously receive GLONASS and GPS/Galileo signals. In general, there has not been much literature published with details on the architecture of a multi-frequency multi-GNSS SDR from the viewpoints of implementation and scalability with respect to the growing number of global/regional satellite navigation systems. To this end, the authors in this paper present a highly scalable and configurable multi-

frequency multi-GNSS SDR architecture that has been under continuous implementation for the last two years at Finnish Geospatial Research Institute (FGI), Finland.

The most apparent advantage of a multi-GNSS receiver is the availability of a greater number of signals than before. The increased number of observations will increase robustness and availability of the position solution as well as offer a better accuracy for the user in certain scenarios. In open sky conditions, this advantage is often not obvious. But especially in urban canyons, where the user is surrounded by high buildings, the total number of available satellites becomes a critical factor. Another perhaps less apparent advantage is the robustness against interference when multiple frequencies are considered from multi-GNSS. A good overview of different kinds of GNSS interference and the options to mitigate them was well described in [24–29].

In this paper, we will first present the design of the implemented multi-GNSS software-defined receiver, then show the experimental results, and finally end with conclusions and future work. The design includes the description of the data flow and functional blocks in the receiver. Three features—the parameter system, assisted GNSS, and multi-correlator tracking—are described in more detail. Next, a description is given of how the acquisition, tracking, data decoding, and positioning can be implemented in a multi-GNSS receiver. The tracking architecture is described in detail together with how the position, velocity, and time are obtained in a multi-GNSS receiver. The time differences between the GNSS systems are indicated with some experimental results. Results from an open sky test case and a simulated urban canyon test case using a GNSS signal simulator are presented in the “[Experimental results](#)” section. The focus of the test cases is to compare the performance of a multi-GNSS solution with a GPS-only solution.

2 Multi-GNSS receiver design

A software-defined multi-GNSS receiver platform, named as FGI-GSRx, has been developed at FGI during the past years. The FGI-GSRx multi-GNSS receiver is mainly a Matlab-based research platform for the analysis and validation of novel algorithms for an optimized GNSS navigation performance. The first version of FGI-GSRx was based on an open-source software receiver platform developed by Borre et al. [12]. Since the receiver by Borre et al. was not originally designed for multi-GNSS operation, the authors have been modifying this receiver significantly to support more GNSS systems simultaneously and to make the receiver more configurable.

The receiver is implemented in Matlab and thus provides a unique and easy-to-use platform for the various research projects at FGI. The receiver is designed for post-processing operation, and it does not support real-time operation. The receiver architecture has been designed so that the intermediate

data from acquisition and tracking can be saved, and the processing can be started from any pre-saved data file. A block diagram of the receiver is shown in Fig. 3.

User parameters are read from the file system together with optional receiver independent exchange format (RINEX) navigation files [30] for assisted GNSS functionality. The parameters specify how the processing of the intermediate frequency (IF) data that has been logged before with the radio frequency (RF) front end will be processed. If requested by the user, the acquisition is executed using the IF data stored in the file system, and the results are stored to the memory on the computer. Optionally, already stored acquisition data can be retrieved from the file system, and the acquisition is bypassed. The result is the same regardless of which approach the user takes; the acquisition output is passed on to the tracking stage.

The same options are available for tracking. Either we process the logged IF data and store the output to the file system or we use already processed data from the file system. The result from tracking is then passed on to navigation.

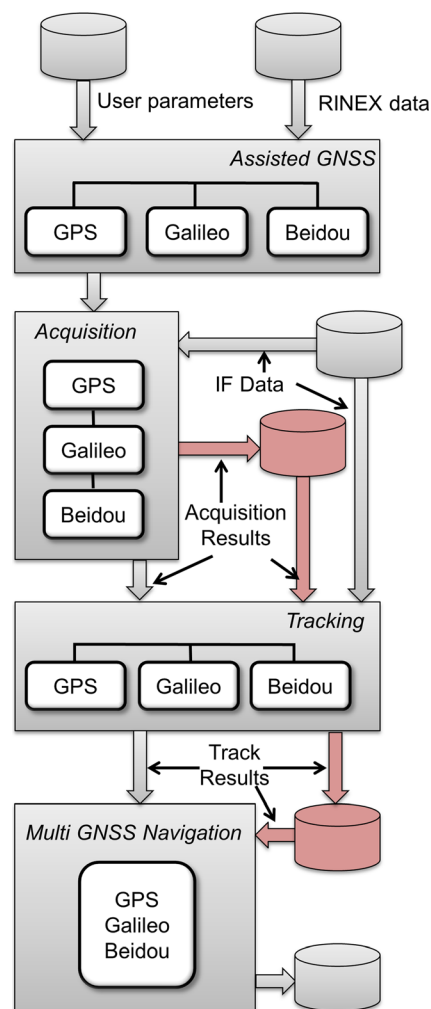


Fig. 3 Functional blocks in the FGI-GSRx receiver. The parts in red indicate the option to use pre-stored output from acquisition and tracking

2.1 Parameter system

Modifying parameters and configuring the receiver is one of the most relevant parts of a receiver intended for research. The goal is to enable all researchers to independently develop and test new algorithms in tracking, acquisition, and navigation. In the FGI-GSRx, we have implemented a parametric system based on text files. This approach makes it very easy to test different algorithms with many different data sets quickly and in a way that can be easily reproduced later. All default values are in a default parameter file, and each user can have multiple personal parameter files containing only parameters that differ from the default values. Therefore, changing and adding parameters for development purposes do not require any changes to the actual receiver code.

2.2 Assisted GNSS

To aid the acquisition process, an assisted GNSS functionality has been added to the receiver. The approximate user position, time, and receiver intermediate frequency for each front end can be provided as input parameters to the receiver. In addition, the ephemeris assistance can be provided either in the RINEX navigation file format [30] or previously saved already decoded broadcast ephemeris files. Utilizing this information, we can estimate the Doppler frequencies of each visible satellite and narrow our frequency search window to speed up the acquisition process.

2.3 Multi-GNSS acquisition

Some suitable algorithms for GNSS signal acquisition can be found in [31] and [32]. At the moment, the acquisition in the FGI-GSRx is done for one GNSS system at a time. Part of the future work is to investigate how signals from different systems should be prioritized at the acquisition stage. The basic algorithm for the acquisition of the GPS and Galileo signals is the same [11], where a fast Fourier transform (FFT)-based parallel code phase search is utilized. The implemented acquisition algorithms to acquire BDS MEO, GEO, and IGSO satellites are discussed in detail in [33] and [34].

The search window, number of coherent and non-coherent integration rounds, and signal thresholds are all configurable parameters. Two different acquisition modes exist, an unaided and an aided acquisition. In the aided mode, only one frequency bin is used, and the threshold can be set separately.

For the Galileo E1 data/pilot channel, both the spreading code and the data bit are 4-ms long so if we want to do longer coherent integrations than 4 ms, we need to take into account the possibility that a bit transition may occur between any two consecutive epochs. In the FGI-GSRx, we can use 8- and 12-ms coherent integration, and we can then search over all combinations of data-bits. For example, for 8-ms coherent

integration, the two data-bits may take any of the following values [+1; +1], [-1; +1], [+1; -1], and [-1; -1].

2.4 Multi-GNSS tracking

Not many good tracking architectures have been presented for GNSS signals in literature. A good architecture can however be found in [35]. Similarly, in the FGI-GSRx presented herein, after acquisition has been completed for all systems, bit edge detection is performed for each satellite signal, and each signal with a detected bit edge will be assigned to a specific tracking channel. Tracking is then initiated with a correlation interval of code length duration for each individual system (i.e., 1 ms for GPS and BDS, 4 ms for Galileo), as shown in Fig. 4.

The tracking of GPS and BDS is done for every ms of data, whereas the Galileo signals are processed only for every 4 ms of data due to the different lengths of one code epoch. The actual amount of data read from the file is adjusted for each epoch based on the true code frequency so that we always aim to process exactly one code epoch. Essentially, we are trying to keep the code phase as close to zero as possible.

The tracking architecture has been designed to be highly configurable with good support for different tracking modes. Each logical unit, such as the discriminators, loop filters, etc., is separated into its own functions. Each logical unit is linked to a certain type; for example, GPS_FREQ_LOOP is used as the frequency-locked loop (FLL) of GPS. Each type can also be in many different states; for example, FREQ_LOOP_PULL_1 can be used in the initial stage of carrier tracking with update rate of 1 ms. Each state also has a predefined update rate. An example of states, types, and functions is shown in Table 1.

As can be seen from Table 1, one type can refer to many different functions and update rates, but each function and update rate is linked to only one state. Depending on what state that type is in, one specific function will be executed at one specific rate. When the state of a type changes, another function will be executed or the update rate will change. With this approach, we can easily switch between, for example, different kinds of discriminator or loop filter and manage the update rate of those functions. Utilizing this kind of an approach, we can easily accommodate for loop pull in, high sensitivity or high dynamic states of the tracking without the risk of unmanageable code.

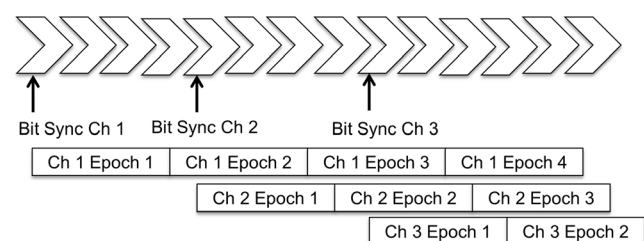


Fig. 4 Tracking is synchronized to the bit edge for each channel

Table 1 Example of how state, function, and type are related

| Type: | GPS_FLL_DISCR | | GPS_FREQ_LOOP | | GPS_DLL_DISCR | |
|--------------------|----------------|-----------|-------------------|-----------|----------------|-----------|
| | Function | Rate (ms) | Function | Rate (ms) | Function | Rate (ms) |
| FLL_DISCR_1 | mulFreqDiscrim | 1 | | | | |
| FLL_DISCR_5 | mulFreqDiscrim | 5 | | | | |
| FLL_DISCR_20 | mulFreqDiscrim | 20 | | | | |
| FREQ_LOOP_PULL_1 | | | gpsFreqLoopPullIn | 1 | | |
| FREQ_LOOP_TRACK_1 | | | gpsFreqLoop | 1 | | |
| FREQ_LOOP_TRACK_20 | | | gpsFreqLoop | 20 | | |
| DLL_DISCR_1 | | | | | mulCodeDiscrim | 1 |
| DLL_DISCR_5 | | | | | mulCodeDiscrim | 5 |
| DLL_DISCR_20 | | | | | mulCodeDiscrim | 20 |

Tracking states are defined as a set of states

2.5 Multi-correlator tracking

In the default configuration, only three correlator fingers are used for tracking and one finger for monitoring the noise level. The finger spacing is configurable, and the default value is 0.25 chips. The FGI-GSRx has a feature called multi-correlator tracking where the user can specify the number of fingers used, fingers’ spacing, and the output rate. This feature is intended for analyzing the channel characteristics in more detail. A typical multi-correlation output for a code delay window of ±2 chips is shown in Fig. 5.

2.6 Data decoding

The sign of the prompt finger is copied into the data decoding buffer, and when the buffer is full, the FGI-GSRx correlates the incoming bit stream with the up-sampled data frame preamble for the respective system. After successful correlation, the start of a data frame is found, and the raw data-bits can be extracted from the signal. The GPS bits are passed through a parity check, and the Galileo bits are de-interleaved and passed through a Viterbi decoder. The final decoded data contains the transmission time for the beginning of the data frame for each channel. Since we know at what sample the data frame started, we can link the transmission time for each signal to a specific sample count.

2.7 Position, time, and velocity solution

The position, time, and velocity can be calculated after tracking has been completed successfully. The input is the decoded data and the measurements for each channel from the tracking engine. The measurements from each channel are aligned with the bit edge of that channel. In

order to have synchronized measurements, we need to realign our observations from the tracking. The decoded data frame in each channel n will provide the transmission time, T_n , for the sample, S_n , that the receiver acquired at the beginning of the frame. In order to obtain synchronization, we have to extrapolate the transmission time for all channels to one common sample, S_0 as shown in Fig. 6.

The measured transmission times for all channels T_n refer to the same sample S_0 in our incoming data. It is worth noting here that since these transmission times are extrapolated from the time stamp in the data frames for each channel, they are in different time domains, namely Galileo standard time (GST), BDS system time (BDT), and GPS time depending on what signal occupies that channel.

To obtain the initial receiver time estimate, T_{rx} , at sample S_0 , we assume that the signal with the shortest traveling distance for each system has traveled for 80 ms. The accuracy of this receiver time estimate is not critical for the position solution, and our time solution will give us the final accurate receiver time. The estimated receiver time is a vector with three components:

$$T_{rx} = \begin{bmatrix} T_{rx}^G \\ T_{rx}^E \\ T_{rx}^B \end{bmatrix} \tag{1}$$

i.e., the estimated receiver time in GST, BDT, and GPS time for the same sample S_0 . The pseudo-ranges ρ_n^k can then be calculated as

$$\rho_n^k = (T_{rx}^k - T_n^k) * c \tag{2}$$

where n is the channel index, k the system index (GPS, Galileo, or BDS), and c is the speed of light.

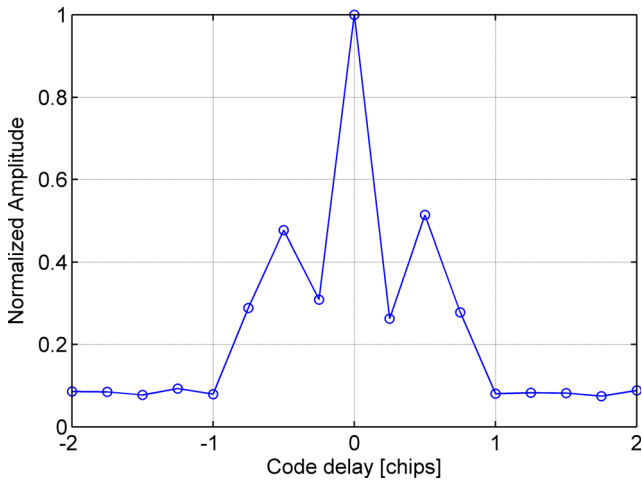


Fig. 5 Example of Galileo E1 signal correlation peak with 17 fingers with 0.25 chip spacing (dots). Blue line is an interpolation to illustrate the peak. Data have been integrated over 200 ms

2.7.1 Position solution

Using an a priori estimate for the user position, $\mathbf{Pos} = [x_0 y_0 z_0]$, and the decoded ephemeris, we can calculate the satellite positions and the predicted range between the user and each satellite, r_i^k , and form the observation matrix with the observed values minus the predicted ones

$$\Delta\rho = \begin{bmatrix} \rho_1^G - r_1^G \\ \vdots \\ \rho_n^G - r_n^G \\ \rho_1^E - r_1^E \\ \vdots \\ \rho_m^E - r_m^E \\ \rho_1^B - r_1^B \\ \vdots \\ \rho_k^B - r_k^B \end{bmatrix} \quad (3)$$

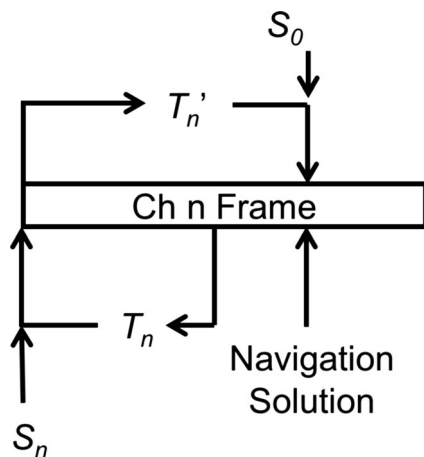


Fig. 6 Transmission time for channel n , T_n , is extrapolated to the sample at which we will calculate our navigation solution. Note here that the T_n ’ is channel-specific and system-dependent

The observation vector is identical regardless of the number of systems we have. For the geometry matrix \mathbf{H} containing the directional cosines, we need to add one clock term for each enabled system. One row in \mathbf{H} can be therefore written as

$$H_{i,j} = \begin{bmatrix} \Delta x_i^j / r_i^j & \Delta y_i^j / r_i^j & \Delta z_i^j / r_i^j \end{bmatrix} \begin{cases} 1, & j = G \\ 0, & j \neq G \end{cases} \begin{cases} 1, & j = E \\ 0, & j \neq E \end{cases} \begin{cases} 1, & j = B \\ 0, & j \neq B \end{cases} \quad (4)$$

where Δx , Δy , and Δz are the differences between the satellite coordinates and the a priori user coordinates, r is the range to the satellite, and the prefixes i indicate satellite number and suffixes j indicate satellite system (G—GPS, E—Galileo, and B—BDS). To obtain the updates to the a priori user position, $\Delta\mathbf{Pos}$, we need to solve a set of normal equations

$$\Delta\rho = H\Delta\mathbf{Pos} + \Delta\varepsilon_\rho \quad (5)$$

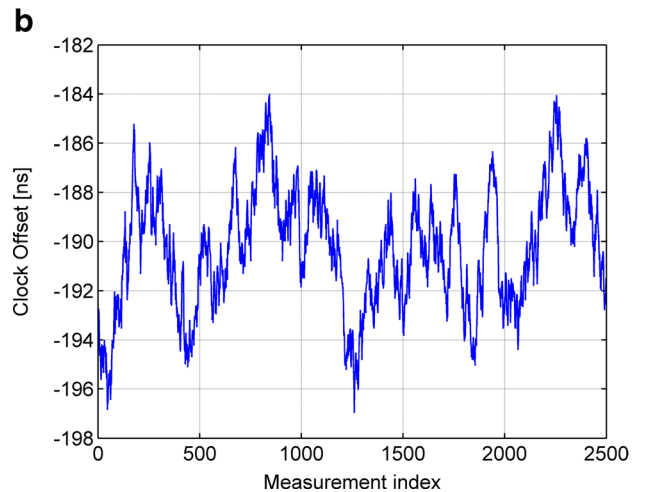
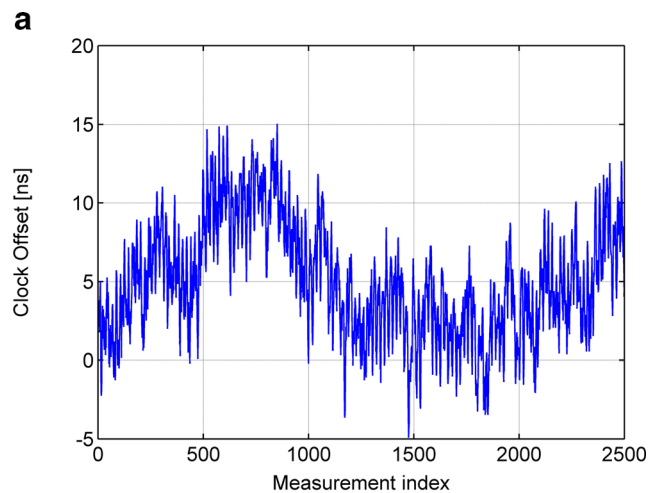


Fig. 7 **a** GPS—Galileo clock offset in ns. **b** GPS—BDS clock offset in ns. The leap second difference between GPS and BDS of 14 s has been removed

Table 2 Configurations for NSL front ends

| Properties | MAX2769B | MAX2112 |
|------------------------------|----------|--------------------|
| Center frequency (MHz) | 1561.098 | 1575.42 |
| 3-dB bandwidth (MHz) | ~4.2 | ~6.6 |
| Max sampling frequency (MHz) | 40 | 30 |
| Reference frequency (MHz) | 26 | 26 |
| Received signal | BDS B1 | GPS L1, Galileo E1 |

where $\Delta\varepsilon_p$ is assumed to be a zero mean residual vector. The least squares solution to (5) can be written as

$$\Delta\text{Pos} = (H^T H)^{-1} H^T \Delta\rho \tag{6}$$

The solution in the multi-GNSS case can then be written as

$$\Delta\text{Pos} = [\Delta x \quad \Delta y \quad \Delta z \quad c\Delta t^G \quad c\Delta t^E \quad c\Delta t^B] \tag{7}$$

where Δt are the clock offsets for the three systems and c is the speed of light. When the a priori user estimate has been corrected by the output of (7), we repeat the steps (3) to (7) until the change in the estimate is sufficiently small.

2.7.2 Time solution

We start by aligning all the measurements to one single sample count, S_0 , in the original data file (see Fig. 6), and then, we assume that the receiver time at this sample is T_{rx} . Note that the measurements from all systems are aligned to the same sample count, but the receiver time is a vector (1), with the time in each system separately. The navigation solution will provide us with the clock offset, Δt , for each system (7), and we can accurately determine the true time for that particular sample count in each system’s time domain by correcting the initial estimate with Δt

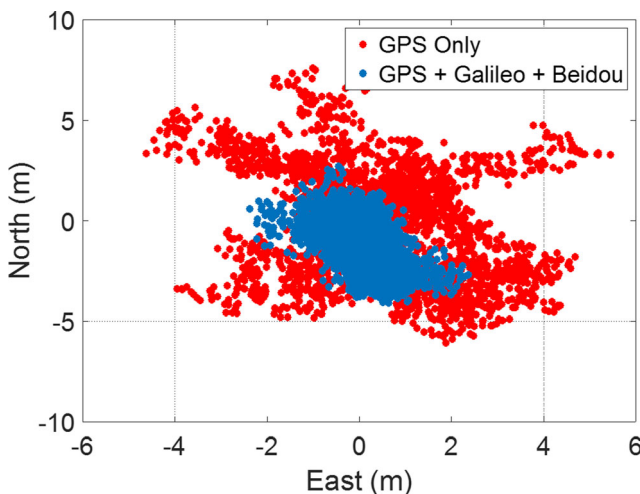


Fig. 8 Ground plot of GPS (6 satellites) solution versus GPS + Galileo + BDS (14 satellites)

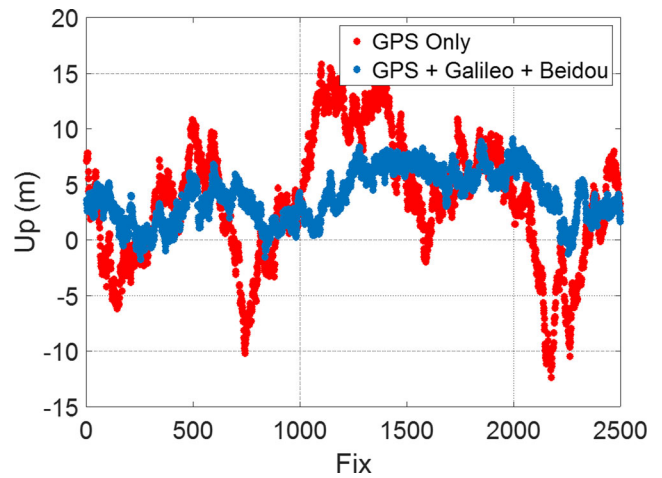


Fig. 9 Altitude error of GPS (6 satellites) solution versus GPS + Galileo + BDS (14 satellites)

$$T_{rx}^{\text{true}} = \begin{bmatrix} T_{rx}^G - \Delta t^G \\ T_{rx}^E - \Delta t^E \\ T_{rx}^B - \Delta t^B \end{bmatrix} \tag{8}$$

The GPS time is semi-synchronized to coordinated universal time (UTC) time in such a way that the time difference is defined as

$$\text{UTC} - \text{GPS time} = -\text{leap}_G + C_0 \tag{9}$$

where leap_G is the number of leap seconds specified for a particular time and date. At the time of writing, the number of leap seconds for GPS was 16. The value of the constant C_0 is continuously monitored by the GPS ground segment, and parameters for a UTC model are broadcasted as a part of the GPS almanacs. The value of C_0 is targeted to be less than 1 μs , but it is typically less than 100 ns.

GST is defined in a very similar fashion

$$\text{UTC} - \text{GST} = -\text{leap}_E + C_1 \tag{10}$$

The number of leap seconds is the same for Galileo as for GPS and the difference between C_0 and C_1 is typically less than 50 ns. BDS time is defined as

$$\text{UTC} - \text{BDT} = -\text{leap}_B + C_2 \tag{11}$$

The value of the constant C_2 is kept less than 100 ns, and for BDS the number of leap seconds is 2. An example of time domain differences is shown in Fig. 7a, b.

As is shown in Fig. 7b, the difference between the two constants C_0 and C_2 for this test was 190 ns.

2.7.3 Velocity solution

The velocity solution is calculated similarly as the position solution. The observation matrix is in this case the difference

Table 3 Horizontal statistics for various combinations of GNSS

| Configuration | Nr. of solutions | Nr. of sat | 50 % error (m) | East offset (m) | North offset (m) | StDev (m) | Max error (m) | HDOP |
|---------------------|------------------|------------|----------------|-----------------|------------------|-----------|---------------|------|
| GPS | 2500 | 6 | 2.98 | 0.48 | -0.06 | 1.51 | 7.67 | 2.25 |
| Galileo | 2500 | 4 | 4.31 | 0.7 | -4.05 | 2.25 | 11.56 | 2.31 |
| BDS | 2500 | 4 | 3.41 | -0.14 | -2.9 | 1.96 | 9.52 | 4.77 |
| GPS + Galileo | 2500 | 10 | 1.78 | 0.41 | -1.28 | 1.24 | 6.18 | 1.44 |
| GPS + BDS | 2500 | 10 | 2.28 | -0.4 | -0.26 | 1.26 | 6.42 | 1.8 |
| Galileo + BDS | 2500 | 8 | 2.78 | -0.02 | -2.57 | 1.3 | 6.62 | 1.94 |
| GPS + Galileo + BDS | 2500 | 14 | 1.38 | 0.04 | -1.2 | 0.93 | 4.16 | 1.34 |

between the measured Doppler frequency obtained directly from the phase lock loop (PLL) and the theoretical Doppler calculated from the a priori user velocity and the satellite ephemeris. The geometry matrix \mathbf{H} (4) is the same as for the position calculation, and the solution is obtained using (5)–(7). Instead of obtaining a position solution, we obtain a velocity solution as

$$\Delta\text{Vel} = \begin{bmatrix} \Delta v_x & \Delta v_y & \Delta v_z & c/L_G \Delta f^G & c/L_E \Delta f^E & c/L_B \Delta f^B \end{bmatrix} \quad (12)$$

where c is the speed of light, L is the center frequency for each system, and Δf is the frequency offset from the nominal intermediate frequency. After iteration, we obtain the true user velocity and the frequency offsets for each system.

At the end of the processing, some additional operations such as coordinate transformations, time corrections, and satellite elevation and azimuth angles are calculated. The satellite elevation is used after the initial position estimate to omit satellites below a user defined cutoff angle. The update rate for the navigation is defined by the user. The default rate is every 20 ms, i.e. 50 Hz.

3 Experimental results

Data was logged with the Nottingham Scientific Limited (NSL) Stereo Software GNSS front end [36]. One of its front

ends uses the Maxim MAX2769B radio chip, and the other one uses the Maxim MAX2112 radio chip. The key configuration parameters for these radios are listed in Table 2.

3.1 Static open sky test with live data

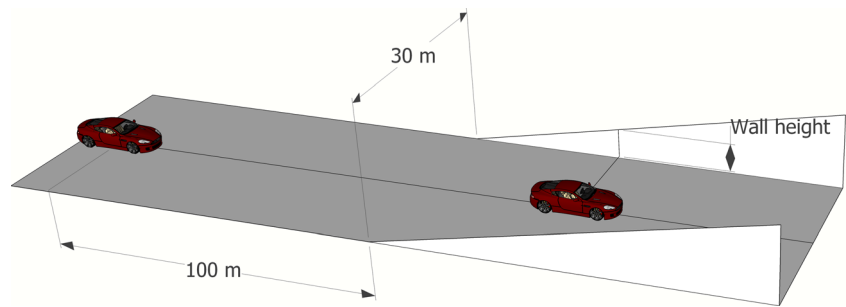
The first test with the multi-GNSS FGI-GSRx receiver was carried out with the roof antenna signal at FGI in Kirkkonummi, Finland. The antenna used in the test was the G5Ant-3AT1 active antenna by Antcom [37]. A suitable time for the test was selected so that a minimum of four satellites were visible from all the three systems; 100 s of data was logged, and the position was calculated at a rate of 50 Hz. The ground plot is shown in Fig. 8. The altitude variations for the GPS-only and multi-GNSS solutions are also shown in Fig. 9.

Additional analysis was performed with the various combinations of satellite systems, and the performance metrics are presented in Tables 3 and 4. Even if BDS and Galileo only can add four satellites each to the multi-GNSS solution, we can still clearly see from the figures and the tables that the accuracy is better for a multi-GNSS solution compared to a GPS-only solution with this particular data set. The offsets vary somewhat for the three systems in both the horizontal and the vertical directions (East, North, and Up offsets). This is partly due to the fact that we have used default values for the ionospheric corrections. The same default parameters were used for all the systems, but the remaining errors in the observations affect the systems differently due to the different

Table 4 Vertical statistics for various combinations of GNSS

| Configuration | Nr. of solutions | Nr. of sat | 50 % error (m) | Up offset (m) | StDev (m) | Max error (m) | VDOP |
|---------------------|------------------|------------|----------------|---------------|-----------|---------------|------|
| GPS | 2500 | 6 | 4.6 | 3.49 | 3.78 | 15.81 | 2.41 |
| Galileo | 2500 | 4 | 2.49 | 0.68 | 2.15 | 11.22 | 3.32 |
| BDS | 2500 | 4 | 9.77 | 9.63 | 5.05 | 22.65 | 2.85 |
| GPS + Galileo | 2500 | 10 | 3.31 | 2.73 | 2.31 | 10.63 | 1.71 |
| GPS + BDS | 2500 | 10 | 3.43 | 3.56 | 3.07 | 11.22 | 1.47 |
| Galileo + BDS | 2500 | 8 | 6.5 | 6.56 | 1.45 | 11.73 | 1.76 |
| GPS + Galileo + BDS | 2500 | 14 | 3.9 | 3.93 | 2.21 | 9.11 | 1.27 |

Fig. 10 Urban canyon for simulator test



horizontal dilution of precision (HDOP) and vertical dilution of precision (VDOP) values.

The horizontal 50 % error is approximately 3 m for GPS only and 1.4 m when using the additional Galileo and BDS satellites. We see an improvement even if the actual position offsets, especially in the north direction, are somewhat bigger for BDS and Galileo compared to that of GPS. If we look at the standard deviation with respect to mean position and the maximum error, we also clearly see an improvement by adding the observations from Galileo and BDS.

For the vertical component (Table 4), the offset is relatively large for the BDS-only solution, which is most likely due to the geometry. This affects the vertical offset of the multi-GNSS solution. However, all other statistical values improve in the same way as for the horizontal values. When looking at the Figs. 9 and 10, the improvement in both the horizontal and in the vertical component is clearly visible when we used a multi-GNSS solution.

3.2 Urban canyon test

The advantages of using multiple GNSSs in a receiver become more apparent in urban canyon and other blocked signal

environments. Unfortunately, due to the very few Galileo satellites and the limited number of BDS satellites visible in northern Europe, it would be very difficult to obtain any conclusive results using live signals. Therefore, we used instead a GNSS signal simulator capable of producing all the signals of interest (GPS, Galileo, and BDS). The simulator does not have any urban canyon capability. Hence, we generated such a scenario instead by artificially blocking out non-visible satellites. The simulator that was used was a Spectracom GSG 6 [38] with default satellite orbit data resulting in full constellation for all the three systems.

The urban canyon was generated by introducing identical walls of specific height on both sides of the receiver. The height of the walls was increased with a rate of 1 m for every 10 m, and for the first 100 m, there were no walls at all. The dimensions of the generated urban canyon are shown in Fig. 10.

As can be seen from Fig. 10, the user started at a distance of 100 m from the urban canyon and driving at a speed of 10 m/s quickly entered the canyon where the walls on both sides are continuously becoming higher with a rate of 1 m/s. The elevation angle of the top of the walls was calculated for each azimuthal angle, and any satellite that was blocked by a wall was disregarded from the navigation solution. The test is repeated for eight different directions of the canyon.

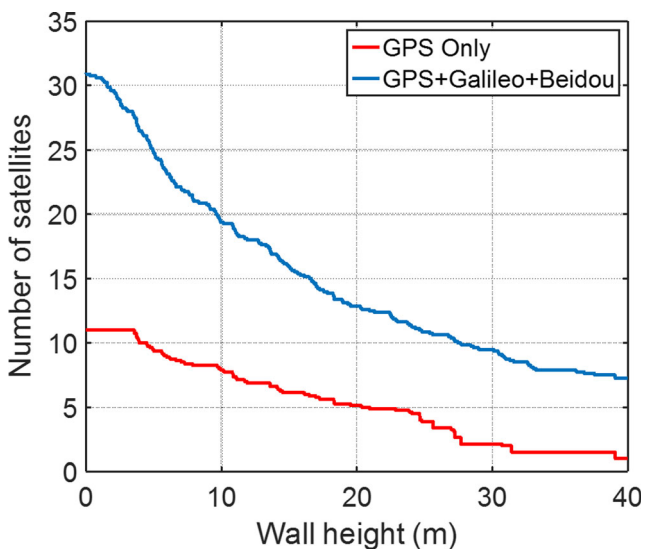


Fig. 11 Average number of satellites for GPS (red) and GPS + Galileo + BDS (blue)

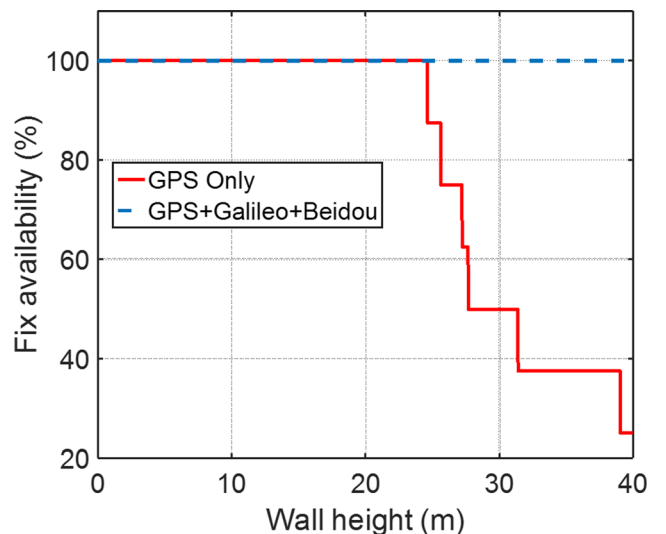


Fig. 12 Fix availability for GPS (red) and GPS + Galileo + BDS (blue)

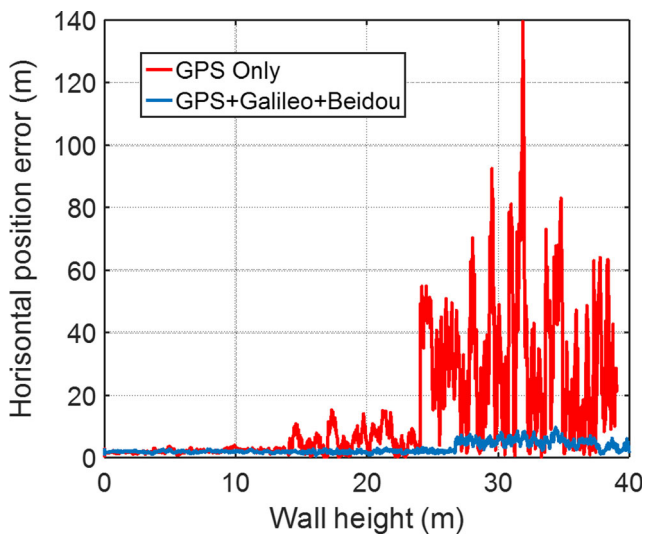


Fig. 13 Horizontal position error for urban canyon with azimuth of 225°

The result is that the number of available satellites becomes less and less and eventually the position will be lost. When fewer satellites are available the geometry becomes worse, affecting the quality of the position. Two options are compared, GPS only and GPS + BDS + Galileo. The average number of satellites over all for eight directions is shown in Fig. 11. The fix availability as a function of the height of the urban canyon wall is shown for GPS only and GPS + BDS + Galileo in Fig. 12.

Besides the low number of satellite visibility and the inevitable low number of navigation solution, i.e., “fix” availability for a GPS-only solution, the performance of the receiver will be worse when using GPS-only signals. The horizontal position errors for the two cases are shown in Figs. 13 and 14.

In the 225° case (Fig. 14), the GPS-only solution managed to maintain a position solution almost until the walls became 40-m high, but the position errors became larger than 100 m.

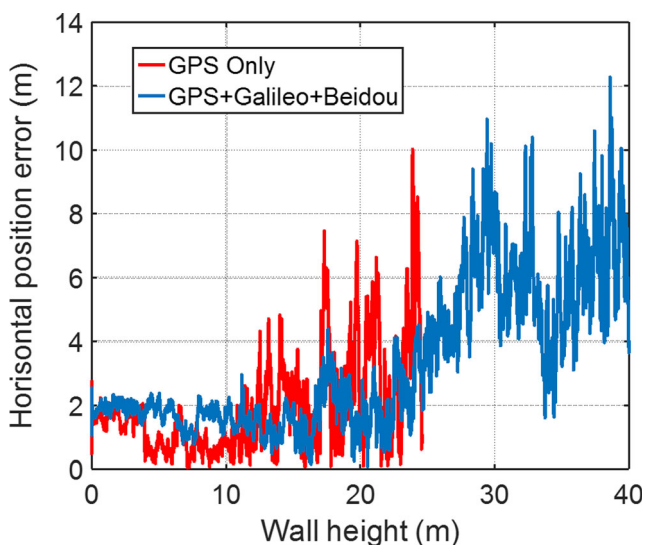


Fig. 14 Horizontal position error for urban canyon with azimuth of 315°

In the 315° case, the GPS-only solution lost its position fixes already much earlier, and the accuracy was degraded.

4 Conclusions

The new Matlab-based multi-GNSS software-defined receiver architecture developed at the Finnish Geospatial Research Institute was presented in this paper. The design of the receiver was described in detail in various processing stages and the impacts of this design when supporting multiple systems were explained. Finally, experimental results were presented where it was distinctively shown that the use of multiple systems simultaneously will result in improved performance, namely availability and accuracy, both in open sky conditions and in urban canyon environments. In the open sky tests, the horizontal 50 % error is approximately 3 m for GPS only, 1.8 to 2.8 m for combinations of any two systems, and 1.4 m when using GPS, Galileo, and BDS satellites. The vertical 50 % error reduces from 4.6 to 3.9 when using all the three systems compared to GPS only. In severe urban canyons, the position error for GPS only can be more than ten times bigger, and the fix availability can be less than half of the availability for a multi-GNSS solution.

Further development of the multi-GNSS receiver is planned, and the following main features will be developed in near future:

- Work on adding support for GLONASS signals has already been started, and this will continue until we have successfully added the fourth system to our receiver.
- A search unit that will contain the logic on how to optimize the search for signals from multiple systems. Some novel algorithms will be developed in this area.
- A Kalman navigation filter to replace the least square estimator (LSE) solution. The LSE solution will in the future be used for initialization only.
- Development of novel multi-GNSS receiver autonomous integrity monitoring (RAIM) algorithms for error detection and exclusion for mass-market-grade receivers.
- A fully integrated multi-GNSS tracking engine with support for all GNSS signals including advanced mode switching techniques. Switching between modes optimized for high dynamic tracking or high sensitivity tracking will improve the overall performance of the receiver significantly.
- In the current implementation, navigation is initiated only after tracking has completed. The plan is to perform navigation for each new measurement from the track engine. This will resemble a more real-time operation, and it will enable feedback from the navigation to the tracking. This makes it possible to integrate any deeply coupled inertial navigation system (INS) algorithm into the receiver.

Acknowledgments This research has been conducted within the projects DETERJAM (Detection analysis and risk management of satellite navigation jamming) funded by the Scientific Advisory Board for Defence of the Finnish Ministry of Defence and the Finnish Geodetic Institute/Finnish Geospatial Research Institute, Finland, and FINCOMPASS, funded by the Finnish Technology Agency TEKES with the Finnish Geodetic Institute/Finnish Geospatial Research Institute, Nokia Corporation, Roger-GPS Ltd., and Vaisala Ltd.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Global Positioning Systems Directorate (2012). Systems engineering & integration interface specification. <http://www.gps.gov/technical/icwg/IS-GPS-200G.pdf>. Accessed 6 May 2015
- Russian Institute of Space Device Engineering (2008). GLONASS interface control document ed. 5.1. <http://www.spacecorp.ru/upload/iblock/1c4/cgs-aaixymt%205.1%20ENG%20v%202014.02.18w.pdf>. Accessed 5 May 2015
- European Space Agency (2015). Galileo satellites—status update. http://www.esa.int/Our_Activities/Navigation/Galileo_satellites_status_update. Accessed 2 May 2016
- Pagny R, Dardelet J-C, Chenebault J (2005) From EGNOS to Galileo: a European vision of satellite-based radio navigation. *Ann Telecommun* 60(3):357–375. doi:10.1007/BF03219825
- European Union (2014) European GNSS open service. http://ec.europa.eu/enterprise/policies/satnav/pubconsult-2/files/galileo_os_sis_icd_v1-2-3_short_version270614_en.pdf. Accessed 5 May 2015
- China Satellite Navigation Office (2013) BeiDou navigation satellite system signal in space interface control document version 2.0. <http://www.beidou.gov.cn/attach/2013/12/26/20131226b8a6182fa73a4ab3a5f107f762283712.pdf>. Accessed 5 May 2015
- Min L, Lizhong Q, Qile Z, Jing G, Xing S, Xiaotao L (2014) Precise point positioning with the BeiDou navigation satellite system. *Sensors* 14:927–943. doi:10.3390/s140100927
- Montenbruck O, Steigenberger P, Khachikyan R, Weber G, Langley RB, Mervart L, Hugentobler U (2014) IGS-MGEX, preparing the ground for multi-constellation GNSS science. *Inside GNSS*, Jan/Feb. 42–49
- Akos DM (1997) A software radio approach to global navigation satellite system receiver design. A dissertation for the degree Doctor of Philosophy presented to The Faculty of the Fritz J. and Dolores H. R., College of Engineering and Technology, Ohio University
- Akos DM, Normark PL, Hansson A, Rosenlind A, Ståhlberg C, Svensson F (2001) Global positioning system software receiver (GPSrx) implementation in low cost/power programmable processors. In: Proceedings of ION 2001, September 11–14, 2851–2858, Salt Lake
- Akos DM, Normark PL, Hansson A, Rosenlind A (2001) Real-time GPS software radio receiver. In: Proceedings of ION 2001, September 11–14, 809–816, Salt Lake City
- Borre K, Akos DM, Bertelsen N, Rinder P, Jensen SH (2007) A software-defined GPS and Galileo receiver: a single-frequency approach. Birkhäuser Verlag GmbH, Boston
- Normark PL, Macgougan G, Stahlberg C (2004) GNSS software receivers—a disruptive technology. In: Proceedings of GNSS Symposium, Tokyo
- Normark PL, Christian S (2005) Hybrid GPS/Galileo real time software receiver. In: Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation, September 13–16, 1906–1913, Long Beach
- Ma C, Lachapelle G, Cannon ME (2004) Implementation of a software GPS receiver. In Proceedings of ION GNSS 17th International Technical Meeting of the Satellite Division, September 21–24, 956–970, Long Beach
- Stöber C, Anghileri M, Ayaz AS, Dötterböck D, Krämer I, Kropp V, Won JH, Eissfeller B, Güixens DS, Pany T (2010) ipexSR: a real-time multi-frequency software GNSS receiver. In: 52nd International Symposium ELMAR-2010, 15–17 September, Zadar
- Jovancevic A, Brown A, Ganguly S, Goda J, Kirchner M, Zigic S (2003) Real-time dual frequency software receiver. In: Proceedings of ION GNSS 16th International Technical Meeting of the Satellite Division, September 9–12, 2572–2583, Portland
- Ledvina BM, Psiaki ML, Sheinfeld DJ, Cerruti AP, Powell SP, Kintner PM (2004) A real-time GPS Civilian L1/L2 software receiver. In Proceedings of ION GNSS 17th International Technical Meeting of the Satellite Division, September 21–24, 986–1005, Long Beach
- Pany T, Förster F, Eissfeller B (2004) Real-time processing and multipath mitigation of high-bandwidth L1/L2 GPS signals with a PC-based software receiver. In: Proceedings of ION GNSS 17th International Technical Meeting of the Satellite Division, September 21–24, 971–985, Long Beach
- Chen YH, Juang JC, De Lorenzo DS, Seo J, Lo S, Enge P, Akos DM (2011) Real-time dual-frequency (L1/L5) GPS/WAAS software receiver. In: Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation, September 19–23, 767–774, Portland
- Juang J-C, Tsai C-T, Chen Y-H (2013) Development of a PC-based software receiver for the reception of Beidou navigation satellite signals. *J Navig* 66(5):701–718
- Viola S, Mascolo M, Madonna P, Sfarzo L, Leonardi M (2012) Design and implementation of a single-frequency L1 multi constellation GPS/EGNOS/GLONASS SDR receiver with NIORAIM FDE integrity. In: Proceedings of the 25th International Technical Meeting of the Satellite Division of the Institute of Navigation, September 17–21, 2968–2977, Nashville
- Ferreira PMLM (2012). GPS/Galileo/GLONASS software defined signal receiver. M.Sc. Dissertation, Universidade Técnica de Lisboa
- Casabona M, Rosen M (1999) Discussion of GPS anti-jam technology. *GPS Solutions* 2(3):18–23
- Bhuiyan MZH, Kuusniemi H, Söderholm S, Airos E (2014) The impact of interference on GNSS receiver observables—a running digital sum based simple jammer detector. *Radioeng J* 23(3):898–906
- Jones M (2011) The civilian battlefield-protecting GNSS receivers from interference and jamming, inside GNSS magazine, March/April, 40–49
- Kraus T, Bauemfeind R, Eissfeller B (2011) Survey of in-car jammers—analysis and modeling of the RF signals and IF samples (suitable for active signal cancellation). In: Proceedings of the 24th International Technical Meeting of the Satellite Division of the Institute of Navigation, September 19–23, 430–435, Portland
- Mitch RH, Dougherty RC, Psiaki ML, Powell SP, O’hanlon BW, Bhatti JA, Humphreys TE (2011) Signal characteristics of civil GPS jammers. In: Proceedings of the 24th International Technical Meeting of the Satellite Division of the Institute of Navigation, September 19–23, 1907–1919, Portland
- Jiang Z, Ma C, Lachapelle G (2004) Mitigation of narrow-band interference on software receivers based on spectrum analysis. In: Proceedings of ION GNSS 17th International Technical

- Meeting of the Satellite Division, September 21–24, 144–155, Long Beach
30. Gurtner W, Estey L (2009) RINEX—the receiver independent exchange format, version 3.00. <ftp://igsceb.jpl.nasa.gov/pub/data/format/rinex300.pdf>. Accessed 5 May 2015
 31. Sagiraju PK, Raju GSV, Akopian D (2008) Fast acquisition implementation for high sensitivity global positioning systems receivers based on joint and reduced space search. *IET Radar Sonar Navig* 2(5):376–387
 32. Juang JC, Chen YH (2010) Global navigation satellite system signal acquisition using multi-bit code and a multi-layer acquisition strategy. *IET Radar Sonar Navig* 4(5):673–684
 33. Bhuiyan MZH, Söderholm S, Thombre S, Ruotsalainen L, Kuusniemi H (2014) Overcoming the challenges of BeiDou receiver implementation. *Sensors* 14:22082–22098
 34. Bhuiyan MZH, Söderholm S, Thombre S, Ruotsalainen L, Kuusniemi H (2014) Implementation of a software-defined BeiDou receiver. In: Chinese Satellite Navigation Conference Proceedings: Volume I, Lecture Notes in Electrical Engineering, 751–762, Vol. 303, ISSN 1876–1100, Springer
 35. Borio D, Lachapelle G (2009) A non-coherent architecture for GNSS digital tracking loops. *Ann Telecommun* 2009(64):601–614. doi:10.1007/s12243-009-0114-1
 36. Nottingham Scientific Limited. NSL stereo datasheet. <http://www.nsl.eu.com/datasheets/stereo.pdf>. Accessed on 10 Jan 2014
 37. Antcom Corporation (2014) G5 L1/L2 glonass + L1/L2 GPS + Omnistar antennas for ground, avionics, marines, buoy and deepsea applications. http://www.antcom.com/documents/catalogs/G5_Antennas_L1L2GLonass_Plus_L1L2GPS_Omnistar.pdf. Accessed 5 May 2015
 38. Spectracom (2014) GSG-6 SERIES. Multi-channel, multi-frequency advanced GNSS simulator. <http://www.spectracomcorp.com/ProductsServices/GPSSimulation/Multi-FrequencyGPSSimulator>. Accessed 5 May 2015