GNSS SOFTWARE RECEIVER

Tomasz Hadaś[1]

Abstract

The idea of replacing hardware components with a software solution in a radio communication system was established more than a quarter century ago, but the available technology allowed to implement this solution in satellite navigation receiver construction not so long ago. The ability to change the signal processing algorithms and their parameters makes the software receiver a flexible tool to work with GNSS systems.

This article provides description of the main differences in the construction of conventional and programmable receivers, bringing the advantages and limitations of both solutions. Based on the ifen GmbH SX-NSR receiver the user interface and programming interface is described. Different user-receiver communications methods are presented and wide range of possible application in satellite positioning development are indicated.

Keywords

GNSS, software receiver, signal processing, SX-NSR.

1 GNSS RECEIVER TYPES

The primary function of GNSS receiver is to receive signals transmitted by navigation satellites and, based on them, determine the position of the receiver in real time or in post-processing. Electromagnetic waves reaching the antenna are converted into digital data. Depending on the capabilities of user access to intermediate processes, different types of GNSS receivers are distinguished.

1.1 Traditional receiver

Commercial GNSS receivers have limited functionality, consist of providing observables (e.g. in RINEX format) or the resulting coordinates. For most users this is sufficient enough, since they are not interested in inter-processes performed by the receiver.

An antenna is connected to a radio frequency signal conditioning, which is responsible for the amplification, noise and frequency filtering, frequency down-conversion, synthezation with and digitization. Digital samples are then sent to dedicated ASIC (Application-Specific Integrated Circuit), responsible for operations of digital correlation and accumulation of results in the code range period. The cumulative results are then transmitted with specified frequency to programmable microprocessor, whose task is to control tracking loops, decoding and processing of navigation data stream in order to determine position, velocity and receiver clock corrections [1].

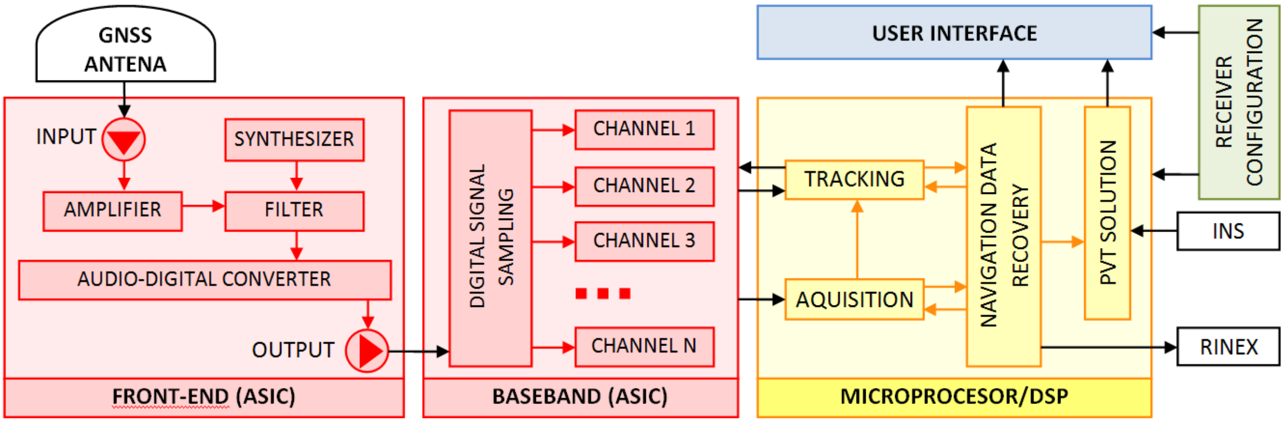

Fig. 1 *Traditional GNSS receiver architecture*

1.2 Software receiver

The idea of software receiver (SR) is to replace components, which are commonly implemented in hardware (eg. Mixers, amplifiers, filters), with software running on computer or any other digital device. Broadband analog-to-digital (ADC) converter should be as closed to antenna as possible.

[1]Tomasz Hadaś, M.Sc. Eng., Wroclaw University of Environmental and Life Sciences, The Faculty of Environmental Engineering and Geodesy, Institute of Geodesy and Geoinformatics, ul. Grunwaldzka 53, 50-357 Wroclaw, tomasz.hadas@igig.up.wroc.pl

It is not possible to completely eliminate hardware elements, because of the analog nature of incoming signal. Operations of amplification, synchronization with the reference signal, converting the original RF (radio frequency) to, the so-called, IF (intermediate frequency) and the signal digitization process are still performed by analog ASIC, called Front-End. It provides an interface of data exchange between the antenna and the software. A well-designed Front-End is a key element of high quality SR, because errors made in the analog stage are most often not possible to compensate at the digital phase [3].

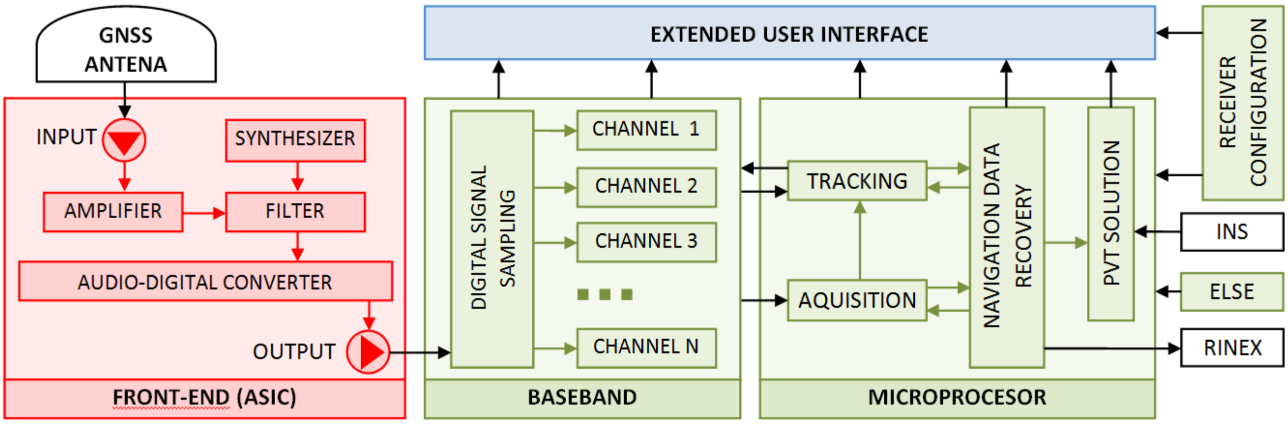

***Fig. 2** Software GNSS receiver architecture*

The ideal software receiver performs direct sampling of radio signal received by antenna using advanced analog-to-digital converter and process all the samples in software using universal CPU [1]. There is no loss of information after analog to digital signal conversion. The whole operation is conducted on wide signal spectrum with very high sampling frequency. Such a solution in impossible to implement using currently available computer technology, due to limited capacity and computational power. Therefore, signal bandwidth is limited and sampling frequency is reduces to a level, that allows efficiently handle the received data. In practice, few well-defined wave frequencies are chosen, which are the subject of further processing in the software.

SR is, in its hardware architecture, rather simple device – consisting of a relatively simple Front-End and software platform, which handles the burden of processing input data and presents it to the user in expected form.

1.3 FPGA circuits

The solution somewhere between software and hardware approach are Field Programmable Gate Arrays (FPGA). These are logic circuits, which may be programmed with usage of specialized tools and hardware description languages (HDL). On the one hand they have a limited ability of reprogramming and on the other hand are much faster than typical software solution.

FPGA approach could be considered as a type of SR, due to the possibility of changing its configuration. However it is accepted, that definition of software receiver concept concerns systems in which signal processing is carried out on a programmable microprocessor or a digital processor, excluding the FPGA approach [1].

1.4 Comparison

The advantage of software receivers is full user control over all process associated with receiving and processing the signal, and determination of receiver PVT. Nevertheless, they are slower than chips, that have advantage in capacity and computational power. ASIC circuits are, by definition, dedicated for specific solution – once designed con not be reprogrammed, thus the change of functionality of receiver is dependent on its complete reconstruction. Similar computational capabilities have FPGA's whose disadvantage is the long reprogramming time, which limits their potential and flexibility.

Software receivers are the ideal platform not only for development and testing new algorithms, but also for receiver integration with other devices and user-defined data formats [2].

2 SOFTWARE RECEIVER HANDLING

The complete software receiver provides extensive user interface and enables for the user to modify various stages of processing zero-one data. Experiments with the individual algorithms should not interfere the work of other modules, so the source code and programming interface must flexible and transparent enough.

2.1 User interface

The user interface (UI) should present the results of data processing data on various stages in an intuitive form, in particular those whose may have changed as a result of user interaction in the algorithms. In addition to providing numerical information in standard or user-defined data exchange formats, the UI should simultaneously present data or results in a form of dynamic plots, to visually inspect applied changes. The user should also be informed about the currently running processes and their status, e.g. in the form of short text messages.

Depending on the purpose to which the SR is used, the user interface may consist of the following modules:

- receiver status view – displays information about the current configuration and operating mode, messages and errors from individual modules; certain information should be recorded in the log file, which detail level is determined by the user;
- signal analysis window – contains information about the signal received by antenna in the domain of time or frequency; this module may consist of power-spectrum-density plot of each incoming stream, a 3D plot of two-dimensional correlation function to search for satellite signals (vertical axis gives the correlation peaks as a function of frequency and code phase) and receiver multi-correlator results plot for selected system and satellite PRN [4];
- channel status – decoded information from each receiver channel: satellite system and PRN, values of observables, signal quality; position of all satellites tracked can be presented on a simple sky plot;
- navigation module – shows (in a typical way for commercial receivers) final results of data processing: receivers coordinates, its velocity, receiver time correction or accurate GPS time, accuracy of results obtained; additional plots of individual parameters may be also present to provide the user a tool for fast and intuitive assessment of the results;
- data export – allows to store intermediate results of studies on hard disk in order to further analysis in external software; user should be able to predefine its own data exchange format or use standardized format such as RINEX, RTCM, NMEA.

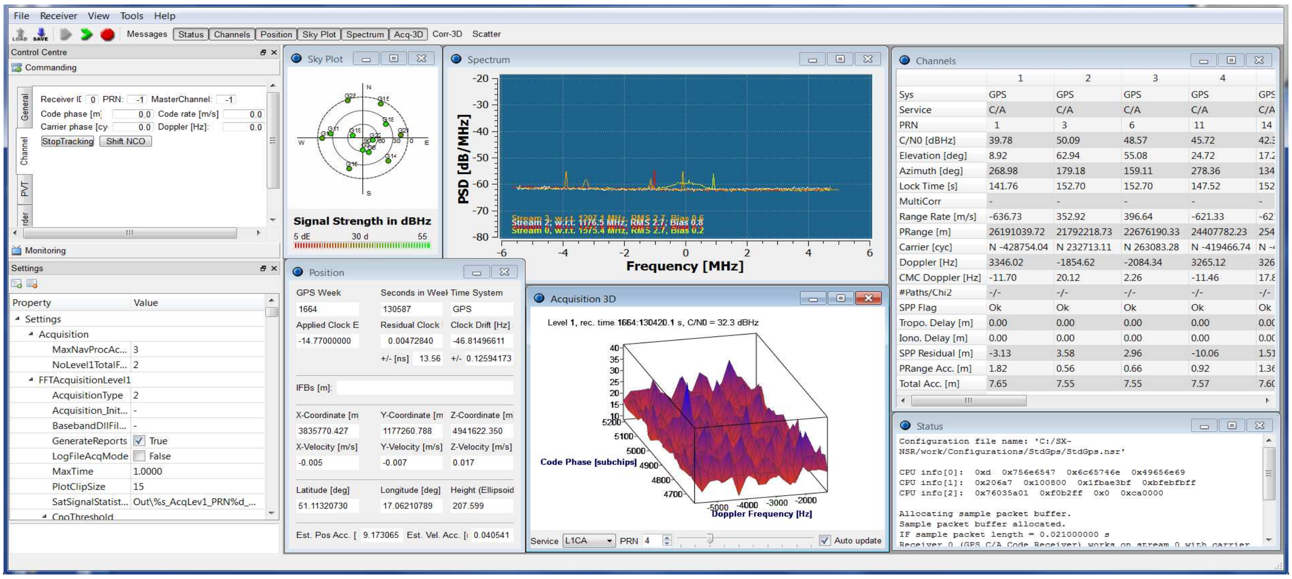

Fig. 3 *SX-NSR software receiver user interface*

2.2 Programming interface

The heart of a software receiver are algorithms responsible of conversion incoming digital stream to final results e.g. receiver state vector. User can interfere with software through an applications programming interface (API), thus obtaining two-sided access to all its functions.

The algorithms are written in high level programming language, which are transparent for less experienced users also. Individual functions are activated automatically in case of specified event occurs, such as start or end of work, minimum number of satellites tracked, new epoch of observations, signal parameter change. User is able to edit existing functions or create own program that is responsible for GNSS data processing.

The easiest way to intervene in the operation of the receiver is to change its parameters through a graphical interface (GUI). It is possible to create users configuration profiles and store them on the hard disk. Most of the changes will result immediately, while some might require a restart or the receiver.

It is also possible to completely omit the prepared software and to capture IF samples directly, in order to analyze raw signal and modify samples. Sampling interface is somewhere between Front-End and receiver software. Data transfer is done via packet data stream. The content of each packet may be defined on receivers start-up. Each time the receiver will record new sample, the corresponding function is called upon to offer an API, which indicates obtained values. User can modify and process the values, and then return them to receiver through the API, which will result in further processing in the receiver.

Remote communication with the receiver can be done via TCP/IP protocol. User sends a command to specific port, which is listened by receiver software. After processing the command in the software, receiver returns feedback in a specific data structure. The command list and returned structures are strictly defined. User may develop its own algorithms in external programming environment (e.g. Matlab), which will control data processing on selected stage in receiver by sending commands dependent on feedback data. Properly prepared command would not disrupt the work of the whole system, so the user can operate only with known issues, without worrying about other stages. This approach allows to continuously receive final results, as prepared by the manufacturer of the receiver.

2.3 Application

Digital signal processing in software lets you deliver high-quality observational data. With access to the structure of the signal at a low level it is possible to optimize the process of its processing, use of advanced algorithms for signal amplification, filtering and tracking, to minimize the effect of multipath [5].

Software receiver can simultaneously serve several functions that are implemented by shared or dedicated modules. By comparing results of different solutions, e.g. in real time, it is easier to analyze the quality, efficiency and influence of the algorithms. Many users may adopt different functions to suit their needs, using always the same hardware, but different parts of the software. Software receiver construction is open - it is limited only by users creativity and the available computing power.

The flexibility of the receiver also revealed in the possibility of adapting it to track new GNSS signals, such as from Galileo, decoding of new information or cooperating with the new external data formats. Software receiver can be modified, while the classical GNSS receiver needs to be rebuilt or totally replaced by more modern hardware. Adapting the software receiver to the new frequency requires only a small firmware update, which consists of change in the Front-End ADC filter parameters [6]. The investment in software receiver gives a solution that will use the full GNSS systems functionality for many years.

Equally interesting, from a research point of view, is the ability to dynamically adapt the SR to the external conditions. Depending on the signal parameters and the available data (such as precise orbits, corrections stream) the receiver may use different algorithms for signal processing and determination of the position. In the case of high buildings area, the receiver should treat the multipath effect in a special way and e.g. eliminate weak signals. If the SR registers only weak signals, it can recognize that it is somewhere in the woods or under a roof, and then the software applies completely different algorithms that work best for this environment. Also, the speed of the antenna can be a decisive factor in choosing the method of calculation of the final solution [5]. Suitable for universal applications commercial receivers are not so flexible.

LITERATURE

[1] MACGOUGAN, Glenn, NORMARK, Per Ludvig, STÅHLBERG, Christian. *The Software GNSS Receiver* [Online]. GPS World, 01.01.2005 [cit. 28-11-2011]. Dostępny z: http://www.gpsworld.com/gps/the-software-gnss-receiver-936?page_id=1

[2] BORRE, Kai, AKOS, Dennis M., BERTELSEN, Nicolaj, RINDER, Peter, JENSEN, Søren Holdt. *A Software-Defined GPS and Galileo Receiver.* Boston: Birkhauser, 2007. ix-x. 978-0-8176-4390-4.

[3] PANY, Thomas. *Navigation signal processing for GNSS software receivers*. Norwood: Artech House, 2010. 17-24. 978-1-60807-027-5.

[4] *SX-NSR Navigation Software Receiver.* Version 2.0. IfenGmbH, 16.06.2010.11-21.

[5] http://www.gps-practice-and-fun.com/software-gps.html

[6] GOLD, Kenn, BROWN, Alison. *A Software GPS Receiver Application for Embedding in Software Definable Radios.* Proceedings of ION GPS 2003, Portland, Oregon, September 2003.

REVIEWER

Jarosław Bosy, dr, Wroclaw University of Environmental and Life Sciences, Faculty of Environmental Engineering and Geodesy, Institute of Geodesy and Geoinformatics, Professor, Vice Head of Institute, ul. Grunwaldzka 53, 50-357 Wroclaw, +48 71 3205688, jaroslaw.bosy@up.wroc.pl