The World of Software Receivers for Satellite Navigation

Benoit Muth

Introduction

Software-Defined Receivers (SDR) are radio receivers that contain as few hardware components as possible. In such a receiver, the signal processing block then starts as close as possible after the antenna, and is carried out with scripts running on a general-purpose processor contrary to traditional hardware radio receivers.

The application of this concept to receivers for Global Navigation Satellite Systems (GNSS) signals was initiated in the latter part of the nineties. Several implementations have been developed all over the world since, at a time where existing GNSS constellations are being modernised and new constellations being constructed. Software receivers not only offer several advantages in terms of processing compared to the traditional GNSS receivers architectures, they also find various and sometimes unexpected applications. This paper contributes with an update on the different GNSS software receivers, a couple of years after the first state of the art [1].

The US GPS is the only fully-operating GNSS since the early nineties. This system has been undergoing several modernisations: new generations of increasingly performant satellites are being launched, and signals offering higher ranging accuracy are being transmitted. Regional augmentation systems to GPS have been developed, such as the European system EGNOS, opening the way to the development of complementary global navigation systems such as Galileo.

The existing GNSS systems, like GPS, are each increasing the number of transmitted signals, with modernised signals in the existing bands and signals in the new frequency bands. This new situation allows for more navigation observables and different combinations of navigation observables, which enables the easier correction of, for example, atmospheric effects and yields more accurate position and velocity estimates. These improvements might be taken further by combining observations from different GNSS systems, as long as these GNSS systems are compatible with each other and interoperable.

For an overview of existing GNSS systems and the future of these systems, we refer to the preceding contribution *Global Navigation Satellite Systems: status, plans and threats* in this year's NL-Arms by Christian Tiberius.

Software Receivers

Software receivers are digital implementations of receivers for radiofrequency signals, that ideally contain as few hardware components as possible. The digital signal processing (DSP) of the stream of discrete-time samples starts right after the antenna, and is achieved by software scripts running on a general-purpose processor. The hardware in a software receiver would then just be made of an antenna attached to an analogue to digital converter (ADC) delivering signal samples to a DSP block. A software receiver does not transmit any signals, to the contrary of a software radio, which is a digital implementation of a transceiver, that not only receives but also transmits signals.

Software-defined receivers

The processing of signals sampled following Nyquist's theorem, an approach called direct digitisation, is not possible or too expensive with the capabilities of *actual* ADCs and DSP boards, and might not become very much easier in the coming years. Indeed, the frequency range and dynamic range of the ADC are usually too limited, and the current available processing powers are too limited, at least for real-time operation. In practice, these limitations are then bypassed by adding analogue components after the antenna of the ideal software receiver architecture, making so-called software-defined receivers. The difference between software and software-defined receivers is illustrated for GNSS in Figure 4. Traditional analogue radio architectures, i.e. radio front-ends, are then still needed to cope with the limitations of the ideal software receiver and efficiently implement a software-defined receiver.

The most spread architecture is the heterodyne architecture that down converts the received signal to a (lower) intermediate frequency (IF), in one or several stages with one or several sets of oscillators, mixers and filters, one set for each down conversion stage. In order to reduce the power consumption of the receiver and the price of analogue components, some receivers implement the homodyne architecture, also called direct conversion architecture, which achieves the down conversion in one step of the RF received signal to baseband or quasi-baseband. This approach is particularly fruitful for the reception of wideband signals or of the combination of several narrowband signals (for the application to GNSS see [2]).

Features of the software receiver

The software receiver approach might first bring cost reduction in the design of radio receivers, since for instance, the displacement of the ADC towards the antenna enables the suppression of analogue components and digital filters in the front-end. Second, the software receiver approach brings flexibility in the design of radio receivers: the single-purpose narrowband components can be replaced by wideband components allowing for the fusion of reception capabilities for different radio systems. The software approach then enables the reception with one device of signals with different modulations and encryptions, or the integration of navigation functions in one electronic device with other non-radio systems.

More important, the software approach enables the re-programmability of the hardware, consequently the updating and improvement of the radio receiver capabilities. Eventually, if the processing power offered by market PCs in the early decennia was not sufficient to have software receivers with real-time processing capabilities, Moore's law ruling the speed of processors has made it now possible for demanding applications, such as GNSS receiver processing to run on ordinary PCs.

GNSS receivers

In this section we concentrate on software receivers for GNSS satellite signals, like the signals involved in GPS.

Purpose of the receiver

GNSS satellites currently transmit so called direct sequence spread spectrum signals, using Code Division Multiple Access (CDMA), i.e. transmission of multiple signals on one frequency using different codes for the signals. This imposes on GNSS receivers to be receivers for CDMA signals, that are either BPSK or BOC modulated [3,4]. Since ideal software receiver currently cannot be built, GNSS receivers depicted, as in Figure 1, implement the classical heterodyne radio architecture to collect the RF signal with an analogue front-end that is located between the antenna and the ADC. In the Digital Signal Processing (DSP) block, usually implemented in hardware through Application Specific Integrated Circuits (ASICs), the received satellite signals are synchronised and demodulated, achieving the simultaneous production of ranging observables and extraction of the navigation information transmitted by the satellites. Once this data has been processed, the receiver can compute its own Position, Velocity and Time (PVT).

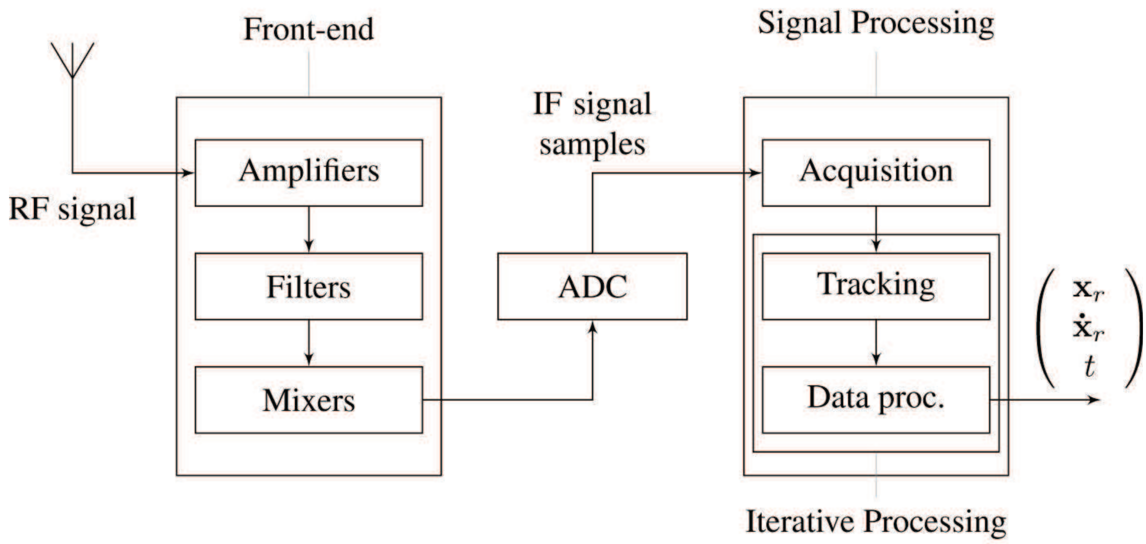

Figure 1: Simplified architecture of a GNSS receiver.

Architecture of the software receiver

The classical receiver implements the heterodyne architecture, that preprocesses the received signal. This means, as depicted in Figure 1 that the received signal is (bandpass) filtered, amplified to an acceptable level, down-converted to a lower, intermediate, frequency (IF) to make it processeable and digitised, i.e. transformed into a stream of discrete-time samples. The receiver then proceeds to signal acquisition, that is, the detection of the received signals and a coarse estimation of the code phase and Doppler frequency synchronisation parameters. The signal processing is then split into different channels corresponding to the detected satellites. In each channel, the received signal is synchronised so as to remove the code and carrier from the signal, i.e. extract the navigation data, and compute navigation observables, pseudorange, carrier phase, accumulated Doppler. In the end, the extracted navigation data is decoded and the receiver PVT estimates $[\mathbf{x}_r^T \dot{\mathbf{x}}_r^T \delta t^T]^T$ can be computed. In Figure 1 the box embedding the tracking and data processing blocks represents the continuous processing part inside the receiver.

A global model of the GPS receiver is presented in [3-7]. To still take advantage of the fastest techologies, one of the trends in GNSS receivers is to consider software routines run by a Field Programmable Gate Array (FPGA), or by a Reduced Instruction Set Computer (RISC) microprocessor instead of a PC. A detailed quantitative analysis of the design, the implementation of such processor architectures for GNSS SDR receivers is presented in [8], including an exhaustive study of existing Application Specific Instruction Processors (ASIP) based architectures.

Receiver starting options

The direct environment and history of a GNSS receiver have such impact that when switched on, the receiver faces different conditions, respectively called cold, warm and hot start conditions. A hot start happens after a relatively short interruption of the reception, from a passage in a tunnel until a shortage of less than two hours. A warm start occurs when the GPS receiver is switched on more than four hours after being switched off at the same place. In this case the receiver does not have any valid ephemeris, although it does have a valid almanac. To the contrary, a cold start happens when no valid timing or location information is available to the receiver, which has not been used for several days, or has travelled far from the last location known to it. These cold start conditions are the only type of conditions that matter to characterise the receiver performance.

The receiver's RF front-end

As illustrated in Figure 1, the RF front-end most of the time implements the heterodyne radio architecture, that filters the spectral band of interest, mixes it to an IF, filters and amplifies the down converted signal that is next transformed by an ADC into a stream of discrete-time samples [3,4,6,7,9-12]. These pre-processing operations in the analogue front-end shall perform identically independently from the reception conditions at the antenna, i.e. independently of the received signal strength and the value of the signal carrier frequency, independently also of potential unintentional interferences from external sources and independently of the self-noise from the front-end. As mentioned above, given technological limitations, the direct sampling of the received signal at RF is not possible, while the sampling at IF or baseband is. Besides the number of quantisation bits, the sampling is characterised by the value of the sampling frequency f_s which in turn is linked to the choice of the sampling strategy [9]. Most front-ends deliver signal streams containing one phase of the received signal. Depending on the signal structure and the receiver specifications, one might want the phase and quadrature components of the received signal.

Signal processing in GNSS receivers

The signal processing in a GNSS receiver is basically made of three blocks: the acquisition block, the tracking block and the data processing block. These three blocks achieve the synchronisation and demodulation of received satellite signals by means of signal processing algorithms, the production of ranging observables and the extraction of the navigation information. Both acquisition and tracking stages are based on the measure of the similarity, correlation, between a received satellite signal and a signal template. For the acquisition, this measure is a two-dimensional quantity depicted in Figure 2. From the tracking outputs the navigation message can be decoded and a position, velocity and time (PVT) solution can be produced.

The acquisition part

While the RF front-end processes one single stream of data, the DSP block separates the different satellite signals. The first task of the DSP, the acquisition, is then to detect which satellites are in view of the receiver and for each satellite, to yield coarse estimates of the synchronisation parameters, the signal travel-time and carrier frequency. This estimation-detection procedure is possible thanks to the properties of the GNSS signal spreading sequences. On the whole, the acquisition is by far the most demanding part of the whole receiver chain in terms of computational cost. This cost is highly variable according to the acquisition method, the initial status of the receiver and the environment of the receiver. Acquisition techniques are based on the correlation of the received signal with a signal replica. Numerically, this correlation measure is computed on a predefined, discrete search grid. The accumulation of the correlation values in a coherent and/or non-coherent way, makes up for a simple but efficient means to enhance the sensitivity of the receiver. Challenging acquisition scenarii include the processing of weak signals and signals with multi-trajectory propagation, multipath, that can be found in urban canyon or indoor environments.

Several factors impact the signal acquisition performance. Satellite-receiver dynamics and characteristics of the receiver, such as the sampling frequency or IF frequency, impact the resolution of the search grid. The implementation of the correlation operation, the integration length, the search strategy are other useful settings of the parameter estimation and the signal detection. The precision (bias and accuracy) of the synchronisation parameter estimates, the sensitivity of the algorithm and the mean acquisition time qualify the performance of the acquisition algorithm. The choice of the acquisition method and the setting of its parameters usually results from a trade-off between computational load / processing speed of the DSP and sensitivity of the algorithm.

The tracking part

The next signal processing step, called tracking, aims at refining and keeping track of the synchronisation parameters, the code phase and carrier frequency, based on the coarse parameter estimates, obtained during the acquisition. The tracking then achieves for the received signal the continuous estimation of the synchronisation parameters and of the GNSS signal parameters : the code phase (travel time), the Doppler frequency shift, accumulated Doppler shift and carrier phase. Of course the accuracy of these parameter estimates has an impact on the precision of the pseudorange measurement, i.e. the distance from the receiver to the satellite. In practice, the signal tracking is divided into two functions, the code tracking and the carrier tracking, that adjust the code phase and carrier phase (fre-

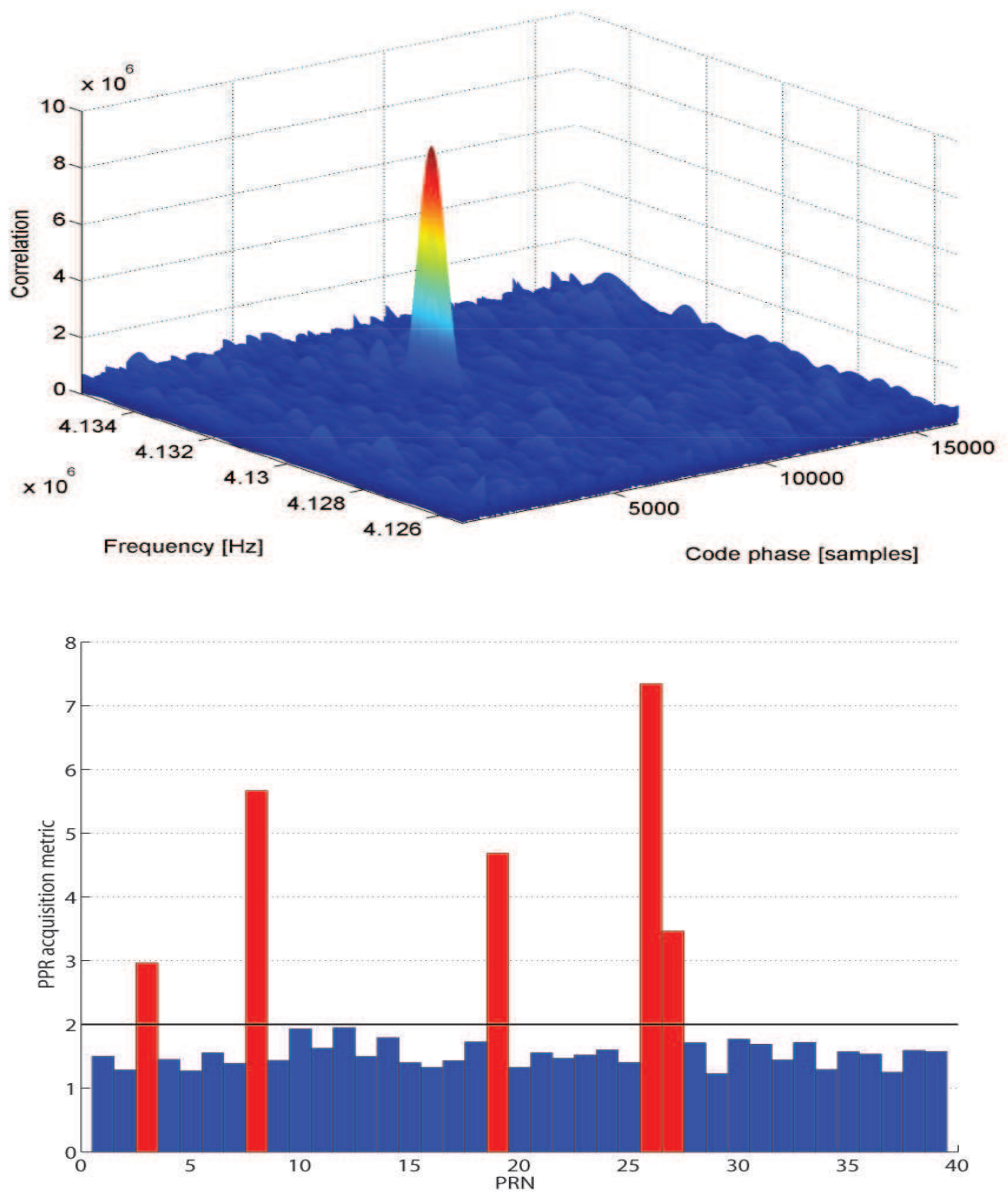

Figure 2: Two-dimensional correlation function of the received signal with the replica for PRN 3 (top) and Acquisition metrics (detection power) of all GPS satellites (bottom): red satellites are detected, blue satellites are considered as absent. The peak on the left plot indicates the presence of the satellite signal. The coordinates of the peak indicate the time-offset and frequency offset of the received signal with regard to the nominal frequency due to the satellite receiver dynamics.

quency) of a replica to the code and carrier phases of the received signal. This tracking is implemented in traditional DSP architectures, with closed-loops called Delay-Lock-Loops (DLL) and Phase-Lock-Loops (PLL). Both code and carrier tracking loops are made from a detector/discriminator, a filter and an oscillator updating the parameter tracked in the loop.

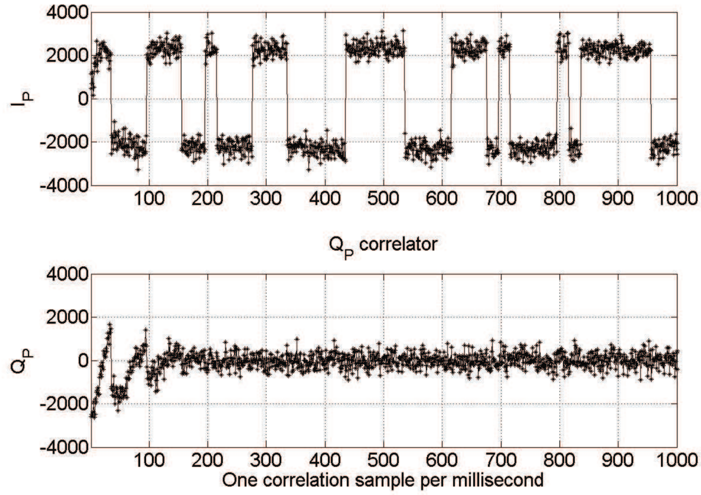

Figure 3: Tracking outputs for satellite signal of space vehicle (SV) 3 recorded in an experiment in the surroundings of Den Helder. In the figure: demodulated phase and quadrature arms (90 degrees shift) showing that the navigation data appear in the prompt-phase arm I_P (in phase).

A lock detector is used to check whether these tracking loops are locked or have lost lock. In this last case, the acquisition must be carried out again for the considered satellite. Otherwise if the tracking is successful, the signal stream at the output of the tracking delivers the navigation data carried by each satellite signal. Next to the navigation data the tracking block yields, for each visible satellite ranging measurements, the code pseudorange, possibly the accumulated Doppler and the carrier phase pseudorange, enabling the computation of a position-velocity-time (PVT) solution. Obtaining these measurements is very well known when the receiver is static and not subject to interference. On the other hand, current receiver challenges include the determination of correlation losses due to multipath or interference.

The main factors impacting the tracking performance are the signal modulation, the chip (one code elemenent) length, the type of discriminator, the early-late correlator spacings, the sensitivity of the tracking loops to the received signal strength, the satellite-receiver dynamics, interferences or multipath. On the other hand, the tracking performance can be characterised mainly by the (non-linear) dynamic and stability properties of the PLL and DLL.

Data processing in the receiver

Once the signal has been demodulated, the data processing first consists of synchronizing and decoding the 50 Hz navigation data. The decoding is achieved in several steps, since the navigation data is structured for GPS, in words containing 30 bits, 10 words forming a (300-bit) subframe, 5 subframes making a (1500-bit) page. Consequently, if a receiver does not have its position, a long enough signal must be read to be sure to obtain the complete navigation data. Once the navigation data has been decoded, the receiver has the knowledge of the satellite's position, of clock corrections terms, can synchronise with the GPS time and can proceed to the computation of pseudoranges and the estimation of its position, velocity and time [3,13].

Software receivers for GNSS

As for other radio frequency receivers, GNSS receivers become all-digital receivers. The software receiver context then not only offers to GNSS receivers advantages in terms of flexibility and reconfigurability, but also enable new applications. Technologically the ideal GNSS software receiver is not feasible, and an analogue front-end is necessary in practice to build a GNSS software-defined receiver, as depicted in Figure 4.

Software-defined receivers first make up for a good framework for developing the GNSS receiver DSP [14,15] and improving the positioning accuracy by integrating other systems to the GNSS receiver platform. Considering the signal processing only, the SDR flexibility is especially needed when designing a new receiver. One can indeed more easily make a multi-frequency receiver, such as a GPS L1 / L2 receiver, with multiple signals on these carrier frequencies such as the C/A code and P(Y) code on L1. One should also implement more easily a multi-constellation receiver, such as a GPS / Glonass receiver. Considering the PVT accuracy, the GNSS receiver can be integrated with, for example, inertial systems such as INS, or other navigation systems such as LORAN. Secondly, considering the reception of one signal from one GNSS system, specific scenarios, such as interference, multipath or weak signals conditions reception can be studied and the fine tuning of the implemented algorithms is made easier. Thirdly, in terms of integration, the software receiver approach not only enables the integration of several navigation functions on one platform to improve the navigation solution but also, by integration, the sharing of one single DSP platform, e.g. a cellphone, with other applications such as cellular communications or embedded applications [16].

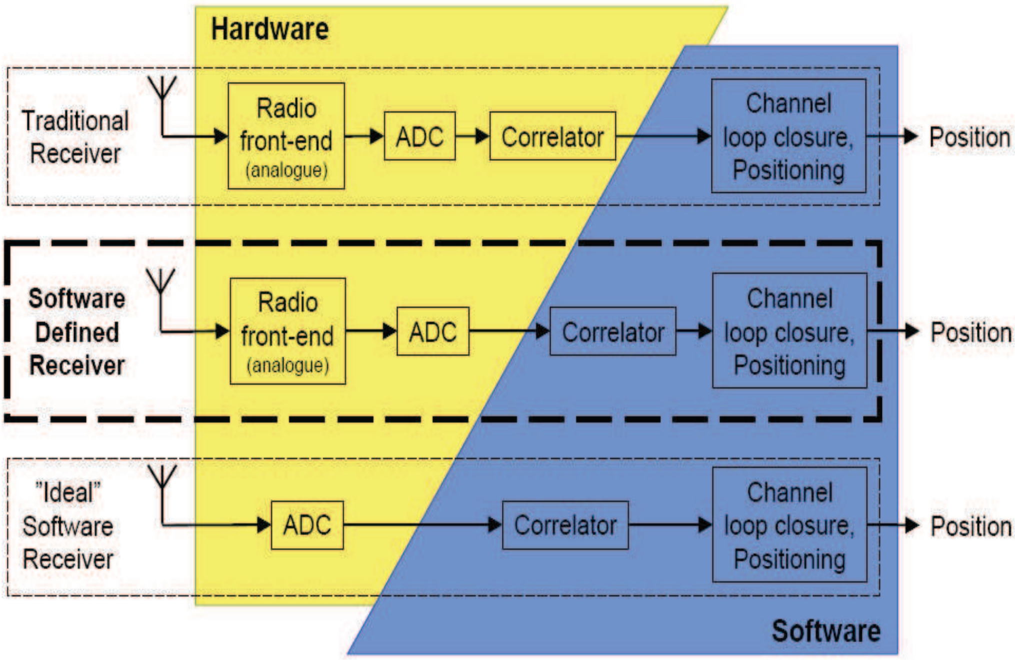

Figure 4: Different implementations for GNSS receivers: software receiver, software-defined receiver and hardware receiver, from [14].

Brief history of GNSS software receivers

The first front-end for GNSS software receivers has been presented in [10], together with the theory of bandpass sampling and other sampling strategies. The first GPS L1 C/A software receiver was then borne. The first realtime software receiver for GPS L1 C/A signals was presented in [17]. Several overview articles have been devoted to GNSS software receivers, the first of which is in the author's opinion the most comprehensive [1]. Other articles with different depth have followed : a list of resources on software receivers in 2005 was established in [14]. Further technical hardware and software-related issues inherent to GNSS software receivers are to be found in [18,19]. Overview articles including safety concerns in civil aviation or military contexts, such as reliability of the position delivered by software receivers, are to be found in [1,20].

The first book on GPS and Galileo software receivers appeared in [4] together with software receiver scripts and data records. These data records

Figure 5: Low-cost antenna and (SiGe) software receiver front-end for pre-processing the received satellite signals. This front-end connects to a PC on which the recording and processing of the data signals take place. Images are from Sparkfun.

were collected with the front-end depicted in Figure 5. Further, [21] presents improvements brought to the software scripts and results that include (1) frequency spectrum monitoring plots, (2) multipath mitigation and tracking results, (3) FPGA and real-time implementation results, (4) implementation of multiple correlators for new GPS and Galileo signals and (5) the bump-jumping technique for removing ambiguities in side-peaks of the auto-correlation function of the new BOC signals.

Software receivers open new perspectives for the improvement of the accuracy of the pseudorange measurements and then of the positioning accuracy with standalone receivers. Indeed, modifications to the receivers such as an increase in the number of correlators, mitigation of physical channel propagation effects acquisition and tracking of weak signals, wipe-off of the navigation data thanks to assisted GNSS, or implementation of the vector delay-lock-loop (VDLL) become possible with a software-defined receiver. Technological trends such as the consideration of Graphical Processing Units, multi-core processing technologies shall supply FPGAs for handling large computational burdens [15,22]. Currently, particular needs concern the availability of hardware for down-converting the received signal, and digitising the variety of RF spectral bands. One example of multifrequency software receiver is illustrated in [23], where GPS L1, L2C and L5 signals are synchronously collected thanks to a triple-frequency front-end. In this case, since the L2 and L5 signals do not bear any navigation message, pseudorange measurements can still be obtained in the software-defined receiver SDR framework by sharing the information from the L1 navigation message.

New applications by software receivers

Besides the new possibilities for standalone processing offered by software receivers in terms of programming flexibility and integration with other systems, software receivers enable the testing and validation of new algorithms. It becomes possible to add in software new functions to the receiver that work independently from the signal processing subsequent to the PVT estimation : such as time-domain histograms for BOC signals run lengths [24] or verification of the integrity of the receiver position [3]. Besides these applications, software-defined receivers have soon become useful to implement signal quality monitoring functionalities at different scales. GPS L1, L2 and L5 signals are being collected triple-frequency thanks to a front-end with very high sampling rate, for monitoring anomalies and exceptional events based on the raw data signal samples only [25].

GIOVE-A and GIOVE-B signals are being monitored using Chilbolton's astronomic telescopes, allowing for the assessment of transmitter non-idealities in a software receiver environment [26-28]. However the first objective of the application of the software-defined receiver approach to GNSS was the search for more computational efficiency for the traditional acquisition and tracking tasks. The implementation of new acquisition techniques such as the FFT-based parallel-search acquisition [29] was one of the first steps in this direction, giving birth to several new ideas for the purpose of signal acquisition [30,31] in the frequency-domain and [32,33] in the time-frequency domain.

As GNSS constellations are multiplying and their utilization is becoming widespread, computational efficiency inside the receiver processing and positioning accuracy remain major challenges for standard receivers. Other challenges arise for aviation or military users needing trust, integrity, on the provided receiver position. Indeed, the receiver has to cope with threats to the receiver accuracy such as unintentional or intentional interference, jamming. These phenomena have been studied for several years and are known. New challenges arise with the protection of receivers against spoofing attacks. In such cases, a signal resembling the broadcast satellite signal, but different, is sent towards the GNSS receiver to cause it to produce a wrong position. Additional checks, signal authentication, are implemented for the receiver to know it produces not only a reliable position, but also in an adequate way and with reliable signals. Signal authentication is then necessary to have the receiver acknowledge the signal processing required for producing ranging measurements and further estimate a position. Signal authentication or anti-spoofing within software receivers, is a more recent field of investigations, where techniques using signals encrypted by the service provider, such as the US GPS P(Y) code, make it possible to the expert user to check the authenticity of the received signal. The first GPS L1 C/A code signal spoofer has been developed using a software radio [34], while the first anti-spoofing technique considers the joint processing of signals collected at different locations, close one to the other [35], to access the military P(Y) code sequence for authentication.

A second application involving the processing of signals received at two different stations is the measurement of the three-dimensional baseline vector between the stations [36,37]. The next contribution in this year's NL ARMS will discuss this application in more detail.

In a similar way to the interferometric positioning, GNSS-reflectometry (GNSS-R) implements the cross-correlation of two received GNSS signals in post-processing. GNSS-R consists in the recuperation of signals reflected by the Earth and the determination of properties of the reflected surface. Indeed, two antennas being connected to a receiver above the surface to study, the receiver collects for each visible satellite, one direct signal from the satellite and one signal reflected by the surface, typically an icy surface, the sea, or the ocean. Reflectometry experiments have been carried out where the receiver was placed on a satellite [38]. Compared to the case of the interferometric positioning, the cross-correlation energy of the direct and reflected signals is more spread and exhibits a specific shape that depends mainly on the reflected surface, and the parameters to retrieve [39-41]. Possible applications of the GNSS reflectometry principally lie in remote, Earth-sensing applications. For instance one can retrieve information on the ocean wind and waves (scatterometry), on the ocean mean height (altimetry) or information on sea ice, land surface topography or near surface soil moisture. These applications are possible thanks to the post-processing capabilities of software receivers.

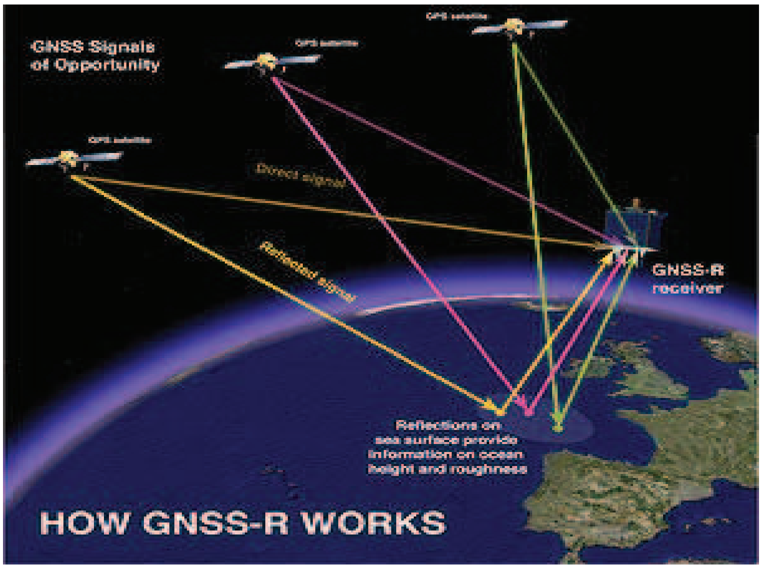

Figure 6: GNSS reflectometry consists in recuperating GNSS signals reflected from a surface. This picture from the University of Southampton depicts the application of GNSS-R to scatterometry or altimetry purposes above the North-Atlantic ocean.

Conclusions

Software receivers result from the breakthroughs in analogue components technology and processing technology. Consideration of GNSS and GNSS receivers have been becoming widespread since the early nineties. As a consequence the late nineties have seen the birth of software receivers for GNSS as research and development tools for traditional GNSS receiver signal processing, processing benchmarks of new signals and new constellations, development platforms for new implementations of old concepts and simply, new applications. GNSS software-defined receivers are tools that enable receivers to be quickly reconfigured and adapted to different environments, which matters in critical environments.

Simple experiments such as the interferometry are already possible with low-cost front-ends, yielding accurate single-difference positioning results. GNSS are tools that enable the study of signal processing immunity to unintentional and intentional interferences or worse, resistance to signal spoofing. Indeed signal authentication algorithms exist, to guarantee the integrity of the signal processing, not only of the estimated position. Eventually, as processing capabilities emerge, software-receivers enable applications that are highly-demanding in terms of computational load, such as GNSS reflectometry.

References

[1] P.-L. Normark, "Software Based GPS : The SDR concept can be applied to receivers for Global Navigation Satellite Systems, giving improved flexibility and reducing development and production costs," *Swedish Journal of Military Technology*, vol. 4, pp. 16–21, 2003.

[2] R. Weiler, P. Blunt, P. Jales, M. Unwin and S. Hodgart, " Performance of an L1 / E5 GNSS Receiver using a Direct Conversion Front-End Architecture," *Proc. of the International Technical Meeting of the ION*, pp. 1478–1489, 2008.

[3] P.Misra and P. Enge, *Global Positioning System - Signals, Measurements, and Performance.* Lincoln, (Massachusetts): Ganga-Jamuna Press, 2006.

[4] D. Akos, K. Borre, N. Bertelsen, P.Rinder and S. Jensen, *A single-frequency software-defined GNSS receiver*, Birkhäuser, 2006.

[5] J. O. Winkel, "Modeling and Simulating GNSS Signal Structures and Receivers", Ph.D. dissertation, Universität der Bundeswehr München, 2003.

[6] B. Eissfeller, "Ein dynamisches Fehlermodell für GPS Autokorrelationsempfänger," Universität der Bundeswehr, München, 1997.

[7] B. W. Parkinson and J. J. Spilker, *Global Positioning System: Theory and Applications.* Washington: AIAA, 1996.

[8] G. Kappen, V. Pieper, L. Kurz and T. Noll, "Implementation and Analysis of an SDR Processor for GNSS Software Correlators," *Proc. 21th ION GNSS Conference*, 2008.

[9] D. M. Akos and J. B. Y. Tsui, "Design and Implementation of a Direct Digitization GPS Receiver Front End" *IEEE Trans. on Microwave Theory and Techniques*, vol. 44, pp. 2334–2339, December 1996.

[10] D. M. Akos, "A Software Radio Approach To Global Navigation Satellite System Receiver Design," Ph.D. dissertation, Ohio University, August 1997.

[11] D. M. Akos, M. Stockmaster and J. B. Y. Tsui, "Direct Bandpass Sampling of Multiple Distinct RF Signals," *IEEE Trans. on Communications*, vol. 47, pp. 983–988, July 1999.

[12] The Danish GPS Center, "The SoftGPS Project", web pages provided http://gps.aau.dk/softgps/," 2010.

[13] E. Kaplan, *Understanding GPS - Principles and applications.* Boston: Artech House, 2005.

[14] R. Babu, "Web-based resources on software GPS receivers," *GPS Solutions*, vol. 9, pp. 240–242, 2005.

[15] M. Hollreiser, "Galileo Receiver Design and Performance," Lecture Notes Galileo Course, Navitec, ESTEC, 2006.

[16] D. M. Akos, "The role of Global Navigation Satellite System (GNSS) software radios in embedded systems," *GPS solutions*, vol. 7, p. 14, 2003.

[17] B. M. Ledvina, M. L. Psiaki, S. P. Powell and P. M. Kintner, "Real-Time Software Receiver Tracking of GPS L2 Civilian Signals using a Hardware Simulator," *Proc. 18th ION GNSS Conference*, pp. 1598 – 1610, 2005.

[18] C. Kelley and D. Baker, "OpenSource GPS, A Hardware/Software Platform for Learning GPS: Part I, Hardware," *GPS World*, 2006.

[19] C. Kelley and D. Baker, "OpenSource GPS, A Hardware/Software Platform for Learning GPS: Part II, Software," *GPS World*, 2006.

[20] J. H. Won, T. Pany and G. W. Hein, "GNSS Software Defined Radio: Real Receiver or Just a Tool for Experts ?," *Inside GNSS*, pp. 48–56, 2006.

[21] E. Vinanden and D. M. Akos, "Improvements to A Software-Defined GPS and Galileo Receiver: Single-Frequency Approach," *Proc. ION International Technical Meeting*, pp. 2230–2236, 2007.

[22] M. Hollreiser, "Galileo Receivers - Challenges and Performance," *Proc. 34th European Microwave Conference*, Amsterdam,pp. 321–324, 2004.

[23] C. O'Driscoll, D. Borio, M. Petovello, T. Williams and G. Lachapelle, "The Soft Approach: A Recipe for a Multi-System Multi-Frequency GNSS Receiver," *Inside GNSS*, pp. 46–51, September 2009.

[24] B. Muth, P. Oonincx and C. Tiberius, "BOC(m,n) signals: a Time-Domain Histogram Pattern," *European Journal Applied Signal Processing*, vol. 2007, April 2007.

[25] S. Gunawardena, "Development of a Transform-Domain Instrumentation Global Positioning System Receiver for Signal Quality and Anomalous Event Monitoring," Ph.D. dissertation, Russ College of Engineering and Technology of Ohio University, June 2007.

[26] O. Montenbruck, C. Gunther, S. Graf, M. Garcia-Fernandez, J. Furthner and H. Kuhlen, "GIOVE-A initial signal analysis," *GPS Solutions*, vol. 10, pp. 146–153, March 2006.

[27] M. Rapisarda, P. Angeletti and E. Casini, "A Simulation Framework for the Assessment of Navigation Payload Non-Idealities," *Proc. Navitec Conference*, 2008.

[28] M. Soellner, C. Kurzhals, W. Kogler, S. Erker,S. Thoelert, M. Meurer, M. Malik and M. Rapisarda, "One Year In Orbit : GIOVE-B CBOC E1 Signal Quality Assessment," *GPS Solutions*, pp. 28–38, 2009.

[29] D.J.R. Van Nee and A.J.R.M. Coenen, "New Fast Code-Acquisition Technique Using FFT," *Electronic letters*, vol. 27, pp. 158–160, 1991.

[30] D. Lin and J. B. Y. Tsui, "Comparison of Acquisition Methods for Software GPS Receiver," *Proc. ION GPS Conference*, pp. 2385–2390, 2000.

[31] J. S. Hong, J. W. Lee, G.-I. Jee, Y. J. Lee, J.W. Kim and C. G. Park, "GPS Signal Processing Algorithm for Software GPS Receiver," *Proc. ION GPS Conference*, pp. 2338–2345, 2000.

[32] P. Oonincx and B. Muth, "A Fractional Fourier Based Correlator for Detecting Joint Time and Frequency Offsets," *Proc. 15th International Conference on Digital Signal Processing*, Cardiff, pp. 623–626, 2007.

[33] P. Oonincx, H. Postma and B. Muth, "Generalized parallel-search strategies using the Fractional Fourier Transform," *Proc. European Navigation Conference (ENC-GNSS) 2008*, Toulouse, France, 2008.

[34] T. Humphreys, B. Ledvina, M. Psiaki, B. O' Hanlon and P. Kintner Jr, "Assessing the Spoofing Threat: Devlopment of a Portable GPS Civilian Spoofer," *Proc. ION International Technical Meeting*, pp. 2314– 2325, 2008.

[35] S. Lo, D. De Lorenzo, P. Enge, D. Akos and P. Bradley, "Signal Authentication: A Secure Civil GNSS for Today," *Inside GNSS*, pp. 30–39, 2009.

[36] B. Muth, "Software-based GNSS L1 Interferometric Positioning," *Proc. 21th ION GNSS Conference*, Savannah, 2008.

[37] B. Muth, P. Oonincx and C. Tiberius, "Differential Observables for Software-Based GPS Interferometry," *Proc. 17th European Signal Processing Conference*, Glasgow, pp. 2166–2170, 2009.

[38] P. Jales, R.Weiler, C. Underwood and M. Unwin, "First Spaceborne Demonstration of Galileo Signals for GNSS Reflectometry," *Proc. ION International Technical Meeting*, 2008.

[39] T. Elfouhaily, D. Thompson and L. Linstrom, "Delay-Doppler analysis of bistatically reflected signals from theocean surface: theory and application," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, pp. 560–573, 2002.

[40] S. Gleason and M. Unwin, "Development and testing of a remote sensing instrument using GNSS reflectometry concepts," *Proc. IEEE International Geoscience and Remote Sensing Symposium IGARSS'03*, vol. 7, pp. 4332– 4334, 2003.

[41] S. Esterhuizen and D. Akos, "The Design, Construction, and Testing of a Modular GPS Bistatic Radar Software Receiver for Small Platforms," Notes of the Workshop on GNSS-R, 2007.