GNSS Software Defined Radio

Real Receiver or Just a Tool for Experts?

JONG-HOON WON, THOMAS PANY, AND GÜNTER W. HEIN,

The ability to replace some hardware components in a GNSS receiver with software-based signal-processing techniques has already produced benefits for prototyping new equipment and analyzing signal quality and performance. Now some developers are attempting to extend the flexibility and cost-benefits of software defined radios to commercial end-user products, including mobile devices incorporating GNSS functionality. This column takes a look at the history of GNSS software receivers, the opportunities and practical engineering challenges that they pose for manufacturers, and the state of the art and related applications of them.

In recent years, a new trend in designing GNSS receivers has emerged that implements digitization closer to the receiver antenna front-end to create a system that works at increasingly higher frequencies and wider bandwidth. This development draws on an earlier software receiver (SR) or software defined radio (SDR) approach originating from signal processing technologies used in military applications.

Today, GNSS software receivers have achieved a level of considerable technological maturity and use, particularly in signal analysis and receiver engineering, and appear poised for much wider adoption in commercial equipment and applications. This column expands on an abbreviated introduction of this topic in our discussion of platforms for future GNSS receivers in a previous "Working Papers" (March, 2006).

In this column, we will briefly describe the history of SR development, introduce the categories of GNSS SR design and implementations to date, provide an overview of commercial SR products and applications, and the future outlook for GNSS SRs.

The flexibility of the programmable implementation of all baseband functions in software allows rapid change and modifications not possible in analog implementations.

History of GNSS Software Receivers

In the early 1990s the U.S. military services were facing several communications-related challenges. These included such matters as ensuring communication with current allies and a global support structure, denying interception of radio messages by hostile elements, taking advantage of the rapid technology changes, and controlling costs of R&D and purchasing.

At that time, military radio designs were based on hardware technology optimized for a single, specific field application and typically assumed a 30-year development life span, that is, the time from product release to the use of its next-generation replacement in new designs. During the 1990s, however, commercial applications started driving technology development so that the effective lifetime of a commercial component design fell to less than two years.

As a result of this change in the equipment design and development environment, a U.S. Department of Defense (DoD) multi-phase joint service project named Speakeasy was undertaken with the objective of proving the concept of a programmable waveform, multiband, multimode radio. (See the paper by R. J. Lackey and D. W. Upmal in the "Additional Resources" section near the end of this article.) The Speakeasy project demonstrated an approach that underlies most software receivers: the analog to digital converter (ADC) is placed as near as possible to the antenna front-end, and all baseband functions that receive digitized intermediate frequency (IF) data input are processed in a programmable microprocessor using software techniques rather than hardware elements, such as correlators.

The flexibility of the programmable implementation of all baseband functions allows rapid change and modifications not possible in analog implementations. This flexibility is an advantage in the military environment where communication at different ranges and to different command levels may require different radio frequency (RF) bands, modulation types, bandwidths, and spreading/despreading and baseband algorithms.

SDR is the underlying technology behind the Joint Tactical Radio System (JTRS) initiative to develop software programmable radios that enable seamless and real-time communication for the U.S. military services with coalition forces and allies. The functionality and expandability of the JTRS is built upon an open framework called the Software Communication Architecture.

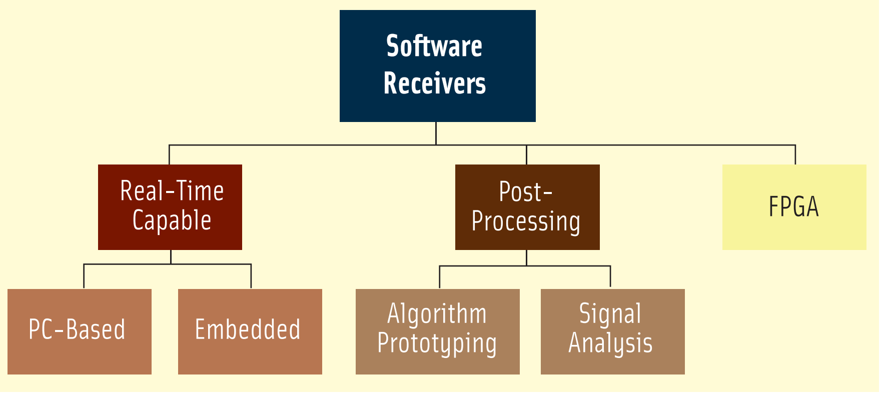


FIGURE 1 Different categories of software receivers

SR: A Functional Definition

Among researchers and engineers in field of communications and GNSS, some confusion has arisen over the terminology used to define a software receiver. For example, some communication engineers regard a receiver that contains multiple hardware parts for diverse systems, which can be reconfigured by setting a software flag or hardware pins of a chipset, to be an SR. In this article, however, we will use the widely accepted SR definition in the field of GNSS, that is, a receiver in which all the internal digital signal processing is carried out in a programmable processor by software techniques.

The internal functions of a modern GNSS receiver include an RF front-end block (typically including an antenna, low-noise amplifier, and RF integrated circuit or RFIC), initial signal acquisition, continuous signal tracking, bit and frame synchronizations, and navigation. A hardware-based receiver accomplishes the mixing of incoming and replica signals in a hardware correlator. Until the late 1990s the signal mixing function could only have been practically implemented in a hardware correlator due to the limited processing power of microprocessors at the time.

In 1990, however, researchers at the NASA/Caltech Jet Propulsion Laboratory introduced a signal acquisition technique for code division multiple access (CDMA) systems that was based on Fast Fourier Transform (FFT). This technique was enhanced by the work of researchers at the Technical University of Delft to apply FFT and inverse-FFT–based signal acquisition techniques for GPS. (For details, see the articles by D. J. R. van Nee, and A. J. R. M. Coenen cited in the "Additional Resources" section.) Since then, this method has been widely adopted in GNSS SR because of its simplicity and efficiency of processing load.

In 1996 researchers at Ohio University provided a direct digitization technique — called the bandpass sampling technique — that allowed the placing of ADCs closer to the RF portions of GNSS SRs. Until this time SRs implemented in university laboratories had a form of postmission processing because of the lack of processing power mentioned earlier. Finally, in 2001 Stanford University researchers implemented a real-time processing–capable SR for the GPS L1 C/A signal. (See the paper by Dennis Akos, P. L. Normark, and P. Enge in the "Additional Resources" section).

SR Types

Nowadays GNSS software receivers can be grouped into three main categories as shown in **Figure 1**. The majority of receivers are definitely found in the postprocessing subgroup "algorithm prototyping," which refers to the possibly countless number of small software tools or lines of code that are developed to test a new algorithm.

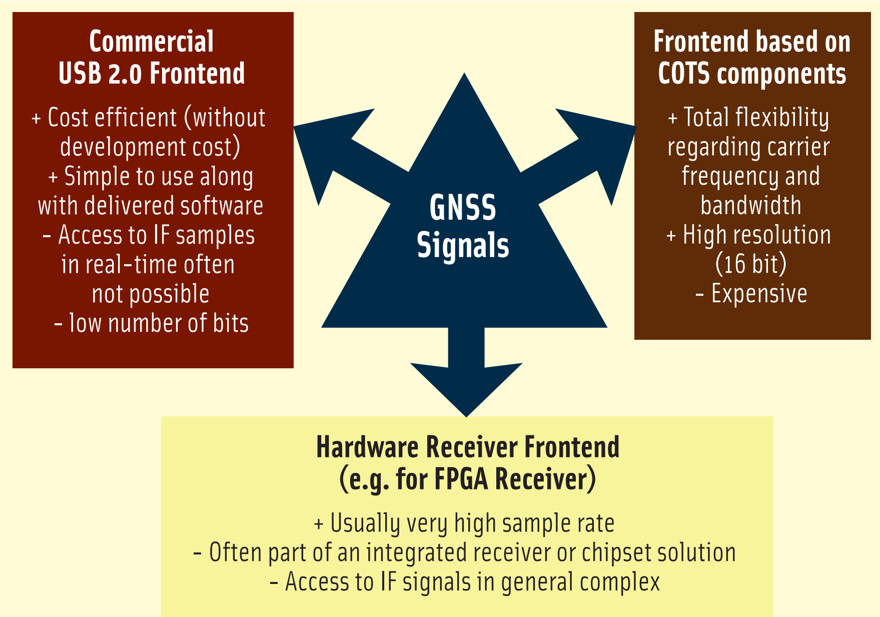
If the algorithm were tested with a real (or realistic) signal, one could already possibly speak of a software receiver. Another typical application of a postprocessing software receiver is GNSS signal analysis, for example, to investigate GPS satellite failures or to decrypt unpublished Galileo codes.

However, the GNSS SR boom really started with the development of real-time processing capability. This was first accomplished on a digital signal processor (DSP) and later on a conventional personal computer (PC). Today DSPs have been partially replaced by specialized processors for embedded applications, which have different features.

The hardware environment (processor speed, memory, available hard disk space) of the two platforms varies considerably; so, PC-based and embedded software receivers also differ substantially in the overall software design, even if they share common signal processing and navigation algorithms.

On a PC, C++ or even Matlab is the development environment; in the embedded sector, the C or assembler language dominates. Because the embedded sector (typically represented by mobile devices) has limited computational and power resources, high-end (and, thus, multifrequency) receivers will most likely always be PC-based receivers or eventually run on a workstation.

The last category, FPGA-based receivers, sometimes is also programmed in a C-like language. As they can be reconfigured in the field, one also can speak of them as being a software receiver. However, because their overall design (especially their integration with other hardware) is so different with respect to other PC-based and embedded GNSS SRs, FPGA-type receivers are not considered further in this article.



Commercial USB 2.0 Frontend

+ Cost efficient (without development cost)
+ Simple to use along with delivered software
- Access to IF samples in real-time often not possible
- low number of bits

GNSS Signals

Frontend based on COTS components

+ Total flexibility regarding carrier frequency and bandwidth
+ High resolution (16 bit)
- Expensive

Hardware Receiver Frontend (e.g. for FPGA Receiver)

+ Usually very high sample rate
- Often part of an integrated receiver or chipset solution
- Access to IF signals in general complex

FIGURE 2 Overview on advantages (+) and disadvantages (-) of GNSS SR front-end developments

Frontends for PC-based Receivers

Software receivers can nowadays be found at the commercial and university level. SR development not only includes programming solutions but also the realization of dedicated front-ends.

From the very beginning, the development of GNSS software receivers was undertaken side by side with the development of dedicated front-ends. PC-based software receivers in particular require a comparably complex interface to transfer the digitized IF samples into the computer's main memory. Historically, most GNSS SR developments started with the GP-2010, awell known GPS L1 RF chip from GEC (or Mitel) that provided the analog plus the digital IF at 4.309 MHz. ADC cards from ICS or National Instruments allowed continuous transfer of data into the PC and eventually storage on a hard disk for postprocessing.

From the very beginning, the development of GNSS software receivers was undertaken side by side with the development of dedicated front-ends.

Two classes of PC-based GNSS SR front-end solutions can be found today. The first one uses a commercially available ADC. This may be connected — for example, via the PCI bus — to the PC, or the ADC works as a stand-alone device. The ADC directly digitizes the received IF signal, which is taken from a pure analog front-end.

The second solution is based on integrating an ADC plus an USB 2.0 interface into the front-end, simultaneously providing the power supply to the ADC and the front-end. Regarding the analog part of the front-end, very different solutions exist based on either superheterodyne, low-IF or direct RF sampling. They are built using existing RF chipsets, discrete analog components, or commercial off-the-shelf (COTS) components. Figure 2 compares the relative advantages and disadvantages of the two types of solutions.

The first type of solution is still used at the university and research institute level, where a high amount of flexibility is required. For example, at the Department of Geomatics Engineering of the University of Calgary, a researcher uses a Novatel Euro-3M board from which they extract the digital IF samples. The samples are transferred via an FPGA board to a National Instruments NI-6534 data acquisition card plugged into a PC. L1 and L2 signals are sampled at 40 MHz but decimated to 5 MHz when they are transferred into the PC. (See the paper by Zheng, B. and G. Lachapelle cited in the Additional Resources section.)

Researchers at Cornell University (in work done cooperatively with the University of Texas at Austin) came up with a very interesting and cost-efficient solution for an L1/L2 front-end using the Zarlink GP-2015 chip. This chip is originally a GPS L1 front-end, and two GP-2015s were used to implement the L1 and the L2 signal path.

Because it was impossible to directly use the GP-2015 for L2, the L2 signal was upconverted to L1 by mixing the L2 RF signal with a signal whose frequency is 347.82 MHz, the difference between L1 and L2. L1 and L2 IF signals were then sampled within the GP-2015 at a rate of 5.714 MHz, and a National Instruments PCI-DIO-32HS (6533) digital I/O card was used to bring the data into the PC. (See the paper by B Ledvina, M. Psiaki, S. Powell, and P. Kittner in Additional Resources.)

At the University FAF Munich's Institute of Geodesy and Navigation, we started front-end development by opening a GPS architect receiver based on the GP-2010 chipset. We connected its analog IF signal via a National Instruments NI-5112 ADC plugged into the PC.

In the next step we specified an L1/L2 wide bandwidth (13 MHz) front-end, which the Fraunhofer Institute for Integrated Circuits then developed using discrete components for the analog part. Again the NI-5112 card was used to digitize the analog IF at about 8 MHz. A maximum transfer rate of 33 MHz (L1 only) or of 2x20 MHz (L1/L2) could be achieved in continuous mode with a resolution of 8 bits.

USB Solutions

Whereas our laboratory front-end solution is quite flexible, the development of front-ends with a universal serial bus (USB) 2.0 connector have arisen quickly, because this allows the front-end to be installed easily on the PC and provides a more cost-efficient solution. Currently, a number of commercial and R&D front-end are available which are summarized in Table 1.

NordNav and Accord were among the first to provide USB-based solutions, and recently the NordNav front-end was extended to permit connecting up to four antennas, which allows users to perform investigations in beam forming, indoor positioning, and GPS reflectometry. Accord is one of the first providers for a L2 frontend, suitable for tracking the new GPS L2 civil signal.

IfEN has announced (available Q3 2006) a high bandwidth L1 front-end, NavPort, shown in Figure 3, capable of processing GPS and Galileo signals. IfEN also plans a second generation NavPort, which additionally allows capturing a second (non-GNSS) signal, which can be synchronized to the GPS L1 signal. NavPort can record these arbitrary analog signals on the PC and automatically time-stamp them with an accuracy better then 1 microsecond on a sample per sample basis.

For instance, these could be signals from geophysical instruments such as a seismograph, generic radio signals, or many others. By using a flexible software signal decoder connected with for example the RS232 output pin of an inertial measurement unit (IMU), the serial output of the IMU could also be decoded and the IMU data is then synchronized to the GNSS measurements.

Another interesting development comes from the University of Colorado, which in an OpenGPS forum published all details on the RF and USB section. (See the paper by S. Esterhuizen in Additional Resources.) The Fraunhofer Institute for Integrated circuits provides single- or dual-frequency front-ends. These are built using discrete components for the analog section; front-end parameters — such as bandwidth, sample rate, or bit resolution — can be adapted to specific user demands.

FIGURE 3 Picture of a typical GPS/Galileo frontend with USB connector

In September, the Birkhäuser unit of Springer Science+Business Media will publish a textbook on software receivers, A Software-Defined GPS and Galileo Receiver, by Kai Borre and others. Accompanying the book will be a USB front-end based on the SiGe SE4110L chipset that, according to the manufacturer SiGe Semiconductor, specifically targets software GPS applications in the embedded sector.

A growing number of commercial software receivers are being developed for the embedded market that supports development of mobile devices with GNSS capabilities.

In summary then, the USB port is altogether very well suited for SR developments. Its maximum transfer rate of 480 MBit/s are even sufficient to realize a GPS/Galileo multi-frequency high bandwidth frontend. Such a front-end is currently specified at the University FAF Munich and under development at the Fraunhofer Institute for Integrated Circuits.

This new SR will receive GPS and Galileo signals on L1, L2, and L5 each with a bandwidth of 18.5 MHz. In triple-frequency mode, the transfer of large amounts of data into the PC uses two USB ports. As it receives all GPS signals and all Open Service Galileo signals, the receiver will facilitate research in high-precision applications and Galileo signal processing algorithm development.

With the availability of USB 2.0, PC-based software receivers have to be considered as a true GNSS SRs, not just an assembly of a bunch of components. The USB approach thus is one of the most important cornerstones of SR development. Currently the high transfer rate already poses few restrictions, and although wireless USB technology is on its way to the market, we expect that USB 2.0 will not be replaced in the near future.

Regarding the RF section, work still remains for the development of multifrequency and high bandwidth chipsets. In the meantime, software receivers for high precision applications will have to incor-

Source	Frequency Band	Bandwidth	Sample rate	Comment
NordNav	L1	Low, High	16 MHz	1, 2, or 4 RF channels
Accord	L1/L2	Low (2 MHz)	2-6 MHz	-
Fraunhofer	L1/L2	4 – 20 MHz	10-40 MHz	Tailored on user demands
Akos, Borre	L1	-	-	-
IfEN	L1	High 10 MHz	23 MHz	-
Univ. Colorado	L1	Low (2.2 MHz)	-	PCB and schematic available
Univ. FAF Munich/ Fraunhofer	L1/L2/L5	18.5 MHz	20 or 40 MHz	In development

TABLE 1. Overview of currently available fronts-ends for PC-based software receivers employing USB 2.0 connectors.

porate discrete components or use the RF part of existing hardware receivers.

SRs for the Embedded Market

A growing number of commercial software receivers are being developed for the embedded market that supports development of mobile devices with GNSS capabilities. Here are brief descriptions of some of the companies developing SRs.

NordNav. NordNav introduced the first software-based satellite navigation package at the commercial level. The embedded SR uses a host CPU to calculate position, saving space and processing power. It consists of a USB-type frontend, digital IF data sample streamer, a real-time 24-channel software receiver, and an application toolkit for research, development, test and verification purposes.

SiRF. As one of the leaders in GPS chip set development, SiRF Technology, also provides a software receiver for wireless handheld devices, called SiRFSoft. Using the software approach, the GPS baseband chip is replaced by optimized software running ultimately on an Intel XScale processor. The XScale family of applications processors is a frequent component in current mobile devices, such as smartphones and wireless PDA market segments, enabling these to maximize GPS performance while minimizing the load on the applications processors.

One of the most obvious and valuable applications of software GNSS receivers is their use in teaching and for training.

According to SiRF, the software receiver has a sensitivity of -159 dBm (which is the same value as for the SiRFstarIII receiver). The software receiver automatically adapts to utilize whatever network assistance data is available.

Center For Remote Sensing (CRS). CRS has released its Software GPS Builder for development and operation of advanced high performance GPS systems based on the SR approach. The company's solution includes the antenna and RF front-end, SR open architecture, a software GPS signal simulator, and the hard disk–based storage system for GPS L1 and L2.

All components are tuned in C-code, allowing direct implementation in a variety of platforms (microprocessors, DSP, FPGA, and ASIC). A series of CRS's utilities mentioned before allows whole design procedures including simulation, test, validation and implementation of a various kind of GNSS SR in one step process.

NAVSYS. A series of SR-related products provided by NAVSYS Corporation seem to be designed to focus on design and development for future GNSS requirements. The company's products have a capability of signal simulation in digital or RF level, logging digital data onto a storage device, and processing in real-time or play-back mode.

In addition, as one of the special applications of NAVSYS's GNSS SR technology, the company provides a network-based communication system in conjunction with a GPS SR, namely, the Position/Location tracking and Communications (POSCOMM) SDR. This unit combines observables from GPS and communication systems, such as pseudorange, carrier phase, and time-of-arrivals (TOAs), in an effort to solve weak signal problems caused by signal blockage, indoor environments, foliage attenuation, or jamming.

Philips. In 2005, Philips announced its new GNSS SR product, Spot, for mobile market. Spot eliminates the need for an expensive baseband processor by performing the required calculations on the application processor of the mobile device. An intellectual property (IP) approach, Spot performs position fixes in either an autonomous mode or, if assistance data is available from the communications network, in assisted-GPS mode. Depending on operational mode, it requires minimum performance specifications from the host processor.

RF Micro Devices (RFMD). RFMD has also unveiled their software-based GPS solution specifically for mobile devices, named as RF8110. As with SiRFsoft, the RF8110 software was optimized for use on the Intel XScale. In order to shorten customers' development cycles, RFMD also provides platform-specific hardware and software evaluation kits, for instance, Intel PXA270 applications processor and the Windows Mobile 5.0 operating system.

CellGuide. CellGuide has also announced the release on their GPS L1 C/A code SR, for mobile devices CDSoft. According to the released specification, CDSoft can run on most ARM and/or DSP-based processors with a variety of operating systems and provide time-to-first-fix of less than 6 seconds and sensitivity down to -157 dBm.

Commercial PC-Based Receivers

NordNav provided the first commercial GNSS SR that can be compared to a normal GPS receiver (and that was not a complete receiver development environment). The pioneer work, a GPS/SBAS L1 receiver, was also probably the first solution based on a USB front-end. Together with a software signal generator, Galileo signals can be tracked as well. Furthermore NordNav provides an application programming interface (API).

IfEN has announced to present at the Institute of Navigation's GNSS 2006 conference an SR solution capable of tracking GPS and Galileo signals on L1. This receiver uses large signal bandwidth to achieve highly accurate code and phase measurements based on a configurable multipath-mitigating correlator.

The receiver outputs two dimensional (delay/Doppler) multi-correlator values as well as FFT acquisition results suitable for signal quality monitoring. IfEN's solution can be reconfigured during runtime and comes with a hardware-based GPS/Galileo signal simulator.

SRs as Teaching Tools

One of the most obvious and valuable applications of software GNSS receivers is their use in teaching and for training.

In particular, receivers for which the source code is available allow inspection of almost all signal data by the researcher. Of course, commercial software receivers are also of interest, because they allow real-time configuration and nice visualization possibilities.

Textbooks already have been published on GNSS SRs. The pioneer in this area is definitely James Bao-yen Tsui who authored the first book on GNSS software receiver algorithms in 2000, *Fundamentals of Global Positioning System Receivers: A Software Approach* (Wiley-Interscience, publisher, updated in 2004).

As mentioned earlier, a new book by Kai Borre et al will appear in September. In it, the authors focus on real-time SR operation. Other web-based resources on software receivers can be also be found in the article by R. Babu cited in Additional Resources.

The European Union is fostering the development of receivers for the upcoming Galileo system. One of the projects it has funded through the Galileo Joint Undertaking is the Galileo Receiver Analysis and Design Application (GRANADA) simulation tool. Running under Matlab, GRANADA is conceived as a modular and configurable tool with a dual role: test-bench for integration and evaluation of receiver technologies, and SR as asset for GNSS application developers.

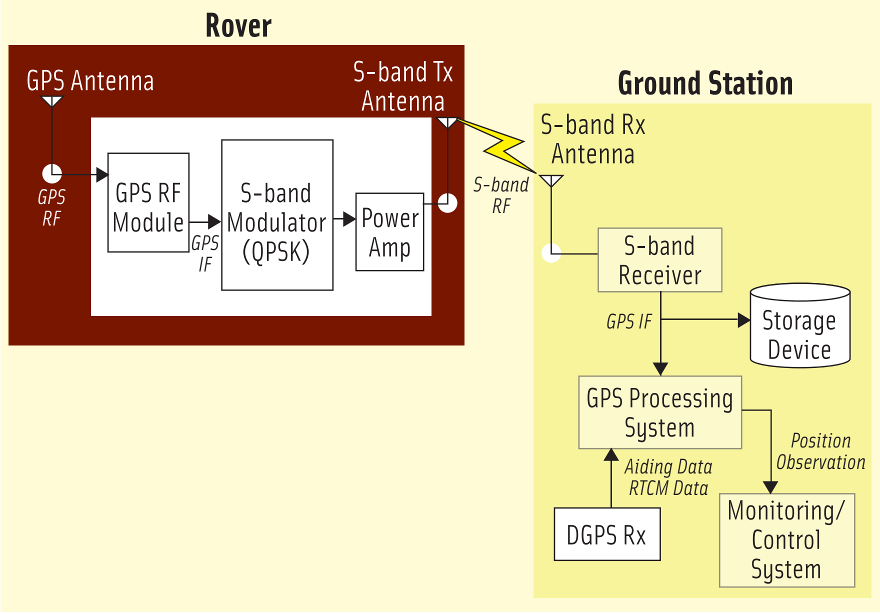


FIGURE 4 Scheme on GPS Translator

The Data Fusion Corporation (DFC) provides an IF-level GNSS SR toolbox in Matlab and C and for DSP configurations, which seems ideal for research and development work in postprocessing mode. The toolbox consists of a GPS baseline receiver and GPS signal generator toolboxes, of which all sources are open to customers and can be modified to a specific algorithm test.

NAVSYS Corporation also provides a signal simulation and analysis tool to simulate the effect of GPS satellite signals on a conventional GPS receiver's code and carrier tracking loops as a form of Matlab Toolbox. This signal simulation tool can be used as an analysis aid to help test and evaluate GPS receivers beyond the capability of conventional RF signal simulators, which frequently do not provide low-level insight into the operation of a GPS receiver.

The NAVSYS product consists of geographical tools, satellite geometry tools, and receiver design and analysis tools. A particular aspect of its signal simulation tool is the ability to simulate both GPS L1 and L2 datasets for playback and analysis by built-in receiver modules within the Matab tools. However, the system is also able to play back data into a GPS receiver under test as live digital or RF signals using an advanced GPS simulator product.

Selected SR Applications

Apart from the previously mentioned uses (algorithm prototyping, the embedded sector, and teaching), SRs are especially suited for some other GNSS applications (shown in Table 2) due to their outstanding flexibility. In this section, we will discuss a selection of these.

GPS Translator System. The GPS translator system was designed initially as a proof-of-principle system for U.S. DoD military missile development programs. This system consists of a missile-based GPS measurement sensor (the rover

Application	Why
GPS/INS integration	Easy access to tracking loops and convenient development environment
GPS reflectometry	Detailed signal analysis in post processing, with arbitrary number of correlators
Indoor reference receiver	Combine indoor and outdoor signals at signal processing level yielding an accurate determination of signal attenuation
Phased array antennas	Simple data flow inside the receiver of the different antennas and easy access to signals
GPS translator system	Flexible receiver at server side. RF frontend as part of weapon which is finally destroyed is cheaper than complete receiver.
Network based positioning of mobile phones for E911	Flexible receiver at server side
Signal analysis tool	Possibility to implement dedicated analysis and monitoring algorithms with visualization capabilities
High-end receiver	Possibility to implement high-end signal processing and navigation algorithms such as frequency domain signal processing, multi-correlator, vector delay lock loop, and so on.

TABLE 2. Current (black) and future (red) applications of software receivers

unit) and a ground-based server system (ground station) as shown in Figure 4.

The rover system comprises a GPS receiver RF front-end and a wireless S-band transmitter. The ground station system incorporates an S-band receiver, a GPS processor, a differential GPS (DGPS) reference receiver, and a monitoring/control system.

The GPS signal (RF) is received at the end of the GPS antenna in the rover system and down-converted into a lower IF. This IF signal is modulated, up-converted, and then transmitted through the S-band channel to the ground station. At the ground station, the received S-band signal is down-converted again to a lower IF and processed by the high performance ground processor to provide the position of the rover. Thus, all the digital signal processing on the GPS signal are performed in the ground station not in the rover.

This system design makes it possible to employ advanced processing techniques without any constraint in cost. The DGPS technique includes Doppler and ephemeris-aiding techniques in signal acquisition/tracking and determining the position of the rover accurately and robustly. In addition, as recommended by the military specifications, the IF signal in the rover should be collected and recorded to storage devices, enabling playback after the test for more precise analysis.

Figure 4 shows that the overall structure of the translator system is similar to that of a software receiver, excluding the S-band data-link.

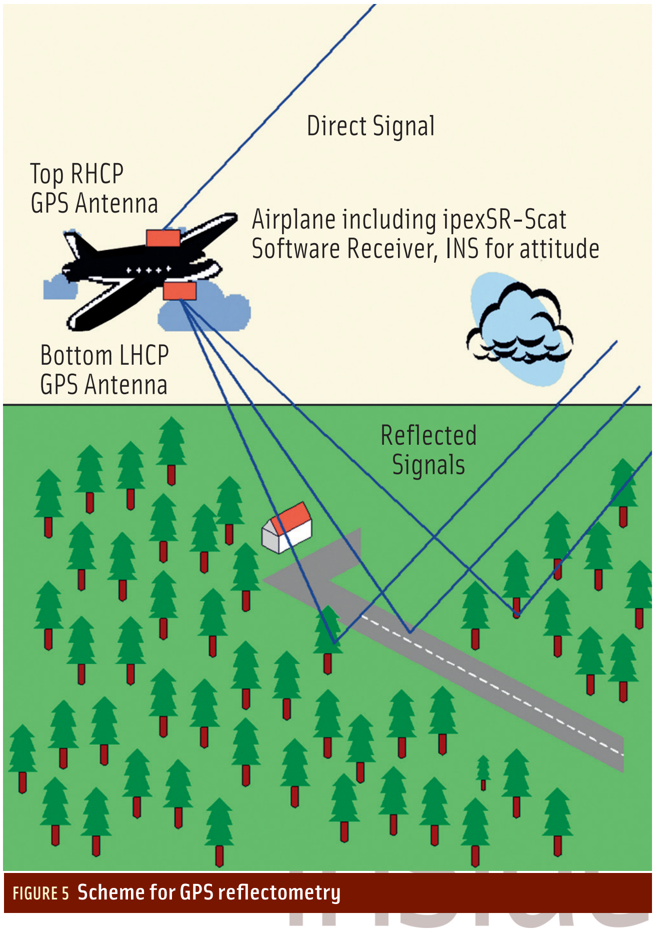


FIGURE 5 Scheme for GPS reflectometry

GPS Science and GPS Reflectometry. In GPS science we use GPS or GNSS signals to investigate scientific phenomena by analyzing the received GNSS waveform in great detail. For example, in GPS meteorology atmospheric parameters such as humidity and temperature are evaluated in terms of the signal path delay.

In GPS reflectometry, reflected GPS signals from the Earth's surface are used to determine parameters like surface height, sea-level height, wave height (and thus wind speed and direction), salinity, humidity, etc. In a typical airborne system, as shown in Figure 5, the GPS receiver is connected to two antennas, one at the bottom and one at the top of the aircraft. The top mounted antenna receives the direct GPS signal, and the bottom antenna the signal reflected from the Earth's surface.

The top-mounted antenna is typically right-handed circular polarized (RHCP) and the bottom-mounted antenna, left-handed circular polarized (LHCP). The GPS satellites emit RHCP signals and, when they are reflected, their polarization changes to LHCP. GPS reflectometry systems have initially been realized using FPGA boards, but recently the PC-based SR approach seems to be more promising.

For example, in the article on GPS backscatter measurement by T. Lingren et al, cited in Additional Resources, about 100 minutes of data were collected using a dual-antenna NordNav front-end. This data was analyzed in a postprocessing phase using dedicated in-house software. This postprocessing method enabled the researchers to re-analyze the reflected signal in very fine detail.

Occasionally their investigation found — apart from the surface echo — an additional echo of unknown origin. The analysis later showed that this echo originated from single objects (for example, a farm). By using signals from multiple satellites, the researchers could locate the reflector on a map. In the context of this column, we should note that this new scientific finding was most likely only possible by having used a software receiver in postprocessing mode.

Generic Signal Analysis Tools. One of the essential goals of a high-end software receiver is its application as the ultimate signal analysis tool. Such a receiver should be an instrument that combines digital storage, oscilloscope, spectrum analyzer, GPS receiver, and possibly also INS data.

An example of such an instrument is described in the article by A. Soloviev, S. Nawardena, and F. van Graas. With it, one simply connects a GNSS antenna and the system reveals everything that you can possibly know about this signal, in real-time or postprocessing (including of course the position). In this context, the processing of the GPS P(Y)-code on L2 is important, because analyzing the propagation effects on that signal — in combination with the L1 signal — is still the only way of eliminating ionospheric errors, and thus allowing high precision applications. The instrument described by Soloviev et al. has accomplished this goal in a software receiver.

Our own developments at the Institute of Geodesy and Navigation also go in this direction. Of special interest are signal quality monitoring and spectral monitoring applications. For signal quality monitoring, multiple correlators are used, each located at a certain code phase offset and Doppler offset with respect to the prompt correlator.

The values (or special combinations of them) are compared against their nominal values permanently. Failures in the transmitter (satellites) or high multipath causes deviations from the nominal values and thus can be detected. By monitoring the spectrum, we are able to assess the quality of the RF environment, including detection of intentional or unintentional jamming.

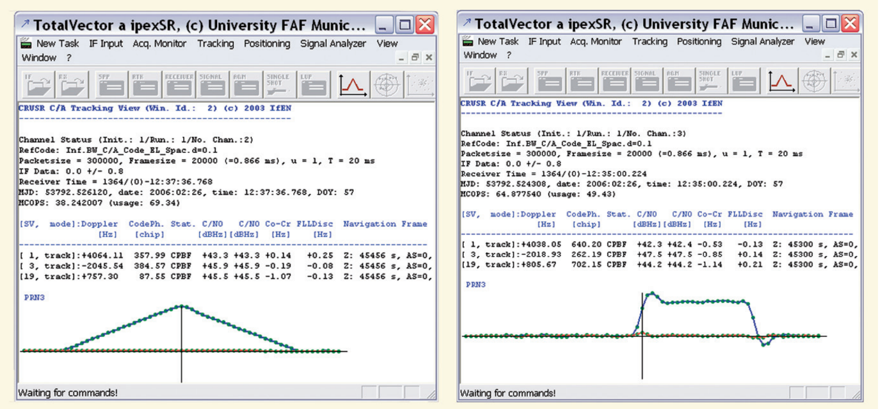
A recent improvement in our software is the waveform multi-correlator (inspired by Novatel's Vision correlator), which correlates the received signal with a PRN code convoluted with Dirac's Delta-function. This method allows us to reconstruct the waveform of the received satellite signal. The waveform is a more direct measure of the GNSS signal compared to the correlation function. Thus, failures — for example, in the satellite — can be more easily detected and analyzed.

In **Figure 6** the same wideband signal for GPS satellite PRN3 is correlated once with a normal multi-correlator (screenshot on left) and once with the waveform multi-correlator (right). In the standard multicorrelator plot the distortions due to the filter in the receiver's frontend can be barely identified, but they are clearly visible in the waveform correlator.

A major SR drawback persists, namely, the slow throughput compared to application specific integrated circuits (ASICs), its hardware counterparts.

Outlook

For a long time, software receivers have already found their place in the field of algorithm prototyping. Nowadays they also play a key role for certain special applications. What remains unclear is whether they will succeed as generic high-end receivers or if they can penetrate the embedded market.



```
CRUSR C/A Tracking View (Win. Id.:  2) (c) 2003 IfEN
----------------------------------------------------

Channel Status (Init.: 1/Run.: 1/No. Chan.:2)
RefCode: Inf.BW_C/A_Code_EL_Spac.d=0.1
Packetsize = 300000, Framesize = 20000 (=0.866 ms), u = 1, T = 20 ms
IF Data: 0.0 +/- 0.8
Receiver Time = 1364/(0)-12:37:36.768
MJD: 53792.526120, date: 2006:02:26, time: 12:37:36.768, DOY: 57
MCOPS: 38.242007 (usage: 69.34)

[SV,  mode]:Doppler  CodePh. Stat. C/NO   C/NO Co-Cr FLLDisc  Navigation Frame
               [Hz]   [chip]       [dBHz][dBHz]  [Hz]     [Hz]
-------------------------------------------------------------------------------
[ 1, track]:+4064.11  357.99 CPBF  +43.3 +43.3 +0.14   +0.25  Z: 45456 s, AS=0,
[ 3, track]:-2045.54  384.57 CPBF  +45.9 +45.9 -0.19   -0.08  Z: 45456 s, AS=0,
[19, track]:+757.30    87.55 CPBF  +45.5 +45.5 -1.07   -0.13  Z: 45456 s, AS=0,
```

```
CRUSR C/A Tracking View (Win. Id.:  2) (c) 2003 IfEN
----------------------------------------------------

Channel Status (Init.: 1/Run.: 1/No. Chan.:3)
RefCode: Inf.BW_C/A_Code_EL_Spac.d=0.1
Packetsize = 300000, Framesize = 20000 (=0.866 ms), u = 1, T = 20 ms
IF Data: 0.0 +/- 0.8
Receiver Time = 1364/(0)-12:35:00.224
MJD: 53792.524308, date: 2006:02:26, time: 12:35:00.224, DOY: 57
MCOPS: 64.877540 (usage: 49.43)

[SV,  mode]:Doppler  CodePh. Stat. C/NO   C/NO Co-Cr FLLDisc  Navigation Frame
               [Hz]   [chip]       [dBHz][dBHz]  [Hz]     [Hz]
-------------------------------------------------------------------------------
[ 1, track]:+4038.05  640.20 CPBF  +42.3 +42.4 -0.53   -0.13  Z: 45300 s, AS=0,
[ 3, track]:-2018.93  262.19 CPBF  +47.5 +47.5 -0.85   +0.14  Z: 45300 s, AS=0,
[19, track]:+805.67   702.15 CPBF  +44.2 +44.2 -1.14   +0.21  Z: 45300 s, AS=0,
```

FIGURE 6 **Comparison of standard (left) and waveform (right) multi-correlator**

A GNSS SR has multi-phase advantages including design flexibility, faster adaptability, faster time-to-market, and easy optimization at any algorithm stage. As a result, it is emerging as an important technology in both commercial and military applications. However, a major SR drawback persists, namely, the slow throughput compared to application specific integrated circuits (ASICs), its hardware counterparts.

In order to overcome this drawback, GNSS SRs embedded into a single microprocessor or DSP in a software form is emerging in the marketplace. Until now and for the near future, this development is driven by the mobile application market, for example, mobile phones with PDA functionality based on location based services (LBS) applications.

This application requires location-enabling chips located in the handset (or handset adaptation package) or as a software package with minimal modification to the handset. Without any doubt, a GNSS SR integrated into existing platform and operating system of mobile devices should be one of the good solutions for this application.

The remaining competition will be over the type of platform and operating system best suited for increasing performance of SRs. Conversely, the question could be posed as how a GNSS software receiver should be designed or modified for the existing platforms and operating systems of mobile devices.

On the opposite end of the spectrum from the mass market, the following factors seem to ensure that, sooner or later, high-end software receivers will be available:

- High bandwidth signals on L1/L2 can already be transferred into the PC in real-time and processed. Development of triple-frequency USB front-ends for GPS and Galileo is underway
- Due to the increasing processing power, real-time processing with a limited amount of multi-correlators is already possible, and software receivers will definitely benefit from the introduction of new multi core processors as discussed in the March issue's Working Papers.
- Postprocessing is one of the major benefits of a software receiver as it allows re-analysis of the same signal several times with all possible processing options, and the rapidly increasing hard disk capacity allows storage of very long time spans and datasets.
- Some signal processing algorithms, such as frequency domain tracking (e.g., the symmetric phase-only matched filter as introduced by the U.S. Air Force Research Lab and Sigtem) or maximum likelihood tracking for high dynamics applications, are much easier to implement in software than in hardware. Those methods require complex operations

at the signal level or multiple reprocessing of the received signal.

Regarding the development and integration costs, embedded and high-end software receivers definitely beat their hardware counterparts and, on the other side, power consumption, especially of high-end SRs, is still problematic. Just one thing is for certain: this new technology increases the options one has to solve a particular positioning problem.

Additional Resources

[1] Akos, D. M., Normark, P. L. and Enge, P., "Real-time GPS Software Radio Receiver", Proc. ION-GPS NTM 2001, Jan. 2001, Long Beach, CA.

[2] Babu, R., "Web-based resources on software GPS receivers", *GPS Solutions,* vol. 9, no. 3, Sept. 2005, pp. 240-242

[3] Borre, K., D. Akos, D. Bertelsen, P. Rinder and S. Jensen, *A Software-Defined GPS and Galileo Receiver – A Single Frequency Approach,* Birkhäuser, ISBN: 0-8176-4390-7, 2006

[4] Cheng, U., Hurd, W. J. and Statman, J. I., "Spread-Spectrum Code Acquisition in the Presence of Doppler Shift and Data Modulation," *IEEE Trans. on Comm.*, Vol. 38, No. 2, Feb. 1990, pp. 241-250

[5] Coenen, A. J. R. M., and D. J. R. van Nee, "Novel Fast GPS/Glonass Code Acquisition Technique Using Low Update Rate FFT," *Electronics Letters,* Vol. 28, No. 9., Apr. 1992, pp. 863-865

[6] Esterhuizen, S., "The Design, Construction, and Testing of a Modular Bistatic Radar Software Receiver for Small Plattforms," Ph.D. Thesis, University of Colorado, http://ccar.colorado.edu/opengps/loki/, 2006

[7] Data Fusion Corporation, http://www.datafusion.com/gps.htm

[8] GRANADA, Galileo Receiver Analysis and Design Application, http://www.deimos-space.com/granada/

[9] Hein, G., T. Pany, S. Wallner and J.-H. Won, "Platforms for a Future GNSS Receiver", *Inside GNSS,* March 2006, pp. 56-62

[10] Lackey, R. J., and D. W. Upmal, "Speakeasy: The Military Software Radio," *IEEE Communications Magazine,* May 1995, pp. 56-61

[11] Ledvina, B., M. Psiaki, S. Powell and P. Kittner, "Real-Time Software Receiver Tracking of GPS L2 Civilian Signals using a Hardware Simulator," Proc. ION-GNSS 2005, Sept. 13-16, Long Beach, pp. 1598-1610

[12] Lingren T., E. Vinande, D. Akos, D. Masters and P. Axelrad, "Measurement of Backscattered GPS Signals," Proc. IEEE/ION PLANS 2006, April 25-27, San Diego, pp. 664-669

[13] Philips, http://www.software.philips.com/about/technologies/spot/index.html

[14] Soloviev, A., S. Nawardena and F. van Graas, "Development of High Performance High Update Rate Reference GPS Receiver," Proc. ION-GNSS 2005, Sept. 13-16, Long Beach, pp. 1621-1631

[15] Tsui, J., *Fundamentals of Global Positioning System Receivers: A Software Approach,* Wiley-Interscience, ISBN 0471712574, 2004

[16] van Nee, D. J. R., and A. J. R. M. Coenen, "New Fast GPS Code Acquisition Technique Using FFT," *Electronics Letters,* Vol. 27, No. 2, January 1991, pp. 158-160

[17] Zheng, B. and G. Lachapelle, "GPS Software Receiver Enhancements for Indoor Use," Proc. ION-GNSS 2005, 13-16 Sept., Long Beach, pp. 1138-1142

Authors

Thomas Pany has a Ph.D. in geodesy from the Graz University of Technology and a MS in Physics from the Karl-Franzens University of Graz. Currently he is working at the Institute of Geodesy and Navigation at the University of Federal Armed Forces Munich. His major areas of interest include GPS/Galileo software receiver design, Galileo signal structure and GPS science.

Jong-Hoon Won received his B.S. and M.S. degrees in control and instrumentation engineering at Ajou University, Suwon, Korea. He also received his Ph.D. at the same university based on the design and development of a software-based GPS receiver and precise navigation algorithm. Since 2005, he has worked as a research associate with the Institute of Geodesy and Navigation of the University of Federal Armed Forces Munich, Germany. His current research interests are in the signal processing and navigation algorithms for GNSS receivers.

"Working Papers" explore the technical and scientific themes that underpin GNSS programs and applications. This regular column is coordinated by **PROF. DR.-ING. GÜNTER HEIN** Prof. Hein is a member of the European Commission's Galileo Signal Task Force and organizer of the annual Munich Satellite Navigation Summit. He has been a full professor and director of the Institute of Geodesy and Navigation at the University FAF Munich since 1983. In 2002, he received the United States Institute of Navigation Johannes Kepler Award for sustained and significant contributions to the development of satellite navigation. Hein received his Dipl.-Ing and Dr.-Ing. degrees in geodesy from the University of Darmstadt, Germany. Contact Professor Hein at <Guenter.Hein@unibw-muenchen.de>. IG

(Continued from page 47)

User costs also are incremental, including all changes that occur with the availability of L2C, and are net of savings from moving to less sophisticated and less proprietary equipment.

Acknowledgments

Steve Bayless, Tyler Duval, Jason Kim, Scott Pace, Mike Shaw, Tom Stansell, Dave Turner, Jack Wells, Rodney Weiher, and Avery Sen offered comments, guidance and assistance to the study. Many others contributed expertise through interviews.

Additional Resources

Kenneth W. Hudnut, and Bryan Titus, *GPS L1 Civil Signal Modernization (L1C),* Interagency GPS Executive Board, July 30, 2004, <http://www.navcen.uscg.gov/gps/modernization/L1/L1C-report-short.pdf>

U.S. Defense Science Board, *The Future of the Global Positioning System,* Washington, D.C.: Office of the Under Secretary of Defense For Acquisition, Technology, and Logistics, October 2005, <http://www.acq.osd.mil/dsb/reports/2005-10-GPS_Report_Final.pdf>

Author

Irving Leveson, president of Leveson Consulting in Jackson, New Jersey, U.S.A., is an economist and strategic planner. He is principal investigator and author of the *L2C Study* analysis for the United States Departments of Commerce and Transportation. He previously held senior positions with Hudson Strategy Group and Hudson Institute and worked with the RAND Corporation and the National Bureau of Economic Research. IG