Simulating GPS Signals

It Doesn't Have to Be Expensive

Alison Brown, Jarrett Redd, and Mark-Anthony Hutton

WHY DO WE SIMULATE REALITY in mathematics, science, engineering, and other pursuits — even in our recreational activities? Well, we do it for a variety of reasons. In mathematics and science, we try to comprehend reality, which is complicated and variable and often has some degree of randomness. We build mathematical models of physical, chemical, or biological processes to better understand them or to predict a particular outcome given some initial conditions. The model may contain a stochastic component to reflect a degree of uncertainty associated with the processes. Weather forecasting is a prime example. Typically, the models are run on a computer where the model parameters and initial conditions can be readily adjusted and the varying outcomes analyzed.

INNOVATION INSIGHTS
with Richard Langley

Simulation: faster, cheaper, safer.

Simulations of reality are often used in teaching where students can more easily grasp the behavior of complicated systems whether they be in the natural sciences or in economics or the social sciences. In medical education, simulated human patients are used initially because it is safer than having students operate on real patients. Similarly, flight simulators are used for the training of pilots because it is cheaper and safer than using real aircraft and a wide variety of "what if" scenarios can be experienced.

Simulation is used for a range of engineering activities to see how an existing system behaves under different conditions because it is faster or cheaper than performing tests in the "real world." It can also be used to estimate how a proposed new system might behave before it becomes a reality — looking at traffic flow in road networks, for example.

We also use simulation for recreation, whether it is playing with the latest computer game or improving our swing with a golf simulator. And simulation is a mainstay of the movie industry.

But getting back to engineering and the main interest of this magazine, simulation is a useful technique in the design and operation of equipment used with global navigation satellite systems. With a radio frequency simulator, we can mimic the radio signals generated by the satellites. These devices allow us to define scenarios, including receiver trajectories, and to replay them while varying the operating parameters of the receiver. Some simulators allow us to record live signals and then to play them back under different assumed conditions.

However, such GNSS signal simulators can be expensive and beyond the limited budgets of many researchers. In this month's column, we look at one company's approach to providing GNSS signal simulation at a low cost — one that virtually any researcher can afford.

As the noted French sociologist and philosopher, Jean Baudrillard, pessimistically once said: "We live in a world where there is more and more information, and less and less meaning." In the field of GNSS engineering, at least, simulation is helping to stem the tide and give us a better understanding of reality.

Embedded GPS receivers have become commonplace with the proliferation of GPS navigation systems into all but the least expensive vehicle and cell-phone lines. As more manufacturers embed low-cost GPS receivers into their products, the need for low-cost GPS signal simulators has also grown. Controlled virtual testing is vital in ensuring the expected system performance.

Many GPS signal generators are available that are designed specifically for high-volume production test applications for devices that use commercial GPS/SBAS, GLONASS, and Galileo receivers. Often the cost of these high-end simulators is beyond the reach of small companies or universities. In response to this need, we have developed our low-cost, software-defined radio (SDR)-based GPS Signal Architect Test Set to address a broad range of research, academic, industrial, and defense applications. The system is designed to be flexible, scalable, and most importantly, inexpensive.

Our test set leverages the capabilities of the Universal Software Radio Peripheral (USRP) radio and our GPS Signal Simulation Toolbox to provide users with a GPS signal generation capability at a much lower cost than currently available on the market. The combination of the GPS signal simulation software coupled with the record and playback capability of the USRP makes for an extremely low-cost, yet highly flexible, GPS signal simulation capability.

FIGURE 1 shows the GPS signal simulator hardware. It is designed for use with commercial software-defined radios and is based on our GPS Signal Simulation Toolbox.

Toolbox. The Toolbox is a complete set of GPS signal simulation, test, and

analysis tools. This Matlab-based signal simulation toolbox simulates the effect of the signal degradation on a conventional commercial GPS receiver, including the effect of the ionospheric activity on the code and carrier tracking loops such as losing lock or cycle slipping. The Toolbox's geographic tools facilitate the transformation of data between the various coordinate systems commonly used in GPS research, including latitude-longitude-altitude; WGS 84 Earth-centered, Earth-fixed (ECEF); north-east-down; and body-fixed reference frames. It also provides tools to read GPS almanacs and ephemerides and compute ECEF and line-of-sight vectors to GPS satellites as a function of user position and time. The receiver design and analysis tools model different receiver architectures and simulate different error scenarios by providing tracking and navigation algorithms, including phase lock loops and delay lock loops.

The user-configurable options allow the operator to define virtually all aspects of a GPS signal environment, including the GPS spreading code(s), navigation message, and interference scenarios. Such flexibility is particularly useful in simulating GPS jamming environments, where time, resources, and repeatability are generally scarce.

Because these tools are linked directly to Matlab, it is relatively simple to define and implement new signal components as they become available. Of primary interest are the GPS modernization codes as well as those of other global navigation satellite systems (GNSS). Also of interest are new and exotic categories of jammers, including frequency-modulated, amplitude-modulated, phase-modulated, and frequency-swept jammers.

An early version of the toolbox was reviewed in a previous Innovation column (see Further Reading).

Configuration. In the configuration shown in Figure 1, the signal control unit (SCU) is used to control the radio for record and playback operation. The USRP includes a 10-MHz frequency standard as well as an input for an external reference clock. The GPS Signal Architect software can produce custom GPS scenario data files, which can use the USRP to produce a GPS signal at RF.

This article provides a review of how the signal simulator uses the USRP family of radios as low-cost RF record and playback devices using the Signal Architect files. In addition, the hardware design and supported signals are described and test results are presented showing the USRP providing simulated GPS signals to conventional GPS user equipment.
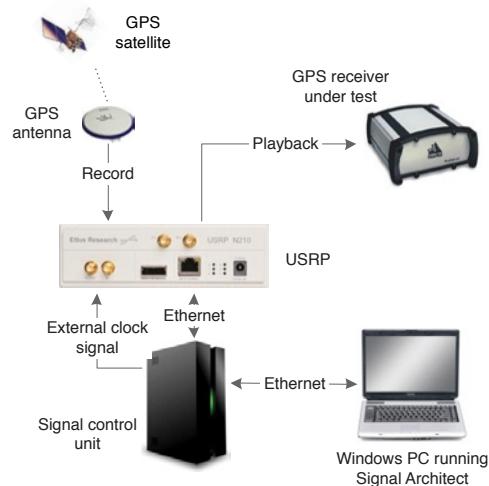
▲ **FIGURE 1** GPS signal simulator hardware.

▲ **FIGURE 2** Univeral Software Radio Peripheral.

Radio Hardware

The USRP radio family provides an inexpensive development platform for software-defined radios. The USRP can also be used to record and play back the GPS signal in a static or mobile environment. The system operator can then reproduce the signal on the bench either from a simulated profile or from a previously recorded test environment. An advantage of the USRP is that it supports a wideband transceiver front-end that can accommodate the full 20 MHz of the GPS signal band and can be tuned to operate at any of the GPS signal frequencies (L1 at 1575.42 MHz, L2 at 1227.60 MHz, or L5 at 1176.45 MHz). This allows record and playback of both the civil and military GPS codes.

While the GPS Signal Architect tools can be easily adapted for use with any commercial SDR, the USRP family was chosen because of its reasonable price, quality construction, and extensive support by the GNU Radio project.

USRPs are SDRs, which can, in principle, transmit or receive signals on any frequency under software control. Typically, USRPs connect to a host computer through a high-speed USB or gigabit Ethernet link, which the host-based software uses to control the USRP hardware and to transmit or receive data. Some USRP models also integrate the general functionality of a host computer with an embedded processor that allows the USRP to operate in a standalone fashion. The USRP hardware is controlled through a hardware driver, which supports Linux, MacOS, and Windows platforms. A framework running on the host computer then accesses the USRP through the driver. Several frameworks, including GNU Radio (a free software toolkit for learning about, building, and deploying SDR systems developed under the GNU Project — "GNU" is a recursive acronym that stands for "GNU's Not Unix"), LabView, Matlab, and Simulink, use the driver. The driver's functionality can also be accessed directly with an application programming interface (API), which provides native support for C++. Any other language that can import C++ functions can also use the driver. This is accomplished in Python through the Simplified Wrapper and Interface Generator, for example. The API allows users to develop their own custom frameworks, as we did with our SCU.

Of the available USRP radios, the N210 was chosen because it has the highest sample rate, greatest

flexibility, and largest capacity for modification (see **FIGURE 2**).

The USRP provides an interface between high-speed analog to digital converters, high-speed digital to analog converters, and an Ethernet interface, as previously mentioned. Daughterboards available for the USRP provide an interface from the baseband signals present at the data converters to the GPS frequency bands and vice versa.

A daughterboard (or daughtercard or piggyback board) is a circuit board meant to be an extension or "daughter" of a motherboard. The USRP uses interchangeable daughterboards, plugging into the main board, to serve as the RF front end. Several classes of daughterboard modules exist: receivers, transmitters, and transceivers. Transmitter daughterboard modules can modulate an output signal to a higher frequency; receiver daughterboard modules can acquire an RF signal and convert it to baseband; and transceiver daughterboard modules combine the functionality of a both a transmitter and receiver.

For this project, a WBX (wide bandwidth) transceiver daughterboard was installed in the USRP. The tunable range of the WBX (50 MHz to 2.2 GHz) covers all the current GNSS frequencies. An RF pre-filter is used to band-limit the GNSS signals to the sample rate selected for use in the SCU to avoid spectral folding from the N210 40-MHz channel bandwidth. For example, a 2-MHz filter centered at L1 is optimal based on the Nyquist sampling frequency of 2 MHz of both the in-phase and quadrature (I/Q) components of the signal. If sampling at 20 MHz, then a 20-MHz pre-filter should be used.

Signal Control Unit

The SCU includes a Linux single-board computer with software developed to run under the GNU Radio Companion (an open-source graphical tool for cre-
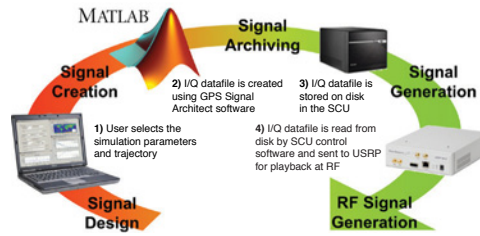
▲ **FIGURE 3** Signal Architect simulation flow.
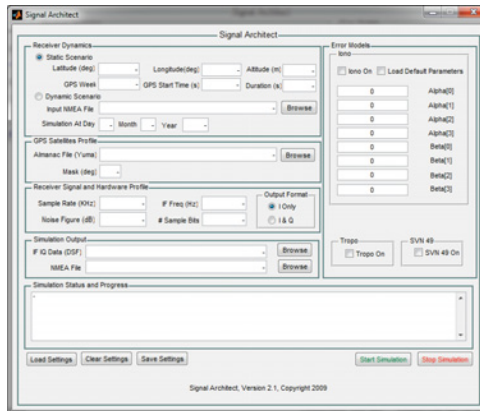
▲ **FIGURE 4** Signal Architect graphical user interface.

Signal Architect SW	
Operating system	Windows XP/Win7/Linux
Multi-core support	Minimum Core i5
Supported GPS signals	GPS L1 (1575.42 MHz) C/A and P
Future system options	GNSS, multi-frequency (GPS/GLONASS)
SDR Control Unit	
RF in/out	SMA connectors for input/output antenna for RF signal record/replay
PC	Interface: gigabit Ethernet (GbE) Operating system: Linux
RF record/replay	Channel 1: center frequency: 1575.42 MHz
	Sample frequency: 2–20MHz
	Sampling: 1 or 2-bit I/Q
	Channel bandwidth set by RF pre-filter to match Nyquist bandwidth (sample rate)
Future system options	Multi-RF channel L1+L2+L5 +GLONASS

▲ **TABLE 1** Signal simulator system specifications.

ating signal flow graphs and generating flow-graph source code using the GNU Radio libraries) and management of the GNU SDR for RF record and playback under control of the GPS Signal Architect software through an Ethernet connection. This enables the user to tap into the excellent USRP community support for his or her project and benefit from the close relationship between the GNU Radio project and the USRP manufacturer. The Ethernet connection is also used to download and upload recorded or simulated signal files from the Signal Architect signal simulation software.

Signal Simulation Software

The GPS Signal Architect hardware and software provides users with a Matlab-based GPS signal generation capability. If our Matlab GPS Toolbox is provided, the Signal Architect GPS simulation can be run under the Matlab environment. For the non-Matlab user, the Signal Architect software is bundled as a stand-alone executable. The signal simulation flow is depicted in **FIGURE 3**.

GUI. Using a simple, intuitive graphical user interface (GUI), the user specifies a trajectory and a complete set of simulation parameters to create an I/Q data file (see **FIGURE 4**). The user specifies a trajectory either from an NMEA file (most GPS receivers use the National Marine Electronics Association 0183 interface standard for logging positions and other data) or a KML file from Google Earth (Google's Keyhole Markup Language has become a standard for describing geographically referenced features), and an almanac file used to define GPS satellites to be simulated. The user defines the mask angle for the satellite selection and the noise figure to be simulated. The Signal Architect software then generates a simulated digital storage file, including the selected pseudorandom noise codes (C/A and/or the unencrypted military P or M′ codes).

Scenarios. The Signal Architect software also ships with preloaded scenario files that the user can run right out of the box into the SCU using the USRP. The Signal Architect software supports computers with multi-core processors and will automatically configure itself to run on all available processors. The Signal Architect software will generate either static or dynamic simulation profiles. The Signal Architect GUI allows the operator to easily modify a wide range of scenario variables from the pre-set defaults. Complete scenarios are easily shared between signal simulation systems, supporting collaborative testing
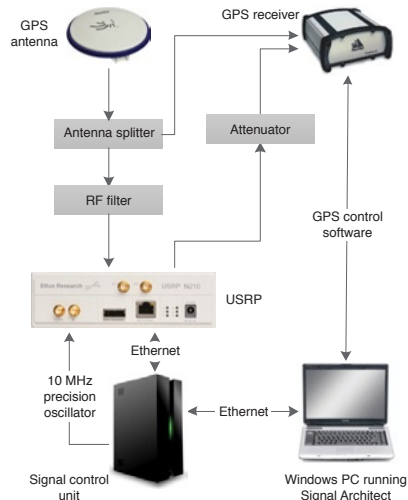
▲ **FIGURE 5** GPS Signal Simulator record and playback GPS simulation.

▲ **FIGURE 6** Simulated satellite constellation.
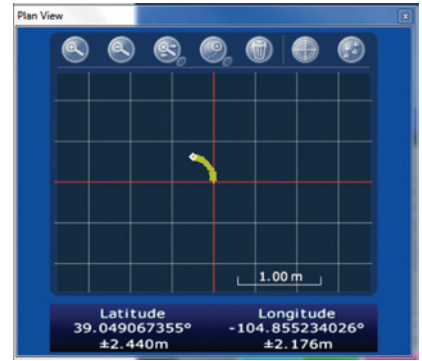
▲ **FIGURE 7** North-east (2-D) position error.

between similar projects and reducing the amount of time spent developing test tools.

The signal data file can then be used for subsequent analysis within Matlab using the Matlab GPS Toolbox, or can be provided to the SCU and USRP to create a GPS signal suitable for playback into a GPS receiver. If the Matlab GPS Toolbox is available, the user has complete flexibility to manipulate the signal at various stages of generation or post-generation to simulate GPS anomalies. Without the toolbox, the user is restricted to using only the standard error modeling provided by the compiled Signal Architect code.

Simulation Test Results

To demonstrate the high fidelity of our Signal Architect signal record and playback capability, a series of stationary GPS simulations were run. In these tests, the USRP was used to record and play back GPS C/A-code signals at the L1 band (1575.42 MHz). The SCU and USRP were connected to a rooftop-mounted GPS L1 antenna. The GPS signal was split between a commercial GPS receiver and the USRP to allow the operator to monitor the GPS receiver while the USRP was recording the GPS signal.

In record mode, the I/Q data is written from the USRP to a file on the SCU. In playback mode, the data is read from the file by the USRP to generate the RF signal. The RF signals are output to the GPS receiver through an external variable attenuator. The attenuator allows the operator to adjust the signal power into the GPS receiver as different lengths of antenna cable are added or as the signal is split to other GPS receivers.

To demonstrate the GPS Signal Architect Test Set performance, representative data was collected in a series of two laboratory tests. The first test demonstrates the system performance as a record and playback GPS signal simulator. The second test results demonstrate the system performance when using the Signal Architect software to generate custom GPS scenario files for playback into the GPS receiver.

In the first test, the GPS simulator hardware was configured as shown in **FIGURE 5**. The GPS receiver and USRP were connected to a commercially available antenna. The antenna was positioned at a known location with a clear view of the GPS constellation. The signal from the GPS antenna was split between the GPS receiver and the USRP so that the data could be logged by the receiver software at the same time as it was being recorded by the SCU.

The simulated satellite constellation is shown in **FIGURE 6**. Seven GPS satellites are in view.

The 2-D position error from the simulated signal is shown in **FIGURE 7**. The errors are representative of the accuracies achievable using GPS C/A-code pseudoranges.

We can examine the performance of our test set by looking at plots of the measurements of carrier-to-noise-density ratio (C/N_0) from the GPS receiver for both the live sky data and for the recorded signal when played into the GPS receiver by the Signal Simulator for three of the GPS receiver channels (see **FIGURE 8**). The C/N_0 data collected from the GPS antenna is shown in blue, while the C/N_0 from the USRP is shown in green.

As we can see, the C/N_0 was 1–2 dB lower in playback mode when compared to the data collected from the GPS antenna. The signal loss is due to the 1-bit sampling of the incoming GPS signal by the Signal Architect software. One-bit and 2-bit quantization are used in many commercial GPS receivers. The rule of thumb states that 1-bit quantization degrades the signal-to-noise ratio by 1.96 dB, and 2-bit quantization degrades the signal-to-noise by 0.55 dB. These results show that 1-bit I/Q sampling is sufficient for reproducing GPS L1 C/A-code signals with the USRP.

In the second test, the Signal Architect software was used to generate a 10-minute static GPS C/A-code L1 scenario file. The SCU used the USRP to generate the GPS signal.

Shown in **FIGURE 9** are the number of satellites the GPS receiver was able to track. When using the GPS Signal Architect Test Set to play back the scenario file, the GPS receiver was able to track all the simulated satellites in the file. The time necessary for the GPS receiver to acquire and track the satellites is consistent with the performance one would expect from

the GPS receiver when connected to an external antenna.

FIGURE 10 shows the C/N_0 measurements from the GPS receiver for three of the receiver channels. There were nine satellites in this static scenario file. The C/N_0 for all the satellites is stable for the duration of the scenario playback.

Another Hardware/Software Option

We have also worked with the manufacturer of the LabSat hardware signal simulator to include some of the software functionality of our USRP system.

The LabSat GNSS simulator (see **FIGURE 11**) can be used to record live navigation satellite RF data streamed onto a hard drive. This can then be played back as an RF signal. When integrated with the SatGen software, simulated digitized RF data can be generated and played back into the LabSat simulator in place of the recorded, digitized GNSS RF signals.

The core of the SatGen software is our Signal Architect software component, which has been adapted to run on the LabSat platform to allow simulation of the multiple GNSS satellite signals.

Hardware. The latest version of the LabSat hardware design (LabSat 2) enables the record and playback of GPS and GLONASS synchronized RF data streams (see **FIGURE 12**). When recording the GPS or GLONASS signals, the RF L1 channels (1575.42 MHz for GPS and 1602.0 MHz for GLONASS) are down-converted to a 0-Hz intermediate frequency. The signals are sampled (2 bit I and Q) at 16.368 MHz to capture the full GLONASS set of frequency channels. The GPS and GLONASS data is then interleaved into a 4-bit data stream and recorded in an internal buffer as a binary file. A high-speed USB link then transmits the data to a PC before streaming onto the PC hard disk. For playback, the PC streams
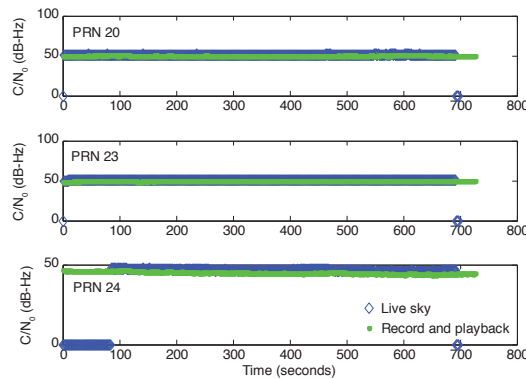
▲ **FIGURE 8** C/N_0 record and playback vs. live sky collection.
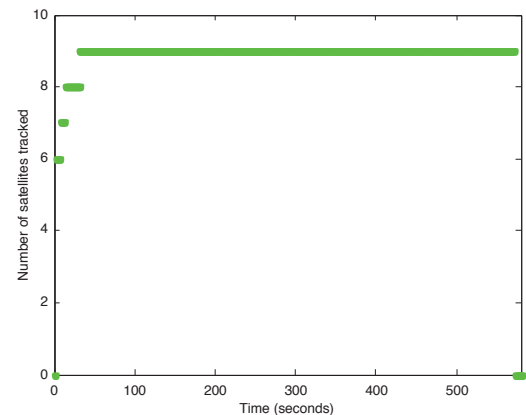
▲ **FIGURE 9** Number of satellites tracked (digital signal file playback mode).
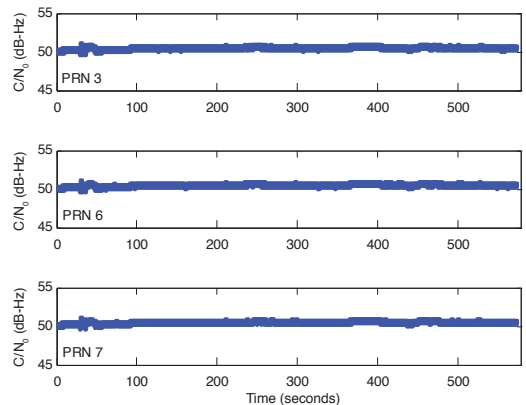
▲ **FIGURE 10** C/N_0 (digital signal file playback mode).

the stored binary data to the LabSat via the USB link. The recorded digital GPS and GLONASS signals are up-converted onto the GPS and GLONASS RF channels and played back into the receiver under test. A digital attenuator on the output can adjust the RF output level.

Using the LabSat record and playback mode, all of the real-world effects on the GNSS signals are recorded, including multipath effects, drop-outs, and atmospheric effects,

allowing repeatable tests to be performed on GNSS receivers under a variety of real-world conditions, such as operating in urban canyons. This is ideal for debugging fault conditions on GNSS equipment and software. For more extensive simulations in different environments, the LabSat SatGen software can be used to generate simulated scenarios at any time or place or for a dynamic environment.

SatGen Scenario Simulation. SatGen is a software package that allows users to define trajectories for use in generating simulated data files for playback into LabSat. A user-defined trajectory file can be used to create a LabSat-simulated scenario for a route anywhere in the world. Routes can be generated directly from NMEA files imported directly into SatGen from a GPS datalogger or from user-defined routes generated using Google Earth.

SatGen users can use Google Earth to define a route by creating a path using its "Add Path" tool. The user can use as many or as few waypoints as the user wants, and can edit routes by moving, adding, or removing waypoints. The path is saved as a standard Google Earth KML file, which is imported into SatGen, which then fills in and smooths the trajectory between the waypoints. The user can also define velocity profiles, or SatGen can provide these automatically. SatGen creates an NMEA file that is used to generate a binary I/Q simulated signal file for replay on the LabSat hardware.

Signal Architect Simulation. The core of the SatGen software is GNSS Signal Architect, an upgraded version of our GPS Signal Architect, which provides the capability to simulate multiple GPS signals and also different GNSS signals.

Signal Architect imports the NMEA trajectory either from a prerecorded file or from one generated using SatGen,

and uses this file to generate a GNSS-simulated scenario. The user specifies the input GPS satellite constellation through a Yuma-format almanac file and the GLONASS constellation through a GLONASS almanac file in ".agl" or RINEX format. These files are then used to generate the simulated pseudorange, Doppler, and carrier phase for the GPS and GLONASS satellites in view of the simulated GNSS receivers above a specified mask angle. This simulated range data is then used to generate the digitized I/Q signals for the GPS and GLONASS satellites. Users who have access to our GNSS Signal Simulation Toolbox (an upgrade of our GPS toolbox) will have the added ability to modify the GNSS signal strength and add additional high-resolution error models to the simulated signals including multipath or GNSS signal error characteristics. The resulting I/Q simulated data file for the GPS plus GLONASS constellation is then recorded in a data file, which can be loaded into the LabSat hardware for playback into a receiver under test.

Test results using the LabSat and SatGen combination have demonstrated that highly accurate navigation solutions can be obtained with a variety of playback modes.

Conclusion

The combination of our GPS Signal Architect software with either the SCU and USRP or LabSat has proven to be an ideal low-cost GPS signal simulation tool with the capability of simulating or recording the complete GPS signal spectrum, including both the civil and the military codes for playback. The initial release of the GPS Signal Architect Test Set supports L1 operation and C/A- and P-code and M′ signal simulation or C/A- and P(Y)-code and M′ record and playback,

▲ **FIGURE 11** LabSat GNSS simulator.
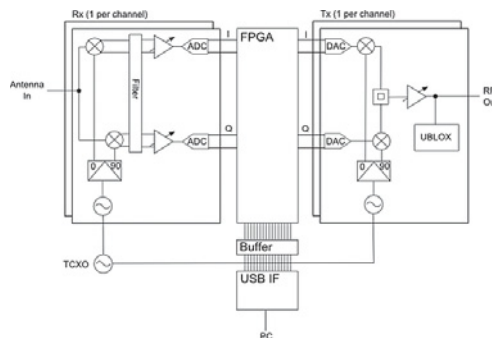
▲ **FIGURE 12** LabSat GNSS simulator hardware design.

while both GPS and GLONASS signal generation and playback is available with LabSat.

Our team of GPS and RF experts is continually developing and updating the system to provide additional functionality. Future releases of our test set will include support for multi-frequency SDR hardware and the capability to simulate other civil and military GPS signals, and also those of other global navigation satellite systems. To reflect this capability, we have branded the latest version of our simulation system, the GNSS Signal Architect Test Set.

Acknowledgments

The authors acknowledge the support of Ettus Research LLC in the development of the technology associated with the USRP system, as well as Racelogic Ltd. for collaboration on the LabSat GNSS simulator. USRP is a registered trademark of National Instruments Corp.

The article is based primarily on the papers "GPS Signal Simulation Using Open Source GPS Receiver Platform" presented at the Virginia Tech Symposium on Wireless Personal Communication in June 2011 and "SatGen GNSS Signal Simulation Software" presented at ION GNSS 2011 in Portland, Oregon, in September 2011.

Manufacturers

The GNSS Signal Architect Test Set was developed by **Navsys Corp.** (*www.navsys.com*). The USRP used for the test set is the **Ettus Research LLC** (*www.ettus.com*) model USRP N210. The LabSat 2 GNSS Simulator and associated SatGen software is produced by **Racelogic Ltd.** (*www.racelogic.co.uk*). The GPS equipment used in our tests was a **Novatel** (*www.novatel.com*) DL-4 plus receiver and a GPS-702GG antenna.

ALISON BROWN is the president and chief executive officer of Navsys Corp., Colorado Springs, Colorado, which she founded in 1986. Brown has a Ph.D. in mechanics, aerospace, and nuclear engineering from UCLA, an M.S. in aeronautics and astronautics from MIT, and an M.A. and B.A. in engineering from Cambridge University. She is a fellow of the Institute of Navigation and an honorary fellow of Sidney Sussex College, Cambridge.

JARRETT REDD is a senior systems engineer with Navsys Corp. working on hardware, firmware, and embedded systems development for signal acquisition, processing, and transmission. He holds an M.S. and B.S. in computer engineering from Texas A&M University.

MARK-ANTHONY HUTTON is a software engineer with Navsys Corp. working on GNSS signal simulation tools and the GPS Jammer Detection and Location System. He holds a B.S. in computer science from the University of Colorado at Colorado Springs.

MORE ONLINE

Further Reading

For references related to this article, go to *gpsworld.com* **and click on** *Richard Langley's Innovation* **in the left-hand navigation bar.**