Q Are there low-cost and low-weight options for GNSS IF storage?

GNSS Solutions is a regular column featuring questions and answers about technical aspects of GNSS. Readers are invited to send their questions to the columnist, **Dr. Mark Petovello**, Department of Geomatics Engineering, University of Calgary, who will find experts to answer them. His e-mail address can be found with his biography below.

MARK PETOVELLO is a Professor in the Department of Geomatics Engineering at the University of Calgary. He has been actively involved in many aspects of positioning and navigation since 1997 including GNSS algorithm development, inertial navigation, sensor integration, and software development. Email: mark.petovello@ucalgary.ca

Over the past decade an increased interest has arisen among researchers and industry for the use of GNSS intermediate frequency (IF) data for post-processing analyses. Intermediate frequency data is the most basic type of GNSS data that can be recorded since it is completely unprocessed and thus allows for many processing options. This contrasts with classical GNSS observables such as pseudoranges and carrier phase measurements that are derived using a specific algorithm to process the IF data. In other words, classical GNSS observables are effectively "fixed" and are thus limited in terms of what you can do with them. The downside to IF data is the data rates and data storage requirements are much higher.

As mentioned, the inherent benefits of IF data is that it allows researchers and users to have control of how the data is processed and, as a result, explore different algorithms and/or settings. In turn, this can have distinct advantages for a wide range of applications such as, for example, multipath mitigation, deep/ultra-tight sensor-fusion, space weather monitoring, reflectometry and signal analysis in general.

The disadvantage of using IF data in post-mission processing is that it requires a much larger storage capacity than classical observables, and data is normally not stored nor output from a typical commercial GNSS receiver due to the high data rates and storage requirements. Commercial options are available for IF recording, but available products typically require a PC for storage, and the entire system thus becomes large, heavy and expensive.

The main challenge for creating a downsized, low-cost solution is the need to support the high data rates and capacity requirements associated with IF data.

In this article, we briefly discuss the fundamentals of IF data acquisition and present a prototype of a simple low-cost and low-weight dual-constellation GNSS IF recorder built from readily-available components. The system was designed specifically for mini–unmanned aerial vehicle (UAV) applications.

IF Sampling

A GNSS receiver can roughly be divided into four main components; antenna, radio frequency (RF) front-end, baseband processor, and navigation processor.

Figure 1 presents a high-level block diagram of a generic GNSS receiver. The antenna receives the electromagnetic waves originating from the satellites. The RF front-end performs amplification, filtering, and down-mixing of the received signal to IF before discretizing it using an analog-to-digital converter (ADC).

The baseband processor first performs signal acquisition and then transitions to signal tracking. Doppler and code-phase measurements from the tracking loops are then used to compute user position, velocity and time (PVT) in the navigation processor.

The IF samples can either be processed in real-time or stored for post-processing. The latter is of particular interest for software receiver testing and development and for applications that may be sensitive to size, weight, and/or power where real-time process-

ing may be infeasible. **Figure 2** shows the workflow for IF data collection and post-mission processing. Real time-processing is effectively the same but since the IF data is processed immediately, the challenges associated with data storage are avoided.

The figure clearly shows that the IF recording system consists of an RF front-end and some kind of storage device. To better explain the nature of IF samples and how they are generated, we turn the attention to the RF front-end. **Figure 3** shows a block-diagram of the principal components forming a typical GNSS RF front-end.

The first stage in the RF front-end is a low-noise amplifier (LNA), which provides initial amplification to the RF signal received from the antenna. The second stage is an in-phase/quadra-phase mixer (I/Q mixer) that is responsible for translating the original RF spectrum down in frequency.

In the mixer, the RF signal is multiplied by a sinusoidal signal generated from a local oscillator (LO) which has a fixed frequency and amplitude. This operation causes the GNSS carrier frequency to be translated to a lower IF frequency. The characteristic equation for this operation can be defined as:

$$f_{IF} = f_{RF} - f_{LO} \qquad \textbf{(1)}$$

where f is frequency and the subscripts are as defined earlier. The equation is actually a little simplified, as the mixer actually produces two translated spectrums originating from both the sum and the difference of the RF and LO frequencies. The spectrum originating from the sum of the signals is of no interest and is easily removed by filtering.

After mixing, the IF signal is either bandpass or low-pass filtered to remove out-of-band signals and to prevent aliasing effects in the ADC. A typical GNSS RF front-end is furthermore equipped with an automatic gain control (AGC) circuit that evaluates the statistical distribution of the IF samples and adjusts the signal amplification for optimum saturation of the ADC.

The output of the RF front-end is either a real or complex (I/Q) signal with sample rates typically ranging between 5–60 MSamples/s (Msps) depending of the frequency bands (i.e., GNSS signals) that are being acquired and tracked. Depending on the number of bits used in the ADC (1 to 16 bits depending on the application) and whether real or complex sampling is used, this translates to data rates (with full data compression) ranging from about 0.6 Mbytes/second (MB/s) to 120 MB/s.

To handle such high data rates, commercial front-ends typically implement a high-speed data link using either PCI, USB, or Ethernet interfaces, which then provide a continuous stream of IF data to be collected by a PC for either IF storage or real-time processing. However, the high sample rates pose a challenge when trying to scale the system down to smaller and more affordable hardware components. Fortunately, low-cost embedded computing platforms are available that can
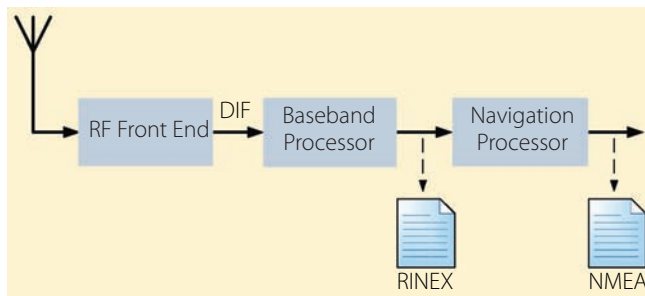

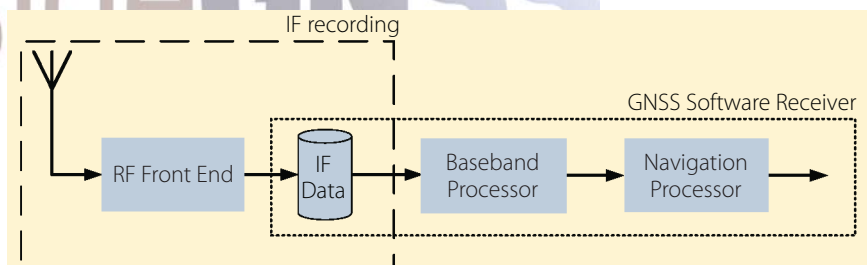**FIGURE 1** Generic GNSS receiver structure

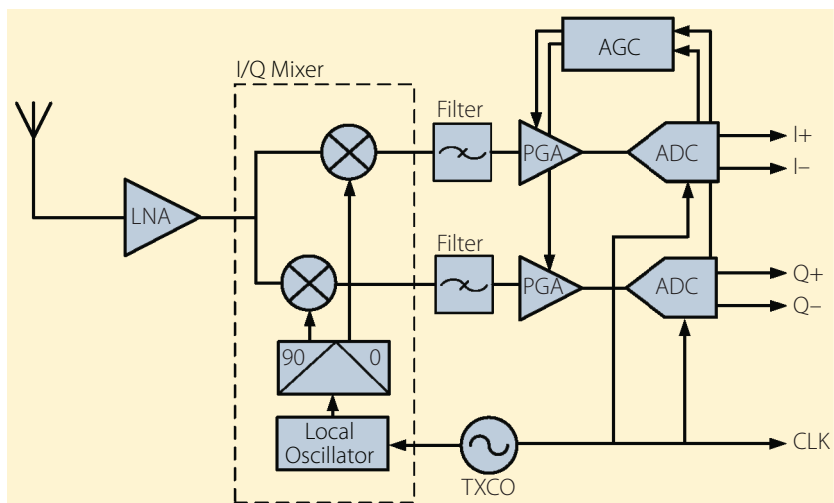**FIGURE 2** IF storage and GNSS software receiver

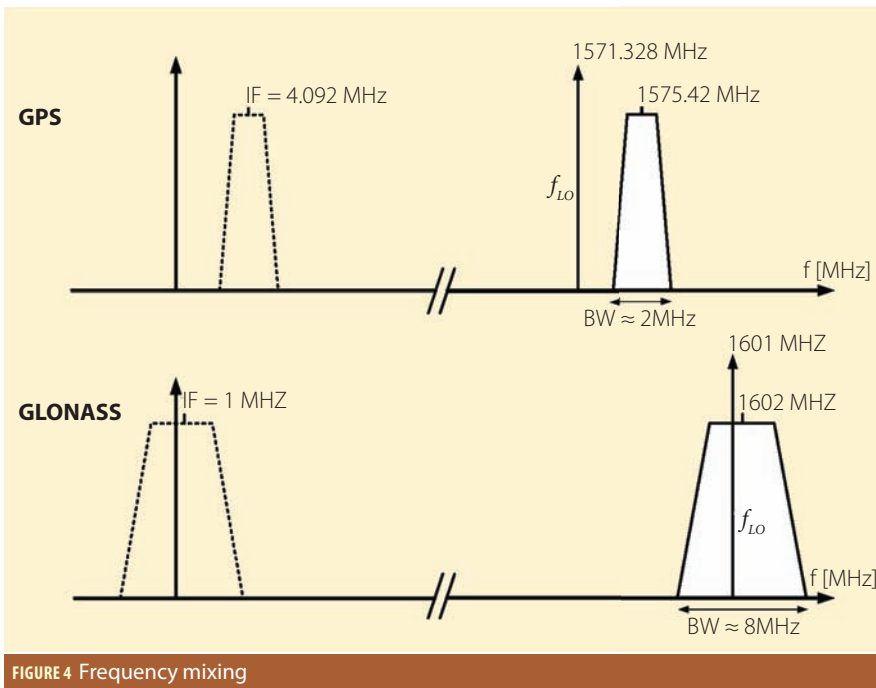**FIGURE 3** Block diagram of GNSS RF front-end

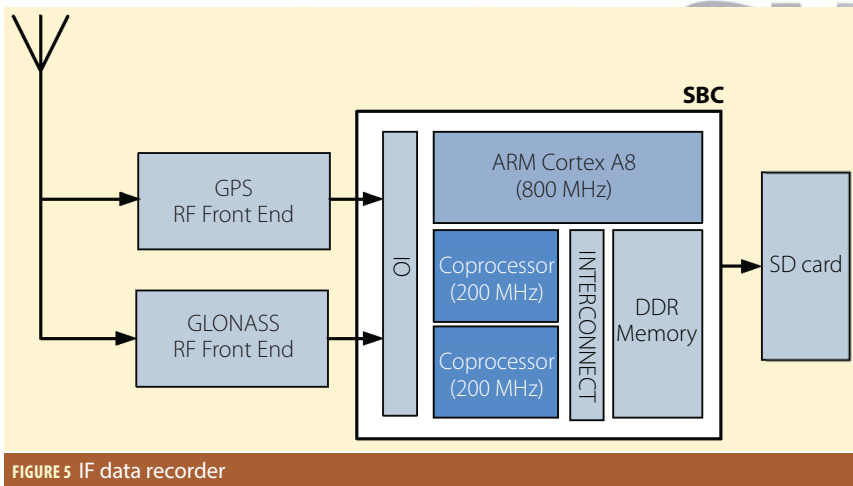**FIGURE 4** Frequency mixing

FIGURE 5 IF data recorder

be used for this task, as discussed in the next section.

A Single-Frequency, Dual-Constellation GNSS Sampler

This section describes a low-cost and low-weight GNSS sampler intended for experiments on mini-UAVs. The development was motivated by the absence of a commercially available IF recorder that met our needs, especially in terms of size and weight. The design is based on readily-available components consisting of a commercial ARM-based single-board computer (SBC) that

can be purchased for around US$55 and an evaluation kit of a GNSS RF front-end capable of receiving GPS, Galileo, BeiDou and GLONASS signals in the L1 frequency band. The system incorporates two RF front-ends; the first is configured for reception of GPS L1/Galileo E1 and the second for GLONASS L1.

In this configuration, the front-ends are capable of providing a dynamic resolution of two bits for I and Q samples, respectively. Further, the ADC-sample clock is also provided as an output for synchronization pur-

poses. The ADCs for both RF front-ends are operated with a sample-rate of 16.368 Msps.

GPS satellites transmit with a L1 carrier frequency of 1575.42 MHz. To capture the GPS C/A code, we have used a local oscillator frequency of 1571.328 MHz. According to equation (1), this gives an IF frequency of 4.092 MHz.

GLONASS employs frequency-division multiple access (FDMA) meaning satellites are using different carrier frequencies for transmission. A total of 14 frequency channels are used with channel numbers ranging from $k = -7, ... , +6$. The carrier-frequencies for each channel are derived from equation (2).

$$f_c = 1602\ MHz + k \cdot 562.5\ kHz \quad (2)$$

In order to use all available GLONASS satellites in a PVT solution, the front-end should capture all the channels. Effectively, this means that GLONASS satellites transmitting in the range 1597.5515 MHz – 1605.8865 MHz should be received simultaneously.

For the GLONASS front-end, a local oscillator frequency of 1601 MHz was used. After mixing, the GLONASS L1 C/A code spectrum is transformed down to -3.4485 MHz – 4.886 MHz. To visualize the mixing process, **Figure 4** shows a frequency domain representation of the mixing process for both the GPS and GLONASS signals. The dotted lines indicate the IF spectrum (i.e., after mixing).

The RF front-ends are connected using a parallel interface to the SBC. For each front-end, the ADC sample clock and the ADC I/Q outputs are connected to input/output (IO) ports on the SBC. For the GPS front-end, only the in-phase (I) outputs have been used — the quadrature (Q) outputs can be omitted as the down-converted spectrum does not overlap 0 Hz (see Figure 4).

For the GLONASS front-end, the IF spectrum has both positive and negative frequencies, and, in this case, complex sampling is necessary to avoid aliasing. **Figure 5** shows a high level

block diagram of the complete system.

The chosen SBC is equipped with a processor using an 800 MHz ARM Cortex A8 CPU and two coprocessors operating at clock frequencies of 200 MHz. For this setup, the two coprocessors have been utilized to read the IO ports connected to the front-ends and transfer the IF samples to two separate circular buffers in the double data rate (DDR) memory on the SBC. Whenever a pre-defined number of bytes have been written to the DDR memory, the coprocessors raise an interrupt that initiates the ARM processor to write the buffer contents to a binary file on the secure digital (SD) card.

One of the advantages of using the coprocessors to read the parallel outputs from the two ADCs is that the ARM processor effectively is off-loaded most of the time. This means that other tasks can be executed while the data collection takes place.

We have currently not taken advantage of this, but an obvious task would be to implement some real-time quality control measures on the IF data. This could, for instance, be to periodically run an acquisition algorithm to determine the satellites in view.

The SBC is shipped with a debian-based (linux) operating system, and the application code is written in C++. The source code for the two coprocessors runs independently from the operating system and is written in assembly.

In terms of throughput, the main bottleneck is the transfer of data from DDR memory to the SD card. The DDR memory has a low latency and operates at high bus speeds, but the software driver and hardware controller for the SD card can only support speeds up to class 10 (10 MB/s). This limitation can however be circumvented by using an external card reader via USB. In this case an average write

speed of approximately 20 MB/s can be achieved.

Another limitation is in terms of sample rate from the front-ends. We have tested sample rates up to approximately 20 Msps, but beyond this limit the coprocessors would start to be overloaded each time an interrupt to the ARM processor is generated. This originates from branching operations, which would cause the coprocessor to skip samples during the transitions.

Results

An accompanying photo shows the components used in developing the IF sampler and storage system. The unit on the ruler is in units of centimeters. To the left is the SBC used for storing the samples to a removable SD card. To the right is one of the front-end evaluation kits. At the bottom is a second revision of the front-end that has been created from a reference design sup-
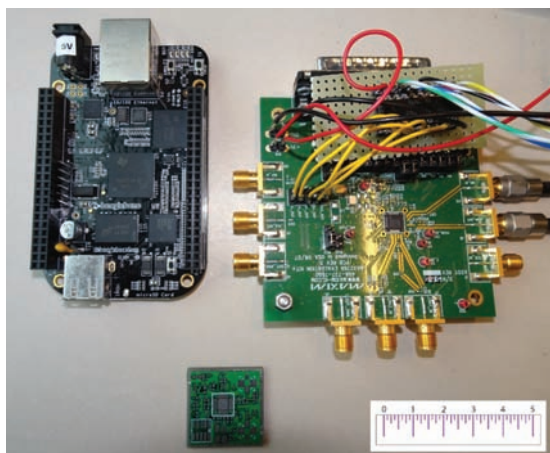
Components for the IF data recorder

plied by the manufacturer (which will ultimately replace the evaluation unit).

To validate that the system was working we performed a static test with a roof antenna at DTU Space Institute Building, Kgs. Lyngby, Denmark. This test was conducted on July 3, 2016, at 16:24 (UTC+1). The recorded IF data was processed by an open-source MATLAB-based software receiver modified to acquire and track GLONASS signals. Results of the satellite acquisition are shown in **Figure 6**.

From the data, we can identify 6 GPS satellites and 4 GLONASS satellites using standard-sensitivity acquisition algorithms. Tracking results from GPS SV26 and GLONASS frequency slot -3 is shown in **Figure 7**. The figure shows the power in the early, late, and prompt correlators (bottom graph) and the output of the in-phase prompt correlator (top).

As a practical note, the GPS front-end has a data-transfer rate of 4 MB/s and the GLONASS front-end has a transfer-rate of 8 MB/s. This means that a mission of approximately 45 minutes can be conducted using a 32-gigabyte SD card. If more storage is needed, the SBC can also be connected to an external hard drive through a USB port.

The SBC in our setup has proven capable of handling two front-ends with sample rates up to 20 Msps each.

If higher sample rates are needed than what this system can provide, we would suggest looking into SBCs based on a hybrid CPU/programmable logic architecture.

These devices have recently gained momentum and are becoming more affordable. A solution based on this type of architecture would, however, require a higher level of hardware expertise and the complexity and cost would normally be higher, too.

Conclusion and Outlook

We have demonstrated a way to design an IF recording system using ready-made equipment where only wired connections between the SBC and RF front-end evaluation kits were necessary. The IF recording system has only been tested with GPS and GLONASS L1 C/A codes but should also support Galileo E1 OS signals, thus making it a triple-constellation system.

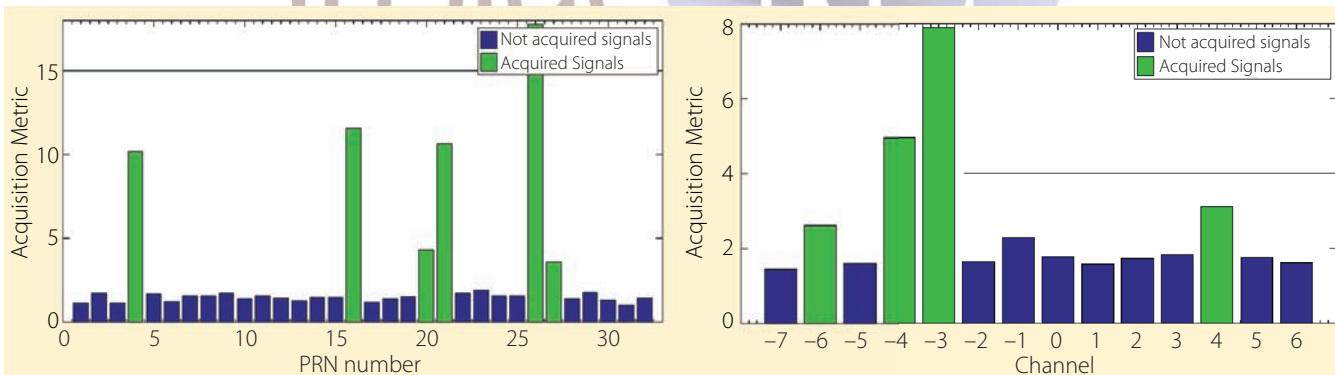The general challenge in using low-cost embedded computers for IF
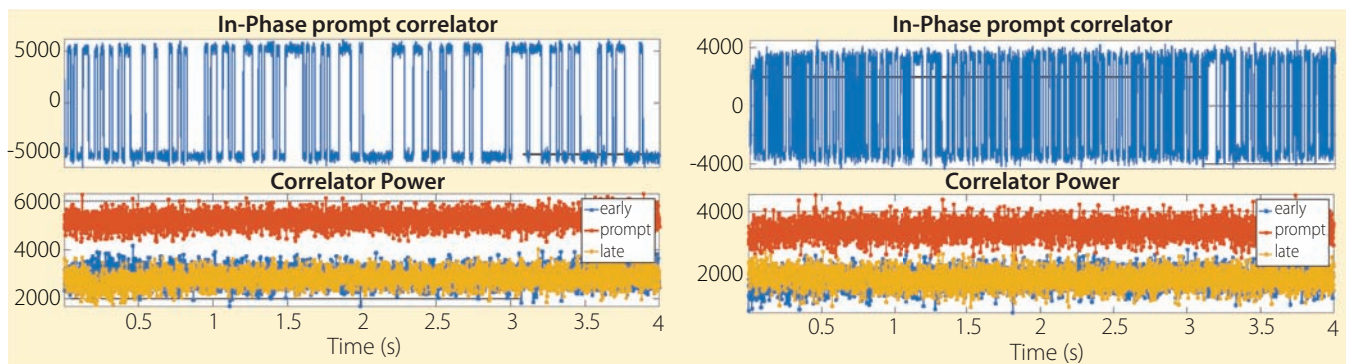

FIGURE 6 Components for the IF data recorder

FIGURE 7 Tracking results. GPS SV26 (left) and GLONASS satellite k= -3 (right)

recording is the performance limitations compared to desk-top computers.

The authors are investigating the possibility of releasing the bill of materials, assembly instructions, and source code in an open source forum for the benefit of the greater GNSS community.

Manufacturers

The system presented here was built using the following components. The two GNSS RF front-ends were based on MAX2769 evaluation eits from **Maxim Integrated**, San Jose, California USA. The SBC is the BeagleBone Black, rev. C from **BeagleBoard.org Foundation**, Oakland Twp, Michigan USA. The software receiver used for processing is derived from the version provided in K. Borre *et alia* (2007) *A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach,* Springer Science & Business Media.

Additional Resources

For more detailed information about the IF recorder described in this article, please refer to:
Olesen, D., and J. Jakobsen, and P. Knudsen, "Low-cost GNSS Sampler Based on the BeagleBone Black SBC," *Proceedings of the 8th ESA Workshop on Satellite Navigation Technologies (Navitec 2016),* December, Noordwijk, Netherlands, December 14–16, 2016 (in press)

For a description of the benefits of IF data storage and processing, refer to:
Lachapelle, G., and A. Broumandan, "Benefits of GNSS IF Data Recording," *Proceedings of the European Navigation Conference (ENC2016),* Helsinki, Finland, May 30 – June 2, 2016

Authors

DANIEL OLESEN obtained his MSc. degree in electronics engineering with specialties in control theory and robot technology from the Technical University of Denmark (DTU). He is currently a Ph.D. candidate at DTU Space, Department of Geodesy. His research interests are navigation for UAVs, GNSS receiver design, GNSS signal processing, and sensor fusion.

JAKOB JAKOBSEN is a researcher at DTU Space, Department of Geodesy. He has worked with research, development and teaching related to GNSS for more than 20 years. His main interests are precise positioning and navigation for small UAVs.

PER KNUDSEN is a professor at DTU Space, and the head of the Department of Geodesy. He is responsible for R&D activities related to GNSS at DTU Space as well as for the operation of permanent GPS reference stations in Greenland. **IG**