SDR for Space, the open way

Manolis Surligas[1,2], Kostis Triantafillakis[1,2], and George Vardakis[1,2]

[1]Libre Space Foundation, Athens, Greece

[2]University of Crete, Computer Science Department, Heraklion, Greece

Email: {surligas, ctriant, vardakis}@csd.uoc.gr

***Abstract*—In this paper we present SatNOGS, a network of ground stations for observing Low Earth Orbit Satellites, and focus our attention to the SDR RF frontend and the GNU Radio module operating it.**

I. Introduction

Satellite Networked Open Ground Station (SatNOGS) [1] is a network of satellite ground stations focused on observing and receiving satellite signals, particularly low earth orbit (LEO) cubesats. The deployed ground stations create a global network that can automatically schedule and accomplish a target satellite observation. The aforementioned functionality is achieved through a cluster of well-defined, completely open-source applications. SatNOGS Database keeps track of all publicly available cubesat-related information including RF communication frequencies, NORAD IDs and more. This information is utilized by SatNOGS Network, which gives online users the ability to schedule satellite observations using the extended network of SatNOGS ground stations around the world. A typical SatNOGS ground station is comprised of three distinctive components: an agile rotator able to follow the requested satellite trajectory, the RF frontend which is an SDR device programmable through GNU Radio, and the SatNOGS Client. The latter is a web application capable of dictating the operation of the rotator as well as the GNU Radio module of the RF.

II. The gr-satnogs GNU Radio OOT module

Each SatNOGS ground station is equipped with an SDR device and operates on a Linux platform. The signal processing that is required for filtering, demodulation e.t.c is accomplished through our GNU Radio Out-Of-Tree (OOT) module called *gr-satnogs*. The source code is available at https://github.com/satnogs/gr-satnogs published under the GPLv3 license.

The *gr-satnogs* module is responsible for handling the incoming signal from the SDR device. It supports a large variety of hardware devices through the *gr-osmosdr* [2] module.

A. Doppler Effect Correction

LEO communications are severely affected from the Doppler effect. The *gr-satnogs* module incorporates a mechanism to correct the Doppler phenomenon in order to increase the accuracy of the demodulation algorithms.

Legacy hardware radios and existing ground stations use *rigctld* daemon of *hamlib* [3] to change their RF frequency. However, this produces glitches and discontinuities at the received signal. This effect is even worse for most of the SDR devices, where the RF frequency changes require a significant amount of time.

For this reason, *gr-satnogs* compensates for the Doppler effect digitally by multiplying the incoming signal with a Numerically Controlled Oscillator (NCO). This technique requires slightly more bandwidth but this is not a problem for modern SDR devices. During the observation *satnogs-client* constantly computes the trajectory of the target satellite and informs every second the GNU Radio block responsible for altering the frequency of the NCO.

For even better performance all the estimated Doppler shift messages are inserted into a 3rd-order polynomial curve fitting system. This curve fitting mechanism estimates the Doppler shift during two consecutive updates from the *satnogs-client* trajectory calculator. Thus, the frequency correction mechanism corrects in a more fine grained way the Doppler shift.

B. Filtering

Filtering is another critical part of the signal processing that the *gr-satnogs* module performs. Due to the variety of the SDR hardware, our module includes a hierarchical GNU Radio block that implements different filtering schemes depending on the SDR device used.

In general, LEO channel bandwidths are a few kHz wide (2 - 100 kHZ). However, most of the SDR devices out there operate on bandwidths that are a few MHz wide. SDR devices like USRPs from Ettus Research can perform decimation and filtering on the onboard FPGA and can filter-out the excess bandwidth to a resulting minimum of 500 kHZ. Cheaper devices however, do not have the hardware capabilities to perform any decimation at all.

This is a serious problem to overcome, because higher bandwidths come along with increased computational requirements. In embedded devices like the Raspberry, filtering out the excess bandwidth may be prohibited without special design considerations. The *gr-satnogs* filtering hierarchical block accepts as parameter the device string and the sampling rate that is configured and tries to decimate and low-pass filter the input signal. If the filter taps exceed a certain threshold, the decimation and low-pass filtering is performed in two steps to exploit parallelism and pipelining.

Taking advantage of the excess bandwidth, the RF frequency of the SDR device is configured with an offset of few kHZ, in order to avoid DC spikes. Using the XLating FIR filter of GNU Radio as the first step filter, we perform low-pass filtering and decimation while avoiding DC issues at the same time.

C. Signal Demodulation

Currently, *gr-satnogs* incorporates a set of generic signal demodulation flowgraphs. We already ship flowgraphs for CW demodulation and generic FSK modems. However, most of them can not extract specific payloads rather than bit streams. Cubesats and other LEO satellites do not follow any specific framing format neither a modulation scheme. AX.25 is a very common framing format among the amateur satellites. Our module implements the AX.25, but unfortunately the synchronization characteristics of this framing scheme are so poor that most of the satellites out there use it for legacy reasons and encapsulated it inside a custom more robust frame format.

Retrieving valid payloads tends to be extremely mission specific. Most Cubesat missions do not provide the framing and synchronization details and makes the payload retrieval a very difficult and time consuming process. Our goal is to provide all the necessary common stuff that are needed for satellite signals reception and simplify the construction of a mission specific transceivers. With enough support from the community and satellite operators *gr-satnogs* satellite database can grow up significantly. Our vision is to encourage Cubesat operators to build their telecommunication subsystem around an open and robust standard framing format. In such cases, development and testing times can be reduced drastically with the help of the *gr-satnogs* module. In the following section we describe such a use case.

III. SATELLITES

Although *gr-satnogs* is a relatively new module and technology, however it already supports a fully functional ground station transceiver for a Cubesat satellite, called UPSat [4].

UPSat is the first Greek open-software, open-hardware Cubesat satellite constructed by University of Patras & Libre Space Foundation. It is part of the QB50 [5] project that will deploy a constellation consisting from 50 cubesats and it is expected to be launched from the ISS at late February.

IV. ON-GOING WORK

A. MIMO beam steering

Multiple Input Multiple Output (MIMO) beamforming is a very promising technique for ground stations receiving communications signals from LEO satellites. The current SatNOGS ground station employs a single antenna for each frequency band of interest, mounted on a rotator which follows the path of a LEO satellite when visible. An alternative to this setup is a MIMO ground station employing multiple elements for each band of interest which can be both low-cost and easier to install than the rotator.

For our implementation we used four custom-made Quad-Helix Antennas(QHA) arranged in square formation with $\frac{5\lambda}{2}$ spacing. The QHA is a favorable antenna for satellite reception because of its very wide beampattern in the broadside. Another appealing characteristic of the QHA is the ease with which one can construct it with the cost remaining at very low levels. A typical QHA can provide up to 5dB gain with the total gain of the array reaching 20dB as a result of beamforming.

The process of beamforming requires a two-step phase manipulation: The device phase correction and the MIMO phase difference calculation. For our implementation two USRPs B210 were used, one for each pair of elements. The challenge presented by the use of two sdr devices is the phase difference of the samples at the output of the devices with reference to their true time of arrival on the elements. Our proposal towards the solution of the lack of synchronization between the devices includes a coarse and a fine phase correction. An element at the center of the square array transmits a known training sequence which is received by all four antennas. The samples produced by the two devices are properly delayed with respect to each other for the coarse phase detection to take place. Then a fine phase correction is achieved by digitally shifting the phase of the produced signals, based on the known training sequence which was transmitted.

After synchronization of the devices is ensured, the MIMO beamforming follows. We assume that the signal arriving at the array is a plane wave. It is mathematically represented as $x(t), t = 1, 2, ..., K$ where K is the number of samples produced in each execution, which happens at intervals of 10 ms. The output of the array is expressed as

$$X(t) = Ax(t) + n(t) \qquad t = 1, \cdots, K \tag{1}$$

where n is zero mean Additive White Gaussian Noise, and A is the steering vector defined as

$$A = \begin{bmatrix} e^{j\frac{2\pi}{\lambda}u_1} & e^{j\frac{2\pi}{\lambda}u_2} & e^{j\frac{2\pi}{\lambda}u_3} & e^{j\frac{2\pi}{\lambda}u_4} \end{bmatrix}^T \tag{2}$$

where λ is the wavelength of the signal and u_n is the unit vector in direction (ϕ, θ) for sensor n and is defined as

$$u_n(\phi, \theta) = d_{x_n} \cos(\theta) \cos(\phi) + d_{y_n} \cos(\theta) sin(\phi) \tag{3}$$

with (d_{x_n}, d_{y_n}) being the position vector of the element n

The use of QHA for MIMO reception at ground stations presents further potential and advantages over traditional setups. It is feasible to construct two QHAs, one for the VHF and one for the UHF band, one inside the other, and with the proper spacing between the elements to perform MIMO reception of satellite signals from both bands, without the need for an extra array. It is also possible for a four element array to receive up to four satellites simultaneously, just by creating a different steering vector for each one. Adding to that, different array weighting techniques can be used that will perform null steering ie. placing nulls at the directions of the satellites perceived as interference, increasing this way the SNR of the satellite being observed.

B. Signal Detection

The observation and reception of LEO satellite signals is a time-consuming and oftentimes fruitless process. Unfortunate observations occur due to inactive satellites or misleading two-line element sets (TLEs). As a result a significant number of long and "empty" observations may accumulate. Thus, the ability of detecting the presence and the exact location of actual satellite transmissions inside a long-lasting observation, in an automated way, is crucial.

In order to overcome the shortcomings described previously, a method based on the eigenvalues of the covariance matrix of the received signal [6] is exploited. More specifically, it is shown that the ratio of the maximum to the minimum eigenvalue can be used to efficiently detect the presence of a signal. The main advantage of the proposed method over other well-studied and less complex techniques such as the conventional energy detection, is the ability to "blindly" decide about the presence of a signal without the need of a priori knowledge on the noise power. Another key characteristic of the eigenvalue-based approach, especially in the context of LEO satellite observations, refers to the efficiency of signal detection under low SNR.

A remark worth mentioning is the fact that unlike energy detection, the threshold γ of the maximum-minimum eigenvalue (MME) detector may be pre-computed based only on N_S, L and P_{fa} and is independent of the noise power.

C. Open Cubesat platform

After the UPSat successful submission, LSF now targets to design and develop an open source and hardware platform for Cubesats. One of the top priorities is the contraction of an affordable SDR based communication board for small satellites among with software implementation of the modem able to run on MCUs or FPGAs for maximum flexibility and reprogrammability.

REFERENCES

[1] D. J. White, I. Giannelos, A. Zissimatos, E. Kosmas, D. Papadeas, P. Papadeas, M. Papamathaiou, N. Roussos, V. Tsiligiannis, and I. Charitopoulos, "SatNOGS: Satellite Networked Open Ground Station." Engineering Faculty Publications and Patents, 2015.

[2] osmocom Gnu Radio Blocks. [Online]. Available: http://sdr.osmocom.org/trac/wiki/GrOsmoSDR

[3] Ham Radio Control Libraries. [Online]. Available: http://hamlib.sourceforge.net/manuals/hamlib.html

[4] A QB50 cubesat by University of Patras & Libre Space Foundation. [Online]. Available: https://upsat.gr/

[5] QB50 European FP-7 project for Facilitating Access to Space. [Online]. Available: https://www.qb50.eu

[6] Y. Zeng and Y.-C. Liang, "Maximum-minimum eigenvalue detection for cognitive radio," in *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2007, pp. 1–5.