A Spectrum Sensor for CubeSat Radios

Dylan J. Gormley, and Anthony A. Stock
Cognitive Signal Processing Branch
NASA Glenn Research Center
Cleveland, OH, USA
{dylan.j.gormley, and anthony.stock}@nasa.gov

***Abstract*—Cube satellite (CubeSat) launches have increased exponentially over the last 20 years. This class of miniature spacecraft is well-suited for a set of nonconventional satellite architectures collectively known as formation flying. With the exponential pace of launches expected to continue, the prospect of spectrum management for these complex formations arises. In previous work, investigators focus on terrestrial applications of spectrum sensing, which have the luxury to utilize hardware with high size, weight, and power (SWaP) resources. In this work, we develop and test a spectrum sensor for CubeSat radio applications.**

Given that CubeSat radios are inherently designed for low SWaP, they cannot implement the computationally expensive spectral correlation analyzer (SCA) algorithms for signal detection. To that end, our investigation focuses on the application of the SCA to square-root-raised-cosine (SRRC) pulse-shaped quadrature amplitude modulation (QAM) waveforms using a field-programmable gate array (FPGA). This model requires no prior knowledge of the radio-frequency (RF) channel. We show that this model can consistently and accurately detect the symbol rate and center frequencies of waveforms located in a spectrum.

***Index Terms*—cubesat, spectrum sensing, spectral correlation analyzer, fpga**

I. Introduction

Over the last two decades, cube satellite (CubeSat) launches have increased exponentially (see Fig. 1). As a result of these CubeSats' flexibility, formation-based satellite architectures are of increasing interest to the public and private sector alike. Modern flight formations are designed to operate with tens [1], hundreds [2], or thousands [3] of CubeSats. In [4], NASA Ames Research Center (ARC) illustrates the usefulness of CubeSat swarm formations in a series of missions known as Starling 1 (see Fig. 2, 3, 4, 5).

In this work, we envision a future where these formations are commonplace. Subsequently, the inter-satellite links become vulnerable to problems experienced terrestrially, such as spectrum congestion and interference.

The use of spectrum sensing to detect these issues requires computationally complex algorithms. Thus, implementation of these algorithms requires robust hardware platforms with high-performance resources. The CubeSat, however, is a resource-limited platform. To overcome this algorithm-resource conflict, we will build upon a CubeSat-oriented feature estimator known as the streaming spectral correlation analyzer (streaming-SCA) [6].

NASA's Cognitive Communications Project supported this work.

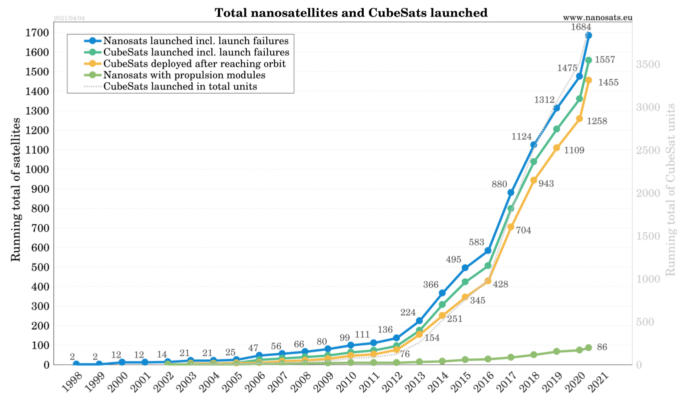

Fig. 1. Running total (cumulative sum) of launched nanosatellites and CubeSats [5].

The streaming-SCA is a memory-efficient field-programmable gate array (FPGA) design to estimate the spectral correlation function (SCF). Realization of the streaming-SCA is limited to digital quadrature amplitude modulated (QAM) waveforms with a square-root-raised-cosine (SRRC) pulse-shaping function (PSF) of an arbitrary modulation order (M). Additionally, only radio-frequency (RF) features are considered: symbol rate and center frequency [6]. With these assumptions, the streaming-SCA is expressed discretely as (1).

$$S_X^v[k] = \frac{1}{N}\sum_{n=0}^{N-1} \left|(X[n]e^{-j2\pi kn}) * h[n]\right|^2 e^{-j2\pi vn} \tag{1}$$

where $*$ is the convolution function, X is a random signal at index n, where $n = \frac{t}{T}$, N is the total number of sample indices per point estimation, $v = \lceil \frac{R_s}{f_s} N \rceil + 1$ is the index of the symbol rate, $k = \lceil \frac{f_c}{f_s} N \rceil + 1$ is the index of the center frequency, h is a low-pass filter of length two, and $S_X^v[k]$ is the estimated power of one SCF point [6].

In the following section, we build upon the streaming-SCA by adding the modules necessary to search, detect, and structure SCF estimates. These additions elevate the streaming-SCA from a simple estimator to an autonomous spectrum sensor capable of maintaining RF awareness. The newly

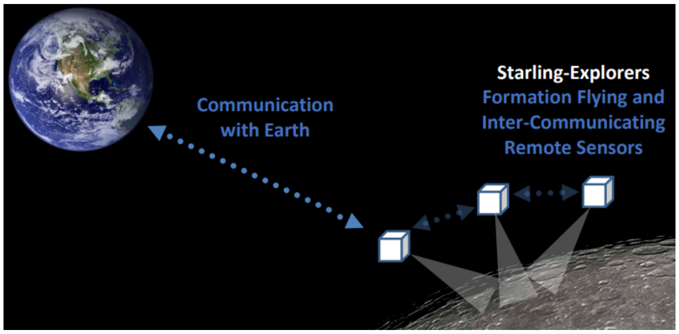

Fig. 2. Starling-Explorers provide coordinated flight and communication with sensors such as ground penetrating radar or magnetometers [4].

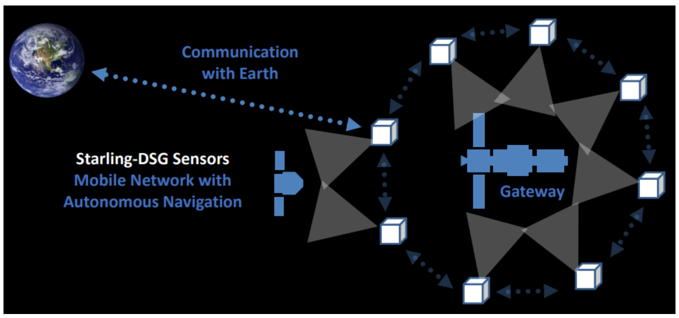

Fig. 3. Starling-DSG sensors provide visual inspection and monitoring for Deep Space Gateway (DSG) operations [4].

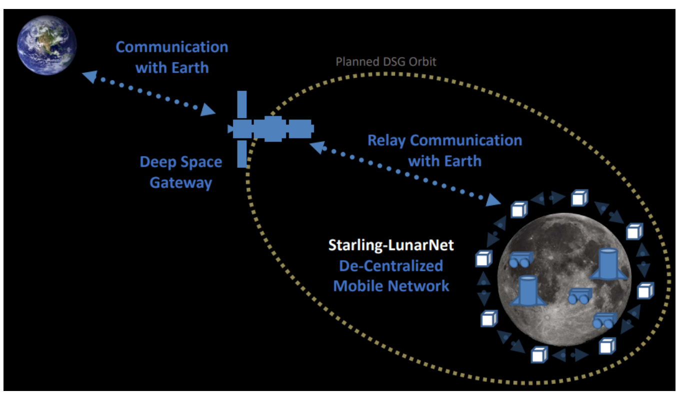

Fig. 4. Starling-LunarNet provides constant monitoring of lunar surface activity by providing an ad hoc mesh network in low Earth orbit (LEO) [4].

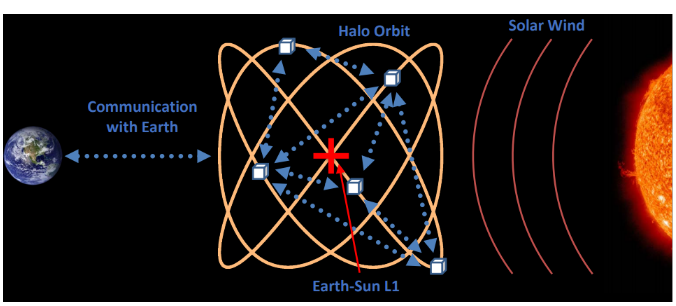

Fig. 5. Starling-SolarWind in L1 halo orbit autonomously performs radio tomography and in-site measurements to monitor and characterize solar wind [4].

formed spectrum sensor will enable dynamic spectrum management (DSM) functionality such as interference mitigation and spectrum sharing on CubeSat radios.

In this design, a scheduling algorithm manages the DSM's symbol rate requests. Each request is processed via the streaming-SCA to produce an SCF estimate $S_X^v[k]$ for the given symbol rate. Detection is then performed with a binary hypothesis test (2).

$$|S_X^v[k]|^2 \underset{H_0}{\overset{H1}{\gtrless}} \beta \left(\frac{1}{N} \sum_{n=0}^{N-1} |X[n]|^2 \right)^2 \tag{2}$$

where β is a scalar used to scale this threshold depending on the ratio between the probability of detection P_d and the probability of false alarm P_{fa}. [6]. Lastly, valid results are stored in a data structure to be used by real-time DSM algorithms.

Following the discussion of our design decisions, we provide a characterization of our spectrum sensor. To find the optimal β, a receiver operating characteristic (ROC) curve was generated (see Fig. 6) for a noisy channel containing two interfering signals (see Table I).

TABLE I
CO-CHANNEL INTERFERENCE USED FOR THE GENERATION OF AN ROC CURVE

#	N	F_s	R_s	f_c	M	r	E_b/N_0
0	2^{12}	$1.0MHz$	$0.1MBd$	$0.1MHz$	32	0.2	$3.0dB$
1	2^{12}	$1.0MHz$	$0.2MBd$	$0.2MHz$	32	0.2	$3.0dB$

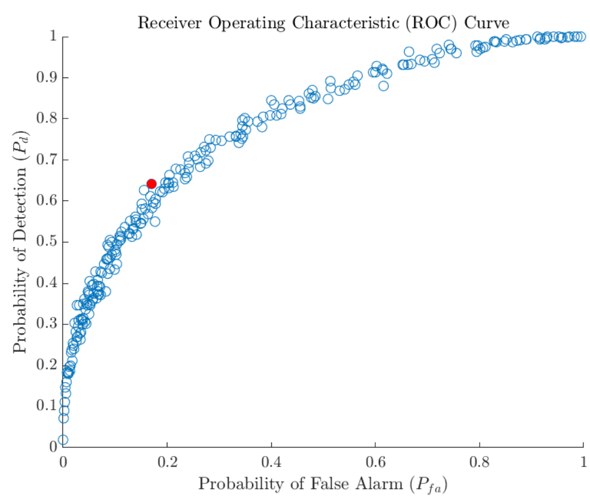

Fig. 6. ROC curve of a noisy channel (see Table 1) in a model of (2) to predict performance given various values of β. Parameters used for curve generation were: confidence level of 95%, β range of 1.0×10^{-6} to 1.0×10^{-3} in steps of 1.0×10^{-6}, and the average of 300 trials per point. The optimal operating point (red) is calculated using (3). Here, J is found to be $(0.64167, 0.47167)$ which was produced using a β of 1.23×10^{-4}.

By observing the trend of P_{fa}, we are able to determine an amount of acceptable risk and select a corresponding β. In

Fig. 6, the optimal point $J = (P_{fa}, P_d)$ is calculated using Youden's J Statistic (3).

$$J = \frac{\sum P_d}{\sum P_d + \sum\left(1 - P_{fa}\right)} + \frac{\left(\sum 1 - P_d\right)}{\left(\sum 1 - P_d\right) + \sum P_{fa}} - 1 \quad (3)$$

As discussed in [6], the threshold is proportional to the sufficient statistic of the received signal's power. A threshold is said to be optimal if it achieves the lowest P_{fa} for a given P_d [6]. To minimize false alarms, the β for this design was found to be 1.23×10^{-4}.

Finally, we experiment with this optimal threshold by parametrically impairing the transmitted waveform's ${}^{E_b}/_{N_0}$ and restricting its excess bandwidth r. To conclude this paper, we summarize the objective, development, and results, followed by a discussion of opportunities for future work.

II. Digital Design

This section discusses the digital design enhancements made to the streaming-SCA. These enhancements allow it to operate autonomously as a spectrum sensor.

A. Threshold

We begin the discussion of our streaming-SCA additions with the threshold. The digital design of our threshold begins by receiving in-phase and quadrature (IQ) samples directly from the analog to digital converter (ADC). As shown in Fig. 7, the samples are serially fed into a magnitude-squared module, accumulated, right-shifted, and finally squared. The value of β is constant and set before synthesis.

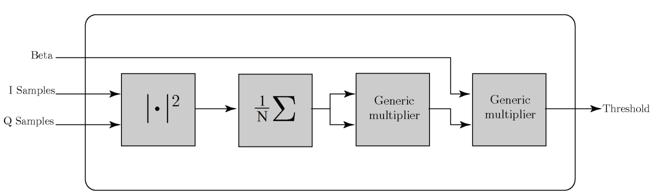

Fig. 7. Block diagram of the threshold module.

B. Center Frequency Detector

The center frequency detector (CFD) module is the full implementation of the binary hypothesis test. The sufficient statistic $l = |S_X^v[k]|^2$ generates a streaming-SCA instance in parallel with the threshold module. These two modules are synchronized, and then evaluated using the test statistic 2.

The digital design operates by receiving a v and an FCW step size, determined by the resolution of k values to be searched. For a given v, the CFD compares each value against the threshold to determine detection. Following this, the test statistic with the highest k is saved, resulting in a detected (v, k) pair. This design is illustrated in Fig. 8.

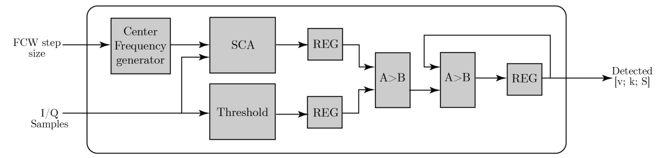

Fig. 8. Block diagram of the center frequency detector module.

C. Spectrum Sensor

The spectrum sensor module is tasked with storing requested values of v from the DSM and storing them into a FIFO. These received values of v are then assigned to available CFD instances. A process polls the CFD instances to determine which ones are ready to receive inputs.

Because the CFD scheduling process can only assign one candidate k at a time, we are guaranteed to have no more than one CFD result at any point in time. Thus, a demultiplexer can assign resulting (v, k, l) tuples to a result FIFO, which are to be read by a DSM. This design is illustrated in Fig. 9.

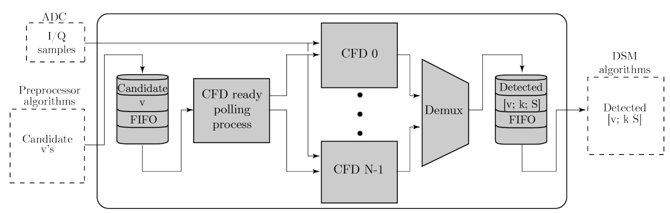

Fig. 9. Block diagram of the spectrum sensor module.

III. Results

In this section, we characterize the results of our spectrum sensor by reporting its performance under various conditions. Three hundred trials were averaged for each of the scenarios listed in Table II.

A. Characterization

TABLE II
Spectrum Sensor Performance

Experiment #	${}^{E_b}/_{N_0}$ (dB)	r	P_d	P_{fa}
baseline	15.0	0.35	1.0	0.0013
1	12.0	0.35	1.0	0.0284
2	9.0	0.35	1.0	0.0584
3	6.0	0.35	1.0	0.0854
4	3.0	0.35	0.9967	0.1146
5	0.0	0.35	0.89	0.1213
6	15.0	0.30	1.0	0.0108
7	15.0	0.25	1.0	0.0108
8	15.0	0.20	1.0	0.0150

We find that for a fixed $\beta = 1.23 \times 10^{-4}$, our spectrum sensor demonstrated a high likelihood of detection. Even for an ${}^{E_b}/_{N_0}$ of zero, a 89% probability of detection is achieved. For all but two experiments, zero misses were encountered.

The P_{fa} is found to be encouragingly low, with the worst-case scenario, experiment 5, yielding a 12.13% probability of erroneous detection. These results confirm the accuracy of the ROC curve method used to calculate the optimal value of β.

In [6], detection of v is found to tolerate little offset error. The v that is sensed must be very close to the true actual v to avoid a miss. However, k is found to be highly tolerant to error. This means that given that the resultant v is correct, the corresponding k is still counted as a successful detection even if it does not accurately represent the actual center frequency of the incident signal.

Depending on properties of the received waveform, such as E_b/N_0 or r, the variance of the detected k will increase or decrease. That is, higher values of E_b/N_0 and r results in less variance of k detection. Fig. 10 illustrates how E_b/N_0 affects the variance of k.

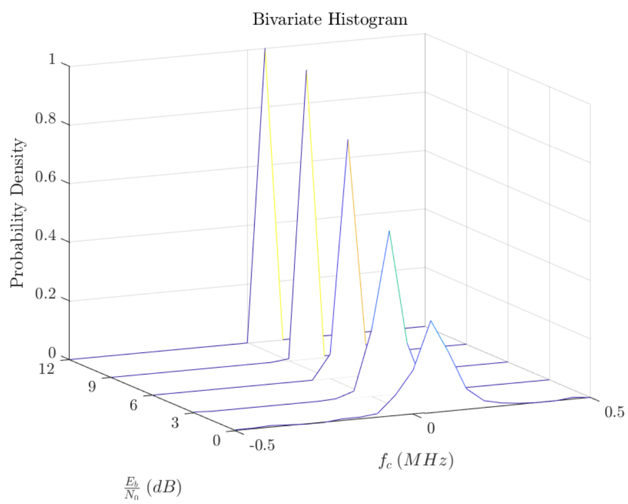

Fig. 10. A bivariate histogram depicting the probability density functions (PDF) of center frequency detection for several values of E_b/N_0. For a received waveform with little impairment, in this case noise, the variance of detected center frequencies is smaller. Conversely, the detected center frequency is more likely to vary from the true center frequency of the received signal when the received waveform is impaired.

B. Timing

To demonstrate the behavior of the spectrum sensor's scheduling scheme, we consider a simple instance of the spectrum sensor with only two CFDs and input/output FIFO buffers of size two. As illustrated in Fig. 11, the upstream system provides v candidates to be searched. The spectrum sensor's internal polling process automatically assigns the first two alpha values $A1$ and $A2$ to the two CFDs, respectively. Each of the following v candidates, $A3$ and $A4$, must wait in the v candidate FIFO until a CFD instance is available. The fifth v candidate, $A5$, is ignored, as the FIFO has no space available. This is indicated to an upstream process with the `inFifo_full` flag.

After some processing time, the results corresponding to $A1$ and $A2$ are pushed to the result FIFO. This allows for $A3$ and $A4$ to be assigned to both CFD instances, one after the other.

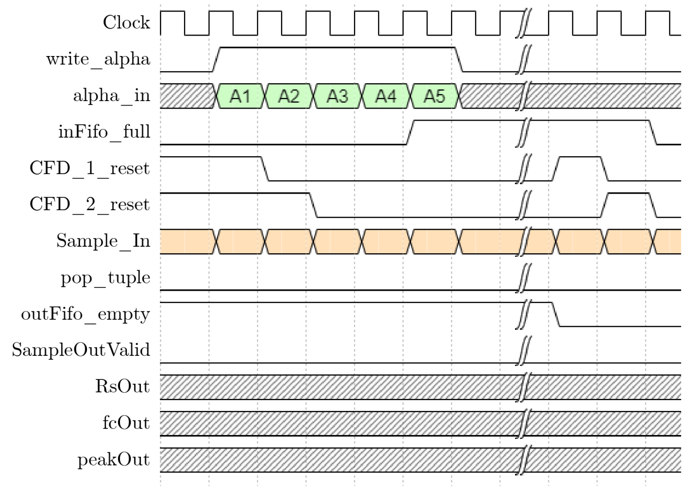

Fig. 11. Timing diagram depicting the process of pushing v candidates to the input FIFO.

In the output of the spectrum sensor module (see Fig. 11), we assume that the downstream module has not yet made any pop requests. When the results corresponding to $A3$ and $A4$ are produced, the oldest results currently stored in the result FIFO are discarded. Finally, two sequential pop requests are made, resulting in the remaining results being driven sequentially to their respective outputs.

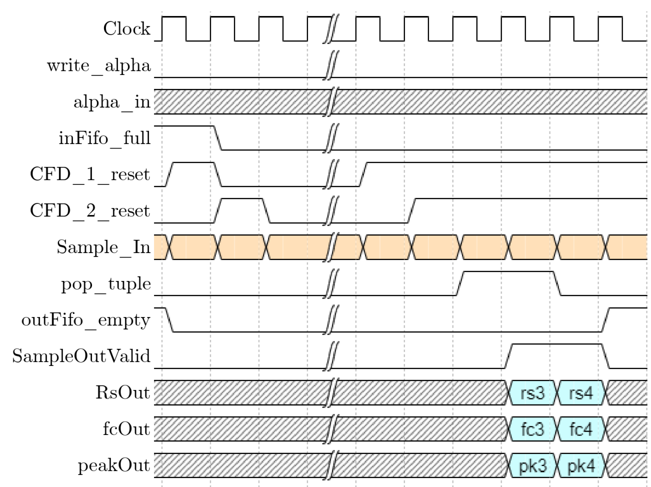

Fig. 12. Spectrum sensor timing diagram, continued. In the event that the result FIFO is full, older results are discarded to prioritize more recent values.

IV. Conclusion

We began this paper by discussing CubeSat flight formations. We then discussed an FPGA algorithm used to blindly estimate the symbol rate and center frequency of an M-QAM-SRRC waveform, known as the streaming-SCA. Following this, we discussed how a spectrum sensor could be designed using the streaming-SCA by adding scheduling algorithms, detection, and data structures. Finally, we performed various experiments to characterize the probability of failure for our spectrum sensor.

With the addition of a scheduling algorithm and data structures, various instances of the CFD can now be processed asynchronously. For future work, we plan to leverage this parallel, asynchronous design to gain speed performance. Two techniques to be investigated are dynamic CFD decimation to decrease CFD processing time and preprocessors to narrow the search space of our spectrum sensor. Lastly, with an optimal threshold now available, it is possible to test waveforms other than M-QAM-SRRC, which require more sensitive thresholds.

ACKNOWLEDGMENT

The authors would like to thank Erica K. Henrichsen for the graphic designs illustrated in this paper.

REFERENCES

[1] L. Plice, A. D. Perez, and S. West, "Helioswarm: Swarm mission design in high altitude orbit for heliophysics," in *Proceedings of the 34nd Annual AIAA/AAS Astrodynamics Specialist Conference*, ser. AAS19. Portland, ME, USA: Digital Commons, 2019, p. 831.

[2] H. Sanchez, D. Mcintosh, H. Cannon, C.Pites, J. Sullivan, S. D'Amico, and B. O'Connor, "Starling 1: Swarm technology demonstration," in *Proceedings of the 32nd Annual AIAA/USU Conference on Small Satellites*, ser. SSC18. Logan, UT, USA: Digital Commons, 2018, pp. VII–08.

[3] G. Giambene, S. Kota, and P. Pillai, "Satellite-5g integration: A network perspective," *IEEE Network*, vol. 32, no. 5, pp. 25–31, 2018.

[4] V. Bennett, K. Bowman, and S. Wright, "Starling 1: Mission technologies overview," NASA Ames Research Center, Moffett Field, California, USA, Tech. Rep. ARC-E-DAA-TN59780, August 2018.

[5] K. Kulu. (2021, April) Nanosats database. [Online]. Available: https://www.nanosats.eu/

[6] D. J. Gormley, "A low-memory spectral-correlation analyzer for digital root-raised-cosine pulse shaped waveforms," Master's thesis, Cleveland State University, Cleveland, OH, USA, May 2021.