A COGNITIVE RADIO PROTOTYPE FOR SPECTRUM SENSING APPLICATION

Michael Steiner (Virginia Polytechnic Institute and State University, Blacksburg, Virginia, United States; mstein02@vt.edu); S. M. Hasan (Virginia Polytechnic Institute and State University, Blacksburg, Virginia, United States; hasan@vt.edu); Tamal Bose (Virginia Polytechnic Institute and State University, Blacksburg, Virginia, United States; tbose@vt.edu); Ahsan Aziz (National Instruments, Austin, Texas, United States; ahsan.aziz@ni.com); Sam Shearman (National Instruments, Austin, Texas, United States; sam.shearman@ni.com)

ABSTRACT

This paper describes the implementation of a cyclostationarity-based spectrum-sensing algorithm on a new radio development platform. The prototype will perform as a cognitive radio node capable of identifying white spaces (unused portions of the frequency band) in the given frequency. This prototype uses LabVIEW, and National Instruments equipment will be distributed across three processing platforms – a field-programmable gate array, a real-time operating system, and a host computer. Signal detection is performed by searching for features in the cyclic spectrum. The ability of the platform to support this spectrum-sensing algorithm and future sensing, classification, and cognitive projects is described.

1. INTRODUCTION

The volume and speed of research being performed in the area of cognitive radios and cognitive radio networks demand new platforms for developing testing algorithms with minimal overhead. This paper discusses a cognitive radio prototype developed for the purpose of studying spectral sensing and modulation classification techniques.

The radio platform was developed by National Instruments (NI) as a versatile development system for real-time applications. The primary advantage of the NI platform is the common interface through the LabVIEW programming environment. This includes programming the field-programmable gate array (FPGA) without coding in VHDL or a similar language. The system provides several components capable of distributing the processing load.

The platform was chosen for spectrum sensing purposes due to its Real-Time Module with a real-time operating system, which allows for processes to be run deterministically. Cyclostationarity-based spectrum sensing is a computationally complex method [1] that was chosen to show the functionality of the prototype radio.

A brief description of the hardware platform with an emphasis on a few select attributes is described in Section 2. Section 3 reviews the fundamentals of cyclostationarity-based detection and introduces the Frequency Accumulation Method (FAM). Implementation of the FAM algorithm on the platform as well as some of the timing details and resource usage consumed by the algorithm is discussed in Section 4. Section 6 states some conclusions about the limitations and value of the system in reference to the findings described in Section 5. Future plans for using the platform in radio development are presented in Section 7.

2. HARDWARE DESCRIPTION

The system consists of six pieces of hardware: an Ettus XCVR2150 transceiver board, an NI interposer board, an NI 5781 Adapter Module, an NI 7695 FPGA Module, an NI 8130 Real-Time Module, and a personal computer, which provides the user interface. Each piece can be programmed through the LabVIEW software interface. This interface is set up as a series of Virtual Instruments (VIs) that can be run on the personal computer or Real-Time Module or can be compiled directly from LabVIEW and downloaded to an FPGA.

The front end of the receiver system is an Ettus XCVR2450 transceiver board operating in the 2.4–2.5-GHz band. The output is analog in-phase and quadrature channels with a controllable receiver bandwidth up to 24 MHz.

The transceiver is connected to an NI 5781 Adapter Module through a proprietary NI interposer board and a pair of differential cables for each channel. The NI 5781 Adapter Module houses a 14-bit analog to digital converter and 16-bit digital to analog converter with sample rates up to 100 MSps.

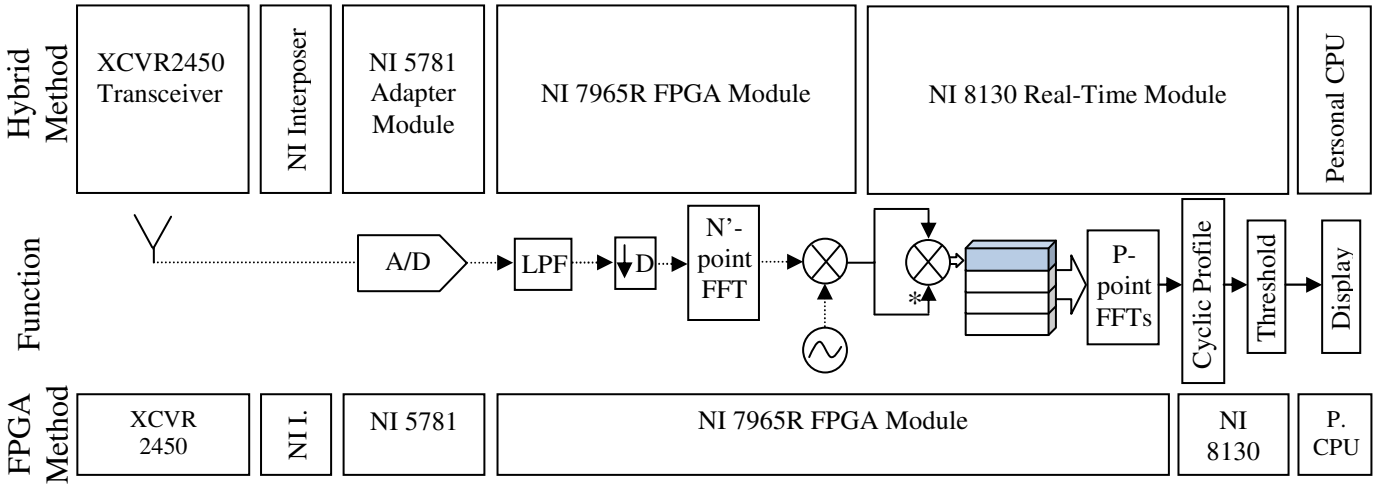

Figure 1. The system layout consists of a series of components with the FAM algorithm distributed between the FPGA Module and the Real-Time Module. The controls are routed to the transceiver and Adapter Module through the FPGA which controls the serial interface connection. The personal computer is used only as a user interface in this implementation.

The digital output is directly interfaced with the NI PXIe 7965 FPGA Module which contains a Vertex-5 FPGA. On-board memory allows for first-in-first-out (FIFO) queues for data transfer between multiple clock domains on the FPGA. Direct memory access (DMA) provides 16 communication channels to the host real-time system reaching a theoretical limit of 800 MB/s.

The NI 8130 Real-Time system operates a dual core 2.3 GHz processor with a real-time version of Windows operating software. The operating system allows for timed loops capable of synchronizing processes or operating function deterministically. The module can be connected to a personal computer through gigabit Ethernet. Figure 1 illustrates the arrangements of the components in the system.

3. CYCLOSTATIONARITY BASED DETECTION

The implementation of a time-smoothed cyclostationary detection algorithm on the radio platform is described in this paper. Cyclostationarity is a characteristic of many man-made signals; it can be defined as a periodicity in the autocorrelation function [1]. The time-smoothed spectral autocorrelation function (SCF) is shown in equation 1 [2].

$$S_x^{\alpha}(f) = \frac{1}{T}\langle X\left(f+\frac{\alpha}{2}\right)X^{*}\left(f-\frac{\alpha}{2}\right)\rangle_{\Delta t} \quad (1)$$

$X(\boldsymbol{f}+\boldsymbol{\alpha}/\mathbf{2})$ is a complex demodulate of the $x(n)$ signal of interest, T is the duration over which the demodulate is formed, and $\boldsymbol{\Delta t}$ the overall duration over which the SCF is estimated. The term α represents the cycle frequency which is the separation in frequency between the demodulates to be correlated.

The advantage of cyclostationarity based detection is that noise is generally considered to be a wide sense stationary process and therefore it has a reduced effect on the estimate [1][2]. The sacrifice for eliminating noise takes the form of greatly increased complexity compared to that of an energy detector. Roberts et al. [2] describe several efficient methods for computing the cyclic spectrum of signals. The frequency accumulation method (FAM) was chosen because of its versatility as a time smoothed algorithm and its judicious use of the fast Fourier Transform (FFT).

The FAM occurs in four steps. First, the signal is windowed and passed through an N'-point FFT where $N' = N/P$, N is the total number of points to be processed, and P is the number of windows. The sampling is performed over Δt seconds. Each output bin is then converted to baseband. Next, these baseband bins are used to compute the cyclic spectrum by correlating bins separated by α, where α is the cyclic frequency related to the sampling

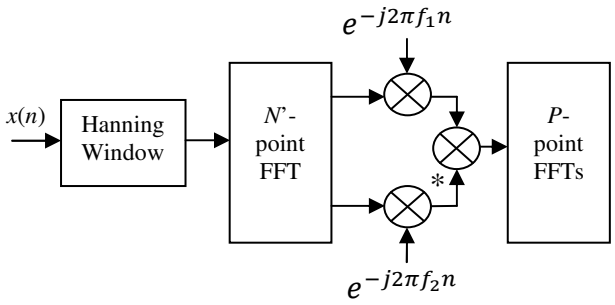

Figure 2. FAM implementation [2]. The signal, $x(n)$, is received in groups of N' samples. A P-point FFT is calculated for each region in the cyclic spectrum formed by the correlation of the complex demodulates.

frequency by f_s/N', creating an N'xN' matrix. Finally, a set of P of these matrices is passed through a P-point FFT to perform the time smoothing. This process is depicted in Figure 2 [2].

A reliable estimate requires that $\Delta t \Delta f \gg 1$ [1][2], where $\Delta f = f_s/N'$. The complexity of the FAM method increases proportionally to $\Delta t \log(\Delta t)$ for a fixed frequency resolution [1].

The correlation coefficients are computed by implementing [3]

$$C_x^{\alpha}(f) \triangleq \frac{S_x^{\alpha}(f)}{\sqrt{S\left(f+\frac{\alpha}{2}\right)S\left(f-\frac{\alpha}{2}\right)}} \tag{3}$$

which yields a number between 0 and 1 to describe the cyclic correlation at each point in the SCF.

A cyclic profile is developed from the maximum cyclic coherence across frequency. The values of the cyclic profile are then compared to a threshold value to determine if a signal is present. An example of a cyclic profile for a BPSK signal is shown in Figure 3.

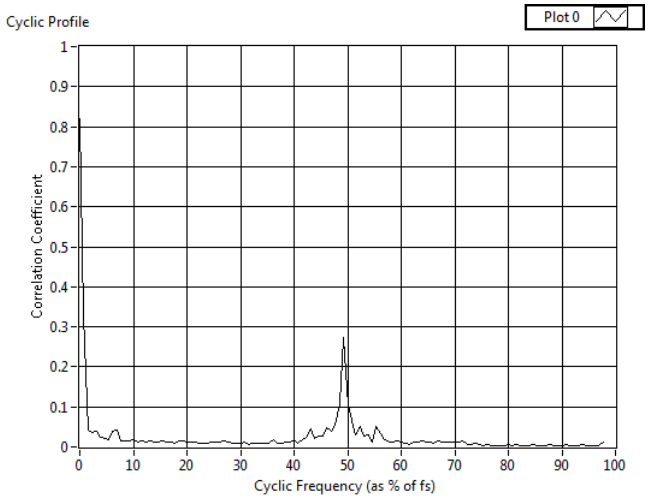

Figure 3. Cyclic profile of a BPSK signal with $N' = P = 128$.

4. IMPLEMENTATION

The algorithm is developed on the platform in two forms. The first places the majority of the algorithm on the FPGA (FPGA implementation) to create a real-time implementation. The second design distributes more of the algorithm on the Real-Time module (hybrid implementation) which has the potential for a higher frequency resolution, although at the expense of being a real-time implementation.

4.1. Shared Implementation

The FPGA and hybrid implementation share a common structure from sampling the received signal to the N'-point FFT on the FPGA.

The input signal is sampled at 25 MSps and then filtered and downsampled to a rate of 2 MSps. After downsampling, the resolution of the data is reduced to 16 bits for transfer to a slower loop where it is windowed and then processed through an N'-point FFT. While the structure is the same for both implementations, the window length and size of the FFT will depend upon the desired frequency resolution in the algorithm output. The choice of N' is restricted in the FPGA implementation based on the choice of P and the available onboard memory.

In the case of the hybrid realization, the output of the FFT is transferred to the RT module through a DMA FIFO for further processing. The FPGA implementation similarly uses an FIFO stored in onboard memory to transfer the FFT output to a higher frequency loop for correlation and smoothing.

4.2. FPGA Implementation

The correlation and smoothing on the FPGA are a pipelined structure, which allows for the high cycle rate necessary to compute the N'xN' correlation matrix. Two points in the correlation matrix are computed simultaneously to reduce the minimum cycle rate to N'xB.

The correlated data are stored in onboard memory until P correlation matrices are computed. Storage is used so that only a pair of P-point FFTs is necessary to smooth the cyclic spectrum instead of an FFT for each bin in the SCF, creating a dramatic savings in space on the FPGA. Figure 4 demonstrates how subsequent sample blocks are computed. Onboard memory storage limits the choice of N' and P. For this implementation, a reasonable value based on the available resources was chosen to be N' = P = 32.

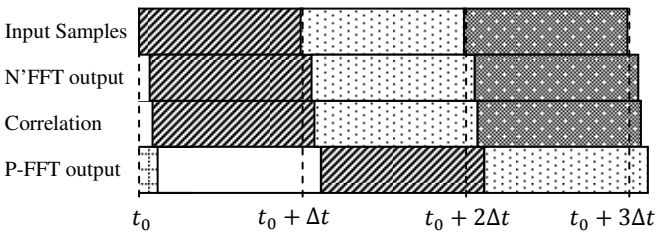

Figure 4. Timing chart demonstrating the parallel processing of sequential input blocks. Delays are introduced by complex functions, such as FFTs and filters, as well as general pipelining to maximize FPGA speed.

4.3. Hybrid Implementation

The hybrid implementation performs the bulk of the computations on general purpose processors controlled by the Real-Time Module. Two distinct advantages of the Hybrid method are the ability to use floating point numbers and a relatively large bank of memory for storing correlated data. Instead of a focus on pipelining as with the FPGA implementation, the Hybrid method focused on dividing the data into two paths after correlation allowed for simultaneous use of both processor cores.

5. RESULTS

The abilities of the FAM cyclostationary detector have been documented in [1], [2], and [3]. For the purposes of cognitive radios, rapid detection and identification of other users is critical. This section focuses on the timing details of the two implementation methods described in this paper.

5.1. FPGA Implementation Results

The FPGA implementation allowed for highly parallelized computation of the cyclostationarity-based detection algorithm. This structure was particularly useful in computing the correlation matrix and *P*-point smoothing FFTs.

Delays are necessarily exhibited throughout the design due to innately pipelined functions, such as the input filter and FFT calculations, as well as the necessary pipelining to meet timing constraints. Figure 4 illustrates how the delays are positioned throughout the algorithm. The most obvious delay occurs because of the need to wait for *P N'*x*N'* correlation matrices to be computed before beginning the smoothing operation. If an FFT structure was devoted to each bin of the SCF, the smoothing computation could be completed sooner relative to the last sample of the set; however, the cost would be an extreme amount of resources on the FPGA. The structure implemented as is still achieves a real-time result. The latency between the last input sample in a data set and the completed smoothing operation is just over one data-set interval.

The other significant source of latency is the delay caused by the input filter and *N*'-point FFT. This delay is relative to the length of the filter and FFT, although modification will result in lower performance of the system. A representative account of the delays for a system with *N'* = *P* = 32 with a 50-tap low-pass filter is found in Table 1. The high tap count for the filter is used for a sharp cutoff of images that would influence the downsampling process. The delay between the first input of a dataset and the output is about 607 μs, although most of this delay is due to the need to wait for all samples to be gathered before beginning the smoothing process. A more representative description of the delay is the time taken from the last input sample in a dataset to the moment the SCF is available, about 95 μs.

The SCF generated on the FPGA is then sent to the RT system for final processing through a DMA FIFO. This is done so that cyclic features can be observed first-hand and the SCF can be used for additional processing if desired.

Table 1. Summary of cycle delays in the FPGA implementation

Delay Type	Count	Cycle Frequency (MHz)	Delay (μs)
Filter/Downsampler	51	25	2.04
Window/*N*'FFT	175	2	87.50
Memory Retrieval	16384	32	512.00
P-point FFTs	129	32	4.03
Pipelining/FIFO	16	-	0.95
Total	-	-	606.52

5.2. Hybrid Implementation Results

While the operations performed on the FPGA operate at a given clock speed, the functions on the real-time system are limited by the processor speed. LabVIEW allows for timed loops to be given dedicated processors on a multi-core machine, speeding up operations. This feature is useful for operations such as the *P*-point FFT calculations which can be performed in parallel. Table 2 details some measured timing results from a few key calculations computed on the Real-Time Module.

Table 2. Timing details of critical FAM functions on the Real-Time Module.

Function	Ideal Number of Complex Multiplications [2]	Time (ms)
Correlation (single core)	$(N')^2P$	0.037
P-point FFTs (single core)	$(N')^2(P/2)\log_2 P$	343.0
P-point FFTs (dual core)	$(N')^2(P/2)\log_2 P$	112.7

The number of ideal complex multiplications is included for comparison. Other operations, such as array manipulation and overhead for using dual cores, may influence the timing results.

Although dual-core operation provides a decrease in the duration of the FFT calculation, using a single core allows for other critical loops, such as data acquisition from the FPGA FIFO, to continue. However, because the duration of the *P*-point FFT calculation exceeds Δt, both cores are best used on this process. As a result, the algorithm cannot

perform in real time and instead hopped over input samples in blocks.

The best use of the Hybrid implementation in cognitive radios is to use a single core for cyclostationary analysis and the dual core for time-critical operations.

6. CONCLUSIONS

Cyclostationarity-based spectrum detection was chosen to test this system due to its high computational complexity as well as to provide versatile means for identifying signals based on unique features.

The FPGA implementation is able to run a real-time, low-resolution version of the cyclostationary-based detection algorithm. By using the versatility of the platform, the algorithm was also spread across both the FPGA and Real-Time Module, allowing for improved frequency resolution in the Hybrid implementation. This gain in resolution came at the expense of loss as a real-time solution. Another approach for implementing time-smoothed cyclostationary analysis in real time include parallelizing the *P*-point FFTs in multiple processing elements [2]. This process was discussed and deemed unrealizable on the current resources due to poor scaling to high-frequency resolutions.

7. FUTURE WORK

The purpose of the radio system is use as an experimental test platform for spectrum sensing and modulation classification techniques. Adding the Hybrid implementation with a less complex sensing method, such as an energy detector, would create a radio platform capable of wideband spectrum sensing with primary user identification through cyclostationary features.

Adding modulation identification and other forms of spectrum awareness allows for the implementation of cognitive functions. The platform is designed for two-way communication, a characteristic not explored in this paper.

The radio platform will be added to the Virginia Tech Cognitive Radio Network Testbed (VT-CORNET). Designed to house 48 radio nodes, the system can be expanded to handle additional hardware such as this platform. Adding heterogeneous hardware to the network is beneficial when simulating a realistic dynamic spectrum access (DSA) protocol due to the likelihood of diverse DSA commercial radio hardware.

8. ACKNOWLEDGMENT

The authors would like to thank WICAT and National Instruments for their donations and support of this project.

9. REFERENCES

[1] W.A. Brown and H.H. Loomis, "Digital Implementations of Spectral Correlation Analyzers," *IEEE Transactions on Signal Processing*, Vol. 41, pp 703-772, 02, February 1993.

[2] R.S. Roberts, W.A. Brown, and H.H. Loomis, "Computationally Efficient Algorithms for Cyclic Spectral Analysis," *IEEE SP Magazine*, pp 38-49, April 1991.

[3] K. Kim, I.A. Akbar, K.K. Bae, J. Um, C.M. Spooner, and J.H. Reed, "Cyclostationary Approaches to Signal Detection and Classification in Cognitive Radio," *2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 212-215, 17-20 April 2007.