

Practical Implementation of Adaptive Threshold Energy Detection using Software Defined Radio

MICHAEL V. LIPSKI^{ID}, Graduate Student Member, IEEE
Pennsylvania State University, University Park PA, USA

SASTRY KOMPELLA^{ID}, Senior Member, IEEE
U.S. Naval Research Laboratory, Washington, DC USA

RAM M. NARAYANAN^{ID}, Life Fellow, IEEE
Pennsylvania State University, University Park PA, USA

Spectrum awareness is a fundamental characteristic of cognitive radio, and spectrum sensing is the local procedure by which the cognitive radio gains knowledge of spectrum users. Energy detection is the most widely used and widely studied form of spectrum sensing. Much of the information about energy detector performance comes from theory and simulation rather than experimental data. We acknowledge a need for empirical data that can be used to evaluate a hardware energy detector and establish expectations on the differences in performance when comparing a hardware energy detector with simulations. In this article, we build and test a real-time, adaptive threshold energy detector using a USRP software-defined radio (SDR). While several groups have built energy detectors using SDRs, we found that there is still a lack of data on the parameters and performance characteristics of SDR-based energy detectors. Our work covers in detail the construction of the SDR energy detector and includes specific hardware and software parameters as well as several practical considerations. We discuss the procedure used to benchmark the energy detector and include experimental results that show how several implementation parameters affect the detector performance. Our work also explores the use of moving average windows to formulate the detection statistic and focuses on the importance of the length of the window as well as the shape of the window.

Manuscript received May 12, 2020; revised September 9, 2020 and November 2, 2020; released for publication November 3, 2020. Date of publication November 24, 2020; date of current version April 10, 2021.

DOI. No. 10.1109/TAES.2020.3040059

Refereeing of this contribution was handled by K. S. Kulpa.

This work was supported by the US Office of Naval Research under Grant N00014-17-1-2386.

Authors' addresses: Michael V. Lipski and Ram M. Narayanan are with the Pennsylvania State University, University Park, PA 16802 USA, E-mail: (mvl5023@psu.edu; ram@engr.psu.edu); Sastry Kompella is with the U.S. Naval Research Laboratory, Washington, DC 20375 USA, E-mail: (sk@ieee.org). (*Corresponding author: Ram M. Narayanan.*)

0018-9251 © 2020 IEEE.

I. INTRODUCTION

Cognitive radio (CR) and dynamic spectrum access (DSA) have been highly active areas of research over the past two decades. This is due in part to the huge number of wireless devices available to the consumer and the continually increasing number of services that require access to the wireless spectrum. Consider the average individual, who almost certainly possesses a smartphone, and is also likely to use some combination of a laptop, tablet, smart watch, and wireless headphones. Beyond personal electronic devices, automation and internet of things (IoT) concepts and technologies, as well as cloud computing and big data, are driving forces behind an increase in the number of home appliances and sensors that are becoming networked. Refrigerators, washing machines, security cameras, thermostats, and electric meters are just a few examples; these common household devices typically communicate with a local hub and have the potential to generate an enormous amount of data that is transmitted using wireless communications protocols. Following the trend of research and development and consumer demands, it becomes readily apparent that the internet is quickly infiltrating all aspects of society in new and fascinating ways. Researchers have worked extensively on developing the concept of the “Smart City,” which proposes ubiquitous connectivity and networking, or IoT on a grander scale for urban settings [1], [2].

Growing bandwidth demands and the oft-stated inefficiency of fixed spectrum allocation motivate novel paradigms. 5G is perhaps the most promising of these: For example, by opening up unused mmWave bands, data rates in the Gbps are feasible, and massive MIMO helps to address the multiple access interference problem [3]. Still, adaptability and dynamic spectrum access are crucial aspects of supporting spectral coexistence of a massive number of devices, particularly if a considerable number of fixed-band legacy devices are in operation. Shifting context, we now consider critical RF connectivity in a contested environment or battlefield, such as a distributed ad hoc network enabling tactical communications among ground units. Spectrum coexistence and reuse are crucial aspects of such wireless systems. Radar and communications systems often operate in overlapping bands, and the addition of hostile transceivers, jamming systems, and environmental factors forces adaptation in order to function in a changing and often unpredictable spectrum landscape. The concept of the cognitive user (CU) is designed around the ideas of flexibility and intelligent adaptation. Spectrum awareness, or knowledge of the spectrum environment, is foundational to the CU paradigm.

In a contested or unpredictable environment, a cognitive radio network (CRN) would need at least one node that is sensing the spectrum locally. An effective spectrum sensing solution incorporates multiple sensing methodologies. For blind spectrum sensing, energy detection is an integral part of the solution because of its low complexity and universal applicability. No *a priori* knowledge of the received signal is needed because energy detection simply looks at received

energy magnitude at a particular time and/or frequency span.

A. Previous Work

One of the biggest challenges to implementing an energy detector is choosing the threshold. The classical energy detection approach employs a fixed threshold; a binary hypothesis testing problem is formulated using dual Gaussian distributions for the signal and the signal plus noise. From the Gaussian distributions, a threshold value can be derived by positing a desired probability of false alarm, yielding the well-known constant false alarm rate (CFAR) detector. In order to calculate the fixed threshold, an estimate of the noise variance is required. Various methods of estimating the noise variance for threshold calculation have been proposed [4], [5].

A fixed threshold can also be calculated using a desired probability of detection, or constant detection rate (CDR), although this approach incurs an increased level of difficulty in implementation because in addition to the noise variance, the energy detector would also need to estimate the signal to noise ratio (SNR). Regardless of which parameter is used to calculate the threshold, energy detection suffers from uncertainty and thus poor performance in low SNR regimes. A number of adaptive threshold approaches have been designed to improve the performance of energy detectors. Some of these methods use a modified CFAR, CDR, or some combination of the two [6]–[10].

One proposed algorithm, which is the basis for the energy detector that was built and tested for this work, calculates the mean and standard deviation of the entire received signal, without needing to differentiate between the signal and the noise-only samples [11], [12]. Other, less conventional detection threshold setting methods have been explored; one group showed the efficacy of iterative methods such as forward consecutive mean excision (FCME) and forward cell averaging (FCA) [13]–[15]. These adaptive detection threshold approaches possess the advantages of being resistant to changes in the noise floor and not needing an explicit estimate of the noise variance.

Interest in energy detection over the past decade has been motivated in part by the desire for energy efficient cooperative spectrum sensing (CSS) in CRNs. Limitations in the performance of energy detection that are present at the level of a single detector can be mitigated to some extent via intelligent fusion of data from multiple sensors spread across an area. One proposed method uses channel estimation to determine the number of samples taken for each detection cycle [16]. Other cooperative schemes explore cognitive relaying with multiple fusion methods [17] and improved energy detection with threshold-based censoring [18] using channels with low SNR and different fading models. This prior work indicates that cooperative energy detection schemes can improve detection performance and decrease false alarm probability compared to conventional energy detection in low SNR and fading conditions.

Previous work has demonstrated energy detection on SDR in a number of contexts [19]–[23]. SDR is an attractive platform for experimentation and it can be seen as a natural precursor to cognitive radio. At its core, the most basic SDR is a transceiver IC (a “radio on a chip”) and an FPGA for handling DSP. What makes it cognitive is the “brains”, or the algorithms and protocols that make the radio agile in time, frequency, bandwidth, and modulation parameters. These capabilities are crucial aspects of the CR that allow it to observe the radio environment and adapt intelligently to best fulfill its role. Higher layer protocols and network function virtualization can enable a CR to fulfill multiple roles under different scenarios and form ad hoc cognitive radio networks (CRNs) or join existing *ad hoc* or infrastructure networks. The cognitive cycle begins with spectrum sensing [24]; it is in this context that we present in detail the development of an adaptive threshold energy detector using an SDR, with an emphasis on the practical considerations and challenges posed by the hardware and software.

B. Present Work

The theory of energy detection is already well developed, and many energy detection methods have been supported with simulation results. This work specifically seeks to help bridge the gap between simulation and hardware implementation.

In this work, we explore the development of a real-time adaptive threshold energy detector using a software defined radio (SDR) and open-source software tools. We then conduct experiments to gauge performance characteristics of the energy detector using a second SDR to transmit several different test signals. Although previous work has modeled energy detection over fading and noisy channels, many real-world environments in which a CR might be deployed are complex and affect signals in ways that are not easily accounted for in simulations. Thus, there is a need for experimentation in hardware. The SDR-based energy detector presented here is distinct from prior implementations because it is a real-time detector employing an adaptive threshold that does not necessarily require a priori information about the channel conditions or SNR. However, as we will see, detection performance can be improved by using knowledge of signal characteristics to tune operating parameters. We also introduce the use of different moving average windows during the sampling process and provide experimental data that demonstrate the performance tradeoffs under different circumstances.

The contributions of this article are as follows: We first introduce the system model and theory of adaptive threshold energy detection. We then cover the specifics of the development of the energy detector in hardware and software. Next, we describe the setup of the experiments, including the details of the test transmitter and waveforms. Experimental results are presented and discussed. Finally, we issue concluding remarks and provide some ideas for future work.

II. BACKGROUND

A. System Model

Energy detectors generally fall into one of two different categories. The first is the FFT-based energy detector. An FFT or periodogram energy detector uses a type of short-time Fourier transform (STFT), which repeatedly computes the FFT of a signal using fixed-length, often overlapping sample windows [25]. The squared magnitude values of the repeated FFTs yield an estimate of the total energy in a number of adjacent frequency bands over some duration. The sampling rate determines the total bandwidth, and the length of the periodogram determines the size and number of sub-bands, or channels.

The energy detector we developed belongs to the other category, which is the single-channel prefilter-type detector. The received signal is passed through an analog band-pass filter (BPF), which removes out-of-band components. The filtered signal is squared and averaged over time to yield the detection statistic. For detection over multiple channels, either type of energy detector would suffice: The FFT coefficients can be used to estimate channel energy, or multiple band-pass filters corresponding to different channel frequency ranges can be employed to distinguish the distribution of the total signal energy across each individual channel.

We assume the perspective of a cognitive user (CU) that is sensing a channel for the presence of a primary user (PU). The signal model begins with the received signal that has been downconverted, filtered, and sampled. The binary hypothesis test is formulated around the input signal $y(t)$ as

$$\begin{aligned} H_0 : & y(t) = n(t) \\ H_1 : & y(t) = s(t) + n(t) \end{aligned} \quad (1)$$

where $s(t)$ is the PU signal and $n(t)$ is the additive white Gaussian noise (AWGN). Instantaneous received energy is defined as $Y(t) = |y(t)|^2$. The test statistic for energy detection is calculated using a moving average (MA) window over the energy samples: to calculate the detection statistic $D(t)$ for a given sample time t , the N most recent energy samples are averaged using

$$D(t) = \frac{1}{N} \sum_{i=0}^{N-1} Y(t-i). \quad (2)$$

The threshold λ is defined such that for a given sample time t , if $D(t) < \lambda$ the detector yields H_0 and if $D(t) > \lambda$ the detector yields H_1 . Important performance metrics are probability of detection (P_D) and probability of false alarm (P_{FA}), which are defined as

$$P_D = P(D(t) > \lambda | H_1) \quad (3)$$

$$P_{FA} = P(D(t) > \lambda | H_0). \quad (4)$$

B. Adaptive Threshold Energy Detection

The approach used to calculate the adaptive threshold is inspired by previous theoretical work [11], [12]. This approach is borrowed from an image processing technique

called image binarization. We selected this method because of its simplicity and robustness. No explicit estimate of noise variance is needed to calculate the threshold, and because the threshold calculation is iterative and continuous, the energy detector can handle changing noise and signal levels.

For a single-channel energy detector, the adaptive threshold is calculated using

$$\lambda = \mu_D(t) + k\sigma_D(t) \quad (5)$$

where $\mu_D(t)$ is the moving average of the detection statistic $D(t)$ defined as

$$\mu_D(t) = \frac{1}{L} \sum_{i=0}^{L-1} D(t-i) \quad (6)$$

and $\sigma_D(t)$ is the standard deviation of $D(t)$ defined as

$$\sigma_D(t) = \sqrt{\frac{1}{L} \sum_{i=0}^{L-1} (D(t-i) - \mu_D(t))^2} \quad (7)$$

where L is the length of the sliding window and k , $0 \leq k \leq 1$ is the standard deviation coefficient. The mean and standard deviation are calculated using the L most recent values of the detection statistic $D(t)$. Because the MA process accumulates many samples, the adaptive threshold can be slow to track changes in signal mean and standard deviation; this effect is more pronounced with larger values of L . Rapid fluctuations in signal level are not immediately reflected in the threshold.

III. IMPLEMENTATION DETAILS

A. Software Defined Radio - Hardware and Software

For the experimental setup, the real-time adaptive threshold energy detector is constructed with an SDR connected to a PC. The SDR is an Ettus Universal Software Radio Peripheral (USRP) X310. The USRP hosts two internal slots, each of which houses a daughterboard. By adopting this design, the USRP is user-configurable based on the selection of daughterboards. The daughterboards are the physical radio front-ends; the USRP itself contains the FPGA, ADCs, and DACs, and power and interfacing hardware.

The USRP X310 motherboard features a Xilinx Kintex-7 FPGA. Each daughterboard slot has two RF ports with SMA connectors: one Rx and one Tx/Rx. The Rx signal path has a 14-b ADC that is capable of sampling at up to 200 MS/s. For our setup, we use the TwinRX daughterboard in the X310; each TwinRX board has two Rx channels that each have a frequency range of 10 MHz–6 GHz and a channel bandwidth of up to 80 MHz each. We use only one of the two Rx channels for our experiments, and the detector's actual bandwidth is determined in software.

An important consideration for spectrum sensing, particularly wideband sensing, is the choice of interface and the maximum supported data rate. SDRs output sample streams either directly to a computer or to a network, so they may have more than one interface method. In the case of USRPs, the models may be equipped with a USB interface

for local connectivity, or an Ethernet port that can support networking or can be used to directly connect to a computer. Ethernet connections typically support 1 Gb/s, although the more expensive models may have maximum data rates of 10 Gb/s. According to the data sheet for the USRP X310, the ADC has a maximum sample rate of 200 MS/s. Assuming each sample is a 32-b complex value, which contains a 32-b I-value and a 32-b Q value, the SDR's interface would need to support a data rate of at least 12.8 Gb/s, assuming zero overhead; in practice, the minimum rate would be higher. Therefore, the interface between the SDR and the host computer could potentially be a limiting factor on the maximum realizable sample rate.

The X310 has two SFP+ ports that can support standard Gigabit Ethernet as well as 10 Gigabit Ethernet. The X310 also has a PCIe x4 interface that can support data rates up to 200 MS/s with the lowest latency via a connection to a computer equipped with a special PCIe Express card. For our experiments, we used a PC running Windows 10 to connect to the USRP X310 via the PCIe Express interface.

In order to measure the performance of the energy detector, a second SDR was set up to transmit various test signals. The second SDR is an analog devices (AD) ADALM-PLUTO, or PlutoSDR, which is a low-cost SDR built around the AD9363 transceiver chip and a Xilinx Zynq-7000 family FPGA. The AD9363 features one Tx and one Rx channel with a frequency range of 325 MHz to 3.8 GHz and up to 20 MHz bandwidth. The PlutoSDR has a 12-b ADC and DAC and can operate in half or full duplex mode. A laptop running Windows 10 is used to interface with the PlutoSDR over USB 2.0. The use of USB 2.0 restricts the maximum capture rate, but the sampling rate was kept relatively low during our experiments, so it was not an issue.

We used GNU Radio (GR) for test signal generation on the Tx side and real-time energy detection and sample output on the Rx side. GNU Radio is a well-known open-source toolkit for implementing software radios, which contains a number of predesigned signal processing blocks written in C. Additionally, the software comes with the GNU radio companion (GRC), which is a graphical tool for building signal flowgraphs in a manner similar to LabVIEW or Simulink. Once the flowgraph is complete, GRC can build the flowgraph into a Python file, which can either be run from the Companion or executed directly from a command line. The GR libraries and hardware drivers for the USRP enable GR to import a sample stream from each Rx channel of the USRP in the form of a signal source block. The user sets various parameters of the USRP source, such as center frequency, channel bandwidth, sampling rate, and gain.

The SDR mixes the received signal down to baseband according to the desired center frequency f_0 . GR then imports the baseband signal as a stream of complex samples, each containing a 32-b I value and a 32-b Q value. We pass the sample stream through a low-pass filter (LPF) block to refine the signal so that it contains only the components within the channel bandwidth. The filtered baseband samples constitute the raw signal $y(t)$ for a single channel. Since

we are only interested in signal energy, the complex values are passed through the *Complex to Mag^2* block to yield real-valued signal energy samples $Y(t)$. The energy samples are then input to the *Moving Average* block to give us the detection statistic $D(t)$. Samples of $D(t)$ are accumulated in order to calculate the evolving mean $\mu_D(t)$ and standard deviation $\sigma_D(t)$. Because the operations needed to implement (7) are not present in the standard GR block library, the $D(t)$ stream is routed to the *Stream to Vector* block, which is periodically sampled by the *Probe Signal Vector* block. We modified the probe function by inserting our own custom Python code to calculate the mean, standard deviation, and threshold values. Each new value of $D(t)$ is compared against the adaptive threshold given by (5) using the *Threshold* block, the output of which is the detector's decision on the channel occupancy state. Fig. 1 is a block diagram of the basic adaptive threshold energy detector flowgraph in GRC.

Collected samples of the filtered baseband signal $y(t)$ and the corresponding energy detector output are saved as binary data streams using the *File Sink* block. On the transmit side, the PlutoSDR saves the transmitter state as a binary value. MATLAB is used to import the saved samples for analysis. Fig. 2 is an example of saved data from the USRP and the PlutoSDR that has been imported and time synchronized in MATLAB. MATLAB can also be used to interface with the USRP and import received samples in real time.

B. Practical Considerations

From the basic prototype shown in Fig. 1, the energy detector can be further adapted for specific regimes. Once the blocks are in place, the next step is to ensure that they are properly configured for the desired application. The *USRP Source* block is the GR interface with the hardware and it allows the programmer to supply the operating parameters. The center frequency of the *USRP Source* block and the bandwidth of the LPF define the channel. Typically, the sampling rate should be set to at least twice the channel bandwidth to conform to the Nyquist–Shannon sampling theorem. However, the objective is not perfect signal reconstruction; the energy detector is simply measuring the total energy magnitude of the band defined by the center frequency and the LPF. If the band is to be subdivided into multiple smaller channels through the use of FFT-based energy detection, then the detector must adhere to the Nyquist rate in order to avoid aliasing. For the single-channel prefilter-type detector used in this study, the Nyquist rate is more of a soft limit.

Operational sampling rate for real-time energy detection is going to depend on application requirements and hardware capabilities. Another consideration is whether or not the samples will be stored for future analysis. With higher sampling rates, a GR binary data file containing the saved samples will grow to be very large very fast, so physical storage will need to be sufficient. For example, assume a sampling rate of 10 MS/s of 32-b complex samples, so

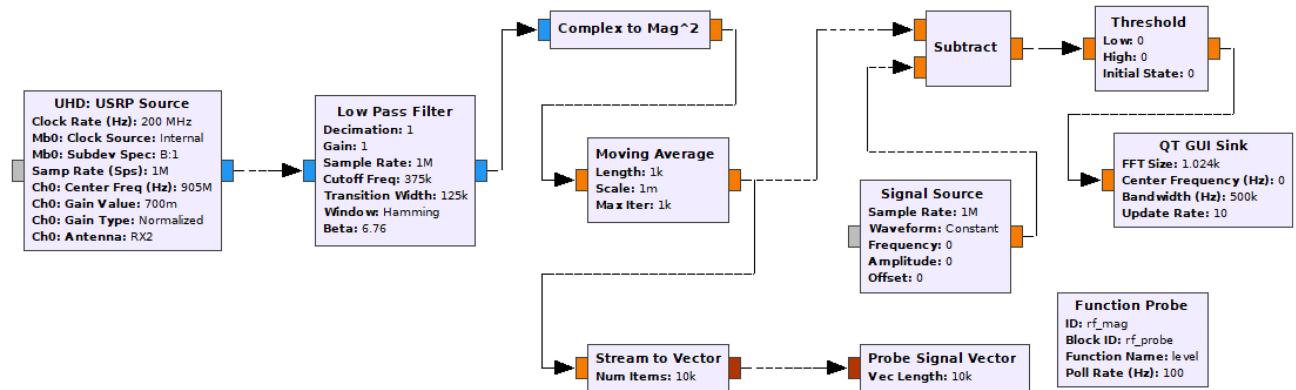


Fig. 1. GNU Radio flowgraph for basic adaptive threshold energy detector.

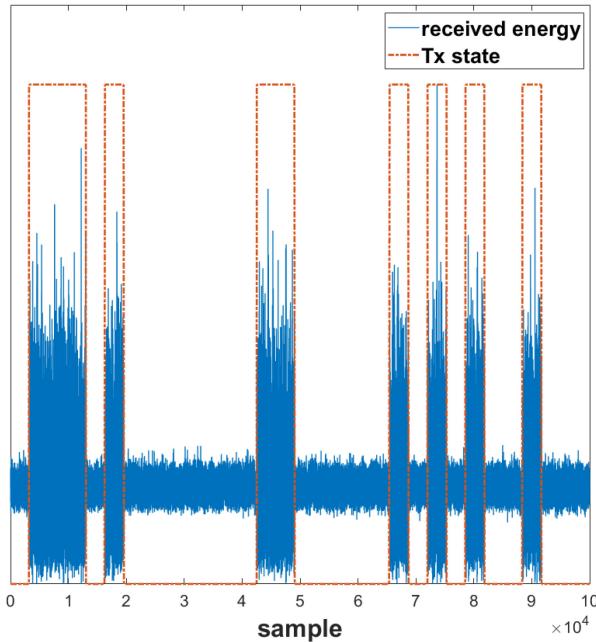


Fig. 2. Transmit signal ON/OFF state and corresponding received energy samples.

$2 \times 32 = 64$ b/sample; $(10 \times 10^6)(64) \div 8 = 80 \times 10^6$ B/s or 80 MB/s. Analysis and postprocessing of such a large quantity of data is computationally intensive. A feasible strategy would be to sample at the higher rate to perform the energy detection in real-time and then downsample before storing the samples.

Among the practical considerations we encountered, of particular importance are the choice of values for N and L , or the lengths of the moving average windows for the detection statistic $D(t)$ and the calculation of $\mu_D(t)$ and $\sigma_D(t)$. The value of L is one or more orders of magnitude greater than N , and the choice for L depends on the mean lengths of the transmit signal's ON and OFF periods. For general application, a larger value for L is sufficient, perhaps on the order of the sampling rate so that the mean and standard deviation estimates are recalculated once per second.

To understand the purpose of the moving average window used to calculate the detection statistic $D(t)$, we must first consider the desired output of an energy detector and its role in the cognitive cycle. An energy detector should provide an estimate of whether or not a PU is transmitting on a channel during a particular sampling interval, and it does so using only sampled energy magnitude over time. Within an individual transmission ON period, the receiver will sample energy at values ranging from the minimum received energy to the maximum received energy, which is evident in Fig. 2. If single energy samples were used for the detection statistic, the detector output would show rapid oscillatory behavior during ON periods, indicating that the intermittent transmitter is switching transmission states much faster than it actually is. It is much more useful for a CR to treat the entire period as an ON state for the purposes of spectrum occupancy modeling and opportunistic spectrum access. A moving average window is an integrator; it attenuates higher frequencies, thus alleviating the rapid fluctuations in received energy. The drawback is that the ideal instantaneous transition between states is stretched into a ramp waveform, which delays the detection of state transitions depending on the threshold level. This feature is shown in Fig. 3. The longer the moving average window, the more pronounced this distortion is. Selection of N also depends mainly on the mean ON and OFF times of the transmitter: More state transitions per time interval leads to more erroneous detections.

IV. EXPERIMENT SETUP AND RESULTS

Once the proper functionality of the USRP energy detector was confirmed, we designed a benchmarking procedure to establish performance metrics. Several test signals were developed using the PlutoSDR as a PU transmitting intermittently. For the real-time energy detection experiments, we chose a center frequency f_0 of 905 MHz. The portion of the spectrum from 902 to 928 MHz is an unlicensed band reserved for industrial, scientific, and medical (ISM) applications in the United States, making it ideal for low-power wireless testing. The USRP and the PlutoSDR were stationed in an indoor lab with a distance d of roughly 4 m

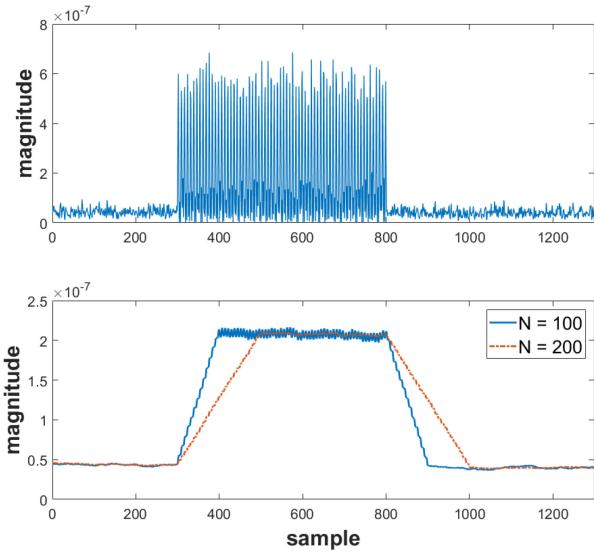


Fig. 3. Top: Instantaneous received energy over time $Y(t)$; bottom: moving averaged energy $D(t)$ using two different window lengths N .

apart. Both SDRs were equipped with basic omnidirectional dipole antennas. The maximum transmission power of the PlutoSDR was kept low to avoid unnecessary interference.

In order to measure P_D and P_{FA} , the pattern of transmitter states needs to be known. By time synchronizing the energy detector output with the corresponding PU ON/OFF states, P_D and P_{FA} can be measured by evaluating each detector output sample as either a correct detection, a missed detection, or a false alarm. The PlutoSDR transmit waveform ON/OFF state is determined by an envelope waveform, and the Tx state during the course of an experiment is saved as a stream of binary-valued samples using the *File Sink* block. MATLAB is used to import the energy detector output samples from the USRP and the Tx state output samples from the PlutoSDR. The imported samples are then aligned in time and compared on a per-sample basis. It is important to make sure that the two imported sample streams represent the same time scale; in other words, both the transmitter and receiver should export their respective sample streams at identical sampling rates, or one of the streams can be decimated to match the rate of the other upon import.

Test data SNR is determined empirically after collection. Stretches of samples containing noise only and sections of samples containing signal plus noise are manually identified and used as references for the two channel states. The variance is estimated for each of the two channel states; the OFF state variance is defined as $\sigma_{y,OFF}^2 = \sigma_n^2$ and the ON state variance is defined as $\sigma_{y,ON}^2 = \sigma_s^2 + \sigma_n^2$. Assuming $s(t)$ and $n(t)$ are uncorrelated, SNR is calculated using

$$\text{SNR(dB)} = 10 \cdot \log_{10} \left(\frac{\sigma_{y,ON}^2}{\sigma_{y,OFF}^2} - 1 \right). \quad (8)$$

By fine-tuning the attenuation level of the PlutoSDR and the USRP gain amount, the average SNR can be reasonably confined near a desired value, but it must always be confirmed during processing. Uncertainty in spectrum

TABLE I
Square Wave Experimental Parameters

Parameter	Value
f_0	905 MHz
Sampling rate	100 kHz
Bandwidth	50 kHz
L	10,000 samples
Tx attenuation	64 dB

conditions, such as spurious noise levels and transient interference, confer a degree of uncertainty to SNR estimates.

A. Square Wave Transmit Waveform

For initial testing of the energy detector, we used a continuous sine wave that is made intermittent by applying a square wave envelope. This is the simplest waveform to test an energy detector: it has periodic state changes and therefore a deterministic behavior. Because of this, the Tx state samples do not need to be saved during the experiment and imported and aligned in postprocessing. The Tx envelope can simply be recreated in MATLAB using the square wave parameters.

For the experiments, square wave frequencies of 100 and 500 Hz were used, both with a 50% duty cycle. Initially, USRP sampling rate was set at 100 kHz and channel bandwidth at 50 kHz, so one period of the square wave envelope was 1000 samples. The value of L , or the number of sample used to calculate the receive signal mean and standard deviation, was set to 10 000 samples. Because the envelope of the received energy is periodic, taking the mean over one period should be sufficient.

The experiments were designed to estimate P_D and P_{FA} with respect to two primary user-variable parameters: Detection statistic moving average window length N and standard deviation coefficient k . To examine behavior as a function of N , we cataloged detection output for different window lengths. Based on behavior observed during initial testing, k was set to 0.1. Table I summarizes the relevant parameters. Experiments were conducted for SNR levels of 10, 15, and 20 dB. Figs. 4 and 5 show the calculated P_D and P_{FA} , respectively, from these experiments.

Detector performance falls when the square wave frequency is changed from 100 to 500 Hz. Also, the detector performance seems to be more severely influenced by moving average window length for the 500 Hz case. This is a reasonable result when we consider the effect of the moving average window. The detection statistic features distortion compared to the instantaneous signal energy, distortion that turns the square-shaped waveform into a trapezoid due to the stretching at the state transitions. These distorted state transitions are where the highest incidence of detection error occurs. In the 500-Hz case, the detector sees a greater number of state transitions per second.

We then conducted experiments with the same test signal scenarios and fixed window length. Window length was fixed at $N = 40$ samples for the 100 Hz square wave and $N = 20$ for the 500 Hz square wave cases. Detector output was observed using different values of k in calculating the

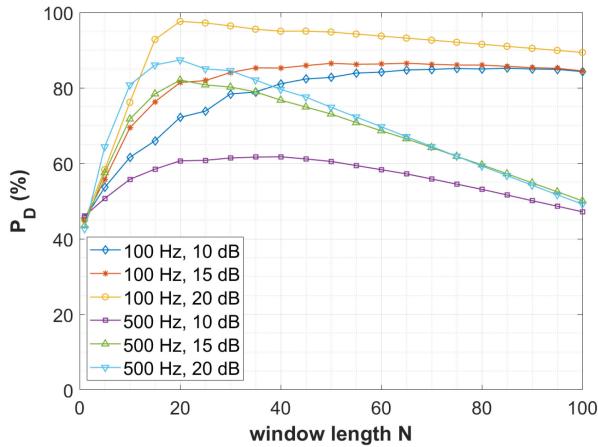


Fig. 4. Probability of detection versus moving average window length N for square wave test signal; adaptive threshold detection with parameter $k = 0.1$.

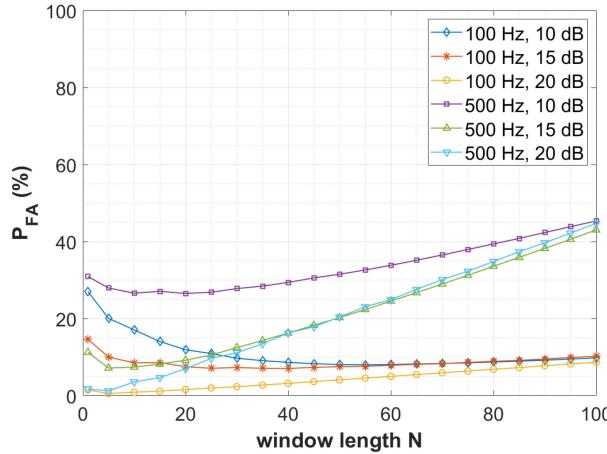


Fig. 5. Probability of false alarm versus moving average window length N for square wave test signal; adaptive threshold detection with parameter $k = 0.1$.

detection threshold. Figs. 6 and 7 show resulting P_D and P_{FA} values from the variable k experiments. Our experiments indicate that smaller values of k yield better detection performance, which means that the adaptive threshold value is mostly comprised of the mean of the received energy. A fixed value of $k = 0.1$ was used for the variable window length experiments because it results in a detector threshold that gives the best balance between P_D and P_{FA} in the experimental scenarios. According to [11], optimum k value should increase with increasing SNR, but this relationship is not reflected in the experimental results, at least not for the range of SNR values observed. We found the range of SNR values from 10 to 20 dB to be the most useful in terms of evaluating the performance of the energy detector: Below 10 dB, the performance is poor and unreliable, while values above 20 dB typically do not result in significant performance gains. Based on our findings, simply using the mean of the received energy taken over a large number of samples would be a simple yet viable strategy for energy detection.

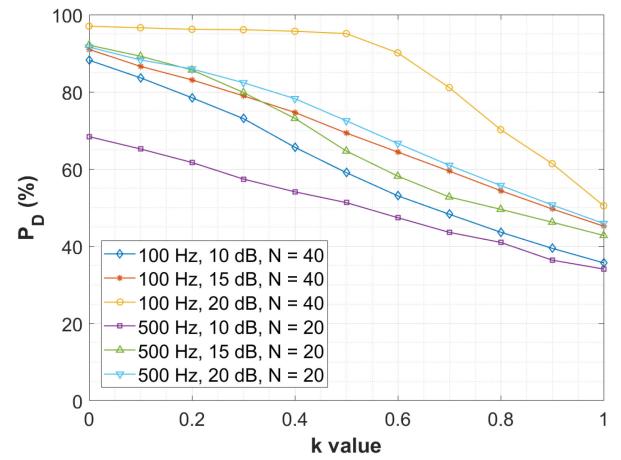


Fig. 6. Probability of detection versus k for square wave test signal; adaptive threshold detection.

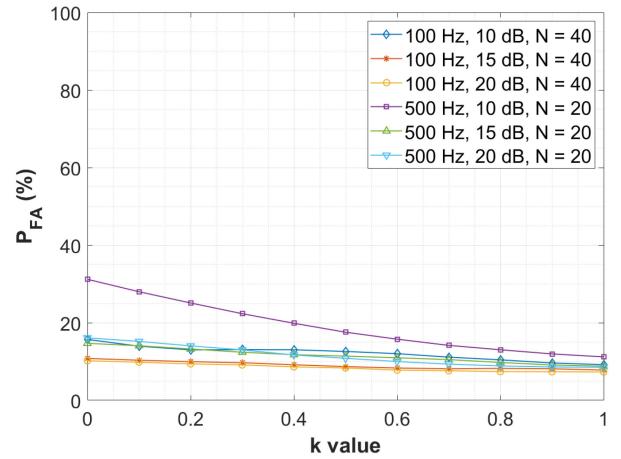


Fig. 7. Probability of false alarm versus k for square wave test signal; adaptive threshold detection.

B. Fixed Threshold Comparison

In order to place the results from the adaptive threshold detection experiments in context, we performed fixed threshold energy detection on the samples received from the square wave test signals. By doing so, we have a point of comparison from which we can further evaluate the adaptive threshold detector performance. Our first approach was to implement a basic CFAR detector with the USRP using a GR flowgraph. We attempted to use different equations for calculating the fixed threshold λ given a desired P_{FA} and an SNR value: [6, (9)] and [7, (7)]. However, across all of our experimental data, the resulting threshold values were always outside of the range of values of the detection statistic, rendering the approach nonviable. Often the calculated threshold was more than an order of magnitude outside of the range of values.

Instead, we decided to obtain the performance metrics empirically in MATLAB. The threshold level was adjusted incrementally until a desired P_{FA} was met. Note that this is not a viable approach to real-time energy detection because the transmitter state is needed to evaluate P_D and P_{FA} . This experiment is designed to present the theoretically optimal

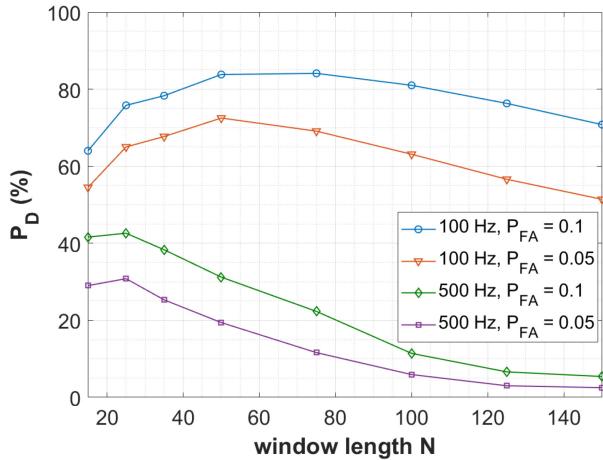


Fig. 8. Probability of detection versus moving average window length N for square wave test signal at 10 dB SNR; fixed threshold detection using constant P_{FA} .

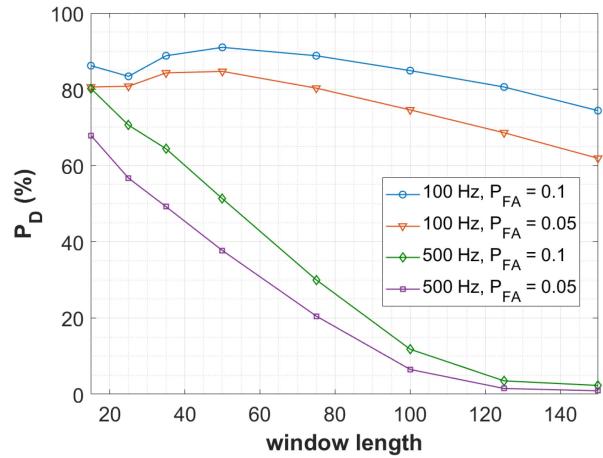


Fig. 9. Probability of detection versus moving average window length N for square wave test signal at 15 dB SNR; fixed threshold detection using constant P_{FA} .

performance of the general purpose USRP energy detector with an omnidirectional antenna. Parameters were kept the same as the adaptive threshold experiments, except for the range of values for N . Sampling rate is 100 kHz, channel bandwidth is 50 kHz, and the transmitted test signals were square waves at 100 and 500 Hz. Values of 0.1 and 0.05 were used for the desired P_{FA} .

Figs. 8, 9, and 10 show calculated P_D corresponding to fixed values of P_{FA} for an SNR of 10, 15, and 20 dB, respectively. A range of moving average window lengths were used in calculating the detection statistic. As is the case with the adaptive threshold results, the energy detection shows much better performance with the 100 Hz square wave compared to the 500 Hz square wave case. P_D drops dramatically with increasing N , which is more pronounced the higher the SNR. This is because the threshold levels needed to sustain the desired values of P_{FA} are so high that most possible detections are missed.

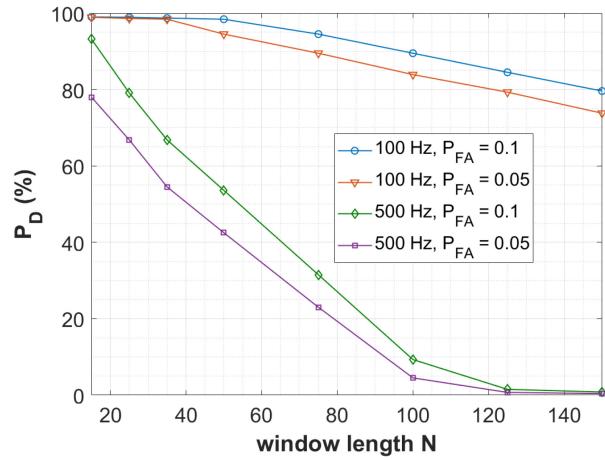


Fig. 10. Probability of detection versus moving average window length N for square wave test signal at 20 dB SNR; fixed threshold detection using constant P_{FA} .

C. OFDM Transmit Waveform

Once we established a baseline performance estimate of the adaptive threshold energy detector with the use of periodic transmit pulses, we wanted to test the detector under more realistic channel occupancy conditions. We designed an intermittent OFDM communications waveform for testing purposes. Fig. 11 is the GR flowgraph of the core of the TX signal generation. The *Random Source* block generates a pseudorandom sequence of bits that are converted into OFDM symbols using the *OFDM Mod* block. The *OFDM Mod* block handles all of the steps involved in creating a baseband OFDM signal; because we were not designing with the intent to mimic a specific technology, we used the default parameters for the OFDM block.

The *Signal Source* block generates the envelope that turns the OFDM signal on and off to simulate a PU transmitting packets in bursts. We used a customized *Python Block* to generate an alternating series of on and off periods with lengths controlled by two different Poisson processes. The custom *Python Block* controls the amplitude of the *Signal Source* block, pushing it back and forth between 0 and 1 in response to the off and on periods. We wanted to demonstrate practical energy detection in a more realistic scenario with a stochastic transmitter state sequence. Based on the mean values of the Poisson point processes, the channel occupancy rate is about 0.36.

For this new set of experiments, the sampling rate is increased to 1 MHz on the USRP, and the channel bandwidth is increased to 500 kHz to ensure that the entire signal is captured. The detector outputs are downsampled to 100 kHz before they are saved to binary data files, so the time scales are the same when the data is imported in MATLAB. The values of N used are larger to compensate for the higher sampling rate; additionally, L is increased to 1 000 000 samples. Table II summarizes the experiment parameters. Since the channel occupancy pattern is stochastic, the transmitter state sequence needs to be recorded so that P_D and P_{FA} can be calculated in MATLAB. Because the

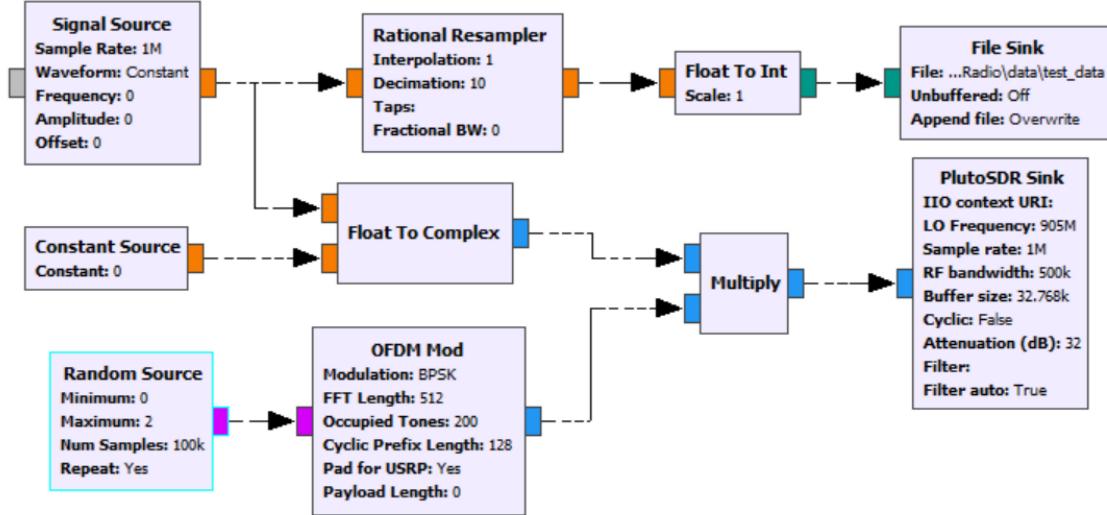


Fig. 11. GNU Radio flowgraph for OFDM transmit test signal.

detector output from the USRP and the transmit signal state from the PlutoSDR are recorded at the same sampling rate, once they are synchronized in time they can be compared sample-by-sample.

Synchronizing the recorded transmit state stream and the corresponding detection statistic and detector output streams can be challenging because they are recorded on two different systems from two different SDRs. The USRP takes significantly longer to start running the GR code once the command to execute it has been entered from the command prompt, so even if the execute command is given to both SDRs simultaneously, the PlutoSDR transmitter code will begin running at least several seconds before the USRP energy detector code starts up. The *File Sink* block starts recording when the GR code begins and records a continuous stream until the code is halted, so there will be a time offset between the two sample streams. Given typical sampling rates, there is frequently a discrepancy in alignment between the two streams on the order of millions of samples. In our experiments, we used the following approach: start up the USRP first, and once it begins running the GR code, start up the PlutoSDR. Captured samples are saved at a sampling rate of 100 kHz, which typically yields an offset between 500 kSamples and 800 kSamples. The initial samples from the USRP are entirely noise, so we parse through them until the first received OFDM signal samples arrive. The alignment phase can be automated by computing the cross correlation between the detection statistic $D(t)$ and the transmit state samples, but this procedure can be time consuming depending on available computer hardware. We found that it was faster to align manually by visually inspecting the two sample streams while advancing the USRP samples until the two streams were correlated. For the combination of sampling rates and transmit state transition rates used, synchronization accuracy was not overly critical. From the point of maximum correlation between the received samples and the transmit state samples, an offset of 5 samples in either direction

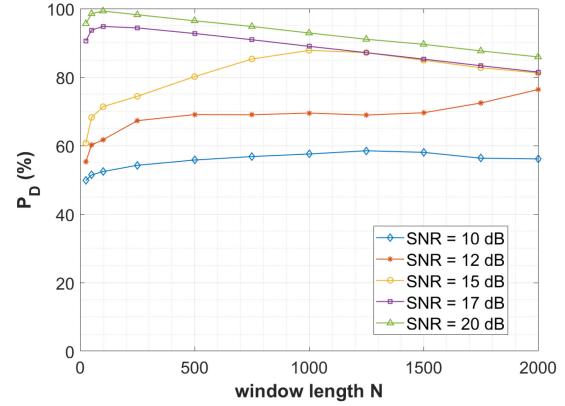


Fig. 12. Probability of detection versus moving average window length N for OFDM test signal using adaptive threshold detection with parameter $k = 0.1$.

would typically result in less than one percent difference in the detection accuracy. An alternative strategy would be to use the *File Meta Sink* block to record the metadata of the samples and then align the Tx and Rx streams using the sample timestamps. This would rely on the transmit and receive systems having synchronized system clocks. It may be possible to run the transmit and receive SDRs from the same computer, which would ensure reliable timestamps.

We ran experiments with the adaptive threshold energy detector using the intermittent OFDM transmit waveform. Sample streams were captured and aligned using the aforementioned procedure. Similar to the initial square wave envelope experiments, we are interested in detector performance as a function of moving average window length N and standard deviation coefficient k . Figs. 12 and 13 show P_D and P_{FA} , respectively, versus N ; k is fixed at 0.1. The optimum window length appears to shrink with increasing SNR. Figs. 14 and 15 show P_D and P_{FA} versus k ; the optimal value of N is varied according to SNR. Performance of the adaptive threshold energy detector is moderately good:

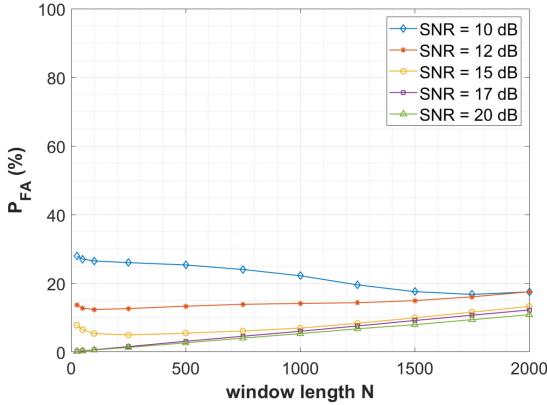


Fig. 13. Probability of false alarm versus moving average window length N for OFDM test signal using adaptive threshold detection with parameter $k = 0.1$.

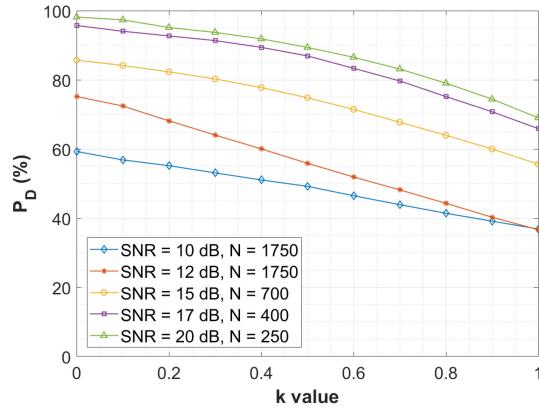


Fig. 14. Probability of detection versus k for OFDM test signal using adaptive threshold.

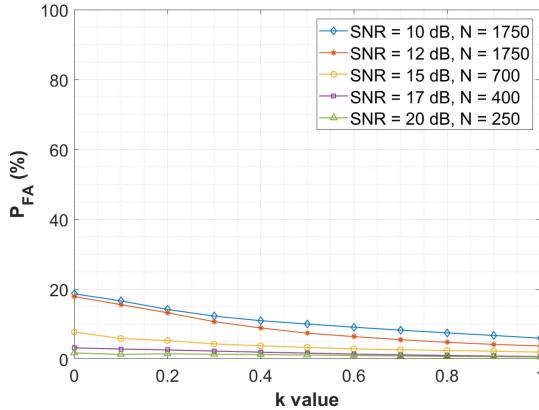


Fig. 15. Probability of false alarm versus k for OFDM test signal using adaptive threshold.

At $\text{SNR} = 15 \text{ dB}$, P_D is around 85% and P_{FA} is below 6%. The P_{FA} trend at higher SNR values is monotonically increasing with window length, whereas it starts to reverse as the SNR drops to 10 dB. Energy detector performance becomes unreliable at low SNR levels; the high P_{fa} at 10 dB using shorter window lengths is likely due to the position of the threshold relative to the noise floor. Noise spikes,

TABLE II
OFDM Experimental Parameters

Parameter	Value
f_0	905 MHz
Sampling rate	1 MHz
Bandwidth	500 kHz
L	1,000,000 samples
Tx attenuation	32 dB

which are attenuated at longer window lengths, breach the threshold and are recorded as detections.

D. Moving Average Window Shape

In our review of previous work, we noticed that similar energy detection strategies also use a simple moving average window for the detection statistic. Through our experiments, we have shown that the length of the moving average window used to calculate the detection statistic has a massive influence on the detection fidelity, which is expected. A question arose as to whether changing the shape of the moving average window would significantly affect performance. We found no reference to the use of anything other than a rectangular moving average window for computing the detection statistic $D(t)$. We designed an experiment in order to lend empirical data to further refine practical energy detection.

A simple moving average uses a rectangular window, where all samples $Y(t - N + 1), \dots, Y(t - 1), Y(t)$ are given equal weights. The detection statistic is redefined as

$$D(t) = \frac{1}{N} \sum_{i=0}^{N-1} w(i)Y(t - i). \quad (9)$$

For our experiments, three new windows are used: a sinusoidal window, an exponential window, and a Gaussian window. The sine window is given by

$$w_{sin}(t) = \cos\left(\frac{\pi t}{2N}\right) \quad (10)$$

the exponential window is given by

$$w_{exp}(t) = e^{-2t/N} \quad (11)$$

and the Gaussian window is given by

$$w_{gaus}(t) = e^{-t^2/2\sigma^2}. \quad (12)$$

Gaussian smoothing is a technique borrowed from image processing. Since the smoothing is being applied in real time, causality constraints mean that we are restricted to using the “half-Gaussian” given by (12). For our experiments, we used a window with $\sigma = N/3$.

The new windowing functions apply different weights to the samples used in the moving average. The most recent sample $Y(t)$ is given the maximum weight of 1. The sample weights decrease monotonically so that the oldest sample $Y(t - N + 1)$ is given the smallest weight. All other parameters are unchanged from Tables I and II. Figs. 16–18 show the P_D versus window length comparison between the four different window shapes (rectangular,

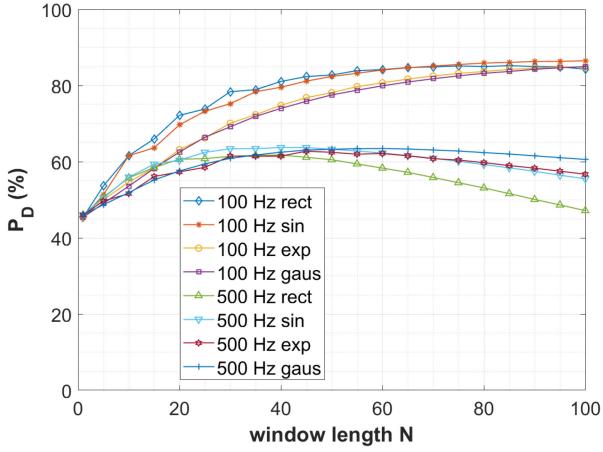


Fig. 16. Probability of detection versus moving average window length N for square wave test signals with $\text{SNR} = 10 \text{ dB}$ using adaptive threshold detection with parameter $k = 0.1$ and different moving average window shapes.

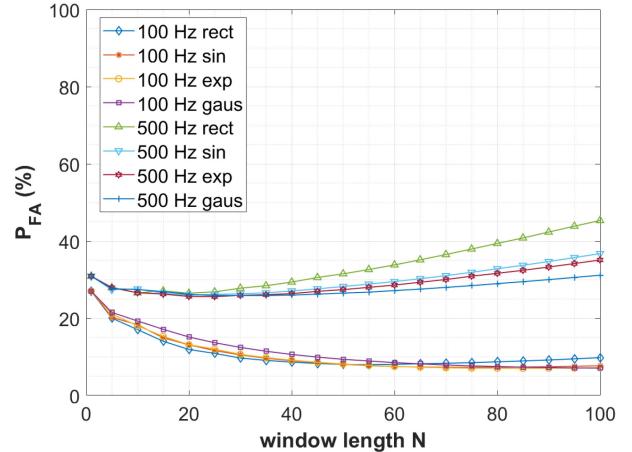


Fig. 19. Probability of false alarm versus moving average window length N for square wave test signals with $\text{SNR} = 10 \text{ dB}$ using adaptive threshold detection with parameter $k = 0.1$ and different moving average window shapes.

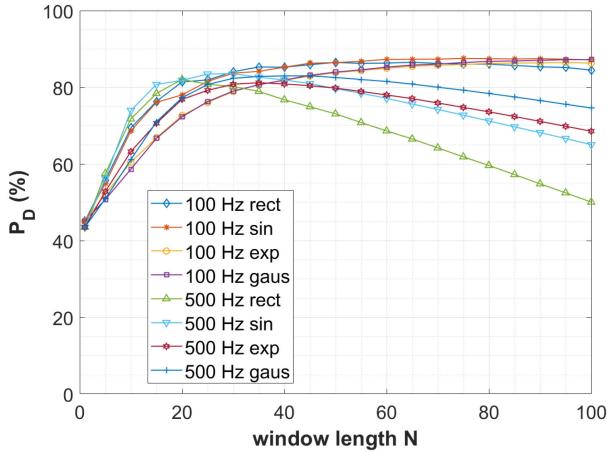


Fig. 17. Probability of detection versus moving average window length N for square wave test signals with $\text{SNR} = 15 \text{ dB}$ using adaptive threshold detection with parameter $k = 0.1$ and different moving average window shapes.

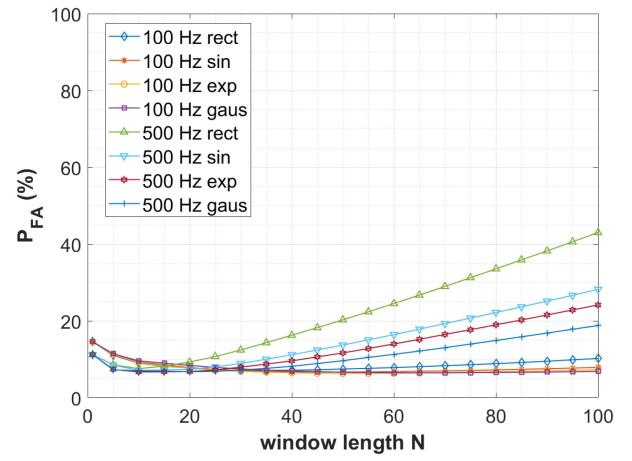


Fig. 20. Probability of false alarm versus moving average window length N for square wave test signals with $\text{SNR} = 15 \text{ dB}$ using adaptive threshold detection with parameter $k = 0.1$ and different moving average window shapes.

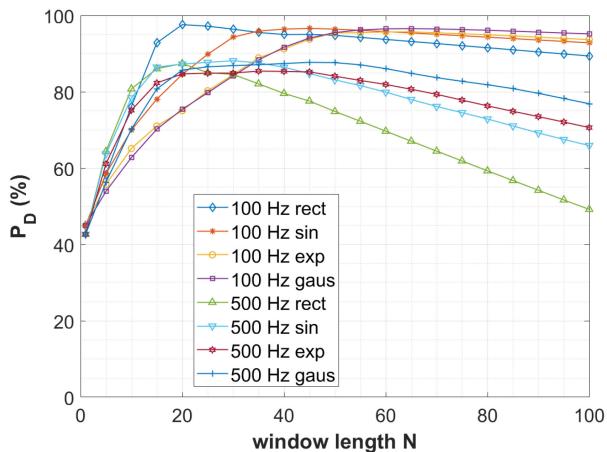


Fig. 18. Probability of detection versus moving average window length N for square wave test signals with $\text{SNR} = 20 \text{ dB}$ using adaptive threshold detection with parameter $k = 0.1$ and different moving average window shapes.

sine, exponential, and Gaussian), and Figs. 19–21 are the P_{FA} versus window length comparisons for the square wave transmit waveforms. The effects of changing the window shape are generally less pronounced for the shorter window lengths, but using an alternative window shape for the moving average smoothing offers a noticeable increase in performance for the longer window lengths. This is especially important for the 500 Hz square wave test signal at 15 and 20 dB SNR, which saw performance drop significantly for the longer window lengths. In particular, the Gaussian window delivers the best P_D and P_{FA} performance for window lengths above about 40 samples. When $N = 100$ for the 500 Hz square wave at $\text{SNR} = 15$ and 20 dB, P_D sees improvements of around a 25% increase, and P_{FA} sees a decrease of around 25%.

Figs. 22–24 show P_D versus window length comparison for the different window shapes for the OFDM test configuration, and Figs. 25–27 show the corresponding P_{FA} comparisons. The alternative window shapes show similar

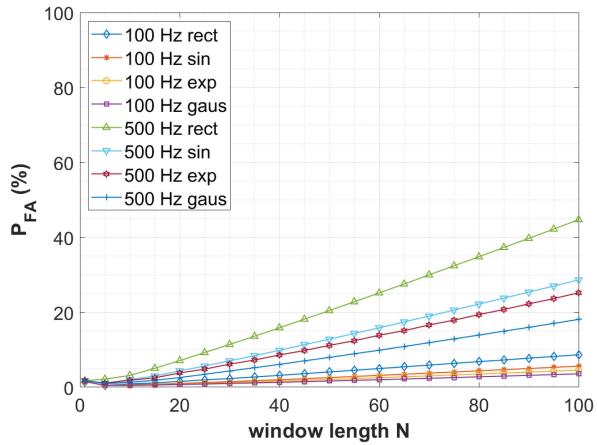


Fig. 21. Probability of false alarm versus moving average window length N for square wave test signals with SNR = 20 dB using adaptive threshold detection with parameter $k = 0.1$ and different moving average window shapes.

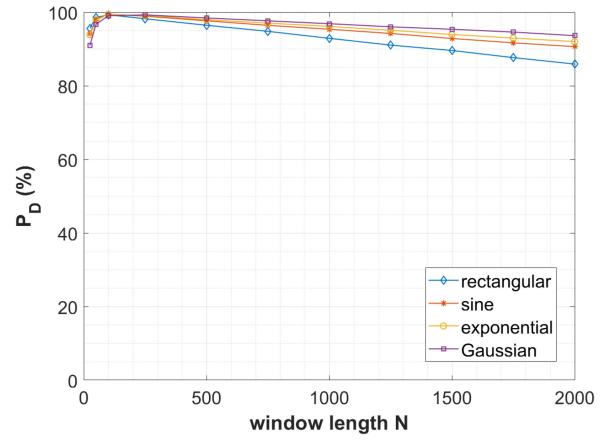


Fig. 24. Probability of detection versus moving average window length N for OFDM test signal with SNR = 20 dB using adaptive threshold detection with parameter $k = 0.1$ and different moving average window shapes.

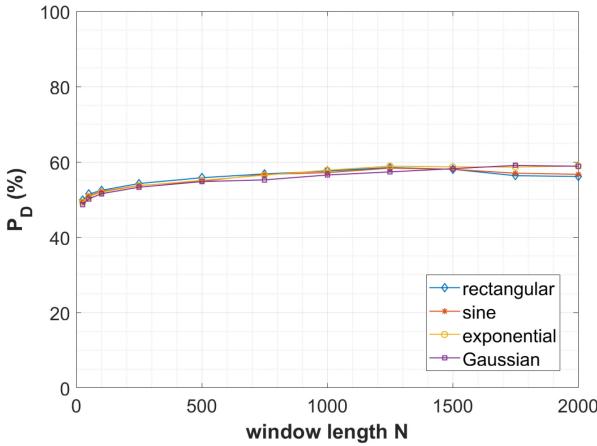


Fig. 22. Probability of detection versus moving average window length N for OFDM test signal with SNR = 10 dB using adaptive threshold detection with parameter $k = 0.1$ and different moving average window shapes.

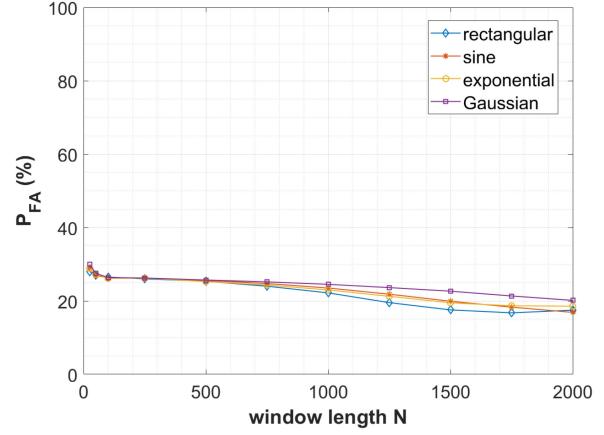


Fig. 25. Probability of false alarm versus moving average window length N for OFDM test signal with SNR = 10 dB using adaptive threshold detection with parameter $k = 0.1$ and different moving average window shapes.

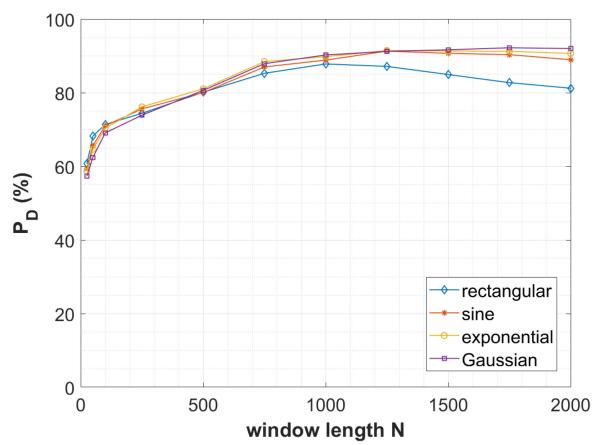


Fig. 23. Probability of detection versus moving average window length N for OFDM test signal with SNR = 15 dB using adaptive threshold detection with parameter $k = 0.1$ and different moving average window shapes.

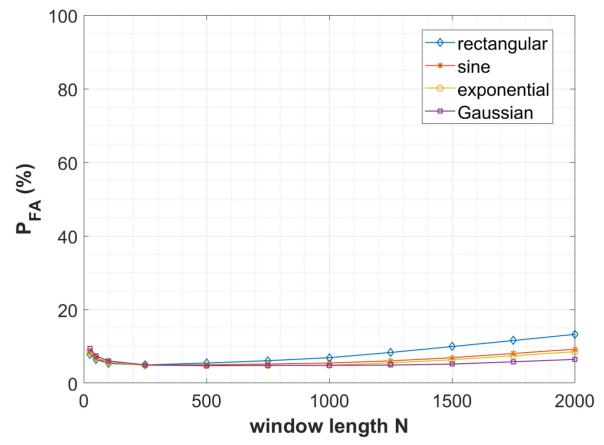


Fig. 26. Probability of false alarm versus moving average window length N for OFDM test signal with SNR = 15 dB using adaptive threshold detection with parameter $k = 0.1$ and different moving average window shapes.

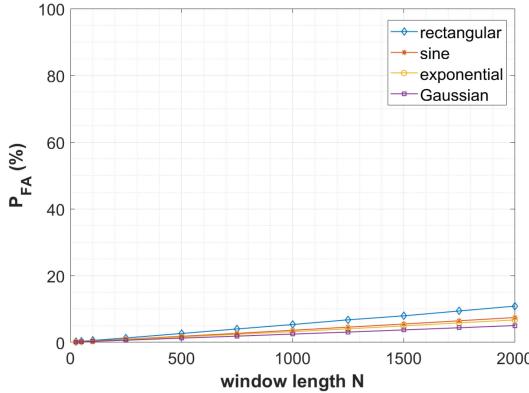


Fig. 27. Probability of false alarm versus moving average window length N for OFDM test signal with $\text{SNR} = 20 \text{ dB}$ using adaptive threshold detection with parameter $k = 0.1$ and different moving average window shapes.

results with the OFDM test waveform, albeit less pronounced than with the square wave. Detector performance at $\text{SNR} = 10 \text{ dB}$ is changed by a few percent at most. At $\text{SNR} = 15$ and 20 dB , the alternative window shapes result in improved performance, particularly when using longer window lengths. Once again, the Gaussian window showed the greatest performance improvements of around 5–10% for both P_D and P_{FA} using the maximum window length of $N = 2000$.

Determining the ideal window length and shape to use is not a trivial task, and it depends on the sampling rate and the characteristics of the signal or signals present on the channel. Based on our results, a good starting value for N is somewhere between $T_s/20$ and $T_s/10$, where T_s is the number of samples taken over the mean ON/OFF period of the channel. For example, the 100 Hz square wave sampled at 100 kHz has an ON/OFF period of 1000 samples, so a good starting value is in the range $N = 50$ to 100 . A value of N greater than around $T_s/5$ is likely to result in too much distortion to the detection statistic relative to the actual transmitter state, as per Fig. 3.

V. CONCLUSION

In this article, we discussed the implementation of an adaptive threshold energy detector using a USRP software defined radio and GNU Radio for the respective hardware and software platforms. We described the GNU Radio blocks used in the signal processing chain and provide block diagrams for the core of the energy detector and the transmit signal generation used to test the detector. Our discussion focused on the practical consideration and challenges faced when bringing energy detection spectrum sensing from theory and simulation into a realized hardware implementation. A procedure for benchmarking performance of an energy detector was outlined.

We performed a series of adaptive threshold energy detection experiments using two different types of transmit test waveforms. Our results showed the performance of our energy detector by examining probability of detection and probability of false alarm as a function of the two main

user-tunable parameters: moving average window length N and standard deviation coefficient k . We also performed a series of control experiments using a fixed threshold in order to show theoretical best-case detection performance for a fixed probability of false alarm. Our experiments demonstrate the feasibility of using the adaptive threshold algorithm for energy detection on an SDR, and the experimental results provide a reference for performance expectations. In addition, three alternative moving average window shapes were explored for use in computing the detection statistic. A moving average window with a Gaussian sample weighting showed the greatest improvement in detection performance and reduction in false alarm probability when detecting an OFDM communications waveform at around 15–20 dB SNR. Provided the channel state transitions are sufficiently separated in time, the USRP implementation of the adaptive threshold energy detector performs quite well at SNR levels of 15 dB and above.

Of course, energy detection efficacy suffers at low SNR levels, and the adaptive threshold alone is not going to overcome that fundamental shortcoming. Still, work on thresholding methods continues to be relevant. Contemporary approaches to wideband spectrum sensing such as compressive sensing and wavelet transform-based edge detection typically require some form of energy detection thresholding in order to classify bands as either occupied or vacant over a sampling period. By integrating spectrum occupancy modeling, detection performance can potentially be improved by increasing decision accuracy under situations of high ambiguity. A cognitive radio can use the detector output to model channel occupancy; by using an algorithm such as the ones proposed in [26], the cognitive radio can access the channel opportunistically by using “white spaces,” or periods when the primary users are not transmitting. Historical spectrum occupancy data can be used to train a superior analysis and decision architecture using Bayesian methods, maximum likelihood estimation, and machine learning techniques. Future work on cognitive radio PHY and MAC layers could develop these ideas and merge them with modern wideband spectrum sensing methods to accomplish accurate, real-time spectrum occupancy assessment across multiple channels.

REFERENCES

- [1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi
Internet of Things for smart cities
IEEE Internet Things J., vol. 1, no. 1, pp. 22–32, Feb. 2014,
doi: [10.1109/JIOT.2014.2306328](https://doi.org/10.1109/JIOT.2014.2306328).
- [2] S. Li, L. D. Xu, and S. Zhao
The internet of things: A survey
Inf. Syst. Front., vol. 17, no. 2, pp. 243–259, Apr. 2015.
- [3] M. Agiwal, N. Saxena, and A. Roy
Towards connected living: 5G enabled Internet of Things (IoT)
IETE Tech. Rev., vol. 36, no. 2, pp. 190–202, Mar. 2018,
doi: [10.1080/02564602.2018.1444516](https://doi.org/10.1080/02564602.2018.1444516).
- [4] Z. Ye, G. Memik, and J. Grosspietsch
Energy detection using estimated noise variance for spectrum sensing in cognitive radio networks
in *Proc. IEEE Wireless Commun. Netw. Conf.*, Las Vegas, NV, USA, 2008, pp. 711–716, doi: [10.1109/WCNC.2008.131](https://doi.org/10.1109/WCNC.2008.131).

- [5] D. Panaitopol, A. Bagayoko, P. Delahaye, and L. Rakotoharison
Fast and reliable sensing using a background process for noise estimation
in *Proc. 6th Int. ICST Conf. Cogn. Radio Oriented Wireless Netw. Commun.*, Osaka, Japan, 2011, pp. 221–225, doi: [10.4108/icst.crowncom.2011.245863](https://doi.org/10.4108/icst.crowncom.2011.245863).
- [6] R. K. Dubey and G. Verma
Improved spectrum sensing for cognitive radio based on adaptive threshold
in *Proc. 2nd Int. Conf. Adv. Comput. Commun. Eng.*, Dehradun, India, 2015, pp. 253–256, doi: [10.1109/ICACCE.2015.70](https://doi.org/10.1109/ICACCE.2015.70).
- [7] H. M. Farag and E. M. Mohamed
Improved cognitive radio energy detection algorithm based upon noise uncertainty estimation
in *Proc. 31st Nat. Radio Sci. Conf.*, Cairo, Egypt, 2014, pp. 107–115, doi: [10.1109/NRSC.2014.6835067](https://doi.org/10.1109/NRSC.2014.6835067).
- [8] R. N. Prashob, A. P. Vinod, and A. K. Krishna
An adaptive threshold based energy detector for spectrum sensing in cognitive radios at low SNR
in *Proc. IEEE Int. Conf. Commun. Syst.*, Singapore, 2010, pp. 574–578, doi: [10.1109/ICCS.2010.5686712](https://doi.org/10.1109/ICCS.2010.5686712).
- [9] M. Lopez-Benitez and F. Casadevall
Improved energy detection spectrum sensing for cognitive radio
in *IET Commun.*, vol. 6, no. 8, pp. 785–796, 22 May 2012, doi: [10.1049/iet-com.2010.0571](https://doi.org/10.1049/iet-com.2010.0571).
- [10] D. Raman and N. P. Singh
An algorithm for spectrum sensing in cognitive radio under noise uncertainty
Int. J. Future Gener. Commun. Netw., vol. 7, no. 3, pp. 61–68, 2014.
- [11] A. Gorcin, K. A. Qaraqe, H. Celebi, and H. Arslan
An adaptive threshold method for spectrum sensing in multi-channel cognitive radio networks
in *Proc. 17th Int. Conf. Telecommun.*, Doha, Qatar, 2010, pp. 425–429, doi: [10.1109/ICTEL.2010.5478783](https://doi.org/10.1109/ICTEL.2010.5478783).
- [12] A. Muralidharan *et al.*
An adaptive threshold method for energy based spectrum sensing in cognitive radio networks
in *Proc. Int. Conf. Control Instrum. Commun. Comput. Technol.*, Kumaracoil, India, 2015, pp. 8–11, doi: [10.1109/ICCI-CCT.2015.7475239](https://doi.org/10.1109/ICCI-CCT.2015.7475239).
- [13] J. J. Lehtomaki, J. Vartiainen, M. Juntti, and H. Saarnisaari
Spectrum Sensing with Forward Methods
in *Proc. IEEE Mil. Commun. Conf.*, Washington, DC, USA, 2006, pp. 1–7, doi: [10.1109/MILCOM.2006.302430](https://doi.org/10.1109/MILCOM.2006.302430).
- [14] J. J. Lehtomaki, J. Vartiainen, R. Vuohtoniemi, and H. Saarnisaari
Adaptive FCME-based threshold setting for energy detectors
in *Proc. 4th Int. Conf. Cogn. Radio Adv. Spectr. Manage.*, Barcelona, Spain, 2011, pp. 1–5, doi: [10.1145/2093256.2093289](https://doi.org/10.1145/2093256.2093289).
- [15] J. J. Lehtomaki, R. Vuohtoniemi, K. Umebayashi, and J.-P. Makela
Energy detection based estimation of channel occupancy rate with adaptive noise estimation
IEICE Trans. Commun., vol. E95-B, no. 4, pp. 1076–1084, Apr. 2012.
- [16] S. M. Budaraju and M. A. Bhagyaveni
A novel energy detection scheme based on channel state estimation for cooperative spectrum sensing
Comput. Elect. Eng., vol. 57, pp. 176–185, Jan. 2017.
- [17] S. Atapattu, C. Tellambura, and H. Jiang
Energy detection based cooperative spectrum sensing in cognitive radio networks
IEEE Trans. Wireless Commun., vol. 10, no. 4, pp. 1232–1241, Apr. 2011, doi: [10.1109/TWC.2011.012411.100611](https://doi.org/10.1109/TWC.2011.012411.100611).
- [18] S. Nallagonda, S. D. Roy, S. Kundu, G. Ferrari, and R. Raheli
Censoring-based cooperative spectrum sensing with improved energy detectors and multiple antennas in fading channels
IEEE Trans. Aerosp. Electron. Syst., vol. 54, no. 2, pp. 537–553, Apr. 2018, doi: [10.1109/TAES.2017.2732798](https://doi.org/10.1109/TAES.2017.2732798).
- [19] D. Cabric, A. Tkachenko, and R. W. Brodersen
Experimental study of spectrum sensing based on energy detection and network cooperation
1st Int. Workshop Technol. Policy Accessing Spectr., Boston, MA, USA, 2006, pp. 1–12, doi: [10.1145/1234388.1234400](https://doi.org/10.1145/1234388.1234400).
- [20] A. Ivanov, N. Dandanov, N. Christoff, and V. Poulikov
Modern spectrum sensing techniques for cognitive radio networks: Practical implementation and performance evaluation
Int. J. Comput. Inf. Eng., vol. 12, no. 7, pp. 572–577, 2018.
- [21] A. Nafkha, M. Naoues, K. Cichon, and A. Kliks
Experimental spectrum sensing measurements using USRP software radio platform and GNU-radio
in *Proc. 9th Int. Conf. Cogn. Radio Oriented Wireless Netw. Commun.*, Oulu, Finland, 2014, pp. 429–434, doi: [10.4108/icst.crowncom.2014.255415](https://doi.org/10.4108/icst.crowncom.2014.255415).
- [22] Prasetyo, R. V. W. Putra, T. Adiono, and A. H. Salman
Kurtosis and energy based spectrum detection for SDR based RF monitoring system
in *Proc. Int. Symp. Intell. Signal Process. Commun. Syst.*, Phuket, Thailand, 2016, pp. 1–5, doi: [10.1109/IS-PACS.2016.7824688](https://doi.org/10.1109/IS-PACS.2016.7824688).
- [23] R. A. Rashid *et al.*
Spectrum sensing measurement using GNU Radio and USRP software radio platform
in *Proc. 7th Int. Conf. Wireless Mobile Commun.*, Luxembourg City, 2011, pp. 237–242.
- [24] I. F. Akyildiz *et al.*
NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey
Comput. Netw., vol. 50, no. 13, pp. 2127–2159, May 2006.
- [25] P. D. Welch
The use of FFT for the estimation of power spectra: A method based on time averaging over short modified periodograms
IEEE Trans. Audio Electroacoust., vol. AU-15, no. 2, pp. 70–73, Jun. 1967.
- [26] T. A. Hall *et al.*
Dynamic spectrum access algorithms based on survival analysis
IEEE Trans. Cogn. Commun. Netw., vol. 3, no. 4, pp. 740–751, Dec. 2017.



Michael V. Lipski (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from The Pennsylvania State University (Penn State), State College, PA, USA, in 2017 and 2020, respectively, where he is currently pursuing the Ph.D. degree in electrical engineering.

In 2016, he worked as an undergraduate Researcher in the Applied Optoelectronics and Photonics Laboratory with Penn State. Since 2017 he has worked as a graduate Researcher in the Radar and Communications Laboratory with Penn State, and since 2020 he has worked as an electrical engineer with the U.S. Naval Research Laboratory, Information Technology Division, Washington, D.C., USA. His current research interests include dynamic spectrum access, cognitive radio, and coherent distributed systems.



Sastry Kompella (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the Virginia Polytechnic Institute and State University, Blacksburg, VA, USA, in 2006.

Currently, he is the Section Head for the Wireless Network Research Section under the Information Technology Division, U.S. Naval Research Laboratory, Washington, DC, USA. His research interests include various aspects of wireless networks, including cognitive radio, dynamic spectrum access and age of information.



Ram M. Narayanan (Life Fellow, IEEE) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Madras, Chennai, India, in 1976, and the Ph.D. degree in electrical engineering from the University of Massachusetts, Amherst, MA, USA, in 1988.

From 1976 to 1983, he was a Research and Development Engineer with Bharat Electronics Ltd., Ghaziabad, India, where he developed microwave communications equipment. In 1988, he joined the Electrical Engineering Department, University of Nebraska-Lincoln, where he last served as the Blackman and Lederer Professor. Since 2003, he is serving as a Professor of electrical engineering with The Pennsylvania State University. His current research interests include passive radar, radar detection through barriers, harmonic and nonlinear radar, noise radar, cognitive radar, medical radar, quantum radar, radar networks, radar reliability, information extraction in radar, and compressive sensing.

Dr. Narayanan is the recipient of the 2017 IEEE Warren White Award for Excellence in Radar Engineering. He is a fellow of the SPIE and IETE.