A Spectrum Sensing Technique Based on Autocorrelation and Euclidean Distance and its Comparison with Energy Detection for Cognitive Radio Networks

Hector Reyes^{1,3}, Sriram Subramaniam¹, Naima Kaabouch¹, Wen Chen Hu²

¹Electrical Engineering, University of North Dakota, Grand Forks, North Dakota, USA

²Computer Science, University of North Dakota, Grand Forks, North Dakota, USA

³Facultad de Ciencias Básicas e Ingeniería, Universidad de los Llanos, Villavicencio, Colombia

ABSTRACT

Spectrum sensing is an important aspect of cognitive radios. This paper describes a method for spectrum sensing based on the autocorrelation of the received samples. The proposed method was evaluated by means of experiments wherein the probabilities of detection and false alarm at different signal-to-noise ratios (SNRs) were observed. The platform used for the experiments was a set of Universal Software Radio PeripheralTM (USRPTM) devices acting as radio frequency front ends in combination with GNU Radio software. Since the signal processing was performed in the software domain, Gaussian noise of different levels was emulated by changing the standard deviation of a Python random number generator. In addition, the output power of a signal generator was varied to obtain different levels of SNR. A metric called the Euclidean Distance was derived to analyze the autocorrelation of the samples received by the USRPTM device in order to decide between two possible situations: only noise present or signal plus noise present. The proposed method was compared with two methods: one based on the value of the autocorrelation at the first lag and another one based on the power of the signal, known as energy detection spectrum sensing technique.

Index Terms – spectrum sensing, autocorrelation, USRPTM, Euclidean distance, energy detection, covariance matrix.

1. Introduction

Studies have shown that the radio spectrum is not utilized constantly by the users this resource has been allocated to; spectrum usage is estimated to be between 5% and 85% on a block of licensed radiofrequency (RF) bands at different geographical locations and at different times [1-4]. Cognitive Radio (CR), introduced by Mitola in 2000, has been proposed as a potential solution to the problem of radio spectrum scarcity and inefficient allocation. As defined in [5], a cognitive radio, built on a software-defined radio (SDR), is an intelligent heuristic wireless communication system with two main purposes: to provide highly reliable communications whenever and wherever needed and to ensure the efficient use of the radio spectrum. Although most of the current research focuses on the spectrum sensing aspect of cognitive radio, this technology has also the potential to enhance the reliability of wireless networks [6]. Spectrum management based on cognitive radio technology can employ the spectrum more efficiently by exploiting it opportunistically when not used by its licensees [5, 7, 8]. As conceived by Mitola [8], a cognitive radio executes the cognitive cycle, which includes observing, deciding, and taking action. Fig. 1 shows a simplified version of this cycle [9].

Fig 1. Cognitive Cycle (modified from [9])

In this paper, we propose and evaluate a spectrum sensing method based on the autocorrelation of the samples received by a radio frequency front end. In this method we compare a metric, the Euclidean distance between the autocorrelation of the samples and a reference line, with a threshold in order to determine the presence or absence of signal. We compared our method with other two techniques: a technique based on the value of the autocorrelation of the samples at the first lag, and a technique based on the power of the received signal, known as energy detection. To perform such comparison we used the probability of detection and the probability of false alarm as criteria. We obtained those probabilities through simulations and real experiments with GNU radio software and Python along with a set of Universal Software Radio Peripheral (USRPTM) devices acting as radio frequency front ends.

The remainder of this paper is organized as follows. Section 2 contains a brief overview of the related work on spectrum sensing. Section 3 presents the mathematical background of spectrum sensing based on autocorrelation, explains the proposed method, and the experimental setup used to assess it. Section 4 presents and analyzes the results of the experiments. This section compares the proposed technique with the energy detection and autocorrelation at the first lag techniques in terms of probability of detection and false alarm. Finally, we draw conclusions in section 5.

2. Related Work

Spectrum sensing is the cognitive radio component that detects the transmissions of primary users (PU). Primary users own the license to use specific portions of the radio spectrum. Spectrum sensing techniques aim to detect spectrum holes, time intervals wherein PU are not transmitting. Several spectrum sensing techniques have been proposed over the last decade [2-20]. Examples of such techniques include energy detection (ED) [10], matched filter (MF) detection [11, 12], cyclostationary feature detection (CSD) [13-15], and covariance-based detection [16].

The Energy Detection technique is the simplest spectrum sensing approach used in cognitive radio [17, 18]. Although it requires no prior knowledge about the primary signal, it does require knowledge of the noise power. In this method, the decision on the occupancy of the spectrum is made by comparing the average energy of the observed samples with a predetermined threshold that depends on the noise floor [10]. Cyclostationary Detection takes advantage of the fact that statistical parameters of practical communication signals vary periodically [19]. Cyclostationary data can be extracted from the spectral-correlation density (SCD) function [13-15]. In Matched Filtering, a matched filter correlates the known primary user signal with the received signal to detect the presence of PU. The matched filtering needs few samples; therefore, it requires short time to achieve low probabilities of missed detection and false alarm [12]. Thus, it is clear that both CSD and MF need information about the primary user signal prior to receiving it.

Covariance spectrum sensing can overcome these problems, since it capitalizes on the fact that the covariance matrix of only noise samples and the one of signal plus noise samples behave differently; therefore, it requires no information about either the noise or the signal power [16]. The covariance matrix of the samples collected by the receiver contains information exploitable for spectrum sensing. Zeng and Liang [16, 20] have proposed two methods that extract information from the covariance matrix of the samples. In [16] these authors introduce two statistics: the sum of the matrix elements that are not in the main diagonal, and the sum of the elements that are in the main diagonal. The comparison of the ratio between these two statistics with a threshold can indicate the presence of either signal plus noise or only noise. In [20] the authors propose other two metrics: the maximum to minimum eigenvalue (MME) ratio and the average received power to minimum eigenvalue ratio, also referred to as the energy with minimum eigenvalue (EME) ratio. As in [16], the comparison of these ratios with a threshold value can differentiate between noise and signal. Several authors have run simulations to evaluate the performance of their methods. Mate *et al.* [21] performed experiments with GNU Radio software and Universal Software Radio PeripheralTM (USRPTM) devices to evaluate the covariance and MME detection methods proposed in [16, 20]. They found that these two methods performed poorly in the practice because the autocorrelation of the noise samples is not a delta function; in other words, they are not delta correlated as assumed in [16, 20].

3. Methodology

3.1 Mathematical Background

Spectrum sensing is essentially deciding between two states: signal absent or signal present. In the technique described herein, the autocorrelation of the received signal samples is analyzed to decide between two hypotheses: $\mathcal{H}_0$, signal is absent, or $\mathcal{H}_1$, signal is present. In equations (1) and (2), the received samples are denoted as $x(n)$, the primary user signal as $s(n)$, and the noise as $\eta(n)$; the argument n denotes the n^{th} sample. Although the transmitted and received signals are continuous, throughout this paper the signal is treated as discrete, since the receiver takes discrete samples of the signal.

$$\mathcal{H}_0 : x(n) = \eta(n) \quad (1)$$

$$\mathcal{H}_1 : x(n) = s(n) + \eta(n) \quad (2)$$

The vectors $\mathbf{x}$ and $\mathbf{s}$ are defined as [16]:

$$\mathbf{x}(n) = [x(n) \ x(n-1) \ \cdots \ x(1)]^T \quad (3)$$

$$\mathbf{s}(n) = [s(n) \ s(n-1) \ \cdots \ s(1)]^T \quad (4)$$

The statistical covariance matrices of the vectors $\mathbf{x}(n)$ and $\mathbf{s}(n)$ are denoted by $\mathbf{R}_x$ and $\mathbf{R}_s$, respectively. $\mathbf{R}_x$ and $\mathbf{R}_s$ are defined as follows [16]:

$$\mathbf{R}_x = E [\mathbf{x}(n) \mathbf{x}^T(n)] \text{ and} \quad (5)$$

$$\mathbf{R}_s = E [\mathbf{s}(n) \mathbf{s}^T(n)] , \quad (6)$$

where E represents the expectation, and the superscript T represents the transpose.

The matrix $\mathbf{R}_x$ can be expressed as

$$\mathbf{R}_x = \mathbf{R}_s + \sigma_\eta^2 \mathbf{I}, \quad (7)$$

where σ_η^2 is the variance of the noise η , and $\mathbf{I}$ is the identity matrix. Therefore, in absence of signal, $\mathbf{R}_s$ is zero as well as the non-diagonal elements of $\mathbf{R}_x$. Based on this assumption, Zeng and Liang [16] have proposed the ratio between the sum of all the elements of $\mathbf{R}_x$: $T_1 = \sum_n \sum_m |r_{nm}|$ and the sum of its diagonal elements, $T_2 = \sum_n |r_{nn}|$ as

a metric to detect the absence or presence of signal. In absence of signal, $\frac{T_1}{T_2} = 1$, whereas with signal present,

$\frac{T_1}{T_2} > 1$. It is clear that in the first case, this ratio is $\frac{\sigma_\eta^2 \mathbf{I}}{\sigma_\eta^2 \mathbf{I}}$, whereas in the second case, it is $\frac{\mathbf{R}_s + \sigma_\eta^2 \mathbf{I}}{\sigma_\eta^2 \mathbf{I}}$. However, in

the reality, even in absence of signal, $\mathbf{R}_x$ is a non-diagonal matrix; therefore $T_1/T_2 > 1$ [18], which can lead into erroneously believing that there is a PU transmitting (i.e., a false alarm).

The authors in [16] assume that the noise η is such that $E(\eta(n)\eta(n+\tau))=0$ for any $\tau \neq 0$ and $E(\eta(n)\eta(n+\tau)) \neq 0$ for $\tau = 0$, which implies that the autocorrelation of the noise η , $E(\eta(n)\eta(n+\tau))$, is a delta-dirac function, or, in other words, the noise is delta-correlated. This condition holds when the noise is Gaussian. As we will see in the next section, in the practice the autocorrelation of noise is not a delta-dirac function, since non-Gaussian noise is also present. In the expression $E(\eta(n)\eta(n+\tau))$, E represents the expected value or expectation, $\eta(n)$ is the n_{th} sample of the noise, and τ is a discrete time shift.

We calculated the autocorrelation of the samples by means of the expression

$$\lambda(l) = \sum_{m=0}^{N_s-1} x(m)x^*(m-l), \quad (8)$$

where $\lambda(l)$ denotes autocorrelation at lag l , N_s the number of samples over which we calculate $\lambda(l)$, $x(m)$ the m_{th} sample, and $*$ the complex conjugate operation.

3.2 Experimental Setup

Fig. 2 shows the experimental setup. A Hewlett Packard 8684C signal generator emulated the signal transmitted by the primary user (PU). The Hewlett Packard 8684C generated a continuous sinusoidal signal tuned at 2410 MHz, the power of which we adjusted to vary the signal-to-noise ratio (SNR) level. We emulated the absence of PU by turning off the signal generator. To emulate the Gaussian noise we generated random numbers by means of a Python function. We varied the noise level by changing the standard deviation of the random numbers. Fig. 2 represents the emulated noise as η_{Sim} . We used a radiofrequency cable between the signal generator and the USRP unit in order to reduce external co-channel interference and noise. We used a USRP® model N200 unit as receiver [23]. A USRP is a radiofrequency front end that interfaces the radiofrequency domain with the binary domain, where the signals pass through a software implemented digital signal processor.

Fig. 2. Experimental setup.

Fig. 3 illustrates the USRP™ block diagram. The USRP™ is made of several blocks: one field programmable gate array (FPGA), one gigabit Ethernet controller, one analog to digital converter (ADC), one digital to analog converter (DAC), one receiver daughterboard, one transmitter daughterboard, and one or two antennas. The receiver and transmitter daughterboards are usually integrated in the same board; they contain the radiofrequency circuitry necessary for receiving and transmitting signals at specific frequency bands. The FPGA performs the digital signal processing for frequency conversion from RF to baseband and vice versa. The signals processed by the FPGA come from the ADC, which samples the RF signal at a rate defined by software, in this case GNU Radio. The gigabit Ethernet controller connects the USRP™ with a computer, where the digital processing takes place. We used GNU Radio [21], an open source software that contains several DSP blocks necessary to define radio systems, to perform the signal processing in the computer. GNU Radio uses C++ to define DSP blocks and Python to glue these blocks in order to build flow graphs that describe how to process the signal. In addition, GNU Radio users can create

customized blocks in either C++ or Python. The GNU Radio flowgraph not only processes the signal but also defines the central frequency and the sampling rate that the USRP® unit will use to operate. Through the flowgraph we configured the USRP™ to operate at 2410 MHz with a sampling rate of 1 million samples per second (MSPS), equivalent to a bandwidth of 1 MHz.

Fig. 3. USRP block diagram.

The GNU Radio software contains a USRP Hardware Driver (UHD™) that allows the computer to capture the samples coming from the USRP™ unit. These samples go to a Python Numpy array, represented by the L-element Numpy Array block in Fig. 2. The autocorrelation of the elements contained in the array is calculated in block ACB (Fig. 2). This calculation, defined by equation (8), yields another array. The distance between this array and a reference line R is the Euclidean distance metric, introduced in this paper. The line R , contained in a Numpy array, obeys the simple equation $R = \frac{-t}{L} + 1$, where t is a vector with L integers starting at 0 and ending at $L-1$, and L is the number of samples captured by the USRP™ device.

In our experiments we expected $\lambda(l)$ to be a delta-dirac function in absence of signal; however, that was not the case. Something similar occurred with the technique based on the value of the autocorrelation at the first lag, $\lambda(1)$, proposed in [22]: ideally, $\lambda(1) = 0$ when only noise exists, and $\lambda(1) \neq 0$ when signal and noise are present; nevertheless, we obtained $\lambda(1) \neq 0$ with no signal. What happened to the values of $\lambda(l)$ and $\lambda(1)$ might be caused by non-Gaussian noise injected by the receiver (USRP), a fact worth considering when deciding between $\mathcal{H}_0$ and $\mathcal{H}_1$ based on the covariance or autocorrelation of the received samples.

We attempted to improve both techniques, our method based on $\lambda(l)$ and the method based on $\lambda(1)$, by subtracting an estimate of the non-Gaussian noise introduced by the USRP™ receiver. We accomplished that by storing in a matrix several vectors containing N_s samples taken by the USRP™ with the signal generator turned off. Then, we averaged all the rows of the matrix, stored the result in the variable η_{USRP} (Fig. 2), and subtracted it from the signal being received by the USRP™ during the experiments.

4. Results and Discussion

Several experiments were performed to validate and test the efficiency of the proposed technique as well to compare it with the autocorrelation at lag 1, $\lambda(1)$, and energy detection techniques. Fig. 4 shows how the autocorrelation $\lambda(l)$ behaved in absence of signal (curve labeled “no signal”) and with signal at different levels of SNR. The curves in Fig. 4a show the results before subtracting η_{USRP} ; it is clear that the autocorrelation obtained with no signal and the one obtained with 0 dB SNR signal are very close. However, the curves in Fig. 4b show that, after

subtracting η_{USRP} , it becomes easier to differentiate the no signal and 0 SNR curves (i.e., between the conditions $\mathcal{H}_0$ and $\mathcal{H}_1$). Fig. 4 also shows that as SNR increases the Euclidean distance between $\lambda(l)$ and the reference line R , identified in this paper as $D_{Euclidean}$, decreases reaching zero when SNR is high. On the other hand, as SNR decreases, $D_{Euclidean}$ increases, reaching its maximum value in the absence of signal. Therefore, $D_{Euclidean}$ surpassing a certain threshold indicates absence of signal. To make it easier to determine a threshold, we have normalized $D_{Euclidean}$ so its maximum is 1 and its minimum 0.

Fig. 4a

Fig. 4b

Fig. 4. Behavior of the autocorrelation $\lambda(l)$ with no signal and with different values of SNR. Notice that the Euclidean distance from the autocorrelation curve $\lambda(l)$ to the reference line decreases as SNR increases.

Fig. 5 (Fig. 5a and Fig. 5b) shows the normalized values of the Euclidean Distance and ACF(1) with respect to SNR for different values of the standard deviation of the emulated noise η_{Sim} . Fig. 5a shows that the Euclidean distance exceeding a certain threshold indicates the absence of signal ($\mathcal{H}_0$), whereas when it falls below that threshold, it indicates presence of signal ($\mathcal{H}_1$). Fig. 5a shows that at very low levels of SNR (-30 dB to -25 dB), the Euclidean distance reaches 1, its maximum. As the SNR increases $D_{Euclidean}$ decreases suffering a steep decay when SNR is between -10 dB and 15 dB. For a SNR greater than 20 dB, $D_{Euclidean}$ becomes zero. On the other hand, Fig. 5b shows that ACF(1) exceeding the threshold indicates the presence of signal ($\mathcal{H}_1$), whereas when it falls below that threshold, it indicates the absence of signal ($\mathcal{H}_0$). The behavior of ACF(1) with respect to the SNR is the opposite of the Euclidean distance behavior: ACF(1) increases along with SNR, exhibiting a steep rise when SNR is between -10 dB and 15 dB.

Fig. 5a

Fig. 5b

Fig. 5. Behavior of $D_{Euclidean}$ as SNR changes with different standard deviations of η_{Sim} . Behavior of autocorrelation at the first lag ACF(1) as SNR changes with different standard deviations of η_{Sim}

Fig. 6a

Fig. 6b

Fig. 6. Behavior of Probability of Detection as SNR changes with different number of samples of $D_{Euclidean}$ and ACF(1).

To evaluate the impact of the number of samples N_s on the performance of the method, we ran experiments with different numbers of samples, and observed the changes in the probability of detection with varying levels of SNR. Fig. 6 (Fig.6a and Fig.6b) shows that when the number of samples increases, the probability of detection increases as well.

Fig. 7a

Fig. 7b

Fig. 7. Behavior of Probability of Detection as SNR changes with different threshold levels of ACF(1) and $D_{Euclidean}$

To evaluate the impact of the threshold selection on the performance of the method, we ran experiments with different thresholds for both methods and calculated the P_d for varying levels of SNR. Fig. 7 (Fig.7a and Fig. 7b) represents the change in probability of detection with respect to SNR for different thresholds for both the Euclidean distance and the ACF(1) methods. Fig. 7 shows that the best threshold for the Euclidean distance method is 0.95 and for the ACF(1) method is 0.1. By best threshold we mean the one that allows the sensing method to yield the highest P_d .

Fig.8. Behavior of Probability of Detection as SNR changes for 1024 samples and threshold levels of $D_{Euclidean} = 0.95$ and ACF(1)

Finally, Fig. 8 compares both methods as they operate with their best thresholds and 1024 samples. In this figure, the results clearly indicate that the Euclidean distance method outperforms the ACF(1) method, since it achieves a probability of detection $P_d=100\%$ at a lower SNR. The gain in SNR for the Euclidean distance method is

approximately 4 dB.

The results shown by Fig. 9b imply that the probability of detection (P_d) for the auto-correlation method is low at a negative levels of SNR, but consistently attains 100% P_d at around 0 dB, whereas the energy detection method attains the same at -5 dB. Comparing the simulation results with those of the real experiments (Fig. 9a), it is apparent that the energy detection achieves a P_d of 100% at -8 dB, whereas the ACF(1) method (*i.e.*, the auto-correlation counterpart of the simulation) achieves the same at -6 dB, showing a better detection performance at levels of SNR -5dB to -8dB lower than the one of the simulation. The Euclidean distance method (labeled as “Distance” in figures 9 and 10) shows better performance than the ACF(1) method regarding the probability of detection. The Euclidean distance method has proven to be more effective than the ACF(1) method by having a SNR gain of 4 dB for a given SNR.

Fig. 9a

Fig. 9b

Fig. 9. Simulation and Real Experiment– Probability of Detection vs. Signal-to-Noise Ratio.

Fig. 10a

Fig. 10b

Fig. 10. Probability of false alarm vs. Signal-to-Noise Ratio.

Fig. 10 shows that according to real experiments (Fig. 10a) and simulations (Fig. 10b), the methods based on autocorrelation perform better than the energy detection method regarding the probability of false alarm. Both graphs show that in the case of the energy detection method the probability of false alarm is around 50 %, whereas in the case of the autocorrelation based methods is approximately 0.

5. Conclusion

Experiments and simulations showed that the Euclidean Distance method proposed herein is more efficient than the ACF(1) method in terms of probability of detection and false alarm, and more efficient than the energy detection method in terms of probability of false alarm. Future work includes characterization of the type of noise introduced by the USRP device, application of the method to scanning several channels in order to estimate their utilization level, and measurement of the speed of the method.

Acknowledgements

The authors acknowledge the support of NSF, grant # 1443861, and EPSCoR/NSF, grant # EPS-0184442.

References

- [1] M. T. Masonta, M. Mzyece and N. Ntlatlapa. Spectrum decision in cognitive radio networks: A survey. *Communications Surveys & Tutorials, IEEE 15(3)*, pp. 1088-1107. 2013.
- [2] N. Kaabouch and WC Hu, Software-Defined and Cognitive Radio Technologies for Dynamic Spectrum Management, IGI Global, Volumes I and II, October 2014.
- [3] S. Subramaniam, H. Reyes, and N. Kaabouch, "Spectrum Occupancy Measurement: An Autocorrelation based Scanning Technique using USRP," IEEE WAMICON, 2015.
- [4] H. Reyes and N. Kaabouch, "Improving the reliability of unmanned aircraft system wireless communications through cognitive radio technology," Communications and Network, Vol. 5 No. 3, pp. 225-230, 2013.
- [5] S. Haykin. Cognitive radio: Brain-empowered wireless communications. *IEEE J. Select. Areas Commun.* 23(2), pp. 201-220. 2005.
- [6] A. Azarfar, J. Frigon and B. Sanso. Improving the reliability of wireless networks using cognitive radios. *Communications Surveys & Tutorials, IEEE PP(99)*, pp. 1-17. 2011.
- [7] I. Akyildiz, Won-Yeol Lee and K. Chowdhury. Spectrum management in cognitive radio ad hoc networks. *Network, IEEE 23(4)*, pp. 6-12. 2009.
- [8] J. Mitola III and G. Q. Maguire Jr. Cognitive radio: Making software radios more personal. *Personal Communications, IEEE 6(4)*, pp. 13-18. 1999.
- [9] L. Doyle. *Essentials of Cognitive Radio* 2009.
- [10] H. Urkowitz. Energy detection of unknown deterministic signals. *Proc IEEE 55(4)*, pp. 523-531. 1967.
- [11] T. Yucek and H. Arslan. A survey of spectrum sensing algorithms for cognitive radio applications. *Communications Surveys & Tutorials, IEEE 11(1)*, pp. 116-130. 2009.
- [12] R. Tandra and A. Sahai. Fundamental limits on detection in low SNR under noise uncertainty. Presented at Wireless Networks, Communications and Mobile Computing, 2005 International Conference On. 2005, .

- 283 [13] W. A. Gardner. Exploitation of spectral redundancy in cyclostationary signals. *Signal Processing Magazine*,
284 *IEEE* 8(2), pp. 14-36. 1991.
- 285 [14] W. Gardner. Spectral correlation of modulated signals: Part I--analog modulation. *Communications, IEEE*
286 *Transactions On* 35(6), pp. 584-594. 1987.
- 287 [15] W. Gardner, W. Brown and C. K. Chen. Spectral correlation of modulated signals: Part II--digital modulation.
288 *Communications, IEEE Transactions On* 35(6), pp. 595-601. 1987.
- 289 [16] Y. Zeng and Y. Liang. Spectrum-sensing algorithms for cognitive radio based on statistical covariances.
290 *Vehicular Technology, IEEE Transactions On* 58(4), pp. 1804-1815. 2009.
- 291 [17] F. de Carvalho, J. S. Rocha, W. T. Lopes and M. S. Alencar. A spectrum sensing algorithm based on statistic
292 tests for cognitive networks subject to fading. Presented at Signal Processing Conference (EUSIPCO), 2013
293 Proceedings of the 22nd European. 2014, .
- 294 [18] M. López-Benítez and F. Casadevall. Improved energy detection spectrum sensing for cognitive radio. *IET*
295 *Communications* 6(8), pp. 785-796. 2012.
- 296 [19] Y. Zeng, Y. C. Liang, A. T. Hoang and R. Zhang. A review on spectrum sensing for cognitive radio:
297 Challenges and solutions. *EURASIP Journal on Advances in Signal Processing* 2010pp. 2. 2010.
- 298 [20] Y. Zeng and Y. Liang. Eigenvalue-based spectrum sensing algorithms for cognitive radio. *Communications*,
299 *IEEE Transactions On* 57(6), pp. 1784-1793. 2009.
- 300 [21] A. Mate, K. Lee and I. Lu. Spectrum sensing based on time covariance matrix using GNU radio and USRP for
301 cognitive radio. Presented at Systems, Applications and Technology Conference (LISAT), 2011 IEEE Long Island.
302 2011, .
- 303 [22] R. K. Sharma and J. W. Wallace. Improved autocorrelation-based sensing using correlation distribution
304 information. Presented at Smart Antennas (WSA), 2010 International ITG Workshop On. 2010, .
- 305 [23] M. Ettus. Universal software radio peripheral (USRP). *Ettus Research LLC* [Http://Www.Ettus.Com](http://Www.Ettus.Com) 2008.
- 306 [24] G. Radio. The gnu software radio. Available from World Wide Web: [Https://Gnuradio.Org](https://Gnuradio.Org) 2007.

307 Biographies

308 **Hector Reyes** received his BSEE and M.Sc. degrees from the Universidad Distrital Francisco Jose de Caldas in
309 1998 and 2003 respectively. In 2014 he received his Ph.D. degree from the University of North Dakota, USA.
310 Currently he is a professor at the Universidad de los Llanos in Villavicencio, Colombia. His research interests
311 include Cognitive Radio, WSN, and Rural Telecommunications.

312 **Sriram Subramaniam**, born in Bangalore, received his Bachelors of Engineering in Electronics and
313 Communication from Visvesvaraya Technological University, India in 2011. With two years of work experience in
314 software development, he is currently pursuing his Masters of Science degree at the Department of Electrical
315 Engineering of the University of North Dakota. His research interests include Cognitive Radio and Wireless
316 Networks.

317 **Naima Kaabouch** is currently an associate professor in the E.E. Department at the University of North Dakota,
318 USA. She received her B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Paris, France.
319 Her research interests include signal/image processing, sensing, smart systems, wireless communications and
320 cognitive radio.

321 **Wen-Chen Hu** is currently an associate professor in the Computer Science Department at the University of North
322 Dakota, USA. He received a PhD in Computer Science from the University of Florida, Gainesville, in 1998. His
323 research interests include handheld-mobile/smartphone/tablet computing, location-based services, Web-enabled
324 information systems such as search engines and Web mining, electronic and mobile commerce systems, and Web
325 technologies.

326

327

328

RF: Radiofrequency