Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 11, Number 5 (2015), pp. 3349-3359

http://www.ripublication.com

Cognitive Radio Implementation with GRC

Hemalatha M[#1], Jagadeeswar Reddy.G[#1], Murali.K[#1], Nageswara Rao E[*2]

[#1]*Department of ECE, Narayana Engineering College,*
Nellore, Andhra Pradesh, India
[*2]*King Khalid University, Abha, Kingdom of Saudi Arabia*

Abstract

The main philosophy behind Cognitive Radio is to efficiently access and utilize an available frequency spectrum in an opportunistic manner. Given today's scenario where the number of private operators are on the rise, frequency seems to be one of the most sought after resources. In contrast to illusion, there is abundance in spectrum availability and the supposed spectrum shortage is partially an artifact of the regulatory and licensing process. In this work, an attempt is made to implement such a Cognitive Radio using GNU Radio Companion (GRC), a software to design, implement and test Software Defined Radio (SDR). Firstly the Bit Error Rate (BER) is measured for different modulation techniques against various available noise distributions for a given channel. Upon comparing the BER for various noise distribution and modulation techniques, the best OFDM technique is determined for each noise distribution. On implementing the SDR, a selector block is assigned as a switching element to select between the best modulation techniques for a given channel with a noise distribution in a real time situation.

Keywords - Cognitive Radio, spectrum sensing, spectrum sharing, WRAN, SDR

I INTRODUCTION AND LITERATURE SURVEY

Due to overlapping of allocations on all the bands reserved by FCC, it creates an illusion of shortage of frequency spectrum for wireless communications [1]. The typical utilization however, is only around 0.5% in the 3-4GHz band and 0.3% in the 4-5 GHz band. The cognitive radio identifies and selects the void in the frequency bands, known as 'spectrum holes' to transmit over them. This procedure is called 'Spectrum sensing' and has been extensively studied over the years. A challenging issue of the emerging IEEE 802.22 wireless regional area networks (WRAN) is to solve the problem ofreliable spectrum sensing and sharing with satisfactory QoS.

The primary objective of CR is detection and occupying of spectrum holes without interference in to the boundary of primary licensed nodes.While this has been the basis for cognitive research, a few alterations to the primary idea have been perceived and implemented.

Depending on the parameters considered a distinction can be made between Cognitive Radios as fully blown cognitive radio, which constitutes every possible parameter and the spectrum sensing radio, which deems only the radio frequency spectrum. The four main functions of Cognitive Radio are spectrum sensing, spectrum management, spectrum mobility and Spectrum sharing. Spectrum Sensing is an important part of Cognitive Radio as pointed out by a lot of researchers, for instance, survey reports on spectrum sensing techniques in cognitive radio [2, 3]. At the heart of the Cognitive Radio lies the Cognitive Engine which can comprehend and make decisions based on the spectrum environment. The characteristic which distinguishes Cognitive Radio is its ability to learn about the spectrum environment and locate spectrum holes which can be allocated to services in need of radio frequency spectrum. A shortage of spectrum for wireless services as a result of traditional management schemes has caused a steep rise in demand for Cognitive Radio Technologies.

Spectrum Sensing is used as a tool to detect unused radio resources and to estimate the interference level is a vital element in Cognitive Radio. Spectrum Sensing relies solely on the Secondary User to detect spectrum holes or “white space” in the RF spectrum. Spectrum Sensing schemes allow secondary users to monitor the licensed frequency bands and opportunistically utilize the frequency band when not used by the primary user while at the same time ensuring that the interference level is below the acceptable limit. Based on the manner in which they are detected they can be classified into Cooperative Detection, Interference based Detection, Receiver Detection and Transmitter Detection.

Frequency is one of the most sought after resource in the current day communication scenario. Given the rapid increase in the number of devices using wireless communication, efficient usage of frequency spectrum has become all the more necessary. One main reason for this situation is that considerable portions of the frequency spectrum have been allocated to the private operators. Although private operators have been provided with ample resources in terms of frequency, much of what has been allotted is still left to be used. Unlike licensed spectrum, the problem with unlicensed spectrum is that the available bandwidth is unable to accommodate theincrease in the number of devices using wireless communication. The scenario of frequency wastage is also the same for Military communication, where much of the allotted frequency is left unused. Cognitive Radio thus aims to use frequencies in an opportunistic manner and in caseof licensed spectrum, prior permission from the license holder must be sought.ThetermCognitive Radio was coined by JosephMitola III in his articleco-writtenwithGeraldQ.Maguire, Jr., in 1999, in which it was described as an SDR which can dynamically change its parameters depending on the need and user requirements[4-7].

The very concept of cognitive radio has been born out of the need to look at the problem of spectrum sharing and utilization in a very dynamic manner. Given a scenario involving a frequency spectrum that is being used by several entities, a cognitive radio is nothing but a transceiver that has been designed and trained to look out for available frequencies, to single them out, to modify its parameters accordingly and also that of the receiver with which it is communicating so that maximum usage of the given frequency spectrum is ensured [8, 9]. The main reason to why cognitive radio seems to be the answer to efficient sharing is the basing of the sharing process on the principles of dynamic spectrum management. The parameters that are prone to change in general are waveform, protocol, operating frequency and networking.The main work of logic in this case lies in how the spectrum is shared amongst several users. Also, if fully fledged usage of this technology is attempted, efforts are required to maintain seamless communication even when there is a transition to a better spectrum by keeping hold of all necessary resources. At the moment, the technology has been implemented in its most primitive form in the television networks in the USA. Conceptually, it is nothing but a software defined radio with a cognitive engine brain [10-12]. The basic working of a cognitive radio is to simply identify a white space, the pattern of noise existing in the space, to identify the method that suits the situation best[13-17]. Beyond this, what remains is how the transceivers at the sending and receiving ends synchronize with one another to pass information. To employ such a technique in the unlicensed spectrum would improve the overall spectrum usage. However, when it comes to licensed frequency spectrum and military communication spectrum, the necessary privileges need to be acquired from the appropriate license holders in order use the channel provided the license holder is not affected in the process [18-24]. The typical wideband analog RF front end consists of several components such as the RF Filter, low noise amplifier and the analog to digital converter. Here, by using the GNU Radio Companion (GRC), the RF front end iscompletely replaced by the equivalent signal processing blocks of the GRC. For a simulation done on a local host, the user is provided with the appropriate sources and sinks depending upon the type of data intended to be transferred. For the implementation of an actual working SDR however, needs the appropriate Universal Software Radio Peripheral. In this case the USRP is left with only the task of transmission and reception while the entire functionality done by the regular RF front end is taken care of by the signal processing blocks included in the GRC flow graph.

II SIMULATION MODELS

A. Measurement of Bit Error Rate (BER)

In order to ascertain the efficiency of a particular modulation technique, the bit error rate measurement of each such technique is taken as the yard stick. Keeping this as the ruling to go by, it uses later in the switching element to flag the technique to be used in a particular channel with a particular level of noise. In this way, cognition becomes possible by having a decision unit like entity that is introduced into a flow graph that shown in fig.1is otherwise full of blocks that perform some function on the incoming signal itself.

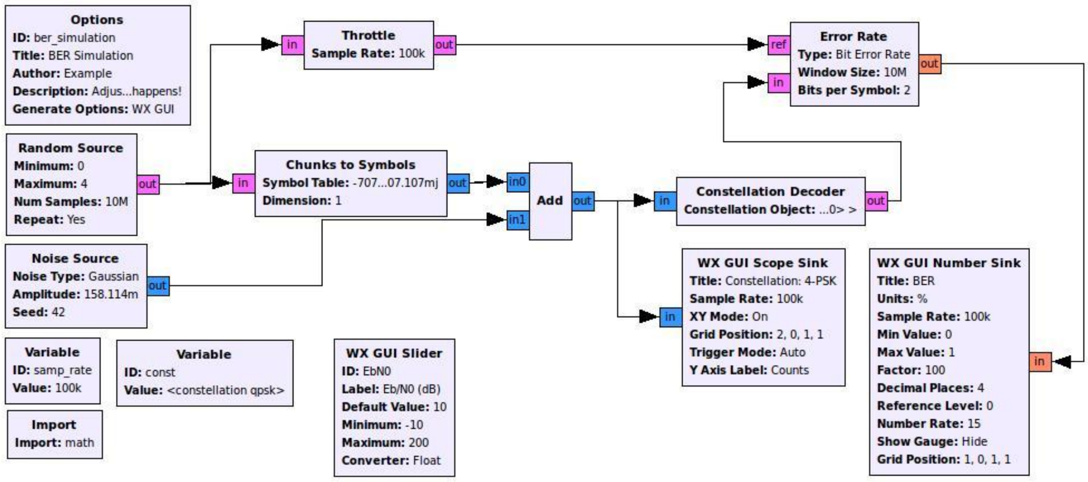


Fig. 1 Bit Error Rate Flow Graph

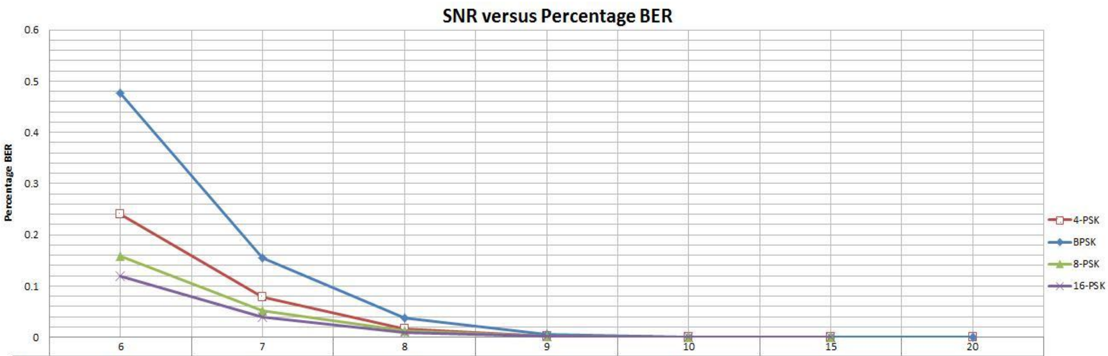
BER is nothing but the number of incorrect bits divided by the total number of bits that is being sent. It may be measured in terms of percentage, or expressed as a power of ten. For any efficient technique it is often of the order 0.001% or so.

In the given flow graph, the bit error rate for each of the different shift keying techniques that are available, namely BPSK, QPSK, 8PSK, 16PSK is measured. The flow graph shown uses a random source as the start point for data flow. A random source is a block that generates a random sequence of bits. It has four parameters that must be given values. It works by generating a stream of bits of different lengths from at different intervals. The minimum length of the bit stream generated is the value given to the first parameter, ‘min’. Likewise, the second parameter, ‘max’ represents the highest value of length that a bit stream being generated can have. It is given the parameter ‘min’ the

value zero and the parameter ‘max’ the value four, meaning that the random source block can generate a bit stream that is not longer than four bits, and at the same cannot be less than or equal to zero.It is to be noted that the ‘min’ is not included in the range of values that the bit stream can take. Also, the ‘random source’ block generates a fixed number of bits for a particular trial of a flow graph. It may be of the order of several millions of bits. For one run of the flow graph, it generates the bits that in total amount to the length specified in the parameter ‘num samples’. Once the ‘random source’ block has generated samples equal to that number it should continue in order for the flow graph to continue working as well. So, to achieve this, fourth parameter, that is the ‘repeat’ parameter is set to ‘yes’. This way, it is seen that once the initial sequence of ten million bits is generated, the block repeats the sequence again. This action takes place so long as the flow graph is made to run until it is stopped.Beyond the ‘random source’ block the graph branches out to two different paths. Since it needs to measure the bit error rate here, the original signal is put through a path where some noise is introduced in to the signal with ‘noise source’ block. The purpose of the 'noise source' block is to generate a particular noise pattern in order to be inserted in to an incoming sequence of bits. There are several types of noise patterns that may be generated using the 'noise source' block. However, some models such as the Gaussian (given as AWGN- Additive White Gaussian Noise), Laplacian, and normal distribution are more common. The level of noise as the bit sequence is generated varies as per the distribution given by their respective functional waveforms.

The noise generated by the 'noise source' block may be of the types float, double, or even complex. Here, since the data is converted to the type 'complex', another parameter of this block that must be tended to is the 'amplitude' parameter. Though noise may be signal that has no inherent pattern whatsoever, this parameter has to be set to give some form of reference to the 'noise source' block. The value for amplitude must be set such that it is not so high that it completely overshadows the signal that must be studied. At the same time, it must not be too low that it has no visible bearing on the source signal at all. It is seen that the value for amplitude is set at 158.114mV. The suffix here indicates a power of ten. 'm' indicates the value that is ten raised to the power of negative three. Likewise, several such suffixes exist, namely M for ten raised to the power of six, that is, one million and so on. A third parameter, called 'seed' exists which is given the value 42. This parameter is what is used to actually generate the noise, that is, it initializes the noise source block. Care should be taken to set the seed value negative for uniform and Gaussian distributions. Once these parameters are set, the 'noise source' block starts generating the noise, which is nothing but a random sequence of bits that must be inducted into the actual source signal. By comparing the actual signal as with a noise induced version of itself, it arrives at the number of bits that have been modified from the actual signal and those that do not comply with the actual bit sequence. This eventually gives the value for the bit error rate (BER), here, as a percentage of the total number of bits in the input sequence. In the part of the graph that generates and sends over the actual signal is situated the 'throttle' block. This block is one of the most important components required to run a flow graph acting under a live input stream. A common issue that arises when running a flow graph is that the rate at which it is run may be faster than the instruction rate accepted by the processor. So, in order to keep the rate under check, the throttle block is introduced. The throttle sets the flow graph to work at a rate that is compatible with the processor on which the flow graph is running. This is done by setting it to work at the same rate as the one at which the source signal is being sampled. So, it is to be noticed that the parameter 'Sample rate' in the 'throttle' block is set to the same value, that is, 100k which is same as the value set in the 'variable' block. After this point, the output of this block is sent to the Bit Error Rate block where actual measurement of the bit error rate takes place. On the other part of the graph that splits from the 'random source' block, is the path in which noise is induced into the source signal. First, the 'random source' block is followed by the 'chunks to symbols' block.

This block is one that can convert data that flows through a graphfrom one data type to another. It basically maps a stream of symbol indexes, of the type unpacked bytes or short to a stream of float or complex constellation points, the type of constellation depending upon the value given to the 'dimension' parameter. That is, the incoming stream of data, of the type 'byte' is mapped to the 'complex' type in 'n' dimensions, where 'n' is the number of dimensions into which the incoming data is to be mapped. By default, the value of this parameter is 1.



SNR	6	7	8	9	10	15	20
4-PSK	0.2406	0.0777	0.0156	0.0025	0.0004	0.0002	0.0001
BPSK	0.4776	0.1545	0.0384	0.0061	0.0007	0.0002	0.0001
8-PSK	0.1593	0.0515	0.0126	0.0022	0.0002	0.0001	
16-PSK	0.1193	0.0386	0.0094	0.0014	0.0002	0.0001	

SNR

Fig. 2 SNR vs Percentage BER for various Modulations

Once the symbols have been mapped to appropriate values on the complex plane, the noise is generated by the 'noise source'. This is achieved using the 'add' block. This block simply adds noise to the signal and yields the resulting value, depicting it on the complex plane. In general, the more the noise, the more is spatial spread of the values plotted on the plane. As the noise reduces, it is observed that the spread of the values reduces and as the noise approaches zero, the spread almost reduces to a point on the complex plane. The 'scope sink' depicts the output waveform of a signal that has been received. Therefore, it is necessary for the user to ensure at this point whether the addition of noise has been effective or not. Also, it gives the user a visual angle to gauge how the original signal has been affected. Care must be taken to ensure that the sample rate of the 'scope sink' is the same as the sample rate mentioned in the 'variable' block. In the 'constellation decoder' block, the values obtained upon inducing noise into the source signal, depicted on the complex plane are once again read and 'byte' type values are once again obtained.

The percentage BER is plotted against SNR for various modulation techniques in fig.2. It is observed that the BER get reduced as the SNR is more and 16PSK proves its best candidature even for lower SNR values amongst its competent techniques such as BPSK, QPSK and 8PSK. The parameter 'constellation object' accepts a raw input, a type which simply takes whatever value is keyed in by the user. Here, the value of the 'constellation object' is 'const.base (2)'. The Bit Error Rate block has two parameters, namely 'window size' and 'bits per symbol'. The first parameter is set to the total length of the bit stream that is generated by the 'random source' block, here the value being 10M. The next parameter, the 'bits per symbol' ascertains the shift keying technique being tested here. It is known that BPSK uses 2 symbols per bit, QPSK uses 1 symbol per 2 bits and so on. This way by setting the value to the appropriate power of 2, BER for a particular modulation technique is estimated.

B) Image Transfer using OFDM in a Noisy Channel, Measurement of Phase Error

Apart from measuring Bit Error Rate for data in general, it may be more practical in nature if more commonly used forms of data, in the form of audio or image are attempted to be transferred through a noisy channel. Although the earlier attempted to transfer a stream of bits, it still does not compare to an actual file transfer that is done on an existent already as it resembles the real time situations to the closest extent possible. The former had the disadvantage no matter how long the stream of bits was, it was repeated in an endless cycle and so, the level of variation that could be seen was always limited as one way or the other the data pattern, or the pattern of the error could not be random always but rather would fall into one of the predictable patterns inevitably. However, when it comes to the case of the image, the length of the stream of bits on which the simulation carried out is limited but the pattern is truly random and every image possesses a unique set of pixel intensities, thereby providing a unique case to be studied. The 'File Source' block has two parameters. The first parameter, source requires the user to specify the full path of where the file is located and follow it with the file name.

Care should be taken to note that the noise variation that can be produced using the flow graph is limited. So, if high resolution images are going to be used, the change that is reflected on them is too small to be observed by the naked eye. So, it is preferable to use a low resolution 'bmp' image since the pixels are more visible to the naked eye when compared to a highresolution image. This helps since any change that is brought to the image by introducing the noise in the channel is just as well perceptible.

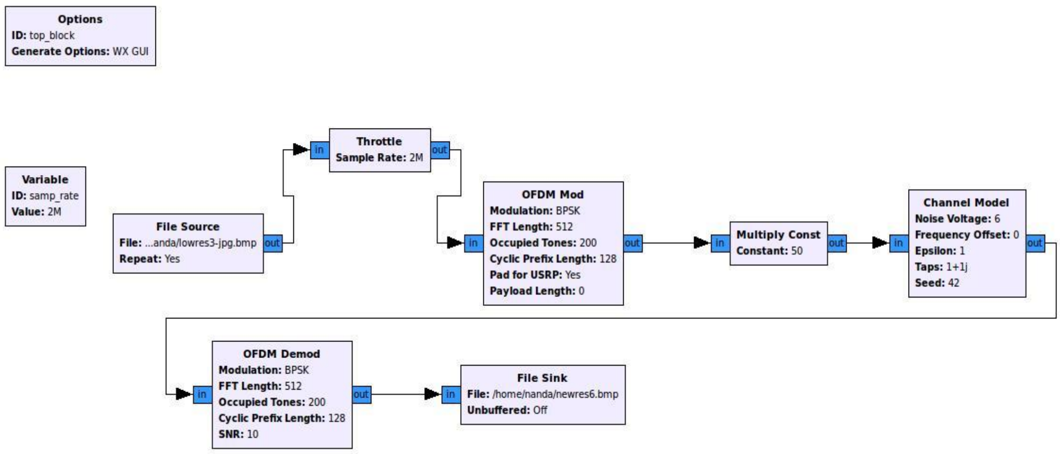


Fig. 3. Image Transmission Flow Graph

The second parameter ‘repeat’, repeatedly sends the data, in the form of the image through the flow graph until it is killed. Following the ‘File Source’ block is the ‘throttle’ block, the sample rate of which is set to 2M, which is two raised to the power of six, or two million. This value is set in the ‘variable’ block first and this change is automatically reflected in the ‘throttle’ block. As mentioned earlier, the throttle block reduces the working speed of the flow graph execution to suit the speed of the processor. The throttle block is then wired to the ‘OFDM Mod’ block which is entrusted with the task of performing the appropriate shift keying technique on the incoming signal. The shift keying technique that is to be used may be modified in the first parameter, ‘modulation’. This parameter provides several options such as BPSK, QPSK, 8PSK, 16PSK. The Image transmission flow graph is shown in fig.3.The parameter ‘Cyclic Parameter Length’ denotes the length of the vector that must be added as prefix to the input signal to size it up to the length specified as the input length. Upon performing the necessary operation, the resulting vector is converted into a bit stream that is as long as specified as the output stream length. Also, it is to be noted that the ‘payload length’ parameter has to be given a length that is a multiple of eight, as the block accepts bit streams that are grouped into bytes. The next block that follows isthe ‘Multiply Cons’ block. It simply boosts the strength of the signal that is being transmitted. This block is available in the module that deals with mathematical operations that can be done on a signal. The block that follows this is the block central to the flow graph in the sense that it holds the very purpose of executing the flow graph in the very first place, the ‘Channel Model’ block. This block is used to set the parameters for the simulation of the channel through which the user’s data is to be transferred. The first parameter in this block is ‘Noise Voltage’. This parameter simply decides how much noise is to be inserted into the channel. Though noise can never be simulated, the level of noise, once ascertained may produce the same effect as having that level of noise in the real time. There are several noise models, namely Additive White Gaussian Noise (AWGN), Laplacian, Normal distribution etc. Here, the default noise type for which the ‘Channel Model’ provides noise level is Gaussian.. Any noise voltage outside this range leads to inability in generating the image. The final parameter, named ‘seed’ is used to set the seed value for random generation. Next comes the ‘OFDM Demod’ block, the parameters of which are mostly set to the same values as that of the corresponding ‘OFDM Mod’ block in the earlier part of the flow graph. One parameter that is different here is the final parameter, ‘SNR’, which is set to a value of 10. This block is finally connected to the ‘File sink’ block where the resulting image is received. The first parameter of this block requires the destination of the result file, followed by the name of the file in which the corrupt image will be stored. Care must be taken to have the image format of the source and resultant files as one and the same. The fig.4 gives the inference on the effect of noise in the communication channel.

Fig. 4. Comparison of transmitted and received images

III CONCLUSION

The study that had been conducted over the past few months served as a good insight into the world of cognitive radio. At the outset, it was intended to ascertain the conditions suitable for the usage of cognitive radio, which in turn prompted a study of its behavior under different conditions that were mimicked by giving different values to the parameters. As a first step towards studying software defined radio, an attempt was made to record its behavioral changes when subjected to varying levels of channel noise for different channel models. The yard stick used for this study was the measurement of the bit error rate, here measured in percentage. This study was carried out by generating a random stream of bits, which was then put through a noisy channel, the noise being induced by the 'noise source' block. Also, provision was made to vary the SNRlevel using a slider. The resulting constellation diagram was studied and the percentage bit error rate value was noted. This procedure was repeated keeping in mind the several shift keying techniques available. So, to simulate different shift keying techniques, the bits per symbol parameter was changed accordingly to simulate techniques such as BPSK, QPSK, 8PSK, 16PSK. In order to provide a comparative view of the efficiency of various techniques under similar sets of conditions, a graph was plotted depicting the varying nature of response. Next, in order to simulate the effect of noise affecting a real time data transfer, an image was attempted to be transferred over a noisy channel and the effect was observed. Here again, in order to study the efficiency of several shift keying techniques, each one was subjected to varying levels of noise and the image obtained at the destination was observed. In almost all cases, the image was found to be distorted, one way or the other. The only difference was in the extent up to which each scenario had affected the process of the image transfer. As a move to progress further in this field, work must not just be restricted simulations on a single computer. In order to be more fruitful, it must be tried with actual USRP, under various architectures. Also, it is important to remember that several cases such as the existing Wi-Fi and Wi-Max networks are yet to be exposed to this technology and so, experimenting under such conditions must be emphasized.

IV REFERENCES

1. http://www.fcc.gov/oet/info/database/spectrum/
2. Hemalatha, M., Prithviraj, V., Jayalalitha, S., Thenmozhi, K., Bharadwaj, D., Koundinya, G., G., A survey report on spectrum sensing techniques in cognitive radio, Journal of Theoretical and Applied Information Technology 2012; 37 (1) : 32-38
3. Hemalatha, M., Prithviraj, V., Jayalalitha, S., Thenmozhi, MIMO in cognitive radio network – A survey, European Journal of Scientific Research 2012; 72 (2):169-175
4. D.Cabric, S.Mishra, R.Brodersen. Implementation issues in spectrum sensing for cognitive radios. Asilomar Conference on Signals, Systems and Computers 2004; 1: 772– 776
5. MitolaJ, Maguire, G. Q., Cognitive radio: making software radios more personal. IEEE Personal Communication Magazine 1999; 6 (4): 13–18.
6. Mitola, J.III. Cognitive radio: An integrated agent architecture for software defined radio. Ph.D dissertation. Royal Institute of Technology, Stockholm, Sweden 2000
7. A.Sahai, D.Cabric. Spectrum sensing : Fundamental limits and practical challenges. IEEE DySPANBaltimore, MD 2005
8. A.Ghasemi, E.S.Sousa. Spectrum sensing in cognitive radio networks: Requirements. challenges and design tradeoffs. IEEE communication Magazine 2008; 46 (4): 32-39
9. D.D.Ariananda, M.K.Lakshmanan, H.Nikookar. Survey on Spectrum Sensing Techniques for CR. Cognitive Radio and Advanced Spectrum Management CogART 2009; 74-79
10. Takeo Fujii. Effect of Sensing Ability for Spectrum Shared Cognitive Radio – Power Control or CSMA. The 3rd International Conference on Ubiquitous and Future Networks (IUFN) 2011; 363 - 368
11. S. Geirhofer, L. Tong, B. Sadler. Dynamic spectrum access in the time domain: Modeling and exploiting white space. IEEE Communication Magazine 2007; 45 (5): 66–72
12. M. Marcus. Unlicensed cognitive sharing of TV spectrum: the controversy at the federal communications commission. IEEE Communication Magazine 2005; 43 (5): 24-25
13. A. Ghasemi, E. Sousa. Optimization of spectrum sensing for opportunistic spectrum access in cognitive radio networks. IEEE Consumer Communication and Networking Conference 2007; 1022–1026
14. S. Mishra, A. Sahai, R. Brodersen. Cooperative sensing among cognitive radios. IEEE International Conference on Communication 2006; 2:1658–1663
15. J. Lehtomaki. Analysis of energy based signal detection. Ph.D. dissertation. University of Oulu. Finland 2005
16. Joseph Font-Segura, Xiaodong Wang. GLRT-Based Spectrum Sensing for Cognitive Radio with Prior Information. IEEE Transactions on Communications 2010; 58 (7): 2137-2146

17. Wendong H, Daniel W, Murad A, James G, George Vlantis, Mario Gerla, Adam Woliz. Dynamic Frequency Hopping Communities for Efficient IEEE 802.22 Operation. IEEE Communication Magazine 2007; 45 (5): 80-87
18. SudhirSrinivasa, Syed Ali Jafar. How much spectrum sharing is Optimal in Cognitive Radio Networks?. IEEE Transactions on Wireless Communication 2008; 7 (10):4010– 4018
19. Danda B. Rawat, Gongjun Yan. Spectrum Sensing Methods and Dynamic Spectrum Sharing in Cognitive Radio Networks: A Survey, International Journal of Research and Reviews in Wireless Sensor Networks 2011; 1-13
20. Ali Tajer, Xiaodong Wang. MultiuserDiversity Gain in Cognitive Networks.IEEE/ACM Transactions on Networking 2010; 18 (6): 1766-1779
21. Cong Shen, Michael P. Fitz. Opportunistic Spatial Orthogonalization and Its Application in Fading Cognitive Radio Networks. IEEE Journal of selected topics in signal processing 2011; 5 (1): 182-189
22. Tao luo, Feilin, Tao jiang, Mohsen Guizani, Wen chen. Multicarrier Modulation and Cooperative Communication in Multihop Cognitive Radio Networks. IEEE Transactions on Wireless Communications 2011; 18 (1): 38-45
23. Zhiguo Ding, Kin K. Leung, Dennis L. Goeckel, Don Towsley. Cooperative Transmission Protocols for Wireless Broadcast Channels. IEEE Transactions on Wireless Communications 2010; 9 (12): 3701-3713
24. Rui Zhang, Ying-Chang Liang. Exploiting Multi-Antennas for Opportunistic Spectrum Sharing in Cognitive Radio Networks. IEEE Journal of selected topics in signal processing 2008; 2 (1): 88-1