Cognitive Spectrum Management

Pooyan Amini, Daryl Wasden, Arash Farhang,
Ehsan Azarnasab, Peiman Amini, Behrouz Farhang-Boroujeny
University of Utah, Salt Lake City
{amini,wasden,arashb,azarnasa,pamini,farhang}@ece.utah.edu

ABSTRACT

The concept of cognitive radio has put together a vast area of expertise from signal processing and communication to data mining algorithms. This is possible because of the recent advances in development of Software Defined Radio (SDR). A cross layer approach in SDR network supplies the MAC layer with more information to take advantage of opportunistic channel access. One major requirement of cognitive radio is to be aware of other legacy radios and avoid using active channels. Using a basestation for channel allocation and node coordination, collaborative spectrum sensing (by filterbanks) determines the best estimate for background noise. Afterwards, the noise temperature is used to profile the presence of the active users in the spectrum based on the location and time of the experiment. A probabilistic approach identifies the model of the spectrum activity as arrival/departure of many narrowband radios in the spectrum. A game theoretic approach for spectrum selection is then developed which sets spectrum usage prices according to space-time statistics of spectrum activity pattern. Furthermore, using a filterbank multicarrier technique, cognitive nodes can keep their transmission power under the noise level in the active parts of the spectrum while filling the spectrum holes.

1. INTRODUCTION

Licensed frequency bands are not used efficiently everywhere and over time [1], [2]. The unutilized part of the spectrum is called a spectrum hole. The next generation of radios will be based on Cognitive Radio (CR) technologies, [3]. CR addresses the inefficiency of the frequency spectrum usage by allowing the coexistence of different radios in the shared unlicensed bands. CR utilizes the spectrum holes over time and space. Legacy devices are the Primary Users (PU) of the spectrum which have priority access to the spectrum. To address the spectrum underutilization issue, the FCC has allowed secondary users (SU) to share some previously dedicated bands under condition of minimal interference to legacy devices of the band [4]. For SU, spectrum holes are opportunities that need to be discovered and exploited [5]. To be transparent to the PU network, SU should operate under interference temperature.

CR should sense the spectrum to detect the activity of PU. Spectrum sensing for CR could be implemented using feature detection or energy detection. Energy detection is often the preferred method for simplicity and because it does not assume prior distribution for spectrum usage. However, one problem of energy sensing is to define a reliable energy threshold which indicates an active PU. Cognitive nodes are required to keep their power level below this threshold over active bands.

Emulab [6] is being used to evaluate the available spectrum. Emulab is a network containing many different types of nodes. The nodes used for this project are those equipped with USRP (Universal Software Radio Peripheral) devices. The software platform for USRP devices is GNU Radio [7]. This software provides users with many different programs to both transmit and receive signals and allows users to write their own programs or modify existing programs as they see fit. Using GNU Radio, the Emulab's USRP nodes were programmed to take measurements of the spectrum by modifying existing programs in the GNU Radio package. This program measures the magnitude and phase of the spectrum over a specified bandwidth. The Emulab requires one to start an experiment by creating what is known as an NS (Network Simulator) file. By creating an NS file, measurements are made on up to 16 different nodes. Within the NS file, program agents (specified lists of programs to be run) have been defined to take measurements on multiple nodes simultaneously. Software defined radio [8] simplifies the digital processing algorithms needed for noise reduction.

The distributed spectrum sensing mechanism reliably determines the background noise level [3]. After the noise floor is detected, we use a threshold above the noise floor to detect the power of the active PU with 95% certainty. The higher the detection threshold the more accurate the model is, but at the same time higher threshold decreases the effective dynamic range of spectrum measurement. The filterbank based power detection technique [9] enables us to initially model the activity of the PU as a renewal process with two distributions for presence and absence of the PU similar to [5]. Higher order statistics can extract some useful information about PU activity. The general random process based modeling improves the cognitive spectrum assignment marginally because it lacks a priori knowledge about individual channels. For example, a channel used by handset devices more resembles a Poisson process than a channel used by a TV station. Handset users may come and leave the channel with exponential inter-arrival times but a TV station broadcasts continually during certain times of the week.

The spectrum assessment algorithm is provided with channel templates containing customized models for each channel. A learning algorithm is thus being devised to fuse the spectrum measurement results with each channel usage template over time. The final spectrum analysis result is saved inside a local database that can be queried by cognitive nodes over the internet.

The problem of spectrum allocation is then modeled using a game theoretic approach. A potential game model is used for this problem and as a result if the users make unilateral decisions the play will converge to a Nash Equilibrium (NE) and it will have a steady-state solution [10]. An action space and a utility function will be defined for this game. The system will be a part of the Software Radio Smart Radio Challenge 2008 [11].

The rest of the paper is organized as follows. A brief introduction to noise temperature detection and distributed channel sensing is presented in Section 2. The algorithm for frequency selection is given in Section 3. We finally talk about the implementation status and draw our conclusions in Section 4.

2. Distributed Spectrum Sensing

One advantage of collaboration among cognitive radios is to share the sensing information for better estimation of the spectrum activities. A base station can fuse the spectrum sensing results for estimation of the background noise. Noise temperature [3] is calculated in an optimal detector [12]. A cognitive radio network may contain many nodes. Each node can sense the spectrum independently of the others with the aid of a filterbank or some other spectral estimator. The purpose of this experiment is to demonstrate through measured data the effectiveness of a distributed sensing algorithm that is described in the next paragraph. The Emulab network of nodes will be used for taking measurements. The Emulab is a network of radio nodes used to run experiments by researchers throughout the world. Currently, the Emulab has 16 Universal Software Radio Peripheral (USRP) devices, 14 of which are active. These devices will be used to gather samples for analysis on a computer. This data will be used to define a probabilistic model of spectrum space-time usage. The Emulab nodes are located in rooms throughout the Merrill Engineering Building (MEB) on the campus of the University of Utah in Salt Lake City, Utah. Figure 1 shows the placement of these nodes.

The algorithm being tested begins by using K complex time samples and dividing the signal at Γ different nodes into sufficiently small slices of spectrum such that each narrow band signal can be considered a single sinusoid multiplied by some gain h. Some noise is present such that the received signal is the transmitted signal multiplied by the channel transfer function plus a random noise variable V. The received signal at one frequency is expressed in the equation $\mathbf{X} = \mathbf{h}\mathbf{s}^H + \mathbf{V}$. $\mathbf{X}$ and $\mathbf{V}$ are Γ-by-K matrices. $\mathbf{V}$ is a matrix of independent identically-distributed random variables (the noise). We strive to estimate the signal power (including channel effects) and the noise power. The best estimate would be a time average of each quantity (i.e. $(1/K) \times \sum |s_k|^2$, $(1/\Gamma) \times \sum |h_\gamma|^2$, and $(1/\Gamma K) \times \sum |v_{\gamma k}|^2$). Since $\mathbf{s}$, $\mathbf{h}$, and $\mathbf{V}$ are unknown, it is impossible to calculate these quantities directly. We can calculate estimates indirectly because $\mathbf{V}$ is a zero mean independent identically-distributed random variable. From these results, we can obtain an estimate of the signal power by first forming the K-by-K matrix $\mathbf{R} = E[\mathbf{X}^H\mathbf{X}]$; then finding its largest eigenvalue, λ_0; and finally using the following equations to obtain estimates of the signal power and the noise power [12].

$$\hat{P}_{\mathrm{sig}} = \frac{K\lambda_0 - \mathrm{trace}[\mathbf{R}]}{\Gamma K(K-1)} \quad (1)$$

$$\hat{P}_{\mathrm{noise}} = \frac{\mathrm{trace}[\mathbf{R}] - \lambda_0}{\Gamma(K-1)}. \quad (2)$$

Originally, the script *usrp_spectrum_sense.py* from the GNU Radio distribution was modified and used to sense the spectrum from about 850 MHz to 950 MHz. It was determined that this method was unreliable for analyzing the spectrum activity. The file performed a large amount of filtering and processing before converting to the frequency domain for spectrum analysis, and there was no way to determine the synchronization between measurements at differing nodes. Due to these shortcomings, it was determined that post-processing could be done in Matlab more efficiently. For this, time samples were needed and not just the spectral estimates provided by *usrp_spectrum_sense.py*.

To truly test the accuracy of the distributed sensing algorithm, synchronization of the time samples is required. Synchronization between nodes is not a trivial task. When the experiment begins, each Emulab node being used is loaded with the Linux operating system. Along with it, several useful programs are loaded to assist in synchronizing the nodes.

Several different options were considered for synchronization of the nodes. Among these, two are worth mentioning. First, the open source program *ntpd* was used to synchronize the nodes and keep them synchronized. According to the official documentation, *ntpd* can synchronize the system clocks to within one millisecond. An entry in the *crontab* (an application scheduler) would then allow the measurement script to run, and measurements could be automated to be taken at synchronized times to within one millisecond. This would have to be repeated at different times for different bands since only a portion of the spectrum can be measured at a time (due to hardware limitations). Because of these limitations, it was concluded that more synchronous measurements could be achieved by the second method.

The second method consists of using a program titled *emulab-sync* which is a simple distributed synchronization client available on all of the Emulab nodes. Using this method, there is a master node and several slaves. The slave nodes wait until the master node tells them to start taking measurements before beginning. The times that these measurements occur is recorded. It has been verified that these recorded times are not more than one millisecond apart (usually they are less). This appears to be the most reliable method of synchronization available at the present

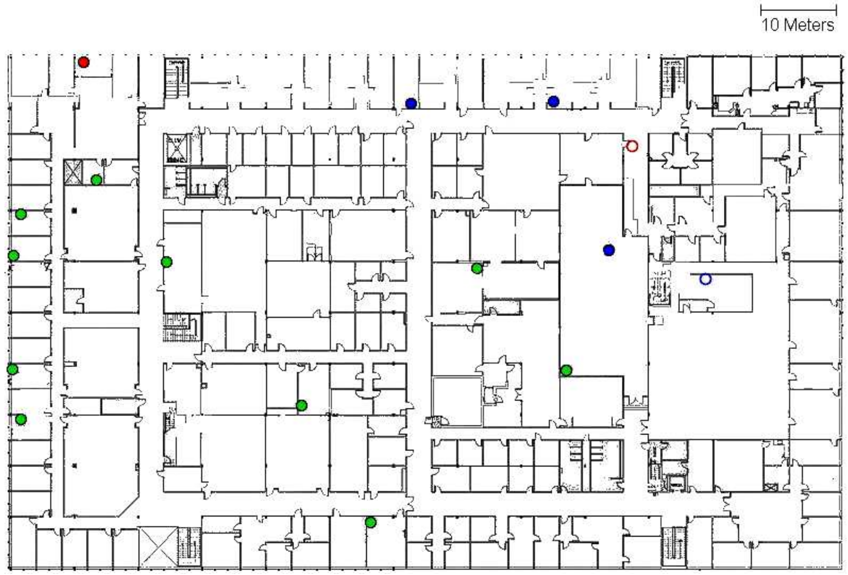

Fig. 1. The placement of the Emulab nodes with USRP devices in the MEB. The green circles represent available nodes. The red circles represent nodes that are out-of-service. The blue circles represent nodes that are in-use. The unfilled circles are the nodes on a different floor than the map.

time. The script file *usrp_rx_cfile.py* that is part of the GNU Radio project was modified and renamed *usrp_rx_fscan.py* to automate the measurement process, utilizing the *emulab-sync* program to synchronize the measurements as much as possible.

The measurements taken with *usrp_rx_fscan.py* are complex time samples that have been demodulated from their center frequency to the baseband. They include 4 MHz of bandwidth for each frequency measured. Assuming that these measurements are synchronized to within one millisecond, the narrow band signals must be much smaller than one kilohertz for justification of the assumption that the narrow band signal gain is constant over its bandwidth. This data will pass through a uniformly modulated DFT filterbank to obtain an estimate of the signal power over narrow bands (much smaller than one kilohertz). Then, the data from all of the nodes will be combined and processed via the algorithm described above. This experiment shall demonstrate the efficacy of the algorithm in improving the results of a distributed spectrum sensing network.

Since a sufficient amount of data for analysis is just beginning to be compiled using the *usrp_rx_fscan.py* script at the writing of this paper, quantitative measurements using experimental data are unavailable. To illustrate that the algorithm works based on the proposed assumptions, a simulation was created in Matlab. The simulation presupposes sufficiently narrow bands and time synchronization of the nodes' measurements.

At the outset of the simulation, a vector $\mathbf{s}$ that contains the source signals at each frequency is created. Each signal is represented internally by a complex number with magnitude

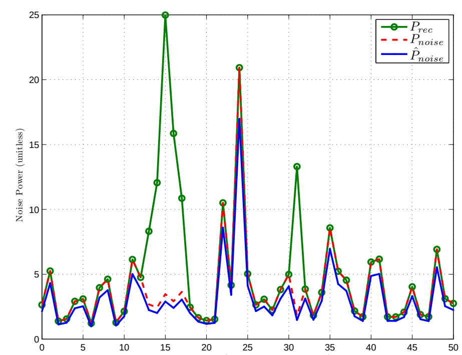

Fig. 2. The received power, the actual noise power at the receiver, and the estimated noise power

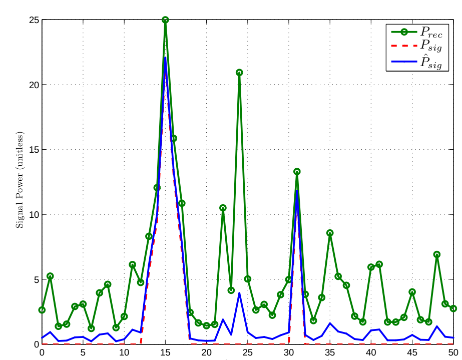

Fig. 3. The received power, the actual signal power at the receiver, and the estimated signal power

and phase. Also, the transfer function h from each signal source to the each node is randomly generated for each frequency (also represented by a complex number). Then the program runs through a loop over all the frequencies. For each frequency in the loop, a noise level is randomly determined and a matrix of normally distributed noise $\mathbf{V}$ is generated. Then, the received signal is calculated using $\mathbf{X} = \mathbf{h}\mathbf{s}^H + \mathbf{V}$. Relevant quantities are recorded in vectors for graphing. These include: the actual signal power, calculated by taking a time average of the magnitude squared of the elements of $\mathbf{s}$ at a specific frequency; the signal power estimate, calculated from the equation for $\hat{P}_{\text{sig}}$; the actual noise power, calculated by taking the time average of $\mathbf{V}$ over all nodes and times; the noise power estimate, calculated from the equation for $\hat{P}_{\text{noise}}$; and the total received power (or simply the received power), calculated by taking the time average of $\mathbf{X}$ over all nodes and times. These were recorded for each of the 51 frequencies used in the simulation. The results are graphed and shown in figures 2 and 3.

For this simulation, five nodes were simulated with 100 sets of measurements taken at differing times (synchronized at each node). Signals with various levels of power were generated at the following frequency indices: 13, 14, 15, 16, 17, and 31. All other signals were zero (considered inactive). Noise was generated at all frequencies. The noise is shown in Fig. 3. Notice in Fig. 2 that the algorithm is able to reduce the peaks of noise at 22 through 25 even though these peaks are comparable to the power being transmitted by the signals. In Fig. 3, it is shown that the noise estimate recognizes this peak as noise and attenuates peaks at 13 through 17 and 31.

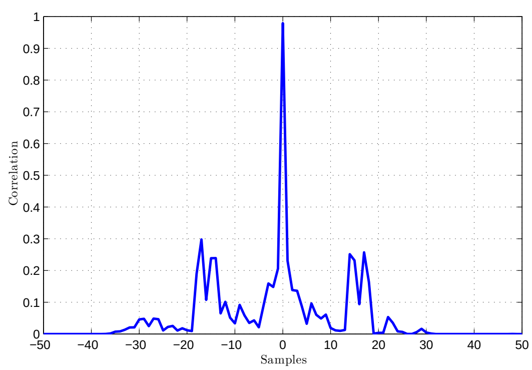

Fig. 4. Normalized correlation of the actual signal power with the signal power estimate

Another plot used to analyze the effectiveness of the algorithm was the normalized cross correlation of the actual signal power with the calculated signal power estimate. This is shown in Fig. 4. It almost exhibits the symmetry one would expect from an autocorrelation function, and its value at zero is 0.98 (almost unity, a perfect match). Fig. 5 shows the crosscorrelation function of the received power (before

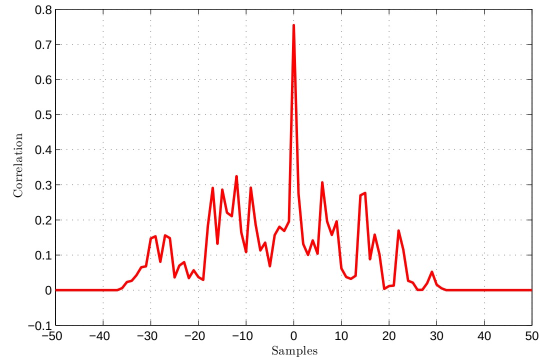

Fig. 5. Normalized correlation of the actual signal power with the average of signal power of the nodes

processing) and the actual underlying signal power. The central peak is approximately 0.75 or 0.23 less than the estimate in Fig. 4. The plot itself doesn't seem to be close to the symmetrical shape one would expect if the received power were a true mirror of the underlying signal power.

These results are satisfying, but more work remains to be done. When sufficient data has been compiled, the algorithm will be applied to the data. The results will be tabulated and graphed. As shown in the simulation, this should demonstrate a noticeable improvement in distinguishing signal power from noise compared to using the received signal power at one node alone. Hence, the proposed algorithm will greatly improve the accuracy of a spectrum sensing model implemented in a cognitive radio network.

3. Spectrum Selection Algorithm

The spectrum assignment in our system has a distributed algorithm and each cognitive radio node selects its own frequency band based on the probability of availability of that frequency band. Game theory is used for this decision making process.

Game theory is a mathematical framework which is used to analyze the choices of players when the outcome for each player depends on both his choice and the choices of other players. This is similar to what we have in a cognitive radio where the outcome for each node depends on the choice of itself and the choices of the other nodes.

A game has three basic elements. These elements consist of a set of players, an action space, and a utility function.The set of players is denoted by N and is usually considered to be equal to $1, 2, \ldots, |N|$,where $|N|$ is equal to the number of players. Each player i has a set of available actions A_i. The action space A is formed by calculating the cartesian product of the sets of actions of all players. Each member of A is called an action profile. Each player has a utility function u_i which maps the action space to the real numbers. The players try to maximize their utility functions.

A Nash Equilibrium (NE) is an action profile for which unilateral deviation from this action profile will not result in a higher utility. In other words, $\mathbf{x} = (x_1, x_2, ..., x_N) \in A$ is a Nash Equilibrium if

$$u_k(\mathbf{x}) \geq u_k(\mathbf{x}_{-k}/x'_k), \forall x'_k \in, \forall k \in N \tag{3}$$

where $\mathbf{x}_{-k} = (x_1, x_2, ..., x_{k-1}, x_{k+1}, x_{k+2}, \ldots, x_N)$ and $(\mathbf{x}_{-k}/x'_k) = (x_1, x_2, \ldots, x_{k-1}, x'_k, x_{k+1}, x_{k+2}, \ldots, x_N)$.

When a NE is reached, the players are not willing to change their decisions hence each NE can be considered as a steady state solution for the game. A game may not have a NE, and if a game has a NE the players actions might not converge. However, for specific types of games there always exists a NE, and the players actions will always converge.

Potential games are one type of game for which the existence of NE and convergence of the player actions is guaranteed [13] [14]. A game is called a potential game if there exists a function $v : A \rightarrow \mathbb{R}$ such that

$$v(\mathbf{x}_{-i}/x'_i) - v(\mathbf{x}) = u_i(\mathbf{x}_{-i}/x'_i) - u_i(\mathbf{x}). \tag{4}$$

The problem of dynamic frequency selection has been modeled by potential games [15].

For our problem, a game G is defined. It will be shown that G is a potential game. The player set N of G consists of the cognitive radios, and the action set of player $i \in N$ is F_i where F_i is the set of available frequencies which that player can use. Unlike [15] for our problem the occupancy probability of every frequency band is known by each of the cognitive nodes and hence we can define the utility function in a different form. Therefore, PU can be considered an active participant in the game. The utility function is defined as:

$$u_i(f_1, f_2, \ldots, f_i, f_{i+1}, \ldots, f_N) = \begin{cases} Q_{f_i} & \text{if } f_i \neq f_j, \forall j \neq i \\ 0 & \text{Otherwise} \end{cases} \tag{5}$$

where Q_{f_i} is the probability of the frequency band f_i being available.

In order to show that G is a potential game. The potential function v is defined in the following way:

$$v(f_1, f_2, \ldots, f_N) = \sum_{i=1}^{i=N} \sum_{j=1}^{\sigma(f_i)} C_{f_i}(j) \tag{6}$$

where $\sigma(f_i)$ is the number of players using the frequency band f_i, and C_{f_i} is equal to

$$C_{f_i}(j) = \begin{cases} Q_{f_i} & \text{if } j = 1 \\ 0 & \text{Otherwise} \end{cases} . \tag{7}$$

In order to demonstrate that this function is a potential function for G, it must be shown that if a player i changes his choice unilaterally the change in his utility function is equal to the change in the potential function. Four cases should be considered. The first case is when neither the current choice f'_i nor the previous choice f_i is occupied. The second case is when f'_i is occupied but f_i is not. The third case is when f'_i is unoccupied but f_i is occupied. The final case is when both f'_i and f_i are occupied. The following equations show that the change in the utility function is equal to the change in the potential function using equation (5).

1) $\forall\ j \neq i,\ f_i \neq f_j$ and $f'_i \neq f_j$

$$u_i(\mathbf{f}_{-i}/f'_i) - u_i(\mathbf{f}) = Q_{f_i} - Q_{f'_i} = v(\mathbf{f}_{-i}/f'_i) - v(\mathbf{f}). \tag{8}$$

2) $\exists\ j \neq i, f_i = f_j$ and $\forall\ k \neq i, f'_i \neq f_k$

$$u_i(\mathbf{f}_{-i}/f'_i) - u_i(\mathbf{f}) = 0 - Q_{f'_i} = v(\mathbf{f}_{-i}/f'_i) - v(\mathbf{f}). \tag{9}$$

3) $\exists\ j \neq i, f'_i = f_j$ and $\forall\ k \neq i, f_i \neq f_k$

$$u_i(\mathbf{f}_{-i}/f'_i) - u_i(\mathbf{f}) = Q_{f_i} - 0 = v(\mathbf{f}_{-i}/f'_i) - v(\mathbf{f}). \tag{10}$$

4) $\exists\ j \neq i, f'_i = f_j$ and $\exists\ k \neq i, f_i = f_k$

$$u_i(\mathbf{f}_{-i}/f'_i) - u_i(\mathbf{f}) = 0 - 0 = v(\mathbf{f}_{-i}/f'_i) - v(\mathbf{f}). \tag{11}$$

Equations (8) to (11) show that v is a potential function for G which will result in having a game which will eventually converge to a NE after a finite number of steps. The numerical analysis of this method will be demonstrated during SDR challenge meeting.

4. Status of Implementation, Conclusion and Future Research

The need for a reliable background noise detection to automate the detection of PU in a cognitive node [16] was addressed in this paper. The noise floor detection and spectrum sensing are improved using the distributed sensing technique discussed in the previous sections. This was demonstrated through a computer simulation in Matlab. We are currently working on the probabilistic modeling of PU activity and also the spectrum selection algorithm.

Acknowledgment

We would like to thank the FLUX Research Group for letting us use their facility at Emulab to setup and run our experiments.

5. REFERENCES

[1] R. Brodersen, A. Wolisz, D. Cabric, S. Mishra, and D. Willkomm, "CORVUS: A cognitive radio approach for usage of virtual unlicensed spectrum," White paper, Berkeley, Available from `http://bwrc.eecs.berkeley.edu/Research/MCMA/CR_White_paper_final1.pdf`, Tech. Rep., July 2004.

[2] N. Shankar, C. Cordeiro, and K. Challapali, "Spectrum agile radios: utilization and sensing architectures," *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pp. 160–169, November 2005.

[3] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, February 2005.

[4] F. C. Commission, "Spectrum Policy Task Force," ET Docket 02-135, Nov 2002.

[5] H. Kim and K. G. Shin, "Efficient Discovery of Spectrum Opportunities with MAC-Layer Sensing in Cognitive Radio Networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 5, pp. 533–545, May 2008.

[6] "Emulab network simulation testbed, university of utah." [Online]. Available: http://www.emulab.net/

[7] "GNU Radio." [Online]. Available: http://www.gnu.org/software/gnuradio/

[8] B. Farhang-Boroujeny, *Signal Processing Techniques for Software Radios*. Morrisville, North Carolina: Lulu Publishing House, 2008.

[9] P. Amini, R. Kempter, R. R. Chen, L. Lin, and B. Farhang-Boroujeny, "Filter bank multitone: A physical layer candidate for cognitive radios," *Software Defined Radio Technical Conference*, November 2005.

[10] J. Neel, J. Reed, and R. Gilles, "Game models for cognitive radio algorithm analysis," *Software Defined Radio Technical Conference*, November 2004.

[11] "Smart radio challenge." [Online]. Available: http://www.radiochallenge.org/

[12] B. Farhang-Boroujeny, *Handbook on Array Processing and Sensor Networks*. John Wiley & Sons, ch. Spectral Estimation in Cognitive Radios, under preparation.

[13] J. Hicks, A. MacKenzie, J. Neel, and J. Reed, "A game theory perspective on interference avoidance," *Globecom2004*, November 2004.

[14] M. Voorneveld, "Potential games and interactive decisions with multiple criteria," Ph.D. dissertation, Tilburg University, 1996.

[15] J. Neel and J. Reed, "Performance of distributed dynamic frequency selection schemes for interference reducing networks," *Milcom 2006*, October 2006.

[16] P. Amini, E. Azarnasab, S. Akoum, X. Mao, H. I. Rao, and B. Farhang-Boroujeny, "Implementation of a cognitive radio modem," *Software Defined Radio Technical Conference*, November 2007.