

Real-time Wide-band Spectrum Sensing for Cognitive Radio

Wei liu, Opher Yaron, Ingrid Moerman, Stefan Bouckaert, Bart Jooris, Piet Demeester
 Department of Information Technology
 Internet Based Communication Networks and Services (IBCN)
 Ghent University - IBBT
 Gaston Crommenlaan 8 Bus 201, B-9050 Gent, Belgium

Abstract—Cognitive radio has received considerable amount of attention as a promising technique to provide dynamic spectrum allocation. Wide-band spectrum sensing is the corner stone for cognitive radio to be functional. Most existing commercial sensing solutions lack either the required flexibility or speed. Software-defined radio (SDR) on the other hand offers very high flexibility and therefore becomes a common platform for CR implementation. Among various SDR platforms, the universal software-defined radio peripheral (USRP) gained broad popularity. This paper presents a real-time wide-band-capable spectrum sensing solution based on USRP. The concept of energy detection and the methodology for wide-band sensing are explained. Finally, the performance of the proposed sensing solution is verified and compared with another popular commercial sensing solution, Airmagnet.

I. INTRODUCTION

As wireless communication technology evolves rapidly, radio spectrum resources become ever more crowded. Cognitive radio (CR) is introduced as a promising technology to improve the efficiency of spectrum utilization by enabling nodes to adapt their transmission parameters to the local spectrum environment [1]. This leads to new challenges in the field of spectrum sensing. The critical design problem is the methodology of processing multi-gigahertz wide bandwidth in real time [2]. Due to the reconfigurability required by cognitive radio, software-defined radio (SDR) becomes a common platform where CR can be implemented [1]. As defined in [3], “software radio” represents radio functionalities defined by software, which comes down to implementing functions in software that are traditionally implemented in hardware.

SDR can be divided into two major categories based on the type of processor used for signal processing. The first category makes use of a general purpose processor (GPP) in a regular PC and the other category typically has a powerful embedded processor on board. Compared to GPP-based SDR, SDR with embedded processors has higher processing speed and lower latency but are also more expensive and difficult to design and debug.

The Universal Software Radio Peripheral (USRP) developed by Ettus Research [4] is a low-cost SDR platform that utilizes a general purpose processor and has gained widespread usage. USRP consists of two parts, a fixed mother board and a plug-in daughter board. The mother board mainly contains ADC/DAC, an FPGA mainly for digital down sampling with

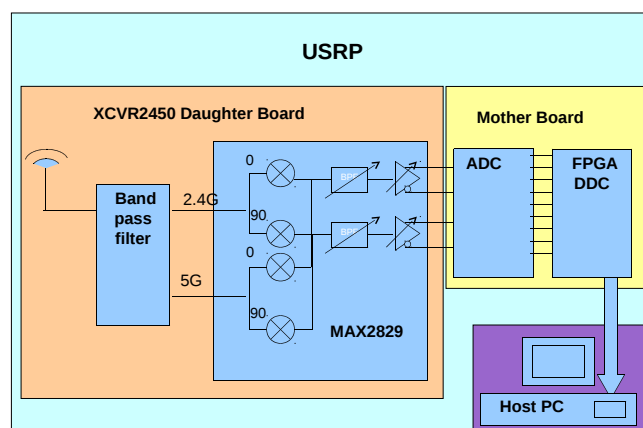


Fig. 1: USRP Block Diagram

programmable decimation rate and an interface connected to host PC. The daughter board provides basic RF front-end functionality. USRP2 — the second generation of USRP — outperforms the original in its more powerful FPGA, faster ADC/DAC and Gigabit Ethernet host connection. A simplified diagram of USRP2 with XCVR2450 daughter board is illustrated in Fig1.

The XCVR2450 is a daughter board that covers the 2.4 and 5 GHz ISM bands and has a configurable analog front-end filter with maximum bandwidth of 30 MHz. Since most commercial sensing solutions such as Airmagnet and Wispy are also limited to the ISM bands, we use this board for implementing our spectrum sensing solution. The identical frequency coverage makes it more meaningful to test and compare with other commercial solutions.

Besides the hardware, Ettus Research also provides the universal hardware driver (UHD) for communication between USRP and host PC [5]. It is available for all major platforms including Linux, Windows, and can be built with many popular compilers such as GCC. Users are able to use the UHD driver standalone or with 3rd party applications such as GNU Radio.

GNU Radio is by far the most well-known software platform to work with USRP. It is an open-source software providing various signal processing blocks accompanied with graphical user interface. Other software platforms such as Simulink and Labview are also readily available [14], [15]. The platform

selected here is Iris, which is a software platform developed by Trinity College Dublin. It has similar component structure as GNU Radio, but is more suitable for reconfigurability on the fly [6]. Both GNU Radio and Iris utilize UHD driver and firmware to communicate with USRP. Compared to GNU Radio, Iris is more transparent due to its simple structure, and hence easier to get access to low level parameters on the hardware. This high transparency and reconfigurability are more desired in our context, hence we selected this platform.

This paper presents a USRP-Iris based sensing solution, and points out several important aspects of spectrum analyzing. Section II reviews existing algorithms and techniques for wide-band spectrum sensing. The implementation of the sensing solution is introduced in depth in Section III. Section IV first examines the proposed solution by comparing it with a quasi-optimal algorithm in matlab, afterwards a set of measurements for comparing the sensing performance with a commercial product — Airmagnet is described. Section V gives theoretical analysis on sensitivity of USRP front-end. Finally some conclusions are drawn in Section VI.

II. EXISTING SENSING ALGORITHMS

A thorough list of sensing algorithms is presented in [7]. Some are more dedicated for specific signals while others are more general; Some have high complexity and good performance while others might be just simple and fast. It is up to the designers to pick what is best suited for a specific goal.

Matched filtering is known as the optimum method for detection of primary users when the transmitted signal is known. However it requires the radio to demodulate the signal, hence needs perfect knowledge of the primary signal. As a result, the receiver’s complexity is proportional to the number of signal types that need to be detected. Therefore it is not suitable for wide-band general purpose sensing.

Waveform based sensing is a method which makes use of certain known patterns in wireless communications. Such patterns include preambles, regularly transmitted pilot patterns, spreading sequences, etc.. Those patterns are usually utilized in wireless communication systems to assist synchronization. Sensing is performed by correlating the received signal with a certain known pattern. This method is only applicable to systems with known patterns and requires the receiver to be aware of those patterns. So certain level of a-priori knowledge is necessary.

Cyclostationary feature detection is a method for detecting primary users by exploiting the cyclostationary features of the primary signal. There is no need to demodulate the signal hence it requires less a-priori knowledge compared with the previous methods. The main advantage here is the ability to distinguish noise from primary user’s signal.

Energy detection based sensing is the most common way of spectrum sensing due to its low computational and implementation complexity. The major advantage is that no a-priori knowledge is required. In the ISM context, all users share the same right to use the spectrum resource. Hence users are

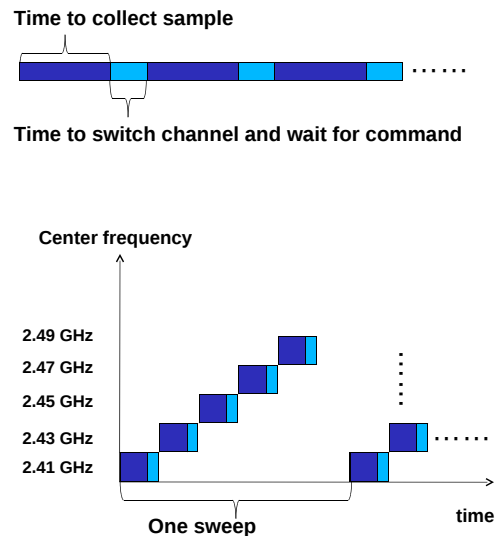


Fig. 2: Sweep

typically interested only in finding a channel with good quality to communicate. This is the concept of horizontal sharing of spectrum, compared with the vertical sharing in licensed bands. Energy detection is best suitable for fast channel quality evaluation and is easy to implement, therefore it is selected to implement our sensing solution.

Many narrow-band sensing algorithms exist, but when it comes to wide-band sensing, much less work has been done. Due to the limitation of ADC and filtering, it is sometimes not possible to sense the entire bandwidth of interest at once. There are methodologies for sensing multiple spectrum bands simultaneously, making use of advanced signal processing techniques [10], [8]. Another style of solution is distributed sensing, covering large bandwidth by multiple devices co-operating with each other. Multi-band joint detection (MJD) proposed in [9] is such a solution.

III. IMPLEMENTATION

USRP2 has powerful ADC, capable of IQ sampling, 100 MHz sample rate and 14 bit resolution for each I or Q sample. But the Gigabit Ethernet link only allows 25 Msps to be passed to the host PC. The most practical way to construct a wide-band spectrum based on energy detection is by tuning the mixer’s center frequency on the RF front-end, e.g. sensing one band at a time and then combining the measurements.

A complete scan of the desired frequency range is called one sweep. This is illustrated in Fig 2. The sweep time should be as short as possible so that the combined results from each center frequency during one sweep can be considered to refer to the same time instance. According to Nyquist theory, the covered bandwidth of Fourier Transform depends on the sample rate. When using complex samples, the covered bandwidth is the same as the sample rate instead of half of the sample rate if real samples are provided. Hence by default the maximum sample rate of 25 Mhz is selected to achieve minimum sweeping time.

Some practical concerns during implementation are discussed below. In reality every time when the USRP switches its operating frequency, it takes time for the host PC to issue a configuration command and the hardware on the front-end to settle down. Hence it is impossible to collect samples continuously. According to the data sheet of the analog chip MAX2829 used on XCVR2450 [11], it takes typically 25 us for channel switching operation in the 2.4 GHz frequency band. However, if the pause would only be the 25 us caused by channel switching, the host PC won't be able to follow the huge data rate from USRP. On top of that the host PC can not control the timing of channel switching accurate enough. If we count the number of samples collected on the host PC and issue the channel switching command when enough samples are collected, then when this message reaches USRP, there are already many more samples streamed to the host PC. It is difficult to tell under which center frequency they are collected and therefore can not be used. Hence the streaming mode of USRP is set to non-continuous mode. The selected channel and the number of required samples are given at the beginning of each streaming session. Once this amount is reached, USRP stops streaming automatically. Then it waits for further commands to switch channel and start again. Another phenomenon is that the first batch of samples arriving from USRP right after tuning frequency or starting up has a very strong DC level. The obtained spectrum from those samples does not reflect the actual environment and hence can not be used. Therefore the first batch of samples are streamed to the host PC and then dropped in the first software component.

We now move to the details of the implementation. We adopt the well-known periodogram algorithm. The advantage of using periodogram rather than performing FFT directly is that it offers a convenient way to control the desired type of spectrum as well as the trade-off between time and frequency resolution. First we collect a certain number of complex samples under a certain center frequency, to be denoted by X . These X samples are then divided into blocks of size N . Each block has 50% overlap with its two adjacent blocks. To avoid introducing high frequency components that are not present in the original signal, each block is multiplied by an equal-size Hamming window. For each windowed block, FFT is performed and the power spectrum density (PSD) is calculated. Finally, the PSD results from all blocks are averaged to produce one clean PSD. When the block size N is smaller, for a given sample size X , the resulting PSD is averaged over more blocks, and therefore it is smoother. This is more suitable for detecting wide-band or stationary signals, but fast and narrow signals are less visible due to the averaging effect and insufficient frequency resolution. If detecting narrow-band signals is more important, N can be increased to achieve better frequency resolution. However, this is not always desirable, since the resulting spectrum is more noisy, which makes it harder to recognize the real signal of interest. The block size N is obviously limited by X , and when it reaches X the periodogram reduces to FFT.

When even better frequency resolution is needed, X has to be increased. This implies longer sample collection time, which compromises the time resolution.

This process, of collecting X samples and calculating the narrow band PSD needs to be repeated for the next center frequency. The question is what size frequency hopping step to take. As previously mentioned, the spectrum obtained for each center frequency covers 25 MHz. However, due to the Hamming window used in periodogram, the samples at the two edges of each block are attenuated, which attenuates the high frequency components in the produced PSD. To overcome this irregularity, we use a 20% overlap in the frequency domain, which means that the difference between consecutive center frequencies is 20 Mhz instead of 25 Mhz. This amount of overlap is sufficient to cover the attenuated spectrum at the edges of each block.

After the spectrum for all center frequencies is obtained, these separate pieces of spectrum need to be combined into one continuous spectrum. When assembling the spectrum, the edges of each PSD block beyond $\pm 10MHz$ around the center frequency are dropped, only the middle 20 MHz part of the spectrum appeared in the final result.

In order to provide flexibility, a number of parameters are exposed to the user. The frequency range can be configured by setting the lower and upper frequency boundaries. The parameters X and N of the periodogram can be used to configure the style and quality of the spectrum. The time resolution can be controlled as well, by configuring the number of sweeps per second. There is an upper limit, however, for this parameter due to the time it takes to complete one sweep. This time depends on several factors: the bandwidth of interest, the values of the parameters X and N , and the processing power of the host PC. If the target sweep time is too short, the system will just sweep as quickly as possible. In order to expose to the user the actual sweep time, an accurate time stamp of each sweep is recorded in the Iris log file.

For example, if we want to monitor the 2.4 to 2.5 GHz ISM band, we will define 2.4 GHz as the starting frequency, and 2.5GHz as the stopping frequency. Based on this input, the program will use 5 center frequencies to cover the whole range: 2.41 GHz, 2.43 GHz, ..., 2.49 GHz. If we choose X to be 2048 and N to be 256, then each PSD segment will contain 256 frequency bins. Due to the overlap in frequency domain, only about 80% of each PSD segment are used, hence in total there should be $256 * 0.8 * 5 = 1024$ PSD bins. For implementation reason, equal amount of bins should be removed from all blocks. Since there are 5 blocks, we remove 52 bins from each block and obtain 1020 bins in total. The resulting frequency resolution can be calculated as $\frac{100Mhz}{1020} = 98KHz$. Suitable selections of parameter values are mainly obtained by trial and error. For the 2.4-2.5 GHz ISM band we discovered that collecting 2048 samples per center frequency and producing 256 PSD bins give relatively good frequency resolution and short sweep time.

It is also possible to change the resolution bandwidth to

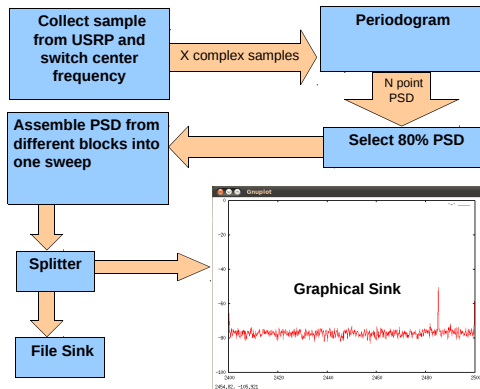


Fig. 3: Iris Component Diagram

less than 25 MHz, with corresponding frequency step size of less than 20 MHz. There are a few step sizes available as configuration options, depending on the required frequency span. The flexibility in step size and frequency span enable our sensing solution to be used with signals of various bandwidths such as Zigbee, WiFi or Bluetooth. Another advantage is the possibility to zoom-in to a specific part of the spectrum. The zoom-in is realized during the actual sampling process rather than just graphical processing.

The architecture of the software within Iris is illustrated in Fig 3. The first component at the top-left of the diagram directly interfaces with USRP. It uses UHD driver to collect samples from USRP and control the operational frequency. The samples are passed to the following components for all the necessary processing as described above. when processing is complete, the calculated spectrum data is presented graphically in real time, and in parallel recorded in a file, for further processing.

IV. VERIFICATION

In order to compare the spectrum obtained by our USRP implementation with the actual spectrum, we build a Matlab model. The first part of the model generates the desired input signal. The second part of the model implements two spectrum sensing algorithms - one is our USRP algorithm, the other is a quasi-optimal algorithm — it fixes its center frequency at 2.45 GHz, right at the middle of the 2.4 GHz ISM band, and covers the entire bandwidth by one single FFT, with no sweeping at all. The model of our USRP implementation includes down sampling by cascaded CIC and half-band filters, identically to the hardware implementation of the USRP's FPGA logic.

First, we examine the algorithm with a 10 MHz wide OFDM signal as input. The result of the simulation is shown in part (a) in Fig 4. The red line indicates the spectrum resulting from our USRP algorithm, the blue line represents the result of the quasi-optimal algorithm. It is obvious that with our algorithm there is always some discontinuity at the edge of blocks. This is partially due to the fact that the overlapping parts of adjacent blocks are dropped. More importantly, a window function is necessary when creating a periodogram, and any window

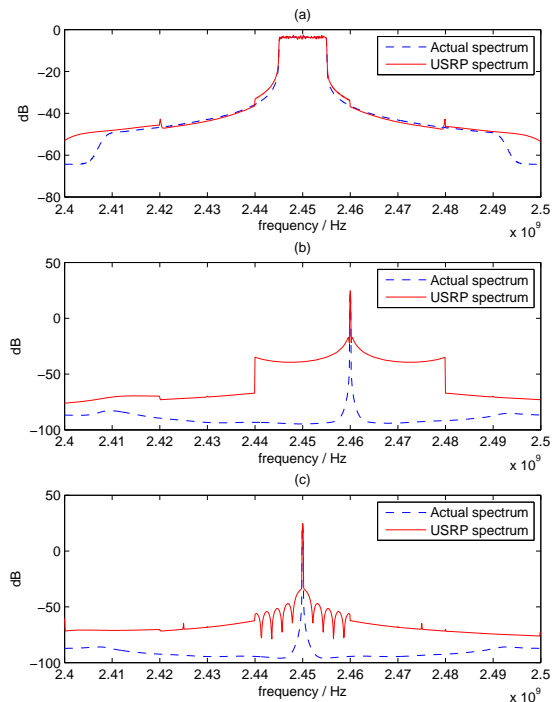


Fig. 4: Comparison of Spectra

function always has certain amount of leakage and smearing. This effect increases the noise floor within the frequency block where signal is present. To quantify the relative difference between the two algorithms, we divide the total energy of the difference between the two spectra, by the total energy of the quasi-optimal spectrum, as follows:

$$\frac{\int |PSD_{quasi}(f) - PSD_{usrp}(f)| df}{\int PSD_{quasi}(f) df}$$

The result we get is 7%, which is rather small.

To gain more insight into the actual influence of sweeping, we examine the behavior of our algorithm with a pure carrier wave as input at different positions within a frequency block — once on the edge and once in the middle. The simulation results are shown in part (b) and part (c) in Fig 4 respectively. We notice an abrupt change in noise floor when signal is positioned at the edge of the block. The situation is much better when the signal is positioned in the middle. This is the typical artifact caused by smearing and leakage. We analyze again the difference between the spectra obtained by the two algorithms. The energy difference is 0.2 % for the single carrier wave at the center, and 21 % at the edge. Note that the later is actually the worst case, and it almost never occurs in real life.

In addition to simulation, some measurements are performed to compare the USRP-Iris based sensing solution to the Airmagnet spectrum analyzer. The test setup is shown in Fig 5. We connect the USRP front-end to a Rohde & Schwarz signal

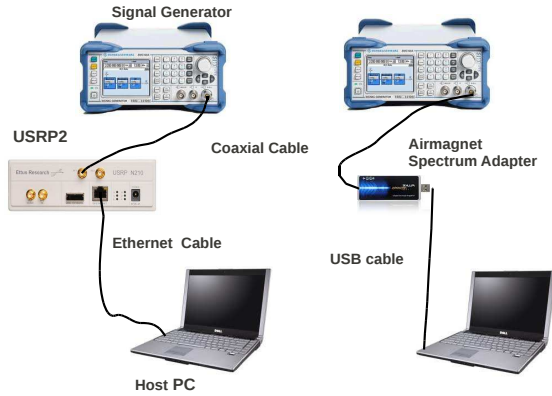


Fig. 5: Test Setup

generator with a coaxial cable, to minimize unpredictable influences of the environment. The same setup is used for Airmagnet measurement, as shown in the lower part of Fig 5. The only differences in hardware between the two setups are replacing the USRP with Airmagnet spectrum adapter and the Ethernet UTP cable with a USB cable. Because Airmagnet has a fixed frequency span of about 100 MHz and resolution bandwidth of 20 MHz [13], we configured our USRP-Iris based solution with similar parameters. Each measurement consists of 20 sweeps over the entire 2.4 GHz ISM band, while the PSD bins as well as the accurate time instance at the beginning of each sweep are recorded by both USRP and Airmagnet.

Our USRP2-Iris implementation has significantly better time resolution at an average of 6 sweeps per second, compared to Airmagnet's one sweep per second. In addition, our solution's resolution bandwidth is 98 kHz, better than 153 kHz from Airmagnet.

First, we measured the noise floor. When no signal is generated by the signal generator, Airmagnet's PSD recording is on average around -105 dBm while USRP is around -88 dBm. Further measurements show that when the transmit power drops below -80 dBm, the signal is buried in the noise for USRP, but Airmagnet can still distinguish the signal from the noise until the transmit power drops below -100 dBm. We conclude that the noise floor of our USRP-Iris solution is approximately -85 dBm, while that of Airmagnet is approximately -105 dBm.

For further measurements we selected the band around 2.485 GHz, which appears to be relatively quiet in our noise floor measurements. We set the signal generator to transmit a sine wave at 2.485 GHz with transmit power of -40 dBm, and then reduced it in steps of 10 dBm down to -100 dBm.

We observe that on the average both devices measure a power level which is a few dBm lower than what is indicated by the signal generator. This difference can be attributed to loss due to mismatch in front-end impedances that always exists. Nevertheless, changes of transmit power are correctly measured by both devices. We conclude that both solutions can accurately measure signal power.

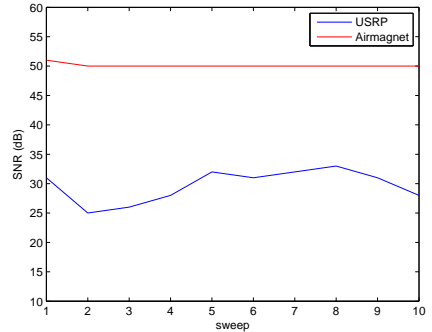


Fig. 6: SNR of -50dBm sine wave

Next we examined stability over time. Figure 6 shows 10 consecutive sweeps from both devices when a sine wave of -50 dBm is transmitted. It is evident that Airmagnet has relatively more stable measurements while the recordings of our USRP-Iris implementation have more fluctuations. One possible explanation is that our implementation is more sensitive to signal variation in the time domain. A more important reason is Airmagnet uses an interval of 30 ms for sample collection at each center frequency [17]. This is much longer than our solution, since we only collected 2048 samples with 25 MHz sample rate. This implies the actual sampling time at each center frequency is only 82 μ s. This long sampling time of Airmagnet gives more accurate measurement but also leads to longer sweep time. It is also noticeable that Airmagnet's SNR is on the average 20 dB higher than ours, which is consistent with the 20 dB difference in noise floor.

V. SENSITIVITY ANALYSIS

Triggered by the big difference in noise floor, we decide to analyze the sensitivity of the XCVR2450 daughter board. It is well known that the noise floor of a receiver in decibel can be calculated as follows [16]:

$$\text{noise floor} = P_{dBm} + NF$$

where P_{dBm} is the thermal noise at the input, NF is the noise figure of the system. P_{dBm} can be written as [12]

$$P_{dBm} = 10 * \log_{10}(k * T * B * 1000)$$

where k is Boltzmann constant, B is the bandwidth of the system in Hz, T is the absolute temperature. Substituting $T = 290K$ for room temperature we can write

$$\text{noise floor} = -174 + 10 * \log_{10}B + NF$$

The noise figure NF is defined as the difference in SNR between the input and output of the system. The general noise figure of a radio receiver can be calculated with Friis' Formula.

$$NF_{receiver} = NF_{LNA} + \frac{NF_{rest} - 1}{G_{LNA}}$$

The above formula shows that the first stage of amplification in the receiver chain denoted as NF_{LNA} , which is often called

the low noise amplifier (LNA), dominates the noise figure of the whole system, if its gain G_{LNA} is sufficiently high. In a typical RF front-end there are two stages of amplification, the amplification before mixer (RF gain) and amplification after mixer (IF gain). In case of the daughter board XCVR2450, both stages are contained in the analog chip MAX2829. Before reaching the MAX2829 all components are passive, which typically have very small contribution to the overall noise figure. The MAX2829 is the first active element in the receiver chain and also the last stop the analog signal passes before reaching the ADC. Consequently, the LNA is the amplification before the mixer. According to the data sheet of MAX2829 [11] the typical noise figure with medium LNA gain is 16 dB, with high LNA gain is 4 dB and with low LNA gain is 30 dB. In our experiment the medium LNA gain is used hence 16 dB is the value to use. As explained above, this noise figure of MAX2829 dominates the noise figure of the complete system. We substitute this number, as well as the bandwidth of the system, and calculate

$$\text{noise floor} = -174 + 10 * \log_{10}(20 * 10^6) + 16 = -85dB$$

which confirms our measurements.

The calculation reveals two dominant factors for receiver's sensitivity: bandwidth and gain. More bandwidth results in higher thermal noise from input. Higher gain setting reduces the system noise figure which eventually reduces the noise floor. The easiest solution to improve sensitivity on a receiver would simply be increasing its gain. This reasoning is confirmed by another group of measurements where we increased the gain of the USRP. The average SNR with -50 dBm transmit power is plotted in Fig 7 . When the gain is increased by 20 dB, the SNR of USRP is on average 47 dB, which makes the sensitivity comparable with Airmagnet. However, when gain is above 30 dB, some amplifiers reach saturation region and distortions are observed in the resulted spectrum. Hence we have a none linear shape at the end of the graph. As opposed to the gain factor, modifying the resolution bandwidth has more noticeable consequences. The direct result of reducing resolution bandwidth is the increase in sweep time. Moreover, more processing time is required also on the host PC due to the smaller step in the frequency domain. Hence there is always a trade-off between sensitivity and speed.

VI. CONCLUSION

We presented a highly flexible sensing solution based on USRP2 and Iris platform. We verified by measurements that our solution is more flexible than a common off-the-shelf solution — Airmagnet — and it is capable of achieving higher resolution in both time and frequency domains. On the other hand the hardware is relatively small and cheap compared to professional spectrum analyzers, yet powerful enough to achieve real-time wide-band sensing. Moreover, there is no “black box” in either the hardware or software implementation. The solution is transparent and can easily be ported to other similar platforms.

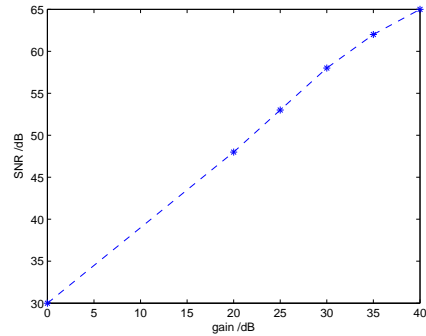


Fig. 7: SNR vs Gain

Currently multiple USRP's are deployed in the Wilab testbed in IBBT and connected to central database. This potentially forms a distributed sensing system. Channel quality assessment is also under development, which can serve as a foundation of cognitive MAC protocol.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union's Seventh Framework Programme FP7/2007-2013 under grant agreements n 257542 (CONSERN project) and n 258301 (CREW project).

We would also like to thank the Telecommunications Research Center of Trinity College, Dublin for their support with the Iris platform and their generosity in sharing their knowledge of USRP.

REFERENCES

- [1] S.Haykin, "Cognitive Radio: Brain-empowered Wireless Communications," *IEEE J. Sel. Areas Comm.*, vol. 23, no. 2, pp. 201-220, Feb.2005
- [2] D. Cabric, S. M. Mishra, and R. Brodersen, Implementation issues in spectrum sensing for cognitive radios, in *Proc. 38th Asilomar Conf. on Signals, Systems and Computers*, vol. 1, Pacific Grove, CA, Nov. 2004, pp. 772-776.
- [3] E.Buracchini, "The Software radio concept", in *IEEE comm. Mag.*, vol. 38, no.9, pp. 138-142, 2000
- [4] Ettus Research. <http://www.ettus.com/>.
- [5] <http://code.ettus.com/redmine/ettus/projects/uhd/wiki>
- [6] P. Sutton et al. "Iris: an architecture for cognitive radio networking testbeds," in *IEEE comm. Mag.*, vol. 48, no.9, pp. 114-122, 2010
- [7] Y. Tevfik, A. Huseyin, "A survey of spectrum sensing algorithms for cognitive radio applications", in *IEEE comm.Survey and Tutorial*, vol. 11, no. 1, pp. 116-130, 2009
- [8] P. Paysarvi Hoseini et al. "An Optimal Algorithm for Wideband Spectrum Sensing in Cognitive Radio Systems", in *Communications (ICC), 2010 IEEE International Conference*, pp. 1-6
- [9] Z. Quan, S. Cui, A. H. Sayed, and H. V. Poor, Optimal multiband joint detection for spectrum sensing in cognitive radio networks, *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 1128-1140, Mar. 2009.
- [10] Z. Tian and G. B. Giannakis, A wavelet approach to wideband spectrum sensing for cognitive radios, in *Proc. 1st Int. Conf. on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, Mykonos, Greece, Jun. 2006
- [11] MAX2829 datasheet <http://www.maxim-ic.com/datasheet/>
- [12] S. Haykin, *Communication Systems*, 4th ed. New York: Wiley, 2001, p. 61
- [13] Fluck Corporation *AnalyzerAir User Manual*, Rev.2, 2006
- [14] Simulink USRP support <http://www.mathworks.nl/discovery/sdr/usrp.html>
- [15] Labview USRP <http://zone.ni.com/devzone/cda/tut/p/id/12985>

- [16] Adrian W. Graham, Nicholas C. Kirkman, Peter M. Paul *Mobile radio network design in the VHF and UHF bands: a practical approach*, 2007
- [17] Airmagnet FAQ <http://airmagnet.flukenetworks.com/faq/>