ISSN:1983 7402 ITA, 24 a 27 de setembro de 2013

Electronic-Warfare Training Using Low-Cost Software-Defined Radio Platforms

Warren P. du Plessis
University of Pretoria, Pretoria, 0002, South Africa

***Abstract*—Skilled electronic warfare (EW) professionals are in greater demand today than ever before. However, the vast range of concepts and technologies relevant to EW makes effective training of large numbers of EW professionals extremely challenging. The use of open software-defined radio (SDR) platforms is proposed as an effective, low-cost means of overcoming this problem in the radio frequency (RF) aspects of EW. A number of examples using a low-cost receiver system are presented to demonstrate the value of this approach to EW training.**

***Keywords*—Education and training, electronic warfare (EW), software-defined radio (SDR).**

I. Introduction

The need for increased and improved training for electronic warfare (EW) professionals at all levels is becoming increasingly apparent in a number of different ways as illustrated by the following examples. Editorials in the Journal of Electronic Defense (JED) have recently argued that every soldier is to some degree involved with and affected by EW [1], that the requirement for EW professionals is increasing while number of EW experts is diminishing [2], and that EW is becoming more complex every year [3]. Of the nine strategic actions to address a number of shortfalls in the EW community presented in a 2010 Association of Old Crows (AOC) position paper titled "21st century electronic warfare" [4], two are "Develop a spectrum enterprise workforce" and "Improve personnel standards." In response to this need, one of the ten sessions at the upcoming AOC International Symposium and Convention is devoted to the topic of "Developing future electromagnetic spectrum operations (EMSO) warriors" [5].

The broad range of concepts and technologies underlying EW are often difficult for non-experts to grasp. Approaches to EW training often exacerbate this problem by being based on lectures on the theory of EW without allowing practical, hands-on experience to be gained. Where practical training is available, it is often based on systems whose expense limit the number of students which can be accommodated and whose complexity discourage experimentation.

The recent development of software-defined radio (SDR) systems targeted at researchers, and radio amateurs and other amateurs holds tremendous potential for training in the radio frequency (RF) aspects of EW. The low cost of many of these systems means that large numbers of these systems can be procured and provided to each student. The availability of free and open-source software (FOSS) and open hardware SDR platforms encourages students to experiment with new concepts and ideas allowing relevant experience to be gained.

A number of low-cost SDR hardware and software platforms are highlighted in Sections II and III. Section IV provides some examples of how the even simplest and cheapest SDR systems can be used to help students understand important EW concepts. Lastly, a brief conclusion is provided in Section V.

W. P. du Plessis, wduplessis@ieee.org, Tel: +27-12-420-2951, Fax: +27-12-362-5000.

This work is based on the research supported in part by the King Abdulaziz City for Science and Technology (KACST) and the National Research Foundation of South Africa (NRF) (Grant specific unique reference number (UID) 85845). The NRF Grantholder acknowledges that opinions, findings and conclusions or recommendations expressed in any publication generated by the NRF supported research are that of the author(s), and that the NRF accepts no liability whatsoever in this regard.

II. SDR Hardware Platforms

The combination of the rapidly-expanding global telecommunications market and development of low-cost RF microchips has led to the development of a number of powerful, yet low-cost, SDR hardware platforms. This section summarises some of these platforms with the focus on systems for which schematics, firmware and software are available.

Arguably the most established supplier in this market is Ettus Research, which offers a range of hardware platforms [6] under the name of Universal Software Radio Peripheral (USRP). The RF and digital portions of the SDR system are purchased separately and then integrated by the user. This approach allows a user to independently select the RF and digital portions of the system which best suit their needs. The schematics and source code of both the firmware and computer drivers are available for the full range of Ettus SDR systems, allowing extensive customisation of these systems. Ettus Research was recently purchased by National Instruments demonstrating the success of the USRP systems.

The bladeRF platform developed by Nuand LLC [7] was completely funded via community support of a Kickstarter campaign [8] demonstrating the tremendous interest in such systems. The bladeRF system is based on low-cost central processing unit (CPU), field-programmable gate array (FPGA) and RF chips leading to a low system cost. The schematics, firmware and software for this system are provided to allow user customisation of the system. Usefully, the system can either be powered via a USB connection or via a separate power supply to allow both tethered and standalone operation. The first bladeRF systems were only shipped in July 2013, so the long-term success of this project remains to be demonstrated. However, initial tests by the author and his students suggest that this system is extremely capable.

Another SDR platform which solicited community support via a Kickstarter campaign is the HackRF system [9]. The HackRF system is unique in that the full documentation for the project, including the printed circuit board (PCB) layout and bill of materials, are provided allowing users to manufacture the device themselves. Despite this option, the Kickstarter campaign raised over USD 600,000 from 1,991 backers despite the required funding being only USD 80,000 [10]. The HackRF platform also covers the widest frequency range of any of the systems considered here and can operate

from 30 MHz to 6 GHz. The main drawback of the HackRF platform compared to the above systems is that it cannot transmit and receive simultaneously (half-duplex operation only).

An interesting addition to this list is a range of digital television receivers which have been found to allow reception of raw samples via USB [11]. While these devices do not have transmitters, they are still extremely useful for electronic support (ES) training. The most remarkable attribute of these devices is their extremely low price which ranges from USD 10 to USD 40 depending on the RF chip used. Despite this low cost, the most capable of these devices operate from tens of megahertz to around 2 GHz and have a maximum sampling rate of 3.2 MS/s with 8-bit resolution for each of the I and Q channels. While practical considerations limit the maximum usable sampling rate to 2.4 MHz, this rate is still high enough to receive a number of different signals. The schematics and firmware for these systems are not available, but their extremely low cost and the existence of open-source drivers justify their inclusion here.

III. SDR Software Platforms

This section provides an overview of some of the software platforms available for SDR system development. FOSS SDR software platforms are emphasised, but a number of commercial products which can be used for this purpose also exist. All the hardware platforms listed in Section II have open-source drivers and integrate with the GNU Radio SDR system. The significance of these observations is outlined below.

The benefit of open-source drivers is that any of these devices can be accessed by custom software written by the user. This allows users to develop systems in mathematical packages like Octave [12], [13] or programming languages like C/C++ [14] and Python [15] – all of which are extremely powerful and are available at no cost. Users can thus work in a development environment they are comfortable with, while being able to exploit the full potential of their SDR hardware.

GNU Radio is an open-source SDR development environment which supports a large number of SDR hardware platforms including those highlighted in Section II [16] [16]. A large number of generic signal-processing blocks are included with GNU Radio and allow the rapid development of complex SDR systems. GNU Radio libraries are available for development using C/C++ and Python, but there is also the option of using a graphical system called the GNU Radio Companion. The GNU Radio Companion retains the power of GNU Radio, yet has a relatively simple interface, allowing students to focus on mastering the relevant material rather than focusing on mastering the SDR system.

IV. EW Training with Low-Cost SDR Platforms

This section provides some examples of how the digital television receivers mentioned in Section II can be used with the GNU Radio Companion considered in Section III to demonstrate important EW concepts.

The hardware platform only has a receiver, so only ES principles can be considered. This limitation is removed by the other hardware platforms considered in Section II.

The receiver was tuned to a centre frequency of 940 MHz because this is in the E-GSM 900 downlink frequency band, and two strong transmitters and a number of weaker transmitters were observed. This parameter selection thus allows the challenges associated with multiple simultaneous signals to be demonstrated.

A. Radar Warning Receiver

Many radar warning receiver (RWR) systems are based on a crystal detector which responds to all signals over a wide range of frequencies [17]. A GNU Radio Companion flowgraph which models a simple RWR is shown in Figure 1. The crystal detector has been modelled by a conversion from the complex representation of the signal to the magnitude of the signal. A frequency-domain output has also been included to provide additional information to facilitate understanding of the operation of the RWR model.

The resulting output when the flowgraph in Figure 1 is run is shown in Figure 2. The top plot shows the time-domain signal in green and the output of the threshold detector in blue. The bottom plot shows the frequency-domain signal in blue and the results of a peak-hold display in green. Note that the time- and frequency-domain plots are unfortunately not synchronised.

From the frequency plot in Figure 2, it is clear that signals from two strong transmitters and a number of weaker transmitters are observed. The time-domain plot in Figure 2 shows a number of detections demonstrating the effectiveness of even this simple RWR model. While two adjacent pulses are clearly observed on the right of the time-domain plot, the left of the plot is much more difficult to interpret because signals from more than one transmitter are received simultaneously. This system thus also demonstrates how a wide-open receiver like the one considered here has difficulty in resolving multiple simultaneous signals.

B. Channelised Receiver

A channelised receiver is similar to the crystal-detector based RWR considered above, but with the addition of filters which separate the received frequency band into a number of channels. A GNU Radio Companion model of a channelised receiver is shown in Figure 3, and the output when the flowgraph is run is shown in Figure 4.

Comparing the flowgraphs in Figures 1 and 3 clearly shows the additional complexity implied by a channelised receiver. The complexity is further emphasised by the fact that the RWR model covers the entire 2.4 MHz band, while the channelised-receiver model has five 200-kHz filters and thus only covers 1 MHz of the band.

The frequency-domain plot in Figure 4 is similar to that in Figure 2 because the same frequency band is considered. The time-domain plot clearly demonstrates how the channel filters are able to isolate the individual channels. Importantly, the weak transmission in channel 2 (green plot) starting slightly before 3.5 ms is observed even in the presence of a powerful simultaneous transmission in channel 4 (purple plot) – clearly demonstrating one of the main advantages of a channelised receiver. Comparing Figures 2 and 4 also suggests that the noise in each channel of the channelised receiver is decreased because weak transmissions in channels 1 (blue plot) and 2 (green plot) can clearly be seen. The improved sensitivity of a channelised receiver resulting from suppressing out-of-band

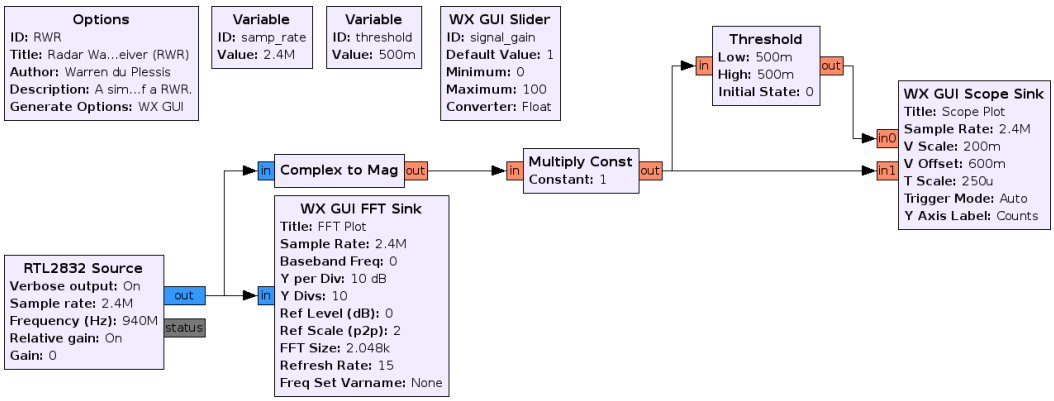

Figure 1. GNU Radio Companion flowgraph of a simple RWR.

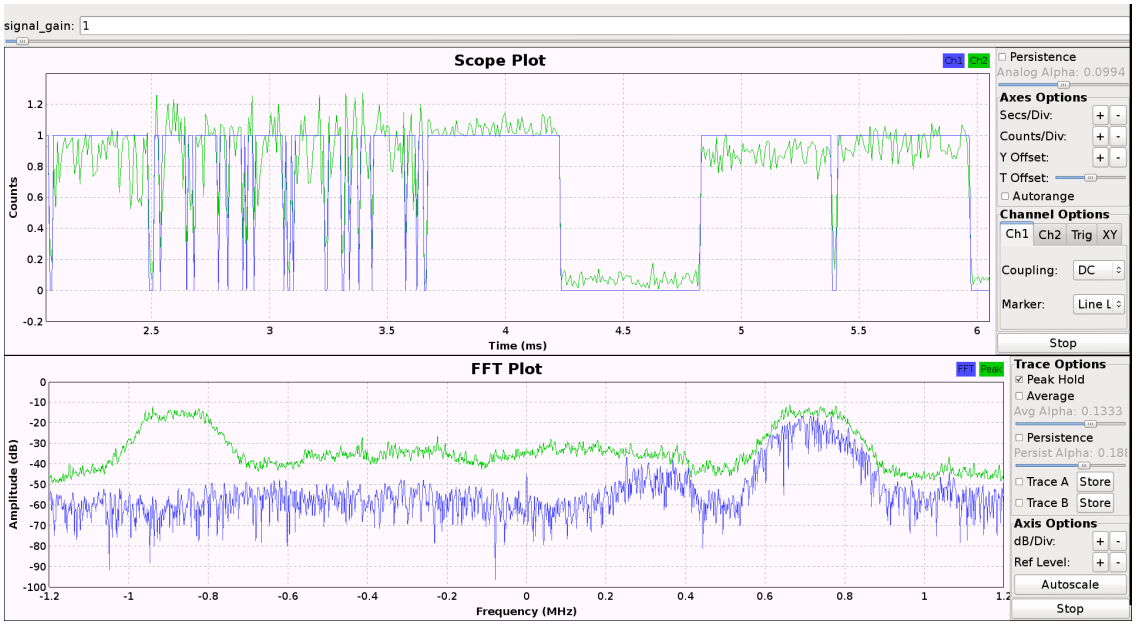

Figure 2. Output generated when the GNU Radio Companion flowgraph in Figure 1 is run.

noise through the use of filters is thus demonstrated without requiring complex explanations.

C. Energy Detection

Another approach to using the GNU Radio Companion to illustrate important concepts is demonstrated in the flowgraph in Figure 5 which gives the display in Figure 6 when run. This case considers a pure simulation (no hardware receiver or transmitter) allowing students to interactively experiment with theoretical concepts. The graphical nature of the GNU Radio Companion interface both allows rapid development of such simulations and allows students to experiment with how the system architecture influences the results.

Figures 5 and 6 consider energy detection by using a complex-to-magnitude-squared block to convert a signal to its energy content and a moving-average block to integrate that energy over time. Three sliders are available at the bottom of the output allowing the threshold, noise amplitude and number of samples integrated to be varied. The display reacts to changes as they are made allowing immediate feedback to be provided.

The top graphs in Figure 6 show the time-domain signal in green and the output of the threshold detector in blue, while the bottom plots show histograms of the output of the detector. The left plots are for noise alone allowing the probability of false alarm to be investigated, while the right plots show the signal with noise to allow the detection probability to be studied.

V. Conclusion

The effective training of large numbers of EW professionals represents a major challenge to the global EW community. Theoretical training without practical, hands-on experiments is only of limited value because students are unable to develop a feeling for the material presented. However, the cost of the necessary systems has historically been prohibitive and their complexity has limited the opportunities for experimentation.

The use of low-cost SDR open hardware platforms in conjunction with FOSS SDR software platforms represents a viable option to allow the training of large numbers of EW professionals in the RF aspects of EW. The cost of these systems means that each student can be provided with their own system, and the open nature and versatility of these systems allows customisation to the requirements of the training and encourages experimentation by students.

A number of examples were presented showing how the use of open SDR hardware and software platforms allows important EW concepts to be demonstrated. The fact that

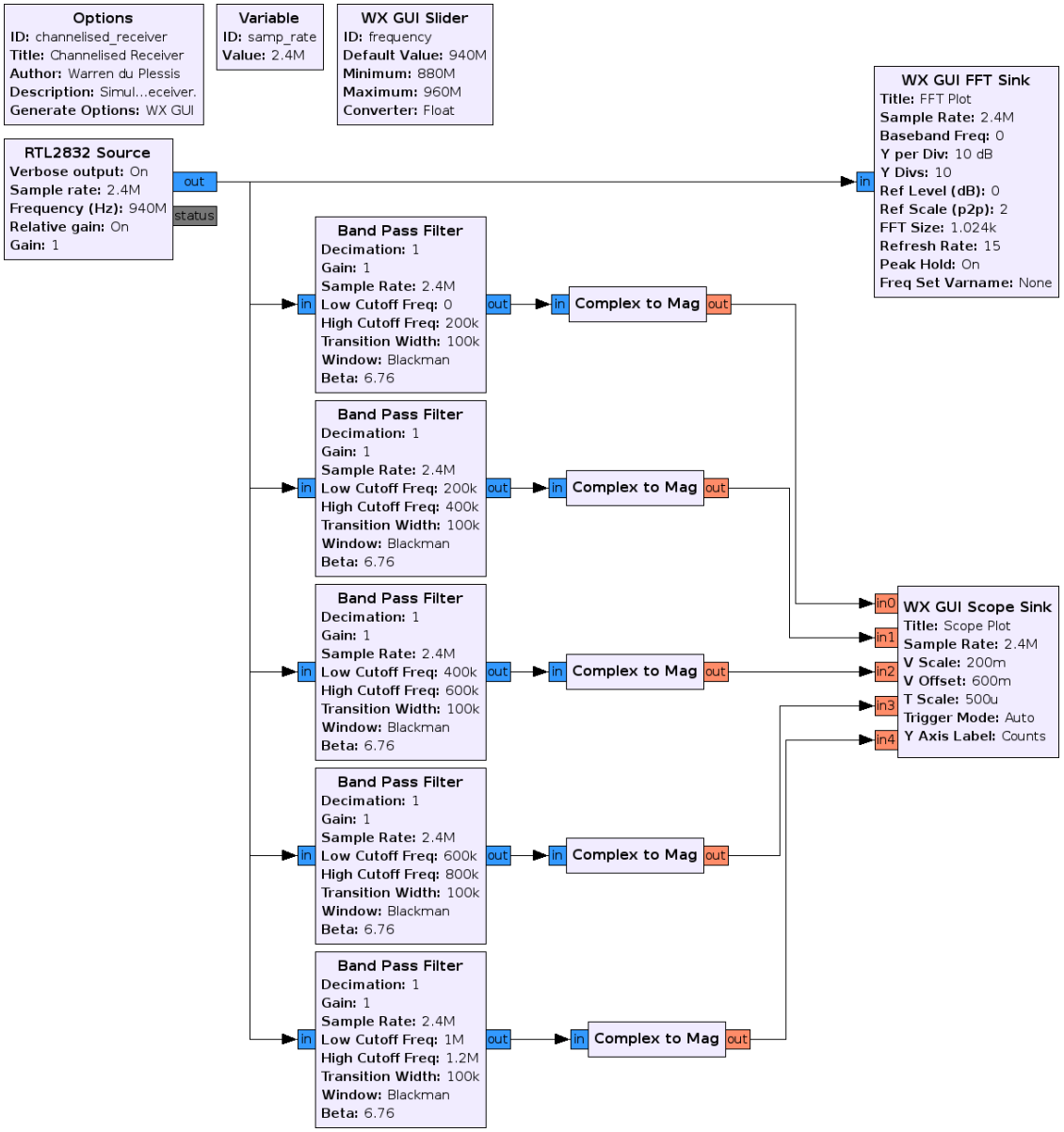

Figure 3. GNU Radio Companion flowgraph of a channelised receiver.

Figure 4. Output generated when the GNU Radio Companion flowgraph in Figure 3 is run.

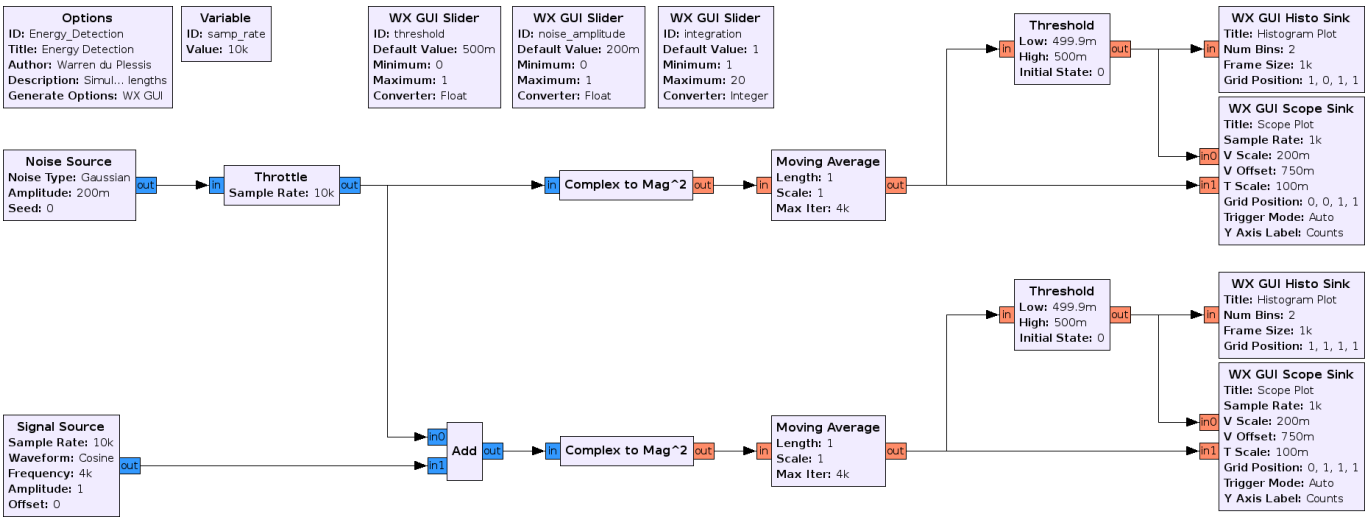

Figure 5. GNU Radio Companion flowgraph to demonstrate energy detection.

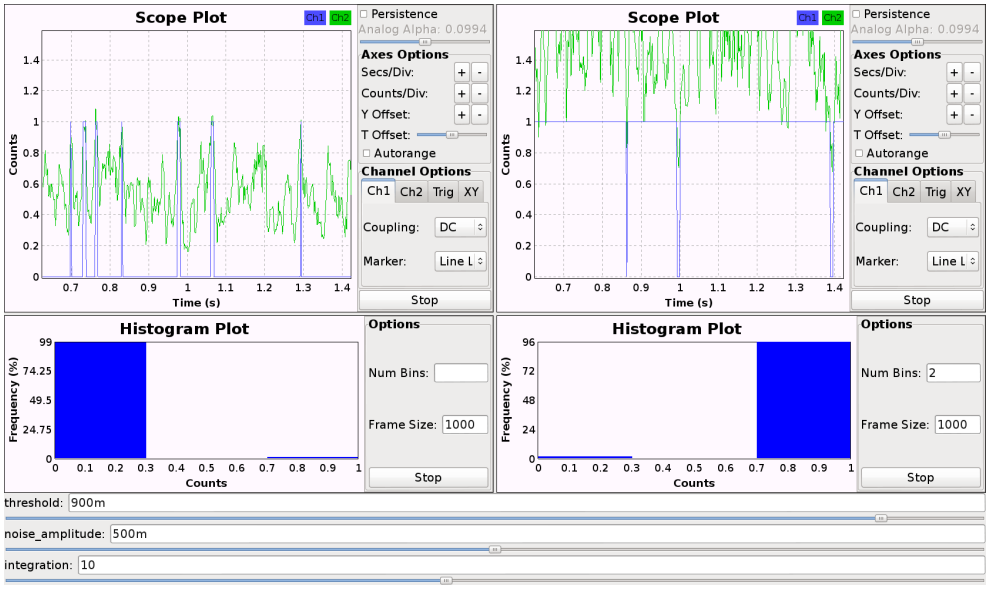

Figure 6. Output generated when the GNU Radio Companion flowgraph in Figure 5 is run.

the software is based on a system block diagram facilitates students' understanding of the relevant concepts and allows comparisons between different system architectures to be directly compared. Furthermore, the simplicity and wide range of available development platforms encourages experimentation and allows customisation.

ACKNOWLEDGMENT

The author would like to thank Albert Lysko, Marcel Gouws and Jan Gutter for making him aware of the USRP and digital television platforms considered in Section II.

REFERENCES

[1] J. Knowles, "Achieving victory," *Journal of Electronic Defense (JED)*, vol. 34, no. 11, p. 6, November 2011.
[2] J. Knowles, "Smart investment," *Journal of Electronic Defense (JED)*, vol. 35, no. 2, p. 6, February 2012.
[3] J. Knowles, "Mind the gap," *Journal of Electronic Defense (JED)*, vol. 35, no. 12, p. 6, December 2012.
[4] R. J. Elder, Jr, *21st century electronic warfare*, Association of Old Crows (AOC), 2010.
[5] (2013, July) Symposium agenda | 50th annual convention | AOC. [Online]. Available: http://www.crows.org/conventions/symposium-agenda-2013.html
[6] (2013, July) Ettus Research. [Online]. Available: http://www.ettus.com
[7] (2013, July) Nuand | bladeRF. [Online]. Available: http://www.nuand.com
[8] (2013, July) bladeRF – USB 3.0 software defined radio by Nuand – Kickstarter. [Online]. Available: http://www.kickstarter.com/projects/1085541682/bladerf-usb-30-software-defined-radio
[9] (2013, September) Great Scott Gadgets HackRF. [Online]. Available: http://greatscottgadgets.com/hackrf/
[10] (2013, September) HackRF, an open source SDR platform by Michael Ossmann –Kickstarter. [Online]. Available: http://www.kickstarter.com/projects/mossmann/hackrf-an-open-source-sdr-platform
[11] (2013, July) rtl-sdr – OsmoSDR. [Online]. Available: http://sdr.osmocom.org/trac/wiki/rtl-sdr
[12] (2013, August) GNU Octave. [Online]. Available: http://www.gnu.org/software/octave/
[13] (2013, August) Octave-Forge. [Online]. Available: http://octave.sourceforge.net/
[14] (2013, August) GCC, the GNU Compiler Collection. [Online]. Available: http://gcc.gnu.org/
[15] (2013, August) Python programming language – official website. [Online]. Available: http://www.python.org/
[16] (2013, July) GNU Radio – WikiStart – gnuradio.org. [Online]. Available: http://gnuradio.org/redmine/projects/gnuradio/wiki
[17] F. Neri, *Introduction to Electronic Defense Systems*, 2nd ed. SciTech Publishing, 2006.