A SDR TESTBED ARCHITECTURE FOR ACM, MIMO AND DSA IN MILITARY APPLICATIONS

Patrik Eliardsson (Swedish Defence Research Agency (FOI) Department of Information Systems, Linköping, Sweden, patrik.eliardsson@foi.se); Ulrika Uppman (Swedish Defence Research Agency (FOI) Department of Information Systems, Linköping, Sweden, ulrika.uppman@foi.se

ABSTRACT

A need for increased link capacity in the upcoming military communication systems can be foreseen where new services such as streaming video will increase the throughput requirements.

This paper describes a testbed architecture based on the GNU Radio Software Defined Radio (SDR) platform together with the Universal Software Radio Peripheral version 2 (USRP2). The architecture is constructed for implementation of techniques like Multiple Input Multiple Output (MIMO), Adaptive Coding and Modulation (ACM) and Dynamic Spectrum Access (DSA) that will increase the link capacity in military applications. The testbed is intended to verify simulated results with realistic channel conditions and environments, and to demonstrate the benefits of the mentioned technologies.

1. INTRODUCTION

In the civilian market the next generation of mobile communication begins to take form with the longterm evolution (LTE) standard. The LTE standard improves the link capacity with use of various technologies such as Multiple Input Multiple Output (MIMO), Adaptive Coding and Modulation (ACM) and spectrum flexibility [1], [2]. The same techniques used in the LTE standard are also interesting for military applications, but the frequency ranges, the operational use and the requirements differ. Furthermore, in some military scenarios there may be no infrastructure to rely on and in addition the communication systems can also be intentionally interfered.

When introducing new military communication services such as blue force tracking, video streaming etc the requirements on the communication links become more demanding. The links have to be more robust against interference and the throughput have to increase without loss in the quality of service.

The operative military communication system is often designed with a fixed modulation and coding scheme to be robust in all types of scenarios. The drawback with a fixed scheme is that the link capacity is limited in favour of making the link robust against channel variations over the time. Instead of using a fixed scheme, ACM can be used to optimize the coding and modulation scheme over the time, where the channel conditions varies. With this technique the link can have maximum throughput and still be robust [3].

Another technique that can increase the throughput and/or the robustness of a link is MIMO. The MIMO technique uses more than one antenna on the receiver and the transmitter to achieve multiple parallel channels [4].

Until now the military have had rights to use a lot of frequencies but the trend now is that the amount of frequencies are decreasing and besides that frequencies are a coveted resource. Frequency utilization measurements on the other hand show that the utilizations varies a lot over the time and that there is an opportunity to utilize the spectrum even more [5]. By finding the time slots where the frequencies are unused, a secondary user could utilize these time slots until the primary user resumes. This concept is called Dynamic Spectrum Access (DSA) and could be useful for military radio communication systems as they often operate in new, unknown environments where the frequency spectrum allocation is unknown and where the radio system may be exposed to intentional interference in electronic warfare attacks. If the system is jammed, this technique would leave the interfered frequency and initialize communication on another frequency.

All together, ACM, MIMO and DSA are promising techniques to be used in the future military communication systems. However, these techniques have to be tested in realistic scenarios and the implementation complexity needs to be studied further.

There are other projects where testbeds are proposed. However, the architecture of these testbeds are often dependent on the current application or the complete testbed tends to be too large to move out of the lab. For example, in [6] a testbed for spectrum sensing and DSA is presented using the Universal Software Radio Peripheral (USRP) hardware. The testbed dynamically allocate different sub-carriers in an OFDM waveform based on the output of an energy detector. In [7] a testbed for MIMO were built, based on field programmable gate arrays (FPGAs). The testbed operates at the industrial, scientific, and medical (ISM) band at 902 - 928 MHz, which are much higher frequencies than we are interested in for military purpose and the testbed is also very specific for MIMO.

In this paper we propose a general architecture for a testbed of high capacity techniques like ACM, MIMO and DSA in military applications. We also suggest a suitable software defined radio (SDR) platform for the testbed to be implemented on. The architecture is designed to be flexible and to utilize the selected SDR-platform and extend it to fulfill the requirements for the different techniques. Furthermore, we have identified and managed crucial design issues for the different high capacity techniques in our architecture.

2. PLATFORM

Although SDR is software-based, RF front-ends are necessary and play a crucial role in all applications. There are several commercially available RF front-ends to choose from, most of them assosiated with different software radio packages. A comparison of the most common SDR platforms is done in [8].

There are a few different available SDR platforms to implement the testbed on. For this work we have chosen GNU Radio together with the Universal Software Radio Peripheral version 2 (USRP2). The reasons behind this decisions are its available signal processing blocks, that the USRP2 is favourably priced, its support for frequencies around 300 MHz, bandwidths around 20 MHz and that this platform has gained a wide-spread use by several researchers [9], [6], [10].

GNU Radio is a free software development toolkit for SDR, with support for use with the USRP2 [11]. There are over 100 signal processing blocks in the library, ready to use. Signal processing blocks are implemented in C++ and later linked together in Python. Instead of connecting the blocks with Python code, there is also a graphical interface where blocks are linked together by clicking on the block and connecting them with wires. All source code is open and it is possible to write your own signal processing blocks.

The USRP2 follows a basic concept with the RF front-end located at plug-in-able daughterboards, ADC/DAC and Digital Down Conversion (DDC) located at the motherboard and a wired Gbit-Ethernet connection to a personal computer (PC). The USRP2 sample clock rate is 100 MHz and it is possible to synchronize the USRP2 with an external reference clock and 1 Pulse Per Second (1PPS) signal [12]. The cost of a motherboard plus a daughterboard is about $1700.

3. ARCHITECTURE

The proposed architecture was generalized in order to be able to use the same foundation for various purposes simply by adding specific functions or blocks. Thus, to introduce modularity and extendability, all signal processing blocks (SPB) were separated from the control unit and the application specific blocks (ASB). The system architecture overview is seen in Fig. 1.

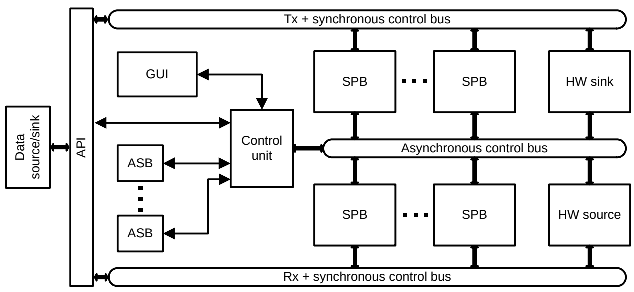


Figure 1: An overview of the proposed GNU Radio testbded architecture.

3.1. SPB

In the suggested architecture the SPBs are essentially complete signal processing blocks in GNU Radio but custom made GNU Radio blocks are also possible to use. These blocks are connected to either the receiver side bus or the transmitter side bus where they are provided with sample data. Coding, orthogonal frequency division multiplexing (OFDM), performance estimation, spectrum sensing etc. are typical SPBs in the architecture.

3.2. ASB

The ASBs are blocks that include functionality that do not perform signal processing based on sample data as the SPBs do. The ASBs make their decisions based on information from various SPBs, passed to the block through the control unit. The decision is then fed back through the control unit to the appropriate SPB.

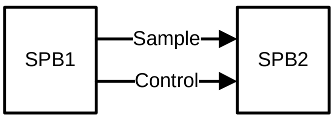
An ACM or a DSA functionality could be an example of an ASB.

3.3. Control unit

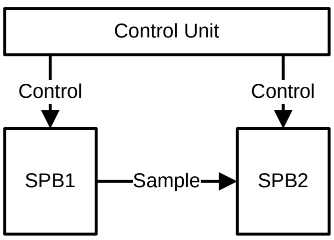
The control unit has control over the SPBs and can change settings based on decisions made by the GUI or by an ASB. Different performance measures and other calculated results from the SPBs are also forwarded to the appropriate blocks by the control unit.

3.4. Control signaling

To achieve a self controlling radio, various parameters of the signal processing blocks (SPB) have to be adjusted by the radio itself. There are different ways the parameters can change; either synchronously with the samples or immediately (asynchronous).



(a) Synchronous



(b) Asynchronous

Figure 2: Control signaling.

In the GNU Radio framework there is not yet any functionality for synchronous control signaling. There was a functionality called m-blocks for this in previous versions but it is now mitigated [13]. Instead of porting the old m-block to the current GNU Radio release 3.3 we plan to use the samples passing functionalities for synchronous control signaling.

When the control signal is passed synchronous with the samples through the blocks, see Fig. 2a, the blocks contain a table of all parameters that can be changed during runtime. The control signal is limited to one byte to minimize the memory usage and will select one of the parameters from the table. Before the block starts to consume samples the control signal will reconfigure the block. The parameter table will however limit the blocks to the parameters in the table.

For the blocks that can be controlled asynchronously, see Fig. 2b, the parameters can be changed in two ways; either they use a table of permitted parameters or the parameters can be passed directly to the block. The parameters can be passed directly to the blocks because they do not need to be passed with every sample and therefore this will not increase the memory usage significantly. This will not limit the parameters to a certain predefined set.

4. VERIFICATION

The verification process of the SPB is of great importance, and for each of the SPBs a unit test will be created. When all the SPBs pass the tests, the system will be verified against a matlab reference model of the complete architecture.

When moving the design to the complete software and hardware based system, there will most likely be some implementation loss compered to the reference model. Possible implementation loss is one of the results that are interesting to investigate further to be able to see the limitations of the testbed and the aspects of a real (non simulated) testbed or system.

5. APPLICATIONS

The proposed testbed architecture is flexible and can be used for several applications. Applications may include their functionality in different ABSs and their signal processing in different SPBs. Our interests are in high capacity techniques like ACM, DSA and MIMO and the next step is to implement these applications in the testbed. Each of the these applications have specific implementation challanges and aspects that have been taken into consideration in the architecture.

5.1. ACM

The simplest form of the ACM block may use a lookup table to choose modulation size and coding rate based on one or several channel parameters. However, it is also possible to extend the ACM block with a cognition node so that the decision of modulation size and coding rate are made using a self-learning algorithm.

In our architecture, the ACM implementation is an ASB. The ACM block is responsible for selecting the optimal settings at the moment so that the link capacity is maximized.

The ACM block will communicate with the control unit which in turn will change settings based on the decisions made by the ACM block. The ACM block will receive feedback and performance measures from the control unit gathered from SPBs.

With the ACM functionality it becomes important that the information about current settings are distributed in a structured manner. When the adaptation is performed on a packet basis, the transmitter can perform asynchronous signaling of settings (Fig. 2b). However, the receiver may need synchronous control signals between blocks (Fig. 2a).

5.2. DSA

Based on the output from a SPB that performs spectrum sensing the DSA ASB finds the optimum frequency allocation for the moment. In the decision phase this ASB considers the current frequency utilization and possibly also the feedback of previous made decisions. The decision could then be stored to later be utilized in future decisions. From the decision of the ASB the control unit will select which sub-carriers to be used in the OFDM-waveform during the next packet and the other sub-carriers will be left unused.

The spectrum sensing is very crucial for the DSA algorithm, therefore several different spectrum sensing methods can be implemented like; energy detection, impulsiveness ratio (IR) [14] and amplitude probability distribution function (APD) [15] as they perform different. The energy detection method is a suitable detector for continuous interference that can be approximated as additive white Gaussian noise (AWGN), for example a TV-transmitter, while this kind of detector is less suitable when the interference has an impulsive behavior. The IR method and the APD method on the other hand have better performance for impulsiveness interference.

5.3. MIMO

It is important to take the amount of data samples into concideration when working with MIMO since multiple antennas implies multiple sample streams. It is important to locate the system bottleneck and minimize the consequences. This has been discussed in [16].

The MIMO setup in Fig. 3 was selected where multiple USRP2 units are connected to multiple Ethernet cards on the host and synchronized with a reference signal from a GPS. Unlike the configuration with a MIMO-cable [12] where the USRP2-units are ordered in a master and slave configuration and only the master is connected to the Ethernet interface to the host, this setup will not introduce a bottleneck in the Ethernet interface when using several USRP2 units.

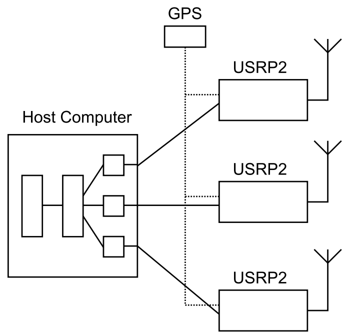


Figure 3: An example on MIMO configuration with multiple USRP2 units.

In a MIMO system implementation it is important to make sure that all antennas in each node are synchronous. The receiver needs to sample the data from the antennas and keep the data synchronous through the digital signal processing. Since the setup with several independent USRP2 units is used, this becomes more crucial. This area has been examined and concluded in [17]. It is also important to synchronize the transmitting antennas, this is supported in the USRP2 and GNU Radio libraries but the accuracy needs to be further investigated.

6. CONCLUSION AND FURTHER WORK

In this work a SDR platform have been suggested and an architecture have been proposed to fit the applications and techniques of interest. The architecture is to be seen as a framework to meet the requirements of the applications to be implemented. Furthermore, critical design issues for ACM, MIMO and DSA have been discussed.

The implementation of the architecture is at the moment in working progress and the first applications to be implemented are DSA and MIMO. Our further work will also include tests of the mentioned high capacity techniques using the testbed.

7. REFERENCES

[1] S. Parkvall and D. Astely, "The Evolution of LTE towards IMT-Advanced," *Journal of Communications*, vol. 4, no. 3, 2009.

[2] S. Sesia, I. Tonfik, and M. Bator, John Wiley & Sons Ltd, 2009*LTE The UMTS Long Term Evolution - From theory to practice*.

[3] A. J. Goldsmith and S.-G. Chua, "Adaptive Coded Modulation for Fading Channels," *IEEE TRANSACTIONS ON COMMUNICATIONS*, vol. 46, no. 5, pp. 595–602, May 1998.

[4] D. Gesbert, M. Shafi, D. shan Shiu, P. J. Smith, and A. Naguib, "From theory to practice: an overview of MIMO space-time coded wireless systems," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 3, pp. 281–302, April 2003.

[5] K. P. Tugba Erpek, Mark Lofquist, "Spectrum occupancy measurements: Loring commerce centre, limestone, maine, september 18-20, 2007," *Shared Spectrum Company Report*, 2007.

[6] Z. Yan, Z. Ma, H. Cao, G. Li, and W. Wang, "Spectrum sensing, access and coexistence testbed for cognitive radio using USRP," in *Circuits and Systems for Communications, 2008. ICCSC 2008. 4th IEEE International Conference on*, May 2008, pp. 270–274.

[7] P. G. Jr, R. Hang, D. Truhachev, and C. Schlegel, "A Portable MIMO Testbed and Selected Channel Measurements," *EURASIP Journal on Applied Signal Processing*, vol. 2006, 2006.

[8] O. Gustafsson, K. Amirix, D. Andersson, A. Blad, C. Bonnetk, J. R. Cavallarox, J. Declerckz, A. Dejonghez, P. Eliardsson, M. Glassez, A. Hayark, L. Hollevoetz, C. Hunterx, M. Joshiy, F. Kaltenbergerk, R. Knoppk, K. Ley, Z. Miljanicy, P. Murphyx, F. Naessensz, N. Nikaeink, D. Nussbaumk, R. Pacaletk, P. Raghavanz, A. Sabharwalx, O. Sarodey, P. Spasojevicy, Y. Sunx, H. M. Tullberg, T. V. Aaz, L. V. der Perrez, M. Wetterwaldk, and M. Wu, "Architectures for cognitive radio testbeds and demonstrators - an overview," in *CrownCom Conference*, 2010.

[9] T. R. Newman and T. Bose, "A cognitive radio network testbed for wireless communication and signal processing education," in *NIEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop, 2009. (DSP/SPE 2009)*, Marco Island, FL, USA, Jan. 2009, pp. 757–761.

[10] R. Dhar, G. George, A. Malani, and P. Steenkiste, "Supporting integrated MAC and PHY software development for the USRP SDR," in *Networking Technologies for Software Defined Radio Networks, 2006. SDR '06.1st IEEE Workshop on*, 2006, pp. 68–77.

[11] "GNU Radio - the gnu software radio." [Online]. Available: http://gnuradio.org

[12] M. Ettus, "Universal software radio peripheral." [Online]. Available: http://www.ettus.com/

[13] "GNU Radio architectural changes." [Online]. Available: http://acert.ir.bbn.com/downloads/adroit/gnuradio-architectural-enhancements-3.pdf

[14] P. Stenumgaard, "A simple impulsiveness correction factor for control of electromagnetic interference in dynamic wireless applications," *IEEE Communications Letters*, vol. 10, no. 3, pp. 147–149, March 2006.

[15] K. Wiklundh, "The relation between the amplitude probability distribution of an interfering signal and its impact on digital radio receivers," *IEEE Trans. on Electromagnetic Compatibility*, vol. 48, no. 3, pp. 5337–544, August 2006.

[16] C. Thein, H. Cao, and A. Wilzeck, "Setup and Characterization of a Flexible Low-Cost 2x2 MIMO Testbed Based on USRP2," in *WSR10 - 6th Karlsruhe Workshop on Software Radios*, 2010.

[17] H. Tullberg and P. Eliardsson, "Properties of USRP2 in communication applications," in *RFMTC09 - 2nd conference on RF Measurement Technology for State of the Art Production and Design*, 2009.