

SDR Based Prototype for Filter Bank based Multi-Carrier Transmission

Ali Dziri, Christophe Alexandre, Rostom Zakaria, Didier Le Ruyet
LAETITIA/CEDRIC Lab, CNAM, 292 Rue Saint Martin 75141 Paris, France.
Email: ali.dziri@cnam.fr

Abstract—This paper investigates the design of a potential Software Defined Radio (SDR) based prototype for Filter Bank based Multi-Carrier Transmission. It allows for implementing and testing transmitter and receiver in real-time on Universal Software Radio Peripheral (USRP) by using two USRP boards. Additionally, an implementation of some classic channel models in a separate USRP board has been used for emulating a propagation channel. The presented prototyping system supports FBMC waveform but also Orthogonal Frequency Division Multi-carriers (OFDM) one using MATLAB routines. With this motivation, we propose a study on real-time decoding of Offset Quadrature Amplitude Modulation FBMC (OQAM-FBMC) radio frames. In this system, data recovery at the receiver is very sensitive to time and frequency synchronization of the frames. This problem was addressed in many theoretical research works. However, considering its practical aspects in the context of a real-time radio transmission, a signal processing block is required to satisfy a trade-off between the accuracy and the latency. We especially recognize different limits, imposed by both hardware and software entities, to the real-time system.

Keywords: SDR, USRP, prototype, FBMC.

I. INTRODUCTION

Until the year 80s, radio equipment technology was generally based on combination of analog components which are inflexible and dedicated to a specific waveform in a specified frequency band. Thus, it was impossible to change the behavior of an electronic system unless we change or replace some electronic components. This paradigm was no longer viable with the explosion of various radio standards. Progress in computer science and digital electronics have led to replace some analog parts of radio systems with a digital and software components, thus allowing a good capacity of configuration and synthesis of different waveforms. In this context, the concept of Software Defined Radio (SDR) has been emerged. It was introduced by Joseph Milota in 1991 [1], is a system wherein all the signal processing works in radio communication are done in software instead of hardware. In other words, in SDR, signal will be processed in digital domain instead of analog domain as in the conventional radio. An SDR consists of two distinct parts:

- Hardware entity which is an RF Front-End (FE). It performs the up/down-conversion of the analog signal as well as a Digital-to-Analog (DAC)/Analog-to-Digital (ADC) conversion at the transmitter/receiver.
- A baseband unity, a software running on a host computer, which performs appropriate signal processing functions

on the digitized signal.

The SDR is defined as a software based communication platform which characteristics can be reconfigured and modified to perform different functions at different times. The RF unity is reconfigurable through a control bus/Ethernet interface. In other words, antenna gain, sampling rate, and central frequency can be adjusted at the transmitter/receiver. Universal Software Radio Peripheral (USRP) is a flexible low-cost platform for SDRs developed by Matt Ettus [2]. USRP consists of two main boards: daughter board and mother board. A mother board is composed mainly by the following subsystems : FPGA, ADC, DAC, host processor interface, and power regulation. These are the basic components that are required for baseband processing of signals. A modular Front End, called a daughter board, is used for analog tasks such as up/down conversion, filtering, and other signal conditioning. This modularity allows the USRP to serve applications that operate between DC and 6 GHz. All products in Ettus Research Bus Series use a USB 2.0 interface to exchange samples with the host computer. These are recommended for applications that do not require high bandwidth and dynamic range provided by the Network Series (USRP N200 and USRP N210). Using a Gigabit Ethernet interface, the devices in the Network Series can transfer up to 50 MS/s of complex baseband samples to/from the host.

The USRP Hardware Driver (UHD) is the device driver provided by Ettus product family. It supports Linux, Mac OS, and Windows platforms. Several frameworks including GNU Radio, LabVIEW, Matlab and Simulink use UHD.

A. related work

Contributions about SDR implementation from the architectural point of view are very limited, see [3] and references therein. In [4], authors have presented a rapid prototyping system for Real-Time MIMO OFDM Transmission. This prototype is based on DSP and FPGA boards as hardware-specific accelerator. In [5], authors have presented methodology and practices of developing software receivers for Global Navigation Satellite System. Their implementation was based on GNU Radio framework. In [3], authors have developed a prototype of FBMC based SDR receiver. They distinguish SISO and MIMO configurations. Their implementation was based on the USRP2 board and the GNU Radio software. In [6], authors have presented a full duplex prototype of OFDM

Physical layer receiver based on NI USRP 2922 board and GNU Radio software. To our knowledge, no demonstrator based on USRP N210 board and MATLAB software has been presented for FBMC waveform.

B. Main paper contributions

In this paper, we present a potential SDR based prototype for FBMC end-to-end Transmission. We have implemented and tested real-time transmitter and receiver based on USRP N210 board and MATLAB routines. A channel emulator has been implemented by using the same hardware and software platform. The presented prototyping system provides various physical layer waveforms such as OFDM, FBMC. Our demonstration in this paper is limited to the latter one. To achieve this goal, we have investigated many practical issues mainly the frequency synchronization and channel estimation and equalization which become more challenging in the context of real-time decoding.

The paper is organized as follows. In section II, we represent briefly the architecture of the proposed prototype. Section III deals with the details about the structure of the frames at the transmitter side. Section IV discusses the baseband processes which are implemented for the recovery phase. Finally we present the experimental results in section V, and we conclude our study in section VI.

II. USRP BASED PROTOTYPE ARCHITECTURE

In the conception step of the presented prototype, we have to answer two fundamental questions : which are software and hardware platforms we will consider? For the first question, the multithread property of GNU Radio made it the first choice for research studies which evaluate the real-time performance of the algorithms. Consequently, most relevant research works are implemented in GNU Radio. However, recently, MathWorks offers support for the USRP N210 and USRP2, using the Communications System Toolbox 5.0, which supports the UHD to provide a real-time connection to the USRP family of radios directly from MATLAB 2011b. RF signals can be received from a USRP radio, and the data processed in real-time using MATLAB functions. Similarly RF signals can be transmitted from MATLAB by streaming data to USRP radios. Parameters such as center frequency, gain and interpolation or decimation rates can be configured directly from MATLAB.

Moreover, the possibility to implement MATLAB blocks which are executable in real-time over transmitters/receivers of type USRP provides a significant flexibility for researchers and scientists to evaluate quickly their proposed algorithms in a real environment. With these motivations, we have chosen MATLAB as a software platform, and jointly the USRP N210 board as a hardware platform. In a final step, we have chosen the WBX (Wide Band Transceiver) as a Frontal End RF module. It provides up to 100 mW of output power and a noise figure of 5 dB. The LO's for the receive and transmit chains operate independently, but can be synchronized for MIMO operation. The WBX provides 40 MHz of bandwidth capability and is ideal for applications requiring access to a

number of different bands within its range between 50 MHz to 2.2 GHz.

Fig. 1 demonstrates the block diagram of the proposed prototype architecture. It is composed of mainly three USRP board based blocks : the USRP based Transmitter, the USRP based channel emulator and USRP based receiver. Each block is composed of one PC host, a USRP N210 board with its WBX board as a radio frontal End. Each USRP board is identified by its IP address.

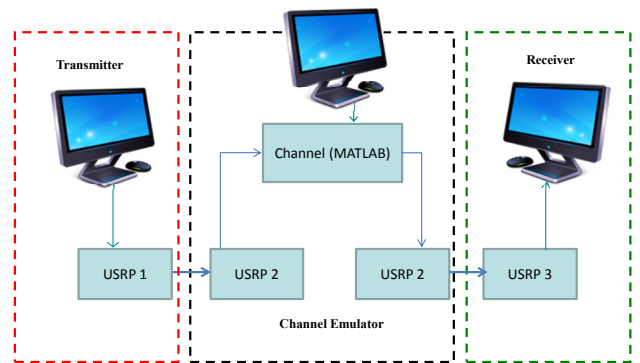


Fig. 1. FBMC based Prototype architecture

At the transmitter, the modulator generates OQAM-FBMC frames in digital complex baseband representation according a specified structure as explained below, and uses the USRP as the digital-to-analog front-end for up-conversion and final transmission. On the other end, at the receiver, the down converted signal is leaded towards the digital signal processing blocks. We address several issues in design of OQAM-FBMC receivers. One issue is the unknown symbol arrival time. A symbol timing error will have little effect as long as all the samples taken are within the length of the guard interval (GI) if we consider an OFDM modulation. However, GI is not considered in an FBMC frame. Once the signal timing is known, an additional issue is that the mismatch of the oscillators in the transmitter and the receiver imposes a carrier frequency offset (CFO). If uncorrected this causes both: (i) a rotation of the constellation and (ii) the spread of the constellation points similar to additive white Gaussian noise (AWGN). Finally, the channel imperfections need to be compensated. Thus, the detection and demodulation unit performs the following processes: time synchronization, estimation/correction of the absolute phase shift, estimation/correction of the carrier frequency offset, and the channel estimation. The channel emulator Unit is dedicated to implement some statistical channel models on USRP board such as vehicular A, vehicular B, ... It is worth noticing that MATLAB code not only executes the baseband signal processing functions, but also provides the proper configurations of the RF parameters through an Ethernet interface.

III. BASEBAND SIGNAL PROCESSING AT THE TRANSMITTER

The OQAM-FBMC can be considered as an evolved OFDM technique; it addresses the main drawbacks of the latter one. It removes the cyclic prefix leading to a higher useful bit rate. Moreover, compared to QAM-OFDM that transmits complex-valued symbols at a given symbol rate, OQAM-based FBMC transmits real valued symbols at twice this symbol rate. Consequently, a similar spectral efficiency is achieved by both systems [7]. Finally, by a judicious choice of the prototype filter, such as the PHYDYAS one, performance can be improved [8]. To implement this OQAM-FBMC scheme, we rely on the polyphase representation which provides a great simplification of theoretical analysis and computationally efficient implementations for several filtering applications [9], [10]. To give an experimental idea about the comparison between OFDM waveform and FBMC one, we have shown in Fig. 2, the full experimental spectrum over 128 sub-carriers of both waveforms measured by the spectrum analyzer. For

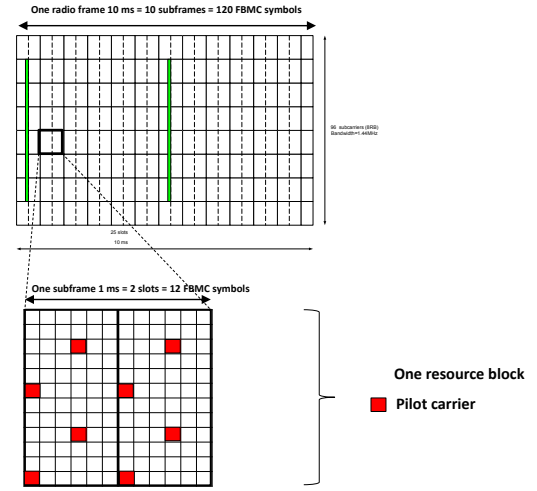


Fig. 3. FBMC LTE-like Frame structure

frequency offset. It is worth noticing that in the proposed prototype, we have considered only the equivalent of two OFDM symbols. Thus, it is more relevant than the demonstrator presented in [13], in which we use 4 OFDM symbols as a preamble.

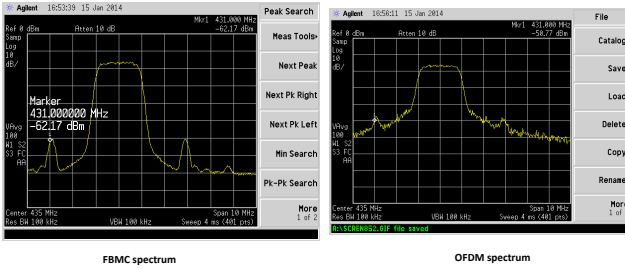


Fig. 2. Comparison between OFDM and FBMC spectrums

the FBMC waveform, we find a difference of almost 60dB between the activated subcarriers spectrum level and the noise level versus 30 dB for the OFDM waveform. Moreover, the spectrum level drops more rapidly compared to the OFDM one. These advantages provides to the FBMC a good opportunity to substitute the OFDM in next generation technology [11].

For the channel estimation, we insert pilots according to the Pair Of Pilots (POP) concept [12].

In order to make the FBMC system as LTE-like as possible, we will only consider only the first approach where the frame structure is based on a 10 ms frame length consisting of 150 FBMC symbols per frame. However, we will have a higher number of resource blocks per frame than the LTE system, with each block having the same number of resource elements. The time duration of each RB will be 25% shorter than for LTE, but this is transparent to higher level functionality. We are considering the scheme represented by Figure 3, on how to organize these frames.

In figure 4, a preamble consisting of six FBMC symbols is available for synchronization purposes. This preamble facilitates the use of a simple correlator in the time domain to find coarse symbol synchronization as well as the fractional



Fig. 4. Pair of Pilots based Preamble structure

When FBMC modulation is used, the complex-valued pilots are split into two real-valued PAM symbols (real and imaginary parts) transmitted with a time shift of $T/2$. This modulation, called OQAM, causes an intrinsic interference since the prototype filter has a length of $K \times T$ with $K > 1$ is the overlapping factor. In our considered case the overlapping factor is set to $K = 4$. One of the characteristics of the FBMC modulation is that the intrinsic interference is purely imaginary, that is, the orthogonality condition is only satisfied

in the real domain. The input-output model of the FBMC transceiver is given by

$$r_{k,n} = h_{k,n} (a_{k,n} + ju_{k,n}) + \eta_{k,n} \quad (1)$$

where $a_{k,n}$ is the PAM data symbol transmitted at the $k - th$ subcarrier and the time index n , $u_{k,n}$ is the real intrinsic interference term, $h_{k,n}$ is the channel coefficient, $r_{k,n}$ is the received signal sample, and $\eta_{k,n}$ is the Gaussian noise contribution, [11].

IV. FRAME RECOVERY MECHANISMS AT THE RECEIVER

The FBMC frame defined above provides us with two degrees of freedom to be exploited in order to estimate the effects of the unknown factors. These two elements are as follows: (i) known training sequence (preamble), (ii) pilot samples. We distinguish two types of synchronization : timing synchronisation and frequency one. We note that, in the presented prototype, the timing synchronization is done by a simple correlation between the transmitted preamble and the received one. Obtained results were robust and efficient. However, the frequency synchronisation is more complicated and it was implemented as follows.

A. Absolute Phase and CFO Estimation

The carrier frequency offset estimation can be done both in time and frequency domains. Note that the experiments are done in a static environment and the channel does not include any fading in the time domain.

A joint effect of the two phenomena at time t can be denoted by a phase shift of $\phi = \phi_a + 2\pi f_{cfo}t$, where ϕ_a and f_{cfo} indicate the absolute phase and the CFO values correspondingly.

Since the preamble signal, as shown in in Fig.4, overlaps with the data signal, the phase of the cross-correlation between the preamble signal and the corresponding part of the received signal is strongly corrupted by the data signal. Therefore, the coarse CFO estimation in time domain is not feasible. Consequently, we have to make use of the particular structure of the preamble and estimate the CFO using the preamble symbols P_k and P'_k as shown in Fig. 4. The effect of the CFO in frequency domain is that the received symbols in a given subcarrier are linearly phase rotated. Since the symbols are time spaced by $T/2$ (half a period), then we can write the received symbols for a given subcarrier k (noise apart) as

$$r_{k,n} = e^{j2\pi f_{cfo}n} h_k a_{k,n} \quad (2)$$

Therefore, we can write the received preamble symbols as:

$$y_{k,n} = e^{j2\pi f_{cfo}n} h_k p_k \quad \text{and} \quad y_{k,n+1} = e^{j2\pi f_{cfo}(n+1)} h_k p'_k \quad (3)$$

Consequently, calculating the ratio between the above expressions, we obtain

$$R_k = \frac{p'_k}{p_k} e^{j2\pi f_{cfo}} \quad (4)$$

Hence, the CFO can easily estimated by retrieving the phase of R_k and meaning the obtained phases over the subcarrier indices k .

TABLE I
SIMULATION PARAMETERS

Parameter description	Parameter value
Bandwidth	1.4 MHz
Sample frequency	1.92 MHz
Subcarriers number	128 subcarriers, 72 useful
Frame length in time	10 ms
Subcarrier spacing	15 kHz
FBMC filter	OFDM/OQAM PHYDYAS
FBMC symbol duration	66.67 μ s
Carrier frequency	430 MHz

B. Channel estimation

For the channel estimation, we are based on the PoP structure : we add an auxiliary pilot to get rid of the interference. In this step, we have considered the same algorithm as presented in [12].

V. EXPERIMENTAL RESULTS

The experiment consists of a complete chain of OQAM-FBMC radio transmission-reception as it is depicted in Figure 1. The host computers are each equipped with Intel Core I7-2600K processors and 4Go RAMs. To be able to work in a real-time condition, MATLAB 2011b is installed on Windows seven 32bits. The data frames are generated as described in section III, using 16-OQAM-FBMC symbols. The RF parameters at transmitter and receivers are configured to the values shown in Table I. We will distinguish two cases : the first one by considering the indoor wireless channel (without the channel emulator), in which the two entities are located in a distance of about 10 meters. This is an indoor scenario where the environment is almost static and there is an interior wall between transmitter and the receiver. The second case, is investigated by using the channel emulator with a static channel with five taps. The obtained results are shown by Fig. 5, and 6, respectively.

At the receiver side, N210 sends the down-converted digital samples to the host computer through the Ethernet interface. The baseband samples which are buffered at the entrance to the computer (namely Windows buffer) will gradually be pre-empted toward a MATLAB buffer, where from the signal processing units read the data. Such dual buffering method is commonly used in real-time systems.

The length of MATLAB buffer can be reconfigured. However, the maximum allowed buffer length is a packet of 46336 samples. Once the current contents of MATLAB buffer is processed, it will leave the buffer, and the next packet will be restored. If the packets are not preempted as fast as they get buffered in the Windows buffer, an overflow will happen when the Windows buffer is full. However, the amount of data which is stored before the overflow happens, can be processed in real-time.

In our experimental results, we have evaluated the execution time of the signal processing block for the frame recovery. The time spent to decode a complete MATLAB buffer is equal to 23 ms which is significantly smaller than the time spent before a packet of the same size get stored in the Windows buffer

(i.e. approximately 46ms when the MATLAB buffer length set to the maximum). Therefore, in our simulation setups, an overflow will never happen.

It is worth noticing that the final results presented by Fig. 5, and 6, of the end-to-end OQAM-FBMC transmission are very interesting and in a good agreement with what is expected. In fact, we see clearly that decisions can be made without ambiguity by considering the adequate threshold levels for 16-QAM symbols. We note also that the spread of constellation points over the imaginary parts can be explained by the intrinsic interference as explained by (1). This latter can be easily removed by taking the real part of the received symbols. From these constellations, one can deduce that the first scheme present a nul BER, and the second one presents an insignificant BER.

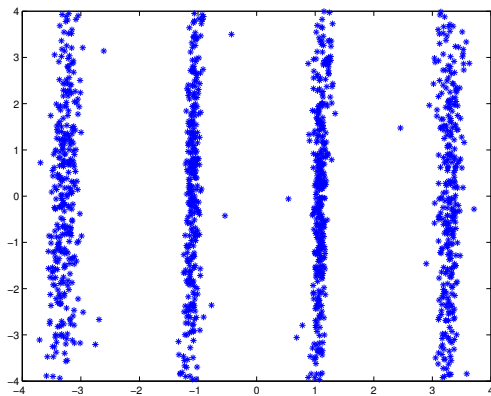


Fig. 5. 16-QAM constellation for indoor wireless channel

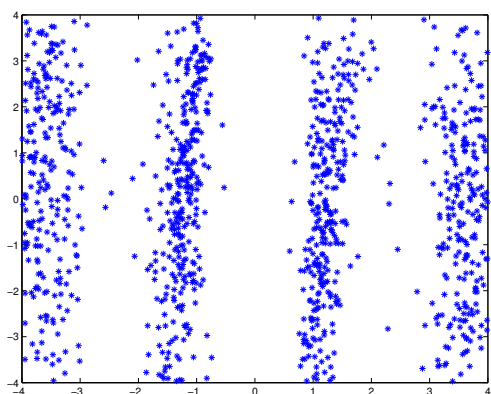


Fig. 6. 16-QAM constellation for multipath (five taps) wireless channel

VI. CONCLUSION

We have proved the feasibility of real time USRP board based prototype using MATLAB software for FBMC waveform. We have solved many practical problems mainly time and frequency synchronization and channel estimation and equalization which become more challenging in the context

of FBMC transmission. A study on real-time decoding of OQAM-FBMC radio frames is provided as an example. The baseband processes involved in decoding of OQAM-FBMC frames, namely the time and frequency synchronization, are implemented in MATLAB 2011b. Although synchronization is a classic problem which is addressed in many theoretical research works, considering the practical aspects of a real-time radio transmission, the signal processing block is required to satisfy a trade-off between the accuracy and the latency.

VII. ACKNOWLEDGMENTS

This work has been supported by the Minalogic competitiveness pole in the context of the SOAPS (Spectrum Opportunistic Access in Public Safety networks) project.

REFERENCES

- [1] J. Mitola, "Software radios: survey, critical evaluation and future directions," *IEEE Magazine on Aerospace and Electronic Systems*, April 1993.
- [2] [2] Matt Ettus, "Universal software radio peripheral," Tech. Rep., <http://www.ettus.com>.
- [3] Jorge Baranda, Pol Henarejos, Yan Grunenberger, and Montse Njar, "Prototyping with SDR: a quick way to play with next-gen communications systems," in *IEEE International Symposium on Wireless Communication Systems (ISWCS)*, 2011, pp. 16–20.
- [4] F. Kaltenberger G. Humer C. Mehlhruer, M. Rupp, "A scalable rapid prototyping system form Real-Time MIMO OFDM Transmissions," in *2nd IEE/EURASIP Conference on DSP enabled Radio*, 2005.
- [5] L. Esteve J. Arribas C. Fernandez-Prades, C. Aviles and P. Closas, "Design Patterns for GNSS Software Receivers," in *NAVITEC, 5th ESA Workshop*, 2010.
- [6] Guillaume Villemaud Wei Zhou and Tanguy Risset, "Full Duplex Prototype of OFDM on GNURadio and USRPs," in *IEEE Radio Wireless Symposium*, 2014.
- [7] P. Siohan, C. Siclet, and N. Lacaille, "Analysis and Design of OFDM/OQAM Systems Based on Filter bank Theory," *IEEE Trans. Signal Process.*, vol. 50, no. 5, pp. 1170–1183, May 2002.
- [8] M.G. Bellanger, "Specification and Design of a Prototype Filter for Filter Bank based Multicarrier Transmission," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01)*, May 2001, vol. 4, pp. 2417 –2420.
- [9] M. Bellanger and J. Daguët, "TDM-FDM Transmultiplexer: Digital Polyphase and FFT," *IEEE Trans. Commun.*, vol. 22, no. 9, pp. 1199 – 1205, Sep. 1974.
- [10] M. Bellanger et al, "FBMC physical layer: A primer," Tech. Rep., PHYDYAS (Physical layer for dynamic access and cognitive radio)[Online]. Available: <http://www.ict-phydyas.org>, Feb. 2010.
- [11] FP7-ICT PHYDYAS - Physical Layer for Dynamic Spectrum Access and Cognitive Radio., <http://www.ict-phydyas.org>.
- [12] C. L  l  , *OFDM/OQAM : m  thodes d'estimation de canal, et combinaison avec l'accs multiple CDMA ou les syst  mes multi-antennes*, Ph.D. thesis, Conservatoire National des Arts et M  tiers, Paris, France, 2008.
- [13] Deliverable D2.1, "Data-aided Synchronization and Initialization (Single Antenna)," Tech. Rep., European project ICT-211887 PHYDYAS, Jan. 2009.