

# SDR in teaching: An OFDM communication primer

Milan Zivkovic, Johannes Schmitz, Rudolf Mathar  
Institute for Theoretical Information Technology  
RWTH Aachen University  
Aachen, Germany  
{zivkovic, schmitz, mathar}@ti.rwth-aachen.de

**Abstract**—In this paper we describe an SDR based framework for adaptive OFDM transmission that enables implementation and experimental evaluation of various communication scenarios using a large set of adaptable parameters. The discussed framework is used as a valuable educational tool that helps students to efficiently internalize different aspects of communication systems including implementation issues and performance degradation caused by hardware imperfections. Moreover, interactive demonstration based on RF laboratory experiments allows for visualization of advanced communication concepts such as adaptive transmission with power/rate control. We describe the laboratory exercise in some detail emphasizing the educational benefit and illustrating synergy effects between research and teaching activities.

**Index Terms**—OFDM, SDR, GNU Radio, USRP, Laboratory, Teaching

## I. INTRODUCTION

During the last decade, Orthogonal Frequency Division Multiplexing (OFDM) [1] has established itself as a reliable high data-rate transmission technology, widely accepted in current broadband wireless standards, such as WiFi, WiMAX, and LTE. OFDM is a multicarrier modulation scheme which provides strong robustness against intersymbol interference (ISI) by dividing the broadband channel into many orthogonal narrowband subchannels in such a way that attenuation across each subchannel stays flat. However, an important task in the design of future OFDM based system is to exploit frequency diversity offered by broadband channels using adaptable transmission parameters (bandwidth, coding/data rate, power) to preserve power and bandwidth efficiency according to the sub-channel conditions at the receiver.

Furthermore, the investigation and assessment of information theoretic concepts for wireless resource management in real-world scenarios requires flexible testbeds with a wide range of reconfigurable parameters. These functionalities are currently offered only by Software Defined Radio (SDR) technology based on general purpose hardware. Additionally, in the last few years SDR is starting to be widely used for educational purposes, as it is shown to be a valuable teaching tool that allows for “painless”

This work was partially supported by the UMIC research cluster of RWTH Aachen University.

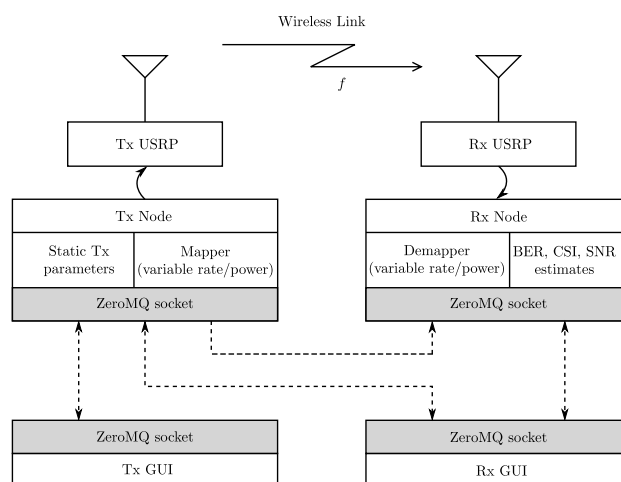


Fig. 1. The system overview

transition between the theory taught in the classroom and real-world issues associated both with the implementation of communication systems and over-the-air scenarios [2].

During recent years, as a part of our research efforts, we developed TIGR [3], a modular, GNU Radio [4] based reconfigurable framework at the Institute for Theoretical Information Technology (TI). TIGR features a library of signal processing blocks, control and feedback mechanisms, and an interactive Graphical User Interface (GUI) that enable for adaptive OFDM transmission with a large set of adaptable parameters. Moreover, TIGR advanced into an interactive educational tool through which students get introduced to SDR technology and the concept of adaptive wireless communication.

This paper introduces the basic architecture and functionalities of TIGR and provides a detailed description of the laboratory experiment that utilizes the flexibility of SDR for educational purposes. Since the duration of the considered laboratory exercise is limited to three hours, the tasks are based on interactive demonstrations, where students can learn about particular concepts through modification of parameters and visualization of the results. Interactive demonstrations and computer-based exercises are often applied in signal processing education. They are shown to improve learning by enhancing understanding, motivating students, and connecting theory with practical

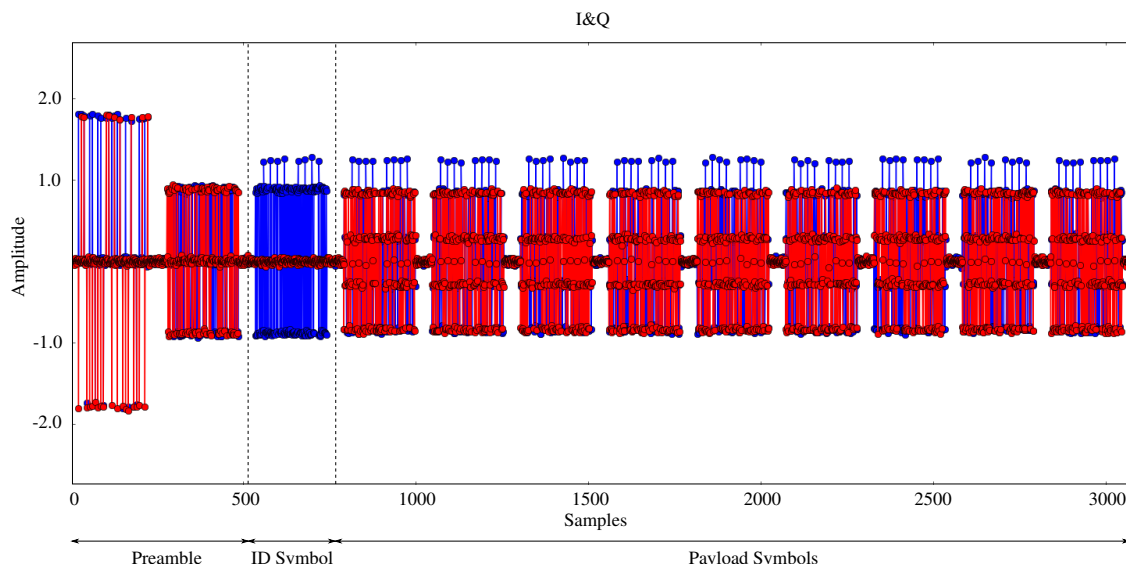


Fig. 2. Frame structure

concerns [5].

The flexible architecture and reconfigurability makes the SDR a convenient tool for interactive demonstrations, which allows exploration of both hardware and software issues and emulation of real RF scenarios.

The remainder of this paper is organized as follows. Section II provides the basic overview of TIGR functionality with separate introduction of different system components. A roadmap of laboratory experiments based on TIGR is described in Section III. Finally, some concluding remarks are given in section IV.

## II. TIGR SYSTEM OVERVIEW

The system diagram of TIGR framework is shown in Fig. 1. The transmitter and receiver nodes are composed of a host commodity computer and general purpose RF hardware, Universal Software Radio Peripheral (USRP) [6]. The baseband signal processing at the host computer is implemented in the GNU Radio framework, while the USRP performs computationally intensive operations such as filtering, up- and downconversion controlled through a robust application programming interface (API) provided by USRP Hardware Driver (UHD) [7]. The communication between the transmit and receive node is organized as a reconfigurable continuous one-way transmission of OFDM symbol frames. As shown in Table I, input configuration parameters are divided into two sets. The set of static parameters containing FFT size, the number of subchannels,

TABLE I  
TIGR OFDM SYMBOL PARAMETERS

|                                      |   |
|--------------------------------------|---|
| Bandwidth ( <i>static</i> )          | Variable, up to $8\text{MHz}$                               |
| FFT length ( <i>static</i> )         | 64 – 1024   |
| Frame length ( <i>static</i> )       | Variable  |
| Carrier frequency ( <i>dynamic</i> ) | 2400 – 2483MHz  |
| Modulations ( <i>dynamic</i> )       | BPSK, QPSK, 8-PSK, 16-QAM, 32-QAM, 64-QAM, 128-QAM, 256-QAM |
| Power ( <i>dynamic</i> )             | Up to 20 mW   |

frame size, etc., is initialized at the beginning of the transmission and is known to both nodes. The set of dynamic parameters, which are reconfigurable at run-time includes the total transmit power, carrier frequency, and allocated power/rate over subcarriers. TIGR OFDM frame structure is shown in Fig. 2. The sequence of nine data symbols is preceded with two preambles (one synchronization preamble and one used for channel estimation) and one ID symbol that is used for the synchronization.

Each frame contains a unique ID associated with the current parameter settings. The decoded ID enables the receiver to perform a parameter lookup necessary for demapping. The forwarding of meta information to the receiver is implemented by the ZeroMQ communication library [8], [9] over Ethernet. Moreover, the ID is utilized both at the transmitter and receiver to determine the random bit sequence used for BER calculation.

The transmission parameters such as carrier frequency, transmission power, and power/bit loading on the subcarriers, can be reconfigured on-the-fly. The GUI is further used to observe the impact of these changes on the received signal by showing estimated SNR, normalized channel state information, carrier frequency offset (CFO), and measured

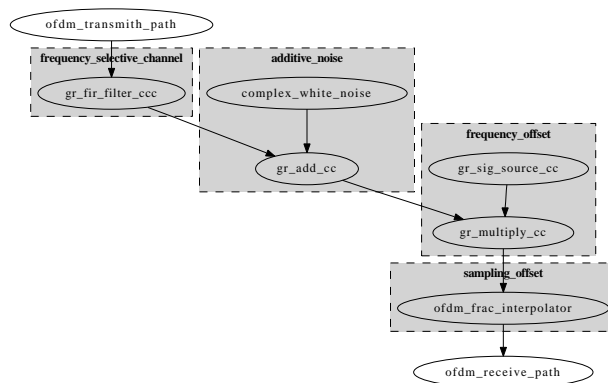


Fig. 3. Simulated channel between the transmit and receive chain

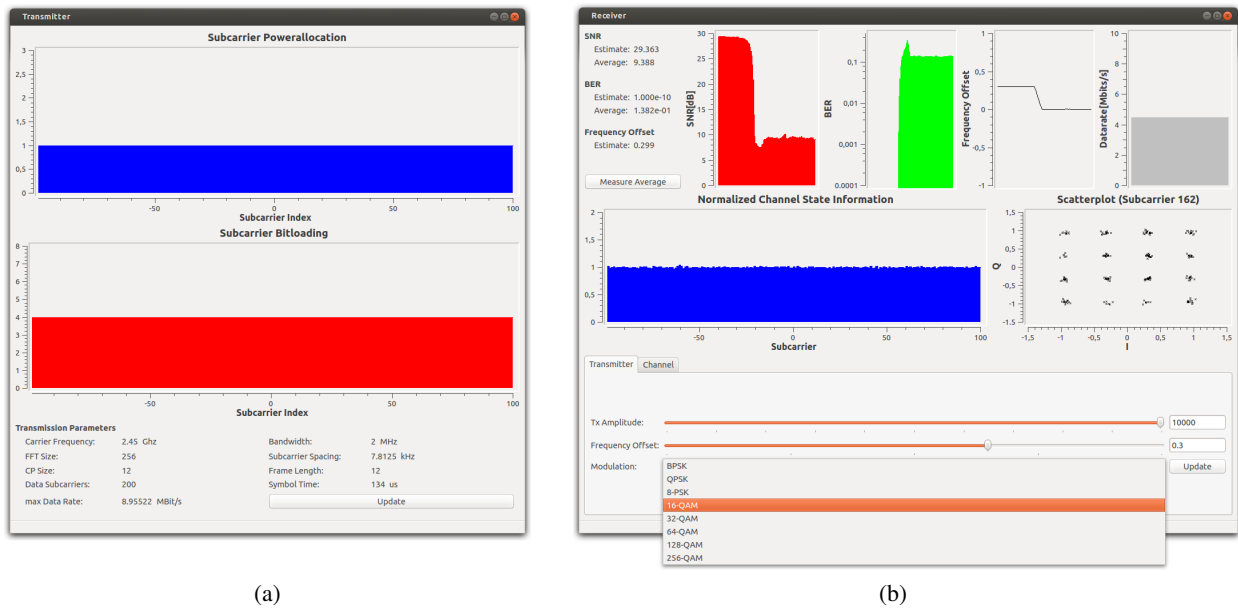


Fig. 4. The transmitter's GUI (a) and receiver's GUI (b)

BER. Moreover, TIGR can operate in two modes: the first mode assumes the RF transmission where the transmitter and receiver are interfaced with the USRPs; the second is the simulation mode where the transmitter and receiver “communicate” over an emulated channel, without the RF interface (USRPs). This allows for the exclusion of unwanted distortions caused by the hardware, but also for system evaluation in the presence of various controllable channel and hardware impairments, such as CFO, sampling frequency offset (SFO), additive noise, interference, and multipath propagation. An example of a simulated channel is shown in Fig. 3.

#### A. GUI

The Graphical User Interface (GUI), facilitating the demonstration, is developed in PyQt [10], a Python binding for the Qt framework. The simplicity, high modularity, and interpreted nature of Python combined with the rich collection of Qt GUI widgets makes PyQT a powerful framework for fast prototyping of the interactive GUI elements. The transmitter's GUI contains the static transmission parameters and the current allocation of rate and power over subchannels, as shown in Fig. 4a. Furthermore, the receiver's GUI, given in Fig. 4b, dynamically shows estimated signal parameters (average SNR, CSI, BER, CFO, and a scatterplot of the specified subcarrier) and contains an interactive interface for on-the-fly configuration of transmitted power and allocation strategy at the transmitter.

#### B. ZeroMQ Integration

The control and feedback backbone as well as parameter visualization of TIGR is realized over local Ethernet using ZeroMQ, a message-oriented middleware library that enables a distributed communication models.

At the beginning of TIGR development in 2007, the backbone implementation [3] was based on CORBA [11],

an object oriented middleware solution, that was standard at the time. However, due to frequent fixes and extensions, CORBA is inherently complex to maintain and further extend. Since then, new transport frameworks with better design, usability and supported concepts have been introduced.

The new releases of the official GNU Radio project base their command and control interface, named Control Port [12], on ICE [13]. Conceptually very similar to CORBA, ICE is an object-oriented middleware library, that eliminates some of CORBA drawbacks. However, our search for a user friendly lightweight solution which would enable easy integration, good scalability, low latency, and small memory footprint led to the decision in favor of ZeroMQ [14].

ZeroMQ (also referred as ØMQ, 0MQ, or ZMQ) is a high-performance asynchronous messaging library for designing distributed and concurrent applications. Unlike other middleware solutions, a ZeroMQ system can run without a dedicated message broker<sup>1</sup>. Moreover, ZeroMQ runs on different architectures and supports more than 40 programming languages.

ZeroMQ transports data using “message batching” (i.e. sending messages in batches rather than one by one), which introduces low latency thanks to smart batching algorithms. Moreover, the ZeroMQ has a small memory footprint compared to the conventional messaging systems [14].

There are two major features of ZeroMQ. At first, the ability to simply bind or connect a socket to multiple endpoints. Secondly, the support of different underlying transport protocols for a socket such as TCP, PGM (used for communication between remote hosts), and inproc (optimized for sending messages between threads within the same application).

<sup>1</sup>Traditional message queuing systems contain a central message server over which each system node communicates with other nodes, thus without direct communication.

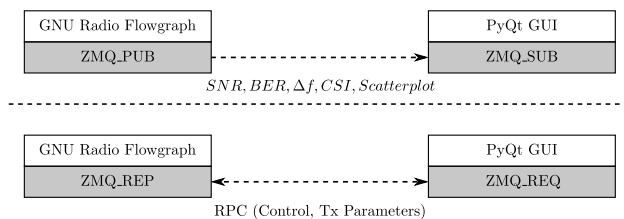


Fig. 5. ZeroMQ communication model (ZeroMQ integration within TIGR)

Beside simple socket send-receive calls, ZeroMQ implements many other complex communication patterns such as request-reply, publish-subscribe, and produce-consume.

We developed a GNU Radio out-of-tree module that implements ZeroMQ [15] communication patterns to enable low latency backbone mechanisms within TIGR. As shown in Fig. 5, estimated parameters that indicate link quality are published from the GNU Radio flowgraph using ZMQ\_PUB sockets implemented as a GNU Radio sink block. At the GUI, a ZMQ\_SUB socket subscribes to the specific publisher to provide data to the GUI objects. In this way, the transceiver system can run independently of the GUI activity since the ZMQ\_PUB sockets make no assumptions on the number of connected subscribers.

Moreover, the control mechanism in TIGR is based on remote procedures calls (RPC) that implement request-reply communication pattern. The interactive receiver's GUI, shown in Fig. 4b, uses a ZMQ\_REQ socket to send requests to the transmitter's GNU Radio flowgraph to change transmit power or change the bit/power allocation over subcarriers. Based on this request ZMQ\_REP socket replies and triggers the change within the transmitter's flowgraph.

### III. THE EXERCISE STRUCTURE

We created the experiment "Implementing an OFDM Transceiver by Software Defined Radio" for a laboratory offered as a part of Communications Engineering and Computer Engineering undergraduate programs at RWTH Aachen University. Most of the TIGR functionality utilized for the laboratory exercise can be depicted using the interactive GUI shown in Fig. 4.

Considering the students' prior knowledge on the topic, the laboratory exercise is accompanied with an appropriate script [16] that contains basic description of the OFDM technology, corresponding implementation issues, as well as an introduction to the SDR concept with emphasis on the GNU Radio framework. Prior to performing the exercise, the students are required to read the script and finish a preparatory exercise for better understanding of the presented concepts.

To get familiar with the concept of communication channels and its capacity and quality, the students are required to plot BER vs. SNR curves according to given analytical expressions such that they can "acquire" the basic feeling of the impact of the received signal energy to data rates. As a part of this task, based on the produced figure, students have to determine SNR values required to achieve given

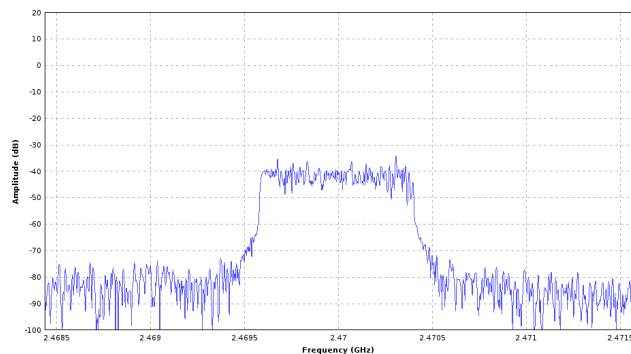


Fig. 6. Spectrum of the generated OFDM signal at the input of the receiver

BER ( $10^{-2}$  and  $10^{-3}$ ) for different modulations. The second preparatory task is also based on analytical results on SNR loss caused by the presence of CFO, as one of the main issues in the OFDM receiver. This helps students to gain basic understanding how different magnitudes of CFO influence the signal quality expressed by the SNR. Corresponding results are shown by solid curves in Fig. 11.

At the beginning of the laboratory exercise, a short presentation of basic concepts given in the script is shown, followed by a short discussion of preparation exercise as an introduction to the laboratory tasks. The duration of the laboratory exercise is three hours and is divided into four parts where each of them introduces different aspects of the OFDM communication. This includes RF transmission of an OFDM signal, experimental derivation of rate-power functions, which are used as main building blocks for the adaptive transmission, and introduction to the hardware imperfections and their influence on the system performance. Each part of the exercise is followed by a discussion where obtained measurements are being automatically logged and given to the students as a part of the final report.

The students are divided into three groups, each with two-to-four participants, working with two PCs, each equipped with a USRP. During the exercises with RF transmission, different carrier frequencies are dedicated for each Tx-Rx pair in order to avoid mutual interference among groups.

In the following, we provide the description of the exercises.

#### A. OFDM Transmission Parameters

In the first (introductory) task, students are required to perform RF transmission of an OFDM signal at the dedicated carrier frequency. Using the GNU radio based integrated tool shown in Fig. 6, the students are required to observe the spectrum of the received signal and determine the carrier frequency and bandwidth of the transmitted signal. Moreover, using the transmitter's GUI shown in Fig. 4a, students are instructed to note down other system parameters such as FFT length, number of data subcarriers, and length of cyclic prefix. Finally, students have to read the value of estimated normalized CFO from the receiver's

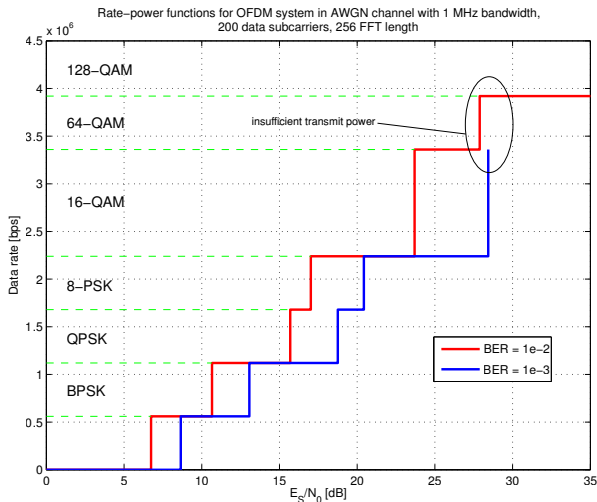


Fig. 7. Rate-power functions for RF transmission of an OFDM system with 1 MHz bandwidth, 200 data subcarriers, 256 FFT length

GUI shown in Fig. 4b and calculate the CFO in Hz given the previously determined system parameters.

### B. Rate-Power Function

In the second task, based on a small measurement campaign, students are required to reproduce the experimental rate-power functions that are used as main building blocks for the adaptive transmission [1], [17]. The task is to match the measured BER to the predefined value (for  $10^{-2}$  and  $10^{-3}$ ) for different modulation schemes (from BPSK to 128-QAM) by changing the transmitted signal power (performed by updating the transmit amplitude in the GUI, as shown in Fig. 4b). After the BER is matched, the students have to note down the corresponding estimated BER, SNR, and achieved data rate. Based on this, students have to produce the experimental rate-power functions using a given Matlab script. The results of the particular experiment performed by one student group is shown in Fig. 7. It can be seen that the maximum transmit power is insufficient for 128-QAM to achieve the BER of  $10^{-3}$ . Besides produced rate-power function, Fig. 8 is used as a control visualization of how well BER requirements are fulfilled during the experimentation.

### C. Efficient Use of Frequency Selectivity

The third task of the laboratory exercise showcases the utilization of a frequency selective channel for the system performance improvement. The goal of this task is to demonstrate the bandwidth efficiency of an OFDM system in frequency selective channels. The experiment is based on the TIGR simulation model, since the real channel environment for the supported transmission bandwidth experiences only minor multipath propagation. First, students have to simulate the OFDM system where *all* subcarriers are loaded with BPSK symbols. The required BER is reached by fitting the transmit amplitude, similarly to the previous task. Then, applying the same average transmit power, the OFDM system is simulated using the *better half* of the subcarriers (in this particular case the middle part of

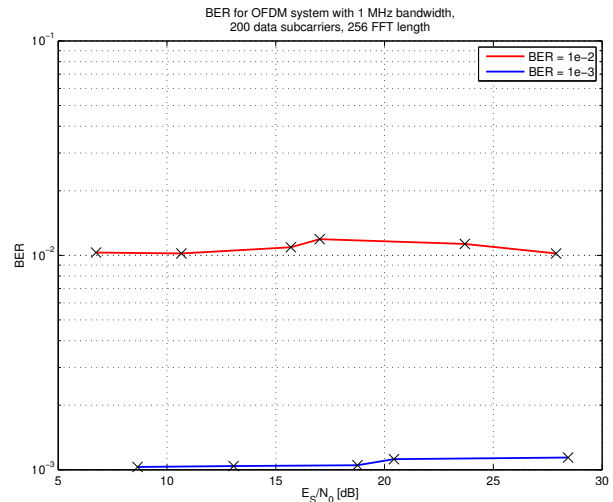


Fig. 8. Measured BERs

the occupied band) loaded with QPSK symbols with double power, thus preserving the same data rate while halving the bandwidth usage. The occupation of subcarriers per modulation scheme is depicted in Fig. 9. The experimentally evaluated performance improvement is shown in Fig. 10. The students are encouraged to deduce that the efficient use of dedicated bandwidth can increase the quality of the transmission. More specifically, lower BER for the same data rate and same transmission power is achieved if the *better half* of the channel is used.

### D. Frequency Offset Analysis

Finally, in the fourth part, while operating TIGR in simulation mode, students get introduced to the issue of CFO offset and its influence on the system performance. Without loss of generality, the experiment is conducted only for the QPSK modulation scheme.

The task is to simulate the SNR loss curves as a function of normalized CFO with SNR as a parameter. The students are required to fit the transmit amplitude by updating its value in the interactive GUI to match the measured SNR value to the nominal SNR value (5, 10, 15, and 20 dB). When the SNR is fitted, students have to note down the transmit amplitude. Afterwards, while using the noted transmit amplitude, students are required to add additional CFO by updating its value in the GUI and measure the SNR. Based on the measured SNR in the presence of CFO, students are required to calculate the *SNR loss* and

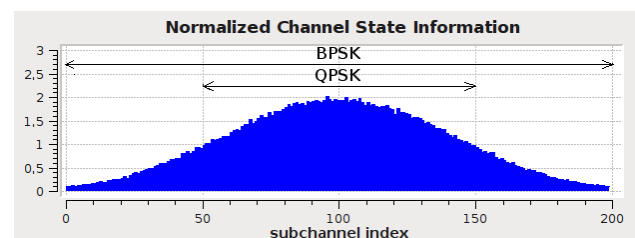


Fig. 9. Using all subcarriers with BPSK and using the better half with QPSK



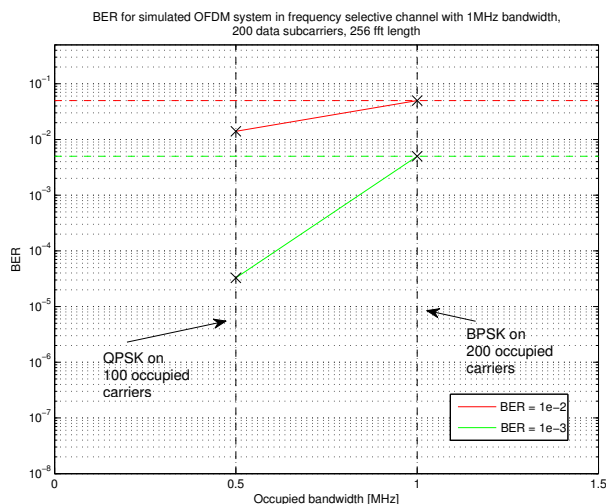


Fig. 10. BERs for simulated OFDM system in frequency selective channel with 1 MHz bandwidth, 200 data subcarriers, 256 FFT length

produce the SNR loss curves using the prepared Matlab script. The corresponding (dashed) curves of one particular experimental trial are shown in Fig. 11 together with the solid ones that correspond to analytical expressions.

Simulation curves suppose to match the analytical results and showcase the feasibility of the analytical model. Moreover, the high variance of SNR estimates in low SNR region influences the results and illustrates the low reliability of estimation algorithms under those conditions.

#### E. Educational Benefits

The students are coming from the fifth semester of Bachelor studies without prior knowledge of OFDM technology and SDR. By using SDR as a tool to demonstrate different aspects of OFDM students get familiar with both concepts. First, they gain in vivo experience about how the high modularity and availability of a large set of adjustable parameters provided by SDR can be used for experimentation and evaluation. Secondly, they get familiar with the concept of adaptive transmission and practical issues of an OFDM transmission such as the influence of the CFO mismatch on the system performance. This interactive demonstrations visualizes how signal quality determines the performance of communication systems and how adaptive systems operates in real RF environment.

#### IV. CONCLUSION

In this paper we have discussed the educational capabilities of TIGR, a reconfigurable framework for an adaptive OFDM transmission based on GNU Radio platform and ZeroMQ backbone implementation. TIGR extends PHY layer functionality of current wireless standards and offers control and feedback mechanisms for easy reconfiguration of transmission parameters. This allows evaluation of different strategies in either simulation or over-the-air scenarios. TIGR furthermore advanced into an interactive educational tool that effectively helps students to internalize the concept of adaptive communication and corresponding

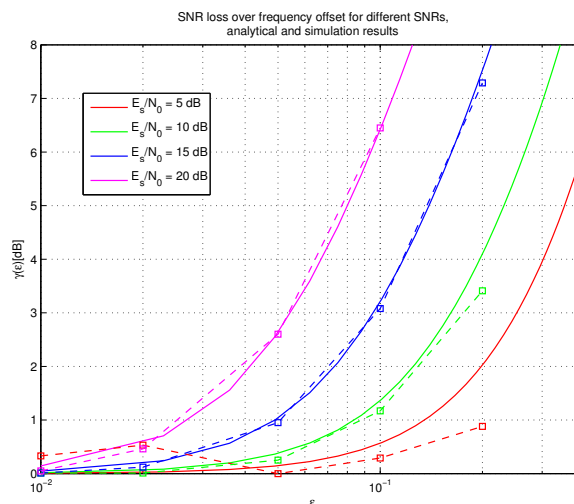


Fig. 11. SNR loss vs. frequency offset for different SNR as parameter, analytical and simulation results

practical issues while introducing them to the capabilities of SDR technology. Finally, after several years of experience with TIGR in a joint research and teaching environment we can claim that the flexibility of SDR has resulted in considerable synergy effects.

#### REFERENCES

- [1] Y. G. Li and G. L. Stuber, *Orthogonal Frequency Division Multiplexing for Wireless Communications*. Springer, 2006.
- [2] J. Gunther and T. Moon, "Synchronization and demodulation programming projects to accompany a first course on digital communications," in *Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE), 2013 IEEE*, 2013, pp. 311–316.
- [3] M. Zivkovic and R. Mathar, "Design issues and performance evaluation of a SDR-based reconfigurable framework for adaptive OFDM transmission," in *Proceedings: ACM international workshop on Wireless network testbeds, experimental evaluation and characterization (WiNTECH'11)*. New York, NY, USA: ACM, 2011, pp. 75–82.
- [4] "GNU Radio." [Online]. Available: <http://gnuradio.org/redmine/projects/gnuradio>
- [5] S. Shearman, S. Hong, and E. Perez, "To improve signal processing concept demonstration software, consider the interactive experience," in *Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop, 2009. DSP/SPE 2009. IEEE 13th*, 2009, pp. 227–231.
- [6] "USRP." [Online]. Available: <http://www.ettus.com>
- [7] "USRP hardware driver." [Online]. Available: <http://code.ettus.com/redmine/ettus/projects/uhd/wiki>
- [8] "ZeroMQ." [Online]. Available: <http://zeromq.org>
- [9] F. Akgul, *ZeroMQ*. Packt Publishing, 2013.
- [10] "PyQt." [Online]. Available: <https://wiki.python.org/moin/PyQt>
- [11] "Object Management Group." [Online]. Available: <http://www.omg.org/>
- [12] T. Rondeau, "GNU radio's control port." [Online]. Available: [http://gnuradio.squarespace.com/storage/grcon13\\_presentations/grcon13\\_rondeau\\_controlport.pdf](http://gnuradio.squarespace.com/storage/grcon13_presentations/grcon13_rondeau_controlport.pdf)
- [13] "ZeroC ICE." [Online]. Available: <http://www.zeroc.com>
- [14] A. Dworak, P. Charrue, F. Ehm, W. Sliwinski, and M. Sobczak, "Middleware trends and market leaders 2011," *Conf. Proc.*, vol. C111010, no. CERN-ATS-2011-196, p. FRBHMULT05. 4 p, Oct 2011.
- [15] "gr-zmqblocks." [Online]. Available: <https://github.com/iohannesz/gr-zmqblocks>
- [16] "Laboratory exercise script: Implementing an OFDM transceiver by software defined radio." [Online]. Available: <http://www.ti.rwth-aachen.de/teaching/laboratory/data/SkriptWS1213.pdf>
- [17] M.-O. Pun, M. Morelli, and C. C. J. Kuo, *Multi-Carrier Techniques For Broadband Wireless Communications: A Signal Processing Perspectives*. London, England, UK: Imperial College Press, 2007.