Rapid Prototyping and Validation on an SDR Platform of a Low-Cost Hybrid ML SNR Estimator over Time-Varying SIMO Channels

Zied Ben Gamra, Faouzi Bellili, Abdelaziz Samet, and Sofiène Affes
INRS-EMT, 800, de la Gauchetière Ouest, Bureau 6900, Montreal, H5A 1K6, Canada.
Emails: {zied.ben.gamra, bellili, samet, affes}@emt.inrs.ca

***Abstract*—An application of a rapid prototyping method on an SDR platform is presented. A novel low-cost hybrid maximum likelihood (ML)signal-to-noise ratio (SNR) estimator over single-input multiple-output (SIMO) time-varying fading channels, is implemented, tested, and validated. The approach adopted to achieve this work is model-based. It requires no coding since it is relies on MATLAB/Simulink and Xilinx Sytsem Generator (XSG) tools that offer a graphical user interface (GUI) and a drag-and-drop method. These software tools allow the creation of a design architecture for the estimator which is then translated into hardware description language (HDL) and implemented on the targeted SDR. Nutaq's PicoSDR2X2 platform and Anite's EB Propsim channel emulator define our experimental environment. The experimental results obtained in real-time and over-the-air corroborate those previously generated off-line by MATLAB.**

I. Introduction

Rapid prototyping is a quite recent design and production method. Owing to its several advantages, it is being used in different fields ranging from simple consumer products to very complex industrial systems. This technique consists in creating, in a rapid and optimized way, prototypes of the desired products and testing them in real time in order to detect design errors or improve the final product. Thus, it follows a time-saving and cost-minimizing strategy.

Due to its evolving aspect and constantly changing standards, the telecommunication sector has taken advantage of this new rapid prototyping technique and has adapted it to its needs. The main reason for that is the fact that this method allows engineers and designers to verify and optimize their designs, measure various varying parameters impacts, try different what-if scenarios early in the development process all in real time with co-design and co-simulation [1].

A concrete application of this technique is the Software-Defined Radio (SDR) or Software Radio which is a reconfigurable wireless system that can support multi-communication standards [2-3]. It allows system reprogramming by using only software commands using FPGAs and DSPs [4]. In an SDR, most radio receiver/processing functions would be defined and changed by user written software programs [5-6]. The concept of SDR was proposed to avoid costly hardware upgrades or total communications device replacement due to rapidly changing wireless communication protocols and standards [7]. Unlike conventional approaches, SDR does not require prior knowledge of hardware description languages since it can be programmed using high-level-language software. Thus, it constitutes a time-saving alternative, suitable for rapid prototyping.

Enhancing the performance of SDR transceivers is best done through the estimation process. Estimating channel parameters is crucial for better performance quality and optimization of adaptive wireless communication systems. Hence, we chose to work with a new low-cost hybrid maximum likelihood (ML) Signal-to-noise-ratio (SNR) estimator over single-input multiple-output (SIMO) time-varying fading channels that has been recently shown to outperform all benchmarks existing in the literature [8]. In this study, the SNR estimator is implemented on Nutaq's Pico SDR platform using MATLAB-Simulink, Xilinx System Generator from the hardware side and Nutaq's software tools (MBDK) from the software side. This work environment is ideal for the rapid prototyping of wireless communication algorithms.

This paper is organized as follows: In section II, we briefly describe the SNR estimator. In section III, we discuss the hardware implementation of this estimator. In section IV, we present our experimental hardware test-bench. In section V, we show and interpret the co-simulations results. Finally, in section VI, we conclude this work.

II. OVERVIEW OF ML SIMO SNR ESTIMATOR

The new low-cost estimator implements ML SNR estimation of linearly-modulated signals over time- varying flat-fading SIMO channels [8]. It offers a new technique to deal with maximum likelihood (ML) estimation of instantaneous (i.e., short-term or local) SNR. Instead of considering the channel to be constant over the entire observation period, its variations are locally tracked using a polynomial-in-time expansion.

It is composed of three different versions: data-aided (DA), non data-aided (NDA), and hybrid. The DA version uses known pilot symbols to achieve the estimation process and is derived in a closed analytical form. On the other hand, the NDA version is iterative and based on the expectation- maximization (EM) concept that guarantees convergence within very few iterations. As a consequence, the way it is initialized has an impact on its ultimate output. Thus, instead of initializing it arbitrarily, we opt for a more interesting solution which consists in using the output of the DA version as an initial input of the NDA one, thereby resulting in the so-called hybrid version SNR estimator. Due to lack of space, all mathematical details can be found in [8].

As illustrated in Figure 1, this new estimator outperforms the only benchmark existing in the literature [9], called WGM,

978-1-5090-1317-3/16/$31.00 ©2016 IEEE

which implementing ML and EM based SNR estimation over time-varying channels.

Moreover, this novel low-cost estimator offers two other main functions on top of SNR estimation: data demodulation and channel identification. Therefore, it can be seen as an integrated-system architecture.

Motivated by all the above, we propose and develop in this work a new model-based hardware architecture and design to implement this estimator and then integrate it and test it on PicoSDR2X2 SDR platfrom.

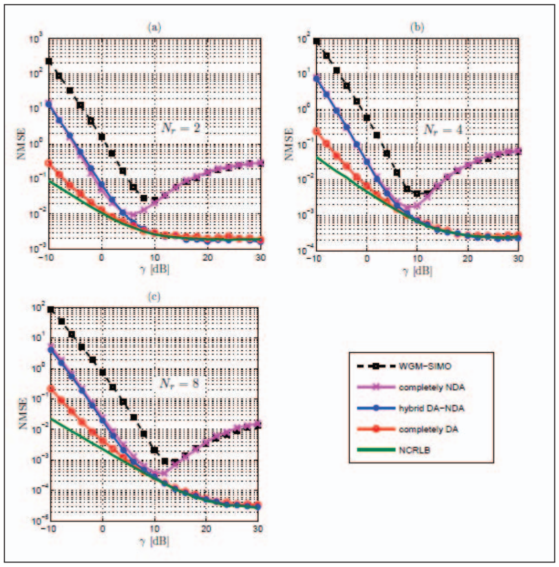

Fig. 1: Comparison between the chosen estimator and WGM-SIMO for different antennae numbers: (a) $N_r = 2$, (b) $N_r = 4$, et (c) $N_r = 8$, with $F_D T_s = 7 \times 10^{-3}$, $N = 112$, $\bar{N}_{DA} = 112$, $\bar{N}_{NDA} = 56$ and $L = 4$, QPSK.

III. New Hardware Design and Implementation

For the design of the hybrid version of the SNR estimator, we had to set the parameters for the most convenient values while taking in consideration the system complexity and hardware available resources. We chose to work with a SIMO antenna configuration i.e., $Nr = 2$. The polynomial-in-time expansion order is $L_{DA} - 1 = L_{NDA} - 1 = 3$. The modulation scheme is QPSK i.e., $M = 4$. The observation window size is $N = 112$ divided into two approximation windows of size $\bar{N}_{NDA} = 56$ i.e., $k = 2$ for the NDA version and $\bar{N}_{DA} = 112$ for the DA one. The symbol period is $T_s = 7.142 \times 10^{-5}$ and the pilot symbol period is $T_s' = 7 \times \mathrm{T}_s$. The maximum Doppler spread is $F_D T_s \leq 7 \times 10^{-3}$.

Implementing the SNR estimator on Nutaq's SDR platform is done in phases. First, we start by the conception phase where the Platform Independent Model (PIM) is created. The system is divided into modules, each one performing a specific task. For each module, the equations and their in/outputs are identified which is known as the data path. The latter is, actually, composed of the totality of resources used to perform the needed task. In other terms, it is the circuit that the in/output data is going to follow. Also, during the conception phase, the finite state machines (FSM) for each module are conceived. This constitutes the control path which contains the resources used to control data flow through data path. It generates a set of control signals that organize and time the functioning of the different entities which execute the treatment. Then, in the development phase, the data and control paths are designed using MATLAB/Simulink and Xilinx System Generator and MBDK Nutaq Sofware. Following this, in the testing phase, software simulations are run and results are validated. Next, is the proper implementing phase where the Platform Specific Model (PSM) is generated and implemented on the hardware platform. Finally, hardware simulations are run in real-world conditions using the channel emulator.

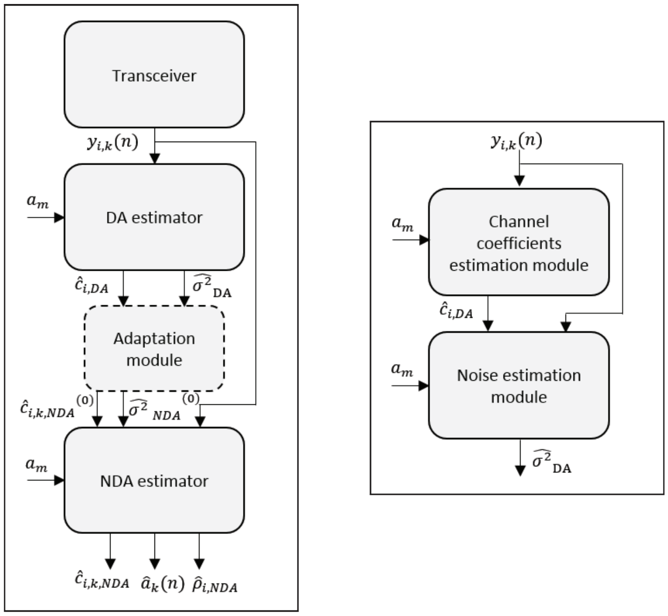

Fig. 3: Bloc diagram of DA estimator.

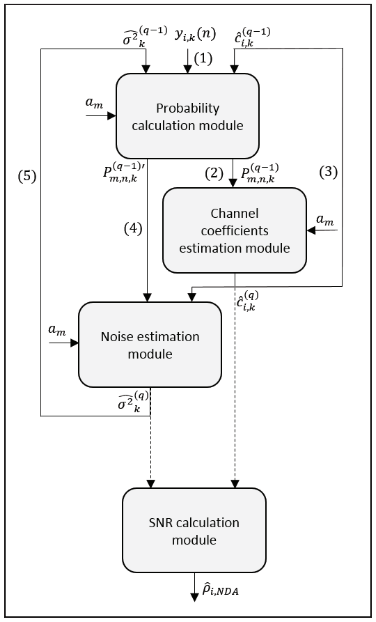

Fig. 4: Bloc diagram NDA estimator.

The MATLAB/Simulink and Xilinx design is composed of four modules as shown in Figure 2:

- The tranceiver module.
- The DA estimator module.

- The adaptation module.
- The NDA estimator module.

First, a QPSK sequence is sent through the wireless transmission channel, received and saved using the SDR 420m radio card. The pilot symbols are then extracted and passed to the DA estimator module. Second, the DA estimator module uses these pilot symbols, as shown in Figure 3, to:

- Estimate the channel coefficients $\widehat{\mathbf{c}}_{i,DA}$ for each antenna i.
- Estimate the noise power $\widehat{\sigma^2}_{DA}$.

Third, these estimates go through the adaptation module to serve as initial guesses for the NDA module which iterative algorithm, shown in Figure 4, runs as follows:

1) The first module calculates the symbol probability $P_{m,n,k}^{(q-1)}$ using the initial guesses in the first iteration and using the same updated parameters in the following iterations.
2) The following module estimates the channel coefficients $\widehat{\mathbf{c}}_{i,k}^{(q)}$ using $P_{m,n,k}^{(q-1)}$.
3) the symbol probability $P_{m,n,k}^{(q-1)}{\prime}$ is calculated again using the new channel coefficients estimates.
4) the noise power $\widehat{\sigma^2}^{(q)}$ is estimated using the new symbol probability $P_{m,n,k}^{(q-1)}{\prime}$ and the channel coefficients $\widehat{\mathbf{c}}_{i,k}^{(q)}$.
5) If the number of iterations q is not reached yet, the new estimated parameters $\widehat{\mathbf{c}}_{i,k}^{(q)}$ and $\widehat{\sigma^2}^{(q)}$ are re-injected into the first module in order to start a new iteration.

At the end of all iterations, the last module calculates the SNR $\widehat{\rho}_{i,\mathrm{NDA}}$ for each antenna i.

IV. Experimental Test-Bench

All rapid prototyping software frameworks have for main purpose the translation of a high-level description to a real-time hardware implementation. This would result in the automation of a big part of the initially needed development work. The benefits are, on the one hand, saving time and in the other hand, allowing non experts in electronic devices to develop applications without requiring a deep knowledge of a hardware description language. Regarding our application, we worked with Nutaq's Pico SDR platform and EB Propsim channel emulator on the hardware side and MATLAB-Simulink, Xilinx System Generator and Nutaq's software tools (MBDK) [10] on the software side. This work environment is ideal for rapid prototyping and wireless telecommunication algorithms implementation which perfectly meets our needs.

A. MATLAB-simulink and Xilinx System Generator

MATLAB-Simulink [11] combined with Xilinx System Generator represent a powerful tool for algorithms conception, test and implementation on embedded systems. It offers the possibility to use MATLAB high-abstraction language and Simulink and Xilinx graphic blocks simultaneously to create the algorithm architecture and the tasks to perform on the hardware platform. The proposed architecture can be tested and simulated before being implemented in order to correct errors and confirm the results. Also, it can be co-simulated on hardware so as to optimize it and verify its real time functioning. More important, the algorithm translation to a hardware description language such as VHDL is done automatically which allows a huge development time saving. To finish, the system generates the bit stream of the application which is directly implemented on the hardware platform. Hence, MATLAB-Simulink and XSG offer complete software set for rapid prototyping.

B. Nutaq's SDR and tools

Nutaq's Pico SDR is a hardware simulation platform used to model, test and deploy wireless telecommunication algorithms rapidly and accurately. The platform is provided with a set of software tools: Model-Based Development Kit (MBDK) that enable remote control and data exchange between Nutaq's hardware and user computer and offer co-simulation tools allowing real time integration of the platform input/output while simulating [12]. It is essentially composed of:

- A large Xilinx Virtex-6 FPGA which has increased digital signal processing resources.
- A Perseus 601X advanced mezzanine card (AMC) which is a digital signal processing card intended for high-performance, high-bandwidth and low-latency processing applications. It is perfect for reducing the size, complexity, risks and costs associated to leadingedge telecommunications.
- A 420m FMC radio card which is a multimode software defined radio transceiver module. It supports a broadband coverage, as well as TDD (time division duplex) and FDD (frequency division duplex) full duplex modes of operation and has a bandwidth going from 1.5 to 28 MHz with excellent channel selectivity.

C. The EB Propsim FS8 channel emulator

In order to simulate radio channel properties, we use the EB Propsim channel emulator commercialized by Anite. It enables us to emulate radio channel characteristics like path loss, multi-path fading, delay spread, Doppler spread, polarization, correlation and spatial parameters critical for MIMO and multiradio system performance. With its 8 RF channels, it is able to generate a 4x4 MIMO bidirectional channel for testing the wireless performance of mobile devices and chipsets and perform studies on various technologies such as 2G, 3G and 4G LTE [13].

V. Simulation results

The last step of testing and validation of the PSM on the SDR platform is achieved through co-simulations using Nutaq's MBDK tools. In order to ensure that the PSM is giving the same results as those of MATLAB, we will set the number of iterations q to 10 and run a Monte-Carlo simulation of 5000 realisations.

For the SNR estimation, as shown in Figure 5, both NMSE curves coincide in each point and give the expected performance. Therefore, the PSM is behaving in the same way as the MATLAB algorithm despite the precision loss due to the fixed-point representation. Consequently, the SNR estimation is validated on the PicoSDR2X2.

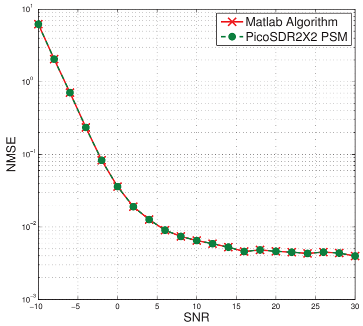

Fig. 5: NMSE of the hybrid estimator MATLAB algorithm compared to that of the same estimator PSM on PicoSDR2X2 in function of the SNR with $F_D T_s = 7 \times 10^{-3}$, QPSK.

Similarly, both BER curves in Figure 6 coincide in each point which confirms that the PSM QPSK demodulation function is valid. This is expected because the SNR estimation exploits the QPSK demodulation.

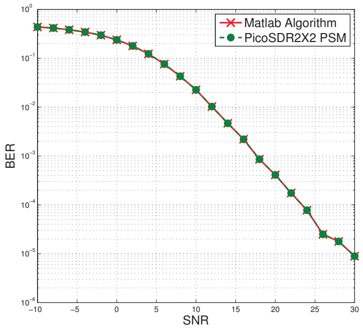

Fig. 6: BER of the hybrid estimator MATLAB algorithm compared to that of the same estimator PSM on PicoSDR2X2 in function of the SNR with $F_D T_s = 7 \times 10^{-3}$, QPSK.

For the channel identification, as presented in Figures 7 and 8, the estimated channel follows the emulated channel variations closely. The channel identification on PicoSDR2X2 is, thus, successful and validated.

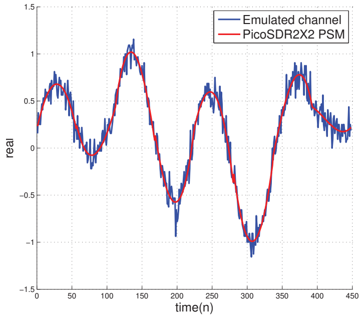

Fig. 7: EB Propsim channel real part identification using the hybrid estimator MATLAB algorithm compared to that of the same estimator PSM on PicoSDR2X2 with an SNR of $24dB$ and $F_D T_s = 7 \times 10^{-3}$, QPSK.

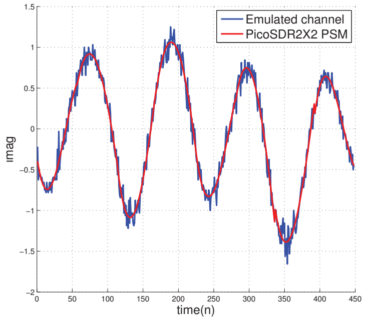

Fig. 8: EB Propsim channel imaginary part identification using the hybrid estimator MATLAB algorithm compared to that of the same estimator PSM on PicoSDR2X2 with an SNR of $24dB$ and $F_D T_s = 7 \times 10^{-3}$, QPSK.

VI. Conclusion

We presented in this paper the steps of the implementation on Nutaq's PicoSDR2X2 of a hybrid SNR estimator using a rapid prototyping method. This time-saving programming approach is based on MATLAB/Simulink and Xilinx tools and Nutaq's MBDK. We were able to test and validate the behavior of the .bit file on the SDR platform. A further extension of this project, on which we are already working, is the integration of this estimator in an OFDM transmission chain in order to get closer to the LTE standard.

References

[1] Matlab. Rapid prototyping. [Online]. Available: http://www.mathworks.com/rapid-prototyping

[2] Louet, Yves, and Al Ghouwayel, Ali. "FPGA implementation of a reconfigurable FFT for software radio multistandard systems." *International Conference on Consumer Electronics*. 2009.

[3] Ahmadian, Mansour, et al. "Model-based design and SDR", DSP enabled Radio, 2005, *the 2ndIEE/EURASIP Conference on* (Ref No. 2005/11086), pp 19/1-19/6.

[4] Zlydareva, Olga, and Sacchi, Claudio. "Multi-standard wimax/umts system framework based on sdr." *Aerospace Conference, 2008 IEEE*. IEEE, 2008.

[5] Muzammil, Rehan, et al. "Design and implementation of BPSK transmitter and receiver for Software Defined Radio on a Model Based Development Platform." *Industrial Electronics and Applications (ISIEA), 2011 IEEE Symposium on*. IEEE, 2011.

[6] Alluri, Veerendra Bhargav, et al. "A new multichannel, coherent amplitude modulated, time-division multiplexed, software-defined radio receiver architecture, and field-programmable-gate-array technology implementation." *Signal Processing, IEEE Transactions on* 58.10 (2010): 5369-5384.

[7] Alluri, Veerendra Bhargav, et al. "A multi-channel coherent amplitude modulated time division multiplexed software defined radio receiver architecture for programmable-logic-device technology implementation." *Radio and Wireless Symposium, 2009. RWS'09*. IEEE. IEEE, 2009.

[8] Bellili, Faouzi, et al. "Maximum likelihood SNR estimation of linearly-modulated signals over time-varying flat-fading SIMO channels." *Signal Processing, IEEE Transactions on* 63.2 (2015): 441-456.

[9] Wiesel, Ami, et al. "SNR estimation in time-varying fading channels." *IEEE Transactions on Communications* 54.5 (2006): 841-848.

[10] Nutaq. Model-Based Design Kit (MBDK). [Online]. Available: http://nutaq.com/en/software/model-based-design-kit

[11] MathWorks. (2015) Model-based design. [Online]. Available: http://www.mathworks.com/help/simulink/gs/model-based-design.html

[12] PicoSDR-User's Guide-v1.0, Nutaq, June 2013.

[13] Propsim quick guide, Anite Telecoms, 2013. [Online]. Available: http://www.gigacomp.de/pdfs/Anite_Propsim_Quick_guide.p