On Real Time Optical Wireless Communication Channel Emulator Design with FPGAs

Luís Rodrigues[1,2], Luis Duarte[1,2], João Pandeirada[2] and Luís Nero Alves[1,2]
[1]Instituto de Telecomunicações, Aveiro, Portugal
[2]Department of Electronics, Telecommunications and Informatics, University of Aveiro
Aveiro, P-3810-193 Portugal
Email: luis.abade@ua.pt, luisdiogoduarte@ua.pt, joao.pandeirada@ua.pt, nero@ua.pt

Abstract— **This paper discusses the implementation details of a channel emulator suited for optical wireless communications (OWC) systems. The channel emulator comprises functional blocks able to model the transmitter, the receiver and the channel. It is in this sense a valuable tool for system performance evaluation. The proposed design uses field programmable gate arrays (FPGAs) as supporting hardware platform, enabling flexible and expedite interconnection with other system blocks. The achieved design operates in real time, allowing timely and realistic assessment of the channel and terminal equipment impairments on the overall system. Achieved results demonstrate the feasibility to address channel emulation with 500ps time resolution, with negligible quantization errors when compared to the predefined channel coefficients.**

Keywords-component; Channel Emulators, Optical Wireless Communications, FPGA

I. INTRODUCTION

Channel emulators constitute the first approach for an initial study and validation of a communication system. Therefore, it is a very important tool that has been in the scope of researchers, especially in radio technology [1-4]. Such emulators offer a controlled and repeatable environment where developers can test their systems in realistic channel propagation conditions. This approach improves the system design stage as it allows to take channel impairments into consideration. They offer a realistic emulation of the propagation environment and a fast testing solution that verifies the performance of a wireless communication system with more flexibility than a physical wireless testbed. Channel emulation comprises a channel emulator block, as well as both the emulated transmitter (Tx Front-end) and the receiver front-ends (Rx Front-end). As can be seen in Fig. 1, such tool is normally connected to the baseband emitter and receiver chains where optimal coding and modulation techniques can be tested. Such emulator reproduces a highly configurable testbed with a very low setup cost when compared to Over The Air testing approaches.

As stated in [5] there are many wireless emulators commercially available [4, 6], able to provide both channel flexibility and bandwidth. Examples of the commercial solutions are the National Instruments PXIe-5644R [7] and the Keysight Propsim Channel Emulation [8], which are used closely in military scenarios where critical radio communication channel needs to be constantly and quickly evaluated [9]. Published work on emulators design usually describes the wireless channel as a multipath fading channel using measured impulse responses in different scenarios and standard radio models. Radio channel emulation is commonly used to speed up radio digital

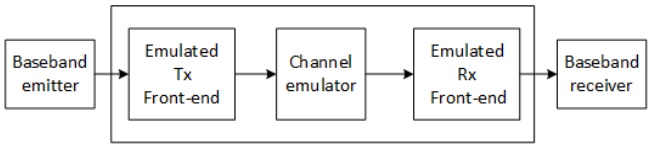

Figure 1. Channel Emulation main blocks.

implementation tuning, as well as, field condition assessment on critical missions [8, 9].

However, to the best of authors knowledge there is no Visible Light Communication Channel Emulator available. With this in mind, this paper proposes a suitable architecture to build a realistic channel emulator for optical wireless communications systems. This architecture comprises the emitter, the channel and the receiving front-end blocks. These blocks have to be tailored to the specific system under development, given that, system design may change considerably. For instance, the emitter and front-end receiver blocks for a visible light communication (VLC) system employing wavelength division multiplexing (WDM) are necessarily different from those employed in an infrared OWC system. To have these specifications taken into consideration, the architecture must be able to support reconfiguration, a feature which justifies the choice for FPGAs. This paper focus on the problem of time resolution need to realistically emulate the OWC channel. Given that FPGAs operate at modest frequencies when compared to the typical delays of an OWC channel, emulating the channel with enough resolution demands parallelization techniques. This paper proposes the usage of polyphase FIR filter decomposition as a mean to achieve the required time resolutions [10].

The rest of the paper is organized as follows, section II presents a summary of channel modeling strategies available in the literature. Section III outlines the general architecture of the channel emulator. Section IV discusses design consideration of a channel emulator with FPGAs. Section V illustrates the potentialities of the emulator, through some illustrative examples. Finally, section VI, concludes the paper.

II. OWC CHANNEL MODELLING

In the design and implementation of an OWC system, it is very important to understand the characteristics of the channel. The most generic way of describing the Channel Impulse Response (CIR), *h(t)*, of an OWC channel is given by

$$h_{\lambda i}(t) = h_{\lambda i_{LOS}}(t) + \sum_{k} h^{(k)}_{\lambda i_{NLOS}}(t) \qquad (1)$$

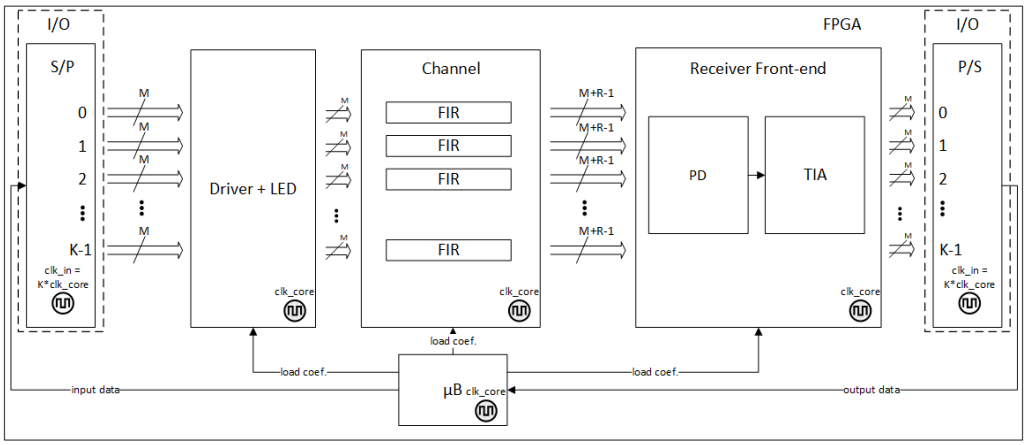

Figure 2. OWC.Channel Emulator Architecture

This description shows that the impulse response is composed by a line-of-sight contribution ($h_{\lambda i_{LOS}}$) and a sum of all the *k* non-line of sight paths ($h^{(k)}_{\lambda i_{NLOS}}$) that reach the receiver [11, 12]. This generic description can also be approximated using a discrete FIR filter topology. In such case, the filter is described by its coefficients retrieved directly from the discrete version of *h(t)*. The spectrum discretization in (1), expressed by λ parameter, is due to the different wavelength response of the channel components, such as the emitter, the reflecting elements and the receiver. A simple classification can divide the wavelength spectrum in ultra violet (UV), visible and infra-red (IR) regions. While UV and IR systems operate usually with a single wavelength, visible light systems require a multiple wavelength approach. This is in part due the nature of visible light which encompasses several wavelengths and the possible presence of wavelength selective environments. Systems using wavelength division multiplexing (WDM) are another example of multiple wavelength systems. For these multiple wavelength systems, the CIR is generally wavelength dependent

The components of (1) can be obtained using raytracing techniques [13]. In order to achieve this, it should be defined the transmitter, reflector and receiver models and then calculate all the possible light ray paths between the emitter and the receiver, as well as the power losses due to reflection and propagation. Raytracing techniques are a reliable approach for producing an accurate CIR. However, the complexity of such an approach increases with the reflection order. Methods able to improve this time-consuming task usually rely on Monte Carlo analysis [14]. Based on these approaches, there are commercial software packages available, such as Zemax and CandLES [15]. Effective methods of simulating the channel often recur to analytical models, such as the ceiling bounce model [16] or to empirical methods based on real CIR measurements [17].

III. OWC Channel Emulator Architecture

Fig. 2 depicts the general architecture of the proposed OWC emulator, comprising, three main blocks: the Driver+LED, the Channel and the Receiving Front-end. Additionally, the complete system includes serializer and de-serializer blocks at both ends of the system chain, and a softcore processor. The Driver+LED block is responsible for the modeling of the electro-optical conversion, entailing device non-linearity and bandwidth restrictions. In terms of hardware resources, it includes a look-up table to model the non-linear effects and a FIR filter to model the frequency response. The channel block implements the channel response based on the selected CIR of the channel. The CIR is implemented using a FIR filter. The receiving front-end implements the model of the optical-electrical conversion on the receiver side. This entails, the frequency response of the amplifier, the addition of noise sources and the characteristics of the photo detector. System reconfiguration and monitoring are handled by an embedded softcore processor. This processor enables signal monitoring, analysis of the channel effects on the transmitted signals and system reconfiguration. Reconfiguration is required to support multiple system parametrizations. For instance, to emulate the channel in different positions inside a room. The serializer and de-serializer blocks are required to handle the time resolution of the CIR. Common implementations of a FIR filter in the targeted FPGA results in *4ns* of CIR resolution since the hardware is not able to compute faster output samples (system clock is limited to *250MHz*). Lower time resolution demand parallel processing architectures. These can be achieved employing polyphase decomposition of the CIR [10]. The number of parallel paths, (*K* in Fig. 2), acts as a frequency multiplying factor, enabling to reduce the channel time resolution to the required values. Considering a Ray-Tracing channel model, the CIR time resolution is related to the target room dimensions by (2).

$$\frac{\sqrt{2}\Delta x}{\Delta t} = \sqrt{2}K\Delta x f_{clk} = c \qquad (2)$$

where Δx is the spatial resolution and Δt is the time resolution, which are bounded by the speed of light, *c*. The time resolution is equal to the inverse of $K \cdot f_{clk}$. To have a precise simulation model, Δx should be small relative to the room size. Since f_{clk}=*250MHz*, choosing *K=8* results in a $\Delta x = 10.6cm$.

The channel output is given by the convolution between CIR and input signal, as shown in (3).

$$y(n) = h(n) * x(nK) + \eta(n) \qquad (3)$$

where, *y(n)* is the output, *h(n)* is the CIR, and the input signal is *x(nK)*. The noise term, $\eta(n)$, considers shot noise induced by

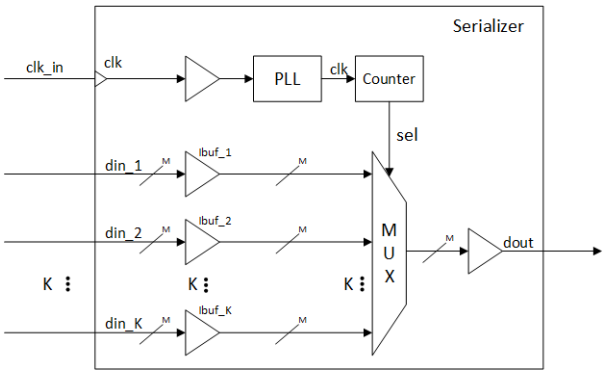

Figure 3. Serializer architecture

ambient light and should be implemented in the receiver block. Since the rate of *n* is *K* times faster than the FPGA clock, *x(nK)* is simply *x(n)* interpolated by *K*, as will be further addressed in section IV.B.

IV. Channel Emulator Design with FPGA

A. FPGA Resources

For the validation of the proposed architecture, a Xilinx KC705 development board was chosen. In this board there are many IO ports available, reconfigurable logic structures, as well as, DSP blocks. The DSPs are very power efficient and are able to operate at higher frequencies than the remaining soft logic circuits inside the chip. They are designed to implement common DSP algorithms, such as addition, subtraction, multiplication and coefficient register storage, with minimum resource utilization and maximum power efficiency in mind.

The SERDES stands for Serializer/Deserializer. These are powerful tools in state of the art FPGAs. FPGAs are the most used technology for wireless systems prototyping. Their clocks are in the range of hundreds of MHz. Higher data rates are possible through parallelization strategies. These strategies rely on these special blocks. It is a widely used technique that allows the serialization and deserialization of data from and to the FPGA fabric pins. When the SERDES is in serialize mode, the output clock rate is *K* times faster than the input clock, where *K* is the number of parallel inputs, each with *M* bits length. With this in mind, SERDES adapts these parallel implementations to the physical layer requirements, which usually relies on serial communication schemes. The SERDES can also operate in the de-serialize mode, converting a serial stream of *K*, *M* bits symbols into *K* parallel paths of length *M*. In this mode, the input clock is *K* times faster than the output. Fig. 3 depicts the overall architecture of the SERDES block, configured in the serializer mode. The SERDES block supports user configuration, enabling to establish the dimension of the symbols and the required parallelization. This is using a dedicated clock. This clock is based on a phase locked loop (PLL) frequency synthesizer, having as its reference the FPGA internal clock. The PLL frequency synthesizer generates a clock with frequency *K* times the larger than the FPGA reference clock.

As previously mentioned an embedded soft-core processor is used to manage system configuration and monitoring. In

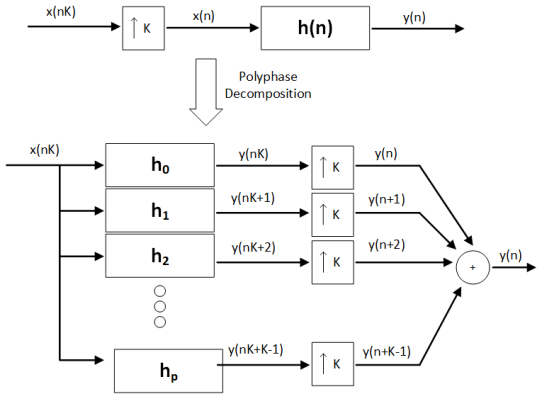

Figure 4. Polyphase decomposition process

Xilinx environment, this soft-core is called Microblaze (µB). It is part of the Vivado IP core library. It is possible to use this soft-core processor for monitoring purposes. This is accomplished through a user graphical interface (GUI), and a connection with MATLAB. This GUI supports visual aids to system configuration as well as probing facilities, which can enable analysis of the signals and a full characterization of the channel effects on the transmitted signal.

B. Polyphase Filter Decomposition

Fig. 4 illustrates the polyphase decomposition of a FIR filter [10]. Polyphase decomposition divides the original filter into smaller parallel filters, h_p, known as phases, which are subsets of *h(n)*, according to (4).

$$h_p = h(nK + p) \quad , p \in \{0,1, \dots, K-1\} \tag{4}$$

where p is the phase index. Equation (5) shows how the decomposition is accomplished, thus stating the equality between the implementations depicted in Fig. 4.

$$\begin{aligned} y(n) &= \sum_{p=0}^{K-1} y(n+p) \\ &= \sum_{p=0}^{K-1} h(nK+p) * x(nK) \\ &= h(n) * x(nK) \end{aligned} \tag{5}$$

The second passage in (5) is possible due to an interpolation process. which keeps the original samples, adding *K-1* zeros between them. The interpolations and the sum at the end can be replaced by a switch, selecting one FIR output each time, which operates *K* times faster than the filters. In hardware, this switch is implemented with a SERDES block.

C. Quantization and normalization

The CIR coefficients exhibit a large dynamic range. Mapping these coefficients in hardware is not straightforward. First, because the amplitude discretization implies quantization errors. Furthermore, if a fixed number of bits is used for each coefficient, then the bit length must be carefully selected in order to be able to represent all the coefficient's amplitude span. To reduce the quantization error, a normalization to the highest value in the CIR can be done. This will ensure the maximum bit

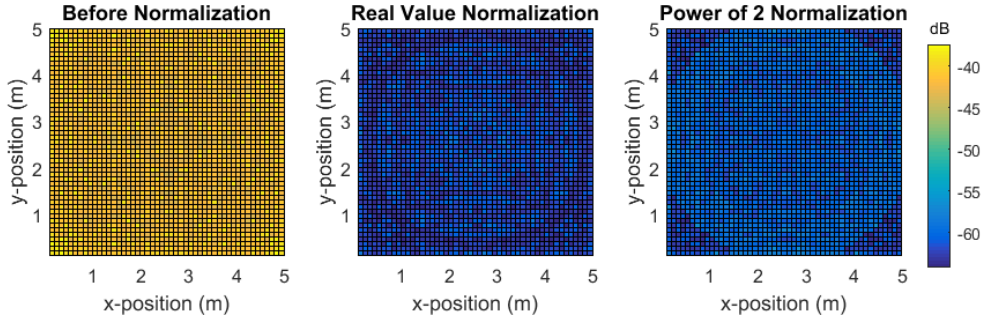

Figure 5. Quantization effects: EVM before and after coefficient normalization

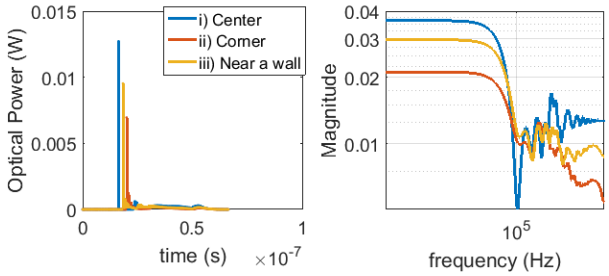

Figure 6. CIR (right) and channel frequency response (left), for three different positions in the room: i)Center; ii)Corner; iii) Near the wall

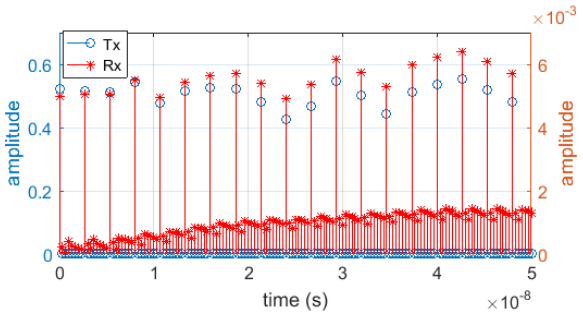

Figure 7. Tx vs Rx OFDM signal

usage by the DSP blocks. Fig. 5 shows the error vector magnitude (EVM) of the implemented CIR, with and without normalization, for all the floor positions inside a room of 5x5x5 meters. The error due to quantization effects can be reduced by *25dB* using normalization by the maximum coefficient value. An alternative is using normalization by a power of 2 near the maximum real value of the CIR. This normalization slightly increases the EVM when compared to previous approach, but it proved to be sufficient to make the quantization error acceptable (in the same order of magnitude). This approach is less hardware demanding, since multiplying by powers of 2 can be implemented with simple shift operations.

V. RESULTS

A. Simulation Set-up

The proposed OWC channel emulator architecture was implemented in System Generator, which is a Xilinx simulation tool using hardware simulation blocks embedded in MATLAB Simulink. For demonstration purposes, the system comprised an emitter, centered in the ceiling of a room of 5x5x5 meters, with white walls and ceiling. The CIR inside the room assuming ideal transmitter and receiver blocks, was obtained using a ray tracing algorithm accounting with two reflections. For testing purposes, the transmitted signal, generated in MATLAB, consisted of an OFDM signal, with 300 load carriers from the total 1024 carriers, QAM constellations, considering 4, 16 and 64-QAM scenarios, at a 250MHz sampling frequency. Fig. 6 depicts the CIR at three different positions inside the room. The same figure also shows the channel frequency response for the same positions.

TABLE I. CHANNEL EMULATOR BLOCK RESOURCES UTILIZATION

Resource	Utilization	Available	Utilization
LUT	600	203800	0.29%
LUTRAM	344	64000	0.54%
FF	1561	407600	0.38%
DSP	40	840	4.76%

B. FPGA Resources

Table I summarizes the usage of FPGA resources. As it can be seen, the implementation has low logic resources occupation. Due to the nature of the system architecture, which requires the implementation of FIR filters, one of the most important parameters is the used DSP resources. From Table I, 40 DSP blocks were used, which corresponds to 4.76% of the total FPGA DSPs, showing the feasibility to support added functionalities in the architecture, such as reconfiguration, monitoring or finer details of the transmitter and receiver blocks. Furthermore, there are enough available logical blocks that allows the emulation of higher complexity systems such as WDM channels.

C. Channel measurements

Fig. 7 shows a sample of the transmitted signal (Tx) and the received signal, properly synchronized and normalized (Rx). The channel introduces intersymbol interference (ISI), due to multipath propagation. This effect is visible in the received signal, where reflected signal from previous symbols is added to subsequent symbols.

A well-known metric used to measure the difference between two signals is the RMS EVM and it is computed in dB as shown in (6).

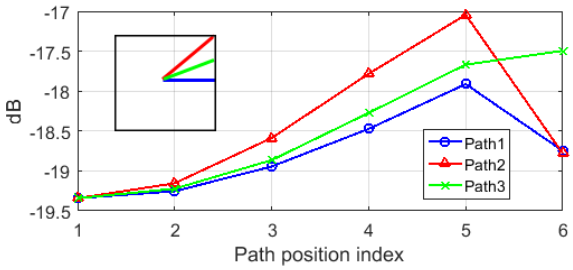

Figure 8. EVM results

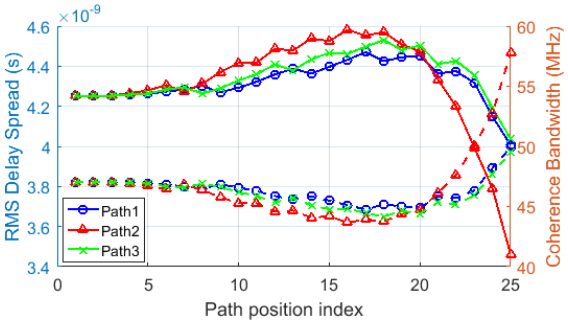

Figure 9. RMS Delay Spread and Coherence Bandwidth

$$EVM_{RMS} = 10\log\left(\sum_{n=0}\sqrt{\frac{(tx(n)-rx(n))^2}{tx(n)^2}}\right) \quad (6)$$

where *tx* is the transmitted signal and *rx* is the received signal. Fig. 8 represents the RMS EVM, in dB, variation of the received signal across three different paths, starting in the center. For each path, a total of 6 locations were considered, denoted in the figure by "Path position index". As expected, the EVM increases as the receiver moves away from the transmitter and get closer to the walls. This behavior is the result of lower LOS amplitude and higher reflection effects. However, near the center of the wall (end of path 1) and in the corner (end of path 2) the RMS EVM value shows a significant improvement, which can be explained by the decrease of reflections (considering higher order reflections this observation may change). When the receiver is located in a quarter of a wall (end of path 3), the reflections from the nearest wall have relative higher amplitude and the EVM does not improve as in the other scenarios. Analyzing the EVM in real time for different positions inside the room is a feature made possible through the usage of a channel emulator.

The RMS Delay Spread, D_{RMS}, is a well-known metric in multipath channels. It is used to measure time dispersion in these types of channels and can be computed as [11, 12]:

$$D_{RMS} = \sqrt{\frac{\sum_t (t-\mu)^2 h^2(t)}{\sum_t h^2(t)}} \quad (7)$$

$$\mu = \sqrt{\frac{\sum_t t h^2(t)}{\sum_t h^2(t)}} \quad (8)$$

The coherence bandwidth measures the interval in which the channel frequency response is considered flat and it is related to the inverse of the delay spread. Fig. 9 shows the RMS delay spread, represented with solid lines, and coherence bandwidth, represented with dashed lines, of the previous considered paths. While moving away from the center, D_{RMS} increases. However, near the walls, the delay spread decreases due to the confinement of the reflections in a shorter time interval, relatively to the LOS path. The best scenario is achieved near the center of a wall. On the other hand, the coherence bandwidth has the inverse behavior: decreases in the beginning of the path and increases near the walls.

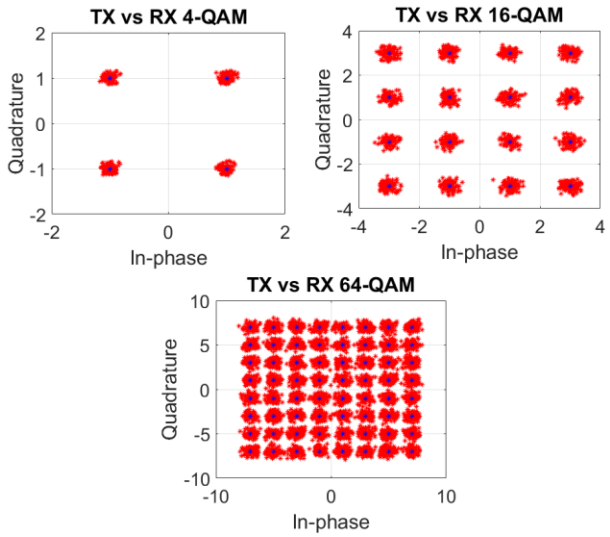

Figure 10. Received QAM constelations in the center of the room

The multipath effects have a significant impact on the received data. Fig. 10 shows the received constellations for 4, 16 and 64-QAM modulation using OFDM. The lower order modulations, 4 and 16-QAM, can be easily recovered. However, 64-QAM already presents dispersion that may induce errors in the demodulation. Note that this result only considers the multipath effect due to ISI. In a real scenario other noise sources, due to ambient light and shot noise and receiver amplifier, may further degrade the received signal.

VI. CONCLUSIONS

This paper proposes a modular architecture suitable for OWC channel emulator design with FPGAs. The work herein reported concentrated on the validation of the proposed approach, as well as, the disclosure of a suitable design approach to implement the fine grain timing considerations required to emulate realistically the channel impulse response. The proposed solution is supported in polyphase FIR filter decomposition methods, and lends itself amenable to implementation with FPGAs. Achieved results illustrate the potential of a real-time channel emulator supported on a GUI interface with MATLAB. Such approach enables fast and reliable system performance analysis to be taken from the very begging of the design process.

ACKNOWLEDGEMENTS

This work is funded by FEDER PT2020 partnership agreement under the project UID/EEA/50008/2013.

REFERENCES

[1] I. Val, F. Casado, P. M. Rodriguez and A. Arriola, "FPGA-based wideband channel emulator for evaluation of Wireless Sensor Networks in industrial environments," *2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pp. 1-7, Barcelona, Spain, September 2014.

[2] G. Joshi, P. R. Prasad and A. Singh, "FPGA implementation of channel emulator for testing of wireless air interface using VHDL," *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pp. 1-6, Bangalore, India, May 2016.

[3] S. Yang and Z. Can, "Design and implementation of multiple antenna channel emulator for LTE system," *9th International Conference on Communications and Networking in China*, pp. 1-6, Maoming, China, August 2014.

[4] S. Picol, G. Zaharia, D. Houzet and G. El Zein, "Hardware Simulator for MIMO Radio Channels: Design and Features of the Digital Block," *2008 IEEE 68th Vehicular Technology Conference*, pp. 1-5, Calgary, Canada, September 2008.

[5] K. C. Borries, G. Judd, D. D. Stancil and P. Steenkiste, "FPGA-Based Channel Simulator for a Wireless Network Emulator," *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*, pp. 1-5, Barcelona, Spain, April 2009.

[6] M. Kahrs and C. Zimmer, "Digital signal processing in a real-time propagation simulator," in *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 1, pp. 197-205, February 2006.

[7] National Instruments, "Real-Time MIMO Channel Emulation on the NI PXIe-5644R.", Internet: http://www.ni.com/example/31556/en/, May 16, 2013 [20-Feb-2018].

[8] Keysight, "Propsim Channel Emulation MANET & Tactical Radio Testing.", Internet: https://www.keysight.com/en/pc-2697407/propsim-channel-emulation-manet-tactical-radio-testing, [20-Feb-2018].

[9] National Instruments, "NI Provides SDR Technology for DARPA Spectrum Collaboration Challenge", November 1, 2016. Internet: http://www.ni.com/newsroom/release/ni-provides-sdr-technology-for-darpa-spectrum-collaboration-challenge/en/ [20-Feb-2018].

[10] O. Gustafsson, K. Johansson, H. Johansson and L. Wanhammar, "Implementation of Polyphase Decomposed FIR Filters for Interpolation and Decimation Using Multiple Constant Multiplication Techniques," *APCCAS 2006 - 2006 IEEE Asia Pacific Conference on Circuits and Systems*, pp. 1-4, Singapore, December 2006.

[11] Y. Qiu, H. Chen and W. Meng, "Channel modeling for visible light communications-a survey", *Wireless Communications and Mobile Computing*, vol. 16, no. 14, pp. 2016-2034, February 2016.

[12] F. Miramirkhani and M. Uysal, "Channel Modeling and Characterization for Visible Light Communications," in *IEEE Photonics Journal*, vol. 7, no. 6, pp. 1-16, December 2015.

[13] J. R. Barry, J. M. Kahn, W. J. Krause, E. A. Lee and D. G. Messerschmitt, "Simulation of multipath impulse response for indoor wireless optical channels," in *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 3, pp. 367-379, April 1993.

[14] F. J. Lopez-Hernandez, R.Perez-Jimeniz and A.Santamaria, "Monte Carlo calculation of impulse response on diffuse IR wireless indoor channels", *Electronics Letters*, vol. 34, pp. 1260–1262, June 1998

[15] M. B. Rahaim, T. Borogovac, and J. B. Carruthers, "CandlES: Communication and lighting emulation software," *Fifth ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, pp. 1-6, September 2010.

[16] J.B. Carruthers and J.M. Kahn, ¨Modelling of non-directed wireless infrared channels¨, *IEEE Transaction on Communication*, vol.45, pp. 1260-1268, October 1997

[17] Cipriano Lomba, "Comunicações Ópticas em Espaço Livre para Ambientes Interiores: Modelação, Simulação e Optimização do Canal Óptico," M.S. thesis, DETI., UA., Aveiro., Portugal, 1997.