Digital Emulation of Time-Varying PMD for Real-Time DSP Evaluations

Digital Emulation of Time-Varying PMD for Real-Time DSP Evaluations

Haorui Kan[1], Hong Zhou[1], Erik Börjeson[1], Magnus Karlsson[2], and Per Larsson-Edefors[1]

[1]*Dept. of Computer Science and Engineering, Chalmers University of Technology, Gothenburg, Sweden*
[2]*Photonics Laboratory, Chalmers University of Technology, Gothenburg, Sweden*
erikbor@chalmers.se

Abstract: We introduce a digital PMD emulator to accelerate BER analysis of coherent receiver DSPs. The emulator creates parameterizable, time-varying impairments, which we use to demonstrate real-time analysis of a CMA equalizer.

1. Introduction

The digital signal processing (DSP) hardware used in fiber-optic communication systems is designed with very strict performance requirements, many of which cannot be evaluated unless time-varying transmission properties are considered. But real-time experiments involving optical and electronic components can become very complex. The CHOICE environment [1] introduces digital emulation as an alternate route to real-time analysis of coherent fiber-optic communication systems. CHOICE contains a set of hardware description (VHDL) components that can be interconnected to configure a field-programmable gate array (FPGA) system to run real-time DSP experiments in which pseudo-random data are generated online. Here, transmitter properties and some channel impairments, such as phase noise, can be digitally emulated and superpositioned on the modulated data. The resulting digitally-encoded symbols are in real-time transferred to the DSP receiver unit which is located in the same FPGA.

In an optical emulator of polarization-mode dispersion (PMD), a number of fiber sections, each with a birefringent crystal whose rotation is under computer control, are concatenated [2]. In contrast, we here introduce a digital emulator of PMD. This emulator digitally models the physical PMD effects based on the waveplate model [3] and can be inserted inside a digital end-to-end system model such as CHOICE. With a digital PMD emulator, we can precisely control polarization rotations and group delays along the fiber's two polarizations and change these as the FPGA emulation run progresses. This makes for parameterizable, programmable real-time DSP evaluations.

2. Digital PMD Emulation

In the waveplate model [3], the fiber is considered as a concatenation of waveplates with random birefringences and rotation angles. The transmission matrix $N(\omega)$ can then be represented as the product of K birefringent waveplates as [4]:

$$N(\omega) = \prod_{k=1}^{K} N_k(\omega) = \prod_{k=1}^{K} \begin{bmatrix} \cos\theta_k & \sin\theta_k \\ -\sin\theta_k & \cos\theta_k \end{bmatrix} \begin{bmatrix} e^{j\omega\tau_k/2+j\delta_k} & 0 \\ 0 & e^{-j\omega\tau_k/2-j\delta_k} \end{bmatrix} \tag{1}$$

Here, θ_k and δ_k is the rotation angle and phase shift of section k, and τ_k is the differential group delay (DGD) between the two channels. For simplicity we will use $\delta_k = 0$ in the following. However, this model requires signals to be transformed back and forth between time and frequency domain, which complicates a real-time implementation. Thus, we move the waveplate model to the time domain. We use Lagrange fractional delay filters, which are finite impulse response (FIR) filters with Lagrange interpolation coefficients [5], to apply a delay equal to fractions of the symbol period, $\tau_k/2$ and $-\tau_k/2$, to each of the channels.

A block diagram of our digital PMD emulator is shown in Fig. 1. We concatenate an arbitrary number of fiber sections; each section with a rotation component and two FIR filters, which are parameterizable with a rotation angle read-only memory (ROM) and a Lagrange interpolation component which takes as input a fractional delay parameter. The final rotation component is needed to complete the waveplate model.

Fig. 1 also shows a QPSK modulated input signal and the outputs of MATLAB and VHDL simulations of the two channels in a one-section PMD emulator that uses a rotation angle θ of 10° and a DGD of $0.6T_{sym}$ where T_{sym} is the symbol period. The results show that the digital PMD VHDL-based emulator agrees well with the MATLAB reference model. (Note that the PMD emulator supports higher modulation formats than QPSK.)

3. System Verification of Digital PMD Emulator

To ensure its functionality, we use VHDL simulations to verify the digital PMD emulator inside the digital system model shown in Fig. 2. Based on CHOICE [1], we develop a two-channel system, in which two random-number generator (RNG) components generate pseudo-random data which go through a QPSK modulator component.

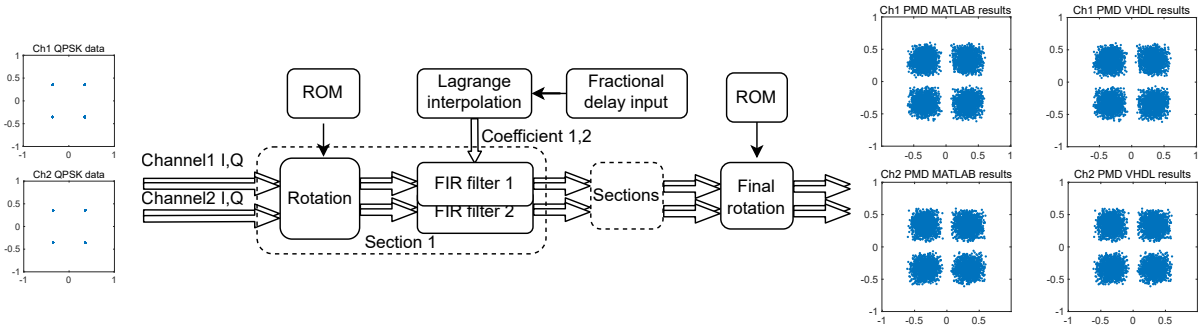

Fig. 1: PMD emulator block diagram with insets of MATLAB and VHDL simulation results.

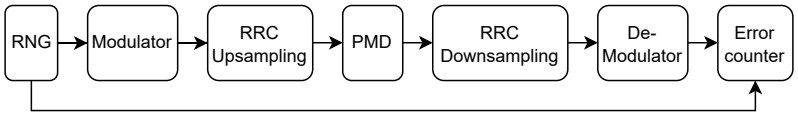

Fig. 2: Block diagram of system used in VHDL simulation to verify the digital PMD emulator.

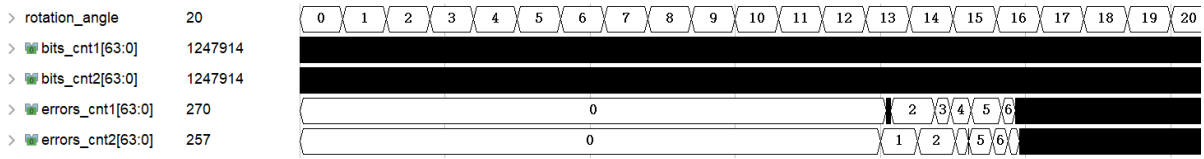

Fig. 3: Example of Vivado simulation result. Here, *bits_cnt1* and *bits_cnt2* represent the accumulated number of data transmitted over each of the channels, while *errors_cnt1* and *errors_cnt2* represent the accumulated number of erroneous data.

Root-raised cosine (RRC) pulse shaping is done using a 51-tap FIR filter structure for a roll-off factor of 0.1. Since the RRC filter uses an oversampling of two, the filter is parallelized in two data lanes. The use of a symmetric transposed filter reduces resource consumption and critical path delay. Next, the two polarizations pass through the emulated fiber, in which the digital effect of PMD is added. At the receiving end, RRC downsampling and demodulation are carried out, after which an error counter logs the number of times the received digital symbols differ from the original data generated by the RNG components.

An example of a VHDL system simulation in Xilinx Vivado is shown in Fig. 3. We use one fiber section with a DGD of $0.6T_{sym}$, and gradually increase θ from 0° to 45°. When θ increases to 13°, an error occurs. In addition, a larger θ produces more errors, which means that PMD has a greater impact on the signals, which is consistent with our expectations.

4. Real-Time System with PMD Emulator and Equalizer

For the real-time emulations, we use a Xilinx Virtex-7 VC709 board which we run at a clock rate of 30 MHz. To emulate a system similar to a practical fiber channel, we now introduce additive white Gaussian noise (AWGN) and use a basic constant-modulus algorithm (CMA) equalizer to downsample the signals and compensate their PMD impairments. The structure of this system is shown in Fig. 4. Here, the PMD emulator component has 10 sections, each with the same DGD value but with different θ. We set the first, center and last section to variable rotations, with different θ updating frequencies. All other sections have fixed θ. We use an 11-tap CMA equalizer with a step size equal to 0.0002 to make the equalizer stable in these feature demonstrations. The total number of bits transmitted over both channels and the total number of erroneous bits are captured by an integrated logic analyzer (ILA). This information is then used to calculate the bit error rate (BER) using MATLAB.

First we use a per-section DGD of $0.06T_{sym}$. The θ of the first, center and last section changes randomly in the range from 0° to 15°. The updating frequency of the first, center and last sections is 0.3, 1.0, and 0.1 Hz, respectively. Since each polarization is emulated at a system clock rate of 30 MHz, these rotations would correspond to, on average, 30, 101, and 10 rad/s, respectively, in a 30-GBaud system. Fig. 5(A) shows how the BER varies with the input signal to noise ratio. The results match the theoretical BER curve in a Gaussian environment, which we expect when rotations are slow and have limited effect on the system performance. To explore how a more pronounced PMD effect influences long-time BERs, we increase the updating frequency of the center section by 100X for the 8, 10, and 12 dB cases; this fast scenario is marked with dashed lines in Fig. 5(A).

Fig. 5(B) shows how the updating frequency of θ influences the BER. We compare two scenarios with different center section updating frequencies. One is for emulating a benign scenario by setting the center section updating

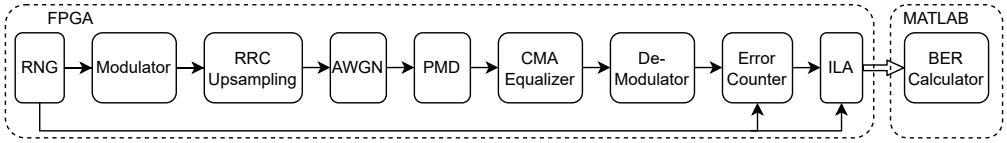

Fig. 4: Block diagram of system used in real-time FPGA emulation.

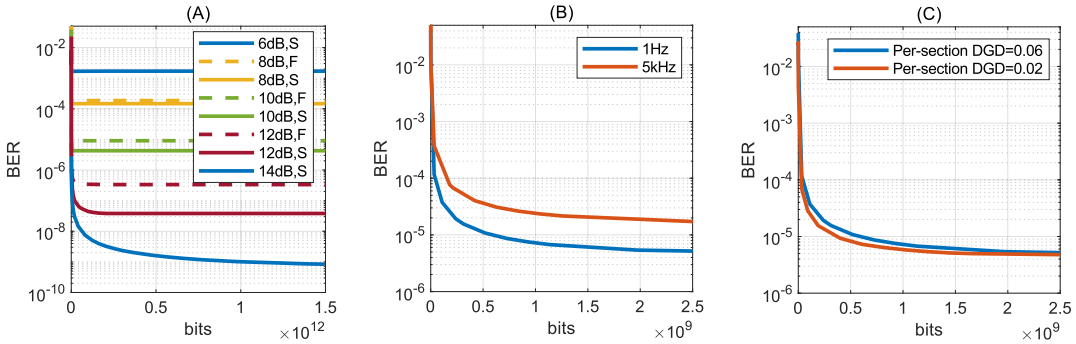

Fig. 5: BER trends for (A) different E_b/N_0 and center section θ updating frequencies (S means slow, F means fast) in long-time runs, (B) different center section θ updating frequencies at $E_b/N_0 = 10$ dB and a per-section DGD of $0.03T_{sym}$, and (C) different DGDs at $E_b/N_0 = 10$ dB and a θ updating frequency of 1 Hz.

frequency to 1 Hz (101 rad/s in a 30-GBaud system). The other one is for emulating a harsh scenario by setting the frequency to 5 kHz (505 krad/s in a 30-GBaud system). We can see that the higher updating frequency will cause a penalty in the BER, since the CMA equalizer has less time to fully converge in that scenario.

Fig. 5(C) shows how the per-section DGD influences the BER trend. We compare two scenarios with a per-section DGD equal to $0.06T_{sym}$ and $0.02T_{sym}$, respectively. The per-section DGD doesn't have much influence on the BER trend since the CMA equalizer can compensate for it, but it does impact the speed of convergence: Smaller delays will make BER decrease faster, but they will converge to the same value.

Finally, Table 1 clearly shows that we can readily fit the whole emulator system on the used FPGA.

VHDL component	LUT utilization	DSP utilization	Slice registers	BRAM
RNG+Modulator	306 (0.07%)	-	191 (0.02%)	-
RRC Upsampling	6508 (1.50%)	96 (3.00%)	6899 (0.80%)	-
AWGN	1883 (0.43%)	24 (0.67%)	3279 (3.78%)	-
PMD	11987 (2.76%)	588 (16.30%)	6407 (0.74%)	12.5 (0.75%)
CMA Equalizer	2116 (0.49%)	380 (10.56%)	7455 (0.86%)	-
Demodulator	24 (0.01%)	-	10 (0.01%)	-
Error Counter	6 (0.01%)	-	254 (0.02%)	-
ILA	1386 (0.40%)	-	2696 (0.30%)	9 (0.61%)
Others	8 (0.01%)	-	33 (0.01%)	-
Total	**25117 (5.60%)**	**1088 (30.22%)**	**28592 (3.30%)**	**21.5 (1.46%)**

Table 1: Resource utilization of Xilinx Virtex-7 FPGA.

5. Conclusion

We introduced a digital PMD emulator, whose polarization rotations and group delays along two polarizations can be precisely controlled in real-time FPGA emulations. Starting from the CHOICE environment [1], we designed a two-polarization real-time system, including our PMD emulator and a CMA equalizer, and tested how PMD parameters influence system BER in long-time runs.

References

[1] E. Börjeson, C. Fougstedt, and P. Larsson-Edefors, "Towards FPGA emulation of fiber-optic channels for deep-BER evaluation of DSP implementations," in *OSA Advanced Photonics Congress, SPPCom,* (2019), p. SpTh1E.4.

[2] J. N. Damask, "A programmable polarization-mode dispersion emulator for systematic testing of 10 Gb/s PMD compensators," in *Opt. Fiber Commun. Conf. (OFC),* (2000), pp. 28–30 vol.3.

[3] F. Curti, B. Daino, Q. Mao, F. Matera, and C. Someda, "Concatenation of polarisation dispersion in single-mode fibres," Electron. Lett. **25**, 290–292 (1989).

[4] J. P. Gordon and H. Kogelnik, "PMD fundamentals: Polarization mode dispersion in optical fibers," Proceedings National Academy Sciences United States America **97**, 4541–4550 (2000).

[5] V. Valimaki and T. I. Laakso, "Principles of fractional delay filters," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing,* vol. 6 (2000), pp. 3870–3873.