FPGA Implementation of a low-complexity fading filter for multipath Rayleigh fading simulator

Kadir Atilla Toker[1], Serdar Özen[2], and Ali Arsal[3]

[1] Izmir University, 35290 ˙Izmir-Turkey, atilla.toker@izmir.edu.tr

[2] Department of Electrical and Electronics Engineering, Izmir Institute of Technology, 35430 ˙Izmir-Turkey, serdarozen@iyte.edu.tr

[3] Faculty of Engineering and Natural Sciences, Sabanci University, 34956 ˙Istanbul-Turkey, aliarsal@su.sabanciuniv.edu

Abstract

A low-complexity high performance Rayleigh fading simulator and its Field Programmable Gate Array (FPGA) implementation are presented. This proposed method is a variant of the method of filtering of the white Gaussian noise where the filter design is accomplished in the analog domain and transferred into digital domain. The proposed method outperforms AR(20) filter and modified Jakes' generators in performance. Although IDFT method achieves the best performance, it brings a significant cost in storage. The proposed method achieves high performance with the lowest complexity, and its performance has been verified on commercially available FPGA platforms.

1. Introduction

In this paper, a low-complexity prototype hardware architecture for modeling of Rayleigh fading process is developed targeting at the Xilinx [1] Virtex4-xc4vsx35 and Spartan3e-xc3s500e Field Programmable Gate Array (FPGA) development platforms. FPGAs offer a lot of flexibility, fixed-point arithmetic units with parameterized precisions, variable-length registers, numerous dedicated signal processing cores, speed, reliability, low turn-around time in the design phase, and very cheap alternative to the expensive emulators available on the market [2,3]. The FPGA realization of the proposed method has been implemented using constant coefficient multipliers and the theoretical performance of the proposed method has been verified using this FPGA implementation. Hardware-based emulators can greatly reduce the simulation time compared to software based simulators [4]. Many laboratory channel simulation tools use hybrid DSP/FPGA solutions [5,6] or stand-alone FPGAs to generate wireless multipath channel models [7,8]. A much more cost-effective approach is to implement the entire emulator on a single FPGA chip [7]. The main contributions of our work are: (1) a novel low-complexity filter design scheme for fading-channel simulators with quantitative performance analysis; (2) demonstration of the hardware implementation of our filter-based simulator, that produces fading channel characteristics, realized on a portion of a commercially available FPGA platform.

2. Derivation of the fading filter

Rayleigh fading process is characterized by the Gaussian WSS uncorrelated scattering fading model [13]. This statistic generally depends on the propagation geometry, the velocity of the mobile, and the antenna characteristics. A common assumption is that the propagation path consists of two-dimensional isotropic scattering with a vertical monopole antenna at the receiver [9]. In this case the in-phase or quadrature part of the received signal envelope must be independent and each must have zero mean for Rayleigh fading, and theoretical spectral density of in-phase, or the quadrature, part of the received faded signal envelope is ;

$$S(f) = \begin{cases} \dfrac{\sigma^2}{2\pi f_d \sqrt{1-(f/f_d)^2}}, & if\ |f\ | \le f_d \\ 0, & else \end{cases} \tag{1}$$

where σ^2 is the rms value of the envelope of the waveform. f_d is the Doppler frequency which is defined as the ratio of the vehicle speed, V, to the wavelength, λ, $f_d = V/\lambda$, and $\lambda = c/ f_c$ where $c = 3 \times 108$m/s is the speed of the light, and fc is the carrier frequency. The corresponding normalized, unit-variance, continuous time autocorrelation function of the fading gain of the wireless channel under these conditions is $R(\tau) = J_0(2\pi f_d |\tau|)$, where $J_0(\cdot)$ is the zerothorder Bessel function of the first kind. For the discrete-time simulation of this model, ideally generated in-phase and quadrature

Gaussian processes should each have the autocorrelation sequence R[n] = $J_0(2\pi f_m|n|)$, where fm = f_dT is the Doppler frequency normalized by the sampling rate 1/T .

3. Novel filter design

A straightforward method to simulate a faded signal is to amplitude modulate the carrier signal with a low-pass filtered Gaussian noise source as shown in Figure. 1. To obtain time varying frequency selective fading channel we must have a bank of these fading filters where each filter generates the corresponding fading channel tap. A fading filter with impulse response g(k) can be designed so that its output spectral density is an approximation to theoretical spectral density of the complex envelope of the faded signal S(f) of (1). We will use filter structures that were proposed in [10].

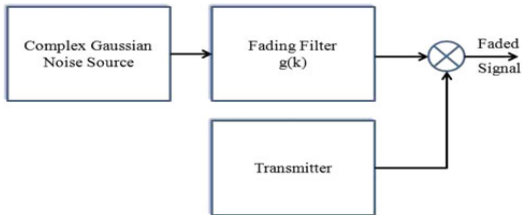

Figure. 1: Faded signal generator that uses low-pass filtered white complex

Consider the elementary first-order filter transfer function $G_1(s)$, and the second-order filter transfer function G2(s) where;

$$G_1(s) = \frac{w_x}{s + w_x}, \qquad and \qquad G_2(s) = \frac{w_x^2}{s^2 + \frac{w_x s}{Q} + w_x^2} \tag{2}$$

in which the parameter Q controls the gain of the peak of the frequency response of the filter where the peak occurs at w = w_x rad/s; for example for the third-order filter if Q =$\sqrt{10}$ then the magnitude of G(·) has a gain of 7dB at w = w_x , in which 10dB gain results from the second-order filter and -3dB results from the first-order part making the overall gain of 7 dB. Then we can have fading filter continuous time transfer functions with higher orders of γ , Gγ (s), that are given by

$$G_\gamma(s) = \begin{cases} G_2^{\gamma/2}(s), & if \quad \gamma \quad even \\ G_1(s) G_2^{(\gamma-1)/2}(s), & if \quad \gamma \quad odd \end{cases} \tag{3}$$

where $G_1(s)$ and $G_2(s)$ are as given in (2). To find the parameters of the fading filter transfer function, G_γ (s), we will first set the filter order γ and Q. Then defining S(f ; ε), as an approximation to the theoretical spectral density of (1), by

$$S(f;\varepsilon) = \begin{cases} \dfrac{\sigma^2}{2\pi f_d \sqrt{1-(f/f_d)^2}}, & if \ |f| \le f_d - \varepsilon, \\ 0, & else, \end{cases} \tag{4}$$

where ε ≍ R^+ is a small positive real number, which can be taken as multiples of the smallest positive number the computing platform that can handle. Then we solved the numerical optimization problem, for fixed γ , f_d and Q,

$$w_x = \text{argmin} \, \| \, S(f;\varepsilon) - |G_\gamma(j2\pi f)|^2 \, \| \, . \tag{5}$$

The result of this numerical optimization (5) gives the frequency, w_x , that minimizes the norm of the distance between the modified theoretical spectral density and the theoretical fading filter spectrum. Theoretical and approximate spectral density, where the approximate spectral density is for the output of the filter $G_3(s)$, are provided in Figure 2.

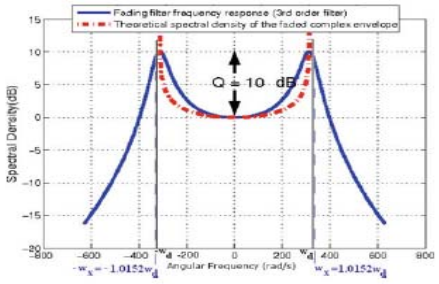

Fig 2: Theoretical and approximate spectral density (for the filter $G_3(s)$)

4. FPGA Implementation

The FPGA implementation of our proposed algorithm, performed using Xilinx's System Generator, is shown in Figure. 3. System Generator Tool produces a design that is targeted towards Xilinx Virtex4 ML402 development kit.

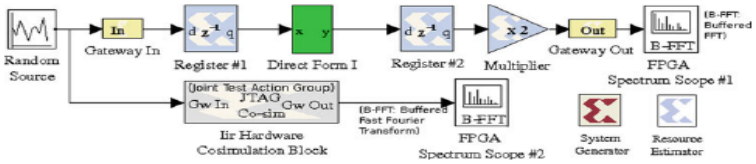

Figure 3: The Simulink, Xilinx co-simulations of the third-order filter for multipath Rayleigh fading simulation on the development board

IIR filters with general purpose multipliers can be used as building blocks for cascade or parallel realizations of higher order IIR filters. Figure 4 shows the detailed implementation of the third-order direct form-I filter, with seven multipliers, using Xilinx System Generator for DSP.

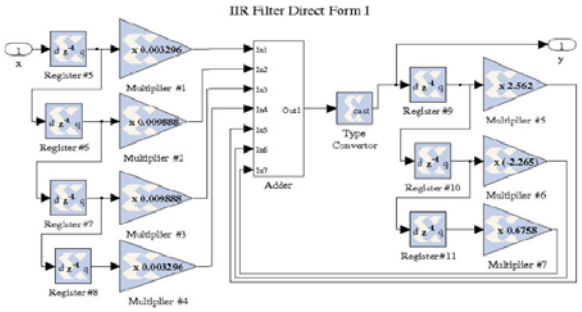

Figure 4: Direct form IIR digital Filter FPGA implementations using System Generator for DSP

A random noise signal generator, using Gaussian-Ziggurat method, is used to generate a discrete time Gaussian white noise signal, and is passed through our ARMA(3,3) filter. Co-simulation has been used to verify our filter frequency response of the real hardware as shown in Figure 5.

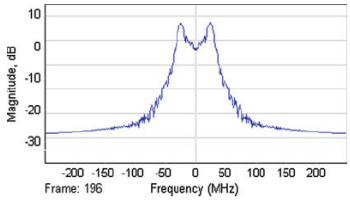

Figure 5: The frequency response of the proposed filter implemented on Virtex4 hardware platform

5. Conclusion

A low-complexity and high-performance implementation of a Rayleigh fading channel emulator was presented. The hardware realization of our channel simulator was implemented using FPGAs, and verified with Xilinx co-simulation.

Our proposed ARMA(3,3) fading generator quantized by 16 bits, outperforms both ARMA(12,12) generator of [11] quantized by 22 bits and modified Jakes' generator quantized by 16 bits. ARMA(12,12) generator fails when using quantization levels lower than 22 bits [11]. Our proposed ARMA(3,3) fading generator works even at 12 bits quantization levels. The main advantage of our ARMA(3,3) Rayleigh fading generator is that it provides accurate performance under all quantization levels for practical usage, such as 12, 14 and, 16 bits of quantization levels, while achieving the lowest complexity of all the Rayleigh fading generators mentioned. Moreover one can fit several ARMA(3,3) generators on a commercially available FPGA board to develop a laboratory tool to implement and realize severe wireless multipath channels similar to those depicted in [12].

FPGA platform as many independent fading generators as the ARMA(3,3) generators would fit in, due to difference in complexity and hardware resource consumption. The coefficients were generated using Matlab 7.2. We report here the hardware co-simulation of the proposed channel emulator uses only 2% of configurable slices, and 4% of the dedicated multipliers and 1% of the available LUTs on the Virtex4 xc4vsx35 platform.

6. References

1. Xilinx. http://www.xilinx.com

2. Agilent Technologies Inc (2005) Baseband studio for fading. Santa Clara, CA

3. Rohde, Schwarz (1999) Baseband fading simulator ABFS, reduced costs through baseband simulation

4. Jämsä T, Poutanen T, Meinila J (2001) Implementation techniques of broadband radio channel simulators. In: Proceedings of the IEEE Vehicular Technology Conference, pp 433–437

5. SalkintzisAK (1999) Implementation of a digital wideband mobile channel simulator. IEEE Trans Broadcasting 45:122–128

6. Wickert MA, Papenfuss J (2001) Implementation of a real-time frequency-selective RF channel simulator using a hybrid DSPFPGA architecture. IEEE Trans Microw Theory Tech 49:1390–1397

7. Alimohammad A, Fard SF, Cockburn BF, Schlegel C (2008) A compact single-FPGA fading-channel simulator. IEEE Trans Circuit Syst II 55:84–88

8. Kahrs M, Zimmer C (2006) Digital signal processing in a real-time propagation simulator. IEEE Trans Instrum Meas 55:197–205

9. Jakes WC (1994) Microwave mobile communications. 2nd edn. IEEE Press, New York

10. Özen S, Zoltowski MD (2001) A fading filter approximation to enable state-space modeling and joint data channel estimation of (time-varying) frequency selective channels with antenna arrays. IEEE Circuits and Systems Society Notre Dame Workshop on Wireless Communication and Networking, South Bend, IN

11. Mehrpouyan H, Blostein SD (2008) ARMA synthesis of fading channels. IEEE Trans Wirel Commun 7(8):2846–2850

12. Homier EA, Scholtz RA (2002) Rapid acquisition of ultrawideband signals in the dense multipath channel. IEEE Conference on Ultra Wideband Systems and Technologies, pp 105–109

13. Bello PA (1963) Characterization of randomly time-variant linear channels. IEEE Trans Commun Syst CS-11:360–393