

An FPGA-Based Simulator for High Path Count Rayleigh and Rician Fading

Saeed Fouladi Fard, Amirhossein Alimohammad
Ukalta Engineering, 4344 Enterprise Square
10230 Jasper Ave, Edmonton, AB, T5J 4P6, Canada
Email: {saeed, amir}@ukalta.com

Bruce F. Cockburn
Department of Electrical and Computer Engineering
University of Alberta, Edmonton, AB, T6G 2V4, Canada
Email: cockburn@ece.ualberta.ca

Abstract—We present an ultra-compact and fast hardware simulator for Rayleigh and Rician fading channels. To ensure numerical robustness and an efficient mapping onto hardware, the fading simulator uses the sum-of-sinusoids technique with $N = 32$ sinusoids added up to model each fading path. Fading samples are generated at a low rate and then are passed to an interpolator, which computes the final samples at the desired baseband rate. We propose a new time-multiplexed datapath that uses a differential approach. Instead of generating the fading samples directly, the datapath generates the discrete difference between fading samples. The proposed simulator is so compact that an entire 4×4 MIMO (multiple-input multiple-output) fading channel can be implemented on a small fraction of a single field-programmable gate array (FPGA). On a Xilinx Virtex-4 XC4VLX200-11 FPGA, up to 1184 different paths can be implemented simultaneously while generating 1184×342 million 2×16 -bit complex-valued fading samples per second.

I. MOTIVATION

Simulating the effect of fading channels is a key step in the design and verification of wireless communication systems. It is generally easier and thus more common to design a fading channel simulator in software rather than in hardware. However, accurate simulation of radio channels is a computationally-intensive process and, indeed, software simulation has become a serious bottleneck to timely design and verification. Hardware-based simulators have shown several orders of magnitude of speed-up over software-based simulators [1], [2]. Speed is an increasingly important advantage when many different scenarios must be simulated over large variety of operating conditions. For example, emerging wireless standards, such as long term evolution (LTE), support MIMO (multiple-input multiple-output) technology and over 300 modes of operation that need to be verified [3].

Commercially-available fading channel simulators are rather bulky and costly. These simulators are standalone units that provide the fading signal in the form of analog signals or digital samples [4]–[6]. Internally, hardware-based simulators all appear to use digital hardware, such as field-programmable gate array (FPGA) devices, to simulate fading channels (for example see [7]–[11]).

In this article, we propose a new compact and accurate fading simulator that can be used for cost-effective and flexible simulation of different fading scenarios ranging from simple Rayleigh and Rician channels to the most complex multipath

MIMO communication systems. The implemented hardware is based on the sum-of-sinusoids (SOS) channel model [12] where the fading process is formed by superimposing sinusoidal waveforms with amplitudes, frequencies and phases that are selected appropriately to generate the desired statistical properties. Our fading simulator is small enough that it can be integrated with the rest of the communication system on a single FPGA.

For an especially compact hardware implementation that supports a large number of propagation paths, we propose a differential sample generation scheme. In this method, instead of generating fading samples directly, the discrete difference between subsequent fading samples are generated at a low sample rate using a time-multiplexed datapath. To increase the efficiency and to provide the target sample rate, we propose an optimized linear interpolator using a simple multiply-accumulate unit. Direct implementation of the SOS-based fading channel model can lead to reduced accuracy when a large range of Doppler frequencies must be modeled. This is due to the fact that in a typical wireless communication system, the Doppler frequency is significantly smaller than the signal sample rate. Therefore, the ratio of these two values, which is used in SOS-based models, is a relatively small value that requires a large number of fractional bits for an accurate fixed-point representation in a hardware implementation of the fading channel simulator. With the proposed technique, the statistical accuracy of the fading simulator is guaranteed provided the Doppler frequency is kept within a certain range. Using our proposed techniques, a relatively large number of independent streams of fading samples can be parameterized separately to simulate different propagation conditions. Therefore, various fading scenarios, such as single and multipath, Rayleigh and Rician fading channels for single and multiple antenna communication, can be simulated on a single FPGA. While the generated fading samples are correlated in time, the fading channel simulator is also extended to model spatial correlation between antennas as well. To ensure the most compact possible design, we developed a set of fixed-point arithmetic and logical libraries in Mex-C [13]. We empirically minimized the wordlength of signals and the size of the on-chip memories using bit-true fixed-point simulations of the fading channel simulator.

The rest of this article is organized as follows. In section II

we briefly review the SOS fading channel models. Section III presents our modified and optimized fading channel model for generating multiple streams of statistically-independent fading processes. Section IV describes an especially compact FPGA simulator design. We also propose a datapath for converting Rayleigh samples into Rician samples. Multiple streams of Rayleigh (or Rician) samples are then combined to simulate different fading channel scenarios. In Section V independent fading paths are used to efficiently implement a 4×4 MIMO fading channel simulator. We then extend this design in Section VI to generate spatiotemporally-correlated fading sequences. Fixed-point simulation results are presented in Section VI. Finally, Section VII makes some concluding remarks.

II. OVERVIEW OF FADING CHANNEL MODELS

Various fading channel simulators based on SOS models have been proposed over the last four decades [14]–[16]. Unfortunately, some of the proposed models, such as the Jakes’ simulator [17] and both its deterministic and stochastic variations, have undesirable statistical inaccuracies. For example, the correlation functions can deviate significantly from those of the theoretical model [18]. Also, some proposed models require a relatively large number of sinusoids to produce multiple uncorrelated fading processes [19], [20]. An improved deterministic SOS-based fading channel model for multiple uncorrelated Rayleigh fading channels was proposed in [21]. Comparative analyses of the popular SOS-based models can be found in [22]–[24]. It was concluded in [22] that the statistical SOS model in [25] has superior properties compared to the other SOS-based models. Although more statistically accurate, the model in [25] is not ergodic, i.e., the time-averaged statistical properties do not converge to the ensemble-average. This implies that multiple simulation runs are required in order to produce accurate fading results. The authors in [9], [10] proposed a modified version of Wu’s model [26]; however, we found that this model is also not ergodic. The model in [25] was further improved in [23] (and later in [11]) to provide more accurate statistics even when averaged over one simulation trial. We utilize the model in [11] for hardware implementation as it provides superior accuracy¹ and its time-averaged statistics match its ensemble-averaged statistics. In this fading model the statistical properties of the generated fading samples are improved by using random processes (RPs) instead of random variables (RVs) that remain fixed after initialization [25]. In this model, each complex discrete-time Rayleigh fading process is described as

$$c[m] = c_i[m] + jc_q[m], \quad (1)$$

¹With similar settings to Wu’s paper [26], we tested the model in [11] and found that the mean-squared-errors (MSEs) of the autocorrelation function for the in-phase and quadrature components were 2.611×10^{-6} and 3.650×10^{-6} , respectively, which are actually lower than the values reported in [26]. Also, the MSEs of the cross-correlation functions of the intra- and inter-faders were 5.386×10^{-5} and 3.298×10^{-5} , respectively, which are again lower than the values reported in [26].

where

$$c_i[m] = \sqrt{\frac{2}{N}} \sum_{n=1}^N \cos(2\pi f_D T_s \cos(\alpha_n[m])m + \varphi_n[m]), \quad (2)$$

$$c_q[m] = \sqrt{\frac{2}{N}} \sum_{n=1}^N \cos(2\pi f_D T_s \sin(\alpha_n[m])m + \psi_n[m]), \quad (3)$$

and

$$\alpha_n[m] = \frac{2\pi n - \pi + \theta[m]}{4N}, \quad n = 1, \dots, N. \quad (4)$$

Here, $m = 0, 1, 2, \dots$ is the discrete time index, f_D is the maximum Doppler frequency for any path, T_s is the symbol period, $\alpha_n[m]$ is the angle of arrival of the n -th sinusoid, and $\varphi_n[m]$ and $\psi_n[m]$ are the phases of the in-phase and quadrature components, respectively, of the n -th sinusoid. The $\psi_n[m]$, $\varphi_n[m]$ and $\theta[m]$ are mutually-independent and uniformly-distributed random walk processes over $[-\pi, \pi)$, for all n .

The motivation behind choosing random walk processes (RWPs) to generate the sinusoid parameters [27] is based on the observation that in an isotropic scattering environment, the angle of arrival changes slowly and continuously. Thus when sampled, the angle of arrival should resemble a RWP [28]. Hence θ , which is uniformly distributed over $[-\pi, \pi)$, should be a highly correlated random process rather than a sequence of uncorrelated uniformly-distributed random variables. Moreover, the behavior of fading channels should change only slowly between consecutive blocks of transmitted symbols. Hence the phase parameter of a sinusoid should also be highly correlated and should not change randomly. Therefore, our improved fading model uses slowly-changing RWPs for both φ_n and ψ_n .

Rician fading samples can be obtained from Rayleigh fading samples by superimposing a term representing line-of-sight (LOS) propagation (or the presence of a strong specular path). This term is commonly assumed to be either a non-zero constant [29], or time-varying and deterministic [30], or time-varying and stochastic [31]. In this article we assume that the LOS term is a time-varying and stochastic process as this assumption produces a more accurate channel model [32]. The resulting Rician fading channel model is expressed in discrete time as follows:

$$r[m] = r_i[m] + jr_q[m], \quad (5)$$

where

$$r_i[m] = \frac{c_i[m] + \sqrt{K} \cos(2\pi f_D T_s \cos(\theta_0)m + \phi_0)}{\sqrt{K+1}}, \quad (6)$$

and

$$r_q[m] = \frac{c_q[m] + \sqrt{K} \sin(2\pi f_D T_s \cos(\theta_0)m + \phi_0)}{\sqrt{K+1}}, \quad (7)$$

where the Rice factor K is the ratio of the power of the LOS (or specular) component to the scattered power, and $c_i[m]$ and $c_q[m]$ are given by (2) and (3), respectively. Also, θ_0 and

ϕ_0 are the angle of arrival and the initial phase of the LOS component, respectively, which are uniformly distributed RVs over $[-\pi, \pi)$.

An important point to note is that the number N of component sinusoids has a significant impact on the statistical properties of the generated fading samples. For small N , the components of the generated complex samples deviate from the ideal Gaussian distribution, and hence the magnitude of the fading samples deviates from the ideal Rayleigh distribution. As N increases, the orthogonal components of the generated samples become increasingly Gaussian according to the Central Limit Theorem. Therefore, it is important to include enough sinusoids for the fading simulation to have an accurate sample distribution. However, increasing the number N of sinusoids directly increases the computational requirement of the fading channel model, especially when many propagation paths needs to be supported (e.g., multiple antennas systems over frequency-selective fading channels). Therefore, it is important to find a computationally-efficient and compact design for the accurate and efficient implementation of fading channels.

III. OPTIMIZED FADING CHANNEL MODEL FOR EFFICIENT HARDWARE IMPLEMENTATION

In a typical wireless communication scenario, the Doppler frequency f_D is significantly smaller than the signal sample rate $F_s = 1/T_s$. This allows us to design much of the fading channel simulator at a much lower sample rate and thereby reduce the required hardware resources. The resulting low-rate signal can then be interpolated to achieve the desired output sample rate. To be able to use a simple linear interpolator, the signal bandwidth must be small enough so that the frequency response of the interpolator does not have a significant impact on the statistics of the generated samples. If not attenuated considerably, the unwanted image signals resulting from upsampling will significantly affect the level-crossing-rate (LCR) and average fade duration (AFD) properties of the fading samples. Given the crucial importance of the LCR and AFD for accurate simulation of wireless links, it is necessary to make these properties match the reference model. Our simulation shows that if the unwanted image signals are kept at least 80 dB less than the signal (fading sample) level, their effect would be negligible on the LCR and AFD properties in the practical signal (data) range. More specifically, if the initial sample rate \hat{F}_s in the SOS stage is larger than $64 \times f_D$ (equivalently, the initial sample time $\hat{T}_s < [64 \times f_D]^{-1}$), a simple linear interpolator can sufficiently attenuate the image signals with no significant effect on the desired signal.

We first alter the representation of the fading model to make its implementation more efficient. It is convenient if all pseudorandomly-generated variables fall between 0 and 1. To this end we can rewrite (4) as

$$\alpha_n[m] = 2\pi \left(\frac{1}{4N}(n - \hat{\theta}) \right), \quad n = 1, \dots, N \quad (8)$$

where $\hat{\theta} \in [0, 1)$. We define $\hat{\alpha}_n[m] = \alpha_n[m]/(2\pi)$, $\hat{\varphi}_n[m] = (\pi + \varphi_n[m])/(2\pi)$ and $\hat{\psi}_n[m] = (\pi + \psi_n[m])/(2\pi)$ to normalize these variables to the range $[0, 1)$. Note that adding π to a random phase does not change its statistical properties. From (8) it can be verified that $\hat{\alpha}$ lies within $[0, 1/4)$, for all n . Consider the function $f(x) = 64 \times (f_D/\hat{F}_s) \sin(2\pi x)$, for $x \in [0, 1/4)$. Since we chose $\hat{F}_s \geq 64 \times f_D$, the value of $f(x)$ is limited to the range $[0, 1]$. Also, we define the function $g(x) = \cos(2\pi x)$. With the above definitions, summation (2) at sample rate \hat{F}_s can be written as

$$\hat{c}_i[m] = \sqrt{\frac{2}{N}} \sum_{n=1}^N g\left(f\left(\frac{1}{4} - \hat{\alpha}_n[m]\right) \frac{m}{64} + \hat{\varphi}_n[m]\right). \quad (9)$$

Note that the inner cosine function in (2) is obtained from $f(\cdot)$ using the identity $\cos(2\pi x) = \sin(2\pi(1/4 - x))$.

Since process $\hat{\alpha}_n[m]$ is highly correlated and changes only slowly with time, one can further simplify the implementation of (9). More specifically, we can write

$$f\left(\frac{1}{4} - \hat{\alpha}_n[m]\right)m \approx \beta_{in}[m] = \sum_{\hat{m}=1}^m f\left(\frac{1}{4} - \hat{\alpha}_n[\hat{m}]\right). \quad (10)$$

Note that $\beta_{in}[m]$ can be written in recursive form as $\beta_{in}[m] = \beta_{in}[m-1] + f(1/4 - \hat{\alpha}_n[m])$ with $\beta_{in}[-1] = 0$. As one can verify in simulation, this modification replaces the repeated multiplications with a running sum with negligible impact on the statistical properties. With similar modifications to $c_q[m]$, simplified summations for the in-phase and quadrature components can be written in discrete time as follows:

$$\hat{c}_i[m] \approx \sqrt{\frac{2}{N}} \sum_{n=1}^N g(\beta_{in}[m]/64 + \hat{\varphi}_n[m]), \quad (11)$$

and

$$\hat{c}_q[m] \approx \sqrt{\frac{2}{N}} \sum_{n=1}^N g(\beta_{qn}[m]/64 + \hat{\psi}_n[m]), \quad (12)$$

where $\beta_{qn}[m]$ is defined recursively as $\beta_{qn}[m] = \beta_{qn}[m-1] + f(\hat{\alpha}_n[m])$ with $\beta_{qn}[-1] = 0$.

Fading samples generated at the slower sample rate \hat{F}_s are then oversampled and interpolated I times to provide samples at the output rate $F_s = I \times \hat{F}_s$. Since we chose $\hat{F}_s \geq 64 \times f_D$, a linear interpolator can be used effectively for this purpose. Specifically, the interpolator output at time $mI + i$, for $m = 0, 1, 2, \dots$ and $i = 0, 1, \dots, I-1$, is given recursively as

$$\begin{aligned} y[mI + i] &= \frac{(y[(m+1)I] - y[mI])i}{I} + y[mI] \\ &= \sum_{j=0}^i \frac{y[(m+1)I] - y[mI]}{I} + y[mI]. \end{aligned} \quad (13)$$

To simplify the hardware implementation, we constrain the interpolation factor I to be a power of 2, i.e., $I = F_s/\hat{F}_s = 2^k$, where $k = \lfloor \log_2(F_s/64f_D) \rfloor$ and $\lfloor x \rfloor$ denotes the largest integer number that is smaller than x . In this case, the interpolator (13) can be implemented without multiplications or divisions. Note that the interpolator requires the discrete

difference between two subsequent low-frequency samples. In addition, since the difference is a linear and time-invariant operation, it can be performed before adding in any Rician specular component. The discrete difference signal for the Rayleigh in-phase component in (11) is thus

$$d_i[m] = \sum_{n=1}^N \frac{g\left(\frac{\beta_{in}[m]}{64} + \hat{\varphi}_n[m]\right) - g\left(\frac{\beta_{in}[m-1]}{64} + \hat{\varphi}_n[m-1]\right)}{\sqrt{N/2}}. \quad (14)$$

By substituting (14) into (13), one can verify that the in-phase Rayleigh fading samples in (2) at sample rate F_s can be approximated by

$$\begin{aligned} \hat{c}_i[mI + i] &= \hat{c}_i[0] + \sum_{\hat{m}=0}^{m-1} d_i[\hat{m}] + 2^{-k} d_i[m]i \\ &= \hat{c}_i[0] + \sum_{\hat{m}=0}^{m-1} d_i[\hat{m}] + 2^{-k} \sum_{j=1}^i d_i[m] \\ &= 2^{-k} \sum_{\hat{m}=0}^m \sum_{j=1}^i d_i[(\hat{m} + 1)I] + \hat{c}_i[0], \end{aligned} \quad (15)$$

where $m = 0, 1, 2, \dots$ and $i = 0, 1, \dots, I - 1$. Equation (15) shows that with the above modifications, the interpolation operation can be simplified to a discrete difference and an accumulation.

Next, the discrete difference of the LOS component is added to the Rayleigh fading samples. The discrete difference signal for Rician samples can be written as

$$s_i[m] = \frac{1}{\sqrt{1+K}} d_i[m] + \sqrt{\frac{K}{1+K}} (g(\lambda[m]) - g(\lambda[m-1])), \quad (16)$$

where $\lambda[m] = \sum_{\hat{m}=1}^m (\eta/64) + \hat{\phi}_0$, $\hat{\phi}_0 = \phi_0/(2\pi)$, and $\eta = 64 \times (f_D \hat{T}_s) \cos(\theta_0)$. The in-phase component of the final Rician samples are interpolated with an accumulator as follows:

$$\hat{r}_i[mI + i] = 2^{-k} \sum_{\hat{m}=0}^m \sum_{j=0}^i s_i[(\hat{m} + 1)I] + \hat{r}_i[0], \quad (17)$$

where $\hat{r}_i[0] = (\hat{c}_i[0] + \sqrt{K}g(\hat{\phi}_0))/\sqrt{1+K}$. The interpolated quadrature component $\hat{r}_q[mI + i]$, for $m = 0, 1, 2, \dots$ and $i = 0, 1, \dots, I - 1$, can be calculated similarly. An efficient hardware design for an especially compact and high-throughput simulator based on the above simplified fading channel model is described in the next section.

IV. IMPLEMENTATION OF THE FADING CHANNEL SIMULATOR

Without loss of generality, we explain the simulator design for 32 channels with each channel providing Rayleigh fading with $N = 32$ component sinusoids. The architecture of our fading simulator consists of two cascaded stages. In the first stage, the complex sinusoids are generated at the sample rate \hat{F}_s . Since \hat{F}_s is much slower than the target sample rate F_s , a common data path can be time-shared to interleave the calculations for different paths. In the second stage every

low-speed stream of fading samples is interpolated with a dedicated interpolator. This way, the low-speed calculations are performed more efficiently with the least amount of hardware.

For a more compact hardware implementation, the random processes $\hat{\theta}$, $\hat{\varphi}_n[m]$ and $\hat{\psi}_n[m]$ are updated according to a modified version of the algorithm described in [11]. Instead of updating the RP values every clock cycle, without significant loss in final accuracy, we update their values every l clock cycles, where $1 < l \leq L$. In this case, we use a time-overlapped approach where one simulation trial is divided into shorter intervals, each of length L time units (e.g., clock cycles).

The proposed updating procedure for random processes is given in Algorithm 1. In this algorithm, χ denotes one of the above RPs and u_χ is a random variable with independent, uniformly-distributed samples over $[0, 1)$. The coefficient ξ is chosen to be small enough to produce highly correlated processes. Some suitable values for ξ were suggested in [23]. The step size δ_χ for each random walk process is the product of ξ , a second coefficient \sqrt{l} used to compensate for the missed update cycles, and a generated uniform sample $u_\chi[m]$. As shown on line 3, the product $\xi\sqrt{l}$ is replaced with the nearest power-of-2, $2^{-\lceil \log_2(\xi\sqrt{l}) \rceil}$, to replace multiplications with shift operations. Lines 5 and 6 ensure that the process $\chi[m]$ remains within the range $[0, 1]$.

Algorithm 1 The updating procedure for random process $\chi[m]$

- 1: **Initialize:** $0 < \xi \ll 1$; $d = \lceil \log_2(\xi\sqrt{l}) \rceil$; $s_\chi = 1$; $\chi[0] \in U(0, 1)$; $u_\chi \in U(0, 1)$;
 - 2: **for** $(m = 0, 1, 2, \dots)$ {
 - 3: $\delta_\chi = s_\chi \cdot (2^{-d} \times u_\chi[m])$;
 - 4: $\chi[m] = \chi[m-1] + \delta_\chi$;
 - 5: **if** $(\chi[m] > 1)$ { $\chi[m] = 1$; $s_\chi = -s_\chi$; }
 - 6: **if** $(\chi[m] < 0)$ { $\chi[m] = 0$; $s_\chi = -s_\chi$; }
 - 7: }
-

Algorithm 1 allows us to share the same functional units for updating three RPs $\hat{\theta}[m]$, $\hat{\varphi}_n[m]$ and $\hat{\psi}_n[m]$. Figure 1 shows the datapath for updating the RPs $\hat{\theta}[m]$, $\hat{\alpha}_n[m]$, $\hat{\varphi}_n[m]$ and $\hat{\psi}_n[m]$. The initial values of three RPs $\hat{\alpha}_n[m]$, $\hat{\varphi}_n[m]$ and $\hat{\psi}_n[m]$ are stored in three block memories “RAM α ”, “RAM φ ” and “RAM ψ ”, respectively, in $Q(16, 15)$ format (i.e., word length = 16 and fraction length = 15) each of depth 1024 (i.e., for 32 paths each using 32 sinusoids). Also, 32 values of $\hat{\theta}$ are stored in the dual-port distributed memory “RAM θ ” in $Q(16, 15)$ format. The RPs $\hat{\theta}[m]$, $\hat{\varphi}_n[m]$ and $\hat{\psi}_n[m]$ are updated according to Algorithm 1 and the value of RPs $\hat{\alpha}_n[m]$ are updated as $\hat{\alpha}_n[m] = (n - \hat{\theta})/(4N)$, for $n = 1, 2, \dots, N$. Note that the updated values of these four RPs are calculated sequentially using time-shared arithmetic resources but distinct memory locations, as shown in Fig. 1. A fading variate generator must create long periods of non-repeating propagation conditions to evaluate accurately the error-rate performance of the communication system under

evaluation. To ensure this, we use a combined linear pseudo-random number generator (PNG) [33] that has a very long period and substantially better randomness and correlation properties compared to conventional linear PNGs.

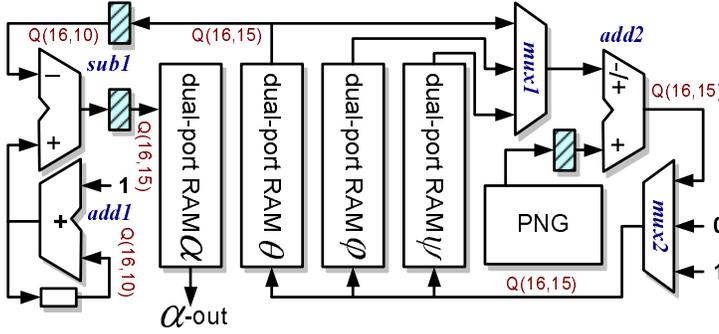


Fig. 1. Datapath for generating the random processes.

Figure 2 shows the proposed datapath for calculating the sequences of differences, $d_i[m]$ and $d_q[m]$. The function $f(x)$, for $x \in [0, 1/4)$, is precomputed and quantized in $Q(16, 15)$ format in 1024 steps and stored in “ROM f ”. To calculate the sine function, the value of $\hat{\alpha}_n[m]$ is passed to “ROM f ” after proper bit selection. To calculate the cosine function, the reformatted $\hat{\alpha}_n[m]$ is first passed through a negation circuit and then passed to “ROM f ”. Through bit-true fixed-point simulations we found that the particular fixed-point representation of $\beta_{in}[m]$ and $\beta_{qn}[m]$ has a great impact on the output statistics. Specifically, we found that the $Q(22, 15)$ fixed-point representation provides enough accuracy for our purposes. 2048 = $2 \times 32 \times 32$ values of $\beta_{in}[m]$ and $\beta_{qn}[m]$ are stored in dual-port block memory “RAM β ”. The $\beta_{in}[m]$ and $\beta_{qn}[m]$ values are updated using adder “add1”, according to (10), after reformatting the $f(\cdot)$ values from $Q(16, 15)$ to $Q(22, 15)$. Moreover, $\hat{\varphi}_n$ and $\hat{\psi}_n$ from two-port block memories “RAM φ ” and “RAM ψ ” are used to compute phases in (11) and (12) after proper bit selection.

In Fig. 2, “cos module” provides $g(x) = \cos(2\pi x)$ values for two inputs from a look-up table. According to our fixed-point simulations, to ensure acceptable statistical accuracy, the look-up table for $g(x)$ in $Q(16, 15)$ format should have at least 4096 entries (requiring 12-bit addressing). For a more efficient implementation, only the first quarter cycle of $g(x)$, i.e., $x \in [0, 1/4)$, is stored in an on-chip block memory. For $x \in [1/4, 1)$, we can find the corresponding values of $g(x)$ based on the values from the first quarter cycle. The outputs of the “cos module” are then passed to accumulators “add4” and “add5” to compute scaled copies of (11) and (12). The outputs of “add4” and “add5” are then passed to “sub1”, which computes the corresponding differences $d_i[m]$ and $d_q[m]$ (see (14)) in $Q(20, 15)$ format. After scaling and proper bit selection, 32 values of $d_i[m]$ and $d_q[m]$ are stored in a distributed memory of depth $64 = 2 \times 32$ in $Q(16, 15)$ format.

The specular component is added to the Rayleigh samples using the datapath shown in Fig. 3. Here, $N = 32$ values of

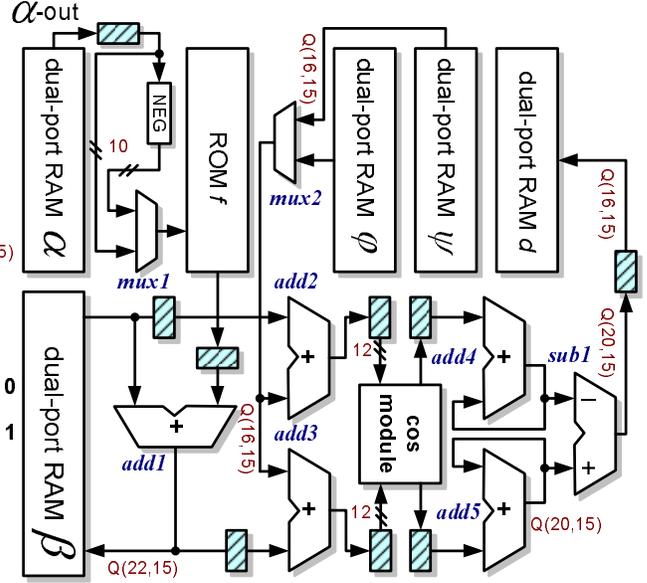


Fig. 2. Datapath for generating Rayleigh fading samples.

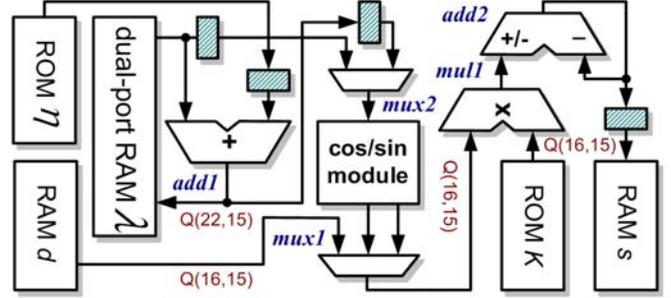


Fig. 3. Datapath for converting Rayleigh fading samples into Rician fading samples.

$\eta = 64 \times (f_D/\hat{F}_s) \cos(\theta_0)$ are stored in single-port memory “ROM η ” and used to update the λ values stored in memory “RAM λ ” in $Q(22, 15)$ format. Memory “RAM λ ” is initialized with $\hat{\phi}_0$ values. The “sin/cos module” reads the sine or cosine of the λ for the specular component (see (6) and (7)) from a look-up table. Memory “ROM K ” holds $64 = 2 \times 32$ values of $1/\sqrt{1+K}$ and $\sqrt{K/(1+K)}$ in $Q(16, 15)$ format. Multiplier “mul1” performs the four multiplications required to calculate $s_i[m]$ and $s_q[m]$ (see (16)). Adder/subtractor “add2” accumulates different components of (16) the result of which, after proper bit selection, is stored using format $Q(16, 15)$ in distributed memory “RAM s ” of depth $64 = 2 \times 32$.

The datapath of the interpolator is shown in Fig. 4. Note that one interpolation branch is dedicated to every in-phase and quadrature stream of samples ($64 = 2 \times 32$ interpolation branches in total). Each interpolation branch consists of a 24-bit accumulator and a register that holds the input signal for an interval of I samples (see (13)). Data from memory “RAM s ” is read and stored in these registers with specific timing. A decoder circuit selects which interpolator branch should store the present output data from “RAM s ”. The interpolator circuit

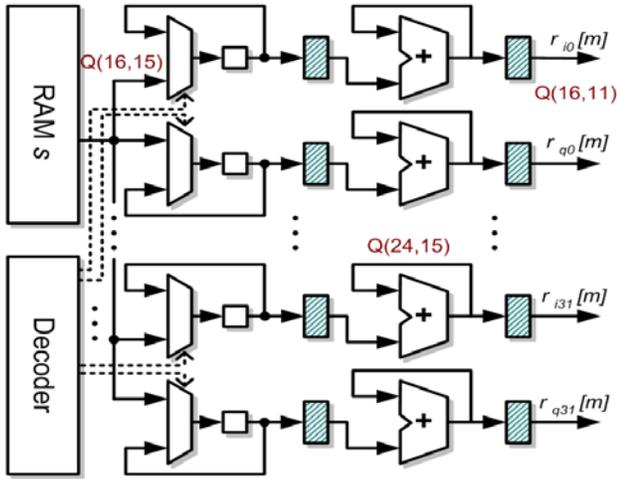


Fig. 4. Datapath for interpolation.

generates the final 32 streams of independent complex Rician fading samples at the desired output sample rate.

We implemented our fading channel simulator on a Xilinx Virtex-II Pro XC2VP100-6 FPGA. As in our example simulator, we configured the hardware to generate 32 independent streams of Rician fading samples using $N = 32$ sinusoids for each stream. For each fading stream, our FPGA implementation uses 2151 of the 44096 configurable slices (4%), only one of the 444 dedicated 18×18 multipliers (< 1%), and nine of the 444 on-chip memory blocks (2%). When implemented with one clock source, our fading generator can generate up to 32×224 million 16-bit complex-valued fading samples per second. Note that the final sample rate of the fading simulator depends on the maximum speed of the interpolator if different clock sources are used for the first (wave superposition) and second (interpolation) stages. When implemented with two clock sources, this figure rises to 32×276 million samples per second, which is the maximum speed of the interpolation circuit in Fig. 4.

Table I summarizes the characteristics of the new fading simulator along with those of the fading channel simulators in [23] and [11]. Note that the proposed fading simulator is more accurate (32 sinusoids versus 8 sinusoids), 18 times smaller (in terms of the number of slices), and 37% faster than the previous design in [11]. We also implemented the new fading simulator on a Xilinx Virtex-4 XC4VLX200-11 FPGA. The hardware was configured to generate $1184 = 37 \times 32$ independent streams of Rayleigh/Rician fading samples using $N = 32$ sinusoids. This implementation uses 90% of the configurable slices, 38% of the XteremeDSP slices, and 99% percent of the 18 Kb on-chip block memories and it can generate up to 1184×342 million 2×16 -bit complex fading samples per second. These flat fading channels can be combined to model different fading scenarios ranging from 1184 independent flat fading channels to frequency-selective fading channels with 1184 resolvable paths. In a frequency-selective fading channel, the transmitted signal typically reaches the

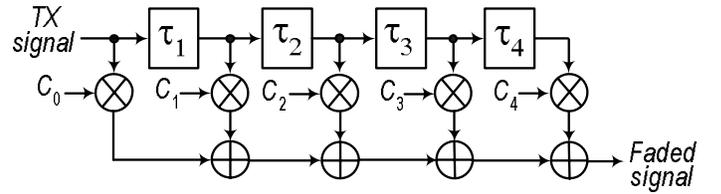


Fig. 5. Effect of a 5-path frequency-selective channel on a transmitted signal.

TABLE I
CHARACTERISTICS OF DIFFERENT SOS-BASED FADING CHANNEL SIMULATORS IMPLEMENTED ON A XILINX VIRTEX-II PRO XC2VP100-6 FPGA

Model	[23]	[11]	Proposed
Fading	Rayleigh	Rician	Rician
# of paths	9	32	32
# of Sinusoids N	8	8	32
# of complex waves	72	256	1024
Clock freq. (MHz)	204	201	224
Output rate (MS/sec)	9×204	32×201	32×276^a
Configurable slices	21996 (50%)	39104 (87%)	2151 (4%)
# of multipliers	48 (97%)	160 (36%)	1 (0.2%)
# of BRAMs	12 (24%)	352 (79%)	9 (2.0%)

^aSample rate of the new fading simulator depends on the maximum speed of the interpolator (maximum 276 MHz). If different clock sources are used for the first (wave superposition running at a maximum frequency of 224 MHz) and the second (interpolation) stages. If one clock source is used, the maximum sample rate will be 32×224 million complex fading samples per second.

receiver over multiple propagation paths, each path having a different relative delay and amplitude. The increased delay spread of the channel corresponds to a reduced coherence bandwidth. Frequency-selective attenuation occurs when the coherence bandwidth of the channel falls below the bandwidth of the signal. Figure 5 shows the datapath of a five-path fading channel. Each path can have a different delay with respect to the zeroth path. These relative delays are not necessarily integer multiples of the sample period. However, path delays are often represented as integer multiples of the sample period for simplicity. This model is also known as the tapped delay channel model [34]. To implement the path delays, one can utilize the dedicated block memories available on the FPGAs. These memories can be configured differently according to the bit-width of the fading coefficients and the delay of each path. They can also be cascaded to obtain longer path delays. The maximum path delay is thus limited by the depth of these on-FPGA BRAMs. The multipliers and adders in Fig. 5 are complex operators. Note that if the clock frequency F_{clk} of the fading channel is greater than the target signal sampling rate F_s , then the complex multipliers and adders can be time-multiplexed to perform the multiply-accumulate operations in Fig. 5.

V. SPATIOTEMPORALLY-CORRELATED MIMO FADING CHANNEL SIMULATION

The independent fading channels can also be combined to simulate MIMO channels. While in the above model we assumed that different fading sequences are correlated in time but uncorrelated in space; however, in a typical MIMO

scenario, the fades usually exhibit spatial correlations [35]. Propagation characteristics of the environment and physical parameters, like antenna spacing and orientation, affect the spatial correlation between different antennas. For simulating multiple antenna systems, these spatial correlations between channels need to be user-controllable. To obtain the space-time correlation characteristics, a temporally-correlated random process can be followed by a linear transformation to be made spatially-correlated [35]. Here we assume that Σ_T and Σ_R are the long-term stable transmitter and receiver correlation matrices, respectively. If $\mathbf{C}[m]$ denotes an $n_R \times n_T$ matrix of unity-variance, temporally-uncorrelated i.i.d Gaussian variables at time m , then the spatiotemporally-correlated MIMO channel model can be written as [35]

$$\mathbf{H}[m] = \mathbf{A}^H \mathbf{C}[m] \mathbf{B}, \quad (18)$$

where \mathbf{A} and \mathbf{B} can be obtained using the Cholesky decompositions of $\Sigma_T = \mathbf{B}\mathbf{B}^H$ and $\Sigma_R = \mathbf{A}\mathbf{A}^H$, respectively. To normalize the channel gain, we set $\text{Tr}(\Sigma_T) = n_T$ and $\text{Tr}(\Sigma_R) = n_R$, where the trace $\text{Tr}(\mathbf{X})$ of a matrix \mathbf{X} denotes the sum of the diagonal elements of \mathbf{X} . The Cholesky decomposition of correlation matrices can be calculated in software in advance and then loaded as constant parameters into the hardware simulator to minimize the run-time computational complexity.

To illustrate the new implementation technique, we start by writing the difference between $\mathbf{H}[m]$ and $\mathbf{H}[m-1]$. The difference matrix $\mathbf{E}[m]$ is

$$\begin{aligned} \mathbf{E}[m] &= \mathbf{H}[m] - \mathbf{H}[m-1] \\ &= \mathbf{A}^H \mathbf{C}[m] \mathbf{B} - \mathbf{A}^H \mathbf{C}[m-1] \mathbf{B} \\ &= \mathbf{A}^H (\mathbf{C}[m] - \mathbf{C}[m-1]) \mathbf{B} \\ &= \mathbf{A}^H \mathbf{D}[m] \mathbf{B}. \end{aligned} \quad (19)$$

From here, since $D_{ij}[m] = C_{ij}[m] - C_{ij}[m-1]$, we can use (14) to find the in-phase (and similarly the quadrature) elements of the matrix \mathbf{D} . Then the elements of $\mathbf{E}[m]$ can be calculated according to (19) at frequency \hat{F}_s . Spatiotemporally-correlated MIMO fading samples are then interpolated I -times with a matrix accumulator as follows:

$$\mathbf{H}[mI+1] = 2^{-k} \sum_{\hat{m}=0}^m \sum_{j=0}^i \mathbf{E}[(\hat{m}+1)I] + \mathbf{E}[0]. \quad (20)$$

Figure 6 shows the datapath for generating spatiotemporally-correlated Rayleigh fading samples for MIMO fading simulation. To simulate MIMO channels, the independent elements of the matrix $\mathbf{D}[m]$ are stored in dual-port memory ‘‘RAM d’’. Matrices \mathbf{A} , \mathbf{B} and \mathbf{E} as well as temporary results are stored in memory ‘‘RAM A-B-H-T’’, which is implemented using a two-port on-chip block memory. This datapath computes the elements of matrix $\mathbf{E}[m]$ according to (19). The matrix entries are then interpolated, as explained in the previous section, to provide the MIMO channel fading samples.

To demonstrate the accuracy of this MIMO fading channel simulator, we utilize the simulator for the bit

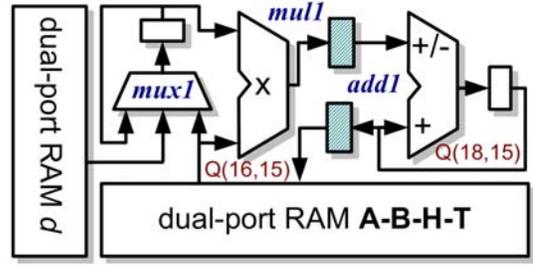


Fig. 6. Datapath for generating spatiotemporally-correlated Rayleigh fading samples for MIMO fading simulation.

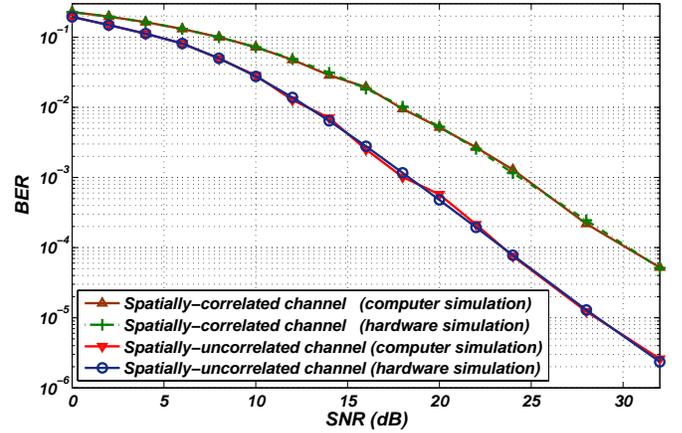


Fig. 7. Bit error rate (BER) performance of a 2×2 MIMO system.

error rate (BER) performance measurement of a 2×2 MIMO system on a GVA-290 FPGA board [36]. In this system, the transmitted bits are modulated with 4-QAM modulation and at the receiver, a maximum-likelihood (ML) detector demodulates the received samples. Also, the receiver is assumed to have perfect knowledge of channel state information. Figure 7 shows the hardware BER performance of this 2×2 MIMO system for spatially-uncorrelated as well as spatially-correlated scenarios using equation (20). In this simulation, signal to noise ratio (SNR) is define as the total transmitted signal power to total noise power at the receiver. Moreover, the sample rate is set to 3.125 million samples per second or 12.5 Mbps and the Doppler frequency is $f_D = 350$ Hz. For the spatially-correlated fading channel, the \mathbf{A} and \mathbf{B} matrices are set to

$$\mathbf{A} = \begin{pmatrix} 0.9027 & 0 \\ -0.2012 - j0.8535 & 0.6480 \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} 0.4163 & 0 \\ -1.1337 + j0.7357 & 0.0472 \end{pmatrix}.$$

The above values are chosen randomly to represent a hypothetical communication scenario. Figure 7 also plots the computer simulation results from a floating-point model of equation (18). As this figure shows, for both spatially-uncorrelated and spatially-correlated fading channels, our

TABLE II
CHARACTERISTICS OF DIFFERENT SOS-BASED MIMO FADING CHANNEL SIMULATORS

Design	I [7]	II [37]	III <i>NEW</i> ^a
# of Channels	1	2	2
# of Sinusoids N	16	8	32
Clock freq. (MHz)	50	201	224
Output rate (MS/sec)	4×1.5	$2 \times 4 \times 201$	$2 \times 4 \times 276$
Configurable slices	22576 (58%)	41198 (93%)	1877 (4%)
# of multipliers	—	272 (61%)	1 (0.2%)
# of BRAMs	(17%)	288 (65%)	9 (2.0%)

^aDesign I was synthesized to a Altera APEX EP20K1000EBC652-3. Designs II and III were synthesized to Xilinx Virtex-II Pro XC2VP100-6 FPGA.

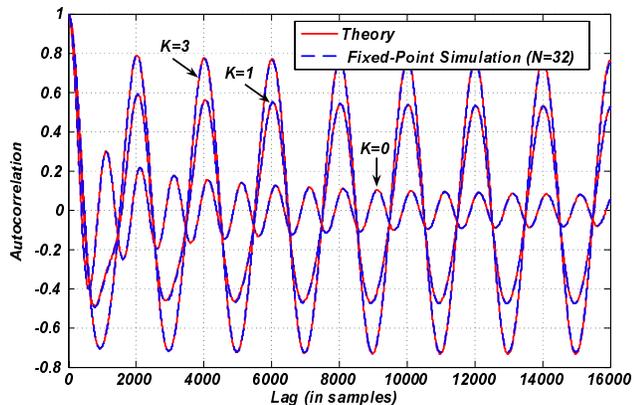


Fig. 8. Autocorrelation of the in-phase and quadrature components of the generated fading samples for one block containing 2×10^6 samples generated using fixed-point simulation with $f_D T_s = 0.001$, $\theta_0 = \pi/3$, and $N = 32$ for $K = 0$ (Rayleigh), 1 and 3.

hardware simulation matches the floating-point results from the original model which verifies the accuracy of our hardware fading simulator.

Table II compares our implementation results with those reported in [7] and [37]. The new MIMO fading simulator was implemented on a Xilinx Virtex-II Pro XC2VP100-6 FPGA. We configured the hardware to generate two spatiotemporally correlated 4×4 MIMO fading channels (i.e., 32 paths altogether) and we used $N = 32$ sinusoids for simulating each Rayleigh fading channel. Our FPGA implementation uses 4% of the configurable slices, one dedicated multiplier, and nine of the 444 on-chip memory blocks (2%). The results in Table II show that the new design is significantly smaller and more efficient than the previous designs.

VI. ACCURACY ASSESSMENT USING FIXED-POINT SIMULATION

To demonstrate the accuracy of the proposed fading channel simulator, we first developed a library of parameterizable fixed-point arithmetic and logical routines in MEX (C for MATLAB). Then we implemented a bit-true model of our proposed fading channel simulator using fixed-point modules and generated sequences of fading samples. We simulated different Rayleigh and Rician fading scenarios for a fading channel with $f_D T_s = 0.001$ and $N = 32$ sinusoids. Figure 8 demonstrates the autocorrelation for 2×10^6 in-phase and

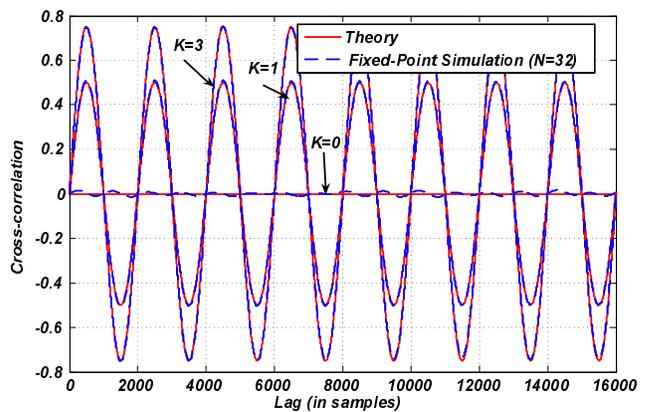


Fig. 9. Cross-correlation between the in-phase and quadrature components of the generated fading samples for one block containing 2×10^6 samples generated using fixed-point simulation with $f_D T_s = 0.001$, $\theta_0 = \pi/3$, and $N = 32$ for $K = 0$ (Rayleigh), 1 and 3.

quadrature components of the generated fading samples for Rician factors $K = 0, 1$ and 3. The theoretical reference ACFs and CCFs are given by the following equations [32]:

$$R_{r_i, r_i}(\tau) = R_{r_q, r_q}(\tau) = \left[\mathcal{J}_0(2\pi f_D \tau) + K \cos(2\pi f_D \tau \cos \theta_0) \right] / (2 + 2K) \quad (21)$$

$$R_{r_i, r_q}(\tau) = -R_{r_q, r_i}(\tau) = \frac{K \sin(2\pi f_D \tau \cos \theta_0)}{2 + 2K} \quad (22)$$

where $\mathcal{J}_0(\cdot)$ is the zeroth-order Bessel function of the first kind. As Fig. 8 shows, there is a close match between the expected analytical autocorrelation plots (computed in floating-point) and the generated fixed-point simulation results. Figure 9 plots the cross-correlation between the in-phase and quadrature components of the generated fading samples and the analytical curves. This figure again shows a close match between the fixed-point simulation results and the desired curves.

As noted above, two important statistical properties of fading channels are the LCR and the AFD. The LCR is the rate at which the envelope crosses a specified level with positive slope. The LCR characterizes important aspects of the dynamic temporal behavior of envelope fluctuations. AFD indicates how long the envelope stays below a given threshold and hence determines the average length of burst errors. These measures can help us better design the wireless systems for different fading channels [38], [39].

Figure 10 shows the LCR (normalized to $f_D T_s$) of the amplitude of generated complex fading samples along with the theoretical LCR of the fading envelope given by [32]:

$$L_{|R|}(\lambda) = \sqrt{\frac{2(1+K)}{\pi}} \lambda f_D \exp(-K - (1+K)\lambda^2) \times \int_0^\pi \left(1 + \frac{2}{\lambda} \sqrt{\frac{K}{K+1}} \cos^2 \theta_0 \cdot \cos \alpha \right) \times \exp(2\lambda \sqrt{K(1+K)} \cos \alpha - 2K \cos^2 \theta_0 \cdot \sin^2 \alpha) d\alpha. \quad (23)$$

Figure 10 again shows excellent agreement between the theoretical and fixed-point simulation results for different values of Rice factor K .

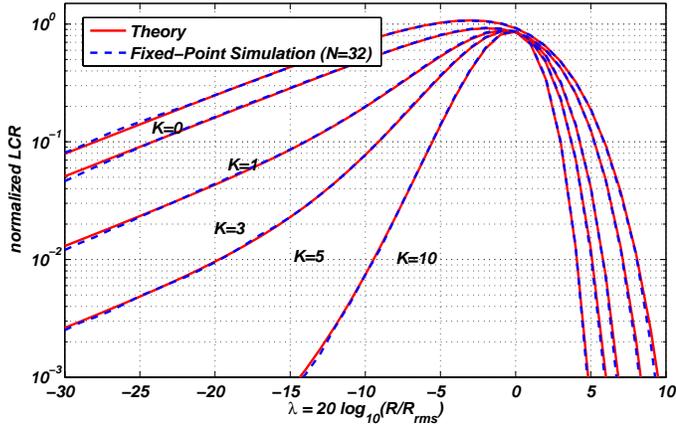


Fig. 10. Normalized level crossing rate function for one block containing 2×10^6 fading samples generated using fixed-point simulation with $\theta_0 = \pi/3$ and $N = 32$ sinusoids, for $K = 0$ (Rayleigh), 1, 3, 5 and 10.

Figure 11 shows the theoretical AFD and the AFD of the generated samples in our fixed-point simulation for 2×10^6 samples and different values for K . The AFD of the fading envelope is given by [32]:

$$T_{|R|}(\lambda) = \frac{1 - Q\left(\sqrt{2K}, \sqrt{2(1+K)\lambda^2}\right)}{L_{|R|}(\lambda)}, \quad (24)$$

where $Q(\cdot)$ is the Marcum Q function [40]. To have distinct curves for illustration, in this simulation the angle of arrival for the specular component is set to $\theta_0 = 0$. As this figure shows, the results of our fixed-point simulation very closely match the theoretical references.

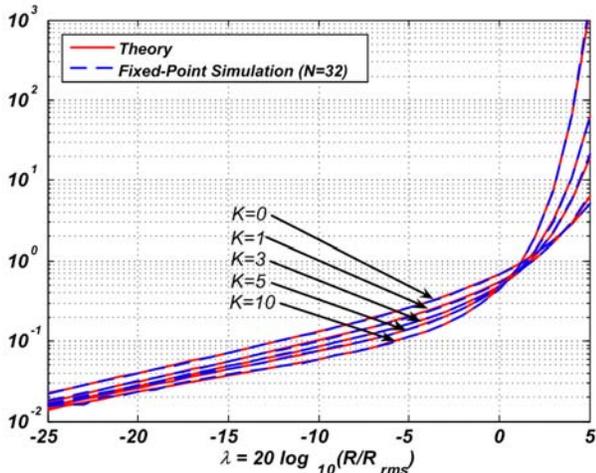


Fig. 11. Normalized average fade duration function for one block containing 2×10^6 fading samples generated using fixed-point simulation with $\theta_0 = 0$ and $N = 32$ sinusoids, for $K = 0$ (Rayleigh), 1, 3, 5 and 10.

Finally, Fig. 12 plots the probability density function (pdf) of the amplitude of the generated samples. Once again it can

be observed that this pdf accurately reproduces its reference curve [32]

$$f_{|R|}(z) = 2(1+K)z \times \exp[-K - (1-K)z^2] \times I_0[2z\sqrt{K(1+K)}], \quad z \geq 0 \quad (25)$$

for five different values of the Rice factor. In (25), $I_0(\cdot)$ denotes the zero-order modified Bessel function of the first kind.

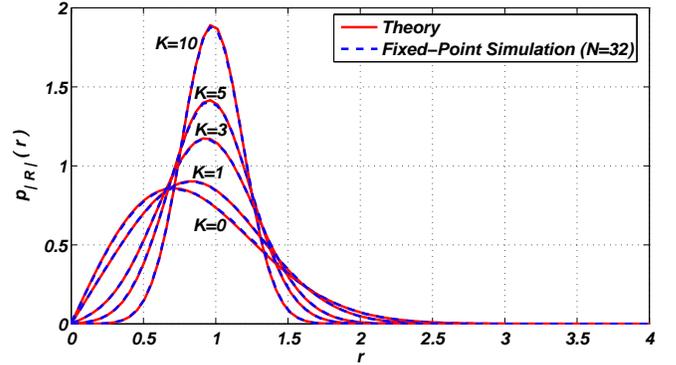


Fig. 12. Probability density function for one block containing 2×10^6 fading samples generated using fixed-point simulation.

VII. CONCLUSIONS

We proposed an especially compact and accurate fading channel simulator that significantly improves on previous designs. We utilized an improved sum-of-sinusoids fading model that permits very compact implementations of Rayleigh and Rician fading channel simulators. Without appreciable loss in accuracy, fading samples are generated at a low rate and interpolated later to match the desired output sample rate. In the first stage of the fading simulator, the discrete difference between fading samples is computed. The discrete difference is then interpolated to produce the final samples. The interpolation factor is chosen to be a power of two to avoid the need for multipliers and thereby reduce the required hardware resources. The statistical accuracy of the new model was confirmed through fixed-point simulation.

The new design can be efficiently mapped onto FPGA hardware. For example, an implementation of a fading channel with 32 paths using 32 complex sinusoids per path on a Xilinx Virtex-II Pro XC2VP100-6 FPGA uses only 4% of the configurable slices, one dedicated multiplier, and nine on-chip memory blocks, while generating over 32×276 million 2×16 -bit fading samples per second. The new simulator is capable of generating up to 1184 independent streams of fading samples on a single Xilinx Virtex-4 XC4VLX200-11 FPGA. These streams can be configured to simulate various fading scenarios including single and multipath, Rayleigh and Rician fading for single and multiple-antenna communication. For example, an entire 4×4 multipath MIMO (multiple-input multiple-output) channel simulator can be implemented on a small fraction of a single FPGA. Each channel can be parameterized independently to support various propagation conditions.

REFERENCES

- [1] M. A. Wickert and J. Papenfuss, "Implementation of a real-time frequency-selective RF channel simulator using a hybrid DSP-FPGA architecture," *IEEE Trans. Microw. Theory Tech.*, vol. 49, no. 8, pp. 1390–1397, 2001.
- [2] D. Derrien and E. Boutillon, "Quality measurement of a colored Gaussian noise generator hardware implementation based on statistical properties," in *Proc. of the IEEE Intl. Symp. on Signal Processing and Information Technology (ISSPIT'02)*, 2002.
- [3] L. M. Correia, *Mobile broadband multimedia networks*. Academic Press, 2006.
- [4] *Baseband Fading Simulator ABFS, Reduced costs through baseband simulation, No. 163*, Rohde & Schwarz, 1999.
- [5] *Baseband Studio for Fading*, Technical Overview, N5115A, Agilent Technologies Inc., 2005.
- [6] *ACE 400NB MIMO Channel Emulator*, Azimuth Systems Inc., Acton, MA, USA, 2006.
- [7] M. Cui, H. Murata, and K. Araki, "Real-time MIMO received signal generator for spatial multiplexing systems," in *Proc. of the IEEE Intl. Veh. Technol. Conf. (VTC'04)*, 2004, pp. 4345–4348.
- [8] A. Alimohammad, S. Fouladi Fard, B. F. Cockburn, and C. Schlegel, "A compact single-FPGA fading channel simulator," *IEEE Trans. Circuits Syst. II*, vol. 55, no. 1, pp. 84–88, Jan. 2008.
- [9] C. H. Liao, T. P. Wang, and T. D. Chiueh, "A novel low-complexity Rayleigh fader for real-time channel modeling," in *Proc. of the IEEE Intl. Symp. on Circuits and Systems (ISCAS'07)*, 2007, pp. 2602–2605.
- [10] T. P. Wang, C. H. Liao, and T. D. Chiueh, "A real-time digital baseband MIMO channel emulation system," in *Proc. of the IEEE Intl. Symp. on Circuits and Systems (ISCAS'07)*, 2007, pp. 2606–2609.
- [11] A. Alimohammad, S. Fouladi Fard, B. F. Cockburn, and C. Schlegel, "An accurate and compact Rayleigh and Rician fading channel simulator," in *Proc. of the IEEE Veh. Technol. Conf.*, 2008, pp. 409–413.
- [12] R. H. Clarke, "A statistical theory of mobile-radio reception," *Bell System Technical Journal*, vol. 47, pp. 957–1000, 1968.
- [13] *MATLAB 7 C and Fortran API Reference*, The Mathworks Inc., Natick, MA, 2008.
- [14] C. X. Wang, M. Pätzold, and D. Yuan, "Accurate and efficient simulation of multiple uncorrelated rayleigh fading waveforms," *IEEE Trans. Wireless Commun.*, vol. 6, no. 3, pp. 833–839, Mar. 2007.
- [15] C. X. Wang, M. Pätzold, and Q. Yao, "Stochastic modeling and simulation of frequency correlated wideband fading channels," *IEEE Trans. Veh. Technol.*, vol. 56, no. 3, pp. 1050–1063, May 2007.
- [16] M. Pätzold, C. X. Wang, and B. O. Hogstad, "Two new sum-of-sinusoids-based methods for the efficient generation of multiple uncorrelated rayleigh fading waveforms," *IEEE Trans. Wireless Commun.*, vol. 8, no. 6, pp. 3122–3131, 2009.
- [17] W. C. Jakes, *Microwave Mobile Communications*. Piscataway, NJ: Wiley-IEEE Press, 1994.
- [18] M. F. Pop and N. C. Beaulieu, "Limitations of sum-of-sinusoids fading channel simulators," *IEEE Trans. Commun.*, vol. 49, pp. 699–708, 2001.
- [19] Y. Li and X. Huang, "The simulation of independent Rayleigh faders," *IEEE Trans. Commun.*, vol. 50, no. 9, pp. 1503–1514, 2002.
- [20] C.-X. Wang and M. Pätzold, "Methods of generating multiple uncorrelated Rayleigh fading processes," in *Proc. of the IEEE Veh. Technol. Conf. (VTC'03)*, 2003, pp. 510–514.
- [21] C.-X. Wang, D. Yuan, H.-H. Chen, and W. Xu, "An improved deterministic SoS channel simulator for multiple uncorrelated Rayleigh fading channels," *IEEE Trans. Wireless Commun.*, vol. 7, no. 9, pp. 3307–3311, 2008.
- [22] C. S. Patel, G. L. Stüber, and T. G. Pratt, "Comparative analysis of statistical models for the simulation of Rayleigh faded cellular channels," *IEEE Trans. Commun.*, vol. 53, pp. 1017–1026, 2005.
- [23] A. Alimohammad, S. Fouladi Fard, B. F. Cockburn, and C. Schlegel, "An improved SOS-based fading channel emulator," in *Proc. of the IEEE Veh. Technol. Conf. (VTC'07)*, 2007, pp. 931–935.
- [24] A. M. M. Donald and J. C. Olivier, "A comparative study of deterministic and stochastic sum-of-sinusoids models of Rayleigh-fading wireless channels," in *Proc. of the IEEE Wireless Communications and Networking Conference (WCNC'07)*, 2007, pp. 2027–2031.
- [25] Y. R. Zheng and C. Xiao, "Improved models for the generation of multiple uncorrelated Rayleigh fading waveforms," *IEEE Commun. Lett.*, vol. 6, pp. 256–258, 2002.
- [26] Z. Wu, "Model of independent Rayleigh faders," *Electronics Lett.*, vol. 40, no. 15, pp. 949–951, July 2004.
- [27] A. Alimohammad, S. F. Fard, and B. F. Cockburn, "A compact Rayleigh and Rician fading simulator based on random walk processes," *IET Commun.*, vol. 3, no. 8, pp. 1333–1342, 2009.
- [28] P. Pampaloni and S. Paloscia, *Microwave Radiometry and Remote Sensing of the Earth's Surface and Atmosphere*. VNU Science Press, 2000.
- [29] K.-W. Yip and T.-S. Ng, "Discrete-time model for digital communications over a frequency-selective Rician fading WSSUS channel," *IEEE Proc. Commun.*, vol. 143, no. 1, pp. 37–42, 1996.
- [30] M. Pätzold, U. Killat, F. Laue, and Y. Li, "On the statistical properties of deterministic simulation models for mobile fading channels," *IEEE Trans. Veh. Technol.*, vol. 47, pp. 254–269, 1998.
- [31] C. Xiao, Y. R. Zheng, and N. Beaulieu, "Statistical simulation models for Rayleigh and Rician fading," in *Proc. of the IEEE Intl. Conf. on Communications*, 2003, pp. 3524–3529.
- [32] C. Xiao, Y. R. Zheng, and N. C. Beaulieu, "Novel sum-of-sinusoids simulation models for Rayleigh and Rician fading channels," *IEEE Trans. Wireless Commun.*, vol. 5, no. 12, pp. 3667–3679, 2006.
- [33] P. L'Ecuyer, "Tables of maximally equidistributed combined LFSR generators," *Math. of Comp.*, vol. 68, no. 225, pp. 261–269, 1999.
- [34] J. G. Proakis, *Digital Communications*. McGraw-Hill, 2001.
- [35] M. Kiessling and J. Speidel, "Statistical transmit processing for enhanced MIMO channel estimation in presence of correlation," in *IEEE Global Telecommunications Conference*, 2003, pp. 2411–2415.
- [36] *GVA-290 Xilinx VirtexE Hardware Accelerator*, GV Associates, Ramona, CA, USA, Aug. 2009. [Online]. Available: <http://www.gvassociates.com>
- [37] A. Alimohammad, S. Fouladi Fard, B. F. Cockburn, and C. Schlegel, "A novel technique for efficient hardware simulation of spatiotemporally correlated MIMO fading channels," in *Proc. of the IEEE Intl. Conf. on Communications*, 2008, pp. 718–724.
- [38] K. Ohtani, K. Daikoku, and H. Omori, "Burst error performance encountered in digital land mobile radio channel," *IEEE Trans. Veh. Technol.*, vol. 30, no. 4, pp. 156–160, Nov. 1981.
- [39] J. M. Morris, "Burst error statistics of simulated Viterbi decoded BPSK on fading and scintillating channels," *IEEE Trans. Commun.*, vol. 40, pp. 34–41, 1992.
- [40] M. K. Simon and M. S. Alouini, *Digital communication over fading channels*, 2nd ed. New York, NY: John Wiley & Sons, 2005.