Hardware Implementation of Nakagami and Weibull Variate Generators

Amirhossein Alimohammad, Saeed Fouladi Fard, Bruce F. Cockburn

***Abstract*—An efficient implementation of Nakagami-m and Weibull variate generators on a single field-programmable gate array (FPGA) is presented. The hardware model first generates a correlated Rayleigh fading variate sequence and then transforms it into a sequence of Nakagami-m or Weibull fading variates. A biquad processor facilitates the compact implementation of a Rayleigh variate generator with arbitrary autocorrelation properties. A combination of logarithmic and linear domain segmentations along with piece-wise linear approximations is used to accurately implement the nonlinear numerical functions required to transform the correlated Rayleigh fading process into Nakagami-m or Weibull fading processes. When implemented on a Xilinx Virtex-5 5VSX240TFF1738-2 FPGA, the fading simulator uses only 1.6% of the configurable slices, 1.2% of the DSP48E modules and 3 block memories, while operating at 120 MHz, generating 120 million complex variates per second. The throughput can be increased up to 373 MHz with this FPGA if two separate clock sources are utilized.**

I. Introduction

Simulating radio propagation channels is a key step in the design and performance verification of wireless communication systems. Fading channel simulators are widely used at the baseband level for the early verification and characterization of wireless transceiver designs. While software simulation of fading channels is easier to develop and thus widely used, hardware-based simulators have been shown to provide several orders of magnitude of speed-up over software-based simulators [1], [2]. Speed is an especially important factor when many different fading scenarios must be simulated the wide variety of operating modes supported by recent wireless standards. While various hardware-based simulators for Rayleigh and Ricean fading channels have already been proposed using field-programmable gate arrays (FPGAs) [3], [4], hardware-based simulators of Nakagami and Weibull fading channels have received far less attention.

The Nakagami-m distribution is commonly used [5], [6] to model moderate to severe fading channels using different values of the parameter m [7]. The Nakagami fading parameter $m \geq 0.5$ determines the fading severity. When $m = 1$, Nakagami-m fading is identical to Rayleigh fading. Values of m in the intervals $[1/2, 1)$ and $(1, \infty)$ correspond to more severe and less severe fading than Rayleigh fading, respectively. It is well-known that the amplitude of a sum of $\xi > 1$ squared, independent Gaussian random variates (RVs), each with zero-mean and variance σ^2 (i.e., $n = \sqrt{c_1^2 + c_2^2 + \cdots + c_\xi^2}$), follows the Nakagami-$m$ distribution with $m = \xi/2$ and a second moment $\Omega = 2m\,\sigma^2$ [8]. Each Nakagami-m variate can thus be found by summing $2m$ independent Gaussian variates. However, this method is limited to integer and half-integer values of m, and it becomes computationally inefficient as m increases.

Other methods have been proposed in the literature for generating Nakagami-m fading variates. In [9], uncorrelated Nakagami-m samples are generated starting with any random number generation method. Autocorrelation is introduced among the random samples by sorting them according to the rank statistics of additional Rayleigh samples with the desired autocorrelation. In [10], a Nakagami-m fading signal with $m < 1$ is simulated using complex Gaussian processes and square-root-Beta random processes. In [11]–[13], Nakagami-m distributed samples are generated from Gamma-distributed samples with the Gamma-distributed samples having been correlated using either the Cholesky decomposition of the covariance matrix or Sim's method [14]. In [15]–[17] nonlinear transformations are proposed for mapping Rayleigh sequences into Nakagami-m sequences. In [18] and [19], two Nakagami-m fading simulators based on the sum-of-sinusoids (SOS) approach are described.

Many of these proposed approaches are not appropriate for an efficient hardware implementation of a parameterizable continuous-time Nakagami-m fading simulator. The main problems include overly large memory requirements [9], the block-based nature of the algorithm [9]–[13], the high computational complexity [9]–[13], [15]–[17], or the need for high-precision arithmetic [15]–[17]. Even though the SOS-based approach is more computationally-efficient than the other proposed schemes [18], [19], the approach does not have the flexibility to support different values of m and impact arbitrary time-correlation properties between fading samples.

In this brief, we propose a computationally-efficient technique for generating random variates from the Nakagami-m distribution. We utilize non-uniform domain segmentations, coefficient scaling and piece-wise linear function approximations to achieve a numerically accurate implementation. We then propose a compact and high-throughput hardware implementation of a parameterizable Nakagami-m fading channel simulator, which can produce arbitrary time-correlation properties between generated fading samples for various values of m. We utilize a custom library of fixed-point arithmetic and logical routines in MEX-C to allow fast bit-true simulation of the proposed Nakagami-m variate generator. This capability allows us to empirically minimize the wordlength of the variables and the size of the memory modules in the datapath of the variate generator. Due to the similarities between the Nakagami-m and the Weibull distributions, with only slight modifications (by reloading on-chip memories), the proposed approach is readily modified to generate Weibull variates [20],

A. Alimohammad is with the Department of Electrical and Computer Engineering, San Diego State University, San Diego, CA, 92182, U.S.A.

Saeed Fouladi Fard is with PMC-Sierra, Calgary, Alberta, Canada.

Bruce F. Cockburn is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, T6G 2V4, Canada

E-mail: aalimohammad@mail.sdsu.edu.

[21]. To the best of our knowledge, these are the first proposed hardware-based Nakagami-m and Weibull variate generators.

The rest of this brief is organized as follows. Section II briefly reviews the transformation-based method for generating variates with the Nakagami-m distribution. Then we describe our new design for the efficient generation of correlated Rayleigh and Nakagami-m variates. Section III presents the compact and high-throughput hardware implementation of the Nakagami-m variate generator. Section IV presents the hardware implementation costs and gives simulation results. Section V describes the Weibull variate generator design. Finally, Section VI makes some concluding remarks.

II. Proposed Nakagami-m Variate Generator

Transforming a uniformly-distributed random variate $u \in [0, 1)$ by an inverse cumulative distribution function (CDF) function $\mathrm{F}^{-1}(u)$ produces a RV having the CDF given by the function $\mathrm{F}(\cdot)$. Consequently, a uniform random variable u can be transformed into a Nakagami-m random variable n_N using the nonlinear transformation

$$n_N = \mathrm{F}_N^{-1}(u) \tag{1}$$

where F_N^{-1} is the inverse Nakagami-m CDF [22]. However, to simulate different fading scenarios, the uniform random variate u should be generated and imparted with specific correlation properties among the generated samples. To generate uniform RVs with a desired autocorrelation function (ACF), a transformation-based approach is proposed in [16]. In this model, first Rayleigh RVs with the desired ACF are generated. Note that the envelope of a sequence of zero-mean unit-variance complex Gaussian variates $c = c_i + jc_q$ (i.e., $r_R = (c_i^2 + c_q^2)^{1/2}$), where c_i and c_q are independent, is Rayleigh-distributed. A sequence of Rayleigh random variates r_R can be transformed into samples with a uniform distribution and the same ACF between samples using the inverse CDF transformation [23]

$$u_U = \mathrm{F}_R(r_R) = 1 - e^{-\frac{r_R^2}{2\sigma_c^2}} \tag{2}$$

where $\mathrm{F}_R(\cdot)$ is the CDF of Rayleigh RVs and σ_c^2 is the variance of c. Then the uniform RVs can be transformed into correlated Nakagami-m RVs using (1).

To generate the baseband complex Nakagami-m fading variates x_X, we use real-valued Nakagami-distributed random variables n_i and n_q as follows:

$$x_X = x_i + jx_q = n_i \cos(\theta) + jn_q \sin(\theta) \tag{3}$$

where $\theta = \arctan(c_q/c_i)$. The key module for generating Nakagami-m RVs using (3) is the calculation of the inverse Nakagami-m CDF $\mathrm{F}_N^{-1}(u)$. While there is no known closed-form expression for $\mathrm{F}_N^{-1}(u)$, except for the special case of $m = 1$, a useful rational polynomial approximation for $\mathrm{F}_N^{-1}(u)$ was proposed in [15], [16] as follows:

$$\mathrm{F}_N^{-1}(u) \approx \eta(u) + \frac{a_1\eta(u) + a_2\eta^2(u) + a_3\eta^3(u)}{1 + b_1\eta(u) + b_2\eta^2(u)} \tag{4}$$

where

$$\eta(u) = \left(\sqrt{\ln \frac{1}{1-u}}\right)^{\frac{1}{m}}. \tag{5}$$

For a given value of m, the five coefficients a_1, a_2, a_3, b_1 and b_2 are calculated to minimize the approximation error. Suitable coefficient values for different values of m are given in [15].

While a polynomial approximation is attractive for the software generation of Nakagami-m variates, an efficient hardware implementation of $\mathrm{F}_N^{-1}(u)$ using (4) and (5) is challenging. Such a design requires several additions and multiplications, two divisions, a square root, an $m-$th root, and a natural logarithm. In addition, generating each complex Nakagami-m RV using (3) requires two multiplications, one division, and evaluation of the $\sin$, $\cos$, and $\arctan$ functions. In addition to these computational requirements, our bit-true fixed-point simulations show that due to the large dynamic range of the intermediate variables, to attain the required accuracy, $\mathrm{F}_N^{-1}(u)$ needs to be implemented in floating-point arithmetic. However, floating-point hardware generally requires much more logic compared to fixed-point computations. While a fixed-point implementation can result in a more compact and efficient design than one using floating-point arithmetic, careless fixed-point implementations can suffer from excessive quantization noise. Moreover, the issues of truncation, rounding and overflow can render a fixed-point implementation inaccurate.

For an accurate and compact fixed-point Nakagami-m variate generator, we start by rewriting $\sin(\theta)$ and $\cos(\theta)$ as

$$\sin(\theta) = \frac{c_q}{\sqrt{c_i^2 + c_q^2}} \quad \text{and} \quad \cos(\theta) = \frac{c_i}{\sqrt{c_i^2 + c_q^2}}.$$

From (3), the complex Nakagami variates can be generated as

$$x_X = g(r_R^2) \times (c_i + jc_q)$$

where $g(r_R^2)$ is the transfer function defined as

$$g(r_R^2) = \frac{\mathrm{F}_N^{-1}(1 - e^{-\frac{r_R^2}{2\sigma_c^2}})}{\sqrt{r_R^2}}. \tag{6}$$

A high-level block diagram of our proposed Nakagami-m variate generator appears in Fig. 1. First the Rayleigh Variate Generator block generates a sequence of zero-mean unit-variance complex Gaussian random variates $c = c_i + j\, c_q$ and the squared envelope r_R^2 of the corresponding Rayleigh process is calculated as $r_R^2 = c_i^2 + c_q^2$. Then the transfer function $g(r_R^2)$ is approximated, and the in-phase and quadrature components of the Nakagami-m variates are found by multiplying $g(r_R^2)$ by c_i and c_q, respectively. In the following two subsections, we describe the details of the design procedure for the Rayleigh variate generator and our computationally-efficient approximation of $g(r_R^2)$.

A. Time-Correlated Rayleigh Variate Generator

The samples generated by the Rayleigh Variate Generator must be properly correlated to model the effects of Rayleigh fading. To generate a sequence of Rayleigh fading samples with a suitable ACF between generated samples of c_i (as

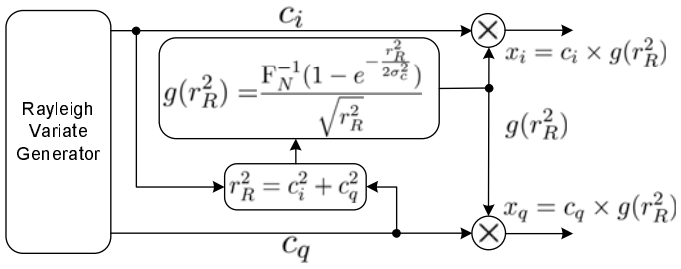

Fig. 1. Block diagram of the complex Nakagami-m variate generator.

well as c_q), statistically independent Gaussian streams of c_i and c_q can be passed through a spectrum shaping filter (SSF) that has a magnitude response equal to the square root of the magnitude of the desired spectrum [22]. The SSF allows us to parameterize the time-correlation between the generated Rayleigh variates to model different fading scenarios.

The SSF can be designed using a finite impulse response (FIR) or an infinite impulse response (IIR) filter. Since IIR filters are typically much smaller than their FIR counterparts for the same filter specifications [24], we approximate the desired magnitude response of the SSF with an IIR filter of order $2K$ [25]. The magnitude response of the SSF can be written as the magnitude response of $K \geq 1$ cascaded second-order canonic sections (biquads) as follows

$$\mathrm{H}(e^{j\omega}) = \prod_{k=1}^{K} \left\{ \lambda_k \times \frac{1 + b_{1,k}\, e^{-j\omega} + b_{2,k}\, e^{-j2\omega}}{1 + a_{1,k}\, e^{-j\omega} + a_{2,k}\, e^{-j2\omega}} \right\} \quad (7)$$

where $b_{1,k}$, $b_{2,k}$, $a_{1,k}$ and $a_{2,k}$ denote the coefficients and λ_k denotes the real-valued scaling factor of the k-th biquad.

Without loss of generality, we design the SSF for an omni-directional receiving antenna and assume two-dimensional isotropic scattering. The ACF associated with either c_i or c_q is expressed as $\mathrm{R}_{ii}(\tau) = \mathrm{R}_{qq}(\tau) = \mathcal{J}_o(2\pi f_D \tau)$ [26], where τ denotes the time lag, f_D is the maximum Doppler frequency, and $\mathcal{J}_o(\cdot)$ is the zeroth-order Bessel function of the first kind [27]. The cross-correlation function (CCF) between c_i and c_q is $\mathrm{R}_{iq}(\tau) = \mathrm{R}_{qi}(\tau) = 0$ (i.e., c_i and c_q are independent stochastic processes). To generate a sequence of Rayleigh variates with the above ACF, a stream of independent Gaussian samples can be passed through a SSF that has a magnitude response equal to the square root of the magnitude of the power spectral density (PSD) function [28]:

$$S_c(f) = \begin{cases} \frac{1}{\pi\sqrt{f_D^2 - f^2}} & \text{if } |f| < f_D, \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

Given the desired frequency response $S_c^{1/2}(f)$, we can design the SSF using the MATLAB function `iirlpnorm` [29], which calculates the optimal filter coefficients and scaling factors by minimizing the p-norm. Then the filter coefficients and scaling factors can be quantized with appropriate fixed-point representations while keeping the SSF stable.

It is important to note that for a typical wireless communication scenario, the Doppler frequency f_D is significantly lower than the signal sample rate F_s. Since the channel changes only slowly compared to the sample rate, we can design the Rayleigh fading simulator (and subsequently, the Nakagami-m fading simulator) at a much lower sample rate with no significant loss in accuracy. This observation allows us to utilize time-multiplexed datapaths, which share functional and storage resources for generating the Nakagami-m variates. The resulting low-rate fading signal can then be interpolated to achieve the desired output sample rate.

We design the SSF at a sampling frequency F_1, where $4f_D < F_1 \leq 8f_D$. Choosing F_1 in this range satisfies the minimum Nyquist rate while keeping the computational complexity low [30]. The SSF output is then up-sampled I_1 times and passed to an elliptic interpolation lowpass filter (EILP), which generates $F_2 = I_1 \times F_1$ samples per second. The samples are further up-sampled and interpolated using a linear interpolator to obtain the desired output sample rate F_s, as will be explained in the next section. Fig. 2 shows both the desired reference with 60 dB target attenuation in the stopband and the achieved frequency responses of the designed fixed-point SSF with four cascaded biquads. As this figure shows, the designed filter accurately produces the desired response within the passband. In the stopband, the designed filter provides more than 55 dB of attenuation.

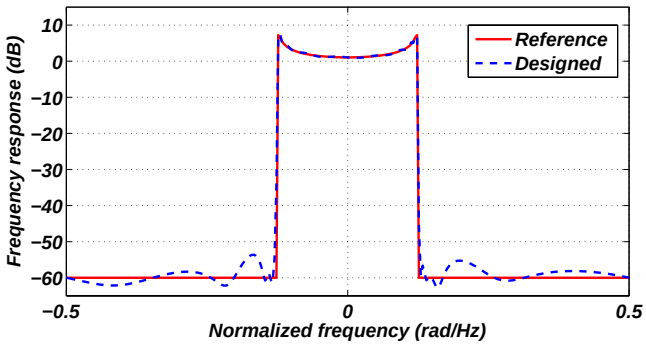

Fig. 2. Frequency response of the PSD in (8) and the designed SSF with $f_D/F_1 = 0.125$.

B. An Efficient Approximation of $g(r_R^2)$

Accurate approximation of $g(r_R^2)$ is crucial for the faithful simulation of Nakagami-m fading channels. Fig. 3 plots $g(r_R^2)$ as a function of $r_R^2 \in [2^{-15}, 2^5)$ for six different values of m on a log-log scale. In this figure and in the rest of this section we assume that the average fading power is $\Omega = 1$. The simulation results in Fig. 3 show that for $m \in [0.5, 25]$ and $r_R^2 \in [2^{-15}, 2^5)$, $g(r_R^2)$ varies over four orders of magnitude. Moreover, $g(r_R^2)$ tends to change faster for small values of r_R^2. For $\sigma_c^2 = 1$, since $\Pr(r_R^2 > 2^5)$ is less than 1.125×10^{-7}, and $\Pr(r_R^2 < 2^{-15})$ is less than 1.526×10^{-5}, we focus on evaluating $g(r_R^2)$ over the interval $[2^{-15}, 2^5)$ without significantly affecting the output statistics (as we will show later). Our fixed-point simulation results show that for a sufficiently accurate representation of the transfer function $g(r_R^2)$, for the lower part of the full domain $r_R^2 \in [2^{-15}, 2^5)$ we require a minimum resolution of 2^{-15}. Such a fine resolution, however, is not required for the upper part of this domain since $g(r_R^2)$ changes slowly in this region. On the other hand, as shown

in Fig. 3, since $g(r_R^2)$ has a wide range of 2^{-7} to 2^7, at least 14 bits are required for the fixed-point representation of this function. One could precompute and store values of $g(r_R^2)$ in a look-up table (LUT). The problem with this approach is that the required size of the LUT for a practical range would be prohibitively large (i.e., $2^{5+15} \times 14 = 14$ megabits).

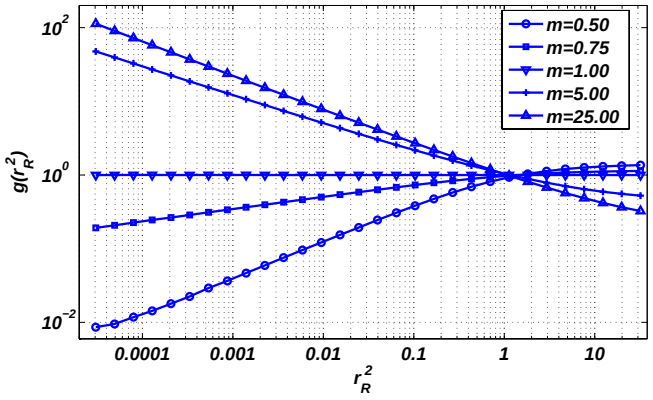

Fig. 3. Log-log plot of $g(r_R^2)$ for $\Omega = 1$ and five different values of m.

An alternative and much more efficient solution is to calculate $g(r_R^2)$ using variable domain segmentation with linear approximations over each segment. In this approach, the domain $[0, 2^5]$ is divided into a number of small segments and the value of $g(r_R^2)$ over each segment is approximated using standard linear regression [31]. However, note that $g(r_R^2)$ changes faster for the smaller values of r_R^2. Therefore, more accurate approximations (i.e., smaller segments) are required for the lower part of the domain. On the other hand, since $g(r_R^2)$ changes slowly for larger values of r_R^2, larger segments can be used in the upper part of the domain. Thus for an accurate approximation of $g(r_R^2)$ that also reduces the size of the LUT, we propose a hybrid logarithmic-linear segmentation [3], [32]. As shown in Fig. 4, the range $(0, 2^\alpha]$ is divided into l logarithmic (i.e., power of 2) segments. Each of these segments is further divided into 2^ζ uniform sub-segments. This segmentation is particularly convenient for hardware implementation as the logarithmic segment for each sample can be determined using a leading-1 detector circuit. Moreover, the next ζ bits after the leading 1s can be used to address the linear sub-segment in the memory. The remaining bits can be used as the argument to the linear function (i.e., $\delta - \delta_0$).

Based on empirical evaluations of the resulting fixed-point accuracy, we divided the domain $[2^{-15}, 2^5]$ into 15 logarithmic segments. Each of the logarithmic segments is further divided into 64 uniform sub-segments. The domain $[0, 2^{-15})$ is also divided into 64 uniform segments, which makes the total number of segments $(1 + 15) \times 64 = 1024$. Then $g(r_R^2)$ over each segment $\delta \in [\delta_0, \delta_1)$ is approximated as $\hat{g}(\delta) \approx a_{\delta_0}(\delta - \delta_0) + b_{\delta_0}$. The number of bits required to store $\{a\}$ and $\{b\}$ can be further reduced by approximating $\hat{g}(\delta)$ as $\hat{g}(\delta) \approx 2^{f_{\delta_0}} \times [\hat{a}_{\delta_0}(\delta - \delta_0) + \hat{b}_{\delta_0}]$, where $\hat{a}_{\delta_0} = 2^{-f_{\delta_0}} \times a_{\delta_0}$ and $\hat{b}_{\delta_0} = 2^{-f_{\delta_0}} \times b_{\delta_0}$. Our bit-true simulations, as shown in Section IV, confirm that if the values of $\{\hat{a}\}$, $\{\hat{b}\}$ are represented in $s18.17$ format, i.e., signed 2's-complement 18-bit fixed-point values with 17-bit fractions representing the signed domain $[-0.5, 0.5)$, the approximation error of $g(r_R^2)$ is relatively low and the statistics of generated samples closely follow the reference statistics. The choice of 1024 segments and a 18-bit representation for $\{\hat{a}\}$, $\{\hat{b}\}$ allows us to store the values of coefficients for each segment in an 18-Kb on-chip read-only memory. The values of f were represented in $s5.0$ format (i.e., signed integers within $[-16, 15]$).

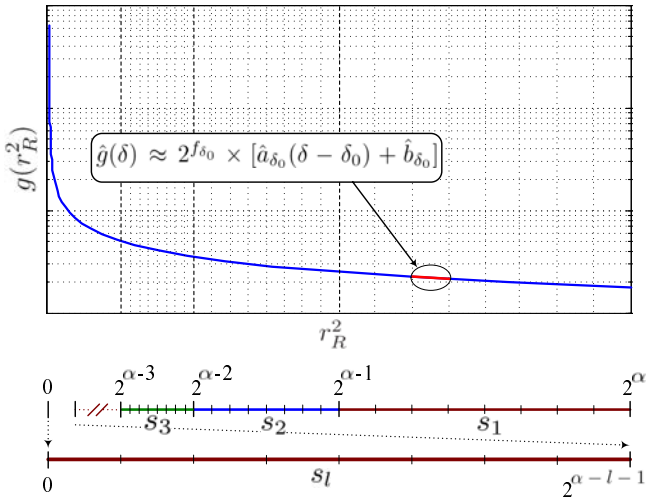

Fig. 4. Hybrid domain segmentation for the transfer function $g(r_R^2)$.

III. Efficient Hardware Implementation of Nakagami-m Variate Generator

Fig. 5 shows the datapath of our proposed Nakagami-m variate generator. This datapath supports different values of the Nakagami parameter m with no impact on the computational complexity. The datapath details are described in the following three sub-sections.

A. Rayleigh Fading Variate Generator Datapath

To generate correlated Rayleigh fading samples, we first generate zero-mean unit-variance independent Gaussian variates c_i and c_q using the compact and high-quality Gaussian variate generator from [32]. The generated samples, c_i and c_q, which are in $s16.11$ format, are passed through the SSF designed in Section II.A. For a compact spectrum shaping IIR filter implementation, we use an optimized fixed-point biquad processor with the architecture shown in Fig. 6. This processor is capable of implementing eight time-multiplexed IIR filters, where the maximum order of each IIR filter is 16 (i.e., eight biquads per filter). The main datapath element is a multiply-accumulator based on a 36×32 multiplier that multiplies the in-phase and quadrature data components by real-valued coefficients. Memory modules RAM a_1, a_2, b_1, b_2 and λ are implemented using distributed memories of depth 64 and they store 32-bit SSF real-valued coefficients and scaling factors in $s32.29$ format. The memories RAM m_1, m_2 and d are implemented using distributed memories of depth 64×2 (for both the in-phase and the quadrature components) in $s36.24$ format. Memories RAM m_1 and m_2 store the values of two

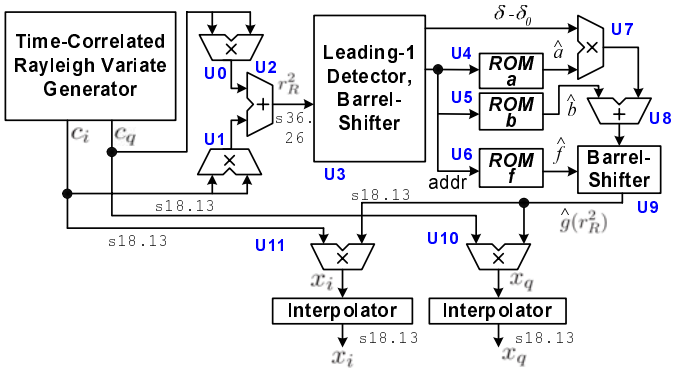

Fig. 5. Datapath of the transformation-based Nakagami-m variate generator.

internal registers for each biquad while RAM d is used to store the output value of each biquad. The Control Unit sequences the operation of a cascade of biquad processors. To control the dataflow through the biquad processor, two flags are assigned to every thread of cascaded biquads (i.e., for each individual filter labeled in-i and out-j in Fig. 6). For example, for the i-th thread (filter calculation), if in-i is high, then input data is ready to be read. Also, if out-i is high, then data can be written to the next cascaded biquad. For each thread, the biquad processor keeps re-evaluating the biquads until either of the input or output flags is de-asserted. To prevent the overwriting of unprocessed data, the biquads in each thread are scheduled to be executed from the last biquad to the first.

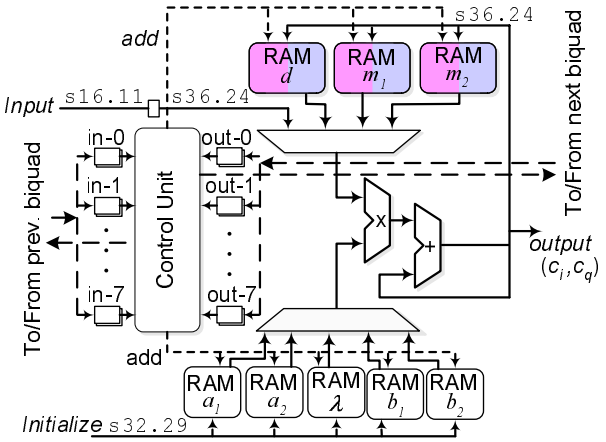

Fig. 6. Architecture of the biquad processor.

B. Nakagami-m Fading Variate Generator Datapath

After the biquad processor generates variates c_i and c_q with the desired ACF, they are passed through modules `U0`, `U1` and `U2` in Fig. 5 to generate r_R^2. Then the calculated r_R^2 is passed to the Leading-1 Detector and Barrel-Shifter `U3` to find the address `addr` of the current hybrid segment, the uniform segment number therein, and the argument $\delta - \delta_0$. Current segment address `addr` is then used to read the coefficient values $\{\hat{a}\}$, $\{\hat{b}\}$, and scaling factors $\{f\}$ from the corresponding read-only memories (ROMs). Note that different sets of coefficients $\{\hat{a}\}$, $\{\hat{b}\}$, and $\{f\}$ can be readily computed off-chip and loaded onto the corresponding memory blocks for different values of m. Then the value of $g(r_R^2)$ is approximated by arithmetic blocks `U7`, `U8` and `U9` as $\hat{g}(r_R^2) \approx 2^f \times [\hat{a}(\delta - \delta_0) + \hat{b}]$. The quadrature components of the fading samples are then calculated as $x_i = c_i \times \hat{g}(r_R^2)$ and $x_q = c_q \times \hat{g}(r_R^2)$ in `U10` and `U11`, respectively.

C. Interpolation Datapath

In the last step, fading variates generated at $\hat{F}_s$ samples per second are oversampled and interpolated I times to provide samples at the target sample rate $F_s = I \times \hat{F}_s$. To simplify the SSF hardware, we exploit power-of-2 interpolation factors. The linear interpolator requires the discrete difference between two consequent low-frequency samples $x[mI]$ and $x[(m+1)I]$ to generate the interpolated fading samples $x[mI+i]$, where $i = 0, 1, \cdots, I-1$, as follows:

$$x[mI+i] = \frac{\big(x[(m+1)I] - x[mI]\big)i}{I} + x[mI]. \qquad (9)$$

To avoid multiplication and division, we accumulate a running sum and choose I to be a power of two. Thus

$$x[mI+i] = \sum_{j=0}^{i} \frac{x[(m+1)I] - x[mI]}{I} + x[mI] \qquad (10)$$

and the interpolator can be implemented simply as shown in Fig. 7. The interpolator contains one 24-bit accumulator and one register that holds the value of the input signal for an interval of I samples.

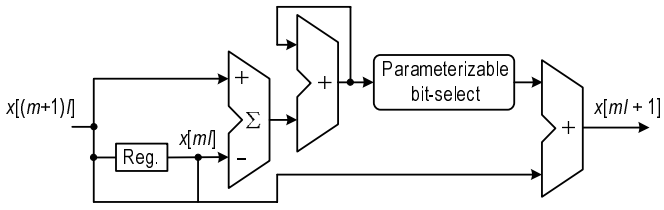

Fig. 7. Interpolator structure.

IV. Implementation and Simulation Results

We implemented the new Nakagami-m fading variate generator in Fig. 5 using device-independent Verilog modules on a Xilinx Virtex-5 5VSX240TFF1738-2 FPGA [33]. For a simulation trial, the parameter values along with the associated contents of memory blocks, are loaded into dedicated registers and block memories. Table I summarizes the implementation costs of the key modules.

TABLE I
Hardware Costs of the Major Modules

Design Parameters	GNG	Biquad Processor	Interpolator	Overall System
# of slices	145	312	62	631
# of DSP48Es	2	6	–	13
# of BRAMs	1	–	–	3
Clock freq. (MHz)	367	120	373	120

The Nakagami-m fading channel simulator requires only 1.6% of the configurable slices, 1.2% of the DSP48E modules, three of the 36 Kbit on-chip block memories (BRAMs), while operating at 120 MHz. Note that the Rayleigh-to-Nakagami variate transformation datapath in Fig. 5 requires 100 configurable slices, five DSP48E modules, and two block memories. Note also that when this datapath is implemented with two clock sources, say one for the fading sample generator and one for the interpolator, the maximum output rate is determined by the clock frequency of the interpolator. With the two-clock configuration, the sample generation rate can be increased up to 373 million complex Nakagami-m fading samples per second using the same FPGA.

To verify the accuracy of our simulator, we performed different simulations using the fixed-point and bit-true models of our proposed hardware simulator and compared the statistical properties of the generated samples against the theoretical references. Fig. 8 shows the relative approximation error of $g(r_R^2)$ (i.e., $e_g(r_R^2) = |1 - \hat{g}(r_R^2)/g(r_R^2)|$) for $m = 10$. The relatively small approximation error shows that the variable-resolution domain segmentation and piece-wise linear approximation can accurately produce the ideal transfer function $g(r_R^2)$. Note that the relative approximation error increases for the largest values of r_R^2. Note also that the probability of large values of r_R^2 is small and that $g(r_R^2)$ goes to zero for such values. However, as we will show, this approximation error does not have a significant impact on the distribution of the generated Nakagami-m samples. Note that the case of $m = 1$ (i.e., Rayleigh fading) has the minimum approximation error error as no envelope transformation is required. As the Nakagami fading parameter m increasingly differs from unity, the greater the approximation error.

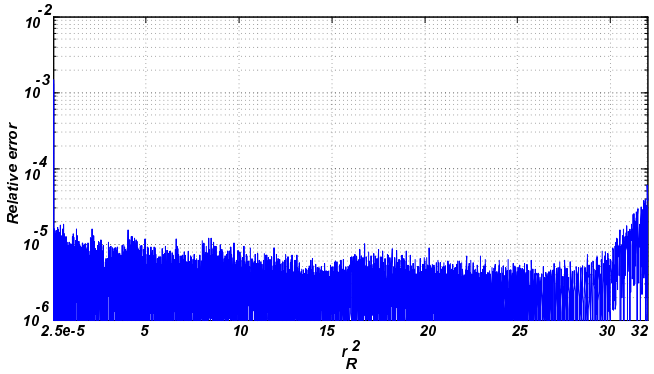

Fig. 8. Relative approximation error of transfer function $g(r_R^2)$ for $m = 10$.

To generate the Nakagami-m fading samples, we utilize the spectrum-shaping PSD function in (8). As an example we will assume a Doppler frequency $f_D = 100$ Hz and a sample rate $F_s = 10$ KHz. Fig. 9 plots the PDF of 10 million generated Nakagami-m samples for eight different values of m. The theoretical functions given by

$$f_N(n) = \frac{2m^m n_N^{2m-1}}{\Gamma(m)\Omega^m} \exp(-\frac{m}{\Omega} n_N^2), \quad n_N \geq 0 \qquad (11)$$

are plotted as well for reference, where $\Gamma(\cdot)$ is the gamma function and Ω is the average fading power of the sequence of fading variates [7]. This figure shows that the PDF of the generated samples closely matches the reference Nakagami-m PDF for the eight m values, which confirms the statistical accuracy of the generated samples. Fig. 10 compares

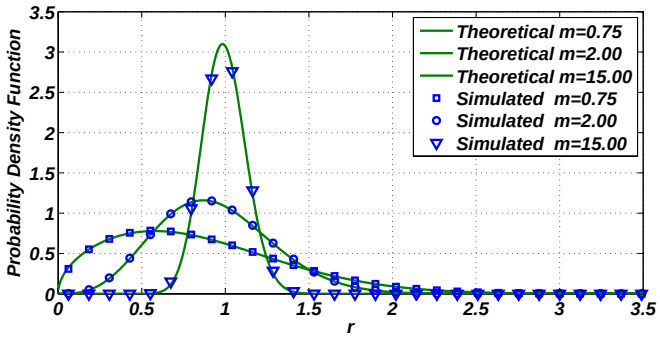

Fig. 9. Comparison between the reference Nakagami-m PDF and the measured PDF of generated samples for eight different values of m.

the autocorrelation of the generated Nakagami-m samples, for six different values of m, and the reference function. This figure also shows a close match between the measured autocorrelation and the reference values. The autocorrelation of the amplitude of the envelope is also widely used to assess the quality of a fading channel simulator. The normalized ACF of the envelope of the generated Nakagami-m samples and the reference values given by [34, eq. 26]

$$\mathrm{R}(\tau)_{ACE} = \frac{{}_2\mathrm{F}_1\left(-\frac{1}{2}, -\frac{1}{2}; m, |\mathcal{J}_o(2\pi f_D \tau)|^2\right)}{\left[\frac{\Gamma(m)}{\Gamma(m+1/2)}\right]^2 \left(\frac{m}{\Omega}\right)} \qquad (12)$$

are compared in Fig. 11, where ${}_2\mathrm{F}_1(\cdot,\cdot;\cdot,\cdot)$ denotes the Gaussian hypergeometric function [27]. Fig. 11 confirms a close match between the generated envelope autocorrelation and the reference functions from equation (12), which further verifies the accuracy of our implementation. Moreover, Fig. 12 shows

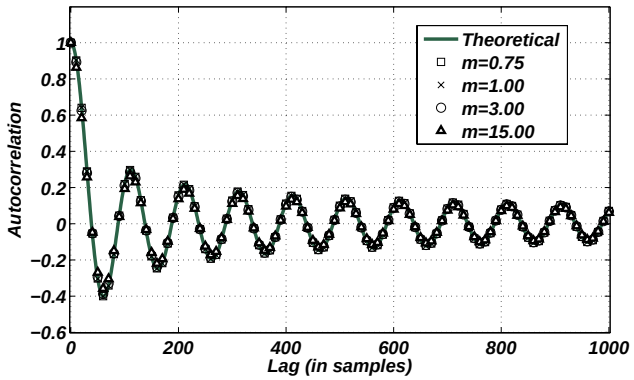

Fig. 10. Comparison between the reference autocorrelation function and that of the generated samples for six different values of m.

the normalized level crossing rate (LCR) of the amplitude of the generated complex Nakagami fading samples. The LCR is the rate at which the envelope crosses a specified level with positive slope. The LCR provides important information concerning the statistics of burst errors in wireless communication

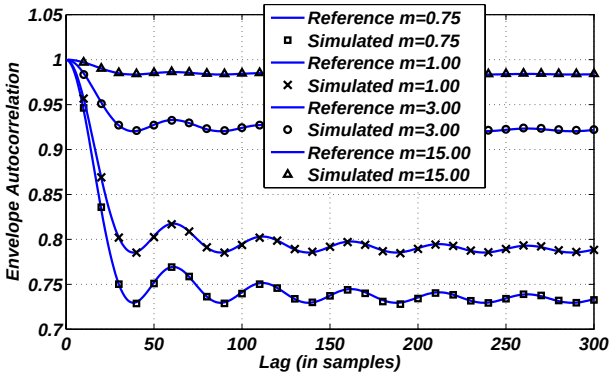

Fig. 11. The reference normalized envelope autocorrelation function and that of the generated samples for six different values of m.

systems. In Fig. 12 the theoretical and simulated LCR plots are normalized to $f_D T_s$. The LCR of the Nakagami fading envelope is given by [35]

$$L_{|R|}(\lambda) = \sqrt{2\pi} f_D T_s \times \frac{m^{m-1/2}\lambda^{2m-1}}{\Gamma(m)} \times \exp(-m\lambda^2). \quad (13)$$

Fig. 12 confirms agreement between the theoretical and fixed-point simulation results for two different values of m.

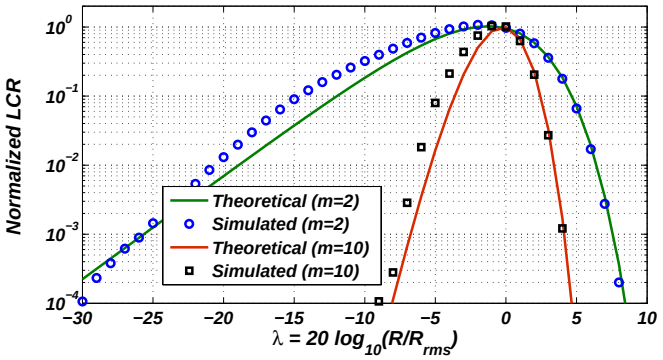

Fig. 12. Comparison between the reference Nakagami-m LCR and the measured LCR of generated samples for two different values of m.

V. Weibull Fading Variate Generator

Our Nakagami-m fading simulator can be readily modified to simulate Weibull fading channels. The CDF of the Weibull distribution can be written as [21]

$$\mathrm{F}_W(r) = 1 - \exp\left\{-\left(\frac{r}{\lambda}\right)^{\beta}\right\}, \quad r \geq 0 \quad (14)$$

where $\lambda > 0$ is the scale parameter of the distribution and β is the shape parameter that controls the fading severity. The transformation function for converting Rayleigh samples into Weibull samples can be written as

$$g_W(r_R^2) = \frac{\mathrm{F}_W^{-1}\left(1 - e^{-\frac{r_R^2}{2\sigma_c^2}}\right)}{\sqrt{r_R^2}} = \frac{\lambda}{\sqrt{r_R^2}} \times \left(\frac{r_R^2}{2\sigma_c^2}\right)^{\frac{1}{\beta}}. \quad (15)$$

The architecture of the Weibull variate generator is similar to that of Nakagami-m, where the transfer function $g_W(r_R^2)$ is used to generate Weibull samples from Rayleigh-distributed random variates instead of $g(r_R^2)$. Algorithm 1 gives the steps of the Weibull variate generator. Note that the same datapath in Fig. 5 can be used to generate Weibull fading samples; however, the contents of ROM a, ROM b and ROM f must be replaced with the appropriate values obtained from a suitable segmentation and linear approximation of $g_W(r_R^2)$.

Algorithm 1 Weibull variate generator procedure

Step 1: Polynomial approximation of $g_W(r_R^2)$ using a hybrid logarithmic-uniform segmentation.
1. The domain $[0, 2^5]$ is segmented into 16 logarithmic segments;
2. Divide each of the logarithmic segments into 64 uniform segments;
3. Approximate $g_W(r_R^2)$ over each segment $\delta \in [\delta_0, \delta_1)$ as $\hat{g}_W(\delta) \approx a_{\delta_0}(\delta - \delta_0) + b_{\delta_0}$;
4. Reduce the number of bits required to store $\{a\}$ and $\{b\}$ by approximating $\hat{g}_W(\delta)$ as $\hat{g}_W(\delta) \approx 2^{f_{\delta_0}} \times [\hat{a}_{\delta_0}(\delta - \delta_0) + \hat{b}_{\delta_0}]$, where $\hat{a}_{\delta_0} = 2^{-f_{\delta_0}} \times a_{\delta_0}$ and $\hat{b}_{\delta_0} = 2^{-f_{\delta_0}} \times b_{\delta_0}$;
5. Store the values of $\{\hat{a}\}$, $\{\hat{b}\}$ and $\{f\}$ for each segment in a ROM;

Step 2: Generate a correlated Rayleigh fading process c as described in Section II.A.
Step 3: Calculate the squared envelope $r_R^2 = c_i^2 + c_q^2$.
Step 4: Read the values of $\{\hat{a}\}$, $\{\hat{b}\}$ and $\{f\}$ from three ROMs.
Step 5: Calculate $\hat{g}_W(r_R^2) \approx 2^f \times [\hat{a}(\delta - \delta_0) + \hat{b}]$.
Step 6: Calculate the quadrature components of the Weibull samples as $w_i = c_i \times \hat{g}_W(r_R^2)$ and $w_q = c_q \times \hat{g}_W(r_R^2)$.
Step 7: Oversample and interpolate the generated fading samples I times.

To verify the statistical accuracy of our hardware model, we generated 10 million Weibull samples with our variate generator. Fig. 13 compares the PDF of the generated Weibull samples with the reference PDF given by

$$f_W(r) = \left(\frac{\beta}{\lambda}\right)\left(\frac{r}{\lambda}\right)^{\beta-1} \times \exp\left\{-\left(\frac{r}{\lambda}\right)^{\beta}\right\}, \quad r \geq 0 \quad (16)$$

for four different values of β [22]. Similar to the Nakagami-m variate generator, we used 1024 segments and represented the values of $\{\hat{a}\}$, $\{\hat{b}\}$ and $\{f\}$ for each segment in $s18.17$, $s18.17$ and $s5.0$ fixed-point formats, respectively. In this simulation, the average power of the Weibull samples was set to $\Omega = 1$. This figure confirms a close match between the generated PDFs and the reference PDF.

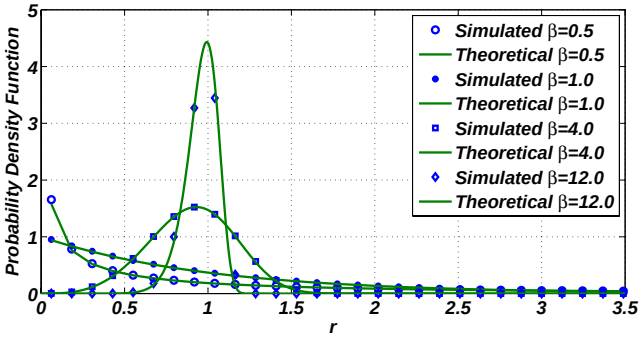

Fig. 13. Comparison between the reference Weibull PDF and the PDF of the generated samples for four different values of β.

VI. Conclusions

A compact and high-throughput hardware-based variate generator for both the Nakagami-m and Weibull distributions was proposed. The new simulator exploits an efficient

implementation of two consecutive nonlinear transformations that uses hybrid domain segmentation and piece-wise linear approximation. We also proposed a compact processor for the efficient implementation of an IIR filter that is required to generate Rayleigh variates with independently-defined auto-correlation properties among the generated samples. To reduce the hardware complexity, the Nakagami-m variate generator operates at a lower sampling rate than the target sampling frequency. A parameterizable linear interpolator up-samples and filters the generated Nakagami-m (or Weibull) variates to achieve the target output rate. Using a custom library of fixed-point arithmetic and logical routines, the new simulators were then further optimized by minimizing the wordlength of the variables and the size of the on-chip memories. To the best of our knowledge, these are the first hardware-based Nakagami-m and Weibull variate generators. The accuracy of the simulator was verified by generating fixed-point fading samples and comparing their important statistical properties with those of the two ideal distributions.

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] A. Alimohammad, S. F. Fard, and B. F. Cockburn, "FPGA-based accelerator for the verification of leading-edge wireless systems," in *IEEE Int. Design Automation Conference*, 2009, pp. 844–847.

[3] ——, "Hardware implementation of Rayleigh and Ricean variate generators," *IEEE Trans. VLSI Syst.*, vol. 99, no. 4, pp. 1–5, 2010.

[4] 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*, 2007, pp. 2602–2605.

[5] T. Aulin, "Characteristics of a digital mobile radio channel," *IEEE Trans. Veh. Technol.*, vol. 30, no. 2, pp. 45–53, May 1981.

[6] A. Mehrnia and H. Hashemi, "Mobile satellite propagation channel. Part 1 - A comparative evaluation of current models," in *Proc. of the Veh. Technol. Conf.*, vol. 5, 1999, pp. 2775–2779.

[7] M. Nakagami, "The m-distribution: A general formula of intensity distribution of rapid fading," in *Statistical Methods in Radio Wave Propagation*, 1960, pp. 3–36.

[8] J. G. Proakis, *Digital Communications*. McGraw-Hill, 2001.

[9] J. C. S. S. Filho, M. D. Yacoub, and G. Fraidenraich, "A simple accurate method for generating autocorrelated Nakagami-m envelope sequences," *IEEE Commun. Lett.*, vol. 11, no. 3, pp. 231–233, Mar. 2007.

[10] K. W. Yip and T. S. Ng, "A simulation model for Nakagami-m fading channels, $m < 1$," *IEEE Trans. Commun.*, vol. 48, no. 2, pp. 214–221, Feb. 2000.

[11] Q. T. Zhang, "A decomposition technique for efficient generation of correlated Nakagami fading channels," *IEEE J. Sel. Areas Commun.*, vol. 18, pp. 2385–2392, Nov. 2000.

[12] Z. Song, K. Zhang, and Y. L. Guan, "Generating correlated Nakagami fading signals with arbitrary correlation and fading parameters," in *Proc. of the IEEE Intl. Conf. on Commun.*, vol. 3, 2002, pp. 1363–1367.

[13] K. Zhang, Z. Song, and Y. L. Guan, "Simulation of Nakagami fading channels with arbitrary cross-correlation and fading parameters," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1463–1468, Sep. 2004.

[14] C. H. Sim, "Generation of Poisson and Gamma random vectors with given marginal and covariance matrix," *Journal of Statistical Computation and Simulation*, vol. 47, no. 1-2, pp. 1–10, 1993.

[15] N. C. Beaulieu and C. Cheng, "An efficient procedure for Nakagami-m fading simulation," in *Proc. of the IEEE Global Telecommunications Conference*, vol. 6, 2001, pp. 3336–3342.

[16] ——, "Efficient Nakagami-m fading channel simulation," *IEEE Trans. Veh. Technol.*, vol. 54, pp. 413–424, 2005.

[17] E. Pajala, T. Isotalo, A. Lakhzouri, E. S. Lohan and M. Renfors, "An improved simulation model for Nakagami-m fading channels for satellite positioning applications," in *Proc. of the 3rd Workshop on Positioning, Navigation, and Commun.*, 2006, pp. 81–89.

[18] T. M. Wu and S.-Y. Tzeng, "Sum-of-sinusoids-based simulator for Nakagami-m fading channels," in *In Proc. of the IEEE Veh. Technol. Conf.*, 2003, pp. 158–162.

[19] D. D. Patil, "On the simulation of Nakagami-m fading channels using sum-of-sinusoids method," Master's thesis, University of Missouri-Columbia, December 2006.

[20] M. A. Taneda, J. Takada, and K. Araki, "A new approach to fading: Weibull model," in *Proc. of the IEEE Intl. Symp. on Personal Indoor Mobile Radio Communictions*, 1999, pp. 711–715.

[21] N. C. Sagias and G. K. Karagiannidis, "Gaussian class multivariate Weibull distributions: Theory and applications in fading channels," *IEEE Trans. Inform. Theory*, vol. 51, no. 10, pp. 3608–3619, Oct. 2005.

[22] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, 4th ed. New York, NY: McGraw-Hill, 2002.

[23] G. L. Stüber, *Principles of Mobile Communication*, 2nd ed. New York: Kluwer Academic Publishers, 2001.

[24] A. Alimohammad and B. F. Cockburn, "Modeling and hardware implementation aspects of fading channel simulators," *IEEE Trans. on Veh. Technol.*, vol. 57, no. 4, pp. 2055–2069, 2008.

[25] K. Steiglitz, "Computer-aided design of recursive digital filters," *IEEE Trans. Audio and Electroacoust.*, vol. 18, no. 2, pp. 123–129, June 1970.

[26] W. C. Jakes, *Microwave Mobile Communications*. Piscataway, NJ: Wiley-IEEE Press, 1994.

[27] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products*, 5th ed. Academic Press Inc., 1995.

[28] P. Bello, "Characterization of randomly time-variant linear channels," *IEEE Trans. Commun.*, vol. 11, no. 4, pp. 360–393, 1963.

[29] *Filter Design Toolbox For Use With Matlab, User's Guide*, The Mathworks, 2005.

[30] C. S. S. Fouladi Fard, A. Alimohammad and B. F. Cockburn, "A single FPGA filter-based multipath fading emulator," in *Proc. of Globecom*, 2009, pp. 1–5.

[31] J.-M. Muller, *Elementary Functions. Algorithms and Implementation*. Birkhauser, 1997.

[32] A. Alimohammad, S. Fouladi Fard, B. F. Cockburn, and C. Schlegel, "A compact and accurate Gaussian variate generator," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 5, pp. 517–527, May 2008.

[33] *Xilinx UG190 Virtex-5 FPGA User Guide*, Xilinx Inc., San Jose, California, USA, November 2009.

[34] C. D. Iskander and P. T. Mathiopoulos, "Analytical envelope correlation and spectrum of maximal-ratio combined fading signals," in *IEEE Pacific Rim Conf.*, 2003, pp. 446–449.

[35] A. Abdi, K. Wills, H. Barger, S. Alouini, and M. Kaveh, "Comparison of level crossing rate and average fade duration of Rayleigh, Rice, and Nakagami fading models with mobile channel data," in *Proccedings of the IEEE Vehic Tech. Conf.*, Boston, MA, 2000, pp. 1850–1857.