

# Modeling and Hardware Implementation Aspects of Fading Channel Simulators

Amirhossein Alimohammad, *Member, IEEE*, Bruce F. Cockburn, *Member, IEEE*

**Abstract**—A channel simulator is an essential component in the development and accurate performance evaluation of wireless systems. Two major approaches have been widely used to produce statistically accurate fading variates, namely, shaping the flat spectrum of Gaussian variates using digital filters and sum-of-sinusoids (SOS) based methods. Efficient design and implementation techniques for these schemes are of particular importance in the design and verification of wireless systems with a relatively large number of channels such as ad-hoc networks. This article considers modeling and implementation aspects of fading channel simulators. First, we present a novel computationally-efficient implementation of a filter-based fading channel simulator on a single field-programmable gate array (FPGA) device. The new technique significantly alleviates the challenges of real-world testing of communication systems by introducing a fast and area-efficient FPGA implementation of the fading channel. Our fixed-point implementation of a Rayleigh fading channel simulator on an FPGA utilizes only 3% of the configurable slices, 10% of the dedicated multipliers and, 1% of the available memories on a Xilinx Virtex-II Pro XC2VP100-6 FPGA, while the simulator operates 12.5 times faster than the example sample rate. Then we describe a compact implementation of the SOS-based fading simulator which uses only 1% of the configurable slices and 1% of the available memories on the same FPGA device while generating over 200 million complex Rayleigh fading variates per second.

**Index Terms**—Fading channel simulator, Rayleigh distribution, Jakes' power spectral density, spectral shaping filter, hardware channel simulator.

## I. MOTIVATION

Wireless communication systems must be designed to operate over radio channels for a wide variety of environmental and weather conditions. Numerous real-time test cases must be applied to a new communication system before shipping the products to market. While it is possible to build prototypes of a proposed system and then field test them in different locations, such an approach is very expensive and will not provide useful feedback in the early stages of design when a relatively large number of candidate designs must be explored. Moreover, propagation conditions in the field are almost impossible to repeat for the comparative analysis of simulation results. A more practical approach is to create appropriate simulation models for the channel and then evaluate designs based on these models. The channel simulator allows the performance evaluation of mobile communication systems under controlled and repeatable conditions that would not normally be possible

in actual field testing. It is important that such a channel model represent all of the relevant behaviour and properties of actual propagation environments as accurately as possible. Numerous wireless channel models have been proposed to characterize time and/or space-variant propagation environments [1]–[13]. An accurate and practical channel model must be chosen and verified before implementation in a fading channel simulator.

Different software-based fading channel simulators using general-purpose processors or digital signal processors (DSPs) have been proposed [14]–[18]. Programmers endeavor to map the parallel operations onto the very long instruction word (VLIW) architecture of high-performance DSPs and/or use the single-instruction stream multiple-data stream (SIMD) instructions of general-purpose processors to speed-up the simulation. Even using optimized software simulators running on high-performance processors, the speed is limited by the inherently sequential instruction execution and the lack of specialized functional units for Monte-Carlo (MC) simulation. In fact, the required run times for the accurate performance evaluation of the most recent low bit error rate (BER) baseband algorithms are becoming prohibitively long, which makes software-based simulation an inefficient technique.

Hardware-based simulators can verify the design at-speed, significantly reducing the simulation time compared to software-based testing schemes [19], [20], and hence they reduce the time-to-market. Hardware-based fading channel simulators use digital hardware [14], [16], [21]–[25] or employ analog techniques [17], [21] for at least part of the baseband signal processing. Regardless of the selected interface (RF, analog baseband, or digital baseband), a digital fading channel simulator is preferred over analog simulators to achieve the best possible accuracy, flexibility, and repeatability [19].

Many commercially available fading channel simulators require complex and relatively expensive hardware consisting of several circuit cards with multiple processors. For example, the NoiseCom MP-2500 Multipath Fading Emulator [26] consists of 11 circuit boards, not including the RF circuitry, cooling fans, or external computer interface for setting various parameters of a frequency-selective fading channel with up to 12 paths. The compactness of the fading channel simulator is especially important when simulating systems with a relatively large number of channels, such as multiple-input multiple-output (MIMO) systems and ad-hoc networks.

Field-programmable gate arrays (FPGAs) are an ideal platform that provide suitable features for the efficient implementation of fading channel simulators. While FPGAs have previously been used as co-processors to implement the computationally-intensive signal processing algorithms of a channel simulator [19], it is especially convenient and

Manuscript received August 9, 2006; revised July 24, 2007. This work was supported by the Natural Sciences and Engineering Research Council of Canada under grant OGP0105567.

A. Alimohammad and B. F. Cockburn are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada.

efficient to realize the baseband algorithms along with the fading channel simulator and noise generator on the same FPGA. Realizing the baseband algorithms along with the fading channel simulator on the same FPGA can simplify the required hardware and significantly speed up the simulation. In addition, integrating baseband algorithm development and implementation early in design is useful to evaluate trade-offs in design complexity and feasibility, to identify promptly the real-time bottlenecks in the proposed systems, to assess alternative algorithms and implementation strategies, and to determine the required hardware resources [27].

Modeling and implementation aspects of the signal processing algorithms used in multipath fading channel simulators are somewhat lacking in the published literature. The rest of the article is organized as follows. Section II reviews the fundamental characteristics of Rayleigh fading channels. Two commonly used fading channel simulator techniques, namely the filter-based method and the sum-of-sinusoids (SOS) based scheme, are presented. Issues affecting fading channel model implementations based on digital filters are considered in Section III. Channel simulation based on the SOS approach is discussed in Section IV. The implementation of a fast discrete-time fading channel simulator based on the SOS model is presented and evaluated in Section V. Finally, Section VI makes some concluding remarks.

## II. MODELING AND SIMULATION TECHNIQUES FOR RAYLEIGH FADING CHANNELS

In a richly-scattered wireless environment, multiple attenuated and delayed copies of the transmitted signal arrive at the receiver along all possible paths. If the difference of the arrival delays of two signal copies is much smaller than the period of the transmitted signal (i.e.,  $T_m \ll T_s$ , where  $T_m$  is the maximum excess delay [28] in the impulse response and  $T_s$  is the symbol period), then the multipath components are not resolvable and the channel is said to be a frequency-nonselective (or frequency-flat) fading channel. In this case, all of the frequency components undergo the same attenuation during transmission through the channel. When the propagation path involves two-dimensional (2-D) isotropic scattering with an omnidirectional receiving antenna at the mobile receiver that collects signals from all possible directions, then the path phases can be assumed to be uniformly distributed over  $(0, 2\pi)$  [1]. According to the Central Limit Theorem (CLT) [29], [30], the sum of a large number of signals with constant amplitudes and uniformly-distributed random phases produces a signal that has a Gaussian distribution with a zero mean. This signal can be represented as a complex Gaussian process  $c(t)$  where the real  $\Re\{c(t)\}$  and imaginary  $\Im\{c(t)\}$  components are independent zero-mean Gaussian [31] with equal variance  $\sigma^2$ . Thus the envelope  $|c(t)| = \sqrt{c_i(t)^2 + c_q(t)^2}$  follows the Rayleigh distribution  $f_{|c|}(c) = \frac{c}{\sigma^2} \exp[-c^2/(2\sigma^2)]$ , where  $\sigma^2$  is the time-averaged power and  $\phi(t) = \tan^{-1}(\frac{c_q(t)}{c_i(t)})$  is the phase of the received signal. Since the processes  $c_i(t)$  and  $c_q(t)$  are independent and Gaussian, it can be shown that  $\phi(t)$  has a uniform distribution,  $\phi \in (-\pi, \pi)$  [3].

The spectrum of the received signal depends on the assumptions made about the angle-of-arrival (AOA) statistics

and the radiation pattern of the receiving antenna. Under the common assumption of a 2-D isotropic scattering environment with an omnidirectional receiving antenna at the receiver with uniformly distributed phases over  $(0, 2\pi)$ , M. J. Gans introduced the notation of a Doppler power spectrum (DPS) in 1972 [3]. Based on the flat fading channel model developed by R. H. Clarke in 1968 [2], the power spectral density (PSD) function associated with the components  $c_i$  and  $c_q$ , denoted by  $G_{c_i}(f)$  and  $G_{c_q}(f)$ , respectively, has a U-shaped form that is bandlimited to  $\pm f_D$ . This is the now well-known Jakes PSD or Jakes power spectrum, given by

$$G_{c_i}(f) = \begin{cases} \frac{1}{2\pi f_D} \frac{1}{\sqrt{1-(f/f_D)^2}} & |f| < f_D \\ 0 & |f| \geq f_D \end{cases} \quad (1)$$

where  $f_D$  is the maximum Doppler frequency.

A frequency-flat channel is usually modeled as a time-correlated Gaussian wide-sense stationary (WSS) random process [1]–[4]. Therefore, the statistical properties of such a temporal and spatially varying channel can be completely described by the first-order and second-order statistics of the channel model [29]. In fact, the important properties of the fading channel model are manifested in the autocorrelations,  $R_{c_i, c_i}(\tau)$  and  $R_{c_q, c_q}(\tau)$ , and the cross-correlation  $R_{c_i, c_q}(\tau)$  of the  $c_i(t)$  and  $c_q(t)$  components of  $c(t)$ , the autocorrelation  $R_{c, c}(\tau)$  of the complex envelope of  $c(t)$ , and the autocorrelation of the squared envelope  $R_{|c|^2, |c|^2}(\tau)$  [2], [12], [32]. The autocorrelation function (ACF) can be obtained by taking the inverse Fourier transform of the PSD given in Equation (1) as follows:

$$\begin{aligned} R_{c_i, c_i}(\tau) &= R_{c_q, c_q}(\tau) = E[c_q(t)c_q(t+\tau)] = J_0(2\pi f_D \tau) \\ R_{c_i, c_q}(\tau) &= R_{c_q, c_i}(\tau) = 0 \\ R_{c, c}(\tau) &= E[c(t)c^*(t+\tau)] = 2J_0(2\pi f_D \tau) \\ R_{|c|^2, |c|^2}(\tau) &= E[|c(t)|^2 |c(t+\tau)|^2] = 4 + 4J_0^2(2\pi f_D \tau) \end{aligned} \quad (2)$$

where  $\tau$  is the time lag,  $E[\cdot]$  denotes expectation, and  $J_0(\cdot)$  is the zeroth-order Bessel function of the first kind. Equations (2) imply that to simulate multiple fading processes that are correlated in time, but uncorrelated between processes, the processes should fulfill the following conditions: (1) the in-phase and quadrature components of each underlying complex Gaussian random process are zero-mean independent Gaussian processes with identical variances and identical ACFs; (2) the ACFs of the  $c_i(t)$  and  $c_q(t)$  components are functions of  $\tau$  only, not functions of  $t$ , (i.e., the fading signal is WSS), and (3) the discrete-time samples of the Rayleigh fading process  $g(t) = |c(t)|$  are correlated in a way that depends on the maximum Doppler frequency normalized by the sampling rate, propagation geometry, and antenna characteristics [33].

The WSS channel model has led to different simulator designs that can be used in the development and accurate error-rate performance evaluation of wireless systems. A well-known approach to the design of a colored Gaussian process is to shape the spectrum of a white Gaussian process using a filter that has a transfer function equal to the square root of the PSD of the desired stochastic process [5], [10], [33]–[35] (i.e., for Rayleigh flat-fading, the appropriate PSD is the Jakes PSD in Equation 1). Another efficient approach to approximate a

suitably correlated Gaussian process with the desired statistical properties in (2) is based on the incoherent superposition of independent complex-valued signals. This model is known as the sum-of-sinusoids approach [2] and is based on Rice's theorem [36]. In the SOS-based channel models, the in-phase  $c_i(t)$  and quadrature  $c_q(t)$  components of a WSS complex Gaussian random process  $c(t)$  are formed by the sum of a finite number of sinusoidal waveforms having amplitudes, frequencies, and phases that are appropriately selected to accurately reproduce the desired statistical properties in (2).

Numerous time-correlated SOS-based Rayleigh fading channel models have been proposed [4], [6]–[9], [11]–[13] and these have led to different simulator designs. Nevertheless, among the different available algorithms for the generation of correlated Rayleigh random variates, some do not produce statistically accurate fading variates and others are impractical for hardware implementation. For instance, it was shown in [30] that the probability density function (PDF) of the quadrature components of the complex envelope and also the squared envelope correlation of the model in [13] are non-stationary, i.e., they are function of time  $t$ . Therefore, it is important to verify the statistical properties of a proposed chosen model before using it as the basis for a fading channel simulator.

The filter-based technique has received more attention for the implementation of fading channel simulators as it can be customized to accurately reproduce the statistical properties required for simulating fading channels [32]. Many commercially available fading channel simulators [37]–[41] employ the filter-based technique. To implement the computationally-intensive multi-rate signal processing algorithms of filter-based techniques, the published filter-based fading channel simulators are commonly realized on heterogeneous architectures (usually consisting of general-purpose processors (GPPs), DSPs, FPGAs, etc.) [18], [19]. In these simulators, those portions of the simulator that are inefficiently simulated on a GPP can be off-loaded to a dedicated device, such as a FPGA or a DSP. For example, the fading simulator in [19] uses a floating point DSP combined with a FPGA to realize a frequency-selective channel simulator. The design in [18] uses two 32-bit floating-point DSP processors to implement a multipath fading simulator. Implementing a parameterizable fading channel simulator on a single FPGA is a challenging task due to the relatively large computational complexity of the filter-based signal processing algorithms [19].

In the next section, we present a novel design and implementation scheme for realizing a parameterizable fading channel simulator on a homogenous architecture. Specifically, the new design is an accurate filter-based fading channel simulator that is compact enough to be integrated on a single FPGA along with different communication circuits of interest.

### III. IMPLEMENTATION OF FADING CHANNEL USING DIGITAL FILTERS

To generate correlated Rayleigh variates, the Gaussian-distributed in-phase and quadrature components can be spectrally shaped by multiplying the frequency domain Gaussian

components by  $\sqrt{G_{c_i}(f)}$ . Then an inverse fast Fourier transform (IFFT) can be applied to the resulting discrete spectrum to obtain time series data [5], [33]. The resulting series is still Gaussian by virtue of the linearity of the IFFT, and it will have the desired Jakes spectrum. The IFFT has a computational complexity of  $O(M \log_2 M)$ , where  $M$  is the number of time-domain sampled Rayleigh channel coefficients. One major disadvantage of the IFFT method is its block-oriented nature, which requires all channel coefficients to be generated and stored before the data is sent through the channel. This implies significant memory requirements and may preclude unbounded continuous transmission, which is usually preferred in long running characterization applications.

An alternative technique to generate the in-phase and quadrature components of fading variates with a particular correlation between variates is to begin with two independent, zero-mean, white Gaussian random variables  $n_i(t)$  and  $n_q(t)$  with identical variance. A (low-pass) filtering operation on the complex Gaussian samples with flat PSD [5], [33], yields samples that also have a Gaussian distribution, with spectrum  $G_{out}(f) = G_{in}(f)|H(f)|^2$ , where  $G_{in}(f)$  is the spectrum of the input samples and  $|H(f)|^2$  is the squared magnitude response of the filter. This filter is often called the shaping filter for it determines the power spectrum shape and the temporal correlation function of the fading process. As mentioned in Section II, the theoretical spectral density of the complex envelope of the signal received by an omnidirectional antenna in a Rayleigh fading wireless channel is given by the Jakes PSD [42]. A shaping filter can be designed with a frequency response equal to the square root of the Jakes PSD,  $\sqrt{G_{c_i}(f)}$ , for a Rayleigh flat-fading process. A correlated Rayleigh process can then be generated by combining the two filtered processes in quadrature.

The reciprocal square root in Equation (1) is an irrational function [34], which cannot be implemented exactly in hardware, so it is common to use a rational approximation of the Jakes PSD. To provide spectral shaping for a rational implementation, we used transformation-based filter designs [18], [19]. An important filter design constraint to consider is that in wireless communication channels  $f_D \ll 1/T_s$ . Consequently, the bandwidth of the spectral shaping filter must typically be much narrower than the bandwidth of the transmitted signal. For example, consider the digital cellular system DSC1800 (GSM1800), which operates at  $f_c = 1.8$  GHz. If the mobile receiver has a maximum speed of  $v = 300$  km/h, then  $f_D = f_c \times (v/c) = 500$  Hz, where  $c$  is the speed of light. If the signal is sampled at  $R_s = T_s^{-1} = 10$  MHz, then the normalized Doppler frequency (also called the Doppler rate) would be  $f_D T_s = 0.00005$ . Since the sampled channel waveform is bandlimited to  $f_D T_s$ , for many common applications with a high sampling rate,  $f_D T_s$  is relatively small (e.g., typical values for mobile radio channels lie in the range  $10^{-5}$  to  $10^{-2}$  [34]). Unfortunately, symbol-rate design of an extremely narrow-band digital filter may run into numerical problems [43]. Instead, to ensure stability and greater computational efficiency it is advantageous to design the PSD shaping filter for a lower sampling frequency  $R_{ch}$ , and then increase the sampling frequency using interpolation

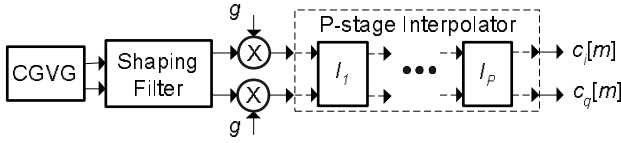


Fig. 1. Architecture of a filter-based fading channel simulator.

to achieve the target symbol rate. The sampling rate of the filter should be increased by an interpolation factor  $I = \lceil \frac{R_s}{R_{ch}} \rceil$  to reach the desired signal sampling rate, where  $R_{ch}$  is the channel sampling rate used to design the shaping filter. After interpolation, a low-pass filter (LPF) is utilized to eliminate the signal replicas introduced in the frequency domain by interpolation. For a practical mobile system, the factor  $I$  can be large and thus the complexity of the real-time interpolation filter can be large. To reduce the computational complexity of the low-pass filters, interpolation is usually accomplished using a multi-stage interpolator design.

The functional structure of a filter-based fading channel simulator is shown in Fig. 1. The generated samples from a complex Gaussian variate generator (CGVG) are spectrally shaped using a digital filter and multiplied by a scaling factor  $g$  to normalize the power of the final resulting channel model waveform  $c[m]$  to one. After generating the fading processes (in-phase and quadrature), the sampling rate of the filter can be increased by an interpolation factor  $I = \prod_{i=1}^P I_i$ , where  $P$  is the number of interpolation stages and  $I_i$  is the interpolation factor of the  $i$ -th stage.

The fading channel simulators in [5], [23], [44], [45] used FIR filters as the shaping filter while the designs in [10], [17], [19], [46], [47] used IIR filters. Several important points should be considered when implementing fading channel simulators using FIR and IIR filters on hardware platforms:

- The degree of the FIR filter is related to the time span of the truncated signal held in the filter and inversely proportional to the Doppler frequency. Specifically, implementation of an extremely narrow-band digital filter with a sharp cutoff and very large attenuation in the stop-band requires a large-order FIR filter [42], [47]. Meeting the same specifications with an IIR filter typically requires fewer hardware resources than an FIR filter. Having both the feedforward and feedback polynomials in an IIR filter permits steeper frequency roll-offs to be implemented for a given filter order than an FIR filter [43]. Thus, rather than designing a high-order FIR filter for an extremely small target  $f_D T_s$  (e.g.,  $10^{-5}$ ), an IIR filter can be designed instead with a smaller order.

- An FIR filter has no feedback path and is thus inherently stable. In an IIR filter, however, as the coefficients are quantized for a fixed-point implementation, the resulting numerical error is fed back and can cause instability [43]. Moreover, such effects can cause significant deviations from the expected filter response. One important decision is thus to determine the maximum number of bits required to represent the constant coefficients and internal signals in the filter. It is well-known that as the order of the polynomial increases, so too does the sensitivity of the obtained roots of a polynomial to the accuracy of its coefficients [48]. Also, for a narrowband filter

with a high sampling rate, the zeros and poles tend to be crowded near the unit circle. If the poles reside on the unit circle or outside of it, the filter becomes unstable [48]. As the coefficients are quantized for a fixed-point platform, due to the rounding or truncation of values, the quantization error can be fed back in the filter, successively magnifying the total error and causing instability. Even if the filter stays stable, the poles can be displaced significantly from their design locations by the quantization of the coefficients, and thus the target specification will not be achieved. Therefore, a careful analysis is required to ensure adequate precision in the coefficients. Since FPGAs, unlike fixed register size DSPs, allow custom bit widths, the filter coefficients representation can be different than the internal signal representation, providing greater flexibility and implementation efficiency.

- For a fixed signal sampling rate of  $R_s$ , the lower the channel sampling rate, the larger must be the interpolation factor. For a high  $R_{ch}$ , the filter order will increase and the filter will become more sensitive to quantization. There is therefore a tradeoff between the computational complexity and numerical stability of the shaping filter and the computation requirements of the interpolation filter. There are many possible combinations of shaping filter designs and interpolator implementations. It is reasonable to design the shaping filter with a small  $R_{ch}$  (compared to  $R_s$ ), which provides stable operation for a fixed-point realization with acceptable computational complexity. Then the interpolator can be designed and implemented efficiently using a variable polyphase filter, to accommodate different Doppler rates, with a windowed  $\text{sinc}(\cdot)$  function impulse response [42]. Note that when the digital filter is designed at a fixed Doppler rate  $f_D T_s$  to provide sufficiently accurate frequency response, then the structure in Fig. 1 is not flexible enough to produce a Rayleigh fading signal with an arbitrary discrete Doppler rate [34]. For an irrational value of  $f_D T_s$ , if the IIR spectral shaping filter is designed for a fixed discrete maximum Doppler rate, then the interpolator must re-sample the process so that rational fractions of the desired rate  $f_D T_s$  can be approximated. The irrational Doppler rate factor may not be a serious limitation when simulating a fading channel because in performance verification, obtaining high accuracy in the Doppler rate is not usually important [34].

We designed a bandlimited Jakes spectral shaping filter using an elliptic IIR filter to closely approximate  $G_{ci}(f)$  in the passband while providing large suppression in the stopband. An elliptic filter is an efficient candidate that has an especially sharp transition from the passband to the stopband for a given order among the different classical IIR structures [43]. A discrete-time elliptic filter can be designed using the cascade Direct-Form II second-order (i.e., two poles and two zeros) section (biquad) structure that is more robust under quantization than the Direct-Form structure [43]. For example, a cascade of four biquads is used in [6] and a cascade of seven biquads is used in [34]. As shown in Fig. 2, for a fixed maximum Doppler rate  $f_D T_s = 0.2$ , the magnitude response of filters designed with  $K = 5$  cascaded biquads closely matches the ideal response in the passband and has a steep roll-off in the transition to the stopband. We also followed

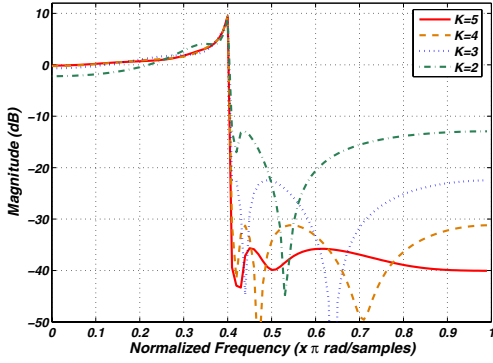


Fig. 2. The magnitude responses of elliptic filters composed of  $K$  cascaded biquads with  $f_D T_s = 0.2$ .

a typical assumption that the Doppler spectrum for mobile speeds of interest is less than 2 kHz wide [49]. A rule of thumb for an appropriate channel sampling rate is to use between 6 to 12 samples/period of the highest frequency of an analog signal [42]. We therefore assumed 10 samples/period at the highest frequency [42], so  $R_{ch} = 10f_D = 20$  Ksamples/sec.

The interpolation step can be performed by adding  $I - 1$  zeros between samples (zero-padding) and filtering the resulting signal with a LPF that has a cutoff frequency of  $\pi/I$  rads/sec. The FIR filter memory must be long enough to span the distance between the channel model samples. Since the interpolator factor  $I$  can in general be large, the FIR filter length becomes significantly long and thus a multi-stage interpolation scheme using FIR filters is commonly employed [18], [19]. However, when the interpolation factor  $I$  is large, we found that this approach may not be efficient enough to be implemented on resource-limited FPGAs. As an alternative we propose to use a low-pass IIR filter with much smaller degree that provides almost linear phase response in the passband as with an interpolation low-pass filter (ILPF). Using an IIR filter as a low-pass interpolation filter decreases the computational complexity significantly. In order to design a  $P$ -stage interpolator, we made the reasonable assumption that the WSS channel model can be used for the urban mobile radio channel over bandwidths of up to 10 MHz [34], [50]. The interpolation factor of our designed system can therefore be up to  $I = \lceil R_s/R_{ch} \rceil = 500$ . Note that the structure in Fig. 1 is sufficiently flexible such that fading channels with different specifications can be readily supported (e.g., various bandwidths using different interpolation factors). The fading channel architecture is also scalable in that it can be instantiated multiple times to simulate various scenarios such as frequency-selective and MIMO channels.

An important IIR filter design decision is the choice of fixed-point representation. Fewer bits in the representation leads to smaller designs at the cost of reduced signal accuracy. Our goal was to use a fixed-point representation while keeping the accuracy close to that of a floating-point design. To ensure this, the numerical behavior of our fixed-point algorithm was verified against that of a more accurate 64-bit floating-point implementation. The internal signal bit widths can be

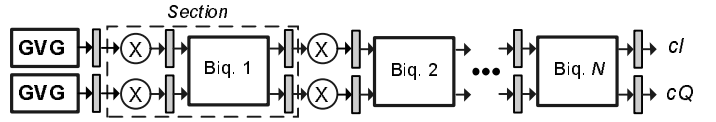


Fig. 3. (a) The structure of cascading Gaussian variate generator and  $N = K + P$  second-order sections.

determined by calculating theoretical bounds on the dynamic ranges of the signals, and on the maximum output errors introduced by truncation in the fixed-point representation. The lower and upper bound values of each signal state how many bits are required at any point in the computation in order to minimize the probability of overflow/underflow while guaranteeing a prescribed degree of accuracy at the filter output. To make sure that the filters are stable under quantization effects, we have designed the filters in 32-bit fixed-point format using Filter Design Toolbox in Matlab [51]. This package offers bit-true implementations of biquads with section scaling and reordering to obtain maximum accuracy.

Realizing the fading channel simulator architecture shown in Fig. 1 requires two instantiations of Gaussian variate generators (for the real and imaginary components),  $K$  cascaded biquads and  $2K$  multipliers to implement the shaping filter, and  $P$  cascaded biquads and  $2P$  multipliers to implement the ILPFs. This is shown in Fig. 3 where  $N = K + P$ . Due to the large computational complexity of these high-precision filters, a parameterizable Rayleigh fading simulator might not be able to fit on one FPGA device and thus might require multiple devices [18]. For example, the fading simulator in [19] uses a floating-point DSP (to implement the random number generator, Doppler spectrum shaping filter, and the first-stage interpolator) combined with a FPGA (to implement the final interpolation stage and other required components) to realize a frequency-selective channel simulator.

As shown in Fig. 3, since both the shaping filter and the ILPFs are designed using standard IIR filters, we propose to exploit time-multiplexed resource sharing to obtain the maximum performance with a minimum amount of FPGA resources. In this technique, the computation resources of the spectral shaping filter are shared with the ILPFs for an efficient on-chip realization of a fading channel emulator. Resource sharing of independent operations of the spectral shaping filter and ILPFs offers significant hardware savings in the fading channel emulator. The throughput of a fading simulator depends upon the binding of the second-order sections to the shared resources. The sampling rate of the hardware-based digital filters was high enough that the throughput reduction of the time-multiplexed scheme compared to the direct instantiation approach did not impact the maximum target sampling rate.

For a meaningful comparison with the results of the SOS-based scheme, we implemented all of our fading channel simulators assuming that  $f_D T_s = 0.01$ . We designed the shaping filter using an elliptic IIR filter with  $K = 5$  cascaded biquads for  $f_D T_s = 0.1$ . We then designed the interpolator for  $I_1 = 10$  using a low-pass Chebyshev Type II IIR filter with  $P_1 = 5$  cascaded biquads. The ILPF has a maximum

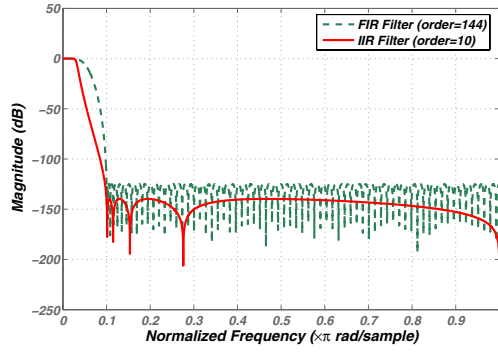


Fig. 4. Magnitude response of the designed ILPF.

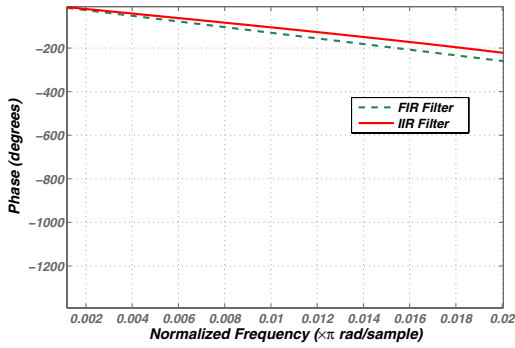


Fig. 5. Phase response of the designed ILPF.

of 0.01 dB attenuation in the passband and a minimum of 127 dB attenuation in the stopband. To optimize the hardware efficiency and accuracy of the design, we designed the shaping filter and the ILPF “together”. The shaping filter was designed to provide a frequency response that closely matches the desired response over the passband and up to frequencies just inside the stop-band. However, at higher frequencies the attenuation is somewhat less (about 35 dB). Then the ILPF provides a larger attenuation over frequencies where the shaping filter might not provide adequate attenuation (127 dB). Therefore the ILPF not only attenuates the out-of-band signals, it can also help the shaping filter to provide more accurate samples. For comparison note that we could use equiripple FIR filters to design the ILPF with the same parameters, but this would require a filter of degree 144. Fig. 4 plots the magnitude response of the IIR and FIR low-pass filters for interpolation. Clearly, designing the ILPF using an IIR filter leads to a significantly smaller order, and this strategy is much more computationally efficient. Fig. 5 shows that the ILPF provides almost linear phase response over the passband region (0 to 2 kHz).

Due to the slow variation of samples at the shaping filter compared to the fast operating rate of biquads implemented on an FPGA, we implemented the shaping filter and the ILPF using one shared biquad. Specifically, we used the Gaussian variate generator (GVG) in [52] that requires only 1.3% of the configurable slices, three dedicated  $18 \times 18$  multipliers, two of the BlockRAMs available in the Xilinx Virtex-II Pro

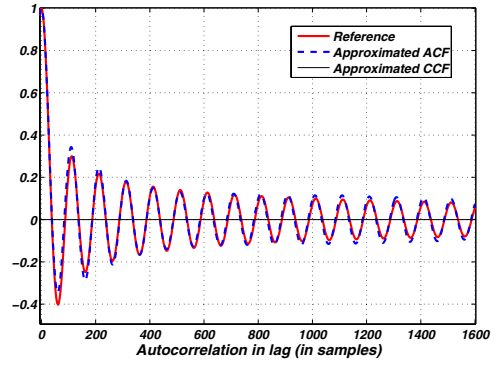


Fig. 6. The ACF and CCF of simulated fading components compared against the theoretical functions.

XC2VP100-6 FPGA, operating at 269 MHz, and generates two Gaussian variates per clock cycle. Our 32-bit fixed-point implementation of the Rayleigh fading channel simulator on an FPGA utilizes only 1421 (3%) of the configurable slices, 47 (10%) of the dedicated  $18 \times 18$ -bit multipliers, and 6 (1%) of the available memories on a Xilinx Virtex-II Pro XC2VP100-6 FPGA, while operating at 50 MHz. Even though the shaping and the low-pass interpolator filters are designed to emulate fading channels sampled at 200 kHz, the fading channel simulator runs at a relatively high clock frequency of 50 MHz. In the example system, where the ILPF generates one fading sample every 20 clock cycles, the simulator operates 12.5 times faster than the target sample rate. The simulator can be slowed down to emulate a wide variety of different channel characteristics.

A hardware description language (HDL) model of the proposed design was simulated to verify the accuracy of the results against the fixed-point software simulation results. We analyzed common statistical properties of  $10^7$  generated fading samples using  $10^6$  generated GVVs. The filters were run initially for a while (e.g., 1 msec) to get past the transient initial conditions and to allow the output signal to achieve asymptotic stationarity. Fig. 6 shows that over a wide range of lags the ACF of the sampled fading variates closely matches the theoretical function, and that the CCF of the sampled real and imaginary fading components is almost zero. The level crossing rate (LCR) of the generated fading signal and its theoretical reference are plotted in Fig. 7. As expected, the LCR deviates from the ideal at the lowest rates, but is a close match at all other rates [34]. As shown in Fig. 8, the envelope PDF of the generated fading variates closely match the Rayleigh PDF.

#### IV. ANALYSIS OF SOS-BASED RAYLEIGH FADING CHANNEL MODELS

As mentioned in Section II, another well-known scheme for generating a sequence of suitably correlated WSS fading variates is the sum-of-sinusoids technique. In order to reproduce the desired properties in Equations (2), numerous SOS-based models have been proposed that differ from one another in the model parameters, which leads to differing statistical properties [1]–[13], [53]–[56].



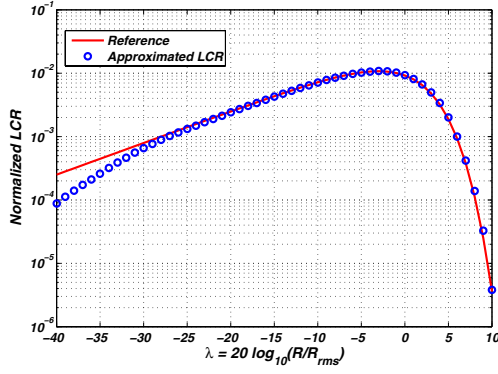


Fig. 7. The LCR of simulated fading components and its reference curve.

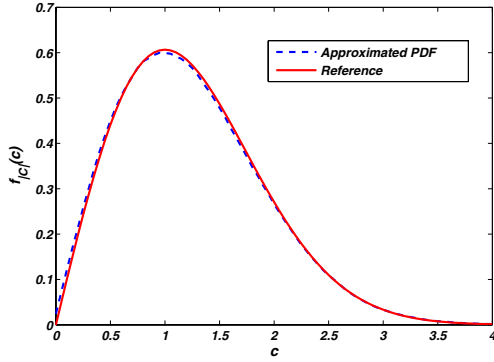


Fig. 8. Envelope PDF for the simulated fading variates and its reference plot.

Clarke proposed a useful mathematical model for the complex channel gain, under the narrow-band flat fading assumption [2]. Clarke showed that the complex channel gain  $c(t)$  at a time  $t$  can be expressed as

$$c(t) = \sqrt{\frac{2}{N}} \sum_{n=1}^N \exp [j(2\pi f_D t \cos(\alpha_n) + \phi_n)] \quad (3)$$

where  $\alpha_n$  and  $\phi_n$  are the angle of arrival and the initial phase, respectively, associated with the  $n$ -th sinusoid,  $N$  is the total number of sinusoids [2], and each low frequency oscillator has equal average amplitude (i.e., the same received power). The phase angles  $\phi_n$  and  $\alpha_n$  are assumed to be mutually independent and uniformly distributed over  $(-\pi, \pi)$  for all  $n$ . The squared envelope correlation of  $c(t)$  in (3) can be written as [30]

$$R_{|c|^2|c|^2}(\tau) = E[|c(t)|^2 |c(t+\tau)|^2] = 4 + 4 \frac{N-1}{N} \mathcal{J}_0^2(2\pi f_D \tau) \quad (4)$$

where as  $N$  approaches infinity, the squared envelope correlation asymptotically reaches the desired value  $4 + 4 \mathcal{J}_0^2(2\pi f_D \tau)$ . Due to the accurate statistical properties of Clarke's model, it has been widely used to model Rayleigh fading channels.

Different sum-of-sinusoids models have been proposed [4], [6]–[9], [11]–[13], [53], [55], [56] based on Clarke's model. These models can be broadly categorized as either deterministic or statistical [30]. In deterministic SOS simulators

[4], [8], [9], [53], [54], all the waveform parameters (i.e., amplitude, Doppler frequency and phase) are selected before the simulation starts and are held constant for all subsequent simulation trials. Hence, the properties of the generated signal are deterministic. On the other hand, in the statistical models (also called Monte Carlo SOS models) at least one of the waveform parameters is taken to be a random variable that changes for every simulation trial. The statistical properties of such a generated signal change for each simulation trial, but converge statistically to the desired properties over a large number of simulation trials [6].

Jakes proposed his deterministic SOS-based model based on Clarke's model to generate time-correlated Rayleigh fading variates [4]. The Jakes model is more computationally efficient than Clarke's model in which the in-phase  $c_i(t)$  and quadrature  $c_q(t)$  components of a stationary complex Gaussian process  $c(t)$  are formed by a *finite* superposition of sinusoids. The model assumes that  $N$  waveforms with equal power arrive at the moving receiver with uniformly distributed arrival angles  $\alpha_n = 2\pi n/N$ , such that waveform  $n$  experiences a Doppler shift  $f_{D_n}$ . Despite the extensive acceptance and application of Jakes model, for simulation results to be meaningful, they must reproduce the important statistics of real channels. It was recently shown that the assumptions and simplifications made by Jakes adversely affect the statistics of the SOS-based fading channel simulator [7], [9], [53]. One problem with Jakes' model is that the cross-correlation between the in-phase and quadrature components is significantly different from zero [9]. Another important problem with the Jakes model is that its output sequence averaged across the ensemble of fading channels is not WSS [9]. In [9] the Jakes model was improved by introducing random phase shifts in the low frequency oscillators. The WSS Jakes model proposed in [9] has the desired complex envelope autocorrelation as the number of sinusoids approaches infinity. However, consistent with Pop and Beaulieu's caution concerning the second-order statistical properties of their proposed model [9], it was proved in [57] that the autocorrelations and cross-correlations of the quadrature components and the autocorrelation of the squared envelope do not approach the desired statistics, even as  $N \rightarrow \infty$ . Pätzold also proposed several deterministic SOS-based modeling schemes that can be applied to simulate multiple Rayleigh fading processes that are correlated in time, but uncorrelated between processes [58]. For example, the method of exact Doppler spread [54] uses a finite number of sinusoids and deterministic discrete Doppler frequencies  $f_{D_n}$ . In order to ensure that the different processes are uncorrelated, this model defines  $f_{D_n}$ s in such a way that they are disjoint (i.e., mutually exclusive) for different processes.

Careful studies of the theoretical fading channel models are important as some models are insufficiently accurate. Recently, Patel *et al.* [30] showed different inaccuracies with the well-known SOS-based models. For example, the model in [13] has non-stationary and non-Gaussian properties. Also, the squared envelope autocorrelation in [13] and [12] is derived incorrectly.

Zheng and Xiao [11] introduced randomness to the Doppler frequency, the initial phase of the sinusoids, and the angles of arrival to have MC simulators with desired statistical prop-

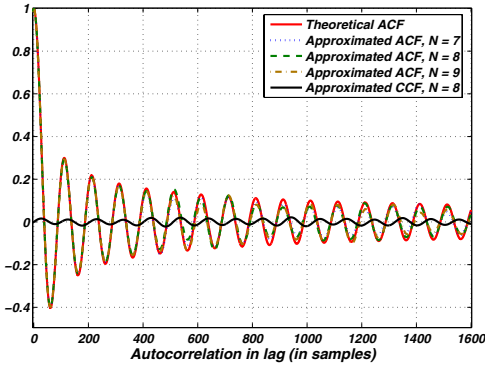


Fig. 9. The ACF and CCF calculated using the Zheng and Xiao's SOS fading channel model by averaging over 100 frames of  $10^4$  fading samples with  $f_D T_s = 0.01$ .

erties. The resulting complex-valued Rayleigh fading process  $c(t)$  is given by

**Model I:**

$$c(t) = \sqrt{\frac{2}{N}} \left\{ \sum_{n=1}^N \cos \left[ 2\pi f_D t \cos \left( \frac{2\pi n - \pi + \theta}{4N} \right) + \phi_n \right] + j \sum_{n=1}^N \cos \left[ 2\pi f_D t \sin \left( \frac{2\pi n - \pi + \theta}{4N} \right) + \varphi_n \right] \right\} \quad (5)$$

where  $\theta$  is a random variable uniformly distributed over  $[-\pi, \pi)$ , and  $\phi_n$  and  $\varphi_n$  are statistically independent and uniformly distributed over  $[-\pi, \pi)$  for  $1 \leq n \leq N$ . *Model I* has several advantages over previous simulation models such as [30]:

(1) It avoids the stationarity problem while maintaining the accuracy of the correlation statistics. The ACFs of the in-phase and quadrature components and the ACFs of the complex envelope match those of Clarke's reference model very closely, even for small  $N$ . Also, the ACF of the squared envelope of the fading signal  $c(t)$  is  $R_{|c|^2|c|^2}(\tau) = 4 + 4\mathcal{J}_0^2(2\pi f_D \tau) + \frac{2 + \mathcal{J}_0(2\pi f_D \tau)}{N}$  [11]. It asymptotically approaches the desired autocorrelation as  $N \rightarrow \infty$ , while good approximation has been observed when  $N$  is not less than eight. The model always produces uncorrelated in-phase and quadrature components, as required for a Rayleigh-distributed envelope.

(2) Due to the proper selection of the simulation parameters in the model in [11], the variance of correlation functions  $\text{Var}[R(\cdot)] = E[|\hat{R}(\cdot) - \lim_{N \rightarrow \infty} R(\cdot)|^2]$  of this model are lower than the variances for most other models for finite  $N$  [12].

Since the improved SOS-based channel models require that a relatively small number of sinusoids ( $8 \leq N \leq 12$ ) converge statistically to the desired properties, they are good candidates for an efficient and compact hardware implementation. For example, the channel simulator in [16] uses  $N = 8$ , the design in [14] uses  $N = 9$ , and the design in [59] uses  $N = 16$ . The commercially available Ascom SIMSTAR fading channel simulator uses  $N = 22$  [39]. Fig. 9 shows the ACF of the in-phase component of the model in [11] for three different small numbers of sinusoids. The figure also shows the cross-correlation of  $c_i$  and  $c_q$  for  $N = 8$ . The LCR and PDF of

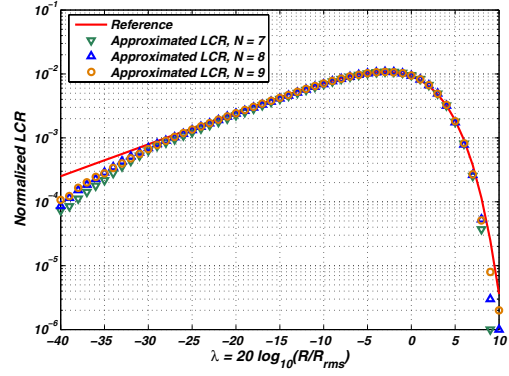


Fig. 10. The normalized LCR of *Model I* calculated by averaging over 100 frames of  $10^4$  fading samples with  $f_D T_s = 0.01$ .

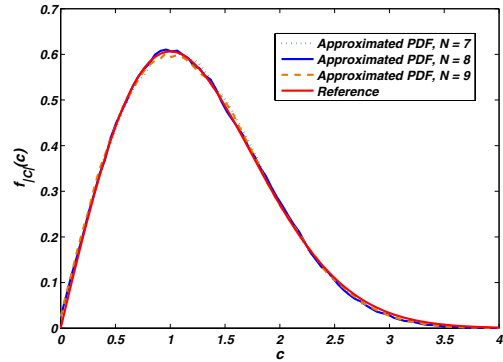


Fig. 11. Envelope PDF of *Model I* calculated by averaging over 100 frames of  $10^4$  fading samples with  $f_D T_s = 0.01$ .

$10^6$  generated fading samples using the model in (5) with for  $N = 7, 8, 9$  and  $f_D T_s = 0.01$  are plotted in Figs. 10 and 11, respectively.

As shown in Figs. 9-11, the statistical properties of the generated fading variates closely match the theoretical functions, when  $N$  is as small as 8. However, an important point to note is that the SOS-based *Model I* is a statistical simulator whose statistical properties converge to the desired properties only over a sufficiently large number of simulation trials. In fact, the channel model is not ergodic and so the statistical properties of a single simulation, no matter how many samples are generated, do not converge to the reference properties. Its statistical properties converge to the desired properties only when they are averaged over a large number of simulation trials. If the channel model is ergodic, then the statistics of the output should converge to the reference ones in a sufficiently long single simulation trial. Fig. 12 plots the ACF of the channel generated using one simulation trial with the model in (5) for one block containing  $10^7$  samples,  $f_D T_s = 0.01$ , and  $N = 8$ . Clearly, the ACF deviates from the reference ACF of the Rayleigh fading channel, especially at the larger lags, while the CCF stays close to zero. Therefore, *Model I* may not be suitable for simulating an ergodic Rayleigh fading channel. In [60], we propose an improved SOS-based model compared to *Model I* to overcome this limitation.



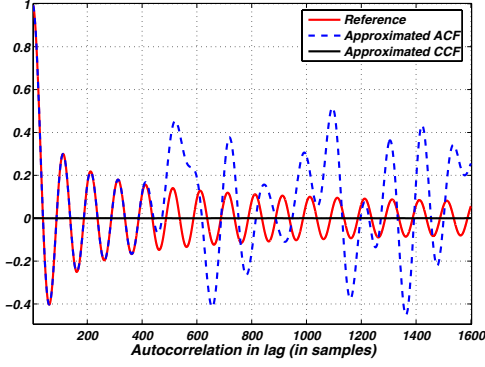


Fig. 12. ACF and CCF for one block containing  $10^7$  fading samples using the Zheng and Xiao's SOS model with  $f_D T_s = 0.01$  for  $N = 8$ .

Even though the model in (5) may not be suitable for emulating an ergodic Rayleigh fading channel, it can still be used for simulating block-based transmission systems. As shown in Figures 9-11, the statistics of the generated fading variates closely match the reference curves for block-oriented fading channel models. Moreover, in some applications it is sufficient to match the ACF in a certain range only. Since these SOS methods converge statistically to the desired properties, it is important to determine the number of simulation trials needed to achieve a desired convergence level. This is directly related to the variation in the time-averaged properties of a single simulation trial from the desired ensemble average properties.

## V. IMPLEMENTATION OF A SOS-BASED FADING CHANNEL SIMULATOR

As discussed in Section IV, the detailed comparison study in [30] concluded that the model in [11] has superior properties among all models for relatively small choices of  $N$ . In this section we propose a compact implementation of SOS-based channel model proposed in [11] that fits on a small fraction of a commonly used FPGA. The continuous-time equations in *Model I* [11] can be written in discrete-time as follows to simplify the computation for hardware implementation:

$$\begin{aligned}
 c[m] &= \sqrt{\frac{2}{N}}(c_i[m] + jc_q[m]), \quad m = 1, \dots, M \\
 c_i[m] &= \sum_{n=1}^N \cos(2\pi(\psi_{in}m + \phi_n)) \\
 c_q[m] &= \sum_{n=1}^N \cos(2\pi(\psi_{qn}m + \varphi_n)) \\
 \psi_{in} &= f_D T_s \cos(\alpha_n), \quad \psi_{qn} = f_D T_s \sin(\alpha_n) \\
 \alpha_n &= \frac{2\pi n - \pi + \theta}{4N}
 \end{aligned} \tag{6}$$

where  $m$  is the discrete time index and  $M$  is the block length. Random variables  $\{\phi_n\}$  and  $\{\varphi_n\}$ , for  $n = 1, 2, \dots, N$ , lie within  $(-0.5, 0.5)$  and can be generated at the beginning of each fading block using two on-chip pseudorandom number generators (PNGs) or read from an external source. We used

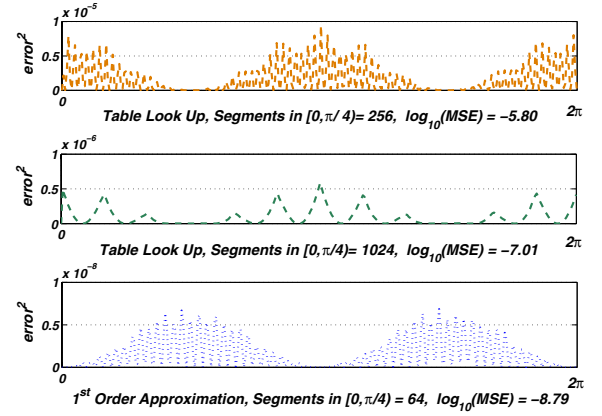


Fig. 13. The 16-bit sine approximation error using table lookup (for different numbers of segments) and linear interpolation.

combined Tausworthe generators to implement the PNG with a period length of  $\rho \approx 2^{88}$  [61].  $\{\psi_{in}\}$  and  $\{\psi_{qn}\}$  are arrays of constant values within a fading block that can be re-initialized at the beginning of each fading block. Their values depend on a uniformly generated random variate  $\theta \in (-\pi, \pi)$  and  $f_D$ . Note that the quality of the uniform random number generator becomes crucial especially when the number of sinusoids is small. If the randomly generated parameter sets are not uniformly distributed, then the statistics become poor and the characterization measurement will be incorrect.

From Equations (6) it is clear that an efficient and accurate implementation of the cosine function will directly improve the performance and overall accuracy of the simulator. There are various standard approaches for approximating trigonometric functions such as CORDIC [62], polynomial approximations, and various table lookup schemes (e.g., direct table lookup and multipartite techniques [63]). The choice of a method and a particular implementation depends on such requirements as throughput, latency, and area as well as the required accuracy. Storing and later retrieving quantized values of the trigonometric function in an on-chip memory, so-called table lookup, is relatively fast, but the approximation accuracy is limited by the size of the on-chip memory. Also the size of the memory grows exponentially with the size of the input word, which confines this solution to relatively small input precisions (i.e.,  $\leq 12$  bits). The table size can be reduced by exploiting the symmetry properties of the  $\sin(\cdot)$  and  $\cos(\cdot)$  functions and then storing only a quarter cycle of the trigonometric function in memory. Another scheme that may require less memory, compared to the table lookup but with similar accuracy, is to use linear (or higher order) interpolation. The channel simulator in [16] uses linear interpolation to implement the sine function, while the design in [59] uses a quarter period, partitioned into 256 segments, to approximate the trigonometric functions. Fig. 13 plots the squared error of the table lookup with different numbers of segments along with the one for the linear interpolation method with 64 segments for the quarter cycle. This figure shows that the MSEs of these three schemes, when implemented using a 16-bit fixed-point format, are relatively close.

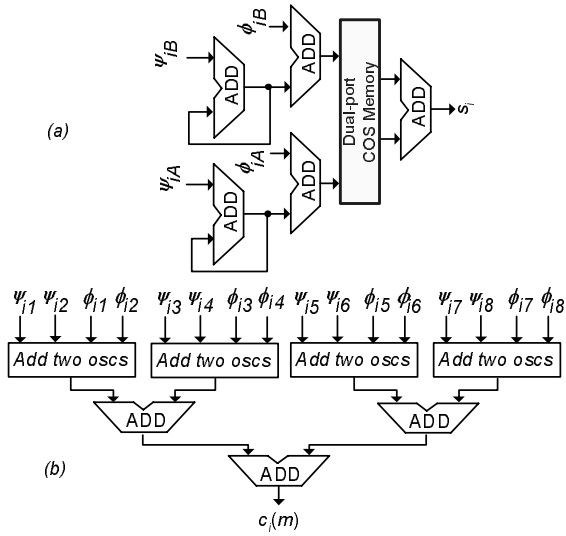


Fig. 14. (a) Circuit for summing  $N = 2$  complex oscillators. (b) Tree-structured adder for summing  $N = 8$  oscillators.

To quickly add the  $N$  quadrature components of  $c_i[m]$  in 6, we took advantage of the fast adder blocks now widely available in FPGAs and organized them into a pipelined tree-structured circuit. The datapath shown in Fig. 14(a) adds two 16-bit in-phase components of  $c_i[m]$  given in (6). Note that the values of  $\psi_{in}m$  (and similarly  $\psi_{qn}m$ ) are calculated using an adder instead of a multiplier. The adder accumulates successive values of  $\psi_{in}m$  for  $m = 0, \dots, M$ ; however, to calculate  $\cos(2\pi(\psi_{in}m + \phi_n))$  where  $\psi_{in}m$  is a floating-point number and  $\phi_n$  is a fractional number, only the fractional part of  $\psi_{in}m + \phi_n$  is required and the integer part of the results can be ignored. Even for a relatively small value of  $N$ , this significantly reduces the size of the implementation. This datapath also requires one dual-port ROM to simultaneously produce two values for  $\cos(2\pi(\psi_{in}m + \phi_n))$  and  $\cos(2\pi(\psi_{i(n+1)}m + \phi_{n+1}))$ . Fig. 14(b) shows the tree-structured datapath for summing  $N = 8$  in-phase components of  $c_i[m]$ . The blocks labeled “Add two oscs” denote instances of the circuit in Fig. 14(a).

The approximated ACF, CCF, PDF, and LCR of  $10^6$  generated fading variates using the SOS-based model in (6), for different fixed-point precisions, are shown along with their theoretical references in Figs. 15, 16, and 17, respectively. These plots show that 16-bit precision is sufficient to obtain fading variates with relatively accurate statistical properties. We assumed a normalized Doppler rate  $f_D T_s = 0.01$ , but the parameterizable model can use any arbitrary Doppler rate with 16-bit precision. The HDL model of the proposed 16-bit datapath was simulated to verify the accuracy of the results against the fixed-point software simulation results. Table I summarizes the implementation characteristics for the new SOS-based fading channel simulator, with  $N = 8$ , on four different FPGAs. The synthesis results verify that the optimized 16-bit datapath on the FPGA can be used to generate over 200 million complex Rayleigh fading variates per second, which is over 500 times faster than a software-based simulator written in C running on a 3.4 GHz Pentium IV processor. The

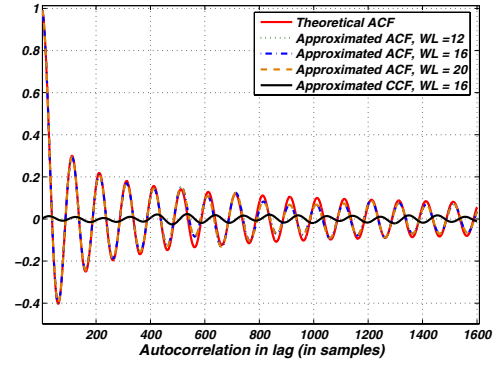


Fig. 15. ACF and CCF of the quadrature component calculated by averaging over 100 frames of  $10^4$  fading samples with  $f_D T_s = 0.01$ .

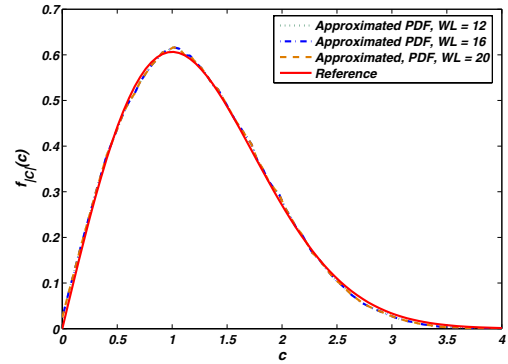


Fig. 16. The PDF of the simulated fading envelope and their references calculated by averaging over 100 frames of  $10^4$  fading samples with  $f_D T_s = 0.01$ .

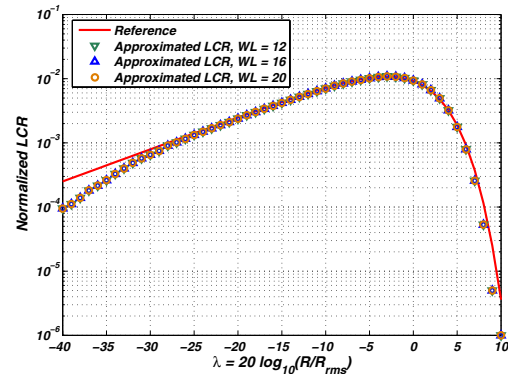


Fig. 17. The LCR of the simulated fading envelope and their references calculated by averaging over 100 frames of  $10^4$  fading samples with  $f_D T_s = 0.01$ .

TABLE I  
IMPLEMENTATION OF FADING SIMULATOR ON DIFFERENT FPGAs

Device	I <sup>a</sup>	II	III	IV
Clock freq. (MHz)	221.92	201.69	179.21	145.33
Output rate (MSamps/sec)	221	201	179	145
Number of slices	542	542	542	1,160
Resource utilization	0.86%	1%	2%	1%
On-chip memory blocks	8	8	8	8

<sup>a</sup>Design I was synthesized for a Xilinx Virtex4 XC4VFX140-11 FPGA. Design II was synthesized for a Xilinx Virtex-II Pro XC2VP100-6 FPGA. Design III was synthesized for a Xilinx Virtex-II XC2V4000-6 FPGA. Design IV was synthesized for an Altera Stratix EP1S80F1508C6 FPGA. The latency of the fading simulator is ten clock cycles in all cases.

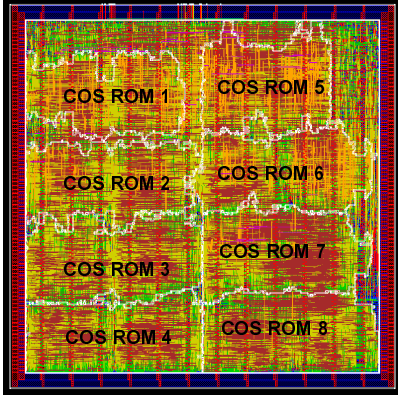


Fig. 18. Layout of the 500 MHz semicustom fading channel variate generator.

implementation complexity of the SOS-based model is directly related to the number of sinusoids used in Equation (6). For example, on a Xilinx Virtex-II Pro XC2VP100-6 FPGA that has 444 block memories, the upper bound for the number of sinusoids that can be used in this implementation is limited by  $N = 444$  (which also uses about 55% of the configurable slices). Since only a relatively small number of sinusoids ( $8 \leq N \leq 12$ ) is sufficient to guarantee that the moments of the realization are close to the reference statistics [6], [11], [30], this scheme is less computationally complex than the filter-based scheme. In addition, while the computational complexity of the filter-based technique depends on the value of  $f_D T_s$ , the resource requirements of an SOS-based scheme are almost independent of the value of Doppler rate. Also, the fading variate generation rate of the proposed SOS-based scheme (that is almost independent of  $N$ ) is significantly greater than the filter-based method. However, the filter-based fading channel simulators have significantly more accurate statistical properties over the SOS-based fading simulators. The SOS model provides a close match with the theoretical statistical properties, but only when the statistics are averaged over an ensemble of fading channels.

Fig. 18 shows the layout of a  $356,409 \mu m^2$  semicustom integrated circuit implementation of fading channel model designed in a 90-nm CMOS technology using a dual-threshold standard cell library. The core was targeted to operate at 500 MHz, generating 500 million complex fading variables per second while dissipating 36.9 mW of dynamic power. Static power dissipation is estimated to be 19.94 mW. As

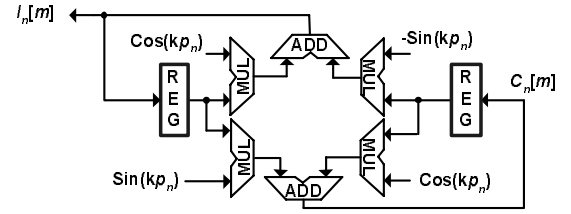


Fig. 19. Datapath for generating one low-frequency oscillator.

shown in Fig. 18, the core area is dominated by the dual-port cosine ROMs. As an alternative to the table look-up method of approximating the cosine function, we propose an iterative method to calculate the in-phase and quadrature components of the complex envelope  $c(t)$ . The continuous-time equations in (5) can be written in discrete-time as follows:

$$c[m] = \sqrt{\frac{2}{N}} \left( \sum_{n=1}^N I_n[m] + j \sum_{n=1}^N Q_n[m] \right), \quad m = 1, \dots, M \quad (7)$$

where  $I_n[m] = \cos(2\pi m f_D T_s \cos(\alpha_n) + \phi_n)$  and  $Q_n[m] = \cos(2\pi m f_D T_s \sin(\alpha_n) + \varphi_n)$ . The  $n$ -th in-phase sinusoid can be expanded as follows:

$$\begin{aligned} I_n[m] &= \cos(2\pi m f_D T_s \cos(\alpha_n) + \phi_n) \\ &= \cos(m p_n + \phi_n) \quad \text{where } p_n = 2\pi f_D T_s \cos(\alpha_n) \\ &= \cos((m-1)p_n + \phi_n + p_n) \\ &= \cos((m-1)p_n + \phi_n) \cos(p_n) - \sin((m-1)p_n + \phi_n) \sin(p_n) \\ &= I_n[m-1] \cos(p_n) - C_n[m-1] \sin(p_n) \end{aligned} \quad (8)$$

where

$$\begin{aligned} C_n[m] &= \sin(2\pi m f_D T_s \cos(\alpha_n) + \phi_n) \\ &= C_n[m-1] \cos(p_n) + I_n[m-1] \sin(p_n) \end{aligned} \quad (9)$$

The  $n$ -th quadrature component can be expanded similarly and expressed as:

$$Q_n[m] = Q_n[m-1] \cos(q_n) - D_n[m-1] \sin(q_n) \quad (10)$$

where  $q_n = 2\pi f_D T_s \sin(\alpha_n)$  and  $D_n[m] = D_n[m-1] \cos(q_n) + Q_n[m-1] \sin(q_n)$ . As given by Equations (8) and (10), the iterative calculation of  $I_n[m]$  and  $Q_n[m]$  requires the previous values  $I_n[m-1]$  and  $Q_n[m-1]$ , respectively. A suitable datapath for generating the  $n$ -th in-phase oscillator,  $I_n[m]$ , is shown in Fig. 19. Note that  $N$  instances of this datapath can be used in parallel to generate the  $N$  oscillators for the in-phase component of  $c[m]$ . It is important to note that for  $n = 1 \dots, N$ , the values of  $I_n[1]$ ,  $C_n[1]$ ,  $Q_n[1]$  and  $D_n[1]$  must be initialized. In addition the maximum Doppler frequency of every low-frequency oscillator must be initialized.

Fig. 20 plots the envelope PDF of  $10^6$  generated fading variates with three different word length precisions used to compute Equation (7). It is evident that 16-bit precision is not acceptable while 20-bit precision provides much improved accuracy. Compared to the 16-bit precision required in the previous implementation scheme, the higher precision is required

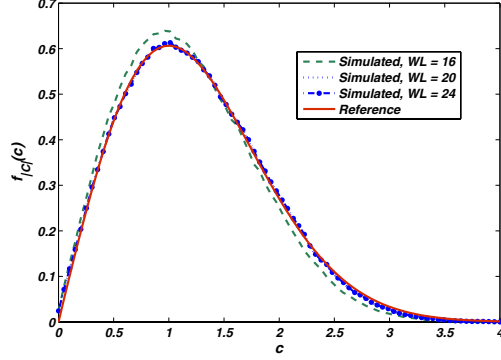


Fig. 20. The PDF calculated by averaging over 100 frames of  $10^4$  fading samples with  $f_D T_s = 0.01$ .

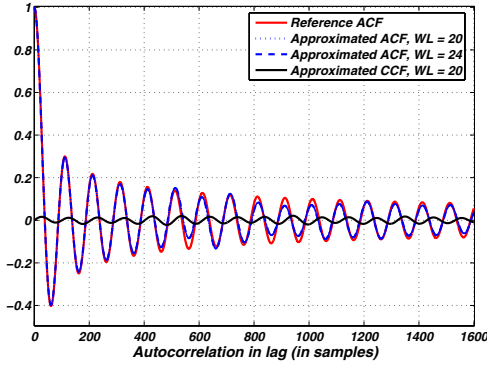


Fig. 21. The ACF calculated by averaging over 100 frames of  $10^4$  fading samples with  $f_D T_s = 0.01$ .

in this model to offset the accumulation of quantization error when evaluating iterative Equations (8) and (10). Fig. 21 shows the ACF of the in-phase component of the iterative model for two different precisions when computing Equation (7). Fig. 22 shows the close match of the approximated LCR with a  $WL = 20$  bit datapath to the reference LCR.

Due to the iterative structure of the  $I_n[m]$  computation, the right hand side of Equation (8) must be computed in each iteration before the next iteration starts. The latency of each iteration is then limited by  $t_m + t_a$ , where  $t_m$  is the delay of the critical path through the multiplier and  $t_a$  is the delay of critical path through the adder. If the implementation on the Xilinx Virtex-II Pro XC2VP100-6 FPGA uses a 20-bit datapath, then  $t_m + t_a$  is about 10 ns and thus the channel simulators is capable of generating 100 million fading variates per second. To speed up the computation, we could use a  $k$ -stage pipelined datapath. Even though a  $k$ -stage pipelined datapath would operate at a higher clock frequency than a non-pipelined circuit, due to the iterative behaviour of the computation, the delay introduced by  $k$  pipeline registers reduces the fading variate generation rate by a factor  $k$ . For example, a four-stage pipelined version of the datapath in Fig. 19 operates at 227 MHz, and the throughput is  $227/4 = 56$  million fading variates per second. To avoid throughput reduction with the pipelined datapath, we can combine more

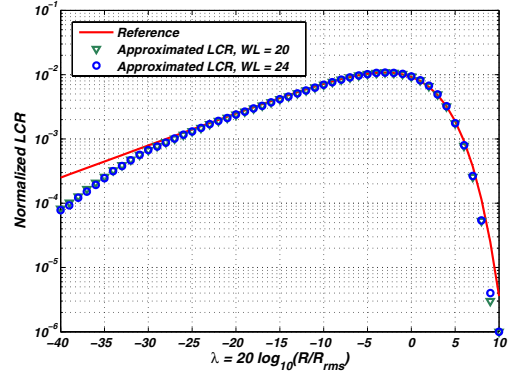


Fig. 22. The LCR calculated by averaging over 100 frames of  $10^4$  fading samples with  $f_D T_s = 0.01$ .

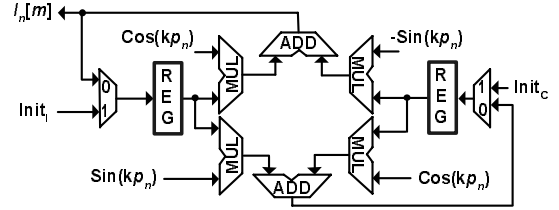


Fig. 23. Pipelined datapath for generating one low-frequency oscillator.

than one iteration into a single stage. Using Equations (8) and (9), the expressions for  $I_n[1]$  and  $C_n[1]$  can be re-written as:

$$\begin{aligned} I_n[1] &= I_n[0] \cos(p_n) - C_n[0] \sin(p_n) \\ C_n[1] &= C_n[0] \cos(p_n) + I_n[0] \sin(p_n) \end{aligned} \quad (11)$$

$I_n[2]$  and  $C_n[2]$  can be written as:

$$\begin{aligned} I_n[2] &= I_n[1] \cos(p_n) - C_n[1] \sin(p_n) \\ C_n[2] &= C_n[1] \cos(p_n) + I_n[1] \sin(p_n) \end{aligned} \quad (12)$$

Equations (11) can be substituted into Equations (12) to give:

$$\begin{aligned} I_n[2] &= I_n[0] \cos(2p_n) - C_n[0] \sin(2p_n) \\ C_n[2] &= C_n[0] \cos(2p_n) + I_n[0] \sin(2p_n) \end{aligned} \quad (13)$$

In general,  $I_n[k]$  and  $C_n[k]$  can be calculated as:

$$\begin{aligned} I_n[k] &= I_n[0] \cos(kp_n) - C_n[0] \sin(kp_n) \\ C_n[k] &= C_n[0] \cos(kp_n) + I_n[0] \sin(kp_n) \end{aligned} \quad (14)$$

Equations (14) simplify the implementation of a  $k$ -stage pipelined datapath. To avoid unused “bubble” cycles in the pipeline, the  $\cos(p_n)$  and  $\sin(p_n)$  constant values should be replaced with  $\cos(kp_n)$  and  $\sin(kp_n)$ , respectively. Also, the first  $k$  initial values of  $I_n[m]$  and  $C_n[m]$ , for  $m = 1, \dots, k$ , should be loaded into the “ $I[m]$ ” and “ $C[m]$ ” registers in  $k$  successive clock cycles, using two multiplexers and external inputs  $Init_i$  and  $Init_c$ , respectively, as shown in Fig. 23. The datapath shown in Fig. 24(a) adds two of the in-phase components of  $c_i[m]$  given in Equation (7). The blocks labeled “ $Calc. I_0$ ” denote separate instances of the circuit in Fig. 23. Then  $N = 8$  in-phase components of  $c_i[m]$  are summed using the tree-structured datapath shown in Fig. 24(b). The datapaths

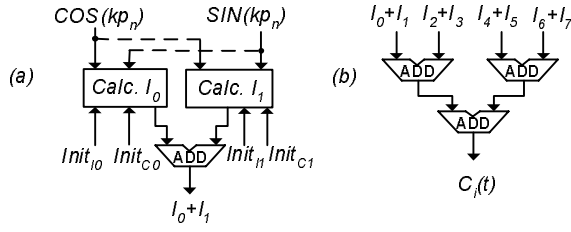


Fig. 24. (a) Circuit for summing two complex oscillators. (b) Tree-structured adder for summing  $N = 8$  oscillators.

TABLE II  
IMPLEMENTATION OF THE SOS-BASED FADING SIMULATOR USING THE  
ITERATIVE SCHEME ON DIFFERENT FPGAS

Device	I <sup>a</sup>	II	III
Clock freq. (MHz)	235.3	210.3	110.03
Output rate (MSamps/sec)	235	210	110
Number of slices	8814 (35%)	8814 (19%)	5830 (7%)
Dedicated resources <sup>b</sup>	256(50%)	256(57%)	176(100%)

<sup>a</sup>Design I was synthesized for a Xilinx Virtex4 XC4VSX55-11 FPGA. Design II was synthesized for a Xilinx Virtex-II Pro XC2VP100-6 FPGA. Finally, Design III was synthesized for an Altera Stratix EP1S80F1508C6 FPGA.

<sup>b</sup>Xilinx Virtex4 FPGAs use dedicated DSP48 blocks, Xilinx Virtex-II Pro FPGAs use dedicated  $18 \times 18$ -bit multipliers, and Stratix FPGAs use 9-bit DSP blocks.

for calculating the quadrature component of  $c(t)$  are similar to the ones used for its in-phase component.

Table II summarizes the implementation characteristics for the SOS-based fading channel simulator, with  $N = 8$ , on three different FPGAs. The datapath in Fig. 23 was implemented in 20-bit fixed-point format while the adder tree was implemented using 16-bit adders. The synthesized layout of this fading channel simulator in a 90-nm CMOS technology is  $967 \mu m^2$ , when the core was targeted to operate at 500 MHz, generating 500 million complex fading variables per second. The core dissipates 185.26 mW of dynamic power and 114.4 mW of static power. Compared to the previous memory-based implementation of SOS fading channel simulators, this approach requires a large number of multipliers and substantially more configurable slices on the same FPGA, while it generates slightly more fading variates per second.

## VI. CONCLUSIONS

Modeling and implementation aspects of two commonly used approaches for the simulation of multipath fading channels, called the sum-of-sinusoids (SOS) based scheme and the filter-based technique, were each considered in turn. The filter-based fading channel simulators show significantly more accurate statistical properties over the SOS-based fading simulators but they require more computational resources and operate more slowly. The SOS model is more efficient for hardware implementation (as it is faster and smaller) and provides a close match with the theoretical statistical properties, but only when the statistics are averaged over a ensemble of fading channels.

An especially efficient implementation of a filter-based fading channel simulator on a single FPGA device was

proposed. The fixed-point implementation utilizes only 3% of the configurable slices, 10% of the dedicated multipliers, and 1% of the available memories on a Xilinx Virtex-II Pro XC2VP100-6 FPGA, while generating arbitrarily long sequences of accurately distributed fading variates. Two different implementation schemes for compact implementations of the SOS-based fading simulators, namely memory-based method and iterative approach, were presented. While the iterative scheme requires substantially more resources than the memory-based implementation, it generates slightly more fading variates per second. Thus, for an FPGA realization, the memory-based method is the more efficient scheme. A typical memory-based implementation of the SOS-based fading channel simulator uses only 1% of the configurable slices and 1% of the available memories on a Xilinx Virtex-II Pro XC2VP100-6 FPGA while generating over 200 million complex Rayleigh fading variates per second. Both schemes are compact enough to be implementable on the same FPGA as the baseband processor during prototype evaluation. These parameterized mobile channel simulators can be reconfigured to accurately simulate a wide variety of different channel characteristics.

## REFERENCES

- [1] P. Bello, "Characterization of randomly time-variant linear channels," *IEEE Trans. Commun.*, vol. 11, no. 4, pp. 360–393, 1963.
- [2] R. H. Clarke, "A statistical theory of mobile-radio reception," *Bell System Technical Journal*, vol. 47, pp. 957–1000, 1968.
- [3] M. J. Gans, "A power spectral theory of propagation in the mobile radio environment," *IEEE Trans. Veh. Technol.*, vol. 21, no. 1, pp. 27–38, 1972.
- [4] W. C. Jakes, *Microwave Mobile Communications*. Piscataway, NJ: Wiley-IEEE Press, 1974.
- [5] J. I. Smith, "A computer generated multipath fading simulation for mobile radio," *IEEE Trans. Veh. Technol.*, vol. 24, no. 3, pp. 39–40, August 1975.
- [6] P. Hoeher, "A statistical discrete-time model for the WSSUS multipath channel," *IEEE Trans. Veh. Technol.*, vol. 41, pp. 461–468, 1992.
- [7] P. Dent, G. E. Bottomley, and T. Croft, "Jakes fading model revisited," *Electronics Letters*, vol. 29, no. 13, pp. 1162–1163, June 1993.
- [8] 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.
- [9] M. F. Pop and N. C. Beaulieu, "Limitations of sum-of-sinusoids fading channel simulators," *IEEE Trans. Commun.*, vol. 49, pp. 699–708, 2001.
- [10] K. E. Baddour and N. C. Beaulieu, "Autoregressive models for fading channel simulation," in *Proceedings of the IEEE Global Telecommunications Conference*, 2001, pp. 1187–1192.
- [11] 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.
- [12] C. Xiao, Y. R. Zheng, and N. Beaulieu, "Statistical simulation models for Rayleigh and Rician fading," in *Proceedings of the IEEE Intl. Conference on Communications*, 2003, pp. 3524–3529.
- [13] Y. R. Zheng and C. Xiao, "Simulation models with correct statistical properties for Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 51, pp. 920–928, 2003.
- [14] E. Casas and C. Leung, "A simple digital fading simulator for mobile radio," *IEEE Trans. Veh. Technol.*, vol. 39, no. 3, pp. 205–212, August 1990.
- [15] P. J. Cullen, P. C. Fannin, and S. S. Swords, "DSP implementation of a wideband frequency selective fading simulator," in *Intl. Conference on Antennas and Propagation*, 1991, pp. 492 – 495.
- [16] J. F. An, A. M. D. Turkmani, and J. D. Parsons, "Implementation of a DSP-based frequency non-selective fading simulator," in *Intl. Conference on Radio Receivers and Associated Systems*, 1990, pp. 20–24.
- [17] M. Lecours and F. Marceau, "Design and implementation of a channel simulator for wideband mobile radio transmission," in *IEEE Veh. Tech. Conf.*, 1989, pp. 652–655.



- [18] A. K. Salkintzis, "Implementation of a digital wide-band mobile channel simulator," *IEEE Trans. Broadcast.*, vol. 45, no. 1, pp. 122–128, 1999.
- [19] 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.
- [20] D. Derrien and E. Boutillon, "Quality measurement of a colored Gaussian noise generator hardware implementation based on statistical properties," in *Proceedings of the IEEE Intl. Symposium on Signal Processing and Information Technology*, 2002.
- [21] G. Arredondo, W. Chriss, and E. Walker, "A multipath fading simulator for mobile radio," *IEEE Trans. Commun.*, vol. 21, no. 11, pp. 1325–1328, November 1973.
- [22] R. N. Kolte, S. C. Kwatra, and G. H. Stevens, "Computer controlled hardware simulation of fading channel models," in *Proceedings of the IEEE Intl. Conference on Communications*, 1988, pp. 1646–1650.
- [23] A. Verschoor, A. Kegel, and J. C. Arnbak, "Hardware fading simulator for a number of narrowband channels with controllable mutual correlation," *Electronics Letters*, vol. 24, no. 22, pp. 1367–1369, 1988.
- [24] Z. Yansheng, "A Rayleigh fading simulator for mobile radio," in *Proceedings of the IEEE Conference on Computer, Communication, Control and Power Engineering*, 1993, pp. 143–145.
- [25] P. Cullen, P. Fannin, and A. Garvey, "Real-time simulation of randomly time-variant linear systems: the mobile radio channel," *IEEE Trans. Instrum. Meas.*, vol. 43, no. 4, pp. 583–591, 1994.
- [26] NoiseCom MP-2500 Multipath Fading Emulator, Technical Manual, NoiseCom, Paramus, NJ, 1996.
- [27] M. Rupp, A. Burg, and E. Beck, "Rapid prototyping for wireless designs: the five-ones approach," *Intl. European Association for Signal Processing Journal*, vol. 83, no. 7, pp. 1427–1444, July 2003.
- [28] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall, 2002.
- [29] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 2002.
- [30] 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.
- [31] R. N. Kolte, S. C. Kwatra, and G. H. Stevens, "Computer controlled hardware simulation of fading channel models," in *Proceedings of the IEEE Intl. Conference on Communications*, 1998, pp. 1646–1650.
- [32] G. L. Stüber, *Principles of Mobile Communication*. New York: Kluwer Academic Publishers, 2001.
- [33] D. J. Young and N. C. Beaulieu, "The generation of correlated Rayleigh random variates by inverse Fourier transform," *IEEE Trans. Commun.*, vol. 48, pp. 1114–1127, 2000.
- [34] C. Kominakis, "A fast and accurate Rayleigh fading simulator," in *Proceedings of the IEEE Global Telecommunications Conference*, 2003, pp. 3306–3310.
- [35] A. Anastasopoulos and A. Chugg, "An efficient method for simulation of frequency selective isotropic Rayleigh fading," in *Proceedings of the IEEE Veh. Tech. Conf.*, 1997, pp. 2084–2088.
- [36] S. O. Rice, "Mathematical analysis of random noise," *Bell System Technical Journal*, vol. 23, pp. 282–332, 1944.
- [37] *Baseband Studio for Fading*, Technical Overview, N5115A, Agilent Technologies Inc., 2005.
- [38] *Baseband Fading Simulator ABFS, Reduced costs through baseband simulation*, no. 163, Rohde & Schwarz, 1999.
- [39] *Full Featured Channel Simulator*, Technical Manual, SIMSTAR-1, Ascom, 2002.
- [40] *Wireless Channel Emulator*, Technical Document, SR5500, Spirent Communications, 2006.
- [41] *Multipath Fading Simulator*, Product Manual, NJZ-1600B, Japan Radio Co., 2005.
- [42] M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, *Simulation of Communication Systems: Modeling, Methodology, and Techniques*. New York: Kluwer Academic Publishers, 2000.
- [43] L. B. Jackson, *Digital Filters and Signal Processing*. Kluwer Academic Publishers, 1989.
- [44] R. A. Goubran, H. M. Hafez, and A. U. H. Sheikh, "Implementation of a real-time mobile channel simulator using a DSP chip," *IEEE Trans. Instrum. Meas.*, vol. 40, no. 4, pp. 709–714, 1991.
- [45] M. S. Lim and H. K. Park, "The implementation of the mobile channel simulator in the baseband and its application to the quadrature type GMSK modem design," in *IEEE Veh. Tech. Conf.*, 1990, pp. 469–500.
- [46] G. J. R. Povey, P. M. Grant, and R. D. Pringle, "A decision-directed spread-spectrum RAKE receiver for fast-fading mobile channels," *IEEE Trans. Veh. Technol.*, vol. 45, no. 3, pp. 491–502, November 1996.
- [47] A. Stéphenne and B. Champagne, "Effective multi-path vector channel simulator for antenna array systems," *IEEE Trans. Veh. Technol.*, vol. 49, no. 6, pp. 2370–2381, 2000.
- [48] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete Time Signal Processing*. Prentice Hall, 1999.
- [49] W. C. Jakes, *Microwave Mobile Communications*. Piscataway, NJ: Wiley-IEEE Press, 1994.
- [50] D. C. Cox, "910 MHz urban mobile radio propagation: multipath characteristics in New York city," *IEEE Trans. Veh. Technol.*, vol. 22, no. 4, pp. 104–109, 1973.
- [51] *Filter Design Toolbox For Use with Matlab, User's Guide*, The Mathworks, 2005.
- [52] A. Alimohammad, S. F. Fard, B. F. Cockburn, and C. Schlegel, "A compact and accurate Gaussian variate generator," to appear in *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*
- [53] M. Pätzold and F. Laue, "Statistical properties of Jakes' fading channel simulator," in *Proceedings of the IEEE Veh. Tech. Conf.*, 1998, pp. 712–718.
- [54] M. Pätzold, R. Garcia, and F. Laue, "Design of high-speed simulation models for mobile fading channels by using table look-up techniques," *IEEE Trans. Veh. Technol.*, vol. 49, no. 4, pp. 1178–1190, July 2000.
- [55] Y. Li and X. Huang, "The simulation of independent Rayleigh faders," *IEEE Trans. Commun.*, vol. 50, no. 9, pp. 1503–1514, 2002.
- [56] C. A. G. de Leon, M. C. Bean, and J. S. Garcia, "Generation of correlated Rayleigh-fading envelopes for simulating the variant behavior of indoor radio propagation channels," in *Proceedings of the IEEE Veh. Tech. Conf.*, 2004, pp. 4245–4249.
- [57] C. Xiao, Y. R. Zheng, and N. C. Beaulieu, "Second-order statistical properties of the WSS Jakes' fading channel simulator," *IEEE Trans. Commun.*, vol. 50, no. 6, pp. 888–891, June 2002.
- [58] C.-X. Wang and M. Pätzold, "Methods of generating multiple uncorrelated Rayleigh fading processes," in *Proceedings of the IEEE Semiannual Vehicular Tech. Conf.*, 2003, pp. 510–514.
- [59] M. Cui, H. Murata, and K. Araki, "FPGA implementation of 4 × 4 MIMO test-bed for spatial multiplexing systems," in *IEEE PIMRC*, 2004, pp. 3045–3048.
- [60] A. Alimohammad, S. F. Fard, B. F. Cockburn, and C. Schlegel, "An improved SOS-based fading channel emulator," in *IEEE Veh. Tech. Conf.*, 2007, pp. 931–935.
- [61] P. L'Ecuyer, "Tables of maximally equidistributed combined LFSR generators," *Math. of Comp.*, vol. 68, no. 225, pp. 261–269, 1999.
- [62] J. Volder, "The CORDIC trigonometric computing technique," *IRE Trans. on Electronic Computers*, vol. EC-8, no. 3, p. 3.4.1, 1959.
- [63] F. de Dinechin and A. Tisserand, "Multipartite table methods," *IEEE Trans. Comput.*, vol. 54, no. 3, pp. 319–330, 2005.



**Amirhossein Alimohammad** is a postdoctoral fellow at the University of Alberta. He received his Ph.D. from the University of Alberta, his M.Sc. from the University of Tehran, and his B.Sc. from the University of Isfahan. In 2001, Amir was with the Electronic and Microelectronic Department at the University of Ulm, Atmel Germany GmbH, and Get2Chip GmbH in Munich, Germany. He was with Informatics Services Corporation (ISC) between 1998–2000 while he was a Ph.D. student at the Sharif University of Technology. His research

interests include reconfigurable computing, VLSI design and test, wireless communications, and signal processing.



**Bruce F. Cockburn** (S'86–M'90) received the B.Sc. degree in engineering physics from Queen's University, Kingston, ON, Canada in 1981, and the M.Math. and Ph.D. degrees in computer science from the University of Waterloo, Waterloo, ON, Canada, in 1985 and 1990, respectively.

He is currently an Associate Professor in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. His research interests include VLSI design and test, parallel signal processing, iterative decoders, and

computer architecture.

Dr. Cockburn is a member of the IEEE Computer Society, the IEEE Communications Society, the IEEE Solid-State Circuits Society, and the Association for Computing Machinery. He is also registered as a Professional Engineer in the Province of Alberta, Canada.