

Single FPGA Simulator for Geometric MIMO Fading Channel Models

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

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

Abstract— We propose a compact and fast hardware channel simulator for the baseband verification and prototyping of multiple-input multiple-output (MIMO) wireless communication systems. The simulator accurately and efficiently implements models of both single-bounce and multiple-bounce geometric propagation conditions. Fading samples are generated at a low rate, comparable to the Doppler frequency, and then interpolated to match the desired sample rate. Bit-true simulations verify the accuracy of the implemented simulator. As an example, when implemented on a Xilinx XC5VLX110 field-programmable gate array (FPGA), our 4×4 MIMO geometric fading channel simulator occupies only 6.6% of the configurable slices while generating more than 16×324 million samples per second. The geometric MIMO fading channel simulator is well suited for use in a FPGA-based error rate performance verification system.

I. INTRODUCTION

Emerging wireless communication standards, such as long-term evolution (LTE) [1], target higher data rates, more efficient spectrum utilization and improved user services. Multiple-input-multiple-output (MIMO) wireless communication technology has been included in all recent broadband wireless standards to increase the data rates and spectrum efficiency over traditional single-antenna transmission. To keep up with the fast evolution of communication technologies and wireless standards, the test system must be controllable and configurable to provide a repeatable test environment in the laboratory. Moreover, for rapid system design and verification and to reduce product time-to-market, hardware-accelerated performance testing is increasingly desirable compared to conventional software simulations.

An accurate and high-throughput fading channel simulator is a crucial component for the baseband design and verification of wireless communication systems. Ensuring product performance in the field by using laboratory channel emulation can give a competitive edge to companies. Commercially-available MIMO fading channel simulators (e.g., [2]–[4]) are standalone units that are relatively costly and cannot be prototyped on a single device with the rest of the system. Various hardware-based fading channel simulators using heterogeneous architectures – including general-purpose processors (GPPs), digital signal processors (DSPs), and field-programmable gate array (FPGAs) – have been proposed (see for example [5] and [6]). FPGA technology has been widely adopted to speed up the prototyping and verification of new digital communication in-

frastructure [7]–[9]. Ideally, the entire communication system chain (i.e., the transmitter, the fading channel simulator, and the receiver) should be prototyped completely on a single device. Such a prototype avoids the need for custom interfaces that connect test equipment to the device under test. Several compact MIMO fading simulators that can be implemented on a small fraction of a single FPGA have been proposed in the literature [10]–[14]. Unfortunately, these simulators are limited to simplistic analytical MIMO channel models, such as the independent and identically-distributed (i.i.d) model [15]. In contrast to the analytical models, physical channel models [15] can be parameterized to accurately mimic wave propagation effects. Physical channel models characterize an environment by describing multipath propagation between the transmitter (TX) and receiver (RX) antennas using geometric parameters of scatterers.

In this work we consider the simulation of geometric channel models in which the fading effects are characterized by the relative location of scatterers on the propagation path. Geometric fading models are especially useful for describing the fading behavior in site-specific scenarios as opposed to averaged conditions. We propose a compact and high-throughput hardware-based geometric fading channel simulator for the rapid prototyping and verification of MIMO systems. Our new single-FPGA implementation of the fading simulator mimics various physical propagation models for different geometric distributions of scatterers. To the best of our knowledge, the proposed versatile simulator is the first single-FPGA geometric fading channel simulator. The channel simulator is compact enough that it can be implemented along with the transmitter and the receiver of a typical MIMO communication system on a single FPGA.

The rest of this article is organized as follows. Section II reviews the narrowband geometric MIMO fading channel models. Section III briefly explains the extension to wideband geometric MIMO fading channels. Section IV presents the compact architecture of the proposed geometric fading channel simulator. Simulation and implementation results are presented in Section V. Section VI describes the use of a geometric fading channel simulator in a single-FPGA error rate measurement system. Finally, Section VII makes some concluding remarks.

II. GEOMETRIC MIMO FADING CHANNEL MODELS

Fig. 1 shows the geometrical representation of a MIMO channel along with the line-of-sight (LOS) paths. We denote the antenna spacing at the TX for antenna elements p and q as δ_{pq} and the antenna spacing at the RX for antenna elements l and m as d_{lm} . The variable ζ_{lp} denotes the distance between the p^{th} TX antenna and the l^{th} RX antenna, ω_p is the approximate direction of the direct path from the p^{th} TX antenna, and γ denotes the direction of RX motion.

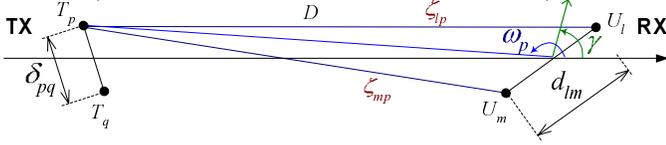


Fig. 1. Geometrical representation of a MIMO channel and the LOS paths.

Here we consider an $N \times M$ MIMO system, where $M > 1$ and $N > 1$ are the numbers of TX and RX antennas, respectively. Thus the baseband input-output relation between the length- M transmit signal vector $\mathbf{s}(t)$ and the length- N receive signal vector $\mathbf{y}(t)$ in discrete time (i.e., $t = nT_s$, where T_s is the symbol period and n is a non-negative integer symbol index) is

$$\mathbf{y}[n] = \mathbf{H}[n] \mathbf{s}[n] + \mathbf{n}[n], \quad (1)$$

where the length- N vector $\mathbf{n}[n]$ represents both additive noise and interference. In equation (1), it is assumed that the MIMO channel is narrow-band, linear, and time-variant and can be fully represented by the $N \times M$ channel matrix $\mathbf{H}[n] = \{h_{lp}[n]\}$. Here each element $h_{lp}[n]$ denotes the time-variant impulse response between the p^{th} TX antenna and the l^{th} RX antenna, called the *channel gain* henceforth.

The channel gains $h_{lp}[n]$ can be derived based on the physical model illustrated in Fig. 1. In the presence of a LOS component, the channel gain $h_{lp}[n]$ between the p^{th} TX antenna and the l^{th} RX antenna is

$$h_{lp}[n] = h_{lp}^{DIF}[n] + h_{lp}^{LOS}[n], \quad (2)$$

where $h_{lp}^{DIF}[n]$ represents the contributions of the diffuse (i.e., scattered) components and $h_{lp}^{LOS}[n]$ denotes the contribution of the LOS component [16]. Assuming that the distance D between the RX and TX is much larger than the antenna spacing at the TX and RX, i.e., $D \gg \max(\delta_{pq}, d_{lm})$, the contribution of the LOS component to the channel gain can be approximated as [16]

$$h_{lp}^{LOS}[n] = \sqrt{\frac{K_{lp}}{K_{lp} + 1}} \times \exp \left\{ j \left(2\pi f_D n \cos(\omega_p - \gamma) - \frac{2\pi \zeta_{lp}}{\lambda} \right) \right\}, \quad (3)$$

where K_{lp} denotes the Rice factor of the link between the p^{th} TX antenna and the l^{th} RX antenna, which is the ratio of the LOS component power to the diffuse component power. In addition f_D is the maximum Doppler frequency, λ is the wavelength and $j^2 = -1$.

When only single-bounce scattering happens on the propagation path, as shown in Fig. 2, the contribution of the diffuse component can be written as [16]

$$h_{lp}^{DIF}[n] = \frac{1}{\sqrt{N_S}} \sum_{i=1}^{N_S} \frac{g_i \exp \{ j(2\pi f_i n + v_{lpi} + \theta_i) \}}{\sqrt{K_{lp} + 1}}, \quad (4)$$

where N_S is the number of independent scatterers, g_i is the amplitude of the received wave from the i^{th} scatterer which is normalized such that $\sum_{i=1}^{N_S} g_i^2 = N_S$. Also, θ_i denotes the phase shift introduced by the i^{th} scatterer, $v_{lpi} = -2\pi(\xi_{ip} + \xi_{li})/\lambda$ is the phase delay due to propagation, and ξ_{ip} and ξ_{li} are the distances of the i^{th} scatterer from the p^{th} TX antenna to the l^{th} RX antenna, respectively. Finally, $f_i = f_D \cos(\omega_i - \gamma)$ and ω_i are the Doppler frequency and the direction-of-arrival observed from the i^{th} scatterer, respectively.

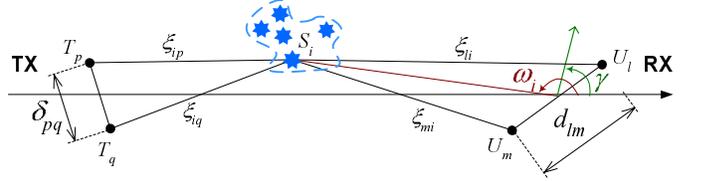


Fig. 2. Geometrical representation of a single-bounce MIMO channel. Diffuse paths from one group of scatterers.

Generalization of single-bounce scattering to multiple-bounce scattering is straightforward. Fig. 3 shows a case of double-bounce scattering in a 2×2 MIMO channel. For this case, the contribution of the diffuse component can be written as

$$h_{lp}^{DIF-2B}[n] = \frac{1}{\sqrt{N_{S1} N_{S2}}} \sum_{i=1}^{N_{S1}} \sum_{k=1}^{N_{S2}} \left[\tilde{g}_{ik} \times \frac{\exp \{ j(2\pi f_i n + v_{lpik} + \theta_{ik}) \}}{\sqrt{K_{lp} + 1}} \right], \quad (5)$$

where N_{S1} and N_{S2} are the number of independent scatterers for first- and second-bounce scattering, respectively. Note that \tilde{g}_{ik} is the amplitude of the received wave through the k^{th} first bounce and i^{th} second bounce scatterer and $\sum_{i=1}^{N_{S1}} \sum_{k=1}^{N_{S2}} \tilde{g}_{ik}^2 = N_{S1} N_{S2}$. Similar to the single-bounce case, θ_{ik} is the random phase shift introduced by the k^{th} first-bounce and i^{th} second-bounce scatterer pairs and $v_{lpik} = -2\pi(\xi_{kp} + \xi_{ik} + \xi_{li})/\lambda$ is the phase delay, where ξ_{kp} is the distance from the p^{th} TX antenna to the k^{th} first-bounce scatterer, ξ_{ik} denotes the distance between the k^{th} first-bounce and the i^{th} second-bounce scatterers, and ξ_{li} is the distance from the i^{th} second-bounce scatterer to the l^{th} RX antenna. Note that only the angle-of-arrival from the second-bounce scatterer affects the observed Doppler frequency.

Comparing equations (4) and (5) we can see that in both cases, the contribution of the diffuse component to the channel gain is calculated by superimposing a number $N_{S1} N_{S2}$ of complex sine waves that are characterized by their amplitude \tilde{g}_{ik} , random phase shift θ_{ik} , phase delay v_{lpik} , and Doppler

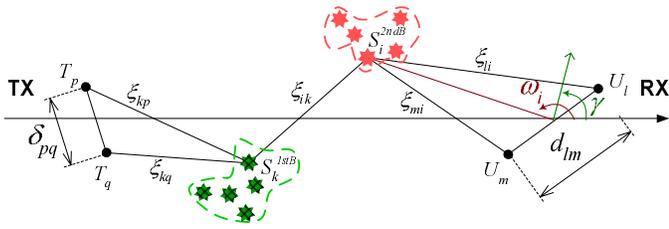


Fig. 3. Geometrical representation of a double-bounce MIMO channel. Diffuse paths from two groups of scatterers.

frequency f_i . In fact, for scattering involving two or more bounces, we can always reformulate equation (5) to have only one summation over all possible paths from the p^{th} TX antenna to the l^{th} RX antenna, hence recreating the same form as equation (4). Due to the similarities between these two systems, when we explain our hardware implementation in Section IV, we will use the simpler equation (4), and we will assume that simulation of double-bounce (and multi-bounce) propagation models is possible with the same hardware.

III. SIMULATION OF WIDEBAND GEOMETRIC MIMO FADING CHANNELS

Different scatterer distribution models have been proposed in the literature. Among the most well known geometric channel models in the literature are: (1) the one-ring model [16], (2) the double-bounce two-ring model [17], and (3) the single-bounce elliptical channel model [18]. The one-ring model assumes that the scatterers are located on a ring around the RX. This model would be appropriate for a downlink connection where the base-station is elevated and thus not obstructed by local scatterers while the mobile-station is surrounded by local scatterers. In the double-bounce fading channels with two-ring scattering, the transmitter and the local scatterers around the transmitter and receiver are stationary while the receiver is moving relative to the rest of the system. Note that in both one-ring and the two-ring channel models it is assumed that any difference between the propagation delays from each TX antenna to each RX antenna is negligible (compared to signal sample period). While this approximation can be useful when modeling narrowband channels, the difference between propagation delays must be considered when simulating wideband channels. Unlike the one-ring and the two-ring models, the elliptical model can be used for simulating wideband channels as well as narrowband channels. In the elliptical model, each ellipsoid represents the scatterers that result in a specific propagation delay.

The underlying difference between narrowband and wideband fading channels is in the characteristics of their propagation delays. While the difference between propagation delays in narrowband channels is negligible, in wideband channels, the difference between propagation delays can be considerably larger. Fig. 4 shows the geometrical representation of a wideband MIMO fading channel. As shown in Fig. 4, frequency-selective wideband fading channels can be modeled as multiple concentric ellipsoids of scatterers, each representing a specific

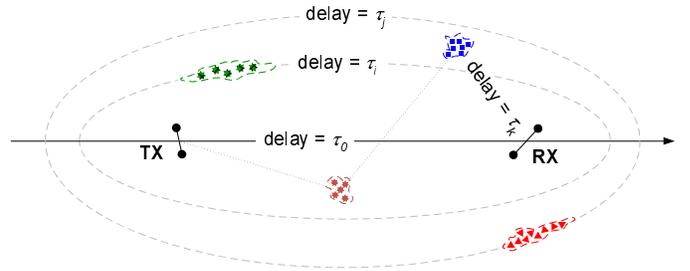


Fig. 4. Geometrical representation of a wideband MIMO fading channel.

propagation delay between the TX and RX antennas arrays. In this figure, the shortest propagation path, corresponding to delay τ_0 , models the LOS path. Note that for each propagation delay $\tau_i > \tau_0$, all of the single-bounce scatterers can be associated with an ellipsoid whose focal points are centered on the TX and RX. For the incoming waves after multiple-bounce scattering, however, no generalizations can be made about the distribution of scatterers.

For a given “normalized” propagation delay $m = \tau/T_s$, the complex channel gain between the p^{th} TX antenna and the l^{th} RX antenna, $h_{lp,m}[n]$, is composed of the contributions of the LOS component (when $\tau = \tau_0$ see equation (3)), the single-bounce scatterers (see equation (4)) and the multiple-bounce scatterers (see equation (5)) with the same normalized propagation delay m . Using this notation, the wideband channel impulse response between the p^{th} TX antenna and the l^{th} RX antenna can be expressed as

$$h_{lp}^{WB}[n, \tilde{m}] = \sum_{\ell=0}^{M-1} h_{lp,m_\ell}[n] \delta[\tilde{m} - m_\ell], \quad (6)$$

where M is the number of taps (discrete propagation paths) and $\delta[n] = 1$ for $n = 0$ and $\delta[n] = 0$ for $n \neq 0$. In equation (6), for simplicity of presentation, we assume a tapped delay channel model [19]. In other words, we assumed that the discrete delays τ_i are an integer multiple of the symbol period, i.e. $\tau_i = m_i T_s$. However, extension of this model to the general case using fractional delay sampling is straightforward (see for example [20]). Generated fading coefficients, $h_{lp,m_\ell}[n]$, are used in a tapped delay channel model to implement a wideband frequency-selective fading channel. For example, Fig. 5 shows the block diagram of a tapped delay model of a wideband fading subchannel between the p^{th} TX antenna and the l^{th} RX antenna, where each path can have a different delay.

IV. HARDWARE IMPLEMENTATION

In a typical wireless communication scenario, the maximum Doppler frequency f_D is significantly smaller than the signal sample rate $F_s = 1/T_s$. This allows us to design much of the fading simulator at a significantly lower sample rate and thereby reduce the required hardware resources by using time-multiplexed datapaths. The resulting low-rate signal can then be interpolated to achieve the desired output sample rate. Here we assume that the fading channel gains are generated

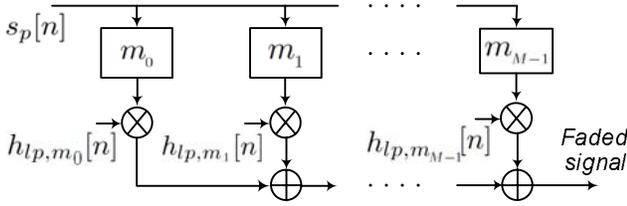


Fig. 5. Tapped delay line model for a wideband fading subchannel between the p^{th} TX antenna and the l^{th} RX antenna.

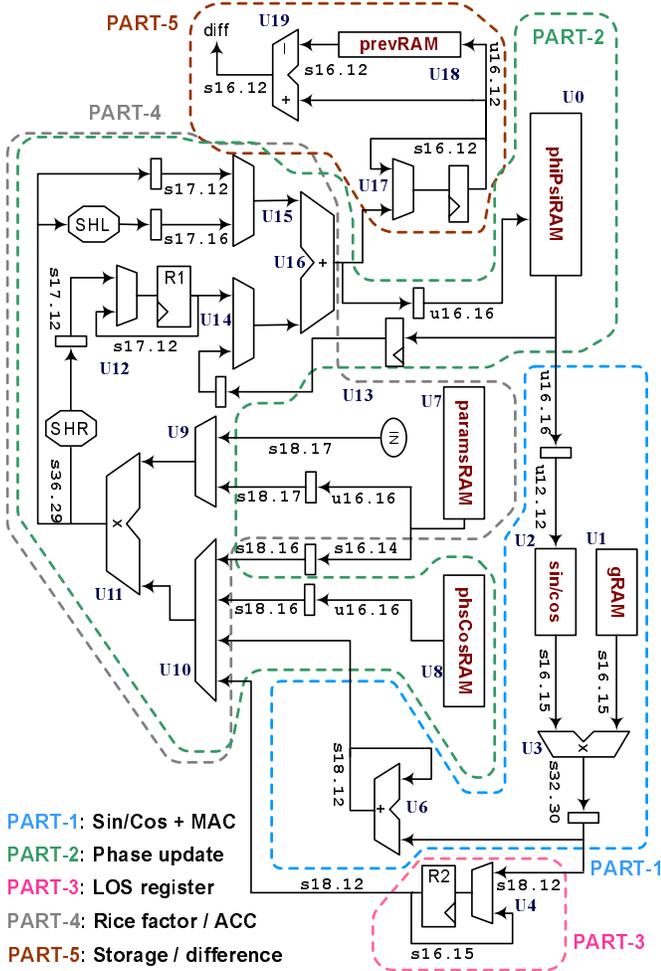


Fig. 6. Datapath of the proposed MIMO fading samples generator.

at $\hat{F}_s = F_s/I$ samples per second and then up-sampled by the interpolation factor I . In the following we explain how our proposed hardware datapath generates the fading samples according to equations (2), (3) and (4).

A. Fading Sample Generation

For a compact implementation, in each cycle, the fading gains are generated in two steps. In the first step, the in-phase components of the channel gains are calculated and in the second step, the quadrature components are generated. In addition, the phases are updated in the second step of each cycle (as explained later).

According to equation (4), the contribution of the diffuse components to the channel gain in quadrature form at sample rate \hat{F}_s can be written as

$$\begin{aligned} h_{lp}^{DIF-I}[n] &= C \times T_{lp} \sum_{i=1}^{N_s} g_i \times \text{cs}(\phi(l, p, i, n)), \\ h_{lp}^{DIF-Q}[n] &= C \times T_{lp} \sum_{i=1}^{N_s} g_i \times \text{sn}(\phi(l, p, i, n)), \end{aligned} \quad (7)$$

where $C = 1/\sqrt{N_s}$, $T_{lp} = 1/\sqrt{K_{lp} + 1}$, $\text{cs}(\phi) = \cos(2\pi\phi)$, $\text{sn}(\phi) = \sin(2\pi\phi)$, $\phi(l, p, i, n) = \hat{f}_i \times n + (v_{lpi} + \theta_i)/2\pi$ is the path-phase for each of the received waves from $N_s \geq 1$ different scatterers at any time $n \geq 0$, and $\hat{f}_i = I \times f_i$.

Fig. 6 shows the datapath of the proposed geometric fading channel simulator. For more clarity, this datapath has been divided into five parts. Time-multiplexed sharing of some modules accounts for the overlapping regions among the parts. The modules in PART-1 calculate the summations $\sum_{i=1}^{N_s} g_i \times \text{cs}(\phi(l, p, i, n))$ and $\sum_{i=1}^{N_s} g_i \times \text{sn}(\phi(l, p, i, n))$. Our bit-true simulation shows that the phases $\phi(l, p, i, n)$ need to be stored with at least 16-bit accuracy. Since the values of $\phi(l, p, i, n)$ are limited to the range $[0, 1)$, the phases are stored in U0 phiPsiRAM in u16.16 format, i.e., 16-bit unsigned values with a 16-bit fraction. The stored phases $\phi(l, p, i, n)$ are received from U0 by module U2 where $\text{cs}(\phi(l, p, i, n))$ and $\text{sn}(\phi(l, p, i, n))$ are calculated. The sine/cosine of $\phi(l, p, i, n)$ are both approximated using a look-up table that contains the first quarter cycle of a sine wave. The coefficients $\{g_i\}$ are stored in memory U1 in s16.15 format (i.e., 16-bit signed values with a 15-bit fraction). Multiplier U3 and accumulator U6 calculate $\sum_{i=1}^{N_s} g_i \times \text{cs}(\phi(l, p, i, n))$ and $\sum_{i=1}^{N_s} g_i \times \text{sn}(\phi(l, p, i, n))$.

The path phases $\phi(l, p, i, n)$ are updated recursively as follows:

$$\phi(l, p, i, n) = \begin{cases} \phi(l, p, i, n-1) + \hat{f}_i & \text{for } n > 0, \\ (v_{lpi} + \theta_i)/2\pi & \text{for } n = 0. \end{cases} \quad (8)$$

Note that in (7), $\text{sn}(\phi)$ and $\text{cs}(\phi)$ are sinusoids with period 1, hence the integer part of the path-phase can be discarded when calculating (8) without affecting the results.

In PART-2 in Fig. 6, the initial path-phase values $\phi(l, p, i, 0) = (v_{lpi} + \theta_i)/2\pi$, for $l = 1, \dots, N$, $p = 1, \dots, M$, $i = 1, \dots, N_s$, are stored in memory U0 phiPsiRAM. These phases are updated as discrete time advances. The maximum Doppler frequency f_D is loaded via multiplexer U9 at initialization. As well, the values $\cos(\omega_i - \gamma)$, for $i = 1, \dots, N_s$, are stored in memory U8 phsCosRAM. These two values are multiplied by U11 to obtain the “projected” Doppler frequencies $f_D \times \cos(\omega_i - \gamma)$. As will be explained later, we set the interpolation factor I to be a power of two. The output of U11 is then shifted to obtain $\tilde{f}_i = f_D \times \cos(\omega_i - \gamma)$ which is used to update the path-phases according to equation (8). Further, a copy of the current phase (through multiplexer U14) is added to \tilde{f}_i (through U15 by adder U16). Finally, the updated path-phase value is stored in phiPsiRAM for the next cycle.

Note that the modules in PART-1 and PART-2 operate in parallel. In each cycle, the summations $\sum_{i=1}^{N_s} g_i \times$

$\text{cs}(\phi(l, p, i, n))$ and $\sum_{i=1}^{N_S} g_i \times \text{sn}(\phi(l, p, i, n))$ are calculated in PART-1 for every TX and RX antenna ($p = 1, \dots, M, l = 1, \dots, N$) and for every scatterer ($i = 1, \dots, N_S$). The path-phases $\phi(l, p, i, n)$ are updated simultaneously in PART-2. The operations of these parts thus require $M \times N \times N_S$ clock cycles for each of the in-phase or quadrature components.

The calculated summations from PART-1 are then scaled in PART-4 by $C \times T_{lp}$ to provide the in-phase and quadrature components in equation (7). Note that for a large number N_S of scatterers, the coefficient $C \times T_{lp} = 1/\sqrt{N_S(K_{lp} + 1)}$ is a small number. The scaling is performed using shifting and multiplication operations. More specifically, the coefficient C can be factored as $C = 2^{-K} \times \tilde{C}$, where $K = \lfloor \log_2(1/C) \rfloor$. Defining $\tilde{T}_{lp} = T_{lp} \times \tilde{C}$, we have $C \times T_{lp} = 2^{-K} \times \tilde{T}_{lp}$. The values of \tilde{T}_{lp} for $l = 1, \dots, N$ and $p = 1, \dots, M$ are stored in memory U7 paramsRAM in u16.16 format and are passed to multiplier U11 through multiplexer U9. These values are then used to scale the summations calculated in PART-1. A right-shift operation (K times) after the multiplier U11 finishes the calculation of the diffuse samples in equation (7). Finally, the diffuse samples are stored in register R1 until they are added to the LOS samples by adder U16.

The LOS fading samples are computed in a similar way but requiring only a single complex sinusoid. From equation (3) the LOS samples in quadrature form at sample rate \tilde{F}_s can be written as

$$\begin{aligned} h_{lp}^{LOS-I}[n] &= W_{lp} \times \text{cs}(\psi(l, p, n)), \\ h_{lp}^{LOS-Q}[n] &= W_{lp} \times \text{sn}(\psi(l, p, n)), \end{aligned} \quad (9)$$

where $W_{lp} = \sqrt{K_{lp}/(K_{lp} + 1)}$, $\psi(l, p, n) = \tilde{f}_p \times n + \eta_{lp}$, $\tilde{f}_p = \cos(\omega_p - \gamma) \times f_D$, and $\eta_{lp} = -2\pi\zeta_{lp}/\lambda$. The path-phases $\psi(l, p, n)$ are updated recursively, in a similar way to equation (8) but with a different initialization value $(1 - \eta_{lp}/2\pi)$. The path-phases $\psi(l, p, n)$, for $l = 1, \dots, N$ and $p = 1, \dots, M$, are stored in memory U0 phiPSiRAM and updated in PART-2. In addition, PART-1 calculates $\text{cs}(\psi(l, p, n))$ and $\text{sn}(\psi(l, p, n))$, which are stored in register R2 in PART-3. Also, the values of W_{lp} , for $l = 1, \dots, N$ and $p = 1, \dots, M$, are stored in memory U7 paramsRAM, and then later passed to multiplier U11 through multiplexer U9. This multiplier then calculates the LOS fading sample, according to equation (9). The LOS sample (from U15) is then added to the diffuse sample (from U14) by adder U16 to obtain the final in-phase or quadrature component of the channel gain. Upsampling of the generated fading samples using interpolation is described in the next sub-section.

B. Interpolation

To simplify the hardware implementation, we constrain the interpolation factor I to be a power of 2, i.e., $I = 2^P$. Using a linear interpolator, the interpolated fading samples $\tilde{h}_{lp}[n]$, $n \geq 0$, at the times $n = 2^P k + u$, $k \geq 0$, $u = 0, \dots, 2^P - 1$ can be written as

$$\tilde{h}_{lp}[2^P k + u] = h_{lp}[0] + \sum_{z=0}^{k-1} d_{lp}[z] + 2^{-P} d_{lp}[k]u, \quad (10)$$

where

$$d_{lp}[z] = \begin{cases} h_{lp}[2^P(z+1)] - h_{lp}[2^P z] & \text{for } z \geq 0; \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

denotes the difference between subsequent fading samples at the slower sample rate $\tilde{F}_s = 2^{-P} F_s$. Note that the expression $2^{-P} h_{lp}[k]u$ in (10) can be calculated using shifting and running-sum operations and, therefore, the linear interpolator can be conveniently implemented using a simple accumulator and shifter.

In PART-5 of Fig. 6, the quadrature components of $h_{lp}[z]$ are calculated. The low frequency channel gains are stored in memory U18 prevRAM and the difference between the current sample and the previous sample for each TX and RX pair (see equation (11)) is calculated and passed to the linear interpolator for generating the final channel gains according to (10).

Fig. 7 shows the datapath of the implemented linear quadrature interpolator. As the fading simulator calculates the difference samples `diff` in PART-5, it causes the corresponding interpolator to latch the updated value in the page registers (`idf0`, `idf1`, `qdf0` and `qdf1`). Depending on the state of the `idf` and `qdf` registers, the updated difference is latched into the inactive register for the in-phase or quadrature path. Note that the operation of scaling by 2^{-P} in equation (10) is performed in parallel by the two barrel shifters M10 and M11 for the in-phase and quadrature parts, respectively.

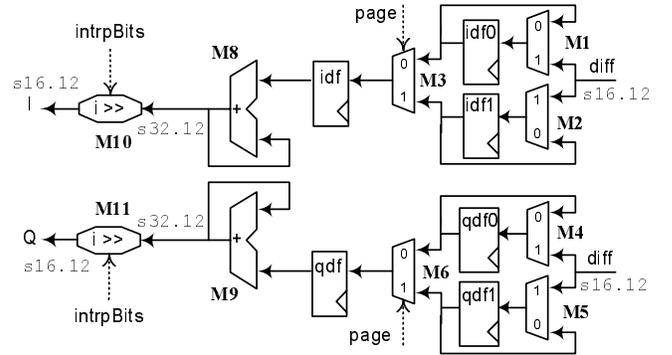


Fig. 7. Datapath of the quadrature linear interpolator.

V. GEOMETRICAL FADING CHANNEL IMPLEMENTATION AND SIMULATION RESULTS

In our experiment, the fading samples were generated at sample rate $\tilde{F}_s = 312.5$ Hz. The interpolator is set to increase the sample rate 32 times to provide a final sample rate of $F_s = 10,000$ samples per second. In the one-ring model, $N_S = 128$ scatterers surround the receiver uniformly on a ring of radius $R^{RX} = 30$ meters. For the two-ring model, $N_{S1} = 32$ and $N_{S2} = 128$ scatterers uniformly surround the TX and RX, respectively, on two circles of radius $R_{TX} = R_{RX} = 30$ meters. In the geometrical elliptical channel model, $N_S = 128$ scatterers surround the transmitter and the receiver uniformly on an ellipsoid with a major axis of $2a = 1460$ meters and a

minor axis of $2b = 1182$ meters. The TX and RX are located at the focal points of this ellipsoid.

To minimize the wordlength of signals and the size of on-chip memories in the fading channel simulator datapath, we developed a parameterizable fixed-point library of arithmetic and logical modules in MEX (compiled C routines for MATLAB). We then implemented the bit-true model of the geometrical fading channel simulator using our fixed-point library. To ensure the accuracy of our geometric fading channel simulator, we compared our fixed-point results against double-precision floating-point results. Figs. 8 (a), (b) and (c) show the fading samples $h_{11}[n]$ (channel gain from the first TX antenna to the first RX antenna) for the one-ring model, the two-ring model and, the geometric elliptical model, respectively. Both the generated floating-point and fixed-point results are shown. These figures show good agreement between the fixed-point and the floating-point results.

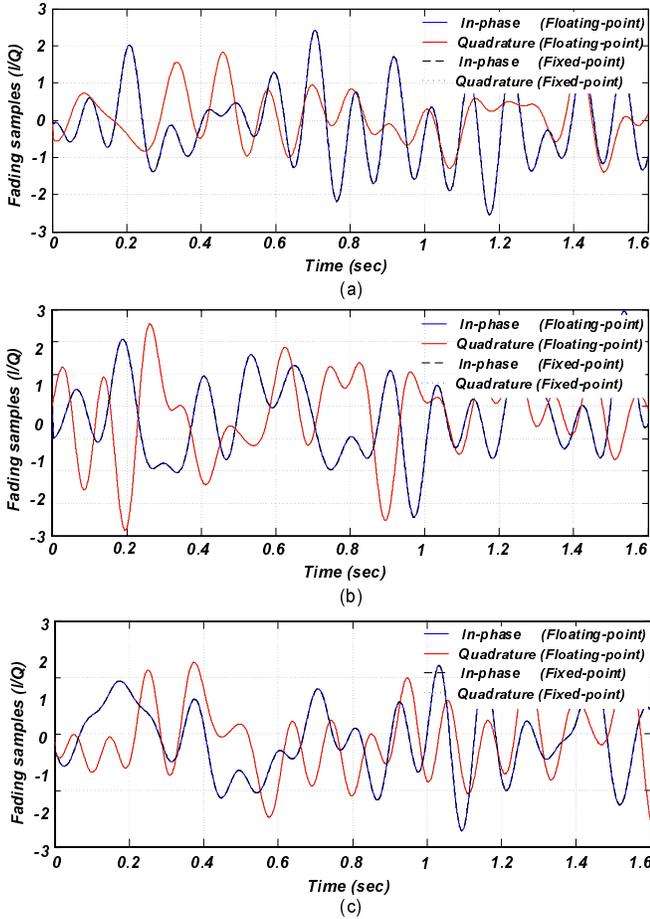


Fig. 8. Fading samples generated using (a) the one-ring model, (b) the two-ring model and, (c) the geometric elliptical model.

Fig. 9 plots the absolute error between the floating-point and the fixed-point results for the two-ring model. Since the oscillator phases are updated using running summations (see equation (8)), the quantization noise can accumulate in the oscillator phases over time. This in turn causes the fixed-point results to drift away from the floating-point results. A major

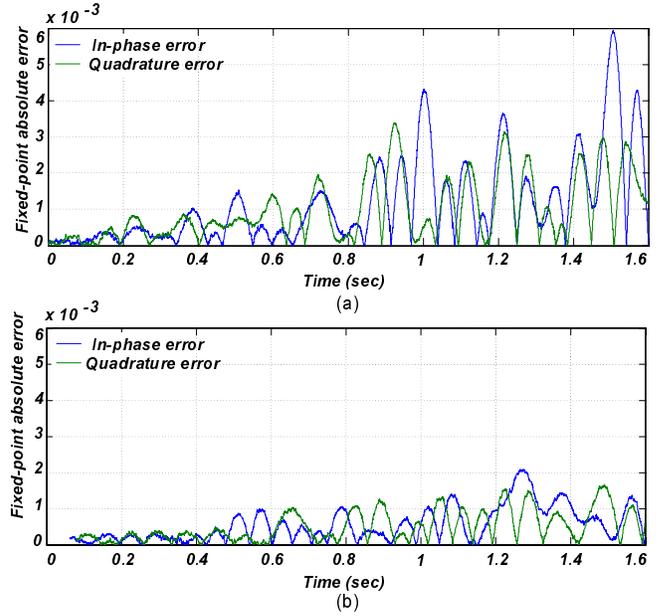


Fig. 9. (a) Absolute error between fixed-point results and the floating-point results when phases are rounded towards zero. (b) Absolute error when phases are rounded towards the nearest integer.

contributor to the quantization noise in the oscillator phases is the rounding operation that happens when calculating $f_D \times \cos(\omega_i - \gamma)$ (see the description of PART-2 in Section IV.A). The result of the multiplication is a 36-bit value that needs to be rounded to a 17-bit value according to the datapath shown in Fig. 6. The simplest rounding method is to discard the extra fraction bits. Since the phases are positive values, discarding the extra bits can be interpreted as rounding towards zero. Fig. 9 (a) plots the absolute error between the fixed-point results and the floating-point results for this case. As Fig. 9 (a) shows, the magnitude of the peaks of the absolute error increases with time. Rounding the 36-bit multiplier outputs to the nearest integer, on the other hand, can reduce the quantization effects as shown in Fig. 9 (b). Nevertheless, as the slow phase drifts do not affect the statistics of the generated fading samples significantly, rounding to the nearest integer is not necessary.

To ensure the statistical accuracy of our geometric fading channel simulator hardware, we compared the measured cross-correlations with their corresponding analytical function approximations. We simulated a 2×2 MIMO channel with the two-ring channel model. In this model, $N_{S1} = 128$ scatterers are assumed to be uniformly distributed around the transmitter on a circle of radius $R^{TX} = 30$ meters. Also, we assumed that $N_{S2} = 128$ scatterers surround the receiver uniformly on a ring of radius $R^{RX} = 30$ meters. In this simulation, the transmitter and the receiver of a 2×2 MIMO system are set $D = 860$ meters apart. The TX and RX antenna arrays are positioned at a 90 degree angle from the horizon. Also, the RX antennas are assumed to be moving in the $\gamma = 60$ degree direction. Further, the maximum Doppler frequency is $f_D = 10$ Hz and samples are generated at

$F_s = 1000$ samples per second. We set the antenna TX and RX antenna spacing to $\lambda/2 = 10$ cm, the Rice factors to $K_{11} = K_{12} = K_{21} = K_{22} = 0$, and we used isotropic scattering and omnidirectional antennas at the TX and RX sides. Fig. 10 plots the measured cross-correlation between $h_{11}[n]$ and $h_{22}[n]$ versus antenna separation at the TX and RX. The theoretical approximation for the cross-correlation from [17] is plotted in Fig. 11. The difference between the simulated cross-correlation and the theoretical approximation is plotted in Fig. 12. Comparing the results in Figs. 10 and 11 shows the difference between the measured results and the expected cross-correlation.

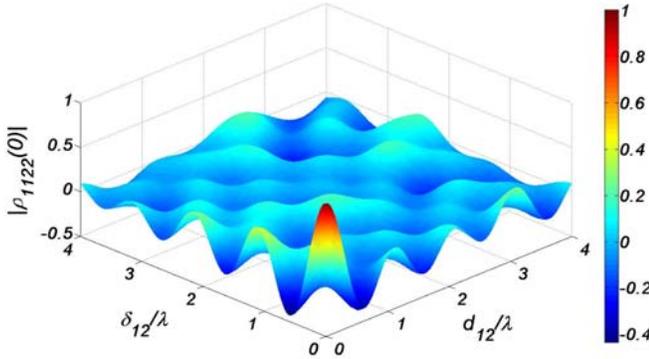


Fig. 10. Simulated cross-correlation between $h_{11}[n]$ and $h_{22}[n]$ versus transmitter and receiver antenna separation. Results are obtained from fixed-point computer simulation of a MIMO channel with the two-ring channel model.

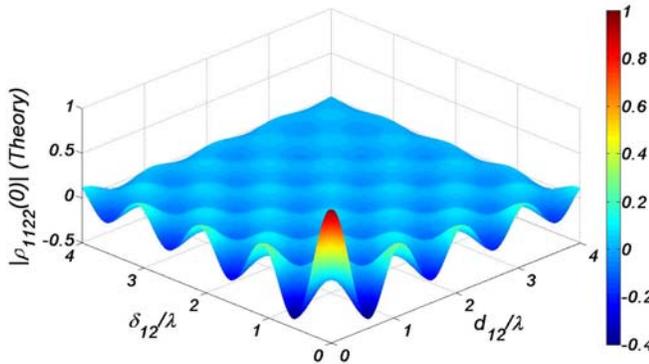


Fig. 11. Theoretical approximation of the cross-correlation between $h_{11}[n]$ and $h_{22}[n]$ for the two-ring model plotted versus antenna separation at the transmitter and the receiver.

The simulation of the fixed-point model for double-scattering scenarios is extremely costly computationally as it needs to generate $100 \times 100 = 10000$ cross-correlation values for each plot. We require a sufficiently large number of channel samples (here we used 10000 samples) to reliably estimate the cross-correlation. Note that $(M \times N) \times (N_{S1} \times N_{S2} + 1)$ complex sinusoids need to be scaled and superimposed when calculating each set of fading samples. With $N_{S1} = 128$ and $N_{S2} = 128$ scatterers and for $M = N = 2$ antennas, our software model required twenty four days to generate 6.554×10^{12} complex sinusoids, and estimate the correlations

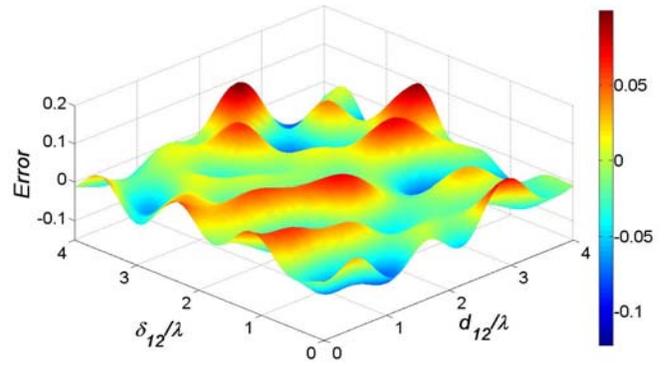


Fig. 12. Difference between the measured cross-correlation and the theoretical approximation of the cross-correlation between $h_{11}[n]$ and $h_{22}[n]$ for the two-ring channel model.

on a dual-core 3.6 GHz Intel Xeon processor. Note that our fixed-point library was developed with key sections coded in 80386 32-bit machine language and is thus extremely fast (80386 Assembly language embedded in C code and linked to MATLAB as a library of MEX files). Software simulation of double-bounce scattering becomes very time consuming because the computational complexity grows so rapidly with the number of scatterers.

VI. FPGA-BASED ERROR RATE MEASUREMENT

We implemented our new geometric fading channel simulator for a 4×4 MIMO system on a Xilinx Virtex-5 XC5VLX110-3 FPGA. The implemented design can be configured to simulate a wide variety of propagation conditions and channel models. As an experiment we configured the hardware to generate 16 streams of channel gains. We also set the maximum number of scatterers to $N_S = 128$. More scatterers can be added easily by increasing the storage capacity. Table I presents the implementation results of this fading simulator. Our FPGA implementation uses only 4597 configurable slices, two multipliers (DSP48E), and three on-chip 36-Kb block memories (BRAMs). In addition, Table I compares the new fading simulator with two 4×4 MIMO fading simulators. As this table shows, the new fading simulator is faster and yet much smaller than the designs reported in [10] and our previous fading simulator in [13], despite the fact that the new design simulates more scatterers and is capable of simulating more fading model types.

TABLE I
COMPARISON OF IMPLEMENTATION RESULTS

Design	I (from [10])	II (from [13])	III (NEW)
MIMO model	i.i.d	i.i.d	i.i.d, GSCM
$M \times N$	4×4	4×4	4×4
N_S	16	8	128
FPGA	EP20K1000-3	XC2VP100-6	XC5VLX110-3
Max. clock	50 MHz	201 MHz	324 MHz
Output rate	16×1.5	$2 \times 16 \times 201$	16×324
# of slices	22576 (58%)	41198 (93%)	4597 (6.6%)
# of MULTs	–	272 (61%)	2 (3%)
# of BRAMs	17%	288 (65%)	3 (2.0%)

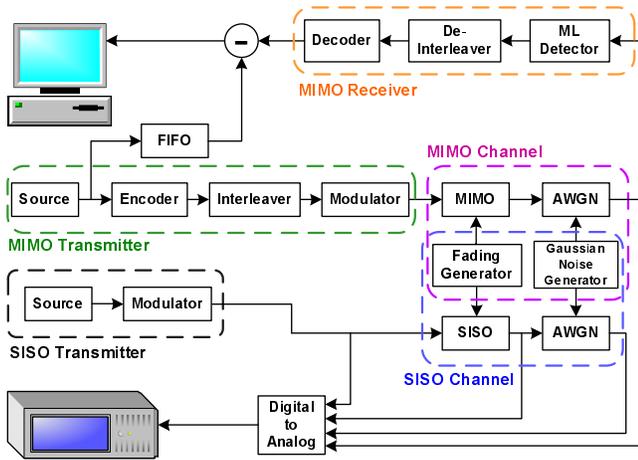


Fig. 13. Block diagram of the implemented fading simulation platform.

Since the performance of a wireless communication system is highly dependent on radio channel characteristics, it is important to use accurate radio channel models. In addition, the execution times of software-based performance simulations of the baseband layer on workstations can be extremely long, especially for increasingly complex communication systems with computationally-intensive signal processing. The accuracy, speed and compactness of the proposed geometric MIMO fading channel simulator makes it a powerful element for accelerating the performance verification of digital communication systems under realistic radio propagation conditions on a single chip.

We developed a parameterizable hardware-based baseband bit error rate tester (BERT) for single and multiple antenna communication systems. Fig. 13 shows the block diagram of the implemented bit error rate (BER) performance measurement system. On this platform, the fading simulator can be configured to simulate both single and multiple-antenna systems for single- and double-scattering fading channel models. In the implemented MIMO prototype system, source bits are encoded using an extended Golay channel code and interleaved with a length 16383 interleaver. Then the interleaved bits are modulated to 4-QAM symbols and passed through the 2×2 MIMO channel where they are affected by the multipath fading and corrupted with additive white Gaussian noise (AWGN). In the receiver, a maximum likelihood (ML) detector tries to estimate the transmitted bits. The ML detector can be configured in real-time to have access to complete and incomplete channel state information. After ML detection, the bit stream is de-interleaved, decoded for the extended Golay code, and compared to the transmitted bit stream. The implemented fading channel simulator can be parameterized in real-time using the host computer to simulate various propagation conditions.

To demonstrate the fading effects on the transmitted symbols, we also implemented a single-antenna transmitter where the bits can be modulated using different schemes (BPSK, QPSK, 4-PAM, 4-QAM, 8-PSK, 16-PSK, 16-QAM, cir-

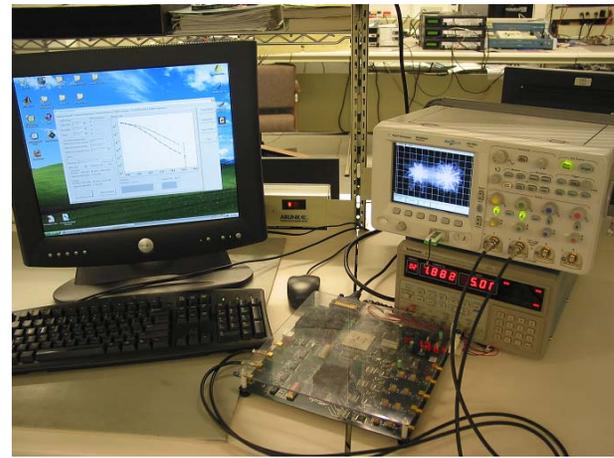


Fig. 14. Fading simulation platform on a GVA-290 FPGA board.

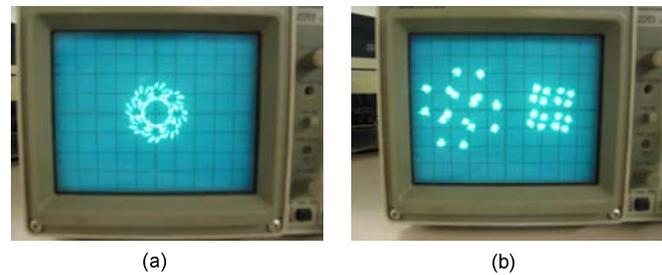


Fig. 15. Pictures of the oscilloscope output for (a) the SISO, and (b) the MIMO systems.

cular 8-QAM, and circular 16-QAM). As shown in the block diagram in Fig. 13, the output of this transmitter can be passed to an oscilloscope through a digital-to-analog converter. Moreover, the faded samples (with and/or without noise) can be monitored on the oscilloscope as well. Fig. 14 shows the picture of the implemented BERT on the GVA-290 board [21] along with the power source, oscilloscope, and the control computer. This board hosts two Xilinx VirtexE-2000 FPGAs and two Xilinx Spartan II FPGAs. This board also includes four 100 Msps analog-to-digital and four 100 Msps digital-to-analog converters. The GVA-290 board is interfaced with the control computer through the parallel port. Fig. 15 shows two outputs of the fading simulation platform on the oscilloscope screen. The Doppler frequency for these simulations was set to $f_D = 0.5$ Hz so that the changes in the scatter-plot could be easily followed. In Fig. 15 (a), the oscilloscope screen shows the scatter-plot of the noisy output of a single-input single-output (SISO) channel. In this picture, 8-PSK modulated samples are passed through a two-path SISO fading channel and corrupted with AWGN. Fig. 15 (b) shows the two noisy outputs of a the 2×2 MIMO channel where the transmitted bits are modulated with 4-QAM and the signal to noise ratio is 20 dB.

On a Xilinx VirtexE XCV2000EBG560-6 FPGA, the implemented MIMO communication system (source, encoder, interleaver, detector, de-interleaver, and decoder) utilizes less

than 9% of the available configurable slices while the rest of the system (fading simulator, BER performance measurement, initialization, and interfacing modules) consume a much larger portion of the available resources (more than 60%). The implemented fading simulation platform and the BER performance measurement cores along with the analog and digital access to different parts of the system on a GVA-290 board can be used for testing and verification of more complex wireless communication systems. More specifically, with one VirtexE-2000 FPGA dedicated to fading simulation and interfacing, the other on-board FPGAs can be used for rapid prototyping of wireless communication systems. In addition, the implemented fading simulation and BER performance measurement platform can be easily adapted to faster and more recent FPGA boards for rapid prototyping of wireless communication system in baseband and intermediate frequency.

Fig. 16 shows the hardware generated uncoded and coded BER performance of this MIMO system under the geometric elliptical and two-ring MIMO channel models. The antenna spacing at the transmitter and the receiver is $0.5 \times \lambda$ and the receiver is placed $D = 860$ meters away from the transmitter. In this simulation, the transmit and receive antenna arrays are positioned at a 90 degree angle from the horizon, the receiver is assumed to be moving in the $\gamma = 60$ degree direction, and we set the maximum Doppler frequency $f_D = 10$ Hz, the wavelength $\lambda = 20$ cm and the Rice factors were $K_{11} = K_{12} = K_{21} = K_{22} = 0$.

In the geometric elliptical model, $N_S = 128$ scatterers surround the TX and RX uniformly on an ellipsoid with the major axis of $2a = 1460$ meters and the minor axis of $2b = 1182$ meters. Also, for the two-ring model, $N_{S1} = 32$ and $N_{S2} = 64$ scatterers uniformly surround the TX and RX, respectively, on two circles of radius $R_{TX} = R_{RX} = 30$ meters. The implemented fading simulator shows the expected coding gain going from the uncoded to the coded systems under test for the two different propagation scenarios.

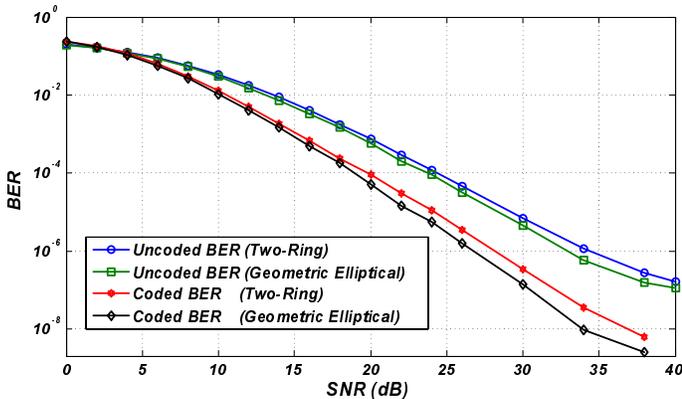


Fig. 16. BER performance of four 2×2 MIMO systems measured using the FPGA-based fading simulator.

VII. CONCLUSION

A fading channel simulator is a key component for the testing and verification of wireless communication systems. In this article we proposed a compact and accurate geometric MIMO fading channel simulator. The new simulator is versatile and can be configured in real-time to simulate different geometric MIMO channel models and propagation scenarios. We verified the accuracy of the implemented fading channel simulator architecture for three common geometric models, i.e., the one-ring model, the double-bounce two-ring model, and the single-bounce elliptical channel model. The proposed geometric MIMO fading channel simulator is compact enough that it can be implemented on a small fraction of a single FPGA along with the baseband transmitter and receiver chains of a MIMO communication system for high-speed performance verification at hardware speeds.

REFERENCES

- [1] S. Sesia, I. Toufik, and M. Baker, *LTE, The UMTS Long Term Evolution: From Theory to Practice*. John Wiley & Sons, 2009.
- [2] *ACE 400NB MIMO Channel Emulator*, Azimuth Systems Inc., Acton, MA, USA, 2006.
- [3] *MIMO Channel Simulator*, ARC SmartSim, 2007.
- [4] *SR5500 Wireless Channel Emulator*, Spirent Communications, Crawley, U.K), 2006.
- [5] 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.
- [6] F. Kaltenberger, G. Steinbock, G. Humer, and T. Zemen, "Low-complexity geometry based MIMO channel emulation," in *European Conf. on Antennas and Propagation*, 2006, pp. 1–8.
- [7] S. Haene, D. Perels, and A. Burg, "A real-time 4-stream MIMO-OFDM transceiver: System design, FPGA implementation, and characterization," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 6, pp. 877–889, Aug. 2008.
- [8] H. Yu, M. S. Kim, E. Y. Choi, T. Jeon, and S. K. Lee, "Design and prototype development of MIMO-OFDM for next generation wireless LAN," *IEEE Trans. Consum. Electron.*, vol. 51, no. 4, pp. 1134–1142, Nov. 2005.
- [9] W. A. Syaifei, Y. Nagao, R. Imashioya, M. Kurosaki, B. Sai, and H. Ochi, "Design of 600 Mbps MIMO wireless LAN system using GLST coding and its FPGA implementation," in *Proceeding of the IEEE Radio and Wireless Symposium*, Jan. 2009, pp. 296–299.
- [10] M. Cui, H. Murata, and K. Araki, "Real-time MIMO received signal generator for spatial multiplexing systems," in *Proc. of the IEEE Intl. Veh. Technol. Conf. (VTC'04)*, 2004, pp. 4345–4348.
- [11] H. Okuni, E. Kudoh, and F. Adachi, "Multipath fading simulator based on distributed scattering model," *IEICE Trans. Commun.*, vol. E87-B, no. 8, pp. 2422–2426, Aug. 2004.
- [12] C. H. Liao, T. P. Wang, and T. D. Chiueh, "A novel low-complexity Rayleigh fader for real-time channel modeling," in *Proc. of the IEEE Intl. Symp. on Circuits and Systems (ISCAS'07)*, 2007, pp. 2602–2605.
- [13] A. Alimohammad, S. Fouladi Fard, B. F. Cockburn, and C. Schlegel, "A novel technique for efficient hardware simulation of spatiotemporally correlated MIMO fading channels," in *Proc. of the IEEE Intl. Conf. on Communications*, 2008, pp. 718–724.
- [14] —, "A single-FPGA multipath MIMO fading channel simulator," in *Proc. of the IEEE Intl. Symp. on Circuits and Systems*. Seattle, Washington, USA, May 2008, pp. 308–311.
- [15] P. Almers et al., "Survey of channel and radio propagation models for wireless MIMO systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, pp. 1–19, 2007.
- [16] A. Abdi and M. Kaveh, "A space-time correlation model for multielement antenna systems in mobile fading channels," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 3, pp. 550–560, Apr. 2002.

- [17] G. J. Byers and F. Takawira, "Spatially and temporally correlated MIMO channels: Modeling and capacity analysis," *IEEE Trans. Veh. Technol.*, vol. 53, no. 3, pp. 634–643, May 2004.
- [18] M. Pätzold and B. O. Hogstad, "A wideband MIMO channel model derived from the geometric elliptical scattering model," in *Proc. of the Intl. Symp. on Wireless Communication Systems*, 2006, pp. 138–143.
- [19] J. G. Proakis, *Digital Communications*. McGraw-Hill, 2001.
- [20] L. Erup, F. M. Gardner, and R. A. Harris, "Interpolation in digital modems - part II: Implementation and performance," *IEEE Trans. Commun.*, vol. 41, no. 6, pp. 998–1008, Jun. 1993.
- [21] *GVA-290 Xilinx VirtexE Hardware Accelerator*, GV Associates, Ramona, CA, USA, Aug. 2009. [Online]. Available: <http://www.gvassociates.com>