

# A Compact Architecture for Simulation of Spatio-Temporally Correlated MIMO Fading Channels

Amirhossein Alimohammad and Saeed Fouladi Fard

**Abstract**—Radio channel impairments have a dramatic impact on the performance of wireless communication systems and hence, utilizing realistic radio channel models is crucial for the accurate performance validation of emerging wireless systems. However, faithful radio propagation channel models are computationally-intensive for software-based simulations, especially for multiple antenna systems. This article presents the design and implementation of a multiple-input multiple-output (MIMO) baseband fading channel simulator on a field-programmable gate array (FPGA). In addition to the well-known independent and identically distributed channel model, the simulator supports three spatio-temporally correlated fading channel models which are commonly used for performance analysis. The implemented MIMO fading channel simulator is compact enough to be integrated with the baseband design under test on the same FPGA for accelerated performance validations.

**Index Terms**—Spatio-temporally correlated fading channels, channel simulation, Kronecker model, Weichselberger model, virtual channel representation model, multiple-input multiple-output (MIMO), field-programmable gate array (FPGA).

**Notations:**  $(\cdot)^T$  denotes matrix transposition,  $(\cdot)^H$  denotes the Hermitian of a matrix,  $(\cdot)^{1/2}$  denotes the matrix square root,  $\text{tr}\{\cdot\}$  denotes the trace of a matrix,  $\odot$  denotes element-wise Schur-Hadamard multiplication,  $\lfloor x \rfloor$  denotes the largest integer number that is smaller than  $x$ .

## I. MOTIVATION

In the rapidly evolving area of digital communications, new wireless systems must be developed in short design cycles. The most accurate performance of wireless communication systems can be measured by building a full prototype of the wireless system and performing experiments at various locations under different conditions in the field. While field testing allows credible performance verification under real-world radio channels, it is preferable to gain insight into system behavior and reliably estimate the performance of the eventual solution prior to the development and deployment of prototype systems in real-world environments.

Simulation is indispensable for the design and performance evaluation of new algorithms before the actual system is implemented. One of the computationally-intensive processes in the design and verification cycle of wireless communication systems is the performance validation of the physical (PHY)

layer under various radio propagation conditions. This is mainly because the baseband layer of emerging broadband communication systems is significantly more complex than the baseband layer of the preceding generations [1]. For example, the long term evolution (LTE) universal mobile telecommunications system (UMTS) supports significantly higher peak data rates (100 Mbps downlink and 50 Mbps uplink), flexible bandwidth options (from 1.4 MHz to 20 MHz), and multi-antenna configurations (ranging from two to four antennas per device). In addition, due to advances in communication techniques, the number of possible configurations and propagation scenarios in which increasingly complex wireless communication systems need to operate and hence, must be verified has increased dramatically. For example, a wireless system should be able to operate reliably in various radio propagation environments (e.g., indoor and outdoor), adapt to unpredictable channel conditions by changing its transmission scheme (e.g., different modulation and/or code rates), and select alternative detection techniques. Therefore, the performance of increasingly complex wireless systems must be evaluated under a relatively large number of settings and options. As the number of possible configurations in emerging standards increases (e.g., more than 300 modulation and coding schemes are present in the IEEE 802.11n standard), timely design and validation of emerging broadband communication systems becomes significantly more challenging. In addition, the bit-true software-based simulation of the baseband layer on workstations is becoming prohibitively time-consuming. As the general purpose processors are not efficient for the bit-level simulation of baseband signal processing algorithms, this makes hardware-accelerated prototyping and validation an increasingly attractive alternative [2], [3]. Hardware-based bit error rate testers (BERTs) can accelerate the performance evaluation of wireless communication systems by several orders of magnitude compared to conventional software-based simulations [4]–[9], hence increasing designer productivity.

The performance of a wireless communication system is highly dependent on the radio channel characteristics and therefore, it is crucial to use accurate radio channel models for reliable evaluations. However, realistic radio propagation channel models are computationally-daunting processes, especially for multiple-input multiple-output (MIMO) systems. Commercial fading channel emulators can be used to generate accurate radio propagation conditions in a laboratory setting, however they are relatively costly [10], [11]. Recently several realizations of MIMO fading channel simulators on field-programmable gate arrays (FPGAs) have been proposed as a low-cost hardware-based solution [12]–[16].

This article presents the design and implementation of a

Copyright © 2013 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org.

A. Alimohammad is with the Department of Electrical and Computer Engineering, San Diego State University, San Diego, CA, 92182, U.S.A. Saeed Fouladi Fard is with PMC-Sierra, Calgary, Alberta, Canada. E-mail: aalimohammad@mail.sdsu.edu.

MIMO baseband fading channel simulator on a single FPGA. The simulator utilizes the fading variate generator from [15] and realizes a new matrix processor for the compact implementation of four MIMO fading channel models, which are widely accepted by the research community for performance analysis [17]. Also, an accurate linear interpolator is designed and implemented to generate the final fading samples at the desired baseband rate. While a hardware-based fading channel simulator can speed up the validation of communication systems over a wide range of radio channel conditions, to further accelerate performance validation at the PHY layer, other computationally-intensive signal processing modules can also be implemented on the same state-of-the-art FPGA [2], [7], [18]. The proposed MIMO channel simulator is compact enough that can be implemented with the baseband design under test on the same low-cost FPGA for bit error rate (BER) performance validation of emerging wireless systems under various channel conditions at hardware speeds.

The rest of this article is organized as follows. Section II briefly reviews four analytical MIMO fading channel models. Section III presents the process of spatio-temporally correlated MIMO fading variate generation along with its compact hardware implementation. Section IV discusses the interpolator design and its novel hardware structure. Section V presents the application of the implemented MIMO fading channel simulator in a developed BER performance measurement system. Section VI makes some concluding remarks.

## II. ANALYTICAL MIMO FADING CHANNEL MODELS

In a frequency non-selective MIMO system with multiple omnidirectional antennas at the transmitter and receiver [19], the received signal at time  $t$  can be represented as

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

where  $\mathbf{H}(t)$  is a time-variant  $n_R \times n_T$  MIMO fading channel matrix between  $n_T$  transmit antennas and  $n_R$  receive antennas,  $\mathbf{s}(t)$  is the  $n_T \times 1$  transmitted signal vector,  $\mathbf{n}(t)$  denotes the  $n_R \times 1$  noise and interference vector, and  $\mathbf{y}(t)$  is the  $n_R \times 1$  received vector. The element of the channel matrix  $h_{lp}(t)$ ,  $l = 1, \dots, n_R$ ,  $p = 1, \dots, n_T$ , represents the complex-valued fading gain between the  $p$ -th transmit antenna and the  $l$ -th receive antenna at any time  $t$ .

Analytical fading channel models were proposed by [20] in the framework of the IST SATURN project [?] and have also been used in the IEEE 802.11 standard (Task Group  $n$ ) [21]. In contrast to propagation-based models [22], [23], analytical fading channel models are site-independent and characterize each  $h_{lp}(t)$  mathematically without explicitly accounting for physical wave propagations. Nevertheless, one can use analytical fading channel models in combination with the propagation-based models. Several different analytical models have been proposed in the literature. They can be sub-classified as (a) correlation-based models, which are well-suited for system design and performance analysis of baseband signal processing algorithms [17], [22], such as the independent and identically distributed (i.i.d.) model [19], the Kronecker model [24], the Weichselberger model [25], and (b) propagation-motivated models, such as the virtual channel representation

(VCR) model [26], the finite scatter model [27], and the maximum entropy model [28]. In this work, we focus on four fading channel models: the i.i.d. model, the Kronecker model, the Weichselberger model, and the VCR model.

The i.i.d. flat-fading channel model assumes that the fades between pairs of transmit and receive antennas are independent and identically distributed. This model corresponds to isotropic scattering and deployment of largely spaced omni-directional antennas [29], [30]. Such a channel arises if both the transmitter and receiver exist in a rich-scattering environment. The entries  $\{h_{lp}(t)\}$  of an i.i.d. channel matrix  $\mathbf{H}(t)$  are zero mean, unit variance temporally-correlated complex Gaussian random variables while no spatial correlation is assumed between different sub-channels  $\{\mathbf{h}_p\}$ , where  $\mathbf{h}_p$  denotes the column vector of complex transfer gains from the  $p$ -th transmit antenna to all  $n_R$  receive antennas.

In a typical MIMO scenario, however, the fades usually exhibit spatial correlations between different transmit-receive antenna pairs [31]. Propagation characteristics of the environment, such as clustered scattering, and the physical parameters of antennas, such as spacing and orientation, affect the spatial correlations between different antennas. For efficient design of the baseband signal processing algorithms, it is essential to utilize faithful MIMO channel models that reproduce the spatio-temporal characteristics of MIMO radio channels accurately. To obtain the space-time correlation characteristics, a temporally-correlated random process can be followed by a linear transformation to be made spatio-temporally correlated [31], as shown in Fig. 1. Therefore, analytical MIMO channel models can be simulated by introducing specific correlation between zero-mean i.i.d. Gaussian samples. The spatial structure of the channel is commonly characterized by channel correlation matrices.

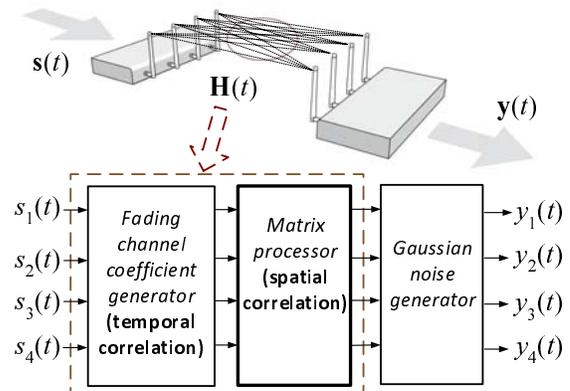


Fig. 1. 4×4 MIMO channel.

The Kronecker model considers separate spatial correlation properties (matrices) at the transmitter and receiver sides and can be expressed as

$$\mathbf{H} = \mathbf{U}\mathbf{G}\mathbf{V}, \quad (2)$$

where  $\mathbf{G}$  is the  $n_R \times n_T$  i.i.d. matrix with zero-mean, unit variance circularly-symmetric complex Gaussian distributed entries (i.e., spatially-independent rich-scattering MIMO fading channel),  $\mathbf{U} = \mathbf{R}_{\text{Rx}}^{1/2}$ ,  $\mathbf{V} = (\mathbf{R}_{\text{Tx}}^{1/2})^T$ , where  $\mathbf{R}_{\text{Tx}}$  and  $\mathbf{R}_{\text{Rx}}$

denote the  $n_T \times n_T$  transmit and  $n_R \times n_R$  receive correlation matrices, respectively [24].

The Weichselberger model relaxes the separability restriction of the Kronecker model and jointly models the correlation properties at the transmitter and receiver as

$$\mathbf{H} = \mathbf{U}(\mathbf{W} \odot \mathbf{G})\mathbf{V}, \quad (3)$$

where  $\mathbf{W} = \tilde{\Omega}_W$  is the element-wise square root of the power coupling matrix  $\Omega$ , whose positive and real-valued elements determine the average power-coupling between the  $p$ -th transmit and the  $l$ -th receive direction. Also,  $\mathbf{U} = \mathbf{U}_{\text{Rx}}$ ,  $\mathbf{V} = \mathbf{U}_{\text{Tx}}^T$ , where  $\mathbf{U}_{\text{Tx}}$  and  $\mathbf{U}_{\text{Rx}}$  are complex unitary matrices containing eigenvectors of  $\mathbf{R}_{\text{Tx}}$  and  $\mathbf{R}_{\text{Rx}}$ , respectively [25]. The  $\mathbf{R}_{\text{Tx}}$  and  $\mathbf{R}_{\text{Rx}}$  matrices can be obtained using the eigenvalue decomposition of the transmit and receive correlation matrices as follows:

$$\begin{aligned} \mathbf{R}_{\text{Tx}} &= \mathbf{U}_{\text{Tx}} \mathbf{\Lambda}_{\text{Tx}} \mathbf{U}_{\text{Tx}}^H, \\ \mathbf{R}_{\text{Rx}} &= \mathbf{U}_{\text{Rx}} \mathbf{\Lambda}_{\text{Rx}} \mathbf{U}_{\text{Rx}}^H. \end{aligned}$$

In contrast to the two prior models, the VCR models a MIMO channel in the beamspace instead of the eigenspace [26]. In particular, the eigenvectors are replaced by fixed and predefined steering vectors as

$$\mathbf{H} = \mathbf{U}(\mathbf{W} \odot \mathbf{G})\mathbf{V}, \quad (4)$$

where  $\mathbf{W} = \tilde{\Omega}_V$  and elements of  $\tilde{\Omega}$  denote the coupling of each scatterer. Also,  $\mathbf{U} = \mathbf{A}_{\text{Rx}}$ ,  $\mathbf{V} = \mathbf{A}_{\text{Tx}}^H$ , where  $\mathbf{A}_{\text{Tx}}$  and  $\mathbf{A}_{\text{Rx}}$  contain steering vectors for  $n_T$  virtual transmit and  $n_R$  virtual receive scatterers.

### III. SPATIO-TEMPORALLY CORRELATED MIMO FADING VARIATE GENERATION

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 variate generator at a significantly lower sample rate and thereby reduce the required hardware resources by using time-multiplexed datapaths. Fading samples generated at the slower sample rate  $\hat{F}_s$  are then oversampled and interpolated  $I$  times to provide samples at the output rate  $F_s = I \times \hat{F}_s$ . We note that the discrete-time interpolator output at time  $mI + k$ , for  $m = 0, 1, 2, \dots$  and  $k = 0, 1, \dots, I - 1$ , is given recursively as

$$\begin{aligned} h_{lp}[mI + k] &= \frac{(h_{lp}[(m+1)I] - h_{lp}[mI])k}{I} + h_{lp}[mI] \\ &= \sum_{j=0}^k \frac{h_{lp}[(m+1)I] - h_{lp}[mI]}{I} \\ &\quad + h_{lp}[mI], \end{aligned} \quad (5)$$

where each element  $h_{lp}[n]$  denotes the time-variant channel gain between the  $p^{\text{th}}$  transmit antenna and the  $l^{\text{th}}$  receive antenna [14], [32]. The quadrature components of channel gains  $h_{lp}[n]$  are independent but temporally-correlated and can be generated using the fading variate generator described in [15]. The amplitude statistics of generated fading gains may

follow a Rician distribution if a direct line-of-sight (LOS) path is present, or a Rayleigh distribution in the absence of a LOS path. The Nakagami- $m$  distribution is a more general distribution that can be utilized to model fading that is more or less severe than Rayleigh fading [33]. It is important to note that since the linear interpolator in (5) requires the discrete difference between two subsequent low-frequency fading samples (i.e., the current sample and the previous sample) for each transmit-receive antenna pair, the fading simulator in [15] also generates the discrete difference fading signals, which can be passed to the linear interpolator for generating the final channel gains.

Our design in [15] uses a time-multiplexed datapath to generate a relatively large number of fading processes on a single FPGA (e.g., 1184 different paths on a Virtex-4 XC4VLX200-11 Xilinx FPGA). These flat fading channels can be used to model standard-compliant frequency-selective MIMO fading channels. Wireless standards typically specify several different power delay profiles (PDPs), such as Extended Pedestrian A model (EPA), Extended Vehicular A model (EVA), and Extended Typical Urban model (ETU) in the LTE standard [34], to model various fading scenarios. In this article we focus on applying spatial correlation properties of a MIMO fading channel to the previously generated temporally-correlated fading samples in [15] using a new compact architecture.

To introduce spatial correlations between temporally-correlated fading variates, instead of performing the matrix calculations (2) and (4) on high-frequency samples, we similarly perform the matrix operations on low-frequency fading samples and later up-sample the resulting streams with appropriate interpolators. Using the fading channel simulator generator in [15], the difference fading channel matrix  $\mathbf{D}[m]$  can be obtained as

$$\mathbf{D}[m] = [d_{lp}[m]] = \mathbf{H}[m+1] - \mathbf{H}[m],$$

where the element  $d_{lp}[m]$  in the  $l$ -th row and  $p$ -th column of  $\mathbf{D}[m]$  denotes the discrete difference fading signal  $h_{lp}[m+1] - h_{lp}[m]$  generated by the temporally-correlated fading variate generator in [15]. The difference samples are then passed to the linear interpolators for up-sampling. Note that due to the linearity of the basic matrix multiplication, we can verify that

$$\begin{aligned} \mathbf{U}\mathbf{H}[m+1]\mathbf{V} - \mathbf{U}\mathbf{H}[m]\mathbf{V} &= \mathbf{U}(\mathbf{H}[m+1] - \mathbf{H}[m])\mathbf{V} \\ &= \mathbf{U}\mathbf{D}[m]\mathbf{V}. \end{aligned} \quad (6)$$

Similarly, since the Shur-Hadamard product is a linear operation, we can write

$$\begin{aligned} \mathbf{U}(\mathbf{W} \odot \mathbf{H}[m+1])\mathbf{V} - \mathbf{U}(\mathbf{W} \odot \mathbf{H}[m])\mathbf{V} \\ &= \mathbf{U}(\mathbf{W} \odot (\mathbf{H}[m+1] - \mathbf{H}[m]))\mathbf{V} \\ &= \mathbf{U}(\mathbf{W} \odot \mathbf{D}[m])\mathbf{V}. \end{aligned} \quad (7)$$

Equations (6) and (7) imply that instead of performing the matrix operations (2) and (4) on the original fading samples, we can use the difference samples and perform (6) and (7), respectively.

For an efficient implementation of the spatial correlation characteristics of analytical MIMO fading channels, we designed a pipelined architecture. The architecture receives the

difference fading samples of matrix  $\mathbf{D}$  from the MIMO fading variate generator in [15], performs the matrix operations of either equation (6) or equation (7) on the generated temporally-correlated fading samples, and passes the spatio-temporally correlated fading samples to the next stage for interpolation. Fig. 2 shows the datapath of the implemented architecture for performing the matrix operations. For convenience of presentation, this architecture will be loosely referred to as the “matrix processor” henceforth. The three RAMs  $u$ RAM,  $w$ RAM, and  $v$ RAM are used to keep the elements of the  $\mathbf{U}$ ,  $\mathbf{W}$ , and  $\mathbf{V}$  matrices, respectively. These memories are dual-port RAMs that can be accessed and programmed externally through the address bus  $xAddrBus$  and the data bus  $xDataBus$  for the real-time configurability. Moreover, the dual-port memory  $t$ RAM is used as a register bank for holding the intermediate results. The elements of  $\mathbf{U}$ ,  $\mathbf{W}$ , and  $\mathbf{V}$  matrices can be read using the address bus  $uwvAddr$  and to access  $t$ RAM, the address lines  $tAddrA$  and  $tAddrB$  are used. The module  $Y1$  is implemented to interface the matrix processor to the fading generator module from [15]. The complex-valued fading samples are presented in 32-bit format, 16 bits of which are used to present the in-phase part and the remaining 16 bits are used for presenting the quadrature part. Before being passed to the arithmetic unit (AU), the input fading samples are converted into 36-bit variables (18 bits for the in-phase part and 18 bits for the quadrature part).

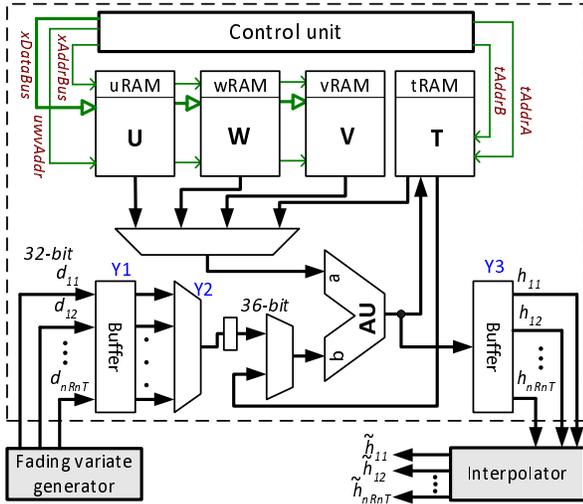


Fig. 2. Datapath of the pipelined architecture for performing the matrix operations of equations (6) and (7).

The control unit is implemented using a set of low-level microinstructions for the elementary matrix operations, including the matrix summation, the matrix product, and the Shur-Hadamard matrix product. We developed specific subroutines of microinstructions to perform these matrix operations efficiently by exploiting the pipelining capabilities of the AU datapath. Moreover, the developed subroutines can be parameterized to perform the matrix operations for different matrix dimensions. The control unit can be reprogrammed to perform these matrix operations in any order.

The core of this architecture is the AU datapath, as shown in Fig. 3, which performs basic complex arithmetic operations,

such as complex products  $(a_i + ja_q) \times (b_i + jb_q)$  and complex additions  $(a_i + ja_q) + (b_i + jb_q)$  for the two complex inputs  $a_i + ja_q$  and  $b_i + jb_q$ , and also real by complex products  $a_i \times (b_i + jb_q)$  [35]. The datapath of the AU contains two 18-bit multipliers U2 and U3 that operate on the quadrature parts. The 36-bit output of the multiplier U2 (U3) is rounded to the nearest 18-bit value by the modules U4, U5, and U8 (U6, U7, and U9). The rounding operation is necessary to reduce the quantization noise and to stabilize the linear interpolators. This will be explained more in the next section.

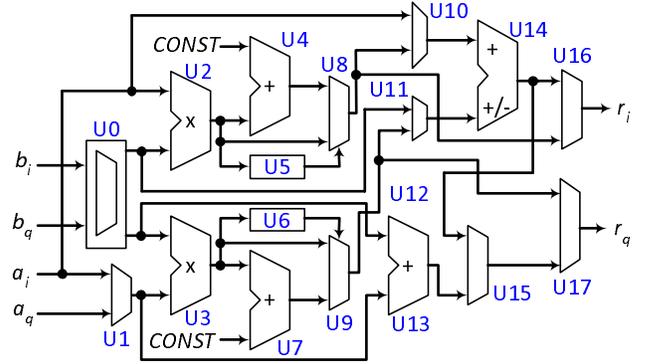


Fig. 3. Datapath of the pipelined arithmetic unit for performing the basic complex operations.

When the addition operation is selected, the in-phase parts of the complex inputs (i.e.,  $a_i$  and  $b_i$ ) are routed to the adder/subtractor U14 either through the multiplexer U10, or through the multiplexers U0 and U11. Also, the quadrature parts of the complex inputs (i.e.,  $a_q$  and  $b_q$ ) are routed to the adder U13 through the multiplexers U0 and U1. The output samples  $r_i = a_i + b_i$  and  $r_q = a_q + b_q$  are then sent to the output via multiplexers U16 and U17. For the real by complex product, the real input  $a_i$  is passed to the multipliers U2 and U3 where it is multiplied by the  $b_i$  and  $b_q$ , respectively. After rounding, the results  $r_i = a_i \times b_i$  and  $r_q = a_i \times b_q$  are passed to the output through multiplexers U16 and U17. Note that the above operations are pipelined and when the pipeline is full, the AU can perform one operation per clock cycle for real by complex products and for complex additions.

In contrast to the above complex operations that can be performed in a single clock cycle, the complex product needs two clock cycles (considering a full pipeline). In the first cycle, the inputs  $a_i$  and  $b_q$  are passed to the multiplier U2 and the inputs  $a_q$  and  $b_i$  are routed to the multiplier U3. After rounding, the results of these two products are passed to the adder/subtractor U14 where the quadrature part of the result (i.e.,  $a_i \times b_q + a_q \times b_i$ ) is calculated and stored in the register U12. In the next cycle, the inputs  $a_i$  and  $b_i$  are passed to the multiplier U2 and the inputs  $a_q$  and  $b_q$  are passed to the multiplier U3. The calculated products are then rounded and passed to the adder/subtractor U14 and the in-phase part of the result (i.e.,  $a_i \times b_i - a_q \times b_q$ ) is calculated. Moreover, the in-phase and the quadrature parts of the result are passed to the output through the multiplexers U16, U15, and U17.

To perform the matrix operations needed for the Kronecker channel model, the matrix processor starts with calculating

$\mathbf{T}_1 = \mathbf{U}\mathbf{D}$ , where the elements of  $\mathbf{U}$  are read from uRAM, the complex elements of the input matrix  $\mathbf{D}$  are read from Y1 through the multiplexer Y2 and are passed to the AU, and the elements of the temporary matrix  $\mathbf{T}_1$  are written to tRAM. To increase the system throughput and efficient use of the pipelined datapath, the control unit of the matrix processor performs the matrix multiplication  $\mathbf{U}\mathbf{D}$  in  $n_R$  steps (assuming that  $\mathbf{U}$  is  $n_R \times n_R$  and  $\mathbf{D}$  is  $n_R \times n_T$ ). More specifically,  $\mathbf{T}_1$  is calculated as follows:

$$\begin{aligned} & \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n_R} \\ u_{21} & u_{22} & \cdots & u_{2n_R} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n_R1} & u_{n_R2} & \cdots & u_{n_Rn_R} \end{pmatrix} \begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1n_T} \\ d_{21} & d_{22} & \cdots & d_{2n_T} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n_R1} & d_{n_R2} & \cdots & d_{n_Rn_T} \end{pmatrix} \\ &= \begin{pmatrix} u_{11}d_{11} & u_{11}d_{12} & \cdots & u_{11}d_{1n_T} \\ u_{21}d_{11} & u_{21}d_{12} & \cdots & u_{21}d_{1n_T} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n_R1}d_{11} & u_{n_R1}d_{12} & \cdots & u_{n_R1}d_{1n_T} \end{pmatrix} + \\ & \begin{pmatrix} u_{12}d_{21} & u_{12}d_{22} & \cdots & u_{12}d_{2n_T} \\ u_{22}d_{21} & u_{22}d_{22} & \cdots & u_{22}d_{2n_T} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n_R2}d_{21} & u_{n_R2}d_{22} & \cdots & u_{n_R2}d_{2n_T} \end{pmatrix} + \\ & \dots \\ & \begin{pmatrix} u_{1n_R}d_{n_R1} & u_{1n_R}d_{n_R2} & \cdots & u_{1n_R}d_{n_Rn_T} \\ u_{2n_R}d_{n_R1} & u_{2n_R}d_{n_R2} & \cdots & u_{2n_R}d_{n_Rn_T} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n_Rn_R}d_{n_R1} & u_{n_Rn_R}d_{n_R2} & \cdots & u_{n_Rn_R}d_{n_Rn_T} \end{pmatrix}. \end{aligned}$$

In the first step, the first column of  $\mathbf{U}$  is multiplied by the first row of  $\mathbf{D}$  and the results are stored in the temporary memory tRAM. In the second step, the second column of  $\mathbf{U}$  is multiplied by the second row of  $\mathbf{D}$  and so on, until the  $n_R$ -th column of  $\mathbf{U}$  is multiplied by the  $n_R$ -th row of  $\mathbf{D}$ . Then the matrix processor accumulates the  $n_R$  sub-product matrices to generate the multiplication result. The reason for this out-of-order processing for complex matrix multiplication is that the AU has different latencies for complex addition and complex multiplication operations. By re-arranging the multiplication and addition operations, unnecessary bubbles in the pipeline can be avoided.

The calculated temporary matrix  $\mathbf{T}_1 = \mathbf{U}\mathbf{D}$  is used to generate the difference samples  $\mathbf{E} = \mathbf{T}_1\mathbf{V}$ , where the elements of  $\mathbf{T}_1$  are read from tRAM, the elements of  $\mathbf{V}$  are read from vRAM, and the elements of the output matrix  $\mathbf{E}$  are written to buffer Y3 to be passed to the interpolators. This matrix multiplication is performed similar to the previous matrix multiplication. This operation is first broken into  $n_T$  multiplication operations between the columns of  $\mathbf{T}_1$  and the rows of the matrix  $\mathbf{V}$ . Then the  $n_T$  sub-product matrices are accumulated to generate the output matrix  $\mathbf{E}$ .

Simulating the Weichselberger model and the VCR model requires an additional Schur-Hadamard (or element-wise) product. More specifically, the matrix processor first calculates  $\mathbf{T}_2 = \mathbf{W} \odot \mathbf{D}$  as

$$\begin{aligned} & \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1n_T} \\ w_{21} & w_{22} & \cdots & w_{2n_T} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n_R1} & w_{n_R2} & \cdots & w_{n_Rn_T} \end{pmatrix} \odot \\ & \begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1n_T} \\ d_{21} & d_{22} & \cdots & d_{2n_T} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n_R1} & d_{n_R2} & \cdots & d_{n_Rn_T} \end{pmatrix} \\ &= \begin{pmatrix} w_{11}d_{12} & w_{12}d_{12} & \cdots & w_{1n_T}d_{1n_T} \\ w_{21}d_{21} & w_{22}d_{22} & \cdots & w_{2n_T}d_{2n_T} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n_R1}d_{n_R1} & w_{n_R2}d_{n_R2} & \cdots & w_{n_Rn_T}d_{n_Rn_T} \end{pmatrix}. \end{aligned}$$

Note that the elements of the matrix  $\mathbf{W}$  are real valued and therefore the AU can calculate each of the real by complex products in a single clock cycle (when the pipeline is full). After calculating  $\mathbf{T}_2$ , the matrix processor proceeds with calculating  $\mathbf{T}_3 = \mathbf{U}\mathbf{T}_2$  and  $\mathbf{E} = \mathbf{T}_3\mathbf{V}$ . These matrix multiplications are also performed with the same matrix multiplication subroutine that breaks the calculations into  $n_R$  (or  $n_T$ ) stages for effective use of the pipelined datapath.

#### IV. INTERPOLATOR DESIGN AND HARDWARE IMPLEMENTATION

To simplify the hardware implementation of the interpolator we constrain the interpolation factor  $I$  to be a power of 2, i.e.,  $I = F_s/\hat{F}_s = 2^k$ . In this case, the interpolator (5) can be implemented without multiplications or divisions. The interpolated fading samples  $\tilde{h}_{lp}[n]$ ,  $n \geq 0$ , at the times  $n = 2^k m + i$ ,  $m \geq 0$ ,  $i = 0, \dots, 2^k - 1$ , can be written as

$$\tilde{h}_{lp}[2^k m + i] = h_{lp}[0] + 2^{-k} d_{lp}[m]i + \sum_{\hat{m}=0}^{m-1} d_{lp}[\hat{m}], \quad (8)$$

where

$$d_{lp}[\hat{m}] = \begin{cases} h_{lp}[2^k(\hat{m} + 1)] - h_{lp}[2^k \hat{m}] & \text{for } \hat{m} \geq 0; \\ 0 & \text{otherwise,} \end{cases} \quad (9)$$

denotes the difference between subsequent fading samples at the slower sample rate  $\hat{F}_s = 2^{-k} F_s$ . Note that the expression  $2^{-k} d_{lp}[m]i$  in (8) can be calculated using shifting and running-sum operations and therefore, the linear interpolator can be conveniently implemented using an accumulator and a shifter.

This interpolator can be used effectively for interpolating samples generated by the fading simulator in [15] because the difference signal has no DC components. In other words, the impulse response of the difference block,  $h_d[n] = \delta[n] - \delta[n-1]$  (see equation (9)), in frequency domain is  $H_d(e^{j2\pi f}) = 1 - e^{-j2\pi f}$ , which has no output at zero frequency (i.e.,  $H_d(e^{j2\pi 0}) = 0$ ). On the other hand, the quantization noise added by the matrix processor is not necessarily DC-free.

This is due to the fact that the number of bits generated by a multiplier is the sum of the number of bits of the input operands. However, implementing a significantly wider datapath for the multiplication results was mainly avoided to reduce the hardware complexity. Instead some of the output bits were trimmed. This, however, can increase the quantization noise that can affect the accuracy of the implemented fading simulator. The rounding technique used in the matrix processor has a significant impact on the DC component of the added quantization noise.

It is important to note that the linear interpolator in equation (8) is implemented with a running summation, which resembles a lowpass infinite impulse response (IIR) filter with the frequency response  $H_i(e^{j2\pi f}) = 1/(1 - e^{-j2\pi f})$ . This IIR filter accumulates the input samples (extremely large gain at zero frequency) and therefore any DC component in the input will add up over time. In contrast to generating temporally-correlated fading variates, which only require superposition of zero-mean sinusoids [15], simulating the spatial correlation characteristics of MIMO fading channel models requires multiple fixed-point multiplication and addition operations.

An effective method, which is used in the matrix processor, is rounding the multiplication results to the nearest fixed-point samples. Rounding the multiplication results can significantly reduce the DC component of the quantization noise, since the multiplication results are rounded up “hopefully” as many times as they are rounded down. As an example, Fig. 4 shows a drift, which is due to the accumulated quantization noise in the interpolated fixed-point samples compared to the floating-point results, for a Weichselberger channel model. Fig. 4 also shows the improvement in reducing the drift and the effectiveness of the rounding technique. As this figure shows, no clear DC bias (or drift) can be observed after two million samples when rounding the multiplier outputs.

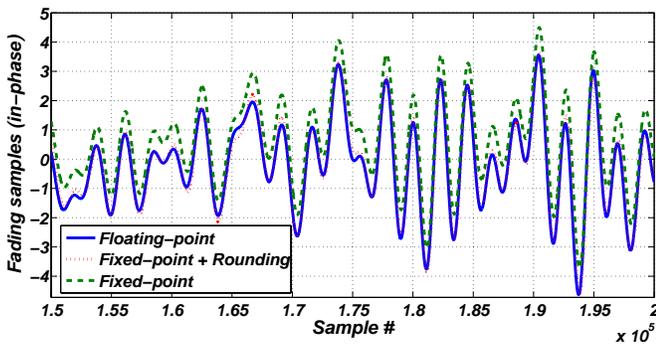


Fig. 4. The effect of rounding on the interpolator output.

We added the rounding functionality to the matrix processor and simulated different analytical channel models and channel conditions. The fixed-point bit-true model was verified based on the computer simulation and we proceeded with the implementation of the fading channel simulator on a GVA-290 FPGA prototyping platform [36]. The fading simulator was set to generate 50 million samples per second. The generated fading samples seemed to have the required statistical properties. However, the mean and variance of the

generated fading samples started to deviate from theoretical results after a few minutes. The direction of deviation for quadrature components of fading samples were related to the simulated scenario and specific values of the  $\mathbf{U}$ ,  $\mathbf{W}$ , and  $\mathbf{V}$  matrices. In one case, a 12% change in the signal variance occurred after five minutes of sample generation, i.e.,  $5 \times 60 \times 50 \times 10^6 = 1.5 \times 10^{10}$  generated fading samples. This deviation happened very slowly in time and could not be easily predicted with the software-based simulations due to the relatively slow computer simulation speed. It was found that the deviation is caused by accumulation of the DC component of quantization noise in the interpolators. More specifically, the assumption that samples are rounded up as many times as they are rounded down on average was not accurate. Even the slightest DC components in the quantization noise accumulate in the interpolator and can render it unstable over long periods of time.

To solve this problem, we modified the interpolator to remove the DC component from the output. The original interpolator is a simple accumulator performing a running sum operation. Fig. 5 shows the datapath of the new interpolator, where the fixed-point format of the signals is denoted using the notation  $(WL, WF)$ , where  $WL$  and  $WF$  denote the wordlength and the fraction length of the variables, respectively. Only the in-phase branch is shown in this figure. The main modification in the new interpolator is the addition of negative feedback to the accumulator N2 (or integrator). In this feedback loop, the sign of the accumulator output (from the most significant bit) is fed back and subtracted from half of the least significant bit of the input. More specifically, 12-bits have been used to represent the fraction part of the difference input  $d[n]$ . The magnitude of the feedback signal is limited to half of the least significant bit of the input. The accumulator output is later divided by the interpolation factor  $I$  in the barrel shifter N3. This would further reduce the relative amount of feedback to the output signal.

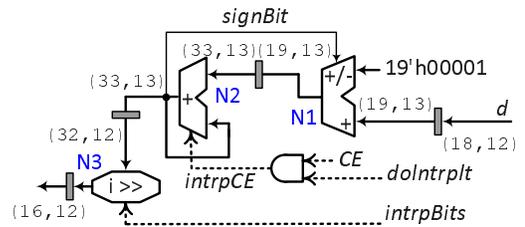


Fig. 5. Datapath of the new interpolator with DC cancellation.

Fig. 6 shows the frequency response of the modified linear interpolator. In this figure, the interpolation factor is set to  $I = 16$ . As this figure shows, the addition of the negative feedback adds a sharp notch at the DC frequency. Particularly, the DC frequency has been attenuated more than 60 dB while the frequency response goes back to 0 dB at  $1.5 \times 10^{-5}$  Hz. The other effect of the added feedback is reduced attenuation at 0.5 radians per sample. However, the attenuation is still sufficient to reduce the unwanted frequency components.

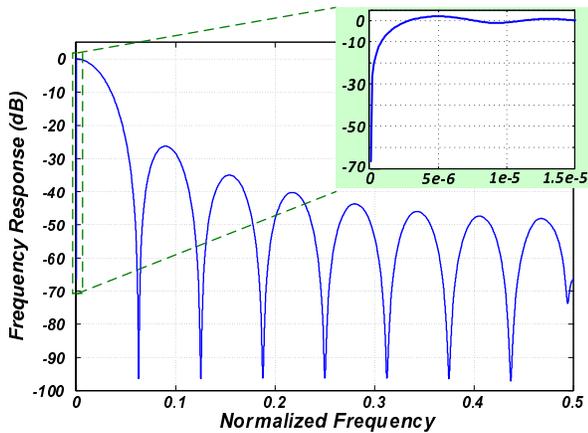


Fig. 6. Frequency response of the implemented linear interpolator.

## V. APPLICATION OF SPATIO-TEMPORALLY FADING CHANNELS IN A BER MEASUREMENT SYSTEM

The analytical MIMO fading channel simulator (including the fading variate generator, matrix processor, and the interpolator) is implemented on different FPGAs. The implemented MIMO fading channel simulator can simulate i.i.d., Kronecker, Weichselberger, and VCR analytical channel models. When implemented on a Xilinx Virtex-5 XCVLX110-3 FPGA, the  $2 \times 2$  matrix processor occupies 2016 (2.9%) of configurable slices, and two DSP48E (3.1%), and can operate at up to 234 MHz. The  $4 \times 4$  matrix processor occupies only 1212 (1.8%) configurable slices, two DSP48E (3.1%), and four 36-Kbit block memories (3.1%) and can operate at the same 234 MHz. The sub-interpolator in Fig. 5 utilizes only 96 (0.1%) of the configurable slices and can operate at up to 448 MHz when implemented on the same FPGA. Notice that the maximum speed of the sub-interpolator corresponds to the maximum sample generation rate of the fading simulator.

For testing and validation of the designed and implemented simulator, we utilized the hardware-based analytical MIMO fading channel in a developed BER performance measurement system [6], [7], [18], [35]. Fig. 7 shows the block diagram of this BERT, which supports single-input single-output (SISO) and  $2 \times 2$  MIMO systems. In the implemented BERT, source bits are encoded using an extended binary (24,12) Golay channel code [37], interleaved with a length 16383 pseudo-random interleaver [38], and modulated using 4-QAM symbols. Then they are passed through the MIMO channel, where they are affected by correlated MIMO fading variates and corrupted with additive white Gaussian noise (AWGN). The fading simulator can be configured to simulate both single- and multiple-antenna systems. In the receiver, a maximum likelihood (ML) detector tries to estimate the transmitted bits. After ML detection, the bit stream is de-interleaved, decoded, and compared to the transmitted bit stream. We also developed a graphical user interface (GUI) through which the BERT can be easily configured for different test scenarios. For example, we can set the spatial correlation parameters of the analytical MIMO fading channel models, vary the signal-to-noise ratio (SNR), and change the noise variance.

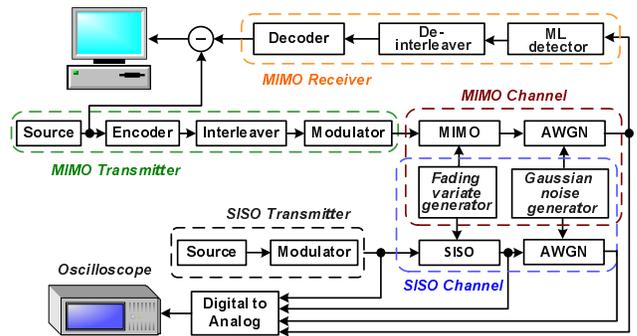


Fig. 7. Block diagram of the implemented BERT system.

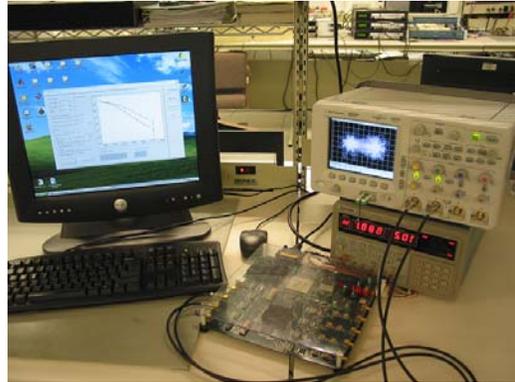


Fig. 8. BER measurement system using implemented MIMO fading channel simulator on a single FPGA.

We implemented a  $2 \times 2$  MIMO communication system on a GVA-290 FPGA development platform [36]. Fig. 8 shows the picture of the GVA-290 board along with the power source, oscilloscope, and the control computer. The GVA-290 board is interfaced with the control computer through the parallel port. This board hosts two Xilinx VirtexE XCV2000E FPGAs in addition to four 100 Msps digital-to-analog and four 100 Msps analog-to-digital converters. Since the VirtexE FPGAs are relatively small with no embedded multipliers, we tried to optimize the size of our communication system with the minimum number of multipliers.

To estimate the BER performance of a communication system with the Monte Carlo (MC) simulation method, we have to measure the BER over a large number of independent problem instances [39]. While simulation of AWGN channels is straightforward as the system performance is averaged over a large number of independent instances of noise and data, BER performance measurement of digital communication systems over time-varying fading channels requires significantly longer simulation times due to the dependence between the channel instances. To accurately estimate the BER performance of a communication system over a time-varying fading channel, the error performance needs to be averaged not only on independent instances of noise and data, but also on the fading channel samples over a long period of time (compared to the channel coherence time  $T_c \approx 0.423/f_D$  [19]). Fig. 9 shows the BER performance of this  $2 \times 2$  MIMO system for different channel models and also for the floating-

point computer simulation using the i.i.d. channel model. In this platform, the sample rate is set to 3.125 million samples per second or 12.5 Mbps (the maximum speed supported by the ML detector), and the Doppler frequency is  $f_D = 350$  Hz. For the Kronecker channel model, the  $\mathbf{U}$  and  $\mathbf{V}$  matrices are set to

$$\mathbf{U}_K = \begin{pmatrix} -0.2281 - j0.6045 & 0.0659 + j0.6270 \\ -0.8782 + j0.6279 & 0.1516 - j0.0198 \end{pmatrix},$$

$$\mathbf{V}_K = \begin{pmatrix} 0.1914 - j0.3442 & -0.1091 - j0.0796 \\ 0.1021 + j1.2781 & 0.4248 + j0.0667 \end{pmatrix}.$$

Also, for the Weichselberger channel model, the values

$$\mathbf{U}_W = \begin{pmatrix} 0.0713 - j0.3103 & 0.2119 - j0.9238 \\ -0.9478 & 0.3184 \end{pmatrix},$$

$$\mathbf{W}_W = \begin{pmatrix} 0.3176 & 0.6355 \\ 0.9077 & 1.6340 \end{pmatrix},$$

$$\mathbf{V}_W = \begin{pmatrix} 0.8012 + j0.5202 & -0.2468 - j0.1602 \\ 0.2941 & 0.9556 \end{pmatrix},$$

are used in the simulator. Moreover, for the VCR channel models we set these matrices to

$$\mathbf{U}_V = \begin{pmatrix} 0.7679 + j0.0821 & -0.6316 - j0.0676 \\ 0.6352 & 0.7723 \end{pmatrix},$$

$$\mathbf{W}_V = \begin{pmatrix} 1.6356 & 0.4370 \\ 1.0452 & 0.2035 \end{pmatrix},$$

$$\mathbf{V}_V = \begin{pmatrix} 0.2519 + j0.7142 & 0.6530 \\ -0.2172 - j0.6158 & 0.7573 \end{pmatrix}.$$

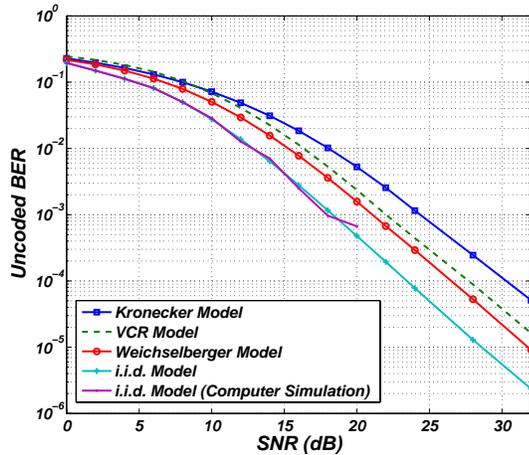


Fig. 9. Bit error rate performance of a  $2 \times 2$  MIMO system with 4-QAM modulated symbols and a ML detector measured using the FPGA-based fading channel simulator for different channel models.

To estimate each BER point at each SNR, we measured the performance over at least 1023 seconds of signal transmission on the hardware platform, and when at least 100 errors were collected from the detector output. Note that due to the extremely slow software-based simulation of MIMO communication systems over time-varying fading channel models, the computer simulation results were given up at 20 dB SNR. This is due to the great computational complexity of the bit-true BER performance measurement using realistic fading

channel models. In fact, 10 seconds of BER measurements using our hardware platform take more than three days of a bit-true software simulation in C running on a 3.6-GHz dual-core Pentium processor with 1 GB of RAM. This corresponds to a speed-up of over 25,000. As this figure shows, the hardware generated BER performance simulation results follow the computer generated floating-point simulation results, which verifies the accuracy of our hardware fading simulation platform. Note that the floating-point BER simulation results for three of the analytical fading channels are not shown as they overlap and are indistinguishable from the fixed-point BER simulation results.

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 validation 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 adapted to faster and more recent FPGA boards for rapid prototyping of wireless communication system in baseband and intermediate frequency.

## VI. CONCLUSIONS

Hardware prototyping of the physical layer is becoming an indispensable technique in the design and validation of rapidly-evolving modern wireless communication systems. This article presented the design and implementation of a compact and accurate spatio-temporally correlated fading variate generator for multiple-input multiple-output (MIMO) channels on a single low-cost field-programmable gate array (FPGA). The presented baseband simulator is versatile and can be configured to simulate different channel models, including the independent and identically distributed model, the Kronecker model, the Weichselberger model, and the virtual channel representation model. The MIMO fading channel simulator was successfully used for the real-time bit error rate (BER) performance validation of a  $2 \times 2$  MIMO wireless system on the same FPGA. The BER performance measurement of a typical  $2 \times 2$  MIMO system was accelerated by four orders of magnitude compared to bit-true software-based simulations.

## REFERENCES

- [1] J. Berkman *et al.*, "On 3G LTE terminal implementation – Standard, algorithms, complexities and challenges," in *IEEE Wireless Communications and Mobile Computing Conference*, 2008, pp. 970–975.
- [2] A. Alimohammad, S. F. Fard, and B. F. Cockburn, "Hardware-based error rate testing of digital baseband communication systems," in *IEEE International Test Conference (ITC)*, 2008, pp. 1–10.
- [3] A. Alimohammad, S. F. Fard, and B. F. Cockburn, "A flexible layered architecture for accurate digital baseband algorithm development and verification," in *Design, Automation and Test in Europe (DATE)*, 2009, pp. 45–50.
- [4] V. Singh, A. Root, E. Hemphill, N. Shirazi, and J. Hwang, "Accelerating bit error rate testing using a system level design tool," in *IEEE Symp. on FCCM*, 2003, pp. 62–68.
- [5] Y. Guo and D. McCain, "Compact hardware accelerator for functional verification and rapid prototyping of 4G wireless communication systems," in *Asilomar Conference on Signals, Systems and Computers*, 2004, pp. 767–771.

- [6] A. Alimohammad, S. F. Fard, and B. F. Cockburn, "FPGA-accelerated baseband design and verification of broadband MIMO wireless systems," in *IEEE International Conference on Advances in System Testing and Validation (VALID)*, 2009, pp. 135–140.
- [7] A. Alimohammad, S. F. Fard, and B. F. Cockburn, "FPGA-based accelerator for the verification of leading-edge wireless systems," in *IEEE International Design Automation Conference (DAC)*, 2009, pp. 844–847.
- [8] *Baseband Studio for Fading*, Technical Overview, N5115A, Agilent Technologies Inc., 2005.
- [9] *Baseband Fading Simulator ABFS, Saving costs through baseband simulation*, Rohde & Schwarz, 2004.
- [10] *Multi-Channel Emulator, EB Prosim C8*, Elektrobit Inc., Oulunsalo, Finland, 2008.
- [11] *ACE 400NB MIMO Channel Emulator*, Azimuth Systems Inc., Acton, MA, USA, 2006.
- [12] F. Ren and Y. R. Zheng, "A novel emulator for discrete-time MIMO triply selective fading channels," *IEEE Trans. Circuits and Systems-I: Regular Papers*, vol. 57, no. 9, pp. 2542–2551, 2010.
- [13] H. Eslami, S. V. Tran, and A. M. Eltawil, "Design and implementation of a scalable channel emulator for wideband MIMO systems," *IEEE Trans. on Vehicular Technology*, vol. 58, no. 9, pp. 2542–2551, 2009.
- [14] A. Alimohammad, S. F. Fard, and B. F. Cockburn, "An accurate MIMO fading channel simulator using a compact and high-throughput reconfigurable architecture," *IET Communications*, vol. 5, pp. 844–852, 2011.
- [15] S. F. Fard, A. Alimohammad, and B. F. Cockburn, "An FPGA-based simulator for high path count Rayleigh and Rician fading," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 6, pp. 2725–2734, 2010.
- [16] A. Alimohammad, S. F. Fard, B. F. Cockburn, and C. Schlegel, "A single-FPGA multipath MIMO fading channel simulator," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2008, pp. 308–311.
- [17] B. Holter and G. E. Oien, "On the amount of fading in MIMO diversity systems," *IEEE Trans. Wireless Commun.*, vol. 4, no. 5, pp. 2498–2507, 2005.
- [18] A. Alimohammad, S. F. Fard, and B. F. Cockburn, "Reconfigurable performance measurement system-on-a-chip for baseband wireless algorithm design and verification," *IEEE Wireless Communications*, vol. 19, no. 6, pp. 84–91, 2012.
- [19] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall, 2002.
- [20] J. P. Keramoal, L. Schumacher, P. E. Mogensen, and K. I. Pedersen, "A stochastic MIMO radio channel model with experimental validation," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 6, pp. 1211–1226, 2002.
- [21] V. Erceg et al., *IEEE 802.11 document 03/940r4 (TGN Channel Models)*, Garden Grove, California, May 2004.
- [22] P. Almers et al., "Survey of channel and radio propagation models for wireless MIMO systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 6, no. 2, pp. 100–106, 2007.
- [23] A. A. M. Saleh and R. A. Valenzuela, "A statistical model for indoor multipath propagation," *IEEE J. Sel. Areas in Commun.*, vol. 5, no. 2, pp. 128–137, 1987.
- [24] J. P. Keramoal et al., "A stochastic MIMO radio channel model with experimental validation," *IEEE J. Sel. Areas in Commun.*, vol. 20, no. 6, pp. 1211–1226, 2002.
- [25] W. Weichselberger et al., "A stochastic MIMO channel model with joint correlation of both link ends," *IEEE Transactions Wireless Commun.*, vol. 5, no. 1, pp. 90–99, 2006.
- [26] A. M. Sayeed, "Deconstructing multiantenna fading channels," *IEEE Transactions Signal Process.*, vol. 50, no. 10, pp. 2563–2579, 2002.
- [27] W. C. Jakes, *Microwave Mobile Communications*. Piscataway, NJ: Wiley-IEEE Press, 1994.
- [28] M. Debbah and R. R. Müller, "MIMO channel modeling and the principle of maximum entropy," *IEEE Transactions on Inf. Theory*, vol. 51, no. 5, pp. 1667–1690, 2005.
- [29] W. C. Jakes, *Microwave Mobile Communications*. Piscataway, NJ: Wiley-IEEE Press, 1974.
- [30] M. Pätzold, *Mobile Fading Channels*. West Sussex, U.K.: Wiley, 2002.
- [31] M. Kiessling and J. Speidel, "Statistical transmit processing for enhanced MIMO channel estimation in presence of correlation," in *IEEE Global Telecommunications Conference*, 2003, pp. 2411–2415.
- [32] S. F. Fard and A. Alimohammad and B. F. Cockburn, "High path-count multivariate Rayleigh fading channel simulator with time-multiplexed datapath," in *IEEE International System on a Chip Conference (SOCC)*, 2009, pp. 271–274.
- [33] A. Alimohammad, S. F. Fard, and B. F. Cockburn, "Hardware implementation of Nakagami and Weibull variate generators," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 99, no. 4, pp. 1–9, 2011.
- [34] *TS 136.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation*, Technical Specification, Version 10, Release 10, 3rd Generation Partnership Project (3GPP), ETSI, 2011.
- [35] S. F. Fard, A. Alimohammad, and B. F. Cockburn, "A versatile fading simulator for on-chip verification of MIMO communication systems," in *IEEE International System on a Chip Conference (SOCC)*, 2009, pp. 412–415.
- [36] *GVA-290 Xilinx VirtexE Hardware Accelerator*, GV & Associates Inc., 2009.
- [37] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2004.
- [38] B. Sklar, *Digital Communications, Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice Hall, 1993.
- [39] M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, *Simulation of Communication Systems: Modeling, Methodology, and Techniques*. New York: Kluwer Academic Publishers, 2000.



**Amirhossein Alimohammad** is an Assistant Professor in the Electrical and Computer Engineering Department at the San Diego State University. He was the Co-Founder and Chief Technology Officer of Ukalta Engineering in Edmonton, Canada, from 2009-2011. He was a Postdoctoral Fellow at the University of Alberta between 2007-2009. He obtained a Ph.D. degree in Electrical and Computer Engineering from the University of Alberta in Canada and a M.Sc. degree from the University of Tehran in Iran. Before starting his Ph.D., he was a Hardware Engineer at Get2Chip GmbH, a Research Fellow in the Institute of Microelectronics at the University of Ulm and Atmel Wireless in Germany. His research interests include digital VLSI systems, reconfigurable architectures, wireless communication circuits, and signal processing algorithms.



**Saeed Fouladi Fard** received a B.Sc. degree in electrical engineering from the Faculty of Communications, Tehran, Iran, in 2000, a M.Sc. degree in electrical engineering from the University of Tehran, Iran, in 2003, and a Ph.D. degree in electrical and computer engineering from the University of Alberta, Edmonton, Canada, in 2009. He was the VP of Engineering at Ukalta Engineering, Edmonton, Alberta, Canada. His general research interests include signal processing, communications, reconfigurable computing, and VLSI design and testing. His current research focuses on multiple antenna transceivers, fading simulation, and efficient hardware computation techniques.