Hardware Emulation of Wideband Correlated Multiple-Input Multiple-Output Fading Channels

Fei Ren, *Student Member, IEEE*, and Yahong Rosa Zheng, *Senior Member, IEEE*

Abstract

A low-complexity hardware emulator is proposed for wideband, correlated, multiple-input multiple-output (MIMO) fading channels. The proposed emulator generates multiple discrete-time channel impulse responses (CIR) at the symbol rate and incorporates three types of correlation functions of the subchannels via Kronecker product: the spatial correlation between transmit or receive elements, temporal correlation due to Doppler shifts, and inter-tap correlation due to multipaths. The Kronecker product is implemented by a novel mixed parallel-serial (mixed P-S) matrix multiplication method to reduce memory storage and meet the real-time requirement in high data-rate, large MIMO size, or long CIR systems. We present two practical MIMO channel examples implemented on an Altera Stratix III EP3SL150F FPGA DSP development kit: a 2-by-2 MIMO WiMAX channel with a symbol rate of 1.25 million symbols/second and a 2-by-6 MIMO underwater acoustic channel with 100-tap CIR. Both examples meet real-time requirement using only 12–14 percent of hardware resources of the FPGA.

Index Terms

Correlated MIMO Fading Channel, Channel Emulator, WiMAX Channel, Underwater Acoustic Channel, FPGA Implementation.

Part of the work has been presented at IEEE Int'l Conf. Audio, Speech, Signal Process (ICASSP) 2010.

F. Ren and Y. R. Zheng are with Department of Electrical & Computer Engineering, Missouri University of Science and Technology, Rolla, MO, 65409, USA. Email: {frrf4, zhengyr}@mst.edu.

The work is supported in part by ONR under grant N00014-07-1-0219 and by NSF under grant ECCS-0846486.

I. Introduction

HARDWARE fading channel emulators provide a fast and low-cost method for testing and verifying new algorithm design, transceiver performance, and channel capacity analysis [1]–[3]. Many products are available commercially for emulating single-input single-output (SISO) or multiple-input multiple-output (MIMO) fading channels. For example, the NoiseCom MP-2500 multipath fading emulator can emulate SISO frequency-selective fading channels with up to 12 delay paths. The Agilent N5115B baseband studio test set is featured with standards-based fading configurations and can support fading channels with up to 48 delay paths. The Rohde&Schwarz ABFS simulator offers two independent six-path baseband fading channels with pre-programmed fading models in mobile radio standards [4]. The Azimuth ACE-400WB supports up to 4-by-4 MIMO fading channels in real time with antenna correlation [5]. The Elektrobit's Propsim F8 RF channel emulator can support up to 16 MIMO fading channels with various radio interfaces such as 802.11n, 3GPP LTE, WiMAX, and Wi-Fi [6]. Although most of them are equipped with advanced features such as fading channel profiles specified by current standards, bi-directional channel modeling, RF interfaces, etc., these existing emulators only provide multiple independent fading subchannels with the temporal correlation function implemented through Doppler spectrum filtering or Sum of Sinusoids (SoS). However, practical MIMO fading channels usually exhibit all three types of correlation functions, referred to as triply-selective channels: time-selectivity due to Doppler (described by temporal correlation), frequency-selectivity due to multipath (described by inter-tap correlation), and space-selectivity (associated with the spatial correlation of transmitters and receivers) [3]. It has been shown in [1] that these correlation functions have significant impact on channel capacity, bit error rate (BER), and transceiver design. Ignoring these correlation functions will lead to impractical testing results.

Incorporating correlation functions into fading subchannels is the key but difficult aspect of accurately generating correlated MIMO fading channels. Many software-based channel simulators [1]–[3], [7]–[10] have successfully simulated doubly-selective or triply-selective correlated fading channels, and they provide the theoretical foundation for hardware-based channel emulators. Recently, research on hardware-based channel emulators for doubly-selective fading channels are reported in [11]–[15], where [11]–[13] propose frequency-selective SISO fading channel emulators, and [14], [15] report MIMO fading channel emulators without spatial or/and inter-tap correlation.

Recently, we developed a hardware-based MIMO fading channel emulator [16] incorporating all three types of correlation functions based on the software simulation method in [3]. This emulator [16] computes the three correlation matrices in the hardware and can emulate a baseband MIMO triply-selective fading channel with $(M \times N \times L)$=160, where M is the number of receive elements, N is the number of transmit elements, and L is the number of taps. It is more challenging for such a correlated MIMO fading channel emulator to meet the real-time requirement in high data-rate, large MIMO size, or long channel impulse response (CIR) fading channels.

In this paper, we improve the MIMO fading channel emulator in [16] through a novel mixed parallel-serial (*mixed P-S*) multiplication structure and two sets of Ping-Pong buffers to achieve real-time implementations of large-dimension MIMO channels. The new emulator is capable of generating MIMO baseband equivalent fading channels with up to $(M \times N \times L)$=1600. This is equivalent to either 1600 independent frequency-flat fading channels, or 16 SISO frequency-selective fading channels with 100 taps each, or a N-by-M $(MN < 16)$ triply-selective fading channel with 100 taps per subchannel. To demonstrate the capability and accuracy of the emulator, two typical MIMO fading channel examples: a 2-by-2 WiMAX channel with a short symbol duration time 0.1 μs and a 2-by-6 underwater acoustic channel with 100 taps CIRs, are implemented on a Stratix III EP3SL150F FPGA DSP development kit, and their outputs are proved to have accurate correlation properties. Less than 15 percent of the hardware resource is required in these two examples and real-time requirements are met. The proposed MIMO channel emulators are tested via Hardware-in-Loop (HIL) models in Simulink.

II. The Mathematic Model

The mathematic model of the proposed emulator is the discrete-time MIMO triply selective fading model in [3]. Consider a MIMO channel with N transmit and M receive elements. The input-output relationship of the channel in the discrete-time domain is described as

$$\mathbf{y}(k) = \sum_{l=-L_1}^{L_2} \mathbf{H}(k,l) \cdot \mathbf{x}(k-l) + \mathbf{v}(k), \tag{1}$$

where the superscript $(\cdot)^t$ is the transpose operator of a matrix or vector, $\mathbf{x}(k) = [x_1(k), x_2(k), ..., x_N(k)]^t$ is the transmitted signal vector, $\mathbf{y}(k) = [y_1(k), y_2(k), ..., y_M(k)]^t$ is the received signal vector, and $\mathbf{v}(k) = [v_1(k), v_2(k), ..., v_M(k)]^t$ is the background white Gaussian noise. Note that we assume the symbol duration being

T_s. The variables L_1 and L_2 are nonnegative integers representing the range of delay taps, and derive that the total channel length is $L = (L_1 + L_2 + 1)$ taps. The MIMO channel matrix $\mathbf{H}(k,l)$ at time index k and delay tap l is defined by

$$\mathbf{H}(k,l) = \begin{pmatrix} h_{1,1}(k,l) & \cdots & h_{1,N}(k,l) \\ \vdots & \ddots & \vdots \\ h_{M,1}(k,l) & \cdots & h_{M,N}(k,l) \end{pmatrix} \tag{2}$$

For the convenience of description, we reshape the matrix $\mathbf{H}(k,l)$ to $(MNL) \times 1$ coefficient vector as

$$\mathbf{h}_{vec}(k) = [\mathbf{h}_{1,1}(k), ..., \mathbf{h}_{1,N}(k) \mid ... \mid \mathbf{h}_{M,1}(k), ..., \mathbf{h}_{M,N}(k)]^t \tag{3}$$

where $\mathbf{h}_{m,n}(k)$ is the complex coefficient vector of the (m,n)-th subchannel at time index k given by $\mathbf{h}_{m,n}(k) = [h_{m,n}(k,-L_1),\ ..., h_{m,n}(k,L_2)]$. Based on the software model in [3], the vector $\mathbf{h}_{vec}(k)$ can be generated by

$$\mathbf{h}_{vec}(k) = \mathbf{C}_{\mathbf{h}}^{\frac{1}{2}}(0) \cdot \mathbf{\Phi}(k) = (\mathbf{\Psi}_{Rx}^{\frac{1}{2}} \otimes \mathbf{\Psi}_{Tx}^{\frac{1}{2}} \otimes \mathbf{C}_{\mathbf{ISI}}^{\frac{1}{2}}) \cdot \mathbf{\Phi}(k) \tag{4}$$

where $\otimes$ denotes the Kronecker product and $\mathbf{X}^{\frac{1}{2}}$ is the square root of matrix $\mathbf{X}$ such that $\mathbf{X} = \mathbf{X}^{\frac{1}{2}} \cdot (\mathbf{X}^{\frac{1}{2}})^h$ with the superscript $(\cdot)^h$ being the Hermitian operator. The spatial correlation matrices $\mathbf{\Psi}_{Rx}$ and $\mathbf{\Psi}_{Tx}$ are determined by properties of the transmit and receive elements, respectively, and are usually known and specified by users. The inter-tap covariance matrix $\mathbf{C_{ISI}}$ is computed according to the power delay profile using (17) in [3]. The $(MNL)\times 1$ vector $\mathbf{\Phi}(k)$ is defined as $\mathbf{\Phi}(k)$=$[Z_1(k), Z_2(k), ..., Z_{(MNL)}(k)]^t$. Each complex coefficient $Z_i(k) = Z_{c_i}(k) + jZ_{s_i}(k)$ $(i = 1, 2, ..., (MNL))$ represents one of multiple uncorrelated Rayleigh fading waveforms and can be efficiently simulated by the sum of sinusoids (SoS) method in [17], [18].

III. HARDWARE IMPLEMENTATION METHOD

For the convenience of describing hardware implementations, we define three new matrices $\mathbf{C} = \mathbf{C}_{\mathbf{ISI}}^{\frac{1}{2}}$, $\mathbf{D} = \mathbf{\Psi}_{R_x}^{\frac{1}{2}} \otimes \mathbf{\Psi}_{T_x}^{\frac{1}{2}}$, and $\mathbf{E} = \mathbf{C}_{\mathbf{h}}^{\frac{1}{2}}(0)$. The coefficients of $\mathbf{h}_{vec}(k)$ in (3) are also rearranged as $H(w,k)$ (where $w =$

$1, 2, ..., (MNL)$) and:

$$\mathbf{h}_{vec}(k) = [H(1,k), H(2,k), ..., H(MNL,k)]^t \tag{5}$$

where $H(w,k)$=$H_c(w,k) + jH_s(w,k)$.

The proposed MIMO fading channel emulator outputs $\mathbf{h}_{vec}(k)$ for N-by-M subchannels with L taps per subchannel within a symbol period. Its hardware implementation consists of five modules: a flat Rayleigh fading generator (FRFG), two Ping-Pong buffers, a correlation multiplier (CM) module, and an interpolation module, as shown in Fig. 1. The FRFG module serially generates (MNL) uncorrelated flat Rayleigh fading waveforms $Z_i(Rk)$ (for $i = 1, 2, ..., (MNL)$) with proper symbol duration T_s, maximum Doppler frequency f_d, and decimation rate R. Its outputs are separated into the real part $Z_{c_i}(Rk)$ and the imaginary part $Z_{s_i}(Rk)$.

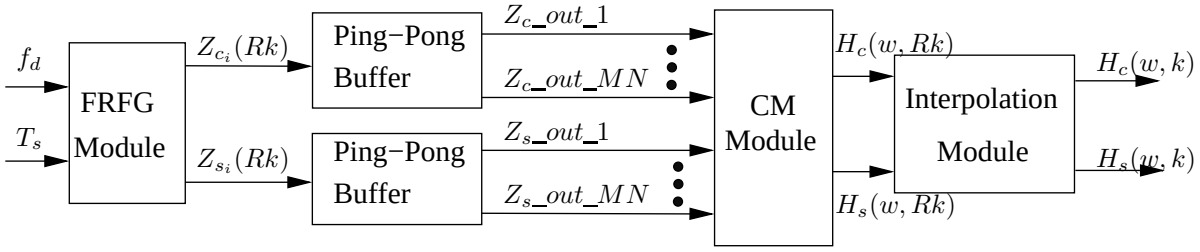

Fig. 1. Block diagram of the proposed emulator for correlated MIMO fading channels.

The Ping-Pong buffers save the serial outputs of the FRFG and convert them into parallel outputs that are required by the following CM module. Utilizing the Ping-Pong buffer ensures that only a single FRFG module is needed to provide all (MNL) uncorrelated Rayleigh fading channel waveforms. The Ping buffer and Pong buffer work alternatively to temporarily store (MNL) uncorrelated fading channel responses. At time instant k, if the Ping buffer is taking new coefficients from the FRFG, then the Pong buffer is outputting the previously stored channel coefficients to the CM module. At the next time instant $(k+1)$, the Ping buffer is ready to output, and the Pong buffer will take the new CIR coefficients. Two sets of Ping-Pong buffers are employed to buffer the real and imaginary parts of complex CIR separately. The design parameter R is carefully chosen to meet the real-time requirements.

The CM module incorporates three types of correlation functions into the uncorrelated fading channel responses via Kronecker product and vector multiplication in (4). It is memory demanding if an all-parallel structure is used,

it is time consuming if an all-serial structure is employed, especially when variables N, M, and L are large. The proposed CM module employs a *mixed P-S* method to implement matrices $\mathbf{D}$ and $\mathbf{E}$ thus drastically reducing memory requirement and processing time. We also exploit the symmetry of the matrix $\mathbf{C}_{\mathbf{ISI}}^{\frac{1}{2}}$ and employ a symmetric storage submodule to save approximate half of the memory space.

Finally, the interpolator module linearly interpolates samples with an interpolation rate R (same to the decimation rate) to output the symbol-rate fading waveforms.

The hardware implementation of the FRFG and interpolation modules are similar to those in [19] and the Ping-Pong buffers and CM module are new structures developed in this work. A brief review of the FRFG and interpolation modules and detailed structures of the Ping-Pong buffers and CM module are given in the next few subsections.

A. The FRFG

One FRFG module is utilized to generate (MNL) independent flat Rayleigh fading coefficients $Z_i(Rk)$ in series with a downsampling factor R. The SoS method [18] is employed to implement the flat Rayleigh fading waveforms via random number generator, LUT for sine and cosine functions, and multipliers and adders, as shown in Fig. 2. The SoS method generates the real and imaginary parts of the coefficients by sum of P sinusoids

$$\begin{aligned} Z_i(k) &= Z_{c_i}(k) + jZ_{s_i}(k), \\ Z_{c_i}(k) &= \sqrt{\frac{2}{P}} \sum_{p=1}^{P} \cos(2\pi(f_d k T_s \cos\alpha_{p,i} + \phi_{p,i})), \\ Z_{s_i}(k) &= \sqrt{\frac{2}{P}} \sum_{p=1}^{P} \cos(2\pi(f_d k T_s \sin\alpha_{p,i} + \varphi_{p,i})), \\ \alpha_{p,i} &= \frac{\pi(p - 0.5 + \theta_i)}{2P}, \quad p = 1, 2, \cdots, P. \end{aligned} \tag{6}$$

where f_d is the maximum Doppler frequency, P is the total number of sinusoids and $j = \sqrt{-1}$. The angle of arrival $\alpha_{p,i}$ is randomized by a θ_i. The random variables $\phi_{p,i}$ and $\varphi_{p,i}$ are the random phases of the in-phase and quadrature components, respectively. The random variables $\phi_{p,i}$, $\varphi_{p,i}$, and θ_i are statistically independent and uniformly distributed on $[-0.5, 0.5)$ for all p.

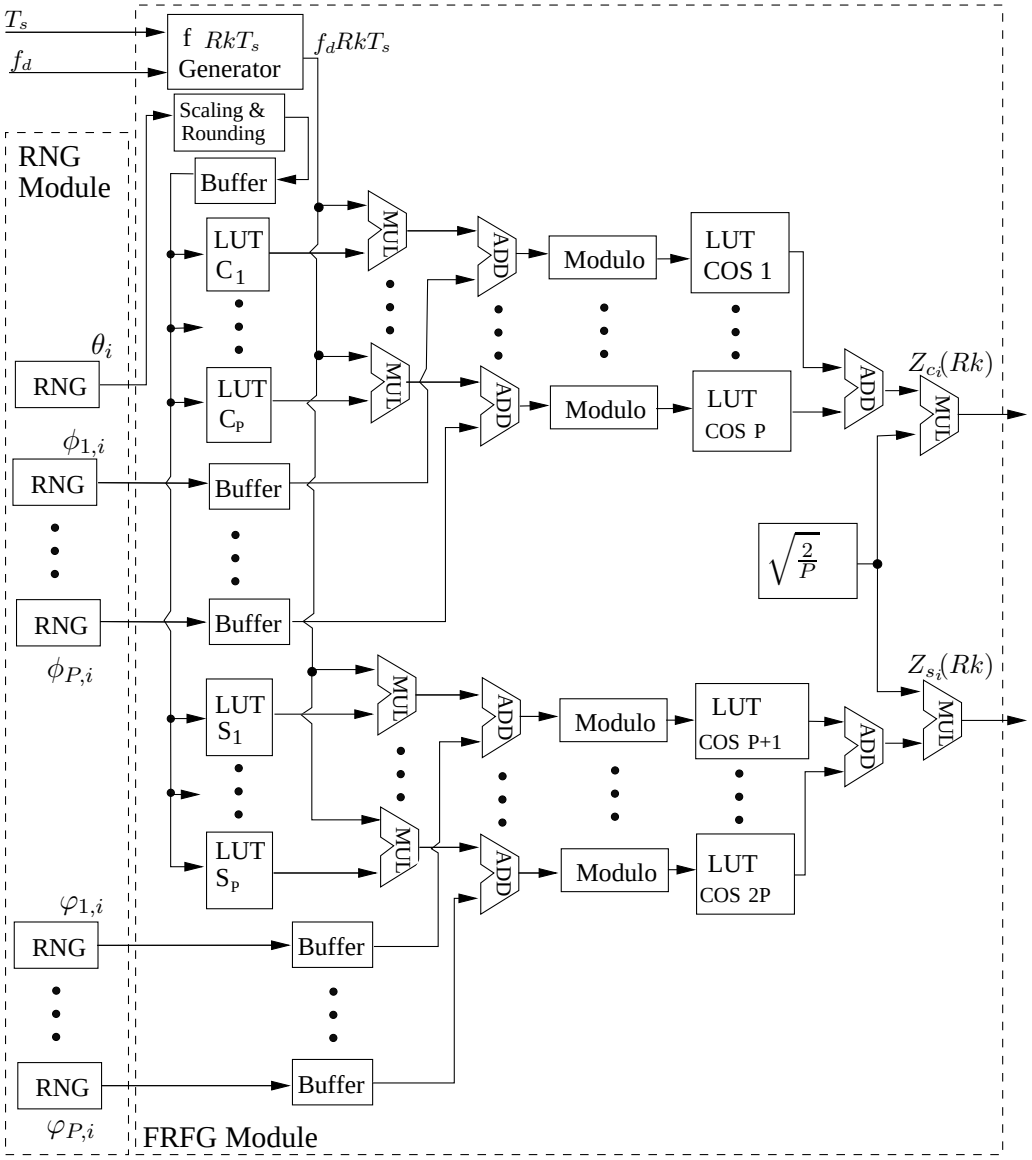

Fig. 2. Implementation blocks of the FRFG module.

B. Ping-Pong Buffers

The Ping-Pong buffers synchronize the FRFG module with the CM module and makes it possible for the single FRFG to continuously provide multiple uncorrelated Rayleigh fading channel responses for the CM module. They perform a serial to parallel data conversion via properly buffering and outputting data. Two identical sets of Ping-Pong buffers are needed to buffer the real part $Z_{c_i}(Rk)$ and the imaginary part $Z_{s_i}(Rk)$ separately. Each Ping-Pong buffer contains two banks of RAMs named $PingRAMs$ and $PongRAMs$. The block diagram of the Ping-Pong buffer storing the real part of coefficients is shown in Fig. 3.

The Ping Pong buffer contains (MN) units of RAMs and each RAM contains L words. The inputs $Z_{c_i}(Rk)$

(where $i = 1, 2, ..., (MNL)$) are fed to the Ping-Pong buffer in the following format. In a period of $(MNLL)$ clock cycles, the serial sequence: $Z_{c_1}(Rk)$, ..., $Z_{c_{MNL}}(Rk)$, is input sequentially in the first (MNL) clock cycles, and then all zeros are input in the rest of $(MNL(L-1))$ clock cycles. In the next period, the variable k is increased by one and then an updating sequence is input in the similar format. The demultiplexer *DEMUX* and up-counter *Counter Sel 1* work together to distribute coefficients $Z_{c_i}(Rk)$ into different RAM units. The up-counter, *Counter Sel 1*, increases by one in every L clock cycles to select one of the (MN) output ports of the *DEMUX*. Another up-counter, *Counter Addr 1*, generates write/read addresses for all RAMs. A periodic pulse generator with the pulse length of L clock cycles is used as the control signal "wren" for the RAMs enabling the write/read operations. The pulse is delayed by $(i-1)L$ clock cycles for the i-th Ping RAM unit, and it is delayed by $(MNLL + (i-1)L)$ clock cycles for the i-th Pong RAM unit. In Fig. 3, some connecting lines between delay blocks and their corresponding "wren" ports are not drawn to avoid increase complexity of the figure.

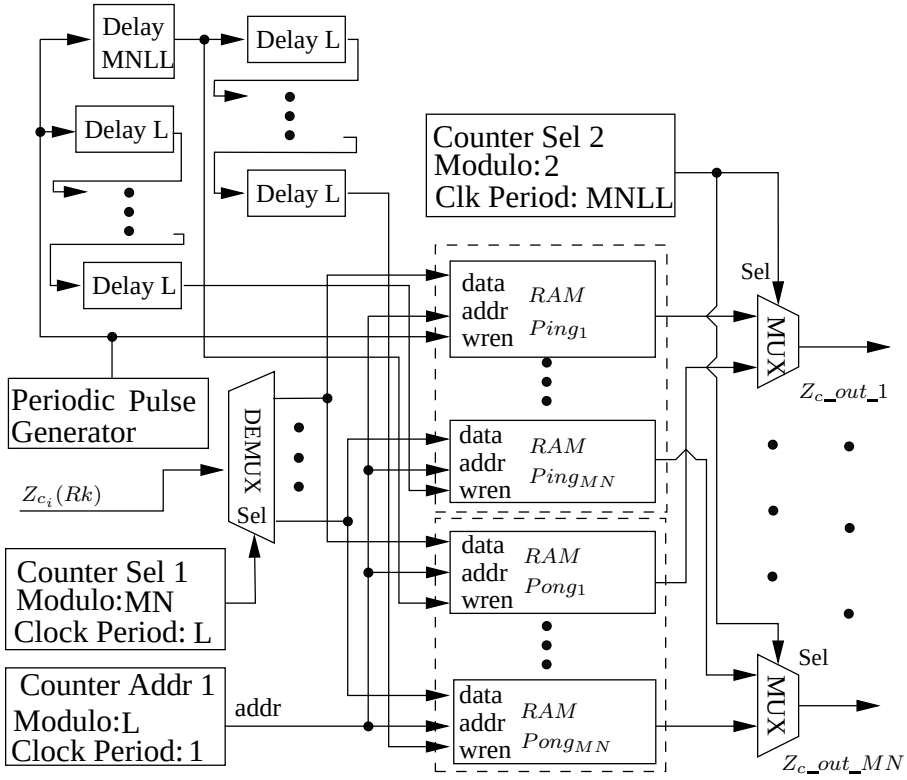

Fig. 3. Hardware implementation of the Ping-Pong buffer module. This diagram shows the data buffer for the real part $Z_{c_i}(Rk)$. The imaginary part uses a similar buffer structure.

Totally, (MN) multiplexers named *MUX* are used to select Ping RAMs or Pong RAMs to be connected to the (MN) parallel output ports named $Z_c\_out\_1 \sim Z_c\_out\_MN$. These multiplexers are controlled by the select signal

generated by the up-counter, *Counter Sel 2*. Each output port sequentially outputs real parts of L uncorrelated fading channels in the following format. In a period of $(MNLL)$ clock cycles, the output port $Z_c\_out\_i$ serially outputs the sequence: $Z_{c_{L(i-1)+1}}(Rk)$, $Z_{c_{L(i-1)+2}}(Rk)$, ..., $Z_{c_{Li}}(Rk)$, for (MNL) times. In the next period, the variable k is increased by one and then an updating sequence is output in the similar format. These outputs are fed to the CM module to be multiplied with the coefficient of matrix $\mathbf{E}$.

C. Correlation Multiplier Module

The proposed CM module is implemented by the *mixed P-S* method, as shown in Fig. 4. It employs $3(MN)$ multipliers, five adders and two accumulators, all capable of outputting results within one clock cycle. Two memory banks $RAMC$ and $RAMD_i$ stores the pre-computed coefficients of matrices $\mathbf{C}$ and $\mathbf{D}$, respectively. If the size of matrix $\mathbf{C}$ is large which is often the case in wideband systems, then only its diagonal and upper-triangular elements are stored to save memory space, thanks to its symmetric property [3]. The j-th row of matrix $\mathbf{C}$ is stored in $RAMC$ with $(L-j+1)$ coefficients. The addresses of $RAMC$ are sequentially allocated ranging from 1 to $\frac{(L+1)L}{2}$. Two up-counters, *Counter 2* and *Counter 3*, are used to generate the proper row and column indices of matrix $\mathbf{C}$, and an address convertor converts these indices into corresponding read addresses of $RAMC$. The address convertor converts these indices into corresponding read addresses which read the specified coefficients. Actually, the address convertor computes the read addresses by:

$$\text{Read Address} = (\min\{l_r, l_c\} - 1)\left(\max\{l_r, l_c\} - \frac{min\{l_r, l_c\}}{2}\right) + \max\{l_r, l_c\} \tag{7}$$

where l_r is the row index; l_c is the column index; *min*$\{\}$ and *max*$\{\}$ find the minimum and maximum values of their arguments, respectively. The address convertor and $RAMC$ build up a storage submodule that implements a symmetric storage method.

The size of matrix $\mathbf{D}$ is often small and coefficients of each column are stored in $RAMD_1$ through $RAMD_{MN}$ separately. The up-counter $Counter1$ and an adder generate the read address for $RAMD_i$ to output (MN) coefficients simultaneously. If the size (MN) is large, then a similar memory scheme as $RAMC$ maybe adopted for $RAMD_i$. In every clock cycle, the output of $RAMC$ is multiplied with the outputs of $RAM\ D_1 \sim RAM\ D_{MN}$ to obtain (MN) coefficients of matrix $\mathbf{E}$ in parallel.

The vector multiplication in (4) is implemented by multiplying the (MN) coefficients of matrix $\mathbf{E}$ with the real

and imaginary parts of the (MN) uncorrelated Rayleigh channel responses stored in the Ping-Pong buffers. Results are added together for the real and imaginary parts respectively, and then two sums are sent to the two accumulators. In a period of L clock cycles, the accumulator sums its inputs in the previous L clock cycles to obtain a single output $H_c(w, Rk)$ or $H_s(w, Rk)$. The outputs of the accumulators are down-sampled with a down-sampling rate L before outputting to the interpolation module. Finally, the interpolation module takes $H_{c/s}(w, R(k-1))$ and $H_{c/s}(w, Rk)$ to produce all coefficients of $\mathbf{h}_{vec}(k)$ in real time. It's worth nothing that the Kronecker product can be computed alternatively by $\mathbf{D} = \mathbf{\Psi}_{\mathbf{Tx}}^{\frac{1}{2}} \otimes \mathbf{C}_{\mathbf{ISI}}^{\frac{1}{2}}$ first and then $\mathbf{E} = \mathbf{\Psi}_{\mathbf{Rx}}^{\frac{1}{2}} \otimes \mathbf{D}$. The proposed *mixed P-S* method can implement this case by simply switching the contents of $RAMD$ and $RAMC$. However, the best implementation is to use $RAMC$ to store the one with the largest dimension of $\mathbf{\Psi}_{\mathbf{Rx}}^{\frac{1}{2}}$, $\mathbf{\Psi}_{\mathbf{Tx}}^{\frac{1}{2}}$, and $\mathbf{C}_{\mathbf{ISI}}^{\frac{1}{2}}$, and use $RAMD$ for the Kronecker product of the other two matrices.

In contrast to the *mixed P-S* method, the emulator in [16] employed a *serial* method and three small RAMs A, B, C to store the coefficients of the matrices $\mathbf{\Psi}_{\mathbf{Rx}}^{\frac{1}{2}}$, $\mathbf{\Psi}_{\mathbf{Tx}}^{\frac{1}{2}}$, and $\mathbf{C}_{\mathbf{ISI}}^{\frac{1}{2}}$. The emulator can meet the real-time requirement only for a small value of (MNL). The *serial* method cannot compute fast enough to meet real-time requirement when the channel has long CIRs and/or the symbol duration reduces. The *mixed P-S* method can solve this problem. It employs (MN) parallel computational paths and can compute the Kronecker product (MN) times faster than the *serial* method does. It also requires significantly less memory space and multiplier utilization than a pure parallel method that can output really fast. Therefore, the *mixed P-S* method achieves the best tradeoff between computational speed and hardware resource utilization.

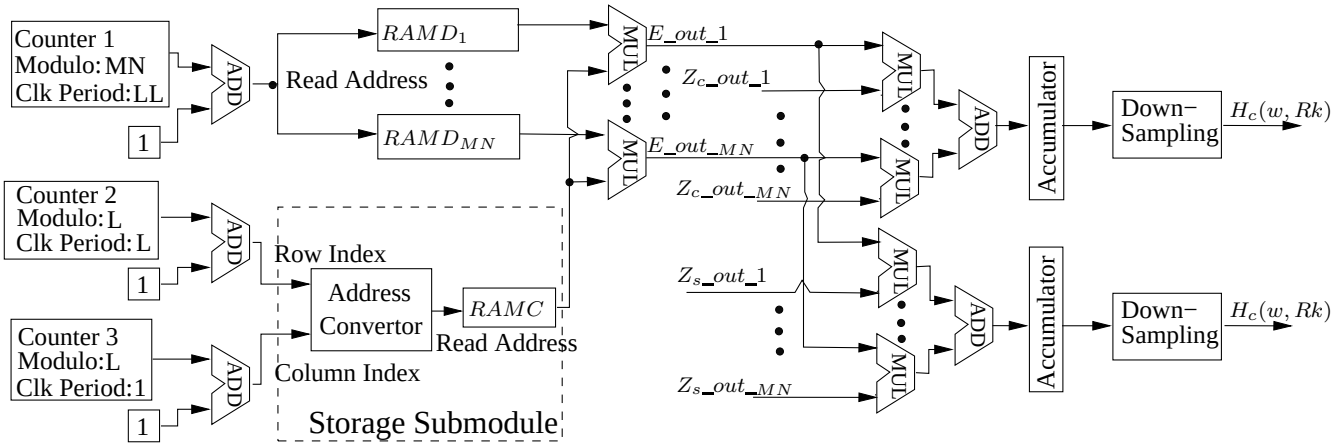

Fig. 4. Hardware implementation of CM module using the *mixed P-S* method. In this design, (MN) coefficients of matrix $\mathbf{E}$ are output in parallel per clock cycle, and one row of $\mathbf{E}$ is output in every L clock cycles. The computational speed and hardware usage are adjustable in the *mixed P-S* method.

D. Interpolator Module

The interpolator module performs a linear interpolation with a rate R to generate fading coefficients at the symbol rate. The structure of the interpolator module is shown in Fig. 5, where the inputs of the real and imaginary parts from the correlation module, $H_c(w, Rk)$ and $H_s(w, Rk)$, are processed separately in parallel through a common control logic. In every (MNL) BCPs, the enable control block controls the counter to increase from 0 to $(R-1)$ in the first R BCPs and to hold at $(R-1)$ in the remaining $(MNL-R)$ BCPs. The counter output is normalized with $1/R$. The real part input, $H_c(w, Rk)$, is delayed by $(MNL)^2$ BCPs and then subtracted from the original input. The result is multiplied with the normalized counter output and then added to the delayed input $H_c(w, R(k-1))$ to obtain the interpolated $H_c(w, k)$. The imaginary part $H_s(w, k)$ is implemented similarly.

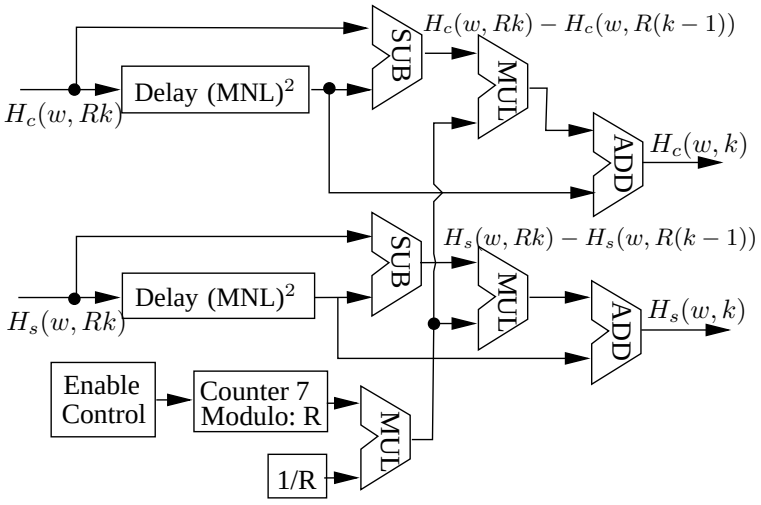

Fig. 5. Implementation of the interpolator module

IV. Implementation Examples

The proposed MIMO fading channel emulator was implemented on an Altera Stratix III EP3SL150F1152C2N FPGA/DSP development kit. The clock frequency in this implementation was F_{clk}=50 MHz, which derived a clock cycle 20 ns. We used Quartus II version 9.0, DSP Builder version 9.0, and Matlab Simulink for this development, and hardware-in-the-loop (HIL) method for testing.

Two MIMO fading channel examples were implemented on the emulator to evaluate accuracy and capability of this emulator. The first example demonstrated feasibility of the emulator in underwater communications by emulating a 2-by-6 underwater acoustic channel with long CIRs L=100 and a long symbol duration T_s=250 μs.

The second example emulated a WiMAX 2-by-2 fading channel with a short symbol duration T_s=0.8 μs and short CIRs L=5, and proved the emulator to be suitable for high data rate communication channels. In order to evaluate accuracy of this emulator, the auto/cross-correlation functions of its output waveforms were computed and compared to theoretical ones.

A. Implementation Example I - Underwater Acoustic Channel

The 2-by-6 underwater acoustic channel was implemented using the following configuration. This underwater communication system consisted of two transmit elements and six hydrophones placed as shown in Fig. 6. The angle of arrival and the angular spread were 90° and 10° respectively. The 100-tap power delay profile linearly ramped up from 0.2 to 1.8 in the first 40 taps, and then fell down from 1.8 to 0.27 in the 40-100 taps. Its total power was normalized to one. The Tx and Rx filters were square-root rased-cosine filters with a roll-off factor 0.3. The cross-correlating matrix $\mathbf{C}_{ISI}^{\frac{1}{2}}$ was computed according to (17) in [3].

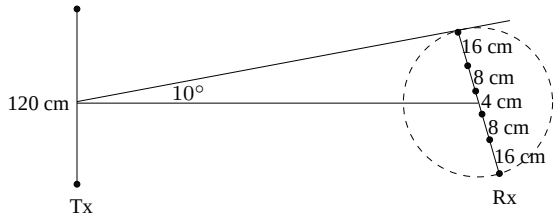

Fig. 6. The placement of transmit elements and hydrophones of the underwater communication system. This is a 2-by-6 MIMO underwater acoustic communication system where the speed of the acoustic carrier is 1500 m/s and the frequency of the carrier is 15 kHz. The wavelength of the carrier is λ=10 cm.

Other implementation parameters were selected as M=6, N=2, L=100, T_s=0.8 μs, f_d=40 Hz, and R=10. The square roots of the correlation coefficient matrices $\mathbf{\Psi}_{Tx}$ and $\mathbf{\Psi}_{Rx}$ were pre-computed and shown as:

$$\mathbf{\Psi}_{Rx}^{\frac{1}{2}} = \begin{pmatrix} 0.9881 & -0.1539 \\ -0.1539 & 0.9881 \end{pmatrix};$$

$$\mathbf{\Psi}_{Tx}^{\frac{1}{2}} = \begin{pmatrix} 0.9793 & -0.1418 & 0.0926 & -0.0725 & 0.0616 & -0.0563 \\ -0.1418 & 0.9716 & -0.1378 & 0.0901 & -0.0711 & 0.0616 \\ 0.0926 & -0.1378 & 0.9697 & -0.1369 & 0.0901 & -0.0725 \\ -0.0725 & 0.0901 & -0.1369 & 0.9697 & -0.1378 & 0.0926 \\ 0.0616 & -0.0711 & 0.0901 & -0.1378 & 0.9716 & -0.1418 \\ -0.0563 & 0.0616 & -0.0725 & 0.0926 & -0.1418 & 0.9793 \end{pmatrix}$$

Based on the outputs of the emulator, auto/cross-correlation functions of several subchannels, including the auto-correlation of $h_{1,1}(75,k)$, the cross-correlation between $h_{1,1}(75,k)$ and $h_{1,1}(76,k)$, and the cross-correlation between $h_{1,1}(75,k)$ and $h_{2,1}(75,k)$, were computed offline and plotted in Fig. 7. According to (19) in [3], their theoretical correlation functions were 0.7155, 0.1177, and –0.1774 multiplying by $J_0[2\pi f_d(k_1-k_2)T_s]$, respectively. As can be seen, the results of hardware outputs closely matched the theoretical ones.

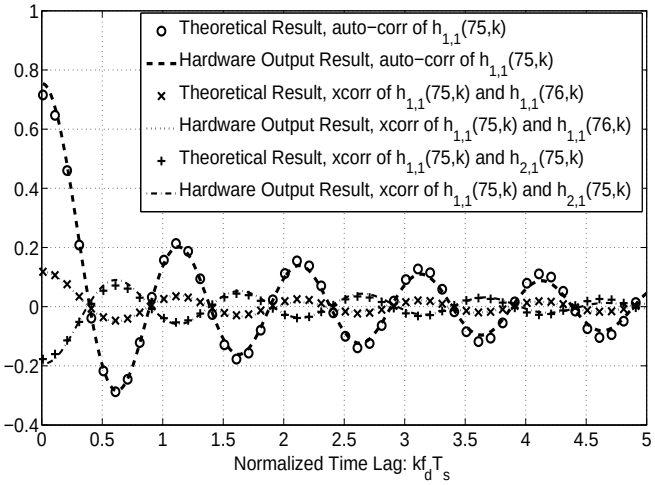

Fig. 7. Performance of underwater acoustic fading channel emulator: Auto-correlation of $h_{1,1}(75,k)$, cross-correlation between $h_{1,1}(75,k)$ and $h_{1,1}(76,k)$, and cross-correlation between $h_{1,1}(75,k)$ and $h_{2,1}(75,k)$. The channel index is according to (3). The results are based on hardware outputs of 200 trials with 2×10^3 samples pre subchannel per trial.

B. Implementation Example II - WiMAX Channel

The proposed emulator also implemented the WiMAX 2-by-2 fading channel example. The implementation parameters were selected as M=N=2, T_s=0.8μs, f_dT_s=0.001 and L=5. The angle of arrival, the angular spread, and the Tx and Rx filters were the same as those used in the underwater example. The distances between two transmit elements and two receive elements were 12λ and 0.5λ, respectively. The power delay profile contained three taps and was given by the SUI-3 model [20], which was suitable for mostly flat terrain with moderate tree densities. The coefficients of $\mathbf{\Psi}_{Tx}^{\frac{1}{2}}$, $\mathbf{\Psi}_{Rx}^{\frac{1}{2}}$, and $\mathbf{C}_{ISI}^{\frac{1}{2}}$ were pre-computed and listed as:

$$\mathbf{\Psi}_{Rx}^{\frac{1}{2}} = \begin{pmatrix} 0.9881 & -0.1539 \\ -0.1539 & 0.9881 \end{pmatrix}; \mathbf{\Psi}_{Tx}^{\frac{1}{2}} = \begin{pmatrix} 0.9941 & 0.1083 \\ 0.1083 & 0.9941 \end{pmatrix};$$

$$\mathbf{C}_{ISI}^{\frac{1}{2}} = \begin{pmatrix} 0.0005 & 0.0015 & -0.0013 & -0.0013 & 0.0036 \\ 0.0015 & 0.0044 & -0.0043 & -0.0123 & 0.0099 \\ -0.0013 & -0.0043 & 0.8776 & 0.0512 & -0.0056 \\ -0.0013 & -0.0123 & 0.0512 & 0.3601 & 0.0048 \\ 0.0036 & 0.0099 & -0.0056 & 0.0048 & 0.0252 \end{pmatrix}$$

The short symbol duration caused a higher real-time requirement that expected 2.5×10^7 complex responses to be generated per second. The short CIRs reduced computational time of Kronecker product and thus lower the real-time requirement, to some extent. Taking the short symbol duration and CIRs into consideration, we set R=3 to met the real-time requirement.

Auto/cross-correlation functions of several subchannels, including the auto-correlation of $h_{1,1}(0,k)$, the cross-correlation between $h_{1,1}(0,k)$ and $h_{1,1}(1,k)$, and the cross-correlation between $h_{1,1}(0,k)$ and $h_{2,1}(1,k)$, were computed offline and plotted in Fig. 8. Their theoretical correlation functions were 0.7728, 0.0634 and –0.0193 multiplying by $J_0[2\pi f_d(k_1 - k_2)T_s]$, respectively. As can be seen, auto/cross-correlation functions of hardware outputs matched the theoretical ones very well.

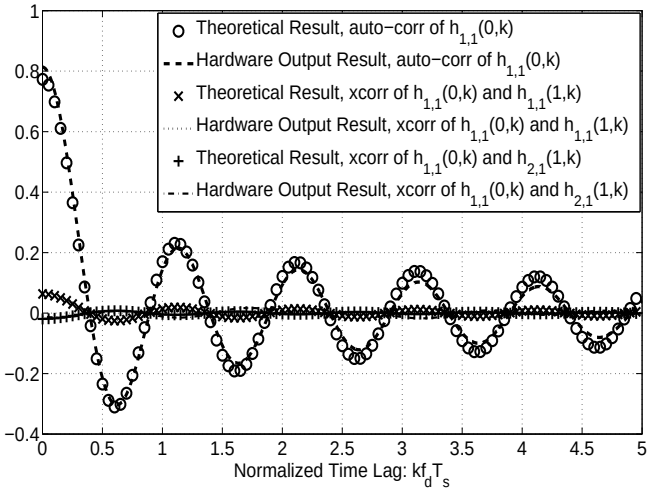

Fig. 8. Performance of WiMAX fading channel emulator: Auto-correlation of $h_{1,1}(0,k)$, cross-correlation between $h_{1,1}(0,k)$ and $h_{1,1}(1,k)$, and cross-correlation between $h_{1,1}(0,k)$ and $h_{2,1}(1,k)$. The channel index is according to (3). The results are based on hardware outputs of 50 trials with 2.8×10^4 samples per subchannel per trial.

V. Performance Evaluation

In addition to accuracy, we evaluated other performances of the proposed emulator including speed and hardware usage. The speed of this emulator was compared to the emulator in [16] which employed a *serial* method. Moveover, multipliers and memory utilization of the *mixed P-S* and *serial* methods were analyzed and compared. Finally, parameter specifications and detailed hardware usage of the proposed emulator were presented.

A. Performance Comparison of Serial and Mixed P-S methods

The proposed emulator with the *mixed P-S* method can compute $\mathbf{C}_{\mathbf{h}}^{\frac{1}{2}}(\mathbf{0})$ and generate correlated fading complex responses much faster than its counterpart in [16] with the *serial* method. The cost was higher hardware utilization, especially multipliers, which were used to construct multiple computational paths. The speed comparison for typical values of M, N, and L was shown in Fig. 9 which clearly demonstrated that the *mixed P-S* method saves a large amount of time. The y-axis indicated the number of clock cycles that were required to generate one correlated fading complex response. As can be seen, when the two methods were set to the same values of M, N, and L, respectively, the *mixed P-S* method was (MN) times faster than the *serial* method. Note that the *serial* method required more clock cycles when either $(M \times N)$ or L increased. But the *mixed P-S* method demanded more clock cycles only when L increased.

The *mixed P-S* method used more multipliers to construct parallel computational paths in the CM module; while

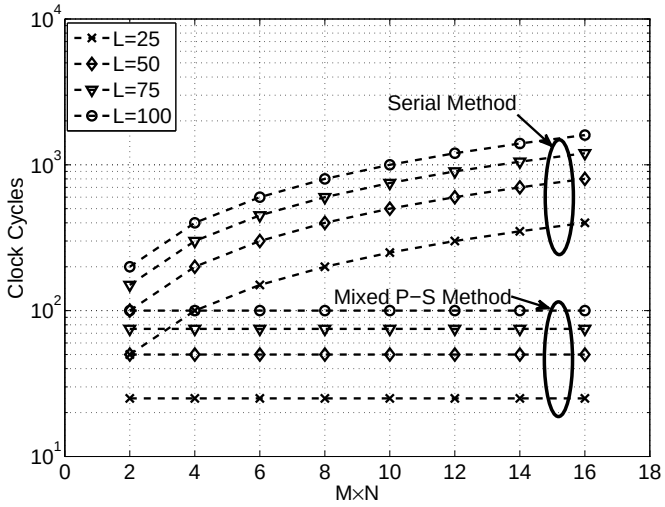

Fig. 9. The time of generating one correlated fading complex response in the *serial* and *mixed P-S* methods. The time is counted by clock cycles.

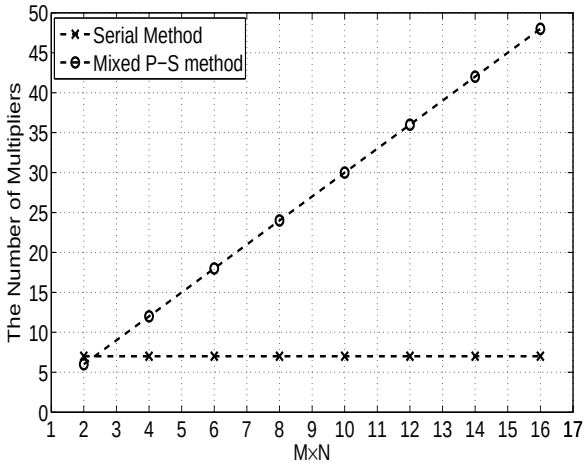

Fig. 10. The numbers of multipliers needed by the *serial* method and the *mixed P-S* method.

the *serial* method used a small constant number of multipliers. The multiplier utilization of the two methods in the CM module was shown in Fig. 10. The *serial* method employed seven multipliers to implement one serial computational path irrespective of the values of M, N, and L. The *mixed P-S* method employed a variable number of multipliers, which was equal to $(3MN)$.

When the fading channel had long CIRs, the memory usage for storing a large size matrix $\mathbf{C}_{\mathbf{ISI}}^{\frac{1}{2}}$ could be drastically reduced by making use of the symmetric property of matrix $\mathbf{C}_{\mathbf{ISI}}^{\frac{1}{2}}$. The numbers of memory words required by a full storage and the proposed symmetric storage methods were shown in Fig. 11. As can be seen, the full storage method needed L^2 words, and the symmetric storage method only needed $\frac{L(L+1)}{2}$ words that approximately saved half number of words.

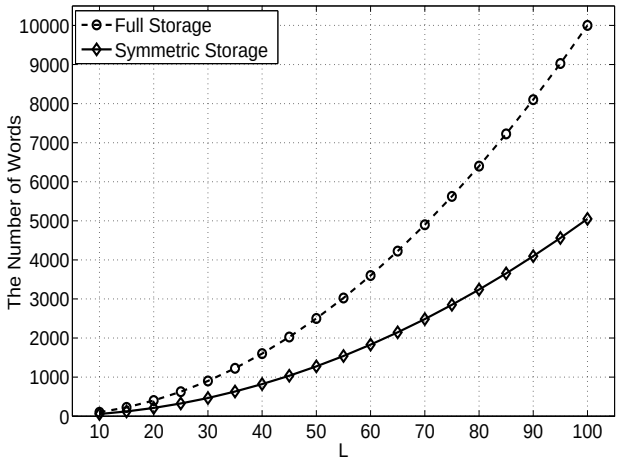

Fig. 11. The memory usage of the full storage and the proposed symmetric storage methods. The memory usage is counted by the number of words.

B. Parameter Specifications and Hardware Usage

The proposed MIMO fading channel emulator is flexible in parameter selection and can be customized to simulate channel scenarios other than the examples presented here. Table I shows the parameter ranges of the emulator with the FPGA chip clock F_{clk}=50 MHz.

TABLE I
PARAMETER RANGES OF THE PROPOSED EMULATOR WITH F_{clk}=50 MHz.

Number of Rx, and Tx	Number of taps	Normalized Doppler $T_s f_d$	Output Speed (Samples/sec)
$(MN) \leq 16$	$L \leq 100$	$1.9\times10^{-6}\sim1$	$3.125\times10^{2}\times R\times(MNL)$

According to Table I, the proposed emulator can emulate any MIMO antenna array combination of Rx and Tx up to (MN)=16, including 2×2, 2×8, 3×3, 4×4 and so on. The maximum number of channel taps L=100 covers most of practical long CIR fading channels including underwater acoustic channels. The proposed emulator stores the normalized Doppler frequency $T_s f_d$ in the Q1.19 format to ensure high accuracy $\frac{1}{2^{19}} = 1.9 \times 10^{-6}$. The emulator can generate $\frac{F_{clk}R(MNL)}{16\times100^2}$ complex samples per second. Each complex sample consists of the real and imaginary parts represented by the Q4.14 format. For the underwater acoustic channel with T_s=250 μs, the real-time requirement can be met by setting R=10. For the WiMAX channel with T_s=0.8 μs, the real-time requirement can be met by setting R=3. For channels with smaller symbol durations, the real-time requirement can be met by increasing the clock frequency and R.

The hardware usage of previous two implementation examples is summarized in Table II, where ALUT, DLR,

BM, DSP, and LU denote adaptive look-up table, dedicated logic register, block memory, DSP block (high-speed 18-bit multiplier), and overall logical utilization, respectively.

TABLE II
RESOURCE USAGE OF TWO EXAMPLES ON STRATIX III EP3SL150F1152C2N FPGA WITH F_{clk}=50 MHZ

	ALUT	DLR	BM bits	DSP	LU
Underwater	14845 (13%)	4042 (4%)	1241613 (22%)	78 (20%)	14%
WiMAX	11926 (10%)	3301 (3%)	659407 (12%)	35 (9%)	12%

Compared to the WiMAX one, the underwater example employs more hardware resources. Especially, it employs approximately double-size BMs and DSPs, since the implementations of Ping-Pong buffers, large size *RAM C*, and parallel computational paths. Note that the total logical utilizations of two examples are only 12% and 14% of the whole FPGA chip, respectively. The low hardware utilization makes it possible to implement other functional modules on the same FPGA chip.

The capability and hardware usage of the proposed emulator are compared with those of the existing emulators in Table III. The numbers of LE, memory block, and DSP elements are based on the WiMAX channel emulator with $MNL = 160$ for the proposed emulator and the one in [19]. The (MNL) for other emulators are listed in the table. It is clear that the capability of the proposed emulator is much higher than the existing ones; while the hardware usage of the proposed emulator remains very low.

TABLE III
RESOURCE USAGE COMPARISONS OF RELATED RAYLEIGH FADING CHANNEL EMULATORS.

	Proposed Emulator	Emulator in [11]	Emulator in [19]	Emulator in [14]	Emulator in [15]
Logic Unit	15227 (LE)	3557 (LC)	44240 (LE)	22272 (LE)	46357 (LC)
Block Memory	659407	Unknown	1920089	Unknown	440960
DSP Element	35	Unknown	194	Unknown	136
Rx×Tx	$M\times N$ [1]	1×1	$M\times N$ [2]	4×4	4×4
Number of Taps	L [1]	3	L [2]	9	Unknown
On-chip $C_{ISI}^{\frac{1}{2}}$ Calculator	No	No	Yes	No	No
Temporal Correlation	SoS	Spectrum filtering	SoS	Spectrum filtering	SoS
Inter-tap Correlation	Yes	Yes [3]	Yes [4]	No	Unclear
Spatial Correlation	Yes	No	Yes	No	Unclear

Note: 1. The numbers of Rx, Tx, and taps meet the relationship: $(MNL)\leq$1600.
2. The numbers of Rx, Tx, and taps meet the relationship: $(MNL)\leq$160.
3. The inter-tap correlation is implemented by upsampling to pass band.
4. The inter-tap correlation matrix $C_{ISI}^{\frac{1}{2}}$ is calculated on chip.

C. Interfacing with Digital Up-Convertor and Down-Convertor

Although the proposed MIMO fading channel emulator was tested by the HIL modules via Simulink, it can be easily integrated with the digital up-convertor and down-convertors to generate intermediate frequency (IF) channel waveforms. The IF channel waveforms can be further converted via analog mixers to generate RF channel waveforms. Altera provides several readily designed digital IF convertors for Stratix III DSP development kit as DSP Builder Simulink models [21]. The Stratix III DSP development kit has two HSMC interfaces that can interface with two daughter boards, each having two ADCs and two DACs, thus a 4-by-4 MIMO channel with IF waveforms can be easily integrated.

VI. CONCLUSIONS

A wideband MIMO fading channel emulator with accurate correlation properties has been proposed. The emulator employs a novel *mixed P-S* method to increase the speed of incorporating correlation functions. This improvement makes the emulator capable of emulating MIMO fading channels with a high data rate, large MIMO size, and long CIRs in real-time. Two MIMO fading channel examples of underwater acoustic and WiMAX have been implemented on one Altera Startix III FPGA/DSP development kit and evaluated in aspects of accuracy, speed, and hardware usage. Results exhibit that the proposed emulator employs low hardware resources and can generate accurate MIMO fading channel responses in real time.

REFERENCES

[1] P. Hoeher, "A statistical discrete-time model for the WSSUS multipath channel," *IEEE Trans. Veh. Technol.*, vol. 41, no. 4, pp. 461–468, Nov. 1992.

[2] K.-W. Yip and T.-S. Ng, "Efficient simulation of digital transmission over WSSUS channels," *IEEE Trans. Commun.*, vol. 43, no. 12, pp. 2907–2913, Dec. 1995.

[3] C. Xiao, J.X. Wu, S-Y. Leong, Y.R. Zheng, and K. B. Letaief, "A discrete-time model for triply selective MIMO Rayleigh fading channels," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1678–1688, Sep. 2004.

[4] Rohde and Schwarz, "R&S ABFS baseband fading simulator," Online: http://www2.rohde-schwarz.com/product/abfs.html.

[5] Azimuth Systems, "ACE 400WB MIMO channel emulator," Online: http://www.azimuthsystems.com.

[6] Elektrobit, "EB Propsim F8-advanced RF channel emulator," Online: http://www.elektrobit.com.

[7] A. Anastasopoulos and K.M. Chugg, "An efficient method for simulation of frequency selective isotropic Rayleigh fading," in *Proc. IEEE Veh. Technol. Conf.*, Phoenix, AZ, May. 1997, pp. 2084–2088.

[8] E. Chiavaccini and G.M. Vitetta, "GQR models for multipath Rayleigh fading channels," *IEEE Journal Selected Areas in Communications*, vol. 19, no. 6, pp. 1009–1018, Jun. 2001.

[9] B. E. Baddour and N. C. Beaulieu, "Accurate simulation of multiple cross-correlated Rician fading channels," *IEEE Trans. Commun.*, vol. 52, no. 11, pp. 1980–1987, Nov. 2004.

[10] K.E. Baddour and N.C. Beaulieu, "Autoregressive models for fading channel simulation," *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1650–1662, July. 2005.

[11] 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, Aug. 2001.

[12] M. Kahrs and C. Zimmer, "Digital signal processing in a real-time propagation simulator," *IEEE Trans. Instrum. Meas.*, vol. 55, no. 1, pp. 197–205, Feb. 2006.

[13] F. Ren and Y.R. Zheng, "A low-complexity hardware implementation of discrete-time frequency-selective Rayleigh fading channels," in *Proc. IEEE International Symposium on Circuits and Systems*, Taipei, Taiwain, May. 2009, pp. 1759–1762.

[14] M. Cui, H. Murata, and K. Araki, "FPGA implementation of 4x4 MIMO test-bed for spatial multiplexing systems," in *Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Barcelona, Spain, Sep. 2004, pp. 3045–3048.

[15] A. Alimohammad, S.F. Fard, B.F. Cockburn, and C. Schlegel, "A novel technique for efficient hardware simulation of spatiotemporally correlated MIMO fading channels," in *Proc. IEEE Int. Conf. Commun.*, Beijing, China, May. 2008, pp. 718–724.

[16] Fei Ren and Y.R. Zheng, "Hardware implementation of triply selective Rayleigh fading channel simulators," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Dallas, Texas, Mar. 2010.

[17] Y.R. Zheng and C. Xiao, "Improved models for the generation of multiple uncorrelated Rayleigh fading waveforms," *IEEE Trans. Commun. Lett.*, vol. 6, no. 6, pp. 256–258, Jun. 2002.

[18] Y. R. Zheng and C. Xiao, "Simulation models with correct statistical properties for Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 51, no. 6, pp. 920–928, Jun. 2003.

[19] F. Ren and Y.R. Zheng, "A novel emulator for discrete-time MIMO triply-selective fading channels," *IEEE Trans. Circuit Systems, Part-I*, vol. 57, no. 9, pp. 2542–2551, Sep. 2010.

[20] J. Milanovic, S. Rimac-Drlje, and K. Bejuk, "Comparison of propagation models accuracy for WiMAX on 3.5 GHz," in *Proc. 14th IEEE International Conference on Electronics, Circuits and Systems*, Marrakech, Morocco, Dec. 2007, pp. 111–114.

[21] Altera Corporation, "DSP Builder – advanced blockset with timing-driven simulink synthesis," Online: http://www.altera.com/products/software/products/dsp/adsp-builder.html.