
Doctoral Dissertations

Student Theses and Dissertations

Fall 2011

Hardware emulation of wireless communication fading channels

Fei Ren

Follow this and additional works at: https://scholarsmine.mst.edu/doctoral_dissertations



Part of the [Electrical and Computer Engineering Commons](#)

Department: **Electrical and Computer Engineering**

Recommended Citation

Ren, Fei, "Hardware emulation of wireless communication fading channels" (2011). *Doctoral Dissertations*. 1912.

https://scholarsmine.mst.edu/doctoral_dissertations/1912

This thesis is brought to you by Scholars' Mine, a service of the Missouri S&T Library and Learning Resources. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

HARDWARE EMULATION OF WIRELESS
COMMUNICATION FADING CHANNELS

by

FEI REN

A DISSERTATION

Presented to the Faculty of the Graduate School of the
MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

2011

Approved by

Yahong Rosa Zheng, Advisor
Jagannathan Sarangapani
Steve Grant
Randy H. Moss
Maggie Cheng

PUBLICATION DISSERTATION OPTION

This dissertation consists of the following five published or accepted papers, formatted in the style used by the Missouri University of Science and Technology, listed as follows:

Paper 1, F. Ren, and Y.R. Zheng, “Hardware Emulation of Wideband Correlated Multiple-Input Multiple-Output Fading Channels,” has been accepted to publish in Journal of Signal Processing Systems, Jun. 2011. Pages 11-34

Paper 2, F. Ren, and Y.R. Zheng, “A Novel Emulator for Discrete-time MIMO Triply-selective Fading Channels,” has been published in IEEE Trans. Circuits and Systems I: Regular Paper, vol. 57, pp.2542-2551, Sep. 2010. Pages 35-65

Paper 3, F. Ren, and Y.R. Zheng, “Hardware Implementation of Triply Selective Rayleigh Fading Channel Simulators,” has been published in Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP10), March 2010. Pages 66-77

Paper 4, F. Ren, and Y.R. Zheng, “A Low-complexity Hardware Implementation of Discrete-time Frequency-selective Rayleigh Fading Channels,” has been published in Proc. IEEE International Symposium on Circuits and Systems (ISCAS09), May 2009. Pages 78-89

Paper 5, S. Subedi, H. Lou, F. Ren, M. Wang, and Y. R. Zheng, “Validation of the Triply Selective Fading Channel Model Through a MIMO Test Bed and Experimental Results,” has been accepted to publish in Proc. International Conference for Military Communications (MILCOM11), Nov. 2011. Pages 90-104

ABSTRACT

This dissertation investigates several main challenges to implementing hardware-based wireless fading channel emulators with emphasis on incorporating accurate correlation properties. Multiple-input multiple-output (MIMO) fading channels are usually triply-selective with three types of correlation: temporal correlation, inter-tap correlation, and spatial correlation. The proposed emulators implement the triply-selective fading Channel Impulse Response (CIR) by incorporating the three types of correlation into multiple uncorrelated frequency-flat Rayleigh fading waveforms while meeting real-time requirements for high data-rate, large-sized MIMO, and/or long CIR channels. Specifically, mixed parallel-serial computational structures are implemented for Kronecker products of the correlation matrices, which makes the best tradeoff between computational speed and hardware usage. Five practical fading channel examples are implemented for RF or underwater acoustic MIMO applications. The performance of the hardware emulators are verified with an Altera Field-Programmable Gate Array (FPGA) platform and the results match the software simulators in terms of statistical and correlation properties.

The dissertation also contributes to the development of a 2-by-2 MIMO transceiver testbench that is used to measure real-world fading channels. Intensive channel measurements are performed for indoor fixed mobile-to-mobile channels and the estimated CIRs demonstrate the triply-selective correlation properties.

ACKNOWLEDGMENTS

I would like to express my gratitude to all the people who have helped and supported me in my Ph.D. study.

First and foremost, I thank my advisor, Dr. Yahong Rosa Zheng, who advised my M.S. thesis and then provided me with the opportunity of Ph.D. study. With her enthusiasm and inspiration, she thoughtfully guided me in research attitude, specific knowledge, and technical writing. Throughout my five years of study, she provided me encouragement, sound advice, good teaching, lots of good ideas, and financial support. I would have not been where I am today without her help.

Next, I would also like to express my deep gratitude to Dr. Chengshan Xiao, for his guidance and support in several joint research projects. I also acknowledge the support of the Office of Naval Research and the National Science Foundation for sponsoring these research projects.

I would like to thank the members of my advisory committee, Drs. Jagannathan Sarangapani, Steve Grant, Randy H. Moss, and Maggie Chen, for their guidance in my Ph.D. studies and suggestions in my dissertation.

I wish to thank all my colleagues at Missouri S&T and friends in Rolla for their kind assistance in my research, study, and rural life during the past five years.

Last but not least, I wish to express my special thanks to my family for their love, encouragement, and support. Particularly, I would like to thank my parents, who not only have raised and supported me through the years, but also are taking care of my baby now. I also thank my parents-in-law for their understanding and support in my most difficult time. I would also like to gratefully thank my wife, Jing Lin, for her precious love, thoughtful understanding, and heartfelt support.

TABLE OF CONTENTS

	Page
PUBLICATION DISSERTATION OPTION	iii
ABSTRACT	iv
ACKNOWLEDGMENTS	v
LIST OF ILLUSTRATIONS	x
LIST OF TABLES	xiii
 SECTION	
1 INTRODUCTION	1
1.1 BACKGROUND	1
1.2 PROBLEM STATEMENT AND DESIGN APPROACH	5
1.3 SUMMARY OF CONTRIBUTIONS	8
1.4 REFERENCES	10
 PAPER	
I. HARDWARE EMULATION OF WIDEBAND CORRELATED MULTIPLE-INPUT MULTIPLE-OUTPUT FADING CHANNELS	11
Abstract	11
1 INTRODUCTION	11
2 THE MATHEMATIC MODEL	14
3 HARDWARE IMPLEMENTATION METHOD	15
3.1 The FRFG	17
3.2 Ping-Pong Buffers	18
3.3 Correlation Multiplier Module	20
3.4 Interpolator Module	22
4 IMPLEMENTATION EXAMPLES	24
4.1 Implementation Example I - Underwater Acoustic Channel	24

4.2 Implementation Example II - WiMAX Channel	26
5 PERFORMANCE EVALUATION	28
5.1 Performance Comparison of Serial and Mixed P-S Methods	28
5.2 Parameter Specifications and Hardware Usage	29
5.3 Interfacing with Digital Up-Convertor and Down-Convertor	31
6 CONCLUSIONS	32
7 REFERENCES	33
II. A NOVEL EMULATOR FOR DISCRETE-TIME MIMO	
TRIPLY-SELECTIVE FADING CHANNELS	35
Abstract	35
1 INTRODUCTION	36
2 DISCRETE-TIME TRIPLY SELECTIVE FADING MODEL	40
3 HARDWARE IMPLEMENTATION METHOD	43
3.1 Random Number Generator and Flat Rayleigh Fading Generator	45
3.2 $C_{ISI}^{\frac{1}{2}}$ Generator Module	47
3.3 Correlation Multiplier Module	49
3.4 Interpolator Module	50
4 EXAMPLES AND PERFORMANCE EVALUATION	50
4.1 $C_{ISI}^{\frac{1}{2}}$ Generator Performance Evaluation	51
4.2 KP Module Memory Usage Evaluation	52
4.3 Frequency Selective Fading Channel Example	54
4.4 Triply Selective Fading Channel Example	55
4.5 Evaluation of Flat Rayleigh Fading Generators	57
4.6 Parameter Specifications and Hardware Usage	59
5 CONCLUSIONS	62
6 REFERENCES	63

III. HARDWARE IMPLEMENTATION OF TRIPLY SELECTIVE RAYLEIGH FADING CHANNEL SIMULATORS	66
Abstract	66
1 INTRODUCTION	66
2 DISCRETE-TIME MIMO TRIPLY SELECTIVE RAYLEIGH FADING MODEL	68
3 HARDWARE IMPLEMENTATION METHOD	69
4 EXAMPLES AND PERFORMANCE EVALUATION	72
5 CONCLUSIONS	77
6 REFERENCES	77
IV. A LOW-COMPLEXITY HARDWARE IMPLEMENTATION OF DISCRETE- TIME FREQUENCY-SELECTIVE RAYLEIGH FADING CHANNELS	78
Abstract	78
1 INTRODUCTION	78
2 DISCRETE-TIME FREQUENCY-SELECTIVE FADING CHANNEL MODELS	79
3 FPGA IMPLEMENTATION	83
4 IMPLEMENTATION EXAMPLE AND PERFORMANCE EVALUA- TION	85
5 CONCLUSIONS	88
6 REFERENCES	88
V. VALIDATION OF THE TRIPLY SELECTIVE FADING CHANNEL MODEL THROUGH A MIMO TEST BED AND EXPERIMENTAL RESULTS	90
Abstract	90
1 INTRODUCTION	90
2 DISCRETE-TIME TRIPLY SELECTIVE FADING MODEL	91
3 TESTBED AND EXPERIMENT	93
4 PROCEDURE, RESULTS AND ANALYSIS	97
4.1 Channel Estimation	97

4.2 Estimation of the Channel Coefficient Covariance Matrix	98
4.3 Decomposition of the Kronecker Product	99
4.4 Estimation of Intertap Covariance Matrix and Spatial Correlation Matrix	100
5 CONCLUSIONS	102
6 REFERENCES	104
SECTION	
2 CONCLUSIONS	105
3 PUBLICATIONS	106
VITA	107

LIST OF ILLUSTRATIONS

Figure	Page
PAPER I	
1 Block diagram of proposed correlated MIMO fading channels emulator. . .	16
2 Implementation blocks of the FRFG module.	18
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.	20
4 Hardware implementation of CM module using the <i>mixed P-S</i> 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.	23
5 Implementation of the interpolator module.	23
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 carrier wavelength is $\lambda=10$ cm.	25
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.	26
8 Performance of the 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.	28
9 Performance comparison for generating one correlated fading complex response using the <i>serial</i> method and the proposed <i>mixed P-S</i> method.	30
PAPER II	
1 Block diagram of discrete-time MIMO triply selective fading emulator. . .	43
2 Implementation blocks of the RNG and FRFG modules.	46
3 Implementation blocks of the $\mathbf{C}_{ISI}^{\frac{1}{2}}$ generator module.	48

4	Implementation of the correlation multiplier module.	51
5	Implementation of the interpolator module.	52
6	The normalized PDPs for the typical urban channel model presented in 3GPP standard [26].	53
7	The squared error and MSE comparison of coefficients of $\mathbf{C}_{ISI}^{\frac{1}{2}}$ between hardware fixed-point and Matlab floating-point. Note that the scales of y axes are different in sub-figures.	53
8	The comparison of memory usage between the pre-compute and store method and the proposed KP method.	55
9	Performance of Doubly selective fading channel emulator. Auto-correlation of $h_{1,1}(k, 0)$, cross-correlation between $h_{1,1}(k, 0)$ and $h_{1,1}(k, 1)$ and between $h_{1,1}(k, 0)$ and $h_{1,1}(k, 2)$. The results were based on hardware outputs of 50 trials with 2.8×10^4 samples per sub-channel per trial.	56
10	Performance of the triply selective channel emulator. Auto-correlation of $h_{1,1}(k, 1)$, cross-correlation between $h_{1,1}(k, 0)$ and $h_{1,1}(k, 1)$, and between $h_{1,1}(k, 0)$ and $h_{2,1}(k, 1)$. The numbers of trials and samples are the same as those in Fig. 9.	58
11	Performance of triply selective channel emulator. Cross-correlation between $h_{1,1}(k, -1)$ and $h_{1,1}(k, 1)$, and between $h_{1,1}(k, -1)$ and $h_{1,1}(k, 2)$. Note the change of scale in y-axis.	58
12	PDF of Z_{c_i} and Z_{s_i} . The numbers of trials and samples are the same as the previous figure.	61
13	PDF of $ Z_i $, where $Z_i = Z_{c_i} + jZ_{s_i}$. The numbers of trials and samples are the same as the previous figure.	62
14	LCR of $ Z_i $. The numbers of trials and samples are the same to the previous figure.	62

PAPER III

1	Hardware implementation block diagram of the triply selective fading simulator.	72
2	The datapath of the $\mathbf{C}_{ISI}^{\frac{1}{2}}$ generator module.	73
3	The datapath of the CM module.	74
4	The memory usage comparison of the pre-compute and store method and the proposed KP method.	75

5	The auto-correlation of $h_{1,1}(1, k)$, the 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 (2). The results are based on hardware outputs of 50 trials with 2.8×10^4 samples in each channel per trial.	76
---	---	----

PAPER IV

1	(a) A typical urban channel PDP with multiple WSSUS rays. (b) Average power/tap of T_s -spaced discrete-time channel response.	82
2	Bandpass filter of the i -th ray sampled at T_{sym} , where the delay τ_i is a fraction of T_{sym}	82
3	Block diagram of FPGA implementation of the frequency-selective Rayleigh fading simulator.	84
4	FPGA implementation of the flat fading generator module.	85
5	FPGA implementation of frequency-selective fading generator module. . .	86
6	Autocorrelation and cross-correlation of the i -th flat fading ray sampled at T_{sym} interval. The normalized Doppler frequency was $f_d T_{sym} = 0.0008$ and $f_d = 125$ Hz.	87
7	Cross-correlation between $H_c(l, k)$ and $H_c(l+1, k)$ of the frequency-selective channel simulator.	87

PAPER V

1	Transmitter setup architecture.	94
2	Receiver setup architecture.	95
3	Floorplan used for the experiment.	96
4	Magnitudes of channel impulse responses for four subchannels.	98
5	Magnitude of estimated channel coefficient covariance matrix.	99
6	Magnitudes of intertap covariance matrices for each subchannel.	102
7	Magnitude of averaged intertap covariance matrix, Ψ_{Tap}	103
8	Kronecker product of estimated $\Psi_{T_{rx}}$ and $\Psi_{T_{ap}}$	103

LIST OF TABLES

Table	Page
PAPER I	
1 Parameter ranges of the proposed emulator with $F_{clk}=50$ MHz.	30
2 Resource usage of two examples on Stratix III EP3SL150F1152C2N FPGA with $F_{clk}=50$ MHz.	31
3 Resource usage comparisons of related fading channel emulators.	32
PAPER II	
1 Parameter ranges of the proposed emulator with $F_{clk}=50$ MHz and $T_s = 3.69 \mu s$	59
2 Resource usage of the MIMO triply selective fading emulator on Stratix III EP3SL150F1152C2N FPGA with $F_{clk}=50$ Mhz.	60
3 Performance comparisons of related Rayleigh fading channel emulators. . .	61
PAPER III	
1 Hardware usage of the simulator on a Stratix III EP3SL150F1152C2N FPGA chip.	76
PAPER V	
1 Comparison of correlation matrices using CMD.	101

1 INTRODUCTION

1.1 BACKGROUND

Wireless fading channel modeling, along with its software simulation and hardware emulation, is an important topic in wireless communications, because it can provide the basis for verification of new algorithm design, testing of transceiver performance, and analysis of channel capacity. Comparing to field tests, wireless fading channel simulation and emulation are more cost-effective, time-efficient, and can provide repeatable and reproducible results. Therefore, it has been widely adopted in academic and industry.

Two types of channel modeling approaches are usually employed: site-specific wave propagation approach and statistical channel modeling approach. The site-specific wave propagation approach uses Maxwell equations to simulate wave propagation through the communication media and it requires detailed physical environment, geometry and dielectric properties [1]. This approach can provide channel models for specific sites but requires high computational power for simulation. In contrast, the statistical channel modeling approach ignores the physical details and only generates the fading channel impulse responses (CIR) with accurate statistical properties [2,3,4]. Well-designed statistical models can model real-world fading channels matching realistic statistical properties such as probability distribution function (PDF), power delay profiles, and auto/cross-correlation. Its computational complexity is much smaller than the wave propagation approach, thus gaining wide spread use in the last two decades.

This dissertation takes the statistical modeling approach to multiple-input multiple-output (MIMO) fading channel modeling and focuses on hardware emulation. The input-output relationship of a MIMO fading channel in the discrete-time

domain can be modeled by using a finite impulse response (FIR) filter method. The math model can be described as follows:

$$\mathbf{y}(k) = \sum_{l=-L_1}^{L_2} \mathbf{H}_l(k) \cdot \mathbf{x}(k-l) + \mathbf{v}(k) \quad (1.1)$$

Where $\mathbf{x}(k) = [x_1(k), x_2(k), \dots, x_N(k)]^t$, $\mathbf{y}(k) = [y_1(k), y_2(k), \dots, y_M(k)]^t$, and $\mathbf{v}(k) = [v_1(k), v_2(k), \dots, v_M(k)]^t$ are the input vector, output vector, and noise vector at time instant k , respectively. The parameters M , N , L_1 , and L_2 are the numbers of receiver (Rx) elements, transmitter (Tx) elements, low and high indices of channel taps per sub-channels, respectively. The matrix $\mathbf{H}_l(k)$ is the channel matrix at time instant k and delay tap l , defined by

$$\mathbf{H}_l(k) = \begin{pmatrix} h_{1,1}(k, l) & \cdots & h_{1,N}(k, l) \\ \vdots & \ddots & \vdots \\ h_{M,1}(k, l) & \cdots & h_{M,N}(k, l) \end{pmatrix} \quad (1.2)$$

Where $h_{m,n}(k, l)$ is the instantaneous channel coefficient at time instant k , and delay tap l for the sub-channel of the m -th Rx element and n -th Tx element. We assume the symbol duration is denoted as T_s . For the convenience of hardware emulation, the matrix $\mathbf{H}_l(k)$ is reshaped to a MIMO channel coefficient vector $\mathbf{h}_{\text{vec}}(k)$ as follows:

$$\begin{aligned} \mathbf{h}_{\text{vec}}(k) &= [\mathbf{h}_{1,1}(k), \dots, \mathbf{h}_{1,N}(k) \mid \dots \mid \mathbf{h}_{M,1}(k), \dots, \mathbf{h}_{M,N}(k)]^t \\ \mathbf{h}_{m,n}(k) &= [h_{m,n}(k, -L_1), \dots, h_{m,n}(k, L_2)]^t \end{aligned} \quad (1.3)$$

A practical MIMO fading channel exhibits three types of correlation. Spatial correlation models space-selectivity; temporal correlation describes the time-selectivity; inter-tap correlation exhibits frequency-selectivity. Therefore, this is referred to as the MIMO triply-selective fading channel [2]. Spatial correlation is usually

measured and predefined according to properties of multiple antennas, and denoted as the Rx correlation matrix Ψ_{Rx} and Tx correlation matrix Ψ_{Tx} . The inter-tap correlation is denoted as the inter-tap correlation matrix, \mathbf{C}_{ISI} . With these correlation matrices, the MIMO channel coefficient vector can be computed as follows:

$$\mathbf{h}_{vec}(k) = \mathbf{C}_h^{\frac{1}{2}}(0) \cdot \Phi(k) = (\Psi_{Rx}^{\frac{1}{2}} \otimes \Psi_{Tx}^{\frac{1}{2}} \otimes \mathbf{C}_{ISI}^{\frac{1}{2}}) \cdot \Phi(k) \quad (1.4)$$

Where the vector $\Phi(k) = [Z_1(k), \dots, Z_{MNL}(k)]^t$ consists of CIRs of multiple frequency-flat fading sub-channels at time instant k . The operator “ \otimes ” and “ $(\cdot)^{\frac{1}{2}}$ ” are the Kronecker product and square root of a matrix, respectively. More details about this equation can be found in [2].

The CIRs of the i -th frequency-flat fading sub-channel, $Z_i(k)$, exhibit temporal correlation, and can be generated by flat Rayleigh fading generators (FRFG) using a sum of sinusoids (SoS) method in [3]. The SoS method is described by the following equations:

$$\begin{aligned} Z_i(k) &= Z_{c_i}(k) + jZ_{s_i}(k), \\ Z_{c_i}(k) &= \sqrt{\frac{2}{M}} \sum_{m=1}^M \cos(2\pi(f_d k T_s \cos \alpha_{m,i} + \phi_{m,i})), \\ Z_{s_i}(k) &= \sqrt{\frac{2}{M}} \sum_{m=1}^M \cos(2\pi(f_d k T_s \sin \alpha_{m,i} + \varphi_{m,i})), \\ \alpha_{m,i} &= \frac{\pi(m - 0.5 + \theta_i)}{2M}, \quad m = 1, 2, \dots, M. \end{aligned} \quad (1.5)$$

Where $Z_i(k)$ is the complex CIR of frequency-flat fading channel at time instant k , f_d is the maximum Doppler frequency, M is the total number of sinusoids, and $j = \sqrt{-1}$. The angle of arrival $\alpha_{m,i}$ is randomized by the variable θ_i . The random variables $\phi_{m,i}$ and $\varphi_{m,i}$ are the random phases of the in-phase and quadrature components,

respectively. The random variables $\phi_{m,i}$, $\varphi_{m,i}$, and θ_i are statistically independent and uniformly distributed on $[-0.5, 0.5]$.

The inter-tap correlation matrix \mathbf{C}_{ISI} is related to channel power delay profiles (PDP), transceiver shaping/matching filters and symbol rates. According to [2], the coefficient of \mathbf{C}_{ISI} , $c(l_1, l_2)$, can be computed by the following equations:

$$\begin{aligned} c(l_1, l_2) &= \sum_{i=1}^K \sigma_i^2 R_{P_T P_R}(l_1 T_s - \tau_i) R_{P_T P_R}^*(l_2 T_s - \tau_i) \\ G(\tau) &= \sum_{i=1}^K \sigma_i^2 \delta(\tau - \tau_i) \end{aligned} \quad (1.6)$$

Where $G(\tau)$ is the power delay profile, K is the number of total resolvable paths and σ_i^2 is the power of the i -th path with delay τ_i , $R_{P_T P_R}(\varepsilon)$ is the convolution function of the Tx shaping filter and Rx matching filter.

Ignoring correlation properties result in inaccurate MIMO channel models, lead to incorrect CIRs, and may cause failure of transceiver design. For example, temporal correlation, caused by Doppler shift, can affect the selection of coding length and receiver performance. Spatial correlation, caused by insufficient spacing between antenna elements, can reduce channel capacity drastically. Inter-tap correlation can affect channel gains at different frequencies, thus influence equalizer design, adaptive modulation selection, and multi-user channelization, etc. Therefore, considering all three types of correlation is the key, as well as difficult, aspect of accurate MIMO channel modeling.

Several software-based fading channel simulators have been developed to generate CIRs of MIMO fading channels employing general-purpose processors and floating-point algorithms. A discrete-time MIMO triply-selective fading channel simulator has been proposed in [2]. It computes the inter-tap correlation matrix according to the power delay profile, and then incorporates the inter-tap and spatial correlation matrices into multiple uncorrelated frequency-flat fading CIRs via Kronecker product.

Another simulator, proposed in [4], synthesizes correlated vector channels (with user-specified correlation function) using the auto-regression modeling method to shape the spectrum of uncorrelated white Gaussian processes. Software-based fading channel simulators are widely used for research algorithm testing in non real-time settings.

For real-time testing and instrumentation, however, hardware-based wireless fading channel emulators are required to generate analog fading waveforms or digital CIRs in real-time. Implementing hardware-based emulators for MIMO fading channels with accurate correlation properties is more difficult than software simulation due to timing and resource constraints. Existing hardware-based channel emulators in commercial products or academic papers often simplify their design by ignoring some of the correlation functions in MIMO channels. For example, existing commercial MIMO emulators, such as NoiseCom MP-2500, Agilent N5115B, Azimuth ACE-MX/440B, etc., only implement the temporal correlation function of fading channels. Many so-called MIMO emulators in the literature [5] and [6] only implement multiple uncorrelated frequency-flat fading sub-channels or consider only spatial or temporal correlation.

1.2 PROBLEM STATEMENT AND DESIGN APPROACH

This dissertation investigates several main challenges in hardware-based triply-selective MIMO fading channel emulators with accurate correlation properties. These challenges include incorporating correlation matrices into channel models, generating CIR outputs in real-time, and making tradeoff between processing speed and hardware resource usage.

The first challenge is to implement some matrix computation modules in hardware. To incorporate the three types of correlation in (1.4) into MIMO fading channels in hardware emulators, extensive matrix computations such as Kronecker product, matrix square root, and matrix multiplication are required. Effectively implementing

them in hardware is challenging due to 2-D properties of matrix computations. In particular, computing the correlation coefficients in (1.6) and matrix square roots are the most complex tasks for hardware implementations.

The second challenge is to meet output real-time requirements for different fading channels. The Kronecker product, matrix square root, and matrix multiplication require a large amount of multiplications and additions, especially when the sizes of matrices are large. In practical RF and underwater acoustic wireless channels, MIMO sizes, $M \times N$, range from 2×2 to 8×12 . The CIR length, L , can vary from ten to several hundreds. The resulting matrix, $\mathbf{C}_h^{\frac{1}{2}}(0)$, can have a size as large as 1000×1000 . The implementation challenges lay in how to speed up the computation via parallel processing, pre-computing, and how to algorithmically reduce computational load.

The third challenge is to achieve good balance between processing speed and hardware usage. For example, using pre-computed correlation matrix, $\mathbf{C}_h^{\frac{1}{2}}(0)$, may eliminate the huge computational load completely and achieve high processing speed, but this approach requires huge memory resource to store the pre-computed data and place stringent requirement on hardware memories. Another example is the selection between parallel and serial structures. Using fully parallel processing for all MIMO sub-channels can increase processing speed but again requires large hardware resources that are linearly proportional to the size of the channel. Using all serial processing saves hardware resources but requires a large amount of processing time. Analyzing real-time requirement of different MIMO channels and designing balanced implementation are the challenges as well as the contributions of this dissertation.

The approach to dealing with the hardware challenges utilizes four techniques:

1. Pre-compute three small-sized matrices, $\mathbf{\Psi}_{Rx}^{\frac{1}{2}}$, $\mathbf{\Psi}_{Tx}^{\frac{1}{2}}$, and $\mathbf{C}_{ISI}^{\frac{1}{2}}$, rather than the large-sized $\mathbf{C}_h^{\frac{1}{2}}(0)$, and then implement Kronecker product to compute $\mathbf{C}_h^{\frac{1}{2}}(0)$

from $\Psi_{Rx}^{\frac{1}{2}}$, $\Psi_{Tx}^{\frac{1}{2}}$, and $\mathbf{C}_{ISI}^{\frac{1}{2}}$. This technique achieves good tradeoff between processing speed and memory usage.

2. Develop a mixed parallel-serial (mixed P-S) computational structure, which employs different numbers of computational paths to compute the Kronecker product and vector multiplication in parallel. The computational speed of his mixed P-S structure is flexible and adjustable to meet various real-time requirements of different triply-selective fading channels.
3. Design a storage module, which takes advantage of the symmetrical property of $\mathbf{C}_{ISI}^{\frac{1}{2}}$ to save memory usage. This module only stores a half of coefficients of $\mathbf{C}_{ISI}^{\frac{1}{2}}$, and can rebuild another half according to the stored half. When long CIR channels with the large-sized $\mathbf{C}_{ISI}^{\frac{1}{2}}$ are emulated, this module can save a large amount of memory usage.
4. Efficiently reuse the FRFG of (1.5) to generate multiple independent signals required in $\Phi(k)$ of (1.4). The proposed method employs one FRFG to generate up to hundreds of frequency-flat fading sub-channels in parallel. It makes use of Ping-Pong buffers to buffer the outputs of the FRFG, and synchronize speed of these outputs with mixed P-S computational paths. A large amount of hardware resources are saved by reducing multiple FRFGs to only one.

The proposed MIMO triply-selective emulators are implemented and tested on an FPGA platform. The Altera Stratix III EP3SL150F1152C2N FPGA/DSP development kits is employed as the hardware platform. This development kit contains Stratix III EP3SL150F1152C2N FPGA chip, 72 MB SRAM, 16 MB flash memory, display LEDs, push-buttons, DIP switches, and data conversion high speed mezzanine daughter cards. The Stratix III FPGA chip features 142000 logic elements (LEs), 5499 Kbits of memory, 384 multiplier blocks, eight phase locked loops (PLLs), 16 global clock networks, and 736 user I/Os. Altera Quartus II v9.1, DSP Builder, and Matlab

Simulink are used for hardware development. Several fading channel examples are implemented on the development kit, and their output results are analyzed and verified in Matlab.

In addition, this dissertation also developed a hardware wireless 2-by-2 MIMO channel testbench. This testbench can be used to study CIRs and correlation properties of real-world MIMO channels. Experimental results have verified the spatial and temporal correlation properties with in-door fixed mobile-to-mobile MIMO channels.

1.3 SUMMARY OF CONTRIBUTIONS

Research of this dissertation addresses the technical challenges in hardware implementation of triply-selective MIMO fading channel emulators and the wireless MIMO channel testbench. My work results in one journal publications, one journal submission, and four conference publications. The complete publication list can be found in Section 3. The technical contributions of the dissertation are.

1. A new hardware implementation method for on-chip inter-tap correlation $\mathbf{C}_{ISI}^{\frac{1}{2}}$ generator is proposed and successfully incorporated to triply-selective fading channel emulators. So far, none of the other existing emulators implement all three types of correlation. The algorithm of computing $\mathbf{C}_{ISI}^{\frac{1}{2}}$ is based on (1.6) and matrix square root. The $\mathbf{C}_{ISI}^{\frac{1}{2}}$ generator employs a LUT scheme and serial computational structure to generate the coefficient of $\mathbf{C}_{ISI}^{\frac{1}{2}}$ one by one, In order to compute the matrix square root, the $\mathbf{C}_{ISI}^{\frac{1}{2}}$ generator implements the Jacobi algorithm for singular-value decomposition, the square root calculation for each diagonal element, and the matrix multiplication among three matrices. The $\mathbf{C}_{ISI}^{\frac{1}{2}}$ generator can be run once at the beginning of each simulation trial, and its results can be stored and used for the entire trial.

2. The mixed P-S computational structure is proposed and incorporated fully serial FRFG to the triply-selective fading channel emulators. Comparing to the serial and parallel structures, the mixed P-S structure makes the best tradeoff between computational speed and hardware usage for extreme time-consuming modules such as the Kronecker product and vector multiplication. It employs parallel computational paths to generate multiple results of Kronecker product and vector multiplication in one clock period. The number of computational paths can be adjusted to meet real-time requirements of different fading channels. Proved by our testing, The mixed P-S structure can handle MIMO channel with (MNL) up to 1600. According to the equation (1.5), the serial FRFG takes Doppler frequency and symbol period as inputs, and generates multiple frequency-flat fading sub-channels in parallel. The linear feedback shift register (LFSR) random number generators (RNG) and an accurate LUT scheme are employed to generate results of $\cos / \sin(\cdot)$ in the equation (1.5) at the precision level of 6.1×10^{-5} . Sub-channels generated by this FRFG are proven to have accurate statistical properties and temporal correlation.

3. The hardware implementation of wireless MIMO channel testbench includes transmitter and receiver design. In the transmitter side, the frame assemble module and digital up convertor (DUC) are implemented on the FPGA development kit. In the receiver side, the bandpass sampler, digital down convertor (DDC), frame synchronization module, carrier phase detection and compensation module, and frame extraction module are implemented. Based on real-world measurement experiments, this testbench can provide experimental results for the CIR estimation and correlation matrices estimation. Experimental results demonstrate that the discrete-time triply selective fading channel can be expressed as separable temporal, inter-tap and spatial correlations. The spatial

and inter-tap correlation matrices can be estimated through the decomposition of channel coefficient covariance matrix.

1.4 REFERENCES

- [1] A. Goldsmith, *Wireless Communications*. Cambridge, England: Cambridge University Press, 2005.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] M. Cui, H. Murata, and K. Araki, "FPGA Implementation of 4×4 MIMO Testbed for Spatial Multiplexing Systems," in *Proc. IEEE Int. Symp. Pers., Indoor, Mobile Radio Commun.*, Barcelona, Spain, Sep. 2004, pp. 3045-3048.
- [6] 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.
- [7] P. L'Ecuyer, "Tables of Maximally Equidistributed Combined LFSR Generators," *Math. Comput.*, vol. 68, no. 225, pp. 261-269, Jan. 1999.
- [8] Y. R. Zheng, "A Non-isotropic Model for Mobile-to-Mobile Fading Channel Simulations," in *Proc. IEEE MILCOM*, Washington DC, USA, Oct. 2006, pp. 1-7.

PAPER

I. HARDWARE EMULATION OF WIDEBAND CORRELATED MULTIPLE-INPUT MULTIPLE-OUTPUT FADING CHANNELS

Fei Ren and Yahong Rosa Zheng

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 to 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.

1 INTRODUCTION

Fading channel emulators provide a fast and low-cost method for testing and verifying new algorithm design, transceiver performance, and channel capacity

analysis [1, 2, 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 [1,2,3,7,8,9,10] 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 [11,12,13,14,15], where [11,12,13] 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 $0.8 \mu s$ 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.

2 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

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

where $\mathbf{x}(k) = [x_1(k), x_2(k), \dots, x_N(k)]^t$ is the transmitted signal vector, $\mathbf{y}(k) = [y_1(k), y_2(k), \dots, y_M(k)]^t$ is the received signal vector, the superscript $(\cdot)^t$ is the transpose operator of a matrix or vector, and $\mathbf{v}(k) = [v_1(k), v_2(k), \dots, v_M(k)]^t$ 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

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

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

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

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

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

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 Ψ_{Rx} and Ψ_{Tx} are determined by properties of the transmit and receive elements, respectively, and are usually pre-known 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 $\Phi(k)$ is defined as $\Phi(k) = [Z_1(k), Z_2(k), \dots, Z_{(MNL)}(k)]^t$. Each complex coefficient $Z_i(k) = Z_{c_i}(k) + jZ_{s_i}(k)$ ($i = 1, 2, \dots, (MNL)$) represents one of multiple uncorrelated Rayleigh fading waveforms and can be efficiently simulated by the sum of sinusoids (SoS) method in [17, 18].

3 HARDWARE IMPLEMENTATION METHOD

For the convenience of describing hardware implementations, we define three new matrices $\mathbf{C} = \mathbf{C}_{ISI}^{\frac{1}{2}}$, $\mathbf{D} = \Psi_{Rx}^{\frac{1}{2}} \otimes \Psi_{Tx}^{\frac{1}{2}}$, and $\mathbf{E} = \mathbf{C}_h^{\frac{1}{2}}(0)$. The coefficients of the channel vector $\mathbf{h}_{vec}(k)$ in (3) are rearranged as $\mathbf{h}_{vec}(k) = [H(1, k), H(2, k), \dots, H(MNL, k)]^t$, and $H(w, k) = H_c(w, k) + jH_s(w, k)$, where $w = 1, 2, \dots, (MNL)$.

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 $i = 1, 2, \dots, (MNL)$) 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 $Z_{s_i}(Rk)$.

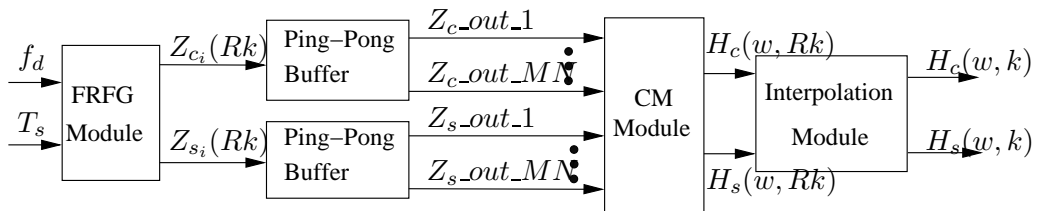


Figure 1. Block diagram of proposed correlated MIMO fading channels emulator.

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. Two sets of Ping-Pong buffers are employed to buffer the real and imaginary parts of uncorrelated complex fading channel responses 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 $\mathbf{C}_{\text{ISI}}^{\frac{1}{2}}$ and employ a symmetric storage submodule to save approximate half of the memory space.

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

The hardware implementation of 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.

3.1 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

$$\begin{aligned}
 Z_i(k) &= Z_{c_i}(k) + jZ_{s_i}(k), \\
 Z_{c_i}(k) &= \sqrt{\frac{2}{P}} \sum_{p=1}^P \cos(2\pi(f_d k T_s \cos \alpha_{p,i} + \phi_{p,i})), \\
 Z_{s_i}(k) &= \sqrt{\frac{2}{P}} \sum_{p=1}^P \cos(2\pi(f_d k T_s \sin \alpha_{p,i} + \varphi_{p,i})), \\
 \alpha_{p,i} &= \frac{\pi(p - 0.5 + \theta_i)}{2P}, \quad p = 1, 2, \dots, P.
 \end{aligned} \tag{5}$$

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 .

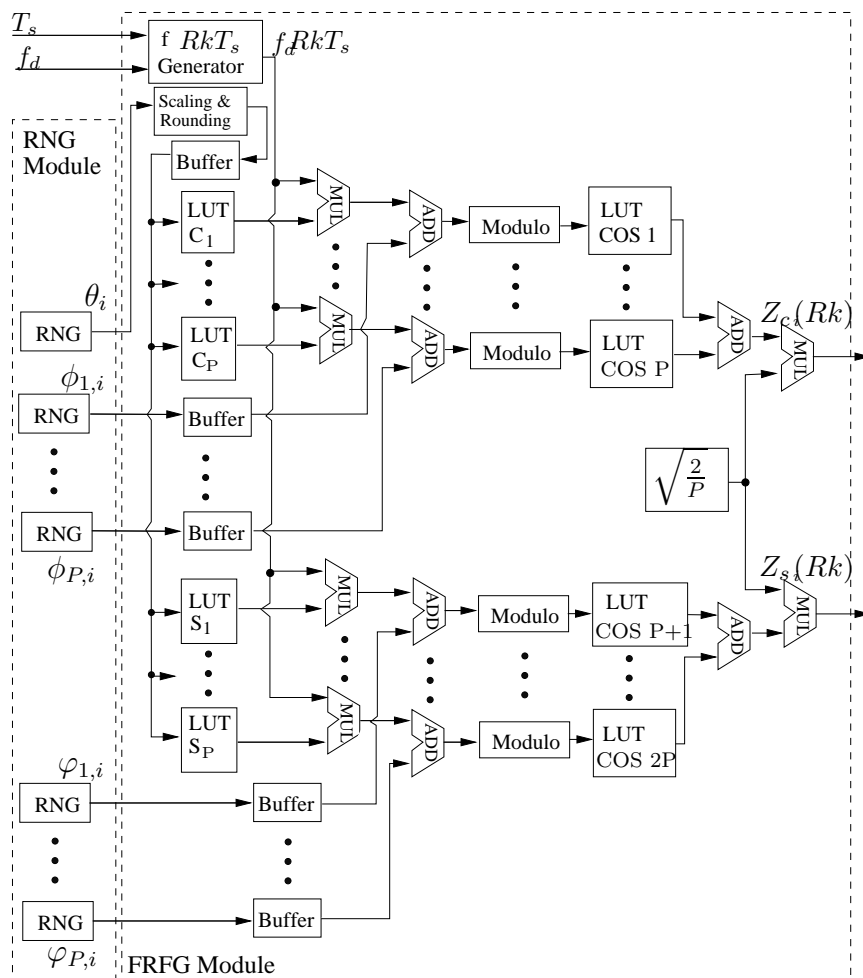


Figure 2. Implementation blocks of the FRFG module.

3.2 Ping-Pong Buffers

The Ping-Pong buffers synchronize the FRFG module with the CM module and make 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 $i = 1, 2, \dots, (MNL)$) are fed to the Ping-Pong buffer in the following format. In a macro period of $(MNLL)$ clock cycles, the serial sequence: $Z_{c_1}(Rk), \dots, Z_{c_{MNL}}(Rk)$, 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 macro 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. The pulse with length of L clock cycles, which is generated by a periodic pulse generator, 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 so as to avoid increasing complexity of the figure.

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 selection 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 $(MNLL)$

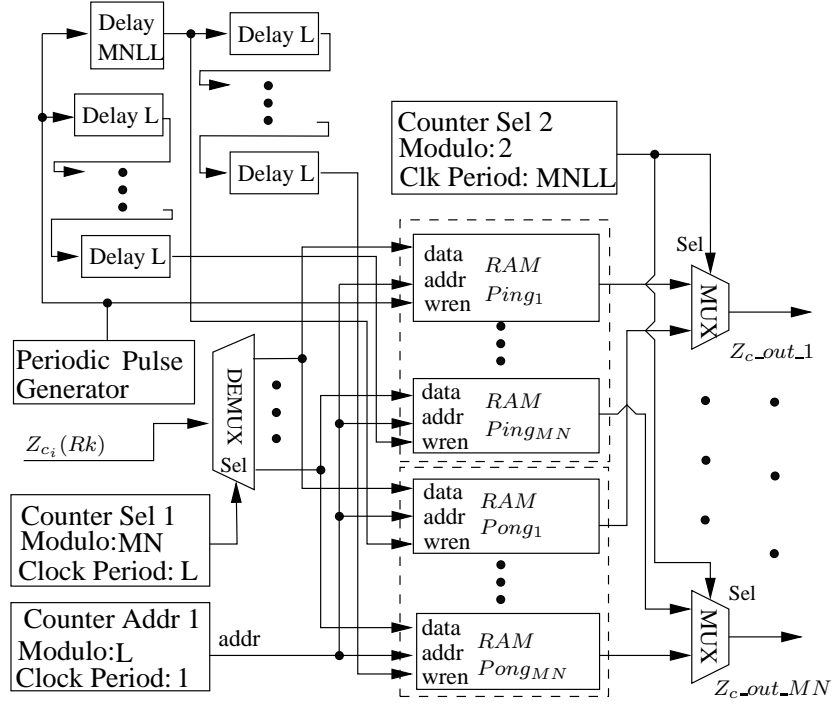


Figure 3. Hardware implementation of the Ping-Pong buffer module. This diagram shows the data buffer for the real part $Z_{c_i}(Rk)$. The imaginary part uses a similar buffer structure.

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 and to be multiplied with the coefficients of matrix \mathbf{E} .

3.3 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 $RAMC$. Actually, the address convertor computes the read addresses by:

$$\text{Read Address} = (\min\{l_r, l_c\} - 1) \left(\max\{l_r, l_c\} - \frac{\min\{l_r, l_c\}}{2} \right) + \max\{l_r, l_c\} \quad (6)$$

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 noting that the Kronecker product can be computed alternatively by $\mathbf{D} = \Psi_{\mathbf{T}\mathbf{x}}^{\frac{1}{2}} \otimes \mathbf{C}_{\mathbf{ISI}}^{\frac{1}{2}}$ first and then $\mathbf{E} = \Psi_{\mathbf{R}\mathbf{x}}^{\frac{1}{2}} \otimes \mathbf{D}$. 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 $\Psi_{\mathbf{R}\mathbf{x}}^{\frac{1}{2}}$, $\Psi_{\mathbf{T}\mathbf{x}}^{\frac{1}{2}}$, 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 $\Psi_{\mathbf{R}\mathbf{x}}^{\frac{1}{2}}$, $\Psi_{\mathbf{T}\mathbf{x}}^{\frac{1}{2}}$, 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.

3.4 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 (basic clock period), 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.

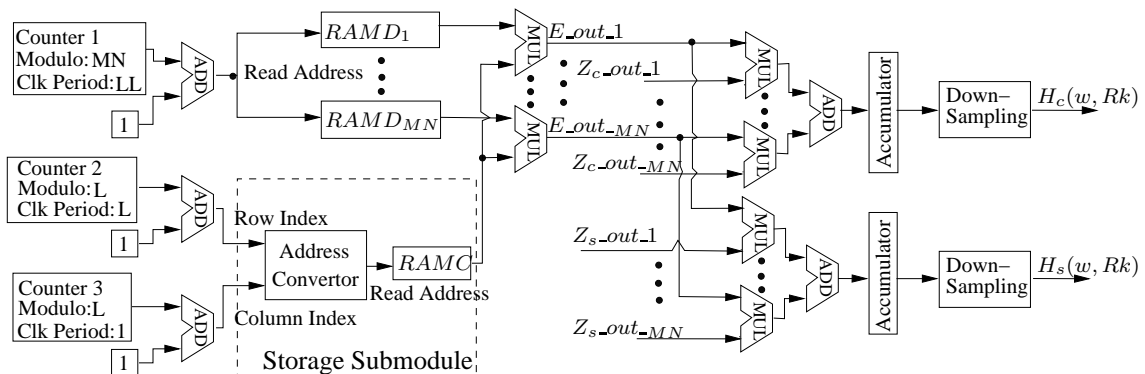


Figure 4. Hardware implementation of CM module using the *mixed P-S* method. In this design, (MN) coefficients of matrix \mathbf{E} are output in parallel per clock cycle, and one row of \mathbf{E} is output in every L clock cycles.

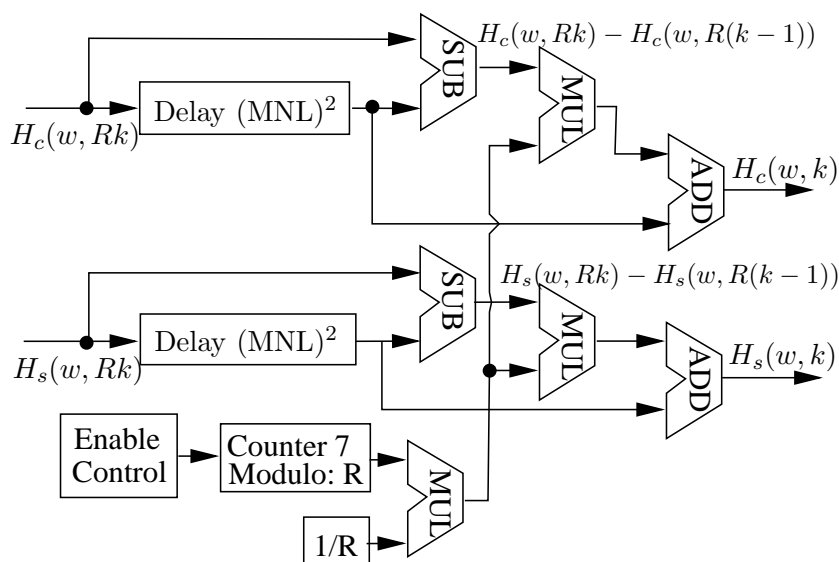


Figure 5. Implementation of the interpolator module.

4 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. The emulator examples can be found at author's website at <http://web.mst.edu/~zhengyr/> as free download.

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 $T_s=250$ μ s. The second example emulated a WiMAX 2-by-2 fading channel with a short symbol duration $T_s=0.8$ μ s 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.

4.1 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 raised-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].

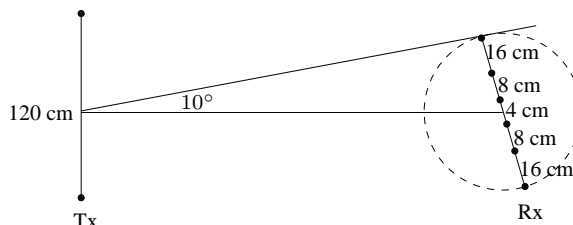


Figure 6. The placement of transmit elements and hydrophones of the underwater communication system. This is a 2-by-6 MIMO underwater acoustic communication system where the speed of the acoustic carrier is 1500 m/s and the frequency of the carrier is 15 kHz. The carrier wavelength is $\lambda=10$ cm.

Other implementation parameters were selected as $M=6$, $N=2$, $L=100$, $T_s=250$ μ s, $f_d=40$ Hz, and $R=10$. The square roots of the correlation coefficient matrices Ψ_{Tx} and Ψ_{Rx} were pre-computed as:

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

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

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.

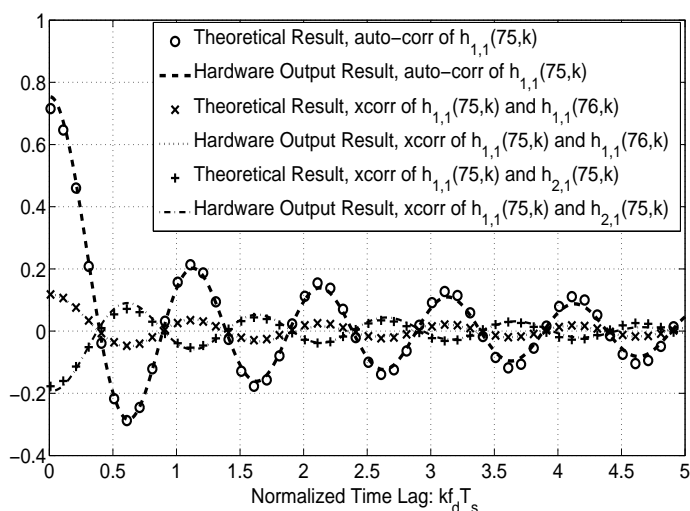


Figure 7. Performance of underwater acoustic fading channel emulator. Auto-correlation of $h_{1,1}(75, k)$, cross-correlation between $h_{1,1}(75, k)$ and $h_{1,1}(76, k)$, and cross-correlation between $h_{1,1}(75, k)$ and $h_{2,1}(75, k)$. The channel index is according to (3). The results are based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

4.2 Implementation Example II - WiMAX Channel

The proposed emulator also implemented the WiMAX 2-by-2 fading channel example. The implementation parameters were selected as $M=N=2$, $T_s=0.8\mu s$, $f_d T_s=0.001$ 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 $\Psi_{Tx}^{\frac{1}{2}}$, $\Psi_{Rx}^{\frac{1}{2}}$, and $C_{ISI}^{\frac{1}{2}}$ were pre-computed and listed as:

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

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

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.

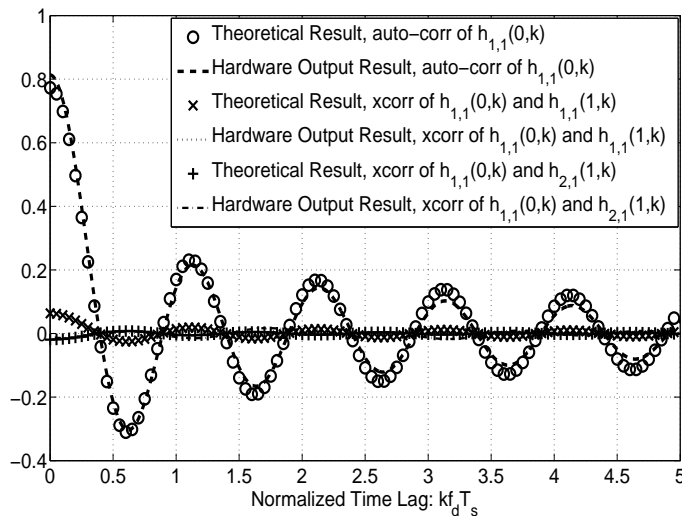


Figure 8. Performance of the WiMAX fading channel emulator. Auto-correlation of $h_{1,1}(0, k)$, cross-correlation between $h_{1,1}(0, k)$ and $h_{1,1}(1, k)$, and cross-correlation between $h_{1,1}(0, k)$ and $h_{2,1}(1, k)$. The channel index is according to (3). The results are based on hardware outputs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

5 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. Moreover, 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.

5.1 Performance Comparison of Serial and Mixed P-S Methods

The proposed emulator with the *mixed P-S* method can compute $\mathbf{C}_h^{\frac{1}{2}}(\mathbf{0})$ 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(a) 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 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. 9(b). 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 $\mathbf{C}_{\text{ISI}}^{\frac{1}{2}}$ could be drastically reduced by making use of the symmetric property of matrix $\mathbf{C}_{\text{ISI}}^{\frac{1}{2}}$. 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.

5.2 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 1 shows the parameter ranges of the emulator with the FPGA chip clock $F_{clk}=50$ MHz.

According to Table 1, 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

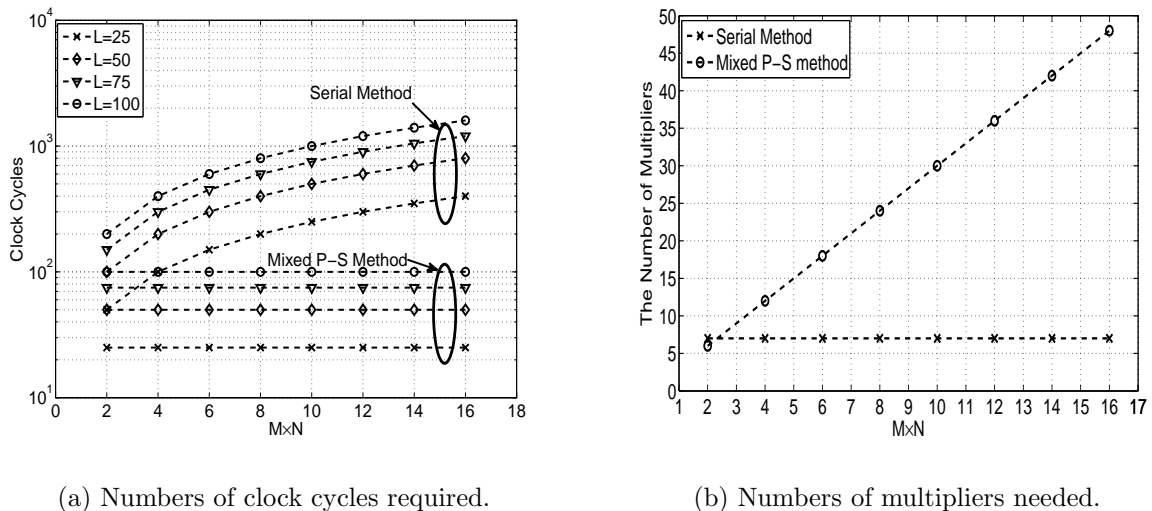


Figure 9. Performance comparison for generating one correlated fading complex response using the *serial* method and the proposed *mixed P-S* method.

Table 1. 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)
$(MN) \leq 16$	$L \leq 100$	$1.9 \times 10^{-6} \sim 1$	$50 \times 10^6 \times \frac{R}{L}$

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 $\frac{F_{clk}R}{L}$ 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 $T_s=250 \mu s$, the real-time requirement can be met by setting $R=10$. For the WiMAX channel with $T_s=0.8 \mu s$, 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 2, 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. Compared to the WiMAX one, the underwa-

Table 2. 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%

ter 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 3. 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.

5.3 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.

Table 3. Resource usage comparisons of related 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.

6 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.

7 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&Schwarz, "R&S ABFS baseband fading simulator," Online: <http://www2.rohde-schwarz.com/product/abfs.html>, May 2009.
- [5] Azimuth Systems, "ACE 400WB MIMO channel emulator," Online: <http://www.azimuthsystems.com>, May 2009.
- [6] Elektrobit, "EB Prosim F8-advanced RF channel emulator," Online: <http://www.elektrobit.com>, Jun. 2009.
- [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] Fei Ren and Y.R. Zheng, "A low-complexity hardware implementation of discrete-time frequency-selective Rayleigh fading channels," in *Proc. IEEE IS-CAS*, Taipei, Taiwan, May 2009, pp. 1759-1762.
- [14] M. Cui, H. Murata, and K. Araki, "FPGA implementation of 4×4 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] F. 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, pp. 1498-1501.
- [17] Y. R. Zheng and C. Xiao, "Improved models for the generation of multiple uncorrelated Rayleigh fading waveforms," *IEEE Trans. Commun.*, vol. 6, no. 6, pp. 256-258, Nov. 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. 2542C2551, Nov. 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. 111C114.
- [21] Altera Corporation, "DSP Builder-advanced blockset with timing-driven simulink synthesis," Online: <http://www.altera.com/products/software/products/dsp/adsp-builder.html>, May 2009.

PAPER

II. A NOVEL EMULATOR FOR DISCRETE-TIME MIMO TRIPLY-SELECTIVE FADING CHANNELS

Fei Ren and Yahong Rosa Zheng

Abstract—Hardware implementation of discrete-time triply selective Rayleigh fading channel emulators is proposed for multiple-input multiple-output (MIMO) communications. The proposed work differs from existing ones in that it incorporates temporal correlation, inter-tap correlation, and spatial correlation matrices into multiple uncorrelated frequency-flat Rayleigh fading waveforms to obtain a triply selective fading channel. The flat fading waveforms with temporal correlation or Doppler spectrum are generated using a Sum-of-Sinusoids (SoS) method. The inter-tap correlation matrix associated with multipath delay spread is computed according to the channel power delay profile and transmit/receive filters. The spatial correlation matrices, including the transmit correlation and receive correlation matrices, are predefined inputs associated with antenna arrangements. The square roots of the three correlation matrices are computed via Singular Value Decomposition (SVD) and then combined in real time via Kronecker product with the flat fading waveforms. Several fading channel examples are implemented on an Altera Stratix III EP3SL150F FPGA DSP development kit with fixed-point arithmetics. A 4×4 MIMO triply-selective channel with 10 correlated delay-taps per sub-channel utilizes one third of the hardware resource of the FPGA chip. The statistical and correlation properties of the emulated fading waveforms match those of the software-based simulators and the theoretical ones. The proposed method achieves good balance between computational complexity and resource utilization.

1 INTRODUCTION

Wireless fading channel modeling, simulation, and emulation are important topics in communications because they provide a fast and low-cost method for testing and verifying new algorithm design, transceiver performance, and channel capacity analysis [1, 2, 3, 4]. To generate correct and realistic fading waveforms, it is important that channel models reproduce accurate properties of actual propagation environments. One significant statistical property for wireless fading channel models is the correlation of fading channel waveforms. For a multiple-input multiple-output (MIMO) system, three types of correlation functions, namely temporal correlation, inter-tap correlation, and spatial correlation, need to be taken into consideration since a practical MIMO fading channel is usually time-selective (described by the temporal correlation), frequency-selective (exhibiting inter-tap correlation), and space-selective (associated with transmitter and/or receiver spatial correlation). This is referred to as the MIMO triply selective fading channel [5]. Incorporating the three types of correlation functions into channel simulators or emulators is the key and yet difficult aspect of accurately simulating MIMO fading channels. The frequency-flat Rayleigh fading channel and the frequency-selective fading channels are two special cases of the MIMO triply-selective fading model [5], when only the temporal correlation, or both the temporal and inter-tap correlation functions are involved, respectively. It is worth noting that the most commonly used Wide-Sense Stationary Uncorrelated Scattering (WSSUS) channel model [1] assumes uncorrelated scatterers in the pass band, which leads to inter-tap correlated fading waveforms in the baseband equivalent channel due to the bandlimited nature of wireless systems [6].

Software-based channel simulators usually employ general-purpose processors and floating-point arithmetic to generate fading channel impulse responses. The flat

Rayleigh fading channels can be simulated by one of the two methods: the spectrum filtering method [7, 8, 9] and the Sum-of-Sinusoids (SoS) method [10, 11, 12, 13]. The spectrum filtering method shapes the spectrum of a white Gaussian waveform using a filter that has a transfer function equal to the square root of the power spectrum density (PSD) of the desired fading process. Doppler spectrum filtering may be implemented by FIR filters [7] or IIR filters [8, 9]. It can simulate fading channels having various PSD shapes and reach accurate statistical properties in every trial. The SoS method sums a finite number of sinusoidal waveforms having amplitudes, frequencies, and phases that are appropriately selected to reproduce the desired statistical properties and Doppler spectra. It is computationally efficient and flexible in parameter reconfiguration (such as the maximum Doppler frequency). Frequency-selective fading channels incorporating inter-tap correlation is more difficult to simulate because the channel is modeled as a time-varying system with a 2-D scattering function. The Doppler frequency is often much smaller than the symbol rate; while the WS-SUS delays are often at fractional spacing of the symbol interval. Many approaches to discrete-time frequency-selective channel simulation are found in the literature, including the delay-weight-and-sum method [6, 14], the correlation matrix multiplication method [15, 5], and the 2-D filtering method [16]. The first one uses fractionally delayed transmit/receive filter taps to weigh and delay multiple uncorrelated flat fading waveforms and sums them together according to the power delay profile (PDP). The correlation matrix multiplication method first computes the inter-tap correlation matrix according to the PDP and then multiplies the square root of the correlation matrix with multiple uncorrelated flat fading waveforms. The 2-D filtering method filters multiple white Gaussian processes by an approximating filter of the delay-Doppler spread function using Gaussian quadrature rules. A MIMO triply selective fading channel model is even more challenging, which needs to consider the transmit

and receive spatial correlation functions in addition to the temporal and inter-tap correlation functions. A discrete-time triply-selective channel, proposed by [5], computes the symbol-spaced inter-tap correlation matrix according to the power delay profile, and incorporates the inter-tap and spatial correlation matrices into multiple uncorrelated SoS flat fading CIRs via Kronecker product. Another MIMO channel simulator, proposed in [17, 8], synthesizes correlated vector channels (with user-specified correlation function) using the Auto-Regression (AR) modeling method to shape the spectrum of uncorrelated white Gaussian processes.

Research on software-based channel simulators provides the basis for the design of hardware-based channel emulators. Several hardware emulators for frequency-flat and doubly-selective fading channels have been reported in the literature. For frequency-flat Rayleigh fading channels, the emulator in [18] is based on the SoS method with a modified random phase variable and its hardware implementation uses reduced rate sinusoids to achieve high speed and low hardware usage. Several other hardware emulators [19, 20] are based on the spectrum filtering method where FIR or IIR filtering with single or multiple interpolation stages is used to improve the accuracy of the narrowband U-shaped Doppler spectrum. For doubly-selective fading channels, the emulator in [21] uses the spectrum filtering method to generate multiple uncorrelated flat fading waveforms, converts them to pass band signals (with upsampling, and then combines them as uncorrelated scatterers according to the power delay profile. Another emulator [22] generates one baseband complex Gaussian process with Doppler filtering and filters it again in the delay-spread domain to generate multipaths. Recently, we propose a hardware emulator [23] based on the software simulation model in [14] which uses the SoS method to generate multiple uncorrelated flat fading waveforms, upsamples them to fractionally spaced baseband signals, and then combines them using weight-delay-and-sum. However, hardware implementation of MIMO triply selective fading channels still presents some challenge

in accurately incorporating correlation functions among the multiple sub-channels. Currently, many so-called MIMO emulators only implement multiple uncorrelated flat fading channels in parallel. For example, the emulator in [24] outputs multiple uncorrelated flat Rayleigh fading channels without considering inter-tap and spatial correlation; the emulator in [25] attempts to incorporate only the spatial correlation matrices into multiple frequency-flat fading waveforms. To the best of our knowledge, no hardware-based channel emulators in the literature or commercial products has properly implemented all three types of correlation functions of MIMO channels.

In this paper, we propose a hardware implementation method for discrete-time MIMO triply selective channel emulators. The proposed method implements the three types of correlation functions of the triply selective channel based on the software simulator in [5]. The emulator consists of five major functional modules: random number generator (RNG), frequency-flat Rayleigh fading generator (FRFG), inter-tap correlation matrix and matrix square root calculation module ($C_{ISI}^{\frac{1}{2}}$ generator), correlation multiplier (CM), and interpolator module. The use of Kronecker product in the CM module saves a large amount of hardware memory storage at the expense of slightly increased computational complexity. This method achieves the best tradeoff between hardware resources and simulation speed. In addition, mixed parallel-serial architecture is used to meet real-time requirements while reducing hardware area. For example, the SoS computation for a single flat Rayleigh fading waveform is performed in parallel, and the generation of multiple flat Rayleigh fading waveforms and correlation combination are performed in series. The proposed emulator is implemented on a Stratix III EP3SL150F FPGA DSP development kit. Several fading channel examples are provided to demonstrate the accuracy and statistical properties of the emulator.

The rest of the paper is organized as follows. Section 2 reviews the software-based model of discrete-time MIMO triply selective fading channels. Section 3 proposes the hardware implementation method for this model. Section 4 presents several FPGA hardware implementation examples and evaluates their performance. Section 5 draws the conclusion.

2 DISCRETE-TIME TRIPLY SELECTIVE FADING MODEL

We choose the discrete-time MIMO triply selective fading model in [5] as the basis for our hardware implementation. Consider a MIMO channel with P transmit and O receive antennas. The input-output relationship of the channel in the discrete-time domain is described as

$$\mathbf{y}(k) = \sum_{q=-Q_1}^{Q_2} \mathbf{H}(k, q) \cdot \mathbf{x}(k - q) + \mathbf{v}(k), \quad (1)$$

where the superscript $(\cdot)^t$ is the transpose operator of a matrix or vector, $\mathbf{x}(k) = [x_1(k), x_2(k), \dots, x_P(k)]^t$ is the transmitted signal vector, $\mathbf{y}(k) = [y_1(k), y_2(k), \dots, y_O(k)]^t$ is the received signal vector, $\mathbf{v}(k) = [v_1(k), v_2(k), \dots, v_O(k)]^t$ is the background white Gaussian noise, k is the time index, and Q_1 and Q_2 are nonnegative integers representing the range of delay taps yielding the total channel length of $Q = Q_1 + Q_2 + 1$. Note we assume the symbol interval is T_s . The MIMO channel coefficient matrix $\mathbf{H}(k, q)$ at time instant k and delay tap q is defined by

$$\mathbf{H}(k, q) = \begin{pmatrix} h_{1,1}(k, q) & \cdots & h_{1,P}(k, q) \\ \vdots & \ddots & \vdots \\ h_{O,1}(k, q) & \cdots & h_{O,P}(k, q) \end{pmatrix} \quad (2)$$

For the convenience of implementation, we reshape the matrix $\mathbf{H}(k, q)$ to an $(OPQ) \times 1$ coefficient vector as

$$\mathbf{h}_{vec}(k) = [\mathbf{h}_{1,1}(k), \dots, \mathbf{h}_{1,P}(k) \mid \dots \mid \mathbf{h}_{O,1}(k), \dots, \mathbf{h}_{O,P}(k)]^t$$

where $\mathbf{h}_{o,p}(k)$ is the coefficient vector of the (o, p) -th sub-channel given by $\mathbf{h}_{o,p}(k) = [h_{o,p}(-Q_1, k), \dots, h_{o,p}(Q_2, k)]$. Based on the software model in [5], the vector $\mathbf{h}_{vec}(k)$ is computed by

$$\mathbf{h}_{vec}(k) = \mathbf{C}_h^{\frac{1}{2}}(0) \cdot \Phi(k) = (\Psi_{Rx}^{\frac{1}{2}} \otimes \Psi_{Tx}^{\frac{1}{2}} \otimes \mathbf{C}_{ISI}^{\frac{1}{2}}) \cdot \Phi(k) \quad (3)$$

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, Ψ_{Rx} and Ψ_{Tx} , are associated with the receiver and transmitter antennas, respectively; \mathbf{C}_{ISI} is the inter-tap covariance matrix which causes intersymbol interference (ISI); and $\Phi(k)$ is an $(OPQ) \times 1$ vector for each time index k .

Define $\Phi(k) = [Z_1(k), Z_2(k), \dots, Z_{(OPQ)}(k)]^t$, where $Z_i(k)$ is one of the uncorrelated flat Rayleigh fading waveforms. The multiple uncorrelated flat Rayleigh fading waveforms $Z_i(k)$ can be efficiently simulated by one of the SoS models [12] and a typical one is

$$\begin{aligned} Z_i(k) &= Z_{c_i}(k) + jZ_{s_i}(k), \\ Z_{c_i}(k) &= \sqrt{\frac{2}{M}} \sum_{m=1}^M \cos(2\pi(f_d k T_s \cos \alpha_{m,i} + \phi_{m,i})), \\ Z_{s_i}(k) &= \sqrt{\frac{2}{M}} \sum_{m=1}^M \cos(2\pi(f_d k T_s \sin \alpha_{m,i} + \varphi_{m,i})), \\ \alpha_{m,i} &= \frac{\pi(m - 0.5 + \theta_i)}{2M}, \quad m = 1, 2, \dots, M. \end{aligned} \quad (4)$$

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

The spatial correlation matrices, $\mathbf{\Psi}_{Rx}$ and $\mathbf{\Psi}_{Tx}$, are usually known and specified by users. The inter-tap covariance matrix, \mathbf{C}_{ISI} , is computed according to the power delay profile (PDP). Define

$$\mathbf{C}_{ISI} = \begin{pmatrix} c(-Q_1, -Q_1) & \cdots & c(-Q_1, Q_2) \\ \vdots & \ddots & \vdots \\ c(Q_2, -Q_1) & \cdots & c(Q_2, Q_2) \end{pmatrix} \quad (5)$$

where $c(q_1, q_2)$ is determined by

$$c(q_1, q_2) = \sum_{n=1}^N \sigma_n^2 R_{P_T P_R}(q_1 T_s - \tau_n) R_{P_T P_R}^*(q_2 T_s - \tau_n) \quad (6)$$

and $R_{P_T P_R}(\xi)$ is the convolution of the transmit and receive filters. Note N is the number of total resolvable paths in the PDP. The superscript $()^*$ is the conjugate operator. Parameters σ_n and τ_n come from the PDPs, $G(\tau)$, which are often specified by standards [26] as

$$G(\tau) = \sum_{n=1}^N \sigma_n^2 \delta(\tau - \tau_n) \quad (7)$$

3 HARDWARE IMPLEMENTATION METHOD

Our hardware implementation of the discrete-time MIMO triply selective fading emulator outputs $\mathbf{h}_{vec}(k)$ for $O \times P$ sub-channels in parallel with Q taps per sub-channel. For the convenience of description, we give elements of $\mathbf{h}_{vec}(k)$ new indices by defining $H(l, k) = h_{o,p}(k, q)$, where $l = Q \cdot [(o-1) \cdot P + (p-1)] + (q + Q_1 + 1)$. Therefore, the channel vector $\mathbf{h}_{vec}(k)$ is converted to $[H(1, k), H(2, k), \dots, H((OPQ), k)]^t$, where $H(l, k)$ is the complex fading coefficient with $H(l, k) = H_c(l, k) + jH_s(l, k)$.

The proposed implementation scheme consists of five major modules: a random number generator (RNG), a flat Rayleigh fading generator (FRFG), a $\mathbf{C}_{ISI}^{\frac{1}{2}}$ generator, a correlation multiplier (CM) module, and an interpolator module, as shown in Fig. 1. Each module implements one or more equations presented in Section 2.

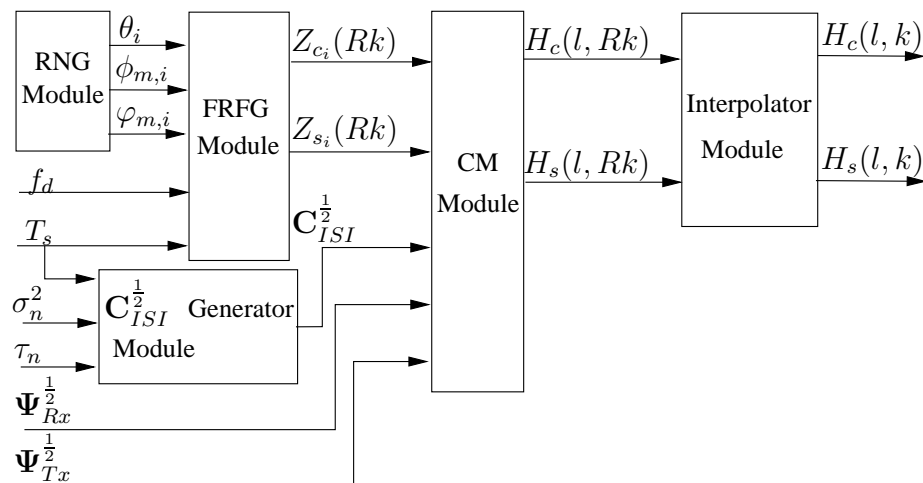


Figure 1. Block diagram of discrete-time MIMO triply selective fading emulator.

The RNG module is a bank of pseudo-random number generators, which generate uniform random variables used in the SoS channel model (3). Note that a set of

the random variables is generated at the beginning of each trial and they are stored and used for all k (time index).

The FRFG module serially generates a large number of uncorrelated flat Rayleigh fading waveforms with proper temporal correlation (or Doppler spectrum) each at a low sampling rate. It takes the random variables from the RNG module as inputs and computes multiple uncorrelated flat Rayleigh fading responses according to (3), but with a decimation factor, R , of the symbol interval. This implementation takes advantage of the fact that the maximal Doppler frequency of typical fading channels is often much smaller than the symbol rate and fading variation within channel coherence time is small. This technique reduces the computational complexity while preserving the accuracy of the channel waveforms.

The $\mathbf{C}_{ISI}^{\frac{1}{2}}$ generator computes the coefficients of the inter-tap correlation matrix using (5) and the square root of \mathbf{C}_{ISI} using the Jacobi algorithm for SVD [27]. Note that this module is also used only once at the beginning of each simulation trial and the square root matrix $\mathbf{C}_{ISI}^{\frac{1}{2}}$ is stored and used, along with the external input, $\mathbf{C}_{TX}^{\frac{1}{2}}$ and $\mathbf{C}_{RX}^{\frac{1}{2}}$, for the entire trial.

The CM module incorporates the three square-root matrices with the multiple uncorrelated flat Rayleigh fading channel responses using the Kronecker product and vector multiplication. Note that the three square root matrices are saved in the on-chip memory of the FPGA development board.

The interpolator module linearly interpolates the triply selective fading channel waveforms into symbol-spaced samples with an interpolation rate R . This module makes sure the emulator meets real-time requirements.

The $\mathbf{C}_{ISI}^{\frac{1}{2}}$ generator and CM module are novel FPGA hardware implementations proposed by this paper. Instead of storing a big square root matrix of size $(OPQ) \times (OPQ)$, this architecture stores the coefficients of three small matrices of sizes $O \times O$, $P \times P$, $Q \times Q$, respectively. The memory savings is accomplished by

slight increase of computational complexity associated with the Kronecker product calculation. The other three modules used in the proposed scheme are similar to the ones in [20, 25, 29] with slight modifications. The following subsections will describe each module in details, with emphasis on the $\mathbf{C}_{ISI}^{\frac{1}{2}}$ generator and CM modules.

3.1 Random Number Generator and Flat Rayleigh Fading Generator

The RNG and FRFG modules work together to generate (OPQ) uncorrelated flat Rayleigh fading channels using (3). The data path of RNG and FRFG modules is shown in Fig. 2. The FRFG module has a parallel-serial mixed structure, which generates $(2M)$ sinusoids in parallel and $Z_1(Rk) \sim Z_{(OPQ)}(Rk)$ in serial. The parallel structure reduces processing time of computing a single $Z_i(Rk)$ to $\frac{1}{M}$ of that required by a serial structure; while a serial structure is a better choice for generating $Z_1(Rk) \sim Z_{(OPQ)}(Rk)$ because (OPQ) is a large and reconfigurable number. The serial structure outputs $Z_1(Rk) \sim Z_{(OPQ)}(Rk)$ sequentially, which matches the requirement of the CM module needing serial inputs.

The RNG module generates all uniform random variables including $\phi_{m,i}$, $\varphi_{m,i}$, and θ_i , where m ranges from 1 to M and i ranges from 1 to (OPQ) . It consists of a set of $(2M + 1)$ RNGs: $(2M)$ of them are for $\phi_{m,i}$ and $\varphi_{m,i}$, and one for θ_i . To provide sufficient accuracy and randomness, we employ the combined linear feedback shift register (LFSR) random number generator (RNG) [29], which has a longer recurrence period, better randomness, and correlation properties than the conventional LFSR RNGs. In our implementation, outputs of the RNGs are scaled and shifted to meet the range requirement before storing in buffers of the FRFG module.

The FRFG module involves a large set of cos and sin functions, whose accuracy affects the performance of the emulator significantly. We propose a simplified but accurate look-up-table (LUT) scheme to compute $\cos \alpha_{m,i}$ and $\sin \alpha_{m,i}$. Since θ_i are uniformly distributed on $[-0.5, 0.5)$, it can be proved that $\alpha_{m,i}$ are uniformly

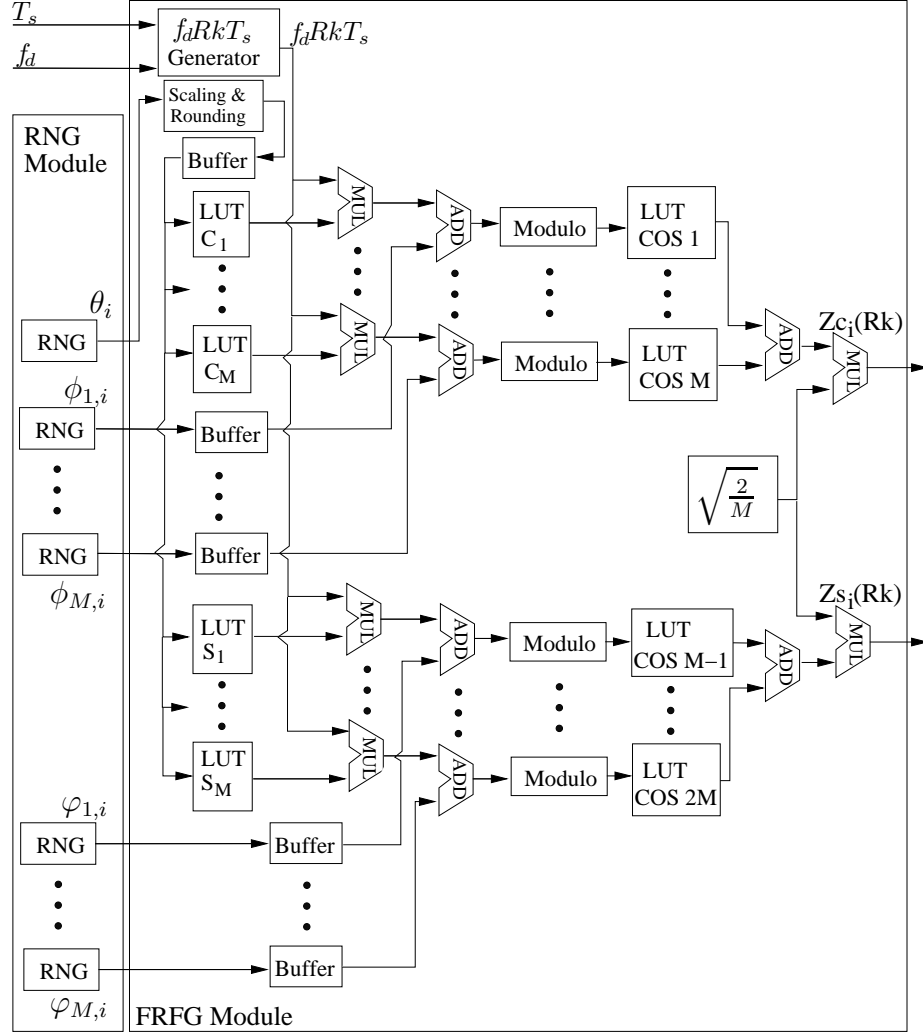


Figure 2. Implementation blocks of the RNG and FRFG modules.

distributed on $[\frac{\pi(m-1)}{2M}, \frac{\pi m}{2M}]$. We build a set of $(2M)$ LUTs named $C_1, C_2, \dots, C_M, S_1, S_2, \dots, S_M$, each of which has D_1 non-overlap entries. The entries in the LUT C_m are $\cos(\frac{\pi(m-1)}{2M} : \frac{\pi}{2MD_1} : \frac{\pi m(D_1-1)}{2MD_1})$; while the entries in the LUT S_m are $\sin(\frac{\pi(m-1)}{2M} : \frac{\pi}{2MD_1} : \frac{\pi m(D_1-1)}{2MD_1})$. The outputs of the RNG for θ_i are rounded and scaled to the range of $[1, D_1]$. Taking these outputs as the read addresses, the set of LUTs output the desired $\cos \alpha_{m,i}$ and $\sin \alpha_{m,i}$ values. This LUT scheme achieves a very high precision which is equivalent to implementing an (MD_1) -entry LUT with range

$\cos(0) \sim \cos(\frac{\pi}{2})$ or $\sin(0) \sim \sin(\frac{\pi}{2})$. The format of the entries in the LUTs is the fixed point Q2.14, which is enough to meet the accuracy requirement.

In the FRFG module, a generator block is employed to output $(f_d Rk T_s)$, where k is generated by an increase counter with a proper updating period, and it is then multiplied by the constants f_d, R , and T_s). The outputs of this block are multiplied by $\cos \alpha_{m,i}$ (or $\sin \alpha_{m,i}$) and added with $\phi_{m,i}$ (or $\varphi_{m,i}$) to obtain $(f_d Rk T_s \cos \alpha_{m,i} + \phi_{m,i})$ (or $(f_d Rk T_s \sin \alpha_{m,i} + \varphi_{m,i})$). A set of modulo operators extract the fractional parts of the inputs and convert them into the read addresses of the $(2M)$ LUTs named $\text{COS}_1, \text{COS}_2, \dots, \text{and } \text{COS}_{2M}$. These LUTs are employed to compute $\cos(2\pi(f_d Rk T_s \cos \alpha_{m,i} + \phi_{m,i}))$ and $\cos(2\pi(f_d Rk T_s \sin \alpha_{m,i} + \varphi_{m,i}))$, respectively. All these LUTs are identical with D_2 entries $\cos(0 : \frac{2\pi}{D_2} : \frac{2\pi(D_2-1)}{D_2})$. The outputs of these LUTs are summed by two accumulators, and multiplied by $\sqrt{2/M}$ to obtain $Z_{c_i}(Rk)$ and $Z_{s_i}(Rk)$. Taking the corresponding parameters of the (OPQ) sub-channels, the FRFG module is re-used to generate (OPQ) uncorrelated flat Rayleigh fading channels. The outputs of the FRFG module are sent to two buffers in the CM module.

3.2 $\mathbf{C}_{ISI}^{\frac{1}{2}}$ Generator Module

The $\mathbf{C}_{ISI}^{\frac{1}{2}}$ generator module takes the PDPs as input and generate $\mathbf{C}_{ISI}^{\frac{1}{2}}$. It consists of two submodules: the \mathbf{C}_{ISI} generator module which computes the coefficients of \mathbf{C}_{ISI} according to (4) and (5), and the matrix square root (MSR) module which finds the matrix square root of \mathbf{C}_{ISI} . The datapath of the $\mathbf{C}_{ISI}^{\frac{1}{2}}$ generator is shown in Fig. 3. The q_1 and q_2 counters range from $-Q_1$ to Q_2 (assuming $Q_1 \leq Q_2$). The q_1 counter increases by one in every (NQ) basic clock periods (BCP); while the q_2 counter increases by one in every N BCPs. Two buffers store values of σ_n^2 and τ_n and sequentially output them while n increases from 0 to N . The outputs of the two buffers are repeated sequences with a period of N BCPs. Computations of $R_{P_T P_R}(\xi)$

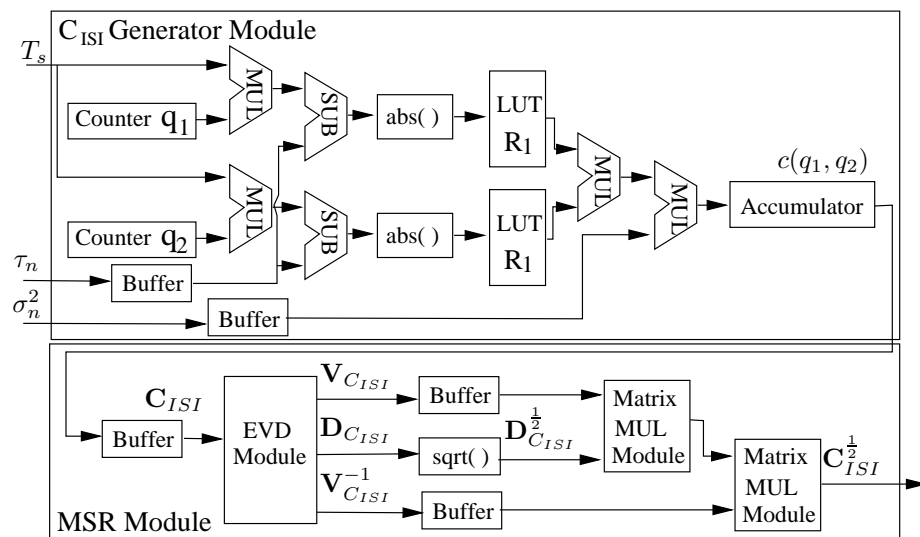


Figure 3. Implementation blocks of the $\mathbf{C}_{ISI}^{\frac{1}{2}}$ generator module.

and $R_{P_T P_R}^*(\xi)$ are implemented using a LUT scheme. Since $R_{P_T P_R}(\xi)$ is a real and even function, the size of the LUT can be reduced by only storing the values corresponding to $\xi \geq 0$. In our implementation, the LUTs R_1 and R_2 have the same D_3 entries: the results of $R_{P_T P_R}(\xi)$ where $\xi = (0 : \frac{Q_1 T_s}{D_3 - 1} : Q_1 T_s + \text{MAX}(\tau_n))$. The read addresses of R_1 and R_2 are computed from $|p_1 T_s - \tau_n|$ and $|q_2 T_s - \tau_n|$, and the outputs are multiplied together and then by the corresponding σ_n^2 . The accumulator following the two multipliers sums the N inputs to obtain one coefficient $c(q_1, q_2)$ in every N BCPs. The \mathbf{C}_{ISI} generator module sequentially outputs the coefficients in a row-wise order.

The MSR module employs the EigenValue Decomposition (EVD) method [27] to find the matrix square root of \mathbf{C}_{ISI} . According to (4) and (5), \mathbf{C}_{ISI} is always a symmetric positive definite matrix, whose eigenvalues are equal to its singular values. Decomposing \mathbf{C}_{ISI} into two matrices: $\mathbf{V}_{C_{ISI}}$ and $\mathbf{D}_{C_{ISI}}$, we have $\mathbf{C}_{ISI} = \mathbf{V}_{C_{ISI}} \mathbf{D}_{C_{ISI}} \mathbf{V}_{C_{ISI}}^{-1}$, where $\mathbf{D}_{C_{ISI}}$ is diagonal with its diagonal elements being the eigenvalues of \mathbf{C}_{ISI} . The square root matrix is then computed as $\mathbf{C}_{ISI}^{\frac{1}{2}} = \mathbf{V}_{C_{ISI}} \mathbf{D}_{C_{ISI}}^{\frac{1}{2}} \mathbf{V}_{C_{ISI}}^{-1}$. The MSR module is implemented by an EVD submodule and two matrix multipliers.

The coefficients of \mathbf{C}_{ISI} are sequentially fed via a buffer to the EVD module and the eigenvalues are computed by the Jacobi rotation algorithm [27] with implementation details found in [30,31]. The outputs of the EVD module, $\mathbf{V}_{C_{ISI}}$ and $\mathbf{V}_{C_{ISI}}^{-1}$, are stored in separate buffers while $\mathbf{D}_{C_{ISI}}$ sequentially pass through a square root calculator to yield $\mathbf{D}_{C_{ISI}}^{\frac{1}{2}}$. The coefficients of $\mathbf{V}_{C_{ISI}} \mathbf{D}_{C_{ISI}}^{\frac{1}{2}} \mathbf{V}_{C_{ISI}}^{-1}$, or $\mathbf{C}_{ISI}^{\frac{1}{2}}$, are computed using two matrix multipliers and are output sequentially in a row-wise order.

3.3 Correlation Multiplier Module

The CM module incorporates inter-tap and spatial correlation matrices into the multiple uncorrelated flat Rayleigh fading waveforms generated by the FRFG. It consists of two submodules: the Kronecker product (KP) module, which computes $\mathbf{C}_h^{\frac{1}{2}}(0) = \mathbf{\Psi}_{Rx}^{\frac{1}{2}} \otimes \mathbf{\Psi}_{Tx}^{\frac{1}{2}} \otimes \mathbf{C}_{ISI}^{\frac{1}{2}}$, and the vector multiplier (VM) module, which implements $\mathbf{C}_h^{\frac{1}{2}}(0) \cdot \mathbf{\Phi}(k)$. Although the coefficients of $\mathbf{C}_h^{\frac{1}{2}}(0)$ are fixed values which can be pre-computed by software and stored in hardware memory, the pre-compute and store method consumes a large amount of hardware memory for $\mathbf{C}_h^{\frac{1}{2}}(0)$ storage, especially, when (OPQ) is large. Our proposed method employs the KP module to compute $\mathbf{C}_h^{\frac{1}{2}}(0)$ in real-time and its results are input to the next module without storing. The datapath of the CM module is shown in Fig. 4.

The KP module consists of three random access memories (RAMs), six counters and a few multipliers and adders. The RAM A, RAM B, and RAM C store the coefficients of the three small matrices, $\mathbf{\Psi}_{Rx}^{\frac{1}{2}}(O \times O)$, $\mathbf{\Psi}_{Tx}^{\frac{1}{2}}(P \times P)$, and $\mathbf{C}_{ISI}^{\frac{1}{2}}(Q \times Q)$, in a row-wise order. The counters, multipliers, and adders work together to generate the proper read addresses for the three RAMs. Counters have different clock periods (integer multiples of one BCP) and modulo operators, O , P , and Q . Two multipliers are employed to multiply outputs of the three RAMs together. Their results are the coefficients of the matrix $\mathbf{C}_h^{\frac{1}{2}}(0)$ in a row-wise order.

In the VM module, two buffers storing $Z_{c_i}(Rk)$ and $Z_{s_i}(Rk)$ are output in a proper order to multiply the corresponding coefficients of $\mathbf{C}_h^{\frac{1}{2}}(0)$. Take the buffer storing $Z_{c_i}(Rk)$ for example, it repeatedly outputs the sequence, $Z_{c_1}(Rk), Z_{c_2}(Rk), \dots, Z_{c_{(OPQ)}}(Rk)$, for (OPQ) times for each time index k . The timing of the $\mathbf{C}_h^{\frac{1}{2}}(0)$ module is aligned with the outputs of the two buffers to ensure correct multiplication. The accumulators sum the (OPQ) results in every (OPQ) BCPs and the final results are down-sampled by the same rate to yield $H_c(l, Rk)$ or $H_s(l, Rk)$. Therefore, for each time index k , it takes (OPQ) BCPs to output one single $H_c(l, Rk)$ and $H_s(l, Rk)$, and $(OPQ)^2$ BCPs to output all results of $H_c(l, Rk)$ and $H_s(l, Rk)$ for $l = 1, \dots, OPQ$.

3.4 Interpolator Module

The interpolator module performs a linear interpolation with a rate R to meet the real-time requirements. The datapath of the interpolator module is shown in Fig. 5. The inputs from the CM module, $H_c(l, Rk)$ and $H_s(l, Rk)$, are delayed by $(OPQ)^2$ BCPs and become $H_c(l, R(k-1))$ and $H_s(l, R(k-1))$, respectively. A enable control block holds “HIGH” for R BCPs and then changes to “LOW” for $(OPQ - R)$ BCPs. Therefore, the output of the counter increases from 0 to $(R - 1)$ in the first R BCPs and holds $(R - 1)$ in the rest of $(OPQ - R)$ BCPs in every (OPQ) BCPs. The counter output is multiplied with $1/R$ as well as $H_c(l, Rk) - H_c(l, R(k-1))$ and $H_s(l, Rk) - H_s(l, R(k-1))$, respectively. The results are added to $H_c(l, R(k-1))$ and $H_s(l, R(k-1))$, respectively, to obtain $H_c(l, k)$ and $H_s(l, k)$ as the final outputs.

4 EXAMPLES AND PERFORMANCE EVALUATION

The proposed discrete-time MIMO triply selective fading channel emulator was implemented on an Altera Stratix III EP3SL150F1152C2N FPGA/DSP development

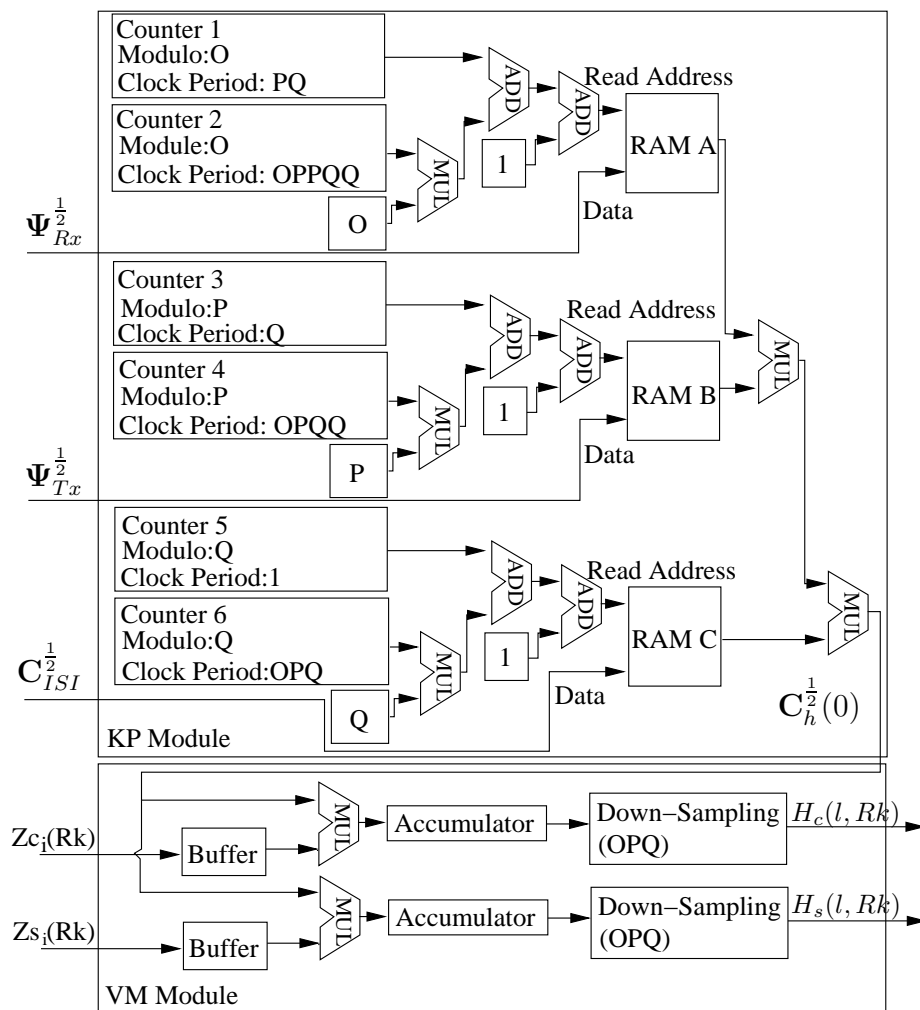


Figure 4. Implementation of the correlation multiplier module.

kit. The basic clock frequency of the FPGA chip was $F_{clk} = 50$ MHz, which provided BCP=20 ns. We used Quartus II version 8.0, DSP Builder version 5.0, and Matlab Simulink for this development and hardware-in-the-loop (HIL) test.

4.1 $C_{ISI}^{\frac{1}{2}}$ Generator Performance Evaluation

As an example, the $C_{ISI}^{\frac{1}{2}}$ generator module was implemented to compute the coefficients for the typical urban channel model with a 12-tap PDP, as shown in Fig.6. This is a commonly used channel model specified in the 3GPP standard [26, pg.70].

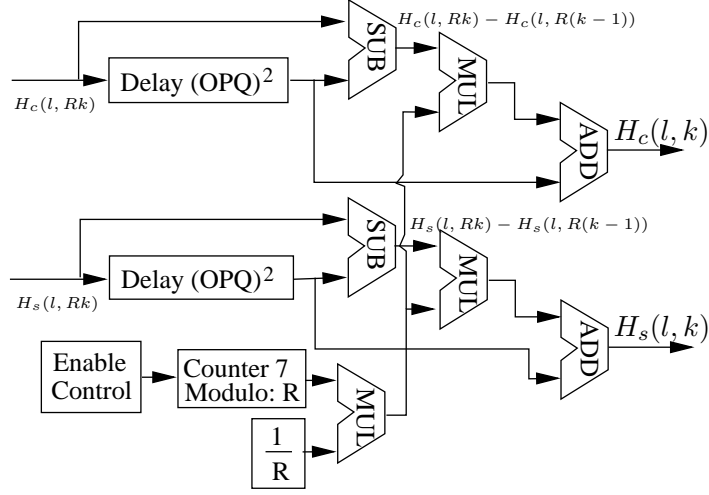


Figure 5. Implementation of the interpolator module.

The transmit and receive filters were the square-root raised cosine (SRC) filters with a roll-off factor 0.3 and group delay $3T_s$, where $T_s = 3.69 \mu s$. We configured $Q_1 = 4$ and $Q_2 = 5$, therefore the size of $\mathbf{C}_{ISI}^{\frac{1}{2}}$ was 10×10 . The coefficients of $\mathbf{C}_{ISI}^{\frac{1}{2}}$ with three fixed-point formats (Q2.6, Q2.10, and Q2.14) were generated by the $\mathbf{C}_{ISI}^{\frac{1}{2}}$ generator and compared to those of Matlab floating-point computation. Their squared error per coefficient and mean squared errors (MSE) are shown in Fig.7, where the x-axis is the coefficient index. All the three fixed-point formats had small MSEs less than -30 dB. The outputs with Q2.10 and Q2.14 resulted in similar MSEs, but the former consumed less hardware resource. Therefore, the Q2.10 format was selected as a good tradeoff between performance and cost.

4.2 KP Module Memory Usage Evaluation

The proposed KP module can save a large amount of memory by computing $\mathbf{C}_h^{\frac{1}{2}}(0)$ in real-time without storing. Instead of pre-computing and storing all coefficients of $\mathbf{C}_h^{\frac{1}{2}}(0)$, the proposed KP method only stores three small matrices, $\Psi_{Rx}^{\frac{1}{2}}$, $\Psi_{Tx}^{\frac{1}{2}}$, and $\mathbf{C}_{ISI}^{\frac{1}{2}}$, and computes the Kronecker product using a serial structure with

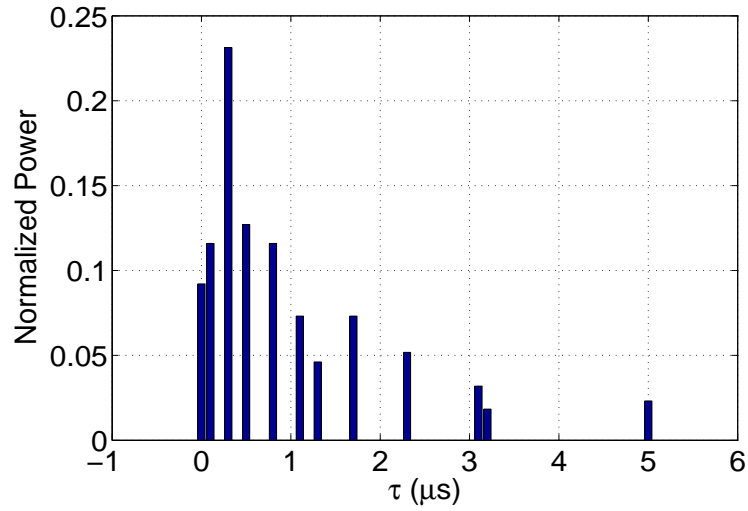


Figure 6. The normalized PDPs for the typical urban channel model presented in 3GPP standard [26].

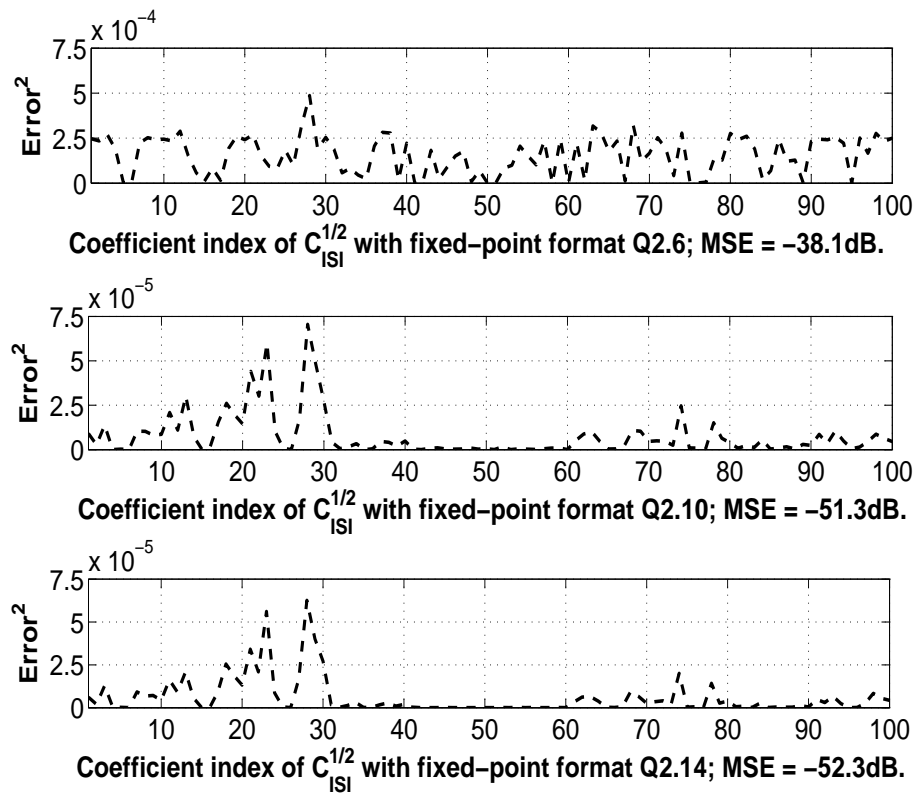


Figure 7. The squared error and MSE comparison of coefficients of $C_{ISI}^{1/2}$ between hardware fixed-point and Matlab floating-point. Note that the scales of y axes are different in sub-figures.

five multipliers, six adders and six counters. The memory usage of the pre-compute and store method is $(OPQ)^2$ words, a quadratic function of the matrix size; while the proposed KP method requires only $(O^2 + P^2 + Q^2)$ words. The comparison for typical values of O , P , and Q is shown in Fig. 8 which clearly demonstrates that the memory required by the pre-compute and store method becomes prohibitively high when the matrix size (OPQ) is large, making the method impractical. A large amount of hardware memory can be saved by the proposed KP method with only slight increase in hardware resource usage, making it a better choice for triply-selective channel emulation.

4.3 Frequency Selective Fading Channel Example

A doubly-selective channel emulator was implemented using the proposed method by configuring Ψ_{Rx} and Ψ_{Tx} to identity matrices thus simultaneously generates $(O \times P)$ doubly-selective channels. The implementation parameters were selected as $M=16$ (number of sinusoids), $T_s=3.69 \mu s$ (symbol duration), $f_d T_s=0.001$ (normalized Doppler), $Q=10$ (channel taps), and $R=140$ (interpolation rate). The PDPs and transmit/receive filters were the same as those in Section 4.1.

The performance of this emulator was evaluated by the auto/cross-correlation of its output waveforms, as shown in Fig. 9. The theoretical autocorrelation function of $h_{1,1}(k, l)$ is given by $c(l, l) \cdot J_0[2\pi f_d(k_1 - k_2)T_s]$; while the theoretical cross-correlation function of $h_{1,1}(k, l_1)$ and $h_{1,1}(k, l_2)$ is given by $c(l_1, l_2) \cdot J_0[2\pi f_d(k_1 - k_2)T_s]$, where the correlation coefficients were $c(0, 0)=0.7794$, $c(0, 1)=0.1551$, and $c(0, 2)=-0.0544$. The auto/cross-correlation functions of the emulator outputs were computed offline using 50 trials with 2.8×10^4 samples per sub-channel per trial. As can be seen, the results of hardware outputs closely matched theoretical ones.

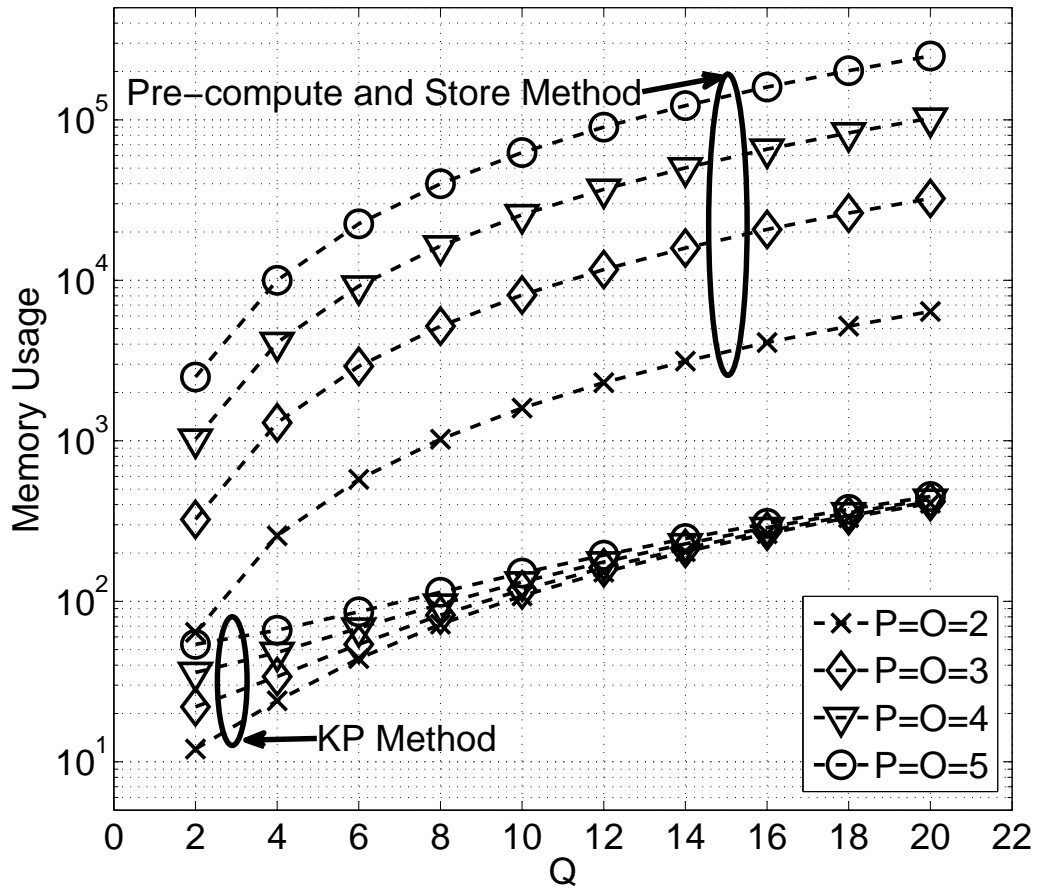


Figure 8. The comparison of memory usage between the pre-compute and store method and the proposed KP method.

4.4 Triply Selective Fading Channel Example

The proposed emulator also implemented the MIMO triply selective channel example presented in [5]. The size of Ψ_{Rx} , Ψ_{Tx} , and \mathbf{C}_{ISI} were $O = P = 2$, and

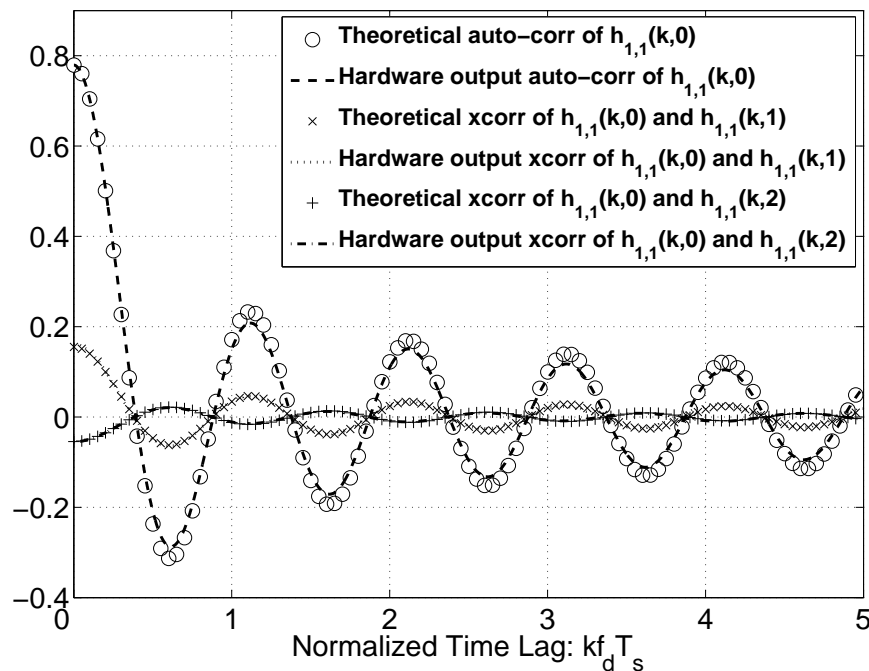


Figure 9. Performance of Doubly selective fading channel emulator. Auto-correlation of $h_{1,1}(k, 0)$, cross-correlation between $h_{1,1}(k, 0)$ and $h_{1,1}(k, 1)$ and between $h_{1,1}(k, 0)$ and $h_{1,1}(k, 2)$. The results were based on hardware outputs of 50 trials with 2.8×10^4 samples per sub-channel per trial.

$Q=4$, respectively, with the spatial correlation matrices given as

$$\Psi_{Tx} = \begin{pmatrix} 1.0000 & 0.2154 \\ 0.2154 & 1.0000 \end{pmatrix}$$

$$\Psi_{Rx} = \begin{pmatrix} 1.0000 & -0.3042 \\ -0.3042 & 1.0000 \end{pmatrix}$$

The PDP was $G(\tau) = A \exp(-\tau/\mu s)$ for $0 \leq \tau \leq 5\mu s$ and zero elsewhere. The transmit filter was a linearized Gaussian filter with a time-bandwidth product equal to 0.3, the receive filter was an SRRC filter with a roll-off factor 0.3. The matrix

\mathbf{C}_{ISI} with $q_1 = -1, 0, 1, 2$ and $q_2 = -1, 0, 1, 2$ were computed by (5) yielding

$$\mathbf{C}_{ISI} = \begin{pmatrix} 0.0091 & 0.0426 & 0.0178 & -0.0016 \\ 0.0426 & 0.3664 & 0.3407 & 0.0367 \\ 0.0178 & 0.3407 & 0.5583 & 0.1414 \\ -0.0016 & 0.0367 & 0.1414 & 0.0602 \end{pmatrix}$$

The parameters, M , f_d , T_s , and R , were the same as those used in the doubly selective fading example. The auto/cross-correlation of several sub-channels was computed in comparison to the theoretical ones and software simulation results reported in [5], as depicted in Fig. 10 and Fig. 11. The theoretical auto/cross-correlations are given by $c_0(l_1, l_2)J_0[2\pi f_d(k_1 - k_2)T_s]$, where $c_0(l_1, l_2)$ is the (l_1, l_2) -th coefficient of $\mathbf{C}_h(0)$. The results of the emulator outputs were also based on 50 trials and they matched the theoretical ones very well.

4.5 Evaluation of Flat Rayleigh Fading Generators

The performance of the flat Rayleigh fading generator was analyzed by statistical properties of FRFG outputs. The FRFG module had the same parameters M , T_s , f_d , and R as those of the previous examples. The probability density function (PDF) of the real/imaginary part of the outputs, the PDF of the envelop, and the level crossing rate (LCR) are compared with the theoretical ones, as shown in Fig. 12–14. The PDF curves of the hardware outputs matched the theoretical ones very well. The LCR of the emulator outputs had slightly lower values than the theoretical ones at lower rates because the number of simulated samples was limited to provide an

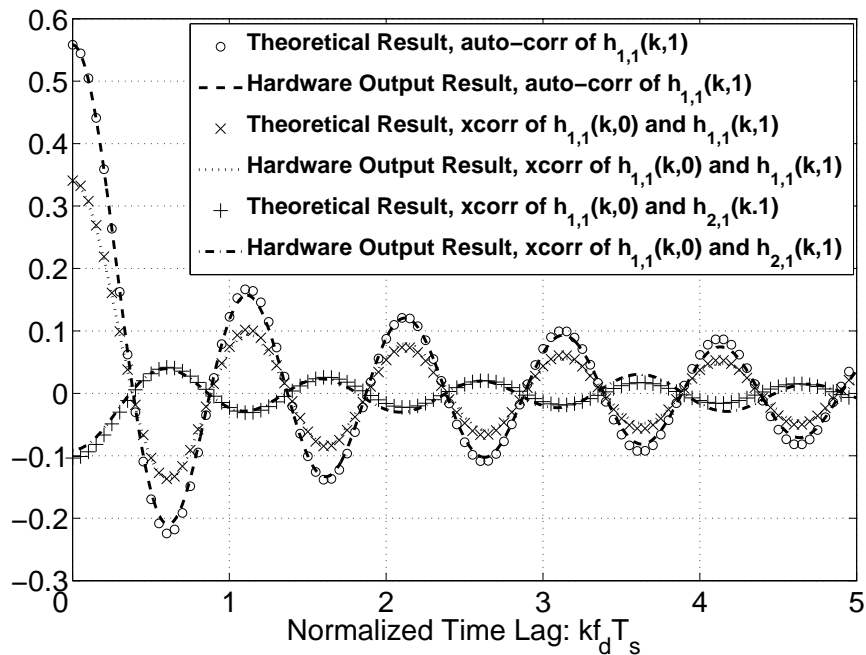


Figure 10. Performance of the triply selective channel emulator. Auto-correlation of $h_{1,1}(k, 1)$, cross-correlation between $h_{1,1}(k, 0)$ and $h_{1,1}(k, 1)$, and between $h_{1,1}(k, 0)$ and $h_{2,1}(k, 1)$. The numbers of trials and samples are the same as those in Fig. 9.

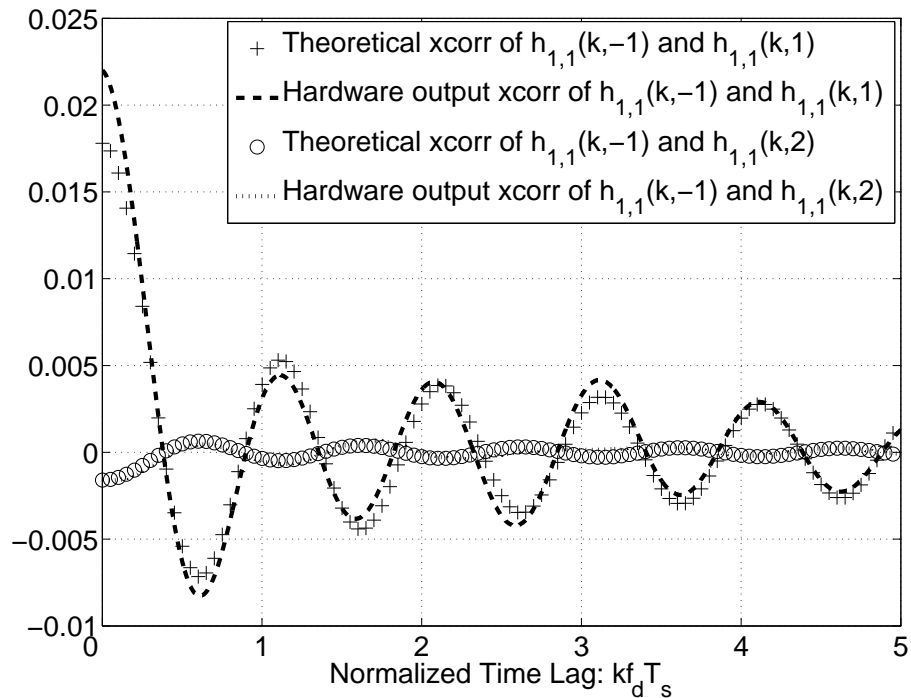


Figure 11. Performance of triply selective channel emulator. Cross-correlation between $h_{1,1}(k, -1)$ and $h_{1,1}(k, 1)$, and between $h_{1,1}(k, -1)$ and $h_{1,1}(k, 2)$. Note the change of scale in y-axis.

accurate LCR count. These results indicated that the hardware implementation of the FRFG module had good accuracy.

4.6 Parameter Specifications and Hardware Usage

The proposed MIMO triply-selective fading emulator is flexible in parameter selection and can be customized to simulate channel scenarios other than the examples presented here. Table 1 shows the parameter ranges of the emulator with the chip clock $F_{clk} = 50$ MHz and symbol duration $T_s = 3.69 \mu s$. The emulator can generate triply-selective channels with any PDPs specified in [26]. For systems with smaller symbol durations, the real-time requirement can be met by increasing the clock frequency F_{clk} or the interpolation rate R such that $(OPQ)^2/(F_{clk}R) < T_s$. This ensures that the number of output coefficients is OPQ/T_s complex samples per second. The normalized Doppler $T_s f_d$ covers most of practical channel scenarios, which is often on the order of 10^{-6} to 10^{-2} . The product (OPQ) is also limited by F_{clk} in this case. Increasing F_{clk} to 200 MHz and keeping T_s and R unchanged lead to $\max(OPQ) = 320$ and the on-chip memory is adequate for channels with $O^2 + P^2 + Q^2 \leq 10^4$. The proposed emulator stores the value of $T_s f_d$ in the Q1.19 format to ensure high accuracy. Each output sample is a complex value whose real and imaginary parts, $H_c(l, k)$ and $H_s(l, k)$, are represented by the Q4.14 format. This is sufficient to avoid overflow and to provide the accuracy of 10^{-4} . The hardware

Table 1. Parameter ranges of the proposed emulator with $F_{clk}=50$ MHz and $T_s = 3.69 \mu s$.

Number of Rx, Tx, and Taps	Normalized Doppler $T_s f_d$	Complex Samples/s	Output Accuracy
$(OPQ) \leq 160$	$1/2^{19} \sim 1$	OPQ/T_s	10^{-4}

usage of the MIMO triply-selective fading emulator with $O = P = 4$ and $Q = 10$ is summarized in Table 2, where ALUT denotes adaptive look-up table, DLR denotes

the dedicated logic register, BM denotes block memory, and DSP means the DSP blocks (high-speed 18-bit multipliers). The percentage uses of the ALUT, DLR, and BM were roughly one third of the total hardware resources of the Stratix III FPGA chip, and the percentage use of the DSP multipliers is about half of the total resource. It is noted that the $C_{ISI}^{\frac{1}{2}}$ Generator consumes the most hardware resources. If the C_{ISI} coefficients and matrix square root calculation were done externally by software, the percentage of ALUT, DLR, BM, and DSP would drop dramatically to 11%, 2%, 13%, and 13% of the total resources, respectively.

Table 2. Resource usage of the MIMO triply selective fading emulator on Stratix III EP3SL150F1152C2N FPGA with $F_{clk}=50$ Mhz.

Module	ALUT	DLR	BM bits	DSP
$C_{ISI}^{\frac{1}{2}}$ Generator	22636	36586	1194944	143
RNG & FRFG	11988	231	618743	16
CM	648	1111	10304	10
Other	120	429	96098	25
Total	35392	38357	1920089	194
percentage	(31%)	(34%)	(34%)	(51%)

The proposed emulator is also compared to other four related emulators of [21,23,24,25], as shown in Table 3, where LE denotes logic elements in Altera FPGA, and LC denotes logic cell in Xilinx FPGA. The LE count is converted from ALUT by $ALUT \approx 1.25LE$ [32]. Although LE and LC are different, one LE is considered equivalent to approximately one LC. Even though it requires slightly more hardware usage, the proposed emulator implements significantly more functionalities than other emulators, including the on-chip $C_{ISI}^{\frac{1}{2}}$ calculation and the CM module incorporating three correlation matrices into the MIMO fading waveforms.

Table 3. Performance comparisons of related Rayleigh fading channel emulators.

	Proposed Emulator	Emulator in [21]	Emulator in [23]	Emulator in [24]	Emulator in [25]
Logic Unit	44240 (LE)	3557 (LC)	19630 (LE)	22272 (LE)	46357 (LC)
Block Memory	1920089	Unknown	822484	Unknown	440960
DSP Element	194	Unknown	Unknown	Unknown	136
Rx×Tx	$O \times P$ ¹	1×1	1×1	4×4	4×4
Number of Taps	Q ¹	3	6	9	Unknown
On-chip $C_{ISI}^{\frac{1}{2}}$ Calculator	Yes	No	No	No	No
Temporal Correlation	SoS	Spectrum filtering	SoS	Spectrum filtering	SoS
Inter-tap Correlation	Yes ²	Yes ³	Yes ⁴	No	Unclear
Spatial Correlation	Yes	No	No	No	Unclear

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

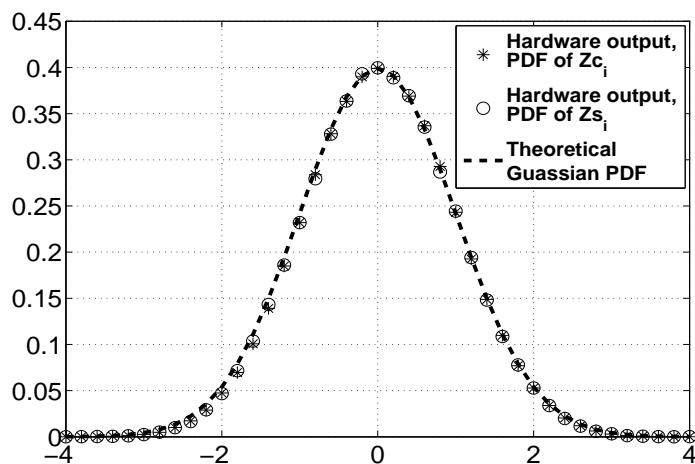


Figure 12. PDF of Z_{c_i} and Z_{s_i} . The numbers of trials and samples are the same as the previous figure.

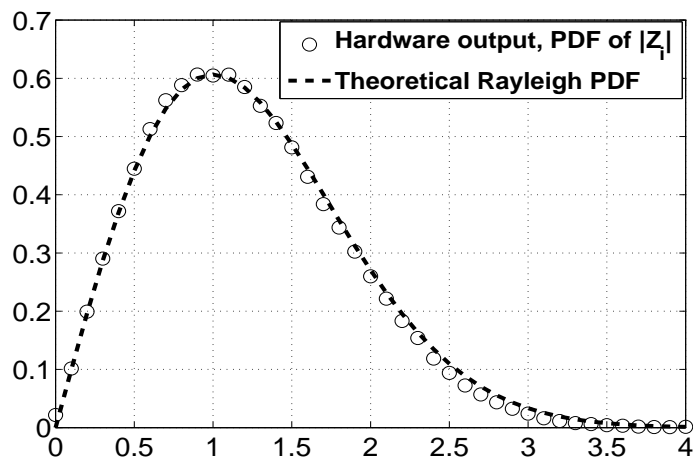


Figure 13. PDF of $|Z_i|$, where $Z_i = Z_{c_i} + jZ_{s_i}$. The numbers of trials and samples are the same as the previous figure.

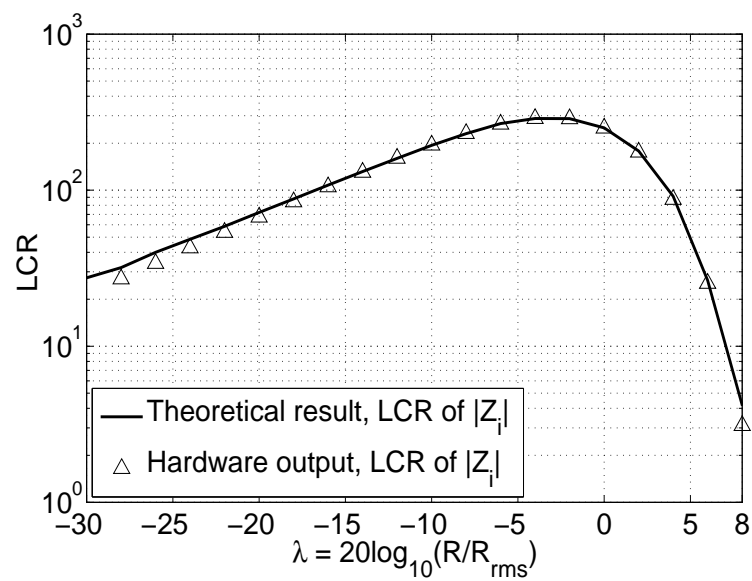


Figure 14. LCR of $|Z_i|$. The numbers of trials and samples are the same to the previous figure.

5 CONCLUSIONS

A hardware implementation scheme has been proposed for discrete-time MIMO triply selective fading emulators which utilizes a mixed parallel-serial structure to

achieve the best tradeoff of hardware usage and output speed. The proposed method is capable of simulating MIMO triply selective fading channels by combining the inter-tap and spatial correlation matrices with uncorrelated flat Rayleigh fading waveforms. The proposed emulator has been implemented on an Altera's Startix III FPGA development kit and meet real-time requirement. The hardware outputs exhibit accurate correlation properties closely matching the theoretical results.

6 REFERENCES

- [1] R. H. Clarke, "A statistical theory of mobile-radio reception," *Bell Syst. Tech. J.*, vol. 47, no. 6, pp. 957-1000, Jul./Aug. 1968.
- [2] H. C. Park, H. S. Lim, and J. W. Yu, "Performance analysis of single-frequency CW signal-based I/Q regeneration in five-port junction-based direct receivers on Rayleigh fading channels," *IEEE Trans. Circuits and Systems II: Express Briefs*, vol. 55, no. 6, pp. 561-565, Jun. 2008.
- [3] L. Wang, C. Zhang, and G. Chen, "Performance of an SIMO FM-DCSK communication system," *IEEE Trans. Circuits and Systems II: Express Briefs*, vol. 55, no. 5, pp. 457-461, May 2008.
- [4] Y. Xia, C.K. Tse, and F.C.M. Lau, "Performance of differential chaos-shift-keying digital communication systems over a multipath fading channel with delay spread," *IEEE Trans. Circuits and Systems II: Express Briefs*, vol. 51, no. 12, pp. 680-684, Dec. 2004.
- [5] 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.
- [6] P. Hoehner, "A statistical discrete-time model for the WSSUS multipath channel," *IEEE Trans. Veh. Technol.*, vol. 41, no. 4, pp. 461-468, Nov. 1992.
- [7] J.I. Smith, "A computer generated multipath fading simulation for mobile radio," *IEEE Trans. Veh. Technol.*, vol. 24, no. 3, pp. 39-40, Aug. 1975.
- [8] 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.

- [9] C. Komninakis, "A fast and accurate Rayleigh fading simulator," in *Proc. IEEE Global Telecommun. Conf.*, San Francisco, CA, Dec. 2003, pp. 3306-3310.
- [10] W. C. Jakes, *Microwave Mobile Communications*, Piscataway, NJ: Wiley-IEEE Press, 1994.
- [11] M. Patzold, U. Killat, F. Laue, and Yingchun Li, "On the statistical properties of deterministic simulation models for mobile fading channels," *IEEE Trans. Veh. Technol.*, vol. 47, no. 1, pp. 254-269, Feb. 1998.
- [12] 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.
- [13] C.S. Patel, G.L. Stuber, and T.G. Pratt, "Comparative analysis of statistical models for the simulation of Rayleigh faded cellular channels," *IEEE Trans. Commun.*, vol. 53, no. 6, pp. 1017-1026, Jun. 2005.
- [14] K.-W. Yip and T.-S. Ng, "Efficient simulation of digital transmission over WS-SUS channels," *IEEE Trans. Commun.*, vol. 43, no. 12, pp. 2907-2913, Dec. 1995.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] A. Alimohammad, S.F. Fard, B.F. Cockburn, and C. Schlegel, "Compact Rayleigh and Rician fading simulator based on random walk processes," *Communications, IET*, vol. 3, no. 8, pp. 1333-1342, Aug. 2009.
- [19] R.A. Goubran, H.M. Hafez, and A.U.H. Sheikh, "Implementation of a real-time mobile channel simulator using a DSP chip," *IEEE Trans. Instrum. Meas.*, vol. 40, no. 4, pp. 709-714, Aug. 1991.
- [20] A. Alimohammad, S.F. Fard, B.F. Cockburn, and C. Schlegel, "A compact single-FPGA fading-channel simulator," *IEEE Trans. Circuits and Systems II: Express Briefs*, vol. 55, no. 1, pp. 84-88, Jan. 2008.
- [21] 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.

- [22] 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.
- [23] Fei Ren and Y.R. Zheng, "A low-complexity hardware implementation of discrete-time frequency-selective Rayleigh fading channels," in *Proc. IEEE IS-CAS*, Taipei, Taiwan, May. 2009, pp. 1759-1762.
- [24] M. Cui, H. Murata, and K. Araki, "FPGA implementation of 4×4 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.
- [25] 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.
- [26] ETSI, "Radio transmission and reception," GSM 05.05, ETSI EN 300910 V8.5.1, 2000.
- [27] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Baltimore, MD: John Hopkins Univ. Press, 1996.
- [28] A. Alimohammad and B.F. Cockburn, "Modeling and hardware implementation aspects of fading channel simulators," *IEEE Trans. Veh. Technol.*, vol. 57, no. 4, pp. 2055-2069, Jul. 2008.
- [29] P. L' Ecuyer, "Tables of maximally equidistributed combined LFSR generators," *Math. of Comp.*, vol. 68, no. 225, pp. 261-269, Jun 1999.
- [30] A. Ahmedsaid, A. Amira, and A. Bouridane, "Efficient systolic array for singular value and eigenvalue decomposition," in *Proc. IEEE International Symposium on Micro-NanoMechatronics and Human Science*, Nagoya, Japan, Oct. 2003, pp. 835-838.
- [31] I. Bravo, P. Jimenez, M. Mazo, J.L. Lazaro, and A. Gardel, "Implementation in FPGAs of Jacobi method to solve the eigenvalue and eigenvector problem," in *Proc. International Conference on Field Programmable Logic and Applications*, Madrid, Spain, Aug. 2006, pp. 1-4.
- [32] Altera Corporation, "Nios II performance benchmarks," Online: http://www.altera.com/literature/ds/ds_nios2_perf.pdf, Oct. 2009.

PAPER

III. HARDWARE IMPLEMENTATION OF TRIPLY SELECTIVE RAYLEIGH FADING CHANNEL SIMULATORS

Fei Ren and Yahong Rosa Zheng

Abstract—In this paper, we implement a real-time hardware triply selective Rayleigh fading simulator. This simulator incorporates the inter-tap and spatial correlation matrices into multiple uncorrelated frequency-flat Rayleigh fading waveforms (including temporal correlation) to simulate a multiple-input multiple-output (MIMO) triply selective Rayleigh fading channel. In the correlation incorporation procedure, this simulator uses a Kronecker product method to save a large amount of hardware memories. Occupying 34% hardware resources of one Stratix III FPGA chip, this simulator can simulate 4×4 MIMO fading channels with 10 correlated delay-taps per subchannel in real-time for a symbol rate of $3.69 \mu s$. Accuracy of this simulator is proved by comparing the statistical properties of its outputs to corresponding theoretical values, and they match perfectly.

1 INTRODUCTION

Wireless fading channel modeling and simulation are very useful for testing and verifying communication algorithm design, transceiver products, and channel capacity analysis. However, the software simulators based on general purpose processors are slow and difficult to meet a real-time simulation requirement. The hardware simulator, which is based on low-cost FPGA and DSP chips, is a preferred solution for the real-time fading channel simulation.

One significant statistical property for wireless fading channel models is the correlation of fading channels waveforms. The subchannels of a MIMO Rayleigh fading channel are time-selective (described by temporal correlation), frequency-selective (exhibiting inter-tap correlation), and space-selective (associated with spatial correlation of transmitters and receivers). This is referred to as the triply selective fading channel containing three types of correlations.

A discrete-time MIMO triply selective Rayleigh fading channel model and software simulation are proposed by [1]. But hardware implementation of MIMO triply selective simulators presents some challenges in accurately computing and incorporating three types of correlations into the discrete-time model. Current reported hardware MIMO simulators do not implement all three types of correlations and may result in inaccurate channel characteristics. For example, the simulator in [2] outputs multiple uncorrelated frequency-flat Rayleigh fading waveforms as MIMO subchannels; while another simulator in [3] attempts to incorporate the inter-tap and spatial correlation matrices into multiple frequency-flat Rayleigh fading waveforms.

In this paper, we propose a hardware implementation method for the discrete-time MIMO triply selective fading simulator on a Stratix III FPGA DSP development kit. This simulator implements all three types of correlations of triply selective channels. The frequency-flat Rayleigh fading waveforms with temporal correlation or Doppler spectrum are generated using a Sum-of-Sinusoids (SOS) method. The inter-tap correlation matrix associated with multipath delay spread is computed according to a channel power delay profile (PDP) and transmit/receive filters. The spatial correlation matrices, including the transmit correlation and receive correlation matrices, are pre-defined inputs associated with antenna arrangements. The matrix square roots of correlation matrices are calculated using an eigenvalue decomposition (EVD) method. Then they are combined with multiple uncorrelated frequency-flat Rayleigh fading waveforms using the Kronecker product and vector multiplicity. The results of

the Kronecker product are computed in real-time for saving hardware memory. Statistical properties of simulator outputs are analyzed and compared to corresponding theoretical ones for performance evaluation.

2 DISCRETE-TIME MIMO TRIPLY SELECTIVE RAYLEIGH FADING MODEL

With accurate statistical properties and computational efficiency for hardware implementation, the discrete-time MIMO triply selective fading model in [1] is chosen as the basis of our hardware implementation. In [1], the MIMO channel matrix at time instant k and delay tap q can be represented as an $(OPQ) \times 1$ coefficient vector $\mathbf{h}_{vec}(k)$, which is defined as

$$\mathbf{h}_{vec}(k) = [\mathbf{h}_{1,1}(k), \dots, \mathbf{h}_{1,P}(k) \mid \dots \mid \mathbf{h}_{O,1}(k), \dots, \mathbf{h}_{O,P}(k)]^t \quad (1)$$

where P and O are the numbers of transmit and receive antennas, respectively (Note we assume the sampling interval being T_s). The vector $\mathbf{h}_{o,p}(k)$ is the (o, p) -th subchannel FIR coefficient vector at time instant k , which is given by

$$\mathbf{h}_{o,p}(k) = [h_{o,p}(-Q_1, k), \dots, h_{o,p}(q, k), \dots, h_{o,p}(Q_2, k)] \quad (2)$$

where Q_1 and Q_2 are nonnegative integers representing the range of q , and $Q = Q_1 + Q_2 + 1$.

In simulation, the vector $\mathbf{h}_{vec}(k)$ can be simulated by

$$\mathbf{h}_{vec}(k) = \mathbf{C}_h^{\frac{1}{2}}(0) \cdot \Phi(k) = (\Psi_{Rx}^{\frac{1}{2}} \otimes \Psi_{Tx}^{\frac{1}{2}} \otimes \mathbf{C}_{ISI}^{\frac{1}{2}}) \cdot \Phi(k) \quad (3)$$

where \otimes denotes the Kronecker product; $\mathbf{X}^{\frac{1}{2}}$ is the square root of matrix $\mathbf{X} = \mathbf{X}^{\frac{1}{2}} \cdot (\mathbf{X}^{\frac{1}{2}})^h$; the matrices Ψ_{Rx} and the matrix Ψ_{Tx} are the spatial correlation matrices determined

by the transmit and receive antennas, respectively; \mathbf{C}_{ISI} is the inter-tap correlation matrix; the vector $\Phi(k)$ is an $(OPQ) \times 1$ vector and $\Phi(k)=[Z_1(k), Z_2(k), \dots, Z_{(OPQ)}(k)]^t$, where $Z_i(k)=Zc_i(k)+jZs_i(k)$ is one of multiple uncorrelated frequency-flat Rayleigh fading waveforms. Each frequency-flat Rayleigh fading waveform $Z_i(k)$ can be efficiently simulated by the SOS method proposed in [4].

For the proposed simulator, the square roots of spatial correlation matrices Ψ_{Rx} and Ψ_{Tx} are specified by users. The inter-tap correlation matrix \mathbf{C}_{ISI} is denoted as

$$\mathbf{C}_{ISI} = \begin{pmatrix} c(-Q_1, -Q_1) & \cdots & c(-Q_1, Q_2) \\ \vdots & \ddots & \vdots \\ c(Q_2, -Q_1) & \cdots & c(Q_2, Q_2) \end{pmatrix} \quad (4)$$

where its coefficients $c(q_1, q_2)$ can be calculated by

$$c(q_1, q_2) = \sum_{n=1}^N \sigma_n^2 R_{P_T P_R}(q_1 T_s - \tau_n) R_{P_T P_R}^*(q_2 T_s - \tau_n) \quad (5)$$

where $R_{P_T P_R}(\xi)$ is the convolution function of the transmit filter and receiver filter; N is the number of total resolvable paths in PDPs; $*$ is the conjugate operator. Parameters σ_n and τ_n are determined by the discrete-time PDPs, $G(\tau)=\sum_{n=1}^N \sigma_n^2 \delta(\tau - \tau_n)$, which are often specified by communication standards like [5].

3 HARDWARE IMPLEMENTATION METHOD

Our proposed hardware simulator can output $\mathbf{h}_{vec}(k)$ in real-time. For the convenience of description, we give new indices elements of $\mathbf{h}_{vec}(k)$,

$$H(l, k) = h_{o,p}(q, k) \quad (6)$$

where $l = Q \cdot [(o - 1) \cdot P + (p - 1)] + (q + Q_1 + 1)$. Therefore, the vector $\mathbf{h}_{vec}(k)$ can be described as

$$\mathbf{h}_{vec}(k) = [H(1, k), H(2, k), \dots, H((OPQ), k)]^t \quad (7)$$

where $H(l, k)$ is a complex fading coefficient and $H(l, k) = Hc(l, k) + jHs(l, k)$.

The proposed hardware simulator consists of four major modules, as shown in Fig. 1. The flat Rayleigh fading generator (FRFG) module generates multiple uncorrelated frequency-flat Rayleigh fading waveforms $Z_i(Rk)$, which have a decimation rate R . The $\mathbf{C}_{ISI}^{\frac{1}{2}}$ generator module computes the inter-tap correlation matrix and its square root. The correlation multiplier (CM) module implements the Kronecker product and vector multiplicity in hardware to perform (3). The interpolator module linearly interpolates samples with an interpolation rate R to increase the output speed for meeting real-time requirement.

Among the four modules, the $\mathbf{C}_{ISI}^{\frac{1}{2}}$ generator and CM module are novel hardware implementations proposed by our paper. The FRFG and interpolator module are similar to the SOS simulator in [3] and will not be described here.

The $\mathbf{C}_{ISI}^{\frac{1}{2}}$ generator module consists of two submodules: the \mathbf{C}_{ISI} generator module that computes the coefficients of \mathbf{C}_{ISI} according to (4) and (5), and the matrix square root (MSR) module that calculates the square root of \mathbf{C}_{ISI} . The datapath of the $\mathbf{C}_{ISI}^{\frac{1}{2}}$ generator module is shown in Fig. 2. The Counters q_1 and q_2 are up counters with same output ranges from $-Q_1$ to Q_2 . The Counter q_1 increases by one in every (NQ) basic clock periods (BCP); while the Counter q_2 increases by one in every N BCPs. Two buffers store σ_n^2 and τ_n , respectively, and sequentially output them. The results of $R_{P_T P_R}(\xi)$ and $R_{P_T P_R}^*(\xi)$ are computed using a lookup table (LUT) scheme. We note the result of $R_{P_T P_R}(\xi)$ is always a real value, which causes $R_{P_T P_R}(\xi) = R_{P_T P_R}^*(\xi)$. Besides, $R_{P_T P_R}(\xi)$ is an even function, so the size of the LUT can be reduced to half by only storing the result of $R_{P_T P_R}(\xi)$ where ξ is a nonnegative value. In our implementation, the LUT R_1 is a D_3 -entry LUT and store the results of $R_{P_T P_R}(\xi)$ where $\xi = (0 : \frac{4T_s}{D_3 - 1} : 4T_s)$. The LUT R_2 is a copy of R_1 . If the results of $|q_1 T_s - \tau_n|$ and $|q_2 T_s - \tau_n|$ are larger than $4T_s$, the R_1 and R_2 output zeros. If

not, they are converted into the proper read addresses of R_1 and R_2 . The outputs of R_1 and R_2 , and corresponding σ_n^2 are multiplied together. In every N BCPs, the accumulator sums N previous inputs to obtain one coefficient of \mathbf{C}_{ISI} , $c(q_1, q_2)$. The \mathbf{C}_{ISI} generator module sequentially outputs the coefficients of \mathbf{C}_{ISI} in a row-wise order.

The MSR module employs the EVD method to find the matrix square root of \mathbf{C}_{ISI} [6]. We note \mathbf{C}_{ISI} is a symmetric positive definite matrix, whose eigenvalues are always positive. The coefficients of \mathbf{C}_{ISI} are stored in a buffer and sent to the EVD module which performs EVD. We employ the Jacobi rotation algorithm to perform EVD, since it is a well-known and accurate method for hardware implementation, the details of which are introduced in [7]. The EVD module outputs three matrices $\mathbf{V}_{C_{ISI}}$, $\mathbf{D}_{C_{ISI}}$, and $\mathbf{V}_{C_{ISI}}^{-1}$, where $\mathbf{C}_{ISI} = \mathbf{V}_{C_{ISI}} \mathbf{D}_{C_{ISI}} \mathbf{V}_{C_{ISI}}^{-1}$. The coefficients of $\mathbf{D}_{C_{ISI}}$ sequentially pass through a square root calculator. Their results are the coefficients of the matrix $\mathbf{D}_{C_{ISI}}^{\frac{1}{2}}$. The coefficients of $\mathbf{C}_{ISI}^{\frac{1}{2}} = \mathbf{V}_{C_{ISI}} \mathbf{D}_{C_{ISI}}^{\frac{1}{2}} \mathbf{V}_{C_{ISI}}^{-1}$ are computed using two matrix multiplier modules. Eventually, the MSR module sequentially outputs them in the row-wise order.

The CM module incorporates the inter-tap and spatial correlation matrices into multiple uncorrelated frequency-flat Rayleigh fading waveforms. It consists of two submodules: the Kronecker product (KP) module, which computes the Kronecker product of $\Psi_{Rx}^{\frac{1}{2}}$, $\Psi_{Tx}^{\frac{1}{2}}$, and $\mathbf{C}_{ISI}^{\frac{1}{2}}$ to obtain $\mathbf{C}_h^{\frac{1}{2}}(0)$, and the vector multiplier (VM) module, which implements $\mathbf{C}_h^{\frac{1}{2}}(0) \cdot \Phi(k)$. Our proposed simulator does not need to store the large size matrix $\mathbf{C}_h^{\frac{1}{2}}(0)$, but employs the KP module to compute it in real-time. The datapath of the CM module is shown in Fig. 3. The RAM A, RAM B, and RAM C store the coefficients of $\Psi_{Rx}^{\frac{1}{2}}(O \times O)$, $\Psi_{Tx}^{\frac{1}{2}}(P \times P)$, and $\mathbf{C}_{ISI}^{\frac{1}{2}}(Q \times Q)$ in the row-wise order. Several counters, multipliers, and adders work together to generate the proper read addresses for three RAMs. The clock periods of Counters 1 – 6 are measured by integer BCPs; while their modulo is related to O , P , and Q . Two multipliers are employed to multiply outputs of three RAMs together. Their results are the coefficients of the matrix $\mathbf{C}_h^{\frac{1}{2}}(0)$ in the row-wise order.

The VM module takes multiple uncorrelated Rayleigh fading waveforms $Zc_i(Rk)$ and $Zs_i(Rk)$ from the FRFG module, and rearranges their order by using two buffers. Taking the buffer storing $Zc_i(Rk)$ for example, the buffer stores the sequence: $Zc_1(Rk)$,

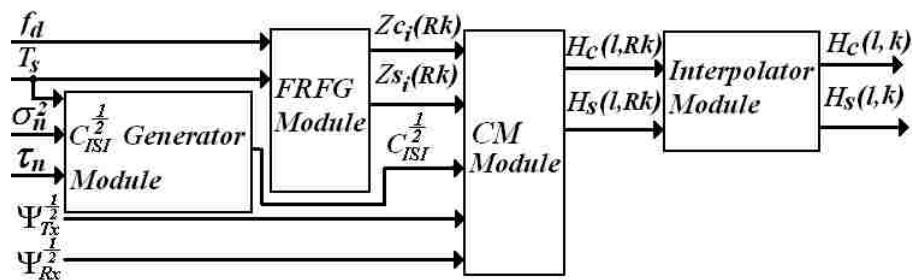


Figure 1. Hardware implementation block diagram of the triply selective fading simulator.

$Zc_2(Rk), \dots, Zc_{(OPQ)}(Rk)$, repeatedly outputs it (OPQ) times, and then do the same to the next sequence: $Zc_1(R(k+1)), \dots, Zc_{(OPQ)}(R(k+1))$. The outputs of two buffers are separately multiplied by the coefficients of $\mathbf{C}_h^{\frac{1}{2}}(0)$. In every (OPQ) BCPs, the accumulator sums the (OPQ) previous inputs to obtain one single $Hc(l, Rk)$ or $Hs(l, Rk)$. Therefore, it takes (OPQ) BCPs to generate one single $Hc(l, Rk)$ or $Hs(l, Rk)$, and $(OPQ)^2$ BCPs to generate all $Hc(l, Rk)$ or $Hs(l, Rk)$ where l ranges from 1 to (OPQ) and k is fixed.

4 EXAMPLES AND PERFORMANCE EVALUATION

The discrete-time MIMO triply selective fading simulator was implemented on an Altera Stratix III EP3SL150F1152C2N FPGA DSP development kit. We used Quartus II version 8.0, DSP Builder version 5.0, and Matlab Simulink for this development.

Memory usage evaluation for the KP module was performed. The proposed KP module computes $\mathbf{C}_h^{\frac{1}{2}}(0)$ in real-time without storing. An alternative method is the pre-compute and store method where the matrix $\mathbf{C}_h^{\frac{1}{2}}(0)$ is pre-computed by software and stored in hardware memory for further access. The memory usage comparison for the two methods is shown in Fig. 4. The y-axis represents the memory usage measured in log, and the x-axis represents the size of $\mathbf{C}_{ISI}^{\frac{1}{2}}$, Q . For the fixed O and P , compared to the pre-compute and

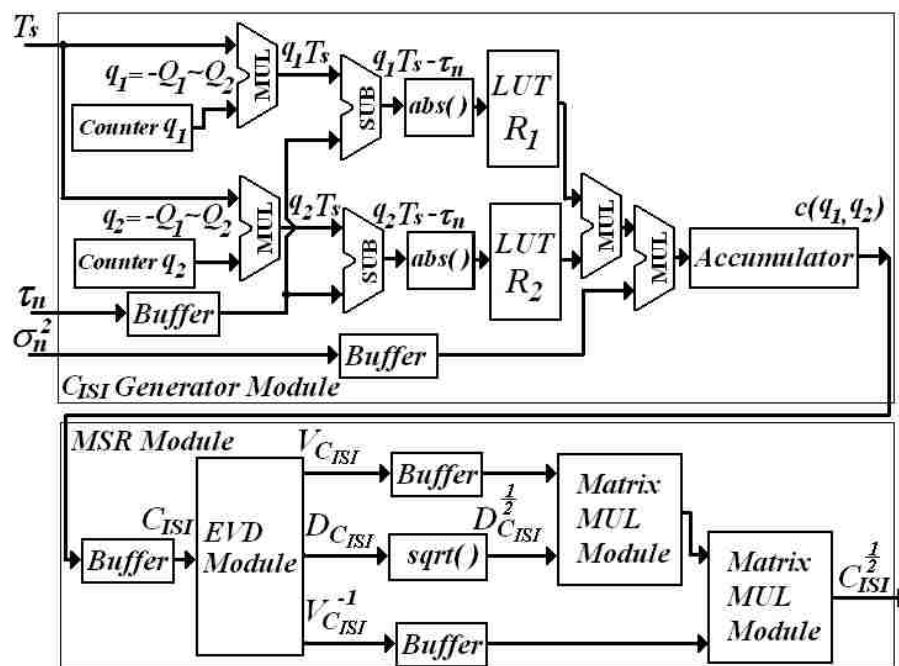


Figure 2. The datapath of the $C_{ISI}^{\frac{1}{2}}$ generator module.

store method, the KP method occupies much less hardware memory, which increases slowly as Q increases.

The performance of the simulator was evaluated through a hardware implementation example with specified parameters. The Matlab simulation and theoretical results with identical parameters have been reported by [1]. We analyzed the statistical properties of hardware outputs and compared them to the theoretical ones. The size of Ψ_{Tx} , Ψ_{Rx} , and C_{ISI} , were $O=P=2$, and $Q=4$, respectively. The matrices Ψ_{Tx} and Ψ_{Rx} were given as follows:

$$\Psi_{Tx} = \begin{pmatrix} 1.0000 & 0.2154 \\ 0.2154 & 1.0000 \end{pmatrix}$$

$$\Psi_{Rx} = \begin{pmatrix} 1.0000 & -0.3042 \\ -0.3042 & 1.0000 \end{pmatrix}$$

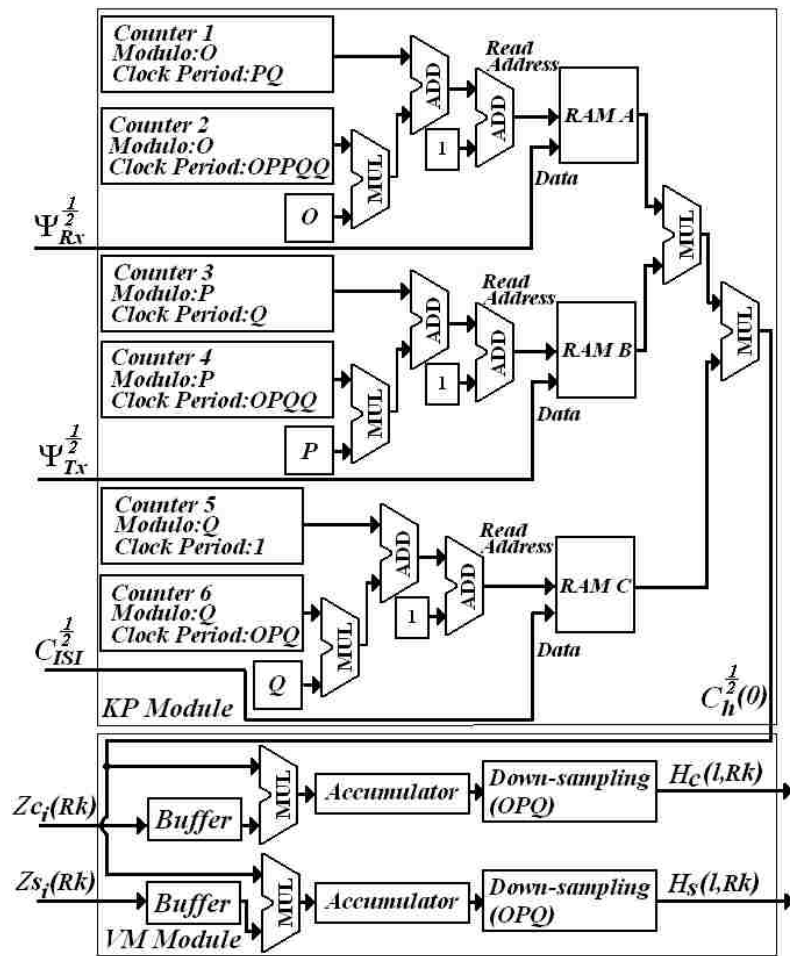


Figure 3. The datapath of the CM module.

The PDP was an exponential function for $0 \leq \tau_n \leq 5 \mu s$. The transmit filter was a linearized Gaussian filter with a time-bandwidth product 0.3, and the receive filter was an SRC filter with a roll-off factor 0.3. Other implementation parameters were: $F_{clock} = 50 MHz$, $T_s = 3.69 \mu s$, $f_d T_s = 0.001$, and the interpolation rate $R = 140$.

The proposed hardware simulator met the real-time requirement and output 4.34×10^6 correlated fading complex coefficients per second. When simulated by matlab, this fading channel scenario took approximate 1 second to output these coefficients. All outputs have

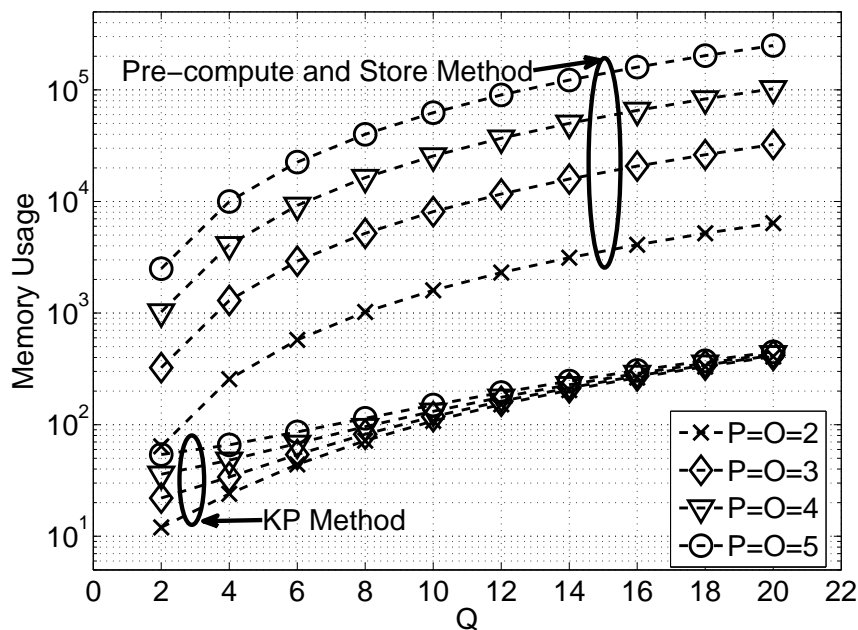


Figure 4. The memory usage comparison of the pre-compute and store method and the proposed KP method.

the fixed-point format Q4.14, which is long enough to provide high accuracy and avoid overflow. Based on the hardware outputs, the auto/cross-correlation between several triply selective channels was computed and depicted in Fig. 5. The matrix \mathbf{C}_{ISI} with $q_1 = -1, 0, 1, 2$ and $q_2 = -1, 0, 1, 2$ was computed and shown as

$$\mathbf{C}_{ISI} = \begin{pmatrix} 0.0091 & 0.0426 & 0.0178 & -0.0016 \\ 0.0426 & 0.3664 & 0.3407 & 0.0367 \\ 0.0178 & 0.3407 & 0.5583 & 0.1414 \\ -0.0016 & 0.0367 & 0.1414 & 0.0602 \end{pmatrix}$$

Therefore, three theoretical curves were 0.5583, 0.3407 and -0.1036 multiplying by $J_0[2\pi f_d(k_1 - k_2)T_s]$, respectively. As can be seen, the correlation curves of hardware outputs matched them very well.

We evaluated hardware resource usage using a hardware implementation example with $O=P=4$ and $Q=10$. Hardware usage is summarized in Table 1, where ALUT denotes

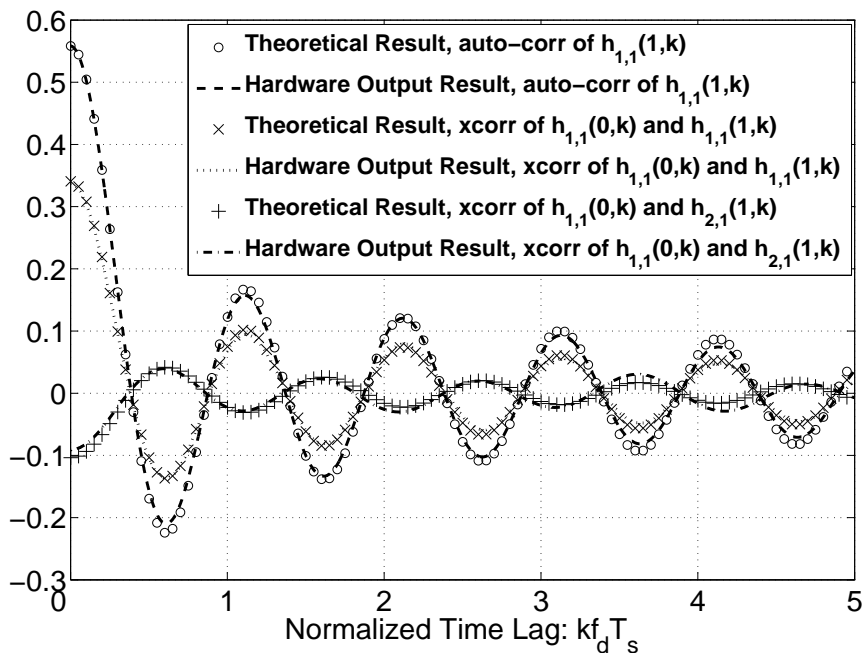


Figure 5. The auto-correlation of $h_{1,1}(1, k)$, the cross-correlation between $h_{1,1}(0, k)$ and $h_{1,1}(1, k)$, and cross-correlation between $h_{1,1}(0, k)$ and $h_{2,1}(1, k)$. The channel index is according to (2). The results are based on hardware outputs of 50 trials with 2.8×10^4 samples in each channel per trial.

adaptive look-up table, DLR is dedicated logic register, BM denotes block memory, and DSP means the DSP blocks (high-speed 18-bit multipliers). The percentage uses of total hardware resources were roughly one third for ALUT, DLR, and BM of one Stratix III FPGA chip and slightly more than a half of the DSP multipliers were utilized.

Table 1. Hardware usage of the simulator on a Stratix III EP3SL150F1152C2N FPGA chip.

	ALUT	DLR	BM bits	DSP
$C_{ISI}^{\frac{1}{2}}$ Generator	22636	36586	1194944	143
FRFG	11988	231	618743	16
CM	648	1111	10304	10
Other	120	429	96098	25
Total	35392 (31%)	38357 (34%)	1920089 (34%)	194 (51%)

5 CONCLUSIONS

A hardware discrete-time MIMO triply selective Rayleigh fading simulator has been implemented on an Altera Startix III FPGA DSP development kit. This simulator is capable of simulating MIMO triply selective fading channels with all three types of correlations in real-time. The outputs of the simulator are evaluated and proved to contain accurate statistical properties as expected.

6 REFERENCES

- [1] C. Xiao, J.X. Wu, S-Y. Leong, Y.R. Zheng, and K. B. Letaief, "A discret-time model for triply selective MIMO Rayleigh fading channels," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1678-1688, Sep. 2004.
- [2] M. Cui, H. Murata, and K. Araki, "FPGA implementation of 4×4 MIMO test-bed for spatial multiplexing systems," in *Proc. IEEE ISPIMRC.*, Barcelona, Spain, Sep. 2004, pp. 3045-3048.
- [3] 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 ICC.*, Beijing, China, May. 2008, pp. 718-724.
- [4] 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.
- [5] ETSI, "Radio transmission and reception," GSM 05.05, ETSI EN 300910 V8.5.1, 2000.
- [6] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Baltimore, MD: John Hopkins Univ. Press, 1996.
- [7] I. Bravo, P. Jimenez, M. Mazo, J.L. Lazaro, and A. Gardel, "Implementation in FPGAs of Jacobi method to solve the eigenvalue and eigenvector problem," in *Proc. International Conference on Field Programmable Logic and Applications*, Madrid, Spain, Aug. 2006, pp. 1-4.

PAPER

**IV. A LOW-COMPLEXITY HARDWARE IMPLEMENTATION
OF DISCRETE-TIME FREQUENCY-SELECTIVE
RAYLEIGH FADING CHANNELS**

Fei Ren and Yahong Rosa Zheng

Abstract—A low-complexity hardware implementation method is proposed for discrete-time frequency-selective Rayleigh fading channels. The proposed method first employs the Sum-of-Sinusoids method to generate multiple independent flat fading channel responses, then utilizes a simple weight-delay-sum filtering method to incorporate the fractionally-delayed multipath rays into inter-tap correlated tap gains. It thus achieves accurate correlation properties in both inter-tap correlation and temporal correlation (or Doppler spectrum). The proposed method is implemented by an Altera Stratix II FPGA development kit and the results show excellent performance match with those by MATLAB software simulations.

1 INTRODUCTION

Wireless fading channel modeling and simulation provide a low-cost means for testing and verification of transceiver products, new algorithm design, and channel capacity analysis. A most commonly used model is the Rayleigh fading Wide-Sense Stationary Uncorrelated Scattering (WSSUS) channel which is often simulated by one of the two methods: the Sum-of-Sinusoid and the Doppler spectrum filtering method [1]. Hardware and software implementations of frequency-flat fading channels have been well studied and reported by, for example, [1, 2, 3, 4] and reference herein. Software implementation of frequency-selective fading channels has also been well investigated [5, 6, 7]. However, hardware simulation of frequency-selective fading channels still presents some challenge in computational complexity and simulation accuracy [8, 9]. The most difficult aspect of frequency-selective fading

channel simulation is to accurately compute and incorporate the cross-correlation between multiple channel taps in the discrete-time model. Although the WSSUS model assumes multiple uncorrelated rays, the sampled discrete-time channel taps are often correlated due to the bandpass nature of wireless communications systems. Many current hardware implementations fail to consider these correlation and result in inaccurate channel characteristics.

In this paper, we propose a simple and elegant method to incorporate inter-tap correlation for hardware implementation of discrete-time frequency-selective fading channels. The proposed method employs the weight-delay-sum filtering method [10] to implement the fractional delays of the multiple WSSUS rays. It combines the weight-delay-sum method with SoS flat fading simulators and ensures low-complexity for real-time hardware implementation. The proposed simulation method is implemented by Altera's Stratix II Field Programmable Gate Array (FPGA) development kit. The results show excellent performance match with those of MATLAB software implementation. The proposed method has advantages in low computational complexity, fast data rate, and more accurate waveforms and correlation properties, in comparison with existing hardware implementation methods.

2 DISCRETE-TIME FREQUENCY-SELECTIVE FADING CHANNEL MODELS

The frequency-selective Rayleigh fading channel model is often expressed as the baseband equivalent channel impulse response consisting of multipath [1]

$$h(\tau, t) = \sum_{i=1}^I P_i g(\tau - \tau_i) \exp[-j(\omega_i(t - \tau_i) - \phi_i)] \quad (1)$$

where P_i , ω_i , and τ_i are the i -th multipath gain, angular Doppler frequency, and relative delay, respectively. The pulse shaping filter $g(\tau)$ is a bandpass filter often implemented by a raised cosine filter [1]. The multipath gains P_i are normalized to yield unit total power

of the response. It is commonly assumed that the multiple rays in (1) are Wide-Sense stationary uncorrelated scattering (WSSUS).

When the delay spread $\tau_{max} - \tau_{min}$ is much smaller than the symbol interval T_{sym} , the channel impulse response can be assumed as frequency-flat fading

$$h(t) = \sum_{i=1}^I P_i g(t) \exp[-j(\omega_i t - \phi_i)] \quad (2)$$

If sampled at T_{sym} interval, the discrete-time flat fading channel can be efficiently simulated by several SoS models [1, 3] and a typical one is

$$\begin{aligned} Z(k) &= Z_c(k) + jZ_s(k), \\ Z_c(k) &= \sqrt{\frac{2}{M}} \sum_{n=1}^M \cos(\omega_d k \cos \alpha_n + \phi_n), \\ Z_s(k) &= \sqrt{\frac{2}{M}} \sum_{n=1}^M \cos(\omega_d k \sin \alpha_n + \varphi_n), \\ \alpha_n &= \frac{2n\pi - \pi + \theta}{4}, \quad n = 1, 2, \dots, M. \end{aligned} \quad (3)$$

where ω_d is the maximum angular Doppler frequency, M is the total number of sinusoids, and $j = \sqrt{-1}$. The angle of arrival α_n is randomized by a uniformly-distributed θ , and ϕ_n and φ_n are the random phases of the in-phase and quadrature components, respectively. The random variables ϕ_n , φ_n , and θ are statistically independent and uniformly distributed on $[-\pi, \pi)$ for all n .

When the channel coherence time is comparable to or larger than the symbol interval, the fading channel is frequency-selective and inter-symbol interference often spans multiple symbol intervals. The sampled channel response (1) becomes a time-varying FIR system

$$H(l, k) = \sum_{i=1}^I P_i g(lT_s - \tau_i) Z_i(k), \quad (4)$$

where $Z_i(k), i = 1, \dots, I$, are independent flat fading CIRs generated by (3). However, the multipath delays τ_i are often fractions of the symbol interval. Sampling the fractional delays at T_{sym} (or at $T_s = T_{sym}/U$, where typically the upsampling rate $U \in [1, 10]$.) results in correlated inter-symbol delay taps [5, 6]

$$E[h(l_1, k)h^\dagger(l_2, k)] = \sum_{i=1}^I \sum_{k=1}^I P_i P_k g(l_1 T_s - \tau_i) g^\dagger(l_2 T_s - \tau_k), \quad (5)$$

note that $R_{gg}(\xi) = E[g(\tau)g^\dagger(\tau + \xi)]$ is the autocorrelation of the bandpass filter $g(\tau)$. The resulting discrete-time power delay profile is shown in Fig.1.

Several methods have been proposed to incorporate the inter-tap correlation in frequency-selective channel modeling including the spectrum factorization method [7] and the correlation matrix factorization method [5, 6]. It has been shown that these methods yield accurate channel models with low computational complexity in software-based simulation. However, the evaluation of correlation coefficients, and the spectrum and/or correlation matrix factorization are costly in hardware implementation. Therefore, we propose a simple weight-delay-sum filtering method [10] to implement the fractional delays,

$$\begin{aligned} H(l, k) &= H_c(l, k) + jH_s(l, k) \\ H_c(l, k) &= \sum_{i=1}^I P_i E_{l,i} Z_{c_i}(k) \delta(l - l_i) \\ H_s(l, k) &= \sum_{i=1}^I P_i E_{l,i} Z_{s_i}(k) \delta(l - l_i) \end{aligned} \quad (6)$$

where $l_i = \lfloor \tau_i/T_s \rfloor$, and $E_{l,i} = g(lT_s - \tau_i)$ are T_s -spaced samples of the delayed bandpass filter, as shown in Fig. 2, where the raised cosine pulse is truncated to $\pm L_g T_s$ with $L_g = 3$.

The simple weight-delay-sum method captures the inter-tap correlation of frequency-selective channels with very low computational complexity. The tradeoff is that it requires I independent flat fading waveforms rather than $L = 2L_g + 1 + \lceil \tau_{max}/T_s \rceil$ required in the correlation matrix factorization method [5]. In practice, the number of multipath I is often slightly larger than the total number of taps L .

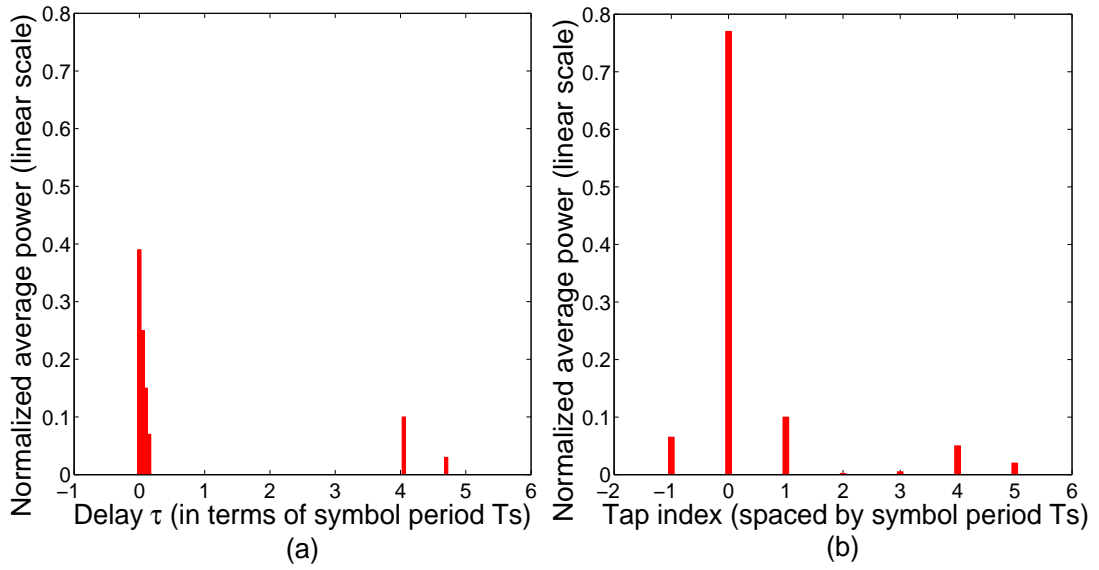


Figure 1. (a) A typical urban channel PDP with multiple WSSUS rays. (b) Average power/tap of T_s -spaced discrete-time channel response.

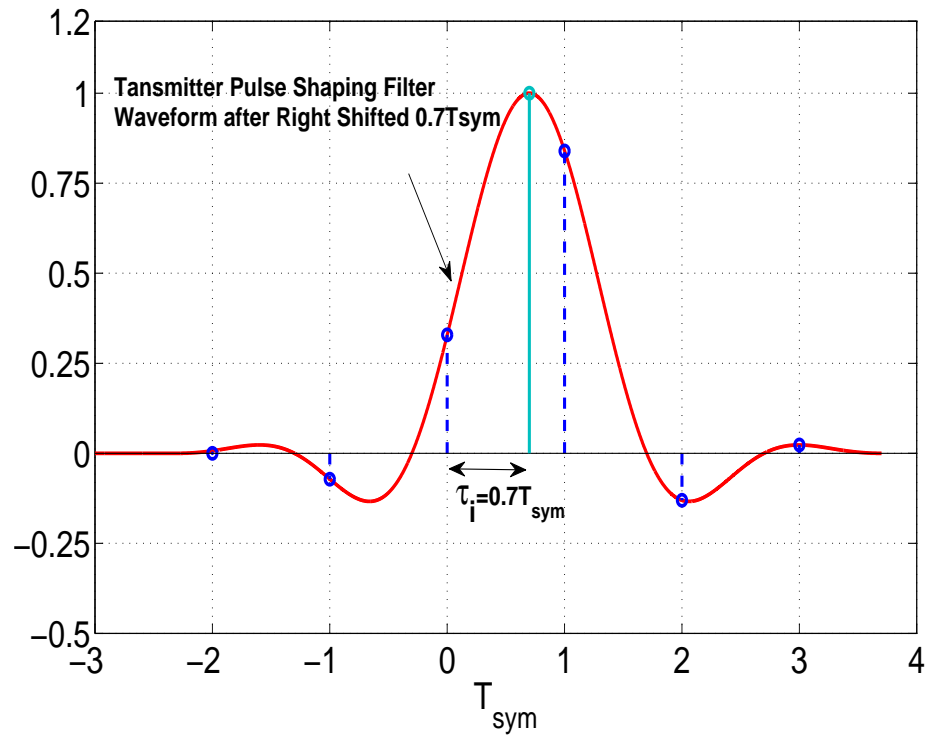


Figure 2. Bandpass filter of the i -th ray sampled at T_{sym} , where the delay τ_i is a fraction of T_{sym} .

3 FPGA IMPLEMENTATION

For real-time hardware implementation, frequency-selective channel waveforms must be sampled at the same rate as the receiver and the received signal (after proper delay) is then

$$y(k) = \sum_{l=0}^{L-1} H(l, k) \cdot x(k-l) + v(k), \quad (7)$$

where $x(k)$ is the transmitted signal and $v(k)$ is the background white Gaussian noise. If the symbol interval $T_{sym} = 1\mu s$ and the upsampling rate is $U = 10$, then L samples of $H(l, k)$ are needed for every $T_s = 0.1\mu s$, where L is on the order of tens. This requirement is stringent for sample-by-sample processing. However, in modern communications systems, block transmission is often employed and channel response is often slowly time varying. We exploit this feature and propose an efficient implementation with block processing.

The proposed hardware implementation scheme consists of three major blocks: a parameter generator bank, a flat fading generator, and a selective fading generator module, as shown in Fig. 3, where MUX is a multiplexer. The parameter generator bank generates and stores all random variables needed for each of the I WSSUS rays. These include the random phase vectors $\Phi_i = [\phi_{1,i}, \phi_{2,i}, \dots, \phi_{M,i}]$ and $\Psi_i = [\varphi_{1,i}, \varphi_{2,i}, \dots, \varphi_{M,i}]$, the maximum Doppler frequencies $\omega_{d,i}$, random phases θ_i , and the power delay profile vectors $\mathbf{P} = \{P_i\}$ and $\mathbf{D} = \{\tau_i\}$. The parameter generator bank also computes and stores the quantities $\cos \alpha_{n,i}$ and $\sin \alpha_{n,i}$ for all n and i . The multiplexer selects the parameters of the i -th ray and sends them to the flat fading generator in series. The flat fading generator generates the real and imaginary components of the i -th flat fading channel responses according to (3) and outputs $Z_{c_i}(k)$ and $Z_{s_i}(k)$ to two buffers of the selective fading generator. When the k -th flat fading samples of all I rays are ready at the buffers, the selective fading generator processes them with the weight-delay-sum filtering method according to (6).

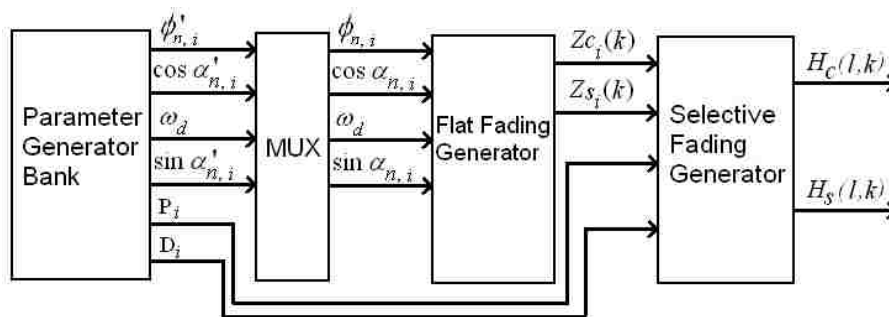


Figure 3. Block diagram of FPGA implementation of the frequency-selective Rayleigh fading simulator.

The implementation of the parameter generator bank is straightforward with several uniform random number generators and the sine and cosine functions are generated by Look Up Tables (LUT). The flat fading generator is implemented as in Fig. 4, where M cosine functions are summed in series to generate the real/imaginary component of the fading response. Flexible data formats are used for different parameters according to their fixed-point precision. For example, the random phase/Doppler parameters use the format (3:20), the number M uses (2:10), the time-index k uses (21:30), and the channel responses use (3:20). Thus, accuracy of output can reach $2^{-20} \approx 10^{-6}$.

The selective fading generator is the core module of the simulator and its structure is shown in Fig. 5. The i -th flat fading channel responses are multiplied with its gain P_i according to the PDP specifications prior to be stored in the buffers. The weights $E_{l,i} = g(lT_s - \tau_i)$ are computed through multiple LUTs which store the raised cosine pulse for $\tau = [-L_g T_{sym} : L_g T_{sym}]$ at a high resolution. The LUTs takes the delay parameter $D_i = \tau_i$ as the inputs and then outputs the corresponding weights $E_{l,i}$ to the multipliers (MUL). Multiple MULs are used to weigh the corresponding flat fading rays in parallel. The accumulators implement the summation of (6) and output a block of $H_c(l, k)$ and $H_s(l, k)$ in parallel.

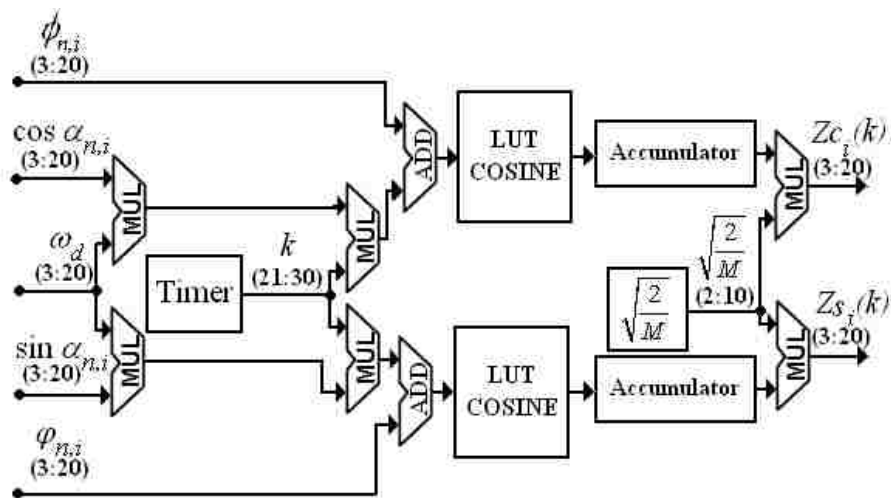


Figure 4. FPGA implementation of the flat fading generator module.

4 IMPLEMENTATION EXAMPLE AND PERFORMANCE EVALUATION

The proposed frequency selective fading channel simulator was implemented by an Altera Stratix II FPGA/DSP development kit. We used Quartus II version 8.0 and DSP Builder version 5.0 for this development. DSP Builder provides a nice interface between the FPGA hardware and MATLAB Simulink so that the parameters of channel specifications were easily input to the channel simulator, and the outputs of the channel simulator were logged in data files in Simulink.

As an example, results for a typical urban channel model of 20 WSSUS rays is presented here. The implementation parameters were: the number of sinusoid $M = 16$, the upsampling rate $U = 10$, the output block size is 10×1 per accumulator, and the channel length $L = 60$ (in terms of T_s). When the clock period of the FPGA chip is set to 20ns, it meets the real-time requirements for symbol interval $T_{sym} = 6.4\mu s$. The logic utilization of the single FPGA chip was 33%, including 15704 (31%) combinational ALUTs and 1383 (2%) dedicated logic registers. Total block memory bits occupied was 822484 (32%). The

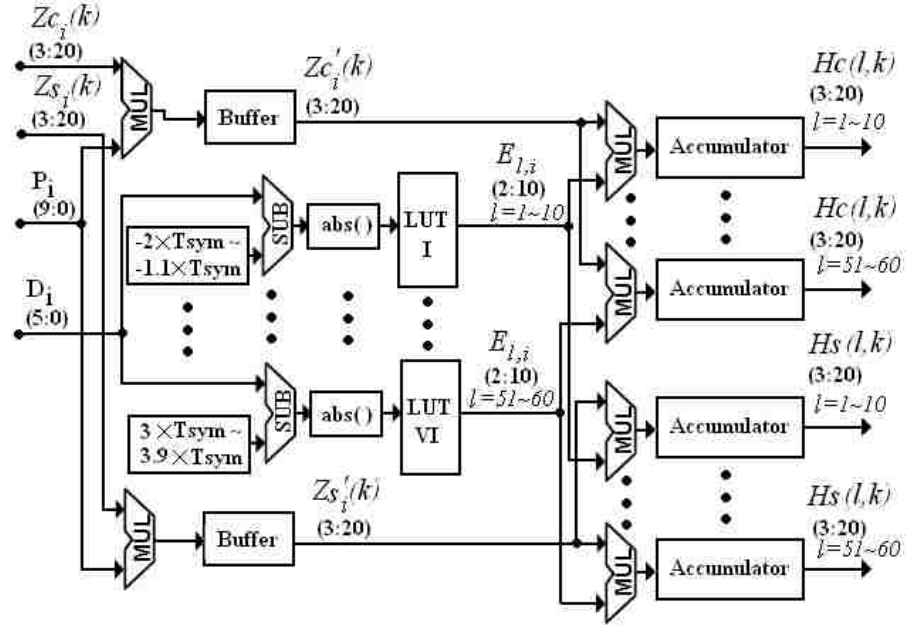


Figure 5. FPGA implementation of frequency-selective fading generator module.

proposed low-complexity hardware implementation occupies less than 1/3 resources on the single FPGA chip.

The performance of the hardware simulator was evaluated by its output waveforms. First, the auto- or cross-correlation of the flat fading generators $Z_{c_i}(k)$ and $Z_{s_i}(k)$ are computed by averaging over five trails and each trial generated $k = 2 \times 10^6$ samples. The results are shown in Fig. 6.

The cross-correlation between $H_c(l, k)$ and $H_c(l + (1, 2, 4), k)$ are shown in Fig. 7. When the accuracy of MATLAB simulations is set to 10^{-6} , which is the same to the accuracy of FPGA outputs. All FPGA outputs match MATLAB simulations very well.

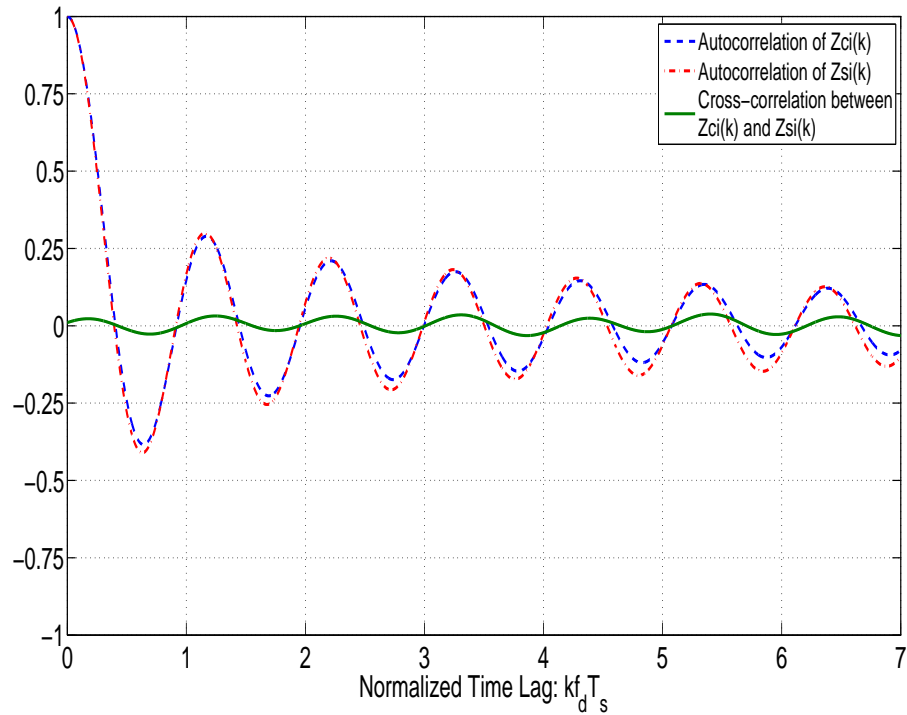


Figure 6. Autocorrelation and cross-correlation of the i -th flat fading ray sampled at at T_{sym} interval. The normalized Doppler frequency was $f_d T_{sym} = 0.0008$ and $f_d = 125$ Hz.

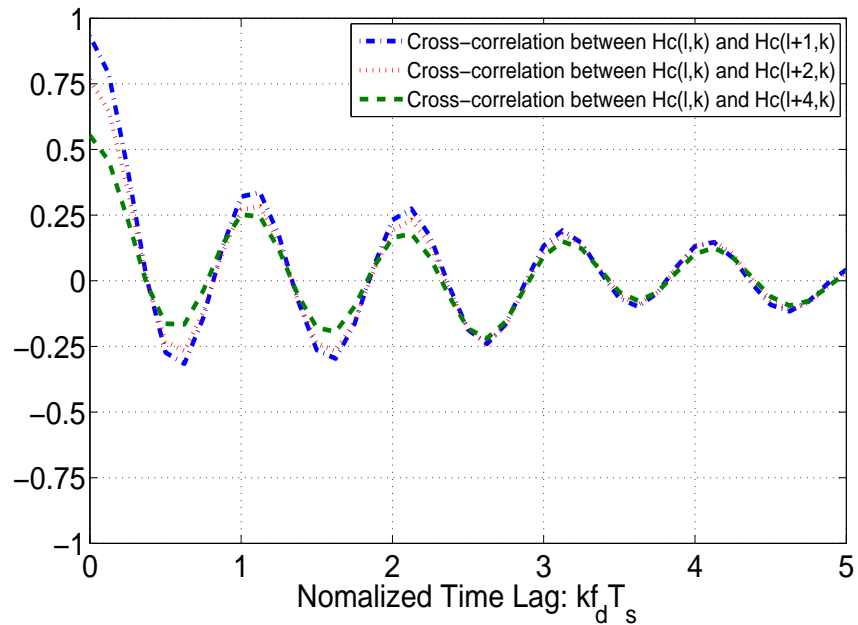


Figure 7. Cross-correlation between $H_c(l, k)$ and $H_c(l+1, k)$ of the frequency-selective channel simulator.

5 CONCLUSIONS

A low-complexity FPGA implementation of frequency selective Rayleigh fading channels has been proposed, which employs a simple weight-delay-sum processing to incorporate the inter-tap correlation of discrete-time channel models. The proposed simulator has been implemented on Altera's Startix II development kits. The results of the hardware simulator match those by the software simulation. The advantages of the proposed simulator include its flexibility for parameter change and its simple, compact implementation.

6 REFERENCES

- [1] W.C. Jakes, *Microwave Mobile Communications*, Piscataway, NJ: Wiley-IEEE Press, 1994.
- [2] C.S. Patel, G.L. Stuber, and T.G. Pratt, "Comparative analysis of statistical models for the simulation of Rayleigh faded cellular channels," *IEEE Trans. Commun.*, vol. 53, no. 6, pp. 1017-1026, Jun. 2005.
- [3] Y.R. Zheng and C. Xiao, "Improved models for the generation of multiple uncorrelated Rayleigh fading waveforms," *IEEE Trans. Communications Letters*, vol. 6, no. 6, pp. 256-258, Jun. 2002.
- [4] A. Alimohammad, S.F. Fard, B.F. Cockburn, and C. Schlegel, "A compact single-FPGA fading-channel simulator," *IEEE Trans. Circuits and Systems II: Express Briefs*, vol. 55, no. 1, pp. 84-88, Jan. 2008.
- [5] 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.
- [6] 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.
- [7] A. Abdi, "Stochastic modeling and simulation of multiple-input multiple-output channels: A unified approach," in *Proc. IEEE Intl. Symp. Antennas Propagat.*, Monterey, CA, Jun. 2004, pp. 3673-3676.
- [8] M. Kahrs and C. Zimmer, "Digital signal processing in a real-time propagation simulator," *IEEE Trans. Instrumentation and Measurement*, vol. 55, no. 1, pp. 197-205, Feb. 2006.

- [9] M.A. Wickert and J. Papenfuss, "Implementation of a real-time frequency-selective RF channel simulator using a hybrid DSP-FPGA architecture," *IEEE Trans. Microwave Theory and Techniques*, vol. 49, no. 8, pp. 1390-1397, Aug. 2001.
- [10] T.I. Laakso, V. Valimaki, M. Karjalainen, and U.K. Laine, "Splitting the unit delay [FIR/all pass filters design]," *IEEE Signal Processing Magazine*, vol. 13, no. 1, pp. 30-60, Jan. 1996.

PAPER

**V. VALIDATION OF THE TRIPLY SELECTIVE FADING CHANNEL
MODEL THROUGH A MIMO TEST BED
AND EXPERIMENTAL RESULTS**

Saurav Subedi, Huang Lou, Fei Ren, Mingxi Wang, and Y. R. Zheng

Abstract—Multiple-input multiple-output (MIMO) channel is often triply selective, meaning that it has spatial, temporal and inter-tap correlation. The temporal correlation is well characterized by its Doppler spectrum, but spatial and inter-tap correlation and their impact on MIMO channels are less studied in the literature. A MIMO testbed has been established to measure the impulse response of MIMO channels and an estimation method is developed to quantitatively measure the correlation matrices from experimental data.

1 INTRODUCTION

The multiple-input multiple-output (MIMO) channel is analyzed as triply selective fading channel in existing literatures, [1], [2]. This model accounts for space-selective, time-selective and frequency-selective nature of MIMO channels. It is shown in [1] that correlation between channel coefficients of the discrete-time MIMO channel can be written as a Kronecker product of temporal correlation, inter-tap correlation and spatial correlations. It is argued in [2] that this model is not accurate and the Kronecker product for the spatial correlations, in general, does not hold in the case of frequency selective channel. The underlying assumptions in [1] are clarified and some emphatic conclusions are drawn in [3] to approve the accuracy of this discrete time model for MIMO triply selective fading channels. A general space-time cross-correlation function incorporating a wide range of parameters of the MIMO fading channel is proposed in [4]. In [5], the vector autoregressive (AR) stochastic models are proposed to simulate multiple cross-correlated Rician fading

channels. The joint effect of spatial and temporal correlation is studied in [6] and analysis of ergodic capacity of a MIMO channel is presented based on the transmit and receive antenna correlations matrices.

This paper validates the triply selective fading channel model through experimental results. We verify the results through the decomposition of the channel coefficient covariance matrix into its Kronecker factors. Approaches for decomposition of Kronecker product into its components are suggested in [7]. However, those methods are applicable only for real matrices. In this paper, we propose a method for approximating factors of a Kronecker product, real or complex. Experimental data from a MIMO testbed is used to estimate the channel impulse response (CIR) and quantitatively estimate the spatial and inter-tap correlation matrices.

This paper is organized as follows. Section II reviews the discrete-time MIMO triply selective fading channel model. Section III describes about the MIMO testbed and the experimental setup. Section IV details the channel estimation procedure, the correlation matrices estimation procedure, results and analysis. Finally, section V concludes the paper.

2 DISCRETE-TIME TRIPLY SELECTIVE FADING MODEL

The input-output relationship of the MIMO channel in discrete-time is described in [1] as

$$\mathbf{y}(k) = \sum_{q=-Q_1}^{Q_2} \mathbf{H}(k, q) \cdot \mathbf{x}(k - q) + \mathbf{v}(k) \quad (1)$$

where k is the time index, Q_1 and Q_2 are non-negative integers representing the range of delay taps yielding the total channel length $Q = Q_1 + Q_2 + 1$, $\mathbf{x}(k) = [x_1(k), x_2(k), \dots, x_P(k)]^t$ is the transmitted signal vector, $\mathbf{y}(k) = [y_1(k), y_2(k), \dots, y_O(k)]^t$ is the received vector and

$\mathbf{v}(k) = [v_1(k), v_2(k), \dots, v_O(k)]^t$ is the additive white gaussian noise. The superscript $(.)^t$ notation represents the matrix transpose operator.

The MIMO channel coefficient matrix $\mathbf{H}(k, q)$ at time instant k and delay tap q is defined by

$$\mathbf{H}(k, q) = \begin{pmatrix} h_{1,1}(k, q) & \cdots & h_{1,P}(k, q) \\ \vdots & \ddots & \vdots \\ h_{O,1}(k, q) & \cdots & h_{O,P}(k, q) \end{pmatrix} \quad (2)$$

We reshape the matrix $\mathbf{H}(k, q)$ to an $(OPQ) \times 1$ coefficient vector as

$$\mathbf{h}_{vec}(k) = [\mathbf{h}_{1,1}(k), \dots, \mathbf{h}_{1,P}(k) \mid \dots \mid \mathbf{h}_{O,1}(k), \dots, \mathbf{h}_{O,P}(k)]^t \quad (3)$$

where $\mathbf{h}_{o,p}(k)$ is the coefficient vector of the (o, p) -th sub-channel given by $\mathbf{h}_{o,p}(k) = [h_{o,p}(-Q_1, k), \dots, h_{o,p}(Q_2, k)]$.

It is stated in [1] that the stochastic fading channel coefficient vector, $\mathbf{h}_{vec}(k)$, is zero-mean gaussian distributed and its covariance matrix, \mathbf{R} is given by

$$\begin{aligned} \mathbf{R} &= E[\mathbf{h}_{vec}(k_1) \cdot \mathbf{h}_{vec}^H(k_2)] \\ &= (\mathbf{\Psi}_{Rx} \otimes \mathbf{\Psi}_{Tx} \otimes \mathbf{\Psi}_{Tap}) \cdot J_0[2\pi f_d(k_1 - k_2)T_s] \end{aligned} \quad (4)$$

where $(.)^H$ denotes the Hermitian operator, \otimes denotes the Kronecker product, $\mathbf{\Psi}_{Rx}$ and $\mathbf{\Psi}_{Tx}$ are the spatial correlation matrices at the receiver and transmitter, respectively, and $\mathbf{\Psi}_{Tap}$ is the intertap covariance matrix. These matrices are defined in (5), (6) and (7). The factor $J_0[2\pi f_d(k_1 - k_2)T_s]$ describes temporal correlation where f_d is the maximum doppler frequency and T_s is the sampling period. $J_0(\cdot)$ is the zeroth order Bessel function.

$$\Psi_{Rx} = \begin{pmatrix} \rho_{Rx}(1,1) & \cdots & \rho_{Rx}(1,O) \\ \vdots & \ddots & \vdots \\ \rho_{Rx}(O,1) & \cdots & \rho_{Rx}(O,O) \end{pmatrix} \quad (5)$$

$$\Psi_{Tx} = \begin{pmatrix} \rho_{Tx}(1,1) & \cdots & \rho_{Tx}(1,P) \\ \vdots & \ddots & \vdots \\ \rho_{Tx}(P,1) & \cdots & \rho_{Tx}(P,P) \end{pmatrix} \quad (6)$$

$$\Psi_{Tap} = \begin{pmatrix} \psi(-Q1, -Q1) & \cdots & \psi(-Q1, Q2) \\ \vdots & \ddots & \vdots \\ \psi(Q2, -Q1) & \cdots & \psi(Q2, Q2) \end{pmatrix} \quad (7)$$

where $\rho_{Rx}(m, p)$ is the receive correlation coefficient between antennas m and p . Similarly, $\rho_{Tx}(n, q)$ is the transmit correlation coefficient between n and q transmit antennas. The elements of intertap covariance matrix is determined according to the power delay profiles.

This paper focuses on the validation of the triply selective fading channel model using (4) through the estimation of the spatial correlation matrices and the intertap covariance matrix.

3 TESTBED AND EXPERIMENT

A 2×2 MIMO-OFDM testbed has been developed using Altera Stratix III EP3SL150F field-programmable gate array (FPGA) DSP development kit. The discrete-time MIMO

triply selective fading channel model in [1] is the basis for the design of this testbed. Hardware implementation of discrete-time MIMO triply selective fading channel emulators is proposed in [8].

In the transmitter side, two independent data streams are generated in Stratix III development kit. The outputs are then up-converted to 17.5 MHz of IF (Intermediate Frequency) and then the signals are fed into RF (Radio Frequency) block, RF2-3000UCV1, to be transmitted at 915 MHz. MPA-10-40 is used for power amplification. Devices AFG3252 and FS725 are clock sources for all other devices. The setup architecture of the transmitter is shown in Fig. 1.

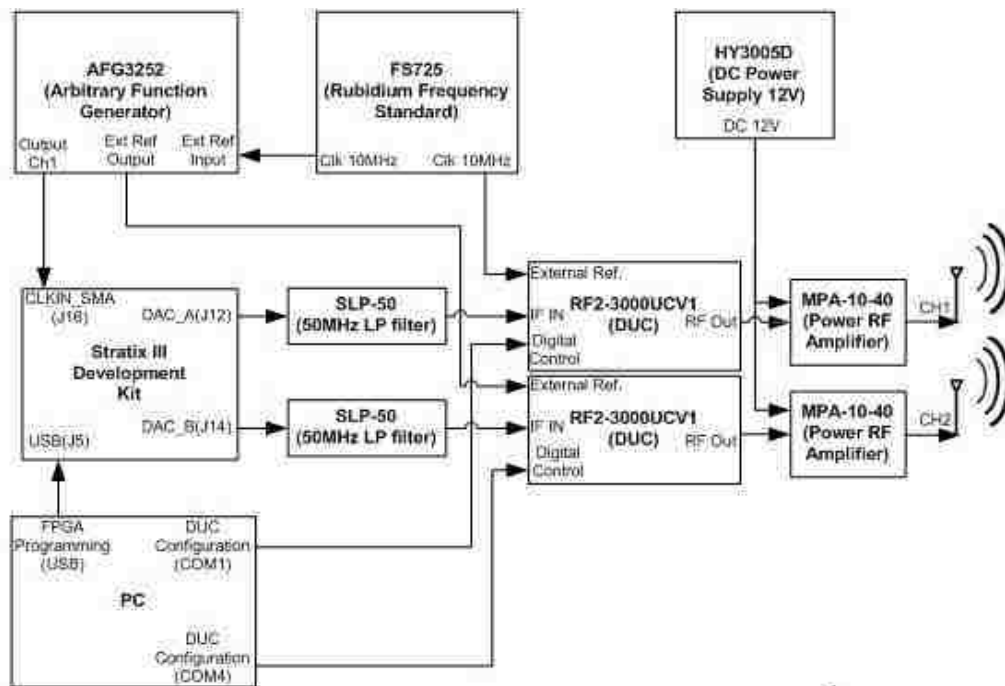


Figure 1. Transmitter setup architecture.

In the receiver side, RF signals are first down-converted to 70 MHz IF by the down-converter block, RF200-2500RV1. Baseband data streams are then generated and recorded in Stratix III development kit and transferred to PC. Devices AFG3252 and FS725 provide clock sources. The receiver setup architecture is shown in Fig. 2.

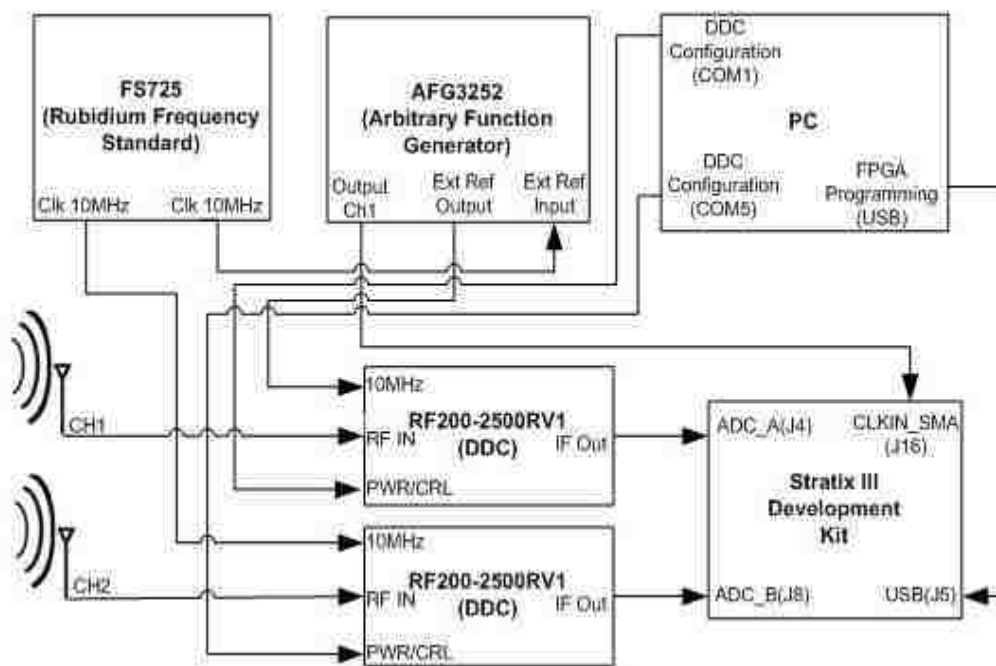


Figure 2. Receiver setup architecture.

A bandwidth configuration of 3.90 MHz is used in this testbed. The number of OFDM subcarriers is 256 and a cyclic prefix length of 64 samples is used. Experiment has been carried out using three different modulation schemes viz. QPSK, 8PSK and 16QAM. These subcarriers are used for channel sounding. Although BPSK is sufficient for channel sounding, the transceiver was originally designed for MIMO communications, thus QPSK, 8PSK and 16QAM modulation schemes are used. Measurements are done for two different experimental setups - one with both transmitter and receiver located in the same room

(inside 208) and the other with transmitter and receiver located in two different rooms (208 and 212) across a hallway as shown in the floor plan in Fig. 3.

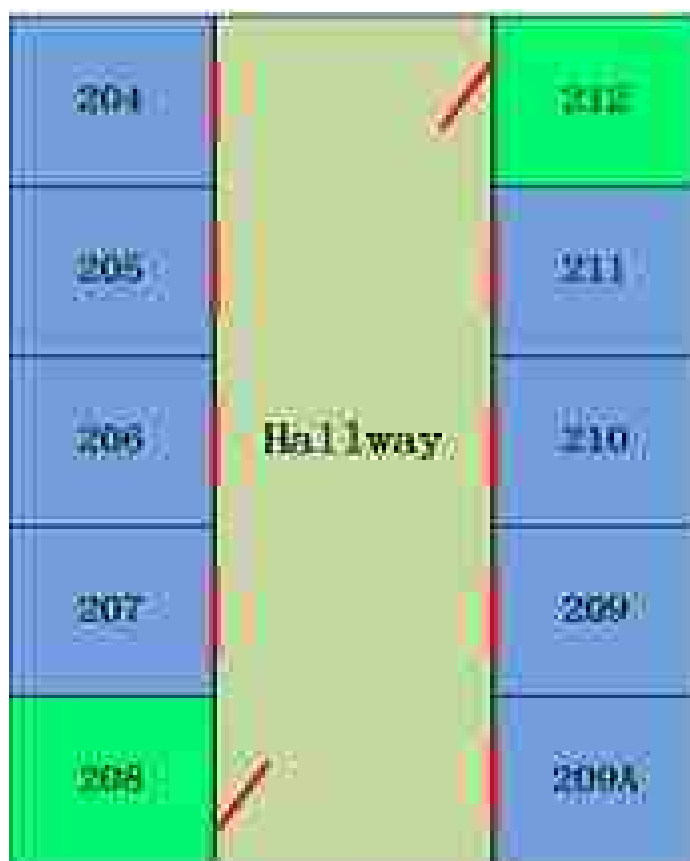


Figure 3. Floorplan used for the experiment.

Experimental data from this testbed is used for channel estimation and subsequent analysis.

4 PROCEDURE, RESULTS AND ANALYSIS

4.1 Channel Estimation

Time domain least squares (LS) method is used for estimation of channel impulse response (CIR) for each subchannel of the 2×2 MIMO system based on the known training sequence and received sequence. The LS estimation, detailed in chapter 8 in [9], is obtained as

$$\hat{\mathbf{h}}_{LS} = (\mathbf{X}^H \mathbf{X})^{-1} \mathbf{X}^H \mathbf{y} \quad (8)$$

where $(\cdot)^H$ and $(\cdot)^{-1}$ represent the hermitian and inverse operations respectively, \mathbf{X} is the circulant training sequence matrix and \mathbf{y} is the received sequence. The matrix \mathbf{X} is formed as

$$\mathbf{X} = \begin{pmatrix} x_Q & \cdots & x_1 & x_0 \\ x_{Q+1} & \cdots & x_2 & x_1 \\ \vdots & \ddots & \vdots & \\ x_{Q+P-1} & \cdots & x_P & x_{P-1} \end{pmatrix} \quad (9)$$

where Q is the number of channel taps and P is the number of pilot data for each antenna.

A long probing sequence is transmitted and CIRs are estimated progressively by using cascading windows of size $N_p = 120$ symbols. An example of 30-tap CIRs of the four subchannels is shown in Fig. 4 where 80 cascading windows are used across the length of the transmitted data sequence. Although the signal bandwidth is only 3.9 MHz, the baseband equivalent channel did experience multipath delay spread spanning 30 taps. This is because both transmitter and receiver antennas were placed very low, only a meter above the floor. This is different from the case where one end is placed very high like a base

station where multipath may not be significant. This demonstrates the difference between mobile-to-mobile channel and base-station to mobile channel. Number of windows can be increased or overlapping windows can be used for the estimation of CIRs of highly scattering channels such as underwater acoustic channel.

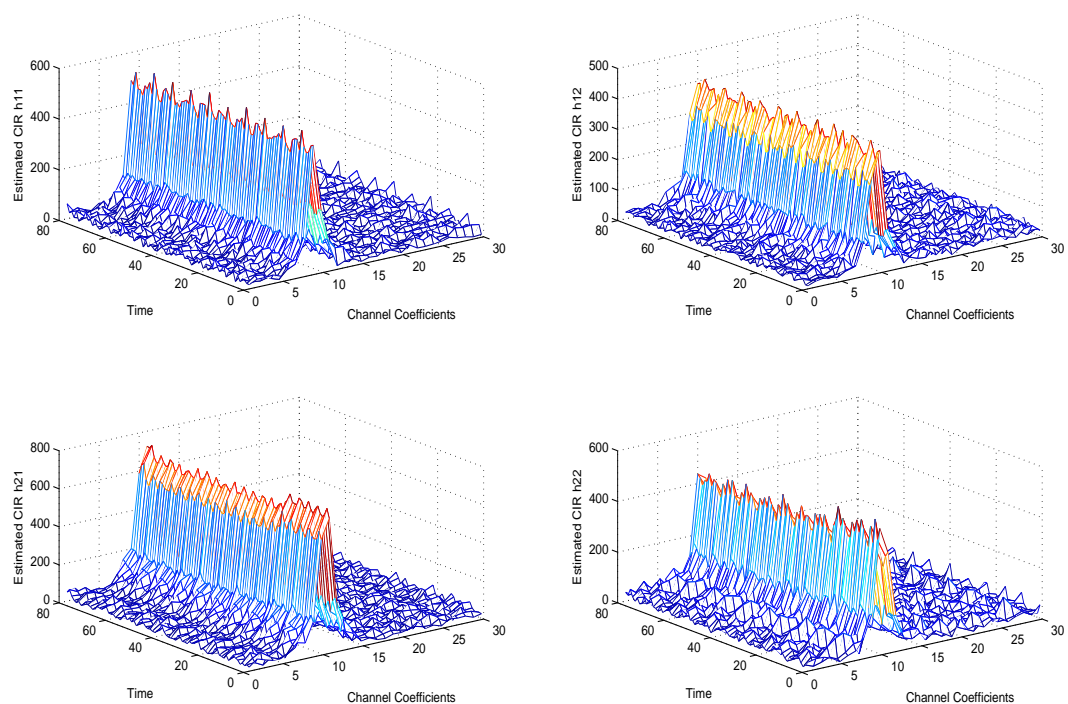


Figure 4. Magnitudes of channel impulse responses for four subchannels.

4.2 Estimation of the Channel Coefficient Covariance Matrix

The channel coefficient covariance matrix is calculated using the estimated channel coefficients. The $(OPQ \times OPQ)$ covariance matrix, \mathbf{R} , is calculated using (4). Estimated channel coefficient covariance matrix is shown in Fig. 5.

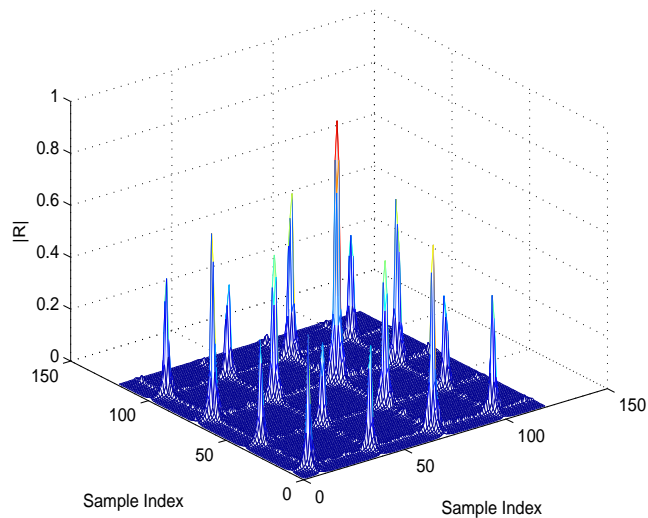


Figure 5. Magnitude of estimated channel coefficient covariance matrix.

4.3 Decomposition of the Kronecker Product

The Kronecker product of two matrices \mathbf{A} and \mathbf{B} is defined as

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \end{pmatrix} \quad (10)$$

where \mathbf{A} is $(m \times n)$, \mathbf{B} is $(p \times q)$ matrix, and \mathbf{C} , the resultant Kronecker product is of size $(mp \times nq)$.

The problem at hand is to find estimations of \mathbf{A} and \mathbf{B} from a given Kronecker product \mathbf{C} . Let us consider the first block of elements of matrix \mathbf{C} , say \mathbf{C}_{11} which is a $(p \times q)$ matrix given by

$$\mathbf{C}_{11} = a_{11}\mathbf{B} \quad (11)$$

If we calculate an ensemble average of all the elements of \mathbf{C}_{11} , that essentially results in scalar multiplication of a_{11} and mean of all the elements of \mathbf{B} as shown in 12

$$E[\mathbf{C}_{11}] = a_{11}E[\mathbf{B}] \quad (12)$$

This isolates the first element of \mathbf{A} from the Kronecker product. We repeat the same process to obtain other elements of \mathbf{A} . The resulting estimation of matrix \mathbf{A} , therefore, is a scaled version of actual \mathbf{A} and retains its spatial properties.

In this paper, we estimate the spatial correlation matrix Ψ_{Tx} from the channel coefficient covariance matrix \mathbf{R} using the method explained in (12).

4.4 Estimation of Intertap Covariance Matrix and Spatial Correlation Matrix

We estimate the $(Q \times Q)$ intertap covariance matrices for each subchannel. Using correlation matrix distance (CMD) as a metric [10], we show that these intertap covariance matrices have identical spatial structure. CMD, the distance between two correlation matrices $\mathbf{R1}$ and $\mathbf{R2}$ is defined as

$$d_{corr}(\mathbf{R1}, \mathbf{R2}) = 1 - \frac{tr\{\mathbf{R1R2}\}}{\|\mathbf{R1}\|_f \|\mathbf{R2}\|_f} \quad (13)$$

where $tr\{\cdot\}$ represents the trace of the matrix and $\|\cdot\|_f$ is the Frobenius norm. CMD becomes zero if the correlation matrices are equal up to a scaling factor and one if they differ from each other. The smaller value, thus, verifies that the matrices are spatially identical. Results are summarized in Table 1 for data obtained from different experimental setups.

These results comply with the assumption in [3] that the power delay profile of the physical channel model is identical for all transmit and receive antenna indices. We compute an average intertap covariance matrix and use it as one of the Kronecker factors of the channel coefficient covariance matrix to estimate the spatial correlation

Table 1. Comparison of correlation matrices using CMD.

Experi. Setup	Atten. (dB)	CMD R_{11}, R_{12}	CMD R_{11}, R_{21}	CMD R_{11}, R_{22}	CMD R_{21}, R_{12}	CMD R_{21}, R_{22}	CMD R, R_{verify}
In 208	22	0.0259	0.0129	0.0959	0.0259	0.0713	0.0196
	26	0.0255	0.0169	0.5789	0.0255	0.0744	0.0966
	30	0.0159	0.0192	0.3133	0.0159	0.0647	0.0465
208 and 212	2	0.2550	0.0822	0.1003	0.2550	0.0908	0.0942
	6	0.0751	0.0230	0.0785	0.0751	0.0569	0.0355
	10	0.1208	0.0244	0.0690	0.1208	0.1433	0.0403

matrix. The spatial similarity among the intertap covariance matrices of four sub-channels can be observed in Fig. 6. Fig. 7 shows the average intertap covariance matrix, $\Psi_{T_{ap}}$.

The elements of spatial correlation matrix are estimated from the channel coefficient covariance matrix \mathbf{R} . The process in (12) gives a matrix spatially identical with $\Psi_{T_{rx}}$. We again calculate the Kronecker product of the estimated spatial correlation matrix, $\Psi_{T_{rx}}$ and the average intertap covariance matrix, $\Psi_{T_{ap}}$ using

$$\mathbf{R}_{verify} = \Psi_{T_{rx}} \otimes \Psi_{T_{ap}} \quad (14)$$

to validate the approach used for the decomposition of Kronecker product. Since the transmitter-receiver setup in this experiment was static, the temporal correlation does not have a significant impact on the results. The CMD metric is used to compare the similarity between the channel coefficient covariance matrix calculated using (11) and (14). Results for 6 different instances are shown in Table 1. The matrices \mathbf{R}_{ij} are the correlation matrices of the ij -th subchannel. Fig.8 shows the channel coefficient covariance matrix estimated using (14). $\Psi_{T_{rx}}$, the (4×4) spatial correlation matrix, is itself a Kronecker product of Ψ_{R_x} and Ψ_{T_x} .

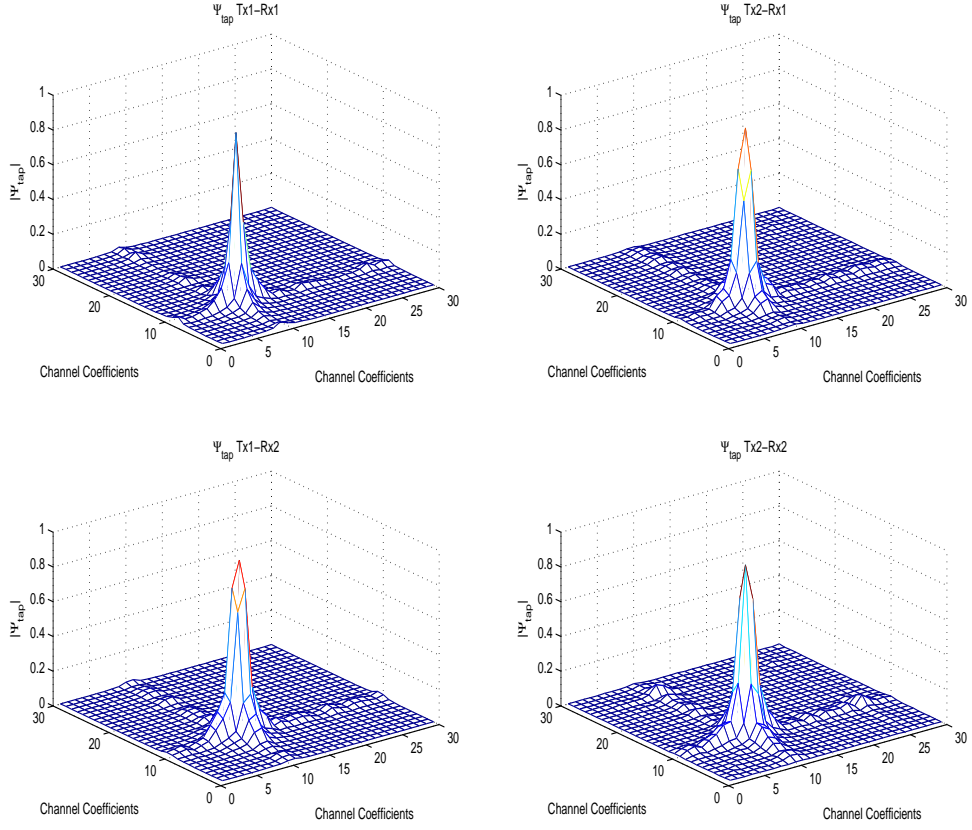


Figure 6. Magnitudes of intertap covariance matrices for each subchannel.

5 CONCLUSIONS

In this paper, we validated the triply selective fading channel model through a MIMO testbed and experimental results. Experimental results demonstrate that the discrete-time triply selective fading channel can be expressed as separable temporal, inter-tap and spatial correlations. Using correlation matrix distance as a metric we show that the intertap correlations for all the subchannels are spatially identical. This permits the estimation of spatial correlations matrices through the decomposition of channel coefficient covariance matrix. Finally, we verify our results by recalculating

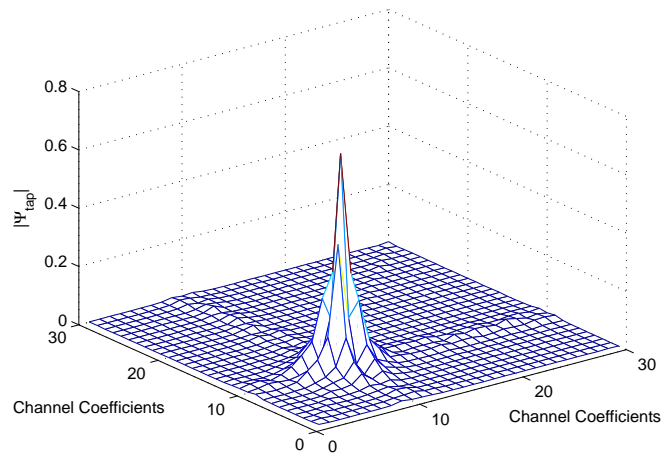


Figure 7. Magnitude of averaged intertap covariance matrix, Ψ_{Tap} .

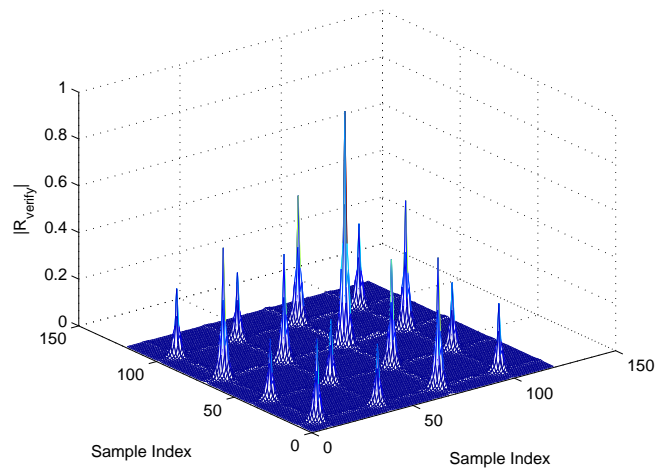


Figure 8. Kronecker product of estimated Ψ_{Trx} and Ψ_{Tap} .

the Kronecker product of the estimated correlation matrices and comparing the result with the covariance matrix obtained directly from the estimated channel coefficients.

6 REFERENCES

- [1] 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.
- [2] J. Mietzner and P. Hoeher, "A rigorous analysis of the statistical properties of the discrete-time triply-selective MIMO Rayleigh fading channel model," *IEEE Trans. Wireless Commun.*, vol. 6, no. 12, pp. 4199-4203, 2007.
- [3] J. Mietzner, C. Xiao, P. Hoeher, and K. Ben Letaief, "A note on discrete-time triply-selective MIMO Rayleigh fading channel models," *IEEE Trans. Wireless Commun.*, vol. 7, no. 3, pp. 837, 2008.
- [4] A. Abdi and M. Kaveh, "A space-time correlation model for multielement antenna systems in mobile fading channels," *IEEE Journal. Selected Areas in Communications*, vol. 20, no. 3, pp. 550-560, Apr. 2002.
- [5] 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.
- [6] G. Byers and F. Takawira, "Spatially and temporally correlated MIMO channels: modeling and capacity analysis," *IEEE Trans. Veh. Technol.*, vol. 53, no. 3, pp. 634-643, May 2004.
- [7] C. V. Loan and N. Pitsianis, *Approximation with Kronecker products*, Kluwer Publications, 1993.
- [8] 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. 2542C2551, Nov. 2010.
- [9] S. Kay, "Fundamentals of Statistical Signal Processing: Estimation Theory," Online: <http://books.google.com/books?id=aFwESQAACAAJ>, Mar. 2010.
- [10] M. Herdin, N. Czink, H. Ozelik, and E. Bonek, "Correlation matrix distance, a meaningful measure for evaluation of nonstationary mimo channels," in *Proc. Vehicular Technology Conf.*, Stockholm, Sweden, May 2005, pp. 136-140.

SECTION

2 CONCLUSIONS

This dissertation focuses on research of hardware-based wireless fading channel emulators. It solves the following main challenges in hardware implementations of wireless fading channel emulators. The hardware implementation methods of triple-selective fading channel emulators with accurate correlation properties are proposed. On-chip FRFGs and inter-tap correlation matrix generators are implemented. A mixed P-S computational structure, which incorporates three types of correlation into sub-channels, is proposed to make the best tradeoff between processing speed and hardware usage. These proposed algorithms and designs have been simulated not only by simulation tools, but also implemented and verified on FPGA development kit platforms. The emulated channels reach excellent statistical and correlation properties, which match those theoretical ones. This dissertation also validates the triply selective fading channel model through a MIMO testbed and experimental results. Experimental results demonstrate that the discrete-time triply selective fading channel can be expressed as separable temporal, inter-tap and spatial correlations.

The contributions of all my research work during Ph.D. study are summarized in two journal papers and five conference papers, among which, two journal papers and three conference papers are included in this dissertation. The complete publication list is included in Section 3.

3 PUBLICATIONS

- [1] F. Ren, and Y.R. Zheng, "Hardware Emulation of Wideband Correlated Multiple-Input Multiple-Output Fading Channels," *Journal of Signal Processing Systems*, accepted to publish, Jun. 2011.
- [2] F. Ren, and Y.R. Zheng, "A Novel Emulator for Discrete-time MIMO Triply-selective Fading Channels," *IEEE Trans. Circuits and Systems I: Regular Paper*, vol. 57, no. 9, pp. 2542-2551, Sep. 2010.
- [3] F. Ren, and Y.R. Zheng, "Incorporating Correlation Matrices into Hardware Triple Selective Fading Channel Emulators Using Kronecker Product," in *Proc. Vehicular Technology Conference (VTC2010-Fall)*, Ottawa, Canada, Sep. 2010, pp. 1-5.
- [4] F. Ren, and Y.R. Zheng, "Hardware Implementation of Triply Selective Rayleigh Fading Channel Simulators," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Dallas, Texas, Mar. 2010, pp. 1498-1501.
- [5] 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 (ISCAS)*, Taipei, Taiwan, May 2009, pp. 1759-1762.
- [6] F. Ren, and Y. R. Zheng, and J. Sarangapani, "Incorporating Forward Error Correction Codes into FlexRay Communications," in *Proc. Sensor, Signal and Information Processing (SenSIP)*, Sedona, AZ, May, 2008.
- [7] S. Subedi, H. Lou, F. Ren, M. Wang, and Y. R. Zheng, "Validation of the Triply Selective Fading Channel Model Through a MIMO Test Bed and Experimental Results," *Proc. International Conference for Military Communications (MIL-COM11)*, accepted to publish.

VITA

Fei Ren was born May 8, 1983, in Chongqing, China. He received his B.S. degree in 2005 in Electrical Engineering from University of Electronic Science and Technology of China, Chengdu, Sichuan Province, China. He received his M.S. degree in 2007 in Electrical Engineering from University of Missouri-Rolla, Rolla, MO, USA. He began his Ph.D. study in January 2008, in the Department of Electrical and Computer Engineering at Missouri University of Science and Technology. His research interests include wireless channel emulators, VLSI, and hardware implementation of wireless communication systems. He is expected to receive his Ph.D. degree in Electrical Engineering from Missouri University of Science and Technology in December 2011.