

A SYSTEM ARCHITECTURE FOR REAL-TIME MULTI-PATH MIMO FADING CHANNEL EMULATION

Elliot Briggs, Tanja Karp, Brian Nutter * †

Texas Tech University
Electrical and Computer Engineering
Lubbock, Texas, USA 79410
elliott.s.briggs@ttu.edu,
tanja.karp@ttu.edu, brian.nutter@ttu.edu

Dan McLane

Chirp, Inc.
21900 Burbank Blvd., Suite 200
Woodland Hills, CA 91367
dmclane@chirpdata.com

ABSTRACT

Creating a flexible, programmable multi-path fading channel emulator in hardware presents many design and implementation challenges. Past work has focused on creating a stochastic Jakes process generator as well as hardware efficient implementations of arbitrary upsampling. Significant advances in the design of the arbitrary ratio upsampler have been made, which has greatly reduced the hardware complexity and increased the system's performance, allowing the design to support MIMO emulation for 2 transmit antennas in a single FPGA device. Also building on previous work, the Kronecker model is introduced, allowing the user to vary the spatial correlation features in the emulated channel. This paper will quickly summarize the theoretical aspect of MIMO channel emulation. The main focus will be given to architectural and implementation details, following the system design process from end to end. A high level of attention will be given to multi-rate signal processing design aspects that minimize computational complexity. The results section highlights metrics from both simulation and FPGA hardware integration.

Index Terms— MIMO, Channel Emulation, Jakes, Kronecker, FPGA, ASIC, Multi-Rate, Arbitrary Resampling

1. INTRODUCTION

The time-varying nature of a SISO channel can be modeled by the Jakes process [1, 2, 3], which models the temporal statistics of a rich scattering channel observed by a mobile device. Numerical and hardware implementations of Jakes process generators exist throughout the literature, often only focusing on approximating the autocorrelation properties of

the Jakes process using various techniques, i.e. [4, 5]. Oftentimes in the literature, implemented architectures neglect practical, fundamental system-level details and overlook the possibility for drastic optimizations and computational complexity reduction, i.e. [6]. This work will highlight some of the possible optimizations that arise when the a full channel emulation architecture is considered.

This work will illustrate the implementation of the Jakes Doppler model used to generate the desired temporal statistics of the channel. The extension to a multi-antenna (MIMO) system requires the introduction of spatial correlation between the antenna arrays in the system, which can be applied using the Kronecker model [7, 8]. The Kronecker model is a simplistic model that is used to correlate a generated set of i.i.d. Jakes processes according to the channel's properties. By combining the Jakes and Kronecker models to achieve programmable temporal and spatial correlation, a MIMO channel emulator can be designed.

First, the generic MIMO channel will be introduced, providing a conceptual framework and establishing notation. Next, the Jakes Doppler model and the Kronecker model will be briefly introduced and discussed. Having introduced the two channel models, a system architecture is presented, from which detailed system-level implementation aspects are highlighted. The system architecture heavily leverages beneficial aspects of multi-rate signal processing techniques to produce a very computationally efficient architecture, targeted for FPGA/ASIC implementation. Finally, system-level simulation results and FPGA integration details are displayed.

2. THE GENERIC MIMO CHANNEL

To establish a conceptual framework, first a generic MIMO system is described. The number of transmit and receive antennas in the MIMO system are indicated by M and N respectively. The $M \times 1$ transmitted symbol vector \mathbf{x} passes through the channel, modeled by the multiplication of \mathbf{x} with the $N \times M$ matrix of complex coefficients \mathbf{H} to form the re-

*This work has been possible under continuous support of Innovative Integration (www.innovative-dsp.com)

†This material is based in part upon work supported by the Texas Higher Education Coordinating Boards Norman Hackerman Advanced Research Program under Project No. 003644-0041-2009.

ceived $N \times 1$ symbol vector \mathbf{y} .

$$\mathbf{y} = \mathbf{H}\mathbf{x}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,M} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N,1} & h_{N,2} & \cdots & h_{N,M} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix} \quad (1)$$

The multiplication between \mathbf{H} and \mathbf{x} model the passage of the N transmitted symbols traveling through $N \times M$ paths in the channel which are then summed accordingly at the receiver's N antennas.

3. APPLYING TEMPORAL CORRELATION: THE JAKES DOPPLER MODEL

Eq. 1 describes the generalized static MIMO system. A time index is added to Eq. 1 so that a continual transmission of symbols through the channel can be represented. Also, if an element of time is included with the channel matrix \mathbf{H} , the channel is allowed to become *time varying*. For simplicity, let the number of transmit and receive antennas $M = 2$ and $N = 2$ respectively and let t indicate the discrete time index in the system. Modifying Eq. 1 accordingly,

$$\mathbf{y}[t] = \mathbf{H}[t]\mathbf{x}[t]$$

$$\begin{bmatrix} y_1[t] \\ y_2[t] \end{bmatrix} = \begin{bmatrix} h_{1,1}[t] & h_{1,2}[t] \\ h_{2,1}[t] & h_{2,2}[t] \end{bmatrix} \begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix}. \quad (2)$$

The time-varying nature of each complex path gain in $\mathbf{H}[t]$ can be modeled by the Stochastic Jakes Doppler process [1, 2, 3]. The Jakes process is one of many stochastic processes that is used to model mobile wireless channel environments. The Jakes process requires two parameters, the carrier wavelength and the mobile's velocity. The ratio of these parameters defines the maximum Doppler frequency f_{max} .

$$f_{max} = \frac{v}{\lambda} \quad (3)$$

To define the Jakes process, the channel coefficients must be decomposed into their real and imaginary components.

$$\hat{j}_{n,m}[t] = \mu_0[t] + j\mu_1[t] \quad (4)$$

In the Jakes process, the continuous-time cross correlation between the real and imaginary components of $\hat{j}_{n,m}$, μ_0 and μ_1 respectively is defined by

$$r_{\mu_0\mu_1}(\tau) = 0, \forall \tau, \quad (5)$$

and the continuous-time autocorrelation of each real and imaginary component is defined by

$$r_{\mu\mu}(\tau) = J_0(2\pi f_{max}\tau), \forall \tau. \quad (6)$$

Taking the Fourier transform of Eq. 6 defines the continuous-frequency Jakes' power spectral density function.

$$S_{\mu\mu}(f) = \begin{cases} \frac{1}{\pi f_{max}\sqrt{1-(f/f_{max})^2}}, & |f| \leq f_{max} \\ 0, & |f| > f_{max} \end{cases} \quad (7)$$

$$f \in (-\infty, \infty)$$

4. APPLYING SPATIAL CORRELATION: THE KRONECKER MODEL

The independent, time-varying complex channel coefficients $\hat{j}_{m,n}[t]$ are generated to have the temporal statistical properties defined by the Jakes process. The spatial features in a MIMO channel introduce an additional dimension of correlation. Intuitively, if the antennas at the transmitter and receiver are closely spaced, the channel conditions between each antenna pair are likely to be correlated.

The spatial statistics resulting from antenna spacing, radiation pattern, polarization, and the local scattering environment can be lumped into a correlation matrix for each transmit and receive antenna arrays. The correlation matrices for the transmitter and receiver antenna arrays are defined by

$$\mathbf{R}_{TX} = \begin{bmatrix} \rho_{1,1}^{TX} & \rho_{1,2}^{TX} & \cdots & \rho_{1,M}^{TX} \\ \rho_{2,1}^{TX} & \rho_{2,2}^{TX} & \cdots & \rho_{2,M}^{TX} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{M,1}^{TX} & \rho_{M,2}^{TX} & \cdots & \rho_{M,M}^{TX} \end{bmatrix} \quad (8)$$

$$\mathbf{R}_{RX} = \begin{bmatrix} \rho_{1,1}^{RX} & \rho_{1,2}^{RX} & \cdots & \rho_{1,N}^{RX} \\ \rho_{2,1}^{RX} & \rho_{2,2}^{RX} & \cdots & \rho_{2,N}^{RX} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{N,1}^{RX} & \rho_{N,2}^{RX} & \cdots & \rho_{N,N}^{RX} \end{bmatrix},$$

where $\rho_{m,n}^{TX}$ and $\rho_{m,n}^{RX}$ indicate the correlation coefficients between the m^{th} and n^{th} antenna at the transmit and receive antenna arrays, respectively [7, 8]. In wireless standards, such as the 3GPP long-term evolution (LTE) standard, several of these matrices are defined for varying scenarios [9]. The Kronecker model attempts to model the spatial correlation between the transmitter and receiver by defining an $NM \times NM$ correlation matrix of the MIMO channel, generated by performing the Kronecker product between the two matrices defined in Eq. 8.

$$\mathbf{R}_{MIMO} = \mathbf{R}_{TX} \otimes \mathbf{R}_{RX}, \quad (9)$$

where \otimes indicates the Kronecker product operator.

Using $M = 2$ and $N = 2$ as an example, the desired spatial correlation properties are generated using \mathbf{R}_{MIMO} to obtain $\mathbf{H}[t]$ from $\mathbf{J}[t]$, the matrix of i.i.d. Jakes processes.

$$\text{Vec}\{\mathbf{H}[t]\} = \mathbf{C}\text{Vec}\{\mathbf{J}[t]\}$$

$$\begin{bmatrix} h_{1,1}[t] \\ h_{2,1}[t] \\ h_{1,2}[t] \\ h_{2,2}[t] \end{bmatrix} = \begin{bmatrix} c_{1,1} & 0 & 0 & 0 \\ c_{2,1} & c_{2,2} & 0 & 0 \\ c_{3,1} & c_{3,2} & c_{3,3} & 0 \\ c_{4,1} & c_{4,2} & c_{4,3} & c_{4,4} \end{bmatrix} \begin{bmatrix} j_{1,1}[t] \\ j_{2,1}[t] \\ j_{1,2}[t] \\ j_{2,2}[t] \end{bmatrix}, \quad (10)$$

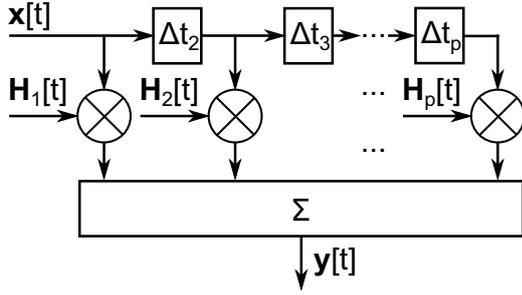


Fig. 1. Tapped Delay Line Model

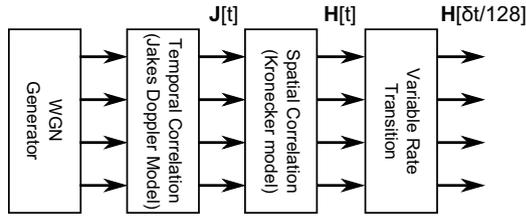


Fig. 2. Channel Matrix Generator

where the lower-diagonal \mathbf{C} matrix is obtained by performing the Cholesky decomposition of \mathbf{R}_{MIMO} .

$$\mathbf{R}_{MIMO} = \mathbf{C}\mathbf{C}^T \quad (11)$$

5. GENERAL IMPLEMENTATION ARCHITECTURE

In addition to spatial and temporal correlation, a wireless MIMO channel typically has multiple paths or echoes. To keep the model simple, assume that each element in the channel matrix has the same $P \times 1$ delay vector $\boldsymbol{\tau} = [\tau_1 = 0, \tau_2, \dots, \tau_P]^T$. Using the $\boldsymbol{\tau}$ vector along with P independent \mathbf{H} matrices, the multi-path MIMO channel can be mathematically described by

$$\mathbf{A}(t, \boldsymbol{\tau}) = \sum_{p=1}^P \mathbf{H}_p[t] \delta(\boldsymbol{\tau} - \boldsymbol{\tau}_p) . \quad (12)$$

Graphically, the system can be described by the tapped-delay line model, illustrated in Fig. 1. This indicates the overall architecture of a multi-path MIMO channel emulator. The tapped delay-line structure that utilizes the P channel matrices will be referred to as the “channel processing” component.

The graphical representation of each \mathbf{H} matrix generator is shown in Fig. 2, which includes the additional “variable rate transition” component. This component performs arbitrary-ratio upsampling for each of the elements in \mathbf{H} so the user can vary the Doppler by programming the desired upsampling factor [10].

6. MULTI-RATE SYSTEM DESIGN

Modern wireless standards are requiring increasingly higher sampling rates, thus wider bandwidths to be processed, specifically LTE-Advanced (release 10), requiring up to 100 MHz of occupied channel bandwidth [11]. As the carrier wavelengths and mobile velocities remain constant, the required rate of the Jakes process generation and the Kronecker spatial correlation components remains constant. To bridge the rate disparity between the high operating rate of channel processing and the low-rate Jakes and Kronecker generators, an arbitrary-rate upsampler is inserted at the end of the processing chain. Assuming the channel processing component operates at a fixed rate, changing the upsampling factor in the variable-rate component will change the rate at which the Jakes and Kronecker components operate, thus allowing the Doppler frequency to be varied!

The rate disparity between the two sections of the channel matrix generator architecture is exploited to form a very computationally efficient implementation. Previous work will be improved [10] and the necessary extensions to support MIMO will be added.

6.1. Jakes Process Generation

As in [2, 3], an FIR filter will be used to process white Gaussian noise to generate each Jakes process. The FIR coefficient set is generated by sampling the square-root Jakes power-spectral density given by Eq. 7. Note that the equality condition in Eq. 7 must be removed to avoid possible divisions by zero.

$$\begin{aligned} H(f_k) &= \sqrt{S_{\mu\mu}(f_k, f_d)}, \\ f_k &= \frac{k-1}{N_{Jakes}} - \frac{1}{2}, \\ k &= 1, 2, \dots, N_{Jakes}, \\ f_d &= \frac{2f_{max}}{f_s}, \end{aligned} \quad (13)$$

where N_{Jakes} indicates the number of sampling points and f_s denotes the operating rate of the filter. For a running design example, $f_d = .8$ and $N_{Jakes} = 256$ will be selected. Performing the discrete Fourier transform of $H(f_k)$ yields the sampled Jakes autocorrelation function. Applying a Kaiser window to the result yields the desired impulse response of the FIR Jakes filter minimizing Gibbs phenomenon. Fig. 3 shows the generated Jakes FIR filter before and after applying the Kaiser window to show its effects.

A 2x2 system requires 4 complex Jakes processes, thus 8 real Jakes processes must be generated, requiring 8 Jakes FIR filters. If an absolute Doppler frequency $f_{max} = 100$ Hz is desired and $f_d = .8$, the required filter operating rate is $f_s = 250$ Hz, thus 250 8-element vectors per second must be produced. If the workload of each Jakes filter, defined by the

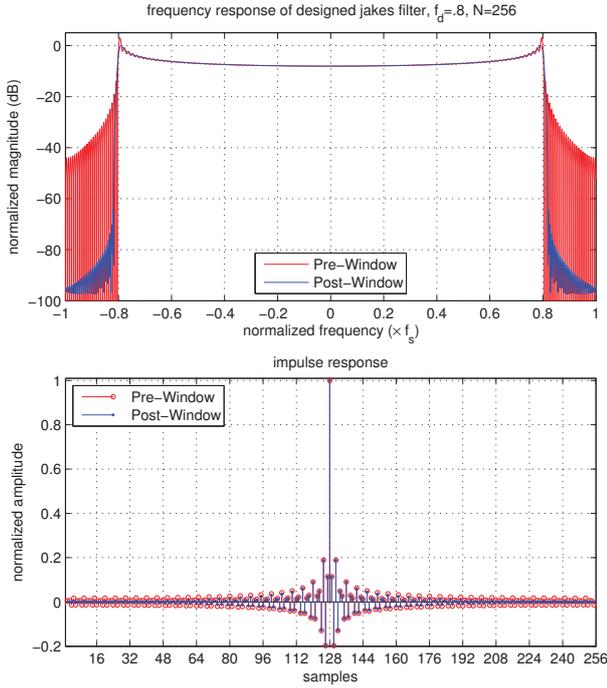


Fig. 3. Jakes FIR Filter

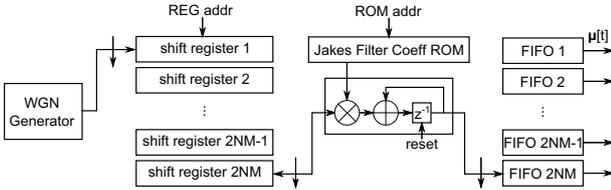


Fig. 4. A Single MACC, $2NM$ Channel Jakes Generator

number of MACC operations required for each output sample, is $W_{Jakes} = 256$, only $8W_{Jakes}f_s = 512,000$ MACCs per second must be performed, a task easily performed using a single MACC element in an FPGA/ASIC implementation.

Additional optimizations can be made when one of the fundamental properties of WGN is exploited. Each sample in a WGN sequence is independent, and uncorrelated with the others, thus a single WGN generator can be time-multiplexed for use with many Jakes process generators! Depending on the maximum possible operating rate of the WGN generator, it is possible to use a single WGN generator across all of the channel matrix generation components!

Fig. 4 depicts the system diagram of an $2NM$ channel Jakes process generator using a single MACC element. The first commutator delivers samples to each $2NM$ shift registers, each containing N_{Jakes} elements. After the shift register is loaded with a new input sample, it is selected by an output commutator, providing access to all N_{Jakes} elements to the MACC component. The register address ROM addresses

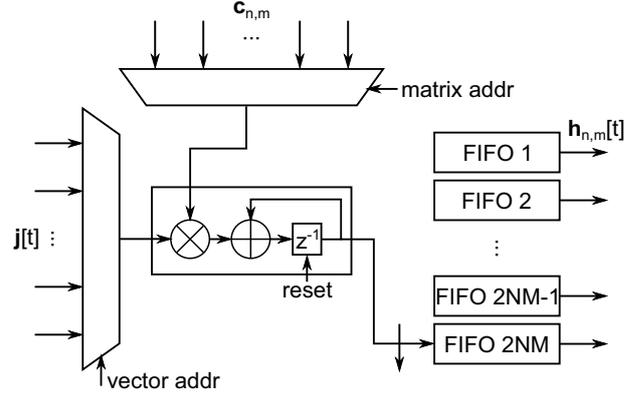


Fig. 5. Matrix-Vector Multiplication Component for the Kronecker Model

are then incremented accordingly, allowing the MACC to perform the necessary operations to generate a single output sample. The output sample is then loaded into its respective FIFO buffer, which are used to time-align each output of the MACC component, resulting in vectors of $\mu[t]$. The Jakes FIR filter is symmetric in nature, requiring only half of the coefficient set to be stored in the Jakes coefficient ROM.

6.2. Applying the Kronecker Model

After generating the $2NM$ Jakes processes, the Kronecker model must be applied to achieve the desired spatial correlation characteristics for the MIMO channel. The required mathematical operation follows from Eq. 10. For a 2×2 MIMO system, the Kronecker model's matrix operation requires 20 real multiplications and 16 real accumulate operations for each produced vector of complex Jakes processes. In the running example, this equates to 250 matrix-vector operations per second, requiring only 5,000 real multiplications and 4,000 real accumulates per second. These operations can be performed using a single MACC element as illustrated in Fig. 5.

6.3. Arbitrary-Rate Upsampling

The arbitrary-rate upsampling component is used to match the very low rate of the Jakes and Kronecker model elements with the potentially high rate channel processing component. The variable rate feature allows the Doppler frequency to be selected by the user. An attractive arbitrary resampler architecture that uses a three-staged approach, capable of bridging the rate disparity in the system has been selected [12, 13]. This architecture utilizes a pair of polyphase upsamplers working in conjunction with a linear interpolation stage to achieve arbitrary-ratio resampling while maintaining levels of dynamic range greater the self-imposed requirement of 16-bits (96 dB). With the assumption that only very large upsampling factors will be required, the mentioned resampling

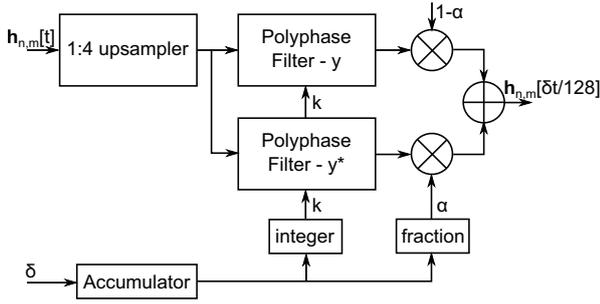


Fig. 6. 3-Stage Arbitrary Resampling Architecture

architecture can be simplified, enabling a highly resource-efficient implementation.

The architecture found in [12, 13] is shown in Fig. 6. To achieve the desired 96 dB dynamic range, the input signal must first be oversampled by a factor of 4, thus the dual polyphase structure is preceded by a 4x upsampling stage. Each of the dual polyphase filters are designed to nominally upsample by a factor of 32x, allowing the final linear interpolation stage to gain the largest jump in overall upsampling.

6.3.1. Cascaded Half-Band Polyphase Preprocessing Upsamplers

A computationally efficient (and resource efficient) 4x upsampling “preprocessing” stage can be implemented using a cascade of 2x upsampling half-band polyphase FIR filters. The half-band FIR filter is specially designed to have almost half of its set of coefficients occupied by zeros, thus the polyphase implementation has two arms, one of which collapses to a pure delay-line, eliminating the need for many MACC operations and dramatically reducing coefficient storage requirements. The overlaid frequency response of the two designed filters is shown in Fig. 7. The half-band filters in Fig. 7 require only 18 and 5 coefficients to be stored for the filters shown by the black and red curves respectively. The total workload of the filter, computed by the number of required MACC operations per output sample, is $W_{HB4x} = \frac{18+5}{4} = 5.75$. The upsampler must produce 4x the number of vectors per second as the Kronecker model component, therefore in the running example, 1,000 vectors per second, each containing 8 independent filter operations. The vector generation requires 8,000 outputs per second, thus 46,000 MACC operations per second. Each 2x upsampling component in the cascade can be implemented using the structure shown in Fig. 8. One arm of the polyphase half-band filter can be implemented as a pure delay line. Advantageously, the input shift register in Fig. 8 and the delay line hold the same set of samples during operation, thus the shift register implements the storage for the polyphase arm containing non-zero coefficients, and the delay-line for the

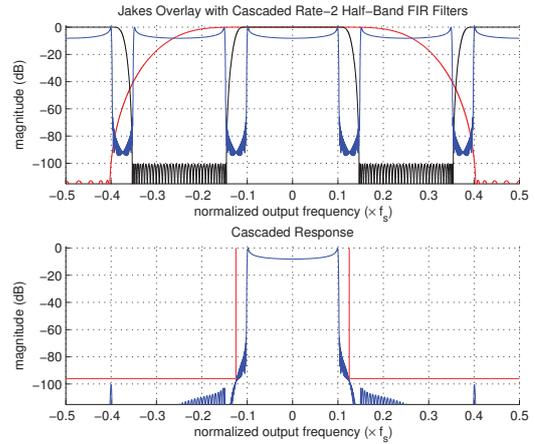


Fig. 7. Cascaded Response of Rate-2 Half-Band Polyphase Upsamplers

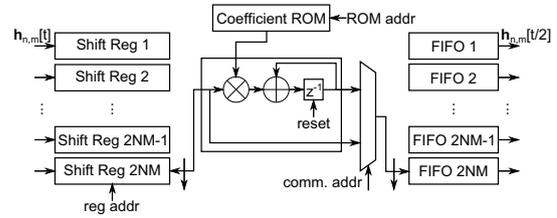


Fig. 8. Rate-2 Half-Band FIR Upsampling

other portion of the filter. A multiplexer is used as a commutator to select samples from the output of the MACC element or the last element in the input shift register.

6.3.2. Fixed-Rate 32x Polyphase Upsampler With Variable-Rate Linear Interpolator

The architecture taken from the literature and shown in Fig. 6 indicates the requirement of two polyphase filters, each of which is designed using the same prototype filter. The prototype filter can exploit the fact that the incoming signal has already been oversampled by a factor of 4, therefore it is only required to provide attenuation in the frequency regions occupied by spectral replicates generated by the upsampling process. Using the Remez algorithm, only the frequency regions that require additional attenuation are included in the stop-band constraints. The remaining regions are considered “don’t care” regions, allowing the size of the coefficient set to be dramatically reduced. The response of the cascaded Jakes and 4x upsampling filter is overlaid with the 32x upsampling prototype filter in Fig. 9, magnified on along the frequency axis for clarity. The designed prototype filter is symmetric and has a 192 tap impulse response, thus the implemented filter requires the storage of 96 coefficients.

Given a low enough operating rate, the dual polyphase

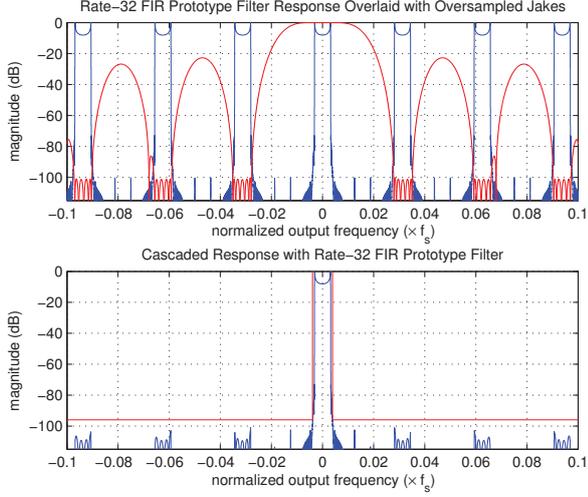


Fig. 9. Prototype Filter Design for the Rate-32 Upsampler

structure shown in Fig. 6 can again be implemented using a single MACC component. Using the running example, 1,000 vectors arrive at the 32x upsampler per second, thus 64,000 vectors per second are produced, recalling that the dual structure produces two separate output vectors for each input. Assuming the variable upsampler is always required to upsample by a factor greater than 32x, one of the produced vectors is always a delayed copy of the previous vector, thus an added memory element reduces the workload of the filter by a factor of two and the dual polyphase filter implementation collapses to the standard single polyphase upsampler structure. After this optimization, the workload of the filter is $W_{32x} = \frac{192}{32} = 6$. Using this optimization, the number of vectors produced per second is reduced to 32,000, thus requiring 256,000 outputs per second and 1.536 MMACs per second. With this low workload, it is realistic to assume this upsampler can also be implemented using a single MACC component in an FPGA/ASIC.

The top half of Fig. 10 illustrates the implementation details for the 32x polyphase upsampler. The integer portion of the accumulator k , depicted in Fig. 6 is used to control the virtual commutator address in the coefficient ROM.

The bottom half of Fig. 10 shows the variable linear interpolation components, which operate at the rate of the channel processing portion of the system (Fig 1). The fractional value of the accumulator α , is used to perform the convex combination between the newest and previous output vectors of the 32x upsampling component. The value of the accumulator input δ , determines the final upsampling factor. To determine the desired value of δ for this architecture, the following equation can be used

$$\delta = \frac{256 f_{max}}{f_d f_s}, \quad (14)$$

where f_s is the operating rate of the channel processing com-

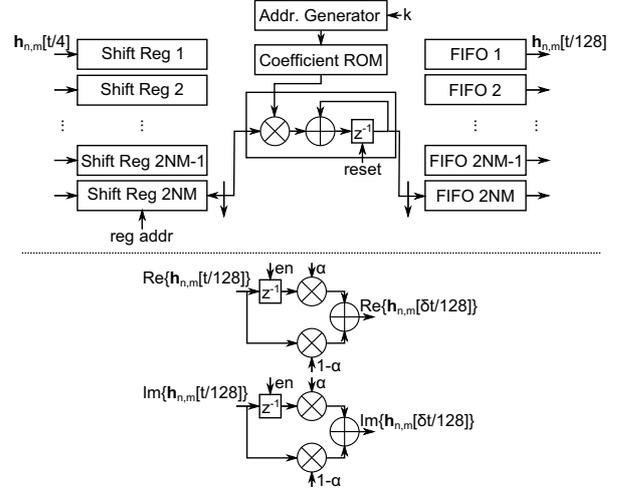


Fig. 10. Arbitrary-Ratio Upsampler: Rate-32 Polyphase Upsampling with Linear Interpolators

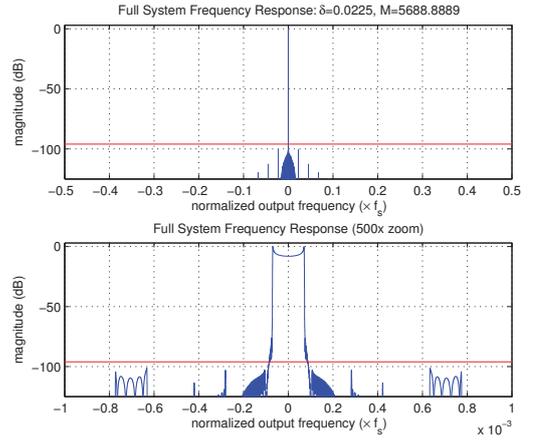


Fig. 11. Frequency Response of the Full System: $\delta = 0.0225$

ponent. If the parameters $f_s = 200$ MHz, $f_d = .8$, $f_{max} = 100$ Hz are selected, $\delta = 1.6 \times 10^{-4}$. Given a value of δ , the upsampling factor of the system, including the 4x preprocessing stage, can be determined by

$$M = \frac{128}{\delta}. \quad (15)$$

Therefore, for $\delta = 1.6 \times 10^{-4}$, $M = 8 \times 10^5$.

The frequency response of the full cascade of resampling filters is shown in Fig. 11 using $\delta = 0.0225$ to approximate an irrational resampling ratio of $\frac{128}{0.0225} \cong 5688.8889$.

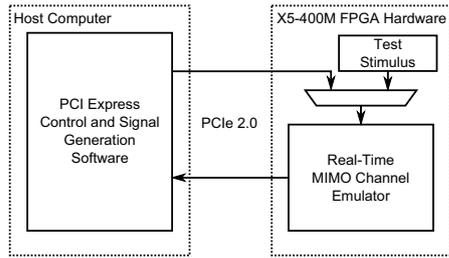


Fig. 12. Hardware Test Setup

	Used	Available	Percentage
slices	1,646	14,720	11%
BRAM	6	244	2%
DSP48E	38	640	6%

Table 1. FPGA Resource Consumption for a Single-Tap Emulator (Excluding WGN Source, $f_s = 200$ MHz)

7. RESULTS

The 2x2 MIMO channel emulator has been implemented using the X5-400M FPGA hardware designed by Innovative Integration. The X5-400M hardware features a Xilinx Virtex 5 SX95T FPGA and communicates with a host computer using PCI Express Gen 2.0. The implemented channel emulator runs at a maximum sampling rate of 200 MHz, processing 1.4901 GBytes/s. The test scenario is depicted in Fig. 12. The host computer generates LTE, or arbitrary signals in software to be processed by the channel emulator. The computer simultaneously and continuously transmits test signals and receives processed signals, which are stored on the host computer for analysis. In this test, the PCI express is capable of providing sustained data transactions of several hundred MBytes/s. In a typical channel emulation configuration, requiring the full processing rate to be sustained, the signal into the channel emulator is sourced from 4 ADCs, processed, and played by 4 DACs (when complex signals baseband signals are used).

Given $f_s = 200$ MHz, 200 MHz of processing bandwidth is achieved. Implementing the arbitrary resampler’s accumulator using a width of 22-bits provides a selectable Doppler resolution of 0.149 Hz. In addition to the programmable Doppler frequency, the spatial correlation matrix can be selected by the user for a given antenna array configuration. The scaling factor of the channel path is performed by scaling the matrix used by the Kronecker model off-line.

The final FPGA resource consumption for a single channel matrix and channel processing element is given in Tbl. 1 (i.e. $P = 1$). The channel processing element multiplies the complex channel matrix with the transmitted complex signal vector (shown in Fig. 1). This matrix-vector multiplication requires 16 real multiplications and 16 real addition operations at the transmitted signal’s sampling rate, accounting for

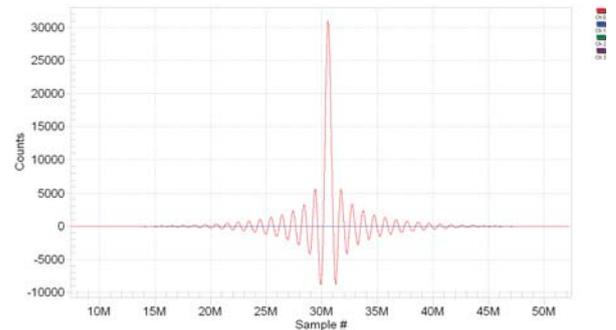


Fig. 13. Hardware-Sourced Jakes Processing Cascade Impulse Response

16 DSP48E elements in the total amount.

To test the Jakes process generation components, a test vector is sourced in hardware, taking the place of the WGN source. Fig 13 shows the impulse responses generated by the 4 complex Jakes signal processing chains by sourcing the vector $[1 - j1, 1 - j1, 1 - j1, 1 - j1]^T$ into the Jakes filter cascade. The transmitted signal vector for both antennas is set to a constant $[1 + j1, 1 + j1]$. For this test, the Kronecker matrix \mathbf{C} is set to the identity matrix. Doing so reveals the cascaded impulse response of the system. The impulse response reveals itself on the real part of the received signal vector and the imaginary part is constant zero. In this test, the Doppler is set to 200 Hz, yielding an impulse response with approximately 50 million taps! The length of the impulse response is determined by the variable Doppler frequency parameter.

Next, the statistical properties of the Kronecker model are verified. To perform verification, WGN is transmitted into the channel emulator and the correlation properties of the received signals are examined. The correlation for each matrix defined by the LTE specification are shown below [9].

$$\begin{aligned}
 \mathbf{R}_{\text{RXlow}} &= \begin{bmatrix} 1 & 0.0011 \\ 0.0011 & 1 \end{bmatrix} \\
 \mathbf{R}_{\text{RXmed}} &= \begin{bmatrix} 1 & 0.2874 \\ 0.2874 & 1 \end{bmatrix} \\
 \mathbf{R}_{\text{RXhigh}} &= \begin{bmatrix} 1 & 0.6683 \\ 0.6683 & 1 \end{bmatrix}
 \end{aligned} \tag{16}$$

8. CONCLUSION

A MIMO channel emulator architecture has been presented that allows the user to select the desired temporal, as well as spatial correlation properties of the channel. A theoretical overview of the Jakes Doppler model as well as the Kronecker spatial correlation model has been given. A detailed architecture has been presented that targets FPGA/ASIC implementation. The implemented design is very resource-efficient, consuming a small fraction of the Xilinx Virtex 5 SX95T

FPGA. Simulation results were given by showcasing the ability of the arbitrary resampling component to achieve very large, high-resolution upsampling factors, allowing the user to select the desired Doppler frequency. Finally, using the three LTE-specified spatial conditions, variable levels of correlation between the received signals has been displayed.

9. REFERENCES

- [1] W.C. Jakes, *Microwave Mobile Communications*, John Wiley & Sons Inc, 1974.
- [2] M.C. Jeruchim, P. Balaban, and K.S. Shanmugan, *Simulation of Communication Systems: Modeling, Methodology, and Techniques*, Information technology: transmission, processing, and storage. Kluwer Academic/Plenum Publishers, 2000.
- [3] M. Pätzold, *Mobile Fading Channels*, J. Wiley, 2002.
- [4] M. Patzold, R. Garcia, and F. Laue, “Design of High-Speed Simulation Models for Mobile Fading Channels by Using Table Look-up Techniques,” *Vehicular Technology, IEEE Transactions on*, vol. 49, no. 4, pp. 1178–1190, jul 2000.
- [5] Fei Ren and Yahong Zheng, “Hardware Emulation of Wideband Correlated Multiple-Input Multiple-Output Fading Channels,” *Journal of Signal Processing Systems*, vol. 66, pp. 273–284, 2012.
- [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 *Communications, 2008. ICC '08. IEEE International Conference on*, may 2008, pp. 718–724.
- [7] K.I. Pedersen, J.B. Andersen, J.P. Kermoal, and P. Mogensen, “A Stochastic Multiple-Input-Multiple-Output Radio Channel Model for Evaluation of Space-Time Coding Algorithms,” in *Vehicular Technology Conference, 2000. IEEE VTS-Fall VTC 2000. 52nd*, 2000, vol. 2, pp. 893–897 vol.2.
- [8] J.P. Kermoal, L. Schumacher, K.I. Pedersen, P.E. Mogensen, and F. Frederiksen, “A Stochastic MIMO Radio Channel Model with Experimental Validation,” *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 6, pp. 1211–1226, aug 2002.
- [9] 3GPP, “User Equipment (UE) Radio Transmission and Reception,” 2010.
- [10] E. Briggs, D. McLane, and B. Nutter, “A Real-Time Multi-Path Fading Channel Emulator Developed for LTE Testing,” in *Wireless Innovation Forum Conference*, Dec. 2011.
- [11] S. Sesia, M. Baker, and I. Toufik, *LTE, The UMTS Long Term Evolution: From Theory to Practice*, Wiley, 2009.
- [12] F.J. Harris, *Multirate Signal Processing for Communication Systems*, Prentice Hall PTR, 2004.
- [13] C. Dick and F. Harris, “Options for Arbitrary Resamplers in FPGA-Based Modulators,” in *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on*, nov. 2004, vol. 1, pp. 777–781 Vol.1.