

Implementation of a Real-Time Rayleigh, Rician and AWGN Multipath Channel Emulator

Peter John Green

Advanced Communication Department
Communication and Network Cluster
Institute for Infocomm Research
Singapore

Abstract—This paper describes the design and baseband processing of a low cost real-time RF channel emulator for the testing of wireless communication systems. It incorporates six independent Rayleigh / Rician fading blocks based on the method of exact Doppler spread (MEDS), an additive white Gaussian noise (AWGN) generator based on the Box-Muller method and programmable delay blocks for up to six multipath channels. Each complex Rayleigh or Rician block is uniquely implemented using one Xilinx direct digital synthesizer (DDS) block consisting of 16 frequency generators. The entire architecture is simulated in Matlab and implemented on the Xilinx ZC706+FMCOMMS2 evaluation board. The statistical properties of the implemented system are also presented.

Keywords— *Fading channel emulator, Rician fading, Rayleigh fading, AWGN, FPGA, sum of sinusoids*

I. INTRODUCTION

This paper covers an RF channel emulator designed, simulated and verified in Matlab and later developed into a real-time RTL implementation on a low cost Xilinx ZC706 evaluation board coupled to an Analog Devices FMCOMMS3 wideband software defined radio (SDR) board. The motivation of this implementation was the need for a low cost emulator capable of emulating a 6 path Rayleigh or Rician channel specific to a project. It is to be used in the testing of a receiver waveform in a mobile fading channel at RF in real-time. Previous works cover either fading channel [1] or AWGN [2] [3] simulation and/or emulation. Previous implementation methods use either a look-up table system, matrix system or a system of shift registers to generate the fading channels [4]. This paper is unique as it incorporates both fading, AWGN and 6 independent multipath channels and uses the Xilinx direct digital synthesizer blocks to generate the complex fading channels.

Commercial RF emulators are available but are extremely expensive and limited in the number of channels or the range of operating frequencies. What is needed is a low cost emulator capable of subjecting the transmitted signal to the receiver through specific Rayleigh or Rician channel characteristics with multipath delays as specified in TABLE 1.

In addition to the characteristics in TABLE 1 it is desired to incorporate an additive white Gaussian noise (AWGN)

generator for setting the signal-to-noise ratio (SNR) of the channel. Although the emulator is specific to the parameters in the table, parameters such as Doppler frequency, power and delay profiles can be easily changed in Matlab to generate the necessary parameters for real-time implementation at baseband.

TABLE 1. PARAMETER OF A SPECIFIC RICIAN CHANNEL

Modified 3GPP – ITU Vehicular A		
Doppler Frequency = 71 Hz, Velocity 148 km/h at 512 MHz		
Rician Channel with K-Factor = 1		
	Relative Delay	Relative Mean Power
Path	(ns)	(dB)
1	0	0
2	375	-1.0
3	750	-9.0
4	1125	-10.0
5	1750	-15.0
6	2500	-20.0

II. EMULATOR SYSTEM ARCHITECTURE

The emulator architecture consists of several blocks as depicted in Figure 1. The baseband emulator consist of the multipath Rician generator block, the AWGN generator block, 2 blocks to measure signal and noise powers, a couple of gain controllers to set the magnitudes of the signal and noise and an adder to sum the signal and noise components.

The RF signal received on the FMCOMMS3 SDR board is down converted into complex baseband I and Q channels, digitized and fed into the ZC706 board for baseband processing. It is first fed into the multipath Rician generator block. The output of this generator is then multiplied by a fixed gain to set the overall signal power. There is a measurement block that measures the signal power so that the signal power can be measured in real time.

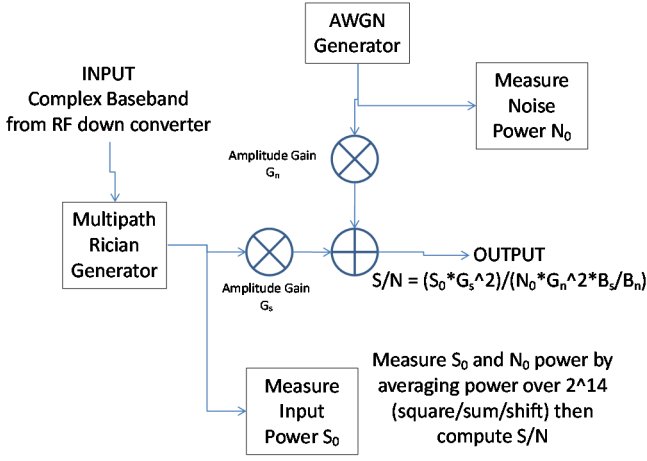


Figure 1: Architecture of the Emulator

The AWGN generator generates complex noise and the magnitude of this AWGN generator power is measured and is adjusted using the gain multiplier. The signal and noise signals are then added up by a complex adder and this baseband output is up converted back to RF.

III. MULTIPATH RAYLEIGH-RICIAN FADING GENERATOR BLOCK

The Multipath Rician Fading Generator consists of the following parts as depicted in Figure 2. It consists of 6 independent Rician channel simulators, several delays tapped at the correct points to meet the specific delay profiles, 6 multipliers to control the gains in each of the 6 paths to meet the power profiles, 6 multipliers to multiply the 6 weighted and delayed paths with the 6 independent Rician fading channel simulators and finally an adder to sum up the 6 signals together.

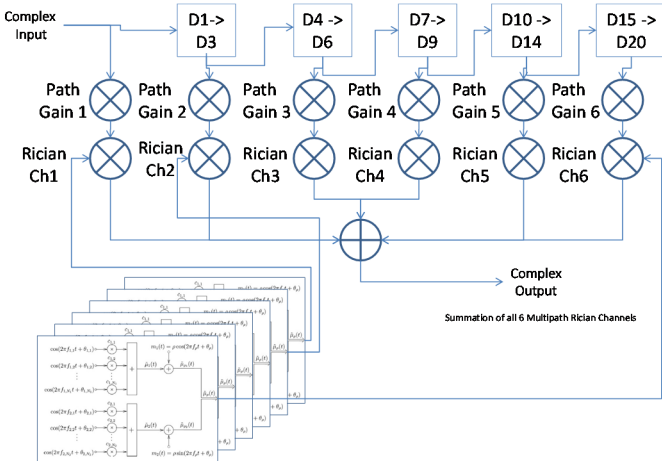


Figure 2: Complex baseband Multipath Rayleigh-Rician Fading Generator Block

The sampling frequency is 8 MSPS giving a minimum delay duration/resolution of 125 ns; path 1 has no delay, path 2 is delayed by 375 ns, path 3 – 750 ns, path 4 – 1125 ns, path 5 – 1750 ns and finally path 6 – 2500 ns. Twenty shift registers are needed and tapped at the correct points as shown in Figure 2.

IV. DETERMINISTIC RAYLEIGH-RICIAN GENERATOR BLOCK

The design of the Rician generator is based on a deterministic Rician fading generator structure based on the concept of the sum-of-sinusoids and using the method of exact Doppler spread (MEDS) approach as described in chapter 5 of [4]. The block diagram is depicted in

Figure 3. This prototype design is based on setting the minimum number of frequencies for N_1 and N_2 in Figure 3 to 7 and 8 respectively. This minimum number ensures that the quality of the generated fading characteristics is not compromised [4].

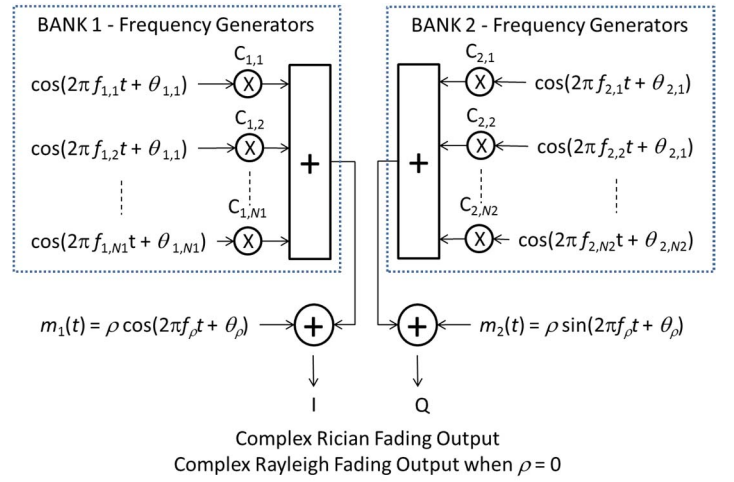


Figure 3: Deterministic Rayleigh / Rician Generator [4]

Each Rician generator consists of two banks of frequency generators to generate one complex output. The first bank consists of 7 $[N_1]$ frequency generators, each with 7 different start phase offsets. The output of each frequency generator is independently weighted and outputs are added. Similarly the second bank consists of 8 $[N_2]$ frequency generators configured in a similar fashion. The outputs of these 2 banks have deterministic Rayleigh fading channel properties. To transform into a Rician channel, the cosine $[m_1(t)]$ and sine $[m_2(t)]$ output of an independent frequency generator is added to the outputs of frequency generator banks N_1 and N_2 . The outputs $m_1(t)$ and $m_2(t)$ are weighed by p making it possible to adjust the K factor of the Rician channel. Setting $p = 0$ turns the Rician channel generator into a Rayleigh fading generator.

The algorithm needed to calculate the necessary frequencies and weighting constants is detailed in [4]. This prototype uses the MEDS algorithm to calculate the necessary constants. A random generator is used to determine the start phases of all

the 15 frequency generators in each Rician generator. The random start phases are important to ensure that the outputs for the Rician generators are independent.

There are many ways to realize the Rician generators using look-up tables or a system of shift registers as stated in Chapter 9 of [4]. In this paper, we develop a unique approach to use the Xilinx direct digital synthesizer (DDS) IP block. Our algorithm calculates the required phase advance needed to synthesize the various frequencies. The required phase advance is pre-calculated and the random start phase offsets is predetermined in Matlab. These constants are loaded to the Xilinx IP to generate the required Rician characteristics in the FPGA. The Xilinx DDS IP blocks consist of 16 DDS which is sufficient to synthesize one complex Rician fading channel.

V. AWGN GENERATOR BLOCK

The architecture of AWGN generator used in the prototype is based on the Box Mueller approach [3] shown in Figure 4.

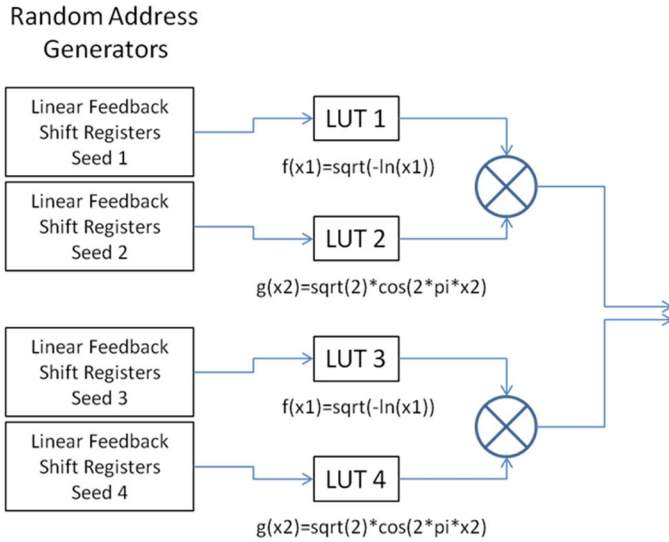


Figure 4: Complex AWGN generator block

There are many possible ways to generate AWGN using thermal noise, method of pre-computed probabilities, method using central limit theorem and the Box-Mueller method [2]. The Box-Mueller is widely used in simulation software and relatively easy to implement on hardware.

There are 3 steps to generate one AWGN output; first two generate independent values x_1 and x_2 of a random variable uniformly distributed over $[0, 1]$. The final step is to compute the functions $f(x_1)$ and $f(x_2)$ where $f(x_1) = \sqrt{-\ln(x_1)}$ and $g(x_2) = \sqrt{2} \cdot \cos(2\pi x_2)$. The final output is $n = f(x_1)g(x_2)$. To generate a complex output, another set is required. 4 seeds are required to generate a complex AWGN output.

VI. SIMULATION AND IMPLEMENTATION RESULTS

The Matlab simulation results of the 6 path Rician channel simulator is as shown in Figure 5.

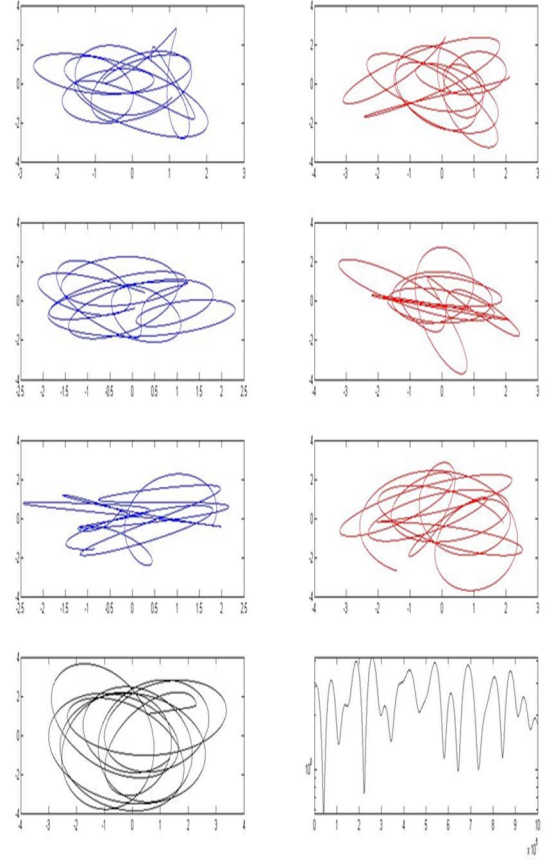


Figure 5: Output of the 6-path Rician Fading Channel Generator Block

The top 6 graphs of Figure 5 show the independent outputs of the 6 Rician channel simulator blocks. The bottom left graph is the final output and is the result of the summation of the 6 Rician channels subjected to the power and delay profiles defined in TABLE 1. The bottom right graph is the magnitude of the final output versus time displayed in log scale on the y-axis.

A large dataset of the output of one Rician fading generator in the FPGA implementation was taken for statistical analysis and the result is shown in Figure 6.

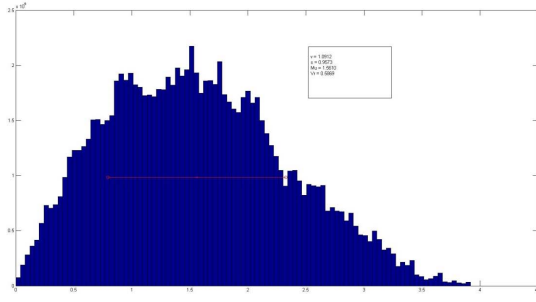


Figure 6: Statistical distribution of the Rician fading generator

The theoretical Rician statistical distribution (PDF) for various values of v and s are shown in Figure 7.

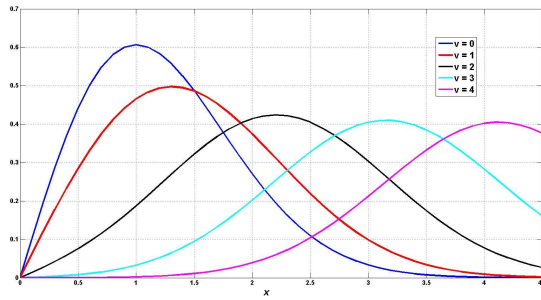


Figure 7: Theoretical Rician PDF for $s = 1$ and various values of v

It can be seen that the fading channel emulator has Rician characteristics with s and v approximately equal to 1 as shown in the red curve of Figure 7.

The Matlab simulation results of the AWGN channel simulator block is as shown in Figure 8.

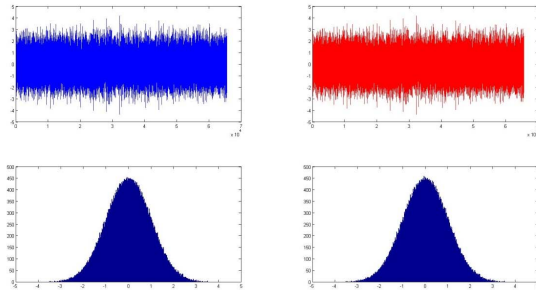


Figure 8: Matlab simulation of the AWGN generator – floating point (left) and fixed point (right)

The top graphs in Figure 8 show the AWGN generated by floating point and 16 bit fixed point versions of Matlab. The bottom graphs of Figure 8 show the statistical properties of the generated waveform. Both floating and fixed point versions have mean of almost zero, i.e. -5.5119×10^{-5} and -4.6186×10^{-5} respectively and a standard deviation of 0.9999 for both

versions. This is sufficiently close to the theoretical mean of zero and standard deviation of 1. The result of the version implemented on the FPGA platform based on a limited number of captured samples is shown in Figure 9.

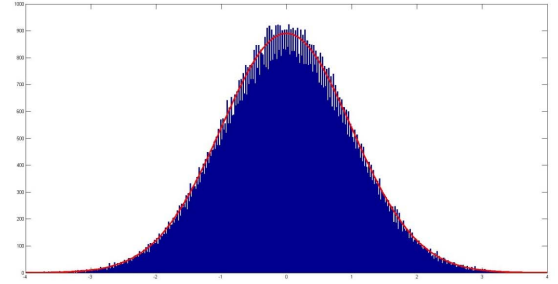


Figure 9: Statistical distribution of AWGN samples captured from FPGA implementation (blue) versus theoretical AWGN distribution (red)

VII. CONCLUSION

In this implementation of a real-time channel emulator, we have used Xilinx DDS blocks to generate 6 independent fading channels and used the Box-Mueller approach to generate the AWGN channel. With the appropriate delays, we have been able to emulate up to 6 independent fading paths to meet the requirements of the multipath fading channel as specified. The design is flexible to allow different Doppler speeds and different paths to be tailored for specific needs.

ACKNOWLEDGMENT

I wish to thank Jin Bo and May Thu Zin for fixed point to RTL conversion, RTL implementation, RTL simulation, RTL/Algorithm model verification, Vivado IPI flow migration, hardware implementation & testing. Zhang Guoping for initial attempt on fixed point to RTL conversion, RTL implementation, and hardware testing on the FMCOMMS2/3 + ZC706. Hsu Ting-Ming for setting up ZC706 + FMCOMMS2/3 platform together with Vivado IPI flow for building the emulator and support the on-board testing. Santhosh Kumar Pilakkat for project initiation, develop solution concept around FMCOMMS3, initial setup of ZC706 and FMCOMMS2/3 environment, build scripts and general coordination.

REFERENCES

- [1] M. Patzold, R. Garcia and F. Lane, "Design of High-Speed Simulation Models for Mobile Fading Channels by Using Table Look-UP Techniques," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 4, pp. 1178-1190, 2000.
- [2] J.-L. D. a. A. G. Emmanuel Boutillon, "Design of High Speed AWGN Communication Channel Emulator," vol. Analog Integrated Circuits and Signal Processing, no. 34, pp. 133-142, 2003.
- [3] J.-L. Danger, A. Ghazel, E. Boutillon and H. Laamari, "Efficient FPGA Implementation of Gaussian Noise

Generator for Communication Channel Emulation," in *7th IEEE Conference on Electronics, Circuits & Systems*, Kaslik, 2001.

[4] M. Patzold, *Mobile Radio Channels*, 2nd ed., Wiley, 2012.