

Weibull and Suzuki Fading Channel Generator Design to Reduce Hardware Resources

Pengda Huang, Dinesh Rajan, and Joseph Camp
Electrical Engineering, Southern Methodist University
{phuang, rajand, camp}@smu.edu

Abstract—A novel and efficient method for emulating a Weibull fading channel is presented. The proposed Weibull fading channel generator employs CORDIC technology which can work without the need for a multiplier, a scarce FPGA resource when emulating wireless channels. Using the proposed hardware implementation structure, the resource consumption is effectively reduced. Further, the proposed generator is able to produce the Weibull fading channel with arbitrary values of the shape parameter γ , in contrast to existing state-of-the-art methods, which can only generate Weibull channels with fixed values of γ . In simulation, the PDF of the Weibull fading channel agrees closely with the theoretical values. The simulation results also show the effectiveness of the proposed structure to generate Lognormal and Suzuki fading channels.

I. INTRODUCTION

Channel emulation is a convenient and powerful technique to study the performance of various wireless networking algorithms and protocols. Consequently, techniques for emulating various fading channels are widely studied [1]–[6]. Xiao [1] proposed a novel method for generating Rayleigh fading channels, which significantly improved the statistical properties of the generated channel. Rayleigh channels have also been implemented in a VLSI architecture [2] and FPGA platforms [3], [4]. Beaulieu and Cheng [7] utilized a rational polynomial to implement a Nakagami- m fading channel. The generation of Lognormal distributed sequence is discussed in [8]. Pätzold et al. [9] studied the simulation of Suzuki channels.

A Weibull fading channel is used to model the land-mobile channel in 800-900 MHz in urban environments [10]. This 800-900 MHz frequency band is an important frequency band and its importance is expected to increase in the coming years due to 4G communications. Urban areas account for the majority of wireless network connections, and thus, understanding the Weibull fading channel is important to improve the performance of wireless systems.

However, little attention has been paid to the generation of Weibull fading channels. To the best of our knowledge, there are only two references that discuss the generation of Weibull fading channels [11], [12]. Karadimas and Kotsopoulos [11] studied the simulation of a Weibull and Lognormal mixed fading channel but did not focus on the hardware implementation. Alimohammad [12] focused on the implementation of Weibull fading channel generation.

One of the main challenges in Weibull fading channel emulation lies in the calculation of a power function. Alimohammad [12] introduced a Weibull fading channel implementation method that uses a look-up-table to calculate the power function. Logarithmic segmentation followed by a uniform segmentation of the power function is undertaken prior to table look-up in order to save memory. However, such segmentation for Weibull fading channel generation encounters three main challenges. First, similar to any other look-up-table method, as the need for accuracy increases, the generation method will require large pieces of memory and long execution times. Second, the logarithmic approximation reduces the accuracy of the approximation, which will be demonstrated in Section VI. The inaccuracy becomes more pronounced when the slope of the curve changes quickly, *i.e.*, the absolute value of the second derivative is large. Third, complex control circuits are required to implement the joint logarithmic and uniform segmentation since the value of power a determines the slope distribution of the function x^a . The change of a induces a varying slope distribution of the power function x^a , which necessitates a modification to the logarithmic segmentation. Thus, more control circuits become necessary to account for different values of a .

This paper proposes an improved, low complexity, Weibull fading channel generation scheme based on a CORDIC methodology. If we use Givens rotation [13] based CORDIC operators to calculate the power function, only bit-shift and addition are needed, which thus reduces the use of hardware elements such as multipliers. Therefore, the proposed Weibull channel generator can be used to generate more channels on the same FPGA chip. The proposed generation scheme can produce a Weibull fading channel with any power values of γ while the existing scheme can only generate the channel data for those values stored in memory. Further, the proposed emulation scheme can also be used to generate Lognormal and Suzuki fading channels and the accuracy of such a generation method is illustrated. To further reduce the memory expenditure of the emulation, a novel Rayleigh channel generation scheme proposed in our prior work [14] is also utilized.

The remainder of this paper is organized as follows: In Section II, Weibull fading channel is introduced. An iterative structure for Rayleigh random variable generation is presented in Section III. Section IV discusses the CORDIC based Weibull fading channel generation method. Section V introduces the application of the proposed method to generate

Lognormal and Suzuki fading channels. In Section VI, numerical simulation results are presented, and conclusions are provided in Section VII.

II. WEIBULL FADING CHANNEL GENERATION

The probability density function (PDF), $f_W(x)$, and the cumulative density function (CDF), $F_W(x)$, of the Weibull distribution are given, respectively, by

$$f_W(x) = \frac{\gamma}{\beta} \left(\frac{x}{\beta}\right)^{\gamma-1} e^{-\left(\frac{x}{\beta}\right)^\gamma}, x > 0, \text{ and} \quad (1)$$

$$F_W(x) = 1 - e^{-\left(\frac{x}{\beta}\right)^\gamma}, \quad (2)$$

where β is the scale parameter, and γ determines the shape of the Weibull PDF. A simple and direct method to generate a Weibull random variable is to generate a uniform random variable and then map it using the inverse of the CDF of the Weibull RV, *i.e.*,

$$F_W^{-1}(u) = \beta [-\ln(1-u)]^{1/\gamma}, \quad (3)$$

where u is a uniformly distributed random variable.

Although there exists several methods to generate a uniform variable [15], [16], the resulting random sequence does not contain the desired Doppler information which is important in characterizing fast fading channels. Fortunately, Stüber [17] reported an Doppler-bearing uniform variable generation method which is derived from Rayleigh fading variable r_R as follows:

$$u = 1 - e^{-r_R^2/2\sigma_c^2}, \quad (4)$$

where r_R contains the Doppler information using the existing methods [1], [18] and σ_c is the variance.

In this paper, the sum-of-sinusoids (SOS) method [1] is used to generate a Rayleigh fading process $r_R(n)$ with the desired Doppler. Subsequently, the Weibull RV is generated as

$$r_W(n) = \beta \left(\frac{r_R^2(n)}{2\sigma_c^2}\right)^{1/\gamma}, \quad (5)$$

Further, the In-phase component (r_W^I) and Quadrature component (r_W^Q) of the Weibull fading channel can be obtained in terms of the In-phase and Quadrature components of the Rayleigh random variable as,

$$\begin{aligned} r_W^I(n) &= \beta \frac{r_R^I(n)}{|r_R(n)|} \left(\frac{r_R^2(n)}{2\sigma_c^2}\right)^{1/\gamma} \\ r_W^Q(n) &= \beta \frac{r_R^Q(n)}{|r_R(n)|} \left(\frac{r_R^2(n)}{2\sigma_c^2}\right)^{1/\gamma} \end{aligned} \quad (6)$$

Details of the Weibull fading channel emulation scheme will be introduced in the following two sections.

III. STRUCTURE OF RAYLEIGH DISTRIBUTED DATA WITH DOPPLER INFORMATION

The in-phase, r_R^I , and quadrature, r_R^Q , components of the Rayleigh fading channel can be generated using the SOS

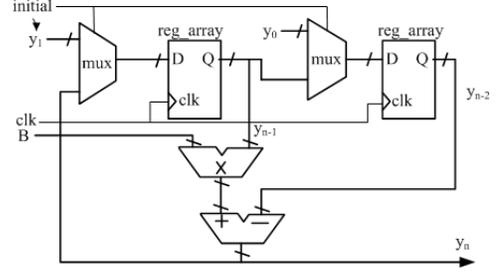


Fig. 1. Iteration architecture of one path.

method as,

$$r_R^I(n) = \frac{1}{P-1} \sum_{p=1}^P \cos(2\pi f_M(\cos(\alpha_p))t + \phi_p), \quad (7)$$

$$r_R^Q(n) = \frac{1}{P-1} \sum_{i=1}^P \sin(2\pi f_M(\cos(\alpha_i))t + \phi_i) \quad (8)$$

where α_p represents the arrival angle at the p -th multipath component, ϕ_p is the carrier phase which is uniformly distributed within $[0, \pi]$, and f_M denotes the maximum Doppler value. The Rayleigh fading channel, $r_R(n)$, can then be calculated as

$$r_R(n) = \sqrt{[r_R^I(n)]^2 + [r_R^Q(n)]^2}, \quad (9)$$

Most implementations of the SOS based Rayleigh channel emulation utilize a look-up table for the sinusoids required in (7) and (8). Clearly, to ensure accuracy of the emulation, large look-up tables and consequently large memories are required. An alternate iterative approach to generate SOS-based Rayleigh channels is proposed in [14], using an AR model. This iterative generation is given by,

$$\begin{aligned} \cos n\omega &= 2 \cos \omega \cos((n-1)\omega) - \cos((n-2)\omega) \\ \sin n\omega &= 2 \cos \omega \sin((n-1)\omega) - \sin((n-2)\omega) \end{aligned} \quad (10)$$

Fig. 1 presents the block diagram for generating one sinusoid of the fading channel using the novel scheme, where $B_p = 2 \cos \omega_p$. From Fig. 1, only two registers are needed to generate a single sinusoid. For the generation of all sinusoids, only one sinusoid value table is needed to initialize the first two samples when starting the iterative computation.

Such an iterative structure is effective in saving memory resources. For instance, consider the generation of a single ITU-Vehicle B channel containing 6 taps. Assume that 20 sinusoids are needed for each tap, and the *sinusoid* initialization table is 1024×2 bytes. Thus, only 2.048 kBytes of memory is needed to generate the I-Q components of Rayleigh fading channel using the novel structure. On the other hand, using a conventional look-up-table, 245.76 kBytes words of memory resources would be needed which is about 100 times the memory usage of the novel structure.

IV. CORDIC BASED WEIBULL FADING CHANNEL GENERATION

CORDIC algorithms, which were originally introduced in [13], are widely used to implement certain mathematical

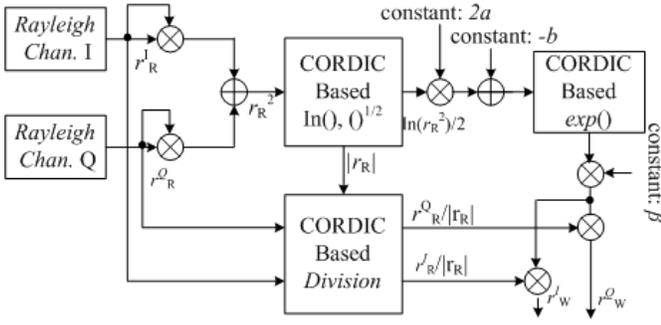


Fig. 2. Block diagram of Weibull fading channel generator.

operators in hardware. The advantage of using CORDIC algorithms is that they do not require multipliers but only adders, look-up tables, and bit shifters. Examples of operators that are implemented using CORDIC methods include exponential, trigonometric and hyperbolic functions.

In this paper, CORDIC methods are leveraged to implement the three main computational modules to generate the Weibull fading channel. Further, the CORDIC based Lognormal and Suzuki channel generation are also considered.

A. Architecture of proposed Weibull fading channel emulator

According to the introduction in Section II, we can generate the Weibull fading channel via (5). Since CORDIC operators can calculate exponential functions, not power functions, we transform (5) into a new form:

$$r_W(n) = \beta e^{\frac{\ln(r_R^2(n))}{\gamma} - \frac{\ln 2\sigma_c^2}{\gamma}} \quad (11)$$

Based on (11), to generate the I-Q components of the Weibull fading channel, the following 4 main steps are needed:

- 1) Generate Rayleigh fading channel ($r_R^2(n)$) using the iterative structure proposed in [14];
- 2) Calculate the natural logarithm ($\ln r_R^2(n)$) and the square root ($|r_R(n)| = \sqrt{r_R^2(n)}$) of $r_R^2(n)$;
- 3) Calculate the exponential function $e^{\ln(r_R^2(n)/2\sigma_c^2)}$;
- 4) Calculate I-Q factors ($r_R^I(n)/|r_R(n)|$, $r_R^Q(n)/|r_R(n)|$) and then generate the I-Q components of Weibull fading channel.

A pictorial description of the major computational blocks in the proposed Weibull fading channel generation scheme is given in Fig. 2. For a certain Weibull fading channel, the parameters σ_c and γ are fixed. Thus, $a = \frac{1}{\gamma}$ and $b = \frac{\ln 2\sigma_c^2}{\gamma}$ can be treated as constants in the hardware implementation.

As noted before, there are several complex computation modules needed in the Weibull channel generator: **exponential function** $e^{(\cdot)}$, **natural logarithm** $\ln(\cdot)$, **absolute value** $|\cdot|$ and **division** $(\cdot)/(\cdot)$. Fortunately, CORDIC operators can be used to compute all the modules in an iterative manner. In the following subsections, the details of the main computation modules are described.

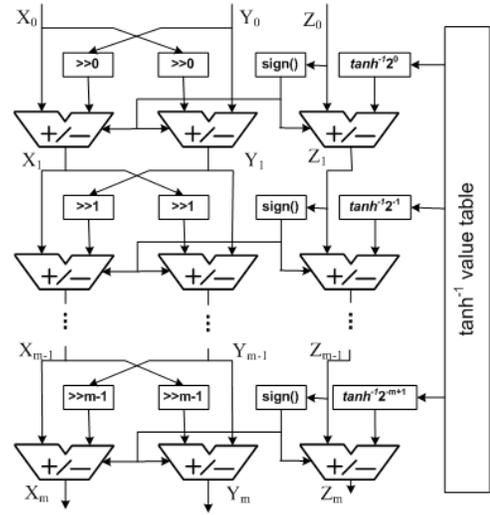


Fig. 3. CORDIC structure for power function calculation.

B. Exponential function module

The CORDIC-based exponential function is implemented in the rotation mode which is described by [19]

$$\begin{cases} x_{i+1} = x_i + \text{sign}(z_i)2^{-i}y_i \\ y_{i+1} = y_i + \text{sign}(z_i)2^{-i}x_i \\ z_{i+1} = z_i - \text{sign}(z_i)\alpha_i \end{cases} \quad (12)$$

where $\alpha_i = \tanh^{-1}(2^{-i})$.

To calculate the exponential function, the values of x , y and z are initialized to be 1, 0 and $\ln r_R^2(n)/\gamma$, respectively ($x_0 = 1, y_0 = 0$ and $z_0 = \ln r_R^2(n)/\gamma$). After several iterations, the outputs are $x_m = \cosh(\ln r_R^2(n)/\gamma)$ and $y_m = \sinh(\ln r_R^2(n)/\gamma)$. The value at $e^{\ln r_R^2(n)/\gamma}$ can be obtained as

$$\begin{aligned} e^{\ln r_R^2(n)/\gamma} &= x_m + y_m \\ &= \cosh(\ln r_R^2(n)/\gamma) + \sinh(\ln r_R^2(n)/\gamma) \end{aligned} \quad (13)$$

To improve the Weibull channel data throughput, a pipelined structure (shown in Fig. 3) is employed to implement the CORDIC based exponential function calculation.

From Fig. 3, the two advantages of the proposed Weibull fading channel generator is evident: i) no multipliers are needed in the computation, and ii) only a single \tanh^{-1} value table is used. The size of the \tanh^{-1} table is determined by the number of iterations. Under the condition of 13 iterations ($m = 13$) [19] and the 16-bit word length, the size of the \tanh^{-1} table is 208 bits. This number of iterations also results in an accurate evaluation of the exponential function and the eventual fading channel as demonstrated by the accuracy of the generated distribution.

C. Natural logarithm and square root module

To calculate $\ln(r_R^2(n))$, the CORDIC operator in the vectoring mode can be used. One level of the CORDIC-based natural logarithm calculator is shown in Fig. 4, and is given

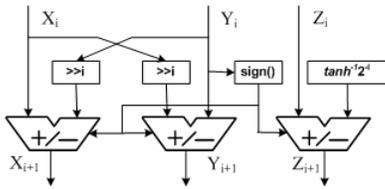


Fig. 4. CORDIC structure for $\ln()$ function calculation.

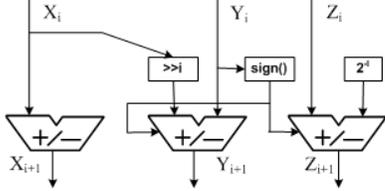


Fig. 5. CORDIC structure for division function calculation.

as

$$\begin{cases} x_{i+1} = x_i - \text{sign}(y_i)2^{-i}y_i \\ y_{i+1} = y_i - \text{sign}(y_i)2^{-i}x_i \\ z_{i+1} = z_i + \text{sign}(y_i)\alpha_i \end{cases} \quad (14)$$

where $\alpha_i = \tanh^{-1}(2^{-i})$.

To calculate the natural logarithm, the values of x , y and z are initialized, respectively, to $r_R^2(n) + 1$, $r_R^2(n) - 1$ and 0 , i.e., $x_0 = r_R^2(n) + 1$, $y_0 = r_R^2(n) - 1$ and $z_0 = 0$. After the iterations, the output $z_m = (\ln r_R^2(n)) / 2$.

As described in Section IV-A, we also need to calculate the absolute value of the square root of r_R^2 , of the Rayleigh fading channel. Fortunately, the CORDIC operator in Fig. 4 is also able to provide the values of $\ln r_R^2(n)$ and $|r_R(n)|$ simultaneously. After the desired number of iterations of (14), the output on the x branch equals the square root ($|r_R^2(n)|$) of $r_R^2(n)$.

D. Division module

To calculate the I-Q components of the fading channel, a division operator is also needed. One level of a CORDIC-based division calculator is shown in Fig. 5, and its mathematical description is given by

$$\begin{cases} x_{i+1} = x_i \\ y_{i+1} = y_i - \text{sign}(y_i)2^{-i}x_i \\ z_{i+1} = z_i + \text{sign}(y_i)\alpha_i \end{cases} \quad (15)$$

where $\alpha_i = 2^{-i}$.

To implement division, the values of x , y and z are initialized to be $r_R^I(n)$, $|r_R(n)|$ and 0 , respectively ($x_0 = r_R^I(n)$, $y_0 = |r_R(n)|$ and $z_0 = 0$). After the desired number of iterations, the output at the z branch is the division result $z_m = r_R^I(n)/|r_R(n)|$.

In this section, we presented the details about how to utilize CORDIC technology in the Weibull fading channel implementation. Compared to existing methods, the CORDIC-based operators architecture effectively saves the hardware resources and generates more accurate fading channel, which will be illustrated in Section VI.

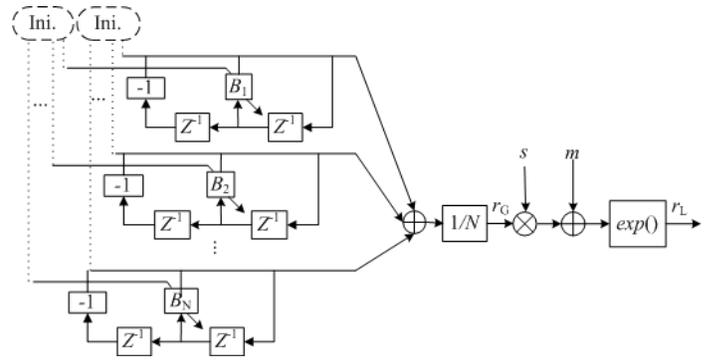


Fig. 6. Block diagram of Lognormal fading process generator.

V. APPLICATION TO LOGNORMAL, SUZUKI FADING CHANNEL GENERATION

For Lognormal and Suzuki fading channel generation, one main challenge lies in the exponential function calculation. Motivated by the proposed Weibull channel generator, CORDIC technology can also produce the Lognormal and Suzuki channel.

Lognormal distributions are used to characterize the shadowing effect of channel fading. The mathematical expression of the Lognormal PDF is given by

$$p_L(r) = \frac{1}{\sqrt{2\pi\sigma r}} e^{-\frac{(\ln r - \mu)^2}{2\sigma^2}}, r > 0, \quad (16)$$

where μ and σ denotes the mean value and variance of random process $\ln r$.

The block diagram of the Lognormal process $r_L(n)$ generation is illustrated in Fig. 6, where $r_G(n)$ denotes the generated Gaussian random process based on the SOS method. The Gaussian process has the mean value of μ_0 and the variance σ_0 , $r_G(n) \sim \mathcal{N}(\mu_0, \sigma_0)$. The iterative structure based Rayleigh channel generator (Section III) is used to produce the Gaussian fading variate in Fig. 6, which can effectively save memory resources. The parameters s and m are the modification coefficients, which are used to transform μ_0 and σ_0 to the actual mean and variance of Lognormal random process.

From Fig. 6, it is clear that an exponential function calculation is needed. We can directly use the CORDIC operator in Fig. 3 to implement the exponential calculation module and complete the Lognormal channel generation.

The Suzuki fading channel model jointly considers fast fading and slow fading [20]. The PDF of Suzuki random variable is written as

$$p_S(r) = \int_0^\infty \frac{r}{\omega^2} e^{-\frac{r^2}{2\omega^2}} \frac{1}{\sqrt{2\pi\sigma\omega}} e^{-\frac{(\ln \omega - \mu)^2}{2\sigma^2}} d\omega \quad (17)$$

The Suzuki variable can be generated by multiplying a Rayleigh fading channel with the Lognormal variate, which is described by

$$r_S(n) = r_R(n)r_L(n) \quad (18)$$

Fig. 7 presents the block diagram of Suzuki fading channel generator. In Fig. 7, the CORDIC operators can be applied in

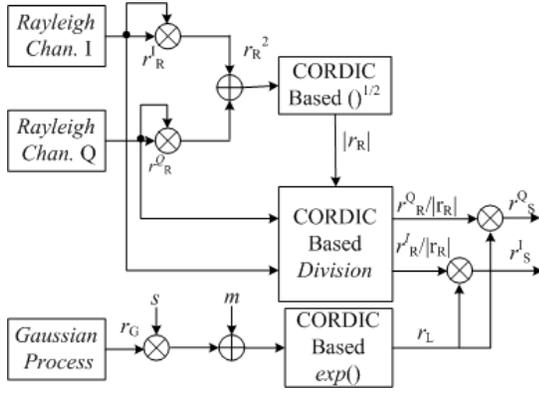


Fig. 7. Block diagram of Suzuki fading process generator.

all of the main calculation modules including, the square root $(\cdot)^{1/2}$, division $(\cdot)/(\cdot)$ and exponential function $e^{(\cdot)}$.

VI. EXPERIMENTS AND SIMULATIONS

In this section, we estimate the hardware resource expenditure of the proposed Weibull fading channel generator on a Xilinx Virtex-4 XC4VFX100FFG1517-11C FPGA.* We also compare the computation complexity between the proposed Weibull channel generator and the method in [12]. For the referenced Weibull channel generator [12], only 10 Weibull channels are considered to be generated in the computation complexity comparison.

Table I lists the hardware expenditure in the items of Slice, LUT, DSP48s and RAMBs with the corresponding ratios to the configurable resources on FPGA which are implemented in VERILOG.

TABLE I
HARDWARE RESOURCE CONSUMPTION FOR WEIBULL CHANNEL GENERATOR

	Slices	LUT	DSP48S	RAMBS
Proposed Generator	1,792(2%)	1994(2%)	5(3%)	1(0.5%)
Total Resources	84,352	84,352	160	376

From Table I, the Weibull fading channel generator requires 2% of the available Slices, 3% of the DSP48s, 2% of 4-input LUT and 0.5% RAMBs, where the number of DSP48s indicate the multiplier consumption and the RAMBs reflect the memory usage. The advantage on saving the memory resources and multipliers is due to the two main reasons: the CORDIC structure based power function module requires no multipliers and needs only 208 bits of memory as the \tanh^{-1} look-up-table. Furthermore, the iterative-structure-based Rayleigh channel generator effectively reduces memory consumption.

TABLE II
COMPLEXITIES OF WEIBULL CHANNEL GENERATION ALGORITHMS

	Mul.	Adder	Units of Memory
Proposed Architecture.	5	65	77
Ref. Architecture from [12]	6	10	320

*We pick this particular FPGA because we have several evaluation boards with this specific FPGA in our lab. A similar exercise can be conducted for other FPGAs and the numbers shown here are to help illustrate the performance of the proposed emulation method.

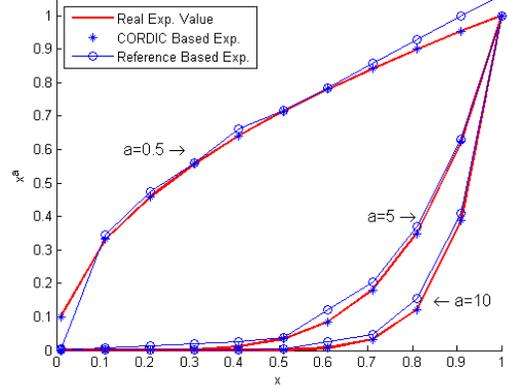


Fig. 8. Power function curves for different power a .

Table II lists certain metrics of computational complexity of the Rayleigh channel generator. From Table II, the proposed generator consumes more adders due to the large data throughput oriented pipeline structure used. Indeed, the volume of the configurable multipliers and block memories are much smaller than the other items, like Slices (components of adders). And the numbers of used multipliers and memory blocks are usually taken to evaluate the algorithm implemented on the FPGA. Thus, the proposed generator is more promising in the engineering practice than the structure of [12].

Furthermore, the proposed generator can produce Weibull fading channel with any value of γ , $\gamma > 0$. For the generator in [12], when the value range of γ increases, the number of required memory blocks will correspondingly increase.

As mentioned in Section I, the cascaded logarithmic and uniform segmentation [12] will induce larger error than the proposed method when calculating the power function. Fig. 8 presents the function (x^a) with different powers ($a = 0.5, 5$ and 10) over the range $x \in [0, 1]$. From Fig. 8, the power function values calculated from the proposed method match the theoretical curve well, while the mismatch can be observed in the calculation based on [12].

Furthermore, in the cascading structure additional control is needed to determine which part of the curve requires more granular segmentation. Because the power a of x^a , $x \in [0, 1]$, determines the shape of the curve. Generally, when $a > 1$, the curve near $x = 0$ has a small slope (requiring less granular segmentation) and the curve near $x = 1$ has a larger slope (requiring more granular segmentation). For $0 < a < 1$, the property of the slope is reversed. Thus, the additional control is needed for adaptive segmentation at the different granular levels for varying values of a .

Fig. 9 plots the PDF of 10^5 Weibull samples of three different γ values ($\gamma = 0.8, 5$ and 10). The Doppler contained in the Weibull samples is 200 Hz, and the sampling frequency is 100 kHz. For comparison, the Weibull PDF curve ($\gamma = 5$) based on the method in [12] is also illustrated, where severe distortion of the PDF curve is observed. The PDF computed from the proposed Weibull channel generation scheme agrees closely with the theoretical curve.

Fig. 10 and Fig. 11 show, respectively, the generated

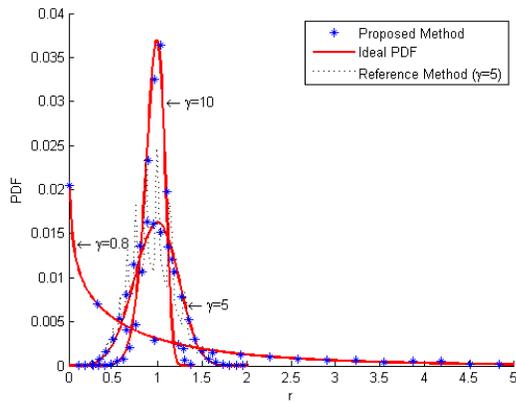


Fig. 9. PDF of Weibull fading channel samples at different γ .

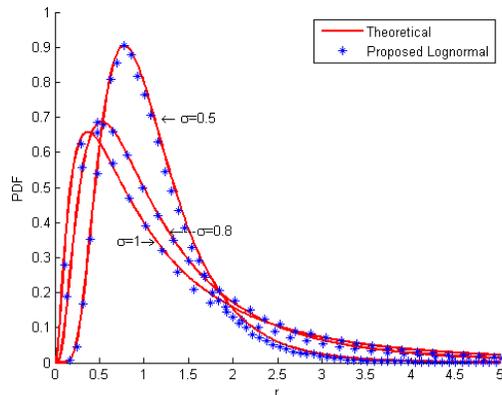


Fig. 10. PDF of Lognormal fading channel samples at different σ .

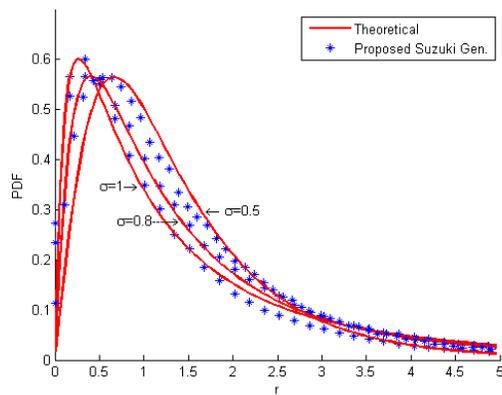


Fig. 11. PDF of Suzuki fading channel samples at different σ .

Lognormal and Suzuki PDFs. A close agreement between the theoretical and the simulated results is observed which confirms the effectiveness of the proposed Lognormal and Suzuki fading channel generators.

VII. CONCLUSION

An efficient fading channel generator for Weibull, Lognormal and Suzuki distributions is proposed. Specifically, we use CORDIC operators to generate all three fading channels, which effectively reduced the consumed multipliers and

memory resources. Moreover, the generator can produce the Weibull channels with any value of γ , while existing methods can only generate limited channels with pre-stored parameters. To reduce memory consumption further, we also used an iterative structure proposed in our prior work to generate Rayleigh fading data, which is the intermediate variate for the Weibull, Lognormal and Suzuki fading channel generation.

REFERENCES

- [1] C. Xiao, Y. R. Zheng, and N. Beaulieu, "Novel sum-of-sinusoids simulation models for Rayleigh and Rician fading channels," *Wireless Communications, IEEE Transactions on*, vol. 5, no. 12, pp. 3667–3679, december 2006.
- [2] G. Rao, R. Bhattacharjee, and S. Nandi, "VLSI architecture for Rayleigh and Rician fading generators," in *TENCON 2004. 2004 IEEE Region 10 Conf.*, vol. C, Nov 2004, pp. 121–124 Vol. 3.
- [3] S. Fard, A. Alimohammad, B. Cockburn, and C. Schlegel, "A single FPGA filter-based multipath fading emulator," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, 30 2009-dec. 4 2009, pp. 1–5.
- [4] O. Nasr and B. Daneshrad, "Design and FPGA implementation an accurate real time 3x4 MIMO channel emulator," in *Signals, Systems and Computers, 2009 Conf. Record of the Forty-Third Asilomar Conf. on*, Nov 2009, pp. 764–768.
- [5] K. Borries, G. Judd, D. Stancil, and P. Steenkiste, "FPGA-based channel simulator for a wireless network emulator," in *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, april 2009, pp. 1–5.
- [6] J.-K. Hwang, K.-H. Lin, J.-D. Li, and J.-H. Deng, "Fast FPGA prototyping of a multipath fading channel emulator via high-level design," in *Communications and Information Technologies, 2007. ISCIT '07. International Symposium on*, oct. 2007, pp. 168–171.
- [7] N. Beaulieu and C. Cheng, "Efficient Nakagami-m fading channel simulation," *Vehicular Technology, IEEE Transactions on*, vol. 54, no. 2, pp. 413–424, march 2005.
- [8] D. Huang, D. Zeng, T. Long, and J. Yu, "Design of a correlated Lognormal distributed sequence generator based on Virtex-IV series FPGA," in *Computer Application and System Modeling (ICCSM), 2010 International Conference on*, vol. 2, oct. 2010, pp. V2–340–V2–343.
- [9] M. Patzold, U. Killat, and F. Laue, "A deterministic digital simulation model for Suzuki processes with application to a shadowed Rayleigh land mobile radio channel," *Vehicular Technology, IEEE Transactions on*, vol. 45, no. 2, pp. 318–331, may 1996.
- [10] M. Alouini and M. Simon, *Digital communication over fading channels*. New York: Wiley-IEEE Press, 2005.
- [11] P. Karadimas and S. Kotsopoulos, "The Weibull-Lognormal fading channel: Analysis, simulation, and validation," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 7, pp. 3808–3813, sept. 2009.
- [12] A. Alimohammad, S. Fard, and B. Cockburn, "Hardware implementation of Nakagami and Weibull variate generators," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 20, no. 7, pp. 1276–1284, july 2012.
- [13] J. E. Volder, "The birth of CORDIC," *The Journal of VLSI Signal Processing*, vol. 25, no. 2, pp. 101–105, 2000.
- [14] P. Huang, M. J. Tonnemacher, Y. Du, D. Rajan, and J. Camp, "submitted to *IEEE INFOCOM 2013* towards scalable network emulation: Channel accuracy versus implementation resources," 2013.
- [15] S. Park and K. Miller, "Random number generators: Good ones are hard to find," *Communications of the ACM*, vol. 3, no. 10, p. 11921201, 1988.
- [16] R. M. Ziff, "Four-tap shift-register-sequence random-number generators," *Computers in Physics*, vol. 12, no. 4, p. 385 392, 1998.
- [17] G. L. Stüber, *Principles of mobile communication*. New York: Kluwer Academic Publishers, 2001.
- [18] K. Baddour and N. Beaulieu, "Autoregressive modeling for fading channel simulation," *Wireless Communications, IEEE Tran. on*, vol. 4, no. 4, pp. 1650–1662, Jul 2005.
- [19] P. Meher, J. Valls, T.-B. Juang, K. Sridharan, and K. Maharatna, "50 years of CORDIC: Algorithms, architectures, and applications," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 56, no. 9, pp. 1893–1907, sept. 2009.
- [20] H. Suzuki, "A statistical model for urban radio propagation," *Communications, IEEE Transactions on*, vol. 25, no. 7, pp. 673–680, jul 1977.