Scientific Research Publishing

Int. J. Communications, Network and System Sciences, 2017, 10, 105-115
http://www.scirp.org/journal/ijcns
ISSN Online: 1913-3723
ISSN Print: 1913-3715

Design of Fading Channel Simulator Based on IIR Filter Using Genetic Algorithm

Haofeng Wang, Siwei Du, Ning Guan

School of Information and Electronics, Beijing Institute of Technology, Beijing, China
Email: bighao.bit@gmail.com

How to cite this paper: Wang, H.F., Du, S.W. and Guan, N. (2017) Design of Fading Channel Simulator Based on IIR Filter Using Genetic Algorithm. *Int. J. Communications, Network and System Sciences*, **10**, 105-115.
https://doi.org/10.4236/ijcns.2017.105B010

Received: March 12, 2017
Accepted: May 23, 2017
Published: May 26, 2017

Abstract

This article designs a Rayleigh fading channel simulator using IIR filter (called Doppler filter), which is used to approximate the Jakes Doppler spectrum. We mainly focus on the design of Doppler filter and model it as an optimization problem. Non-convexity of the problem is proved in this paper and genetic algorithm (GA) is used to optimize it, which has never been used before in this area. Simulation result shows that GA converges at the solution corresponding to very accurate approximation of Jakes PSD. Finally, several statistical characteristics of the simulator are verified, including correlation functions and level-crossing rate (LCR), all of which match the theoretical predictions pretty well.

Keywords

Channel Simulator, Genetic Algorithm, Jakes, Biquad, IIR

1. Introduction

Channel simulator is usually used in laboratory to verify system performance, which is convenient and efficient. This article designs an accurate and highly efficient channel simulator to produce fading process $h(t)$. Here $h(t)$ is a complex Gaussian random process whose amplitude $|h(t)|$ is a Rayleigh process, used to characterize the Rayleigh fading channel.

According to [1], when receiver moves relatively to the transmitter in a uniform scattering environment, Doppler shift will occur, which makes correlation functions of $h(t)$ satisfy:

$$\begin{gathered} R_h(\tau) = J_0(2\pi f_D \tau) \\ R_{h_I,h_Q}(\tau) = 0 \end{gathered} \tag{1}$$

where f_D is the maximum Doppler frequency. h_I and h_Q are the real and imaginary parts of $h(t)$ respectively and they are independent. J_0 is the zero-order Bessel function of the first kind and its Fourier transform (2) is the power spectral density (PSD) of $h(t)$, which is called the Jakes Doppler spectrum [1].

$$S_h(f)=\begin{cases}\dfrac{1}{\pi f_D\sqrt{1-(f/f_D)^2}}, & |f|\leq f_D\\ 0, & otherwise\end{cases} \tag{2}$$

Channel simulator is designed to generate the fading process $h(t)$ whose correlation functions and PSD satisfy (1) and (2) respectively. In the open literature, two methods have been proposed to design the simulator.

The first method, called "the sum of sinusoids", was first proposed by Jakes [2], where a number of sine waves were summed up to simulate the fading process. Though the method is simple and easy to be implemented, fading process generated by this method lacks of randomness and some of its high order statistic characteristics are difficult to be consistent with theoretical ones [3].

The second method is to filter noise, where complex white Gaussian noises pass through a filter to generate the fading process. This filter, which we call Doppler filter, can be implemented by both FIR [4] and IIR filter [5]. Since the transition band is very steep in Jakes PSD (2), if using FIR filter to approximate it accurately, filter order needs to be high, which is expensive in terms of hardware resources. The IIR filter, however, can achieve excellent approximation with considerably low order.

Reference [5] proposed a simulator structure consisting of an IIR filter and an interpolator. It calculated coefficients of IIR filter using ellipsoid algorithm (EA) [6]. To keep the filter stable, any pole gone out of the unit circle in iterations would be reflected back. This makes the iteration start from a new point, like random searching.

Reference [7] [8] also used EA but proposed a barrier function to constrain poles lying inside the unit circle. However, the convergence will be affected by the starting point, because non-convexity of the problem. To the best of my knowledge, previous articles used cascaded low order IIR filter only mentioned the non-convexity of their objective functions, however, without proving.

Based on the review above, this article realizes a channel simulator proposed in [5] using different algorithm. We focus on the design of Doppler filter inside it, which is actually an IIR filter fitting problem. Non-convexity of the problem is proved for the first time and genetic algorithm is used to solve it, which is a kind of global optimization method and performs well in solving non-convex problems.

This paper is organized as follows. Section 2 introduces the structure of the channel simulator and defines the optimization problem to design the Doppler filter. Section 3 uses genetic algorithm to solve the problem and section 4 testifies correlation functions and LCR of the fading process generated. Finally, Section 5 concludes the paper.

Notation: The following notations are used throughout this paper. Boldface lowercase letter denotes vectors. Operator $||$ denotes the absolute value and $\|\mathbf{x}\|_2$ represents Euclidean norm of vector $\mathbf{x}$. The superscript $*$ and T denote complex conjugation and transposition respectively.

2. Channel Simulator

2.1. Structure of the Simulator

Figure 1 shows the structure of the channel simulator used in this paper, which consists of three parts. Noise generator produces complex white Gaussian noise. Doppler filter filters white noise to generate correlated noise with its PSD complying with Jakes PSD. Interpolator makes Doppler rate of the fading process output satisfy the requirement of the physical channel. The Doppler rate is defined as $\rho = f_D / f_s$, where f_s is the sampling rate of fading process $h(t)$ [5]. The ρ value at the Doppler filter usually takes 0.2. When implemented, Doppler filter remains unchanged and interpolation factor I of the interpolator is changed to adjust ρ at the output: $\rho = 0.2/I$. This article will focus mainly on the design of the Doppler filter next.

One important consideration in designing Doppler filter is to ensure the stability. Generally, the larger the filter order is, the more difficult to keep it stable. On account of this, a good choice is to use the cascaded low order IIR sections as the Doppler filter. Except for being easy to be stable, this structure has additional advantage that is less sensitive to quantization error compared with high order IIR filter.

Formula (3) shows the system function of the traditional biquad IIR filter, where *a*, *b*, *c*, *d* are real-valued optimization variables. Reference [9] studied the stable condition of this biquad, which was called the "Triangle Stability Region" but not intuitive.

$$H_{biquad}(z) = \frac{1 + az^{-1} + bz^{-2}}{1 + cz^{-1} + dz^{-2}} \tag{3}$$

We factorize its numerator and denominator, writing it into the form of conjugate poles and zeros (4), where *zr*, *zi*, *pr* and *pi* are real-valued optimization variables, representing the real and imaginary parts of *zero* and *pole* respectively.

$$\begin{gathered} H_{SOS}(z) = \frac{(z - zero)(z - zero^*)}{(z - pole)(z - pole^*)}, \\ zero = zr + j \cdot zi, \\ pole = pr + j \cdot pi. \end{gathered} \tag{4}$$

This Second-Order Section (SOS) is inherently stable, as long as all *poles* are strictly inside the unit circle. When implementing in a physical system, restore

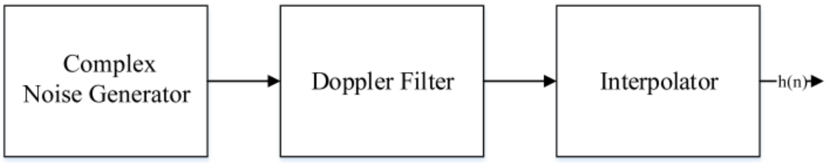


Figure 1. Structure of the channel simulator.

the conjugate form into its traditional biquad form.

$$|\,pole\,| = \sqrt{pr^2 + pi^2} < 1 \tag{5}$$

Usually, multiple of SOSs need to be cascaded to approximate the objective response. Assuming we use K SOSs, and uniformly sample variable z into $M+1$ points points along the unit circle: $z_i = e^{j\pi i/M}, i = 0,1,\ldots,M$ (usually M = 500), to get the frequency response of Doppler filter after frequency sampling:

$$H_i = H(z_i;\mathbf{x}) = A\prod_{k=1}^{K}\frac{(z_i - zero_k)(z_i - zero_k^*)}{(z_i - pole_k)(z_i - pole_k^*)}, i = 0,1,\ldots,M \tag{6}$$

wherein A is the scaling factor and $\mathbf{x}$ is a vector containing all optimization variables:

$$\mathbf{x} = [A, zr_1, zi_1, pr_1, pi_1, \ldots, zr_K, zi_K, pr_K, pi_K] \tag{7}$$

2.2. Optimization Model

Ideal amplitude response of Doppler filter is the square root of Jakes PSD (2) but cannot be realized with a physical system according to Paley-Winner theorem, because at the frequency point $f = f_D$ the amplitude of Jakes PSD changes from $+\infty$ into zero immediately. For this reason, [4] gives a modified objective response that can be achieved by a stable system. Referring to it herein, this article uses (8) as the objective amplitude response, where i is the index of frequency point z_i.

$$Y_i = \begin{cases} \sqrt{1/\sqrt{1-(i/L)^2}}, & i = 0,1,\ldots,L-1 \\ \sqrt{L(\pi/2 - \arcsin(1-1/L))}, & i = L \\ \sqrt{10^{SA/10}}, & i = L+1,\ldots,M \end{cases} \tag{8}$$

In the formula above, parameter SA denotes the stopband attenuation relative to the passband. Objective function could be defined as:

$$f(\mathbf{x}) = \sum_{i=0}^{M} v_i(log(Y_i) - log(|\,H_i\,|))^2 \tag{9}$$

where $\mathbf{v} = [v_0, v_1, \ldots, v_M]$ is a weight vector to emphasize the error minimization for certain frequency band [7]. The complete optimization problem is:

$$\begin{aligned} &minimize \quad f(\mathbf{x}) \\ &subject\ to \quad -1 \le A \le 1 \\ &\qquad \sqrt{pr_k^2 + pi_k^2} < 1,\ k = 1,2,\ldots,K \\ &\qquad \sqrt{zr_k^2 + zi_k^2} < 1,\ k = 1,2,\ldots,K \end{aligned} \tag{10}$$

wherein constraint $-1 \le A \le 1$ is an empirical condition. $\sqrt{pr_k^2 + pi_k^2} < 1$ limits *poles* of the k^{th} SOS located inside the unit circle to keep Doppler filter stable. Constraint $\sqrt{zr_k^2 + zi_k^2} < 1$ confines *zeros* lying inside the unit circle too, so that the Doppler filter will be a minimum phase system and have the minimum group delay.

Non-convexity of (10) confirms only if one point $\mathbf{x}_0$ exists in the feasible region, around which the objective function is not convex. We will find such $\mathbf{x}_0$

next.

According to the first-order conditions of convexity, $f(\mathbf{x})$ is convex if and only if the feasible region ($dom\ f$) is convex and (11) holds for all $\mathbf{x}, \mathbf{x}+\Delta\mathbf{x} \in dom\ f$ [10].

$$f(\mathbf{x}_0+\Delta\mathbf{x}) \geq f(\mathbf{x}_0)+\nabla f(\mathbf{x}_0)^T \cdot \Delta\mathbf{x} \tag{11}$$

Here ∇f denotes the gradient of $f(\mathbf{x})$ respect to $\mathbf{x}$, and $\Delta\mathbf{x}$ is the incremental step of $\mathbf{x}$.

For problem (10), $dom\ f$ is obviously a convex set. To show the non-convexity of $f(\mathbf{x})$, we randomly generate $\mathbf{x}_0$ from $dom\ f$ and approximately calculate $\nabla f(\mathbf{x}_0)$ using its finite difference in computer. Then change $\Delta\mathbf{x}$ gradually and plot $f(\mathbf{x})$ and its first-order approximation to see if equation (11) is satisfied.

An example is showed in **Figure 2**, in which we see that $f(\mathbf{x})$ is not convex around $\mathbf{x}_0$. Indeed, much more than one points like $\mathbf{x}_0$ could be found in simulation, at which $f(\mathbf{x})$ is not convex.

3. Genetic Algorithm

3.1. Introduction of Genetic Algorithm

Genetic Algorithm (GA) simulates the evolution process of biotic population with selection, crossover and mutation operations. The population evolves after several generations to achieve a stable situation, which corresponds to the convergence of optimization problem [11]. Procedures in simulating GA have been described as pseudocodes in **Figure 3**.

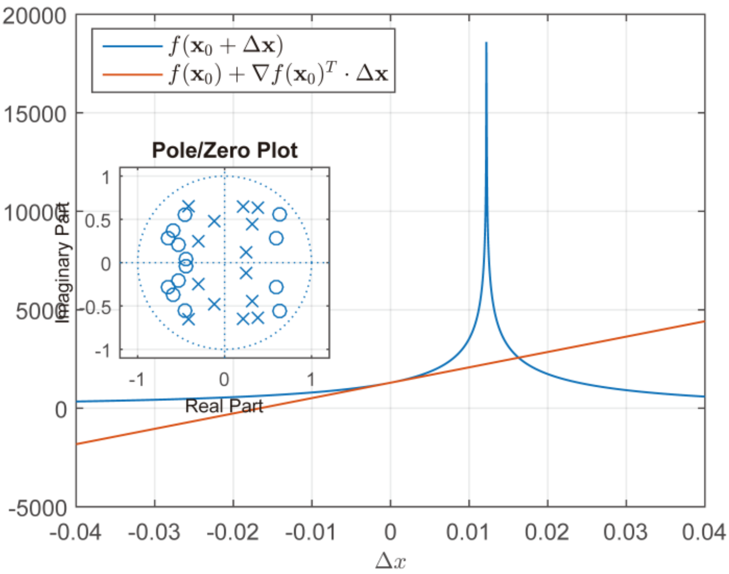
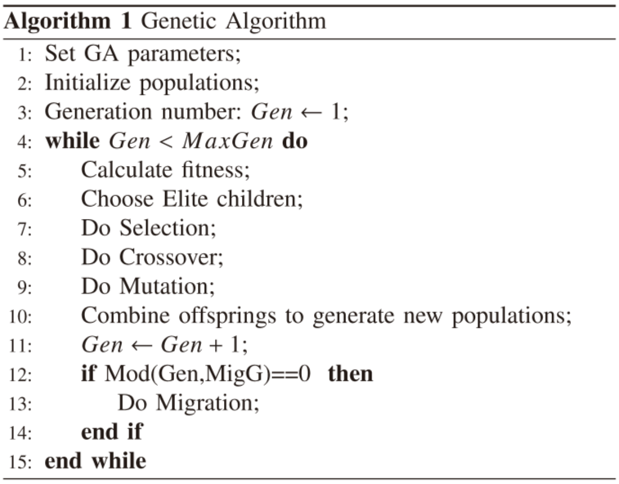


Figure 2. Non-convexity of the objective function. Here $\Delta\mathbf{x} = t\cdot\mathbf{1}$ where $\mathbf{1}$ is a 4K + 1 dimensional unit vector and t changes from −0.04 to 0.04. We see condition (11) does not hold in the neighbourhood of $\mathbf{x}_0$, meaning $f(\mathbf{x})$ is not convex. Here $\mathbf{x}_0$ = [−0.0122, 0.5939, 0.2823, 0.3822, −0.6351, −0.5309, 0.2066, −0.3018, 0.2474, −0.5919, −0.3708, 0.2478, 0.1198, −0.6504, −0.2830, −0.4147, −0.6512, −0.4563, −0.5530, 0.3174, −0.4442, −0.4444, −0.0413, 0.2127, −0.6475, 0.6329, −0.5572, −0.1189, 0.4793].



```
Algorithm 1 Genetic Algorithm
Set GA parameters;
Initialize populations;
Generation number: $Gen \leftarrow 1$;
while $Gen < MaxGen$ do
    Calculate fitness;
    Choose Elite children;
    Do Selection;
    Do Crossover;
    Do Mutation;
    Combine offsprings to generate new populations;
    $Gen \leftarrow Gen + 1$;
    if Mod(Gen,MigG)==0 then
        Do Migration;
    end if
end while
```

Figure 3. Pseudocodes describing GA.

Some terminologies of GA will be used in the subsequent content: A possible solution of the optimization problem is called an individual in GA. Each individual consists of a number of genes and each gene corresponds to an optimization variable. For example $\mathbf{x}$ in (7) is called an individual, and variables *A*, *zr*, *zi*, *pr* and *pi* in $\mathbf{x}$ are genes. Technical details of the pseudocodes will be given in the following.

3.2. Solving the Problem Using GA

GA parameters: Supposing that Doppler filter uses $K = 7$ cascaded SOSs, the referenced GA parameters are listed in.

Table 1. Generally, the larger *Nind* is, the greater probability that the global optimal solution may be contained in the population, so *Nind* should be configured big enough to keep a good diversity of the genes.

Initializing population: This step generates the initial population and each member must be feasible, meaning that individuals must satisfy constraints in (10). Firstly, generate genes *A*, *zr*, *zi*, *pr* and *pi* from uniform distribution in the range [−1, 1] as an individual.

$$gene = Uniform(-1,1) \tag{12}$$

Then checking the feasibility, add the individual into the initial population if its *poles* and *zeros* are all within the unit circle, otherwise discard. Repeat this process until *Nind* feasible individuals are created to form an initial subpopulation.

Calculating fitness: Fitness is used to quantitatively describe each individual's adaption to the environment. This article chooses the value of objective function as fitness and the value smaller represents better adaptation.

$$fitness = f(individual) \tag{13}$$

In this step, compute the objective function value of individuals in each subpopulation first, then sort all individuals in an ascend order according to their fitness values.

Table 1. Parameters of GA.

Variable	Value	Annotation
Nvar	29	Optimization variable number
Nind	300	Individual number in subpop
Nsub	20	Number of subpopulations
Relite	0.1	Ratio of Elite children
Rxov	0.7	Individual ratio to do Crossover
Rmut	0.2	Individual ratio to do Mutation
MigR	50	Generation gap to do Migration
MaxGen	10,000	Maximal generation number

Selecting elite individuals: Elite individuals refer to those who have the best adaptabilities in every subpopulation. They are preserved and passed to the next generation directly, which ensures that the performance of the new population will not degrade. This step selects *Nelite* individuals in each subpopulation, whose objective function values are smallest.

$$\begin{gathered} Nelite = Nind \cdot Relite \\ Nxov = Nind \cdot Rxov \\ Nmut = Nind \cdot Rmut \\ Nparent = 2 \cdot Nxov + Nmut \end{gathered} \tag{14}$$

Selection: Selection is to select *Nparent* individuals from each subpopulation to produce offspring. Individuals stronger will have more opportunities to be selected and mate, while those with poor adaptability will be eliminated gradually. All individuals selected will be divided into two parts, where the first part will do crossover operation, and another part will mutate. In crossover, $2 \cdot Nxov$ individuals need to be selected as parents because two parental individuals will produce one child. As for mutation, *Nmut* parents are needed because one single parent mutates to produce one child.

In this paper, a tournament selecting method [12] has been used. *S* (tournament size, e.g. 4) players are randomly chosen from each subpopulation in the tournament. A player is an individual and the one that has the best fitness value will be selected. Repeat this process until *Nparent* parental individuals are selected in each subpopulation.

Crossover: Crossover simulates the mating process of natural population. In the operation, parental individuals are paired first and each pair consists of a father and a mother. Then randomly select half of genes from the father, and the complementary half from the mother. Finally combine these two parts of genes together to produce a child.

$$child_{i,Gen+1} = \begin{cases} father_{i,Gen}, & rnd_i = 1; \\ mother_{i,Gen}, & rnd_i = 0; \end{cases} \tag{15}$$

The subscript $i = 1, \ldots, 4K + 1$ is the index of gene in the individual and *rnd* is a random integer generated from 0 and 1. Totally *Nxov* children in each subpopulation are need to be produced in this step.

Mutation: In a biotical population, gene mutation happens when passing genes from parents to children, which will produce new genes and increase the genetic diversity of the population. Gaussian mutation [13] is one kind of classical mutation methods in GA, where a random variable, generated from the Gaussian distribution (16) with zero mean and standard variance σ, will be superimposed on the original gene as a mutation variation.

$$\begin{gathered} mut(gene) = gene + Gaussian(\sigma) \\ \sigma = std(gene) \end{gathered} \tag{16}$$

At the beginning of the GA, each gene is dispersedly distributed in the entire feasible region. The variation magnitude can be relatively big at this moment to offer more possibilities for each gene. However, as the evolution progresses, each gene begins to converge and variation magnitude should be reduced accordingly. Because after evolution, gene values that were far away from their values, now have been eliminated already by natural selection. If the variation amplitude generated is too large, the gene newly mutated will still be eliminated with great probability in the subsequent evolution, which indicates that such mutation has no meaning.

This article chooses the standard variance of $\mathbf{x}$ to be σ of the Gaussian distribution. As the iteration progresses, $\mathbf{x}$ will gradually converge, σ tracking this tendency becomes smaller. Thus the variation magnitude will also decrease, which avoids invalid searching.

Combination: After mutation, all offspring have been produced and they will be combined to generate a new population, which consists of three parts. The first one is the elite individuals retained from the population of the last generation. The second one is the offspring created by crossover operation and the last part is the children generated by mutation, whose genes are mutated versions of their parents' genes.

Population newly generated will be delivered to calculate fitness again and the generation number *Gen* will be updated. So the evolution continues and optimization variables converge to the global or near global optimum solution gradually.

Migration: Migration occurs every *MigG* generations past promoting the communication between subpopulations. Elites of each subpopulation are gathered up first in this step, then they are randomly divided into *Nsub* parts, passed to each subpopulation to replace their worst fit individuals. Subpopulation and migration mechanism can efficiently improve the probability of getting the global optimal in GA.

4. Simulation Results

Simulating GA described above in Matlab, we get the Doppler filter. **Figure 4** shows the amplitude response of the designed Doppler filter. We see that it matches closely with the targeted response in passband, the transition band is very sharp and the response in stopband fluctuates slightly around

$SA = -60$ dB.

There are two criterions to measure the statistical accuracy of fading channel simulator. The first one is the correlation function. As an example, **Figure 5** shows the correlation characteristics of the generated fading process and we see that both auto-correlation and cross-correlation functions simulated match well with theoretical curves.

Another statistic characteristic of the fading channel is the LCR, which indicates the speed of the envelope of $h(t)$ passing through level Z in the downward direction. For Rayleigh fading channel, LCR satisfies (17), where $\lambda = Z / Z_{rms}$ is the signal level being crossed normalized with the rms of the fading process [1].

$$L_Z = \sqrt{2\pi} f_D \lambda e^{-\lambda^2} \tag{17}$$

Figure 6 shows the simulated LCRs, which are in good agreement with theo-

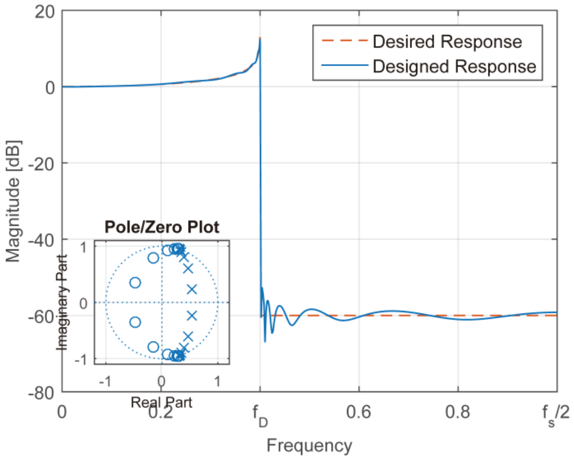


Figure 4. Amplitude response and pole-zero plot of the Doppler filter, with SOS number $K = 7$, Doppler rate $\rho = 0.2$ and stopband attenuation $SA = -60$ dB.

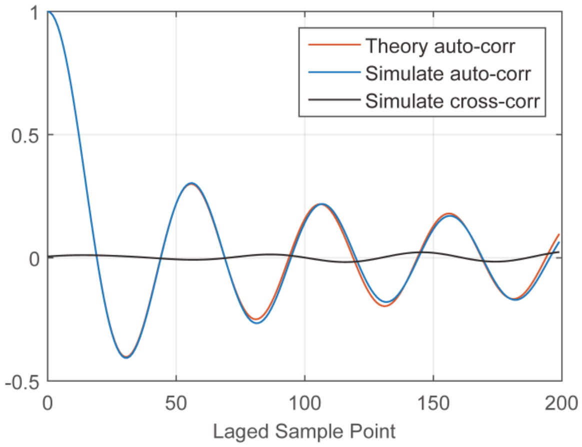


Figure 5. Correlation characteristics. The theoretical auto-correlation function of $h(t)$ is zero-order Bessel function and the cross-correlation function of $h_I(t)$ and $h_Q(t)$ is zero. We see that both auto-correlation and cross-correlation functions simulated match well with theoretical curves.

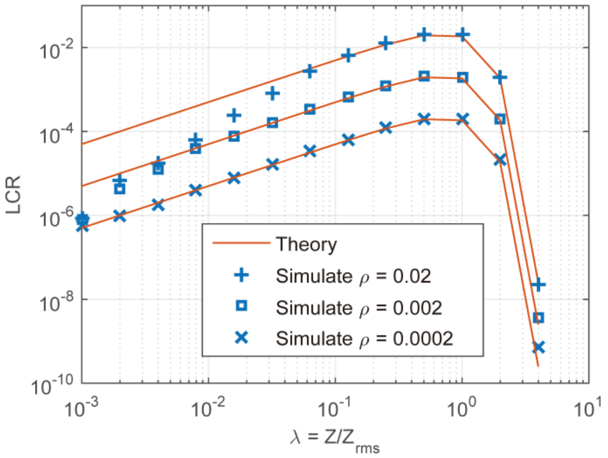


Figure 6. Level-crossing rate with interpolation factor $I = 10,100,1000$ respectively. The simulated results match well with theoretical ones, expect for two situations. The first one is when the channel changes extremely fast ($\rho = 0.02$) and another one is when λ is small ($Z < Z_{rms}/10$). However, [5] explained that these two situations and both caused by sparse sampling of the fading process, unrelated to the particular simulation techniques used.

retical ones, expect for two situations where the LCR values are underestimated. However, according to [5], these situations are common problems and unrelated to the particular simulation techniques used. We see that mismatches disappear as interpolation factor I increases.

5. Conclusion

This paper designs the channel simulator producing correlated Gaussian noises as the Rayleigh fading waveforms. An optimization model has been built to design the Doppler filter, and its non-convexity is proved with the aid of computer simulation. Genetic algorithm is used to solve the problem and it converges to the solution corresponding to very accurate approximation of Jakes PSD. Also statistical characteristics of the generated fading process match well with theoretical predictions, which confirms the validity of the simulator we designed.

References

[1] Goldsmith, A. (2005) Wireless Communications. Cambridge University Press, Cambridge, 71-82. https://doi.org/10.1017/cbo9780511841224

[2] Jakes, W.C. and Cox, D.C. (1994) Microwave Mobile Communications. Wiley IEEE Press. https://doi.org/10.1109/9780470545287

[3] Pop, M.F. and Beaulieu, N.C. (2001) Limitations of Sum-of-Sinusoids Fading Channel Simulators. *IEEE Transactions on Communications*, **49**, 699-708. https://doi.org/10.1109/26.917776

[4] Young, D.J. and Beaulieu, N.C. (2000) The Generation of Correlated Rayleigh Random Variates by Inverse Discrete Fourier Transform. *IEEE Transactions on Communications*, **48**, 1114-1127. http://doi.org/10.1109/26.855519

[5] Komninakis, C. (2003) A Fast and Accurate Rayleigh Fading Simulator. *GLOBECOM'03*, *IEEE Global Telecommunications Conference*, **6**, 3306-3310.

https://doi.org/10.1109/GLOCOM.2003.1258847

[6] Boyd, S. and Barratt, C. (2008) Ellipsoid Method. Notes for EE364B, Stanford University.

[7] Alimohammad, A., Fard, S.F. and Cockburn, B.F. (2012) Filter-Based Fading Channel Modeling. *Modelling and Simulation in Engineering*, **2012**, 3. https://doi.org/10.1155/2012/705078

[8] Alimohammad, A. and Fard, S.F. (2013) Fpga Implementation of Isotropic and Nonisotropic Fading Channels. *IEEE Transactions on Circuits and Systems II*: *Express Briefs*, **60**, 796-800. https://doi.org/10.1109/TCSII.2013.2281768

[9] Lu, W.-S. and Hinamoto, T. (2003) Optimal Design of Iir Digital Filters with Robust Stability Using Conic-Quadratic-Programming Updates. *IEEE Transactions on Signal Processing*, **51**, 1581-1592. https://doi.org/10.1109/TSP.2003.811229

[10] Boyd, S. and Vandenberghe, L. (2004) Convex Optimization. Cambridge University Press, 69-70. https://doi.org/10.1017/CBO9780511804441

[11] Chipperfield, A., Fleming, P., Pohlheim, H. and Fonseca, C. (1994) Genetic Algorithm Toolbox for Use with Matlab.

[12] Filipovic, V. (2012) Fine-Grained Tournament Selection Operator in Genetic' Algorithms *Computing and Informatics*, **22**, 143-161.

[13] Higashi, N. and Iba, H. (2003) Particle Swarm Optimization with Gaussian Mutation. *Proceedings of the* 2003 *IEEE Swarm Intelligence Symposium*, *SIS*'03, 72-79. https://doi.org/10.1109/SIS.2003.1202250