

## Article

# An Operation Reduction Using Fast Computation of an Iteration-Based Simulation Method with Microsimulation-Semi-Symbolic Analysis <sup>†</sup>

Vladimir Mladenovic <sup>1</sup> , Danijela Milosevic <sup>1</sup>, Miroslav Lutovac <sup>2</sup>, Yigang Cen <sup>3</sup> and Matjaz Debevc <sup>4,\*</sup>

<sup>1</sup> Faculty of Technical Sciences Cacak, University of Kragujevac, 32000 Cacak, Serbia; vladimir.mladenovic@ftn.kg.ac.rs (V.M.); danijela.milosevic@ftn.kg.ac.rs (D.M.)

<sup>2</sup> The School of Electrical Engineering and Computer Science of Applied Studies Belgrade, 11000 Belgrade, Serbia; lutovac@gmail.com

<sup>3</sup> School of Computer and Information Technology, Beijing Jiaotong University, 100044 Beijing, China; ygcen@bjtu.edu.cn

<sup>4</sup> The Faculty of Electrical Engineering and Computer Science, University of Maribor, 2000 Maribor, Slovenia

\* Correspondence: matjaz.debevc@um.si or matjaz.debevc@gmail.com; Tel.: +386-31-346-300

<sup>†</sup> This paper is an extended version of our paper published in *Studies in Informatics and Control* **2016**, *25*, 303–312, complemented by new results.

Received: 24 October 2017; Accepted: 11 January 2018; Published: 18 January 2018

**Abstract:** This paper presents a method for shortening the computation time and reducing the number of math operations required in complex calculations for the analysis, simulation, and design of processes and systems. The method is suitable for education and engineering applications. The efficacy of the method is illustrated with a case study of a complex wireless communication system. The computer algebra system (CAS) was applied to formulate hypotheses and define the joint probability density function of a certain modulation technique. This innovative method was used to prepare microsimulation-semi-symbolic analyses to fully specify the wireless system. The development of an iteration-based simulation method that provides closed form solutions is presented. Previously, expressions were solved using time-consuming numerical methods. Students can apply this method for performance analysis and to understand data transfer processes. Engineers and researchers may use the method to gain insight into the impact of the parameters necessary to properly transmit and detect information, unlike traditional numerical methods. This research contributes to this field by improving the ability to obtain closed form solutions of the probability density function, outage probability, and considerably improves time efficiency with shortened computation time and reducing the number of calculation operations.

**Keywords:** microsimulation-semi-symbolic analysis; iteration-based simulation method; Kummer's transformation

## 1. Introduction

In general, theoretical, experimental, and computational approaches are the basis for the study of observed phenomena. Every scientific and experimental result is expected to be placed into a function for its use, so the commercial use of products and services, and many engineering uses, emanate from a scientific approach that has been translated into an engineering approach.

Emerging developments are posing challenges in information technology [1–3] that include searching large databases [1,2], solving complex processes described by mathematical models, analyzing phenomena in communications in the information space, such as the transmission of wireless signals in urban environments [4–6], and the continuous high speed delivery of information without

stagnation in software engineering [7]. The common feature among all these issues is how to directly obtain results for further processing or exploitation. To address this challenge, complex mathematical tools are used to perform analyses and simulations of the performance of the observed processes and systems. Most often, a classical mathematical analysis does not provide final answers in closed form for complex phenomena, so special functions are used to obtain a solution. When results cannot be obtained with mathematical models, we use numerical methods. Most numerical tools include complex calculations, such as differential and integral equations and algebraic structures, using numerical mathematics algorithms such as Newton-Cotes, Romberg integration, Gauss-Christoffel, trapezium rule, Gauss formulas, etc. [8,9]. Due to this complexity, students may not understand the complete process or system, or cannot perform the method performance analysis to the end within the bounds of the classroom. Engineers and researchers may not have appropriate insight into the impact of the parameters necessary for the effective investigation or design. Additionally, the numerical computation generates a large amount of data, which may sometimes lead to erroneous results [10,11]. These incorrect results may be due to the finite word length in the records, or errors occurring during the shortening of numbers in fractions, for example. These methods do not provide the ability to manipulate with analytic expressions. These issues have been overcome by introducing a new method that treats variables and parameters as symbols. The method is called the iteration-based simulation method (IBSM) [11,12]. In addition to the IBSM that provides symbolic analysis, we allow the analysis to be partially observed through the concept of microsimulation analysis, which additionally enhances the ability to influence the parameters and variables. Also, we created an effective method with fast computation time that, together with the operation reduction, provides accurate results quickly. Computer algebra systems are important tools for these analyses, developments, and research, as they provide a completely new approach for understanding and solving complex cases. In this paper, we present the above methods directly applied to two examples. Both examples require complex analysis and use symbolic closed form expressions, and their numerical analysis is time-consuming.

The paper is organized as follows: in Section 2, the problem statement is described. Section 3 illustrates the complete methodology and procedure for applying the method, as well as examples with results in details. In Section 4, the collected results are discussed in detail.

## 2. Problem Statement

A large number of simulations do not guarantee that tolerances will not be exceeded. This is one of the numerous drawbacks of numerical-based tools. Our study had three goals. The first was to solve any analysis in closed form to allow further simplification and manipulation by using an iteration-based simulation method. The second goal was to develop an algorithm to quickly compute the method. Finally, we wanted to reduce the number of operations in the algorithm prior to its implementation. All phases of development and testing were observed by microsimulation-semi-symbolic analysis.

The IBSM was developed using the computer algebra system (CAS) to simplify complex algebraic expressions that offers an acceptable reduced analytic form for further manipulation or simulation as a closed form solution as previously published [12]. As integrals are present in the majority of the analyses, we approached the analyses with elementary calculating when the integrals are presented using Riemann sum. The method converts low-complexity implementation into a high-complexity structure. This approach allows implementation in the hardware environment.

The CAS performs symbolic mathematical operations and is used in the fields of mathematics and computer science. The CAS is based on algebraic calculations and manipulations performed using the same process as manual derivations. The CAS exclusively includes working with symbols, and numerical calculation is a special case for a CAS. Since symbols are used as variables, CAS deals with symbolic processing. Symbolic processing (SP) involves the development, implementation, and application of algorithms that manipulate and analyze mathematical expressions. CAS provides a deeper understanding and helps students to learn and engineers to simulate and design. The Wolfram language (WL) is the programming language suitable for CAS. WL has the ability to manipulate

symbolic expressions using a method similar to traditional manual derivation [13]. The WL is characterized by high-performance computing and the generation of compact and short program codes.

The goal of IBSM is to introduce a new parameter to obtain a closed form expression. Since the iteration is a new parameter, we used a transformation to change the integral into a sum, i.e., a series. For this purpose, we used Riemann sum transformation for the features of the improper integrals. Using this method, we obtained closed form expressions that can be manipulated and simplified, with a short computation time, while reducing the operations. The results were tested and verified. The general form of the Riemann integral transformation into a series is given as follows [14,15]:

$$\int_a^b f(x)dx = \lim_{\|\Delta x\| \rightarrow 0} \sum_{i=1}^n f(x_i) \Delta x_i \quad (1)$$

By observing the integrals in the previous session, we defined two types of Riemann sums: a single sum and a double sum. We first solved the single sum, then solved the double integrals. So, given Equation (1), the Wolfram language code is shown in Figure 1, where  $q$  is the value of the iteration in the defined transformation.

```
RiemannSumSingle[f_, {x1_, a1_, b1_}, q_] :=
  Sum[Evaluate[(f /. {x1 -> (a1 + (b1 - a1) k / q)}), {k, 1, q}], {k, 1, q}] (b1 - a1) / q
RiemannSumDouble[f_, {x2_, a2_, b2_}, {x3_, a3_, b3_}, q_] :=
  Sum[Evaluate[(f /. {x2 -> (a2 + (b2 - a2) k / q), x3 -> (a3 + (b3 - a3) k / q)}), {k, 1, q}], {k, 1, q}] (b2 - a2) / q
  (b3 - a3) / q
```

Figure 1. Wolfram language code for a Riemann sum.

Microsimulation mimics a complex phenomenon by describing its micro-components. Essentially, the system is left free to develop without too many constraints and simplifying assumptions [16]. However, when microsimulation is used with only symbolic content, and the particular numerical values are changed in the final stage, it is called microsimulation semi-symbolic analysis (MSSA). We observed each element of the symbolic calculation through MSSA, which provides faster and better testing and verification as well as a reduction in the operations [17,18]. MSSA also directly calculates in the first run without requiring more simulation attempts.

The next step was the development of an algorithm to allow fast computation. To achieve this, we treated the expression as a series. As a reminder, a short explanation of the concept of fast computation follows. A series is said to converge slowly if a large number of members of the series need to be added to determine a sum with the required accuracy. During the addition of series members using the member-by-member technique, the process automatically occurs and is interrupted when a selected criterion for error evaluation is fulfilled. Given the ultimate summation, the absolute value of the relationship between the last member and the calculated sum is most often used. This criterion is not always reliable, especially for the addition of a trigonometric series. An error caused by an interrupted summing is always higher than estimated. Conversely, contemporary computing machines can quickly add a large number of members in the series. However, due to the limitation on the format of the records in the registers, a certain number of decimal places are eliminated, which leads to the accumulation of errors and to completely absurd results in the process of summing. Therefore, procedures exist for accelerating the convergence of a series, such as Kummer, Aitken, Cesar, and Euler. This paper presents an effective method for accelerating the convergence of a series based on Kummer's transformation.

We adhered to two theorems. The first states that if  $\sum_{k=1}^n a_k$  converges, then  $\lim_{k \rightarrow \infty} a_k = 0$ . The second states that if  $\sum_{k=1}^n a_k$  and  $\sum_{k=1}^n b_k$  are positive series, and if  $\lim_{k \rightarrow \infty} \frac{a_k}{b_k} = \rho$ , ( $b_k \neq 0$ ), then

convergence or divergence occur simultaneously. Kummer's transformation, better known as Kummer's acceleration method, accelerates the convergence of many series. The method subtracts from a given convergent series  $\sum a_k$ , and another equivalent series  $\sum b_k$ , whose sum  $C = \sum_{k \geq 0} b_k$  is well known and finite. Kummer's transformation is described as:

$$\sum_{k=0}^{\infty} a_k = \rho \sum_{k=0}^{\infty} b_k + \sum_{k=0}^{\infty} \left(1 - \rho \frac{b_k}{a_k}\right) a_k = \rho C + \sum_{k=0}^{\infty} \left(1 - \rho \frac{b_k}{a_k}\right) a_k \quad (2)$$

The convergence of the right hand side of Equation (2) is faster because  $1 - \rho \cdot b_k/a_k$  tends to 0 as  $k$  tends to infinity [19]. The complete procedure is shown in Figure 2.

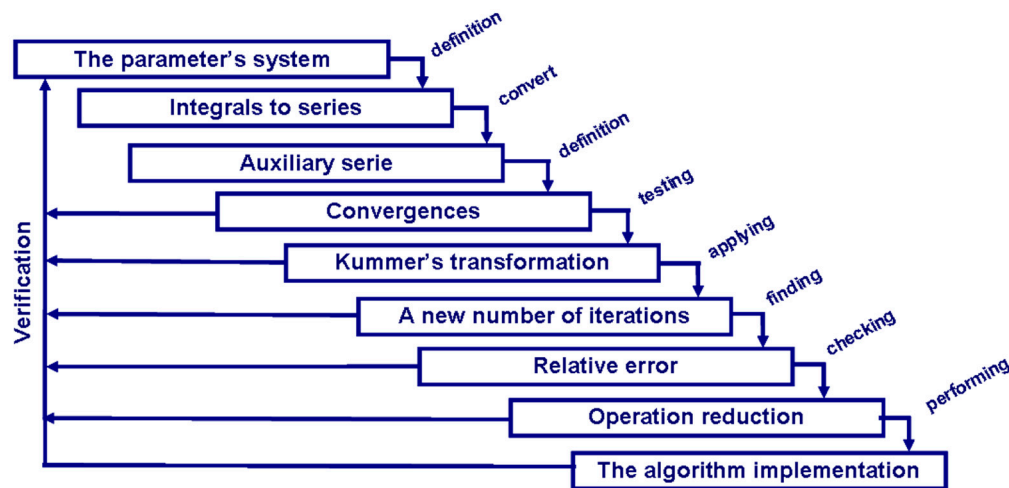


Figure 2. Steps of the speeding up and operation reduction process.

The reduction in operations was performed by counting all math operations and functions contained in the final expressions. Wolfram language allows the performance of direct counting. Mathematical operations and functions in WL can be viewed both symbolically and as commands. Operations are recognized using the FullForm command, and the counting is performed using the StringPosition command. Since we had sums where the numbers are repeated  $q$  times, the WL code for completely counting the operations is:

```
InnerOperations=q*FullForm[aK[z,q]];
StringPosition[InnerOperations, {"Times","Power","Plus","Rational", "BesselI", "Log","Exp"}];
```

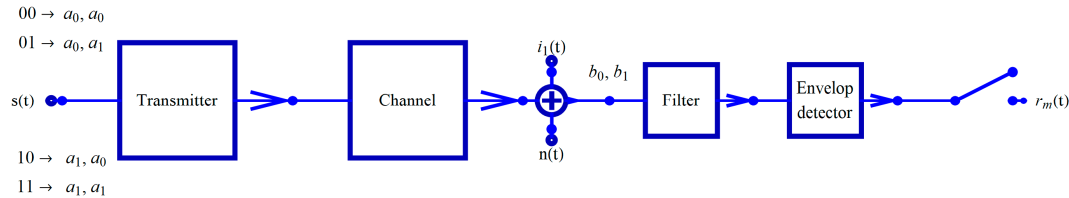
The orders of Times, Plus, BesselI, Log, and Exp are functions used in close form expressions. Similarly, substituting  $s$  for  $aK[z,q]$ , we obtained the number of operations in the accelerated algorithm.

### 3. Applications of the Accelerating Procedure and Operation Reduction with Microsimulation Semi-Symbolic Analysis

In this section, the operation reduction using fast computation of an iteration-based simulation method with microsimulation-semi-symbolic analysis was applied to two processing problems to illustrate the shorter computation time of the algorithm, and to demonstrate the variety of applications for which the operation may be used. A case with complex calculation is illustrated in the example with non-coherent Amplitude-Shift Keying (ASK) with shadowing, interference, and correlated noise. The second example treats second-order statistics in the SC macrodiversity system operating over Gamma shadowed Nakagami- $m$  fading channels [20].

### 3.1. Non-Coherent Amplitude Shift Keying (ASK) with Shadowing, Interference, and Correlated Noise

Non-coherent ASK is a modulation scheme used to send digital information between digital equipment and it is shown on Figure 3. Similar part of the system, where real-time estimation is needed, can be found in [21]. The data is transmitted by the non-coherent system without a carrier in a binary manner.



**Figure 3.** Non-coherent amplitude shift keying (ASK) system with interference  $i_1(t)$ .

Shadowing with interference is one of the most common models used in wireless communications to describe the phenomenon of multiple scattering [21–24]. The basic components of the system are shown in Figure 1. Both shadowing and interference cause strong fluctuations in the amplitude of the useful signal. This occurs in urban areas and is described as a log-normal distribution. In our analysis, we performed an outage probability. Transmitting signals using two symbols were observed in the non-coherent ASK system [25,26]. The noise, as a narrow-band stochastic process, is correlated and the coefficient of correlation is denoted by  $R$  ( $R \neq 1$ ). Mathematically, the noise can be described as  $n_i(t) = x_i(t) \cdot \cos(\omega t) - y_i(t) \cdot \sin(\omega t)$ . The receiver is sheltered, and no optical visibility exists toward the transmitter, but interference  $i_1(t) = A_1 \cdot \cos(\omega t)$  is present. If the system sends logical zero, then the signal  $s_0(t) = a_0 \cdot \cos(\omega t)$  has been sent, but if the system sends a logical unit, then the signal  $s_1(t) = a_1 \cdot \cos(\omega t)$  has been sent. The parameters  $a_0$  and  $a_1$  are the signal elements from which the code words are formed. The receiver detects information signal  $b_0 \cdot \cos(\omega t)$  and  $b_1 \cdot \cos(\omega t)$  with envelopes  $z_0$  and  $z_1$  after passing through a transmitting channel. The  $b_m$  ( $m = 0, 1$ ) are the elements of the detected signals. The receiver system includes a filter and detector envelope. In the receiver input, the signal is:

$$r_m(t) = b_m \cdot \cos(\omega t) + A_1 \cdot \cos(\omega t) + x_m \cdot \cos(\omega t) - y_m \cdot \sin(\omega t) = z_m \cdot \cos(\omega t + \phi_m), m = 0, 1 \quad (3)$$

with envelopes  $z_0$  and  $z_1$ , and phases  $\phi_0$  and  $\phi_1$ , respectively.

The general form of the condition joint probability density function is:

$$p(x_0, x_1, y_0, y_1) = \frac{1}{4\pi^2\sigma^4\sqrt{1-R^2}} \exp\left\{-\frac{x_0^2 + x_1^2 + y_0^2 + y_1^2 - 2R(x_0x_1 + y_0y_1)}{2\sigma^2(1-R^2)}\right\} \quad (4)$$

where  $R$  is the coefficient of correlation and  $\sigma$  is variance. To ensure the set of expressions is solved continuously, using the polar coordinates is necessary, as follows:

$$\begin{aligned} x_0 &= z_0 \cos \phi_0 - b_0 - A_1 \\ y_0 &= -z_0 \sin \phi_0 \\ x_1 &= z_1 \cos \phi_1 - b_1 - A_1 \\ y_1 &= -z_1 \sin \phi_1 \end{aligned} \quad (5)$$

The next step was determining the condition joint probability density function (JPDF). Substituting Equation (5) into Equation (4), we obtained:

$$p(r_0, r_1/b_0, b_1, \phi_0, \phi_1, A_1) = p(x_0, y_0, x_1, y_1) \cdot |J| \quad (6)$$

where  $|J|$  is Jacobian. A joint probability density function has a log-normal distribution described as [22]:

$$p_{ij}(b_0, b_1 / A_1) = \frac{1}{2\pi\sigma^2(b_0 + A_1)(b_1 + A_1)\sqrt{1-R^2}} \times \\ \times \exp\left\{-\frac{(10\log(b_0 + A_1) - a_i)^2 + (10\log(b_1 + A_1) - a_j)^2}{2\sigma^2(1-R^2)}\right\} \\ \times \exp\left\{\frac{2R(10\log(b_0 + A_1) - a_i)(10\log(b_1 + A_1) - a_j)}{2\sigma^2(1-R^2)}\right\} \quad (7)$$

For  $i = j = 0$ , the code word 00 was sent; for  $i = 1$  and  $j = 0$ , the code word 01 was sent; for  $i = 0$  and  $j = 1$ , the code word 10 was sent; and for  $i = 1$  and  $j = 1$ , the code word 11 was sent. So:

$$p(z_0, z_1 / b_0, b_1, \varphi_0, \varphi_1, A_1) = \frac{z_0 z_1}{(2\pi)^2 \sigma^4 \sqrt{1-R^2}} \times \\ \times \exp\left\{-\frac{z_0^2 + z_1^2 - 2R(b_0 + A_1)(b_1 + A_1) - 2Rz_0 z_1 \cos(\varphi_0 - \varphi_1)}{2\sigma^2(1-R^2)}\right\} \times \\ \times \exp\left\{\frac{2(1+R)(b_0 + b_1 + 2A_1)(z_0 \cos \varphi_0 + z_1 \cos \varphi_1)}{2\sigma^2(1-R^2)}\right\} \quad (8)$$

The last expression can be transformed using a modified Bessel function [27] before derivation of the closed form expression:

$$e^{x \cos \alpha} = \sum_{n=-\infty}^{\infty} I_n(x) \cos n\alpha \quad (9)$$

and applying trigonometric transformation:

$$\cos \alpha \cos \beta \cos \gamma = \frac{1}{4} [\cos(\alpha + \beta + \gamma) + \cos(\alpha - \beta + \gamma) + \cos(\alpha + \beta - \gamma) + \cos(\alpha - \beta - \gamma)] \quad (10)$$

Equation (8) becomes:

$$p(z_0, z_1 / b_0, b_1, \varphi_0, \varphi_1, A_1) = \\ = C \cdot e^{C_0} \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} I_n(C_1) I_m(C_2) I_k(C_3) \times \cos[n\varphi_0] \cos[m\varphi_1] \cos[k(\varphi_0 - \varphi_1)] \quad (11)$$

where:

$$C = \frac{z_0 z_1}{(2\pi)^2 \sigma^4 \sqrt{1-R^2}} \\ C_0 = -\frac{z_0^2 + z_1^2 - 2R(b_0 + A_1)(b_1 + A_1)}{2\sigma^2(1-R^2)} \\ C_1 = \frac{(b_0 + A_1)(b_1 + A_1)}{\sigma^2(1-R)} z_0 \\ C_2 = \frac{(b_0 + A_1)(b_1 + A_1)}{\sigma^2(1-R)} z_1 \\ C_3 = \frac{R \cdot z_0 z_1}{\sigma^2(1-R^2)} \quad (12)$$

Using the Bessel identity  $I_n(x) = I_{-n}(x)$ , it follows that:

$$p(z_0, z_1 / b_0, b_1, A_1) = 2\pi^2 C \cdot e^{C_0(b_0, b_1, A_1)} \times \sum_{n=0}^{\infty} I_n[C_1(b_0, b_1, A_1)] \cdot I_n[C_2(b_0, b_1, A_1)] \cdot I_n[C_3] \quad (13)$$

The present interference is described with the Rayleigh distribution over the probability density function (PDF) [23,24] as:

$$p(A_1) = \frac{A_1}{\sigma^2} \exp\left\{-\frac{A_1^2}{2\sigma^2}\right\}; 0 \leq A_1 \leq \infty \quad (14)$$

To eliminate the interference, performing averaging is necessary for all values of interference  $A_1$ .

$$p_{ij}(z_0, z_1/b_0, b_1) = \int_0^\infty p(z_0, z_1/b_0, b_1, A_1) \cdot p(A_1) dA_1 \quad (15)$$

The integral in Equation (15) is solved using integral:

$$\int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \cos n\varphi_0 \cos m\varphi_1 \cos k(\varphi_0 - \varphi_1) d\varphi_0 d\varphi_1 = \begin{cases} \pi^2, & |n| = |m| = |k| \\ 0, & n \neq m \neq k \end{cases} \quad (16)$$

The distribution is obtained by averaging  $\varphi_0$  and  $\varphi_1$  for all values between  $-\pi$  and  $\pi$ .

$$p(z_0, z_1/b_0, b_1) = \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} p(z_0, z_1/b_0, b_1, \varphi_0, \varphi_1) d\varphi_0 d\varphi_1 \quad (17)$$

For all code word combinations, distributions of envelopes are obtained by integrating all values between  $b_0$  and  $b_1$ . So, when the code word  $|ij|$  ( $i = 0, 1; j = 0, 1$ ) has been sent, marked with  $H_i H_j$  in Equation (18), and when the same is detected in the input of the receiver marked with  $D_i D_j$ , the detection of the signals is described as:

$$P(D_i D_j / H_i H_j) = p_{ij}(z_0, z_1) = \int_0^\infty \int_0^\infty p(z_0, z_1/b_0, b_1) \cdot p_{ij}(b_0, b_1) db_0 db_1 \quad (18)$$

The outage probability is:

$$P_{outage} = 1 - \sum_{i=0}^1 \sum_{j=0}^1 P(H_i H_j) P(D_i D_j / H_i H_j) \quad (19)$$

where  $P(H_i H_j) = P(H_i) \cdot P(H_j) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$ ,  $i = 0, 1$ , and  $j = 0, 1$ . For the outage probability, Equation (19) represents closed form expression and is often not present in the closed form solution. Closed form expression represents an implicit solution that is contained in a mathematical expression [12]. A closed form solution provides a solved problem in terms of functions and mathematical operations from a given and generally-accepted set [28]. In other words, a closed form solution provides an explicit solution to an observed problem, whereas closed form expression shows an implicit or insufficient solution.

From Equation (7), the joint probability density function is shown in Figure 4.

$$\begin{aligned} \text{JPDF}[x_-, y_-, s_-, R_-, A1_-] := & \frac{1}{2 * \pi * (s)^2 * (b0 + A1) * (b1 + A1) * \sqrt{1 - R^2}} * \\ & \text{Exp}\left[-\frac{(10 * \text{Log}[10, (b0 + A1)] - x)^2 + (10 * \text{Log}[10, (b1 + A1)] - y)^2}{2 * (s)^2 * (1 - R^2)}\right] * \\ & \text{Exp}\left[\frac{2 * R * (10 * \text{Log}[10, (b0 + A1)] - x) * (10 * \text{Log}[10, (b1 + A1)] - y)}{2 * (s)^2 * (1 - R^2)}\right] \end{aligned}$$

**Figure 4.** Condition joint probability density function using Wolfram language for shadowing and interference.

Figure 5 shows the manipulating of Equation (8) and substituting into Equation (12) for the changing of coefficients for simplification.



$$\begin{aligned}
C_c &= \frac{z_0 * z_1}{(2 \pi)^2 (s)^4 \sqrt{1 - R^2}}; \\
C_0 &= -\frac{(z_0)^2 + (z_1)^2 - 2 R * (b_0 + A_1) * (b_1 + A_1)}{2 (s)^2 (1 - R^2)}; \\
C_1 &= \frac{(b_0 + A_1) * (b_1 + A_1)}{(s)^2 (1 - R)} z_0; \\
C_2 &= \frac{(b_0 + A_1) * (b_1 + A_1)}{(s)^2 (1 - R)} z_1; \\
C_3 &= \frac{R}{(s)^2 (1 - R^2)} z_0 * z_1;
\end{aligned}$$

Figure 5. Changing of coefficients for simplification.

Interference  $A_1$  is present per Equation (14) and described by Wolfram language code in Figure 6,

$$pARayleight[A_-, s_-] := \frac{A}{(s)^2} * \text{Exp}\left[-\frac{A^2}{2 (s)^2}\right]$$

Figure 6. Rayleigh distribution for interference coded by Wolfram language.

Averaging all  $A_1$  values is necessary, according to Equation (15). The general form of the condition joint probability density function is defined in Equation (14), and is described in Figure 7.

$$\begin{aligned}
&\text{LogNormalP}[v_-, s_-, R_-, A1_-] := \\
&2 \pi^2 * C_c * \text{Exp}[C_0] * \sum_{k=1}^v (\text{BesselI}[k, C_1] * \text{BesselI}[k, C_2] * \text{BesselI}[k, C_3])
\end{aligned}$$

Figure 7. Log-normal distribution for non-coherent ASK in the presence of shadowing and interference.

$s$  is marked variance  $\sigma$ ,  $R$  is the correlation coefficient, and  $v$  is the order of the iterations. The finalization of IBSM obtains closed form expressions of the probability density function, and outage probability in term of iterations (Figure 8).

$$\begin{aligned}
&\frac{1}{q^2} (h_0 - h_1)^2 \\
&\sum_{k=1}^q \frac{1}{q} (h_1 - h_0) \sum_{k=1}^q \left( z_0 z_1 \left( \frac{(h_1 - h_0)k}{q} + h_0 \right) I_1 \left( \frac{R z_0 z_1}{(1 - R^2) \sigma^2} \right) I_1 \left( \frac{\left( 2 h_0 + \frac{2k(h_1 - h_0)}{q} \right)^2 z_0}{(1 - R) \sigma^2} \right) I_1 \left( \frac{\left( 2 h_0 + \frac{2k(h_1 - h_0)}{q} \right)^2 z_1}{(1 - R) \sigma^2} \right) \right. \\
&\exp \left[ -\frac{\left( \frac{10 \log \left( \frac{2(h_1 - h_0)k}{q} + 2 h_0 \right)}{\log(10)} - a_0 \right)^2 + \left( \frac{10 \log \left( \frac{2(h_1 - h_0)k}{q} + 2 h_0 \right)}{\log(10)} - a_1 \right)^2}{2 (1 - R^2) \sigma^2} + \right. \\
&\frac{R \left( \frac{10 \log \left( \frac{2(h_1 - h_0)k}{q} + 2 h_0 \right)}{\log(10)} - a_0 \right) \left( \frac{10 \log \left( \frac{2(h_1 - h_0)k}{q} + 2 h_0 \right)}{\log(10)} - a_1 \right)}{(1 - R^2) \sigma^2} - \frac{2 R \left( \frac{2(h_1 - h_0)k}{q} + 2 h_0 \right)^2 + z_0^2 + z_1^2}{2 (1 - R^2) \sigma^2} - \\
&\left. \left. \frac{\left( \frac{(h_1 - h_0)k}{q} + h_0 \right)^2}{2 \sigma^2} \right] \right) / \left( 4 \pi (1 - R^2) \sigma^8 \left( \frac{2(h_1 - h_0)k}{q} + 2 h_0 \right)^2 \right)
\end{aligned}$$

Figure 8. Closed form solution of probability density function ( $PDF_{outage}$ ) of a non-coherent ASK system.



The closed form of  $PDF_{outage}$  in Figure 8 provides the next parameters: iteration  $q$ ,  $h_0$ , and  $h_1$  are the resolution of the iteration,  $z_0$  and  $z_1$  are envelopes,  $R$  is the coefficient of correlation, and  $\sigma$  is variance. This expression cannot be manually obtained by using numerical tools. The resultant closed form solution of  $P_{outage}$  is an expression that is ready for further processing. Accordingly, the viewpoint is an insight into the parameters and variables that participate in obtaining all the features of this case study. Drawing the characteristics is now possible, but this calculation would take too long, regardless of the chosen accuracy. On the other hand, for greater accuracy, a number of iterations is required, which is not beneficial for this form of expression.

Finally, the closed form solution of  $P_{outage}$  is shown in Figure 9.

$$\begin{aligned} & \left( \frac{1}{q^2} \sum_{k=1}^q \frac{\sum_{k=1}^q \left[ q z^2 I_1 \left( \frac{R z^2}{(1-R^2)\sigma^2} \right) I_1 \left( \frac{4 k^2 z}{q^2 (1-R)\sigma^2} \right)^2 \exp \left( -\frac{2 z^2 - \frac{8 k^2 R}{q^2}}{2(1-R^2)\sigma^2} - \frac{k^2}{2 q^2 \sigma^2} + \frac{R \left( \frac{10 \log(\frac{2k}{q})}{\log(10)} - 1 \right)^2}{(1-R^2)\sigma^2} - \frac{\left( \frac{10 \log(\frac{2k}{q})}{\log(10)} - 1 \right)^2}{(1-R^2)\sigma^2} \right)}{16 \pi k (1-R^2) \sigma^8} \right)}{q} + \right. \\ & \left. \sum_{k=1}^q \frac{\sum_{k=1}^q \left[ q z^2 I_1 \left( \frac{R z^2}{(1-R^2)\sigma^2} \right) I_1 \left( \frac{4 k^2 z}{q^2 (1-R)\sigma^2} \right)^2 \exp \left( -\frac{2 z^2 - \frac{8 k^2 R}{q^2}}{2(1-R^2)\sigma^2} - \frac{k^2}{2 q^2 \sigma^2} + \frac{R \left( \frac{10 \log(\frac{2k}{q})}{\log(10)} + 1 \right)^2}{(1-R^2)\sigma^2} - \frac{\left( \frac{10 \log(\frac{2k}{q})}{\log(10)} + 1 \right)^2}{(1-R^2)\sigma^2} \right)}{16 \pi k (1-R^2) \sigma^8} \right)}{q} + \right. \\ & \left. 2 \sum_{k=1}^q \frac{1}{q} \left( \sum_{k=1}^q \frac{1}{16 \pi k (1-R^2) \sigma^8} q z^2 I_1 \left( \frac{R z^2}{(1-R^2)\sigma^2} \right) I_1 \left( \frac{4 k^2 z}{q^2 (1-R)\sigma^2} \right)^2 \exp \left( -\frac{2 z^2 - \frac{8 k^2 R}{q^2}}{2(1-R^2)\sigma^2} - \right. \right. \right. \\ & \left. \left. \left. \frac{k^2}{2 q^2 \sigma^2} + \frac{R \left( \frac{10 \log(\frac{2k}{q})}{\log(10)} - 1 \right) \left( \frac{10 \log(\frac{2k}{q})}{\log(10)} + 1 \right)}{(1-R^2)\sigma^2} - \frac{\left( \frac{10 \log(\frac{2k}{q})}{\log(10)} - 1 \right)^2 + \left( \frac{10 \log(\frac{2k}{q})}{\log(10)} + 1 \right)^2}{2(1-R^2)\sigma^2} \right) \right) \right) \right) \end{aligned}$$

**Figure 9.** Closed form solution of outage probability  $P_{outage}$  of a non-coherent ASK system with shadowing and interference.

In our case, a member  $a_k$  represents a general member of the series in  $P_{outage}$ , from the closed form solution in Figure 9.

Convergence testing of the  $a_k$  verified that:

$$\begin{aligned} \lim_{k \rightarrow q} a_k &= 0 \\ \lim_{q \rightarrow \infty} a_k &= 0 \end{aligned} \quad (20)$$

Convergence testing was performed with assumptions that  $0 \leq R < 1$ ,  $\sigma > 0$ ,  $z \geq 0$ , and  $q \geq 1$ .

The selection of the auxiliary function is one of the most important aspects of the MSSA [29]. In testing many series, the authors of this paper highlighted the series that shows the best performance to accelerate convergence, meaning a shorter computation time with the optimum number of iteration. Comparative analysis of different auxiliary series can be the subject of particular surveys, and the reader(s) are encouraged to do so. Therefore, in our case, the auxiliary series is:

$$C = \sum_{k=1}^{\infty} \frac{1}{k^5 2^{k-1}} \quad (21)$$

The series converges to  $2\log 2$ . To fully use Equation (2), we made a minor modification to the member  $b_k$ , with respect to the convergence theorems that have been mentioned above. The new member becomes  $b_k \rightarrow a_k + c_k$ , so:

$$s = \sum_{k=0}^{\infty} a_k = \rho \sum_{k=0}^{\infty} a_k + \sum_{k=0}^{\infty} \left(1 - \rho \frac{a_k + c_k}{a_k}\right) a_k = \rho C + \sum_{k=0}^{\infty} \left(1 - \rho \frac{a_k + c_k}{a_k}\right) a_k \quad (22)$$

where  $c_k$  is general term in Equation (21). We obtain the general member of  $P_{outage}$  marked as  $a_k$  in Figure 10, separating it from Figure 9. Following the next step in MSSA, we derived the term  $\rho$  (Figure 11).

$$\begin{aligned} & -\frac{1}{\pi q^2 (R^2 - 1) \sigma^8} z^2 2^{-\frac{20}{(R+1)\sigma^2 \log(10)} - 2} I_1\left(\frac{R z^2}{(1 - R^2) \sigma^2}\right) \\ & \left( e^{\frac{2R}{(R^2-1)\sigma^2}} 2^{\frac{20}{(R+1)\sigma^2 \log(10)} + 1} \left(\frac{k}{q}\right)^{\frac{20}{(R+1)\sigma^2 \log(10)}} + 2^{\frac{40}{(R+1)\sigma^2 \log(10)}} \left(\frac{k}{q}\right)^{\frac{40}{(R+1)\sigma^2 \log(10)} + 1} \right) \\ & \left(\frac{k}{q}\right)^{-\frac{20}{(R+1)\sigma^2 \log(10)} - 1} I_1\left(\frac{k^4 z}{q^4 (1 - R) \sigma^2}\right)^2 \exp\left(\frac{\frac{2q^2(-R+z^2+1)-k^2(R^2+8R-1)}{q^2(R-1)} - \frac{200 \log^2\left(\frac{2k}{q}\right)}{\log^2(10)}}{2(R+1)\sigma^2}\right) \end{aligned}$$

Figure 10. General term in a series of  $P_{outage}$  marked as  $a_k$ .

$$\begin{aligned} & \left( 2^q z^2 \left( \exp\left(\frac{R(100 \log^2(2) + 7 \log^2(10) + \log^2(\frac{512}{5})) + \log^2(10240)}{(R^2 - 1) \sigma^2 \log^2(10)}\right) + \right. \right. \\ & \quad \exp\left(\frac{R(100 \log^2(2) + 7 \log^2(10) + \log^2(10240)) + \log^2(\frac{512}{5})}{(R^2 - 1) \sigma^2 \log^2(10)}\right) + \\ & \quad \left. \left. 2 \exp\left(\frac{R(8 \log^2(10) + \log^2(\frac{512}{5}) + \log^2(10240)) + \log^2(10) + 100 \log^2(2)}{(R^2 - 1) \sigma^2 \log^2(10)}\right) \right) \right) \\ & I_1\left(\frac{z}{(1 - R) \sigma^2}\right)^2 I_1\left(\frac{R z^2}{(1 - R^2) \sigma^2}\right) \exp\left(\frac{(2 z^2 + 1) \log^2(10) - 2 R (100 \log^2(2) + 7 \log^2(10))}{2 (R^2 - 1) \sigma^2 \log^2(10)}\right) \Bigg/ \\ & \left( 2^q z^2 e^{\frac{2z^2+1}{(R^2-1)\sigma^2}} \left( \exp\left(\frac{R \log^2(\frac{512}{5}) + \log^2(10240)}{(R^2 - 1) \sigma^2 \log^2(10)}\right) + \exp\left(\frac{R \log^2(10240) + \log^2(\frac{512}{5})}{(R^2 - 1) \sigma^2 \log^2(10)}\right) + \right. \right. \\ & \quad \left. \left. 2 \exp\left(\frac{R(-100 \log^2(2) + \log^2(10) + \log^2(\frac{512}{5}) + \log^2(10240)) + \log^2(10) + 100 \log^2(2)}{(R^2 - 1) \sigma^2 \log^2(10)}\right) \right) \right) \\ & I_1\left(\frac{z}{(1 - R) \sigma^2}\right)^2 I_1\left(\frac{R z^2}{(1 - R^2) \sigma^2}\right) - 8 \pi q (R^2 - 1) \sigma^8 \\ & \exp\left(\frac{R(R \log^2(10) + 2(4 \log^2(10) + \log^2(\frac{512}{5}) + \log^2(10240)))}{2(R^2 - 1) \sigma^2 \log^2(10)}\right) \Bigg) \end{aligned}$$

Figure 11. The element Kummer's transformation  $\rho$ .

We checked that the value  $\rho$  tends to 1 after convergence testing. The quicker computation was performed by assuming how much iteration is required to calculate the outage probability  $P_{outage}$  obtained by the IBSM. Otherwise, a large number of iterations are required to calculate the closest exact values of  $P_{outage}$ , but the computation is time consuming. Then, the resulting  $P_{outage}$  equalizes with a new series obtained by the Kummer's transformation, and performs point matching for the various values of the envelopes, followed by a new reduced number of iterations. After that, the verification of the obtained results was performed by checking the relative error, which determined the degree of adjustability of the algorithm [29]. Finally, we checked the number of operations of calculations in the expression in Figure 9, and then obtained a reduced number of operations with a new decreased number of iterations.

After all symbolic derivations, we used closed form solutions to directly obtain results in the first attempt. To obtain concrete numerical results, we needed to set the initial parameters. We supposed that the closest exact value was obtained after 500 iterations by using the outage probability  $P_{outage}$  in Figure 9, and the resolution of the iteration was  $h_0 = 0$  and  $h_1 = 1$ . We also used  $z_0 = z_1 = z$  to simplify the analysis. The next step was calculating the new numbers of iterations that are reduced for various values of the envelope  $z$ . This was performed using the command `FindRoot[s==Poutage,{q,1}]`.  $s$  is a new expression obtained by Kummer's transformation in Equation (22), and  $P_{outage}$  is a closed form solution in Figure 8. We took the range of values  $z = \{1, 15\}$  for a concrete case [29]. Experiments were performed for various values of the coefficient of correlation  $R$  ( $R = 7/10, 8/10$ ) and the variance  $\sigma$  ( $\sigma = 2, 3$ ). All calculations were performed with a precision of  $10^{-6}$ . All tests were performed on a PC with: Intel® Core™ i5-6500 CPU@ 3.2 GHz, 8 GB RAM, 64-bit Operating System, Windows 10, and Mathematica Wolfram 11.1. The reduced number of iterations are shown in Table 1.

**Table 1.** Reduced number of iterations.

$z$	$q_1$ $R = 7/10; \sigma = 2$	$q_2$ $R = 7/10; \sigma = 3$	$q_3$ $R = 8/10; \sigma = 2$	$q_4$ $R = 8/10; \sigma = 3$
1	25	32	21	30
2	20	27	16	25
3	19	25	14	22
4	18	24	12	21
5	17	23	11	20
6	17	23	10	20
7	18	23	9	19
8	18	23	9	19
9	20	24	9	20
10	21	25	9	19
11	23	26	10	20
12	27	27	11	21
13	29	28	12	21
14	32	30	14	22
15	36	31	16	23

In Figure 12, the changing iteration values in terms of the envelope  $z$  are shown for the accelerated algorithm. Notably, the reduced number of iterations is not the same for each envelope value. The minimum number of iterations is  $z = 10$ , where the value provides a true detection. However, the number of iterations is in range of 9 to 35 if we observe the total range of the envelope, which is a significant reduction compared to the original 500 iterations.

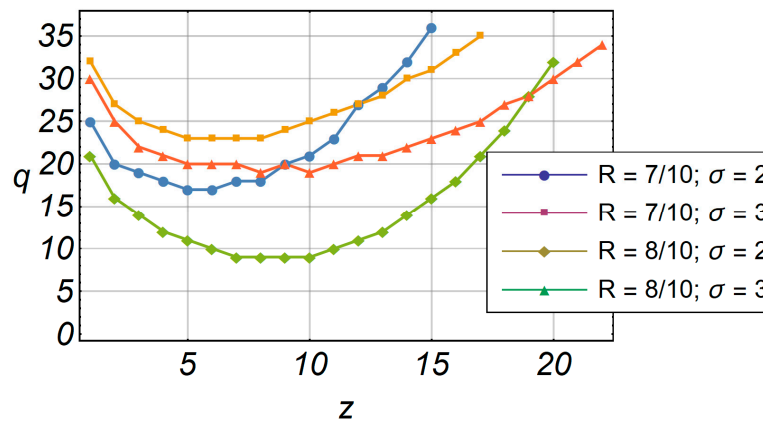


Figure 12. The number of iterations in term of envelope  $z$ .

Since the absolute error is not precisely characterized by accuracy, the relative error is used as:

$$\delta = \frac{s - P_{outage}}{P_{outage}} \quad (23)$$

Relative errors do not exceed more than 10% of the value as it is shown in Figure 13. This indicates that the algorithm is quite accurate. In Figure 14, the comparative characteristics of  $P_{outage}$  and  $s$  are shown. The accelerated algorithm  $s$  is marked as  $P_{e,approx}$ .

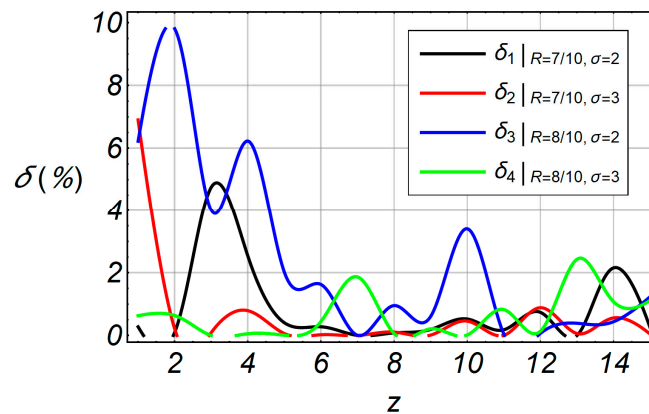


Figure 13. Relative error functions in term of the envelope  $z$ .

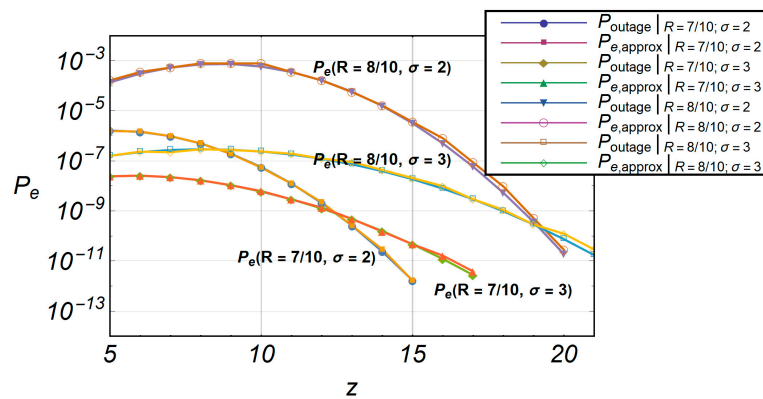
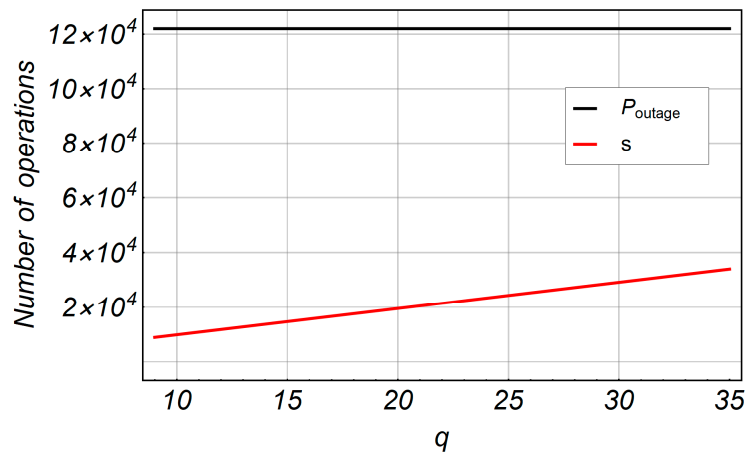


Figure 14. Comparative characteristics of  $P_{outage}$  and accelerated outage probability  $s$ .

The total calculation of formula  $P_{outage}$  required 1193.97 s, or 19 min and 54 s, so the average time per iteration was 70.2335 s. The sped up algorithm's total calculation time for the accelerated formula was 1.25 s, so the average time per iteration was 0.0735294 s. Wolfram language code for time consumed is: `Table[Timing[N[Poutage]],{z,15}] // Total`. Command `Table` provides a calculation for any value of envelope  $z$ , and command `Timing` provides the exact time of calculation. Command `Total` summarizes total time per envelope. Similarly, changing the parameter `Poutage` with `s` for the accelerated algorithm in the previous WL command line provides the time consumed for fast computation. Our algorithm is accelerated as:

$$Ratio = \frac{time(P_{outage})}{time(s)} = \frac{1193.97}{1.25} \approx 955times \quad (24)$$

Figure 15 shows the number of operations in terms of the number of iterations  $q$  for fast computation. The number of iterations is fixed at  $q = 500$  for  $P_{outage}$  because we initially assumed that this number of iterations was satisfied for the closest exact value of  $P_{outage}$ . The number of operations for fast computation of IBSM is less than  $P_{outage}$ . For 500 iterations, we counted 120,000 math operations for  $P_{outage}$ . The number of math operations changes in the range of 9000 to 34,000, which is the result of variety in the number of iterations for fast computation.



**Figure 15.** Number of operations in terms of number of iterations  $q$  for fast computation. The number of iterations is fixed with  $q = 500$  for  $P_{outage}$ .

### 3.2. Second-Order Statistics in Wireless Channels

The level crossing rate (LCR) and the average duration of fade (ADF) are important second-order statistical characteristics describing the fading channel in mobile communications. These values are suitable for designing mobile radio communication systems and for analyzing their performance. In digital telecommunications, a sudden drop in the value of the received signal directly leads to a drastic increase in the probability of error. For optimizing the coding system required to correct errors, the number of times the received signal passes through the given level in time and how long, on average, the signal is below the specified level must be known. The LCR and ADF are the appropriate measures closely related to the quality of the received signal [24].

The LCR of signal  $Z(t)$ , marked as  $N_Z(z)$ , is defined as the signal speed crossing through level  $z$  with a positive derivative at the intersection point  $z$ . The ADF, marked as  $T_Z(z)$ , represents the mean time for which the signal overlay is below the specified  $z$  level.

The LCR at envelope  $z$  is mathematically defined by [22]:

$$N_Z(z) = \int_0^\infty \dot{z} p_{\dot{z}Z}(z, \dot{z}) d\dot{z} \quad (25)$$

where  $z$  is the envelope of the received signal,  $\dot{z}$  is its derivative in time, and  $p_{\dot{z}Z}(z, \dot{z})$  is the joint probability density function. The average fade duration (AFD) is determined as [22]:

$$T_Z(z) = \frac{F_Z(z \leq Z)}{N_Z(z)} \quad (26)$$

where  $F_Z(z \leq Z)$  represents the probability that the signal level  $Z(t)$  is less than the level  $z$ . Evaluation and calculation of LCR and ADF are trivial in an environment where no large reflections exists with a large number of transmission channels and shadowing, which simplifies the mathematical description of the distribution of the signal. However, in complex environments, obtaining LCR and ADF characteristics is time-consuming. An example of a complex environment is described in Stefanovic et al. [20]. In this example, the LCR and ADF expressions were obtained. Their analytical shapes are closed forms, but the complexity shows a long computation time. Thus, the LCR value is normalized by the Doppler shift frequency  $f_d$  [20] through Equation (15):

$$\begin{aligned} \frac{N_Z(z)}{f_d} &= \frac{z^{M_1-1}}{\Gamma(M_1)\Gamma(c_1)\Gamma(c_2)} \left( \frac{N_1 m_1}{r_1} \right)^{M_1-1} \sqrt{\frac{2\pi}{m_1}} \times \\ &\times \sum_{k=0}^{\infty} \frac{(N_1 m_1 z / r_1 ((1/\Omega_{01}) + (1/\Omega_{02})))^{\frac{M_1-c_1-c_2+k-1/2}{2}}}{c_2(1+c_2)_k \Omega_{01}^{c_1} \Omega_{02}^{k+c_2}} K_{(M_1+c_1+c_2+k-1/2)} \left( 2\sqrt{\frac{N_1 m_1 z (\Omega_{01} + \Omega_{02})}{r_1 \Omega_{01} \Omega_{02}}} \right) + \\ &+ \frac{z^{M_2-1}}{\Gamma(M_2)\Gamma(c_1)\Gamma(c_2)} \left( \frac{N_2 m_2}{r_2} \right)^{M_2-1} \sqrt{\frac{2\pi}{m_2}} \times \\ &\times \sum_{k=0}^{\infty} \frac{(N_2 m_2 z / r_2 ((1/\Omega_{01}) + (1/\Omega_{02})))^{\frac{M_2-c_1-c_2+k-1/2}{2}}}{c_2(1+c_2)_k \Omega_{01}^{c_1} \Omega_{02}^{k+c_2}} K_{(M_2+c_1+c_2+k-1/2)} \left( 2\sqrt{\frac{N_2 m_2 z (\Omega_{01} + \Omega_{02})}{r_2 \Omega_{01} \Omega_{02}}} \right) \end{aligned} \quad (27)$$

where  $\Gamma(x)$  denotes the Gamma function,  $M_i$  is  $(m_i N_i^2)/r_i$ ,  $m_i$  is the Nakagami- $m$  fading severity parameter,  $N_i$  denotes the number of identically assumed channels at each microlevel,  $r_i$  is related to the exponential correlation  $\rho_i$ ,  $c_i$  denotes the order of Gamma distribution,  $\Omega_{0i}$  is related to the average powers of the Gamma long-term fading distributions, and  $K_v(x)$  is the modified Bessel function of the second order. Similarly, the AFD is obtained as [20] per Equation (16):

$$\begin{aligned} T_Z(z) &= \frac{F_Z(z \leq Z)}{N_Z(z)} \\ P_z(Z) &= \frac{2(N_1 m_1 / r_1)^{M_1}}{\Gamma(M_1)\Gamma(c_1)\Gamma(c_2)M_1} \times \\ &\sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \left( \frac{N_1 m_1}{r_1} \right)^k \frac{(N_1 m_1 z / r_1 ((1/\Omega_{01}) + (1/\Omega_{02})))^{\frac{c_1+c_2+l-k-M_1}{2}}}{c_2(1+c_2)_k \Omega_{01}^{c_1} \Omega_{02}^{l+c_2}} K_{(c_1+c_2+l-k-M_1)} \left( 2\sqrt{\frac{N_1 m_1 z (\Omega_{01} + \Omega_{02})}{r_1 \Omega_{01} \Omega_{02}}} \right) + \\ &\frac{2(N_2 m_2 / r_2)^{M_2}}{\Gamma(M_2)\Gamma(c_1)\Gamma(c_2)M_2} \times \\ &\sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \left( \frac{N_2 m_2}{r_2} \right)^k \frac{(N_2 m_2 z / r_2 ((1/\Omega_{01}) + (1/\Omega_{02})))^{\frac{c_1+c_2+l-k-M_2}{2}}}{c_2(1+c_2)_k \Omega_{01}^{c_1} \Omega_{02}^{l+c_2}} K_{(c_1+c_2+l-k-M_2)} \left( 2\sqrt{\frac{N_2 m_2 z (\Omega_{01} + \Omega_{02})}{r_2 \Omega_{01} \Omega_{02}}} \right) + \end{aligned} \quad (28)$$

As in the previous example, we defined a general term  $a_k$  from Equation (27), shown in Figure 16.

$$\begin{aligned}
& \left( \sqrt{\pi} \sqrt{m} \Gamma(c+1) \Gamma(q+1) \Omega^{-2c-q} \left( -\frac{L m (R-1)^2}{L (R^2-1) - 2 R (R^L-1)} \right)^{-\frac{L^2 m (R-1)^2}{L (R^2-1) - 2 R (R^L-1)}} \right. \\
& 2^{\frac{1}{20} z} \left( -\frac{L^2 m (R-1)^2}{L (R^2-1) - 2 R (R^L-1)} - 1 \right)^{\frac{5}{4}} \times 5^{\frac{1}{20} z} \left( -\frac{L^2 m (R-1)^2}{L (R^2-1) - 2 R (R^L-1)} - 1 \right) \\
& \sqrt[4]{\frac{L m (R-1)^2 \Omega 10^{z/20}}{2 R (R^L-1) - L R^2 + L}} \left( -\frac{L m (R-1)^2 \Omega 2^{\frac{z}{20}-1} \times 5^{z/20}}{L (R^2-1) - 2 R (R^L-1)} \right)^{c+\frac{L^2 m (R-1)^2}{2 L (R^2-1) - 4 R (R^L-1)} + \frac{q}{2}} \\
& K^{\frac{L^2 m (R-1)^2}{L (R^2-1) - 2 R (R^L-1)} + 2c+q+\frac{1}{2}} \left( 2 \sqrt{2} \sqrt{-\frac{10^{z/20} L m (R-1)^2}{(-2 R^{L+1} + L R^2 + 2 R - L) \Omega}} \right) / \\
& \left( c q! \Gamma(c)^2 \Gamma(c+q+1) \Gamma \left( -\frac{L^2 m (R-1)^2}{L (R^2-1) - 2 R (R^L-1)} \right) \right)
\end{aligned}$$

**Figure 16.** General term of the level crossing rate (LCR) marked as  $a_k$ .

Using the expression in Figure 17, we derived the term  $\rho$  that tends to 1 when  $q \rightarrow \infty$ .

$$\begin{aligned}
& \left( \sqrt{\pi} \sqrt{m} e^{q^2} \Gamma(c+1) \Gamma(q+1) 2^{\frac{1}{20} z} \left( -\frac{L^2 m (R-1)^2}{L (R^2-1) - 2 R (R^L-1)} - 1 \right)^{\frac{5}{4}} \times 5^{\frac{1}{20} z} \left( -\frac{L^2 m (R-1)^2}{L (R^2-1) - 2 R (R^L-1)} - 1 \right) \right. \\
& \sqrt[4]{\frac{L m (R-1)^2 \Omega 10^{z/20}}{2 R (R^L-1) - L R^2 + L}} \left( -\frac{L m (R-1)^2 \Omega 2^{\frac{z}{20}-1} \times 5^{z/20}}{L (R^2-1) - 2 R (R^L-1)} \right)^{c+\frac{L^2 m (R-1)^2}{2 L (R^2-1) - 4 R (R^L-1)} + \frac{q}{2}} \\
& K^{\frac{L^2 m (R-1)^2}{L (R^2-1) - 2 R (R^L-1)} + 2c+q+\frac{1}{2}} \left( 2 \sqrt{2} \sqrt{-\frac{10^{z/20} L m (R-1)^2}{(-2 R^{L+1} + L R^2 + 2 R - L) \Omega}} \right) / \\
& \left( c q! \Gamma(c)^2 \Omega^{2c+q} \Gamma(c+q+1) \left( -\frac{L m (R-1)^2}{L (R^2-1) - 2 R (R^L-1)} \right)^{\frac{L^2 m (R-1)^2}{L (R^2-1) - 2 R (R^L-1)}} \right. \\
& \Gamma \left( -\frac{L^2 m (R-1)^2}{L (R^2-1) - 2 R (R^L-1)} \right) + \sqrt{\pi} \sqrt{m} e^{q^2} \Gamma(c+1) \Gamma(q+1) \\
& 2^{\frac{1}{20} z} \left( -\frac{L^2 m (R-1)^2}{L (R^2-1) - 2 R (R^L-1)} - 1 \right)^{\frac{5}{4}} \times 5^{\frac{1}{20} z} \left( -\frac{L^2 m (R-1)^2}{L (R^2-1) - 2 R (R^L-1)} - 1 \right) \\
& \sqrt[4]{\frac{L m (R-1)^2 \Omega 10^{z/20}}{2 R (R^L-1) - L R^2 + L}} \left( -\frac{L m (R-1)^2 \Omega 2^{\frac{z}{20}-1} \times 5^{z/20}}{L (R^2-1) - 2 R (R^L-1)} \right)^{c+\frac{L^2 m (R-1)^2}{2 L (R^2-1) - 4 R (R^L-1)} + \frac{q}{2}} \\
& K^{\frac{L^2 m (R-1)^2}{L (R^2-1) - 2 R (R^L-1)} + 2c+q+\frac{1}{2}} \left( 2 \sqrt{2} \sqrt{-\frac{10^{z/20} L m (R-1)^2}{(-2 R^{L+1} + L R^2 + 2 R - L) \Omega}} \right)
\end{aligned}$$

**Figure 17.** The element Kummer's transformation  $\rho$ .



In this case, Equations (27) and (28) have already been provided in advance in a closed form where the iteration parameter  $q$  is present, so applying the IBSM would be excessive. To compute the closest exact values of LCR and AFD, 100 iterations were required in Stefanovic et al. [20]. Using Kummer's transformation, both LCR and AFD were calculated in the first iteration. All computations were performed using the values of  $m = 1$ ,  $L = 2$ ,  $\Omega = 1$ ,  $c = 2$ , and  $R = 1/5$ . An auxiliary series was used:

$$C = \sum_{k=1}^{\infty} e^{-k^2} \quad (29)$$

The series  $C$  converges to  $(1/2) \cdot (\vartheta_3(0, e^{-1}) - 1)$ , where  $\vartheta_a(u, x)$ , ( $a = 1, \dots, 4$ ) is the theta function, defined as [30]:

$$\vartheta_3(u, x) = 1 + 2 \sum_{k=1}^{\infty} x^{k^2} \cos(2k \cdot u) \quad (30)$$

Figure 18 shows the comparative characteristics of LCR and accelerated LCR. The deviation of the accelerated series is small in relation to the original series, and the relative error is shown in Figure 19, in the specified range of envelope  $-35 \leq z \leq 30$ .

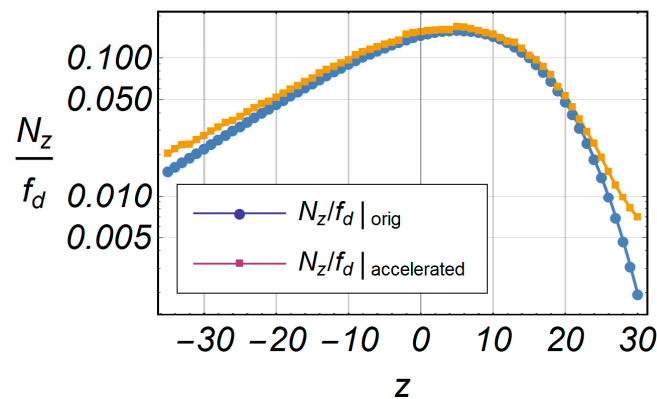


Figure 18. Comparative characteristics of LCR and accelerated LCR.

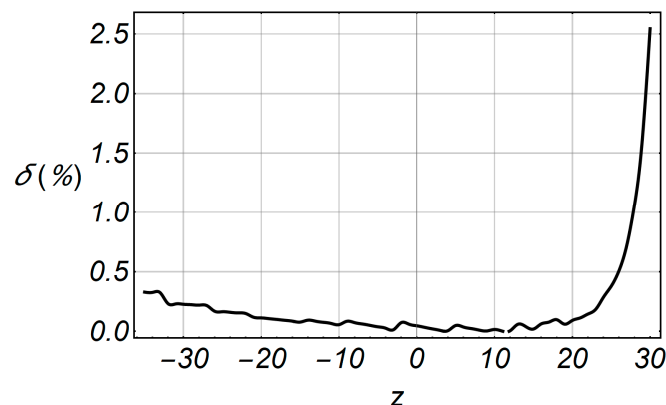


Figure 19. Relative error function of LCR in term of the envelope  $z$ .

The total calculation of the LCR formula required 30.6563 s, so the average time per iteration was 0.437946 s. The total calculation time with the sped up algorithm in the accelerated formula was 1.53125 s, so the average time per iteration was 0.021875 s. Our algorithm is accelerated as:

$$\text{Ratio} = \frac{\text{time}(\text{LCR}_{\text{orig}})}{\text{time}(\text{LCR}_{\text{accelerated}})} = \frac{30.6563}{1.53125} \approx 20 \text{ times} \quad (31)$$

Figure 18 shows the number of operations of LCR ( $N_z$ ) in terms of the number of iterations  $q$  for fast computation. The number of iterations was fixed at  $q = 100$  for  $LCR_{orig}$  because we initially assumed that this number of iterations satisfied the closest exact value of  $LCR_{orig}$ . For 100 iterations, we counted 20,200 math operations for  $LCR_{orig}$ . The number of math operations was 1184 for  $LCR_{accelerated}$  calculated in the first iteration using fast computation. Using the same method, the AFD was obtained by applying Equation (22). Figure 20 shows the comparative characteristics of AFD and accelerated AFD. A small deviation in the range of  $-35 \leq z \leq -28$  was observed, perceived through the relative error in Figure 21.

The total calculation of formula  $AFD_{orig}$  required 19,553.1 s, or 5 h and 25 min, so, the average time per iteration was 279.33 s, or 4 min and 19.33 s. The sped up algorithm total calculation time with the accelerated formula was 1.29688 s, so, the average time per iteration was 0.0185268 s. An obvious difference in the time calculation exists because the number of sums for AFD increased in Equation (27), where we have sums for  $k$ ,  $l$ , and  $q$ . In this case, our algorithm is accelerated as:

$$Ratio = \frac{time(AFD_{orig})}{time(AFD_{accelerated})} = \frac{19,553.1}{1.29688} \approx 15 \cdot 10^3 times \quad (32)$$

For 100 iterations, we counted the  $344 \times 10^6$  math operations for  $AFD_{orig}$ . The number of math operations was 5619 for  $LCR_{accelerated}$  calculated in the first iteration for fast computation.

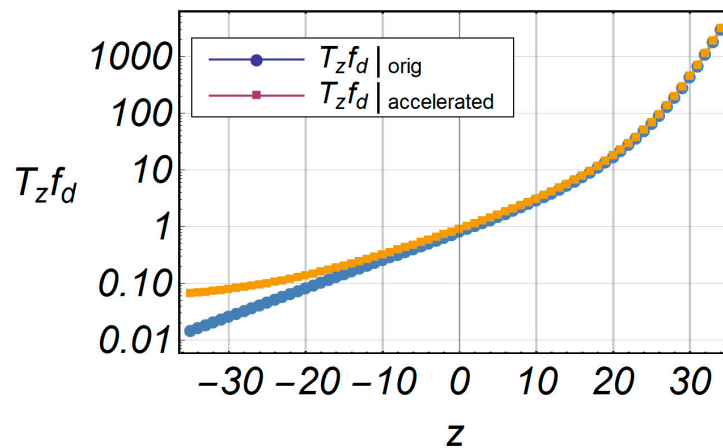


Figure 20. Comparative characteristics of the average fade duration (AFD) and accelerated AFD.

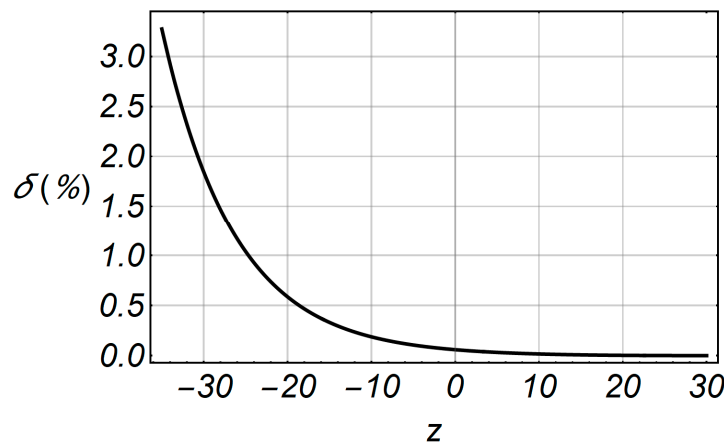


Figure 21. Relative error function of AFD in terms of envelope  $z$ .

#### 4. Conclusions

This paper presents a new method to accelerate the computation and reduce the number of calculation operations in the iteration-based simulation method. The method was developed to simulate the systems and processes when obtaining mathematical formulas in the final closed form is not possible. Often, many phenomena show that closed form expressions and simulations are executed with numerical based tools. In these cases, the users do not have insight into the phenomena that affect the flow of processes, which can lead to incorrect assumptions and results. The method provides insight into processes and systems using symbolic processing, with significant acceleration and reduction in the number of computation operations required. For symbolic derivation, the computer algebra system was used, and Kummer's transformation was used to shorten the computation time. The complete method to reduce the number of operations and shorten the computation time was illustrated in two examples. Both cases require complex and time-consuming calculations. Due to the large number of operations, the memory resources can also play a significant role in the speed of the calculation. The acceleration of the algorithm and the reduction of the number of operations significantly affected efficiency in terms of time savings and the rapid production of results. The method can be used in many fields where fast computation in one-step simulation runs is required.

**Acknowledgments:** This paper is partially funded by Serbian Ministry of Education, Science and Technological Development within the project No. TR 32023.

**Author Contributions:** Vladimir Mladenovic developed algorithms and performed testing; Yigang Cen and Vladimir Mladenovic proposed the field of application and microsimulation concept; Miroslav Lutovac wrote Wolfram language code involving by the symbolic processing; Danijela Milosevic and Matjaz Debevc introduced the semi-symbolic concept combining symbolic and numerical data, analyzed data, and performed their verifications; Vladimir Mladenovic and Matjaz Debevc wrote the paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

#### References

1. Kan, S.C.; Cen, Y.G.; Wang, Y.H.; Mladenovic, V.M.; Voronin, V. SURF Binarization and Fast Codebook Construction for Image Retrieval. *J. Vis. Commun. Image Represent.* **2017**, *49*, 104–114. [[CrossRef](#)]
2. Zhanga, F.; Cen, Y.G.; Zhaoa, R.; Hua, S.; Mladenovic, V. Multi-separable dictionary learning. *Signal Process.* **2017**. [[CrossRef](#)]
3. Voronin, V.; Marchuk, V.; Makov, S.; Mladenovic, V.; Cen, Y.G. Spatio-Temporal Image Inpainting for Video Applications. *Serbian J. Electr. Eng.* **2017**, *14*, 229–244.
4. Lin, L.; Ma, X.; Liang, C.; Huang, X.; Bai, B. An Information-Spectrum Approach to the Capacity Region of the Interference Channel. *Entropy* **2017**, *19*, 270. [[CrossRef](#)]
5. Lin, P.H.; Jorswieck, E.A. Multiuser Channels with Statistical CSI at the Transmitter: Fading Channel Alignments and Stochastic Orders, an Overview. *Entropy* **2017**, *19*, 515. [[CrossRef](#)]
6. Kakar, J.; Sezgin, A. A Survey on Robust Interference Management in Wireless Networks. *Entropy* **2017**, *19*, 362. [[CrossRef](#)]
7. Chen, L. Continuous Delivery: Overcoming adoption challenges. *J. Syst. Softw.* **2017**, *128*, 72–86. [[CrossRef](#)]
8. Ferrari, G.; Colavolpe, G.; Raheli, R. *Detection Algorithms for Wireless Communications*; John Wiley & Sons Ltd.: Chichester, UK, 2004; ISBN 978-0-470-85828-8.
9. Glover, I.A.; Grant, P.M. *Digital Communications*, 3th ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2010.
10. Pavlovic, V.; Mladenovic, V.; Lutovac, M.D. Computer Algebra and Symbolic Processing in Modern Telecommunication Applications: A New Kind of Survey. In *Telecommunications: Applications, Modern Technologies and Economic Impact*; Barringer, J.P., Ed.; Nova Science Publishers: New York, NY, USA, 2014; pp. 29–116; ISBN 978-1-63117-141-3.
11. Mladenovic, V.; Makov, S.; Voronin, V.; Lutovac, M. An Iteration-Based Simulation Method for Getting Semi-Symbolic Solution of Non-coherent FSK/ASK System by Using Computer Algebra Systems. *Stud. Inform. Control* **2016**, *25*, 303–312.

12. Mladenovic, V.; Milosevic, D. A novel-iterative simulation method for performance analysis of non-coherent FSK/ASK systems over Rice/Rayleigh channels using the Wolfram language. *Serbian J. Electr. Eng.* **2016**, *13*, 157–174. [CrossRef]
13. Wolfram, S. *An Elementary Introduction to the Wolfram Language*; Wolfram Media, Inc.: Campaign, TN, USA, 2017; ISBN 9781944183059.
14. Anton, H. *Calculus: A New Horizon*, 6th ed.; Wiley: New York, NY, USA, 1999; pp. 324–327; ISBN 978-0471243496.
15. Weisstein, E.W. Riemann Sum, Wolfram Web Resource. Available online: <http://mathworld.wolfram.com/RiemannSum.html> (accessed on 1 September 2017).
16. Castiglione, F. Microsimulation of Complex System Dynamics: Automata Models in Biology and Finance. Ph.D. Thesis, Universität zu Köln, Köln, Germany, 2001.
17. Grabowski, D.; Grimm, C.; Barke, E. Semi-symbolic modeling and simulation of circuits and systems. In Proceedings of the IEEE International Symposium on Circuits and Systems, Kos, Greece, 21–24 May 2006; pp. 983–986. [CrossRef]
18. Gil, L.; Radetzki, M. Semi-symbolic operational computation for robust control system design. In Proceedings of the 22nd International Conference on Methods and Models in Automation and Robotics (MMAR), Miedzyzdroje, Poland, 28–31 August 2017; pp. 779–784. [CrossRef]
19. Abramowitz, M.; Stegun, I. *Handbook of Mathematical Functions*; National Bureau of Standards: Dover, NY, USA, 1964.
20. Stefanovic, D.; Panic, S.; Spalevic, P. Second-order statistics of SC macrodiversity system operating over Gamma shadowed Nakagami-*m* fading channels. *Int. J. Electron. Commun. (AEÜ)* **2010**. [CrossRef]
21. Radwan, A.; Rodriguez, J. *Energy Efficient Smart Phones for 5G Networks*; Springer: Berlin, Germany, 2015; ISBN 978-3-319-10313-6.
22. Borhani, A.; Patzold, M. Modeling of Vehicle-to-Vehicle Channels in the Presence of Moving Scatterers. In Proceedings of the 2012 IEEE Vehicular Technology Conference (VTC Fall), Quebec City, QC, Canada, 3–6 September 2012; pp. 1–5. [CrossRef]
23. Patzold, M. *Mobile Radio Channels*, 2nd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2011; ISBN 978-0-470-51747-5.
24. Patzold, M. *Mobile Fading Channels*; John Wiley & Sons, Ltd.: New York, NY, USA, 2003; ISBN 978-0471495499.
25. Lutovac, M.; Mladenovic, V.; Lutovac, M. Development of Aeronautical Communication System for Air Traffic Control Using OFDM and Computer Algebra Systems. *Stud. Inform. Control* **2013**, *22*, 205–212. [CrossRef]
26. Martinez, W.; Martinez, A. *Computational Statistics Handbook with MATLAB*; Chapman & Hall/CRC: London, UK, 2015; ISBN 9781466592735.
27. Andrews, G.; Askey, R.; Roy, R. *Special Functions (Encyclopedia of Mathematics and its Applications)*; Cambridge University Press: Cambridge, UK, 2001; pp. 212–222. ISBN 978-0521789882.
28. Borwein, J.M.; Crandall, R.E. Closed forms: What they are and why we care. *Not. Am. Math. Soc.* **2013**, *60*, 55. [CrossRef]
29. Mladenovic, V.; Makov, S.; Cen, Y.G.; Lutovac, M. Fast Computation of the Iteration-Based Simulation Method—Case Study of Non-coherent ASK with Shadowing. *Serbian J. Electr. Eng.* **2017**, *14*, 415–431. [CrossRef]
30. Available online: <http://reference.wolfram.com/language/ref/EllipticTheta.html> (accessed on 19 December 2017).



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).