Scientific Journal of Riga Technical University
Telecommunications and Electronics
2010
Volume 10

Choice of Gaussian Noise Generator for Noise Immunity Estimation of Binary Channels with Viterbi Detection

Albert Zelenkov, *Riga Technical University,* Sergey Zelenkov, *Riga, Radar Systems, Inc.*

Abstract – **This article reveals three methods of noise generating with Gauss distribution by using uniformly distributed random-numbers in computer imitation modeling of the digital communications systems – the uniformly distributed random-numbers summing method, Box-Muller method and Marsaglia-Tsang ziggurat-method. The results of simulation by the Monte Carlo include histograms of noise generators and limits for accuracy and computational time of various methods of Gaussian noise generating to evaluate the noise immunity of transmission of binary data by antipodal signals in symmetric channels with intersymbol interference and the Viterbi detection.**

Keywords: **BER, bit error probability, impulse response, intersymbol interference, noise immunity, noise generators, signal-to-noise ratio, state diagram, Viterbi detection.**

I. INTRODUCTION

In the computer imitation modeling of the digital communications systems, normal (or Gaussian) white noise (*Additive White Gaussian Noise* or AWGN) is being added to the information signals, based on the given signal-to-noise ratio (SNR or S/N). This noise has zero mean value and variance value of 1 (standard normal distribution law N(0,1)). Such noise can be generated by random-number generators.

Wide spread occurrence of the Gaussian distribution law is caused by fact, that it is the limit law for another distributions and is very common in practice, for instance as the noise in the electronics or noise in the radio-electrical circuits.

There are few methods for calculation of normal numbers with distribution N(0,1) by using the uniformly distributed source numbers. One of the simplest and prevalent methods is based on the limit of the random values sum distribution laws (distribution of the sample average), which has the common name of the central limit theorem. It has been proved in [1], that the sum of the large number of independent random components with the same distribution law, so that each of these components has nearly zero influence on this sum, has the distribution, which is very similar to normal one. In practice, by using this method, the numbers from N(0,1) can be calculated with simple adding of the uniformly distributed (over the interval of [-1,1]). These numbers can be calculated by the formula $x_i = 2y_i - 1$, where y_i numbers are uniformly distributed over the interval of [0,1]. It is recommended [2], [3] to sum at least MG=12 base numbers x_i, in order to get random number with distribution law similar to Gaussian one. The result of the sum after dividing by multiplier of

`sqrt(1./3.*(double)MG)` $<->$ $\sqrt{\mathrm{MG}/3}$,

that normalizes variance to 1, is gradually converging estimation of the distribution N(0,1) with MG increasing. In C/C++ language standards for calculation of the uniformly distributed numbers y_i there is implemented standard function `int rand()`, which returns integer pseudo-random number from the interval of [0,`RAND_MAX`], where `RAND_MAX` is constant, defined by `#define RAND_MAX 32767`. In order to get y_i values from the interval of [0,1], it is necessary to convert result of the rand() function to the real number and divide it by `RAND_MAX`. If the probabilities are equal, the relative frequency of any y_i number approaches to the probability of 1/32767, when the length of generated sequence is increasing.

However, the random-number summing method is relatively slow, or, when MG values are small, it is inaccurate description of the Gaussian distribution of the momentary noise values, especially in the area of the large momentary noise values. At the same time, the simplicity of the Gaussian-like numbers calculation is attractive, when using this method.

There is not much known about how such inaccuracy affects the estimation of the channel quality with intersymbol interference, when using imitation modeling, except common considerations. We are interested in getting such estimation by making modeling of the data processing in the binary data transmission channels with use of the Viterbi detection.

Another original method of generator building for Gaussian numbers is based on the Box and Muller transformation [3]. This generating method is faster and more precise, than central limit theorem based method, in the same time it is simple for realisation. Its algorithm can be reduced to following. Let x_i and y_i be independent random numbers with uniform distribution over interval of [-1,1]. We now can calculate $s_i = R_i^2 = x_i^2 + y_i^2$. If $R_i > 1$ or $R_i = 0$, then x_i and y_i values should be “discarded”, then we make another pair of the numbers (x_i, y_i) and calculate s_i again. Once the condition $0 < R_i \le 1$ has been satisfied, we calculate z_{i1} and z_{i2} following formulas $z_{i1} = x_i \cdot \sqrt{\frac{-2 \cdot \ln s_i}{s_i}}$ and $z_{i2} = y_i \cdot \sqrt{\frac{-2 \cdot \ln s_i}{s_i}}$, these numbers are independent values, that satisfies standard normal distribution N(0,1).

At the present moment the fastest and the most precise method of the normal distribution law satisfying random-numbers generating methods is so-called truncated pyramids method (The Ziggurat Method for Generating Random Variables), which has been offered in 1984 by Marsaglia and Tsang [4] and improved by same authors in 2000 [5]. The [5] describes in details this method from mathematical viewpoint and its programming realization in C language. The mathematical and programming realization difficulties can be considered as the methods weaknesses.

By using computer imitation modeling with Monte-Carlo method [2, 3], we made histograms and analyzed limit possibilities of all three described methods of Gaussian number generators for the Gaussian noise imitation in the binary data transmission channels. The purpose of this modelling was quality estimation (noise immunity) for binary data transmission digital systems with antipodal signals by using symmetric channel with intersymbol interference (ISI) with AWGN. In order to make optimal detection of the received signals we used Maximum-Likelihood Sequence Estimation method (MLSE), realised according to the Viterbi algorithm [6,7], that is known also as the Viterbi detection [8] or the Viterbi equalizing [9]. The length (the number of samples) of the channel's discrete impulse response is parameterized by the constant value V, which determines state-diagram structure (the trellis-diagram, same as Forney trellis) and decoding procedures in program, described in [10]. The discrete values of the impulse response are inputted from the keyboard. In real digital communications systems, e.g., in GSM (Global System for Mobile Communications), the impulse response of the channel is estimated during the process of periodical channel probing with testing pseudo-random sequence.

Below we give noise immunity graphs, which we've received after imitation modelling, for different Gaussian noise forming methods and different types of the transmission channel's frequency response linear parts. We have compared noise immunity of the simple threshold detecting of binary signals as well as detecting with Viterbi algorithm for the channels with spectral zeroes (zeroes of the channel's frequency response). The lag time of data reading from the Viterbi detector's state diagram "survivor" (or active) paths has been defined more accurately in order to get lower probability of the bit error. As an estimate of the bit error probability, we used bit error rate (BER), denoted below as p_{er}.

The Fig.1 shows flow block of calculation program for binary signal processing modeling in the channels with ISI and AWGN when using Viterbi detector as a receiver. The schematic in Fig.1 includes two random number generators (RNG). The first one is used for the source "information" binary sequence generation with two symbols of equal probabilities. Before sending it to the channels input, this sequence is converted to the binary impulse sequence with pulse-amplitude modulation (PAM), which consists of the ± 1 values. The second random-number generator models the discrete white Gaussian noise with the zero mean value and variance of 1. The AWGN is added to the convolution result of the ± 1 sequence with channel's impulse response. The noise effective value is set according to the given SNR in dB. The SNR step is chosen to be 0.5 dB. The number of noise immunity graph points is set by global constant in program.

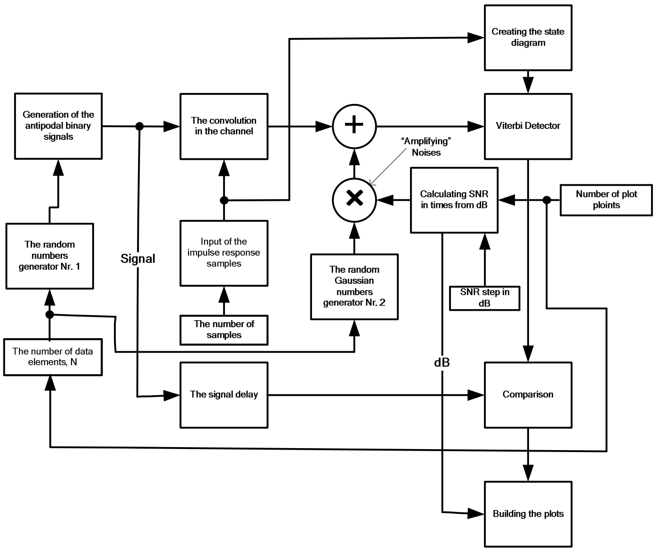

Fig.1. Processing algorithms flow block.

The additive mixture, which consists of the convolution signal and noise, is sent to the Viterbi detector input. The impulse response samples (inputted from the keyboard) for the linear portion of the transmission channel are used for convolution calculation as well as in realization of the process of useful signal receiving with noise by using Viterbi detection. The binary data after detection is being compared with really transmitted for the estimation of relative frequency p_{er} of the bit error, i.e. BER. The probability of the bit error is estimated through relative frequency value p_{er} for big enough number of binary data. Afterwards the program makes graph of noise immunity curve representing dependency of BER logarithm ($\lg p_{er}$) on SNR measured in dB. This graph determines the quality of the digital communications system. The processing program, which corresponds flow block in Fig.1, is written in C/C++ programming language, and its full listing is available in [10].

Let's examine three methods of the normalized Gaussian number generators mentioned above and compare them by their influence degree on the noise immunity estimation for the binary channel with ISI and AWGN.

II. THE SUM OF RANDOM VALUES

Let's use the central limit theorem for quasi-Gaussian number generating. As it has been mentioned before, in order to get appropriate approximation for Gaussian distribution law, it is recommended to make a sum of at least 12 numbers with same uniform distribution law. However, despite the simpliness, this operation can be time-consuming. Just to make sure, how fast normalization becomes, when the number of components increases, Fig.2 shows histograms, where MG=1, 2, 3, 4, 5, ... – the number of components in a sample sum of random values. The number of these sample sums, which were used for histogram building, is $M = 10^5$. The step of histogram in probability axis is $dX = 0.1$ and it belongs to a range of the random valueв of $-5 \le X \le +5$.

The values of the random numbers with uniform distribution in the range of [0,1) can be calculated in four ways. The first one is to use standard library C-function `int rand()`. Another example of uniformly distributed numbers generator is function `urand()` used from [3], which is defined in C language as `float unirand(long int *iy)` [10]. For the third example we used generator described in [4] by macros `UNI` and `SHR3`, which can be launched by macros `UNI`. The fourth generator is Wichman and Hill random numbers generator [11], which has been realised in *MS Excel* 2003/2007 as a function `RAND` [12]. The algorithm of this generator (Algorithm AS 183) is available in [11,12] for FORTRAN programming language. In case of C programming language it is described by function

```
float frandom()
{
/* Wichmen-Hill Random Number Generator
for Excel 2003
(http://support.microsoft.com/kb/828795/r
u)*/
  double amod, x;
  ix=(171*ix)%30269;
  iy=(172*iy)%30307;
  iu=(170*iz)%30323;
  amod=(double)ix/30269.0+
(double)iy/30307.0+(double)iz/30323.0;
  return (float)modf(amod,&x);
}
/* The global variables long int ix, iy
,iu are given values in range between 1
and 30000 until the first handling. The
standart C/C++ function modf()  separates
the fractional part and the integer part
of the value assigned to the variable
amod*/
```

When MG=1, each sample sum consists of one random number from the uniformly distributed interval of $\left[-\sqrt{3},\sqrt{3}\right]$, with zero mean value and variance of 1. The histogram is shown in Fig.2.

When MG=2 each sample sum is the sum of two random numbers with the same uniform distribution for each. The sum of two random numbers follows the Simpson distribution law in the interval of $\left[-\sqrt{6},\sqrt{6}\right]$. In this case the probability density has a shape of the isosceles triangle, since this law is a convolution of the two same uniform laws. The histogram for Simpson law is given in Fig.3.

When MG=3 the histogram in Fig.4 already looks like the histogram of the Gaussian numbers generator, which can be generated by, for instance, ziggurat-method (or Box-Muller method), which is shown in Fig.5. The difference of the histograms for the random values, calculated as the sums of the uniformly distributed values (when $\mathrm{MG} \ge 3$), in general, lies in the accuracy of description of the probability for large (by absolute value) momentary noise values, i.e. description of

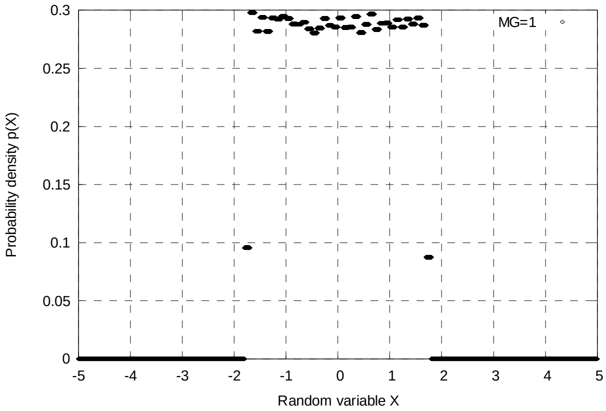

Fig.2. The histogram of uniformly distributed probability density. MG=1, the step dX=0.1, $M=10^5$.

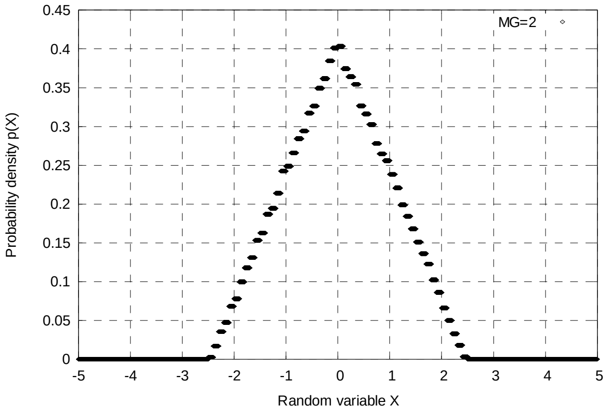

Fig.3. The histogram of the normalized Simpson distributed probability density for the sum of 2 uniformly distributed random values. MG=2, the step dX=0.1, $M=10^5$.

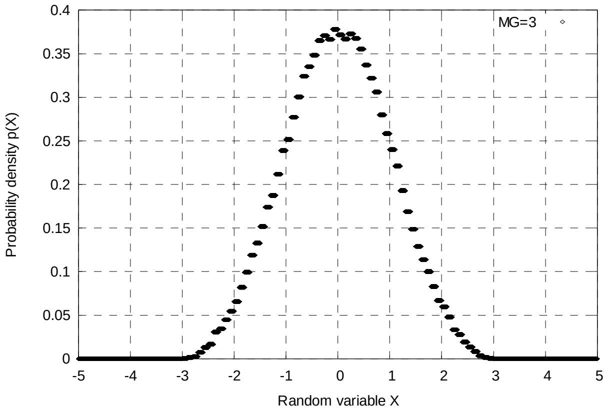

Fig.4. The histogram of the normalized probability distribution density for the sum of 3 uniformly distributed random values. MG=3, the step dX=0.1, $M=10^5$.

integral curves “trails” for Gaussian probability distribution, where it’s values are close to 0 or 1. When MG=3 the sum of random values is distributed over the interval of [-3,3], i.e., the generator doesn’t generate numbers with absolute value over 3, meanwhile, theoretically, at the output of the generator there can be numbers with any numeric value (the bigger the numbers absolute value, the less probable it will be generated).

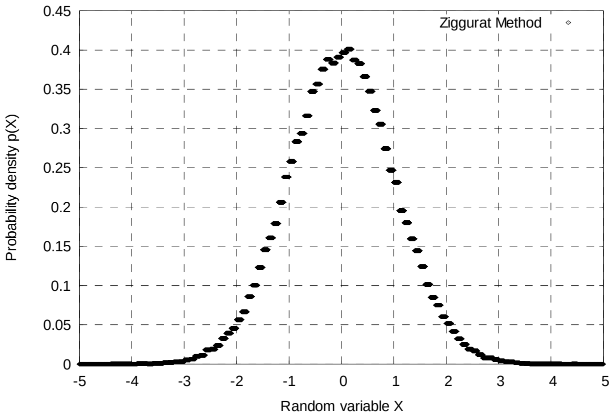

Fig.5. The histogram of normalized Gaussian distributed numbers. RNOR (Ziggurat-method), the step dX=0.1, M=10^5 .

When MG=4, the sum of random values is distributed over the interval of $\left[-2\sqrt{3},2\sqrt{3}\right]$, when MG=5 – over the interval of $\left[-\sqrt{15},\sqrt{15}\right]$ and, at last, when MG=12 – over the interval of $[-6, 6]$ and so on. Generally, when summing MG uniformly distributed random numbers, we get random asymptotic Gaussian numbers over the interval of $\left[-\sqrt{3\cdot \mathrm{MG}},\sqrt{3\cdot \mathrm{MG}}\right]$ with zero mean value and variance of 1.

III. THE RESULTS OF IMITATION MODELING OF THE BINARY TRANSMISSION CHANNEL WITH VITERBI DETECTION WHEN ISI AND AWGN ARE PRESENT

According to the table of the probability integral available in hand-books (e.g., in [13]), it’s easy to determine, that the probability of the Gaussian numbers with zero mathematical expectation and variance of 1, that exceed the absolute value of 6, is very small and is equal merely 1.97e-9 . For that reason, the number of random values $\mathrm{MG} \geq 12$ when using the central limit theorem should indeed give good approximation to Gaussian numbers for the most of the Gaussian process modeling application, for instance, modeling of the Gaussian noises in digital communications systems. This assumption has been checked in example of MLSE modeling task in binary channels with antipodal signals when ISI and AWGN are present. When modeling such channels, the task is to get small error probability values ($10^{-7} \ldots 10^{-5}$ and less) at the output of the Viterbi detector, and the noise itself should be Gaussian, since it is assumed that the channel is linear.

The imitation modelling was made by the method of Monte-Carlo according the flow block in Fig.1. We used different types of generators described before for the source of the Gaussian noise. We also used Viterbi detection with tact-by-tact refreshing and analysis of the survivor paths. Symbolical estimations of the transmitted binary data in this case are read from the survivor paths with a delay, the value of this delay is assigned to the variable W ($\mathrm{W} \geq 0$).

The discrete impulse response of the channel, which may consist of any number of the samples assigned to the variable V ($\mathrm{V} \geq 1$), is sequentially input step-by-step from the PC keyboard at the beginning of the processing procedure. For analysis we have chosen the discrete impulse response h(k)={1,2,3,4}, which consists of 4 (V=4) samples.

The decibel-log frequency response, which corresponds to impulse response h(k)={1,2,3,4}, is shown in Fig.6 as a dependency of $20\lg\left[\left|K(j\omega T)\right|\right]$ on ωT (where $\omega = 2\pi f/(NT)$, $f = 0,1,2,...,N/2$, T – sampling period). Such frequency response is substantially irregular. At the interval from 0 to ¼ of sampling frequency ($0 \leq \omega T < \pi/2$), it is falling from 0dB to -10...-12dB and with further increase of the frequency it doesn’t exceed value of -11dB. Due to the intersymbol interference, the data transmission without Viterbi detection wouldn’t be possible in this channel, since the bit error probability is close to $1/2$ in case of threshold detecting.

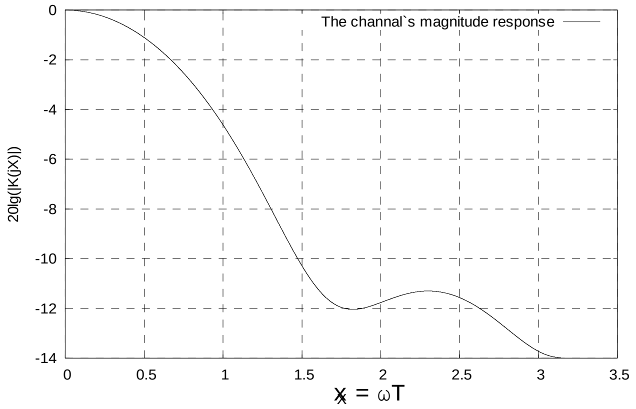

Fig.6. The decibel-log frequency response of the channel with impulse response h(k)={1,2,3,4}, x=ωT=2πf/N, f=0,1,2,...,N/2, N=1024 .

The number of the binary symbols at the channel input was set to M=10^5, 10^6 and 10^7 for each point of the noise immunity curve.

As it was mentioned before, the simplest generator to realize is the uniformly distributed random numbers summing generator. In C programming language the program is made the following way:

```
s=.0;
for(k=0;k<MG;k++)
s+=(2.*rand()/32767.01.)/
sqrt(1./3.*(double)MG);
```

Instead of the expression `rand()/32767.0`, it is possible to use the function `unirand()` or the macro `UNI` or, at last, the function `frandom()`, described before.

In order to realize ziggurat-method in C programming language, the pre-processor directive `#include`

`"ZigGauss.cpp"` (see [10]) must be included in the processing program. In this case the Gaussian numbers generator is called by macro RNOR (it is necessary to call `zigset(unsigned long)` function before doing this, [5]). The Gaussian numbers generation algorithm by Box-Muller method is realized by function `gaussrand()` (see description in [10]).

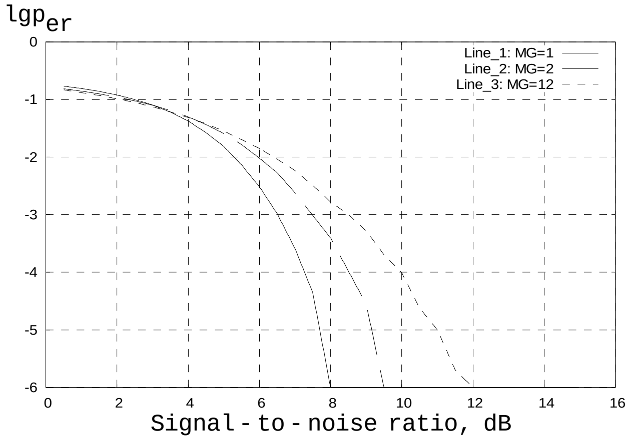

Fig.7. The noise immunity curves when using rand() generator, MG=1 (Line 1), MG=2 (Line 2) и MG=12 (Line 3), M=10^6, h(k)={1,2,3,4}, W=25.

The curves of noise immunity as the dependence of lg p_{er} ($p_{er} = BER$) of SNR for all three values of MG, when $M = 10^6$, are shown in Fig.7. For Fig.8 MG = 12 and $M = 10^7$. For comparison, the Fig.9 shows the noise immunity curve for Gaussian numbers generator `RNOR` when $M = 10^7$.

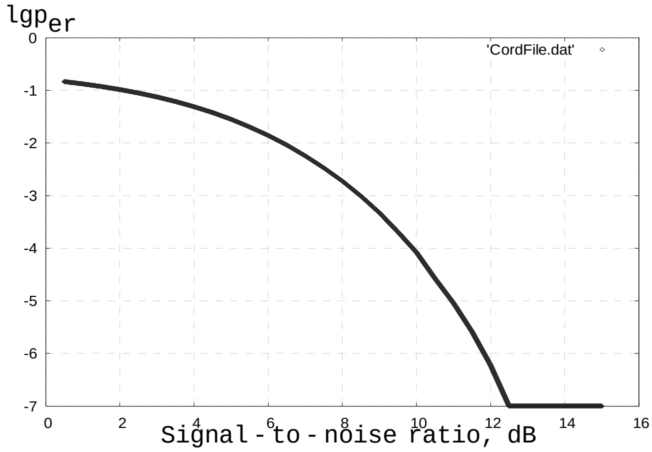

Fig.8. The noise immunity curve when using unirand() generator, MG=12, $M = 10^7$, h(k)={1,2,3,4}, W=20.

By comparing plots in Fig. 7 and Fig. 8, we can see, that when MG is increasing, the noise immunity curves, which nearly overlap over interval of $-\infty$ <SNR<2...4dB, are separating when SNR>4dB and they become even more flat with the further increase of MG approaching to the Gaussian numbers generator's curve in Fig. 8 from the lower bound. At the same value of SNR, the bit error rate is increasing with the increase of MG. For instance, in Fig. 7, where MG=1 and SNR=8dB, there is lg $p_{er} = -6.0$ ($p_{er} = 10^{-6}$), and in Fig.8, where MG=12, at the same value of SNR there is lg $p_{er} = -2.722$ ($p_{er} = 1.8967 \cdot 10^{-3}$), i.e. the order of error probability estimation p_{er} has increased by three, which can be explained by expansion of the possible values interval for generated random numbers by increasing the number of components in the sum. In all figures the estimation of the bit error probability (as a sample average value of the relative frequency [1]) becomes less accurate when the probability itself approaches to the value of 1/M (i.e. the value 10^{-6} in Fig.7 or value of 10^{-7} in Fig.8, 9). This can be explained by the fact, that corresponding the 1/M value, there must be in average a one error on M (10^6 or 10^7) binary symbols at the channels input. The decrease of the error probability estimation accuracy when reaching 1/M is caused by decrease of the number of errors in the sample over interval of M cycles. The changes in the noise immunity curves appearance are the evidence of the bit error probability estimations increasing statistical instability – from monotone decreasing at the beginning they have more or less noticeable breaks of the curves in the reference points at the end of the SNR range. This can be seen, e.g., in Fig.7 (Line 3). The bit error probability below 1/M can't be estimated.

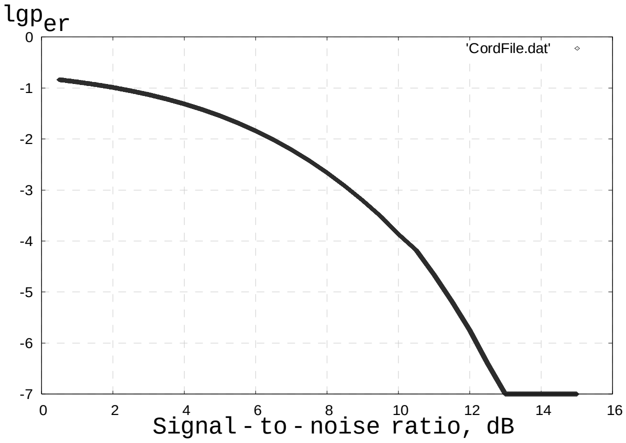

Fig.9. The noise immunity curve when using RNOR generator, $M = 10^7$, h(k)={1,2,3,4}, W=20.

By comparing Fig.7 and Fig.8, we can see that when MG=12, the noise immunity curve is close enough to one plotted for the Gaussian numbers generator `RNOR`. E.g., when SNR=12dB, the bit error probability estimation, calculated as the sum of 12 random numbers generated by `unirand()`, is 3 times less than the estimation of the `RNOR` generator. At the same time, when SNR=12dB the lg $p_{er} = -6.222$ ($p_{er} = 6.0e-7$), and in Fig.8 the lg $p_{er} = -5.745$ ($p_{er} = 1.8e-6$). The increase of MG from 12 to 24 causes the increase of the value of error probability estimation logarithm up to lg $p_{er} = -5.854$ ($p_{er} = 1.4e-6$), i.e. increases the estimation itself by 2.33 times, which is just 1.29 times less than estimation, calculated for the `RNOR` generator. However, the time required for calculation of the whole noise immunity curve (consists of 30 points with step 0.5dB on SNR axis (from 0.5dB to 15dB), the $M = 10^7$ for every point, W=20) for MG=12 when using unirand() generator is taking almost 2 times long (43 min, 22.00 sec), and for MG=24 almost 3 times long (72 min 0.26 sec), than the one when using Gaussian numbers generator RNOR (24 min 37.98 sec). The Gaussian numbers generator Box-Muller (`gaussrand()`) for same calculations requires comparable with `RNOR` amount of time (30 min 2.190 sec). Almost same time – 28 min 15.35 sec – is required when using generator rand() for MG=12 and 41 min 33.25 sec for MG=24. When using generator frandom() for MG=12 it takes 37 min 46.36 sec, and for MG=24 – 46 min 55.56 sec, i.e. a bit longer than in case of `rand()`.

The measurements of the calculation time were made by means of the C function `int gettime(struct time *)`. For experiments we have used PC with dual-core processor *Core 2 Duo E6550* with clock speed 2.325*GHz* on the *Intel DQ35MP LGA 775* motherboard with 2.048*GB Kingston DDRII RAM* (the available physical memory is 1.45*GB*), that is evenly divided between two cores.

Thus, we can note, that the calculation of the Gaussian-like values by summing uniformly distributed numbers is not always convenient way. In order for such case to get the results of the digital communications system modeling when using Viterbi detection, which would be close to ones using the straight Gaussian number calculation methods (Box-Muller method or ziggurat-method of Marsaglia and Tsang), it is necessary to significantly increase the calculation time.

In conclusion, we’d like to note the good quality characteristics of the channels when using Viterbi detection. Compared to the equalizers of other types [9] this detector provides the least loss of SNR at the presence of ISI and AWGN, and at the given value of SNR it provides the best channel quality, since it realizes the least probability of the bit error. Viterbi detection makes it possible to transfer data with small bit error probability even in channels with very bad frequency responses, which has zeroes in the working bank, that leads to partial spectrum suppression and (as consequence of that) intersymbol interference. Such channels can be encountered in digital radio-communications [9]. At the presence of ISI (even if there are no noises) it is practically impossible to transfer information in such channels without Viterbi detection, since the bit error probability is reaching $1/2$. During the process of Viterbi detection testing realized in the C/C++ function `viterbi_m()` [10], we have received the noise immunity curves for channels with spectral zeroes. These curves match similar quality characteristics in Fig.10.3.3 in [9].

It’s important to note, that the bit error probability in the channel with Viterbi detection at the presence of ISI and AWGN can be decreased by increasing the recommended value of a delay for decision-making on transferred binary data by survivor paths. This recommended value of the delay comes from the expression $\mathrm{W}=5\times(\mathrm{V}-1)$, given in [9].

The research of such possibility for the channel with spectral zeroes was made on example of the impulse response h(k)={0.2275, 0.46, 0.688, 0.46, 0.2275}, which, according to [9], belongs to category of “very bad” quality channels. This impulse response is shown in Fig.10, the V value for it is $\mathrm{V}=5$ and a recommended value of data reading delay should be $\mathrm{W}=20$. The channel’s gain-frequency response in Fig.11 corresponds this impulse response. This frequency response has a spectral zero at $\mathrm{X}=\omega\mathrm{T}=2.1$, where T is sampling period.

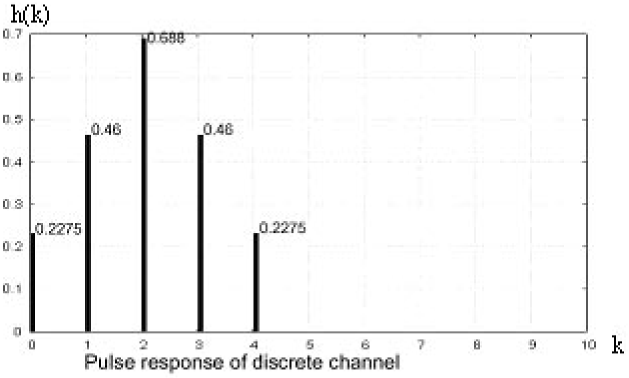

Fig.10. The impulse response of the discrete channel h(k)={0.2275, 0.46, 0.688, 0.46, 0.2275} with spectral zero.

The curves of the noise immunity for the channel with impulse response illustrated by Fig.10 for the various values of a delay for the binary data reading from survivor paths of the Viterbi detector state diagram are shown in Fig.12,13 and 14.

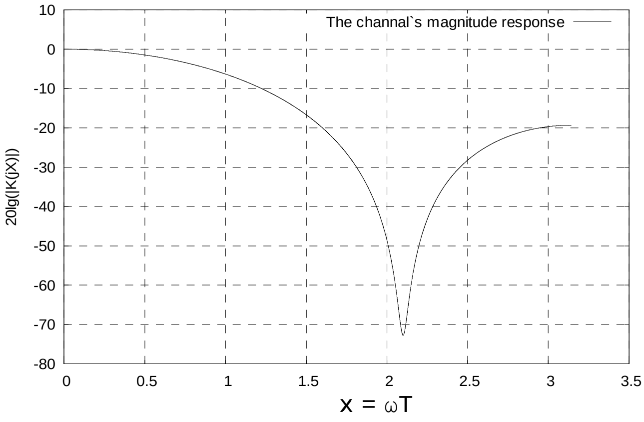

Fig.11. The gain-frequency response for a channel with impulse response h(k)={0.2275, 0.46, 0.688, 0.46, 0.2275}, $x=\omega T=2\pi f/N$, $f=0,1,2,...,N/2$.

For comparison, the Fig.12 illustrates noise immunity curve Line 1 for the channel with no intersymbol interference (i.e. for the Nyquist channel), when h(k)={1,0,0,0,0,...}). Line 1 shows the dependency of the $\lg(\mathrm{BER})$ on SNR for the binary antipodal signals (see example in рис.5.2.4 in [9] or BPSK curve in 2.7.att. in [10]).

By comparing Fig.12,13 and 14, we can see, that the increase of the reading delay from $\mathrm{W}=15$ to $\mathrm{W}=25$ for big enough value of SNR, which is at least 10dB, causes a noticeable decrease of the bit error rate p_{er}.

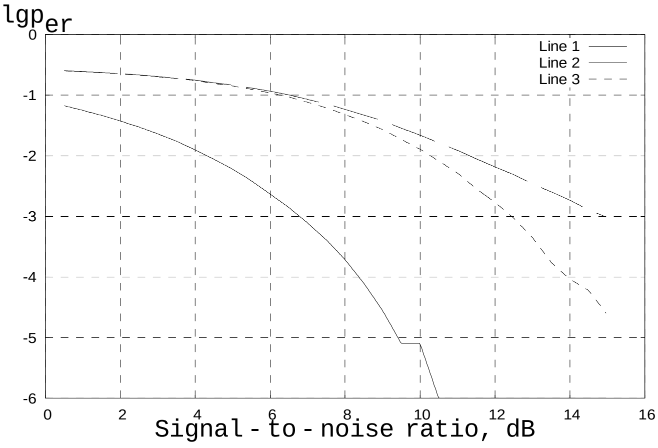

Fig.12. The noise immunity curves. RNOR generator, $M=10^6$. Line 1 – no ISI, W=5; Line 2 – ISI present, W=15; Line 3 – ISI present, W=25.

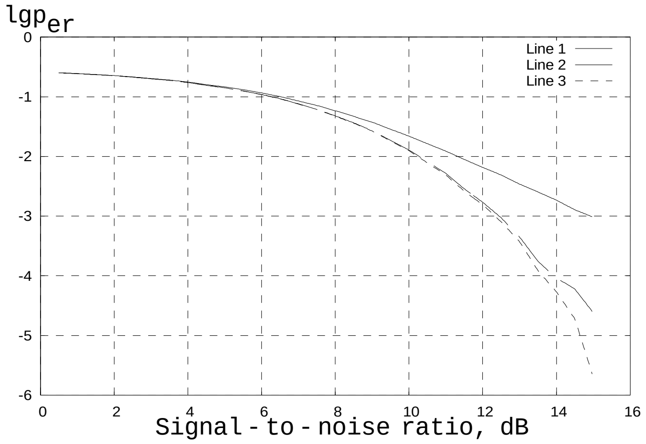

Fig.13. The noise immunity curves. RNOR generator, $M=10^6$. Line 1 – ISI present, W=15; Line 2 – ISI present, W=25; Line 3 – ISI present, W=35.

E.g., for SNR=14dB an estimate of the bit error probability is decreasing from $p_{er} = 1.85 \cdot 10^{-3}$ ($\lg p_{er} = -2.732$) for W=15 to $p_{er} = 9.2 \cdot 10^{-5}$ ($\lg p_{er} = -4.036$) for W=25, i.e., by 20 times. The further increase of W up to 35 cycles, as it's shown in Fig.13, where for SNR=14dB there is $p_{er} = 5.4 \cdot 10^{-5}$ ($\lg p_{er} = -4.268$), and even more up to W =45, as it's shown in Fig.14, where for the same SNR value there is $p_{er} = 4.6 \cdot 10^{-5}$ ($\lg p_{er} = -4.337$), gives a little improvement of the channels quality, proportionally increasing calculation time value instead.

The modelling of different channels (with spectral zeroes) shows, that the choice of the reading delay value by expression $W=5 \times V$ is more optimal by quality (i.e. it provides a smaller bit error probability) rather than the expression $W=5 \times (V\text{-}1)$.

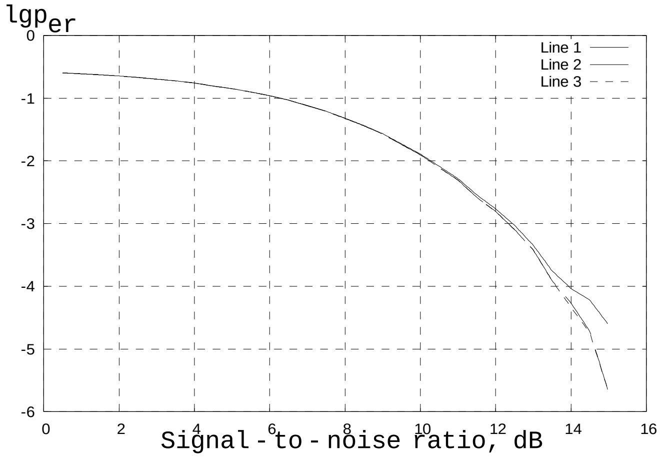

Fig.14. The noise immunity curves. RNOR generator, $M=10^6$. Line 1 – ISI present, W=25; Line 2 – ISI present, W=35; Line 3 – ISI present, W=45.

The fact, that the channel with impulse response from Fig.10 is "very bad" from the viewpoint of reliability of the data transfer, is confirmed by Fig.15, which shows the quality characteristics of three channels with spectral zeroes. Fig.16 also proves that, which shows gain-frequency responses of these three channels. Two of them – Line 1 and Line 2 – have two zeroes in range from 0 to $1/(2T)$ Hz, but "vary bad" Line 3 one – just one zero, however this zero has the widest suppression frequency band on any level.

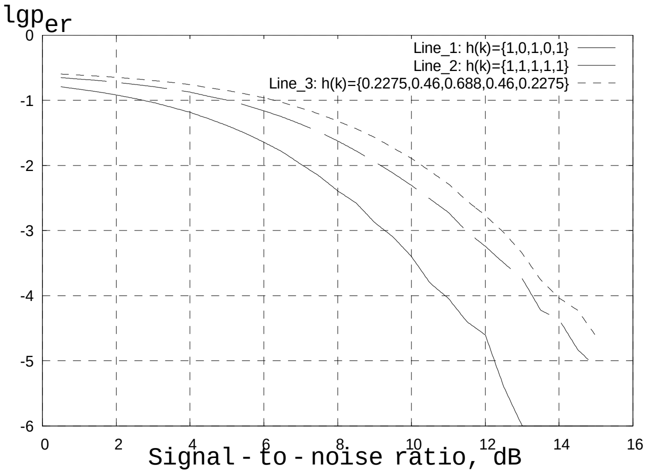

Fig.15. The noise immunity curves for impulse responses with spectral zeroes: Line 1 – h(k)={1,0,1,0,1}; Line 2 – h(k)={1,1,1,1,1}; Line 3 –h(k)={0.2275, 0.46, 0.688, 0.46, 0.2275}. RNOR, $M = 10^6$, $W = 25$.

Thus, noise immunity curve Line 3 in Fig.15 for the channel's impulse response in Fig.10 is to the right, compared to the other two. To get the given value of the bit error probability in such digital communications system it's necessary to provide additional 1dB for SNR at the Viterbi detector input compared to the Line 2, and additional 3dB compared to the Line 1.

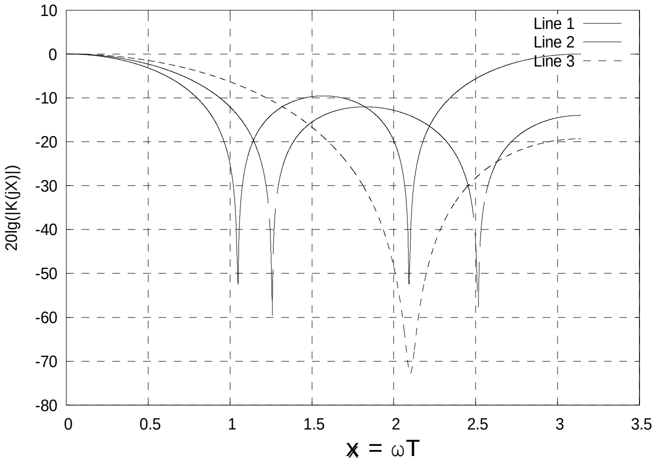

Fig.16. The channels' gain-frequency responses for impulse responses with spectral zeroes: Line 1 – h(k)={1,0,1,0,1}; Line 2 – h(k)={1,1,1,1,1}; Line 3 – h(k)={0.2275, 0.46, 0.688, 0.46, 0.2275}, $x=\omega T=2\pi f/N$, $f=0,1,2,...,N/2$.

In the case of a simple threshold detection all three channels with spectral zeros can not be used for data transmission, since the probability of errors in these channels has a value between 0.25 and 0.5 due to ISI and for large values of signal to noise ratio, and even in the complete absence of noise.

IV. CONCLUSION

In conclusion, we can say that when estimating noise immunity of the digital communications systems by using means of the imitation modeling, it is possible to use the central limit theorem for white Gaussian noise generation. However, in order to achieve more accurate results it's necessary to make sum of at least 10…12 uniformly distributed over interval of [-1,1] base random values. At the same time, it doesn't matter which random numbers generator exactly has been used – `rand()`, `unirand()`, `UNI` or `frandom()`. In numeric sequences, created by these generators, the random values are linearly independent (delta-correlated).

Both the method of Box-Muller and ziggurat-method of Marsaglia and Tsang gives same result at the end for the bit error probability, when modeling data transfer systems with Viterbi detection.

By comparing the computing time used by processing program, the central limit theorem based method requires significantly more time, than Box-Muller and ziggurat-methods. The least time is required by ziggurat-method for imitation modelling. This method should be recommended as the most appropriate for noise immunity estimation when modelling digital communications systems with ISI and AWGN with use of Monte-Carlo method. By it's accuracy, Box-Muller method is similar to ziggurat-method, but it requires several more time (for about a quarter).

In order to achieve as small bit error probability value as it's possible with a little additional computing time loss, the data reading delay from the survivor paths of the Viterbi detector state diagram should be chosen by following the expression $W=5\times V$, rather than $W=5\times(V-1)$ one.

REFERENCES

[1] В. Б. Давенпорт, В. Л. Рут, *Введение в теорию случайных сигналов и шумов*. Москва: Из-во Ин. литературы, 1960.

[2] Michel C. Jeruchim, Philip Balaban and K. Sam. Shanmungan,, *Simulation Communication Systems: Modeling, Methodology, and Technique*, 2-nd edition. –N.Y.: Kluwer Academic, 2000

[3] Дж. Форсайт, М. Малькольм, *Машинные методы математических вычислений*. Москва: Мир, 1980.

[4] George Marsaglia and Wai Wan Tsang, "*A fast, easily implemented method for sampling from decreasing or symmetric unimodal density functions*", SIAM Journ. Scient. and Statis. Computing, vol. 5, 1984, pp. 349-359.

[5] George Marsaglia and Wai Wan Tsang, "*The Ziggurat Method for Generating Random Variables*", Journal of Statistical Software, vol. 5, Issue 8, 2000. Available: http://www.jstatsoft.org/v05/i08/paper [Accessed: Sept. 29, 2010]

[6] G. David, Jr. Forney, "*The Viterbi Algorithm*", Proc. IEEE, vol. 61, no.3, March 1973, pp. 268-279 (Дж. Д. Форни-мл., "*Алгоритм Витерби*", ТИИЭР, 1973, т. 61, №3, pp. 12–25).

[7] G. David, Jr. Forney, "*Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference*", IEEE Trans. Inform. Theory, IT-18, May 1972, pp. 363-378

[8] М. Вернер, *Основы кодирования*. Москва: Техносфера, 2006.

[9] Дж. Прокис, *Цифровая связь*. Москва: Радио и связь, 2000.

[10] Alberts Zeļenkovs, *Informācijas pārraides un ciparu sakaru sistēmu pamati: Mācību grāmata*. Rīga: RTU Izdevniecība, 2008. (Applications 6 and 7)

[11] B. A. Wichman and I. D. Hill, "*Algorithm AS 183: An Efficient and Portable Pseudo-Random number Generato*r", Applied Statistics, vol. 31, 1982, pp.188-190.

[12] "*Описание функции RAND в Excel 2007 и в Excel 2003*". Available: http://support.microsoft.com/kb/828795/ru [Accessed: Sept. 29, 2010]

[13] Л. Дж. Комри, *Шестизначные математические таблицы Чемберса*. М.осква: Наука, 1964.

Albert Zelenkov was born in Moscow, Russia in 1939. In 1963 he graduated Riga Civil Aviation Engineering Institute (also known as RIIGA or RAU), Riga, Latvia with an Engineer of Radio electronics. In 1967 he graduated RIIGA with Sc. Eng. Doctor degree (first higher degree in USSR) in signals spectrum analysis field. In 1992 he graduated St. Petersburg State University of Aerospace Instrumentation (LIAP), St. Petersburg, Russia with a High Doctor Degree of Engineering Science (second higher degree in USSR) in digital signal processing field and in 1993 graduated RAU with a doctor habilitation degree in engineering. Currently is a leading researcher of Riga Technical University (RTU) ETF TET department, Riga, Latvia.

Sergey Zelenkov was born in Riga, Latvia in 1980. In 2001 he graduated Riga Technical University (RTU) with a Bachelor of Engineering Science degree in Radio Electrical Engineering. In 2003 he graduated RTU with a Master of Engineering Science degree in Telecommunications. Since 1999 he has been working in Institute of Solid State Physics (CFI), Riga, Latvia as a software and Oracle Data Base programmer. Since 2000 he has been working in Radar Systems, Inc., Riga, Latvia as a software programmer and an embedded systems engineer in telecommunication and radiolocation field. From 2001 and up to 2004 he had participated in International European Project - Geowaters (geowaters.com) as an engineer.

Alberts Zelenkovs, Sergejs Zelenkovs. Gausa trokšņu ģeneratora izvēle bināru kanālu trokšņnoturības novērtēšanai ar Viterbi detektoru.
Šajā rakstā tiek apskatītas 3 metodes gadījumskaitļu formēšanai ar Gausa sadalījumu N(0,1), izmantojot gadījumskaitļus ar vienmērīgu sadalījuma likumu un nulles vidējo vērtību, ciparu sakaru sistēmu datora imitācijas modelēšanā. Tādas metodes ir: gadījumskaitļu summēšanas metode, kas balstās uz centrālo robežas teorēmu, Boksa-Mullera metode un Marsaļja-Tsanga zikurāt-metode (piramīdas metode). Dažādas gadījumskaitļu ģenerēšanas metodes tika izmantotas Gausa trokšņu modelēšanai bināro datu pārraides ciparu sakaru sistēmas uztvērēja ieejā ar diametrāli pretējo signālu izmantošanu (NRZ jeb BPSK veids) simetriskajos kanālos ar starpsimbolu interferenci un detektēšanu uztvērēja izejā ar Viterbi algoritmu. Kanāls ar trokšņiem un gadījuma rakstura binārajiem datiem tika

modelēts ar Monte-Karlo metodi pie uzdotās attiecības signāls pret trokšņiem (dB). Modelēšanas rezultātā tika iegūtas trokšņu ģeneratoru histogrammas, ka arī tika novērtētas precizitātes un skaitļošanas laika robežas iespējas dažādām Gausa trokšņu ģenerēšanas metodēm, ar kurām novērtē bināru datu pārraides ciparu sakaru sistēmu trokšņnoturību. Tika izpētīts, kā ietekmē trokšņnoturības novērtējumu dažādā vienmērīgi sadalīto saskaitāmo skaita izvēle, gadījumā kad tiek izmantota centrālā robežas teorēma. Tika apskatīta sakaru kanāla lineārā trakta dažāda veida impulsa reakciju ietekme, kas noved pie nuļļu rašanās kanāla frekvenču raksturlīknē. Tika precizēts datu nolasīšanas aizkaves laiks Viterbi detektora izejā, kas nepieciešams lai nodrošinātu pēc iespējas zemāku bitu kļūdas varbūtību pie nelielām skaitļošanas laika izmaksām.

Альберт Зеленков, Сергей Зеленков. Выбор генератора гауссова шума для оценки помехоустойчивости двоичного канала с детектором Витерби.

В данной статье рассматриваются три метода получения случайных чисел с гауссовским распределением N(0,1) из случайных чисел с равномерным законом распределения и нулевым средним значением с целью использования при компьютерном имитационном моделировании цифровых систем связи. Такими методами являются метод суммирования случайных чисел, основанный на центральной предельной теореме, метод Бокса и Мюллера и зиккурат-метод (метод пирамиды) Марсальи и Тсанга. Различные методы получения случайных чисел использовались для моделирования гауссовых шумов на входе приемника цифровых систем передачи двоичных данных на основе противоположных сигналов (типа NRZ или BPSK) в симметричных каналах с межсимвольной интерференцией и детектированием на выходе приемника с помощью алгоритма Витерби. Канал с шумами и случайным потоком двоичных данных моделировался методом Монте-Карло при заданном отношении сигнала к шуме в dB. В результате моделирования получены гистограммы генераторов шума и оценены предельные возможности по точности и вычислительному времени различных методов генерирования гауссова шума, используемых при оценке помехоустойчивости каналов связи. Исследовано влияние на оценку помехоустойчивости выбора разного числа суммируемых случайных чисел с равномерным законом распределения при использовании центральной предельной теоремы. Рассмотрено влияние выбора разного типа импульсных реакций линейной части канала связи, приводящих к нулям в частотной характеристике канала. Уточнено время задержки считывания данных на выходе детектора Витерби, необходимое для получения предельно малой вероятности битовой ошибки при небольших затратах вычислительного времени.