

# MOBILE SOFTWARE PLATFORM FOR RADIO INFORMATION SYSTEMS STATISTICAL MODELING

**Alexander A. Totilas,**

*Department of Nanotechnology in Electronics, Kazan National Research Technical University named after A.N. Tupolev-KAI,  
KNRTU-KAI, Kazan, Russia;*

*German-Russian Institute of Advanced Technologies (GRIAT), [totilas.aleks@gmail.com](mailto:totilas.aleks@gmail.com)*

**Renat F. Zaripov,**

*Department of Nanotechnology in Electronics, Kazan National Research Technical University named after A.N. Tupolev-KAI,  
KNRTU-KAI, Kazan, Russia;*

*German-Russian Institute of Advanced Technologies (GRIAT), [rfzaripov@kai.ru](mailto:rfzaripov@kai.ru)*

**Marat M. Fatykhov,**

*Department of Nanotechnology in Electronics, Kazan National Research Technical University named after A.N. Tupolev-KAI,  
KNRTU-KAI, Kazan, Russian Federation;*

*German-Russian Institute of Advanced Technologies (GRIAT), [mmfatykhov@prof.kai.ru](mailto:mmfatykhov@prof.kai.ru)*

DOI: 10.36724/2664-066X-2020-6-4-2-10

## ABSTRACT

For the static research purposes in the field of radio signals reception under freely fluctuating interference conditions, specialized software for the Android platform based on the Kotlin language has been developed using. The program design allowing to work with both the information recorded by the radio receiver analog system and the signals obtained using the simulation model. CDMA system operation under Gaussian white noise and chaotic pulse interference conditions has been simulated. The dependences of the bit error probability on additive radio channel effects of a different nature were obtained. The work originality is an attempt to implement a software-defined binary signal processing system on an accessible platform.

**KEYWORDS:** *software defined radio, chaotic pulse interferences, code division multiple access, the clean architecture, mobile platform.*

## I. INTRODUCTION

Nowadays, telecommunications such as mobile cellular communication networks and wireless local area networks (WLANs) are actively developing. The number of users in such networks is constantly increasing, and at the same time, the requirements applied to communication systems, necessary for comfortable communication, are growing - for voice, text messages, image transmission, video communication, the Internet surfing and other services. Over time, the load on the network grows the requirements for speed and quality of data transferring. At the same time, more and more wireless networks are appearing, which leads to very limited frequency resource decrease and rising cost, as well as to an increase in noise and interference in the used radio frequency ranges. All of this makes the research in the field of wireless data transmission efficiency improving very relevant.

A promising and relevant solution to the above problems is the use of so-called complex signals, including broadband signals, which allow occupied frequency bands to be used, while introducing minimal interference to existing systems.

The use of such signals is associated with complex processing, calculations that are difficult to implement with analog elements, but which are easily solved using digital systems. In addition, the rejection of analog processing in favor of digital allows achieving a number of advantages, such as increased noise immunity, long range with relatively low transmitter power, data encryption, and much other functionality. [1]

One of the ways to study the digital communication network capabilities is its computer simulation. It provides available tools with minimal cost and sufficient accuracy for determining the characteristics, for experiment on models close to real objects, when they are inaccessible.

### Software Defined Radio System

A software-defined radio (SDR) system is a radio communication system where the most of the components (amplifiers, mixers, filters, detectors, etc.) are implemented on a personal computer, embedded system or other programmable digital equipment instead of hardware. Such systems require great computing capabilities, however, modern personal computers, gadgets, embedded systems with their relatively low cost and wide distribution have a sufficient number of processor cores with high clock frequencies - up to 5 GHz per core.

The simplest SDR system can consist of a personal computer with a sound card (or other analog-to-digital converter), which has connected radio-frequency receiving system. A significant part of the signal processing is supported by the processor, not in specialized hardware components (electrical circuits). This design allows you to receive and transmit a very large number of different radio protocols only on the single software basis.

Software-defined radio systems are widely used for military purposes, as well as in cellular mobile communication networks, since they must serve a wide range of radio protocols in real time.

According to engineers, software and hardware developers, providers from the International Organization for Wireless Innovation (The Wireless Innovation Forum, formerly SDRForum), in the long run, SDR will become the dominant technology in radio communications. In addition, SDR, together with programmable antennas, allows cognitive radio -the further telecommunications development.

Previous types radio systems developers had to deal with such problems as the limited frequency resource, its occupancy and the presence of mutual interference. However, software-defined radio systems can be flexible enough to get around these problems in various ways, for example, using spread spectrum technologies that allow multiple transmitters to transmit at the same time at the same frequencies with very little mutual interference. In addition, error detecting and correcting algorithms are introduced that make it possible to completely get rid of the interference effect.

Turning in the signal direction Software-defined antennas application allows receivers to better suppress interference from other directions, and therefore receive weaker signals.

In cognitive radio technologies, each system measures the current spectrum conditions and transmits this information to other similar systems, so transmitters can avoid mutual interference, for example, by choosing unused frequencies. Also, each radio system, according to this technology, is associated with a geolocation data for given area spectrum occupancy information obtaining and tuning the effective frequencies and transmitter power.

Based on the communication with the receiver, transmitter power dynamic tuning reduces it to the

required minimum, solving the near-far problem and reducing interference, as well as reducing the power consumption from transmission on portable devices.

### System design general principles

In the system development, it is very important to pay attention to its architecture. So, if the system is designed as a single block, it is not free for expansion, modifications. It is difficult to test it, verify all its functions reliability, but it is easy to allow a flaws and shortcomings. Any change in one part of it requires processing of the entire system, and this introduces additional labor and financial costs. In addition, the design of such system carries a high complexity itself.

The modularity concept, based on the problem decomposition helps to help solve these problems. According to it, the system must be divided into functionally complete modules. Thus, the system can be easily supplemented and improved by adding and/or changing these blocks.

At the highest level, the SDR system can be divided into hardware and software parts. It is assumed that in this way each of them can be developed independently. In their turn, these parts should also be decomposed.

Historically, equipment has been developed from functionally complete units, thereby fulfilling the principle of modularity. However, this mainly concerns only design, but often does not apply to the period of operation. The components of the finished product are usually rigidly connected with each other and the environment structure that unites them (circuit board): the procedure for replacing them (soldering) is relatively complicated and time-consuming, and it is also difficult to configure the functionality. In part, these problems are solved by connecting parts - connectors and interfaces connecting devices to each other.

Therefore, in this projected system, the hardware is completely represented as several interchangeable modules having a single unified connector for connection. The hardware module is completely replaced depending on the task and the required functional purpose.

In the developed system, the equipment is divided into a data source (for example, radio frequency) and a digital processing center. Different source implementations may, for example, work with different frequencies and powers, but must have the same connectors and a single interface for "communication" with the digital part.

A digital device can also be easily replaced in the next cases: it has the same connection interface, supports software for data generating and processing, and has sufficient performance for real-time operation.

The software part takes over most of the functions previously performed by the hardware equipment, which significantly reduces the size and cost of the device. Also, software with proper design is divided into modules, which sometimes makes it possible to change the functionality and processing algorithms on-line, and in the future will make it easy to expand and modify. In addition, the modular architecture of the software simplifies its testing, which means entire system overall reliability and accuracy increase.

## System elements implementation

The developed system in its hardware part contains two devices: the first is the data source for the system, and the second is processing and display device.

An important part of the system is the equipment devices interaction interface. For physical wiring task, a universal serial bus interface is suitable as a version, since it is widespread - it is the standard for most modern digital devices, also, it can have high data transfer speeds - up to 20 Gb / s (for USB 3.2 Gen 2x2 ) [2]. An important connector feature is also power transmission to the device. In cases where the wired connection between devices is impossible, irrational or inconvenient and the distance between them is no more than 10 meters, wireless connection through Bluetooth modules can be used.

A common digital interface between parts of the equipment allows using a wide design implementations range, depending on the task and capabilities.

For example, if the goal is radio data receiving and subsequent processing (in this case, the devices act as an SDR real-time system), then a RF device is connected to the digital equipment. In general, an RF hardware can have one or more antennas, a preselector, a local oscillator and a mixer for transferring the signal to an intermediate frequency, a band-pass filter for suppressing the mirror channel, an ADC, a USB converter, and/or a Bluetooth module. The radio receiver of the system is generally connected to the processing device via USB for a more reliable connection and power supply.

There may be situation when a radio signal is received by some device and its digitized data is stored in memory, but processing at the reception time is impossible (for example, due to the fact that the ADC has high bit depth and the processing device has low performance - thus the data stream does not manage to be processed in real time). The solution to this problem is to use a storage device, such as a flash drive, as a data source. Then the processing device reads the information directly from the memory via the USB connector.

A data processing device can be any computing device that supports system software and is capable of performing the necessary calculations in a reasonable amount of time. It is possible to use:

- single-board computers;
- FPGA
- PC
- mobile devices (smartphones).

Single-board computers are mobile and cheap, but they usually have low performance and RAM.

FPGAs are portable and have a great performance, but require much more time for software development, while programming uses hardware description languages that do not allow developing a complex user interface.

PCs have high performance, but high cost and large size. In addition, they are stationary, tied to a general power supply system.

Modern mobile devices have several advantages - they are portable, have small dimensions, a large amount of RAM, low cost, and are quite productive (a mid-range smartphone usually has 8 processor cores with a clock speed of about 1.8-2 GHz) and combine a display device.

In addition, due to their widespread prevalence, they are so included in the everyday life of people that almost every person has it. Thus, a mobile phone (smartphone) was selected for use as system data processing equipment.

An important feature of the smartphone is the presence of a built-in touch screen, which not only allows you to display the information necessary for the operator, but also serves as a controlling element, which means it allows the user to interact with the software. This provides software configuration depending on the task and the type used data source.

Currently, smartphones with operating systems (OS) iOS and Android are common. The choice of platform is considered from the device and software development opportunities availability point of view - devices with the iOS operating system are much more expensive and less common [3]. In addition, for the software full development and distribution, this platform requires a PC with a similar OS from the manufacturer, which requires huge additional costs. Thus, an Android device was chosen due to its widespread use, low cost and the free unlimited software development possibility.

## SOFTWARE DEVELOPMENT

Based on the processing system hardware digital part implementation - a mobile device based on the Android operating system - software can be developed in the Java, Kotlin, and JavaScript programming languages (through the React Native framework).

JavaScript is not officially supported by the Android platform, the React Native framework does not have the same functionality as native languages, in addition, it is unstable and often leads to application inoperability. Therefore, this language is not suitable for the mobile devices complex and resource-intensive software package development.

Java is the official programming language for Android, stable and has a large number of libraries from third-party developers, which allows quickly adding various functionality to the application. However, this language also has a number of drawbacks: it is strongly typed and verbose, requires a lot of development time.

Kotlin is also the official (and recently, the main) programming language for Android applications. Compared to Java, it is concise, so the amount of code base is significantly reduced, which means that development time is also reduced. In addition, it contributes to a significant reduction in the number of errors, which means it increases the stability of the software. Also, this language, like Java, is compiled into byte code for the Java Virtual, therefore it allows the use of third-party libraries written in the Java language.

Thus, based on the advantages described above, Kotlin has been chosen for software development.

### Software architecture

The system software design based on the “The Clean Architecture” [4] concept, proposed by engineer Robert Martin (Fig. 1).

This concept means dividing the entire program into several main layers representing various program areas:

- Entities layer - the core of the architecture, contains business logic objects. This can be either classes with methods or sets of data structures. They contain the most general, high-level rules.
- Layer of business rules — application-specific laws and processes that manipulate data flows between entities.
- Interface layer - an intermediary between entities with business rules and an implementation layer, designed to transform data into a form suitable for layers.
- Implementation layer - includes elements of an objects specific implementation, a graphical interface, external libraries and frameworks, a connection to the network and external interfaces, databases - that is, everything that depends on the platform.

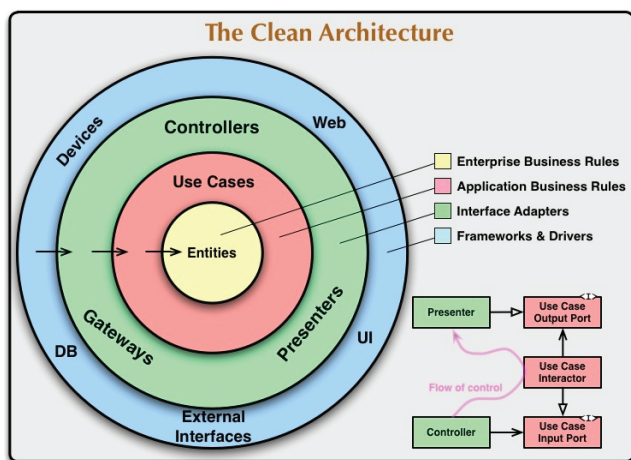


Fig. 1. “The Clean Architecture” concept by Robert Martin

Between layers it is necessary to observe the following Dependency Rule: external layers depend on internal, but not vice versa. For example, the business logic layer uses the central layer entities, while the latter do not interact with the business logic in any way.

Thus, this architecture fulfills the following requirements necessary for reliable, easily extensible, and modifiable software:

- the software core does not depend on third-party libraries, frameworks, databases;
- the software core does not depend on the user interface execution;
- software is easy to test.

Through this architecture and JVM-based language, the software core can be distinguished and used both for a mobile solution and for use in a desktop PC.

### Data source

The designed system is developed both for practical purposes, such as the complex radio signals reception and processing, and for research - the study of multi-user access by encoding with orthogonal or pseudorandom sequences and their noise immunity under the noise, interference, and reflections conditions. Therefore, the system software implements two types of operation modes corresponding to two types of data source:

- received signal processing mode (data "from the outside");
- simulation mode (software data generation).

For the SDR system, the data source is a radio signals transmitted through radio channel with noise and interference. The mixture of noise, interference and signals received by the RF device antennas can pass through pre-filtering, then it is digitized in the hardware ADC. Thus, a digital signal with a flow rate equal to the ADC sampling frequency is received at the mobile device input via USB or Bluetooth, while the number of bits in the word that determines the signal level is equal to the ADC bit depth.

The software is developed for receiving and modeling radio signals with various parameters used in communication systems (including nonstandard ones), therefore a large number of various settings are provided in the data source module:

- ADC sampling rate;
- ADC bit depth;
- channel code type;
- carrier frequency;
- number of channels;
- data flow rate;
- frame length;
- number of frames;
- code length.

For modeling purposes, it is also possible to generate additive white Gaussian noise (AWGN) with given signal to noise ratio and chaotic pulse interferences (CPI) with a scarcity factor.

First, according to the user entered settings, the software generates random digital data for each channel with a uniform zero probability distribution.

Then, in the encoder module, data is encoded with a code corresponding to each channel. Encoding is the modul 2 addition for each data frame bit  $a$  with the entire sequence of code. Thus, the length of the encoded data is:

$$L_{cd} = L_d \cdot L_c \quad (1)$$

where  $L_d$  – data frame length,  $L_c$  – code sequence length.

The encoded data from each channel undergoes a change in the bipolar form. In this case, the pulse durations correspond to half the data flow rate selected by the user (the symbol rate is a half the bit rate). Then the bipolar pulse sequences are linearly combining, forming a group signal.

Next, the generator block creates a carrier harmonic with a frequency specified by the user, which, at the same time as the signal from the encoder, is supplied to the QPSK modulator block. The following operations are performed here:

- The pulse sequence is divided into two streams: even pulses go to channel I, and odd pulses go to channel Q (given that the numbering starts from zero).
- Channel I data is multiplied with the carrier harmonic, and channel Q data is multiplied with the same harmonic, but 90 degrees out of phase.
- Q and I components add up to form a QPSK signal and are output.



Obtained signal is additively combining with noise and interference. The mixture is displayed on the oscilloscope on the source module page and is forwards to the demodulator unit.

### Noise and interference generation

In the signal source module simulation section, the user has the ability to turn on/off the noise and interference under the radio channel simulation, as well as configure their parameters.

When the noise is turned on, the user adjusts the signal-to-noise ratio. Each time the parameter is changed, an AWGN is generated. Generation is based on the function included in the Kotlin programming language standard library. All samples obtained by this function have the same probability density of the form of a Gaussian distribution with zero mean and unit dispersion. The noise power is controlled by multiplying the density of the noise power coefficient by a random value obtained by the above function.

For the continuous analog white noise according to the formula [5]:

$$P_N = \int_{-\infty}^{\infty} S_N(f)df = \int_{-\infty}^{\infty} \frac{N_0}{2}df = \infty \quad (2)$$

the dispersion is infinite, however, in this case, the noise is represented by discrete samples, and the frequency range at which the noise is considered white is limited. Thus, its dispersion is finite, and noise can be implemented by software.

The required noise power calculation based on the user specified SNR and bit rate, as well as the known QPSK signal average power. SNR in software is interpreted as the normalized signal-to-noise ratio in decibels, therefore, based on the formula:

$$\frac{E_b}{N_0} = \frac{S}{N} \frac{W}{R} \quad (3)$$

where the energy of the bit  $E_b$  is the energy per one bit of information [6]:

$$E_b = ST_b = \frac{S}{R} \quad (4)$$

and the spectral density of the noise power  $N_0$  is the power per unit frequency interval in the noise spectrum having an energy amount.

Then, the power spectral density is expressed as follows:

$$N_0 = \frac{E_b}{SNR_{db}} = \frac{S}{10^{\frac{SNR_{db}}{10}} \cdot R} \quad (5)$$

Thus, when interferences are turned off and there are no channel signals in the simulated broadcast, white Gaussian noise is observed, in which the power spectral density is directly proportional to the signal-to-noise ratio.

When interferences are turned on, the user has the ability to adjust the sparseness coefficient and signal-to-noise ratio. Turning on and changing the parameters

starts the generation of CPI in the simulated radio channel.

First, for simulation each data point, taken with a sampling frequency (specified by the user), the value of the same function, which was used to generate the AWGN, from the Kotlin library is calculated. Then it is compared with the sparseness coefficient, which in this case is the threshold of the pulse – if the function value is greater than the coefficient, then it goes on for further processing, otherwise the generation process for this data point is completed and the interference pulse does not occur.

Further, values exceeding the threshold are multiplied by a power factor expressed from a known signal-to-noise ratio.

Thus, when the noise is turned off and there are no channel signals on the air, a sequence of pulses with chaotic amplitude and an interval between them is observed, the average value of the pulses occurrence frequency being inversely proportional to the rarefaction coefficient, and the amplitude is directly proportional to the signal/noise ratio. It should also be noted that the number of generated pulses for the considered time interval approximately corresponds to the Poisson distribution law.

### Demodulation

The signal data received through USB or Bluetooth, or generated by the program, enters the demodulation module.

In it, QPSK demodulation is carried out in two channels, each of which is a demodulation of the BPSK signal. Thus, in the module, data is divided into two streams:

I-channel and Q-channel. The input signal in the I-channel is multiplied with the signal of the harmonic signal generator generating the cosine. The frequency of the generator is user-configurable and must be equal to the carrier frequency of the input QPSK signal for proper demodulation. As a result of multiplication, the sum of the double frequency harmonic and the constant component, proportional to the phase difference between the signal of the generator and the input signal, is formed [7].

Next, the resulting mixture enters a digital FIR filter, where noise is suppressed and a constant component is extracted. It is programmed to operate as a low-pass filter (LPF) using the ideal pulse response formula:

$$\begin{aligned} h_d(n) &= 2f_c \frac{\sin(n\omega_c)}{n\omega_c}; n \neq 0 \\ h_d(n) &= 2f_c; n = 0 \end{aligned} \quad (6)$$

The ideal filter characteristic is an infinite decaying series of values, however, for practical FIR use, this series must be limited, and the so-called Gibbs phenomenon appears – a transitional frequency band appears between the suppression and transmission bands, and emissions forms on the transfer characteristic.

Multiplication the pulse response by the window smoothing weight function helps to solve this problem. For this, the following are used [8]:

- Hannah function

- Hamming function
- Gauss function
- Blackman function

For each window, the resulting signal spectrum minor lobes level is different, and can act as a parameter characterizing the quality of filtering.

The filtering part is also user-configurable – its order, cutoff frequency and weight function are set.

After filtering, the signals from both channels enter the data processing module. Since the filter introduces a delay, the processing start time is also software-based delayed to avoid errors.

In the module, the signals are divided into time sections of length  $T_s$ , each of them is integrated over time. Thus, data array that carries information about the average signal value at each such section is obtained. Next, two arrays of in-phase and quadrature channels are combined as follows:  $D = [i_0, q_0, i_1, q_1, \dots, i_n, q_n]$ , where  $i_k$  is the  $k$ th value of the in-phase values array,  $q_k$  is the  $k$ th value of the quadrature values array.

Then, the resulting data array passes to the decoder module.

### Decoding

A digital values array is supplied to the input of the decoding module, which is a group signal – a mixture (linear sum of encoded data) of each active communication channel information.

To decode data, it needs first to be determined which channels should be processed. The developed software allows defining channels in several ways:

- manual channel definition;
- all possible channels identification;
- automatic channel detection.

In the manual user determination case, the channel codes that need to be decoded are manually entered. When selecting all channels for subsequent processing, all possible channels are determined for a user-selected family of codes of a specified length: for example, for Walsh codes of length 16, 16 channels with all possible codes (including service sequences) are determined. If the user selects the automatic channels detection, the program sets the channels whose codes give the maximum response to the cross correlation function, determined by the beginning of the transmitted sequence.

To highlight the transmitted information messages of each channel, their code sequences are converted into bipolar values according to the following rule: “-1” corresponds to the zero bit, and “1” corresponds to the single bit.

The values of the group signal are multiplied by the values of the bipolar code sequence with a period equal to the length of the code. The result of the multiplication will depend on the presence of a channel message: if the information has been transmitted through channel, then a bipolar sequence was obtained, where each value corresponds to a channel information bit (a threshold considered is a zero level: a negative value is a zero bit, a positive is a single), otherwise the output is a sequence of zeros.

Under the influence of noise and interference, the group signal is distorted. If we assume that the total set of noise is distributed according to the Gaussian law according to the expression of the probability density:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right) \quad (7)$$

where  $\sigma$  is a root-mean-square deviation, and  $\sigma^2$  is a process dispersion, demodulated mixture defines as:

$$x(t) = S_i(t) + n(t), i = 1..k \quad (8)$$

where  $S_i(t)$  is an information signal in the form of binary bipolar symbols with amplitude  $a$ ,  $n(t)$  is zero-mean AWGN.

Based on the last 2 expressions, the nominal probability densities of receiving signals  $S_1$  and  $S_2$  can be expressed as:

$$p(x|S_1) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-a_1}{\sigma}\right)^2\right) \quad (9)$$

$$p(x|S_2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-a_2}{\sigma}\right)^2\right) \quad (10)$$

These densities are called the likelihood densities of signals and show the probability density of the random signal  $x(t)$  (provided that the corresponding symbols are transmitted (Figure 2)).

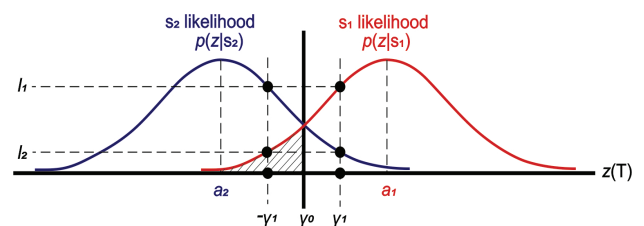


Fig. 2. Symbols receiving nominal probability densities [9]

As can be seen from the graphs (Fig. 2), the densities do not have clear boundaries and intersect with each other, as a result of which errors may occur in determining which symbol was received. Despite this, the final solution (choice of hypothesis) is still issued in comparison with the threshold – the zero level ( $\gamma_0$ ).

The program decoding module enables the display of bits that are supposedly erroneous. This is done by entering another threshold value that defines the zone of uncertainty ( $\pm\gamma_n$ ). All values are compared with this threshold: a value is considered erroneous if it's modulo less than the  $\gamma_n$ . Erroneous bits are remembered by the system and are subsequently highlighted in red when displayed. It is possible to use the error correction scheme in further.

### Statistics collection

The statistics module displays the data collected as a result of coding, modulation, demodulation and decoding, as well as graphing the digital communication network characteristics.

If the signal source is channels generated by the program, then some parameters of the system are known in advance, in the case of signal processing from an external device or from a data file, some parameters are determined during the program operation.

The statistics window shows the number of active channels, the total number of transmitted bits and received bits per communication session (includes a certain number of frames), the number of bits detected by the decoder as erroneous, and the percentage of erroneous bits to the total number of received (which is the current bit error probability).

The module has the ability to collect statistics for plotting the BER/SNR dependence. The input parameters entered by the user in this case are the initial and final values of the SNR in decibels, and the desired graph samples number. The counting process is carried out by repeating the simulation of the reception session with the parameters of the source, demodulator and decoder specified in advance, and at each iteration, noise power is swiped corresponding to the current SNR:

$$SNR_i = SNR_0 + i \frac{(SNR_K - SNR_0)}{K} \quad (11)$$

where  $i = 0..K-1$  is the iteration number;  $SNR_0$  – initial SNR;  $SNR_K$  – final SNR;  $K$  – number of graph points.

It is necessary to note that it is not recommended to specify a large number of points – this entails a long processing, collecting statistics and plotting.

## RESEARCH IN DEVELOPED SOFTWARE

The developed software for a software-defined radio system allows researches in the field of noise immunity and data transmission using the multiple access method such as code division multiplexing (CDMA), when several communication channels are operating simultaneously.

### Initial data for the study

Data Source Settings:

- A. Operation mode: simulation.
- B. Sampling frequency: 10 MHz.
- C. Channel settings:
  - Number of channels: 4.
  - Channel codes type: Walsh codes.
  - Code length: 4 bits.
  - Carrier frequency: 500 kHz.
  - Data transfer rate: 100 kBit / s.
  - Information parcel length: 100 bits.
  - Number of frames: 1.

Demodulator Settings:

- A. Generator frequency: 500 kHz.
- B. FIR Filter Settings:
  - Order: 10.
  - Cutoff frequency: 1000 Hz.
  - Weight function: Blackman.
- C. Filter delay compensation: 0.1.
- D. Information parcel length: 100 bits.

E. Code Length: 4 bits.

F. Data transfer rate: 100 kBit.

Decoder Settings:

A. Detection mode: manual.

B. Number of channels: 4.

C. Channel codes type: Walsh codes.

D. Code Length: 4 bits.

E. Threshold signal level: 0.3.

### Noise immunity study

The performance calculation module calculates the theoretical  $BER_{theor}$  and practical BER in percent, the theoretical system throughput in kbit/s and the real system transmission rate D in kbit/s at various levels of the  $E_b/N_0$  ratio. AWGN acts as a noise model.

The theoretical bit error probability for systems with coherent QPSK modulation is calculated using the following formula:

$$P_{err.theor.} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad (12)$$

$$Q(x) = \frac{1}{2\pi} \int_x^\infty \exp\left(-\frac{t^2}{2}\right) dt \quad (13)$$

Theoretical throughput is calculated by Shannon's theorem as follows:

$$C = W \log_2 \left(1 + \frac{E_b}{N_0}\right) \quad (14)$$

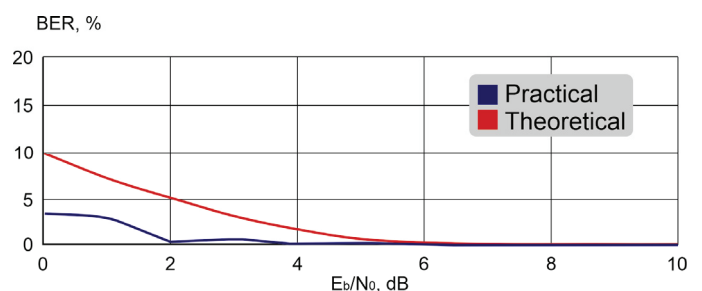
where  $W \approx 1/T_b$  is available bandwidth in Hz.

Numerical simulation results are collected in Table 1.

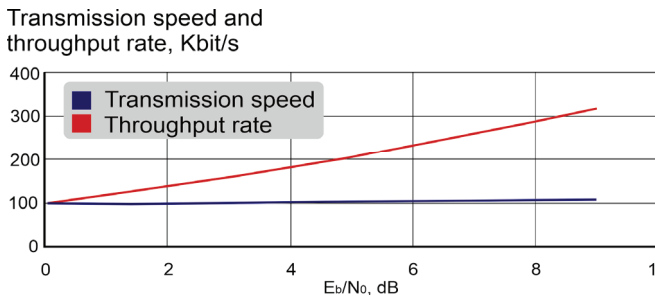
Table 1  
BER dependence from  $E_b/N_0$

$E_b/N_0$ , dB	$P_{err.theor.}$ ( $BER_{theor}$ ), %	$P_{err}$ (BER), %	C, kbit/s	D, kbit/s
0.0	9.571986	3.5	100.000	96.500
1.0	7.087514	3.0	117.564	97.000
2.0	4.914345	0.5	137.010	99.500
3.0	3.138237	0.75	158.268	99.250
4.0	1.806955	0.25	181.225	99.750
5.0	0.913139	0.25	205.737	99.750
6.0	0.391355	0.0	231.646	100.000
7.0	0.136200	0.0	258.781	100.000
8.0	0.036432	0.0	286.979	100.000
9.0	0.006988	0.0	316.080	100.000

The obtained dependencies are presented in Fig. 3 and 4.



**Fig. 3.** Theoretical and practical BER dependencies on the  $E_b/N_0$  ratio Graphs



**Fig. 4.** Bandwidth and bit rate dependencies on the  $E_b/N_0$  ratio

Thus, according to the obtained results, developed model does not meet theoretical expectations. In practice, the simulation program achieved half of the probability bit error over the entire  $E_b/N_0$  range. This phenomenon can be explained by a very limited selection and a small amount of transmitted data (100 bits for each of the four channels), as well as imperfect program algorithms. At the same time, the transmission speed drops slightly with a high noise level compared to the declared one, but does not exceed the bandwidth values, which is normal.

As a result, according to the developed model, the system is capable to correctly receive, demodulate, and decode data transmitted by at least four channels in one frequency band under noise conditions with an  $E_b/N_0$  ratio more than 6 dB.

Since the software can simulate not only AWGN, but also CPI in the radio channel, we study the effect of chaotic pulse interference on the system with the same settings. In this case, this interference is a model of other radio transmitters that may interfere with signal transmission.

Measurements are taken in the absence of background white noise in order to investigate only the interference effect. The simulation program allows adjusting the interference by changing the ratio of the energy of the bit  $E_b$  to the spectral density of the interference power  $P_0$  and the so-called sparseness coefficient  $K_s$  inversely proportional to the interference pulses appearance frequency, that is, the higher its value, the wider pulses appear in the channel. Studies are conducted to these parameters combining in order to identify an influence to the system tendency for each of them. The results are listed in Table 2.

Table 2

Bit error probability depending on CPI parameters

$N_b$	$E_b/N_0$ , dB	$K_s$	$P_{err}$ (BER), %
1		0.4	4.00
2		0.6	3.75
3		0.8	2.50
4		1.0	2.75
5		1.2	2.00
6		1.5	1.00
7		0.4	1.00
8		0.6	1.00
9		0.8	1.00

$N_b$	$E_b/N_0$ , dB	$K_s$	$P_{err}$ (BER), %
10		1.0	0.50
11		1.2	0.00
12		1.5	0.00
13		0.4	0.00
14		0.6	0.00
15		0.8	0.00
16		1.0	0.00
17		1.2	0.00
18		1.5	0.00

According to the results of the study, it is clear that with an increase in the  $E_b/N_0$  ratio, the error probability decreases, and at the level of 6 dB the system no longer produces influential amount errors. That is, the probability is too small to detect it with a selected transmitted information amount. The study also shows that with decreasing rarefaction coefficient, the error probability increases.

In general, the error probability with CPI on the communication channel is lower compared to the influential AWGN for the same signal-to-noise and noise ratio, respectively.

The smaller CPI influence on the system compared to the AWGN can be explained by the fact that pulsed noise occupies a smaller band in the spectrum and is narrow-band compared to the broadband signals used in CDMA systems.

### Conclusion

During the project implementation, the processes modeling theoretical basis, the nature of complex signals, noise and interference was studied. The wireless digital communication systems receivers and transmitters generalized structural diagrams were studied and the main characteristics of such systems were also considered. Phase manipulation and, in particular, quadrature modulation, which has great potential and is widely used in existing wireless networks, were examined in detail. The basic multiple access methods, necessary for the systems with a large number of users (subscribers) developing possibility were studied. Code separation of channels was considered in more detail as the most interesting, relevant and effective channels separation solution, and also some actual groups of channel codes with their implementation methods were studied.

The result of the work is the developed software, primarily intended for broadband CDMA signal receiver using QPSK at the presence of AWGN and CPI in the radio channel modeling. The program implements the data, noise and interference generation processes, channel coding and decoding, QPSK modulation and demodulation, statistics collection and system characteristics graphing versus S/N ratios.

The software product, in addition to modeling, also acts for receiver digital data processing – it has the ability to read and process the received signal digitized by the ADC data from a memory (file with UTF-8 encoding). After refinement and optimization, for some cases, the software can process data not only from memory, but also directly from an external radio module connected via a USB interface to the processing device – thereby



forming a software-defined radio system. For SDR, the program is a key link – in addition to performing most of the work previously carried out using a large number of analog devices, it enables the operator to configure the entire system while performing to solve a wide range of problems, which also significantly reduces the final cost, facilitates easy development and the rapid spread of SDR.

Another important feature of the developed system is the platform on which it is built. The software operates on the mobile devices basis, which gives a number of advantages compared to traditional stationary equipment solutions. This makes the system:

- Portable – it has small dimensions and weight of devices.
- Modular – an easy replacement individually for both the radio module and the processing device.
- Cheap – relatively low prices for devices sufficient to operate the system.
- Affordable – mobile devices widespread use on the market, as well as their availability for almost every system user; easy distribution and installation of lightweight software; the ability to transfer the current configuration to the other devices.
- Productive – because of presence of a sufficiently powerful multi-core processor, a large amount of RAM in modern mobile devices.

Studies have been conducted to the system noise immunity at the certain settings. As a result of measurements with AWGN in the radio channel, the bit error probability indicators turned out to be less (better) than theoretical ones - this is explained by the small amount of transmitted information in the experiment, as well as the imperfection of the program algorithms. At the same time, the transmission speed slightly decreased with strong noise, but did not exceed the declared speed and throughput, which is normal. The bit error probability with the CPI in the radio channel was also measured at various interference settings. As a result of the study, it turned out that CPIs have less effect on the system than AWGN, due to the narrow-band interference. Moreover, with the tendency of the refraction coefficient to zero, CPI in form and parameters tends to AWGN.

Thus, the developed system is relevant for private research and practical purposes. It has great possibilities and, after refinement and optimization, can serve as the basis for educational or commercial use.

#### REFERENCES

[1] Fajzullin, R.R. (2011), An integrated approach to solving the mobile infocommunication systems signals processing multiprocessor devices structural synthesis problem [“Kompleksnyj podhod k resheniyu zadachi strukturnogo sinteza mul'tiprocessornyh ustrojstv obrabotki signalov

mobil'nyh infokommunikacionnyh sistem”] *Nelinejnyj mir*, Vol. 9. No. , pp. 78-85.

[2] USB Implementers Forum, Inc. (2019). USB 3.2 Specification Language Usage Guidelines from USB-IF. [online] Available at: [https://www.usb.org/sites/default/files/USB\\_3\\_2\\_Language\\_Product\\_and\\_Packaging\\_Guidelines\\_FINAL.pdf](https://www.usb.org/sites/default/files/USB_3_2_Language_Product_and_Packaging_Guidelines_FINAL.pdf) [Accessed 6 Dec. 2019].

[3] StatCounter (2020). Mobile Operating System Market Share Worldwide (1999-2020). [online] StatCounter Global Stats. Available at: <https://gs.statcounter.com/os-market-share/mobile/worldwide> [Accessed 10 Jun. 2020].

[4] Martin, R.C. (2019). The Clean Architecture. [online] Cleancoder.com. Available at: <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>.

[5] Nadeev, A.F., Podkurkov, I.A. (2015), “Adaptive estimation of non-Gaussian interference distribution parameters based on the application of the EM algorithm and its modifications”, [“Adaptivnaya ocenka parametrov raspredeleniya negaussovskih pomekh na osnove primeneniya EM-algoritma i ego modifikacij”] *Nelinejnyj mir*, Vol.13. No 8. pp. 64-72.

[6] Kadushkin, V. V. (2016), “CDMA systems non-Gaussian channel optimal receiving algorithm considering intersystem interference.” [“Algoritm optimal'nogo priema v negaussovskih kanalah CMDA sistem s uchedom vliyaniya vnutrisistemnyh pomekh”], *Zhurnal radioelektroniki*, No11, available at: <http://jre.cplire.ru/jre/nov16/index.html>.

[7] Fajzullin, R.R., Kadushkin, V.V., Zaripov, R.F. (2015), Poly-Gaussian algorithm for mobile communication systems channels joint demodulation-decoding [Poligaussovskij algoritm sovmestnoj demodulyacii-dekodirovaniya v kanalah mobil'nyh sistem svyazi] *Nelinejnyj mir*, Vol.13. No 8, pp. 4-9.

[8] Kadushkin, V.V., Fatykhov, M.M., Zaripov, R.F. (2016) “Sufficiency of a poly-Gaussian approximation of arbitrary probability distributions in the communication systems with mobile objects radio links”, *New Technologies, Materials and Equipment of the Russian Aerospace Industry: All-Russian Scientific and Practical Conference with International Participation, Collection of reports. Vol. 2* [“O dostatochnosti poligaussovoj approksimacii proizvol'nyh veroyatnostnyh raspredelenij v radioliniyah sistem svyazi s podvizhnymi ob'ektami.” *Novye tekhnologii, materialy i oborudovanie rossijskoj aviakosmicheskoy otrasli: Vserossijskaya nauchno-prakticheskaya konferenciya s mezhdunarodnym uchastiem. Sbornik dokladov. Vol 2*] *Izd-vo Akademii nauk RT, Kazan*, pp. 521-523.

[9] Sklar, B. (2016). *Digital communications: fundamentals and applications*. Upper Saddle River, Nj: Prentice Hall Ptr.

[10] Lerner, I.M., Fayzullin, R.R. and Chernyavskii, S.M. (2018), “To a Matter of Increasing the Spectral Efficiency of Phase Radio-Technical Data Transmission Systems Operating under Strong Intersymbol Interference”, *Russian Aeronautics. Vol. 61, No.1*, pp. 120-126.

[11] Zaripov R.F., Fayzullin R.R., Fatykhov M.M. and Kaduskin V.V. KNRTU-KAI. (2017), *Programma dlya modelirovaniya algoritmov obrabotki signalov v mobil'nyh sistemah svyazi* [A program for simulation of signal processing algorithms in mobile communication systems], Russian Federation, Patent № 2017615223.