Calculation of reliability of radio-electronic systems using the Monte Carlo method

Timofey Legkiy*

MIREA - Russian Technological University, Moscow, Russia

Abstract. In various fields of knowledge, modeling methods are used to study objects and systems. The use of modeling is especially effective in cases where direct study of the object or system itself is either impossible or leads to a large expenditure of resources. For example, to assess the reliability of objects located in low-Earth orbit. The most popular method for predicting reliability is the Monte Carlo method. The Monte Carlo method is widely used in various fields of science and technology to analyze the reliability of technical systems. It has also found application in reliability calculations and forecasting of radio engineering systems. The article discusses the Monte Carlo method in relation to the calculation of the reliability of electronic means. The algorithm of the solution is given, an example of calculation is considered.

1 Introduction

There are several methods for predicting the reliability of electronic devices [1-3]. Among them, the Monte Carlo method occupies a special place in predicting reliability. The Monte Carlo method is a numerical method of solving by modeling random variables in order to calculate the characteristics of their distributions [4-6]. Count de Buffon is framed as the founder of this method of research of mathematical and physical processes. He was the first to apply the method, based on random processes, to calculate the number "π".

This group of methods has a number of advantages [7-9]:

- relatively simple structure of the computational algorithm;
- the error is usually inversely proportional to the number of calculations.

In the most general form, the scheme of the Monte Carlo method looks like this:

Let it be required to find some value *I*. It is assumed that it is possible to construct a random variable ζ with mathematical expectation $E\zeta$ equal to *I* and variance *D(*. Constructing a sufficiently large number of values ζ an approximate value of the desired quantity is obtained:

$$I = \mathrm{E}\zeta \approx Z_n = \frac{\zeta_1+\cdots+\zeta_n}{n}, \tag{1}$$

where *n* is the number of sample values.

How accurate the approximate value will be depends on the value of *n*.

* Corresponding author: legki@rambler.ru

This paper will consider the so-called "rough" Monte Carlo method, which is based on a structural function that states that the probability of failure-free operation of the entire system is equal to the product of the probability of failure-free operation of all its components:

$$x_i = \begin{cases} 1, \text{if } e_i \text{ component works} \\ 0, \text{in any other state} \end{cases} . \quad (2)$$

Then the probability of failure-free operation of the entire system P_r (t) can be described by expressions in the case of parallel connection of components:

$$P_r(t) = \prod_{i=1}^{n} x_i * P_i, \quad (3)$$

where P_i is the probability of failure-free operation of an individual element.

In the case of a serial connection, the formula takes the form:

$$P_r(t) = 1 - \prod_{i=1}^{n}(1 - x_i * P_i). \quad (4)$$

2 An example of the algorithm

Let's analyze the Monte Carlo method based on the simplest geometric problem: finding the area of a figure.

Figure 1 shows the figure, the area to be found.

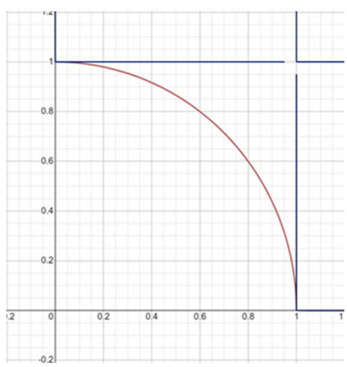

Fig. 1. Figure, the area to be found.

At the next stage, a square with the side "L" is drawn, and random values x1 and y1 ∈ [0;1] start generating. Based on these random numbers, points for plotting X1 = L*x1, Y1 = L*y2 are created and plot appears (Figure 2).

Area of the figure will be calculated using the formula S = L2 * n/N, where n is the number of points that fall into the circle, N is the total number of points.

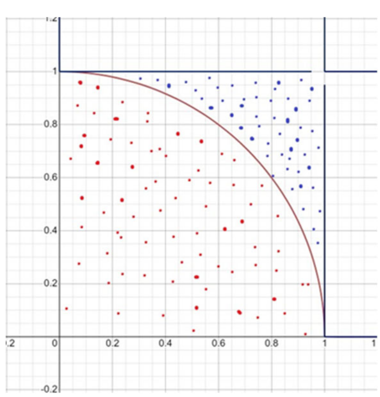

Fig. 2. Complement to the square.

To confirm the operability of this algorithm, it was implemented it in MatLab. Figure 3 shows the algorithm code.

True value: S=(pi*R^2)/4=0.7854.

: Simulation results:

For n = 10; S=[0.7;1].

For n = 100; S = [0.76;0.81].

For n = 1000; S = [0.768;0.8041]

For n = 10000; S = [0.7787;0.789].

For n = 100000; S = [0.7843;0.7865].

As can be noted, with an increase in the number of random numbers, the accuracy of the algorithm increases.

Consider the application of the Monte Carlo method for calculating the reliability of radio-electronic systems (RES).

The general algorithm for calculating the reliability is shown in Figure 4

```
clear;
hold off;
n=0;

N = 100;
L = 1;

for k=1:1:N
    x = rand()*L;
    y = rand()*L;
    if(x*x + y*y <1)
        n=n+1;
    end
end
S = L^2 * (n/N)
```

Fig. 3. Program.

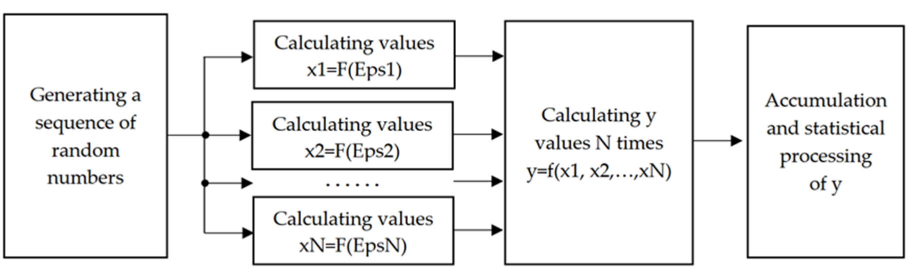

Fig. 4. Calculation algorithm.

After the operation of this algorithm, an array of "y" values is obtained, which is easy to process: finding the mathematical expectation, variance, median, etc.

3 Results

Let's take an example of how the algorithm works.

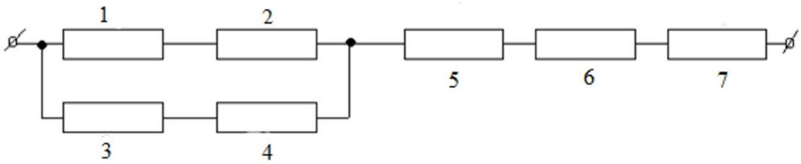

Fig. 5. Circuit example.

Figure 5 shows the structural diagram of RES, as well as the probability of failure-free operation of each element of this system Pn = [0.905; 0,819; 0.887; 0.852; 0.961; 0.942; 0869]. Usually, the probability of failure-free operation for such schemes is calculated by the formula:

$$P_{\Sigma}(t) = [1 - (1 - P_1(t) \cdot P_6(t)) \cdot (1 - P_2(t)P_7(t))] \cdot P_3(t) \cdot P_4(t) \cdot P_5(t) =$$
$$= [1 - (1 - 0{,}852)(1 - 0{,}711)] \cdot 0{,}887 \cdot 0{,}852 \cdot 0{,}961 = 0{,}689.$$

MatLab algorithm processing this scheme is shown in Figure 6.

```
clear;
hold off;
P = [0.905,0,819,0.887,0.852,0.961,0.942,0869];
N = 10000;
Elem = 7;
n = 0;

for k=1:1:N
    vc = [0,0,0,0,0,0,0];
    Eps = [0,0,0,0,0,0,0];
        for(j=1:1:Elem)
            Eps(j) = rand();
        end
        for(c=1:1:Elem)
            if(Eps(c)<P(c))
                vc(c)=1;
            else
                vc(c)=0;
            end
        end
      kof = 0;
      kof =  (vc(1)&&vc(2))||(vc(3)&&vc(4))&&vc(5)&&vc(6)&&vc(7);
      n = n + 1*kof;
end

disp(n/N)
```

Fig. 6. MatLab algorithm.

The result of the work.

For n = 10; P=[0.5;0.9].

For n = 100; P = [0.64;0.75].

For n = 1000; P = [0.65;0.72].

For n = 10000; P = [0.6758;0.7].

For n = 100000; P = [0.6814;0.69].

The dependence of the percentage of deviation of the reference value on the number of cycles passed by the algorithm is shown in Figure 6.

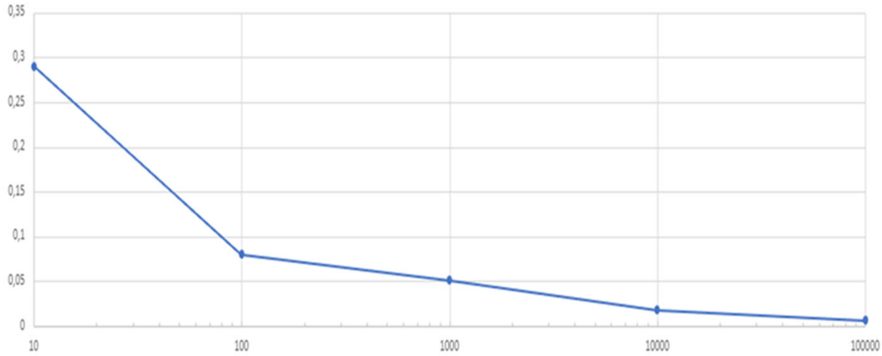

Fig. 7. Graph of the deviation of the true value from the number of cycles.

As can be seen, with an increase in the number of cycles, the accuracy of the algorithm increases. But one problem arises from this: with the first increase in the number of cycles by 10 times (from 10 to 100), the accuracy increased by about 6 times, while with a subsequent increase by 10 times (from 100 to 1000), the accuracy increased by about 2 times. From here arises the problem that was described in [4]. In the "rough" Monte Carlo algorithm, the deviation of the final result is inversely proportional to the square root of the sample size. On average the sample size should be increased by about k2 to reduce the deviation in k. To solve this problem, many improvements of the algorithm have been developed to reduce the sample size. But they were not considered in this article.

Using the Monte Carlo algorithm on such an elementary scheme is not advisable, since coding a program may take longer than calculating such a scheme manually.

4 Conclusion

The Monte Carlo algorithm has a wide range of applicability due to the ability to simulate the process without conducting a large number of real tests, which is of great importance when testing RES, since it is often not possible to conduct an unlimited number of experiments.

The algorithm will reveal its effectiveness in the analysis of complex schemes, when the complexity of the system makes the formulation of accurate models almost impossible.

References

1. F. Ehiagwina, O. Adewunmi, E. Seluwa, O. Kehinde, N. Sidiq, Majlesi Journal of Telecommunication Devices **5**, 129-137 (2016).
2. R. Alzbutas, G. Dundulis, Probabilistic Assessment of Structural Integrity **13(3)**, 154 (2024).
3. H.M. Alshanbari, H. Al-Mofleh, S. Jin-Taek, S.K. Khosa, Symmetry **15(11)**, 2066 (2023).
4. D. Reiter, *The Monte Carlo Method, an Introduction* (Springer, 2007).
5. A. Joseph, *Markov Chain Monte Carlo Methods in Quantum Field Theories: A Modern Primer* (Springer Briefs in Physics, 2020). arXiv:1912.10997v3.
6. R. Alzbutas, G. Dundulis, Probabilistic Assessment of Structural Integrity **13(3)**, 154 (2024).

7. J. Hao, L. Yaohang, *Monte Carlo Methods and Their Applications in Big Data Analysis* (Springer, 2016).
8. Sh. Wang, T. Weber, D. Schramm, Thorben Berns, Future Transp **4(2)**, 380-408 (2024).
9. H. M. Alshanbari, H. Al-Mofleh, S. Jin-Taek, S.K. Khosa, Symmetry **15(11)**, 2066 (2023).
10. J.M. Hammersley, D.C. Handscom, *Monte Carlo methods* (New York, John Wiley & Sons, Inc.R. C. Mikkelson, 1964)
11. E.C. Cipu, R. I. Cipu, Ş.M. Michnea, Mathematics **12(7)**, 1090 (2024).