MATEC Web of Conferences **141**, 01040 (2017) DOI: 10.1051/matecconf/201714101040
*Smart Grids 2017*

# Mathematical simulation of analog-to-digital converter

*Mikhail* Andreev[1,*], *Aleksey* Suvorov[1], and *Aleksey* Khlebov[1]

[1]National Research Tomsk Polytechnic University, 634050 Tomsk, Russia

**Abstract.** Numerical relay protection devices (NRP) are a promising area for the development of protection equipment for electric power systems. For its development, research and, in some cases, setting mathematical models are used. At the same time, the converting part of the NRP, which realizes the analog-to-digital conversion (ADC) of the analog input signal to the digital one, with which the microprocessor of the NRP works directly, is often ignored in models. The purpose of this work is to study the characteristics and parameters of the ADC, as well as to evaluate its impact on the operation of the ADC itself and need to take into account those or other characteristics in the simulation. To develop the ADC model, the C programming language was used. The ADC model is takes into account the cutoff frequency of the analog signal, sampling period, and number of samples required. The characteristics and parameters of the ADC are studied, their effect on the operation of the ADC is estimated. The result of this study – the ADC must be modeled as completely as possible, since its operation depends on the correct functioning of the NRP algorithm.

## 1 Introduction

Relay protection devices play a significant role in ensuring the manageability and reliability of power systems. Until recently, the main share of the relay protection devices utilized was made up of electromechanical and microelectronic devices [1]. Technical means of relay protection morally and physically “grow old”. These circumstances indicate that in this area, modernization of relay protection is needed, based on the use of microprocessor systems and use of hardware and software systems.

Modern relay protection devices use logic circuits. For their implementation it is necessary to use a discrete binary information transfer system. The signals necessary for monitoring the state of the provided system come from the transformers in analog form. In order to convert the analog signal into a binary code, use an ADC (analog-to-digital converter). The goal of this work is to simulate the algorithm of the ADC operation in the MathCAD software package.

* Corresponding author: andreevmv@tpu.ru

## 2 Mathematical model of ADC

The basic principle by which the analog signal is converted into a discrete one consists in sampling the values from an analog signal after a certain period of time and translating these values into binary code. These values repeat the basic form of the analog signal.

The analog signal to the ADC passes through a low-pass filter that passes the frequency spectrum below a certain frequency. The cutoff frequency usually is 50 Hz. The program can specify any number of samples to be performed. In the course of the algorithm, all the numbers are rounded up to larger ones with the help of the function *ceil(x)*, since working with non-integer values when using the binary system is complex and in most cases not advisable. During the operation of the ADC, the simulation program only analyzes the measured value modules. After selecting the initial and associated conditions, we create a matrix of sample values, consisting of twenty-five elements, the number of elements can be very easily changed. The function *num2str(x)* convert numbers to a string. This is necessary to further convert the decimal number to binary. A sampling matrix is obtained consisting of 25 elements of whole decimal numbers written in the form of a string. Next, you need to translate the numbers in the matrix from the decimal to binary, for this a function is created, the form of which is presented in Fig.1.

```
base2base(tal, base_in, base_out) ≡ | NBD ← "0123456789ABCDEF"                                   (1)
                                    | InputNumberLength ← strlen(tal)                          (2)
                                    | NBD_base ← substr(NBD, 0, base_in)                       (3)
                                    | DecimalValue ← 0                                         (4)
                                    | for j ∈ 0..InputNumberLength − 1                         (5)
                                    |   | k ← substr(tal, j, 1)                                (6)
                                    |   | cif ← search(NBD_base, k, 0)                         (7)
                                    |   | error("Found a digit in the number that isn't in this base") if cif = −1   (8)
                                    |   | DecimalValue ← DecimalValue + floor[cif·(base_in)^(InputNumberLength−j−1) + 1/2]   (9)
                                    | Out ← ""                                                 (10)
                                    | while DecimalValue > 0                                   (11)
                                    |   | X ← floor(mod(DecimalValue, base_out))               (12)
                                    |   | Out ← concat(substr(NBD, X, 1), Out)                 (13)
                                    |   | DecimalValue ← floor(DecimalValue / base_out)        (14)
                                    | Out                                                      (15)
```

**Fig. 1**. Operating algorithm of ADC in MathCAD.

Description of the functions of each line in the program is following: (1) – the name of the function and enumeration of its arguments, creating a string that includes all the digits; (2) – counting the number of digits in the entered number; (3) – substring obtained from the NBD string by selecting 10 characters starting at position 0 in the row NBD; (4) – assigns a variable to a value of 0; (5) – beginning of a cycle executed from 0 to a value equal to the number of digits in the entered number -1; (6) – select a specific number in the entered number; (7) – search for the selected digit in NBD and select its serial number; (8) – if there is no digit in the list that is assigned the value -1; (9) – an increment to some variable of an integer, depending on the number of passed stages of the cycle; (10) – assigning an empty string to a variable; (11) – beginning of the loop, while the variable is still greater than zero; (12) – finding the remainder of dividing by 2 and rounding up to a smaller integer; (13) – adding the resulting residue to a string of final value; (14) – decreasing the variable number, then how many times it can be reduced determines the number of values in the response line; (15) – output of the final answer.

## 3 Research results

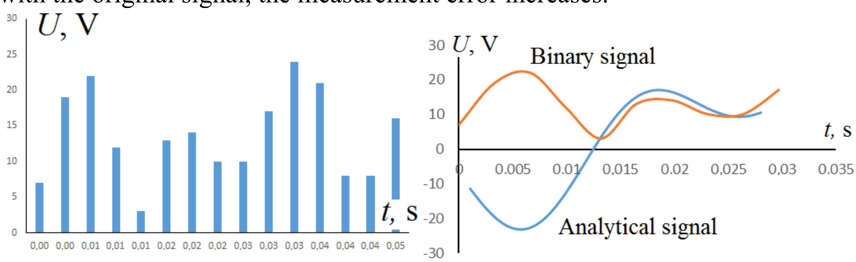
To test the algorithm, the ADC input signal is generated by the formula (1):

$$U(t) = 10\cdot(\sin(314\cdot t - 60) + \sin(157\cdot t + 60) + \sin(78\cdot t + 150)) \tag{1}$$

here $U(t)$ – the dependence of the voltage applied to the analog-to-digital converter on the time; $t$ – time, during which an analog-to-digital converter is supplied with voltage.

In Fig. 2a is a diagram of the obtained sample values in accordance with the ADC algorithm (Fig. 1). Also, for clarity, the original analytical curve with the converted curve of discrete sampling is compatible (Fig. 2b). It can be seen from the graph that the sample almost perfectly repeats the outlines of the original analog signal with a small error in some places, which is caused by rounding of the selected numbers.

Let us analyze the behavior of a rectified discrete curve with increasing or decreasing sampling period within acceptable limits without changing the cutoff frequency of the input signal and its shape. First, consider the change in the discrete signal with a decrease in the sampling period to 0.0025 s. Below is a graph of a combined rectified discrete signal and an analytical signal (Fig. 3a). The Fig. 3b shows a combined graph of voltage changing in time as the sampling period increases to 0.005 s. It can be seen that with a decrease in the sampling period, the output signal is closer to the original analytical form. At the same time, with an increase in the sampling period, the output signal more and more diverges with the original signal, the measurement error increases.




a) b)

**Fig. 2**. a) The diagram of values of the received sample in decimal form; b) a combined voltage changing in time.

## 4 Conclusions

1. In the work done, the operation of the analog-digital converter was investigated. With the use of software complexes, an algorithm for the operation of the ADC has been created. The process of modeling the converter in its general form is described in full.

2. Some binary values are obtained for the received analog signal. The operation of the algorithm is considered for a specific example, but if the input signal is changed, the algorithm will continue to work with observance of the maximum frequency conditions.

3. The effect of sampling period selection is investigated: if the less the period is selected, the more accurate results will be obtained at the output of the ADC.

4. The developed algorithm can be used in the simulation of relay protection [2-8], including power system simulator [9-12]. The implementation of the algorithm will allow

 DOI: 10.1051/matecconf/201714101040

detailed analysis of the operation of the ADC at each of its stages, while varying its parameters.

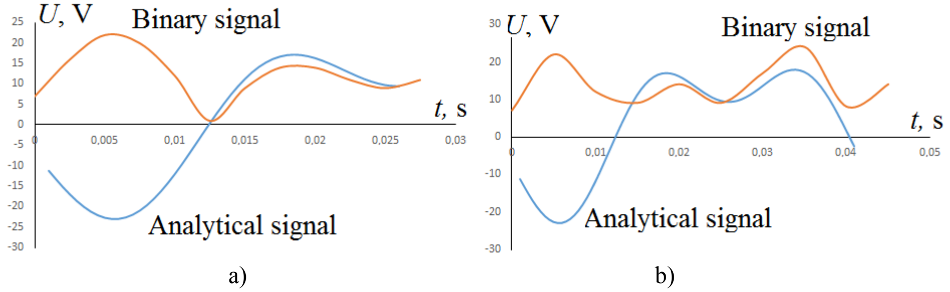



**Fig. 3**. a) Combined graph of voltage changing in time with decreasing sampling period; b) combined graph of voltage changing in time with increasing sampling period.

The reported study was funded by RFBR, according to the research project No. 16-38-60043 mol_a_dk.

## References

1. G. Ziegler, *Numerical Differential Protection* (Erlangen: Publicis Publishing, 2012)
2. A.M. Carter, D.C. Humphreys, A. Dysko, J.R. McDonald, DPSP, 391 (2001)
3. A. Dysko, J.R. McDonal, G.M. Burt, J. Goody, B. Gwyn, DPSP, 381 (1987)
4. A. Dysko, J.R. McDonal, G.M. Burt, J. Goody, B. Gwyn, IEEE Transmission Distribution Conf., 406 (1999)
5. C.-S. Chen, C.-W. Liu, J.-A. Jiang, IEEE Transactions on Power Delivery, **21** (2), 619 (2006)
6. A. Suvorov, Y. Borovikov, A. Gusev, A. Sulaymanov, M. Andreev, N. Ruban, R. Ufa, Electrical Engineering, **99**, 3 (2017), DOI: 10.1007/s00202-016-0464-4
7. N.Yu. Ruban, A.S. Gusev, A.Yu. Pishchulin, MATEC Web Conf. **37** (2015), DOI: 10.1051/matecconf/20153701046
8. N.Y. Ruban, A.O. Sulaymanov, R.A. Ufa, I.A. Razzhivin, MATEC Web Conf. **91** (2016), DOI: 10.1051/matecconf/20179101050
9. N.Y. Ruban, Y.S. Borovikov, A.O. Sulaymanov, A.S. Gusev, IFOST, 311 (2016), DOI: 10.1109/IFOST.2016.7884255
10. R.A. Ufa, A.S. Vasilev, A.S. Gusev, A.A. Suvorov, IFOST, 113 (2016), DOI: 10.1109/IFOST.2016.7884204
11. R.A. Ufa, V.A. Sulaymanova, Y.S. Borovikov, MATEC Web Conf. **37** (2015), DOI: 10.1051/matecconf/20153701061
12. Y.S. Borovikov, A.Y. Pischulin, R.A. Ufa, IOP Conf. Series: Mater. Sci. Eng. **93** (2015), DOI: 10.1088/1757-899X/93/1/012018