On the Problem of Choosing Optimal Methods for Approximating Functions

I A Bordanov¹, S N Zhiganov², S N Danilin¹

¹Department of Information Technology, Murom Institute of Vladimir State University, 23, Orlovskaikaya Street, Murom, 602264, Russia
²Department of Radio Engineering and Computer Systems, Murom Institute of Vladimir State University, 23, Orlovskaikaya Street, Murom, 602264, Russia

E-mail: bordanov2011@yandex.ru

Abstract. The materials of the article relate to the field of optimization of control systems and signal processing when preparing models for technical implementation. The informational level of structural and functional decomposition of models of approximators of square root functions is considered. The article investigates two classes of computational methods: sequential - polynomials of the best approximation and parallel - multilayer feedforward neural networks. For each of the classes, using particular examples, the approximation error was calculated according to the criteria of the maximum absolute error and the area of the error function, as well as the computational costs as the sum of the number of mathematical operations and queries in the memory of the calculator.

1. Introduction

Many researchers have been involved in the development of methods for approximating complex functional dependencies for their practical application. The theoretical foundations of numerical methods and approximation methods were laid by the greatest scientists of their time: I. Newton, L. Euler, N.I. Lobachevsky, K.F. Gauss, P.L. Chebyshev, S. Hermit, M. Savage and others. The mathematical base created by them is still developing, new approaches, methods and algorithms are proposed.

The appearance of the first analog and then digital computing devices that automate the processing of various kinds of information requires researchers to develop applied methods of approximation. One of the first domestic and foreign scientists involved in the introduction of these methods into computing facilities are V.D. Baikov, V.B. Smolov, B. Walder, P. Bezier, R.W. Hamming and others.

An important stage in the development of the theory is associated with the names of Sh.Zh. Balle Poussin, D. Jackson, S.N. Bernshtein, who conducted research on the relationship between the error of approximation of functions by polynomials of various degrees. In the works of the named scientists, the reduced error was about 0.1%, which is insufficient for modern radio engineering systems. This was due to the fact that the algorithms used had to be significantly simplified to a level corresponding to the level of development of computing technology of that period of time.

The emergence of high-performance computing tools for information processing opens up the possibility for scientists to create new applied approximation methods, which are based on more
complex algorithms for information processing, which will increase the accuracy of approximating functions by several orders of magnitude.

1.1. Problem
The problem that has not been solved by the theory of approximation of functions is the development of a general approach and methods for optimal provision of the given technical and economic parameters of approximators at all stages of the product life cycle (design, production, operation) under given restrictions under certain operating conditions [1]. The functional properties and economic parameters achieved at the stage of computer modeling change significantly during the transition to the technical implementation of approximators [2].

1.2. Solution to the problem
World, domestic and author's experience of applying the basic principles, methods and technologies of systems engineering (SE) to all periods of the life cycle of objects and processes gives a large increase in the efficiency of any projects due to their complex optimization [3-5]. The theory and practice of SE showed that in each project it is necessary to optimize functional parameters, technical resources and economic costs [6-9].

The fundamental technique (technique) of SE is decomposition - dividing the original system or its model into parts of ISO/IEC 24765. Decomposition is applied both for processes and for results in order to divide the project into smaller, easily manageable parts for parallel work, their coordination, refinement and control [10].

The authors have developed a general approach (GA) to solving the problem under consideration, based on the SE methodology [11].

On the basis of the GA in this article, a technology for optimizing the main component of the algorithmic support of the information level of the structural and functional decomposition of models of approximators of square root functions is proposed and applied.

A variant of the search for the optimal ratio between the permissible error of the solution of the problem and the computational costs of the algorithms used is investigated.

2. Polynomial approximation
Approximation methods existing at the present time are presented in works [12-15]. When reproducing functional dependencies, the polynomial approximation method has found wide application, which is used in many scientific and applied technical problems: from the approximation of standard mathematical functions in modern specialized microprocessors to the implementation of calibration characteristics when reproducing working standards, calibrating sensors and measuring systems. The ubiquitous distribution of the polynomial method is due to its simplicity, clear geometric interpretation, and most importantly - low computational costs when calculating the values of the function \( f(x) \) using the polynomial.

\[
\varphi(x) = a_0 + a_1 x + \ldots + a_n x^n = \sum_{k=0}^{n} a_k x^k
\]  

(1)

When using Horner’s scheme, expression (1) can be represented as:

\[
\varphi_h(x) = a_0 + a_1 x + \ldots + a_n x^n = [(a_n x + a_{n-1}) x + a_{n-2}] x + \ldots + a_1 + a_0
\]  

(2)

The computational complexity of calculating a polynomial of degree \( n \) is \( 2n \) operations: \( n \) multiplications and \( n \) additions \( (n(\times) + n(+) ) \). In addition, it is necessary to store \( n + 1 \) constants \( a_0, a_1, \ldots, a_n \) in memory and request them to implement the calculation of the polynomial.

There are various methods for finding the coefficients of the polynomial (2). The generalized Chebyshev theorem determines the polynomial of the best approximation of a function when all \( n + 2 \)
extreme values of the errors \( x_i, \ i = 0, 1, ..., n + 1 \) on the interpolation interval \( x \in [a, b] \) alternately change sign and are equal to each other in absolute value [3]. Usually, the accuracy of the approximation of functions using the obtained polynomials is compared with the polynomial of the best approximation.

Another approach to finding the coefficients of the polynomial function (1) is based on minimizing the area of the figure bounded from above by the curve \( \delta(x) = |f(x) - \varphi(x)| \), and from below by the abscissa on the interval of approximation of the function \([a; b]\). When approximating the function \( f(x) \), a polynomial of the form (1) of order \( n \) is used, the found coefficients of which \( a_0, a_1, ..., a_n \) provide the minimum area of the error function \( S_{err} \), determined by the expression

\[
S_{err} = \int_a^b |\delta(x)|\,dx = \int_a^b |f(x) - \varphi(x)|\,dx \rightarrow \min
\]

Such a criterion is called integral or minimizing the error area [16].

Using the method described in [14], the coefficients of the polynomials of the best approximation of the function \( f(x) = \sqrt{x} \) were calculated on the range of values \( x \in [0; 1] \), the results of calculating the main parameters characterizing the quality of the approximation and the requirements for the implementation of the obtained polynomials are shown in table 1.

3. Artificial neural networks
Another promising approach to the approximation of functional dependencies is the use of artificial neural networks (ANN) [17]. In accordance with D. Tsybenko's theorem (universal approximation theorem of 1989), feedforward ANN as a minimum with one hidden layer can approximate any continuous function of many variables with any accuracy [18]. In this article, to solve the approximation problem, we will use an ANN with one hidden layer. The number of inputs \( N_i = 1 \) and the number of output neurons \( N_o = 1 \) correspond to the number of arguments. According to the Kolmogorov-Arnold-Hecht-Nielsen theory, a sufficient number of neurons in the hidden layer to solve this problem \( N_h = 2N_i + 1 = 3 \). Figure 1 shows the structure of the ANN.

![Figure 1](image-url)
To assess the performance, we introduce an additional parameter $f$ - the number of calculations of the activation function [19].

### Table 1. Square root function approximation polynomials.

| №  | Polynomial formulas                      | $\delta_{MM}$ | $A + m$ | $S_{err}$ |
|----|----------------------------------------|---------------|---------|-----------|
| 0  | $\varphi_0(x) = 0.5$                    | 0.500         | 1       | 0.250     |
|    | $a_0 \neq 0$                           |               |         |           |
|    | $a_1 = 1$                              |               |         |           |
|    | $a_0 = 0$                              |               |         |           |
|    | $a_1 \neq 1$                           |               |         |           |
|    | $a_0 \neq 0$                           |               |         |           |
| 1  | $\varphi_1(x) = 0.125 + x$             | 0.125         | 2       | 0.076     |
|    | $a_0 \neq 0$                           |               |         |           |
|    | $a_1 = 1$                              |               |         |           |
|    | $a_0 = 0$                              |               |         |           |
|    | $a_1 \neq 1$                           |               |         |           |
| 2  | $\varphi_2(x) = 0.07 + 1,869x - 0.997x^2$ | 0.070         | 7       | 0.039     |
|    | $a_0 \neq 0$                           |               |         |           |
|    | $a_1 = 1$                              |               |         |           |
|    | $a_0 = 0$                              |               |         |           |
|    | $a_1 \neq 1$                           |               |         |           |
| 3  | $\varphi_3(x) = 3.497x - 5.75x^2 + 3.32x^3$ | 0.074         | 6       | 0.048     |
|    | $a_0 \neq 0$                           |               |         |           |
|    | $a_1 = 1$                              |               |         |           |
|    | $a_0 = 0$                              |               |         |           |
|    | $a_1 \neq 1$                           |               |         |           |
| 4  | $\varphi_4(x) = 4.65x - 14.275x^2 + 19.77x^3 - 9.2x^4$ | 0.056         | 8       | 0.036     |
|    | $a_0 \neq 0$                           |               |         |           |
|    | $a_1 = 1$                              |               |         |           |
|    | $a_0 = 0$                              |               |         |           |
|    | $a_1 \neq 1$                           |               |         |           |
| 5  | $\varphi_5(x) = 5.8x - 28.44x^2 + 68.858x^3 - 73.2x^4 + 28.03x^5$ | 0.045         | 9       | 0.028     |
|    | $a_0 \neq 0$                           |               |         |           |
|    | $a_1 = 1$                              |               |         |           |
|    | $a_0 = 0$                              |               |         |           |
|    | $a_1 \neq 1$                           |               |         |           |
| 6  | $\varphi_6(x) = 6.95x - 49.559x^2 + 182.65x^3 - 327.47x^4 + 12,97,5x^5$ | 0.037         | 11      | 0.024     |
|    | $a_0 \neq 0$                           |               |         |           |
|    | $a_1 = 1$                              |               |         |           |
|    | $a_0 = 0$                              |               |         |           |
|    | $a_1 \neq 1$                           |               |         |           |
| 7  | $\varphi_7(x) = 8.12x - 79.52x^2 + 412.29x^3 - 1095.31x^4 + 1537.25x^5 - 1085.34x^6 + 303.543x^7$ | 0.032         | 12      | 0.02      |
|    | $a_0 \neq 0$                           |               |         |           |
|    | $a_1 = 1$                              |               |         |           |
|    | $a_0 = 0$                              |               |         |           |
|    | $a_1 \neq 1$                           |               |         |           |
| 8  | $\varphi_8(x) = 9.21x - 117.55x^2 + 810.3x^3 - 2955.09x^4 + 6026.384x^5 - 6909.19x^6 + 4159.83x^7 - 1022.91x^8$ | 0.028         | 14      | 0.018     |
|    | $a_0 \neq 0$                           |               |         |           |
|    | $a_1 = 1$                              |               |         |           |
|    | $a_0 = 0$                              |               |         |           |
|    | $a_1 \neq 1$                           |               |         |           |
| 9  | $\varphi_9(x) = 10.23x - 164.07x^2 + 1443.58x^3 - 6870.08x^4 + 18932.24x^5 - 31119.53x^6 + 30076.71x^7 - 15763.41x^8 + 3455.34x^9$ | 0.023         | 15      | 0.015     |

The approximation error for the ANN will be estimated according to the same criteria as for the polynomial, but we will carry out their calculation numerically for 10,000 values of the input argument by the trapezoid method.
During the experiment, we will change the activation functions of the neurons of the hidden layer: hyperbolic tangent (TanSig), logistic function (LogSig), semi-linear function (ReLU), linear with saturation (SatLin). The output neuron has a linear activation function. We will conduct the training in accordance with the Broyden-Fletcher-Goldfarb-Shanno algorithm. Loss function - mean square of errors (MSE), target set to $10^{-8}$. A summary is presented in table 2.

**Table 2.** ANN approximation of the square root function.

| Function of activation of neurons of the hidden layer | $\delta_{M\Phi}$ | A+m+f | $S_{err}$ |
|-------------------------------------------------------|------------------|-------|----------|
| TanSig                                                | 0.021            | 25 (f=3) | 0.00064  |
| LogSig                                                | 0.021            | 25 (f=3) | 0.00067  |
| ReLU                                                  | 0.097            | 25 (f=3) | 0.01414  |
| SatLin                                                | 0.097            | 25 (f=3) | 0.01377  |

As can be seen from table 2, the ANN provides an approximation accuracy no worse than a 9th degree polynomial, while requiring large computational costs. A visual comparison of the absolute error and the square of the error of polynomials and ANNs is shown in figures 2 and 3, respectively.

**Figure 2.** Comparison of the absolute error of function approximation by polynomials and neural networks. The horizontal dashed line marks the best result $\delta_{M\Phi}$ equal to 0.021.
Figure 3. Comparison of the area of the error of approximating a function by polynomials and neural networks. The horizontal dashed line marks the best result $S_{err}$ equal to 0.00064.

4. Conclusions
As a result of the work:

- A brief analysis of the development in time of methods for the approximation of complex functional dependencies is given.
- The main problem is indicated, which is not solved by the theory of approximation of functions - the lack of a general approach and methods for optimal provision of the specified technical and economic parameters of approximators at all stages of the product life cycle (design, production, operation) at given restrictions under certain operating conditions.
- The authors have developed a general approach to solving the problem under consideration, based on the SE methodology.
- A description of two classes of methods for approximating complex functional dependencies is given: polynomials of the best approximation and ANN.
- The described methods are applied to solve the problem of approximating the square root function.
- Accuracy and computational costs are calculated for each method. The results obtained allow us to conclude that the opinion of a number of researchers about the absolute advantages of parallel computational algorithms over sequential ones is premature.
- It is shown that, depending on the criteria, parameters, level and type of decomposition position, both sequential (best approximation polynomials) and parallel (neural network) methods and algorithms can be optimal for solving problems of approximation of functions.
- The results of the work can be useful both in scientific research and in engineering applications.
5. References

[1] Schuman C D, Potok T E, Patton R M, Birdwell J D, Dean M E, Rose G S, and Plank J S 2017 A survey of neuromorphic computing and neural networks in hardware Preprint 1705.06963

[2] Danilin S N, Makarov M V, Shchanikov S A Numerical simulation of neural network components of controlling and measuring systems 2014 Proceedings of 2014 International Conference on Mechanical Engineering, Automation and Control Systems pp 6986873

[3] Kossiakoff A, Sweet W N, Seymour, S J and Biern S M 2011 Systems engineering principles and practice 83 (USA: John Wiley & Sons) p 560

[4] Elm J P and Goldenson D R 2012 The business case for systems engineering study: Results of the systems engineering effectiveness survey (USA: Carnegie Mellon University Software Engineering Institute) p 173

[5] Danilin S N, Shchanikov S A 2015 Neural network control over operation accuracy of memristor-based hardware Proceedings of 2015 International Conference on Mechanical Engineering, Automation and Control Systems pp 1-5

[6] Nikolenko V Yu 2018 Basic course of system engineering: a tutorial (Moscow: MIPT) p 330

[7] Shtanko A N and Kulik S D 2021 Preliminary Experiment on Emotion Detection in Illustrations Using Convolutional Neural Network Advances in Intelligent Systems and Computing 1310 pp 490-494

[8] Kulik S D and Levin E O 2021 Specialized Software Tool for Pattern Recognition of Biological Objects Advances in Intelligent Systems and Computing 1310 pp 173–180

[9] 2010 ISO/IEC/IEEE 24765:2010 Systems and software engineering – Vocabulary IEEE Standards Association 24765 p 418

[10] Shchanikov S A, Bordanov I A, Zuev A D, Danilin S N, Korolev D, Belov A and Mikhaylov A 2021 Fault Tolerance of Memristor-Based Perceptron Network for Neural Interface BioNanoScience 11(1) pp 84–90

[11] Shchanikov S, Zuev A, Bordanov I, Danilin S, Lukoyanov V, Korolev D, Belov A, Pigareva Y, Gladkov A, Pimashkin A, Mikhaylov A, Kazantsev V and Serb A 2021 Designing a bidirectional, adaptive neural interface incorporating machine learning capabilities and memristor-enhanced hardware Chaos, Solitons and Fractals 142 pp 110504

[12] Bakhvalov N S, Zhidkov N P and Kobelkov G M 2000 Numerical methods (Moscow: Laboratory of Basic Knowledge) p 624

[13] Verzhbitsky V M 2001 Numerical methods (mathematical analysis and ordinary differential equations): Textbook manual for universities (Moscow: Higher school) p 382

[14] Chekushkin V V and Zhiganov S N 2018 Computational methods in optimization of engineering problems (USA: Open Science Publishing) p 202

[15] Chekushkin V V, Zhiganov S N, Mikheev K V and Bykov A A 2017 Density matrix theory of coherent ultrafast dynamics Bulletin of the Concern VKO "Almaz - An-tei" (1) pp 98–104

[16] Zhiganov S N and Mikheev K V 2018 Search for Chebyshev polynomials that minimize the area of errors Methods and devices for transmitting and processing information: Scientific and technical journal 20 ed V V Romashova and V V Bulkin (Murom: MI VIGU) pp 100–102

[17] Danilin S N, Makarov M V, Shchanikov S A The development of a neuronetwork component for technical systems of mechanical engineering Proceedings of 2014 International Conference on Mechanical Engineering, Automation and Control Systems

[18] Cybenko G V 1989 Approximation by Superpositions of a Sigmoidal function Mathematics of Control Signals and Systems 2(4) pp 303–314

[19] Kulik S D and Tkachenko 2009 K I Evaluation of the efficiency of technical systems using neural networks Neurocomputers: development, application (9) pp 47–60

Acknowledgments

The work was supported by the RFBR grant No. 19-07-01215.