Volume 87, Number 5,
May 2026

ISSN 0005-1179
CODEN: AURCAT

AUTOMATION AND REMOTE CONTROL

http://ait.mtas.ru

Automation and Remote Control

Vol. 87, No. 5, May 2026

Automation and Remote Control

ISSN 0005-1179

SCOPE

Automation and Remote Control is one of the first journals on control theory. The scope of the journal is control theory problems and applications. The journal publishes reviews, original articles, and short communications (deterministic, stochastic, adaptive, and robust formulations) and its applications (computer control, components and instruments, process control, social and economy control, etc.).

Automation and Remote Control is abstracted and/or indexed in *ACM Digital Library, BFI List, CLOCKSS, CNKI, CNPIEC Current Contents/Engineering, Computing and Technology, DBLP, Dimensions, EBSCO Academic Search, EBSCO Advanced Placement Source, EBSCO Applied Science & Technology Source, EBSCO Computer Science Index, EBSCO Computers & Applied Sciences Complete, EBSCO Discovery Service, EBSCO Engineering Source, EBSCO STM Source, EI Compendex, Google Scholar, INSPEC, Japanese Science and Technology Agency (JST), Journal Citation Reports/Science Edition, Mathematical Reviews, Naver, OCLC WorldCat Discovery Service, Portico, ProQuest Advanced Technologies & Aerospace Database, ProQuest-ExLibris Primo, ProQuest-ExLibris Summon, SCImago, SCOPUS, Science Citation Index, Science Citation Index Expanded (Sci-Search), TD Net Discovery Service, UGC-CARE List (India), WTI Frankfurt eG, zbMATH.*

Journal website: http://ait.mtas.ru

0005-1179/26. *Automation and Remote Control* (ISSN: 0005-1179 print version, ISSN: 1608-3032 electronic version) is published monthly by Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, 65 Profsoyuznaya street, Moscow 117997, Russia.

Volume 87 (12 issues) is published in 2026.

Publisher: Trapeznikov Institute of Control Sciences, Russian Academy of Sciences.

65 Profsoyuznaya street, Moscow 117997, Russia; e-mail: redacsia@ipu.rssi.ru; http://ait.mtas.ru, http://ait-arc.ru

Contents

Automation and Remote Control

Vol. 87, No. 5, 2026

Special Issue[1]

Models for Managing the Elimination of Consequences of Critical Situations at Oil Refining and Chemical Enterprises

A. F. Rezchikov, O. I. Dranko, O. V. Kushnikov, A. S. Bogomolov, A. A. Dnekeshev, and I. A. Stepanovskaya 387

Using Machine Learning Methods for Analyzing and Forecasting of Small Samples of Macroeconomic Indicators in the Energy Sector of the Russian Federation

V. A. Ivanyuk 404

Consideration of Soft Dependencies under Stochastic Uncertainty in Multi-Project Program Implementation

I. V. Burkova and A. V. Shchepkin 418

Controlled Random Search and Likelihood Ratio in Boolean Programming Problems

A. I. Ermolaev, A. V. Akhmetzyanov, and A. R. Latipov 428

Reviews

Biology and Phenomenology of Temporal Encoding: A Review of Studies and Models

L. Yu. Zhilyakova 437

Robust, Adaptive, and Network Control

Design of Self-Checking Discrete Devices Based on Boolean Signal Correction with Weighted Sum Codes in the Residue Ring of a Given Modulus

D. V. Efanov and Y. I. Yelina 465

[1] Articles published in this heading, are the termination of special issue (see Autom. Remote Control, 2026, no. 4).

ISSN 0005-1179 (print), ISSN 1608-3032 (online), Automation and Remote Control, 2026, Vol. 87, No. 5, pp. 387–403.

SPECIAL ISSUE

Models for Managing the Elimination of Consequences of Critical Situations at Oil Refining and Chemical Enterprises

A. F. Rezchikov[*,a], O. I. Dranko[*,b], O. V. Kushnikov[,c], A. S. Bogomolov[**,***,d], A. A. Dnekeshev[**,e], and I. A. Stepanovskaya[*,f]**

[*] *Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia*
[**] *Saratov Scientific Center, Russian Academy of Sciences, Saratov, Russia*
[***] *Saratov University named after N.G. Chernyshevsky, Saratov, Russia*
e-mail: [a] rw4cy@mail.ru, [b] olegdranko@gmail.com, [c] kushnikoff@iptmuran.ru, [d] bogomolov@ya.ru, [e] dnekeshev1991@gmail.com, [f] irstepan@ipu.ru

Received July 8, 2025
Revised November 16, 2025
Accepted November 20, 2025

Abstract—The article is devoted to the development and application of system-dynamic models for managing the process of liquidation of emergency situations at oil-refining and chemical enterprises. A complex of heterogeneous system-dynamic models has been developed, allowing enterprise management according to a criterion that minimizes the deviation of relevant indicators of safe functioning from the values recommended by the decision-maker. A formulation of the problem of managing the process of liquidation of emergency situations is presented, and a comprehensive methodology is proposed, including the construction of cause-and-effect graphs, regression analysis of functional dependencies, numerical solution of a system of nonlinear differential equations, as well as a procedure for correcting the system-dynamic model. The complex of models makes it possible to take into account key safety indicators, external factors, and nonlinear effects, ensuring high accuracy of forecasting and risk analysis. The obtained results can be used in the development of control systems for the process of liquidation of consequences of emergency situations at oil-refining and chemical enterprises of the country, as well as in training systems for facility-level units of the Ministry of Emergency Situations.

Keywords: oil-refining enterprises, chemical enterprises, system dynamics, control systems, chemically hazardous substances, critical situations, nonlinear differential equations, training systems

DOI: 10.7868/S1608303226050015

1. INTRODUCTION

At present, in connection with the development of new technologies and the complication of production processes, the issues of increasing the safety of the functioning of oil-refining enterprises are becoming one of the priority tasks. Accidents at such enterprises can lead to serious consequences for human health, the environment, and the economy, which makes the issues of increasing the level of industrial safety of oil refineries highly relevant (see Fig. 1).

Industrial safety at oil-refining enterprises is the most important aspect determining the sustainability of production, the prevention of accidents, and the reduction of the economic and environmental consequences of possible incidents. In recent decades, various researchers have studied methods for assessing and managing risks at such enterprises, applying both traditional statistical methods and modern approaches, including machine learning and intelligent data analysis. The

Fig. 1. State of the Achinsk oil refinery after the fire [1].

analysis of the existing literature makes it possible to identify several main directions of research in the field of the safety of oil-refining enterprises, including quantitative risk assessment, modeling of emergency situations, forecasting of equipment failures, and analysis of historical accident data.

One of the effective methods of risk assessment is Fault Tree Analysis (FTA), which is used to analyze possible scenarios of equipment failures and their consequences. In [2], the application of the Fuzzy Fault Tree Analysis (FFTA) method for assessing the risks of fires and explosions in oil storage tanks is considered. The authors note that traditional methods of quantitative risk analysis assume the availability of accurate data on the probability of failure of various elements of the system; however, in practice, such data may be incomplete or inaccurate due to insufficient statistical information. In this work, a combined methodology is proposed that combines traditional fault tree analysis with fuzzy set theory, which makes it possible to take uncertainties into account when assessing the probability of accidents.

Historical analysis of accidents also plays an important role in the development of methods for preventing incidents. In [3], a detailed analysis of 44 accidents at the oil refinery in Skikda (Algeria) over the period from 2002 to 2013 was carried out. The authors considered various incidents, including fires, explosions, and leaks of toxic substances, and identified the key factors contributing to the occurrence of accidents. On the basis of the data analysis, it was established that a significant part of the accidents was caused by equipment failures resulting from outdated infrastructure and insufficient maintenance.

Another important aspect of ensuring safety at petrochemical enterprises is the analysis of fire and explosion risks. In [4], methods for assessing and modeling risks associated with fires and explosions at petrochemical facilities are considered. The study uses a comprehensive approach, including the HAZID methodology for hazard identification, Dow's Fire & Explosion Index (Dow's F&EI) for quantitative risk assessment, as well as software-based modeling of emergency situations using the Process Hazard Analysis Software Tool (PHAST).

Forecasting equipment failures is another important area of research in the field of industrial safety. In [5], a methodology for predicting pipeline failures based on machine learning methods is proposed. The authors use three models—a multilayer perceptron neural network (MLP), a radial basis function neural network (RBF), and multinomial logistic regression (MNL)—for the analysis of historical data on pipeline damage.

Modern technologies for processing textual information are also finding application in the field of industrial safety. In [6], a system for automated risk analysis of oil refineries based on natural language processing (NLP) and the Bidirectional Encoder Representations from Transformers (BERT) model is proposed.

Equipment reliability management and maintenance planning also play an important role in ensuring industrial safety. In [7], a Risk-Based Inspection and Maintenance (RBI&M) methodology is presented, including six stages: identification of the scope of application, functional analysis, risk assessment, quantitative expression of risk, planning of measures, and implementation of works.

Special attention is paid to the study of the domino effect—the escalation of emergency situations due to a primary incident. In [8], a detailed analysis of methods for assessing the domino effect at industrial facilities over the past 30 years was carried out.

Thus, modern studies in the field of industrial safety demonstrate a wide range of methods for risk assessment and management, from classical probabilistic models to intelligent data analysis. The integrated application of these approaches makes it possible to significantly increase the safety of oil-refining enterprises by reducing the probability of emergency situations and mitigating their consequences.

Many researchers and engineers are engaged in developing innovative solutions to reduce risks and increase the reliability of production processes. The developed systems and methods, including control and diagnostic means, undergo thorough testing and demonstrate high efficiency in practice.

Nevertheless, despite many years of research and the achieved results, the existing models and methods do not always make it possible to take into account the complex system of interrelations between internal and external factors affecting the safety of processes, which may negatively affect their reliability. This circumstance makes the application of the mathematical apparatus of system dynamics expedient for increasing the safety of the functioning of oil-refining enterprises.

System dynamics is a method of computer modeling and analysis of complex systems that makes it possible to study their behavior over time under the influence of various factors. This approach is based on the use of differential equations and feedback loops to describe the interrelations between the elements of the system.

Historically, system dynamics was developed by J. Forrester and his colleagues in the middle of the twentieth century for the analysis of industrial and social systems [9].

At present, it is applied in various fields, including production, emergency management, healthcare, analysis of natural disasters, and many others. In production, it is used for process optimization, increasing the efficiency of equipment operation, and resource management, which makes it possible to minimize costs and predict possible failures.

In the field of emergency management, system dynamics helps to model the development of events, assess the consequences of technogenic disasters, and develop effective response plans, which significantly increases the level of safety. In healthcare, it finds application in diagnostics, treatment planning, and resource management, as well as in combating epidemics, where models predict the spread of infections and assess the effectiveness of preventive measures.

In addition, system dynamics is actively used for the analysis and management of the consequences of natural disasters, such as floods, making it possible to develop strategies for minimizing damage and restoring infrastructure. This approach is indispensable in situations where it is necessary to take into account many interrelated factors and predict the development of complex systems.

As shown in [10], this approach makes it possible to take into account nonlinear interactions, feedback loops, and time delays that are characteristic of industrial systems. The authors carried out an analysis of 63 studies in which system dynamics was applied to assess external factors, organizational impacts, and internal causes of accidents.

In the case of the oil-refining industry, a classical example is the analysis of the Bhopal accident, where system dynamics was used to construct models describing the influence of the human factor, technical failures, and organizational decisions on the safety of the functioning of the Union Carbide

chemical enterprise [11]. The work considered cause-and-effect relationships between key variables affecting the safety of functioning, such as equipment failure, operator errors, and insufficient maintenance.

In [12], the main shortcomings of traditional methods are emphasized, including their limitations in modeling nonlinear processes and time delays. The authors analyze the application of system dynamics in the context of risk analysis for the oil-refining industry. They also highlight the need to integrate this methodology into the decision-making process, which makes it possible to better predict emergency situations and increase resilience to critical situations in the process of functioning of an oil-refining enterprise. This study presents a model of the actions of the population in the event of an accident at a chemically hazardous facility, taking into account the level of awareness of the enterprise personnel.

In [13], a system-dynamic model used in the system for managing the process of informing the population under emergency situations at oil-refining enterprises is presented. The authors developed a stock-and-flow model that analyses the influence of the frequency of message distribution and the quality of information on people's behavior during chemical accidents.

This article is devoted to the development of new system-dynamic models intended for use in managing the process of liquidation of emergency situations at oil-refining enterprises. These models make it possible to comprehensively assess possible risk factors that may lead to disruption of the normal course of technological processes, to take timely necessary measures to prevent emergency situations, and to reduce the damage from their occurrence.

2. STATEMENT OF THE PROBLEM

Develop mathematical models and methods for managing the process of liquidation of emergency situations at an oil-refining enterprise according to an efficiency criterion that makes it possible, over the time interval $t \in [t_0; t_N]$, to determine control actions in the form of an action plan $p(t) \in P$ and to minimize, under admissible values of environmental disturbances $R(t) \in R$, the objective function

$$Z(p(t)) = \int_{t_0}^{t_N} \sum_{i=1}^{n} \left(K_i^* - K_i(t, R(t), p(t))\right)^2 \gamma_i \, dt \to \min \tag{1}$$

subject to the constraints:

$$\frac{dK_i(t, R(t), p(t))}{dt} = f_i\left(t, K_1(t), \ldots, K_n(t), R(t), p(t)\right), \quad i = \overline{1,n}, \tag{2}$$

$$K_i(t_0) = K_{i0}, \quad i = \overline{1,n},$$

$$t > 0, \quad K_i > 0, \quad i = \overline{1,n},$$

$$K_i^{\min} \leqslant K_i(t, p(t)) \leqslant K_i^{\max}, \quad i = \overline{1,n} \tag{3}$$

and boundary conditions:

$$F_i^{t_0}(K, K', p) = 0, \quad F_j^{t_N}(K, K', p) = 0, \quad i = \overline{1,k_1}, \quad j = \overline{1,k_2}.$$

Here, $K_i(t, p(t))$, $i = \overline{1,n}$, and K_i^* are the relevant indicators of the efficiency of liquidation of a critical situation and their values recommended by the decision-maker, respectively; γ_i is the weight coefficient of the ith indicator; $K_i^{\min}$ and $K_i^{\max}$ are the minimum and maximum values of the efficiency indicator of liquidation of the emergency situation.

Develop mathematical models and algorithms for forecasting, over the time interval $[t_0; t_N]$, changes in the relevant indicators of the efficiency of liquidation of the emergency situation $K_i(t, p(t))$, $i = \overline{1,n}$, for the purpose of determining such time instants $t_j \in [t_0; t_N]$, $j = \overline{1,m}$, when they go beyond the values K_i^* recommended by the decision-maker.

3. MATHEMATICAL MODEL

To solve problems (1)–(3), it is expedient to use the system-dynamic approach and the method of system dynamics [5, 9, 12, 13, 16–19]. Within the framework of this approach, a complex of interconnected mathematical models has been developed, the construction of which involves the implementation of the following stages:

1. Selection of the input variables of the model affecting the safety of the functioning of oil-refining and chemical enterprises in accordance with [14, 15].

2. Development of a cause-and-effect graph between the system variables and external influences.

3. Construction of a system of system-dynamics equations in the general form, the solution of which will make it possible to forecast the values of the variables related to the safety of the functioning of an oil-refining enterprise over various time intervals.

4. Determination of functional dependencies between the model variables using the apparatus of regression analysis.

5. Solution of the system of nonlinear differential equations by a numerical method.

6. If necessary, correction of the mathematical model in order to achieve the required accuracy of calculations.

Selection of Input Variables. The assessment of the safety of the functioning of oil-refining and chemical enterprises is determined in accordance with the requirements of the Federal Law "On the Protection of Population and Territories from Natural and Man-Made Emergencies" [14] and GOST R 22.1.10–2002 "Safety in Emergencies. Monitoring of Chemically Hazardous Objects" [15] (see Fig. 2).

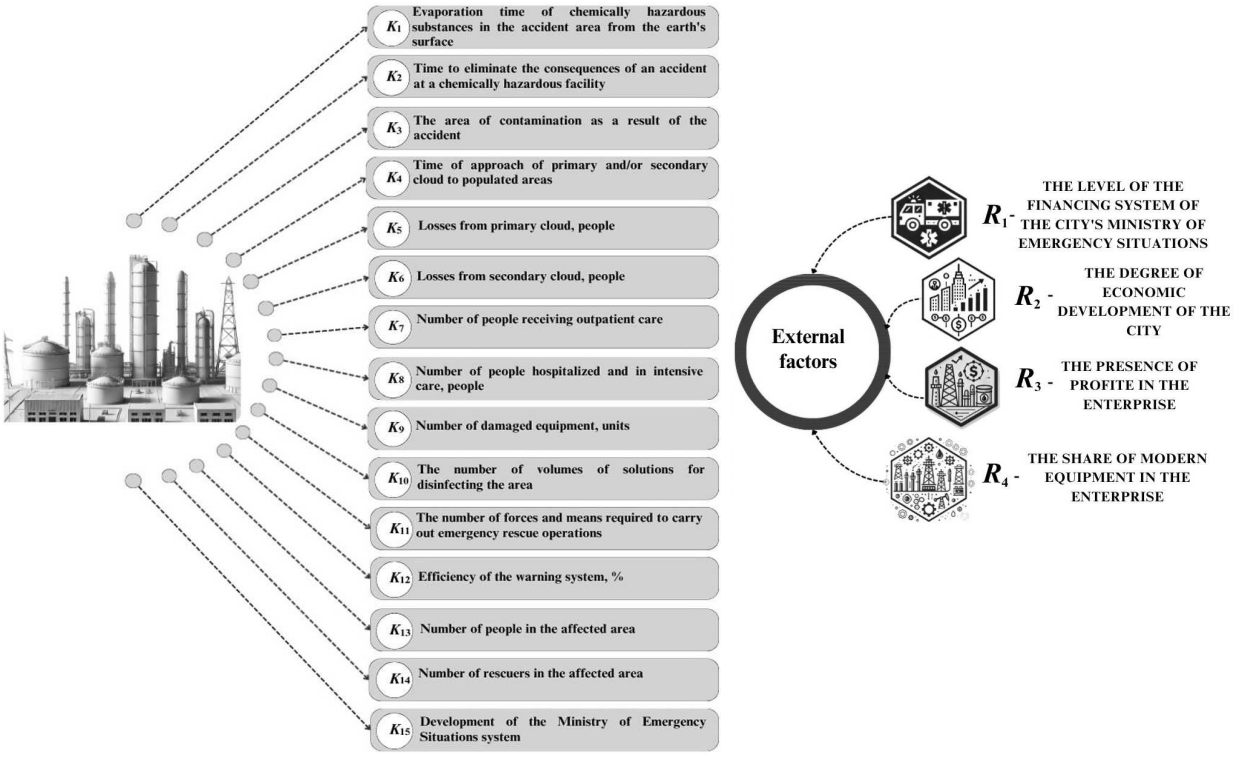


Fig. 2. Input variables of the model.

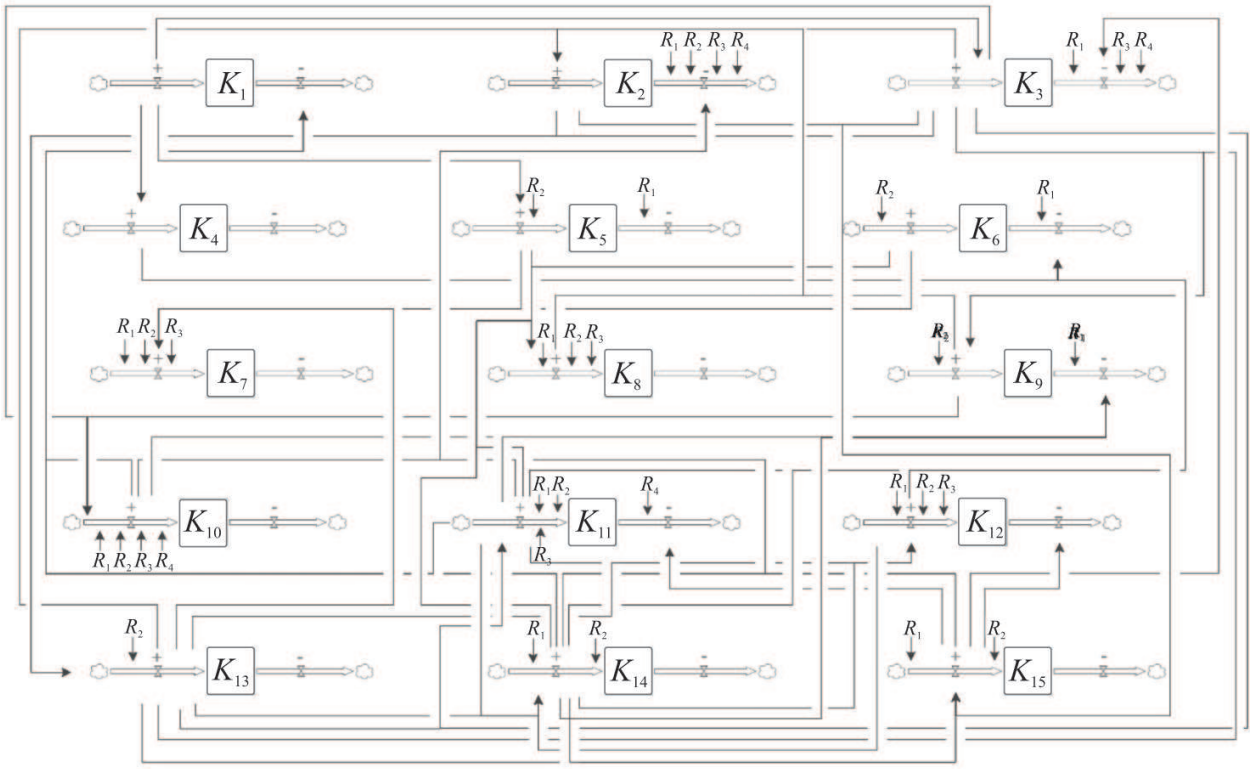


Fig. 3. Cause-and-effect graph.

Cause-and-Effect Graph between System Variables and External Influences. The cause-and-effect graph was constructed according to the methodology developed in [16, 21, 25] (see Fig. 3).

Construction of the System of System-Dynamics Equations in the General Form. Below is a part of the system of differential equations (4), describing the change of the variables $K_i(t, p(t))$, $i = \overline{1,n}$, over time, taking into account the positive and negative relationships existing between them and the influence of external factors:

$$\begin{cases} \dfrac{dK_1(t)}{dt} = -f_1(K_{10}(t))f_2(K_{11}(t))f_3(K_{14}(t)), \\ \dfrac{dK_2(t)}{dt} = f_4(K_3(t))f_5(K_7(t))f_6(K_8(t))f_7(K_9(t))f_8(K_{13}(t)) \\ \qquad\qquad - f_9(K_{10}(t))f_{10}(K_{11}(t)), \\ \qquad\qquad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ \dfrac{dK_{14}(t)}{dt} = f_{49}(K_{11}(t))f_{50}(K_{12}(t))f_{51}(K_{13}(t)) \times R_1(t) + R_2(t), \\ \dfrac{dK_{15}(t)}{dt} = f_{52}(K_2(t))f_{53}(K_3(t))f_{54}(K_{13}(t))f_{55}(K_{14}(t)) \times R_1(t) + R_2(t) \end{cases} \tag{4}$$

Determination of Functional Dependencies between the Model Variables. To solve system (4), it is necessary to establish the dependencies between the model variables $f_1, f_2, \dots, f_{55}$ using regression analysis, expert assessments, or the corresponding regulatory and reference documentation. When the developed models are used as part of training systems, these dependencies may also be specified at the stage of describing the scenario of the emerging critical situation.

In the article, the dependencies are approximated mainly by linear functions or second-degree polynomials possessing sufficient flexibility for modeling.

In the article, the dependencies are approximated mainly by linear functions or second-degree polynomials possessing sufficient flexibility for modeling nonlinear effects that are often present in complex systems. In addition, such polynomials can be well interpreted from the point of view of physical meaning, which makes them a convenient tool for analysis and presentation of results.

Solution of the System of Nonlinear Differential Equations. System (5), obtained after substituting into (4) the dependencies $f_1, f_2, \ldots, f_{55}$ determined at the previous stage of model development, is solved.

$$
\begin{cases}
\dfrac{dK_1(t)}{dt} = -(0.14K_{10}^2 - 0.86K_{10} + 1.54)(0.6K_{11}^5 - 1.64K_{11} + 1.87) \\
\qquad \times (0.03K_{14}^2 - 0.03K_{14} + 0.92), \\
\dfrac{dK_2(t)}{dt} = (-0.53K_3^2 + 0.34K_3 + 0.75)(-5.75K_7^2 + 3.98K_7 + 2.36) \\
\qquad \times (-7.87K_8^2 + 8K_{14} + 0.45)(-3.17K_9^2 + 5.01K_9 - 1.2) \\
\qquad \times (-14.03K_{13}^2 + 25.61K_{13} - 10.89)(-4.77K_{10}^2 - 9.69K_{10} - 5.5) \\
\qquad \times (6.47K_{11}^2 - 12.68K_{11} + 6.81)(-0.41K_{14}^2 + 0.47K_{14} + 0.64) \\
\qquad \times (13.28K_{15}^2 - 27.1K_{15} + 14.41)(0.1R_1(t) + 2) \\
\qquad \times (0.2R_2(t) + 2.5)(2\sin R_3(t) + \varphi)(4R_4(t) + 4), \\
\dfrac{dK_3(t)}{dt} = (18.06K_1^2 - 35.01K_1 + 17.45)(22.81K_{15}^2 - 39.91K_{15} + 17.94) \\
\qquad \times (0.1R_1(t) + 2) + (2\sin R_3(t) + \varphi) + (4R_4(t) + 4), \\
\ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \\
\dfrac{dK_{13}(t)}{dt} = (-1.34K_2^2 + 2.13K_2 + 0.07)(-0.02K_3^2 - 0.06K_3 + 0.94) \\
\qquad \times (0.2R_2(t) + 2.5), \\
\dfrac{dK_{14}(t)}{dt} = (-12.24K_{11}^2 + 21.08K_{11} - 8.38)(-3.5K_{12}^2 + 5.21K_{12} - 1.23) \\
\qquad \times (3.21K_{13}^2 - 5.21K_{13} + 2.67)(0.1R_1(t) + 2)(0.2R_2(t) + 2.5), \\
\dfrac{dK_{15}(t)}{dt} = (0.67K_2^2 - 1.42K_2 + 1.6)(0.41K_3^2 - 0.4K_3 + 1) \\
\qquad \times (5.63K_{13}^2 - 10.37K_{13} + 5.67) \\
\qquad \times (0.03K_{14}^2 - 0.05K_{14} + 0.93)(0.1R_1(t) + 2)(0.2R_2(t) + 2.5).
\end{cases}
\tag{5}
$$

The initial conditions used to solve the system of nonlinear differential equations are given in Table 1.

Table 1. Initial conditions

K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}	K_{11}	K_{12}	K_{13}	K_{14}	K_{15}
1	1	0.43	0.91	1	1	0.94	0.94	0.65	0.71	0.73	0.48	0.85	0.53	0.84

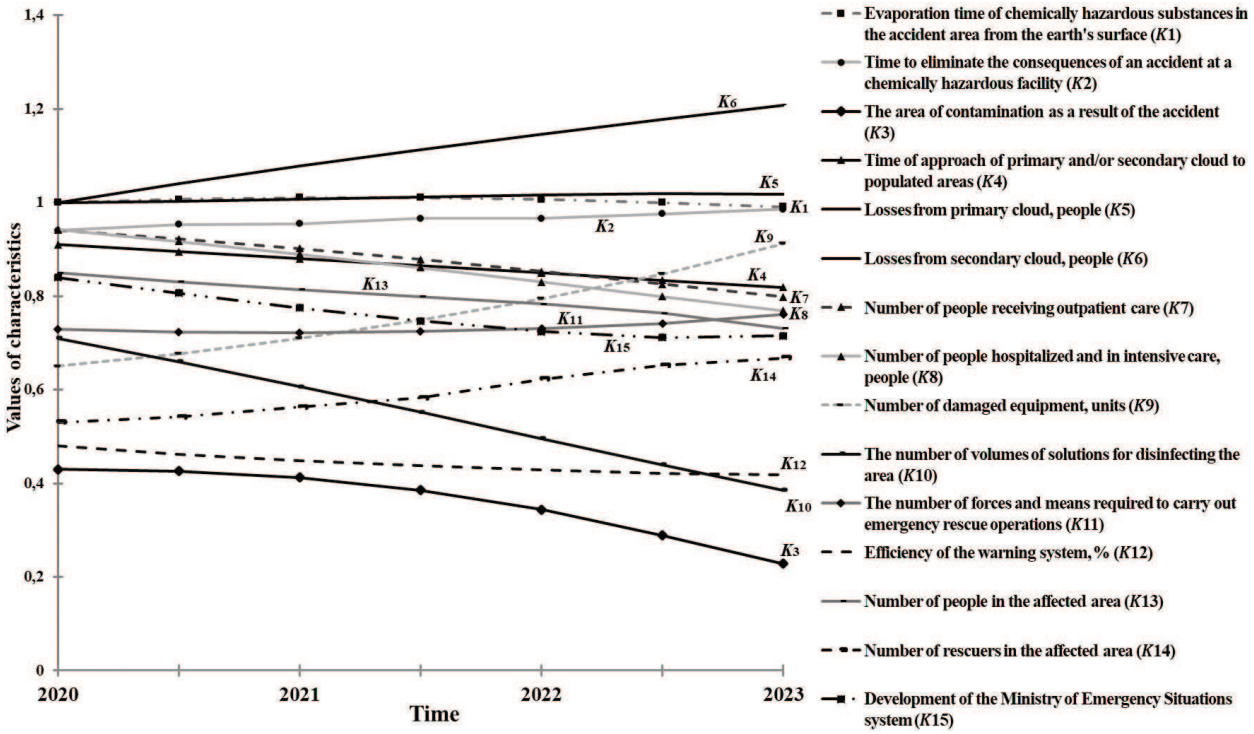


Fig. 4. Change in the normalised model variables over the machine time interval $[0; 0.6]$ of the training system.

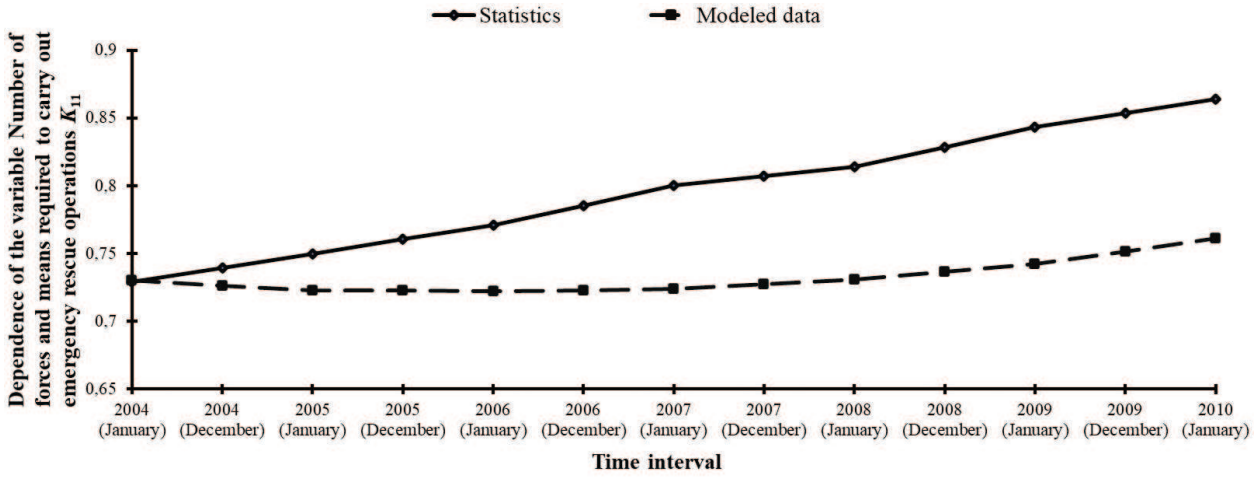


Fig. 5. Comparison of statistical and calculated data for the characteristic $K_{11}(t)$.

Figure 4 shows the graph of the normalized model variables $K_i(t, p(t))$, $i = \overline{1, n}$, obtained as a result of solving system (5).

To assess the model error, Fig. 5 presents a comparison of calculated data characterizing the change in the normalized variable $K_{11}(t)$ over the machine time interval $[0; 1.2]$ with statistical data obtained from the analysis of materials on the investigation of the explosion at the Texas City oil refinery [26].

From Fig. 5 it follows that the maximum deviation of $K_{11}(t)$ from the statistical data is 11.98%.

4. CORRECTION OF THE SYSTEM-DYNAMICS MODEL

The necessity of periodic correction of the system-dynamics model is explained by the following considerations.

Modern oil-refining and chemical enterprises are complex large-scale systems of high dimensionality, the functioning processes of which depend on many hundreds and thousands of parameters, many of which are qualitative in nature.

The development of a complete mathematical model characterizing the efficiency of the functioning of an oil refinery or a chemical enterprise is hardly possible due to the high dimensionality of the problem being solved. Moreover, such a model will have low accuracy due to the systematic accumulation of errors in determining all its numerous input variables, and the modeling error will only increase over time.

From the above, it follows that almost any mathematical model of a complex system becomes less accurate over time without periodic correction.

Below, an algorithm for correcting a system-dynamics mathematical model is presented, used in solving the problem of managing the liquidation of consequences of critical situations at oil-refining and chemical enterprises.

The algorithm makes it possible to maintain the error of the mathematical model at a specified level so that it does not affect the reliability of conclusions and recommendations formed for the decision-maker. The algorithm is based on the assumption that the accuracy of forecasting the behavior of a complex large-scale system can often be increased by reducing the forecasting interval.

Verification of the validity of this assumption should be carried out at the stage of adapting the developed mathematical support to the specific features of functioning within the control system of a particular enterprise.

At this stage, it is also necessary to determine the periodicity with which it is economically feasible and organizationally and technically possible to carry out correction of the mathematical model.

If the duration of the selected forecasting interval does not allow the required accuracy of mathematical modeling to be ensured, then the developed correction algorithm cannot be used at this oil refinery or chemical enterprise.

The presence of significant inertia in many complex large-scale systems makes it possible to conclude that the algorithm for correcting the system-dynamics model may be effective for many enterprises.

The experience of implementing the procedure for correcting system-dynamics models of large-scale systems allows one to conclude that for many oil refineries and chemical enterprises the modeling error can be maintained at a level not exceeding 10% when the forecasting interval is reduced to 12–14 months of real time, which is acceptable when solving problems (1)–(3) as part of a training complex.

Algorithm.

Step 1. Select an action plan $p_i(t) \in \{P(t)\}$ implemented during the liquidation of consequences of critical situations at oil-refining and chemical enterprises.

Step 2. Select relevant disturbances that must be taken into account when solving problems (1)–(3).

Step 3. Determine functions approximating the change of the selected disturbances over a unit time interval of machine time.

Step 4. Construct a system-dynamics model (1) in the general form.

Step 5. Determine the internal functions of the model by approximating them with low-degree polynomials $f_1, f_2, \ldots, f_{55}$.

Step 6. Present the mathematical model in a form suitable for numerical calculations by substituting $f_1, f_2, \ldots, f_{55}$ into the mathematical model (4).

Step 7. Based on statistical information, construct dependencies $K_i(t, p(t))^{\text{Stat}}$, $i = \overline{1,15}$, characterizing the change of the characteristics $K_i(t, p(t))$, $i = \overline{1,15}$ over the time interval used in modeling within the training system.

Step 8. Solve system (5) and determine the calculated values of the characteristics $K_i(t, p(t))$, $i = \overline{1,15}$.

Step 9. Set the value of the counter of characteristics $K_i(t, p(t))$, $i = \overline{1,15}$ equal to one, i.e. $i = 1$.

Step 10. Compare the calculated value of the characteristic $K_i(t, p(t))$ with the value $K_i(t, p(t))^{\text{Stat}}$ obtained from statistical analysis.

Step 11. If the condition

$$\forall t \in [t_0; t_N] \quad \left| K_i(t, p(t))^{\text{Stat}} - K_i(t, p(t)) \right| \leqslant 0.1\, K_i(t, p(t))$$

is satisfied, i.e. if the modulus of the difference between the compared characteristics exceeds 10%, initiate the correction procedure by proceeding to the next step. Otherwise, proceed to Step 15.

Step 12. Determine the value of the correction coefficient

$$K^{\text{cor}}(K_i(t, p(t))) = \frac{K_i(t, p(t))^{\text{Stat}}}{K_i(t, p(t))}$$

for the characteristic $K_i(t, p(t))$ as the value by which the calculated value of the characteristic $K_i(t, p(t))$ must be multiplied in order to obtain the value of this characteristic derived from statistical analysis $K_i(t, p(t))^{\text{Stat}}$.

Step 13. Modify the cause-and-effect graph G_{PSS}, the degrees, or the coefficients of the polynomials $f_1, f_2, \ldots, f_{55}$ in such a way that the condition

$$\forall t \in [t_0; t_N] \quad \left| K_i(t, p(t))^{\text{Stat}} - K_i(t, p(t)) \right| < 0.1\, K_i(t, p(t))$$

is satisfied.

Step 14. Generate a message for the decision-maker on the completion of the correction operation of the main characteristic $K_i(t, p(t))$ and record in the database the value of the correction coefficient $K^{\text{cor}}(K_i(t, p(t)))$.

Step 15. Increase the value of the counter of characteristics by one, i.e. assign $i = i + 1$.

Step 16. If the condition $i \leqslant 15$ holds, proceed to Step 9.

Step 17. Generate a message for the decision-maker on the completion of the correction operation of all characteristics $K_i(t, p(t))$, $i = \overline{1,15}$, and record in the database the values of the correction coefficients of these characteristics $K^{\text{cor}}(K_i(t, p(t)))$, $i = \overline{1,15}$.

Step 18. Replace the mathematical model used before correction with a model having modified cause-and-effect relationships and coefficients of the polynomials $f_1, f_2, \ldots, f_{55}$.

End of the algorithm.

An example of correcting the model variable K_{12} over the machine time interval $[0; 1, 3]$ of the training system is shown in Fig. 6.

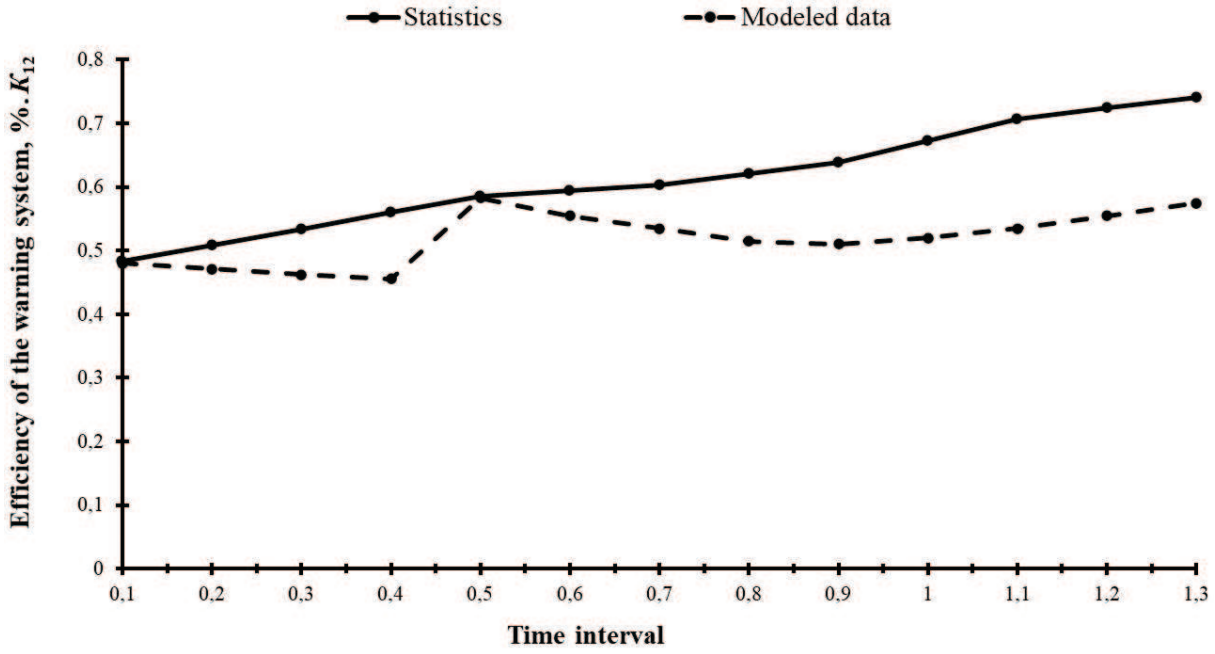


Fig. 6. Correction of the model variable K_{12} over the machine time interval $[0; 1.3]$ of the training system.

In accordance with the developed algorithm for correcting the mathematical model, correction is performed if the discrepancy between the modeling results and the statistical data exceeds 10%. Based on the data in Fig. 6, it can be concluded that over the machine time interval $[0; 1, 3]$ the model correction was performed only once at the time instant $t = 0.5$.

5. MODEL EXAMPLE

Let us consider an example of using the developed models as part of a training system intended for acquiring and improving skills in managing the processes of liquidation of consequences of critical situations at oil-refining and chemical enterprises.

Problem Statement. Suppose that accidents have occurred at an oil refinery in a large city and at a nearby chemical plant as a result of an external impact. In the training system of the facility-level unit of the Ministry of Emergency Situations, it is required to verify the feasibility of three plans for localization and liquidation of the resulting emergency situation (LLES) and to select the one that will minimize the objective function (1) subject to constraints (2) and (3).

Description of the Accident and Its Consequences. On March 23, 2005, a cloud of hydrocarbon vapors ignited and exploded at the isomerization unit of an oil refinery in Texas City (see Fig. 7). As a result, 15 workers were killed, 180 were injured, and the plant suffered severe damage amounting to \$322 million in 2024 prices. Taking into account compensation of \$2.1 billion, repair costs, production downtime, and fines, this explosion became the most costly accident at an oil refinery in the world.

Solution of the Problem. When performing model calculations, the system of differential equations (5), the initial conditions given in Table 1, and the dependencies between the model variables $f_1, f_2, \ldots, f_{55}$ determined based on the results of investigations of the largest oil refinery accident in Texas City, USA, as well as a number of accidents at domestic and foreign oil-refining and chemical enterprises [2, 4, 8, 26], were used.

Fig. 7. Consequences of the explosion at the oil refinery in Texas City.

In the model example for plans p_1–p_3, a number of model dependencies $f_1, f_2, \ldots, f_{55}$ are approximated by second-degree polynomials:

$$\begin{cases} f_{21}(K_5) = -0.144K_5^2 + 0.108K_5 + 0.98, \\ f_{22}(K_6) = -0.158K_6^2 + 0.128K_6 + 0.97, \\ f_{23}(K_{13}) = -1.85K_{13}^2 - 3.39K_{13} + 2.52, \\ f_{24}(K_{15}) = -1.54K_{15}^2 + 3.19K_{15} - 0.65. \end{cases} \tag{6}$$

To assess the approximation accuracy, the dependencies $f_7(K_8)$ and $f_{12}(K_{15})$ obtained by calculation are compared with statistical data.

In Fig. 8 and Fig. 9, the red line represents the approximation curve obtained on the basis of the polynomial model, while the black line shows real statistical data. The comparison results indicate a good agreement between the considered curves.

Substituting $f_1, f_2, \ldots, f_{55}$ into the system of equations (5), we solve this system numerically under the given initial conditions (Table 1). Substituting the obtained results into (1) in the form of dependencies $K_i(t, p(t))$, $i = \overline{1, 15}$, we calculate the value of the objective function $Z(p_1(t))$ corresponding to plan $p_1(t)$. The values of the objective function $Z(p_2(t))$ and $Z(p_3(t))$, corresponding to plans $p_2(t)$ and $p_3(t)$, are determined in a similar way (Table 2).

Table 2. Results of solving the problem.

Plans for liquidation of consequences of critical situations	$p_1(t)$	$p_2(t)$	$p_3(t)$
Value of the objective function	1.356	1.678	1.935

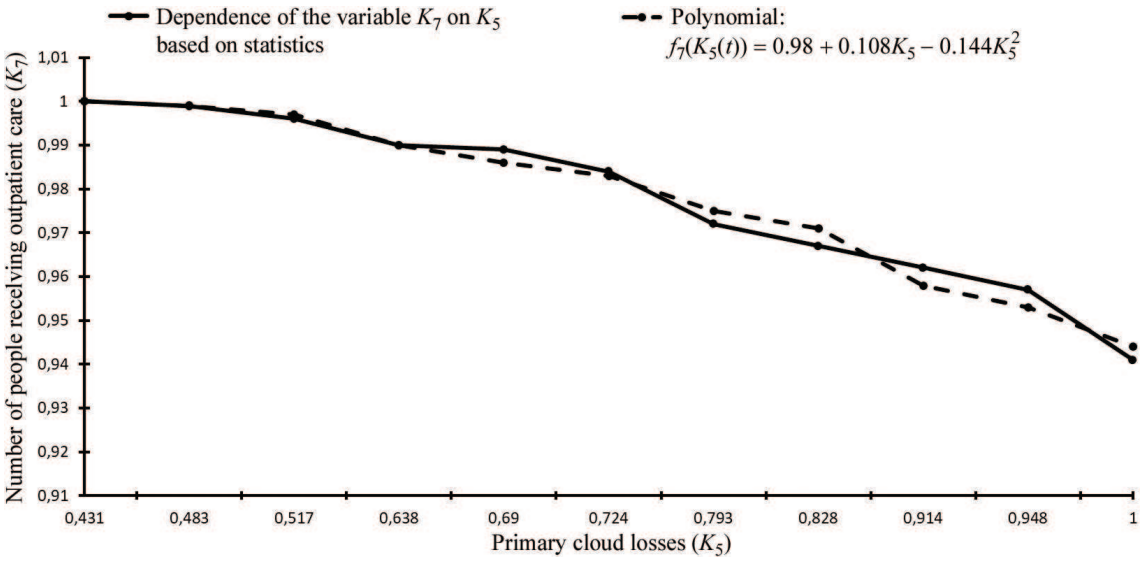


Fig. 8. Dependence of the number of people who received outpatient care (variable K_7) on losses caused by the primary cloud K_5.

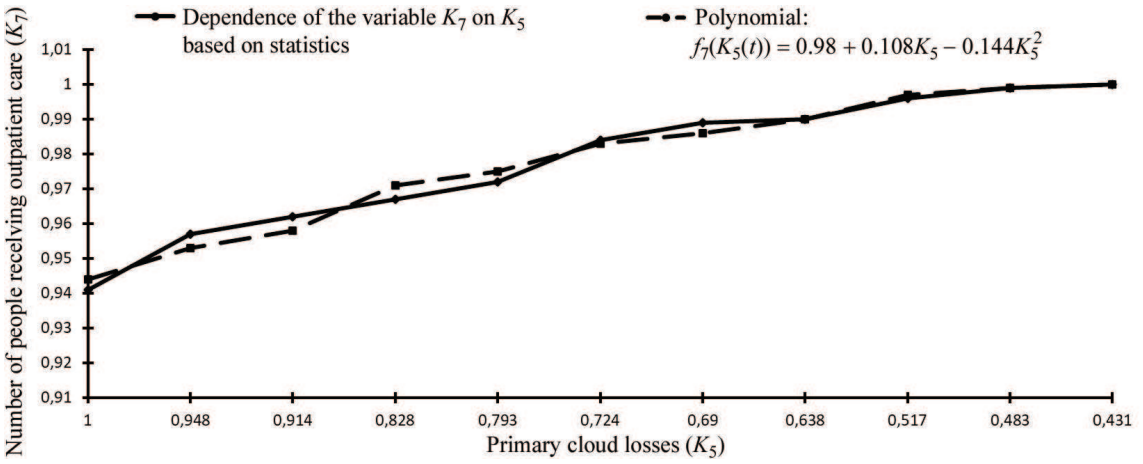


Fig. 9. Dependence of the number of people who received outpatient care (indicator K_7) on the development level of the emergency service system K_{15}.

The calculations performed show that the solution of problems (1)–(3) is the plan $p_1(t)$.

6. INFORMATION–LOGICAL SCHEME FOR SOLVING THE PROBLEM

The procedure for solving problems (1)–(3) over time intervals [minimum possible; month], [quarter; year] is presented in Fig. 10 in the form of an information–logical scheme (ILS). The scheme characterizes the main stages of minimizing damage during the liquidation of consequences of critical situations at oil-refining and chemical enterprises, the production processes of which are associated with the production, storage, and processing of toxic and potentially hazardous substances.

The following designations are used in Fig. 10: 1—technological equipment for preparing raw materials at the oil refinery; 2—equipment for primary processing of raw materials at the oil refinery; 3—equipment for secondary processing of raw materials at the oil refinery; 4—equipment for hydrotreating at the oil refinery; 5—mixing of components of finished products; 6—local automation devices; 7—recording information in the database (DB) and knowledge base (KB); 8—expert

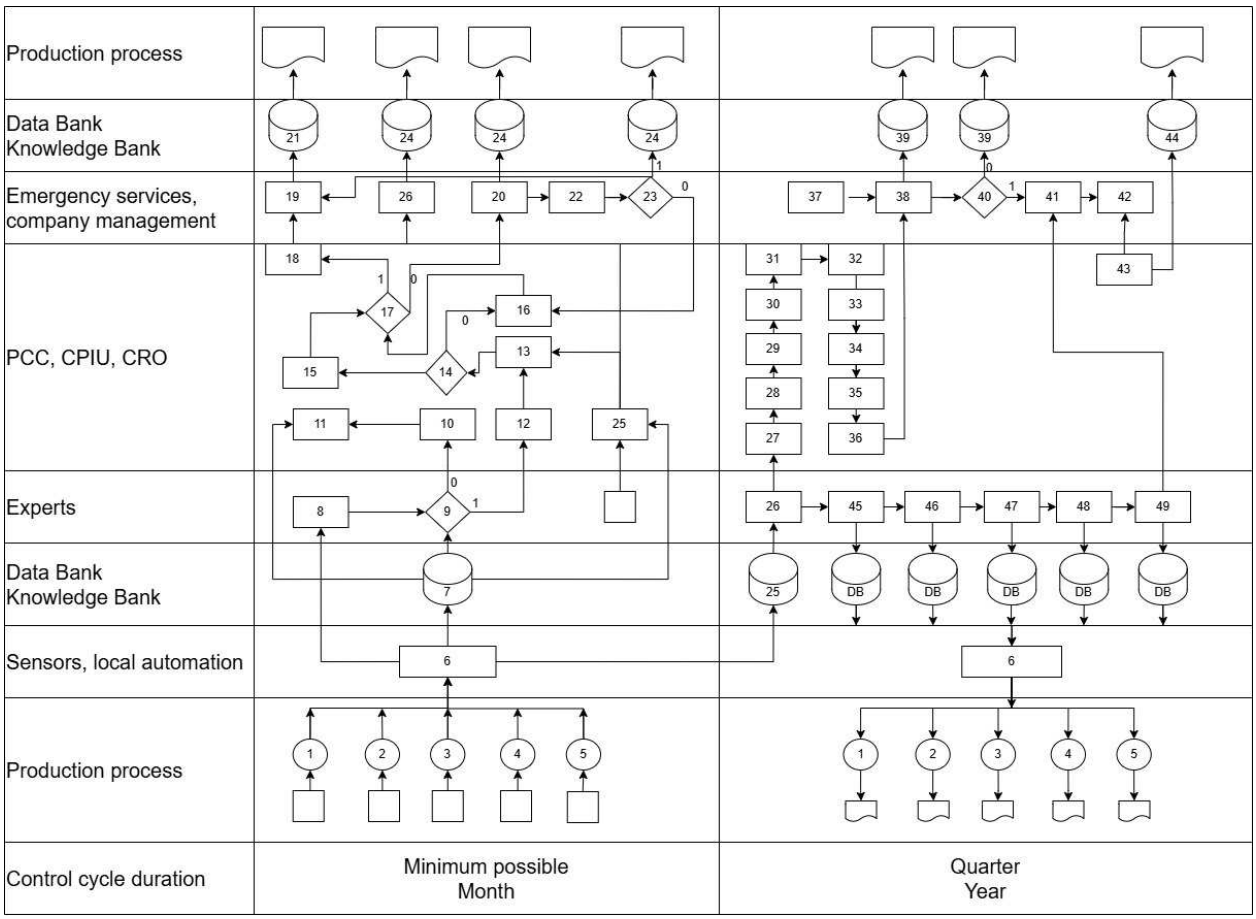


Fig. 10. Information–logical scheme for solving the problem in the system of the facility-level unit of the Ministry of Emergency Situations deployed at the enterprise [23–25].

assessment of the hazard of the emerging situation; 9—emergency situation; 10—recording information on the normal production situation in the shift dispatchers log on an electronic medium; 11—determination of measures to eliminate the normal situation; 12—collection of information on the emergency situation; 13—identification of the emergency situation; 14—for the emergency situation, plans for localization and liquidation (LLES) have been developed; 15—verification of the plans for feasibility; 16—preparation of the LLES plan; 17—is the plan feasible?; 18—notification of the feasibility of plan p_i; 19—management of plan execution p_i; 20—informing the management decision-maker about the impossibility of executing plan p_i; 21—recording in the database the results of plan execution p_i; 22—correction of plan p_i; 23—is the corrected plan p_i feasible?; 24—recording in the database information on the implemented plan p_i; 25—selection from the enterprise database of information on the release of toxic substances as a result of the accident; 26—expert assessment of the severity of the emergency situation [12, 13]; 27—determination of the mass of the toxic substance released into the atmosphere in the primary and secondary cloud; 28—determination of the concentration of the toxic substance in the affected area; 29—determination of the rise height of the toxic substance; 30—determination of the dispersion zone of the toxic substance; 31—determination of the duration of the atmospheric release; 32—determination of damage from increased morbidity of the population; 33—determination of damage to agriculture; 34—determination of damage from changes in the natural environment; 35—determination of damage due to deterioration in the quality of life; 36—determination of damage to the enterprise; 37—assessment of personnel performance efficiency; 38—informing the enterprise management and the

emergency service of the facility-level unit about the calculated values of damage and personnel performance efficiency; 39—recording in the enterprise database information on the magnitude of damage caused by the atmospheric release of a toxic substance; 40—is correction of the LLES required?; 41—modification of the LLES; 42—approval of new LLES by the enterprise management; 43—training of personnel in actions for localization and liquidation of accidents using the training system; 45–49—recommendations for reducing losses specified in items 32–36 of the ILS; LLES—plan for localization and liquidation of emergency situations.

It follows from Fig. 10 that the problem of managing the liquidation of consequences of critical situations at oil-refining and chemical enterprises is solved over two time intervals of different duration: [minimum possible; month], [quarter; year].

During the solution of the problem at the first time interval, the degree of hazard of the emerging critical situation that led to atmospheric releases of pollutants is determined. The maximum size of the contamination zone by toxic or potentially hazardous substances is determined, and measures are taken to limit it and evacuate personnel. The feasibility of existing plans for localization and liquidation of emergency situations is assessed, and, if necessary, they are corrected. Information about the incident is communicated to the enterprise management and the emergency service of the facility-level unit and is also entered into the enterprise database. If necessary, the information is transmitted to the unified state system for prevention and liquidation of emergency situations.

At the second time interval, the damage caused by atmospheric emissions of pollutants is assessed; the corresponding information is entered into the database of the information system, transmitted to the enterprise management, and, in agreement with it, communicated to all interested external organizations.

7. CONCLUSION

The developed complex of system-dynamics models constitutes a tool for analysis, forecasting, and control under emergency situations, intended for use as part of information-control and training systems of oil-refining and chemical enterprises. The mathematical model, taking into account a significant number of relevant feedback relationships between the system variables, as well as external disturbing factors, makes it possible to comprehensively assess the safety of technological processes and to minimize damage during the liquidation of consequences of critical situations at oil-refining and chemical enterprises.

The use of the apparatus of regression analysis for forming functional dependencies between the internal variables of the model makes it possible to take nonlinear effects into account and to increase the accuracy of calculations. Second-degree polynomials used for data approximation made it possible to obtain the required level of calculation accuracy, while the numerical solution of the system of nonlinear differential equations provided reliable modeling results. Real statistical data served as the basis for the calculations, and their use made it possible to confirm the correspondence between the results of model calculations and statistical information.

The development of an original algorithm for correcting the system-dynamics model and for operational decision-making when variables deviate from permissible values allowed the required modeling error of approximately 10% to be ensured.

A comparison of modeling results with actual data obtained at various industrial enterprises [3] confirmed the reliability and the required level of accuracy of mathematical modeling. The results of the conducted research may serve as a basis for the further improvement of safety-management systems and risk minimization at enterprises of the oil-refining industry.

REFERENCES

1. Federal Service for Environmental, Technological, and Nuclear Supervision. Accident at the Achinsk Oil Refinery [Electronic resource], 2014. Available at: `https://www.gosnadzor.ru/industrial/oil/lessons/2014%20%d0%b3%d0%be%d0%b4`. Accessed March 13, 2025. (In Russian.)
2. Wang, D., Zhang, P., and Chen, L., Fuzzy fault tree analysis for fire and explosion of crude oil tanks, *J. Loss Prev. Process Ind.*, 2013, vol. 26, no. 6, pp. 1390–1398. https://doi.org/10.1016/j.jlp.2013.08.022
3. Chettouh, S., Hamzi, R., and Benaroua, K., Examination of fire and related accidents in Skikda Oil Refinery for the period 2002–2013, *J. Loss Prev. Process Ind.*, 2016, vol. 41, pp. 186–193. https://doi.org/10.1016/j.jlp.2016.03.014
4. Saloua, B., Mounira, R., and Salah, M., Fire and Explosion Risks in Petrochemical Plant: Assessment, Modeling and Consequences Analysis, *J. Fail. Anal. Prev.*, 2019, vol. 19, pp. 903–916. https://doi.org/10.1007/s11668-019-00732-6
5. Elshaboury, N., Al-Sakkaf, A., Alfalah, G., and Abdelkader, E., Data-Driven Models for Forecasting Failure Modes in Oil and Gas Pipes, *Processes*, 2022, vol. 10, Art. 400. https://doi.org/10.3390/pr10020400
6. Macêdo, J.B., Moura, M.C., Aichele, D., and Lins, I.D., Identification of risk features using text mining and BERT-based models: Application to an oil refinery, *Process Saf. Environ. Prot.*, 2022, vol. 158, pp. 382–399. https://doi.org/10.1016/j.psep.2021.12.025
7. Bertolini, M., Bevilacqua, M., Ciarapica, F.E., and Giacchetta, G., Development of risk-based inspection and maintenance procedures for an oil refinery, *J. Loss Prev. Process Ind.*, 2009, vol. 22, no. 2, pp. 244–253. https://doi.org/10.1016/j.jlp.2009.01.003
8. Necci, A., Cozzani, V., Spadoni, G., and Khan, F., Assessment of domino effect: State of the art and research needs, *Reliab. Eng. Syst. Saf.*, 2015, vol. 143, pp. 3–18. https://doi.org/10.1016/j.ress.2015.05.017
9. Forrester, J., *Mirovaya dinamika: per. s angl.* (World Dynamics, transl. from English), Moscow: OOO "Izd-vo AST", 2003. (In Russian.)
10. Shirea, M.I., Jun, G.T., and Robinson, S., The Application of System Dynamics Modelling to System Safety Improvement: Present Use and Future Potential, *Safety Science*, 2018, vol. 106, pp. 104–120, https://doi.org/10.1016/j.ssci.2018.03.010
11. Di Nardo, M., Madonna, M., Murino, T., and Castagna, F., Modelling a Safety Management System Using System Dynamics at the Bhopal Incident, *Appl. Sci.*, 2020, vol. 10, no. 3. https://doi.org/10.3390/app10030903
12. Durnev, R.A., Kotosonova, A.S., and Galiullina, R.L., System-dynamic model of informing the population in an accident at a chemically hazardous object, *Issues of Risk Analysis*, 2015, vol. 12, no. 2, pp. 44–47. (In Russian.)
13. Durnev, R.A., Kotosonova, A.S., and Galiullina, R.L., Results of system-dynamic modeling of the process of informing the population in a chemical accident, *Issues of Risk Analysis*, 2016, vol. 13, no. 1, pp. 46–51. (In Russian.)
14. Federal Law "On the Protection of Population and Territories from Natural and Man-Made Emergencies," no. 68-FZ, December 21, 1994 (latest edition) [Electronic resource], 1994. Available at: `https://www.consultant.ru/document/cons_doc_LAW_5295/`. Accessed March 13, 2025. (In Russian.)
15. State Standard of the Russian Federation. Safety in Emergencies. Monitoring of Chemically Hazardous Objects. General Requirements, July 1, 2003 [Electronic resource], 2003. Available at: `https://docs.cntd.ru/document/1200030864`. Accessed March 13, 2025. (In Russian.)
16. Dnekeshev, A., Kushnikov, V., and Tsvirkun, A., System-Dynamic Model for Analysis and Forecasting Emergency Situations of Oil Refinery Enterprises, *Proc. 17th Int. Conf. on Management of Large-Scale System Development (MLSD)*, Moscow, Russian Federation, 2024, pp. 1–4.
17. Tsvirkun, A.D., Rezchikov, A.F., Dranko, I.O., et al., Optimization and simulation approach to determining critical combinations of company parameters, *Autom. Remote Control*, 2024, vol. 85, no. 10, pp. 972–980.

18. Tsvirkun, A.D., Bogomolov, A.S., Dranko, O.I., et al., System dynamics models for controlling the road transport system of a megacity, *Autom. Remote Control*, 2024, vol. 85, no. 10, pp. 981–994.

19. Tsvirkun, A.D., Rezchikov, A.F., Kushnikov, V.A., et al., Models and Methods for Checking the Attainability of Goals and Feasibility of Plans in Large-Scale Systems Using the Example of Goals and Plans for Elimination of the Consequences of Flood, *Autom. Remote Control*, 2023, vol. 84, no. 12, pp. 1437–1448.

20. Kusheleva, E., Rezchikov, A., Kushnikov, V., et al., Mathematical model for prediction of the main characteristics of emissions of chemically hazardous substances into the atmosphere, *Stud. Syst. Decis. Control*, 2019, vol. 199, pp. 594–607.

21. Kushnikova, E., Kulakova, E., Alipchenko, S., et al., Models and Methods for Determining Damage from Atmospheric Emissions of Industrial Enterprises, *Stud. Syst. Decis. Control*, 2019, vol. 199, pp. 256–267.

22. Zhmud, V. and Dimitrov, L., Using the Fractional Differential Equation for the Control of Objects with Delay, *Symmetry*, 2022, vol. 14, no. 4, Art. 635.

23. Kushnikova, E., Ivaschenko, V., and Dvoryashina, M., Coordination of the Functional Structure of the Decision Support and Management System, *Proc. 14th Int. Conf. on Management of Large-Scale System Development (MLSD)*, Moscow, Russian Federation, 2021, pp. 1–5.

24. Rezchikov, A., Kushnikova, E., Kushnikov, O., and Baryshnikova, E., Analysis of the Goals Achievement Degree and Plans Implementation in Large-Scale Systems, *Proc. 16th Int. Conf. on Management of Large-Scale System Development (MLSD)*, Moscow, Russian Federation, 2023, pp. 1–4.

25. Fominykh, D., Kushnikov, V., and Rezchikov, A., Control of the Welding Process in Robotic Technological Complexes Using the System Dynamics Model, *Proc. Int. Multi-Conf. on Industrial Engineering and Modern Technologies (FarEastCon)*, Vladivostok, Russia, 2019, pp. 1–6.

26. Ramos, B., López Droguett, E., Mosleh, A., et al., Revisiting Past Refinery Accidents from a Human Reliability Analysis Perspective: The BP Texas City and the Chevron Richmond Accidents, *Can. J. Chem. Eng.*, 2017, vol. 95, no. 12, pp. 2293–2305. https://doi.org/10.1002/cjce.22996

This paper was recommended for publication by A.A. Galyaev, a member of the Editorial Board

ISSN 0005-1179 (print), ISSN 1608-3032 (online), Automation and Remote Control, 2026, Vol. 87, No. 5, pp. 404–417.

SPECIAL ISSUE

Using Machine Learning Methods for Analyzing and Forecasting of Small Samples of Macroeconomic Indicators in the Energy Sector of the Russian Federation

V. A. Ivanyuk

Financial University under the Government of the Russian Federation, Moscow, Russia
e-mail: VAIvanyuk@fa.ru

Received July 8, 2025
Revised October 14, 2025
Accepted November 20, 2025

Abstract—Forecasting methodologies are widely applied in the analysis of socio-economic systems. Employing robust forecasting techniques enables organizations to anticipate future developments, optimize resource allocation, and mitigate potential risks. In the energy sector, accurate forecasting of supply and demand is essential for maintaining grid stability, reducing operational costs, and enhancing reliability. This study aims to assess the effectiveness of statistical and neural network modeling methods in forecasting macroeconomic indicators within the energy sector of the Russian Federation.

Keywords: machine learning methods, ensemble forecast, energy balance, statistics

DOI: 10.7868/S1608303226050025

1. INTRODUCTION

Amid significant transformations in the energy sector and increasing emphasis on sustainable development, examining the challenges of forecasting the energy balance is essential. The energy balance represents the interdependence between production capacity and consumption within the energy system. Effective analysis of energy balance fluctuations requires consideration of a comprehensive range of economic, environmental, and technical factors.

The energy balance encompasses more than energy production and consumption. It also requires consideration of both renewable and non-renewable resource utilization, seasonal fluctuations in demand, and the influence of industrial and other economic sectors. These factors are essential for a comprehensive analysis.

The relevance of the chosen topic is dictated by the need to overcome the general economic problem of electricity shortage.

The article has the following structure. Section 2 describes the collection and processing of data for making a forecast. Section 3 contains an analysis of forecasting methods. Section 4 presents the results of the ensemble forecast. The final results are presented in Section 5.

2. DATA COLLECTION AND PROCESSING FOR FORECASTING

In order to obtain up-to-date data, annual macro-economic reports for the period 2005–2020 from the Unified Interdepartmental Information and Statistical System of the Russian Federation (UIISS) were studied [1, 2]. The data summary sheets had different formats, encoding, and indicator

names. Hence, manual transposition was performed, the data were reduced to a common format, and the names of the indicators were unified.

For the 2021–2022 data collection, annual summary statistical reports on the energy balance of the Russian Federation were obtained from the official website of the Ministry of Energy [3]. Since the data in these reports were based on the year-to-year relative change (percentage change) in indicators, a consistent conversion of relative values to natural indicators (in million tons of reference fuel) was carried out. At the next stage, an in-depth analysis of data sources was performed, which proved the absence of freely available energy balance data for the Russian Federation in the period 2023–2024. After receiving confirmation that the highest possible completeness of the macroeconomic data is reached, the 2005–2020 and 2021–2022 data were combined and reduced to a single format. As a result, a table of indicators has been formed, which contains data sorted by date, type of energy, and consumption sector.

An array of data consisting of 29 blocks on 8 time series (sampled values) was obtained, where each series had a length of 18 successive values. In this case, two values in each block are aggregators, so that the total number of input data in the table can be defined as 4032 values. The fragments of the input data are presented in Table 1.

Table 1. Fragment of the input data.

Supply				Ext. demand		Demanc																								
In mln ton ref. Fuel.		Production (generation)	Imports	Exports	Total consumption, RF	Conversion into other forms of energy	Processing into fuel	Processing into non-fuel products	As non-fuel resources	Agriculture and forestry	Mining and quarrying	Manufacturing	Manufacture of food products	Manufacture of textiles	Manufacture of footwear and leather	Manufacture of products of wood	Manufacture of paper, printing trade	Manufacture of coke and refined petroleum products	Chemical industry	Manufacture of rubber and plastic	Manufacture of mineral products	Manufacture of metals	Machine tool manufacture	Manufacture of electronics	Manufacture of motor vehicles	Generation and distribution of electric power, gas and water	Construction	Logistics	Population	Other
Oil and natural gas liquids	2021	754,3	0,0	349,5	404,8	0,7	370,3	33,3	0,2	0,0	0,2	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
Gas		856,0	11,4	257,1	610,4	316,6	38,3	62,3	12,4	2,1	29,0	50,7	2,0	0,1	0,0	0,3	0,8	5,8	3,5	0,1	15,1	17,8	2,6	0,3	0,8	1,7	4,6	37,2	6,2	0,1
Coal		250,5	20,1	156,1	114,5	76,0	27,9	0,1	0,3	0,1	0,3	2,8	0,1	0,0	0,0	0,0	0,0	0,0	0,3	0,0	0,8	1,6	0,0	0,0	0,0	2,8	0,0	0,2	1,0	0,0
Refined fuels		411,8	7,0	168,8	250,0	87,0	96,6	21,2	2,9	0,5	6,5	11,9	0,5	0,0	0,0	0,1	0,2	1,3	0,8	0,0	3,5	4,3	0,6	0,1	0,2	1,0	1,0	8,3	1,6	0,0
Combustible energy waste		22,8	0,0	0,0	22,8	5,8	0,0	0,1	0,1	0,1	0,0	8,3	0,2	0,0	0,0	0,6	0,3	1,2	0,2	0,0	0,1	5,5	0,0	0,0	0,0	0,2	0,0	0,0	0,0	0,0
Electrical energy		417,2	1,1	5,1	413,2	0,9	0,0	0,0	0,0	7,2	49,3	114,4	6,0	0,5	0,1	1,7	6,6	10,2	16,9	1,8	6,0	55,8	1,2	1,8	4,3	44,8	4,3	30,9	45,9	2,5
Thermal energy		172,9	0,0	0,0	172,9	0,0	0,0	0,0	0,0	4,4	5,8	67,0	6,3	0,3	0,1	1,8	6,8	12,8	20,1	0,7	3,2	9,0	1,5	1,1	2,9	12,9	0,5	3,0	12,5	0,3
Renewable sources		1,2	0,0	0,0	1,2	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,2	0,0
		2886,8	39,6	936,6	1989,9	487,1	533,0	116,9	15,9	14,3	91,2	255,1	15,1	1,0	0,2	4,5	14,7	31,3	41,9	2,7	28,7	94,1	5,9	3,3	8,1	63,4	10,5	79,6	68,4	3,0
Oil and natural gas liquids	2022	770,2	0,0	356,9	413,3	0,7	378,0	34,0	0,2	0,0	0,2	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
Gas		827,4	9,9	222,6	614,7	318,8	38,5	62,7	12,5	2,1	29,2	51,1	2,0	0,1	0,0	0,3	0,8	5,9	3,5	0,1	15,2	17,9	2,7	0,3	0,8	1,7	4,7	37,4	6,2	0,1
Coal		255,6	21,5	161,8	115,3	76,6	28,1	0,1	0,3	0,1	0,3	2,9	0,1	0,0	0,0	0,0	0,0	0,0	0,3	0,0	0,8	1,6	0,0	0,0	0,0	2,9	0,0	0,2	1,0	0,0
Refined fuels		410,1	6,9	164,0	253,0	87,7	98,4	21,4	2,9	0,5	6,6	11,9	0,5	0,0	0,0	0,1	0,2	1,3	0,9	0,0	3,5	4,3	0,6	0,1	0,2	1,0	1,0	8,3	1,6	0,0
Combustible energy waste		21,9	0,0	0,0	21,9	5,6	0,0	0,1	0,1	0,1	0,0	8,0	0,2	0,0	0,0	0,6	0,3	1,2	0,2	0,0	0,1	5,3	0,0	0,0	0,0	0,2	0,0	0,0	0,0	0,0
Electrical energy		423,1	1,1	5,1	419,0	1,0	0,0	0,0	0,0	7,3	50,0	116,0	6,1	0,5	0,1	1,7	6,7	10,4	17,1	1,8	6,1	56,6	1,2	1,8	4,4	45,5	4,4	31,3	46,5	2,6
Thermal energy		166,0	0,0	0,0	166,0	0,0	0,0	0,0	0,0	4,2	5,6	64,3	6,0	0,3	0,1	1,7	6,5	12,2	19,3	0,7	3,1	8,7	1,4	1,0	2,8	12,3	0,5	2,9	12,0	0,3
Renewable sources		1,3	0,0	0,0	1,3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,3	0,0
		2875,6	39,4	910,4	2004,6	490,3	543,1	118,3	16,0	14,3	91,9	254,1	14,9	1,0	0,2	4,5	14,5	30,9	41,3	2,7	28,8	94,5	5,9	3,2	8,1	63,5	10,6	80,2	68,7	3,0

Based on machine learning methods and statistics, we will create forecasts for the next 7 years (2023–2030).

3. FORECASTING SMALL SAMPLES OF MACROECONOMIC INDICATORS OF THE RUSSIAN ENERGY SECTOR

3.1. Machine-Learning-Based Linear Forecast

Linear regression estimation and forecasting imply a number of conditions, such as the normality of the sample distribution, the robustness of the predicted trend, a small quantity of outliers, and the homoscedasticity of the series. Despite the fact that the linear method is the fastest, it is extremely sensitive (not robust) to outliers and produces the largest confidence interval (forecast

error). This method should be used only in cases where the immediacy is much more important than the accuracy of the result [4–6].

Methodology. To use a machine-learning-based linear forecasting method, it is necessary to determine the permissible limits of the values of the predicted variable. It should be noted that a preliminary determination of the permissible limits of the values of functions and their arguments is mandatory in almost all machine learning methods using gradient or stochastic optimization. Usually, the limits are determined by judgmental methods (so, for example, the annual gas production limit depends not so much on planning as on the availability of actual natural, material, and human resources). If it is impossible to obtain the values collected from experts, the limits are defined as the upper and lower likelihood bounds corresponding to the standard data spread being two interquartile ranges of sample variance, or as logically reasonable values corresponding to theoretically achievable maximum and minimum.

It should also be noted that in addition to the boundaries that ensure values are within the likelihood range, the Tikhonov regularization method is also applied to linear forecasts, which consists of eliminating sample outliers based on the assumption of a normal input data distribution, followed by linear imputation of the average values to replace identified anomalies. Another widely used method is linear forecasting with scheduling imputation, where outliers are determined not by higher-order statistical moments, but rather based on the hypothesis that they are related to the known crisis events.

In this case, the values corresponding to crisis moments are imputed using the smart method (less commonly, the linear method).

1. The limits of permissible forecast values are defined.
2. If necessary, scheduling or statistical regularization is carried out.
3. The linear regression equation is calculated.
4. After checking the results, forecast values that exceed the limits are replaced.

For the mathematical implementation of the logical component of the fourth algorithm step, piecewise constant functions are usually used, such as:

$$\text{Kronecker function: } \delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases}$$

$$\text{Sign function: } \operatorname{sgn} i = \begin{cases} +1, & i > 0, \\ 0, & i = 0, \\ -1, & i < 0, \end{cases}$$

$$\text{Heaviside function: } \mathrm{H}i = \begin{cases} 1, & i \geqslant 0, \\ 0, & i \leqslant 0. \end{cases}$$

Using these functions as product elements that take values of either 1 or 0, they can be employed to include or exclude blocks of values in additive equations.

In general, the linear forecast equation looks as follows:

$$\begin{aligned} y_{n+f} = {} & H(k_1)H\big(Y_{+0} - (k_1(n+f)+k_2)\big)\big(k_1(n+f)+k_2\big) \\ & + H(-k_1)H\big(-Y_{-0} + (k_1(n+f)+k_2)\big)\big(k_1(n+f)+k_2\big) \\ & + H(k_1)H\big(-Y_{+0} + (k_1(n+f)+k_2)\big)Y_{+0} \\ & + H(-k_1)H\big(Y_{-0} - (k_1(n+f)+k_2)\big)Y_{-0}, \end{aligned}$$

where

$$k_1 \text{ (slope coefficient)} = \frac{n\Sigma_{i=1}^{n} iY_i - \Sigma_{i=1}^{n} i\Sigma_{i=1}^{n} Y_i}{\Sigma_{i=1}^{n} i^2 - (\Sigma_{i=1}^{n} i)^2},$$

$$k_2 \text{ (base coefficient)} = \frac{\Sigma_{i=1}^{n} Y_i - \left(\frac{\Sigma_{i=1}^{n} iY_i - \Sigma_{i=1}^{n} i\Sigma_{i=1}^{n} Y_i}{\Sigma_{i=1}^{n} i^2 - \left(\Sigma_{i=1}^{n} i\right)^2}\right)\Sigma_{i=1}^{n} i}{n},$$

y_{n+f}—forecast value for a given date, Y_i—sample elements, Y_{+0}—upper limit of permissible forecast values, Y_{-0}—lower limit of permissible forecast values, n—sample size, f—prediction interval, $y_i = k_1 i + k_2$—linear regression equation.

Explanation of the logical components: $H(k_1) = 1$, if the function increases; $H(-k_1) = 1$, if the function decreases; $H\big(Y_{+0} - (k_1(n+f) + k_2)\big) = 1$, if the function did not cross the upper limit; $H\big(-Y_{-0} + (k_1(n+f) + k_2)\big) = 1$, if the function did not cross the lower limit; $H\big(-Y_{+0} + (k_1(n+f) + k_2)\big) = 1$, if the function crossed the upper limit; $H\big(Y_{-0} - (k_1(n+f) + k_2)\big) = 1$, if the function crossed the lower limit.

3.2. ETS Exponential Smoothing Forecasting Model

The abbreviation ETS (error/trend/seasonality) means that the following predictive factors are taken into account when making a forecast:

1. The general tendency of growth or decline (trend).
2. Seasonal fluctuations (periodic/harmonic motion).
3. The error significance level increases as we approach the starting point of the forecast.

The ETS forecast is also referred to as the exponential smoothing forecast, which is related to the method of calculating the significance of the error (exponential amplification).

As in linear forecasting, the main approximation accuracy criterion in the ETS method is the squared deviation, but in this case, instead of calculating the sum of squares, we obtain the sum of the products of the squares by their weights, which increase exponentially as they approach the forecast point [7–9].

The result of the forecast is an additive function consisting of linear and periodic components [10–14].

The forecast algorithm:

Step 1. Definition of the permissible forecast value limits.
Step 2. Identification of the linear trend using the classic diagonal method.
Step 3. Approximation of the trend residuals by a periodic function.
Step 4. Expert definition of the "data inertia" value.
Step 5. Sample approximation by a linear function, taking into account the error weights.
Step 6. Summation of linear and periodic functions.
Step 7. Forecast generation based on the overall trend.
Step 8. Verification and replacement of the forecast values that exceed the limits.

Let us formalize the implementation of the ETS forecast as part of the algorithm steps mathematically:

Step 2. Identification of the linear trend using the classic diagonal method:

$$y_{i_1} = \frac{n\left(n\Sigma_{i=1}^{n} iy_i - \Sigma_{i=1}^{n} i\Sigma_{i=1}^{n} y_i\right)}{\Sigma_{i=1}^{n} i^2 - (\Sigma_{i=1}^{n} i)^2} + \frac{\Sigma_{i=1}^{n} y_i - \left(\frac{\Sigma_{i=1}^{n} iy_i - \Sigma_{i=1}^{n} i\Sigma_{i=1}^{n} y_i}{\Sigma_{i=1}^{n} i^2 - \left(\Sigma_{i=1}^{n} i\right)^2}\right)\Sigma_{i=1}^{n} i}{n},$$

where y_{i_1}—initial linear trend for obtaining seasonal residuals, y_i—sample elements, i—element indices, n—sample size,

$$k_1 \text{ (slope coefficient)} = \frac{n\Sigma_{i=1}^{n} iY_i - \Sigma_{i=1}^{n} i\Sigma_{i=1}^{n} Y_i}{\Sigma_{i=1}^{n} i^2 - (\Sigma_{i=1}^{n} i)^2},$$

$$k_2 \text{ (base coefficient)} = \frac{\Sigma_{i=1}^{n} Y_i - \left(\frac{\Sigma_{i=1}^{n} iY_i - \Sigma_{i=1}^{n} i\Sigma_{i=1}^{n} Y_i}{\Sigma_{i=1}^{n} i^2 - \left(\Sigma_{i=1}^{n} i\right)^2}\right)\Sigma_{i=1}^{n} i}{n},$$

where Y_i—sample elements.

Step 3. Approximation of the trend residuals by a periodic function:

$$\sum_{i=1}^{n}\left(k_3\varphi(k_4 i) - (k_1 i + k_2 - Y_i)\right)^2 \to \min,$$

where k_1—slope coefficient, k_2—base coefficient, k_3—selected amplitude coefficient of the periodic function, k_4—selected frequency coefficient of the periodic function.

Step 5. Sample approximation by a linear function with regard to the error weights:

The value of the data inertia coefficient $\alpha \in]0; 1]$ is usually determined according to the expert opinions; in the absence of such, it is recommended to take a value within 0.25–0.30. The significance of up-to-date data grows with the increase of the coefficient, while the significance of "stale" data decreases.

Step 6. Summation of linear and periodic functions:

$$\sum_{i=1}^{n}\left(\frac{e^{\alpha i}(k_5 i + k_6 - Y_i)}{e^{\alpha n}}\right)^2 \to \min,$$

where α—coefficient of "data inertia" (staleness), k_5—selected slope coefficient, k_6—selected base coefficient, Y_i—sample elements.

Step 7. The final prediction equation without limits looks as follows:

$$y_{n+f} = k_5(n+f) + k_6 + k_3\varphi(k_4(n+f)).$$

Step 8. Next, we check the results and replace forecast values that exceed the limits.

Thus, we obtain the forecast with limits:

$$\begin{aligned} y_{n+f} = {} & H(k_5)H\Big(Y_{+0} - \Big(k_5(n+f) + k_6 + k_3\varphi(k_4(n+f))\Big)\Big)\big(k_5(n+f) + k_6 + k_3\varphi(k_4(n+f))\big) \\ & + H(-k_5)H\Big(-Y_{-0} + \Big(k_5(n+f) + k_6 + k_3\varphi(k_4(n+f))\Big)\Big)\big(k_5(n+f) + k_6 \\ & + k_3\varphi(k_4(n+f))\big) + H(k_5)H\Big(-Y_{+0} + \Big(k_5(n+f) + k_6 + k_3\varphi(k_4(n+f))\Big)\Big)Y_{+0} \\ & + H(-k_5)H\Big(Y_{-0} - \Big(k_5(n+f) + k_6 + k_3\varphi(k_4(n+f))\Big)\Big)Y_{-0}. \end{aligned}$$

3.3. Neural-Network-Based Forecast

A neural network functions through interconnected nodes called neurons. These nodes are organized into a multi-layered structure that resembles the structure of the human brain.

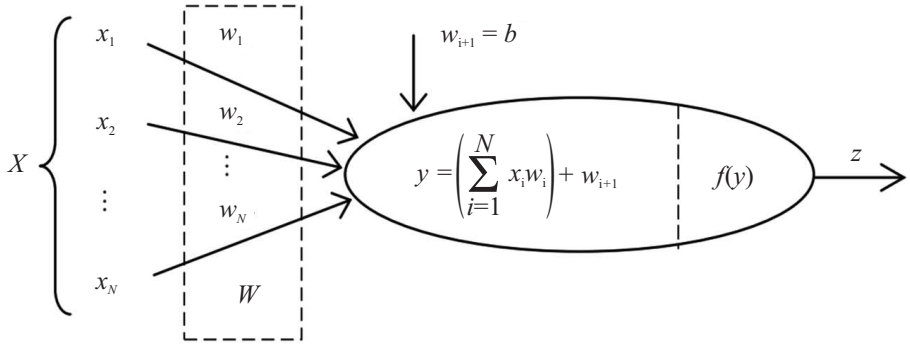


Fig. 1. Schematic drawing of an artificial neural network.

The most elementary neural network, called a perceptron, is a simplified simulation of the activity of a biological network consisting of neurons and connections between them.

The structure of a formal neuron is shown in Fig. 1.

A set of signals is coming to the input. Each signal is subjected to a weighting procedure, i.e., multiplied by a preset coefficient:

$$net_j = \sum_{i=1}^{N} x_i w_{ij},$$

where net_j—total value that would result from processing all incoming signals of neuron j (neuron's summed input), N—total number of elements whose input parameters affect the input signal j; w_{ij}—weight determining the strength of the connection linking neuron i with neuron j.

Summing up all the input signals, previously multiplied by the corresponding weights, we get the total input signal of the neuron.

Each neuron functions according to its own internal algorithm that determines the input signal. This algorithm calculates the result based on the total value received by the neuron. This rule is called the activation function $f(x)$.

Let us turn to the Widrow–Hoff learning rule, also known as the delta rule. Its objective is to minimize the root-mean-square error that has arisen in the neural network. This error is calculated for the input data using the formula:

$$E = \frac{1}{2}(Y - d)^2,$$

where d—desired output value.

Each neuron performs weighted sum calculations using the following formula:

$$S = w_1 X_1 + w_2 X_2 - b,$$

where b—defines the critical level (threshold).

For example, consider the linear activation function given as $Y = x$. In this case, the functional of the error will be determined by the following formula:

$$\frac{\partial E}{\partial w_1} = (X - d)X_1,$$
$$\frac{\partial E}{\partial w_2} = (X - d)X_2,$$
$$\frac{\partial E}{\partial b} = -(X - d).$$

The weights and biases of the neuron are determined by the following formulas:

$$w_i(t+1) = w_i(t) - \alpha(Y-d)X_i,$$
$$b(t+1) = b(t) + \alpha(Y-d).$$

In this particular scenario, the error functional is defined as:

$$E = \frac{1}{2}\Sigma_j(Y_j - d_j)^2.$$

The coefficients determining the weights and biases of neurons are calculated using the following formulas:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha(Y_j - d_j)X_i,$$
$$b_j(t+1) = b_j(t) - \alpha(Y_j - d_j).$$

The error back-propagation algorithm serves as the core tool for training multilayer neural networks based on the principle of the forward signal propagation. Training essentially boils down to two consecutive stages covering each layer of the network: forward and backward passes. Synaptic weights, denoted as $w^k_{i,j}$ (i is the weight number, j is the neuron number, k is the layer number), are adjusted to achieve the best match between the actual signal output and the desired output. The output value of the jth neuron located in the kth layer is determined as follows:

$$Y^k_j = F\left(\Sigma w^k_{i,j} Y^{k-1}_i - b^k_j\right).$$

The input signal for the jth neuron of the last layer is calculated as follows:

$$Y_j = F\left(\Sigma w_{i,j} Y^{n-1}_i - b_j\right).$$

The error functional of the neural network is defined as:

$$E = \frac{1}{2}\Sigma_j(\gamma_j)^2,$$

where $\gamma_j = Y_j - d_j$—error of the jth neuron located in the output layer.

Error of the jth element located on the kth hidden layer:

$$\gamma^k_j = \frac{\partial E}{\partial Y^k_j} = \Sigma_j \frac{\partial E}{\partial Y_j}\frac{\partial Y_j}{\partial S_j}\frac{\partial S_j}{\partial Y^k_j} = \Sigma_j \frac{\partial E}{\partial Y_j}\frac{\partial Y_j}{\partial S_j} w_{i,j} = \Sigma_j(Y_j - d_j)F'(S_j)w_{i,j} = \Sigma_j \gamma_j F'(S_j)w_{i,j}.$$

Error gradients:

$$\frac{\partial E}{\partial w_{i,j}} = \frac{\partial E}{\partial Y_j}\frac{\partial Y_j}{\partial S_j}\frac{\partial S_j}{\partial w_{i,j}} = \gamma_j F'(S_j)Y^k_j,$$
$$\frac{\partial E}{\partial b_j} = \frac{\partial E}{\partial Y_j}\frac{\partial Y_j}{\partial S_j}\frac{\partial S_j}{\partial b_j} = -\gamma_j F'(S_j),$$
$$\frac{\partial E}{\partial w^k_{i,j}} = \Sigma_j \frac{\partial E}{\partial Y_j}\frac{\partial Y_j}{\partial S_j}\frac{\partial S_j}{\partial Y^{k-1}_j}\frac{\partial Y^{k-1}_j}{\partial S^{k-1}_j}\frac{\partial S^{k-1}_j}{\partial w^k_{i,j}} = \gamma_j F'\left(S^k_j\right)Y^k_j.$$

The weights and biases of the neurons are determined by the following formulas:

$$w^k_{ij}(t+1) = w^k_{ij} - \alpha\gamma^k_j F'\left(S^k_j\right)Y^k_j,$$
$$b^k_j(t+1) = b^k_j - \alpha\gamma^k_j F'\left(S^k_j\right),$$

where $\alpha(0 < \alpha < 1)$—learning rate.

The algorithm will be executed until $E > E_m$, E_m, where Em is the desired value of the root mean square error of the neural network.

Let us create a simple, non-recurrent, perceptron neural network for the autoregression of each data series:

$$y_{n+f} = k_{n0}\tanh k_{n1}\tanh\left(\sum_{i=1}^{i=m} k_{ni}\tanh\left(\sum_{j=i}^{i=1} k_{nij}\tanh(Y_{ij})\right)\right),$$

where y_{n+f}—forecast at time $n+f$, Y_{ij}—sample elements, $\tanh(x)$—hyperbolic tangent function, k_{nij}—coefficients of the neuron input weights calculated during training, f—forecast lead time, m—number of input neurons, n—number of inputs of each first-layer neuron.

Let us train the neural network using the standard stochastic Hopfield method or the gradient descent method (EBP), minimizing the sum of squared deviations.

3.4. The Ensemble Forecast

Figure 2 shows graphs for all three of the above types of forecasts (linear forecast, ETS forecast, neural forecast).

As one can see, the forecasts are considerably different:

1. The *linear* forecast shows a significant decrease.
2. The *ETS* forecast shows a decrease, an increase, and then a decrease again.
3. The *neural* forecast shows an increase.

Such significant discrepancies in forecasts usually appear when predicting small and ultra-small time series (less than 30 elements), since their statistical deviations are very large and, for example, in a sample of 20 elements, may be up to 2σ.

In such cases, a predictive ensemble (combined forecast) is created from a set of disparate forecasts, the main task of which is to give a consensual prediction depending on the degree of

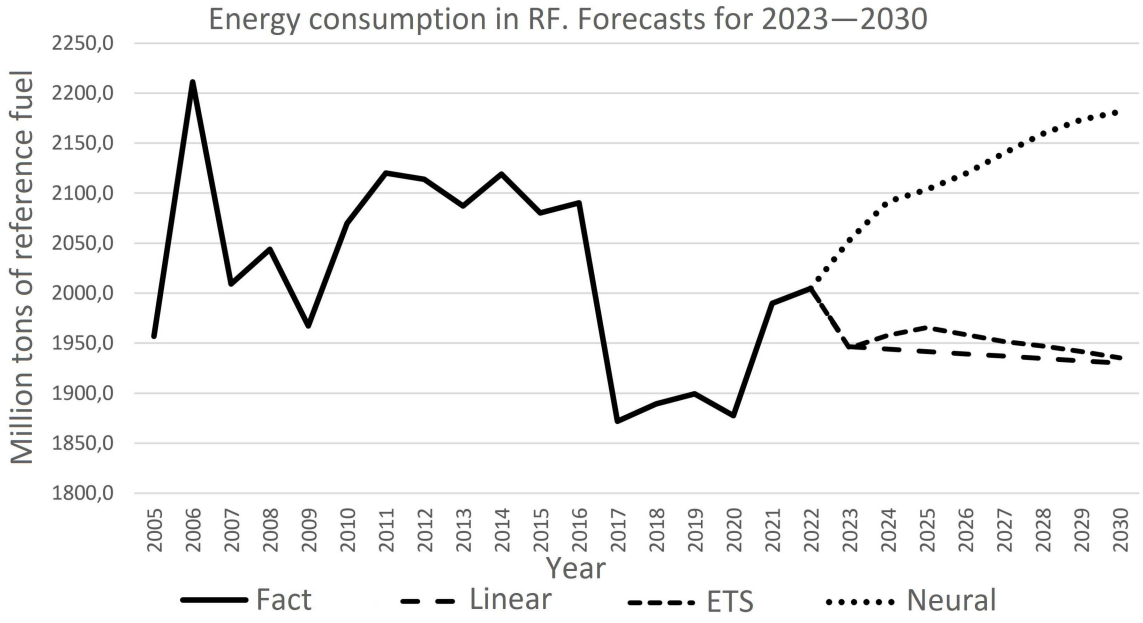


Fig. 2. Linear forecast, ETS forecast, neural forecast.

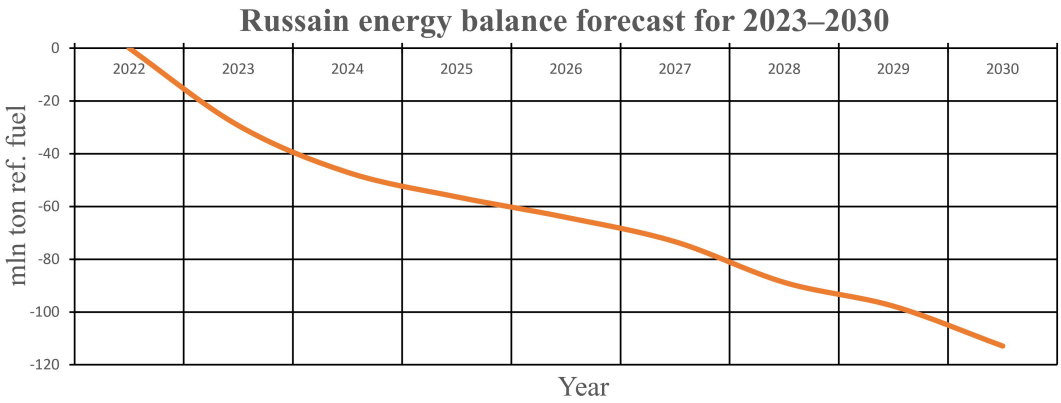


Fig. 3. Russian energy balance forecast.

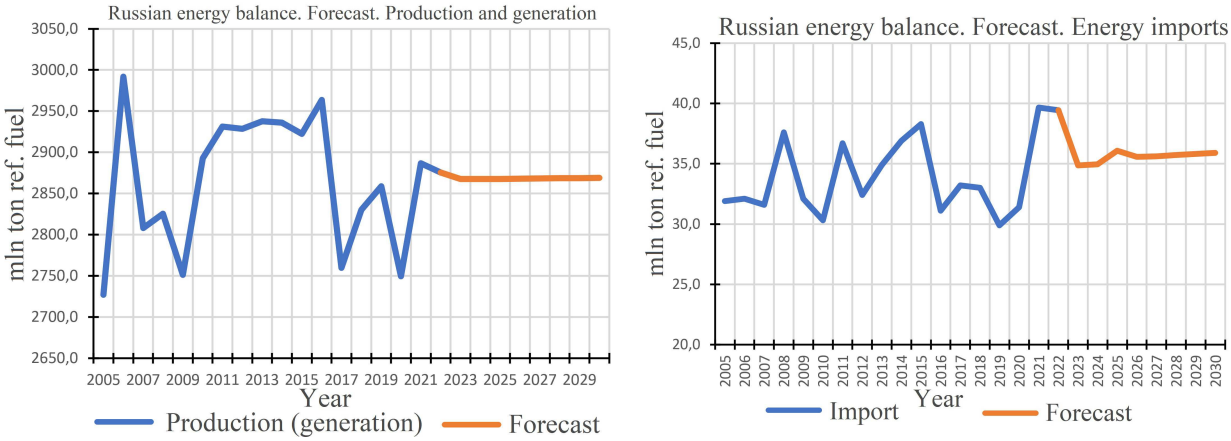


Fig. 4. Forecast. Production and generation. Energy imports.

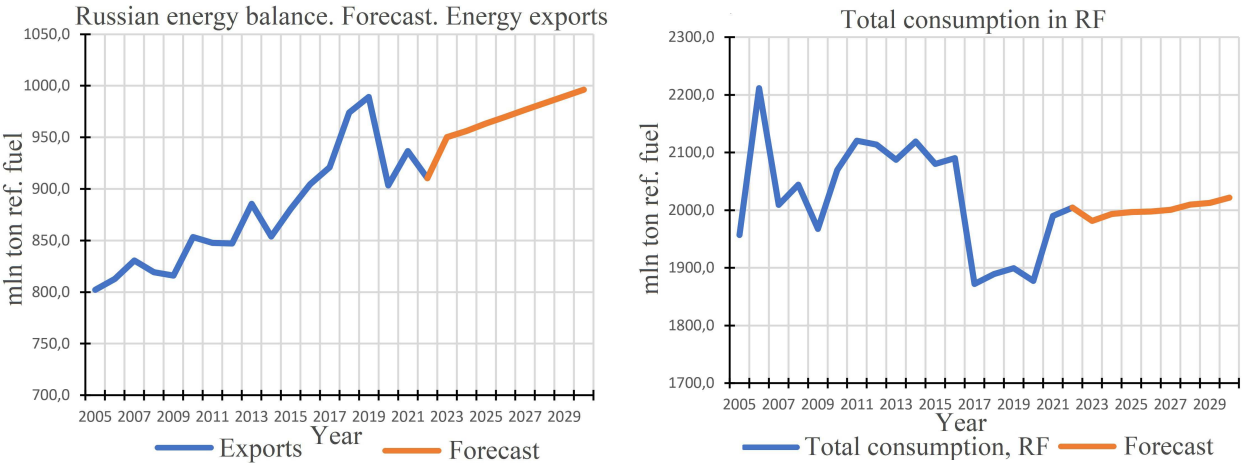


Fig. 5. Forecast. Energy exports. Total consumption in RF.

confidence in a particular methodology:

$$y_{n+f} = \sum_{i=1}^{m} d_i y_{n+f_i}; \quad \sum_{i=1}^{m} d_i = 1,$$

where y_{n+f}—forecast at a time $n+f$, y_{n+f_i}—forecast according ith method, d_i—confidence coefficient of the ith method, m—number of forecasting methods.

The simplest ensembling method is the Bayesian method, which assumes an equal degree of confidence in all forecasting methods [15–17]:

$$y_{n+f} = \sum_{i=1}^{m} \frac{y_{n+f_i}}{m}.$$

Using this method, we will calculate ensemble forecasts and plot such forecasts both for individual indicators and the energy balance as a whole.

Let us consider the general ensemble forecast of the energy balance (Fig. 3).

As we can see, if current trends continue in the period 2023–2030, an energy deficit in the amount of 20 to 120 million tons of reference fuel is expected, which, with the total volume of current production, ranges from 0.7% to 4.2%. Such a deficit can be easily covered by a planned increase in production by 0.6–0.7% per year.

Now let us find out the causes of a possible shortage. Figure 4 shows that total production will remain unchanged, while energy imports will increase slightly.

Figure 5 shows that exports and domestic consumption will increase significantly.

Thus, even an intuitive analysis of the graphs let us conclude that the growth of domestic consumption and exports of energy resources will be the main causes of a possible energy shortage.

4. RESULTS. ASSESSMENT OF FORECAST QUALITY

The estimate of the possible deviation of the predicted value represents the estimated limits of fluctuations in the predicted value, defined as the confidence interval of the forecast. They are usually designated as "pessimistic" and "optimistic" limits of the forecast. The value of the confidence interval is based on the assumption that the predicted value has a Gaussian normal distribution, as well as on the amount of uncertainty allowed in the forecast.

The value of the confidence interval is calculated for each forecast period using the formula:

$$\Delta X_n = nZ_\alpha RMSE,$$

where n—prediction interval, Z_α—multiplier of the significance level, $RMSE$—standard deviation of the trend.

Thus, each subsequent value of the predicted variable expands its probabilistic bounds by at least one standard deviation, and the maximum significance level of the forecast corresponds to a deviation of 68%.

Let us estimate the confidence interval of the components of linear, exponential, neural, and ensemble forecasts of the energy balance, which include exports, imports, production, and consumption of energy resources.

For this estimation, it is first necessary to find the standard sample approximation error for the linear, exponential, and neural trends of these components. In all three cases, it can be estimated as:

$$RMSE = \sqrt{\frac{\sum\limits_{i=1}^{n}(X_i - x_i)^2}{n-1}},$$

where X_i—actual values of the predicted variable, x_i—values of the predictive trend, n—sample size of the actual values.

In this case, the limits of the confidence interval for the predicted values are defined as:

$$D_{i>n} = x_i \pm Z_\alpha(i-n)RMSE,$$

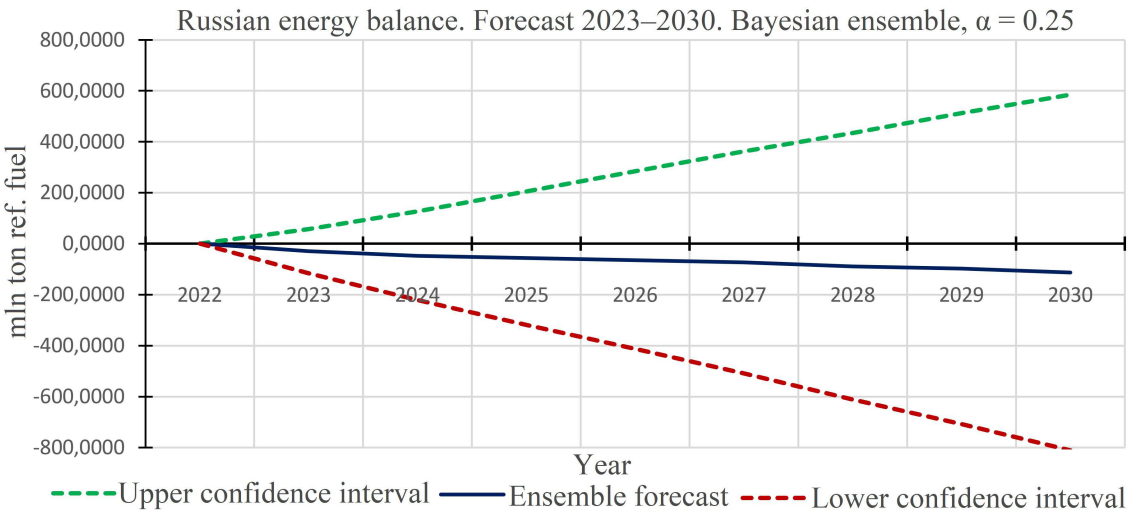


Fig. 6. Ensemble forecast of the Russian energy balance with confidence interval.

where x_i—values of the predictive trend, n—sample size of the actual values, Z_α—multiplier of the significance level, $RMSE$—standard deviation of the trend.

Since the actual energy balance is a value reduced to zero, it is impossible to estimate its deviation from any predictive trends. In this case, calculated estimates of the error in energy balance components can be used, and regardless of whether the component has a negative or positive effect on the energy balance, the fraction of error it introduces should be accounted for as increasing the total error.

For each type of energy balance forecasts, we obtain the values of the cumulative RMSE using Bayesian ensembling and estimate the error of the ensemble forecast as an arithmetic mean (see Table 2).

Table 2. Benchmarking of forecasts

Energy balance forecast	Absolute cumulative RMSE	Relative cumulative RMSE
Linear	74.5223	5.14%
ETS (exponential)	75.4225	5.22%
Neural	77.5736	5,67%
Ensemble	75.8395	5.23%

Let us plot the resulting ensemble forecast with confidence intervals (see Fig. 6).

It should be noted that the confidence interval reflects outer limits beyond which the predicted value will not exceed at the specified significance level [18–20]. However, the display of the confidence interval, especially for long-term prognostic periods, suggests a low forecast accuracy and a significant spread of possible values. Hence, for graphs, it is recommended to use a predictive corridor, which is a visual representation of the standard deviation or the results of frequency analysis. In this case, the display of the forecast limits will be smoother, narrower, and non-expansive. Let us consider the limits of the predictive corridor for displaying the ensemble forecast of the energy balance.

In this case, the limits of the predictive corridor for the predicted values are defined as:

$$F_{i>n} = x_i \pm Z_\alpha RMSE,$$

where x_i—values of the predictive trend, n—sample size of the actual values, Z_α—multiplier of the significance level, $RMSE$—standard deviation of the trend.

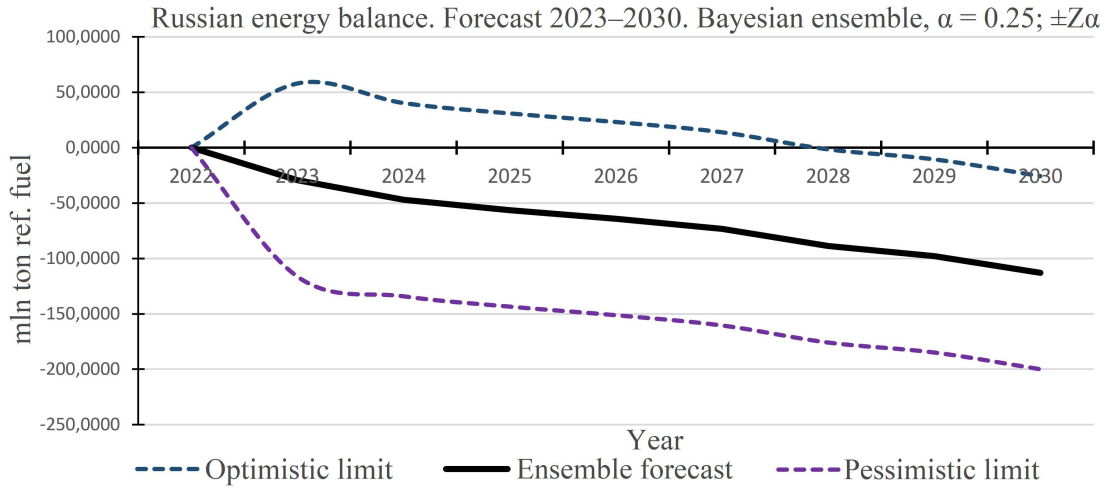


Fig. 7. Ensemble forecast of the Russian energy balance with predictive corridor.

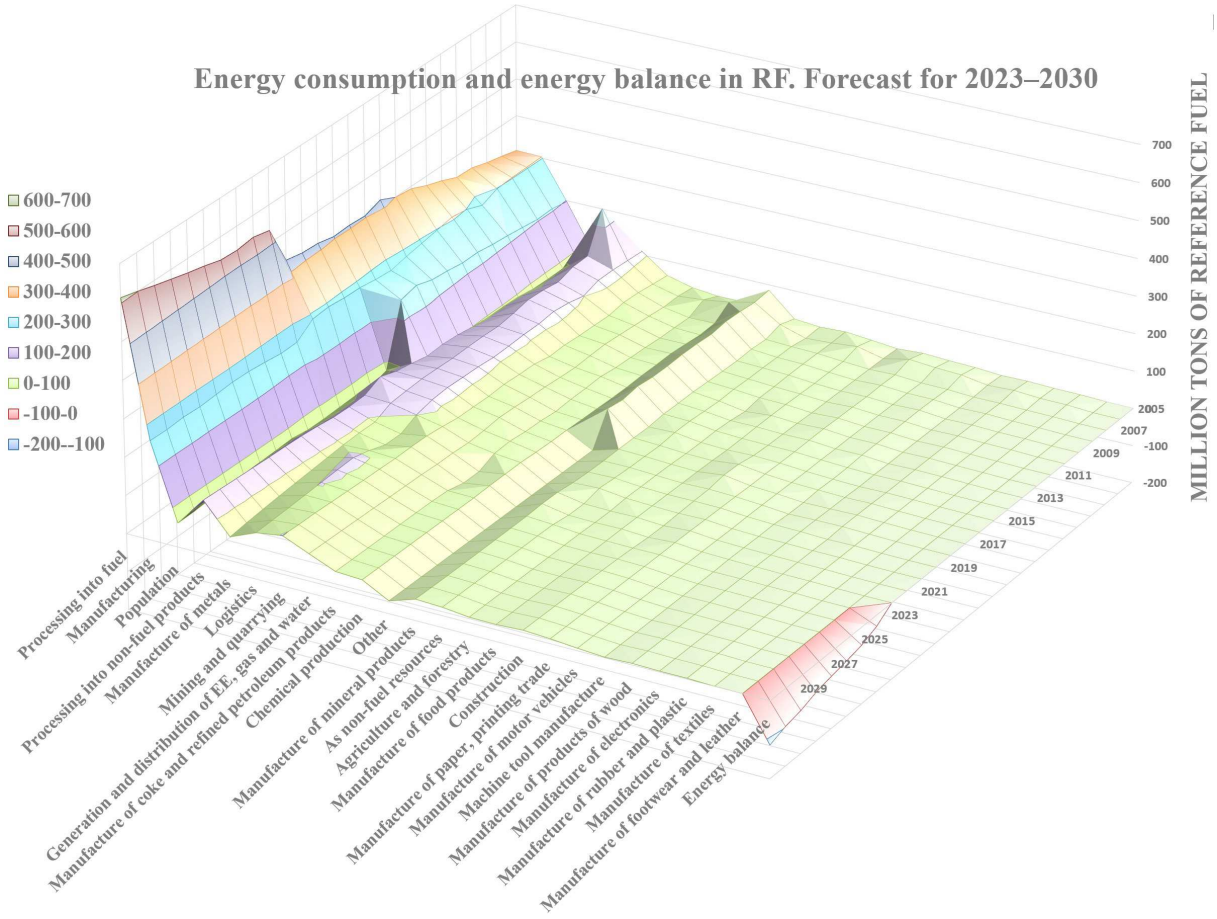


Fig. 8. Three-dimensional graph of sectoral forecasts.

So, for $\alpha = 0.25$, the expected deviation will be 1.15σ, which is reflected in the graph in Fig. 7.

As the figure shows, this representation of the forecast is much more favorable in terms of psychological perception.

As a conclusive outcome, a three-dimensional graph of sectoral forecasts is constructed (Fig. 8) in order to identify industries where energy supply requires special attention.

From the graph (the green area of near-zero values on the right), it immediately becomes clear which industries in the Russian Federation are not receiving enough attention. In order to better diversify the economy, these industries need to be stimulated.

5. CONCLUSION

Based on the time series analysis of indicators of the energy balance of the Russian Federation, both classical statistical models and deep learning methods, as well as their ensemble combination, were implemented and investigated.

The results revealed the advantages and limitations of each forecasting model. It is shown that the use of an ensemble model combining the predictions of various methods provides higher stability and accuracy of the results compared with individual models.

The economic value of the developed forecasting system is confirmed by the possibility of its application for strategic and operational planning, risk management, and the formation of competitive advantages in the energy market. The practical implementation of the models and the constructed ensemble based on real data has shown their applicability to the tasks facing energy companies, traders, and government regulators.

REFERENCES

1. UIISS. State Statistics. https://www.fedstat.ru
2. Rosstat. Federal State Statistics Service. https://www.rosstat.gov.ru
3. Ministry of Energy. https://www.minenergo.gov.ru
4. Vasil'ev, M.V. and Dranko, O.I., Two-level revenue forecasting model for a large-scale energy system, *Sensors and systems*, 2023, vol. 267, no. 2, pp. 71–78.
5. Dranko, O.I., On Forecasting Enterprise Conversion Financing, *South Ural State University Bulletin. Series: Computer technology, management, radio electronics*, 2020, vol. 20, no. 4, pp. 74–82.
6. Dranko, O.I. and Blagodarnyj, E.V., Modeling the Destruction of the Value of Russian Energy Companies, *Information and mathematical technologies in science and management*, 2022, vol. 27, no. 3, pp. 104–112.
7. Dranko, O.I., Rezchikova, A.F., Stepanovskaya, I.A., et al., Scenario Modeling of the Country's Development Based on Indicative Planning, *Problems of Management*, 2024, no. 5, pp. 25–41.
8. Dranko, O.I. and Taroyan, K.K., Revenue Forecasting for a Fast-Growing Company Using a Logistic Curve, *Automation and Modeling in Design and Management*, 2024, vol. 24, no. 2, pp. 84–92.
9. Dranko, O.I. and Taroyan, K.K., On a Revenue Forecasting Model for a Fast-Growing Enterprise, *South Ural State University Bulletin. Series: Computer Technology, Management, Radio Electronics*, 2023, vol. 23, no. 4, pp. 66–75.
10. Ivanyuk, V.A., A Long-Term Forecasting Technique Based on a Multi-Trend Forecast, *Soft Measurements and Calculations*, 2023, vol. 73, no. 12–1, pp. 128–138.
11. Brockwell, P.J. and Davis, R.A., *Introduction to Time Series and Forecasting*, New York: Springer, 3rd ed., 2016.
12. Brown, R.G., *Smoothing Forecasting and Prediction of Discrete Time Series*, New York: Prentice Hall, 1963.
13. Chen, W., Xu, H., Liu, Z., et al., Hybrid Modeling of Energy Price Forecasting Combining Temporal Features and Economic Indicators, *Energy*, 2022, vol. 243, pp. 123–135.
14. Hamid, A., Islam, M.S., and Hasan, M.R., A Comprehensive Review of Deep Recurrent Neural Networks for Energy Price Forecasting , *Renewable and Sustainable Energy Reviews*, 2020, vol. 135, art. no. 110354.

15. Ivanyuk, V., The Method of Residual-Based Bootstrap Averaging of the Forecast Ensemble, *Financial Innovation*, 2023, vol. 9, no. 1, pp. 37.

16. Ivanyuk, V., Forecasting of Digital Financial Crimes in Russia Based on Machine Learning Methods, *Journal of Computer Virology and Hacking Techniques*, 2024, vol. 20, no. 3, pp. 349–362.

17. Li, K., Liu, X., Zhu, J., and He, Z., Electricity Price Forecasting Using Gradient Boosting and Random Forests, *Applied Energy*, 2019, vol. 253, pp. 113–122.

18. Smith, J. and Reeve, D., Forecasting Oil and Natural Gas Prices Using ARIMA and Exponential Smoothing Models, *Energy Economics*, 2021, vol. 93, pp. 105–120.

19. Wang, L., Zhou, Y., and Yang, G., Ensemble Forecasting Model Combining ARIMA, XGBoost, and LSTM for Short-Term Natural Gas Price Prediction, *Energy Reports*, 2022, vol. 8, pp. 1430–1440.

20. Zhang, G.P., Time Series Forecasting Using Hybrid ARIMA and Neural Network Models, *Neurocomputing*, 2003, vol. 50, pp. 159–175.

This paper was recommended for publication by A.A. Galyaev, a member of the Editorial Board

ISSN 0005-1179 (print), ISSN 1608-3032 (online), Automation and Remote Control, 2026, Vol. 87, No. 5, pp. 418–427.

SPECIAL ISSUE

Consideration of Soft Dependencies under Stochastic Uncertainty in Multi-Project Program Implementation

I. V. Burkova[*,a] **and A. V. Shchepkin**[*,b]

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
e-mail: [a] *irbur27@mail.ru,* [b] *av_shch@mail.ru*

Received July 8, 2025
Revised November 17, 2025
Accepted November 20, 2025

Abstract—This paper addresses the problems of managing a multi-project program under soft dependencies between some program projects. As a rule, the use of soft dependencies saves the time and/or cost of executing the next project, which decreases the time and/or cost of implementing the entire program. An example shows how rarely dealing with soft dependencies can be reduced to simple schemes without losing the problem essence. Consideration of stochastic uncertainty in project implementation is introduced. According to the conclusion, the effect from considering soft dependencies depends on the probability of realizing them.

Keywords: program, project, hard dependencies, soft dependencies, stochastic uncertainty in the realization of soft dependencies

DOI: 10.7868/S1608303226050035

1. INTRODUCTION

In a program consisting of interdependent projects (multi-project program), the results of one project can often be used in the execution of others, thereby decreasing the implementation time (duration) and/or cost of the entire program. Such dependencies between projects, without mandatory fulfillment but with a positive effect on program parameters, are called soft dependencies.

Problems with soft dependencies were investigated in [1–4]. Note that rather simple setups were studied in [1, 2], e.g., the effect from considering several soft dependencies was supposed to be additive, and no probabilistic characteristics [6–9] of the effect of soft dependencies, which is certainly present, were taken into account. This practice is not always applicable; see an illustrative example in the next section.

2. AN EXAMPLE OF ESTIMATING SOFT DEPENDENCIES IN A PROGRAM

Let a program consist of 5 projects. Their numerical parameters are combined in the table below.

This table has the following notation: $t_{\min}$ and $t_{\max}$ are the minimum and maximum project execution times (terms), respectively (in days); p is the probability of project completion within the corresponding term; t_c is the time saving of the next project (in days); finally, r_c is the resource saving of the next project (in million rubles).

The residential complex construction project has a hard deadline of 500 days. In the example, this is the only project that may be executed in the maximum term with a probability below 1. This situation occurs when the project deadline is set “from the outside” (e.g., by the customer),

Table 1. Numerical parameters of program projects

Project	Minimum term, $t_{\min}/p$	Maximum term, $t_{\max}/p$	Impact on projects	Saving t_c/r_c
1. Start-up of C1 concrete production line	12/0.5	20/1	2, 5	7/20, 5/100
2. Start-up of C15 concrete production line	15/0.6	22/1	5	8/150
3. Start-up of R0 rebar production line	7/0.8	11/1	4, 5	4/15, 6/120
4. Start-up of R15 rebar production line (can be used only with C15 concrete; when used together, $t_c = 2$ and $r_c = 5$)	9/0.7	12/1	5	7/180
5. Construction of a residential complex	400/0.75	500/0.95	—	—

and everything possible will be done in the calendar plan to complete the work by the specified date.

Now we explain this table. Residential complex construction can be initiated independently of the start-up of production lines. By assumption, the pit has already been dug. Then both concrete and rebar, needed immediately, will have to be purchased from suppliers, requiring additional time (to wait for delivery) and additional money. Currently, all these items are included in the construction project. Obviously, the cost of components purchased "on the side" is higher than in-house production. Therefore, given the plans to organize production lines for concrete and rebar, it makes sense to consider the self-supply of the construction project with these materials. This approach is financially reasonable for the contractor, and it remains to assess its time feasibility: first, the production lines have to be launched.

It is possible to start up a production line for concrete of universal grade C1 or improved (modified) grade C15. For example, the use of C15 eliminates the need for some additives, reduces the curing time of the structure, etc. The production of C15 can be established independently or based on the available line for C1. In the second option, the start-up time of the production line is reduced by 7 days, but it is possible to initiate this work only after launching the C1 line. This line is started in 20 days with probability 1 (surely). That is, with saving both time and resources, the production of C15 is definitely launched in 35 days. Under favorable conditions (everything can be accomplished in the shortest term), this time is reduced to 20 days with a probability of

$$p_{12\,\min} = p_{1\,\min} \times p_{2\,\min} = 0.5 \times 0.6 = 0.3,$$

where $p_{1\,\min}$, $p_{2\,\min}$, and $p_{12\,\min}$ are the probabilities of completing the first, second, or first and second projects, respectively, under soft dependencies within the minimum term.

Rebar can be considered similarly. The rebar of grade R15 is stronger, lighter, and more reliable, but can be used only with C15 concrete. Their simultaneous use adds 5 to the resource saving and 2 to the time saving. R0 rebar (universal) can be used with any grade of concrete. Based on the R0 line, the production of R15 can be launched in 12 days with the probability of

$$p_{34\,\min} = p_{3\,\min} \times p_{4\,\min} = 0.8 \times 0.7 = 0.56$$

and in 19 days with probability 1 (surely).

Obviously, in the described situation, all dependencies are soft and their effects cannot be simply summed. First, there are different scenarios for their use:

1. Start all projects independently; then, the time to launch production lines is likely to be reduced since, in a sequential start-up, it is possible to start working with complex lines at best only after 12 and 7 days (in both cases, these figures exceed the time saving due to soft dependencies). Construction can be started at least a week earlier as well. Note that the time saving due to soft dependencies is not considered here.

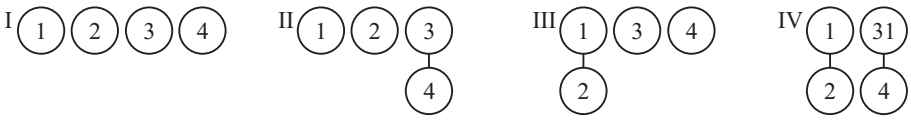


Fig. 1. Interaction options of projects 1–4.

2. Use in-house concrete (plain or improved), waiting 12 to 35 days and saving 100 to 150 million and 5 days. Due to the hard construction deadline, the 5 days also count appreciably.

3. Use in-house rebar, either universal or improved, waiting 7 to 19 days and saving 120 to 180 million and 6 days. Due to the hard construction deadline, the 6 days also count appreciably.

4. Use both in-house concrete and in-house rebar with their compatibility in mind (R15 can be used only with C15). R0 rebar is universal and can be used with any grade of concrete, but does not offer several advantages. In this scenario, the waiting time is taken into account only for concrete: the production of rebar is established earlier. But on the other hand, 11 days can be saved on the term reduction.

In reality, the above 4 scenarios generate a much larger number of particular options. We will deal with them by full enumeration, considering all logical and simultaneously reasonable options to combine the projects. There are not so many of them. Note that the first four projects interact with each other. The fifth one can be combined with any of their combinations. Figure 1 shows the interaction options of projects 1–4.

Figures 2–5 present the interaction options of all five projects for combinations I–IV, respectively. We briefly discuss their features.

I) The soft dependencies between the first four projects are neglected, all projects are executed in parallel and as fast as possible, albeit without resource and time savings. In this combination, there are five logical and reasonable options for including project 5:

1. Project 5 is also included without soft dependencies, in parallel with the others. This case corresponds to the situation without resource deficiency but with a desire or need to complete all projects as fast as possible. For I.1, we have $t_c = 0$ and $r_c = 0$.

2. Project 5 is included after project 3 and uses in-house rebar. When executing after project 3, the corresponding soft dependency (3–5) is used, and $t_c = 6$ and $r_c = 120$. The duration of project 3 does not exceed 11 days (and merely 7 days with a probability of 0.8). Subtracting the time saving, we obtain a shift of 1–5 days and resource saving.

3. Project 5 is included after project 4 and uses improved in-house rebar. The soft dependency (4–5) is considered, and $t_c = 7$ and $r_c = 180$. As a result, project 5 is shifted by 2–5 days, and the resource saving increases markedly to 180. An additional constraint is that R15 rebar can only be used with C15 concrete, so the contractor has to purchase concrete of that grade. Or, when the C15 concrete line is running, the contractor can start using in-house concrete.

4. Project 5 is included after projects 1 and 3, and both soft dependencies (1–5) and (3–5) are considered. Note that the option of executing project 5 after projects 1 and 4 is eliminated as illogical: the use of R15 requires concrete of a higher grade, and it makes no sense to wait for the production of unnecessary concrete. As a result, $t_c = 5 + 6 = 11$ and $r_c = 100 + 120 = 220$. We estimate the shift of project 5 by the largest time of projects 1 and 3, i.e., 12–20 days, or 1–9 days, taking the savings into account.

5. Project 5 is included after projects 2 and 4, and both soft dependencies (2–5) and (4–5) are considered. The option of executing project 5 after projects 2 and 3 makes no sense: during the start-up of the C15 line, the production of R15 has enough time to start, and these materials together give a much higher gain than when using C15 jointly with R0. As a result, $t_c = 8 + 7 + 2 = 17$ and

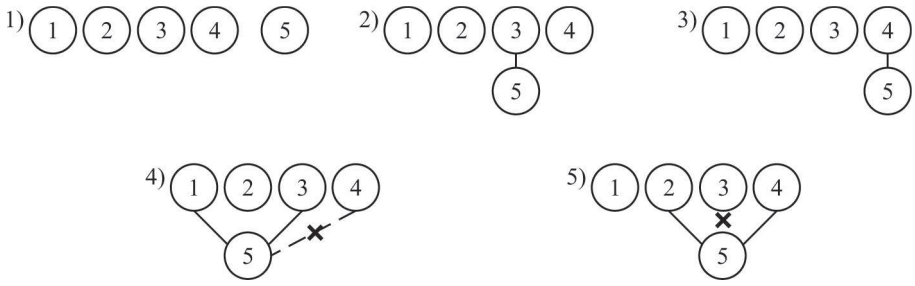


Fig. 2. Inclusion options for Project 5 in combination I.

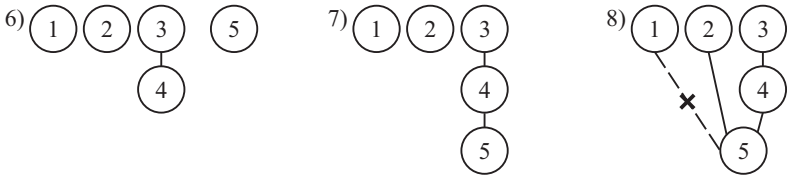


Fig. 3. Inclusion options for Project 5 in combination II.

Fig. 4. Inclusion options for Project 5 in combination III.

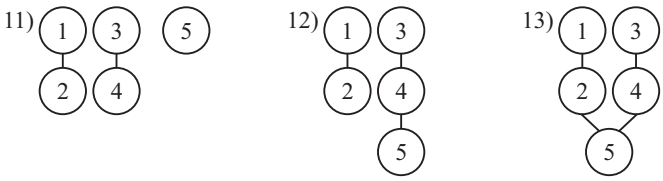


Fig. 5. Inclusion options for Project 5 in combination IV.

$r_c = 150 + 180 + 5 = 335$. The third term of this sum is the synergistic effect, representing the gain from the joint use of R15 and C15. We estimate the shift of project 5 by the largest time of projects 2 and 4, i.e., 15–22 days, or (−2)–5, taking the savings into account. The value (−2) shows that, at best, the execution time of project 5 can be reduced even more compared to the case of neglecting the soft dependencies. In addition, the resource saving is appreciable as well.

II) Among the first four projects, only the dependency (3–4) is considered, giving the savings $t_c = 4$ and $r_c = 15$, and the production of R15 rebar is eventually set up in 12–19 days. In this combination, there are three logical and reasonable options for including project 5:

6. Project 5 is included in parallel without soft dependencies. The features of this situation have been discussed above. The time and resource savings due to the soft dependency between projects 3 and 4 are $t_c = 4$ and $r_c = 15$.

7. Project 5 is included after project 4 (and project 3 as its component) and starts using in-house R15 rebar. With executing this project after project 4, we use the corresponding soft dependency

(4–5), and $t_c = 7$ and $r_c = 180$. As a result, the R15 line is launched in 12–19 days (and will be further used in project 5), the term of project 5 is shifted by 5–12 days, and C15 grade is needed to purchase only for project 5. Note that the longest term for launching the rebar production line does not exceed the shortest term for launching the concrete production line. Therefore, it may be logical for project 5 to use in-house rebar but not in-house concrete; on the other hand, by the time of using in-house concrete, the rebar production line will already be running, and it makes no sense to use in-house concrete without in-house rebar.

8. Project 5 is included after project 4 (and project 3 as its component) and project 2, and uses R15 rebar and in-house C15 concrete. The soft dependencies (2–5), (4–5), and, of course, (3–4) are used. The saving from the joint use of R15 and C15 is also taken into account. Accordingly, $t_c = 8 + 7 + 4 + 2 = 21$ and $r_c = 150 + 180 + 15 + 5 = 350$. The term of project 5 is shifted by (-6)–1 days, i.e., it is very likely to be completed earlier than required. By the above reasoning, it makes no sense to consider project 1 instead of project 2 in such a combination.

III) Among the first four projects, only the dependency (1–2) is taken into account, giving the savings $t_c = 7$ and $r_c = 20$, and the production of C15 concrete is eventually set up in 20–35 days. This combination describes the least probable situation: when considering the dependency (1–2), it makes sense to take the dependency (3–4) into immediate account (see combination IV below) due to no extra costs and a significant gain. Nevertheless, such a situation is theoretically possible under a set of circumstances (e.g., there is enough rebar of the required grade in stock). In this combination, there are two logical and reasonable options for including project 5:

9. Project 5 is included in parallel without soft dependencies. The features of this situation have been discussed above. The time and resource savings due to the soft dependencies between projects 1 and 2 only are $t_c = 7$ and $r_c = 20$.

10. Project 5 is included after project 2 (and project 1 as its component) and project 4, and uses R15 rebar and in-house C15 concrete. The soft dependencies (2–5), (4–5), and, of course, (1–2) are used. The savings from the joint use of R15 and C15 are also taken into account. Accordingly, $t_c = 8 + 7 + 7 + 2 = 24$ and $r_c = 150 + 180 + 20 + 5 = 355$. The term of project 5 is shifted by (-4)–11 days. By the above reasoning, it makes no sense to consider project 1 instead of project 2 in such a combination.

IV) Among the first four projects, both dependencies (1–2) and (3–4) are taken into account, giving the savings $t_c = 7 + 4 = 11$ and $r_c = 20 + 15 = 35$, the productions of C15 concrete and R15 rebar are set up in 20–35 days and 12–19 days, respectively. In this combination, there are three logical and reasonable options for including project 5:

11. Project 5 is included in parallel without considering soft dependencies. The features of this situation have been discussed above. The time and resource savings due to the soft dependencies between projects 1–4 only are $t_c = 7 + 4 = 11$ and $r_c = 20 + 15 = 35$.

12. Project 5 is included after project 4 (and project 3 as its component) and starts using in-house R15 rebar. With executing this project after project 4, we use the corresponding soft dependency (4–5), and $t_c = 7$ and $r_c = 180$. As a result, the R15 line is launched in 12–19 days (and will be further used in project 5), the term of project 5 is shifted by 5–12 days, and C15 grade is needed to purchase only for project 5.

13. Project 5 is included after project 4 (and project 3 as its component) and project 2 (and project 1 as its component) and uses R15 rebar and in-house C15 concrete. Besides the above dependencies (1–2) and (3–4), the soft dependencies (2–5) and (4–5) are fulfilled (actually, all the soft dependencies). The savings from the joint use of R15 and C15 are also considered. Accordingly, $t_c = 7 + 8 + 7 + 4 + 2 = 28$ and $r_c = 20 + 150 + 180 + 15 + 5 = 370$. The term of project 5 is shifted by (-8)–7 days.

We summarize the results in the table below.

Table 2. The interaction of program projects: a comparison of options

Combination	Option	t_c	r_c	The number of soft dependencies	The shift of project 5's term (in days)
I	1	0	0	0	0
	2	6	120	1	7–11
	3	7	180	1	2–5
	4	11	220	2	1–9
	5	17	335	2	(−1)–5
II	6	4	15	1	0
	7	7	120	2	5–12
	8	21	350	3	(−6)–1
III	9	7	20	1	0
	10	24	355	3	(−4)–11
IV	11	11	35	2	0
	12	7	180	3	5–12
	13	28	370	3	(−8)–7

Analysis shows no obvious patterns in Table 2, which is the main conclusion. Moreover, efficiency criteria in this situation can be different, e.g., the saving of resources or the shift of the term of project 5, as the most ambitious part of the program. Despite some correlation between the above criteria (e.g., options 5, 8, 10, and 13 are quite effective in terms of both criteria), we note that in each of the combinations I–IV, the option with the maximum number of dependencies used is most effective. Generally speaking, it is not easy to make an unambiguous choice.

At the beginning of this section, we have emphasized the importance of the probabilistic component, which is neglected in the illustrative example. Also, the term of each project is not two limit points but any point on the interval between them, and each such point has a particular probability (in the simplest case, a linear probability distribution). These facts are not considered in the example as well. In reality, this probability evolves over time in a much more complicated way.

The example is very simple for a real construction program, the more so in the elementary setting: it ignores the stochastic reduction of the project execution time and/or resources according to some temporal distribution. Therefore, dealing with soft dependencies can rarely be reduced to simple schemes without losing the problem essence. This paper takes a step towards considering the probabilistic and uncertain parameters of soft dependencies.

3. THE SET OF REALIZABLE SOFT DEPENDENCIES

Consider the problem of reducing the implementation time (duration) of a multi-project program with soft dependencies using the basic definitions provided in [4, 5].

By assumption, there are many options for including soft dependencies into the program. Each option is characterized by the probability of its realization and the corresponding program duration.

We introduce the following notation:

τ_i is the project execution time when neglecting soft dependencies;

m is the number of projects in the program;

$T = \sum_{i=1}^{m} \tau_i$ is the program duration when neglecting soft dependencies.

An algorithm for including soft dependencies into a program to reduce its duration was described in [4].

In this paper, we optimally select soft dependencies using the probability of soft dependence realization and the program duration reduction as the optimality criteria.

4. THE PROBABILITY OF SOFT DEPENDENCE REALIZATION

Suppose that the program has a set S of n soft dependencies. Let p_i, $i = 1, 2, \ldots, n$, be the probability of realizing the ith soft dependency, and t_i, $i = 1, 2, \ldots, n$, be the program duration reduction under the realization of the ith soft dependency.

According to [10], if all soft dependencies are realized, the program duration will be $T - \tilde{T}$, where

$$\tilde{T} = \sum_{i=1}^{n} t_i.$$

Since the realization of each soft dependency is treated as an independent event, the probability of realizing all soft dependencies will be

$$P = \prod_{i=1}^{n} p_i.$$

Following [10], the efficiency of different options to realize a soft dependency will be estimated by the mean program duration reduction. We denote by M_S the mean program duration reduction under the realization of all soft dependencies. As demonstrated in [10], this value is given by

$$M_S = \sum_{i=1}^{n} t_i \prod_{i=1}^{n} p_i.$$

Consider a situation when all soft dependencies provide the same reduction of the project execution times, but the probabilities of realizing these dependencies differ.

In this case, the mean program duration reduction under the realization of all soft dependencies can be written as

$$M_S = nt \prod_{i \in S} p_i.$$

Without loss of generality, we arrange all soft dependencies in ascending order of the probabilities of realization:

$$p_1 \geqslant p_2 \geqslant \ldots \geqslant p_n. \tag{1}$$

If all soft dependencies provide the same reduction of the project execution times and inequality (1) holds, for k successively realized soft dependencies, the mean is given by

$$M_m = kt \prod_{i=m}^{k+m-1} p_i, \quad m = 1, 2, \ldots, n - k + 1.$$

In view of condition (1), we have

$$M_1 > M_2 > \ldots > M_{n-k+1}.$$

Let

$$p_i = p_1 b^{\,i-1}. \tag{2}$$

In this case,

$$M_1 = ktp_1 b^{\,k(k-1)b/2}.$$

We find the optimal number of realizable soft dependencies by minimizing the mean:

$$\frac{\partial M_1}{\partial k} = tp_1 b^{\,k(k-1)b/2} + ktp_1 \frac{b}{2}(2k-1)\ln b = 0. \tag{3}$$

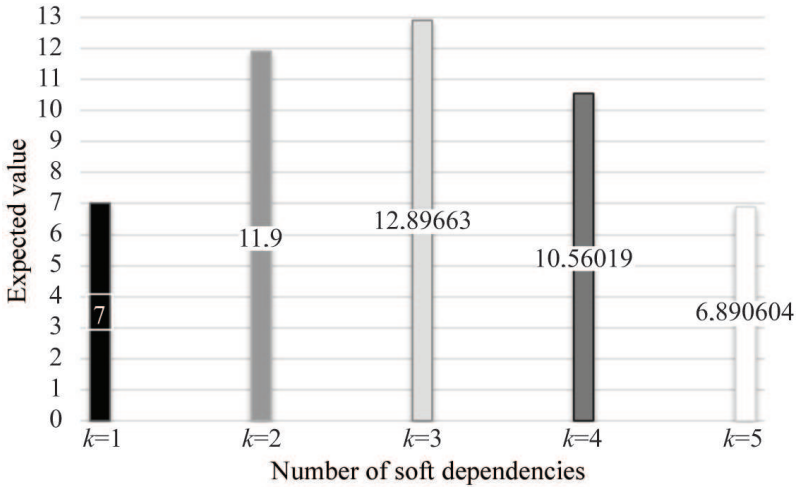


Fig. 6. Change in the expected value versus the probability of soft dependency realization.

From (3) it follows that

$$1 + k\frac{b}{2}(2k-1)\ln b = 0.$$

The solution of this equation has the form

$$k = \frac{1 \pm \sqrt{1 - \frac{16}{b \ln b}}}{4}. \tag{4}$$

Example 1. There are five soft dependencies. The program duration reduction by each soft dependency is $t = 10$, and the probability of realizing the first soft dependency is $p_1 = 0.7$. The probabilities of realizing soft dependencies form the geometric progression (2); let its ratio be $b = 0.85$. Accordingly, the probabilities of realizing the soft dependencies are $p_1 = 0.7$, $p_2 = 0.6$, $p_3 = 0.51$, $p_4 = 0.43$, and $p_5 = 0.37$. Figure 6 shows the graph of the mean program duration reduction as a function of the number of realized soft dependencies.

In Example 1, it suffices to realize only three soft dependencies to maximize the mean program duration reduction. By the way, this fact follows from the expression (4). For the above value of b, the calculations yield $k = 2.95$. Note that increasing the ratio actually increases the number of realized soft dependencies maximizing the mean program duration reduction; accordingly, decreasing the ratio decreases this number.

Consider now the following situation: when realizing the ith soft dependency, the project execution time is reduced to the value

$$t_i = t_1 a^{i-1} \tag{5}$$

(another geometric progression), and each soft dependency is realized with the same probability.

In this case, the mean program duration reduction under the realization of all soft dependencies can be written as

$$M_S = \frac{t_1(a^n - 1)}{a - 1}\, p^n. \tag{6}$$

Assuming that only the first k soft dependencies are realized, the expression (6) becomes

$$M_1 = \frac{t_1(a^k - 1)}{a - 1}\, p^k.$$

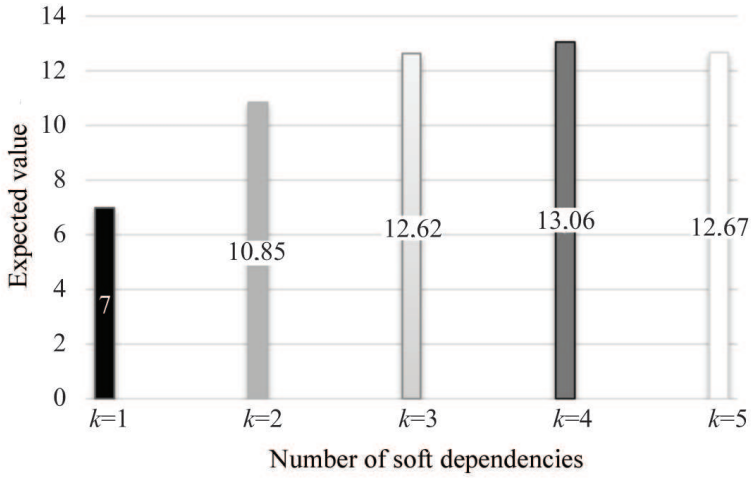


Fig. 7. Change in the expected value versus the program execution time reduction.

We find the number of realizable soft dependencies maximizing the mean program duration reduction:

$$\frac{d}{dk}\left(\frac{t_1(a^k-1)}{a-1}\,p^k\right) = t_1 p^k \frac{a^k \ln a}{a-1} + t_1 \frac{a^k-1}{a-1} p^k \ln p = 0. \tag{7}$$

From (7) it follows that

$$k = \frac{\ln \frac{\ln p}{\ln ap}}{\ln a}. \tag{8}$$

Since $k < n$, we have

$$p < a^{a^n/1-a^n}. \tag{9}$$

Example 2. As in Example 1, there are five soft dependencies. For $a = 1.05$, formula (9) implies $p < 1$, so the probability of realizing each soft dependency can be chosen to be 0.7. The program duration reduction by each soft dependency is described by (5): $t_1 = 10$, $t_2 = 10.7$, $t_3 = 11.45$, $t_4 = 12.25$, and $t_5 = 13.11$. Figure 7 shows the graph of the mean program duration reduction as a function of the number of realized soft dependencies.

In Example 2, it suffices to realize only four soft dependencies to maximize the mean program duration reduction. By the way, this fact follows from the expression (8). For the above value of a, the calculations yield $k = 3.11$. Note that changing the ratio actually changes the number of realized soft dependencies maximizing the mean program duration reduction.

Thus, we have considered two settings: in the first, all soft dependencies provide the same reduction of the project execution time, but the probabilities of realizing these dependencies are different; in the second, the reduction of the project execution time when realizing soft dependencies differs, but the probabilities of realizing soft dependencies are the same.

5. CONCLUSIONS

This paper has analyzed the use of soft dependencies in multi-project program implementation. An example has been provided to demonstrate how rarely dealing with soft dependencies can be reduced to simple schemes without losing the problem essence. The problem of determining the optimal number of realizable soft dependencies maximizing the mean program duration reduction under a stochastic uncertainty in the realization of soft dependencies has been considered. As shown, the maximum value of the mean program duration reduction depends on the number of

realized soft dependencies and, moreover, is largely determined by the probability of realizing each soft dependency. Numerical examples have been given in which the maximum value of the mean program duration reduction is achieved for a smaller number of available soft dependencies.

REFERENCES

1. Barkalov, S.A., Burkov, V.N., Burkova, I.V., Voropaev, V.I., Sekletova, G.I., et al., *Matematicheskie osnovy upravleniya proektami* (Mathematical Foundations of Project Management), Moscow: Vysshaya Shkola, 2005. (In Russian.)
2. Barkalov, S.A., Burkova, I.V., Kolpachev, V.N., and Potapenko, A.M., *Modeli i metody raspredeleniya resursov v upravlenii proektami* (Models and Methods of Resource Allocation in Project Management), Moscow: Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, 2004. (In Russian.)
3. Barkalov, S.A., Kolpachev, V.N., Perelygin, A.L., and Likhotin, Yu.P., Project Management Problem with Soft Dependencies Between the Works, *Scientific Herald of the Voronezh State University of Architecture and Civil Engineering. Constr. and Arch.*, 2004, no. 3, pp. 125–129. (In Russian.)
4. Burkov, V.N., Burkova, I.V., and Shchepkin, A.V., Soft Dependencies between Projects in Program Management, *Proc. 16th Int. Conf. on Management of Large-Scale System Development (MLSD)*, 2023, Moscow, Russian Federation, pp. 1–5. https://doi.org/10.1109/MLSD58227.2023.10304063
5. Burkova, I.V., Cost Minimization in Projects Based on Soft Dependencies, *Proc. All-Russian Scientific and Practical Conference on Automation Systems (in Education, Science and Production) (with international participation)*, 2023, Novokuznetsk, pp. 219–224. (In Russian.)
6. Golenko-Ginzburg, D.I., *Stokhasticheskie setevye modeli planirovaniya i upravleniya razrabotkami* (Stochastic Network Models of R&D Planning and Management), Voronezh: Nauchnaya Kniga, 2010. (In Russian.)
7. Gelrud, Yu.D., The Generalized Stochastic Network Models for Management of Complex Projects, *Vestn. Novosib. Gos. Univ., Ser. Mat. Mekh. Inform.*, 2010, vol. 10, no. 4, pp. 36–51. (In Russian.)
8. Voropaev, V.I. and Gelrud, Yu.D., Generalized Stochastic Network Models for Management of Complex Projects (Parts 1 and 2), *The Project Management Journal*, 2008, no. 1, pp. 18–27; no. 2, pp. 114–125. (In Russian.)
9. Smith, A., Stochastic Methods in Project Management, *Journal of Optimization*, 2021, vol. 12, pp. 45–60.
10. Burkov, V.N., Burkova, I.V., and Shchepkin, A.V., Conditions of Application of Soft Dependencies between Projects in Program Management under Stochastic Uncertainty of Their Implementation, *Proc. Int. Scientific and Practical Conference "Theory of Active Systems – 55 Years" (TAS-55)*, Moscow: Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, 2024, pp. 95–99. (In Russian.)

This paper was recommended for publication by A.A. Galyaev, a member of the Editorial Board

ISSN 0005-1179 (print), ISSN 1608-3032 (online), Automation and Remote Control, 2026, Vol. 87, No. 5, pp. 428–436.

SPECIAL ISSUE

Controlled Random Search and Likelihood Ratio in Boolean Programming Problems

A. I. Ermolaev[*,a], **A. V. Akhmetzyanov**[**,b], **and A. R. Latipov**[**,c]

[*] *National University of Oil and Gas "Gubkin University", Moscow, Russia*
[**] *Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia*
e-mail: [a] *ermolaev.a@gubkin.ru,* [b] *atlaswa@gmail.com,* [c] *latipov257@gmail.com*

Received July 8, 2025
Revised October 30, 2025
Accepted November 20, 2025

Abstract—Based on the sequential probability ratio test (likelihood ratio) method, a controlled random search algorithm is proposed for the approximate solution of large-scale discrete programming problems. The reduction in the exhaustive search of feasible sets of the decision variables of the problem is achieved by introducing non-zero probabilities of false recognition of the optimal solution. As a practical application of the algorithm, the problem of forming optimal well placement patterns in oil and gas reservoirs is considered. The results of computational experiments are presented, the purpose of which was to study the accuracy of the problem's solution depending on its dimension (the number of blocks where well placement is possible and the number of wells to be placed were varied). The optimal solution of the problem obtained by one of the exact methods of discrete programming was used as a reference solution, against which the accuracy of the approximate solution generated by the proposed algorithm was evaluated.

Keywords: algorithm, Boolean programming, probability, statistical hypotheses, maximum likelihood method, probability density, optimization, transportation problem

DOI: 10.7868/S1608303226050045

1. INTRODUCTION

One of the main approaches to solving large-scale discrete programming problems is the use of approximate algorithms that allow estimating the error in determining the exact (optimal) solution [1–3]. This approach is quite justified when solving applied problems, for which a typical situation is not only their large dimensionality but also a significant error in the values of the initial parameters. In this case, the value of the exact solution decreases, and it becomes possible to reduce the number of analyzed feasible solutions of the problem by abandoning the guarantee of obtaining an exact solution.

In accordance with [2], a large-scale problem will be understood as a problem for which the search for an optimal solution on a specific computer takes a time exceeding a specified value, or requires an amount of memory exceeding the allocated volume.

A significant class of approximate methods is composed of controlled random search methods [1, 2]. Unlike "blind" random search, these methods allow accumulating information obtained at previous stages of the search for an optimal solution to correct subsequent stages.

The algorithm proposed in this paper belongs to the controlled random search methods; it represents an iterative procedure, at each step of which, given the probabilities of erroneous inference, the truth of one of two statistical hypotheses is tested: can the feasible solution, which is the most preferable relative to all previously generated feasible solutions in terms of the objective function, be considered optimal, or should the search be continued?

2. PROBLEM STATEMENT AND ALGORITHM DESCRIPTION

Consider the linear Boolean programming problem:

$$F(X) \to \min \tag{1}$$

$$X = (x_1, \dots, x_k) \in A \tag{2}$$

$$x_i \in \{0, 1\}, \quad i = 1, \dots, k, \tag{3}$$

where A is a subset of k-dimensional Euclidean space represented by linear constraints, and $F(X)$ is a linear function defined on the subset A.

For the large-scale problem (1)–(3), solving it would require a computationally prohibitive exhaustive search of all feasible solutions (in the general case, such an exhaustive search is impossible due to the practically unlimited volume of required memory and, accordingly, computation time). Let us introduce the set B:

$$B \equiv \{X = (x_1, \dots, x_k) : X \in A, 0 \le x_i \le 1,\ i = 1, \dots, k\}. \tag{4}$$

Assuming that an exhaustive search of all feasible solutions is impossible due to enormous time costs, let us consider the main steps of the algorithm.

1. Two problems are solved by any available linear programming method:

$$F(X) \to \min_{X \in B}, \tag{5}$$

$$F(X) \to \max_{X \in B}. \tag{6}$$

Let $F_{\min}$ and $F_{\max}$ be the objective function values in the optimal solutions of problems (5) and (6), respectively. Then, taking into account (4), for the values of the objective function (1) under constraints (2) and (3), it will follow that:

$$F_{\min} \le F(X) \le F_{\max}. \tag{7}$$

2. A set of n feasible solutions to problem (1)–(3) is generated randomly (according to some probability distribution): $X_1, \dots, X_n$, and a set $F_1 \equiv F(X_1), \dots, F_n \equiv F(X_n)$ is formed, which represents a realization of a random variable—the value of the objective function of problem (1)–(3).

If for $r, j \in \{1, \dots, n\}$ it turns out that the conditions $F_r \le F_j$ and $F_r = F_{\min}$ are satisfied, then it follows from inequalities (7) that X_r is the optimal solution to problem (1)–(3), and all calculations are terminated. Otherwise, the following testing stages are required, where the case $F_r > F_{\min}$ is considered. If over the next $n+1, n+2, \dots$ "trials" the condition $F_r \le F_j$ is satisfied, where $j \in \{n+1, n+2, \dots\}$, a suspicion arises: is X_r the optimal solution to problem (1)–(3)?

The main goal of the subsequent stages of the proposed modification of the random search method is to answer the question: how many additional trials, in each of which $F_{\min} < F_r \le F_j$ is satisfied, where $j \in \{n+1, n+2, \dots\}$, should be conducted so that, given the probabilities of false inference, one could assert that X_r is the optimal solution to problem (1)? The answer to these questions is achieved as follows.

3. Let us introduce the notations $X^* \equiv X_r$ and $F^* \equiv F(X_r)$. Based on the obtained values of the objective function (1), a histogram is constructed. From the shape of the histogram, one of the possible probability distributions for $F(X)$—the values of the objective function (1)—is selected. This selected probability distribution is applied to the values of the random variable $F(X)$ belonging both to the interval $[F^*, F_{\max}]$ and to the interval $[F_{\min}, F_{\max}]$. In this regard, two hypotheses H_1 and H_2 are put forward:

$$\mathrm{H}_1 : F(X) \in [F^*, F_{\max}]; \tag{8}$$

$$\mathrm{H}_2 : F(X) \in [F_{\min}, F_{\max}]. \tag{9}$$

To test the proposed hypotheses, it is suggested to use the sequential probability ratio test (or likelihood ratio) [4]. One of its advantages is the ability not to specify the required number of trials (measurements, iterations) in advance, since after each trial, either the validity of hypothesis H_1 is established or a conclusion is made about the need to continue the trials. Thus, unnecessary trials are not conducted.

4. The implementation of the sequential probability ratio test (SPRT) when applied to problem (1)–(3) consists of the following steps.

4.1. The probabilities of false recognition are introduced: γ_{lt} is the probability of accepting H_l when H_t is actually true, where $l, t \in \{1, 2\}$. Two thresholds are determined: α and β, $\alpha > \beta$ [4]:

$$\alpha \equiv (1 - \gamma_{21})/\gamma_{12}, \quad \beta \equiv \gamma_{21}/(1 - \gamma_{12}). \tag{10}$$

4.2. Based on the selected type of probability distribution of the random variable $F(X)$ and its realizations $F_1, \ldots, F_n$ obtained during the trials, the vectors of parameters Y_1 and Y_2 of two conditional probability density functions of the objective function values (1) are determined using the maximum likelihood method [5]. The first conditional probability density function $p_{1n}(f/Y_1, F^*)$ corresponds to hypothesis H_1 (condition (8) is valid), i.e., $F^* \le F(X) \le F_{\max}$. The second conditional probability density function $p_{2n}(f/Y_2, F_{\min})$ corresponds to hypothesis H_2 (condition (9) is valid), i.e., $F_{\min} \le F(X) \le F_{\max}$. In other words, two probability density functions $p_{1n}(f/Y_1, F^*)$ and $p_{2n}(f/Y_2, F_{\min})$ are reconstructed, which is reduced to solving two problems:

$$\prod_{j=1}^{n} p_{1n}(F_j/Y_1, F^*) \to \max_{Y_1}, \tag{11}$$

$$\prod_{j=1}^{n} p_{2n}(F_j/Y_2, F_{\min}) \to \max_{Y_2}. \tag{12}$$

After determining Y_1 and Y_2, one can proceed to testing hypotheses H_1 and H_2, i.e., to answering the question: is the random variable $F(X)$ distributed according to the first law $p_{1n}(f/Y_1, F^*)$ or the second law $p_{2n}(f/Y_2, F_{\min})$?

4.3. A new feasible solution X_{n+1} is randomly generated, and F_{n+1}—the value of the objective function (1) corresponding to this solution—is calculated.

4.4. The likelihood ratio δ_{n+1} [4] is introduced:

$$\delta_{n+1} = \frac{p_{1n}(F_{n+1}/Y_1, F^*)}{p_{2n}(F_{n+1}/Y_2, F_{\min})}.$$

4.5. The conditions for stopping or continuing the trials [4] are checked:

a) if $\delta_{n+1} \ge \alpha$, then H_1 is considered true, i.e., the hypothesis is accepted: X^* is the optimal solution to problem (1)–(3);

b) if $\beta < \delta_{n+1} < \alpha$, then it is necessary to generate a new feasible solution X_{n+2}, calculate F_{n+2}, determine δ_{n+2} using the formula

$$\delta_{n+2} = \frac{p_{1n}(F_{n+1}/Y_1, F^*)}{p_{2n}(F_{n+1}/Y_2, F_{\min})} \cdot \frac{p_{1n}(F_{n+2}/Y_1, F^*)}{p_{2n}(F_{n+2}/Y_2, F_{\min})},$$

again compare δ_{n+2} with the thresholds α and β, etc. (with each iteration, the number of factors in the likelihood ratio increases by one);

c) if the condition $\delta_{n+1} \le \beta$ is satisfied (H_2 is true), i.e., $\delta_{n+1} = 0$ because $F_{n+1} < F^*$ and $p_{1n}(F_{n+1}/Y_1, F^*) = 0$, then it is necessary to set $X^* \equiv X_{n+1}$, $F^* \equiv F(X_{n+1})$, find new values of Y_1 and Y_2 by solving problems (11) and (12) where $j = 1, 2, \ldots, n+1$, and repeat steps 4.2–4.5, setting $n \equiv n + 1$.

Let $(m-1)$ be the number of iterations of the recognition process in each of which the condition of item b) was satisfied, and realizations of the random variable $F(X)$ were obtained: $F_{n+1}, F_{n+2}, \ldots, F_{n+m-1}$. Then, at the mth iteration after calculating F_{n+m}, the likelihood ratio takes the form:

$$\delta_{n+m} = \prod_{j=n+1}^{n+m} \frac{p_{1n}(F_j/Y_1, F^*)}{p_{2n}(F_j/Y_2, F_{\min})}. \tag{13}$$

3. APPLICATION OF THE ALGORITHM TO FORM OPTIMAL WELL PLACEMENT PATTERNS IN OIL AND GAS RESERVOIRS

Consider the application of the algorithm to solve the problem of placing a given number of wells in oil and gas reservoirs, which is a key task in the list of problems for designing the development of oil and gas fields. Numerous studies are devoted to its solution (see, for example, reviews [6, 7]), most of which use placement models where the objective functions represent final performance indicators of the reservoir development processes, such as the hydrocarbon recovery factor from the subsurface. This necessitates reservoir simulation of oil and gas field development processes [8], the implementation of which is associated with significant time costs.

Consider one of the alternative approaches to designing gas (oil) well placement patterns, in which the well placement problem is reduced to a linear Boolean programming model [9]. In this model, the objective function is a formalization of heuristic rules for rational well placement, verified by years of practice in oil and gas field development. In this case, calculating the objective function values will not require involving reservoir simulation. According to these rules, the placement of a given number of wells (more precisely, bottom-holes) should ensure [9]:

a) the smallest possible distance from the wells to any point of the productive formation;

b) equality of the drainage areas of the wells;

c) proximity of the wells to the reservoir areas with higher reserve values.

Fulfilling these rules aims to achieve the maximum possible reservoir coverage by a given number of wells and to increase the hydrocarbon recovery factor from the reservoir.

The substantive formulation of the problem is as follows: let the reservoir be divided into blocks, in each of which a bottom-hole can be placed; the number of wells to be placed is given; it is required to determine the set of blocks containing bottom-holes for which the heuristic rules of rational well placement (rules *a, b, c*) are violated to a minimum extent.

Let us proceed to constructing the optimization model, focusing on the results of paper [9]. Preliminarily, the productive area is approximated by a two-dimensional domain consisting of equal-area square blocks. The gas (oil) reserves of each block are known. It is assumed that when placing a well in any block, the coordinates of the bottom-hole coincide with the center of this block.

Let us introduce the initial parameters. Let S be the number of production wells, K be the number of blocks per well, N be the total number of blocks, $N = KS$. In order for the problem to admit non-trivial solutions, we set $N > S \geq 1$, i.e., $K > 1$.

Let V_j be the geological reserves of the jth block, $V_j > 0$, and $V \equiv \max\{V_j\}, j = 1, \ldots, N$. We introduce the parameter λ_j, characterizing the productivity ("importance," "usefulness") of the jth block or the potential efficiency of a well placed in this block. Then the ratio $\lambda_j = V_j/V$ can be chosen as λ_j [9]. Let us introduce the parameter R_{ij}—the distance between the centers of the ith and jth blocks, $R_{ij} > 0$, $j \neq i$, $R_{ii} = 0$, $R \equiv \max\{R_{ij}\}$, $i = 1, \ldots, N$, $j = 1, \ldots, N$. Let $r_{ij} = R_{ij}/R$.

Let us define the parameter c_{ij}—the "weighted distance" between the ith and jth blocks:

$$c_{ij} = \begin{cases} (\lambda_j)^{1-\gamma}(r_{ij})^{\gamma}, & i \neq j, \\ 0, & i = j, \end{cases} \tag{14}$$

where γ is an expert assessment of the importance of the "distance" indicator relative to the "reserves" indicator, $0 \leq \gamma \leq 1$. The choice of γ for oil and gas wells may vary. Taking into account the higher mobility of gas compared to oil, γ for gas reservoirs will have lower values. The parameter c_{ij} can be interpreted as a penalty for violating rules *a, b, c* when including the jth block in the area of influence of a well located in the ith block. The area of influence is understood as the set of blocks around the well from which the well receives the main influx of formation fluids.

To increase the adequacy of well placement models, instead of the Euclidean distance expressed in units of length, one can use parameters characterizing the resistance to the flow of formation fluids from one block to another as the distance between any pair of blocks (a detailed description of such a substitution can be found in [10]).

Let us introduce the decision variables x_{ij}: $x_{ij} = 1$ if the jth block is included in the area of influence (drainage area) of a well located in the ith block, and $x_{ij} = 0$ otherwise. It follows from the definition of x_{ij} that if $x_{ii} = 1$, then there is a well in the ith block, otherwise $x_{ii} = 0$.

Taking into account the above-formulated concept of rationality (rules *a, b, c*), the formation of the best well placement pattern is reduced to finding such decision variables x_{ij} that

$$\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}x_{ij} \to \min_{x}, \tag{15}$$

$$\sum_{i=1}^{N} x_{ii} = S, \tag{16}$$

$$\sum_{j=1}^{N} x_{ij} = Kx_{ii}, \quad i = 1, \ldots, N, \tag{17}$$

$$\sum_{i=1}^{N} x_{ij} = 1, \quad j = 1, \ldots, N, \tag{18}$$

$$x_{ij} \in \{0, 1\}, \quad i = 1, \ldots, N, \quad j = 1, \ldots, N. \tag{19}$$

In model (15)–(19): the objective function in criterion (15) is the total penalty for violating rules *a, b, c*; constraint (16) is the limit on the number of wells; constraint (17) is the condition that each area of influence consists of an equal number of blocks; constraint (18) is the condition that any block can be included in only one area of influence. Since it follows from (14) that $c_{ij} = 0$ for $j = i$, the variables x_{ii} for $j = i$ are absent in the objective function (15). Considering (14) and $V_j > 0$, $R_{ij} > 0$, $j \neq i$, $i = 1, \ldots, N$, $j = 1, \ldots, N$, the objective function (15) will be strictly greater than zero for any feasible solution.

The algorithm considered in the previous section utilizes the specificity of problem (15)–(19), which lies in the following. First, the system of equations (16)–(19) is linearly dependent due to the condition $N = KS$. Second, with a certain fixed well placement, i.e., with known feasible values of the variables x_{ii}, problem (15), (17)–(19) becomes a classical balanced transportation problem with a cost objective (T-problem) [11]. We will demonstrate the transformation of problem (15), (17)–(19) into a T-problem for one of the feasible well placements, according to which wells are placed in the first S blocks:

$$x_{ii} = 1, \quad i = 1, \ldots, S, \quad x_{ii} = 0, \quad i = S + 1, \ldots, N. \tag{20}$$

Taking into account relations (17), (18), (20), problem (15)–(19) takes the form:

$$\sum_{i=1}^{S}\sum_{j=S+1}^{N} c_{ij}x_{ij} \to \min_{x}, \tag{21}$$

$$\sum_{j=S+1}^{N} x_{ij} = K-1, \quad i=1,\ldots,S, \tag{22}$$

$$\sum_{i=1}^{S} x_{ij} = 1, \quad j=S+1,\ldots,N, \tag{23}$$

$$x_{ij} \geq 0, \quad i=1,\ldots,S, \quad j=S+1,\ldots,N. \tag{24}$$

In problem (21)–(24), considering that $N = KS$, the sum of the right-hand sides of equations (22) and the sum of the right-hand sides of equations (23) coincide and equal $(N-S)$. Due to constraint (23) and conditions (24), the values of the decision variables cannot be greater than 1. Since the right-hand sides of constraints (22) and (23) are strictly positive integers and condition (24) is introduced, condition (19) will be satisfied "automatically" [11]. Therefore, when condition (20) is satisfied, the optimal values of x_{ij} (where $i=1,\ldots,S$, $j=S+1,\ldots,N, j\neq i$) from the point of view of criterion (21) can be obtained using linear programming algorithms, which are more efficient compared to discrete programming algorithms when solving large-scale problems. The remaining decision variables of the original problem (15)–(19) are set to zero to satisfy constraints (16)–(19). Thus, a value of the objective function (15) will be obtained that is equal to the value of the objective function (21) in the optimal solution of problem (21)–(24).

By assigning values equal to one to some S elements of the main diagonal of the square matrix $\{x_{ij}\}_{N\times N}$ using a random number generator and setting the remaining elements of the main diagonal to zero (to satisfy constraint (16)), it is possible to generate optimal solutions to problem (21)–(24), which are feasible solutions to problem (15)–(19). Each feasible solution corresponds to a value of the objective function (15), which makes it possible to apply the proposed controlled random search algorithm to solve the original problem (15)–(19).

Below are the results of testing the proposed algorithm with various sets of initial data for problem (15)–(19). Natural and technological parameters of the development of a hypothetical gas field were used as initial data. Computational experiments were conducted by varying the key parameters of the problem: the total number of blocks (N), the number of placed wells (S), and γ—the parameter of the objective function (15), $0 \leq \gamma \leq 1$. The parameter γ determines the relative "weights" of rules a and c, namely: at $\gamma = 0$, only the block reserves are taken into account (complete dominance of rule c); at $\gamma = 1$, only the distances between wells are considered (complete dominance of rule a). Note that at $\gamma = 0$, the solution of problem (15)–(19) is trivial: wells are placed in the S blocks that exceed the other $(N-S)$ blocks in terms of reserves.

The proposed algorithm (PA) was compared with the Branch and Bound Method (BBM) [11], which allows obtaining an exact solution to problem (15)–(19). A computation time limit of 1200 seconds (20 minutes) was set for the BBM. The analysis is based on a comparison of the obtained objective function values (15): F^* corresponds to the best solution found by PA; F_{opt} to the optimal solution found by BBM (if available); $F_{\min}$ is the theoretical lower bound of the objective function (15), when problem (15)–(18), (24) is solved instead of problem (15)–(19). In addition, the time spent obtaining the final solutions was compared: T^* for PA, T_{opt} for BBM. The comparison results are presented in Table. Besides the parameters mentioned above, Table shows the error of the best (approximate) solution obtained using PA relative to the exact solution obtained using BBM: $\varepsilon = (F^* - F_{\text{opt}})/F_{\text{opt}}, \%$. The symbol ">" means that the allowed solution time was exceeded. In the PA implementation, the limit on the number of iterations was 5000.

Comparison of the approximate algorithm (PA) and the exact algorithm (BBM)

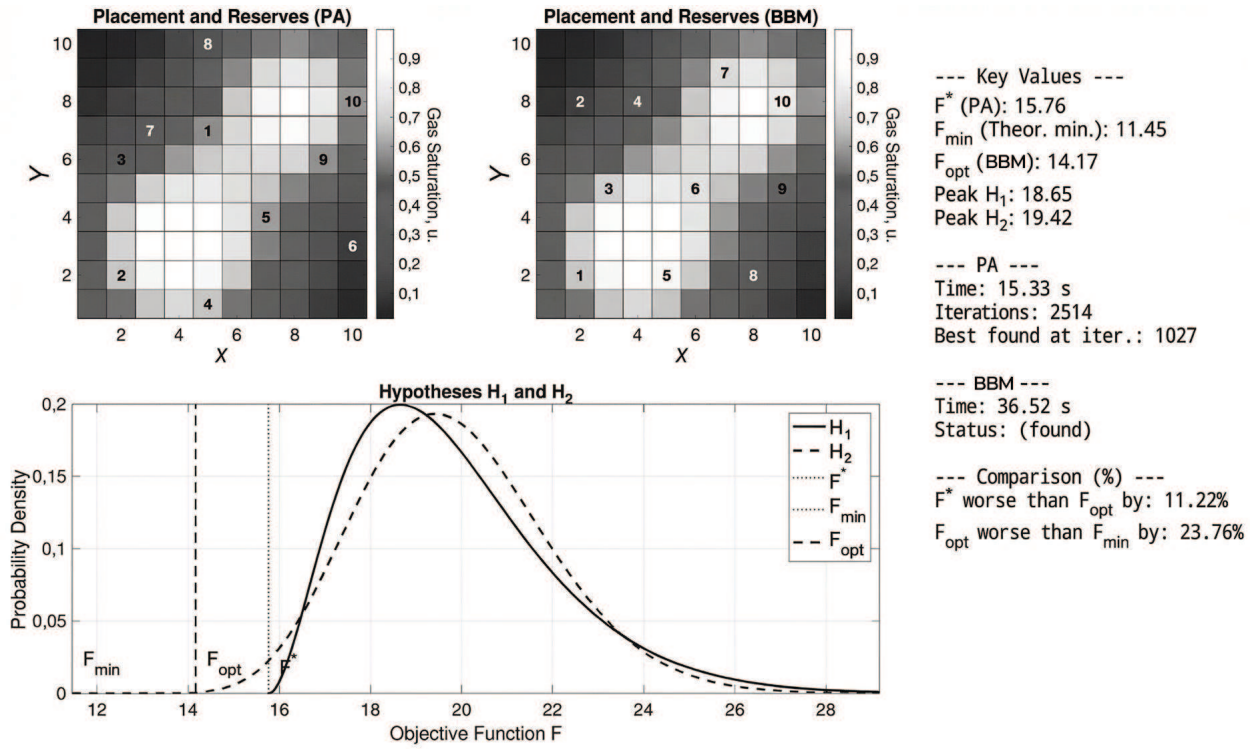
N	S	γ	F^*	F_{opt}	$F_{\min}$	T^*, s	T_{opt}, s	Number of PA iterations	ε, %
16	4	0.0	11.06	11.06	11.06	13.63	0.02	5000	0.00%
16	4	0.3	7.45	7.45	7.34	9.36	0.03	3562	0.00%
16	4	0.7	4.28	4.28	4.26	13.19	0.02	5000	0.00%
16	4	1.0	2.83	2.83	2.83	13.08	0.03	5000	0.00%
100	5	0.0	45.94	45.45	45.45	26.71	0.13	5000	1.08%
100	5	0.3	30.36	29.93	25.43	15.05	5.62	3203	1.44%
100	5	0.7	19.61	18.65	12.26	14.43	28.09	3102	5.15%
100	5	1.0	14.65	13.95	7.46	11.70	28.31	2743	5.02%
100	10	0.0	42.56	41.10	41.10	32.74	0.16	5000	3.55%
100	10	0.3	26.94	26.01	23.32	19.56	14.67	3151	3.58%
100	10	0.7	15.76	14.17	11.45	15.33	36.52	2514	11.22%
100	10	1.0	10.78	9.53	7.07	15.31	51.19	2728	13.12%
100	25	0.0	33.32	29.46	29.46	49.85	0.20	5000	13.10%
100	25	0.3	20.70	19.50	17.46	26.94	2.88	2943	6.15%
100	25	0.7	10.80	9.86	9.11	29.30	147.14	3217	9.53%
100	25	1.0	7.03	>	5.89	36.29	> 1200	3608	>
400	10	0.0	184.80	182.74	182.74	218.29	13.15	5000	1.13%
400	10	0.3	110.43	>	81.97	95.08	> 1200	2792	>
400	10	0.7	60.47	>	29.53	91.40	> 1200	2852	>
400	10	1.0	41.83	>	14.51	85.58	> 1200	2712	>
400	40	0.0	168.14	156.33	156.33	462.45	10.43	5000	7.55%
400	40	0.3	88.06	>	71.75	337.64	> 1200	3435	>
400	40	0.7	39.15	>	26.65	285.09	> 1200	3279	>
400	40	1.0	22.66	>	13.40	247.32	> 1200	2749	>
400	100	0.0	137.21	110.68	110.68	959.47	10.12	5000	23.97%
400	100	0.3	67.65	>	53.29	592.15	> 1200	2835	>
400	100	0.7	26.91	>	21.13	639.56	> 1200	3273	>
400	100	1.0	14.14	>	11.16	577.67	> 1200	2921	>

From the results presented in Table, the following conclusions can be drawn.

1. Impact of problem dimensionality on computation time. For the problem with $N = 16$, both methods find the solution almost instantly (BBM is faster). Already at $N = 100$, T_{opt}—the computation time for BBM—increases noticeably and becomes comparable to T^*—the computation time for PA (or $T_{\text{opt}} > T^*$), for example, at $N = 100$, $S = 5$, $\gamma = 0.70$: $T_{\text{opt}} = 28$ s, $T^* = 14$ s; $N = 100$, $S = 10$, $\gamma = 1.00$: $T_{\text{opt}} = 51$ s, $T^* = 15$ s; $N = 100$, $S = 25$, $\gamma = 0.70$: $T_{\text{opt}} = 147$ s, $T^* = 29$ s. At $N = 400$, the BBM consistently fails to meet the 1200-second limit. In contrast, the PA completed its work significantly faster than this time threshold in all experiments without exception, finding a solution in a time ranging from one and a half minutes to ∼16 minutes (90–960 s), which demonstrates its applicability to large-scale problems.

2. Deviation from the exact solution. For small N ($N = 16$), the PA finds the exact solution ($F^* = F_{\text{opt}}$). For $N = 100$, the PA finds solutions that are only slightly inferior to the exact ones. The deviation of F^* from F_{opt} varies within reasonable limits (1–13%), as it does not exceed the error in the initial data for real production facilities. However, for extreme values of γ or a large number of wells, the deviation can reach 24%.

3. Influence of the parameter γ. The extreme values $\gamma = 0$ and $\gamma = 1$ simplify the problem for BBM, which finds the solution quickly (for $N \le 100$). In these cases, the PA shows poorer relative performance: it often reaches the iteration limit, and the error of the approximate solution may be higher than at intermediate γ values (0.3; 0.7). At $\gamma = 0$ or $\gamma = 1$, solutions with similar quality



Graphs of the probability density functions of the objective function values (15) for hypotheses H_1 and H_2 at $N = 100$, $S = 10$, $\gamma = 0.7$.

can arise, which hinders the operation of the PA. As noted above, at $\gamma = 0$ the solution is trivial: wells are concentrated in zones with maximum reserves; at $\gamma = 1$, wells are distributed as evenly as possible. Intermediate γ values (0.3; 0.7) create a more complex objective function landscape, slowing down BBM, but allowing the PA to demonstrate its speed advantages while maintaining high solution quality.

4. Estimation of the number of iterations required to obtain the final solution using PA. The PA stops either when the iteration limit (5000) is reached or when hypothesis H_1 is accepted ($\delta_{n+1} \geq \alpha$). The termination of the algorithm in many cases by the condition $\delta_{n+1} \geq \alpha$ (hypothesis H_1 is true) confirms its operability.

The figure shows the probability density functions of the random variable—the objective function (15) at $N = 100$, $S = 10$, $\gamma = 0.7$, corresponding to hypotheses H_1 (solid curve) and H_2 (dashed curve). Based on the histogram constructed using the realizations of this random variable, the β-distribution was chosen as the probability distribution.

4. CONCLUSION

It is advisable to use the proposed algorithm in integer programming problems where the process of generating feasible solutions is not computationally demanding. An example of such a problem is the model of optimal well placement in oil and gas reservoirs considered in this article.

The formation of oil and gas well placement patterns belongs to the main tasks of designing hydrocarbon field development processes. The application of the algorithm to solve this problem allows reducing the time for generating and selecting acceptable field development scenarios, from which the final option to be implemented is selected. It is important to note that the positive qualities of the algorithm manifest themselves specifically when solving large-scale problems, which is typical when designing the development of real oil and gas production facilities. The significant uncertainty in the values of the initial parameters perfectly justifies the use of approximate opti-

mization methods. The algorithm consistently finds near-optimal solutions, spending significantly less time on this than exact methods. This difference in performance is especially noticeable on large-scale problems, where the Branch and Bound Method regularly reached the 1200-second limit, while the proposed algorithm always remained far from this threshold.

The results of computational experiments showed the operability of the proposed controlled random search algorithm, which is based on the likelihood ratio implemented in the form of a sequential probability ratio test.

The reduction in the exhaustive search of feasible solutions is achieved not only by introducing non-zero probabilities of false inference but also by using all the information contained in the probability distributions of the random variable, whose realizations are the values of the optimization problem's objective function.

Since any linear discrete programming problem in which each decision variable takes integer values from some finite set, e.g., $\{0, 1, 2, \ldots, l\}$, where $l < \infty$, can be reduced to an equivalent linear Boolean programming model, the proposed controlled random search algorithm can be extended to solve discrete optimization problems of a more general form.

FUNDING

This work was supported by the Russian Science Foundation, project no. 25-71-20008.

REFERENCES

1. Finkelstein, Yu.Yu., *Priblizhennye metody i prikladnye zadachi diskretnogo programmirovaniya* (Approximate Methods and Applied Problems of Discrete Programming), Moscow: Nauka, 1976.
2. Sigal, I.Kh. and Ivanova, A.P., *Vvedenie v prikladnoe diskretnoe programmirovanie: modeli i vychislitel'nye algoritmy* (Introduction to Applied Discrete Programming: Models and Computational Algorithms), Moscow: Fizmatlit, 2003.
3. Khachaturov, V.R., Veselovskii, V.E., Zlotov, A.V., et al., *Kombinatornye metody i algoritmy resheniya zadach diskretnoi optimizatsii bol'shoi razmernosti* (Combinatorial Methods and Algorithms for Solving Large-Scale Discrete Optimization Problems), Moscow: Nauka, 2000.
4. Fu, K.S., *Sequential Methods in Pattern Recognition and Machine Learning*, Academic Press, 1968. Translated under the title *Posledovatel'nye metody v raspoznavanii obrazov i obuchenii mashin*, Moscow: Nauka, 1971.
5. Cramér, H., *Mathematical Methods of Statistics*, Princeton University Press, 1946. Translated under the title *Matematicheskie metody statistiki*, Moscow: Mir, 1975.
6. AlQahtani, G.D., Vadapalli, R., Siddiqui, S., and Bhattacharya, S., Well Optimization Strategies in Conventional Reservoirs. *SPE 160861*, 2012.
7. Latipov, A.R., Analysis of Models and Methods for Optimal Well Placement in Oil and Gas Reservoirs, *Avtom. Inform. TEK*, 2025, no. 6 (623), pp. 50–57.
8. Kanevskaya, R.D., *Matematicheskoe modelirovanie gidrodinamicheskikh protsessov razrabotki mestorozhdenii uglevodorodov* (Mathematical Modeling of Hydrodynamic Processes of Hydrocarbon Field Development), Moscow-Izhevsk: Institut komp'yuternykh issledovanii, 2003.
9. Ermolaev, A.I. and Ibragimov, I.I., Modeli ratsional'nogo razmeshcheniya skvazhin pri razrabotke gazovykh i gazokondensatnykh mestorozhdenii (Models of Rational Well Placement in the Development of Gas and Gas Condensate Fields), *Trudy Instituta problem upravleniya im. V.A. Trapeznikova RAN*, 2006, vol. XXVII, pp. 118–123.
10. Ermolaev, A.I. and Kuvichko, A.M., HPC-Based Optimal Well Placement, *Proc. of the EAGE Conference (ECMOR XIII)*, Biarritz, France, 10–13 September 2012.
11. Korbut, A.A. and Finkelstein, Yu.Yu., *Diskretnoe programmirovanie* (Discrete Programming), Moscow: Nauka, 1969.

This paper was recommended for publication by A.A. Galyaev, a member of the Editorial Board

ISSN 0005-1179 (print), ISSN 1608-3032 (online), Automation and Remote Control, 2026, Vol. 87, No. 5, pp. 437–464.

REVIEWS

Biology and Phenomenology of Temporal Encoding: A Review of Studies and Models

L. Yu. Zhilyakova

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
e-mail: zhilyakova@ipu.ru

Received June 16, 2025
Revised November 17, 2025
Accepted December 10, 2025

Abstract—Encoding of temporal information (also termed time or temporal encoding in the literature) is a key function of biological systems underlying perception, learning, decision-making, and behavior synchronization. Despite a large amount of empirical research data and many theoretical models, a unified concept of temporal encoding has not yet been formulated. This paper overviews research works devoted to time encoding in living organisms. It examines current views on the neural correlates of time encoding. The evolution of approaches and models is traced from classical scalar timing models to more complex network, population, and Bayesian concepts. The main trends and paradigm shifts in modeling time memorization and prediction are highlighted. The key properties of time encoding observed in most vertebrate species are indicated. The first fundamental phenomenological models—the *internal clock model* and the *Scalar Expectancy Theory* (SET)—are described in detail. Their significance for control theory, artificial intelligence, and robotics is substantiated.

Keywords: temporal encoding, memorization of time intervals, rhythmic activity, internal clock, timing models

DOI: 10.7868/S1608303226050055

1. INTRODUCTION

The problem of temporal encoding is fundamental in neuroscience, psychology, and mathematical modeling of behavior. For control theoreticians, the problem of encoding time intervals is primarily related to robotics (in general) and biologically inspired technologies (in particular). Time is a universal parameter that determines processes such as motor planning, learning, synchronization, rhythm and speech perception, and event prediction. The precise neural mechanisms used by biological systems to represent, memorize, and reproduce time intervals are still the subject of intensive research. There exists no unified theory due to the multiplicity of time scales, differences between sensory and motor tasks, and limitations of experimental methods.

Numerous studies aim to understand the brain's operation of temporal information, the degree of (local or global) distribution of time perception and memorization mechanisms, and investigate the properties of neural network dynamics involved in these mechanisms. This review is mainly intended to systematize approaches to modeling the memorization and reproduction of time intervals. We trace the evolution of ideas regarding the perception and memorization of time by living organisms and, accordingly, the evolution of models for encoding time intervals. The existing approaches are classified, key conceptual directions are identified, and current trends in the development of theoretical models are revealed. Special attention is paid to the first internal clock models that emerged in the 1960s and became firmly established as fundamental concepts of timing, guiding further research for several decades. Since then, experimental scientists have obtained a huge

amount of new data, leading to several hypotheses and more complex models. The internal clock model and the Scalar Expectancy Theory (SET) have been much criticized. However, they remain the most influential theories of memorization and reproduction of time intervals and deserve close attention, even despite the variety of new approaches and methods; see below.

The practical significance of internal clock models is due to their application in neurointerfaces, robotics, adaptive control systems, and neuropsychological diagnostic tools. Particularly in robotics, temporal encoding is necessary for organizing sequential actions, synchronizing with external events, processing rhythmic signals, and predicting operation durations. Biologically inspired models of time can lead to more flexible and adaptive behavior of artificial agents in dynamically changing environments.

Some research into temporal encoding up to 2020 was reviewed in [1]. The paper described the main internal clock models used by the authors to search for correlates in brain activity patterns. The local and global encoding of duration was considered (at the levels of individual cells and interaction between brain areas, respectively). The degree of matching between experimental data and the models was analyzed; according to the authors' conclusion, the matching is still insufficient. The above review will be repeatedly mentioned below. From a psychological perspective, cognitive theories of time perception were surveyed in [2]. Unfortunately, this is one of the few Russian-language papers devoted to temporal encoding. Note also that it contains no references to relevant studies by Russian scholars. A more thorough search has revealed one more Russian-language publication on a similar topic, devoted to the psychophysiological assessment of the sense of time in football referees [3]. This lacuna in research makes our review even more topical, as it will help the Russian scientific community to form an idea of this fairly extensive field of knowledge and models.

The remainder of this paper is organized as follows. Section 2 is devoted to a brief description of neurobiological research in the field of temporal encoding. We present a small part of experimental works related to the search for correlates of temporal encoding in the brain and in nervous systems in general [4–24] and show the complexity and degree of distribution of the system responsible for temporal encoding. Section 3 discusses in detail the two early phenomenological models, namely, the *internal clock model* by Treisman [25–28] and the *Scalar Expectancy Theory* by Gibbon and Church [29–37]; in addition, the methodological principles for constructing models of temporal encoding are outlined, and the relevance of scalar models at the present time is justified. In Section 4, we continue the historical review, providing the further periodization of research associated with the emergence of new technologies and new discoveries in the neurobiology of time. The transition from local mechanistic models to dynamic network models and distributed systems [38–46] is traced. For current trends, the main directions of development are identified and briefly described as follows: the discovery of time cells in the hippocampus [7–9], the transition to population models [47–49], Bayesian approaches to time perception [50–53], the models linking timing with decision processes [54, 55], and some studies of the influence of neurochemistry on the encoding and memorization of time intervals [56–61]. The presentation in Section 4 is rather abstract: this paper mainly aims to provide a broad, albeit inevitably less detailed, overview of modern research and timing models, as well as to show that the latest data confirm the key properties of time perception in animals and humans formulated in the first phenomenological models. In the Conclusions, we summarize the review's outcomes and discuss the advantages of phenomenological models as applied to control theory, including some ways to improve them.

2. SEARCH FOR THE NEUROBIOLOGICAL CORRELATES OF TIME

In neuroscience, as in other experimental disciplines, the validity and accuracy of models directly depend on the quality of empirical data and the correctness of their interpretation. In contrast to

the exact sciences, biological systems are characterized by a high degree of variability and multifactorial causality of phenomena observed. Experimental data in neuroscience inevitably contain a significant noise component due to methodological constraints and the complexity and heterogeneity of biological objects. Moreover, the same effects can be obtained by activating different structures, and the study of distributed mechanisms is accompanied by the danger of data incompleteness. The multiplicity of possible interpretations and the probability of erroneous conclusions create fundamental difficulties for testing hypotheses, constructing theories, and building models.

This section outlines directions of neurobiological research, not so much to immerse the reader in this field deeply, but rather to "map" time processing in the brain schematically and illustrate the challenges faced by scholars when modeling timing and related processes.

The neurobiological correlates of temporal encoding are found in different parts of the vertebrate brain and the nervous system as a whole in simpler organisms. Temporal encoding is studied in different species of living organisms, from crickets [4] to humans [5]. Localizing and detecting brain areas involved in time processing in vertebrates is a complex task partially solved by many researchers, but a complete picture has not yet been obtained. The mechanisms of *sensorimotor control* in humans and non-human primates and their connection with time processing were examined in [6]. As emphasized therein, neural dynamics in sensorimotor areas are involved in timing based on rhythms generated. Encoding time and temporal sequences of events in the *hippocampus* was investigated in [7–11]. As claimed in [12, 13], neurons in the *entorhinal cortex* are responsible for accurate time perception. The neurons responsible for different stages of timing behavior in the *prefrontal* and *anterior cingulate cortex* were identified in [14]. The studies and modeling of temporal encoding by *cerebellar* neurons were presented in [5, 15–20]. Based on experimental functional magnetic resonance imaging (fMRI) data, a model of time processing in the human *visual cortex* was constructed in [21]. Even this list of references makes it evident how complex the processing of durations and sequences of time intervals by nervous systems is. As a result, research is shifting toward *distributed systems* [22, 23]. In opposition to the above studies, Robbe places the function of time perception in humans outside the brain [24].

However, one should keep in mind that not only vertebrates but also simpler species have the ability to estimate time. Referring to [42, 56], the authors of [1] suggested that ancient evolutionary mechanisms are responsible for timing. Most species, from insects to primates, process temporal information as if they were using a stopwatch; this indicates the existence of an internal clock function preserved throughout evolution.

Nevertheless, despite the cross-species similarity of time processing and evolutionary continuity, researchers currently have no doubt that temporal encoding in the mammalian brain is extremely complex. It is distributed across different brain areas that generate coordinated activity patterns. Different durations may be handled by different brain areas, and the areas responsible for perception, planning, and decision-making are closely interconnected.

3. THE PRECLINICAL PERIOD OF TIME PERCEPTION RESEARCH. PHENOMENOLOGICAL MODELS

The neural correlates of timing is a young research area associated with the emergence of technical capabilities for the precise detection of neural activity. However, the study and modeling of temporal encoding have been ongoing for over 60 years. The chronology of models and methods can be conditionally divided into several periods with a common theme. Of course, the boundaries of these periods are blurred and have many intersections, but certain milestones can be identified to mark new research directions. The main interest for our review is the earliest, so-called *preclinical period*, when the main properties of subjective time estimation in humans and animals were discovered. They continue to find new confirmations in the modern era.

The preclinical period began with Treisman's paper [25], published in 1963. His internal clock model was generalized to the *Scalar Expectancy Theory* by Gibbon and Church. The two models transferred the study of time perception from a descriptive field into an exact science with predictive power. They showed that even such a subjective phenomenon as the sense of time obeys strict mathematical laws and can be modeled quantitatively. With all their simplicity, the models reproduce the basic properties of perception and memorization of time intervals inherent in both animals and humans.

3.1. Treisman's Internal Clock Model

The term "internal clock" was introduced, and the model itself was proposed, by Treisman in the two-part paper [25]. The first part described seven series of experiments with humans on memorizing intervals of various durations. The experimental design varied from series to series. The second part was devoted to the description of the model, developed in an attempt to explain and connect the psychophysical data.

The experiments revealed fundamental and stable patterns of human time memorization errors. First, humans demonstrate a predictable bias in duration estimation: short intervals are systematically overestimated, and long intervals are underestimated (the so-called Vierordt's law, see the discussion below). Second, the variance of estimates increases with increasing duration (Weber's law). Third, as the series of trials continues, all time estimates and reproductions have a bias to longer durations (the lengthening effect), and this bias becomes more pronounced for shorter original intervals. These errors manifest regardless of the method (reproduction, production, comparison, or verbal estimation), indicating deeply ingrained mechanisms of internal timing. Additional experimental conditions (changes in stimulus intensity, motor response format) can enhance or modify these patterns, but do not eliminate them.

The author's goal was to construct a minimal model, i.e., to find a minimum set of components and links between them necessary for the perception, memorization, and estimation of time intervals. The resulting mechanism must be capable of partial error correction: a deviation or limitation at one point in the system must be compensated by adaptation in others, ensuring overall stability. In addition, it must reproduce the systematic errors characteristic of human perception. Considering the fundamental limitations and features underlying typical human errors in time memorization and estimation allows constructing practical models both for explaining experiments and for engineering applications.

The internal clock model includes the following main components and operating principles:

- a *pacemaker*, a generator that produces a series of equal-duration pulses propagating along a certain path;
- a *counter*, an accumulator that sums the number of pulses in a given time interval and transfers this value to the store;
- a *store*, a memory block to write or retrieve the counting results for further comparison;
- a *comparator*, a decision mechanism that retrieves data from the store and compares them with the current counter results to determine the counting stop moment or to select an appropriate response;
- a *verbal selective mechanism*, a block that facilitates the retrieval of information from the store by referring to symbolic labels (e.g., "1 second," "2 minutes");
- a *specific arousal center*, a regulator that acts on the pacemaker to change its frequency, thereby regulating pulse generation.

In subsequent reviews, this model is often reduced to a four-component one (*pacemaker—counter—store—comparator*) and sometimes even to a three-component one (*pacemaker—counter—store*); for example, see [1]. The last two blocks are usually eliminated; in the three-

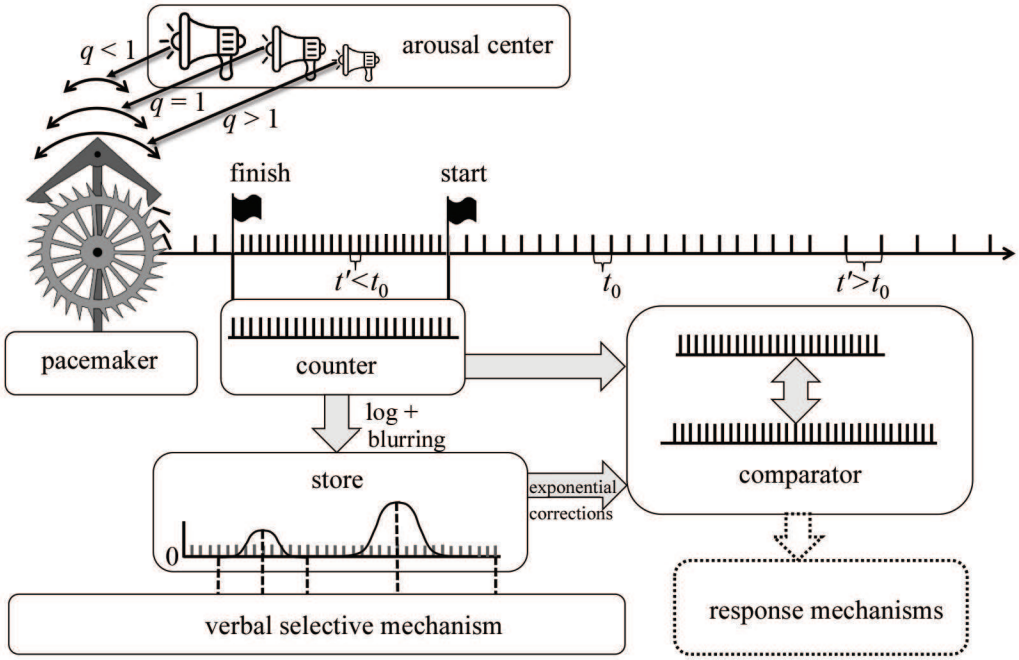


Fig. 1. Treisman's internal clock model. The figure reflects the model scheme and its textual description in [25].

component model, the comparison mechanism is transferred from the structural part to the functional one. The removal of the verbal mechanism from subsequent schemes is natural: the internal clock model turned out to be applicable not only to humans but also to many animal species. The frequency modulator is apparently considered secondary to the basic blocks of the model. However, in this section, the model is described in its original form; see the structure in Fig. 1.

The model's operation is described by eight postulates.

1. The *pacemaker* generates sequential pulses moving along a certain path (Fig. 1) at a constant speed. The basic interval between sequential pulses is t_0 with variance σ_t^2. Treisman wrote that his predecessors drew a parallel between such a mechanism and the action of a pacemaker neuron propagating an action potential along an axon. The same principle is used in this model, albeit without direct analogies with neural mechanisms.

2. The pacemaker's activity is regulated by a *specific arousal center*, which influences the rate of pulse generation. A definite level of specific arousal sets an average interval t' between pulses, which is calculated as $t' = qt_0$ with $q > 1$ and $q < 1$ for low and high arousal levels, respectively. The specific arousal level may depend on features of the experimental procedure, significant aspects of the situation, or other factors; under stable conditions (e.g., during presentation of a time interval), it usually remains constant or changes very slowly. Therefore, within a single trial, variations in t' are minimal, but its value may change from trial to trial.

3. The counter can operate in two modes: for writing to the store or for comparing the store with the current count. The interval durations in these modes are calculated by different formulas. This dual approach is necessary to minimize transformation errors (see below).

4. The measured interval is recorded in the *store* and retrieved from it by the comparator. The memory is a one-dimensional array of cells or addresses starting at a *zero point*. The model uses a logarithmic scale, so the best analogy for the memory block is a slide rule. Each cell corresponds to a certain number of pulses. If the counter detects n pulses, this measure will be read into a

cell at a distance of $\log n$ from the zero point. Activation spreads to neighboring cells, forming a distribution with a peak at $\log n$.

5. The *verbal selective mechanism* is an analog of long-term memory. It is intended to store symbolic time labels associated with certain points in the store ("two seconds," "one minute"). If the connection between the label and real experience is misaligned, the association will be corrected.

6. Retrieving a value (measure) from the store, the comparator transforms it back to a linear scale, thereby leveling possible errors. If the comparator has a systematic bias to the zero point, $\log n$ will be transformed to $\log n + \log m$, where $\log m$ is the bias value. After exponential transformation, a possible error constant r is added. The complete transformation is as follows:

$$RM = \exp(\log n + \log m) + r = nm + r,$$

where RM is the retrieved measure.

This measure is compared by the comparator with the current counter value to make a decision (e.g., when to stop or which response to choose). Different modifications of the formula for RM are given for different experiments.

7. The model provides feedback for the dynamic adjustment of memory parameters (logarithmic bias) based on the frequency of a particular response or on the total degree of store activation, thereby ensuring automatic error compensation and adaptation to experimental conditions.

8. Activation of any area in the store shifts $\log m$ toward the optimal value for the corresponding interval; thus, the value of $\log m$ at any moment will partially depend on the average effect of all areas active at this moment.

The value of Treisman's model lies not in the particular formulas but in the fundamental blocks, their structure and interrelation, as well as in the introduction of principles and parameters correcting errors and, conversely, adding systematic biases. These principles include the logarithmic storage of an interval with the correction term $\log m$, the arousal and inhibition effect of the pacemaker (the term q), and the additive correction r. The values of these corrections can be varied considerably without significantly affecting many predictions derived from the model's basic assumptions.

Treisman's model explains and generalizes a number of known psychophysiological phenomena, such as Weber's law for time intervals, the indifference interval, Vierordt's law, the crawling lengthening effect (the tendency for reproduced intervals to increase gradually during an experimental session), and other experimental effects observed in the studies of subjective time perception.

Let us dwell on several phenomena that have proven universal for many animal species, not only humans.

Timing obeys Weber's law. According to this empirical psychophysiological law [62], the minimum increase in a stimulus that will produce a detectable increase in sensation is proportional to the existing stimulus.

Treisman used the *Weber function*, i.e., a linear generalization of the original Weber's law [62] that better approximates the *difference threshold for time intervals*:

$$\Delta T = k(T + a),$$

where k and a are multiplicative and additive parameters, respectively. They take different values for different experiments. Due to this formula, the longer the time interval is, the greater variance its estimate will have. But the coefficient of variation remains approximately constant.

Timing obeys Vierordt's law. Vierordt's law [63] is a psychological phenomenon in the field of time perception, stating that humans systematically overestimate short time intervals and underestimate long ones. The law was established by German physiologist Vierordt in the mid-19th

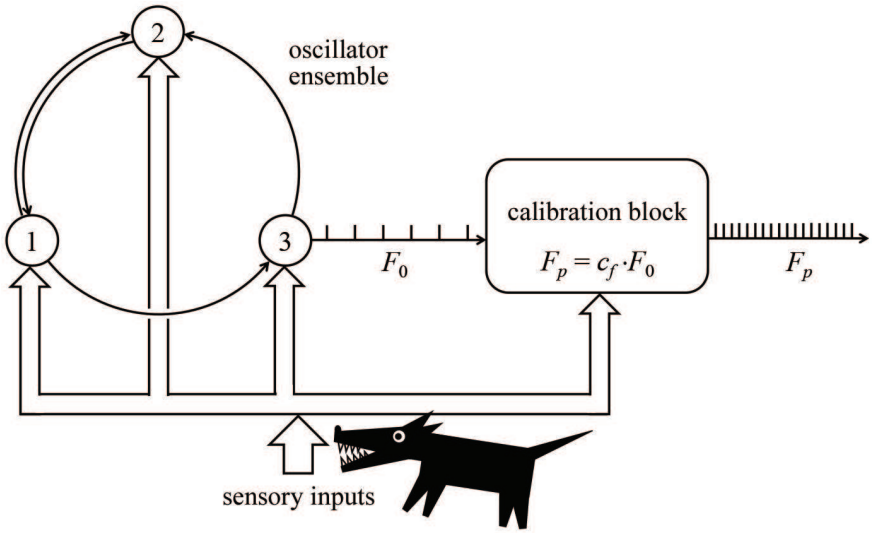


Fig. 2. The temporal pacemaker model consisting of two components: an oscillator and a calibration block. The scheme corresponds to the model from [27]; the sensory input, indicating "danger" and accelerating the rhythm, was added by the author.

century through experiments with stimuli of different durations. It explains one of the persistent errors observed in subjective time estimation and is often observed in a wide variety of experiments on time psychology.

If short intervals are overestimated and long ones underestimated, there must exist a boundary value that can be estimated accurately by a human. This value was called the *indifference interval*. During his experiments, Treisman tried to find it. The indifference interval varied across series; moreover, it grew due to the lengthening effect of reproduced intervals as fatigue accumulated during long experiments.

Nevertheless, three modes of time perception and reproduction can be distinguished:

- the subsecond interval, characterized by high sensitivity, low variance, and overestimation;
- the indifference interval (0.5–2 s), where the value of $\frac{\Delta T}{T}$ is minimal: $\frac{\Delta T}{T} \approx 0.07$;
- the supersecond interval, characterized by increasing errors and underestimation during reproduction.

The Weber function is usually considered for intervals exceeding the indifference interval.

Despite its simplicity, the model became very popular, and Treisman's pioneering paper [25] is still actively read today.

Note that Treisman is an expert in cognitive psychology, and his model has great practical and theoretical value for him and his followers. His subsequent studies, such as [26, 27] and others, presented evidence confirming the theory based on new experimental data. In [27], the pacemaker model was made more complex by adding a new block, the so-called "temporal pacemaker." It consists of two components, namely, an oscillator and a calibration block. In turn, the oscillator consists of three elements with excitatory and inhibitory links that generate a rhythm of a given frequency F_0. The output of the oscillator leads to the calibration block. The activity of the calibration block depends on a sensory input. Upon receiving a signal from the sensors, it forms a calibration factor c_f and outputs the new frequency $F_p = c_f F_0$ (Fig. 2).

This modification of the model makes a step toward biological plausibility. The elements of the oscillator block were called "conceptual neurons" by the authors. Indeed, the above scheme corresponds to a simple neural ensemble generating activity with a given frequency that is modulated by external inputs.

Summarizing this subsection, Treisman's scheme was the first to link a number of empirical phenomena via a single simple mechanism, unified Weber's and Vierordt's principles, substantiated the scalar nature of errors, and laid the foundation for subsequent quantitative theories such as SET. The model proved surprisingly robust to experimental tests and, despite its simplicity, remains one of the most influential in the psychology of time.

3.2. Scalar Expectancy Theory

Treisman's internal clock model was generalized to the *Scalar Expectancy Theory* by Gibbon and Church. This theory is a minimal information-processing model of the internal timer in animals and humans that explains the scalar property of errors in temporal tasks. It was proposed almost one and a half decades after the internal clock model in the paper [34].

According to the *scalar property*, which gave the theory its name, the error (spread) in the estimate of a time interval is proportional to the duration of the interval itself, so that the relative accuracy (coefficient of variation) of duration perception is preserved for different time ranges. Based on these estimates, animals form intervals of reward expectation; the difference between real and expected time is estimated as the ratio of these two values. The results confirmed Weber's law in the perception of time by animals.

A large amount of experimental data on the perception and memorization of time by animals was analyzed in [34]. Experiments with rats and pigeons were described: the animals' response to time intervals under various reinforcement learning procedures was studied. As shown, the animals estimate reinforcement time using a so-called "scalar synchronization process" that scales estimates for different time interval durations.

Gibbon analyzed experiments on behavior under fixed reinforcement intervals, with the tasks to estimate and reproduce duration. In fixed-interval experiments, animals learn to expect reinforcement after a strictly defined time period, and their response changes in a specific way when approaching the time of reinforcement. A large number of published experimental data on the behavior of rats and pigeons were analyzed. According to the analysis results, the spread of time estimation errors is proportional to the time interval itself. This makes the coefficient of variation constant for different intervals, analogously to Weber's law in sensory systems.

The model developed by Gibbon is similar to Treisman's internal clock model. However, there are several important differences to note. First, the Scalar Expectancy Theory was proposed for estimating time intervals in animals, not humans. This leaves an imprint on the model: there is no verbal mechanism. Second, the scalar law is realized using a much simpler procedure. SET has no logarithmic scale, as it analyzes the *ratio* of the remembered interval to the current one. Activation occurs if this ratio exceeds a given threshold. Changes also affected the memory block, split into *working* and *reference* memory. Another distinctive feature of SET is that, unlike Treisman, Gibbon strived for biological plausibility only in the reproduced phenomena, not in the structure. For instance, the pacemaker in his model is just a generator of ticks or pulses, produced at some average rate without binding to real time. In other words, they are some internal units of the model. Thus, SET models temporal encoding in a more general way; moreover, further research has shown that it is valid for both animals and humans.

Here are the key components of the first SET model:

- a *pacemaker*, generating a sequence of pulses (time units) at some average rate;
- an *accumulator*, collecting pulses from the pacemaker while the animal counts a given interval;
- *memory*, storing the accumulated pulses as a reference interval (reference memory) and retaining the current count (working memory);
- a *comparator/decision mechanism*, comparing the current accumulator (ongoing) with a sample from reference memory and determines the response moment or decision moment.

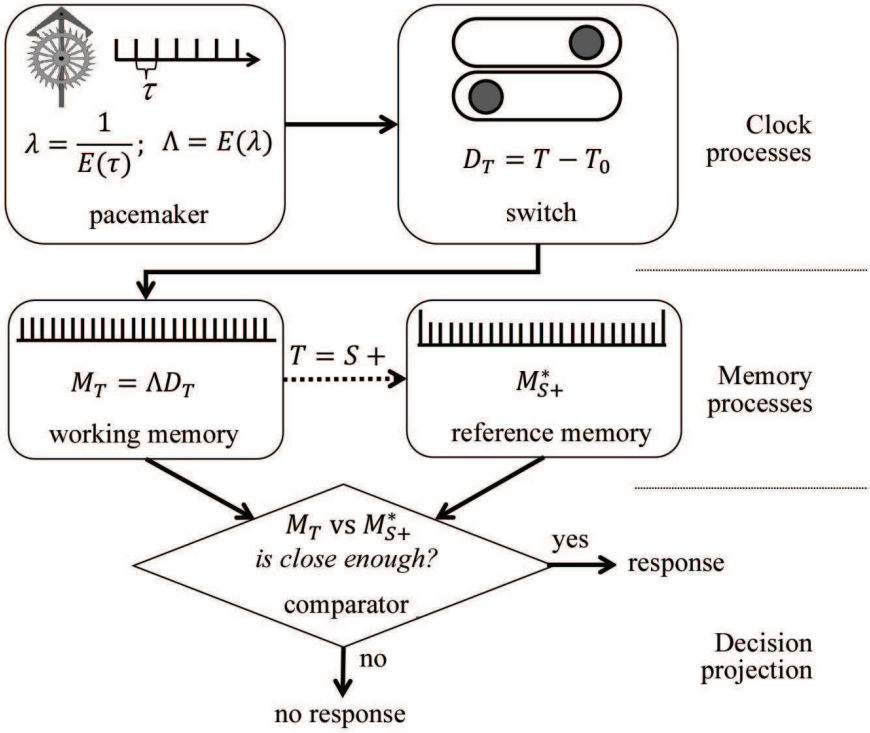


Fig. 3. The model of temporal information processing.

The operation of this model can be schematically described as follows. At the start of an interval, the pacemaker is triggered, and pulses begin to be counted by the accumulator. At the moment of reinforcement, the result enters memory and becomes a reference. In subsequent trials, the animal counts time, comparing the current number of pulses with the value from reference memory. When the current count is close to the reference (their ratio exceeds a threshold), the animal begins or stops responding.

Comparison of ratios explains the scalar property: the error and variance of time estimates increase proportionally to the interval length, and the shapes of psychometric functions for different intervals superimpose when converted to a relative time scale.

The scalar expectancy theory received its final form in [36]. The model of temporal information processing presented therein is shown in Fig. 3. It consists of three rows. The first row represents the clock process, including a pacemaker and a switch that allows pulses into an accumulator of working memory. The pacemaker generates pulses at an average frequency Λ, which is assumed to be sufficiently high relative to the time values used in experiments (from seconds to minutes). After appropriate training, the switch opens and allows pulses to pass for an average time of D_T into the accumulator of working memory (the second row) during a synchronization signal. The accumulator detects and stores the number of pulses (the mean value M_T). When, at the end of some trial, the subject exhibits the desired response and receives reinforcement ($T = S+$), the time value recorded in working memory during this trial is stored in longer-term reference memory for reinforced values (the mean value M^*_{S+}). The third row shows the decision process. A response occurs when the comparator judges that the current value in working memory on this trial is sufficiently close to the value in reference memory for the reinforced duration to warrant a response.

In [36], as in many subsequent publications by these authors (for example, see another program paper [37]), an experimental verification of the scalar property in time interval estimation by animals was associated with the so-called “peak procedure.”

The *peak procedure* is a technique used to study the memorization and reproduction of time intervals in animals: food (or other reinforcement) becomes available after performing some action, usually pressing a pedal or lever (operant tasks), or is provided after a fixed period of time in training trials (Pavlovian tasks). After training, in test trials, reinforcement is omitted, and the animals' responses are tracked. In operant tasks, the pedal pressing rate typically increases toward the expected time of reinforcement, reaches a maximum (peak), and then declines. This procedure allows assessing how an animal subjectively perceives the time interval, as well as how accurately it can predict the time of reward. Both types of experiments confirm the main tenets of scalar expectancy theory and Weber's law.

The peak procedure is often used in modern behaviorist research related to timing. Its application to study interval time estimation in a large number of bird, fish, and mammal species, including domestic chickens, pigeons, wood pigeons, black-capped chickadees, goldfish, mice, rats, fox possums, and humans, was described in [64]. The resulting data were analyzed in two ways. The first is an analysis of data averaged across multiple trials [65]. A common characteristic of interval timing is that animals, including humans, measure short intervals more accurately than long ones. Remarkably, when plotting the response-to-peak time ratio, the response curves overlap and are almost identical regardless of the interval measured. This feature, confirming the scalar property of time processing across different species, was documented in birds, fish, and mammals.

The second way is the analysis of individual trial data, in which three indicators are typically extracted from response patterns in individual trials: the start time (the transition from low to high level, i.e., to more intense pedal or lever pressing), the stop time (the transition from high to low level), and the run time (the duration of the intense phase). In [66], these data were used to compare scalar and connectionist models of time estimation; the results unequivocally illustrated the advantage of scalar models.

3.3. Methodological Principles for Constructing Temporal Encoding Models

When developing models for memorizing and reproducing time intervals for robotic systems and engineering applications, a key requirement is finding a compromise between biological plausibility and technical simplicity. On the one hand, a model must reproduce the fundamental principles of temporal encoding characteristic of living systems: the dependence of estimation accuracy on duration, the shifting of estimates toward the mean, and adaptation to changing conditions. This is necessary for robotic or artificial intelligence (AI) systems to act in real time as flexibly and efficiently as biological organisms, e.g., to respond to signals in a timely manner or to learn complex sequences of actions. However, a direct copy of highly complex neurophysiological mechanisms is not the best solution: hardware and software require resource-intensive computations, and excessive architectural complexity does not always improve the quality of results.

In this context, the most rational way is to construct phenomenologically grounded models that do not represent exact copies of biological processes but accurately embody their main properties. Treisman's internal clock model and Gibbon and Church's Scalar Expectancy Theory are examples of such a compromise: despite their extremely simple and straightforward mechanisms (compared to real brain networks), these models are consistently confirmed by empirical studies, in experiments with animals and with humans as well. The effects they predict—the scalar ratio of errors, variability, and shifts in interval estimates—are found in all types of time estimation tasks. Hence, these models are a convenient theoretical basis for constructing effective timing algorithms suitable for a wide range of practical problems in engineering, robotics, and cognitive science.

As an additional argument, we present a brief polemic between two groups of scholars. The authors of [67] criticized the classical internal clock concept as overly mechanistic and limited in understanding causality in biological systems, proposing to look at timing more integratively and

abandon rigid mechanistic models. A response to that paper was published the same year [68], where internal clock models were defended by arguing that they retain important explanatory power and serve as an effective foundation for understanding temporal processes; also, the incorrect or incomplete interpretation of the models by critics was underlined therein. The same opinion was stated by the authors of the review [1], mentioned at the beginning of this paper. Based on the above arguments, here we give close attention to internal clock models and the Scalar Expectancy Theory stemming from the former. The subsequent periods of research into the perception and memorization of time intervals will be described in less detail below. However, a thorough description of the properties and operating principles of the network models mentioned in subsection 4.3, together with a detailed discussion of state-of-the-art results, deserves a separate large paper.

4. FURTHER CHRONOLOGY OF TEMPORAL ENCODING STUDIES

The models and theoretical results obtained in the preclinical period turned out to be so universal that they continue to receive new confirmations. Behaviorist studies of timing mechanisms permeate all stages up to the present. Nevertheless, with the advent of new equipment and new data, research on temporal encoding required new methods.

4.1. The Neurophysiological Turn (1980–1990)

The transition from behavioral to neurophysiological studies of timing in the 1980s was associated with the emergence of new experimental methods and paradigms for investigating the neural mechanisms of temporal processing. Starting from the 1980s, the focus was gradually shifting from the models of human and animal behavior and the search for formal psychophysical regularities (such as Weber's law) to the analysis of the functioning of neural circuits, brain structures, and neurotransmitter systems involved in temporal processing. Here, we highlight the research works of Ivry, Meck, and their colleagues.

Are common neural mechanisms used in time perception and the motor reproduction of time intervals? The studies for answering this question were carried out in [69]. According to the experiments, there are significant individual differences in the accuracy of motor timing (rhythmic finger and foot tapping) and perceptual estimation of short intervals, supporting the idea of a common "clock mechanism" involved in both perception and motor production of time, with additional contributions from specific motor processes. These results laid the groundwork for subsequent theories of distributed timing.

A series of experiments with patients with cerebellar lesions was described in [16–19]. Note that these experiments were conducted over many years, and the cited works belong to different periods; for a more holistic picture, they will be described together. The influence of various types of neurological disorders on time measurement functions was studied in [16]. As shown therein, the cerebellum plays a key role in the accurate perception and reproduction of time intervals in both motor and perceptual tasks. Patients with cerebellar damage have significant impairments in these functions, whereas other neurological groups do not suffer from such problems. This indicates a specific role of the cerebellum as a central timing mechanism operating independently of the task type. Thus, the cerebellum is considered an important component of the distributed timing system in the brain.

According to [17], patients with cerebellar lesions have difficulty estimating the duration of a short auditory stimulus and the speed of a moving visual stimulus. The cerebellum was described not as a single oscillator but as a network of many timers activated for different tasks, i.e., from motor to sensory learning. The authors also discussed that cerebellar activation in cognitive experiments may reflect the preparation of temporal response patterns rather than the analysis of

abstract data. The studies of patients with cerebellar lesions in tasks requiring timing coordination of events and actions were continued in [18]. Previous research into temporal encoding was summarized in the paper [19]. It considered motor and perceptual tasks in which critical events occur at different time intervals (hundreds of milliseconds and several seconds). One key question is whether the representation of temporal information depends on a specialized system, is distributed, or is computed locally, depending on the task. The authors again emphasize the crucial role of the cerebellum, as it is associated with tasks requiring precise time measurement. In addition, they paid attention to the basal ganglia as a specialized system for time measurement, especially for longer intervals. By the authors' hypothesis, this structure is associated with decision processes. The authors were already less categorical in the later publication [70], formulating two alternative models for the perception of time flow, namely, specialized and universal (intrinsic). The former model emphasizes that the stimulus duration is estimated by specialized neural mechanisms intended to represent temporal relationships between events. The latter model is based on the assumption that duration representation may be universal, arising from the intrinsic dynamics of non-specialized neural mechanisms. The authors analyzed both models, albeit without final conclusions, merely outlining directions for further research related to the integration of these two paradigms.

Another line of studies is associated with neurotransmitter systems and neuromodulation in temporal encoding. Pharmacological research by Meck et al. demonstrated the different effects of dopaminergic and cholinergic drugs on time measurement [57–59]. The results of the experiments were interpreted within the scalar timing theory. As discovered in [58], an impact on the dopaminergic system mainly affects the perception phase of duration: with dopamine deficiency, time intervals are subjectively stretched. The impact of cholinergic drugs on time memory was investigated in [59]. The main experiment was to assess how increasing or blocking the level of acetylcholine in the brain with drugs affects memory for time intervals when performing a time discrimination task. It was suggested that the level of acetylcholine in the brain determines the "transfer rate" between the internal clock and memory. A cholinergic agonist increases the speed of duration memorization, causing underestimation of time; an antagonist slows it down, leading to overestimation.

In addition to transmitter modulation of time perception, Meck and colleagues also investigated neural correlates of temporal encoding. Their paper [71] is one of the most influential works on temporal encoding in the hippocampus. It showed a crucial role of the hippocampus in the retention and integration of temporal traces in working and long-term memory, not just in sensitivity to signal length. This work was one of the first to link temporal and spatial memory directly with hippocampal function in animals.

In [72], Meck and Church proposed a model where the brain's internal system can switch between counting events and measuring time intervals. This universal system uses the same mechanisms for both tasks, changing settings depending on the demands of the situation. According to animal experiments, both counting and timing obey similar laws and exhibit the scalar property of errors. Thus, the model explains how the brain flexibly manages time and quantity using a single adaptive system. In a continuation of that research, the authors experimentally proved the possibility of simultaneously measuring multiple intervals (clock multitasking) [73]. This result strengthened the position of models with several independent or context-tunable internal timers and expanded the universal timing model proposed in [72].

While investigating the neural mechanisms of working memory and action control, Niki and Watanabe tried to identify the neurophysiological correlates of temporal processing in various areas of the monkey cerebral cortex, primarily in the prefrontal and anterior cingulate cortex [14]. Neurons responsible for various stages of timing behavior–expectation, action preparation, and responding—were identified. The work was one of the first to associate neural activity in the pre-

frontal cortex with counting time intervals and mechanisms of temporal expectation. These results became the basis for subsequent studies of working memory, temporal encoding, and cognitive control.

Thus, during the period under consideration, it became possible to build behavioral models and, moreover, to link timing directly to definite brain structures (cerebellum, basal ganglia, prefrontal cortex, and hippocampus). The use of pharmacological interventions to "dissociate" different components of the scalar timing model was initiated. And the concept of a distributed timing system appeared: temporal processing began to be seen as a function of distributed neural networks rather than a single "biological chronometer." The above advances prepared the transition to the next period.

4.2. Distributed Neural Networks and Network Modeling of Timing (1990–2000)

The 1990s in neuroscience were called the "decade of the brain" due to large-scale initiatives to study brain structure and function [74]. In the research of neural temporal encoding, this period marked a transition from the search for a single center of time to the concept of distributed neural timing systems involving the cerebellum, basal ganglia, supplementary motor area (SMA), and frontal and parietal cortex. Neuroimaging methods (fMRI, PET) began to be introduced into research to map brain networks involved in temporal processing.

During those years, ideas about temporal processing as a function of distributed, dynamically connected brain systems took shape. The first distributed neural network models of temporal encoding were presented [32, 38, 40–42]. As a rule, the main research method was observation of patients with various brain lesions and identification of dysfunctions associated with these lesions. In the previous section, such experiments were described in [16–19]. They localized brain regions responsible for some aspects of time perception. A new important task was to separate the internal clock components from other sources of temporal variability.

A key review on the neurobiology of time during that period was provided in [43]. The main models and neural mechanisms underlying the perception and reproduction of time intervals in humans and animals were considered therein. Hazeltine et al. summarized experimental data indicating that timing is realized by distributed neural systems. Different brain areas (cerebellum, basal ganglia, premotor cortex, and SMA) participate differently in various tasks, e.g., motor and perceptual. Thus, there is no single center of time; rather, temporal processing functions are distributed and context-dependent. As demonstrated, there exist common mechanisms for motor and perceptual timing: the variation of errors in time is similar in both motor and sensory tasks, indicating the operation of a single or closely related timing system. At the same time, the difference in processing short and long intervals may be explained by the involvement of different neural pathways and the interaction of memory and attention. The authors considered the hypothesis of multiple specialized timers (e.g., independent processes for different limbs or for different contexts) that can be integrated at a higher level when performing complex sequential or parallel actions. That paper played an important role in the transition from searching for a neural time center to understanding timing as the result of distributed, specialized, and interacting brain networks.

With the new instrumental base, research into the impact of neurotransmitters on time perception continued. The role of neurochemistry (dopamine and acetylcholine) was demonstrated, and it was confirmed that disturbances in the systems cause specific timing deficits. Also note a fundamental review devoted to the role of brain neurochemical systems in the mechanisms of perception, estimation, and storage of time [60]; data from laboratory experiments, pharmacology, and clinical research were integrated therein to reveal how different neurotransmitters influence the internal clock and memory processes. The new data confirmed that dopamine regulates the speed of the internal clock mechanism, while acetylcholine is responsible for storing temporal information

in memory. Studies with pharmacological manipulations and clinical observations demonstrated that disturbances in these neurochemical systems lead to characteristic errors in time estimation: either accelerating or slowing down subjective time, or impairing memory of time intervals.

Based on experimental data, the theoretical preclinical models of time were tested in humans and animals. The scalar expectancy theory was confirmed by the results of several new experiments and gained a second wind [32, 33]. The new work by Church, Meck, and Gibbon [33] aimed to determine the characteristics of the internal clock, temporal memory, and decision processes involved in temporal integration, based on analysis of individual trials in rats. Three groups of 10 rats each were trained with reinforcement after 15, 30, or 60 seconds. Along with the reinforced intervals, long non-reinforced test trials were included in the consideration. On each test interval, the times of start, stop, and peak of the active response period (e.g., lever pressing) were measured to analyze the behavior structure not by averages but by individual sessions and responses. The following behavioral model was proposed and validated: in each individual trial, the animal is guided with a random sample of reinforcement time retrieved from memory, and the decisions to start and stop responding are determined by independent thresholds. According to the covariance analysis of different behavioral characteristics (the start and stop times, the peak and width of the active period), the scalar properties—the interval's spread proportional to its length—manifest at the level of individual trials as well, not only in averaged data.

One of the first connectionist models of temporal encoding was proposed by Church and Broadbent [32]. As pointed out in their work, the scalar timing theory explains well the ability of animals to estimate time intervals (on the one hand) but relies on some cognitive phenomena that are difficult to reproduce with known biological mechanisms (on the other hand). Therefore, a new connectionist model of the scalar expectancy theory was developed to explain how organisms can accurately measure time intervals at the level of neural networks. The model is based on the operation of a group of neural oscillators. Each oscillator has its own frequency; combinations of their states at the time of reinforcement (e.g., receiving a reward) are stored in memory. When it is necessary to reproduce this interval again, the current phase of each oscillator is compared with the stored pattern from memory. The decision to start/stop an action is made when the current set of phases matches that recorded during learning. The emphasis is on real or simulated neural network mechanisms instead of an abstract pulse generator and accumulator. The model describes the realization of temporal processing in biological neural networks without the need for a single center of time. The model successfully explains experimental data on operant timing tasks and confirms the distributed nature of temporal processing in the brain.

Theoretical models explaining how neural networks and individual neurons can encode, store, and reproduce time intervals were surveyed [40]. According to the author, although the central nervous system is obviously capable of accurately encoding time, the realization of the mechanisms for storing and processing temporal information is completely unclear. It is difficult to understand how neurons can accurately operate on different time scales from seconds to minutes; however, this occurs in the control of everyday behavior. The main difficulties of neural time encoding, as well as some ways of using neurons to encode temporal information, were described in the paper. Models based on the following different principles were reviewed: accumulation of pulses or signals; use of groups of neural oscillators (oscillator models); and dynamics of large neural networks for encoding long intervals. Two original models where not individual cells but a population of neurons jointly encodes a time interval were discussed in detail. Such approaches enable more robust and plastic time reproduction even under the variability and noise in neural systems. It was described how artificial neural network models learn to recognize and extract temporal patterns, and which of these properties could potentially be realized in the brain. As shown by computer simulations of various approaches (oscillator, population, and network models), such mechanisms can explain

experimentally observed timing properties in animals and humans. The author wondered which of the model solutions are actually realized in real neural tissue and emphasized the need for further empirical research to find an answer.

A three-layer neural network reproducing the results of classical experiments on duration discrimination in animals was proposed in [38]. The first layer consists of clusters of neurons with probabilistic internal feedback that maintain short-term (working) activity for a random period. The second layer is a spiking neuron that continues to be active as long as a sufficient number of clusters from the first layer remain active. The third layer responds to the end of activity in the second layer by generating a short burst of spikes. The model generates time intervals with a distribution depending on three parameters, namely, the number of clusters in the first layer, their activity time, and the threshold of the neuron in the second layer. Intervals can be trained at different layers of the network by changing the corresponding parameters to obtain different Weber fraction characteristics, from S-shaped to linear and saturating curves. The model shows how populations of interacting neuron clusters can measure time intervals with characteristics corresponding to the scalar regularities observed in animals and humans.

The concept of state-dependent timing was laid out in [41] and widely developed later. Buonomano and Merzenich investigated how biologically plausible neural networks can transform temporal information (structure in time, e.g., a sequence of signals or intervals) into a spatial code, i.e., a stable pattern of activity in neural populations. A neural network was created in which neurons are connected according to some real properties of the brain (with time delays and synaptic plasticity). The network is capable of memorizing a definite temporal sequence of input stimuli, so that a unique spatial pattern of neural activation corresponds to a specific temporal sequence. The model uses parameters analogous to real cortical networks, confirming the possibility of such a temporal-to-spatial transformation in living neural circuits. Such a mechanism may explain how the brain recognizes and distinguishes complex temporal patterns (words, rhythms, or sequences of stimuli) without requiring specialized clocks or universal timing. The above work was one of the first demonstrations that neural networks can represent temporal information spatially, which is extremely important for understanding sensory processing, learning, and memory. The flourishing of state-dependent network models came in the next decade. They will be discussed in detail in subsection 4.3.

The authors of [42] formulated one of the most interesting problems in the field of interval timing: how the brain measures events lasting minutes using neural processes based on milliseconds. The main attention was paid to models resting on the coincidence of activity in different neural circuits, i.e., the so-called coincidence-detection mechanisms. By assumption, time intervals can be encoded according to the principle of coincidence in the operation of many oscillating and interacting neural units. The authors emphasized the evolutionary aspect by considering the role of similar mechanisms in different animal species. The universality and fundamentality of this temporal encoding method for the brain were underlined. Interval encoding mechanisms were linked to the neurobiological foundations of learning, memory, and integration of sensory information. As in most works of this period, it was concluded that timing is realized in the brain by distributed networks, not a single chronometer. In addition, the coincidence mechanism may be a universal principle of time representation in vertebrates. As in other Meck's works, attention was also paid to the key neurotransmitters involved in temporal information processing.

Thus, the main trend of the period under consideration is the shift of research toward distributed mechanisms of temporal encoding and the search for the neural correlates of time interval processing in several brain areas. Attempts were made to build biologically plausible models that, on the one hand, would satisfy Weber's law, and on the other hand, would have some properties of biological neural ensembles.

4.3. Systems Neuroscience (2000–2010)

The period 2000–2010 can be characterized as an era of neuroimaging revolution and integration of cognitive neuroscience in the study of timing. While the 1990s were devoted to functional mapping and the conceptualization of distributed systems, the 2000s were marked by the widespread introduction of modern neuroimaging methods to investigate the neurobiological correlates of temporal encoding. During this period, functional MRI (fMRI) became a standard tool for studying temporal processing in humans, as it maps brain activity during timing tasks with high spatial resolution. The development of positron emission tomography and high-resolution electroencephalography became key to studying the temporal dynamics of neural processes.

There was a transition from the analysis of individual (albeit multiple) structures to the study of their network interactions and functional connectivity between brain areas. In the 2000s, the idea that the brain uses several specialized systems for time processing depending on the temporal scale and task type finally took shape. Among them are "automatic systems" for short intervals (milliseconds to seconds), associated with motor control, and "cognitively controlled systems" for long intervals (seconds to minutes), requiring attention and working memory.

The neurochemistry of timing received a deeper comprehension. Studies detailing the role of various neurotransmitter systems in temporal processing appeared, especially dopaminergic and cholinergic systems, with an emphasis on clinical studies of several diseases (Parkinson's disease, schizophrenia, etc.).

The active development of computational neuroscience led to the creation of biologically plausible timing models integrating neuroimaging data with theoretical constructs.

The main shift of the 2000s–2010s was the transition from simple structural mapping and experimental models to a comprehensive analysis of distributed brain networks in cognitive, computational, and clinical contexts.

A classical paper of this period, also coauthored by Meck, has the telling title *What Makes Us Tick?* [56]. This is a fundamental review devoted to the functional and neural mechanisms of interval timing, i.e., the brain's ability to estimate and control time intervals from seconds to minutes. The authors presented the brain as a complex time machine, where understanding the mechanisms of time tracking is key to understanding all brain functions. The paper considered interval timing as a central mechanism occupying an intermediate position between two other temporal systems: circadian rhythms (24-hour cycle) and millisecond timers (motor control, speech). Buhusi and Meck described how organisms have developed multiple timing systems active over more than 10 orders of magnitude of duration with varying degrees of precision. In mammals, the main intervals include (1) circadian clocks in the suprachiasmatic nucleus, (2) an automatic timer for millisecond motor control (cerebellum), and (3) a common flexible timer for seconds to hours (thalamo-cortico-striatal circuits).

The central conclusion was as follows. The brain represents time in a distributed manner, and timing occurs by detecting the coincident activation of different neural populations. Each brain structure contributes its resonance, and all these oscillations are tracked and integrated by basal ganglia circuits, acting like a conductor of an orchestra. As emphasized in the paper, the hallmark of interval timing is the proportionality of the estimation error to the interval length (duration), a property known as scalar timing, corresponding to Weber's law for most sensory modalities. Thus, the scalar expectancy theory, proposed thirty years earlier, was once again confirmed using advanced research methods.

The above paper continued neurochemical research as well. As shown therein, dopamine regulates the pace of subjective time, while acetylcholine is responsible for the storage of temporal information. Disturbances in these systems lead to specific errors in time perception and reproduction, as demonstrated in laboratory and clinical studies by the authors and their colleagues. The

paper discussed changes in timing mechanisms for patients with Parkinson's disease depending on the level of dopamine in the brain. According to the studies, patients taking medication estimate time quite normally, but as the effect of the medication wears off, their clock slows down.

The neural mechanisms of cognitive time measurement in humans were investigated in [75]. Special attention was paid to the functions of the right hemisphere prefrontal cortex. The work presented an analysis of two previously published experiments using different cognitive tasks for time estimation (sub-second and supra-second intervals). Based on their comparison, commonalities and differences in brain activation during time measurement were identified. The authors hypothesized the existence of a specialized cognitive system in the right hemisphere prefrontal cortex that provides a universal mechanism for time estimation in different tasks (from simple duration comparison to complex temporal sequences). The right-hemisphere specifics are consistent with clinical observations: damage to the right prefrontal area leads to deficits in temporal estimation. Lewis and Miall supported the idea of separate automatic (motor, short intervals) and cognitive (long and complex) clock systems in the human brain, noting the unique role of the right dorsolateral prefrontal cortex in cognitive timing.

Many publications on the neural mechanisms of time perception in humans belong to Harrington and colleagues. Let us dwell on the two most cited studies of the decade under consideration. How are different areas of the human brain activated during tasks of perception and comparison of time intervals? This issue was investigated in the paper [76]. Using event-related functional magnetic resonance imaging, the authors traced the evolution of activation of various brain areas at different stages of temporal information processing. According to their conclusion, time processing is the result of dynamic interaction between cortical and subcortical structures, and different stages of the task involve different brain areas. In [54], using fMRI, Harrington et al. aimed to discriminate and study two key processes of temporal processing: encoding time intervals and decision-making regarding their duration. The main goal of that work was to separate the neural systems involved in forming representations of time from the systems and processes associated with decision-making regarding interval durations. This important discrimination provides a better understanding of how the brain processes temporal information at different stages. The results showed the independence of the systems ensuring interval encoding and decision processes. Based on this significant finding, temporal processing involves several separate, albeit interacting, neural components.

The work of neurobiologist Buonomano and his colleagues laid the foundations for research into state-dependent network (SDN) models. These models assume that temporal information is encoded in the state evolution of a neural network, i.e., in how activity parameters (excitation flows, spiking activity, etc.) change over time within the network itself, rather than in a separate timer or oscillator. An extensive review of time processing mechanisms in the brain at different time scales was presented in the paper [77]. The authors analyzed how neural networks encode time intervals—from milliseconds to seconds and minutes—and discussed which biological systems, structures, and processes may provide this ability in animals and humans. Two main groups of models were discussed in detail. The first one is oscillator or clock-based models, where time is determined by the accumulation of pulses or the coincidence of oscillator phases, most often for subsecond and second intervals. The second group is SDN models, where the duration of an interval is represented by the state evolution of a dynamic neural network in which the temporal sequence of exciting individual neurons reflects time processing. Experimental data on the role of the cerebellum, basal ganglia, prefrontal cortex, and local neural networks were analyzed. Particular attention was paid to how synaptic plasticity, neural network dynamics, and the properties of individual neurons realize temporal selectivity. The work emphasized the importance of learning and reorganization of synaptic connections for fine-tuning temporal processing in different tasks. The results of this paper agree with the conclusions of other researchers: temporal processing is the outcome of the

cooperation of many structures and mechanisms specific to different task scales. Plasticity and learning ability make the timing system adaptive to external and internal changes: it is possible both to memorize new intervals and to adapt old ones to changed conditions.

This theme was developed in the paper [39]. As shown therein, time is encoded by the state evolution of multidimensional neural networks: each pattern of network activity carries information about the time elapsed since the last event. The SDN model explains why time estimation depends on context and previous stimulation, not just on the current stimulus. The work under consideration laid the foundations for the modern understanding of the internal network mechanisms of brain timing. The SDN model was further developed in [44, 45, 78, 79]. According to the arguments provided in [78], the brain's ability to perceive and discriminate time intervals (hundreds of milliseconds) can arise spontaneously from the dynamics of local neural networks, without special clock mechanisms. Temporal information is encoded by the time-varying patterns of activity in neural populations; this property was confirmed by data from both in vivo and in vitro experiments. Some experiments on cortical slices were described in [79]; as shown, if the same time interval is repeatedly supplied to a neural culture, the network will "learn" this interval, i.e., change its dynamic activity so as to predict the expected moment of the signal. Moreover, even isolated neural networks can store and reproduce time intervals only through internal changes in their dynamic properties, without external "time organizers."

A computer model of an internal clock based on a simple recurrent inhibitory neural network was presented in [80]. Temporal information in the model is encoded as the evolution of unique patterns of neural activity after an external signal. This model shows that the sequence of activation patterns can serve as an internal chronometer: time from the start signal is represented through changes in network state. In addition, the system is noise-resistant, reproducible, and can be restarted by a strong transient signal.

The role of neural oscillations in the operation of the brain's internal chronometer was examined in the paper [81]. The authors showed that not only individual frequencies (e.g., alpha or beta rhythms) but also the interaction of different brain rhythms are important for time estimation; the latter rhythms change depending on the task and time. Such a comprehensive approach better explains temporal information encoding by the brain through the dynamics of oscillatory processes.

Internal clock models (dedicated models) and SDN models (intrinsic models) were also compared by other researchers. Based on the analysis of the two models, Ivry, Spencer, and Karmarkar investigated how the brain encodes short time intervals (hundreds of milliseconds) [82]. Subjects were presented with sound signals of different structures (simple and with irrelevant tones) and asked to compare durations. Contextual variations helped test the sensitivity of temporal processing to interference. As it turned out, the presence of irrelevant sounds of variable duration, participants discriminated time worse (an argument in favor of intrinsic models that are sensitive to context). For longer intervals (300+ ms), the above effect disappeared, indicating a limitation of SDN models in terms of time range. However, the authors offered another explanation: factors related to attention may also influence performance on duration discrimination tasks. For instance, processing of short intervals strongly depends on the context; on the one hand, this may be an argument in favor of SDN models, and on the other hand, evidence of the influence of attention and task structure on time perception.

A biologically plausible model of time measurement based on neural integration (accumulation of signal to a threshold) was developed and analyzed in [83]. A neurobiological mechanism for measuring time intervals was presented; the neural integration of a signal to a threshold under noise explains the main behavioral regularities of interval timing (the scalar property and variations in errors) at the physiological level.

In a slightly later review of computational models of interval timing [84], the approaches existing at that time were divided into four main types: pacemaker–accumulator models (Treisman and Gibbon–Church models); multiple oscillator models; memory trace models based on SDN; and drift–diffusion models. Drift–diffusion models are a class of stochastic models that explain the brain's decisions on interval duration by gradual accumulation of a noisy signal until reaching a certain threshold. They clarify the distributions of time errors and responses. The longer the interval is, the greater the impact of noise (and hence, the scalar increase in the variability of decisions) will be. These models explain why, under the impact of attention, fatigue, or drugs, either the "drift rate" (the intensity of signal accumulation) or the width of the error band changes. For all four types, the authors proposed assessment criteria: the scalar property, the ability to reproduce retrospective/prospective timing effects, and the sensitivity to attention and neurochemical impacts.

Summarizing the intermediate results, the period 2000–2010 completed the transition from localist models to understanding timing as a function of complex, interacting neural networks. The widespread introduction of neuroimaging methods allowed shifting the focus of research from animals to humans and forming modern ideas of the multiplicity and flexibility of temporal processing systems in the brain.

The period under consideration laid the methodological and theoretical foundations for all subsequent research in the neuroscience of time, establishing standards for experimental design and creating a conceptual framework that researchers still use today.

4.4. The Era of Distributed and Integrative Approaches (2010–2020)

That period was characterized by several key shifts and trends. There was a sharp increase in the number of multidisciplinary publications, specialized journals appeared, and international forums and conferences were set up. The experimental arsenal was expanded by introducing high-density multi-electrode arrays, two-photon microscopy, genetic and optogenetic tools for direct investigation of neural network dynamics and causal interventions. Different models were synthesized and integrated. Hybrid models were built from parts of internal clock models, SDN, oscillator, and hybrid schemes. Such models are capable of explaining both retrospective and prospective time effects, rhythm, the scalar property, and the impact of attention and motivation. Active studies were devoted to interval timing and rhythm and, moreover, to temporal predictions, the order of events, and synchronization between agents. There was a growing interest in clinical aspects: models based on new clinical data were developed to explain timing changes in diseases (Parkinson's disease, depression, schizophrenia, ADHD, etc.), and neurorehabilitation protocols were created based on them.

One of the most important events of that period was the discovery of time cells in the hippocampus. The concept of time cells was introduced in the early 2010s by Eichenbaum's group of researchers. These neurons were described as cells that encode sequential moments of time within episodic memory, similar to how place cells encode position in space.

Neurons with properties of time cells were first described in 2008, see [7]. However, the term "time cells" appeared later, in 2011 ([8]). As is believed, the concept of time cells was substantiated in detail in the review [9], and the term became conventional.

Time cells are specialized neurons first discovered in the hippocampus of animals and then in humans. These cells fire actively at specific moments, marking the sequence of events in time within an episode or task. Using them, the brain forms the chronology and order of events, which is critical for episodic memory. As shown by studies, a novel result in the literature, the brain has temporal tags, similar to how place cells encode the animal's position in space. The discovery solved an old puzzle: how does the brain determine sequential events (e.g., word order, daily events)? Time cells sort moments within an episode to recall not only what and where happened, but also when it

happened. After rodents, time cells were found in monkeys and humans, confirming the universality of this mechanism in mammals. A notable outcome was that the cells support the properties of scalar timing (the spread of errors in determining time is proportional to the interval length), which is consistent with leading theories of timing.

The same hippocampal neurons were demonstrated to act as place cells (encoding location) and as time cells, depending on context. Therefore, the hippocampus builds not only spatial but also temporal maps of experience.

The discovery of time cells proved crucial for understanding how the brain reconstructs the chronology and duration of events, why strong emotions compress the time scale, and why humans overestimate unusual moments. A new focus of researchers was placed on the learning and remapping of time cells: these cells can change their activity depending on the task, demands, and experience, showing plasticity of time representations. With the development of intracranial recordings and neuroimaging, time cells are being actively studied in cognitive and clinical tasks in humans.

Time cells became the basis for a new wave of research into episodic memory, memory flexibility, and time awareness. As a result, the emphasis shifted from abstract theories to direct studies of neural network mechanisms of time and chronology at the level of nerve cells. For example, experiments on rats with various motor activity tasks were described in [10]; time cells and "distance cells" were identified in the hippocampus.

The research work of Merchant, a leading researcher in the neurobiology of time, covered key aspects of understanding how the brain processes temporal information [85–89]. Experimental studies in primates demonstrated the existence of multiple neural chronometers in different brain areas that are activated differently in synchronization and rhythm continuation tasks [85]. In [86, 87], the reader will find an introduction to the neurobiology of interval timing, systematizing the main concepts, methods, and discoveries in the study of time measurement mechanisms from milliseconds to seconds, as well as a fundamental review of the neural foundations of time perception and estimation, summarizing knowledge on how different brain structures (cerebellum, basal ganglia, and cortex) participate in temporal processing at different scales. In the medial premotor areas of primates, the realization of multiple levels of neural clocks for estimating and reproducing time intervals in the hundreds of milliseconds was investigated in the paper [88]. The authors found that individual neuronal populations encode elapsed and remaining time, as well as the duration and order of intervals, forming a hierarchical multilayer system in which temporal information is represented by the dynamic activity of cell populations. According to these results, temporal processing is realized as a distributed network function, where each layer contributes to precise control of rhythmic movements and their coordination in time. A comparative cross-species study of the ability to find rhythm in humans and primates was provided in the paper [89], indicating the evolutionary and neural foundations of rhythmic behavior and synchronization with external temporal patterns. As shown by the studies, the brain uses distributed specialized timing systems depending on the task and temporal scale. Special emphasis was placed on rhythmic behavior as a fundamental ability linking time perception, motor control, and evolutionary aspects of temporal processing in primates and humans. According to another study [23] as well, time is encoded by different neural networks depending on the task.

How is temporal information encoded and used during nervous system development? This issue was the subject of the biological review [90]. The authors considered how neural stem cells read and decode temporal signals arising from complex interactions between molecules, cells, and tissues to correctly determine cell fate during brain development.

Note another review [91], describing studies of how the cerebral cortex realizes the temporal processing necessary for cognitive functions. The authors showed that temporal encoding is an

intrinsic function of cortical neural networks, arising due to the temporal features of their activity and ability to plasticity. The role of neural circuit dynamics in maintaining "cognitive time" was highlighted: from moment-to-moment tracking of events to forming expectations and decision-making. As emphasized therein, understanding how the cortex encodes and uses time on short scales is crucial for building general models of cognitive brain activity, including perception, planning, prediction, and learning.

On the contrary, a highly specialized study was presented in [92]. It was demonstrated that professional percussionists reproduce time intervals with high accuracy and minimal errors compared to other people. This is due to their high sensory precision: they rely less on interval averaging and use more "optimal" temporal encoding.

Weber and Fairhall examined the role of neural adaptation in encoding and the dependence of the efficiency of processing changing environmental properties on the dynamic adjustment of neural sensitivity at different time scales [93].

Bayesian models of time interval memorization [50–53] radically changed the understanding of time perception, showing that systematic "errors" in duration estimation are not flaws but an optimal strategy of the brain to reduce uncertainty under internal noise. They are closely related to the concept of predictive processing in the brain; the main idea is that the brain constantly forms and updates hierarchical predictions about the external world based on an internal model, comparing them with real sensory data and minimizing prediction errors. These models treat the memorization and estimation of time intervals as a process of probabilistic inference, where prior experience is integrated with current sensory signals to obtain the most probable estimate. The brain combines current sensory data about time with prior experience, which shifts regression type toward the mean; however, exactly this feature makes the system more reliable in the long run. Neurochemical disturbances, such as dopamine deficiency in Parkinson's disease, change the balance between precise counting and the use of prior experience in favor of the latter, explaining specific patterns of temporal distortions in different clinical groups. Essentially, Bayesian models show that the human brain is not a precise chronometer, but an intelligent prediction system that sacrifices absolute accuracy for overall efficiency in an unpredictable world.

4.5. Modern Period

A review of recent research should be much more detailed. Many breakthrough works have been published, and new discoveries have been made. We hope to devote a separate paper to the results of recent years, with modern neurobiological studies and a focus on network models of encoding time intervals. This section only outlines the key milestones.

Modern studies have significantly expanded the class of computational and biological models by including recurrent neural networks, spiking systems, and hybrid architectures closely approximating real biological networks. A significant breakthrough has been the experimental confirmation of the role of neural oscillators and dynamic ensembles in the generation of time intervals, as well as evidence of the existence of a flexible and multimodal time encoding system in the brain. Works from this period have been integrating plasticity, adaptation, and error prediction mechanisms, as these concepts link temporal processing with memory and decision-making. According to several fresh studies, the distributed representation of time and its neural correlates are much more variable and complex than previously assumed, which calls into question rigid mechanistic models and stimulates a transition to ensemble and hierarchical models of temporal processing.

A new understanding of the computational mechanisms of time perception is being formed using neural network models, which are becoming a tool for explaining temporal computations in the brain [94]. A recurrent neural network model with plasticity capable of considering variability in temporal estimates has been presented in [95]. The scope of application of SDN models is expand-

ing; they are now used to describe neuromodulation, particularly modulation of locomotion [96]. A model for learning temporal sequences based on spiking neural networks has been developed, with application to the study of musical memory [97]. It has been shown that time intervals in ensembles of neural networks can be encoded on a logarithmic principle [98], which refers back to Treisman's first internal clock model. An integrative model combining neural oscillators with computational memory mechanisms has been proposed, going beyond the traditional scalar timing theory [99]. In a review of current data on the neural bases of duration estimation [100], Tsao et al. have distinguished between prospective (current, clock-like) and retrospective (recollective, reconstructive) time perception; as demonstrated, both types of processes rely on the dynamics of neural population states, but are implemented by different computational and neurobiological mechanisms. According to [101], memory and time perception use partially common neural mechanisms; without considering time representation, it is impossible to fully understand how memory works. Each cognitive action is treated as unfolding in time, and memory as a mechanism organizing this temporal flow. The impact of time constraints on cognitive performance, including mathematical productivity, has been studied in [102]. The control principles of neural dynamics revealed through the study of timing are being analyzed [103]. Neural mechanisms that simultaneously track several independent and asynchronous time intervals and rapidly switch between them in response to changes in the environment are being investigated [104]. Brain's synchronization of movements with external (music) and internal (mental rhythm reproduction) signals is being studied [105]. As discovered, the brain is capable of compensating for functional impairments under different methods of rhythm cueing. Collectively, these works significantly deepen the understanding of the mechanisms of time perception and processing in the brain at different levels and from different perspectives.

Let us particularly emphasize one review and two historical publications. The main neurocomputational models of interval timing have been reviewed in [106]. Models have been classified according to several principles: the presence or absence of threshold and adaptive clocks, and whether timing is a function of building a dedicated clock or emergent temporal encoding at the population level. The classification highlights both the common statistical properties of models (e.g., the scalar property and the shift of temporal estimates) and the diversity of computational solutions by which the real brain can realize timing in behavioral and neurophysiological tasks. The authors have considered key models from the classical internal clock to modern neural trajectory models, underlining the need to see the overall picture of the field's development (see the woods for the trees).

Meck's scientific legacy and contribution to the study of time perception and interval timing have been described in the paper [107], including recollections of his colleagues and students about his personality and role in shaping the scientific community in the field.

Treisman's classical work [25] on the internal chronometer and its role in time perception, published in 1963, has been analyzed in detail in [108]. Highlighting the internal clock model proposed by Treisman, the author has examined the main experiments, demonstrated how this model explains key effects (e.g., Weber's law), and discussed the relationship between Treisman's theory and later models such as the Scalar Expectancy Theory. In addition, the significance of this work for the modern understanding of timing psychology has been noted.

5. CONCLUSIONS

According to this review, the two interrelated lines of research—Treisman's internal clock model and the Scalar Expectancy Theory (SET)—remain methodologically and empirically fruitful after decades. Their key assumptions are surprisingly robust: with increasing measurement precision and complication of biologically plausible models, the basic properties reproduced by these models are again confirmed. This robustness indicates the fundamental nature of the principles laid down in the

models, making them a convenient support both for new data interpretations and for engineering applications.

The practical value of the models lies in their structural clarity and computational simplicity. In robotics and AI tasks, this is critical: biological plausibility here is reasonably understood as the reproducibility of observed phenomena (the scalar property, the bias of estimates toward the mean, and the superposition of curves in time) rather than as a literal reconstruction of physiology. The modular architecture (pacemaker, switch, working and reference memory, comparator, and decision mechanism) provides transparent error tracing, easy calibration, and portability across platforms and environments.

At the same time, a significant limitation of classical models is their low tolerance to noise and variability of natural signals. A promising direction is the transition to population encoding and network implementations, where temporal information is distributed across the states of dynamic systems. Network approaches, including state-dependent networks and recurrent architectures, can inherit the phenomenological advantages of internal clocks and SET, while increasing noise robustness, representational flexibility, and online adaptability. In this regard, we intend to systematically review the latest investigations of timing, with an emphasis on network models and mechanisms of the scalar property in distributed state representations.

We have started developing a model for memorizing and reproducing intervals that combines phenomenological verifiability with the robustness of population encoding. The first theoretical results have already been presented in [109]. We plan to create a simulation model, conduct experiments, and analyze the experimental data.

In general, the continuity from simple clocks to network time appears not as a rejection of classics, but as their natural development: the proven phenomena and criteria are preserved, but the carrier and encoding method change. This junction opens up opportunities for integrating cognitive theories of time with AI and robotics practices, taking into account real constraints of noise, resources, and adaptability.

REFERENCES

1. Tallot, L. and Doyere, V., Neural Encoding of Time in the Animal Brain, *Neuroscience & Biobehavioral Reviews*, 2020, vol. 115, pp. 146–163.
2. Kulieva, A.K., Berezner, T.A., Shishunova, A.N., et al., Cognitive Theories of Time Perception, *Vestnik of Saint Petersburg University. Psychology*, 2025, vol. 15, no. 1, pp. 51–65.
3. Tarabrina, N.Yu. and Kraev, Yu.V., Psychophysiological Assessment of Sense of Time in Football Referees of Different Qualifications, *Science and Sport: Current Trends*, 2018, vol. 21, no. 4, pp. 152–157.
4. Aldworth, Z.N., Dimitrov, A.G., Cummins, G.I., et al., Temporal Encoding in a Nervous System, *PLOS Computational Biology*, 2011, vol. 7, no. 5, art no. e1002041.
5. Li, L., Hou, C., Peng, C., et al., Encoding, Working Memory, or Decision: How Feedback Modulates Time Perception, *Cerebral Cortex*, 2023, vol. 33, no. 19, pp. 10355–10366.
6. Balasubramaniam, R., Haegens, S., Jazayeri, M., et al., Neural Encoding and Representation of Time for Sensorimotor Control and Learning, *Journal of Neuroscience*, 2021, vol. 41, no. 5, pp. 866–872.
7. Pastalkova, E., Itskov, V., Amarasingham, A., et al., Internally Generated Cell Assembly Sequences in the Rat Hippocampus, *Science*, 2008, vol. 321, no. 5894, pp. 1322–1327.
8. MacDonald, C.J., Lepage, K.Q., Eden, U.T., et al., Hippocampal "Time Cells" Bridge the Gap in Memory for Discontiguous Events, *Neuron*, 2011, vol. 71, no. 4, pp. 737–749.
9. Eichenbaum, H., Time Cells in the Hippocampus: a New Dimension for Mapping Memories, *Nature Reviews Neuroscience*, 2014, vol. 15, no. 11, pp. 732–744.
10. Abramson, S., Kraus, B.J., White, J.A., et al., Flexible Coding of Time or Distance in Hippocampal Cells, *Elife*, 2023, vol. 12, art. no. e83930.

11. Reddy, L., Zoefel, B., Possel, J.K., et al., Human Hippocampal Neurons Track Moments in a Sequence of Events, *Journal of Neuroscience*, 2021, vol. 41, no. 31, pp. 6714–6725.
12. Montchal, M.E., Reagh, Z.M., and Yassa, M.A., Precise Temporal Memories Are Supported by the Lateral Entorhinal Cortex in Humans, *Nat. Neurosci.*, 2019, vol. 22, pp. 284–288.
13. Dias, M., Ferreira, R., and Remondes, M., Medial Entorhinal Cortex Excitatory Neurons Are Necessary for Accurate Timing, *Journal of Neuroscience*, 2021, vol. 41, no. 48, pp. 9932–9943.
14. Niki, H. and Watanabe, M., Prefrontal and Cingulate Unit Activity During Timing Behavior in the Monkey, *Brain Research*, 1979, vol. 171, no. 2, pp. 213–224.
15. Narain, D., Remington, E.D., Zeeuw, C.I.D., et al., A Cerebellar Mechanism for Learning Prior Distributions of Time Intervals, *Nat. Commun.*, 2018, vol. 9, no. 1, art. no. 469.
16. Ivry, R.B. and Keele, S.W., Timing Functions of the Cerebellum, *Journal of Cognitive Neuroscience*, 1989, vol. 1, no. 2, pp. 136–152.
17. Ivry, R., Cerebellar Timing Systems, *International Review of Neurobiology*, 1997, vol. 41, pp. 555–573.
18. Ivry, R.B., Spencer, R.M., Zelaznik, H.N., et al., The Cerebellum and Event Timing, *Annals of the New York Academy of Sciences*, 2002, vol. 978, no. 1, pp. 302–317.
19. Ivry, R.B. and Spencer, R.M., The Neural Representation of Time, *Current Opinion in Neurobiology*, 2004, vol. 14, no. 2, pp. 225–232.
20. Yamazaki, T. and Tanaka, S., Computational Models of Timing Mechanisms in the Cerebellar Granular Layer, *The Cerebellum*, 2009, vol. 8, pp. 423–432.
21. Stigliani, A., Jeska, B., and Grill-Spector, K., Encoding Model of Temporal Processing in Human Visual Cortex, *Proceedings of the National Academy of Sciences*, 2017, vol. 114, no. 51, pp. e11047–e11056.
22. Howard, M.W., Shankar, K.H., Aue, W.R., et al., A Distributed Representation of Internal Time, *Psychol. Rev.*, 2015, vol. 122, pp. 24–53.
23. Paton, J.J. and Buonomano, D.V., The Neural Basis of Timing: Distributed Mechanisms for Diverse Functions, *Neuron*, 2018, vol. 98, no. 4, pp. 687–705.
24. Robbe, D., Lost in Time: Relocating the Perception of Duration Outside the Brain, *Neuroscience & Biobehavioral Reviews*, 2023, vol. 153, art. no. 105312.
25. Treisman, M., Temporal Discrimination and the Indifference Interval: Implications for a Model of the "Internal Clock", *Psychological Monographs: General and Applied*, 1963, vol. 77, no. 13, pp. 1–31.
26. Treisman, M., Cook, N., Naish, P.L., et al., The Internal Clock: Electroencephalographic Evidence for Oscillatory Processes Underlying Time Perception, *Quarterly Journal of Experimental Psychology Section A*, 1994, vol. 47, no. 2, pp. 241–289.
27. Treisman, M., Faulkner, A., Naish, P.L., et al., The Internal Clock: Evidence for a Temporal Oscillator Underlying Time Perception with Some Estimates of Its Characteristic Frequency, *Perception*, 1990, vol. 19, no. 6, pp. 705–742.
28. Treisman, M., The Information-Processing Model of Timing (Treisman, 1963): Its Sources and Further Development, *Timing & Time Perception*, 2013, vol. 1, no. 2, pp. 131–158.
29. Church, R.M., Theories of Timing Behavior, in *Contemporary Learning Theories*, Routledge, 2019, pp. 41–72.
30. Church, R.M., Properties of the Internal Clock, *Annals of the New York Academy of Sciences*, 1984, vol. 423, pp. 566–582.
31. Church, R.M. and Gibbon, J., Temporal Generalization, *Journal of Experimental Psychology: Animal Behavior Processes*, 1982, vol. 8, no. 2, pp. 165–186.
32. Church, R.M. and Broadbent, H.A., A Connectionist Model of Timing, in *Neural Network Models of Conditioning and Action*, Michael, S.G.J.E.R.S. and Commons, L., Eds., Hillsdale: Lawrence Erlbaum Associates, 1991, pp. 225–240.
33. Church, R.M., Meck, W.H., and Gibbon, J., Application of Scalar Timing Theory to Individual Trials, *Journal of Experimental Psychology: Animal Behavior Processes*, 1994, vol. 20, no. 2, pp. 135–155.
34. Gibbon, J., Scalar Expectancy Theory and Weber's Law in Animal Timing, *Psychological Review*, 1977, vol. 84, no. 3, pp. 279–325.

35. Gibbon, J. and Church, R.M., Time Left: Linear versus Logarithmic Subjective Time, *Journal of Experimental Psychology: Animal Behavior Processes*, 1981, vol. 7, no. 2, pp. 87–108.

36. Gibbon, J., Church, R.M., and Meck, W.H., Scalar Timing in Memory, *Annals of the New York Academy of Sciences*, 1984, vol. 423, no. 1, pp. 52–77.

37. Gibbon, J. and Church, R.M., Representation of Time, *Cognition*, 1990, vol. 37, no. 1–2, pp. 23–54.

38. Bugmann, G., Towards a Neural Model of Timing, *Biosystems*, 1998, vol. 48, no. 1–3, pp. 11–19.

39. Karmarkar, U.R. and Buonomano, D.V., Timing in the Absence of Clocks: Encoding Time in Neural Network States, *Neuron*, 2007, vol. 53, no. 3, pp. 427–438.

40. Miall, C., Models of Neural Timing, *Advances in Psychology*, 1996, vol. 115, pp. 69–94.

41. Buonomano, D.V. and Merzenich, M.M., Temporal Information Transformed into a Spatial Code by a Neural Network with Realistic Properties, *Science*, 1995, vol. 267, no. 5200, pp. 1028–1030.

42. Matell, M.S. and Meck, W.H., Neuropsychological Mechanisms of Interval Timing Behavior, *Bioessays*, 2000, vol. 22, no. 1, pp. 94–103.

43. Hazeltine, E., Helmuth, L.L., and Ivry, R.B., Neural Mechanisms of Timing, *Trends in Cognitive Sciences*, 1997, vol. 1, no. 5, pp. 163–169.

44. Buonomano, D.V. and Maass, W., State-Dependent Computations: Spatiotemporal Processing in Cortical Networks, *Nature Reviews Neuroscience*, 2009, vol. 10, no. 2, pp. 113–125.

45. Buonomano, D.V., Bramen, J., and Khodadadifar, M., Influence of the Interstimulus Interval on Temporal Processing and Learning: Testing the State-Dependent Network Model, *Philosophical Transactions of the Royal Society B: Biological Sciences*, 2009, vol. 364, no. 1525, pp. 1865–1873.

46. Buonomano, D.V., The Neural Mechanisms of Timing on Short Timescales, in *Subjective Time: The Philosophy, Psychology, and Neuroscience of Temporality*, Arstila, V. and Lloyd, D., Eds., The MIT Press, 2014, pp. 329–342.

47. Buonomano, D.V. and Laje, R., Population Clocks: Motor Timing with Neural Dynamics, *Trends in Cognitive Sciences*, 2010, vol. 14, no. 12, pp. 520–527.

48. Zhou, S. and Buonomano, D.V., Neural Population Clocks: Encoding Time in Dynamic Patterns of Neural Activity, *Behavioral Neuroscience*, 2022, vol. 136, no. 5, pp. 374.

49. Zhou, S., Masmanidis, S.C., and Buonomano, D.V., Encoding Time in Neural Dynamic Regimes with Distinct Computational Tradeoffs, *PLOS Computational Biology*, 2022, vol. 18, no. 3, art. no. e1009271.

50. Shi, Z., Church, R.M., and Meck, W.H., Bayesian Optimization of Time Perception, *Trends in Cognitive Sciences*, 2013, vol. 17, no. 11, pp. 556–564.

51. Gu, B.-M., Jurkowski, A.J., Lake, J.I., et al., Bayesian Models of Interval Timing and Distortions in Temporal Memory as a Function of Parkinson's Disease and Dopamine-Related Error Processing, *Time Distortions in Mind: Temporal Processing in Clinical Populations*, Vatakis, A. and Allman, M.J., Eds., Leiden–Boston: Brill, 2015, pp. 281–327.

52. Gu, B.-M., Jurkowski, A.J., Shi, Z., et al., Bayesian Optimization of Interval Timing and Biases in Temporal Memory as a Function of Temporal Context, Feedback, and Dopamine Levels in Young, Aged and Parkinson's Disease Patients, *Timing & Time Perception*, 2016, vol. 4, no. 4, pp. 315–342.

53. Sadibolova, R. and Terhune, D.B., The Temporal Context in Bayesian Models of Interval Timing: Recent Advances and Future Directions, *Behavioral Neuroscience*, 2022, vol. 136, no. 5, art. no 364.

54. Harrington, D.L., Boyd, L.A., Mayer, A.R., et al., Neural Representation of Interval Encoding and Decision Making, *Cognitive Brain Research*, 2004, vol. 21, no. 2, pp. 193–205.

55. Balcı, F. and Simen, P., A Decision Model of Timing, *Current Opinion in Behavioral Sciences*, 2016, vol. 8, pp. 94–101.

56. Buhusi, C.V. and Meck, W.H., What Makes Us Tick? Functional and Neural Mechanisms of Interval Timing, *Nature Reviews Neuroscience*, 2005, vol. 6, no. 10, pp. 755–765.

57. Meck, W.H., Selective Adjustment of the Speed of Internal Clock and Memory Processes, *J. Exp. Psychol. Anim. Behav. Process*, 1983, vol. 9, pp. 171–201.

58. Meck, W.H., Affinity for the Dopamine D2 Receptor Predicts Neuroleptic Potency in Decreasing the Speed of an Internal Clock, *Pharmacol. Biochem. Behav.*, 1986, vol. 25, pp. 1185–1189.

59. Meck, W.H. and Church, R.M., Cholinergic Modulation of the Content of Temporal Memory, *Behav. Neurosci.*, 1987, vol. 101, pp. 457–464.
60. Meck, W.H., Neuropharmacology of Timing and Time Perception, *Cognitive Brain Research*, 1996, vol. 3, no. 3–4, pp. 227–242.
61. Balcı, F., Interval Timing, Dopamine, and Motivation, *Timing & Time Perception*, 2014, vol. 2, no. 3, pp. 379–410.
62. Ekman, G.O.S., Weber's Law and Related Functions, *The Journal of Psychology*, 1959, vol. 47, no. 2, pp. 343–352.
63. Glasauer, S. and Shi, Z., The Origin of Vierordt's Law: The Experimental Protocol Matters, *Psycholog. J.*, 2021, vol. 10, no. 5, pp. 732–741.
64. Colwill, R.M. and Balsam, P.D., Peak Procedure, in *Encyclopedia of Animal Cognition and Behavior*, Vonk, J. and Shackelford, T.K., Eds., Cham: Springer, 2022, pp. 5102–5107.
65. Balsam, P., Sanchez-Castillo, H., Taylor, K., et al., Timing and Anticipation: Conceptual and Methodological Approaches, *European Journal of Neuroscience*, 2009, vol. 30, no. 9, pp. 1749–1755.
66. Cheng, K. and Miceli, P., Modelling Timing Performance on the Peak Procedure, *Behavioural Processes*, 1996, vol. 37, no. 2–3, pp. 137–156.
67. Eckard, M.L. and Lattal, K.A., The Internal Clock: a Manifestation of a Misguided Mechanistic View of Causation?, *Perspectives on Behavior Science*, 2020, vol. 43, pp. 5–19.
68. Sanabria, F., Internal-Clock Models and Misguided Views of Mechanistic Explanations: a Reply to Eckard & Lattal (2020), *Perspectives on Behavior Science*, 2020, vol. 43. no. 4, pp. 779–790.
69. Keele, S.W., Pokorny, R.A., Corcos, D.M., et al., Do Perception and Motor Production Share Common Timing Mechanisms: a Correlational Analysis, *Acta Psychologica*, 1985, vol. 60, no. 2–3, pp. 173–191.
70. Ivry, R.B. and Schlerf, J.E., Dedicated and Intrinsic Models of Time Perception, *Trends in Cognitive Sciences*, 2008, vol. 12, no. 7, pp. 273–280.
71. Meck, W.H., Church, R.M., and Olton, D.S., Hippocampus, Time, and Memory, *Behavioral Neuroscience*, 1984, vol. 98, no. 1, pp. 3–22.
72. Meck, W.H. and Church, R.M., A Mode Control Model of Counting and Timing Processes, *Journal of Experimental Psychology: Animal Behavior Processes*, 1983, vol. 9, no. 3, pp. 320–334.
73. Meck, W.H. and Church, R.M., Simultaneous Temporal Processing, *Journal of Experimental Psychology: Animal Behavior Processes*, 1984, vol. 10, no. 1, pp. 1–29.
74. Jones, E.G. and Mendell, L.M., Assessing the Decade of the Brain, *Science*, 1999, vol. 284, no. 5415, pp. 739–739.
75. Lewis, P.A. and Miall, R.C., A Right Hemispheric Prefrontal System for Cognitive Time Measurement, *Behavioural Processes*, 2006, vol. 71, no. 2–3, pp. 226–234.
76. Rao, S.M., Mayer, A.R., and Harrington, D.L., The Evolution of Brain Activation During Temporal Processing, *Nature Neuroscience*, 2001, vol. 4, no. 3, pp. 317–323.
77. Mauk, M.D. and Buonomano, D.V., The Neural Basis of Temporal Processing, *Annual Review of Neuroscience*, 2004, vol. 27, no. 1, pp. 307–340.
78. Goel, A. and Buonomano, D.V., Timing as an Intrinsic Property of Neural Networks: Evidence from in Vivo and in Vitro Experiments, *Philosophical Transactions of the Royal Society B: Biological Sciences*, 2014, vol. 369, no. 1637, art. no. 20120460.
79. Goel, A. and Buonomano, D.V., Temporal Interval Learning in Cortical Cultures Is Encoded in Intrinsic Network Dynamics, *Neuron*, 2016, vol. 91, no. 2, pp. 320–327.
80. Yamazaki, T. and Tanaka, S., Neural Modeling of an Internal Clock, *Neural Computation*, 2005, vol. 17, no. 5, pp. 1032–1058.
81. Kononowicz, T.W. and Van Wassenhove, V., In Search of Oscillatory Traces of the Internal Clock, *Frontiers in Psychology*, 2016, vol. 7, art. no. 224.
82. Spencer, R.M., Karmarkar, U., and Ivry, R.B., Evaluating Dedicated and Intrinsic Models of Temporal Encoding by Varying Context, *Philosophical Transactions of the Royal Society B: Biological Sciences*, 2009, vol. 364, no. 1525, pp. 1853–1863.

83. Simen, P., Balci, F., Desouza, L., et al., A Model of Interval Timing by Neural Integration, *Journal of Neuroscience*, 2011, vol. 31, no. 25, pp. 9238–9253.

84. Addyman, C., French, R.M., and Thomas, E., Computational Models of Interval Timing, *Current Opinion in Behavioral Sciences*, 2016, vol. 8, pp. 140–146.

85. Merchant, H., Zarco, W., Perez, O., et al., Measuring Time with Different Neural Chronometers During a Synchronization-Continuation Task, *Proceedings of the National Academy of Sciences*, 2011, vol. 108, no. 49, pp. 19784–19789.

86. Merchant, H., Harrington, D.L., and Meck, W.H., Neural Basis of the Perception and Estimation of Time, *Annual Review of Neuroscience*, 2013, vol. 36, no. 1, pp. 313–336.

87. Merchant, H. and De Lafuente, V., Introduction to the Neurobiology of Interval Timing, in *Neurobiology of Interval Timing*, Merchant, H. and De Lafuente, V., Eds., Springer, 2014, pp. 1–13.

88. Merchant, H., Bartolo, R., Perez, O., et al., Neurophysiology of Timing in the Hundreds of Milliseconds: Multiple Layers of Neuronal Clocks in the Medial Premotor Areas, *Neurobiology of Interval Timing*, Merchant, H. and De Lafuente, V., Eds., Springer, 2014, pp. 143–154.

89. Merchant, H., Grahn, J., Trainor, L., et al., Finding the Beat: a Neural Perspective Across Humans and Non-Human Primates, *Philosophical Transactions of the Royal Society B: Biological Sciences*, 2015, vol. 370, no. 1664, art. no. 20140093.

90. Toma, K., Wang, T.C., and Hanashima, C., Encoding and Decoding Time in Neural Development, *Development, Growth & Differentiation*, 2016, vol. 58, no. 1, pp. 59–72.

91. Finnerty, G.T., Shadlen, M.N., Jazayeri, M., et al., Time in Cortical Circuits, *Journal of Neuroscience*, 2015, vol. 35, no. 41, pp. 13912–13916.

92. Cicchini, G.M., Arrighi, R., Cecchetti, L., et al., Optimal Encoding of Interval Timing in Expert Percussionists, *Journal of Neuroscience*, 2012, vol. 32, no. 3, pp. 1056–1060.

93. Weber, A.I. and Fairhall, A.L., The Role of Adaptation in Neural Coding, *Current Opinion in Neurobiology*, 2019, vol. 58, pp. 135–140.

94. Bi, Z. and Zhou, C., Understanding the Computation of Time Using Neural Network Models, *Proceedings of the National Academy of Sciences*, 2020, vol. 117, no. 19, pp. 10530–10540.

95. Hallez, Q., Mermillod, M., and Droit-Volet, S., Cognitive and Plastic Recurrent Neural Network Clock Model for the Judgment of Time and Its Variations, *Scientific Reports*, 2023, vol. 13, no. 1, art. no. 3852.

96. Bi, Z., Zhou, C., Pernia-Andrade, A.J., et al., Circuits for State-Dependent Modulation of Locomotion, *Frontiers in Human Neuroscience*, 2021, vol. 15, art. no. 745689.

97. Liang, Q., Zeng, Y., and Xu, B., Temporal-Sequential Learning with a Brain-Inspired Spiking Neural Network and Its Application to Musical Memory, *Frontiers in Computational Neuroscience*, 2020, vol. 14, art. no. 51.

98. Ren, Y., Allenmark, F., Muller, H.J., et al., Logarithmic Encoding of Ensemble Time Intervals, *Scientific Reports*, 2020, vol. 10, no. 1, art. no. 18174.

99. Shi, Z., Gu, B.-M., Glasauer, S., et al., Beyond Scalar Timing Theory: Integrating Neural Oscillators with Computational Accessibility in Memory, *Timing & Time Perception*, 2022, vol. 11, no. 1–4, pp. 198–219.

100. Tsao, A., Yousefzadeh, S.A., Meck, W.H., et al., The Neural Bases for Timing of Durations, *Nature Reviews Neuroscience*, 2022, vol. 23, no. 11, pp. 646–665.

101. Buonomano, D.V., Buzsaki, G., Davachi, L., et al., Time for Memories, *Journal of Neuroscience*, 2023, vol. 43, no. 45, pp. 7565–7574.

102. Hallez, Q. and Rebecchi, K., Evaluation and Time Constraint: Impact of Time Processing on Mathematical Performance, *The Journal of Experimental Education*, 2026, vol. 94, no. 3, pp. 464–481. https://doi.org/10.1080/00220973.2024.2403558

103. Stine, G.M. and Jazayeri, M., Control Principles of Neural Dynamics Revealed by the Neurobiology of Timing, *Annual Review of Neuroscience*, 2025, vol. 48, pp. 43–63.

104. Haim, S., Ofir, N., Deouell, L.Y., et al., Neural Signatures of Flexible Multiple Timing, *Journal of Neuroscience*, 2025, vol. 45, no. 24, art no. e2041242025.

105. Harrison, E.C., Grossen, S., Tueth, L.E., et al., Neural Mechanisms Underlying Synchronization of Movement to Musical Cues in Parkinson Disease and Aging, *Frontiers in Neuroscience*, 2025, vol. 19, art. no. 1550802.

106. Balcı, F. and Simen, P., Neurocomputational Models of Interval Timing: Seeing the Forest for the Trees, *Advances in Experimental Medicine and Biology*, 2024, vol. 1455, pp. 51–78.

107. Balcı, F., Vatakis, A., and Gu, B.-M., Remembering Warren H. Meck, *Timing & Time Perception*, 2023, vol. 11, no. 1–4, pp. 1–13.

108. Wearden, J.H., Treisman (1963): an Appreciation, *Timing & Time Perception*, 2024, vol. 13, no. 1, pp. 1–24.

109. Zhilyakova, L., Direct and Inverse Problems of Time Encoding by Neuron-Like Agents, in *Advances in Neural Computation, Machine Learning, and Cognitive Research VIII. NI 2024*, Redko, V., Yudin, D., Dunin-Barkowski, W., Kryzhanovsky, B., and Tiumentsev, Y., Eds., Selected Papers from the XXVI International Conference on Neuroinformatics, October 21–25, 2024, Moscow, *Studies in Computational Intelligence*, Cham: Springer, vol. 1179.

This paper was recommended for publication by A.I. Michalski, a member of the Editorial Board

ISSN 0005-1179 (print), ISSN 1608-3032 (online), Automation and Remote Control, 2026, Vol. 87, No. 5, pp. 465–481.

ROBUST, ADAPTIVE, AND NETWORK CONTROL

Design of Self-Checking Discrete Devices Based on Boolean Signal Correction with Weighted Sum Codes in the Residue Ring of a Given Modulus

D. V. Efanov[*,a,**] **and Y. I. Yelina**[*,b]

[*] *Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia*
[**] *Solomenko Institute of Transport Problems, Russian Academy of Sciences, St. Petersburg, Russia*
e-mail: [a] *TrES-4b@yandex.ru,* [b] *eseniya-elina@mail.ru*

Received June 16, 2025
Revised November 17, 2025
Accepted December 10, 2025

Abstract—It is proposed to design concurrent error-detection (CED) circuits for discrete automation and computing devices using Boolean signal correction (BSC) with weighted sum codes. Within this approach, a CED circuit corrects signals from all outputs of an object under diagnosis and involves a particular subset of codewords of a preselected weighted sum code. An algorithm is developed for selecting codewords used to design a CED circuit based on BSC; this algorithm allows choosing the best variant to cover faults at the object's outputs and ensures the self-checking property of the circuit. The features of organizing CED circuits based on the above method are shown.

Keywords: concurrent error-detection (CED) circuit, Boolean signals correction, weighted sum code, self-checking device, computation control at device outputs

DOI: 10.7868/S1608303226050065

1. INTRODUCTION

Many units and nodes of automation and computing systems must be self-checking for the timely detection of faults in their structures and mitigation of their manifestations [1]. This is especially important in critical systems, where a fault (error) may cause an accident or even a catastrophe.

In the design of self-checking discrete devices, methods of information theory, coding, and Boolean function theory are widely used [2–6]. Considering the properties of error-detecting and error-correcting codes makes it possible to design devices with given reliability characteristics and fault detection capabilities by manifestations in the form of signal distortions at specially selected checkpoints or operating outputs [7, 8].

This paper is devoted to further developing the theory of self-checking discrete device design through implementing external fault diagnosis means in the form of concurrent error-detection (CED) circuits using the properties of binary redundant codes for error detection only (without error correction). As such codes, we consider a wide class of weighted sum codes in the residue ring of a given modulus [9]. According to the authors' research, these codes can be effectively applied in the design of CED circuits based on Boolean signal correction (BSC); the idea to use such signal correction for building self-checking discrete devices was proposed in [10, 11]. With BSC, when each set of argument values is supplied to the inputs of a CED circuit, the signals from an object under diagnosis are converted into signals treated by the circuit as codewords of a preselected code. (This approach is in contrast to the conventional one, where the object's signals are supplemented with check signals without any conversion [12, 13].) Due to the properties of weighted sum codes, it is possible to design self-checking devices based on BSC and choose implementation options with the best values of the structural redundancy and testability indicators of CED circuit gates.

2. PROBLEM STATEMENT

Existing CED circuit design methods involve the error-detection properties of uniform block codes and the diagnostic properties of Boolean functions. In CED circuit design, the general error-detection properties of uniform block codes and Boolean functions are used. In fact, a CED circuit "inherits" the detecting properties of the redundant code (or a particular class of Boolean functions) selected at the design stage. It becomes impossible to consider in a CED circuit the structural features of objects under diagnosis (the configurations of gates and their connections with each other, as well as with inputs and outputs of the device, which affects the multiplicity and types of allowed errors at their outputs caused by internal structure faults). Therefore, two approaches emerge for further development of CED circuit design methods. The first is to construct a new redundant code by taking the object's structure and a given fault model into account. Such a problem was solved in [14]. The second approach is to use a particular subset of the set of codewords of some uniform block code in order to orient the selected codewords to cover faults at the object's outputs with certain properties. Research shows that this is possible using BSC. Let us formulate the following problem.

Consider a given combinational object under diagnosis $F(X)$, which computes the values of Boolean functions $f_1(X), f_2(X), \ldots, f_n(X)$ when supplying the sets of argument values $< x_t x_{t-1} \ldots x_2 x_1 > = < X >$ at its inputs. It is required to build a self-checking discrete device based on BSC and a wide class of weighted sum codes with a particular subset of the set of its codewords. (This class covers codes well known in the theory of self-checking discrete device design.) In addition, it is necessary to develop a method for extracting a particular subset of codewords of a weighted sum code whose codewords have desired diagnostic properties (e.g., capabilities to detect faults with given multiplicities or to implement Boolean functions from special classes in a CED circuit), a method for modifying the encoder of this code to work with a particular subset of its codewords, and a CED circuit design algorithm based on BSC and a particular subset of codewords of this code.

Note that in a special case, the object under diagnosis can be a subcircuit of the device $F(X)$ whose outputs have certain properties of error propagation under internal structure faults, being controllable for the chosen CED circuit organization using a particular subset of codewords of a weighted sum code.

3. THE ORGANIZATION STRUCTURE OF A CED CIRCUIT

In a CED circuit based on BSC, the signals coming from an object under diagnosis $F(X)$ are converted by a signal correction block (SCB) so as to obtain a codeword of a given code (Fig. 1). For correction, two-input mod 2 adders (XORs) are used; they uniformly generate 0 and 1 when all sets of argument values are supplied to the inputs and can correct any signal to 0 and 1. The first inputs of the gates receive signals $f_1(X), \ldots, f_n(X)$ from the object $F(X)$, and the second inputs receive signals from the block $G(X)$, which computes the values of correction functions $g_1(X), \ldots, g_n(X)$, predetermined for each set $< X >$. The conversion is given by

$$h_i(X) = f_i(X) \oplus g_i(X), \quad i = \overline{1,n}. \tag{1}$$

Initially, at the CED circuit design stage, the values of the functions $g_1(X), \ldots, g_n(X)$ are not specified. The main task is to obtain their values on each set of argument values. For organizing computation control, the type of functions computed at SCB outputs (the functions $h_1(X), \ldots, h_n(X)$) is important. According to (1), for known values of the functions $f_1(X), \ldots, f_n(X)$, the values of the functions $g_1(X), \ldots, g_n(X)$ are uniquely determined by the values of the functions $h_1(X), \ldots, h_n(X)$:

$$g_i(X) = f_i(X) \oplus h_i(X), \quad i = \overline{1,n}. \tag{2}$$

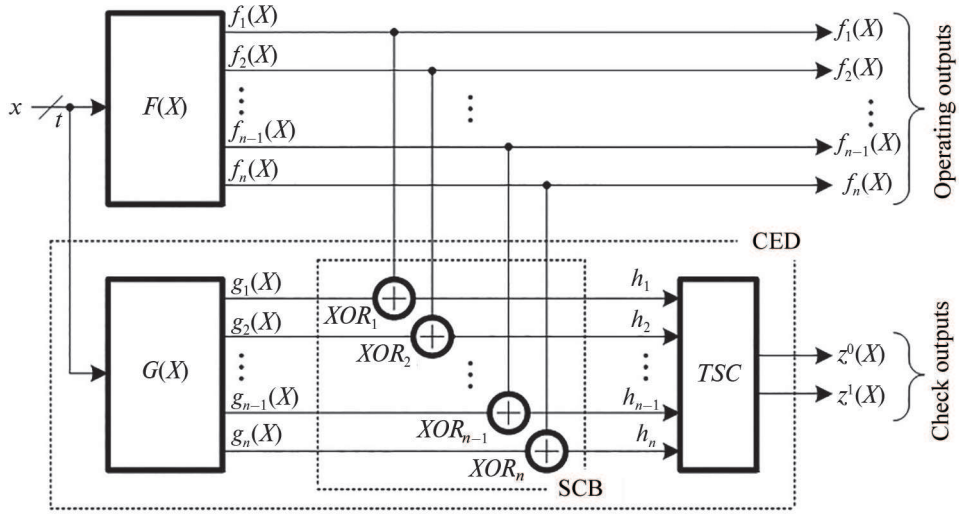


Fig. 1. The block diagram of a CED circuit based on BSC.

The methods for determining the values of the functions $h_1(X), \ldots, h_n(X)$ depend on the binary redundant code chosen for computation control. These methods are categorized as heuristic (a complement on each set of argument values considered sequentially, under the existing constraints on this procedure due to the requirements for generating test combinations for SCB gates and the checker [15]) and functional (an initially established relationship between the values of the functions $g_1(X), \ldots, g_n(X)$ and $f_1(X), \ldots, f_n(X)$ before the CED circuit design stage [16]). Thus, when supplying each set of argument values $< x_t x_{t-1} \ldots x_2 x_1 >$ to the device inputs, a codeword belonging to the chosen code is formed at SCB outputs; this is checked in the CED circuit by the totally self-checking checker (TSC) [17]. As the "watch dog," this checker has two outputs $z^0(X)$ and $z^1(X)$. In the absence of errors at its inputs and internal faults, the checker generates the two-rail signal $< 01 >$ or $< 10 >$. Violation of the two-rail nature indicates the presence of errors in computations and, indirectly, the presence of faults in the object under diagnosis or the CED circuit itself.

The difference between the above approaches to CED circuit design lies precisely in the fact that the conventional approach implies no correction but complements the signals generated at the outputs of the block $F(X)$ with check signals computed by an additional block in the CED circuit. Therefore, the methods widely used in the conventional approach take into account special properties of signal propagation to the device outputs under faults, e.g., the unidirectionality of occurring errors [18–20] or unidirectionality and their asymmetry (inequality in the number of corrupted 0s and 1s) [21, 22]. For the block diagram presented in Fig. 1, it is impossible to consider the type of error occurring at the object's output directly; therefore, computation control methods based on isolating subsets of independent [23] and r-independent outputs (outputs at which errors with multiplicities $d < r$ are allowed [24]) can be effective here.

Within the conventional approach to CED circuit design for a selected error-detecting code, there exists only one option to form the functions computed by control devices. (In other words, it is impossible to build various CED circuits, except for different implementations of the additional block for computing check functions and the checker.) In the case of BSC for a selected error-detecting code, there are numerous variants to build a CED circuit, determined only by the resulting codeword, i.e., the conversion of a particular codeword $< f_n(X) f_{n-1}(X) \ldots f_2(X) f_1(X) > = < F >$ on each set of argument values $< x_t x_{t-1} \ldots x_2 x_1 >$. Owing to the large number of variants to build a CED circuit, the self-checking property can be ensured in practice by choosing an appropriate

implementation. For example, BSC allows building self-checking devices in several cases where this property cannot be achieved by conventional approaches with duplication and parity checking [11].

The following peculiar constraints are imposed on the design procedure with BSC:

(1) For a complete check of *TSC*, when supplying sets of argument values to the inputs, it is necessary to form all codewords of a given code that belong to the set of combinations making up the check test. (This is not always the complete set of codewords of the code under consideration, which is determined by the method for building *TSC* [17].)
(2) For a complete check of the SCB, the inputs of each gate must receive check tests when supplying sets of argument values to the inputs.
(3) The devices $F(X)$ and $G(X)$ must be checking (self-testing), which requires each fault to be manifested as a distortion of their output values on at least one set of argument values [25].

The check test is determined according to the structures of the devices in a CED circuit. The SCB involves XORs; for their canonical implementation, a complete check requires the use of all four combinations $\{00, 01, 10, 11\}$ [26]. In the noncanonical implementation case, a smaller number of test combinations may be taken, i.e., one of the following sets of combinations: $\{00, 01, 11\}$, $\{00, 01, 10\}$, $\{00, 10, 11\}$, and $\{01, 10, 11\}$ [27]. The set of test combinations necessary to check the checker in a CED circuit depends on its implementation. For example, for separable codes and an implementation in the form of an encoder and a comparator, it is required to generate the complete set of check vectors when supplying given sets of argument values to the inputs.

Application of BSC in CED circuit design implies computation control of the functions implemented at SCB outputs via a predetermined diagnostic attribute. For example, in the case of BSC, the self-duality of the functions [28] or the belonging of the generated codewords to preselected constant-weight codes [10, 11] can be controlled. Both of these diagnostic attributes were combined in [29].

When designing a CED circuit based on BSC, it is unnecessary to correct all signals from an object under diagnosis. In some cases, it suffices to convert only part of them. For example, in the case of a 2-out-of-4 constant-weight code, the object's outputs are divided into groups of four, and only two signals in each group are converted in the SCB [30]. However, converting all object's signals gives the designer greater flexibility in building self-checking devices, including the ability to choose the best implementation method. Let us demonstrate this below, using CED circuit design based on BSC and weighted sum codes as an example.

4. WEIGHTED SUM CODE IN THE RESIDUE RING OF A GIVEN MODULUS

Weighted sum codes represent a large class of uniform block codes that can be effectively used in the design of self-checking, controllable, and fault-tolerant automation and computing devices [8]. There are many variants to construct them under given redundancy constraints. The codewords of weighted sum codes are obtained by the following algorithm.

Algorithm 1 (the rules for forming codewords of weighted sum codes).

(1) Choose and fix m as the number of data symbols of the code. Order and combine them into the data vector $< f_m(X)f_{m-1}(X)\dots f_2(X)f_1(X) >$, where $f_i(X) \in \{0, 1\},\ i = \overline{1, m}$.
(2) Fix an array of weights $[w_m, w_{m-1}, \dots, w_2, w_1]$ assigned to the bits of the data vector, where $w_i \in \mathbb{N},\ i = \overline{1, m}$.
(3) Specify a modulus $M \in \mathbb{N},\ M \geqslant 2$.
(4) For each data vector, determine the smallest nonnegative residue for the sum of the weights of the significant bits:

$$W_M = W(\mathrm{mod}M) = \left(\sum_{i=1}^{m} w_i f_i(X)\right)(\mathrm{mod}M). \tag{3}$$

(5) Represent the resulting number in binary form and write it in $k = \lceil \log_2 M \rceil$ bits of the check vector.

(6) Concatenate the check vector to the data vector and write it in the lower bits of the codewords.

Hereinafter, we will designate weighted sum codes by $WS(m, k, M)$-codes, separately indicating the array of the weights $[w_m, w_{m-1}, \ldots, w_2, w_1]$ assigned to the bits of the data vector.[1]

An arbitrary modulus can be taken to construct a weighted sum code; meanwhile, special properties are observed for the codes with moduli $M \in \left\{2^1, 2^2, \ldots, 2^{\lceil \log_2(\sum_{i=1}^{m} w_i + 1)\rceil}\right\}$. For these codes, when considering the complete set of data vectors, all possible check vectors with k bits are generated at least once. (Their set has cardinality 2^k.) Therefore, the self-checking property of the checkers of such codes can be easily ensured.

Moreover, for definite values of the weights $[w_m, w_{m-1}, \ldots, w_2, w_1]$ and the moduli $M \in \left\{2^1, 2^2, \ldots, 2^{\lceil \log_2(\sum_{i=1}^{m} w_i + 1)\rceil}\right\}$, it is possible to achieve an important property from the standpoint of using $WS(m, k, M)$-codes in CED circuit design, i.e., the uniform distribution of the complete set of data vectors (of cardinality 2^m) among all check vectors with k bits from their complete set. With this property, it is easier to ensure the self-checking property of the CED circuit.

In CED circuit design, $WS(m, k, M)$-codes with various parameters can be used. The choice of code parameters is determined by the individual features of the object under diagnosis. If the detecting characteristics of the code allow covering the complete set of errors at the object's outputs, then a code with $n = m + k$ bits in codewords can be chosen. If complete error coverage is unachievable, separate codes are used to control subsets of outputs. This method of computation control in a CED circuit has other advantages over control of the complete set of outputs as well: it is much easier to ensure the self-checking property of individual control devices, and the checkers of the codes have simpler structures.

Below, we describe a CED circuit design method with BSC and $WS(m, k, M)$-codes that is based on correcting all signals from the object under diagnosis but uses a particular subset of codewords of the code instead of all codewords. This method makes it possible to fix the multiplicities of detectable errors at the object's outputs and adjust the structural redundancy and controllability indicators of CED circuits.

5. EXTRACTING A PARTICULAR SUBSET OF CODEWORDS OF A WEIGHTED SUM CODE FOR CED CIRCUIT DESIGN

To extract a particular subset of codewords of the $WS(m, k, M)$-code for CED circuit design, recall that all 2^k check vectors must be generated at least once. Otherwise, it will be impossible to ensure a complete check of the checker of the $WS(m, k, M)$-code. In view of the aforesaid, we propose the following algorithm for extracting a particular subset of codewords of the $WS(m, k, M)$-code for CED circuit design.

Algorithm 2 (the generalized algorithm for selecting a particular subset of codewords of the $WS(m, k, M)$-code).

(1) Fix the parameters of the $WS(m, k, M)$-code and the array of weights $[w_m, w_{m-1}, \ldots, w_2, w_1]$ assigned to data vector bits.

(2) Generate the complete set of codewords of the $WS(m, k, M)$-code of cardinality 2^m.

(3) Classify the codewords of the $WS(m, k, M)$-code into M subsets corresponding to the numbers W_M.

[1] More precisely, they can be called codes with summation of the weights of data vector bits in the residue ring of a given modulus.

(4) Form an M-partite graph (for codes with the uniform distribution of data vectors among check vectors, a Turán graph $T(2^m, M)$):
   (a) Assign to the vertices of each part the codewords corresponding to the numbers W_M (3). Then each part of the graph will contain the codewords with the same check vector.
   (b) Assign to the edges connecting the ith and jth vertices (v_i and v_j, respectively) the weights d_{ij} equal to the Hamming distance between the corresponding codewords.
(5) On the graph $T(2^m, M)$, select the maximal cliques (their size will be M).
(6) Determine a criterion for selecting a clique among the maximal ones.
(7) Select the cliques satisfying this criterion among the maximal cliques.
(8) Take an arbitrary clique among the remaining maximal ones.
(9) The codewords corresponding to the vertices in the selected maximal clique form a particular subset of codewords of the $WS(m, k, M)$-code for CED circuit design.

Let us demonstrate Algorithm 2 on an example of selecting a particular subset of codewords of the $WS(4, 2, 4)$-code with the array of weights $[w_4, w_3, w_2, w_1] = [1, 1, 2, 3]$; it can be used to build the "basic" CED circuit structure for groups of $n = 6$ outputs. For example, its analog with the conversion of some signals from the object under diagnosis, responsible for forming the data bits of the $WS(4, 2, 4)$-code, was considered in [31].

We generate all codewords of the $WS(4, 2, 4)$-code and classify them into groups corresponding to the numbers W_4 (Table 1). Next, we form a four-partite graph (in this case, a Turán graph $T(16, 4)$; see Fig. 2).

Table 1. Classification of codewords of the $WS(4, 2, 4)$-code

W_4			
0	1	2	3
g_2g_1			
00	01	10	11
Codewords			
0000 00	0011 01	1011 10	1010 11
0101 00	0100 01	0111 10	0110 11
1001 00	1000 01	0010 10	0001 11
1110 00	1101 01	1100 10	1111 11

In a Turán graph, all 2^m vertices are split into M parts of equal size $l = \frac{2^m}{M}$. Since $M \in \left\{2^1, 2^2, \ldots, 2^{\lceil \log_2(\sum_{i=1}^m w_i + 1)\rceil}\right\}$, the graph is regular ($2^m$ is divisible by M). Each vertex of the graph $T(2^m, M)$ has degree

$$\deg(v) = 2^m - \frac{2^m}{M} = 2^m\Big(1 - \frac{1}{M}\Big). \tag{4}$$

In the case under consideration, the graph $T(16, 4)$, each of the 16 vertices has degree $2^4\Big(1 - \frac{1}{4}\Big) = 12$.

The number of edges in a graph $T(2^m, M)$ for $WS(m, k, M)$-codes is given by

$$N_{\text{ver}} = \frac{(M-1)(2^m)^2}{2M} = \frac{M-1}{M}2^{2m-1}. \tag{5}$$

For the $WS(4, 2, 4)$-code, we have $\frac{4-1}{4}2^{2\cdot 4-1} = \frac{3}{4}2^7 = 96$.

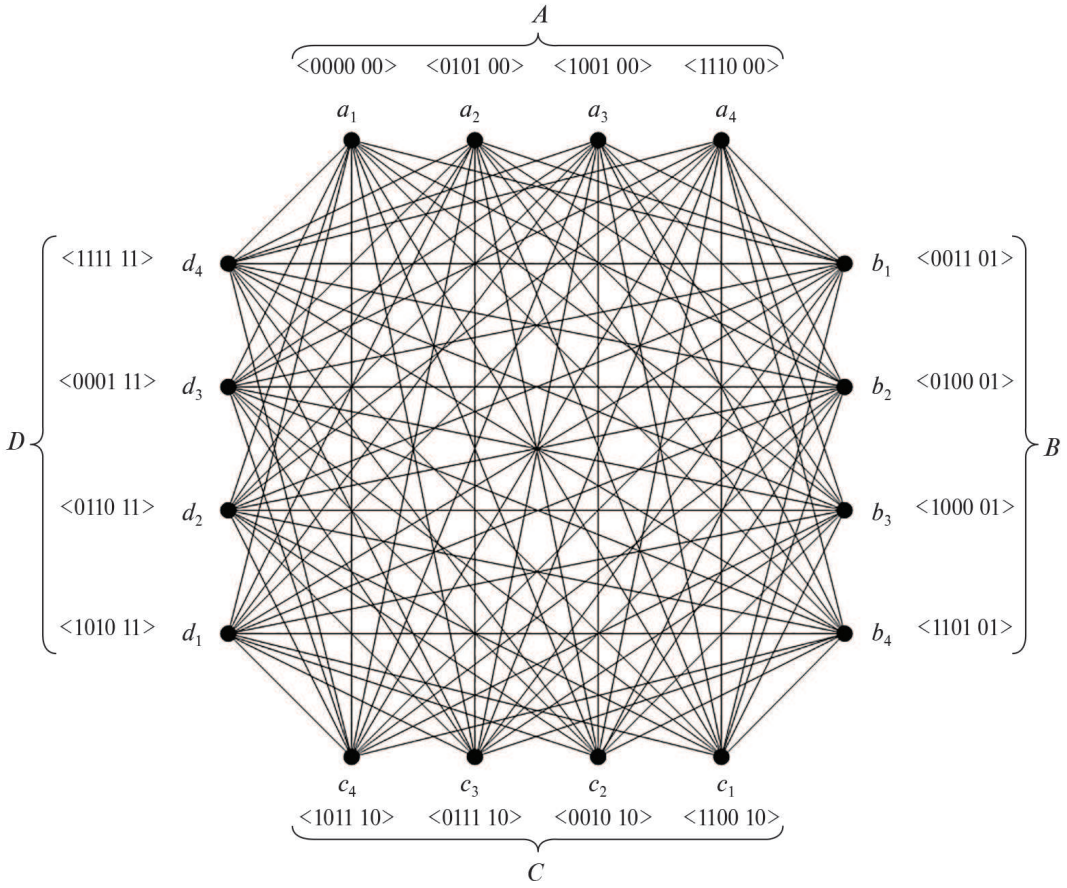


Fig. 2. The Turán graph $T(16,\ 4)$ for the $WS(4,\ 2,\ 4)$-code with the array of weights $[w_4, w_3, w_2, w_1] = [1,\ 1,\ 2,\ 3]$.

Next, it is necessary to extract all maximal cliques from the resulting graph $T(2^m, M)$. Their size equals M, and the number of edges is given by

$$R = \frac{M(M-1)}{2}. \tag{6}$$

For example, in the maximal cliques of the Turán graph built for the $WS(4, 2, 4)$-code, there are $\frac{4(4-1)}{2} = 6$ edges.

Each maximal clique corresponds to a particular subset of codewords of the $WS(m, k, M)$-code for CED circuit design. The number of maximal cliques in a graph $T(2^m, M)$ depends on the number of vertices in each part $(\frac{2^m}{M})$:

$$N_C = \Big(\frac{2^m}{M}\Big)^M. \tag{7}$$

For the $WS(4, 2, 4)$-code, we have $(\frac{2^4}{4})^4 = 2^8 = 256$.

Among the set of maximal cliques, it is necessary to select those with definite properties (satisfying a desired criterion).

The choice of an appropriate maximal clique will be associated with fixing certain values of the Hamming distance between the corresponding codewords. If the structure of the object under diagnosis (i.e., the configurations of gates and their connections to the device outputs) is unknown, then a reasonable criterion is the maximum total weight of the edges under the maximum shift of

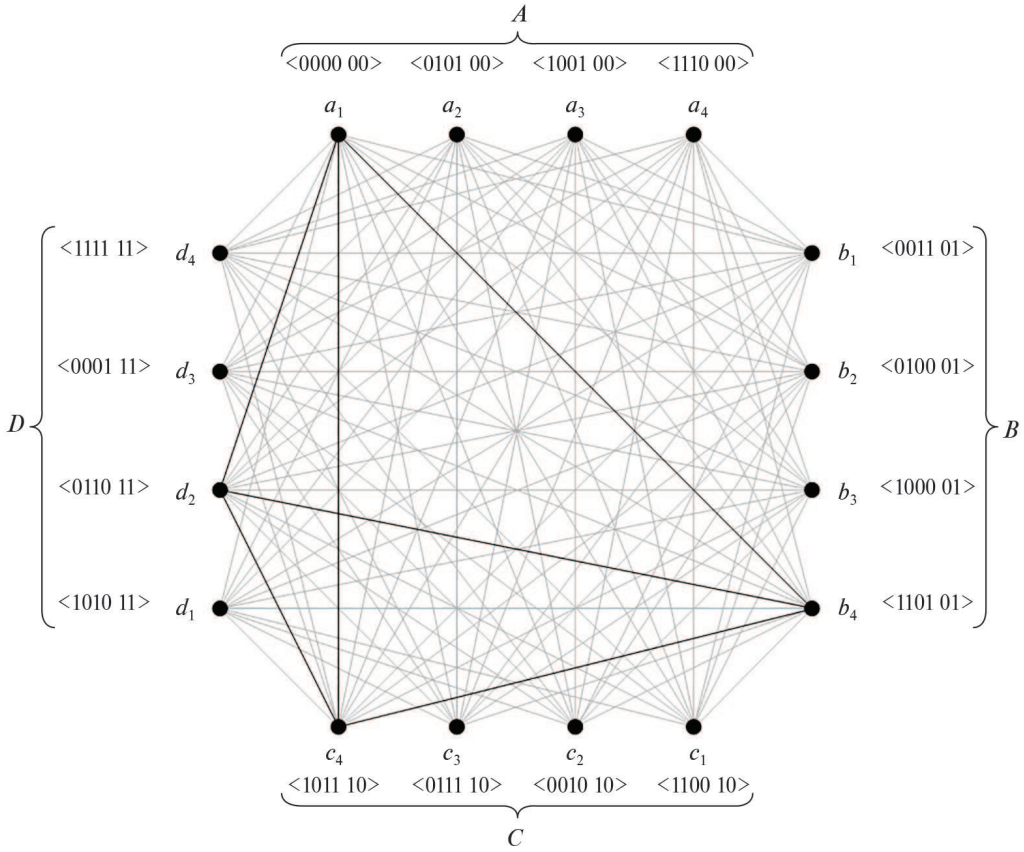


Fig. 3. Extracting a maximal clique in the Turán graph $T(16,\ 4)$ for the $WS(4, 2, 4)$-code with the array of weights $[w_4,\ w_3,\ w_2,\ w_1] = [1,\ 1,\ 2,\ 3]$.

the absolute values of the edge weights upward:

$$D_C = \sum_{v_i, v_j} d_{ij}, \quad d_{ij} \to \max, \tag{8}$$

where C is an expression containing all vertices of the clique (for large cliques, simply the clique number after ordering them), d_{ij} is the weight of the edge, and v_i and v_j are vertices belonging to this clique.

The maximum shift of absolute edge weights upward will be achieved if each weight of an edge in the maximal clique with total weight is as close as possible to

$$\gamma = \frac{D_C}{R} = \frac{2\sum_{v_i, v_j} d_{ij}}{M(M-1)}. \tag{9}$$

In the case of the $WS(4, 2, 4)$-code, formula (9) gives $\gamma = \frac{D_C}{6}$.

Note that the problem considered here (selecting a particular subset of codewords of the $WS(m, k, M)$-code for CED circuit design) is similar to the problem of selecting controllable probabilistic tests addressed, e.g., in [32, 33].

There are 256 maximal cliques in a graph $T(2^m, M)$. After an exhaustive search of all these, we identified a unique maximal clique for which all edge weights equal 4, and the total weight of the clique is 24. ($D_{a_1,b_4,c_4,d_2} = 24$ and $\gamma = \frac{24}{6} = 4$.) The clique includes vertices a_1, b_4, c_4, and d_2 (Fig. 3). The particular subset of codewords of the $WS(4, 2, 4)$-code corresponding to this clique is $\{< 0000\ 00 >, < 1101\ 01 >, < 1011\ 10 >, < 0110\ 11 >\}$. They will be used below to design of a CED circuit based on BSC with the $WS(4, 2, 4)$-code.

There exist other maximal cliques with the value $D_C = 24$. For example, the same number is provided by extracting a maximal clique with vertices a_1, b_1, c_1, and d_4. However, for the codewords corresponding to this clique, four pairs will have code distances equal to 3, and two pairs equal to 6. The above criterion allows discarding all cases except those with the maximum shift of weights upward.

When selecting a maximal clique on a Turán graph for codewords of the $WS(m, k, M)$-code, the number of maximal cliques grows astronomically with m. Therefore, in practice, one can establish either particular subsets of codewords of $WS(m, k, M)$-codes with various properties (one-time) or a criterion for selecting a maximal clique corresponding to the desired conditions (not necessarily the maximum possible Hamming distance between codewords). For example, such a condition is to select a maximal clique with all edge weights $d_{ij} \geqslant 3$.

Some $WS(m, k, M)$-codes can be used in CED circuit design based on two diagnostic attributes, i.e., the belonging of codewords to a preselected code and the self-duality of functions [34]. Not all $WS(m, k, M)$-codes possess this property. The $WS(4, 2, 4)$-code with the array of weights $[w_4, w_3, w_2, w_1] = [1, 1, 2, 3]$ (see above) is a representative of the class of codes with check symbols described by self-dual Boolean functions. Such codes have the following feature: on the complete set of their codewords, it is possible to select 2^{m-1} pairs with words orthogonal over all bits. Direct analysis of Table 1 shows that the $WS(4, 2, 4)$-code possesses this property. When building a CED circuit based on BSC with computation control via two diagnostic attributes, this fact can be utilized to establish a criterion for extracting a particular subset of codewords. For example, consider the Turán graph for the $WS(4, 2, 4)$-code (Fig. 2) and select two different vertex pairs of the Turán graph $(a_{i_1}, b_{i_2}) \,\&\, (c_{i_3}, d_{i_3}) \vee (a_{i_1}, c_{i_3}) \,\&\, (b_{i_2}, d_{i_3}) \vee (a_{i_1}, d_{i_4}) \,\&\, (b_{i_2}, c_{i_3})$, where i_1, i_2, i_3, i_4 denote the vertex numbers in the corresponding parts of the graph $(1 - A, 2 - B, 3 - C, 4 - D)$, with the maximum edge weight $\nu_{ij} = 6$. Then, the selected codewords will allow computation control via two diagnostic attributes under definite conditions for generating codewords $< h_n(X)h_{n-1}(X) \dots h_2(X)h_1(X) >$. (On orthogonal input combinations over all arguments, it is necessary to fix codewords also orthogonal over all arguments.) An example of a maximal clique satisfying this criterion is the clique with vertices a_1, b_1, c_1, and d_4. The corresponding codewords are $\{< 0000\ 00 >, < 0011\ 01 >, < 1100\ 10 >, < 1111\ 11 >\}$.

6. BLOCK DESIGN FEATURES IN A CED CIRCUIT

In the CED circuit structure (Fig. 1), the SCB has a standard implementation of n XORs. The block $G(X)$ is essentially an encoder of the $WS(m, k, M)$-code, intended to compute the bits of its codewords when supplying sets of argument values to the inputs; and TSC is a device for recognizing the codewords of this code. In the presented implementation, one could use the standard structure of a checker of any separable code in the form of a cascade connection of the encoder $G(F)$ and the comparator $kTRC1$: the former generates a check vector from the values of the data vector, and the latter compares the bits of the check vector obtained directly from SCB outputs and the check vector generated by the encoder $G(F)$, converting k two-rail signals into one. The self-checking comparator is implemented in two-rail logic from standard modules for compressing two-rail signals (TRC, two-rail checkers) [35]; therefore, in this checker implementation, it is necessary to invert the bits of the check vector generated at SCB outputs. One could immediately design the block $G(X)$ considering the need to obtain not the check vector for the $WS(m, k, M)$-code but the inverted vector. This would be sufficient for the correct operation of the CED circuit. However, in this case, it becomes impossible to take into account a particular subset of codewords of the $WS(m, k, M)$-code instead of its complete set. For instance, any distortion converting a codeword of the $WS(m, k, M)$-code into a codeword of the same code, even when using some particular subset of codewords of this code at the CED circuit design stage, will not be detected by the checker: a two-rail signal will

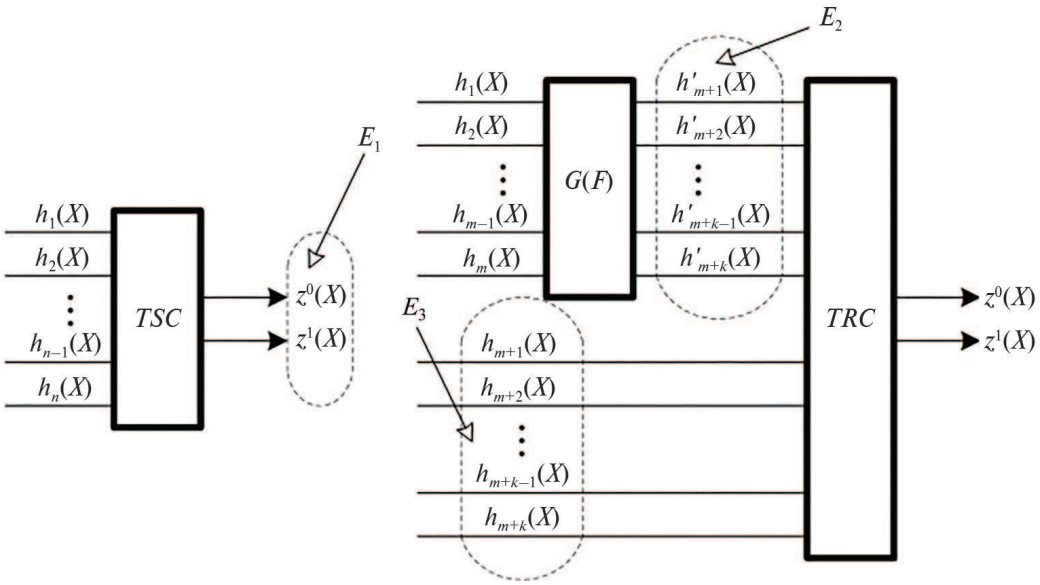


Fig. 4. Areas in the circuits of checkers of the $WS(4, 2, 4)$-code suitable for structural modifications.

be generated at its outputs. Therefore, to consider the error detection properties in the codewords of a particular subset of codewords of the $WS(m, k, M)$-code, the checker must be modified to "respond" only to the codewords preselected by the CED circuit designer.

Figure 4 shows the structures of checkers of the $WS(m, k, M)$-code and possible circuit nodes suitable for modifying signals on the checker lines, i.e., the areas E_1, E_2, and E_3. The checker can be implemented as a codeword detector: when codewords from a particular subset are received at its inputs, a two-rail signal is generated at the checker outputs; otherwise, a non-two-rail signal is generated. This modification area is E_1. The checker can be implemented as a two-stage structure consisting of an encoder and a comparator. In this case, there are two possible modification areas: modifying signals at the output of the encoders $G(F)$ and $G(X)$ (the areas E_2 and E_3, respectively).

Let us demonstrate possible structural modifications of the checker of the $WS(m, k, M)$-code on an example of the $WS(4, 2, 4)$-code with the array of weights $[w_4, w_3, w_2, w_1] = [1, 1, 2, 3]$ and the particular subset of codewords $\{< 0000\ 00 >, < 1101\ 01 >, < 1011\ 10 >, < 0110\ 11 >\}$.

Table 2 presents the options for fixing signals in the highlighted areas for modifying the control devices in the CED circuit.

Consider the option of designing the checker as a detector; then it is implemented in the form of a converter so that for each codeword belonging to the $WS(4, 2, 4)$-code, one of the two vectors from column E_1 is generated. Note that in the case of computation errors, the device $F(X)$ or the CED circuit blocks can generate a codeword not belonging to the $WS(4, 2, 4)$-code; hence, it is required to generate the values $< 00 >$ or $< 11 >$ for all codewords not belonging to this code. Accordingly, there exist $2^{2^6} = 2^{64}$ variants to specify the required values and build such a detector.

We have the following options for building a modified encoder of the $WS(4, 2, 4)$-code. When implementing the encoder $G(F)$ or the encoder $G(X)$, it is necessary to exclude the generation of a correct check vector when a codeword not belonging to the $WS(4, 2, 4)$-code arrives at the inputs. In the case under consideration, this can be done in three variants for each codeword unused. For codewords not belonging to the $WS(4, 2, 4)$-code, no special complement is required: if such a word appears, a mismatch between the data and check vectors will be recorded at SCB outputs. There are 3^{12} variants to specify the required values and build the block $G(F)$ and, accordingly, the block $G(X)$.

Table 2. Options for fixing signals under control device modifications in the CED circuit

Codewords of the code			Signals in modification areas	
Data vector	Check vector	Belonging to the particular subset of codewords	E_1	E_2, E_3
0000	00	+	01 / 10	00
0001	11	–	00 / 11	00 / 01 / 10
0010	10	–	00 / 11	00 / 01 / 11
0011	01	–	00 / 11	00 / 10 / 11
0100	01	–	00 / 11	00 / 10 / 11
0101	00	–	00 / 11	01 / 10 / 11
0110	11	+	01 / 10	11
0111	10	–	00 / 11	00 / 01 / 11
1000	01	–	00 / 11	00 / 10 / 11
1001	00	–	00 / 11	01 / 10 / 11
1010	11	–	00 / 11	00 / 01 / 10
1011	10	+	01 / 10	10
1100	10	–	00 / 11	00 / 01 / 11
1101	01	+	01 / 10	01
1110	00	–	00 / 11	01 / 10 / 11
1111	11	–	00 / 11	00 / 01 / 10

The trivial issue of the structural design of modified encoders, checkers, and detectors of $WS(m, k, M)$-codes (in particular, the $WS(4, 2, 4)$-code) goes beyond the scope of this paper.

7. PRINCIPLES FOR SPECIFYING VALUES AT SCB OUTPUTS

There are numerous variants to form codewords at SCB outputs. Even for a preselected particular subset of codewords, a large number of variants can be used to specify the values of the functions $h_1(X), \ldots, h_n(X)$ and, accordingly, $g_1(X), \ldots, g_n(X)$.

Generally, it is required to fix appropriate values of the functions $h_1(X), \ldots, h_n(X)$ on each of the 2^t sets of argument values in order to form codewords from a preselected particular subset of codewords of the $WS(m, k, M)$-code. In doing so, the above conditions for ensuring the self-checking property of the CED circuit blocks should be considered. Clearly, each check vector of the $WS(m, k, M)$-code can be generated at least once under the condition $t \geqslant M$; otherwise, this is impossible. If the condition holds, the BSC functions can be obtained by sequentially selecting codewords from the particular subset of codewords of the $WS(m, k, M)$-code for each row of the truth table that describes the object under diagnosis and the outputs of the CED circuit blocks. The particular subset of codewords has cardinality M, and on each of the 2^t rows, there are M options for obtaining codewords. However, the upper bound is M^{2^t} (while neglecting the generation of each unique codeword once). With the latter factor taken into account, on M rows we have to generate at least each codeword from the selected particular subset of codewords of the $WS(m, k, M)$-code, and on the remaining $2^t - M$ rows proceed arbitrarily. In total, there are $M^{2^t - M}$ variants to specify the values of the functions $h_1(X), \ldots, h_n(X)$ (accordingly, the same number of variants to design the CED circuit with a fixed particular subset of codewords). For example, for the $WS(4, 2, 4)$-code with $t = 4$, we have $4^{2^4 - 4} = 4^{12} = 16\,777\,216$ options for CED circuit design. Among these options, some may be less efficient, e.g., those not ensuring the self-checking property of the CED circuit

blocks or leading to high structural redundancy of the device (as compared to the redundancy of a self-checking implementation of the device by duplication or another method).

From the standpoint of ensuring the testability of CED circuit gates during the operation of a self-checking device, a method of interest is to specify appropriate values of the functions $h_1(X), \ldots, h_n(X)$ so that the probabilities of generating each test combination for the checker are close (if not equal). This can be achieved by uniformly distributing the codewords from a particular subset of codewords of the $WS(m, k, M)$-code among all 2^t input combinations. However, the uniform distribution of codewords does not guarantee the possibility of generating tests for SCB gates, but one can always return to the step of determining the values of the BSC functions and change their fixation method.

Algorithm 3 (the rules for designing CED circuits based on BSC with a fixed particular subset of codewords of the $WS(m, k, M)$-code).

(1) Order the codewords from a fixed particular subset of codewords of the $WS(m, k, M)$-code and number them by $q \in \{1, \ldots, 2^M\}$.
(2) Form a truth table with 2^t rows to describe the logic of the object under diagnosis (as noted above, the method works correctly if $t \geqslant M$.)
(3) Determine the so-called *repetition coefficient* of codewords:

$$\delta = \frac{2^t}{M}. \tag{10}$$

(4) On the rows of the truth table with decimal numbers (they are assigned decimals $0, \ldots, 2^{t-1}$, corresponding to the binary numbers written in the set of argument values) from the range $(q-1)\delta \ldots q\delta - 1$, fix the signals at SCB outputs so that the codeword with number q is generated.
(5) Obtain the values of the BSC functions using (2).
(6) Verify the condition for generating tests for all SCB gates. If a complete test cannot be generated, change the method for fixing the signals.
(7) Design the block $G(X)$.
(8) Select a method for modifying the control part of the CED circuit and modify one of the devices in the circuit.

Let us emphasize several important aspects in the operation of Algorithm 3. First, it is unnecessary to control the formation of the complete set of test combinations for the checker: all check vectors will be generated automatically. Second, a complete test for each SCB gate will not be formed in some cases (see Step 6 of Algorithm 3). If this happens, the variant for determining the values should be changed. One could initially focus on the row-by-row analysis of the tables describing the operation of devices, considering the generation of test combinations for gates, as done in [36]; in this case, however, the algorithm will run longer. (This is the main vulnerability of the algorithm.)

We demonstrate the operation of Algorithm 3 to obtain a description of a CED circuit on each line using the $WS(4, 2, 4)$-code with the array of weights $[w_4, w_3, w_2, w_1] = [1, 1, 2, 3]$ and a particular subset of its codewords $\{< 0000\ 00 >, < 1101\ 01 >, < 1011\ 10 >, < 0110\ 11 >\}$ for control of the device specified by Table 3.

In the CED circuit (see Fig. 1), the SCB has a standard implementation. It is required to design $G(X)$ and *TSC*.

Following the steps of Algorithm 3, we obtain the values of the functions describing the device $G(X)$. After ordering, the codewords are assigned the numbers: 1—$< 0000\ 00 >$, 2—$< 1101\ 01 >$, 3—$< 1011\ 10 >$, and 4—$< 0110\ 11 >$. By formula (10), the repetition coefficient of codewords from the particular subset is $\delta = \frac{2^4}{4} = 4$. Next, on rows with numbers $0, \ldots, 3$, we specify the

Table 3. The operation of a device with four outputs and the CED circuit for it

nos.	The sets of argument values				The values at the outputs of the device $F(X)$							The values at the outputs of the block $G(X)$							The values at the outputs of the SCB							Test combinations for SCB gates						
	x_4	x_3	x_2	x_1	f_6	f_5	f_4	f_3	f_2	f_1	g_6	g_5	g_4	g_3	g_2	g_1	h_6	h_5	h_4	h_3	h_2	h_1	XOR_6	XOR_5	XOR_4	XOR_3	XOR_2	XOR_1
0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	00	11	11	11	00	00
1	0	0	0	1	0	1	0	0	1	1	0	1	0	0	1	1	0	0	0	0	0	0	00	11	00	00	11	11
2	0	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	0	0	0	0	0	0	00	00	11	11	00	11
3	0	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	11	00	00	00	00	00
4	0	1	0	0	1	1	0	0	0	1	0	0	0	1	0	0	1	1	0	1	0	1	10	10	00	01	00	10
5	0	1	0	1	0	0	1	0	0	1	1	1	1	1	0	0	1	1	0	1	0	1	01	01	11	01	00	10
6	0	1	1	0	1	0	0	1	0	0	0	1	0	0	0	1	1	1	0	1	0	1	10	01	00	10	00	01
7	0	1	1	1	0	1	1	1	1	1	1	0	1	0	1	0	1	1	0	1	0	1	01	10	11	10	11	10
8	1	0	0	0	1	0	1	0	0	0	0	0	0	1	1	0	1	0	1	1	1	0	10	00	10	01	01	00
9	1	0	0	1	0	0	1	0	1	0	1	0	0	1	0	0	1	0	1	1	1	0	01	00	10	01	10	00
10	1	0	1	0	1	0	0	1	1	0	0	0	1	0	0	0	1	0	1	1	1	0	10	00	01	10	10	00
11	1	0	1	1	1	0	0	1	0	0	0	0	1	0	1	0	1	0	1	1	1	0	10	00	01	10	01	00
12	1	1	0	0	1	1	0	0	0	0	1	0	1	0	1	1	0	1	1	0	1	1	11	10	01	00	01	01
13	1	1	0	1	0	1	1	0	0	1	0	0	0	0	1	0	0	1	1	0	1	1	00	10	10	00	01	10
14	1	1	1	0	0	0	1	1	0	0	0	1	0	1	1	1	0	1	1	0	1	1	00	01	10	11	01	01
15	1	1	1	1	0	0	0	1	0	0	0	1	1	1	1	1	0	1	1	0	1	1	00	01	01	11	01	01

signals at SCB outputs so that codeword 1 is formed; on rows with numbers $4, \ldots, 7$, codeword 2; on rows with numbers $8, \ldots, 11$, codeword 3; finally, on rows with numbers $12, \ldots, 15$, codeword 4. Using (2), we determine the values of the BSC functions; see the results in Table 3, where the last six columns present the combinations generated at the inputs of the gates. Direct analysis of these columns indicates that a test combination is formed for each gate.

Next, we select a method for modifying the control part of the CED circuit and design the self-checking device. Here, the issues of structural design and efficiency indicators of the resulting CED circuit are omitted: the presentation is intended to show only the fairly simple and understandable principles of implementing a self-checking device.

According to experimental studies on a variety of examples, in a large number of cases, Algorithm 3 yields less redundant self-checking combinational discrete devices compared to duplication. The effect is higher for more complex initial functions computed by the block $F(X)$. When designing self-checking finite state machines (FSMs) with CED circuits for logic and output converters, this effect only increases since duplication of all Boolean blocks of FSMs is not required.

8. CONCLUSIONS

This paper has described the features of the authors' method for designing self-checking devices based on BSC in a CED circuit with a particular subset of codewords of the $WS(m, k, M)$-code and conversion of all signals from an object under diagnosis.

The advantages of this CED circuit design method are obvious. First, by choosing a particular subset of codewords of the $WS(m, k, M)$-code, it is possible to consider the characteristics of errors occurring at the object's outputs. Second, there are numerous variants to specify the values of the functions formed at SCB outputs, which provides "flexibility" and the ability to choose the CED circuit structure during its design. Third, it is easy to modify the control part of the CED circuit so that it detects any errors except those distorting codewords from the selected particular subset of codewords of the $WS(m, k, M)$-code into each other. In fact, this method is closely related to those involving inseparable codes in CED circuit design with BSC [10, 11, 15, 16, 30], and the above methods for modifying the control circuits are based on modifying the $WS(m, k, M)$-code.

Note the following drawbacks of the CED circuit design method. The most serious one, in the authors' opinion, is the necessity to ensure the self-checking property of SCB gates and the checker of the $WS(m, k, M)$-code. This cannot be done for all devices. For example, all SCB gates are checked only if each of the functions computed by the object under diagnosis takes value 1 (or 0) on at least two sets of argument values [31]. With a small number of the object's inputs, this is not always feasible. The second drawback is that, to ensure a complete check, a definite subset of argument values has to be supplied during the operation of the self-checking device. This cannot be done, e.g., in critical systems where input actions change infrequently [37], and a combination of test and functional diagnosis methods is then required [38]. Concerning the third drawback, the method itself requires the selection of a particular subset of codewords of the $WS(m, k, M)$-code, which (albeit being performed once) is difficult for large values of the code parameters m, k, M. In addition, there are numerous variants to specify the values of the functions at SCB outputs on each set of argument values; despite the possibility to choose a single variant for fixing their values, in many practical cases, an optimal one cannot be chosen due to the complexity (or even infeasibility) of an exhaustive enumeration of all variants.

From the authors' point of view, the above method for designing self-checking devices is of interest for implementation on modern programmable logic devices.

REFERENCES

1. Efanov, D.V., *Metody sinteza samoproveryaemykh diskretnykh ustroistv* (Design Methods for Self-Checking Discrete Devices), Moscow: LENAND, 2025.
2. Sagalovich, Yu.L., *Algebra, kody, diagnostika* (Algebra, Codes, Diagnosis), Moscow: Institute for Information Transmission Problems, Russian Academy of Sciences, 1993.
3. Fujiwara, E., *Code Design for Dependable Systems: Theory and Practical Applications*, John Wiley & Sons, 2006.
4. Goessel, M., Ocheretny, V., Sogomonyan, E., and Marienfeld, D., *New Methods of Concurrent Checking*, 1st ed., Dordrecht: Springer Science+Business Media B.V., 2008.
5. Drozd, A.V., Kharchenko, V.S., Antoshchuk, S.G., et al., *Rabochee diagnostirovanie bezopasnykh informatsionno-upravlyayushchikh sistem* (Operational Diagnosis of Safe Information and Control Systems), Khar'kov: Zhukovskii National Aerospace University, 2012.
6. Mikoni, S.V., Sokolov, B.V., and Yusupov, R.M., *Kvalimetriya modelei i polimodel'nykh kompleksov* (Qualimetry of Models and Polymodel Complexes), Moscow: Russian Academy of Sciences, 2018.
7. Sapozhnikov, V.V., Sapozhnikov, Vl.V., and Efanov, D.V., *Kody s summirovaniem dlya sistem tekhnicheskogo diagnostirovaniya. Tom 1: Klassicheskie kody Bergera i ikh modifikatsii* (Sum Codes for Technical Diagnosis Systems. Vol. 1: Classical Berger Codes and Their Modifications), Moscow: Nauka, 2020.
8. Sapozhnikov, V.V., Sapozhnikov, Vl.V., and Efanov, D.V., *Kody s summirovaniem dlya sistem tekhnicheskogo diagnostirovaniya. Tom 2: Vzveshennye kody s summirovaniem* (Sum Codes for Technical Diagnosis Systems. Vol. 2: Weight-Based Codes), Moscow: Nauka, 2021.
9. Das, D., Touba, N.A., Seuring, M., and Gossel, M., Low Cost Concurrent Error Detection Based on Modulo Weight-Based Codes, *Proceedings of the IEEE 6th International On-Line Testing Workshop (IOLTW)*, Spain, Palma de Mallorca, July 3–5, 2000, pp. 171–176. https://doi.org/10.1109/OLT.2000.856633
10. Gessel, M., Morozov, A.V., Sapozhnikov, V.V., et al., Logic Complement, a New Method of Checking the Combinational Circuits, *Autom. Remote Control*, 2003, vol. 64, no. 1, pp. 153–161.
11. Goessel, M., Morozov, A.V., Sapozhnikov, V.V., et al., Checking Combinational Circuits by the Method of Logic Complement, *Autom. Remote Control*, 2005, vol. 66, no. 8, pp. 1336–1346.
12. Sogomonyan, E.S. and Slabakov, E.V., *Samoproveryaemye ustroistva i otkazoustoichivye sistemy* (Self-Checking Devices and Fault-Tolerant Systems), Moscow: Radio i Svyaz', 1989.
13. Mitra, S. and McCluskey, E.J., Which Concurrent Error Detection Scheme to Choose?, *Proceedings of the International Test Conference*, Atlantic City, NJ, October 3–5, 2000, pp. 985–994. https://doi.org/10.1109/TEST.2000.894311
14. Efanov, D.V., The Equal-Length Redundant Code Development for the Self-Checking Combinational Devices Synthesis Based on Data on Their Structures, *Electronic Modeling*, 2022, vol. 44, no. 1, pp. 43–52. https://doi.org/10.15407/emodel.44.01.043
15. Sapozhnikov, V., Sapozhnikov, Vl., and Efanov, D., Concurrent Error Detection of Combinational Circuits by the Method of Boolean Complement on the Base of "2-out-of-4" Code, *Proceedings of 14th IEEE East-West Design & Test Symposium (EWDTS'2016)*, Yerevan, Armenia, October 14–17, 2016, pp. 126–133. https://doi.org/10.1109/EWDTS.2016.7807677
16. Sapozhnikov, V.V., Sapozhnikov, Vl.V., and Efanov, D.V., Formation of Totally Self-Checking Structures of Concurrent Error Detection Systems with Use of Constant-Weight Code "1-Out-Of-3", *Electronic Modeling*, 2016, vol. 38, no. 6, pp. 25–43.
17. Sapozhnikov, V.V. and Sapozhnikov, Vl.V., *Samoproveryaemye diskretnye ustroistva* (Self-Checking Discrete Devices), St. Petersburg: Energoatomizdat, 1992.
18. Sogomonyan, E.S. and Gossel, M., Design of Self-Testing and On-Line Fault Detection Combinational Circuits with Weakly Independent Outputs, *Journal of Electronic Testing: Theory and Applications*, 1993, vol. 4, no. 4, pp. 267–281. https://doi.org/10.1007/BF00971975

19. Busaba, F.Y. and Lala, P.K., Self-Checking Combinational Circuit Design for Single and Unidirectional Multibit Errors, *Journal of Electronic Testing: Theory and Applications*, 1994, vol. 5, no. 5, pp. 19–28. https://doi.org/10.1007/BF00971960

20. Morosow, A., Saposhnikov, V.V., Saposhnikov, Vl.V., and Goessel, M., Self-Checking Combinational Circuits with Unidirectionally Independent Outputs, *VLSI Design*, 1998, vol. 5, no. 4, pp. 333–345. https://doi.org/10.1155/1998/20389

21. Efanov, D.V., Sapozhnikov, V.V., and Sapozhnikov, V.V., Conditions for Detecting a Logical Element Fault in a Combination Device under Concurrent Checking Based on Berger's Code, *Autom. Remote Control*, 2017, vol. 78, no. 5, pp. 891–901.

22. Efanov, D.V., Sapozhnikov, V.V., and Sapozhnikov, Vl.V., Organization of a Fully Self-Checking Structure of a Combinational Device Based on Searching for Groups of Symmetrically Independent Outputs, *Automatic Control and Computer Sciences*, 2020, vol. 54, no. 4. pp. 279–290. https://doi.org/10.3103/S0146411620040045

23. Aksenova, G.P. and Sogomonyan, E.S., Design of Self-Checking Built-In Check Circuits for Automata with Memory, *Autom. Remote Control*, 1975, vol. 36, no. 7, pp. 1169–1177.

24. Efanov, D.V., Synthesis of Self-Checking Computing Devices Based on a Complete System of Special Groups of the Diagnostic Object Outputs, *Journal of Instrument Engineering*, 2023, vol. 66, no. 5, pp. 355–372. https://doi.org/10.17586/0021-3454-2023-66-5-355-372

25. Parkhomenko, P.P. and Sogomonyan, E.S., *Osnovy tekhnicheskoi diagnostiki. Tom 2: Optimizatsiya algoritmov diagnostirovaniya, apparaturnye sredstva* (Foundations of Technical Diagnosis. Vol. 2: Optimization of Diagnostic Algorithms, Hardware Means), Moscow: Energoatomizdat, 1981.

26. Aksenova, G.P., Necessary and Sufficient Conditions for the Synthesis of Completely Testable Modulo 2 Convolution Circuits, *Autom. Remote Control*, 1979, vol. 40, no. 9, pp. 1362–1369.

27. Yelina, Y.I. and Efanov, D.V., Weight-Based Bose–Lin Codes in Concurrent Error-Detection Circuit Based on Boolean Signal Correction, *Journal of Computer and Systems Sciences International*, 2025, vol. 64, no. 1, pp. 17–35.

28. Saposhnikov, Vl.V., Dmitriev, A., Goessel, M., and Saposhnikov, V.V., Self-Dual Parity Checking — A New Method for On-line Testing, *Proceedings of the 14th IEEE VLSI Test Symposium*, USA, Princeton, 1996, pp. 162–168. https://doi.org/10.1109/VTEST.1996.510852

29. Efanov, D.V. and Pivovarov, D.V., The Hybrid Structure of a Self-Dual Built-In Control Circuit for Combinational Devices with Pre-Compression of Signals and Checking of Calculations by Two Diagnostic Parameters, *Proceedings of the 19th IEEE East-West Design and Test Symposium (EWDTS'2021)*, Batumi, Georgia, September 10–13, 2021, pp. 200–206. https://doi.org/10.1109/EWDTS52692.2021.9581019

30. Sapozhnikov, V.V., Sapozhnikov, Vl.V., and Efanov, D.V., Design of Self-Checking Concurrent Error Detection Systems Based on "2-out-of-4" Constant-Weight Code, *Probl. Upr.*, 2017, no. 1, pp. 57–64.

31. Efanov, D.V. and Yelina, Y.I., Design of Self-Checking Digital Devices with Boolean Signals Correction Using Weight-Based Bose–Lin Codes, *Control Sciences*, 2024, no. 4, pp. 22–36. http://doi.org/10.25728/cs.2024.4.3

32. Yarmolik, S.V. and Yarmolik, V.N., The Synthesis of Probability Tests with a Small Number of Kits, *Automatic Control and Computer Sciences*, 2011, vol. 45, no. 3, pp. 133–141. https://doi.org/10.3103/S0146411611030072

33. Yarmolik, V.N., Petrovskaya, V.V., and Mrozek, I., A Measure of Differences for Test Sets in Reducing Controllable Random Tests, *Informatics*, 2022, vol. 19, no. 4, pp. 7–26. https://doi.org/10.37661/1816-0301-2022-19-4-7-26

34. Efanov, D.V. and Pivovarov, D.V., Design of Self-Checking Discrete Devices Based on Polynomial Codes with Computation Control via Several Diagnostic Attributes, *Autom. Remote Control*, 2025, vol. 86, no. 5, pp. 402–416.

35. Lala, P.K., *Self-Checking and Fault-Tolerant Digital Design*, San Francisco: Morgan Kaufmann Publishers, 2001.

36. Efanov, D.V., Synthesis of Self-Checking Combinational Devices Based on the Boolean Signal Correction Method Using Bose–Lin Codes, *Information Technologies*, 2023, vol. 29, no. 10, pp. 503–511. https://doi.org/10.17587/it.29.503-511

37. Drozd, A., Kharchenko, V., Antoshchuk, S., et. al., Checkability of the Digital Components in Safety-Critical Systems: Problems and Solutions, *Proceedings of the 9th IEEE East-West Design and Test Symposium (EWDTS'2011)*, Sevastopol, Ukraine, 2011, pp. 411–416. https://doi.org/10.1109/EWDTS.2011.6116606

38. Litikov, I.P. and Sogomonyan, E.S., Test and Functional Diagnosis of Digital Devices and Systems, *Autom. Remote Control*, 1985, vol. 46, no. 3, pp. 375–384.

This paper was recommended for publication by L.Yu. Filimonyuk, a member of the Editorial Board