ISSN 2782-2427

CONTROL SCIENCES

4/2025

[1]Russian Academy of Sciences.

CONTROL SCIENCES
Scientific Technical Journal

6 issues per year
ISSN 2782-2427
Open access

Published since 2021

Original Russian Edition
Problemy Upravleniya
Published since 2003

FOUNDER AND PUBLISHER
V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences

Editorial address
65 Profsoyuznaya st., office 410,
Moscow 117997, Russia

☎/🖷 +7(495) 198-17-20, ext. 1410

✉ pu@ipu.ru

URL: http://controlsciences.org

Published: September 24, 2025

Registration certificate of
Эл № ФС 77-80482
of 17 February 2021
issued by the Federal Service for Supervision of Communications, Information Technology, and Mass Media

CONTROL SCIENCES

4.2025

CONTENTS

Surveys

Bukov, V. N., Bronnikov, A. M., Popov, A. S., and Shurman, V. A. Technical Condition Monitoring Methods to Manage the Redundancy of Systems. Part III: Nonclassical Models in Fault Diagnosis 2

Systems Analysis

Kalyanov, G. N. and Novikov, D. A. Classification of Activity Processes 17

Mathematical Problems of Control

Kononov, D. A. and Furugyan, M. G. Managing a Complex of Jobs with Uncertain Execution Request Arrivals 25

Control in Social and Economic Systems

Tsyganov, V. V. Sustainable Development Mechanisms for a Transport Corporation with Supervised Learning 32

Sych, V. V. The Correlation Between the Social Network Activity and Academic Performance of Technical University Students: A Case Study of *VKontakte* 43

Control of Moving Objects and Navigation

Moscowsky, A. D. Application of Subdefinite Models in the Global Localization of a Mobile Robot 55

TECHNICAL CONDITION MONITORING METHODS TO MANAGE THE REDUNDANCY OF SYSTEMS. PART III: Nonclassical Models in Fault Diagnosis

V. N. Bukov*, A. M. Bronnikov**, A. S. Popov***, and V. A. Shurman****

*Airborne Aeronautical Systems, Moscow, Russia
**Bauman Moscow State Technical University, Moscow, Russia
***Zhukovsky–Gagarin Air Force Academy, Voronezh, Russia
****Institute of Aircraft Equipment, Zhukovsky, Russia

*✉ v_bukov@mail.ru, **✉ bronnikov_a_m@mail.ru, ***✉ saga30@yandex.ru, ****✉ shurman@niiao.ru

Abstract. Redundancy management in a technical system involves a monitoring procedure to reconfigure the system as needed. The four-part survey presents modern diagnosis methods for dynamic systems as an integral function of monitoring. Part III is devoted to diagnosis methods employing neural networks, fuzzy models, structural models, set-based models, and a statistical approach. The fundamentals of creating and training neural networks to perform diagnostic functions are considered. The approach with fuzzy models is described, including general modeling rules and the features of their use in diagnosis tasks. The approach with structural models is demonstrated, including its features in failure detection. The fundamentals of set-theory methods, particularly the formalism of zonotopes, are presented. Finally, the approach based on statistical pattern recognition is briefly discussed.

Keywords: artificial neuron, neural network, fuzzy models, membership function, fuzzy clustering, structural models, Dulmage–Mendelsohn decomposition, diagnosability, zonotopes, gradations in discrete feature space, empirical likelihood ratio estimate.

INTRODUCTION

The main limitation of approaches with analytical models is the need for precise knowledge of such models (both their structure and parameters), which is often impossible in practical applications. In the case of mathematical models with partial uncertainty, there exist well-developed procedures for their parametric or structural identification [1–3]. However, here the focus is on the "initial" elimination of this drawback (i.e., the one inherent in the approach itself). Such approaches include the use of neural networks, fuzzy models, structural models, the apparatus of set theory, and statistical estimates.

Proponents of such methods, deliberately or due to insufficient information, refrain from a detailed description of the processes occurring in a system diagnosed. In one way or another, this makes the models used rough (inaccurate, approximate) but provides greater flexibility in fault diagnosis, reducing or completely bypassing negative effects. That is why such approaches are applied for the effective monitoring of dynamic systems [4].

1. NEURAL NETWORK-BASED DIAGNOSIS METHODS

Neural networks (NNs) are widely used in fault diagnosis [5–9]. One modification of such networks [10, 11] uses *Nonlinear AutoRegressive with eXogenous inputs* (NARX) models in combination with learning algorithms.

1.1. Formation of an ARX Neural Network

One approach involves developing and training a set of NN-based estimators to reproduce the behavior of a system under consideration. The structure of the *i*th individual neuron [12] uses a *Multiple Input Single Output* (MISO) system in which the output signal y_i is calculated as a function of the weighted sum of all inputs u_i of the neuron, $u_{i.1},\dots,\ u_{i.n_i}$, with corresponding weights $w_{i.1},\dots,\ w_{i.n_i}$ (Fig. 1). The function f is called the activation function.

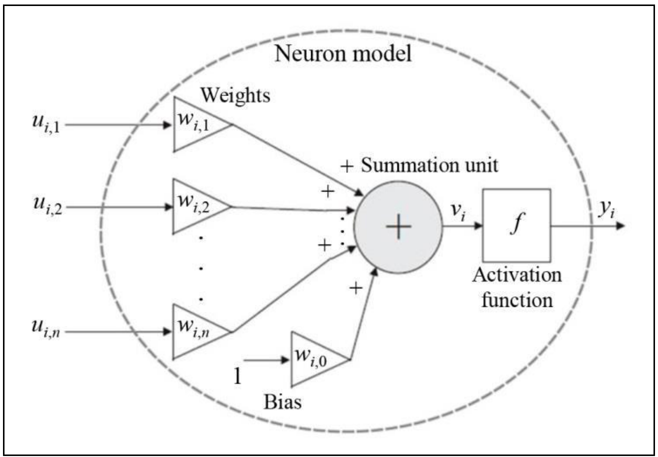


Fig. 1. One example of an artificial neuron.

NNs are classified according to the ways of connecting their elements [13, 14]. In a feedforward NN, the elements are grouped into unidirectional layers. The first (input) layer receives information directly from the network inputs; then, each sequential hidden layer receives input data from the neurons of the previous layer and transmits the output data to the neurons of the next layer, up to the last output layer, where the final network output data are generated. Therefore, neurons are connected layer-to-layer but not within the same layer. The only limitation is the number of neurons in the output layer, which must coincide with the number of actual output channels of the network.

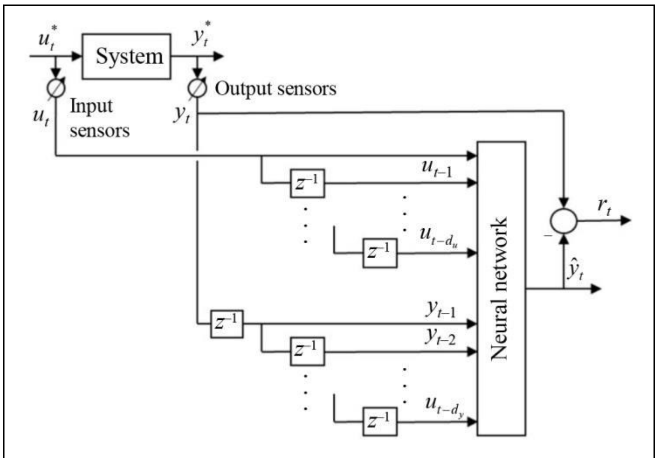


Fig. 2. An open-loop dynamic NN.

On the other hand, recurrent NNs [13] are multilayer networks in which the output data of some neurons is fed back to neurons of previous layers. Thus, information is transmitted both forward and backward and provides dynamic memory within the network.

An intermediate solution is an NN with a connected delay line. Networks of this type are suitable for modeling or predicting the evolution of a dynamic system. In particular, a properly trained open-loop NARX network can estimate current (or future) results based on past measurements of the system's input and output data.

Generally speaking, for *Multiple Input Multiple Output* (MIMO) systems, open-loop NARX networks follow the law

$$\hat{y}_t = f_{\text{net}}(u_t,\dots,u_{t-d_u},\ y_{t-1},\dots,\ y_{t-d_y}), \tag{1}$$

where $\hat{y}_t$ is the estimate of the system output at step t; u and y are the measured system inputs and outputs, respectively; d_u and d_y are the number of input and output delays, respectively; finally, f_{net} is a function implemented by the network that depends on the architecture of the layers, the number of neurons, their weights, and activation functions. Figure 2 shows the structure of an open-loop NARX network used as an estimator.

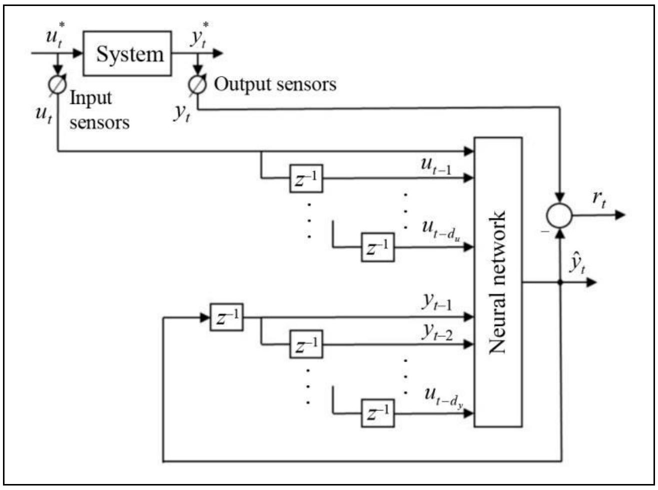


Fig. 3. A closed-loop dynamic NN.

The difference between the measured output y_t and its estimate (1) is taken as the diagnosis residual r_t.

When only input measurements are available, the NARX network can become a dynamic NN by closing the feedback loop of the network outputs to the inputs (Fig. 3).

The developer of a fault diagnosis system can vary the number of neurons and the connections between them, while the weights $w_{i.j}$ within each neuron are assigned by training the network.

Since NN components are often chosen subjectively, they may be incompletely adequate for the system or process modeled, which affects NN implementations.

1.2. Neural Network Training

An NN becomes a diagnosis tool after its training (assignment of the weights w) on pre-selected examples of the functioning of the system diagnosed, both in operable and inoperable states (in particular, due to the occurrence of a single fault or a combination of faults). Besides the basic network with fixed weights after training, there are modifications with refinement (retraining) of network parameters during its operation. In addition, an NN should be retrained each time, even for homogeneous objects operating in different conditions and with different historical operation data.

An NN can be trained only to the "diagnostic events" (normal, operable, pre-fault, and inoperable states) that are initially known and used in the training process. If the set of admissible states (including faulty ones) is not specified for an NN, it will not distinguish between them or find an unforeseen state.

The use of NNs allows performing self-diagnosis in practice, since a properly trained NN is robust to anomalies in data and recognizes a diagnostic event itself, even under inaccuracies in the source data.[1] This will not happen when using "hard" diagnosis algorithms.

The formal objective of training is to minimize the loss function (residual) E [14], which depends on the vector of weights w, and it can be implemented in two different modes:

– the incremental mode (each input-target pair independently generates an update of the network weights);

– the batch mode (all input data and the loss function are applied to the network simultaneously).

Although the second training mode requires more memory than the first, it is characterized by faster convergence and results with smaller errors.

Given a set of S patterns corresponding to both the absence and presence of various faults in the system, for each of $P = C_P^2 = S!/2(S-2)!$ possible input-output pairs, the error vector can be written as

$$e_p = y_p - \hat{y}_p = \begin{bmatrix} e_{p.1} & \cdots & e_{p.M} \end{bmatrix}^{\mathrm{T}},\ p = \overline{1, P}, \quad (2)$$

where M is the number of outputs (i.e., the residuals of the system diagnosed). The total loss function, which depends on the choice of all weights $w = (w_1, ..., w_N)$ (see formula (1) and Fig. 1), takes the form:

$$E(w) = \frac{1}{2}\sum_{p=1}^{P}\sum_{m=1}^{M}(y_{p.m} - \hat{y}_{p.m}(w))^2 = \frac{1}{2}\sum_{p=1}^{P}\sum_{m=1}^{M}e_{p.m}^2(w).$$

Any standard numerical optimization algorithm [14–16] can be applied to update the values of the parameters w_i in order to minimize the residual $E(w)$. Iterative algorithms are most widespread; they use such characteristics as:

– the gradient of the loss function,

$$\mathrm{grad}E = \frac{\partial E}{\partial w} = \begin{bmatrix} \dfrac{\partial E}{\partial w_1} & \cdots & \dfrac{\partial E}{\partial w_N} \end{bmatrix}^{\mathrm{T}};$$

– the Hessian of the loss function,

$$H(E) = \begin{bmatrix} \partial^2 E / \partial w_1^2 & \cdots & \partial^2 E / \partial w_1 \partial w_N \\ \vdots & \ddots & \vdots \\ \partial^2 E / \partial w_N \partial w_1 & \cdots & \partial^2 E / \partial w_N^2 \end{bmatrix};$$

– the Jacobian of the estimation errors,

$$J(e) = \begin{bmatrix} \partial e_{1.1} / \partial w_1 & \cdots & \partial e_{1.1} / \partial w_N \\ \partial e_{1.2} / \partial w_1 & \cdots & \partial e_{1.2} / \partial w_N \\ \vdots & \ddots & \vdots \\ \partial e_{P.M} / \partial w_1 & \cdots & \partial e_{P.M} / \partial w_N \end{bmatrix}.$$

Sequential iterations of these algorithms consist of updating the parameter values and calculating the new value of the loss function until the stopping condition is satisfied.

The achievable sensitivity of the diagnosis procedure to individual faults is a complex issue that is unlikely to have a generalized assessment.

Note finally that NNs approximate any nonlinear and dynamic function under a suitable structure of weights. Moreover, online training makes it easy to modify the diagnosis system when changes are made to the physical process or control system. An NN can generalize when available input data are not present in the training data and make reasonable decisions in cases of noisy or corrupted data. NNs are also easily applicable to multiparameter systems and have a high level of structural parallelism. An NN can work with both qualitative and quantitative data simultaneously. In addition, an NN can be very useful in the absence of any mathematical model of the system (i.e., when analytical models cannot be applied for some reason). As expected, all these factors together may provide a higher degree of fault tolerance.

On the other hand, a fundamental feature of NNs is that they operate as a "black box" without qualitative and quantitative information about the model they represent. This circumstance currently restrains develop-

[1] This statement belongs to one of the paper's reviewers. We have accepted it with gratitude.

ers and customers from applying such solutions, especially for high-responsibility tasks (e.g., in civil aviation, for safety-critical tasks). It is generally recognized that there exists an ambiguity regarding the operation of NNs in unforeseen situations [17].

The approach described above has a structural problem. It is practically impossible to preselect a combination of measurable parameters and network structure that would guarantee reliable discrimination of faults from a given list. This is usually achieved through trial and error. In addition, the set of training examples must be representative, i.e., contain a sufficient number of combinations of conditions for the task tackled.

2. DIAGNOSIS WITH FUZZY MODELS

The fuzzy model approach also helps in dealing with the precise knowledge problem for the model of a system diagnosed. This approach can be applied at two levels: first, fuzzy descriptions are employed to generate residuals, and then faults are detected using fuzzy logic again [18–24].

Fuzzy logic systems offer a linguistic model of system dynamics that can be easily understood using definite rules. They are also able to handle inaccurate or noisy data.

2.1. Fuzzy Model Formation

An effective approach to designing a fuzzy logic system begins with partitioning the available data into subsets (clusters) [19] characterized by simpler (linear or affine) behavior. A cluster can be defined as a group of data that are more similar to each other than to data from another cluster. Data similarity is usually expressed in terms of their distance to a given element within a cluster (used as the latter's prototype). Fuzzy clustering is an effective data partition tool with smooth, rather than abrupt, transitions between subsets, determined by the so-called membership functions.

In general, nonlinear MISO systems can be approximated using fuzzy inference [25, 26]. However, according to the approach proposed in [27], implications become crisp functions of the input data, and each fuzzy rule takes the form

$$R_i : \text{IF } x \in X_i \text{ THEN } \hat{y}_i = a_i^{\mathrm{T}} x + b_i, \; i = \overline{1, n_c}, \qquad (3)$$

where R_i is the ith rule; X_i is the ith cluster; n_c is the total number of clusters; a_i is the parameter vector; b_i is a scalar bias; $\hat{y}_i$ is the inference of the ith rule; finally, y_i is the output of the system diagnosed. Figure 4 illustrates the rule (3) for a one-dimensional variable x with $n_c = 5$.

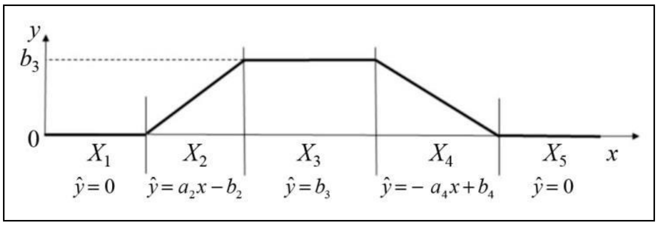


Fig. 4. The graphical illustration of the rule (3).

With the membership function $\lambda_i(x)$ considering the logic of cluster selection (IF … THEN …), the resulting formula for the entire set of rules (3) can be written as the weighted sum

$$\hat{y}_t = \frac{\sum_{i=1}^{n_c} \lambda_i(x_t) y_t^i}{\sum_{i=1}^{n_c} \lambda_i(x_t)} \qquad (4)$$

with the following notation: x_t is a suitable combination of the input and output signals; y_t^i is the output signal of the local linear (or affine) ith model, defined as

$$y_t^i = \sum_{k=1}^{n} a_{i.k} y_{t-k} + \sum_{k=0}^{n-1} b_{i.k} u_{t-k} + c_i, \qquad (5)$$

where $a_{i.k}$, $b_{i.k}$, and c_i are the parameters of the ith model in the subspace (cluster) X_i.

Formula (5) contains both the latest obtained u_t, y_{t-1} and the previous u_{t-1}, $y_{t-2}, \ldots$ samples of the input and output data, which reflects the dynamic behavior of the system. Therefore, the observation is treated as the linear autoregressive model with exogenous inputs (ARX) [9] of order s in which the regressor vector takes the form

$$x_t = \Big[\underbrace{y_{t-1} \;\cdots\; y_{t-s}}_{\text{output sample}} \; \underbrace{u_t \;\cdots\; u_{t-s+1}}_{\text{input sample}} \Big]^{\mathrm{T}},$$

where t is the current discrete time instant.

Unlike the classical counterpart, fuzzy clustering [20, 28, 29] allows any element to belong to several clusters simultaneously.

Another approach to fuzzy modeling is the fuzzy finite-state automaton model[2], which is used to describe discrete-event systems [22].

[2] In the below description of this approach, we have departed from the convention of part II of the survey to consider only dynamic systems with models (2) and (3) [30]: the symptom formation method is also applicable to the systems specified above.

2.2. Fault Diagnosis

The difference between the system outputs measured and their values obtained using the implicit model (2) can be taken as the residual:

$$r_t = y_t - \hat{y}_t.$$

Figure 5 shows the simplified block diagram of the residual formation based on an adaptive fuzzy model. Here, a fuzzy prototype and a recursive computing unit that implements some identification algorithm for the parameters use the same input and output data independently.

If several different faults have to be detected, the block diagram is reduced to another form with the multiplication of prototypes (models). Such a diagram will be discussed in part IV of the survey.

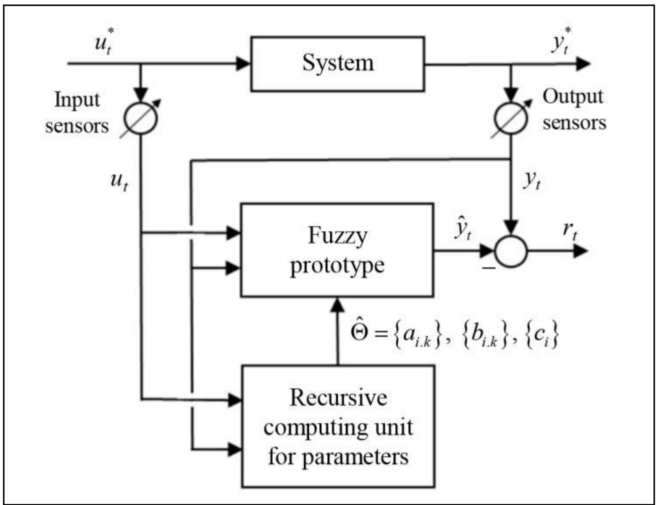


Fig. 5. Residual computation based on a fuzzy model.

A separate but very important problem is the fuzzy clustering of the system's state space [31]. Most fuzzy clustering algorithms optimize the *c*-means objective function $J(Z, U, V)$ as follows:

- The data are represented in the matrix form

$$Z = \begin{bmatrix} z_{11} & \cdots & z_{1N} \\ \vdots & \ddots & \vdots \\ z_{n1} & \cdots & z_{nN} \end{bmatrix},$$

where n and N denote the data dimension (the number of measurements taken per observation) and the number of available observations, respectively.

- The fuzzy partition matrix $U = [\mu_{ik}]$ is compiled from the values of the membership function, where i and k indicate the ith measurement and the kth cluster, respectively.
- The centers of the so-called prototypes $V = [v_1 \ \cdots \ v_{n_c}]$ are determined, i.e., the points used to estimate the distance between the cluster and the current state of the system.

The formula for minimizing the widespread *c*-means objective function [32] is

$$v_k = \arg\min J(Z, U, V)$$
$$= \arg\min \sum_{k=1}^{n_c} \sum_{i=1}^{N} (\mu_{ik})^m (z_i - v_k)^{\mathrm{T}} A(z_i - v_k)$$

with a weight $m > 1$, where the matrix A specifies the shape of the cluster.

To diagnose failures under measurement noises and modeling uncertainties, it is necessary to satisfy the noise separation principle: in this case, the residual generator is not affected by modeling uncertainty and input disturbances. This can be achieved using generalized observation schemes, consistency conditions [33], or particular design approaches [34].

For discrete-event systems, which often behave randomly [35, 36] and are described by a fuzzy finite-state automaton model, a fault diagnosis method based on the mathematical apparatus of fuzzy logic was proposed in [22]. The key element therein is the use of the determinizer of a fuzzy finite-state automaton, i.e., a finite-state automaton that simultaneously describes both the normal and abnormal behavior of the original automaton.

Compared to the algebraic approaches [37, 38] to the determinization of fuzzy finite-state automata, which preserve the dimension of the original automata, the approach under consideration involves additional useful information about the degree of confidence in the implementation of each possible transition during diagnosis, which potentially increases (if necessary) the depth of the fault search. The price paid is a significant dimension of the determinizer's transition table.

The fault diagnosis scheme in [22] contains several channels (exceeding by one the number of possible system faults). Each channel includes an observer in the form of the determinizer of a fuzzy finite-state machine. A decision block based on a comparison of the outputs of the discrete-event system and observers decides on the (in)operability of the system.

A distinctive feature of this method is that its implementation does not require the preliminary formation of a tabular description of diagnostic tools; all calculations are performed directly during the fault diagnosis process using compact analytical relationships, which is very important for technical condition monitoring.

The use of fuzzy logic for diagnosing a specific three-tank system was presented in [39]. Note that this system as a test object was taken in numerous publications, in particular, with consideration of various observers [40–42] (see part II of the survey [30]), parity equations with neuro-fuzzy identification [43], residuals from physical nonlinear equations [44] (as in part II of the survey [30], albeit with nonlinear equations), estimation of physical parameters using fuzzy NNs [45], and a fuzzy model based on the B-spline[3] of a network [46].

In [39], a *fault isolation system* (FIS) was defined as a quadruple

$$\text{FIS} = \langle F,\ S,\ V,\ \phi \rangle,$$

with the following notation: F is the set of faults; S is the set of symptoms with the two-level $\{0,\ 1\}$ or three-level $\{-1,\ 0,\ +1\}$ scale; V is the set of symptom values from the interval $[0, 1]$ or $[-1...+1]$, respectively; finally, ϕ is a function defined as the Cartesian product of the sets F and S, i.e., the set of all pairs $(f_i,\ s_j)$. Figure 6 illustrates the peculiarities of symptom computation with the following notation: r_j is a residual; $s_j \in S$ is a symptom; $v_{ij} \in V$ is the continuous value of the *j*th symptom for the *i*th fault; P and $N-$, $N+$ are the positive and negative symptom assessment domains, respectively.

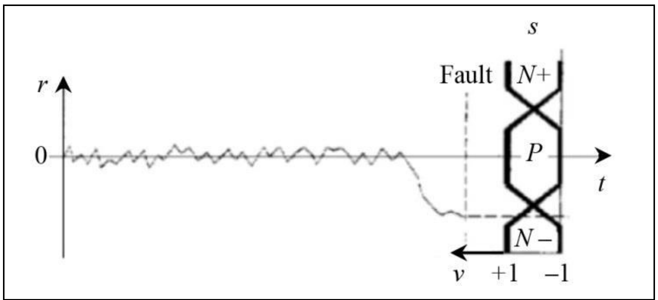


Fig. 6. Three-level symptom assessment.

Fault diagnosis is performed on such a set of symptoms using threshold elements and rules of the form (3) for each fault. For example, for three-digit symptoms, the rules for the number of faults are as follows:

$$R_i : \text{IF } s_1 = 0, s_2 = +1 \text{ or } -1,$$

$$s_3 = +1 \text{ or } -1, \ldots \text{THEN fault } f_i.$$

Thus, FIS is a table assigning a value or a subset of symptom values s_j to each fault f_i.

To isolate faults, the continuous values of symptoms v_{ij} are additionally calculated using formulas similar to (4), and the significance (grade of membership) of symptoms is thereby taken into account.

This interpretation of residuals allows considering the main uncertainty in the fault diagnosis process, i.e., the uncertainty of symptoms s_j. As claimed [40], this method provides much better robustness of the generated diagnoses to measurement noise compared to algorithms with threshold residual checking and logical inference.

[3] Abbreviation for "basis spline," i.e., a spline with the smallest carrier for specifying the degree.

3. STRUCTURAL FAULT DIAGNOSIS METHODS

Structural analysis involves a structural representation of a model, which is a rough description taking into account only the appearance of variables in each equation. Hence, large-scale problems can be analyzed efficiently and without numerical difficulties. However, the price paid is the excessive generality of the results. A fault diagnosis approach based on structural analysis was proposed in numerous works [47–50].

Structural analysis has good computer support, both in MATLAB and in Python.

A graph-theoretical algorithm called the Dulmage–Mendelsohn decomposition is fundamental to the analysis of the diagnostic properties of models [47, 49]. It consists in permuting the rows and columns of the so-called structural matrix of the system, containing special symbols that reflect the appearance of variables in different equations of the mathematical model of this system.

3.1. Structural Models

Let E be the set of constraints or equations in a model, and let V be the set of its variables. Then the structural model can be represented by a bipartite graph $G = (E \cup V, A)$, where A denotes the set of edges between nodes in two node sets E and V. An edge $(e_i, v_j) \in A$ if and only if the variable $v_j \in V$ appears in the model relation $e_i \in E$. A common way to visualize a structural model is the so-called biadjacency matrix of the graph, i.e., a matrix with rows and columns corresponding to constraints and variables. An element $(i,\ j)$ of the biadjacency matrix is empty if the variable v_j does not appear in the constraint (equation) e_i. A biadjacency matrix representing the structure of a model is also simply called a structural matrix.

The set of variables V can be classified as unknown $X \subset V$, known $Z \subset V$, or faulty $F \subset V$ under the condition $V = X \cup Z \cup F.$ For the purposes of analysis, the most important part of the structure is the one related to unknown variables, and the subgraph containing only the constraints (equations) and unknown variables is called the reduced structural graph.

Structural analysis is often explained on a demonstration example. As such an example, we take the mathematical model of an electric motor from the monograph [49]. This model is described by the equations

$$
\begin{aligned}
&e_1 : U = h(R + f_R) + Lh' + K_a h\omega,\\
&e_2 : h = \sqrt{\frac{T_m}{K_a \omega}},\\
&e_3 : J\omega' = T - b\omega,\\
&e_4 : T = T_m - T_l - f_{\text{load}},\\
&e_5 : y_h = h + f_h,\\
&e_6 : y_\omega = \omega + f_\omega,\\
&e_7 : y_T = T + f_T,\\
&e_8 : \frac{dh}{dt} = h',\\
&e_9 : \frac{d\omega}{dt} = \omega',
\end{aligned}
\tag{6}
$$

with the following notation: U is supply voltage; T_m is torque; T_l is rated load torque; h is the winding current of the motor; ω is the angular speed of the motor; R is the rated resistance of the motor winding; L is the inductance of the motor winding; K_a is the electromechanical coupling coefficient; J is the total rotational inertia of the rotor and load; finally, b is the friction coefficient.

The motor is equipped with sensors measuring the current y_h, the shaft torque y_T, and the angular velocity y_ω. Faults under consideration include the faults of all sensors, f_h, f_ω, and f_T, the winding resistance fault f_R, and the neglected additional motor load f_{load}.

In this case, with the above designations, the sets of variables V include

$$X = \{h, h', \omega, \omega', T, T_m, T_l\} = \{x_i\}_{i=\overline{1,7}},$$

$$Z = \{y_h, y_\omega, y_T, y_U\} = \{y_j\}_{j=\overline{1,4}},$$

$$F = \{f_R, f_h, f_\omega, f_T, f_{\text{load}}\} = \{f_k\}_{k=\overline{1,5}}.$$

In addition, the values of the following parameters are known: R, L, K_a, b, and J.

The structural matrix is represented as a table (e_i is the ith row, v_j is the jth column, here $v_j = \{x_{k=\overline{1,7}}, y_{l=\overline{1,4}}, f_{m=\overline{1,5}}\}$), and its entries are filled with the following symbols: "•" if the variable v_j directly appears in e_i; "D" if the variable v_j is represented by its derivative; "I" if the variable v_j is represented by its integral.

Figure 7 shows the initial structural matrix corresponding to model (6).

The result of the Dulmage–Mendelsohn decomposition is presented in Fig. 8: by permuting the constraints (equations) $e_1, \ldots, e_5$ and the variables $x_1, \ldots, x_4$, the biadjacency matrix of the reduced structural graph is transformed into a block triangular form. Three blocks are highlighted on the diagonal: the first two represent the well-defined part M^0 (the number of "internal" variables is equal to the number of equations), and the third represents the overdefined part M^+ (the number of "internal" variables is less than the number of equations) of the system model.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	f_1	f_2	f_3	f_4	f_5	y_1	y_2	y_3	y_4
e_1	•	•	•					•								•
e_2	•					•										
e_3			•	•	•											
e_4					•	•	•					•				
e_5	•								•				•			
e_6			•							•				•		
e_7					•						•				•	
e_8	I	D														
e_9			I	D												

Fig. 7. The structural matrix of system (6).

	x_7	x_6	x_1	x_2	x_3	x_4	x_5
e_4	•	•					•
e_2		•	•				
e_1			•	•	•		
e_8			I	D			•
e_3					•	•	•
e_9					I	D	
e_7							•
e_5			•				
e_6					•		

Fig. 8. The reduced structural matrix (diagnostic model) of system (6).

According to Fig. 8, the model has no underdefined part M^-. The well-defined part is used to uniquely determine the input unknowns, and the overdefined part forms the basis for fault diagnosis.

3.2. Fault Diagnosability Analysis

Fault diagnosability is understood as a combination of two properties:

– fault detectability (the potential capability to establish the fact of its existence),

– fault isolability[4] (the potential capability to determine the type, location, and time of occurrence of a detected fault).

The set of detectable and isolable (mutually distinguishable) faults in a system depends on the physical properties of the system and its relationships as well as on the available measurements. Fault detectability and isolability are system properties limiting the diagnostic performance that can be achieved by any fault diagnosis system.

Without loss of generality, by assumption, a single fault f_i can violate a single equation e_j of the model. If this is not the case, then additional equations are introduced to obtain an appropriate form.

For a model M, let O_f and O_{NF} denote the set of all its observations during operation with a fault f and without any faults, respectively. Obviously, the fault f in the model M is detectable under the condition[5] $O_f \setminus O_{\text{NF}} \neq \varnothing$.

[4] The term "localizability" is occasionally used in the literature as well [51, 52].

[5] It reads: the set of observations minus those corresponding to normal (fault-free) operation is non-empty.

Therefore, the structurally detectable fault f affects the equation e_f located in the overdefined part M^+ of the model:

$$e_f \in M^+ .$$

As applied to model (6), the fragment of the Dulmage–Mendelsohn decomposition takes the form shown in Fig. 9: the overdefined part M^+ (highlighted in gray) of the model includes the faults $f_1, \ldots, f_4$, whereas the fault f_5 (f_{load}) is not detectable ($e_{f_5} \in M^0$).

A fault f_i is considered to be isolable (and distinguishable from a fault f_j) if there exists an observation corresponding to f_i (allowing it to be detected) and simultaneously not associated with f_j. This is expressed by the formula $O_i \setminus O_j \neq \varnothing$ or

$$e_{f_i} \in \left(M \setminus \{e_{f_j}\}\right)^+ .$$

Detection is a necessary condition for isolation, i.e., the isolability of a fault always implies its detectability, and the converse is false:

$$e_{f_i} \in \left(M \setminus \{e_{f_j}\}\right)^+ \quad \Rightarrow \quad e_f \in M^+ .$$

To illustrate isolability using the example (6), we continue the Dulmage–Mendelsohn decomposition to the result shown in Fig. 10.

According to the analysis of Fig. 10, the detectable faults f_1 and f_2 (f_R and f_h) belong to one overdefined block (denoted by M_1^+) whereas the faults f_3 and f_4 (f_ω and f_T) to another overdefined block M_2^+. Thus, the specified groups of faults are isolated from each other, but they are not isolated within the groups.

In general, the isolability of a single fault can be calculated by analyzing ordered pairs of faults using the formula

$$\bigcap_{f_j \in F} \left\{ (f_i, f_j) \mid e_{f_i} \in \left(M \setminus \{e_{f_j}\}\right)^+ \right\},$$

with one Dulmage–Mendelsohn decomposition performed for each fault in the model. Thus, for the entire model, it is necessary to perform as many Dulmage–Mendelsohn decompositions as there are single system faults being analyzed.

Appropriate observations are required to implement potential detectability and isolability. Structural

	x_7	x_6	x_1	x_2	x_3	x_4	x_5	f_1	f_2	f_3	f_4	f_5
e_4	•	•					•					•
e_2		•	•									
e_1			•	•	•			•				
e_8			I	D			•					
e_3					•	•	•					
e_9					I	D						
e_7							•				•	
e_5			•						•			
e_6					•					•		

Fig. 9. The structural matrix (fault detectability) of system (6).

	x_7	x_6	x_1	x_2	x_4	x_5	x_3	f_1	f_2	f_3	f_4	f_5
e_4	•	•				•						•
e_2		•	•									
e_1			•	•			•	•				
e_5			•						•			
e_8			I	D								
e_3					•	•	•					
e_7						•					•	
e_9					D		I					
e_6							•			•		

Fig. 10. The structural matrix (fault isolability) of system (6).

analysis methods allow handling available observations, such as those in model (6) and in Fig. 6. In addition, it is possible to place sensors for achieving a given or possible level of fault detectability and isolability as well as to form recommendations for the development of diagnostic tests.

3.3. Fault Detection Based on Structural Analysis

There exist various modifications of structural analysis [53] and descriptions of their implementation algorithms [54]. Basically, mechanisms for forming and analyzing structurally overdefined sets (SOSs) M^+ or, more precisely, the minimal SOS are used. (Any subset of this SOS is not an SOS.) The minimal SOS can be achieved by partitioning the set M into equivalence classes [16] according to the appearing equations e_i to avoid searching for the same SOS in different ways.

As a result, each SOS contains a set of equations where at least one equation can be used for analytically expressing the residual. A set of equations resolved for the unknowns is used to design a residual generator.

As an illustrative example, we consider the SOS $M = \{e_1, e_3, e_5, e_7, e_8, e_9\}$ for an electric motor described by model (6). This set has five variables in six equations, and any of the equations can be used to form the residual. Figure 11 shows the diagrams of computation flows when selecting e_5, e_1, or e_3 as the redundant equation, respectively. For instance, Fig. 11a shows the computation flow from the differentiated variable ω' to the undifferentiated variable ω (i.e., integration). Figure 11c presents the opposite case, from h to h' (i.e., differentiation). Finally, Fig. 11b demonstrates both cases, integration and differentiation.

The diagrams in Fig. 11 are organized as follows: the input variables are indicated on the left edge, and the vertical line means the resolution of an appropriate equation with respect to the subsequent variable.

Each diagram mentioned has particular computational characteristics and sensitivity to faults.

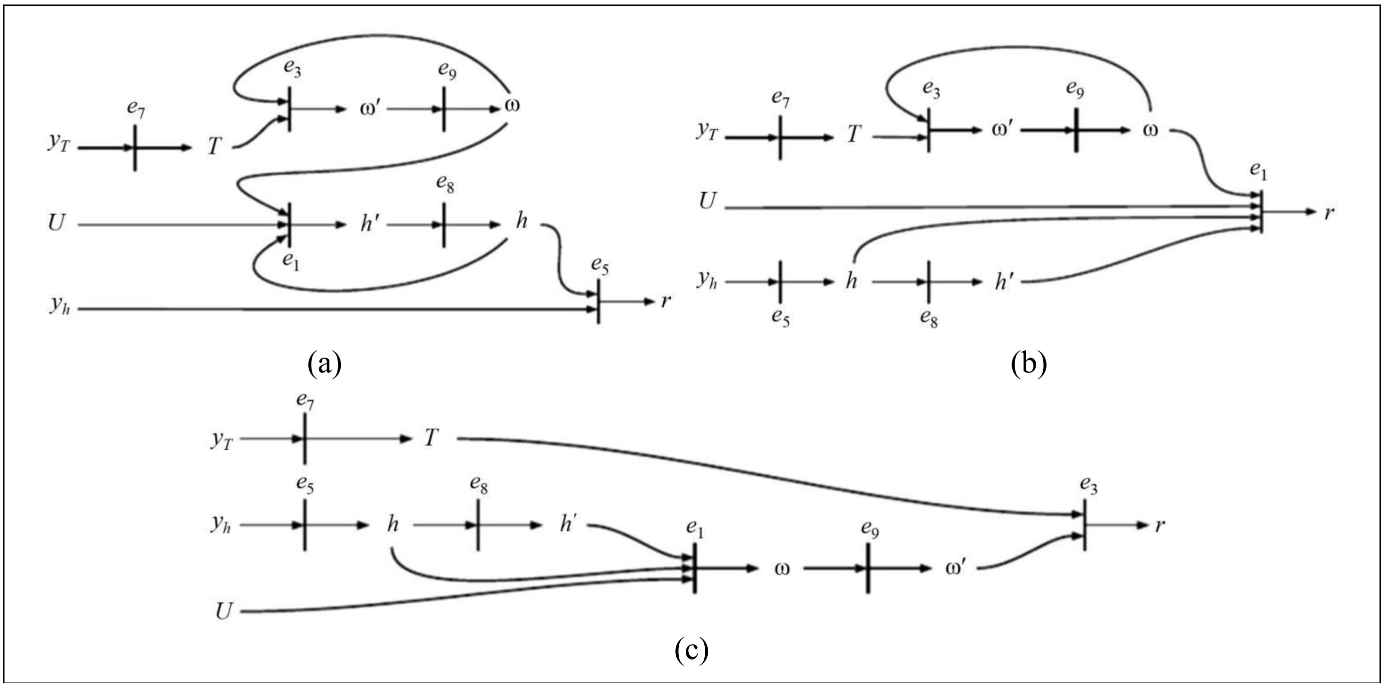


Fig. 11. SOS diagrams with different residual formation methods: (a) integral, (b) mixed, and (c) differential.

Now we derive the final computation formula for fault diagnosis based, e.g., on the diagram in Fig. 11c. This formula determines the content of any algorithm implemented. By making the corresponding substitutions, introducing the differentiation operator $p = d/dt$, and performing simple transformations, we arrive at the "exact" expression for the residual:

$$r = \frac{Jp+b}{K_a}\left(\frac{U}{y_h - f_h} - Lp\ln(y_h - f_h) - R - f_R\right) - y_T + f_T.$$

However, by definition, the faults f_h, f_R, and f_T are unavailable for direct measurement; thus, the formula for the residual in the neighborhood of the point $f_h = 0$, $f_R = 0$, $f_T = 0$, reduced to the signal of the motor shaft torque sensor, becomes

$$r(f_h, f_R, f_T) = \frac{Jp+b}{K_a}\left(\frac{U}{y_h} - Lp\ln(y_h) - R\right) - y_T. \quad (7)$$

According to this formula, the measurable signals U, y_h, and y_T are used to detect at least the separate faults f_h, f_R, and f_T. In the absence of faults, equality (7) takes a zero value. The values of the system parameters J, b, K_a, L, and the nominal (fault-free) value of R must be known.

The sensitivity to the "permanent" faults ($pf_h = 0$, $pf_R = 0$, $pf_T = 0$) is characterized by the values

$$\frac{\partial r}{\partial f_h} = \frac{Jp+b}{K_a}\left(Lp\frac{1}{y_h} - U\frac{1}{y_h^2}\right), \quad \frac{\partial r}{\partial f_R} = -\frac{b}{K_a}, \quad \frac{\partial r}{\partial f_T} = 1.$$

Hence, if the sensor's signal y_h of the motor winding current changes in accordance with the nonlinear differential equation

$$JLp^2\frac{1}{y_h} + bLp\frac{1}{y_h} - JUp\frac{1}{y_h^2} - bU\frac{1}{y_h^2} = 0,$$

the sufficient condition $\partial r / \partial f_h \neq 0$ of the sensitivity (7) to the fault f_h will not hold.

This circumstance suggests that, on certain phase trajectories of the system's motion, fault diagnosis may not be performed.[6]

4. FAULT DIAGNOSIS METHODS BASED ON SET THEORY

When using dynamic system models for monitoring purposes, there is always some deviation between the modeling results and the actual behavior of the systems. This phenomenon is due to both the neglect of some known "insignificant" relationships and the presence of unknown or inaccurately known relationships in the object modeled. The resulting modeling errors introduce uncertainty, which most often lies within the estimated ranges.

There are several ways to deal with model-related uncertainty, depending on whether it appears in the parameters (structured uncertainty) or in the model structure (unstructured uncertainty). The most developed group of approaches, known as active, is based on generating residuals insensitive to given uncertain-

[6] This applies not only to structural methods.

ties but sensitive to system faults. At the same time, there is another family of approaches, known as passive, that increase the reliability of a fault detection system by extending uncertainty to the residual values with creating appropriate adaptive thresholds.[7] The second approach was proposed in the fundamental works [55] (the time domain) and [27] (the frequency domain).

4.1. Initial Conditions and Problem Statement

The idea behind the set-membership approach is to use a geometric set to bind uncertain states. Well-known geometric sets include intervals, ellipsoids, polyhedra, zonotopes, etc. Among all the set-based approaches, zonotopes are remarkable for the lowest computational complexity of their implementation.

The approach under consideration involves the specific concept and notation of a zonotope—a mathematical construct[8] of the form

$$Z = \langle c, H \rangle = \{ c + Hz, \|z\|_\infty \le 1 \}, \tag{8}$$

where c and H are the center and segment matrix of the zonotope. The expression (8) is used for the formal designation of an interval in a multidimensional space. On the plane, this is a rectangle with sides $2h_1$ and $2h_2$; in the 3D space, a parallelepiped with edges $2h_1$, $2h_2$, $2h_3$, etc.

Zonotopes are characterized by linearity in the sense of

$$\langle c_1, H_1 \rangle \oplus \langle c_2, H_2 \rangle = \langle c_1 + c_2, [H_1, H_2] \rangle,$$
$$L\langle c, H \rangle = \langle Lc, LH \rangle,$$

where $\oplus$ denotes the Minkowski sum;[9] L is an arbitrary matrix of compatible dimensions. Here, $[H_1, H_2]$ means the formation of a new segment matrix from the initial ones H_1 and H_2. As a rule, it leads to the “growth” of the zonotope limits.

For a zonotope $Z = \langle c, H \rangle$, the weighted reduction operator $\downarrow^H_{q,W}$ [56] satisfies the property

$$\langle c, \downarrow^H_{q,W} \rangle \subseteq \langle c, H \rangle,$$

where $q \ge n$ indicates the maximum number of the columns of $\downarrow^H_{q,W}$ and W is a weight matrix of compatible dimensions. This operator reduces the segment matrix determining the “spread” of possible values in the neighborhood of the zonotope center.

Consider a linear system with discrete time $t = 0, 1, 2, \ldots$:

$$x_{t+1} = Ax_t + Bu_t + D_w w_t, \; y_t = Cx_t + D_v v_t, \tag{9}$$

where $x_t \in \Re^{n_x}$, $u_t \in \Re^{m_u}$, and $y_t \in \Re^{n_y}$ are the system state vectors, known inputs, and outputs, respectively; $w_t \in \Re^{m_w}$ and $v_t \in \Re^{m_v}$ are the disturbance and measurement noise vectors, respectively.

The values of the vectors w_t and v_t and the initial state x_0 are bounded by the zonotopes[10]

$$w_t \in \langle 0, I_{m_w} \rangle, \; v_t \in \langle 0, I_{m_v} \rangle, \; x_0 \in X_0 = \langle c_0, H_0 \rangle \tag{10}$$

and are supposed to be unknown at each time instant.

4.2. Zonotopic Observer

In this subsection, we briefly present one fault diagnosis approach using zonotopes [16, 49].

For system (9), an observer with a gain $G(k) \in \Re^{n_x \times n_y}$ is introduced:

$$\hat{x}_{t+1} = A\hat{x}_t + Bu_t + D_w w_t + G_t(y_t - C\hat{x}_t - D_v v_t). \tag{11}$$

As claimed [58], for any $t+1 \in N$, the observer (11) satisfies the inclusion

$$\hat{x}_{t+1} \in \aleph_{t+1} = \langle c_{t+1}, H_{t+1} \rangle, \tag{12}$$

where the center and segment matrix are given by

$$c_{t+1} = (A - G_t C)c_t + Bu_t + G_t y_t,$$
$$H_{t+1} = \left[(A - G_t C) \downarrow^H_{q,W}, D_w, G_t D_v \right]. \tag{13}$$

According to formulas (13), the inclusion (12) is expanded as follows:

$$\begin{aligned} \hat{x}_{t+1} &\in \langle c_{t+1}, H_{t+1} \rangle \\ &= \left((A - G_t C) \langle c_t, \downarrow^H_{q,W} \rangle \right) \\ &\oplus \left(B \langle u_t, 0 \rangle \right) \oplus \left(C_t \langle y_t, 0 \rangle \right) \\ &\oplus \left(D_w \langle 0, I_{m_w} \rangle \right) \oplus \left(-G_t D_v \langle 0, I_{m_v} \rangle \right). \end{aligned} \tag{14}$$

Formula (14) defines the temporal transformation of the zonotope limiting the subsequent values of the vector $\hat{x}_t$ $\forall t+1 \in N$.

[7] Although the approach uses traditional models of the form (1) [30] (see part II of the survey), they do not constitute its core.

[8] A polyhedron representing the Minkowski sum of a set of vectors.

[9] The Minkowski sum of sets *A* and *B* of a linear space *V* is the set *C* consisting of the sums of all possible vectors from *A* and *B*: $C = \{ c \mid c = a + b, \; a \in A, \; b \in B \}$, see the paper [57].

[10] The first two zonotopes are called unitary.

4.3. Fault Detection

Consider a system where formulas (9) are replaced by the equations

$$\begin{gathered} x_{t+1} = Ax_t + Bu_t + D_w w_t + Q^u f_t^u, \\ y_t = Cx_t + D_v v_t + Q^y f_t^y, \end{gathered} \tag{15}$$

where $f_t \in \Re^{m_f}$ and $Q \in \Re^{n_y \times m_f}$ are the fault effect vector and the fault effect matrix of the actuator (the superscript *u*) and sensor (the superscript *y*), respectively. To detect faults in this system, the following actions are performed:

– Input $U_t = \{u_t\}$ and output $Y_t = \{y_t\}$ data are acquired and stored.

– The set of estimated states $Y_t = \{y_t\}$ is calculated using equation (14) and the interval constraints (10)

$$X_0, u_t, y_t, w_t, v_t \to X_t^e.$$

– The set of estimated outputs Y_t^e is calculated using the second equation (9) and the interval constraints (10)

$$X_t^e, v_t \to Y_t^e.$$

– The intersection of the sets of estimated and measured outputs is calculated:

$$X_t^{f=0} = Y_t^e \cap Y_t. \tag{16}$$

If for each t the intersection (16) is non-empty, then the actual value of the system output is considered to coincide with its forecast; therefore, the absence of faults is concluded. The emptiness of this intersection is a sign of system faults.

In this case, the sensitivity condition of the test (16) to sensor faults in equations (15) is the non-strict inequality

$$\sum_{t=0}^{\infty} (c_t^{rf})^{\mathrm{T}} c_t^{rf} \ge \beta_s^2 \sum_{t=0}^{\infty} f_t^{\mathrm{T}} f_t \tag{17}$$

given the existence of matrices[11] $P \in S_{\succ 0}^{n_x}$, $K \in \Re^{n_x \times n_y}$, and $N \in \Re^{n_y \times n_y}$ as well as scalars $\alpha > 0$ and $\beta > 0$ such that[12]

$$\begin{bmatrix} \alpha P & * & * \\ \alpha A^{\mathrm{T}} P - \alpha C^{\mathrm{T}} K^{\mathrm{T}} & \alpha P + C^{\mathrm{T}} N C & * \\ -\alpha Q^{\mathrm{T}} K^{\mathrm{T}} & Q^{\mathrm{T}} N^{\mathrm{T}} C & Q^{\mathrm{T}} N Q - \beta^2 I_{m_f} \end{bmatrix} \succ 0.$$

In (17), c_t^{rf} is the center of the zonotope $R_t = \langle c_t^r, H_t^r \rangle, \forall t \in N$, of the residual $r_t^f = y_t - Cx_t$ with the parameters

$$c_t^r = M(y_t - Cc_t),\ H_t^r = \begin{bmatrix} MCH_t & MD_v \end{bmatrix},$$

where $M \in \Re^{n_y \times n_y}$ is the gain matrix of the zonotope of the residual; as an option, the equalities $G = P^{-1}K$ and $M = N^2$ can be used.

The corresponding proofs and examples of using zonotopes can be found in [51, 59].

5. STATISTICAL FAULT DIAGNOSIS

The pattern recognition-based diagnostic approach is based on the statistical decision method, which consists in the following.

Consider a system with a discrete feature space (measurements) of dimension k with axes ξ_1, ξ_2,..., ξ_k and n gradations along the axes π_1, π_2,..., π_n, respectively. The gradations provide for both the operable technical condition of the system and the presence of various faults. By assumption, there are sufficiently many such systems,[13] and depending on the relevant characteristics, they are divided into classes K_{ij}, $i = \overline{1, k}$, $j = \overline{1, n}$, including variants without faults $\mathrm{K}_{\text{no fault}}$ or with different faults $\mathrm{K}_{m\text{th fault}}$, where $m \le kn - 1$.

The main parameter is an empirical (statistical) estimate of the likelihood ratio. In a simplified version (a "local" estimate) of the method, it is calculated by the formula [60]

$$L(\xi) = \frac{f_l(\xi)}{f_h(\xi)} \approx \frac{r_l + 1}{r_h + 1} \frac{N_h + 2}{N_l + 2}, \tag{18}$$

where $f_l(\xi)$ and $f_h(\xi)$ are the Bayesian estimates of the probabilities that the system from the *l*th and *h*th class, respectively, will fall into a certain point ξ (or its neighborhood); r_l and r_h are the number of systems of the *l*th and *h*th classes falling into the point ξ according to the a priori information; finally, N_l and N_h are the number of systems of the *l*th and *h*th classes, respectively, in the training sample. More accurate, albeit complex, formulas with probability densities can be found in [60, 61].

[11] The first matrix mentioned is positive definite and symmetric, $P \succ 0$.

[12] Here, asterisks serve to simplify the form of the symmetric matrix.

[13] Due to the ergodicity hypothesis of random processes [62], for a priori data, the matter concerns the number of systems used for training systems; for a posteriori data, the number of time-separated observations of the system diagnosed.

To decide on the system's belonging to a particular class, the following threshold rule is applied:

- the system is operable if $L(\xi) \in K_{\text{no fault}}$;

- the system has *m*th fault if $L(\xi) \in K_{m\text{th fault}}$.

The training stage lies in memorizing the systems from the training sample, and the statistical estimate (18) is calculated during the current recognition (fault diagnosis).

Formally, the solution of the fault diagnosis (prediction) problem using the mathematical apparatus of statistical classification does not differ in principle from that of other technical problems requiring recognition. However, there are several peculiarities that must be taken into account. The main stages of solving the problem include:

• selection of a recognition model, which can be deterministic or probabilistic and depends on the degree of mixing of the sets belonging to different classes;

• description of the patterns R_λ of different classes (operable and different faults) based on a priori information; evaluation of the informativeness of different parameters (their significance for recognition) allows optimizing the description of patterns;

• comparison of current information about the system monitored with a priori specified patterns R_λ of the classes;

• deciding on the degree of operability of the system or its parts based on the monitoring data.

The main peculiarity of the statistical approach is that, for acceptable performance, it requires a significant amount of both a priori data (training) and a posteriori data (current monitoring measurements). Nevertheless, the "local" estimate (18) allows solving problems with $k \leq 50$ parameters (measurements) and $\pi \leq 8$ gradations for each parameter (related to the number of possible faults). In other words, the matter concerns not too large samples, which is essential for monitoring tasks.

Statistical fault diagnosis based on pattern recognition has become widespread and developed in Russian scientific literature and technology; for example, see [63–65].

CONCLUSIONS

The use of nonclassical representations or descriptions for dynamic systems has significantly expanded the possibilities for their fault diagnosis, primarily in terms of overcoming the traditional difficulties for engineering applications: noise in real measurements and distortions inevitably introduced into formal models. As a result, diagnostic and monitoring systems with fundamentally new capabilities and areas of practical application have been created and widely adopted.

At the same time, this path poses additional problems: besides complicating the apparatus used, the results obtained may be opaque, vague, ambiguous, or overly general. A lack of specifics can seriously restrict the application of the relevant approaches.

Part IV of the survey will analyze new approaches to fault diagnosis and the integration of various models and methods.

REFERENCES

1. Forssell, U. and Ljung, L., Closed-Loop Identification Revisited, *Automatica*, 1999, vol. 35, no. 7, pp. 1215–1241.
2. Haber, R. and Unbehauen, H., Structure Identification of Nonlinear Dynamic Systems – A Survey on Input/Output Approaches, *Automatica*, 1990, vol. 26, no. 7, pp. 651–677.
3. Isermann, R., *Fault-Diagnosis Systems. An Introduction from Fault Detection to Fault Tolerance*, Berlin–Heidelberg: Springer, 2006.
4. Kalyavin, V.P. and Davydov, N.A., *Nadezhnost' i diagnostika avtomototransportnykh sredstv* (Reliability and Diagnosis of Motor Vehicles), St. Petersburg: Elmor, 2014. (In Russian.)
5. Palyukh, B.V., Shprekher, D.M., and Bogatikov, V.N., Electromechanical Systems Diagnosis Based on Neural Network Technologies, *Software & Systems*, 2015, no. 3, pp. 5–11. (In Russian.)
6. Pushkarev, D.O., Application of Neural Network Expert Systems for Monitoring, Diagnosis and Forecasting of Technical Condition of Aircraft Engines, *Inform.-Tekhnol. Vestn.*, 2020, no. 3(25), pp. 51–59. (In Russian.)
7. Kulida, E.L. and Lebedev, V.G., Modern Approaches to Prognostics and Health Management of an Aircraft Electromechanical Actuator, *Control Sciences*, 2024, no. 3, pp. 3–15.
8. Vasilescu, M.-V., Epikhin, A.I., and Toriya, T.G., Diagnosis of the Technical Condition of a Boiler Plant Using Bayesian Networks, *Operation of Maritime Transport*, 2023, no. 3, pp. 113–120.
9. Loskutov, A.I., Stolyarov, A.V., and Ryakhova, E.A., Method for Control of Technical State of Onboard Radioelectronic Equipment of Spacecrafts Based On Interaction of Elements of Distributed Technical Diagnosis System, *Testing. Diagnosis*, 2021, vol. 24, no. 1 (271), pp. 44–50. (In Russian.)
10. Haykin, S.S., *Neural Networks and Learning Machines. Vol. 3*, Upper Saddle River, NJ: Pearson Education, 2009.
11. Diaconescu, E., The Use of NARX Neural Networks to Predict Chaotic Time Series, *WSEAS Trans. Comp. Research*, 2008, vol. 3, pp. 182–191.
12. Liu, G.P. and Patton, R.J., *Eigenstructure Assignment for Control System Design*, New York: John Wiley & Sons, 1998.
13. Medsker, L. and Jain, L.C., *Recurrent Neural Networks: Design and Applications*, Cleveland: CRC Press, 1999.
14. Kruglov, V.V. and Borisov, V.V., *Iskusstvennye neironnye seti. Teoriya i praktika* (Artificial Neural Networks: Theory and Practice), Moscow: Goryachaya liniya – Telekom, 2002. (In Russian.)
15. Reizlin, V.I., *Chislennye metody optimizatsii* (Numerical Optimization Methods), Tomsk: Tomsk National Research Technical University, 2013. (In Russian.)

16. Krysander, M., Åslund, J., and Nyberg, M., An Efficient Algorithm for Finding Minimal Overconstrained Subsystems for Model-Based Diagnosis, *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 2008, vol. 38, no. 1, pp. 197–206.
17. Korbicz, J., Patan, K., and Obuchowicz, A., Dynamic Neural Network for Process Modelling in Fault Detection and Isolation Systems, *Applied Mathematics and Computer Science*, 1999, vol. 9, no. 2, pp. 519–546.
18. Dexter, A.L. and Benouarets, M., Model-Based Fault Diagnosis Using Fuzzy Matching, *IEEE Trans. on Sys. Man. and Cyber. Part A: Sys. & Humans*, 1997, vol. 27, no. 5, pp. 673–682.
19. Isermann, R., On Fuzzy Logic Applications for Automatic Control, Supervision and Fault Diagnosis, *IEEE Trans. on Sys. Man. and Cyber. Part A: Sys. & Humans*, 1998, vol. 28, no. 2, pp. 221–235.
20. Polkovnikova, N.A. and Kureichik, V.M., Neural Network Technologies, Fuzzy Clustering and Genetic Algorithms in an Expert System, *Izvestiya SFedU. Eng. Sci.*, 2014, no. 7 (156), pp. 7–15. (In Russian.)
21. Akhmetkhanov, R.S., Dubinin, E.F., and Kuksova, V.I., Application of Methods and Models of Fuzzy Logic in Technical Diagnosis Systems, *Machine Drives and Parts*, 2018, no. 1–2 (27), pp. 6–11. (In Russian.)
22. Shumsky, A.E. and Zhirabok, A.N., A Fault Diagnosis Method for Discrete-Event Systems Based on the Fuzzy Finite State Automaton Model, *Control Sciences*, 2025, no. 2, pp. 31–41.
23. Klyuchkin, A.K. and Khamatov, A.A., Synthesis of the Diagnostic Model Based on Fuzzy Logic Apparatus, *Aerospace Instrument-Making*, 2023, no. 7, pp. 34–41. (In Russian.)
24. Ablyazov, E.K. and Ablyazov, K.A., Assessment of the Technical Condition of the Port Lifting and Transport Equipment (LTE) Using Fuzzy Logic, *Marine Intellectual Technologies*, 2024, no. 2-1 (64), pp. 242–251. (In Russian.)
25. Fantuzzi, C. and Rovatti, R., On the Approximation Capabilities of the Homogeneous Takagi-Sugeno Model, *Proceedings of the 5th IEEE International Conference on Fuzzy Systems*, vol. 2, NJ: IEEE, Piscataway, 1996, pp. 1067–1072.
26. Rovatti, R., Takagi-Sugeno Models as Approximators in Sobolev Norms: The SISO Case, *Proceedings of the 5th IEEE International Conference on Fuzzy Systems, vol. 2*, NJ: IEEE, Piscataway, 1996, pp. 1060–1066.
27. Takagi, T. and Sugeno, M., Fuzzy Identification of Systems and Its Applications to Modeling and Control, *IEEE Transactions on Systems, Man and Cybernetics*, 1985, no. 1, pp. 116–132.
28. Jun, W., Shitong, W., and Chung, F.-L., Positive and Negative Fuzzy Rule System, Extreme Learning Machine and Image Classification, *Int. J. of Machine Learning and Cybernetics*, 2011, no. 2 (4), pp. 261–271.
29. Graaff, A.J. and Engelbrecht, A.P., Clustering Data in Stationary Environments with a Local Network Neighborhood Artificial Immune System, *Int. J. of Machine Learning and Cybernetics*, 2012, vol. 3, no. 1, pp. 1–26.
30. Bukov, V.N., Bronnikov, A.M., Popov, A.S., and Shurman, V.A., Technical Condition Monitoring Methods to Manage the Redundancy of Systems. Part II: Classical Models, *Control Sciences*, 2025, no. 3, pp. 2–11.
31. *Diagnosis and Fault-Tolerant Control 2: From Fault Diagnosis to Fault-Tolerant Control*, coord. by V. Puig and S. Simani. London: ISTE Ltd.; Hoboken: John Wiley & Sons, 2021.
32. Bezdek, J.C., *Pattern Recognition with Fuzzy Objective Function Algorithms*, Berlin: Springer Science & Business Media, 2013.
33. Chow, E.Y. and Willsky, A.S., Analytical Redundancy and the Design of Robust Detection Systems, *IEEE Trans. Automatic Control*, 1984, vol. 29, no. 7, pp. 603–614.
34. Duan, G., How, D., and Patton, R., Robust Fault Detection in Descriptor Systems via Generalised Unknown Input Observers, *Int. J. Systems Science*, 2002, vol. 32, no. 2, pp. 7724–7729.
35. Zaytoon, J. and Lafortune, S., Overview of Fault Diagnosis Methods for Discrete Event Systems, *Annual Reviews in Control*, 2013, vol. 37, no. 2, pp. 308–320.
36. Cassandras, Ch. and Lafortune, S., *Introduction to Discrete Event Systems*, 2nd ed., New York: Springer Science+Business Media, 2008.
37. Zhirabok, A.N., Kalinina, N.A., and Shumskii, A.E., Technique of Monitoring a Human Operator's Behavior in Man-Machine Systems, *Journal of Computer and Systems Sciences International*, 2018, vol. 57, no. 3, pp. 443–452.
38. Zhirabok, A.N., Kalinina, N.A., and Shumskii, A.E., Method for the Functional Diagnosis of Nondeterministic Finite State Machines, *Journal of Computer and Systems Sciences International*, 2020, vol. 59, no. 4, pp. 565–574.
39. Kościelny, J.M., Application of Fuzzy Logic for Fault Isolation in a Three-Tank System, *Proc. of the 14th IFAC World Congress*, Beijing, 1999, vol. 7, pp. 73–78.
40. Hengy, D. and Frank, P.M., Component Failure Detection Using Local Second-Order Observers, in *Fault Detection & Reliability*, Singh, M.G., Ed., Oxford: Pergamon Press, 1987, pp. 147–154.
41. Frank, P.M., Fault Diagnosis in Dynamic Systems via State Estimation – A Survey, in *System Fault Diagnosis, Reliability and Related Knowledge-Based Approaches*, Tzafestas. E.A., Ed., Dordrecht: Dr. Reidel Publ., 1987, vol. 1, pp. 35–98.
42. Koenig, D., Nowakowski, S., and Cecchin, T., An Original Approach for Actuator and Component Fault Detection and Isolation, *Proceedings of the CAA Symposium on Fault Detection Supervision and Safety for Technical Processes (SAFEPROCESS '97)*, Kingston upon Hull, 1997, vol. 1, pp. 95–105.
43. Garcia, F.L., Izquierdo, V., de Miguel, L., and Peran, J., Fuzzy Identification of Systems and Its Applications to Fault Diagnosis Systems, *Proceedings of the CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS '97)*, Kingston upon Hull, 1997, vol. 2, pp. 705–712.
44. Koscielny, J.M., Sedziak, D., and Sikora, A., The "DIAG" System for Fault Detection and Isolation in Industrial Processes, *Proceedings of the CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS '94)*, Helsinki, 1994, vol. 2, pp. 790–795.
45. Han, Z. and Frank, P.M., Physical Parameter Estimation Based FDI with Neural Networks, *Proceedings of the CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS '97)*, Kingston Upon Hull, 1997, vol. 1, pp. 294–299.
46. Benkbedda, H. and Patton, R., Information Fusion in Fault Diagnosis Based on B-spline Networks, *Proceedings of the CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS '97)*, Kingston upon Hull, 1997, vol. 2, pp. 681–686.
47. Cassar, J.P. and Staroswiecki, M., A Structural Approach for the Design of Failure Detection and Identification Systems, *IFAC Proceedings Volumes*, 1997, vol. 30, no. 6, pp. 841–846.
48. Ploix, S., Désinde, M., and Touaf, S., Automatic Design of Detection Tests in Complex Dynamic Systems, *IFAC Proceedings Volumes*, 2005, vol. 38, no. 1, pp. 478–483.

49. *Diagnosis and Fault-Tolerant Control 1: Data-Driven and Model-Based Fault Diagnosis Techniques*, coord. by V. Puig and S. Simani, London: ISTE Ltd.; Hoboken: John Wiley & Sons, 2021.
50. Kopkin, E.V. and Ivanyu, A.Yu. Structural-Stochastic Computational Models for Monitoring the Technical Condition of Onboard Spacecraft Systems, *Izv. Tula State Univ. Eng. Sci.*, 2022, no. 10, pp. 9–14. (In Russian.)
51. *Recommendations for Standardization R 50.1.048-2004: Information and Telecommunication Gaming Systems. Terms and Definitions*, Moscow: Gosstandart of Russia, 2004. (In Russian.)
52. Bruskin, S.N., Druzhayev, A.A., Maron, A.I., and Maron, M.A., Effective Methods for Creation of Malfunction Search Algorithms in Information Systems, *Intellekt. Sist. Proizv.*, 2017, vol. 15, no. 3, pp. 88–93. (In Russian.)
53. Armengol, J., Bregyn, A., Escobet, T., et al., Minimal Structurally Overdetermined Sets for Residual Generation: A Comparison of Alternative Approaches, *IFAC Proceedings Volumes*, 2009, vol. 42, no. 8, pp. 1480–1485.
54. Krysander, M., Åslund, J., and Frisk, E., A Structural Algorithm for Finding Testable Sub-models and Multiple Fault Isolability Analysis, *Proceedings of the 21st International Workshop on Principles of Diagnosis (DX-10)*, Portland, USA, 2010. DOI: https://doi.org/10.36001/phmconf.2010.v2i1.1940
55. Horak, D., Failure Detection in Dynamic Systems with Modelling Errors, *AIAA J. of Guidance, Control and Dynamics*, 1988, vol. 11, no. 6, pp. 508–516.
56. Song, Y., Sai, M.P.D., and Yu, H., Zonotope-Based Nonlinear Model Order Reduction for Fast Performance Bound Analysis of Analog Circuits with Multiple-Interval-Valued Parameter Variations, *Proceedings of IEEE Design, Automation & Test in Europe: Conference & Exhibition*, Dresden, Germany, 2014. DOI: 10.7873/DATE2014.024
57. Panyukov, A.V., The Linear Inequalities Set Representation of Minkowski's Sum for Two Polyhedrons, *Bulletin of the South Ural State Univ. Math. Model. Program.*, 2012, no. 14, pp. 108–119. (In Russian.)
58. Combastel, C., Zonotopes and Kalman Observers: Gain Optimality under Distinct Uncertainty Paradigms and Robust Convergence, *Automatica*, 2015, no. 55, pp. 265–273.
59. Himmelblau, D.M., *Fault Diagnosis in Chemical and Petrochemical Processes*, Amsterdam: Elsevier, 1978.
60. Gaskarov, D.V., Golinkevich, T.A., and Mozgalevskii, A.V., *Prognozirovanie tekhnicheskogo sostoyaniya i nadezhnosti radioelektronnoi apparatury* (Predicting the Technical Condition and Reliability of Radio-Electronic Equipment), Moscow: Sovetskoe Radio, 1974. (In Russian.)
61. Mozgalevskii, A.V. and Koida, A.N., *Voprosy proektirovaniya sistem diagnostirovaniya* (Design Issues for Diagnosis Systems), Leningrad: Energoatomizdat, 1985. (In Russian.)
62. Ventsel', E.S., *Teoriya veroyatnostei* (Probability Theory), Moscow: KnoRus, 2010. (In Russian.)
63. Obozov, A.A. and Tarichko, V.I., Stochastic Image Recognition Theory As a Means to Improve Technical Diagnosis of Car Engines, *Dvigatelestroenie*, 2013, no. 3 (253), pp. 16–22. (In Russian.)
64. Usik, B.A., Luchin, A.V., and Obynochnyi, I.A., Selection of Parameters for Diagnosing Converter Devices with Redundant Elements by Pattern Recognition Methods, *Izv. Inst. Inzhen. Fiz.*, 2022, no. 3 (65), pp. 14–16. (In Russian.)
65. Shamolin, M.V., Generalized Checking Problem in Diagnosis Problems, *Itogi Nauki Tekh. Sovremenn. Mat. Prilozh. Tematich. Obz.*, 2022, vol. 209, pp. 117–126. (In Russian.)

This paper was recommended for publication by V.G. Lebedev, a member of the Editorial Board.

Received November 13, 2024,
and revised August 21, 2025.
Accepted August 27, 2025.

Author information

Bukov, Valentin Nikolaevich. Dr. Sci. (Eng.), JSC Airborne Aeronautical Systems, Moscow, Russia
✉ v_bukov@mail.ru
ORCID iD: https://orcid.org/0000-0002-5194-8251

Bronnikov, Andrei Mikhailovich. Dr. Sci. (Eng.), Bauman Moscow State Technical University, Moscow, Russia
✉ bronnikov_a_m@mail.ru
ORCID iD: https://orcid.org/0009-0009-1216-3521

Popov, Aleksandr Sergeevich. Cand. Sci. (Eng.), Zhukovsky–Gagarin Air Force Academy, Voronezh, Russia
✉ saga30@yandex.ru

Shurman, Vladimir Aleksandrovich. JSC Research Institute of Aviation Equipment, Zhukovsky, Russia
✉ shurman@niiao.ru

Cite this paper
Bukov, V.N., Bronnikov, A.M., Popov, A.S., and Shurman, V.A., Technical Condition Monitoring Methods to Manage the Redundancy of Systems. Part III: Nonclassical Models in Fault Diagnosis. *Control Sciences* **4,** 2–16 (2025).

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com

CLASSIFICATION OF ACTIVITY PROCESSES

G. N. Kalyanov* and D. A. Novikov**

*,**Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia,

*✉ kalyanov@ipu.ru, **✉ novikov@ipu.ru

Abstract. Nowadays, methodology—the general theory of activity organization and control—continues to develop. This paper introduces a classification system for processes, determining the place of any process and establishing the presence of interrelations with other processes. Several classification attributes are identified, and a corresponding classification scheme of elementary processes is proposed. The concept of a complex process is considered, and its division into elementary processes in accordance with the classification scheme is demonstrated. An example of a complex process of motor vehicle repair is given; its functional and behavioral schemes are presented, and the corresponding elementary processes of various types are described in detail, namely, supporting processes (the processes of its life cycle), organizational and management processes (business processes), as well as informational and computational processes.

Keywords: process, activity, methodology, classification, life cycle.

INTRODUCTION

In recent years, within the framework of interdisciplinary research, a new scientific direction has been formed to unify the description and formalization of various types of human activity. (*Activity* is a goal-oriented active effort of an individual.) This direction, devoted to developing a unified theory of activity as an aggregate of general and universal models, is represented by the works of A.M. Novikov and D.A. Novikov and their colleagues [1–5]. The corresponding doctrine (theory) of the general regularities of activity organization is called *methodology*.

Methodology includes general methodology (the methodology of elementary activity) [1], the methodologies of various types of human activity (the so-called partial methodologies: research methodology [5], the methodology of practical activity, including control methodology [4] as its variant, as well as the methodologies of educational, artistic, and play activities) [2, 6], and the methodology of complex activity, generalizing the above methodologies to the case of non-elementary activity. The concept of complex activity was defined as "a goal-oriented active effort of an individual that has a nontrivial internal structure, multiple and/or varying subject, technology and role of the object of activity in its goal context" [1]. Of course, these types of human activity are not exhaustive; in particular, philosophical and religious activities were mentioned in [1–5]. This list can probably be enlarged.

Since methodology is regarded as the doctrine of activity organization [2], the concept of organization should be explained. The Philosophical Encyclopedic Dictionary [7] provides three meanings of this concept. For the purposes of this paper, we will use the following: organization is an aggregate of processes or actions that form and improve the interrelations between the parts of a whole. In other words, to organize an activity, one should arrange it into a complete system with clearly defined characteristics, logical structure, and implementation process [2]. In the monograph [1], the process structure of activity was treated as an integral part of it, along with the logical and cause-effect (causal) structures.

1. THE PROCESS STRUCTURE OF ACTIVITY AND PROCESS THEORY

The concept of "process" is used to describe natural phenomena and is widespread in various fields of human activity. We can talk about physical, chemical, technical, technological, computational, organizational, and many other processes. The process semantics in these contexts differ, but there are some common features: a temporal character and a definite internal structure. The German *prozess* came *from Latin prōcēdēre*, meaning "to go forward, to proceed, to advance" (*prō* – "forward" + *cēdēre* "to go", as one of their meanings).

The most general definition of a process is given in the Philosophical Encyclopedic Dictionary [7]: a process is a category of philosophy characterizing an aggregate of irreversible, interrelated, long-term changes, both spontaneous and controlled, both self-organized and organized, that result in some kind of novelty or innovation (new morphological forms of organisms, new varieties, social, scientific, cultural, etc. innovations).

Processes can be divided into *natural* and *artificial* (goal-oriented). The first ones run in accordance with the laws of nature regardless of one's will; examples are physical, chemical, biological, etc., processes. Goal-oriented processes are the subject, result, or content of some human activity.

The processes of various categories are studied by the corresponding branches of science; see their simplest correlation diagram in Fig. 1.

In this paper, a process is considered exclusively in the context of the methodology of human activity: natural processes are not studied below.

National and international standards (GOST R ISO 9000, ISO/IEC 12207, and ISO/IEC 15288) define a *process* as an aggregate of interrelated or interconnected actions that transforms inputs into outputs [8–10]. However, this definition reflects only the functional and informational aspects of a process, neglecting its behavioral aspect.

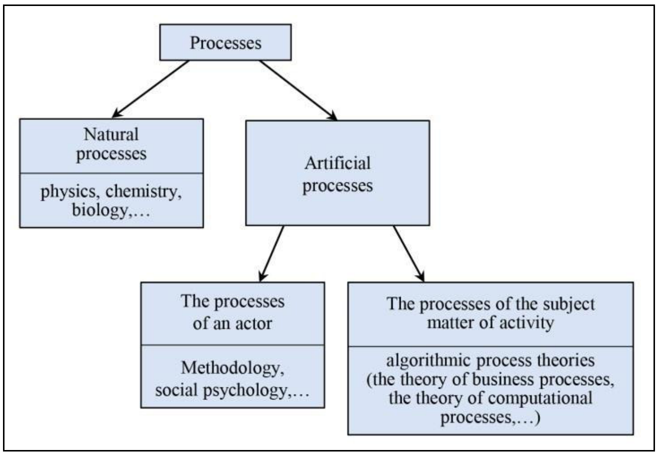


Fig. 1. Processes as an object of research.

For the purposes of this paper, we will use the following definition, going back to the definitions from the Modern Encyclopedic Dictionary and the Great Soviet Encyclopedia ("a sequential change of phenomena, states in the development of something" or "an aggregate of sequential actions to achieve some result"): a ***process*** is an aggregate of sequential states of an object and functions ensuring transitions between these states.

Note that the functional part of a process is a hierarchical tree divided into functions and operations [11, 12].

A *function* is a set of operations grouped by a definite feature that ensure, in the aggregate, the transition of an object to a new state.

An *operation* is an elementary (indivisible) action performed on an object or its element (a structural unit of action; a way to perform an action in definite conditions).

An integral part of any process is its inputs, outputs, and resources.

Process *inputs* are the objects incoming to a process to be further changed and transformed. The nature of incoming objects is quite diverse. Inputs can be something tangible (raw materials, semi-finished products) or intangible (information (data), documents, messages, requests, and services).

Process *outputs* are the objects obtained by transforming the inputs of a process. In other words, they are process results. We emphasize that new objects or processes can be born within a process.

Process *resources* are tangible or intangible objects necessary to sustain a process. Like process inputs, these objects have a very diverse nature. Resources may include equipment and tools, personnel, information, finances, space (e.g., an office), infrastructure, information systems, algorithms, technology, and many more. Note that in different processes, the same objects can be both inputs and resources or outputs.

Nowadays, rich literature on the study of processes of various categories is available. However, most authors pay their attention to only two categories of processes, *computational* [13] and *organizational/management/business* processes [11]. Appropriate theories have been developed for these categories of processes, while there are only scattered models and methods of study (in particular cases) for other categories of processes. The existing approaches and the corresponding results provide no uniform description of the processes of creation and behavior of modern complex systems. Therefore, it is topical to develop appropriate theories for other categories of processes as well as to create a general theory of processes, integrating and generalizing the above partial theories.

Proceeding from the classification of activity by its goal orientation (namely, play – learning – labor), we can speak about:

- the methodology of play activity,
- methodology of learning (educational) activity,
- the methodology of labor (professional) activity, which can be further divided into:

○ practical activity, both in the sphere of material and spiritual production, and

○ specific forms of professional activity: philosophical, scientific (research), artistic, and religious [2].

Table 1 summarizes the types of activities in accordance with their goal object.

In view of the aforesaid, the object of research in the general theory of processes is the processes of (complex) activity, and the subject matter of research is the general regularities of the organization of activity processes and control of such processes.

Table 1

The types of activity: classification by the goal object

The type of activity	Goal object
Play	Play
Educational	Human (education)
Practical	Good, service
Scientific (research)	Scientific result
Artistic	Work of art
Philosophical	Scientific result
Religious	Human (faith)

Note that the general process theory being created should provide a systematized basis for solving several problems:

- process modeling,
- engineering (design) and re-engineering of processes, including alternative solutions of these problems,
- process analysis and verification;
- process automation.

Thus, the primary problem of process theory is to create a classification system for processes, determining the place of any process and establishing the presence of interrelations with other processes.

Many experts in the field of management and production organization of various economic sectors presented classifications of processes [13–22]. One of the most complete classifications of business processes was proposed in [14] with the following six attributes as classification bases: the presence of links with the environment (internal processes and external processes); the dependence on the subject area (technological, organizational, and business processes); the relation to the main product (main processes, auxiliary processes, management business processes, network business processes, production processes, and administrative processes); scale (organization-level processes and department-level processes); the production cycle stage of goods (goods movement processes, preproduction processes, and infrastructure processes); the type of activity (processes directly ensuring the release of products, planning and management processes, resource processes, transformation processes, meta processes, and product-oriented processes).

However, the proposed classifications have common shortcomings:

– no unification, narrow focus, orientation to particular industries, fields, etc.;

– orientation exclusively on economic categories, which complicates the use of such classifications when unifying the description and formalization of various types of human activities;

– no complex integration with the subject matter (object) and the subject of activity.

2. PROCESS CLASSIFICATION BASES

As shown in Fig. 1, the artificial processes of activity are divided into two categories, namely, the processes of an actor and the processes of the subject matter of activity.

The **processes of an actor** (elementary or complex activity), common to all types of activity, are summarized in Table 2; also, see Table 1 in the monograph [1]. In fact, Table 2 describes the life cycle of an activity as a complex system evolving over time.

Further detailing is carried out for a particular activity. The process models of complex activity were discussed at length in [Chapter 5, 1] and are not studied here.

The correlation between the processes of an actor and the processes of the subject matter of activity was considered in [Chapter 2, 1].

We propose a classification for **the processes of the subject matter of activity** based on their division into the following four groups (Fig. 2).

1. **The life cycle** (LC) of a process, which includes:

- concept and design;
- production (creation and organization);
- testing;
- application;
- support, maintenance, and modernization;
- utilization.

Further detailing of the processes of this group is carried out for a particular practical activity and includes at least two additional levels.

Table 2

Phases, stages, and steps of complex activity

Phase	Stage	Step
Design	I. Fixing demand and understanding needs	1. Fixing demand and understanding needs
	II. Setting goals, structuring goals and tasks	2. Creating a logical model
	III. Selecting and developing technology	3 Checking the readiness of technology and the sufficiency of resources
		4. Creating a cause-effect model
		5. Creating technology of lower-level elements
		6. Forming/modernizing resources
		7. Calendar-network scheduling and resource planning
		8. Performing optimization
		9. Assigning actors and defining responsibilities
		10. Allocating resources
Implemen-tation	IV. Performing actions and obtaining results	11. Performing actions and obtaining results
Reflection	V. Assessing results and reflecting	12. Assessing results and reflecting

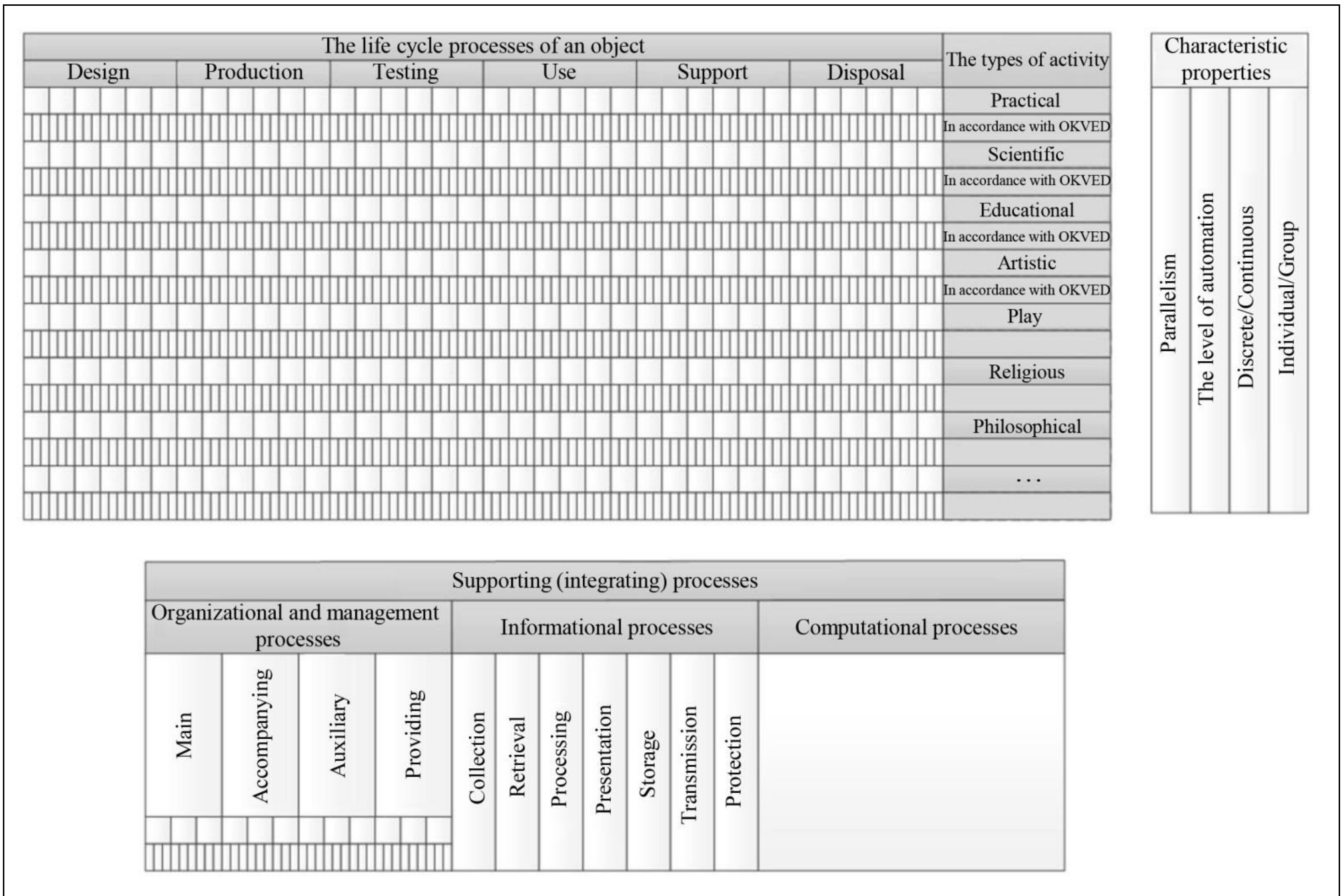


Fig. 2. The classification scheme for processes.

2. **The type of activity**, in accordance with the partial methodologies of activity reflecting the specifics of each activity within the life cycle stages of its goal object, namely:

- play activity,
- educational activity,
- practical activity,
- scientific (research) activity,
- artistic activity,
- philosophical activity,
- religious activity.

For practical, educational, scientific (research), and artistic activities, further detailing can be carried out, e.g., using the Russian Classification of Economic Activities (OKVED) [23] with at least four additional levels.

3. **Supporting** processes (integrating processes), which penetrate the processes of groups 1 and 2 above and link their functional components:

- organizational and management processes (business processes),
- informational processes, and
- computational processes.

Further detailing of such processes is also multilevel. In particular, organizational and management processes can be divided into main, accompanying, auxiliary, and providing processes; in turn, they can be further detailed in accordance with the functional hierarchy of an enterprise or organization (for example, see [6, 12]).

Main business processes are those oriented towards producing goods or rendering services that are the goal objects of enterprise formation and generate income. For example, for a fat-processing integrated works, such processes are butter production and margarine production; for a road transport enterprise, the main process is rendering transportation services.

Accompanying business processes are those oriented towards producing goods or rendering services that are the results of the enterprise's concomitant production activities and also generate income. For a fat-processing integrated works, such processes are soap production and glycerin production.

Auxiliary business processes are those intended to maintain the main and accompanying processes and oriented to support their specific features. For a road transport company, such processes are vehicle repair and maintenance and transportation safety assurance; for a fat-processing integrated works, the process of repair of oil refining equipment.

Providing business processes are those intended to ensure the main and accompanying processes and oriented to keep their universal features. For any enterprise, such processes are the financial provision of its activity, the provision of personnel, legal provision, etc.

Similarly, *informational processes* (processes related to changes in information or actions using information) can be divided into the processes of information collection, retrieval, processing, presentation, storage, transmission, and protection.

Information *collection* is a process that involves finding, gathering, and extracting primary data, which precedes the solution of any problem.

Information *retrieval* is a process of extracting stored data necessary to achieve a particular goal.

Information *processing* is a goal-oriented process of changing the content or form of information presentation. Two types of information processing can be distinguished: processing related to obtaining new content or new information (transformation by rules, including formulaic calculations), research of cognition objects by their models, logical reasoning, generalization, etc.); processing related to changing the form of information presentation without changing its content (structuring, coding, etc.).

Information *presentation* is a process of bringing information material into a more convenient form depending on the situation and addressee.

Information *storage* is a process of fixating information on some carrier, in one way or another.

Information *transmission* is a process executed according to the following scheme: information source – encoding device – communication channel – decoding device – information receiver.

Information *protection* is a process of restricting the access of unauthorized actors to information to ensure security.

4. **The characteristic properties of processes**, which any process from groups 1–3 may have, in particular:

- the level of automation (manual, automated, and automatic);
- parallelism (synchronous, asynchronous, conveyor, and sequential);
- the composition of executors (individual, group, and multi-role);
- character (discrete, continuous), etc.

The characteristic properties of processes are not a classification basis, as they merely introduce additional requirements for process description languages. For example, to describe parallel processes, one needs branch synchronization tools in the language, etc.

Within systems engineering and software engineering as the most developed disciplines for LC, the LC processes of systems fit into this scheme. For instance, the technical processes presented in the book [8] detail all the processes of elementary activities for a particular practical activity (e.g., software systems development). Agreement processes, organizational project provision processes, and project processes (except for the information management process) are organizational and management processes, and the information management process is an aggregate of information processes.

The processes included in the above classification are, in essence, elementary processes. In real life, the following *complex processes* (also called integrating processes) dominate:

• one of the processes of a particular elementary activity, represented by an element of the matrix in Fig. 2 for any level of its detail;

• one of the organizational and management processes, whose type corresponds to the selected process of a particular elementary activity;

• a set of information processes linking the functions of the selected process of a particular elementary activity;

• a set of computational processes detailing some functions of the selected process of a particular elementary activity.

Note that a computational process is a special case of a lower-level control process. The former process has the highest level of formalization and, as a rule, is modeled by a system of finite-state automata or a special state transition diagram. Therefore, one can build a set of methods to study a computational process, predict its behavior, and control the process [24].

3. AN ILLUSTRATIVE EXAMPLE OF A COMPLEX PROCESS

Figures 3 and 4 show an example of a complex process, i.e., the repair process of a motor vehicle. Consider its fragment indicated by solid lines in Fig. 3; this is a supporting process according to LC processes and a process of practical activity according to the types of activities. Further functional detailing allows attributing this process to OKVED's section 45 "Wholesale and Retail Trade in Motor Vehicles and Motorcycles and Their Repair" and, further, to OKVED's subsection 45.2 ("Maintenance and Repair of Motor Vehicles"), including (in accordance with items 45.20.0–45.20.5) the following types of works:

• repair of motor vehicles, including: mechanical repair, repair of electrical systems, injection system repair, routine maintenance of motor vehicles, body repair, chassis repair, washing and polishing, painting and drawing, windshield and window repair, car seat repair;

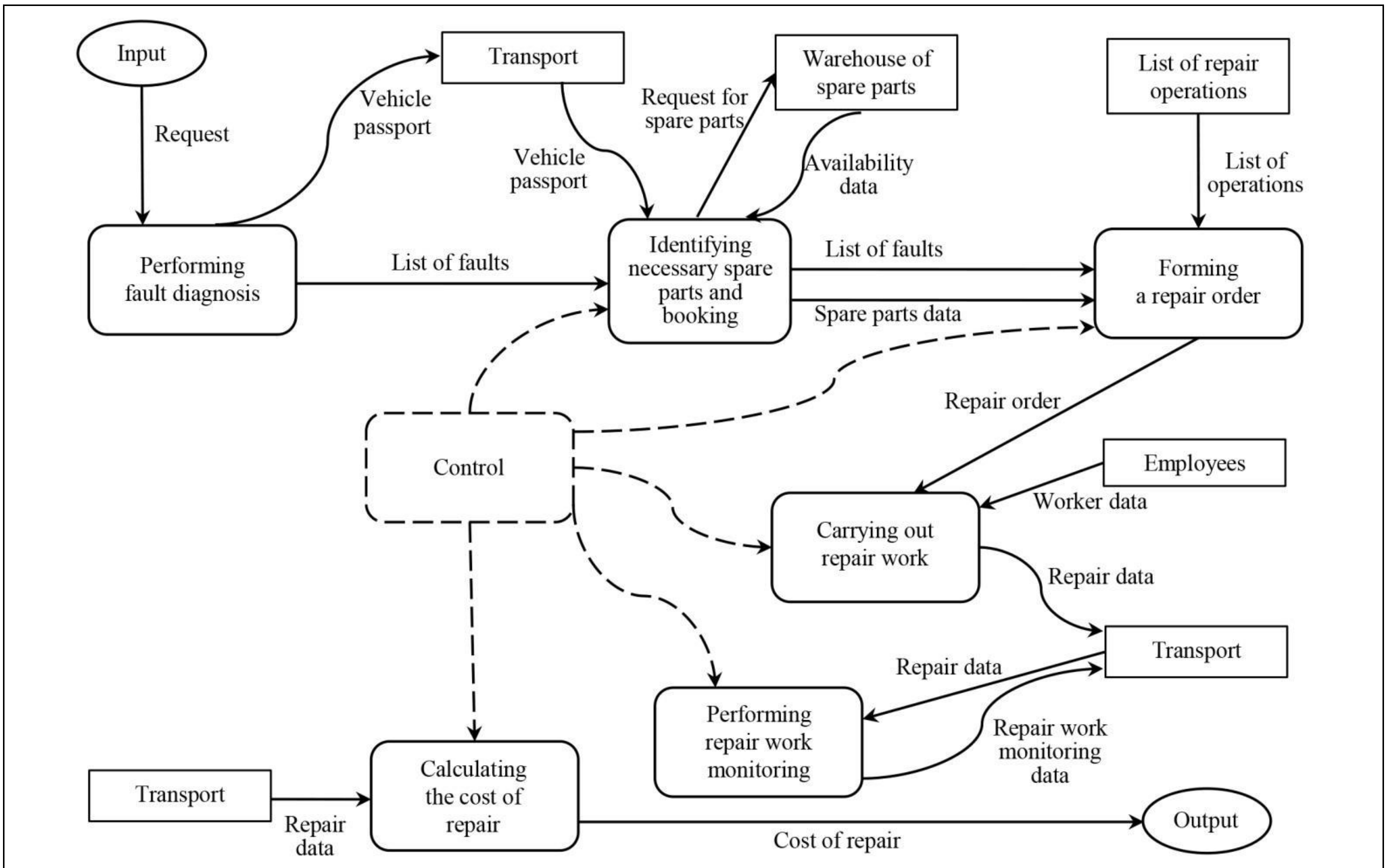


Fig. 3. The functional diagram of the motor vehicle repair process.

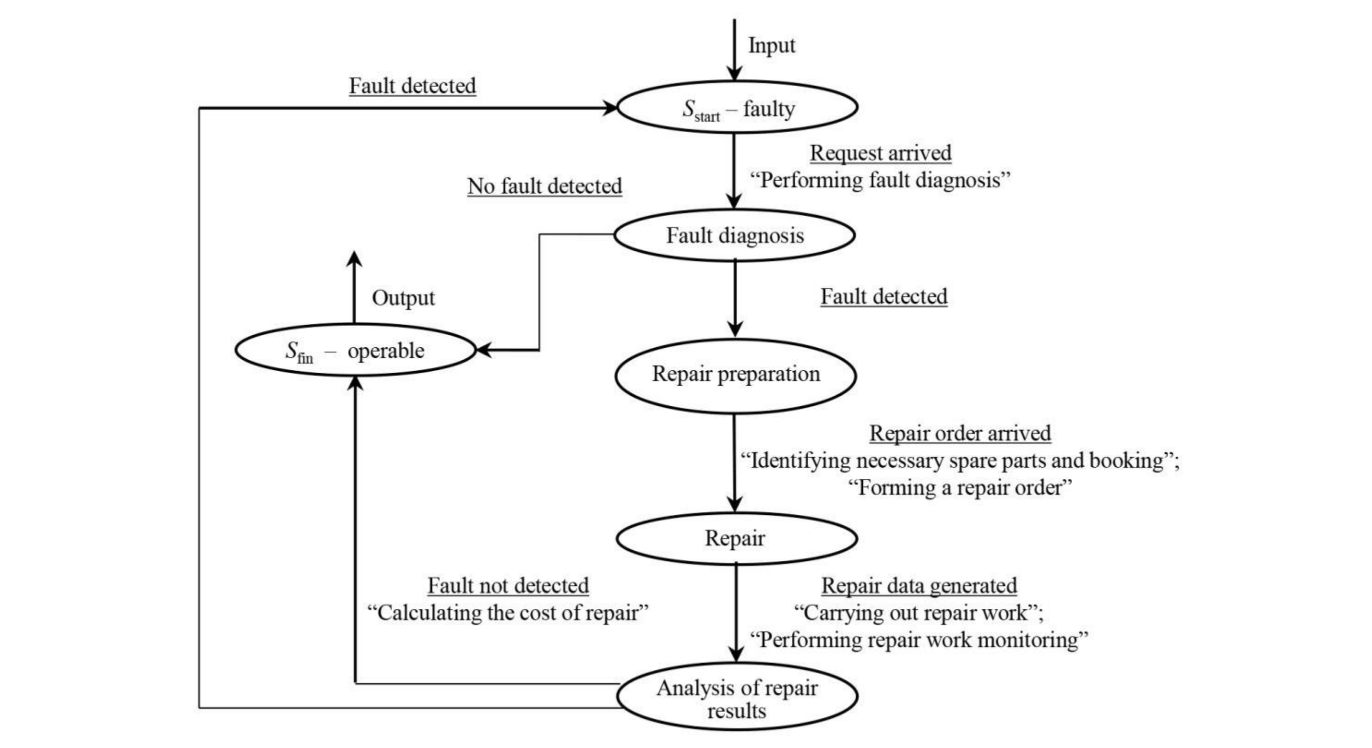


Fig. 4. The behavior diagram of the motor vehicle repair process.

- tire fitting and all kinds of related works;
- anti-corrosion treatment;
- installation of additional equipment (alarms, radio equipment, additional headlights, etc.), spare parts and accessories not directly related to the production process;
- pre-sale preparation;
- roadside assistance;
- transportation of disabled (non-runner) vehicles to the place of repair or parking.

The fragment indicated by dashed lines in Fig. 3 (and its detailing process with the state transition diagram, see Fig. 4) is an organizational and management process, and its type may vary depending on the enterprise type. For example, it is the main process for a service center and an auxiliary one for a motor depot rendering transportation services.

The fragments depicting the functions related to data storage are informational processes, and their types depend on the way the information is changed or used. Finally, the "Repair bill" fragment can be a computational process if represented by a formula for calculating the required result.

CONCLUSIONS

This paper has proposed a classification system for processes, providing a foundation for solving the relevant problems (see Section 2) within process theory and a systematized approach to study the processes of activity. In this research direction, further priority challenges include:

- development of a unified process model to integrate its functional, informational, and behavioral aspects. Such a model can be based on the corresponding models from algorithmic theories of processes (first of all, the theory of computational processes and the theory of business processes);
- development of a set of research methods for processes based on the stages and steps of its life cycle.

REFERENCES

1. Belov, M.V. and Novikov, D.A., *Methodology of Complex Activity. Foundations of Understanding and Modelling*, Cham: Springer, 2020.
2. Novikov, A.M. and Novikov, D.A., *Metodologiya* (Methodology), Moscow: SINTEG, 2007. (In Russian.)
3. Novikov, A.M. and Novikov, D.A., *Metodologiya: slovar' sistemy osnovnykh ponyatii* (Methodology: The Glossary of Basic Terms), Moscow: Librokom, 2015. (In Russian.)
4. Novikov, D.A., *Control Methodology*, New York: Nova Science, 2013.
5. Novikov, A.M. and Novikov, D.A., *Research Methodology: From Philosophy of Science to Research Design*, London: CRC Press, 2013.
6. Kalyanov, G.N., *Modelirovanie, analiz, reorganizatsiya i avtomatizatsiya biznes-protsessov* (Modeling, Analysis, Reorganization and Automation of Business Processes), Moscow: Finansy i Statistika, 2006. (In Russian.)

7. *Filosofskii entsiklopedicheskii slovar'* (Philosophical Encyclopedic Dictionary), Moscow: Sovetskaya Entsiklopediya, 1983. (In Russian.)
8. Batovrin, V.K., *Sistemnaya i programmnaya inzheneriya. Slovar'-spravochnik* (Systems and Software Engineering: Dictionary and Reference Book), Moscow: DMK Press, 2010. (In Russian.)
9. *ISO/IEC/IEEE 15288: 2023 Systems and Software Engineering – System Life Cycle Processes*, Geneva: International Organization for Standardization, 2023.
10. *ISO/IEC 12207: 2017 Systems and Software Engineering – Software Life Cycle Processes*, Geneva: International Organization for Standardization, 2017.
11. Kalyanov, G.N., *Teoriya biznes-protsessov* (Theory of Business Processes), Moscow: Goryachaya liniya – Telekom, 2023. (In Russian.)
12. Kalyanov, G.N., *Teoriya i praktika reorganizatsii biznes-protsessov* (Theory and Practice of Business Process Reorganization), Moscow: SINTEG, 2000. (In Russian.)
13. Magomedova, N.G., The Classification of the Business Processes in the Agro-Industrial Enterprises, *Voprosy Strukturizatsii Ekonomiki*, 2012, no. 1, pp. 41–45. (In Russian.)
14. Amirkhanov, K.G., Business Processes of an Industrial Enterprise: Concept and Classification, *Voprosy Strukturizatsii Ekonomiki*, 2005, no. 2, pp. 140–149. (In Russian.)
15. Ostroukhova, N.G., Peculiarities of Business Processes of Energy Companies, *Siberian Financial School*, 2012, no. 1 (90), pp. 118–122. (In Russian.)
16. Patruseva, A.M., Classification of Processes: Some Aspects of Application, *Vestnik MGSU*, 2016, no. 6, pp. 142–151. (In Russian.)
17. Frolov, T.Yu., Classification of Innovative Processes in Oil and Gas Enterprises, *Izv. Sankt-Peterburg. Ekon. Gos. Univ.*, 2016, no. 2 (98), pp. 131–134. (In Russian.)
18. Fleaca, E. and Fleaca, B., The Business Process Management Map – An Effective Means for Managing the Enterprise Value Chain, *Procedia Technology*, 2016, vol. 22, pp. 954–968.
19. Hassen, M. and Gargouri, F., Multi-dimensional Classification of Sensitive Business Process Modeling Aspects, *Procedia Computer Science*, 2024, vol. 239, pp. 2158–2167. DOI: https://doi.org/10.1016/j.procs.2024.06.404
20. Latiffianti, E., Siswanto, N., Wiratno, S., and Saputra, Y., CIMOSA Process Classification for Business Process Mapping in Non-Manufacturing Firms: A Case Study, *AIP Conference Proceedings*, 2017, vol. 1902, art. no. 020040. (Proceedings of 3rd International Materials, Industrial and Manufacturing Engineering Conference (MIMEC2017), Miri, Malaysia, 2017.) DOI: https://doi.org/10.1063/1.5010657
21. Mancioppi, M., Danylevych, O., Karastoyanova, D., and Leymann, F., Towards Classification Criteria for Process Fragmentation Techniques, in *Lecture Notes in Business Information Processing*, 2012, vol. 99, pp. 1–12. DOI: https://doi.org/10.1007/978-3-642-28108-2_1
22. Santos, R., Salgado, M., and Pereira, V., Business Process Prioritization Criteria: A Case Study in the Financial Market, *RAUSP Management Journal*, 2022, vol. 57, no. 1, pp. 35–48.
23. *OK 029 – 2014 (KDES Ed. 2): Russian Classification of Economic Activities.* Approved by Order N14-st of Rosstandard dated January 31. 2014. (In Russian.)
24. Mironov, A.M., *Teoriya protsessov* (Theory of Processes), Pereslavl-Zalessky: The University of Pereslavl, 2008. (In Russian.)

This paper was recommended for publication by O.V. Loginovskiy, a member of the Editorial Board.

Received April 14, 2025,
and revised May 21, 2025.
Accepted May 21, 2025.

Author information

Kalyanov, Georgy Nikolaevich. Dr. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ kalyanov@ipu.ru
ORCID iD: https://orcid.org/0000-0003-2429-0703

Novikov, Dmitry Aleksandrovich. Academician, Russian Academy of Sciences; Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ novikov@ipu.ru
ORCID iD: https://orcid.org/0000-0002-9314-3304

Cite this paper

Kalyanov, G.N. and Novikov, D.A., Classification of Activity Processes. *Control Sciences* **4**, 17–24 (2025).

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com

MANAGING A COMPLEX OF JOBS WITH UNCERTAIN EXECUTION REQUEST ARRIVALS

D. A. Kononov* and M. G. Furugyan**

*Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
**Federal Research Center "Computer Science and Control," Russian Academy of Sciences, Moscow, Russia

*✉ dmitrykon52@gmail.com, **✉ rtsccas@yandex.ru

Abstract. This paper considers the scheduling problem for a complex of basic jobs under the condition that at some uncertain times, execution requests for supplementary higher-priority jobs are received. If a supplementary job request arrives during the execution of a basic job, then the latter is terminated and must be restarted at some time upon the complete service of the former. All jobs (basic and supplementary) are executed without interruption. By assumption, during the execution of a basic job, two or more supplementary job requests are unlikely to arrive, and such cases are not analyzed. Also, a supplementary job request can arrive only after the complete service of the previous supplementary request. Two problem formulations are studied as follows. In the first, the performance criterion is the completion time of the basic job complex, and the problem is to minimize this time. In the second formulation, the probability of a collision is minimized, as a situation where a supplementary job request arrives during the execution of a basic job. The problems are solved via their reduction to infinite zero-sum two-player (antagonistic) games and the discrete approximation of the latter by finite games. Model examples are considered. The problem formulation with non-fixed durations of the basic jobs, linearly dependent on the amount of additional resources allocated, is investigated as well. In this case, job scheduling is reduced to a linear programming problem.

Keywords: job scheduling, collision, zero-sum two-player game, antagonistic game, non-renewable resources, mixed strategy, optimal strategy.

INTRODUCTION

Special automated control systems (ACSs) of various classes and purposes are widely used in the development, operation, and management of complex objects. Such systems are oriented to perform several control functions in given fields and have essential properties (system parameters) and characteristics reflecting the efficiency of their realization [1].

In modern conditions, it is important to develop control actions in real-time automated systems. The main characteristic of such problems is an essential upper bound on the time to process input information and output the result, in the form of control actions on the object or messages to the user. The problem becomes even more complicated under uncertainty, when it is necessary to find a schedule in a changing environment, i.e., when new (unpredicted) requests for control signals arrive.

Besides pure science, the above class of problems is of practical value. The need for fast algorithms composing multiprocessor schedules often arises in real-time distributed computing and operational control based on the processing and analysis of incoming real-time data. The following examples can serve as an illustration of the wide practical spread of real-time multiprocessor ACSs and the importance of efficient computation and control algorithms:

• Modern space monitoring systems are real-time systems that continuously process incoming data on the motion of objects in near-Earth space.

• Nuclear reactor control systems at nuclear power plants receive real-time data from many sensors and must promptly implement control actions on the reactor based on these data.

• In developed countries, real-time systems are used in government analytical centers to monitor and analyze continuous economic or environmental information coming from various points. Such systems must efficiently process huge amounts of data and, based on these data, promptly notify of any problems identified in the early stages of occurrence.

- When testing aircraft and other complex technical objects, it is crucial to receive and promptly process, by a real-time computing system, periodic input information about the state of various object nodes.
- At a modern airport with many runways, decision-making includes the assignment of aircraft to different runways as well as their takeoff and landing order.
- In complex logistics systems, decisions must be made in real time when critical situations occur.
- In emergencies, it is necessary to process incoming information urgently and calculate elimination forces and means (make schedules) in real time.

Note that the correctness of a real-time ACS depends not only on the computational accuracy but also on the time to obtain the results. Real-time scheduling is an important part of such systems: the system designer must ensure that all jobs will be executed in due time.

Along with the problem of constructing a feasible schedule for a known real-time computing system, the inverse problem is also topical: design a real-time system of some minimal possible configuration in which a feasible schedule can always be found for a given complex of jobs. This problem is crucial for onboard computing systems, which usually designed by minimizing the necessary computational resources in order to save their mass and power consumption. An algorithm for designing such systems was described in the paper [2].

Job scheduling arises in many spheres of human activity, such as construction, economics, warfare, ecology, mining, management of complex technical objects (airplanes, power plants, and nuclear reactors), transport scheduling, management of computational processes, particularly in real-time systems, and other industries. This topic was widely addressed in the literature. For example, we mention the fundamental monographs [3, 4], where various problems of scheduling theory and discrete optimization were studied, classified (into polynomially solvable and NP-hard), and solved via algorithms. In addition, the authors analyzed the computational complexity of the proposed algorithms. Based on the concept of the distance between problem instances, methods for solving several NP-hard problems with the maximum delay minimization criterion and time-optimal scheduling problems were developed in the book [5].

Various mathematical apparatus is used in scheduling problems. For example, a technique for managing computational processes with directive deadlines was described in the papers [6, 7]. The technique involves finite automata with a stopwatch and time diagrams. This approach is especially relevant in the design of real-time systems.

Job scheduling under uncertainty and risk is of great interest. For example, such problems with non-fixed parameters were investigated in [8–10]. By assumption, the durations of jobs, as well as available and required resources, are given by probabilistic characteristics or their possible ranges. In the latter case, an algorithm for partitioning the set of all possible parameter values into the so-called stability polyhedra was developed. For all parameter values belonging to each such polyhedron, the structure of the optimal schedule remains unchanged. Hence, it is possible to construct a schedule for each polyhedron in advance and choose the necessary solution in real-time computations as soon as the values of uncertain parameters become known. This approach is especially topical in the design and operation of real-time systems with strictly limited computation time.

According to the production planning methodology proposed in [11], the schedule of job execution is compiled together with the analysis of possible changes in production capacities. The original problem was reduced to a nonlinear integer mathematical programming problem. The scheduling problem of job completion dates with the stochastically varying amounts of resources required for job execution was investigated in the paper [12].

This work continues the research initiated in [13, 14]. The scheduling problem of a complex of basic jobs under uncertainty is considered. By assumption, at some uncertain times, there arrive execution requests for supplementary higher-priority jobs. Two problem formulations are studied using a game-theoretic approach. The first one is to minimize the completion time of the basic job complex. In the second formulation, the performance criterion is the probability of no collision. (A collision is a situation where a supplementary job request arrives during the execution of a basic job.) We also investigate the formulation with non-fixed durations of basic jobs, linearly depending on the amount of supplementary resources allocated for this purpose. In this case, a feasible schedule is found by solving a linear programming problem.

1. PROBLEM STATEMENT

There is a complex of basic jobs $W=\{w_1, w_2,\ldots, w_n\}$ with known durations $t_1, t_2,\ldots, t_n$ and a given execution sequence $w_1 \to w_2 \to \ldots \to w_n$.

At some uncertain times $0 \le y_1 \le y_2 \le \ldots \le y_m \le T$, there may arrive execution requests for supplementary jobs $Z = \{z_1, z_2, \ldots, z_m\}$ with known durations $\tau_1, \tau_2, \ldots, \tau_m$, respectively. The upper threshold (deadline)

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

is given.

Supplementary jobs have a higher priority than the basic ones. If a supplementary job request arrives during the execution of a basic job, then the latter is terminated and must be restarted at some time upon the complete service of the former. This situation is called a collision.

All jobs (basic and supplementary) are executed without interruption. By assumption, during the execution of a basic job, two or more supplementary job requests are unlikely to arrive, and such cases are not analyzed. Also, a supplementary job request can arrive only after the complete service of the previous supplementary request. Let the basic and supplementary jobs represent program modules solving some application tasks by an available computing device. It is a renewable resource, i.e., can be reused. When a supplementary job request arrives, this device is immediately passed for its execution. By assumption, in case of no collisions, the basic job must be completed no later than the deadline *T*. We consider two problem formulations as follows.

Problem 1. It is required to design an optimal execution strategy for the basic job complex by minimizing the completion time of the last job w_n (the entire set of jobs, including the supplementary ones).

Problem 2. Assume that the execution requests for the complex of basic jobs *W* and supplementary jobs *Z* are received repeatedly. It is required to design an optimal scheduling strategy for the basic jobs by maximizing the probability of no collision.

Note that in both formulations, the arrival of a request for supplementary job z_j becomes known only at a time y_j, $j = \overline{1, m}$. Such problems arise, e.g., during flight tests. In normal mode, computations are performed using application modules w_i, $i = \overline{1, n}$. At uncertain times y_j, $j = \overline{1, m}$, an abnormal situation may occur, e.g., the values of some important parameters may go beyond admissible limits. In this case, the computations planned are interrupted, and supplementary higher-priority jobs are executed.

We also investigate the problem formulation with non-fixed durations of basic jobs, linearly dependent on the amount of additional (non-renewable) resources allocated.

2. SOLUTION OF PROBLEM 1

First, consider the case $m = 1$. Let a request for supplementary job *z* arrive at a time *y*, and let its duration be τ. We introduce a zero-sum two-player (antagonistic) game with a payoff function $F(x_1, x_2, \ldots, x_n, y)$. In this game, the strategy of the first player determines the times x_i to start the basic jobs $w_i \in W$, $i = \overline{1, n}$, and the strategy of the second player determines the arrival time *y* of the request for the supplementary job *z*. The payoff function is defined as follows:

$$F(x_1, x_2, \ldots, x_n, y) = \begin{cases} x_n + t_n \text{ if } 0 \le y \le x_1 - \tau, \\ \text{or } x_k + t_k \le y \le x_{k+1} - \tau \\ \text{for some } 1 \le k \le n-1, \\ \text{or } x_n + t_n \le y \le T; \\ \\ y + \tau + \sum_{i=k}^{n} t_i \\ \text{if } x_k \le y < x_k + t_k \text{ for some } 1 \le k \le n \\ \text{or } x_k - \tau < y \le x_k \text{ for some } 1 \le k \le n-1. \end{cases}$$

In other words, in the absence of a collision, we have $F(x_1, x_2, \ldots, x_n, y) = x_n + t_n$. If a collision occurs during the execution of some job $w_k \in W$,

$$F(x_1, x_2, \ldots, x_n, y) = y + \tau + \sum_{i=k}^{n} t_i.$$

This means that the optimal guaranteeing strategy of the first player to schedule the execution of the basic job complex *W* is $x_1^0 = 0$, $x_i^0 = x_{i-1}^0 + t_{i-1}$, $i = \overline{2, n}$. In this case, all the basic jobs *W* will be completed at the time $x_n^0 + t_n$ (no collision) or $x_n^0 + t_n + \tau$ (collision occurrence). With any other strategy $x_1, x_2, \ldots, x_n$, in the worst case, the basic job complex *W* will be completed at the time $x_n + t_n + \tau >$ $x_n^0 + t_n + \tau$ since $x_n > x_n^0$. Thus, the strategy $(x_1^0, x_2^0, \ldots, x_n^0)$ is the optimal guaranteeing strategy under $m = 1$.

Consider an illustrative example.

Example 1. For $n = m = 1$,

$$F(x,\ y) = \begin{cases} x+t \\ \text{if } 0 \le y \le x-\tau, \\ \text{or if } x+t \le y \le T \\ \text{(no collision);} \\ \\ y+\tau+t \\ \text{if } x-\tau < y < x+t \\ \text{(collision occurrence).} \end{cases}$$

The graph of this payoff function is shown in Fig. 1.

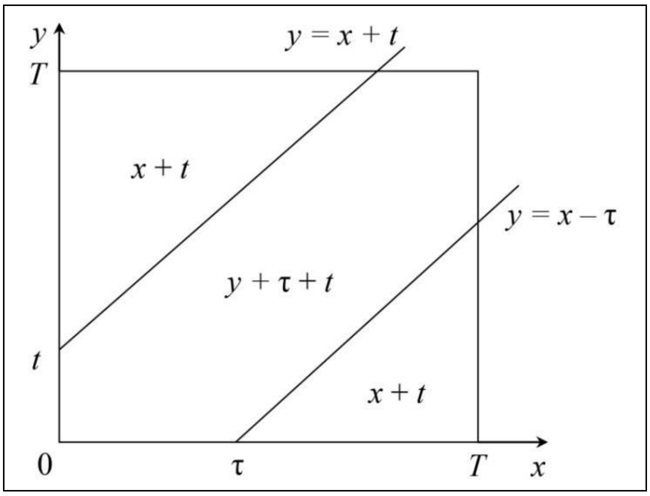


Fig. 1. The payoff function in Problem 1 with *n* = *m* = 1.

In this case, we have $F(0,\ y) < 2t + \tau$ for all $y \in [0,\ T]$ and $F(x,\ y) > 2t + \tau$ for $x \ne 0$ and any $y \in (x - \tau,\ x + t)$. Hence, $x^0 = 0$ is the optimal guaranteeing strategy of the first player. ♦

The case $n > 1,\ m > 1$ is considered by analogy. As a scheduling strategy we choose $\left(x_1^0,\ x_2^0, \ldots,\ x_n^0\right)$. In the absence of a collision on the interval $\left[x_1^0,\ x_1^0 + t_1\right]$, we fix x_1^0 and let $x_2^0 = x_1^0 + t_1$. If a collision occurs on the interval $\left[x_1^0,\ x_1^0 + t_1\right]$ at the time y_1, we let $x_1^0 = y_1 + \tau_1$. Similarly, in the first case, the interval $\left[x_2^0,\ x_2^0 + t_2\right]$ is examined for collisions; in the second case, the interval $\left[x_1^0,\ x_1^0 + t_1\right]$, and so on. Thus, the optimal job scheduling strategy *W* is constructed dynamically, depending on the arrival of supplementary job requests.

3. SOLUTION OF PROBLEM 2

Now we define the payoff function of the antagonistic game as follows:

$$F(x_1,\ x_2, \ldots,\ x_n,\ y_1,\ y_2, \ldots,\ y_m) = \begin{cases} 1 \text{ if } x_i + t_i \le y_j \le x_{i+1} - \tau_j \\ \text{for all } i = \overline{1,\ n} \text{ and } j = \overline{1,\ m}, \\ \text{or } 0 \le y_j \le x_1 - \tau_j, \\ \text{or } x_n + t_n \le y_j \le T; \\ \\ 0 \text{ if } x_i \le y_j \le x_i + t_i \\ \text{for some } 1 \le i \le n,\ 1 \le j \le m, \\ \text{or } x_i - \tau_j \le y_j \le x_i. \end{cases}$$

Let $\bar{x} = (x_1,\ x_2,\ \ldots,\ x_n)$ and $\bar{y} = (y_1,\ y_2,\ \ldots,\ y_m)$. Then $F(\bar{x},\ \bar{y}) = 1$ if there is no collision and $F(\bar{x},\ \bar{y}) = 0$ otherwise. Let $X = \{\bar{x} :\ x_i + t_i \le x_{i+1},\ i = \overline{1,\ n-1},\ x_1 \ge 0,\ x_n \le T\}$, $Y = \{\bar{y} :\ y_j + \tau_j \le y_{j+1},\ j = \overline{1,\ m-1},\ y_1 \ge 0,\ y_m \le T\}$, and $f(\bar{x})$ be the mixed strategy of the first player, i.e., a probability measure on the set *X*. By the definition of the payoff function $F(\bar{x},\ \bar{y})$,

$$E(f,\ \bar{y}) = \int_X F(\bar{x},\ \bar{y})\, df(\bar{x})$$

is the probability of no collision under a fixed $\bar{y}$. The optimal mixed strategy of the first player, $f^0(\bar{x})$, maximizes the value of

$$\min_{\bar{y} \in Y} E(f,\ \bar{y}):$$

$$\max_f \min_{\bar{y} \in Y} E(f,\ \bar{y}) = \min_{\bar{y} \in Y} E(f^0,\ \bar{y}),$$

where the maximum is taken over all probability measures on the set *X*. In other words, $f^0(\bar{x})$ is the best guaranteeing strategy of the first player.

Example 2. Consider the case $n = m = 1$, $X = [0,\ 1]$, $Y = [0,\ 1]$, and $T = 1$. We will write $x,\ y,\ t,$ and τ instead $x_1,\ y_1,\ t_1,$ and τ_1, respectively. Let $t \le 0.25$ and $\tau \le 0.25$. Then

$$F(x,\ y) = \begin{cases} 1 \text{ if } y \ge x + t \\ \text{or } y \le x - \tau; \\ \\ 0 \text{ if } x - \tau < y < x + t \end{cases},$$

$$x \in [0,\ 1],\ y \in [0,\ 1].$$

The graph of the payoff function *F* is shown in Fig. 2.

Let $f^0(x)$ be a probability measure uniformly distributed on the interval [0, 1]. By the definition of the payoff function $F(x,\ y)$, we have

$$\int_0^1 F(x,\ y)\,df^0(x) \geq 1-\tau-t$$

for all $y\in[0,\ 1]$. If the probability measure $f(x)$ is not uniformly distributed on the interval [0, 1], then there exists a segment $[x_0,\ x_0+\tau+t]\subseteq[0,\ 1]$ such that

$$\int_{x_0}^{x_0+\tau+t} df(x) > \tau+t.$$

Hence, there exists a value $y\in[0,\ 1]$ for which

$$\int_0^1 F(x,\ y)\,df(x) < 1-\tau-t.$$

Therefore,

$$\int_0^1 F(x,y)\,df^0(x) = \max_f \min_{y\in[0,1]} \int_0^1 F(x,y)\,df(x),$$

where the maximum is taken over all probability measures on the interval [0, 1]. So, $f^0(x)$ is the optimal mixed strategy of the first player. ♦

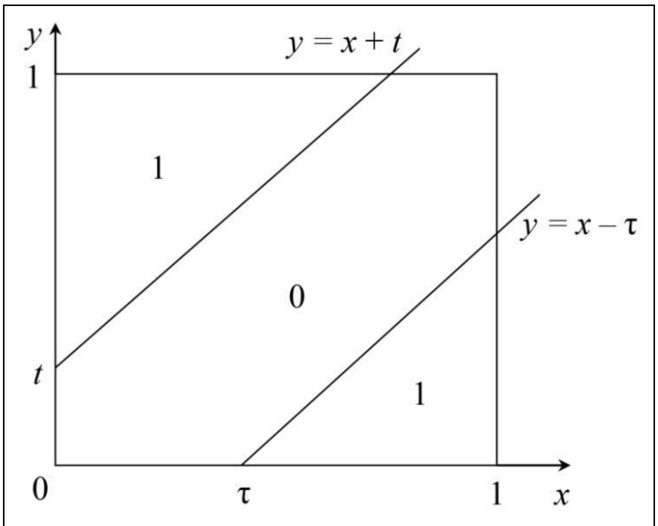


Fig. 2. The payoff function in Problem 2 with *n* = *m* = 1, *t* ≤ 0.25, τ ≤ 0.25, and *T* = 1.

Next, we can apply the discrete approximation method of an infinite game by a finite one [15]. For any $\varepsilon>0$, this method yields the ε-optimal mixed strategy of the first player, $f_\varepsilon(\bar{x})$, concentrated in a finite number of points. Let $f_\varepsilon(\bar{x})$ be concentrated in points $v_1, v_2, \ldots, v_p$ with jumps

$$q_1,\ q_2,\ldots,\ q_p,\ q_j>0,\ j=\overline{1,\ p},\ \sum_{j=1}^{p} q_j = 1.$$

Each point v_j, $j=\overline{1,p}$, is associated with some schedule for the job set W, which should be executed with probability q_j.

Example 3. Let $n=m=1$, $X=[0,\ 1]$, $Y=[0,\ 1]$, $T=1$, $t=0.25$, and $\tau=0.25$. The payoff function has the following form:

$$F(x,\ y)=\begin{Bmatrix} 1 \text{ if } y\geq x+0.25 \\ \text{or } y\leq x-0.25; \\ \\ 0 \text{ if } x-0.25<y<x+0.25 \end{Bmatrix},$$
$$x\in[0,\ 1],\ y\in[0,\ 1].$$

According to Example 2, $f^0(x)$ is the optimal mixed strategy of the first player and

$$\int_0^1 F(x,\ y)\,df^0(x) \geq 1-0.25-0.25=0.5$$

for all $y\in[0,\ 1]$. Let $\bar{f}(x)$ be a probability measure on the segment [0, 1] concentrated in two points, $x=0$ and $x=1$, with jumps of 0.5. Then

$$\int_0^1 F(x,\ y)\,d\bar{f}(x) \geq 0.5$$

for all $y\in[0,\ 1]$. Hence, like $f^0(x)$, $\bar{f}(x)$ is the optimal mixed strategy of the first player. Thus, the job w_1 should be started at time 0 or time 1 equiprobably (with probability 0.5).

The graph of the payoff function *F* is shown in Fig. 3.

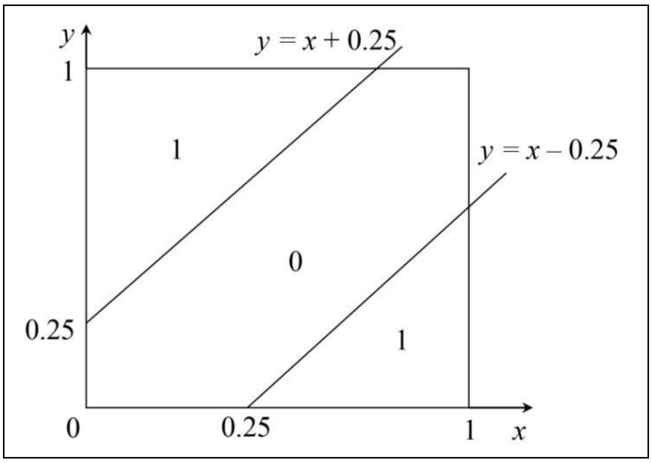


Fig. 3. The payoff function in Problem 2 with *n* = *m* = 1, *t* = 0.25, τ = 0.25, and *T* = 1.

4. THE PROBLEM WITH NON-FIXED JOB DURATIONS

In this section, we assume the availability of *L* types of additional non-renewable resources to execute the basic job complex *W*, in amounts $R_1,\ R_2,\ldots,\ R_L$, respectively. (Non-renewability means that the resources cannot be reused.) If a job $w_i\in W$ is allocated an amount r_{il} of the lth resource type, $l=\overline{1, L}$, its duration will be reduced to

$$t_i = t_i^0 - \sum_{l=1}^{L} a_{il} r_{il}, \; i=\overline{1, n},$$

where a_{il} are given nonnegative numbers; t_i^0 denotes the duration of this job without additional resources allocated. By assumption, the following constraints hold:

$$0 \le r_{il} \le r_{il}^0, \; i=\overline{1, n}, \; l=\overline{1, L}, \quad (1)$$

$$\sum_{i=1}^{n} r_{il} \le R_l, \; l=\overline{1, L}, \quad (2)$$

$$t_i^0 - \sum_{l=1}^{L} a_{il} r_{il}^0 > 0, \quad (3)$$

where r_{il}^0, $i=\overline{1, n}$, $l=\overline{1, L}$, are given positive numbers (the maximum admissible amounts of resources that can be allocated to the job). Inequalities (1) limit the amounts of each resource type that can be allocated to each job. Next, inequalities (2) limit the total amount of each resource type allocated to all jobs together. Finally, inequalities (3) limit the durations of the jobs. A resource allocation r_{il}, $i=\overline{1, n}$, $l=\overline{1, L}$, is called feasible if conditions (1)–(3) are valid.

The objective is to determine a feasible resource allocation facilitating the solution of Problems 1 and 2. According to Sections 2 and 3, such a resource allocation minimizes the total duration of the job complex *W*. Thus, we arrive at the following linear programming problem:

$$\min_{r_{il}, i=\overline{1,n}, l=\overline{1,L}} \sum_{i=1}^{n} \left(t_i^0 - \sum_{l=1}^{L} a_{il} r_{il} \right)$$

subject to the constraints (1) and (2). The solution of this problem will give the optimal feasible resource allocation.

CONCLUSIONS

In this paper, we have studied the scheduling problem for a complex of basic jobs under the condition that at some uncertain times, execution requests for supplementary higher-priority jobs are received. The execution sequence of the basic jobs is fixed. If a supplementary job request arrives during the execution of a basic job, then the latter is terminated and must be restarted at some time upon the complete service of the former. All jobs (basic and supplementary) are executed without interruption. Two problem formulations have been considered. In the first, the performance criterion is the completion time of the basic job complex, and the problem is to minimize this time. In the second formulation, the probability of a collision is minimized, as a situation where a supplementary job request arrives during the execution of a basic job. The problems have been solved via their reduction to infinite zero-sum two-player (antagonistic) games and the discrete approximation of the latter by finite games. The scheduling method has been illustrated on model examples. Also, the problem formulation with non-fixed durations of the basic jobs, linearly dependent on the amount of additional resources allocated, has been investigated. In this case, a feasible schedule is found by solving a linear programming problem.

The results of this paper can be used to plan computations during the testing and operation of complex technical objects (such as airplanes and nuclear reactors). In the planned mode, computations are performed using application modules, and an abnormal situation may occur at uncertain times (e.g., the values of some parameters may go beyond an admissible range). In this case, scheduled computations are interrupted and supplementary higher-priority jobs are executed.

Scheduling problems under uncertainty were studied in [8–10] under the assumption of renewable resources and the non-fixed values of some parameters (such as job durations or the amounts of available resources). The parameters were defined through either their admissible ranges or probabilistic characteristics. The solution algorithms were based on the branch-and-bound method. In contrast to the cited works, this paper has addressed a scheduling problem with uncertain request arrivals. Also, the case of additional non-renewable resources has been investigated. Problems with a heterogeneous set of resources were considered in [13, 14] in the deterministic setup.

In the future, we intend to analyze a more general problem formulation with several computing devices for basic and supplementary jobs.

REFERENCES

1. Kononov, D.A., Security Research of Control Systems Based on the Analysis of Their System Parameters, *Trudy 28-oi Mezhdunarodnoi konferentsii "Problemy upravleniya bezopasnost'yu slozhnykh sistem"* (Proceedings of the 28th International Conference on Problems of Complex Systems Security Control), Moscow, 2020, pp. 102–108. (In Russian.)
2. Kononov, D. and Furugyan, M., Control of a Complex of Works in Multiprocessor Real-Time ACS, *Proceedings of the 1st International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA2019)*, Lipetsk, 2019. URL: https://ieeexplore.ieee.org/document/8947570.
3. Tanaev, V.S., Gordon, V.S., and Shafranskii, Ya.M., *Teoriya raspisanii. Odnostadiinye sistemy* (Scheduling Theory: Single-Stage Systems), Moscow: Nauka, 1984. (In Russian.)

4. Brucker, P., *Scheduling Algorithms*, Heidelberg: Springer, 2007.
5. Lazarev, A.A., *Teoriya raspisanii. Metody i algoritmy* (Scheduling Theory. Methods and Algorithms), Moscow: Trapeznikov Institute of Control Sciences RAS, 2019. (In Russian.)
6. Glonina, A.B. and Balashov, V.V., On the Correctness of Real-Time Modular Computer Systems Modeling with Stopwatch Automata Networks, *Modeling and Analysis of Information Systems*, 2018, vol. 25, no. 2, pp. 174–192. https://doi.org/10.18255/1818-1015-2018-2-174-192 (In Russian.)
7. Glonina, A.B., Tool System for Testing Real-Time Constraints for Modular Computational System Configurations, *Moscow Univ. Comput. Math. Cybern.*, 2020, vol. 44, no. 3, pp. 120–132. https://doi.org/10.3103/S0278641920030036
8. Koshelev, P.S. and Mishchenko, A.V. Optimizing Management of Jobs in a Logistic Project under Conditions of Uncertainty, *Journal of Computer and Systems Sciences International*, 2018, vol. 60, no. 4, pp. 595–609.
9. Gorskii, M.A., Mishchenko, A.V., Nesterovich, L.G., and Khalikov, M.A., Some Modifications of Integer Optimization Problems with Uncertainty and Risk, *Journal of Computer and Systems Sciences International*, 2022, vol. 61, no. 5, pp. 813–823.
10. Kosorukov, O.A., Lemtyuzhnikova, D.V., and Mishchenko, A.V., Methods and Models of Project Resource Management under Uncertainty, *Journal of Computer and Systems Sciences International*, 2023, vol. 62, no. 2, pp. 305–323.
11. Yao, X., Almatooq, N., Askin, R.G., and Gruber, G., Capacity Planning and Production Scheduling Integration: Improving Operational Efficiency via Detailed Modelling, *Intern. J. Production Research*, 2022, vol. 60, no. 1, pp. 7239–7261.
12. Graves, S.C., How to Think About Planned Lead Times, *Intern. J. Production Research*, 2022, vol. 60, no. 1. DOI: http://dx.doi.org/10.2139/ssrn.3485059
13. Kononov, D.A., Furugyan, M.G. Planning a Complex of Works with Heterogeneous Resources under Uncertainty, *Proceedings of 2022 15th International Conference on Management of Large-Scale System Development (MLSD)*, Moscow, 2022. DOI: 10.1109/MLSD55143.2022.9934381
14. Kononov, D.A. and Furugyan, M.G., Effective Means of Regional Management: Optimal Use of Heterogeneous Resources, *Proceedings of 2021 14th International Conference on Management of Large-Scale System Development (MLSD'2021)*, Moscow, 2021. DOI: 10.1109/MLSD52249.2021.9600251
15. Furugyan, M.G., On Approximating the Solution of a Certain Class of Infinite Antagonistic Games, *Vestn. Mosk. Gos. Univ. Ser. 15. Vych. Mat. Kibern.*, 1978, no. 2, pp. 81–85. (In Russian.)

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

Received April 4, 2025, and revised June 30, 2025. Accepted July 16, 2025

Author information

Kononov, Dmitry Alekseevich. Dr. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ dmitrykon52@gmail.com
ORCID iD: https://orcid.org/0000-0002-6059-5590

Furugyan, Meran Gabibullaevich. Cand. Sci. (Phys.–Math.), Federal Research Center "Computer Science and Control," Russian Academy of Sciences, Moscow, Russia
✉ rtsccas@yandex.ru
ORCID iD: https://orcid.org/0000-0003-0373-9233

Cite this paper

Kononov, D.A. and Furugyan, M.G., Managing a Complex of Jobs with Uncertain Execution Request Arrivals. *Control Sciences* **4,** 25–31 (2025).

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com

SUSTAINABLE DEVELOPMENT MECHANISMS FOR A TRANSPORT CORPORATION WITH SUPERVISED LEARNING

V. V. Tsyganov

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

✉ bbc@ipu.ru

Abstract. Sustainable development of a transport corporation is achieved through a consistent and balanced improvement of its economic and environmental performance indicators. This is largely facilitated by reducing fuel consumption by corporate vehicles, which saves costs and decreases harmful emissions into the environment. In this regard, a three-level structure for managing fuel consumption in a transport corporation is considered under uncertainty caused by both random external factors and the undesirable activity of corporate employees associated with their private goals not coinciding with the goal of the entire corporation. The top level of this structure is the Principal, the middle level is the Intendant, responsible for fuel consumption in a corporate division, and the bottom level is the Driver of a corporate vehicle. The sustainable behavior of corporate employees—the Intendant and Driver—to reduce fuel consumption within their competence is modeled. An organizational mechanism for the Intendant's sustainable behavior is developed, including supervised learning procedures for the Principal and incentive procedures for the Intendant. An organizational mechanism for the Driver's sustainable behavior is also proposed: the Principal provides the Intendant with supervised learning opportunities and delegates his authority to incentivize the Driver. As proven, the integration of these two mechanisms reduces fuel consumption and ensures sustainable development of the transport corporation. In addition, the sustainable behavior of corporate employees—the Intendant and Driver—ensures the sustainability of the entire corporation. The results are illustrated by an example of the fuel consumption management mechanism in JSC Russian Railways.

Keywords: transport, corporation, sustainable development, control, digitalization, supervised learning, sustainable behavior.

INTRODUCTION

Sustainable development is "a development that meets the needs of the present without compromising the ability of future generations to meet their own needs" [1]. The current global sustainable development goals [2] include:

– efficient use of resources,

– reduction of emissions harmful to human health and the environment,

– establishment of planning and management mechanisms related to climate change.

Among these goals, the creation of cost-saving and environmentally friendly transport systems stands out [2]. Transport accounts for 20–25% of global energy consumption and carbon dioxide emissions [3]. Almost 97% of harmful substance emissions of transport into the atmosphere are gases formed during fuel combustion [4]. They include nitrogen oxides and particulate pollutants, which negatively impact human health [5], as well as carbon oxides and hydrocarbons, which are greenhouse gases [6]. Among all sectors of the economy, the transport industry is characterized by the fastest growth of greenhouse gas emissions [7].

Thus, both efficiency and environmental impact are important for the sustainable development of transport [8]. To achieve global goals, it is necessary for companies to apply sustainable business practices [2]. In this regard, ESG—the concept of sustainable corporate development based on the principles of environmental responsibility (E), social responsibility (S), and effective corporate governance (G)—has become widespread in recent decades [9]. Accordingly, the sustainable development goals of a transport corporation are related to the improvement of its economic and environmental performance indicators by both improving technologies and activating, training, and developing employees; for example, see [10, 11]. The study of the related psychological aspects of sustainable development was initiated in [12]. Further research led to the concept of sustainable behavior [13]. Such behavior implies the employee's understanding of the importance of increasing production efficiency while preserving the environment. The consequence of sustainable behavior is a thrifty attitude to the resources and environment, which brings satisfaction and intrinsic motivation of employees [13].

Note that the vast majority of corporate vehicles (CVs) are equipped with internal combustion engines. These vehicles account for the greater part of the fuel consumption by a corporation. For example, up to 80% of the fuel of JSC Russian Railways is consumed by locomotives with diesel engines (diesel shunters) [10]. Therefore, fuel saving by CVs directly reduces the resource utilization and costs of a corporation as well as its harmful emissions in exhaust gases (including greenhouse gases). On this basis, transport corporations strive to minimize CV fuel consumption by training and activating employees; for example, see [10, 11]. Inculcating employees with the psychology of sustainable behavior helps to conserve CV fuel, thereby reducing both operating costs and pollutant emissions, as well as ensuring compliance with environmental regulations.

Nevertheless, motionless CVs with idling engines are regularly encountered. The point is that in the planning practice of a large transport corporation, the future limits (norms) of fuel consumption by a corporate division and an individual employee are usually lowered when the current fuel consumption decreases. However, the lower the limits (norms) are, the less fuel will be available to a corporate division and an employee to fulfill their tasks. Since fuel consumption depends on random factors, a corporate division and an employee may fail to fulfill these tasks under unfavorable circumstances, with all the ensuing negative consequences. Therefore, a farsighted division manager, as well as an experienced driver, may be uninterested in reducing fuel consumption below the norm. This is a typical problem of adaptive planning from the achieved level, which is studied within the theory of control in organizations [14]. For this kind of problems, a solution approach is to design organizational mechanisms for the operation of corporations [15]. For example, an incentive mechanism to implement environmental requirements for locomotives in JSC Russian Railways was considered in [16].

This paper develops the models of training, activation, and sustainable behavior of the manager and employee of a corporate division, with a focus on reducing the fuel consumption of CVs, and the corresponding organizational mechanisms of sustainable development of a transport corporation.

1. THE SUSTAINABLE BEHAVIOR OF THE INTENDANT

1.1. General Assumptions

Fuel in a corporation is consumed both to meet the needs of CVs (e.g., trucks or diesel shunters) and for miscellaneous demands. These needs and demands depend on random external and internal factors, and many of the latter become known only to a narrow circle of involved persons directly in the course of performing their duties. Assume accordingly that the person responsible for fuel consumption in a corporation (further called the Principal) has inaccurate information about the real possibilities of reducing fuel consumption in his division and CVs. But the Principal can take advice from the Consultant. Both of them get information about the factual fuel consumption of a division. In addition, the Principal can set consumption norms. Based on the deviation of the factual fuel consumption from the norms, the Principal can incentivize the person responsible for fuel consumption in the division (further called the Intendant). However, the Principal cannot determine whether the factual fuel consumption in the division is the minimum achievable in the current conditions.

In view of the aforesaid, we consider a three-level system for managing fuel consumption in a corporation, with a control authority (the Principal) at the top level, a person responsible for fuel consumption in a corporate division (the Intendant) at the middle level, and a CV driver (the Driver) at the bottom level. For this system, it is required to build an organizational mechanism minimizing fuel consumption under uncertainty caused by both random external factors and the undesirable activity of corporate employees (the Intendant and Driver), associated with their private goals not coinciding with the Principal's goal.

We introduce the following notation: t is the time period, $t = 0, 1, \ldots$; d_t is the fuel consumption of the division in period t, consisting of the fuel consumption c_t of the CV and the miscellaneous fuel consumption s_t of the division: $d_t = c_t + s_t$. The value of c_t is reported by the Driver to the Intendant in period t. By assumption, $c_t \in C_t = \left[L(t),\ \gamma\right]$, where

$L(t)$ is a stationary random process determining the realization (value) l_t of the random variable of the minimum CV fuel consumption in period t, $L(t) \in \Lambda = [\delta, \varepsilon]$ with $\delta > 0$ and $\varepsilon \le \gamma$. Hence, $c_t \in \mathrm{C} = [\delta, \gamma]$, $t = 0, 1, ...$

The miscellaneous fuel consumption s_t in period t is specified by the Intendant, $s_t \in S_t = [M(t), \alpha]$. Here, $M(t)$ is a stationary random process determining the realization (value) m_t of the random variable of the minimum miscellaneous fuel consumption, $M(t) \in \mathrm{M} = [\mu, \beta]$ with $\mu > 0$ and $\beta \le \alpha$. Hence, $s_t \in \Sigma = [\mu, \alpha]$, $t = 0, 1, ...$

As $d_t = c_t + s_t$, we obtain $d_t \in \Psi_t = [A(t), \alpha + \gamma]$, where $A(t) = L(t) + M(t)$ is a stationary random process determining the realization (value) a_t of the random variable of the minimum fuel consumption by the division, $a_t = l_t + m_t$, $t = 0, 1, \ldots$ It follows from $c_t \in \mathrm{C} = [\delta, \gamma]$ and $s_t \in \Sigma = [\mu, \alpha]$ that $d_t \in \Delta = [\delta + \mu, \alpha + \gamma]$.

Suppose that in period t, the Principal knows the factual fuel consumption d_t of the division. However, he is unaware of the realization (value) a_t of the random variable of the division's minimum fuel consumption. To incentivize the Intendant's fuel consumption reduction under uncertainty, the Principal assigns him one of two categories, namely, 1 (reasonable consumption) and 2 (unreasonable consumption), using the known consumption d_t, $t = 0, 1, \ldots$ This has to be done with minimum losses.

1.2. The Complete and Partial Awareness of the Principal

In this subsection, the Principal is assumed to know, in period t, the realization a_t of the random variable of the minimum fuel consumption of the division, $a_t = l_t + m_t$, $t = 0, 1, \ldots$ This realization is characterized by the dimensionless relative rate $e_t = a_t / (\alpha + \gamma), 0 < e_t \le 1, e_t \in D = [(\delta + \mu)/(\alpha + \gamma), 1]$, $t = 0, 1, \ldots$ The problem is to assign category 1 or 2 to the Intendant by relating e_t to one of two unknown subsets D_1 and D_2 forming the set D: $D_1 \cup D_2 = D$.

Under complete awareness, the Principal knows the rate e_t to belong, with a conditional distribution density $\epsilon(e_t \mid \eta)$ and a prior probability ζ_η, $\eta = \overline{1, 2}$, to one of the two subsets D_1 and D_2, $D_1 \cup D_2 = D$. We denote by q_{12} the losses due to the Principal's erroneous relation of e to the subset D_2 if e actually belongs to the subset D_1. Similarly, q_{21} is the losses due to the Principal's erroneous relation of e to the subset D_1 if e actually belongs to the subset D_2.

We introduce a classification parameter p separating the subsets D_1 and D_2: $e \in D_1 = [(\delta + \mu)/(\alpha + \gamma), p]$ if $e \le p$ and $e \in D_2 = (p, 1]$ otherwise. Then the optimal dichotomy $\{D_1, D_2\}$ is determined by solving the following optimization problem over p, with the mean categorization losses as the objective function:

$$\sum_{\eta=1}^{2} \sum_{\theta=1}^{2} q_{\eta\theta} \int_{D_\theta} \zeta_\eta \epsilon(e \mid \eta) de \underset{\{D_1, D_2\}}{\rightarrow} \min. \tag{1}$$

Here, θ is the variable of summation, $\theta = \overline{1, 2}$, and D_θ, $\theta = \overline{1, 2}$, are subsets of D_1 and D_2.

Under partial awareness, the Principal does not know the above probability characteristics. Hence, the value of the parameter p cannot be determined by solving problem (1). However, by observing the random realization a_t, $t = 0, 1, \ldots$, of the division's minimum fuel consumption, it is possible to obtain and sequentially refine the estimates p_t of the value p, using the supervised learning algorithm with the help of the Consultant [17]. Knowing the fuel consumption rate (e_t), the Consultant reports his opinion $B(e_t)$ to the Principal: the rate is excessive ($B(e_t) = 1$) or not ($B(e_t) = 0$):

$$B(e_t) = \begin{cases} 1 \text{ if } e_t > p \\ 0 \text{ if } e_t \le p, \end{cases} \tag{2}$$
$$t = 0, 1, ...$$

where $B(\cdot)$ is the consulting procedure. Then the Principal can apply the supervised learning algorithm [17]

$$p_{t+1} = P(p_t, e_t) = p_t - \iota_t [p_t - 0.5 - q_{12} + (q_{12} + q_{21}) B(e_t)], \tag{3}$$
$$p_0 = p^0, \ t = 0, 1, ...$$

Where $\iota_t \in \mathrm{I} = \left\{ \iota_t > 0 \mid \iota_t > \iota_{t+1}, \sum_{\sigma=1}^{\infty} \iota_\sigma < \infty \right\}$. If $e_t \le p_t$, the Principal will assign category 1 to the Intendant; otherwise, category 2. Thus, the Intendant's category is given by

$$k_t = G(p_t, e_t) = \begin{cases} 1 \text{ if } e_t \le p_t \\ 2 \text{ if } e_t > p_t, \end{cases} \quad (4)$$
$$t = 0, 1, ...$$

1.3. The Unawareness and Supervised Learning of the Principal

Suppose now that the Principal and Consultant are unaware of the above probabilities and the minimum fuel consumption a_t. However, they know the factual fuel consumption d_t. We introduce a relative rate of this consumption: $i_t = d_t / (\alpha + \gamma)$, $0 < i_t \le 1$, $i_t \in D = [(\delta + \mu)/(\alpha + \gamma), 1]$, $i_t \ge e_t$, $t = 0, 1, \ldots$ Then the Consultant determines his opinion using the procedure (2) with the observed rate i_t instead of the unknown e_t. More precisely, having obtained i_t, the Consultant informs the Principal of his opinion $B(i_t)$: the consumption is accurate $(B(i_t) = 1)$ or not $(B(i_t) = 0)$:

$$B(i_t) = \begin{cases} 1 & \text{if } i_t > p \\ 0 & \text{if } i_t \le p, \end{cases} \quad (5)$$
$$t = 0, 1, ...$$

Accordingly, the Principal calculates an estimate b_t of the parameter p_t using formula (5), replacing the unknown e_t in algorithm (3) with the observed one i_t:

$$b_{t+1} = P(b_t, i_t) = b_t - \iota_t [b_t - 0.5 - q_{12} + (q_{12} + q_{21}) B(i_t)], \quad (6)$$
$$b_0 = p^0, t = 0, 1, ...,$$

where $P(\cdot)$ is the learning procedure. By comparing i_t and b_t similar to formula (4), the Principal determines the fuel consumption category g_t of the Intendant's division:

$$g_t = G(b_t, i_t) = \begin{cases} 1 \text{ if } i_t \le b_t \\ 2 \text{ if } i_t > b_t, \end{cases} \quad (7)$$
$$t = 0, 1, ...,$$

where $G(\cdot)$ is the categorization procedure. The value of b_t can be interpreted as the normative value of the rate i_t, depending on which the Intendant's category is assigned. Since $i_t = d_t / (\alpha + \gamma)$, the value $h_t = b_t(\alpha + \gamma)$ means the fuel consumption threshold for the division (d_t). If this threshold is not exceeded (i.e., $d_t \le h_t$), the Intendant will receive category 1 $(g_t = 1)$ and an incentive; otherwise, $g_t = 2$ and the Intendant will not be incentivized (or even penalized).

The consulting (5), learning (6), and categorization (7) procedures make up the Principal's organizational mechanism $\Phi = (B, P, G)$ for managing the division's fuel consumption.

1.4. The Intendant's Goals and Decisions under the Principal's Supervised Learning

Using the mechanism $\Phi = (B, P, G)$, the Principal learns with the prompting of the Consultant, assigns a category, and incentivizes the Intendant based on the observation d_t and the calculated rate i_t, $t = 0, 1, ...$ However, the Intendant is more aware of the fuel consumption than the Principal and can use this to his advantage. By assumption, at the beginning of period t, the Intendant knows the factual CV fuel consumption c_t and the realization m_t of the random variable of the minimum miscellaneous fuel consumption. The Intendant then selects a value of d_t from the condition $d_t \ge f_t \equiv c_t + m_t$ so that $d_t \in Y_t = [f_t, \beta + \gamma]$ and $f_t \in H = [\delta + \mu, \beta + \gamma]$, $t = 0, 1, \ldots$

Consider how the Intendant makes his decisions under the mechanism $\Phi = (B, P, G)$. A farsighted Intendant seeks to choose d_t in period t to improve both current and future categories. For this purpose, he may control the division's miscellaneous fuel consumption s_t and the CV fuel consumption c_t. Formally, the Intendant chooses d_t by the desire to increase the utility of the categories in period t:

$$T_t = T[g_t, g_{t+1}, ..., g_{t+\lambda}],$$
$$T_t \downarrow g_\nu, \nu = \overline{1, \lambda}, t = 0, 1, \ldots, \quad (8)$$

where $T[\cdot]$ is a monotonically decreasing function of its arguments (the utility of any category for the Intendant decreases with its number) and λ is the Intendant's foresight.

According to the procedure (7), the category g_t depends only on the Intendant's choice of the fuel consumption d_t, $d_t \in Y_t$. However, to increase the current category utility (8), the Intendant should consider the impact of d_t on the category g_ξ in a future period ξ, $\xi = \overline{t+1, t+\lambda}$. Due to formula (7), this category g_ξ depends on the Intendant's choice d_ξ and the estimate b_ξ in the period. In view of the expression (6), the estimate b_ξ depends on the Intendant's choice d_ω and the estimates b_ω in the previous periods ω, $\omega=\overline{t, \xi-1}$. With formula (6) treated as a recurrence relation, it can be easily established that the category g_ξ depends on the Intendant's previous choices d_ω, $\omega=\overline{t, \xi-1}$.

Note that at the beginning of period ξ, the Intendant will know the realization $f_\xi = c_\xi + m_\xi$, $f_\xi \in H$, $\xi=\overline{t+1, t+\lambda}$. Suppose that the Intendant will also know the future CV fuel consumption c_σ and the miscellaneous fuel consumption m_σ, $\sigma=\overline{\xi+1, t+\lambda}$. Then the Intendant can choose d_ξ by maximizing the objective function (8) with the known realization f_σ, $\sigma=\overline{\xi+1, t+\lambda}$ subject to the condition $d_\xi \in Y_\xi$, $\xi=\overline{t+1, t+\lambda}$. As shown above, the Intendant should consider the impact of d_ξ on the future categories g_σ, $\sigma=\overline{\xi+1, t+\lambda}$.

Repeating this reasoning sequentially from $\xi=t+\lambda$ to $\xi=t+1$, we obtain that it is reasonable for the farsighted Intendant to predict the choice d_ξ backwards. As a result, the objective function (8) is transformed into the predicted category utility

$$T_t^o = T_t^o\left(g_t,\ f_{t+1},.., f_{t+\lambda}\right) = \max_{d_{t+1}\in Y_{t+1}} \max_{d_{t+2}\in Y_{t+2}} \dots \max_{d_{t+\lambda}\in Y_{t+\lambda}} T_t, \quad (9)$$

which depends on the realizations f_ξ, $\xi=\overline{t+1, t+\lambda}$.

However, the Intendant is unaware of the future CV fuel c_ξ and miscellaneous fuel m_ξ consumptions. Therefore, to choose a value of d_t that will increase the predicted category utility (9), the Intendant should predict the fuel consumptions $f_\xi = c_\xi + m_\xi$, $\xi=\overline{t+1, t+\lambda}$. Let the Intendant be guided by the principle of maximum guaranteed result [14, 15], expecting the worst-case predictions $f_\xi \in H$, $\xi=\overline{t+1, t+\lambda}$. Then the function (9) is transformed into the Intendant's objective function

$$\Gamma_t\left(d_t\right) = \min_{f_{t+1}\in H} \max_{d_{t+1}\in Y_{t+1}} \min_{f_{t+2}\in H} \max_{d_{t+2}\in Y_{t+2}} \dots \min_{f_{t+\lambda}\in H} \max_{d_{t+\lambda}\in Y_{t+\lambda}} T_t. \quad (10)$$

In this case, the set of his optimal choices d_t^* maximizing the objective function (10) in period *t* has the form

$$\Xi_t = \left\{d_t^* \in \Delta \mid \Gamma_t\left(d_t^*\right) \geq \Gamma_t\left(d_t\right), d_t \in \Delta\right\},\quad t=0, 1,... \quad (11)$$

Below, we accept the hypothesis of the Intendant's benevolence towards the Principal: if $f_t \in \Xi_t$, then $d_t^* = f_t$, $t=0, 1,\dots$ This means that the division consumes more fuel only if it increases the Intendant's objective function (10).

1.5. An Organizational Mechanism for the Sustainable Behavior of the Intendant

The mechanism $\Phi=\left(B, P, G\right)$ is designed to ensure the Intendant's sustainable behavior, aimed at improving both the economic and environmental performance indicators of the corporation. Therefore, such behavior should be expressed in terms of the Intendant's desire to reduce the division's fuel consumption. This is formally reflected as follows.

Definition 1. The Intendant's behavior is sustainable if, for any given CV fuel consumption c_t, the division's fuel consumption is minimal:

$$d_t^* = f_t,\ t=0, 1,...\blacklozenge \quad (12)$$

From a practical viewpoint, as already indicated, large corporations often plan fuel consumption from the achieved level. In this case, the future norm of this consumption is reduced when the factual fuel consumption decreases. In the model under consideration, this means that the norm b_{t+1} (the normative value of the rate i_{t+1} in period $t+1$) is reduced when the current fuel consumption rate i_t decreases. However, according to the procedure (6), the lower the norm b_{t+1} is, the smaller value the fuel consumption rate i_{t+1} in period $t+1$, sufficient for the Intendant to receive the highest category, will take. Since $i_{t+1} \geq e_{t+1}$, where e_{t+1} is a random variable, the Intendant may receive a lower category under unfavorable circumstances. Therefore, the farsighted Intendant may be uninterested in reducing the fuel consumption rate i_t below the norm b_t. (He gets the highest category in period *t* for fulfilling this norm.) Thus, a control mechanism is needed to incentivize the Intendant's sustainable behavior.

Proposition 1. *Under the mechanism* $\Phi=\left(B, P, G\right)$, *the Intendant's behavior is sustainable*.

P r o o f. By formula (5), $B\left(i_\xi\right)$ is a non-decreasing function of i_ξ under the mechanism $\Phi=\left(B, P, G\right)$,

$\xi = \overline{t+1, t+\lambda}$. Due to formula (6), $b_{\zeta+1}$ does not grow with increasing $B(i_\zeta)$ and, hence, the same property applies to $b_{\zeta+1}$ when increasing i_ζ, $\zeta = \overline{t, t+\lambda-1}$. Furthermore, according to the expression (7), $g_\xi = G(b_\xi, i_\xi)$ does not grow with increasing b_ξ, $\xi = \overline{t+1, t+\lambda}$. It follows from the above considerations that $g_\xi = G(b_\xi, i_\xi)$ is a non-decreasing function of i_t, $\xi = \overline{t+1, t+\lambda}$.

In addition, by formula (7), $g_t = G(b_t, i_t)$ is a non-decreasing function of i_t. Thus, all arguments of the function $T[g_t, g_{t+1}, ..., g_{t+\lambda}]$ do not decrease with increasing i_t. Therefore, according to the expression (8), T_t does not grow with increasing i_t for any realizations $c_{t+1}, .., c_{t+\lambda}$, $m_{t+1}, .., m_{t+\lambda}$. Then, due to formula (10), the function $\Gamma_t(d_t)$ achieves maximum at the minimum value of i_t. As $i_t = d_t / (\alpha+\gamma)$, the function $\Gamma_t(d_t)$ achieves maximum at the minimum value of d_t. Recalling that $d_t = c_t + s_t$, we obtain $d_t = f_t \in \Xi_t$ by the expression (11). On the other hand, $f_t \in \Xi_t$ implies $d_t^* = f_t$ based on the hypothesis of the Intendant's benevolence towards the Principal. Hence, by definition (12), the Intendant's behavior is sustainable. ♦

Under the mechanism $\Phi = (B, P, G)$, the division's fuel consumption reduction, first, does not worsen the Intendant's current category (7) and, second, does not tighten the fuel consumption rates in the future. Being benevolent to the Principal, the Intendant minimizes the division's fuel consumption (12).

However, the Intendant's desire to minimize the division's fuel consumption is not enough. In practice, it is common to see motionless CVs with idling engines. To avoid this, the driver of each CV should have a vested interest in minimizing fuel consumption.

2. THE SUSTAINABLE BEHAVIOR OF THE DRIVER

2.1. The Unawareness and Supervised Learning of the Intendant

With the mechanism $\Phi = (B, P, G)$, the Principal ensures the sustainable behavior of the Intendant aimed at minimizing the fuel consumption $f_t = c_t + m_t, t = 0, 1, ...$ Here, c_t is the fuel consumption of the CV reported by the Driver, $c_t \in C_t = [l_t, \gamma]$ (see subsection 1.1). But the Intendant does not know the realization l_t of the minimum CV fuel consumption. The fuel consumption is reduced to the minimum value if $c_t = l_t, t = 0, 1, ...$

Suppose that the minimum CV fuel consumption (l_t) becomes known to the Intendant at the beginning of period t. Since the Intendant is unaware of l_t, the Driver can manipulate the CV fuel consumption c_t, choosing $c_t > l_t$ if it is profitable for him. On the other hand, the Intendant should minimize the CV fuel consumption $c_t, t = 0, 1, \ldots$ To do this, the Principal provides the Intendant with supervised learning capabilities and delegates to the Intendant the right to establish a Driver's operation mechanism similar to $\Phi = (B, P, G)$. (The Principal uses this mechanism to control the Intendant, see subsection 1.3.)

We introduce the following dimensionless relative rates to measure the CV fuel consumption:

$$k_t = l_t / \gamma, \; K_t = c_t / \gamma, \quad t = 0, 1, \ldots \tag{13}$$

Then the Intendant should reduce the value of the rate K_t to the minimum k_t. Suppose that for this purpose, the Intendant classifies the Driver's performance depending on the CV fuel consumption as satisfactory (class 1) or unsatisfactory (class 0). Incorrect classification incurs costs. To improve the validity of his decisions, the Intendant learns classification with the help of the Estimator, in the same way as the Principal learns categorization with the help of the Consultant (see subsection 1.3).

We denote by ρ_{10} the losses due to the incorrect assignment of class 0 to the Driver (although Driver deserves class 1) and by ρ_{01} the losses due to the incorrect assignment of class 1 to him. The Estimator observes the CV fuel consumption (c_t), calculates K_t by formula (13), and reports his opinion $R(K_t)$ to the Intendant (whether the CV fuel consumption in period t is excessive or not). If the Estimator considers the consumption to be excessive, then $R(K_t) = 1$; otherwise, $R(K_t) = 0$. This is formally written as

$$R(K_t) = \begin{cases} 1 \text{ if } K_t > \varsigma \\ 0 \text{ if } K_t \le \varsigma, \end{cases} \tag{14}$$
$$t = 0, 1, \ldots,$$

where $R(\cdot)$ denotes the estimation procedure and ς is the estimation parameter, $\varsigma > 0$. Note that the estimation procedure (14) is similar to the consulting procedure (5). With the considerations of Section 1 repeated, a supervised learning algorithm similar to the procedure (6) can be used to minimize the average classification losses. The tunable classification parameter (briefly, the norm u_τ) is calculated using a recurrence procedure similar to (6):

$$u_{t+1} = U(u_t, K_t) = u_t - \varrho_t [u_t - 0.5 - \rho_{01} + (\rho_{01} + \rho_{10}) R(K_t)], \quad (15)$$
$$u_0 = u^0, t = 0, 1, \ldots,$$

where $U(u_t, K_t)$ is the normalization procedure and $\varrho_t \in \mathrm{P} = \left\{ \varrho_t > 0 \mid \varrho_t > \varrho_{t+1}, \sum_{\sigma=1}^{\infty} \varrho_\sigma < \infty \right\}$, $t = 0, 1, \ldots$ Comparing K_t and u_t, by analogy with the procedure (7), the Intendant determines the Driver's class (v_t):

$$v_t = V(u_t, K_t) = \begin{cases} 1 \text{ if } K_t \le u_t \\ 0 \text{ if } K_t > u_t, \end{cases} \quad (16)$$
$$t = 0, 1, ...,$$

where $V(\cdot)$ is the classification procedure. The value u_t means the norm of the rate K_t used to classify the Driver. Since $K_t = c_t / \gamma$, the value $n_t = \gamma u_t$ can be interpreted as the threshold of the Driver's fuel consumption (c_t). If this threshold is not exceeded ($c_t \le n_t$), the Driver will receive class 1 $(v_t = 1)$ and an incentive. The estimation (14), normalization (15), and classification (16) procedures make up the Intendant's organizational mechanism $\Pi = (R, U, V)$ to manage the Driver's fuel consumption.

2.2. The Goals and Decisions of the Driver Given the Unaware Intendant

Consider how the Driver makes decisions under the mechanism $\Pi = (R, U, V)$. The norm u_t in this mechanism is interpreted as an upper bound for the CV fuel consumption rate acceptable to the Intendant. If the fuel consumption rate K_t does not exceed the norm $u_t (K_t \le u_t)$, the Driver will be assigned class 1 ($v_t = 1$) and incentivized. Therefore, the utility of the farsighted Driver increases with current and υ future classes:

$$Z_t = Z[v_t, v_{t+1}, ..., v_{t+\upsilon}],$$
$$Z_\varphi \uparrow v_\varphi, \varphi = \overline{t, t+\upsilon}, t = 0, 1, ... \quad (17)$$

According to formulas (13)–(16), the utility (17) depends on the CV fuel consumption c_φ in period φ, $\varphi = \overline{t, t+\upsilon}$. In this case, the Driver knows the current realization of the random variable l_t but not the future realizations l_χ, $\chi = \overline{t+1, t+\upsilon}$.

Suppose that the Driver is aware only of the inclusions $l_\chi \in \Lambda$ and $c_\chi \in C_\chi$, $\chi = \overline{t+1, t+\upsilon}$. When eliminating the uncertainty regarding l_χ and c_χ, $\chi = \overline{t+1, t+\upsilon}$, the Driver is guided by the principle of maximum guaranteed result [14, 15]. By repeating the considerations of Section 1, we establish that the objective function $\Theta_t(c_t)$ of the farsighted Driver is equal to the maximum guaranteed utility (17):

$$\Theta_t(c_t) = \min_{l_{t+1} \in \Lambda} \max_{c_{t+1} \in C_{t+1}} \min_{l_{t+2} \in \Lambda} \max_{c_{t+2} \in C_{t+2}} \ldots \min_{l_{t+\lambda} \in \Lambda} \max_{c_{t+\lambda} \in C_{t+\upsilon}} Z_t. \quad (18)$$

Accordingly, in period t, the set of rates c_t^* maximizing the objective function (18) is

$$\Omega_t = \{c_t^* \in C_t \mid \Theta_t(c_t^*) \ge \Theta_t(c_t), c_t \in C_t\}, \quad t = 0, 1, \ldots \quad (19)$$

Let us accept the hypothesis of the Driver's benevolence to the Intendant: if $c_t \in \Omega_t$, then $c_t^* = l_t$, $t = 0, 1, ...$ This means that the CV consumes more fuel only when increasing the Driver's objective function (18).

2.3. An Organizational Mechanism for the Sustainable Behavior of the Driver

The Intendant's goal is to reduce the fuel consumption c_t to the minimum l_t. This can be prevented by the Intendant's planning from the achieved level (see the discussion above). In the case under consideration, this means that the norm u_{t+1} is reduced as the fuel consumption rate K_t decreases. However, according to formula (16), the lower the norm u_{t+1} is, the less fuel the Driver should consume to obtain class 1 in period $t+1$ ($v_{t+1} = 1$). Since $K_{t+1} \ge k_{t+1}$, and the minimum possible fuel consumption value k_{t+1} is a random variable, under unfavorable circumstances the Driver may obtain class 0 ($v_{t+1} = 0$). For these reasons, the farsighted Driver may be uninterested in reducing the fuel consumption rate K_t below the norm u_t.

Definition 2. The Driver's behavior is sustainable if the CV fuel consumption in each period is minimal:

$$c_t^* = l_t, t = 0, 1, \ldots \blacklozenge \quad (20)$$

Proposition 2. *Under the mechanism* $\Pi = (R, U, V)$*, the Driver's behavior is sustainable.*

P r o o f. With the mechanism $\Pi = (R, U, V)$ and the procedure (16), the current class $v_t = V(u_t, K_t)$ is not reduced with decreasing K_t. Consider the dependence of the future class $v_\varphi = V(u_\varphi, K_\varphi)$, $\varphi = \overline{t+1, t+\upsilon}$, on K_t. According to the expression (14), the estimate $R(K_\varphi)$ in period φ does not grow with decreasing K_φ, $\varphi = \overline{t+1, t+\upsilon}$. Due to formula (15), the norm u_φ is not reduced with decreasing $R(K_{\varphi-1})$. In addition, by the expression (16), the class $v_\varphi = V(u_\varphi, K_\varphi)$ is not reduced with decreasing u_φ. Based on the three monotonic dependencies above, the class $v_\varphi = V(u_\varphi, K_\varphi)$ in period φ is not reduced with decreasing K_t for $\varphi = \overline{t+1, t+\upsilon}$.

According to formula (17), the Driver's objective function (18) increases with the growth of the current and future classes v_φ, $\varphi = \overline{t, t+\upsilon}$. From the considerations of the previous paragraph it follows that this objective function $\Theta_t(K_t)$ does not decrease with decreasing K_t. On the other hand, as $K_t \ge k_t$, we obtain $\Theta_t(k_t) \ge \Theta_t(K_t)$, $K_t \in \mathrm{K}_t$, and, by definition (19), $k_t \in \Omega_t$. Hence, based on the Driver's benevolence hypothesis, $K_t^* = k_t$, and the expression (13) leads to formula (20). ♦

3. SUSTAINABILITY MECHANISMS FOR A TRANSPORT CORPORATION

Definition 3. A transport corporation is sustainable if the fuel consumption of its division (including the CV fuel consumption) in each period is minimal:

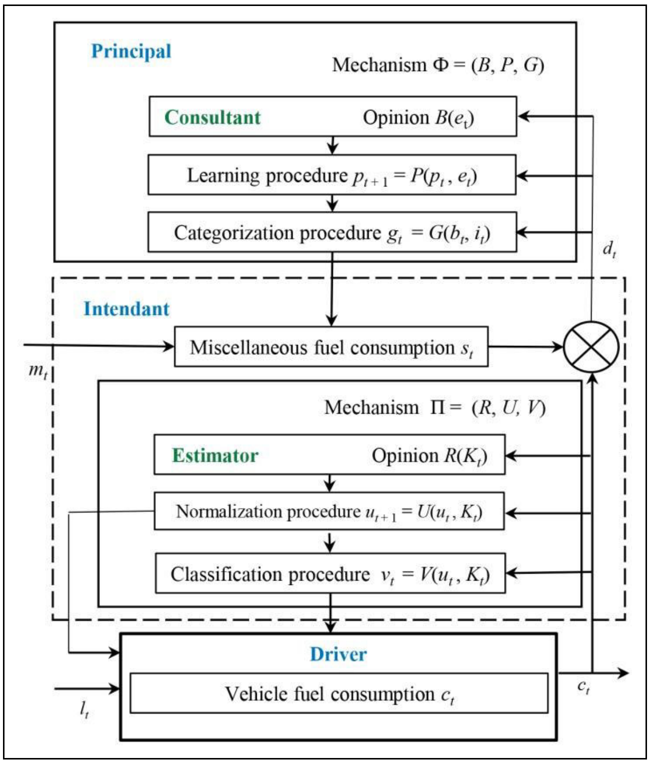
$$d_t^* = a_t,\ t = 0,1,\ldots \ ♦ \tag{21}$$

Proposition 3. *For the sustainability of a transport corporation, it is sufficient to apply the integrated mechanism* $\Sigma = (\Phi, \Pi)$ *consisting of the mechanisms* $\Phi = (B, P, G)$ *and* $\Pi = (R, U, V)$.

P r o o f. Under the mechanism $\Phi = (B, P, G)$, the Intendant's behavior is sustainable (Proposition 1). By Definition 1, this means that for any given CV fuel consumption c_t, the fuel consumption of the division is minimal: $d_t^* = f_t$, $t = 0,1,\ldots$

Under the mechanism $\Pi = (R, U, V)$, the Driver's behavior is sustainable (Proposition 2). By Definition 2, the CV fuel consumption in each period is minimal: $c_t^* = l_t$, $t = 0, 1,\ldots$ Substituting the last equality into formula (12) and using $f_t = c_t + m_t$ and $a_t = l_t + m_t$, we finally arrive at the expression (21). ♦

Actually, the proof of Proposition 3 is based on the fact that the sustainable behavior of the corporate employees (the Intendant and Driver) ensures the sustainability of the entire corporation. Consequently, the integrated mechanism $\Sigma = (\Phi, \Pi)$ can be called the organizational mechanism of the transport corporation's sustainability. The structure of this mechanism is illustrated by the figure below.



The organizational structure and sustainability mechanism of a transport corporation.

4. AN ILLUSTRATIVE EXAMPLE: A SUSTAINABILITY MECHANISM FOR A RAILWAY CORPORATION

Large-scale railway corporations design their own strategies for sustainable (cost-saving and environmentally friendly) development based on the ESG principles; for example, see [10, 11]. To attract passengers and shippers, they need to reduce the fuel consumption of diesel locomotives. This leads to significant cost reductions for interested persons while reducing the environmental load.

Consider such an approach to improving the sustainability of a railway corporation using the example of JSC Russian Railways, further also referred to as the holding. Since about 80% of fuel in the holding is consumed by diesel locomotives [10], they account for about 80% of fuel costs and about 80% of environmental damage due to the emissions of exhaust gases (including greenhouse gases). Therefore, reducing the fuel consumption of diesel locomotives is crucial for the sustainable development of Russian Railways.

Organizational mechanisms for reducing fuel consumption by the holding's locomotives were developed in the paper [18]. These mechanisms include supersized learning procedures for corporate employees [19]. The models and methods for managing the creation and implementation of innovative means and technologies for reducing fuel consumption by the holding's locomotives were presented in the report [10]. These R&D works resulted in a mechanism for reducing fuel consumption that ensures the sustainable development of Russian Railways [10, 18, 19]. The mechanism is an analog of the integrated mechanism presented in the figure.

Consider this integrated mechanism at the regional level. Here, the Principal is an official of the regional railway traction directorate (a branch of Russian Railways) responsible for reducing fuel consumption on the branch's railway network. The Intendant is an employee of the locomotive operation depot (LOD) of this branch responsible for fuel consumption (in short, the manager), and the Driver is a diesel locomotive motorman assigned to the LOD. The responsible persons mentioned are assisted by advisors (as a rule, appointed from former responsible persons).

As a CV we take TEM18DM, a shunting diesel locomotive manufactured by Bryansk Machine Building Plant, a division of CJSC Transmashholding (hereinafter briefly referred to as a/the shunter) [20]. About two hundred shunters of this brand operate on the Oktyabrskaya Railway. Consider the organizational mechanism $\Pi = (R, U, V)$ as applied to the interaction between the manager and driver in the LOD of the Oktyabrskaya Railway when operating a shunter (e.g., in the "Saint Petersburg–Finlandsky" LOD [20]).

According to Proposition 2, under the mechanism $\Pi = (R, U, V)$, the motorman's behavior (as the Driver) is sustainable. Therefore, to reduce fuel consumption, it is sufficient for the LOD manager (as the Intendant) to apply this mechanism, e.g., during the weekly control of fuel consumption in the LOD. Assume that the motorman reports to the manager the average hourly fuel consumption of his shunter over the last week. We denote by t the week number, $t = 0, 1, \ldots$ The hourly fuel consumption of TEM18DM (c_t) varies within 10–12 l/h [20]. Therefore, according to subsection 1.1, we set $\delta = 10$, $\gamma = 12$, and $c_t \in \mathrm{C} = [10, 12]$, $t = 0, 1, \ldots$

Thus, within the mechanism $\Pi = (R, U, V)$, the manager and estimator use the average data on hourly fuel consumption over the previous week (c_t). Following (13), we introduce the dimensionless relative rate of fuel consumption by the shunter: $K_t = c_t / 12$, $5/6 < K_t \le 1$, $t = 0, 1, \ldots$ Next, according to formula (14), the estimator informs the manager of his opinion $R(K_t)$: $R(K_t) = 1$ if the consumption is excessive or $R(K_t) = 0$ otherwise. The manager then calculates the fuel consumption norm u_t of the shunter using the procedure (15). Finally, the manager determines the motorman's class v_t by the procedure (16).

The estimation (14), normalization (15), and classification (16) procedures make up the organizational mechanism $\Pi = (R, U, V)$ used by the manager to reduce the shunter's fuel consumption. To perform model calculations, we suppose that at the end of week t during the subsequent quarter (i.e., $t = \overline{0, 11}$), the motorman reports to the manager the average hourly fuel consumption c_t of his shunter (see the top row of the table). To calculate $R(K_t)$, v_t, u_t by formulas (14)–(16), we take the following parameter values of the mechanism $\Pi = (R, U, V)$: $\varsigma = 0.92 \cong 11/12$, $u_0 = 0.90$, $\rho_{01} = 0.5$, $\rho_{10} = 1$, and $\varrho_t = 1/(t+10)$, $t = \overline{0, 11}$. In this case, formulas (14) and (15) turn into:

$$R(K_t) = \begin{cases} 1 \text{ if } K_t > 0.92 \\ 0 \text{ if } K_t \le 0.92, \end{cases} \quad t = \overline{0, 11}, \tag{22}$$

$$u_{t+1} = \left\{ u_t (t+9) + 1.5 \left[1 - R(K_t) \right] \right\} / (t+10), \quad u_0 = 0.9, \; t = \overline{0, 11}. \tag{23}$$

The resulting values of v_t, $R(K_t)$, and u_t calculated by formulas (16), (22), and (23), respectively, for 12 weeks of the quarter ($t = \overline{0, 11}$) are given in the table.

This example illustrates the simplicity and transparency of the organizational mechanism $\Pi = (R, U, V)$ as well as the applicability of the proposition.

Fuel consumption by the shunter and the calculated values of the organizational mechanism for the quarter

t	0	1	2	3	4	5	6	7	8	9	10	11
c_t	10.7	11.1	11.0	11.3	10.9	11.0	11.2	10.8	11.2	10.7	10.6	11.1
K_t	0.89	0.93	0.92	0.94	0.91	0.92	0.93	0.90	0.93	0.89	0.88	0.93
v_t	1	1	0	0	0	0	1	0	0	0	1	1
$R(K_t)$	0	1	0	1	0	0	1	0	1	0	0	1
u_t	0.90	0.96	0.87	0.92	0.85	0.90	0.94	0.88	0.92	0.87	0.90	0.93

To ensure the sustainability of Russian Railways, the requirements for the holding's locomotives in the field of environmental protection were supplemented with this organizational mechanism, including the flexible adjustment of fuel consumption rates and norms for the branch and depot level of the traction management hierarchy, and incentives for the employees engaged.

CONCLUSIONS

To develop sustainably, a transport corporation should reduce both economic costs and environmental damage due to its operations. The fuel economy of corporate vehicles significantly reduces both the operating costs of the corporation and the emission of pollutants into the environment. Therefore, the sustainability of a transport corporation depends on the effectiveness of CV fuel consumption management.

Nevertheless, motionless CVs with idling engines are regularly encountered. To avoid this, a transport corporation should ensure the sustainable behavior of middle managers and drivers that improves its economic and environmental performance indicators.

The integrated mechanism proposed in this paper ensures the sustainable behavior of transport corporation's employees by incentivizing them to reduce fuel consumption. The results were used to develop conceptual environmental protection requirements for locomotives of JSC Russian Railways. It is recommended to apply the results also to the vehicles of automobile and water transport corporations.

Further R&D works in this area can be related to:

– designing sustainable development mechanisms for transport corporations with alternative learning procedures;

– generalizing the results for solving other sustainable development problems;

– implementing the theoretical results in practice.

REFERENCES

1. *Our Common Future: Report of the World Commission on Environment and Development*, New York: United Nations, 1987.
2. *Transforming Our World: the 2030 Agenda for Sustainable Development*, UN General Assembly Resolution A/RES/70/1. New York: United Nations, 2015. URL: https://www.un.org/en/development/desa/population/migration/generalassembly/docs/globalcompact/A_RES_70_1_E.pdf (Accessed April 17, 2025.)
3. Caille, A., Al-Moneef, M., Castro, F.B., et al., *Transport Technologies and Policy Scenarios to 2050*, London: World Energy Council, 2007. URL: https://web.archive.org/web/20081204051628/http://www.worldenergy.org/publications/809.asp (Accessed April 17, 2025.)
4. *About Transportation & Climate Change: Transportation's Role in Climate Change*, Washington: US Department of Transportation, 2015. URL: https://web.archive.org/web/20151031215015/http://climate.dot.gov/about/transportations-role/overview.html (Accessed April 17, 2025.)
5. Kim, M.-K., Park, D., and Kim, D.Y., Quantification of the Ecological Value of Railroad Development Areas Using Logistic Regression Analysis, *International Journal of Environmental Research and Public Health*, 2021, vol. 18, no. 22, art. no. 11764.
6. Iliev, S., Stanchev, H., and Mitev, E., An Experimental Comparison of Emissions of a Common-Rail Diesel Engine Fueled with Ethanol and Butanol Additives, *AIP Conference Proceedings*, 2023, vol. 2868, art. no. 020018.
7. Metz, B., Davidson, O.R., Bosch, R., et al., Transport and Its Infrastructure, in *Climate Change 2007: Mitigation of Climate Change*, New York: Cambridge University Press, 2007, pp. 323–386. URL: https://www.ipcc.ch/report/ar4/wg3/transport-and-its-infrastructure/ (Accessed April 17, 2025.)
8. Savosina, M.I., Assessment of Effectiveness of Sustainable Transport Development, *World of Transport and Transportation*, 2020, vol. 18, no. 2, pp. 50–66.
9. Pollman, E., The Making and Meaning of ESG, *Harvard Bus. L. Review*, 2024, vol. 14, p. 403.
10. Burkov, V.N., Enaleev, A.K., Strogonov, V.I., et al., Models and Methods for Managing the Development and Implementation of Innovative Energy-Efficient Tools and Technologies in JSC Russian Railways, *Report on RFBR Project no. 17-20-05216*, Moscow: Trapeznikov Institute of Control Sciences RAS, 2018. (In Russian.)
11. Union Pacific: Sustainability. URL: https://www.up.com/aboutup/esg/index.htm (Accessed April 17, 2025.)
12. Corral-Verdugo, V., Frías, M., and García, C., Introduction to the Psychological Dimensions of Sustainability, in *Psychological Approaches to Sustainability*, Corral-Verdugo, V., García, C., and Frías, M., Eds., New York: Nova Science, 2010, pp. 3–18.
13. Tapia-Fonllem, C., Corral-Verdugo, V., and Fraijo-Sing, B., Sustainable Behavior and Quality of Life, in *Handbook of Environmental Psychology and Quality of Life Research*, Fleury-Bahi, G., Pol, E., and Navarro, O., Eds., Cham: Springer, 2017, pp. 265–268.
14. Novikov, D.A., *Theory of Control in Organizations*, New York: Nova Science, 2013.
15. Burkov, V., Kondratiev, V., Korgin, N., and Novikov, D., *Mechanism Design and Management. Mathematical Methods for Smart Organizations*, New York: Nova Science, 2013.
16. Enaleev, A., Comprehensive Assessment and Incentive Mechanism to Implement Environmental Requirements for Locomotives, *Proceedings of the 6th International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency*, Lipetsk, 2024, pp. 804–809.
17. Tsyganov, V., Strategic Adaptive Transport Management, *Proceedings of 2024 17th International Conference on Management of Large-Scale System Development (MLSD)*, Moscow, 2024, pp. 1–5. DOI: 10.1109/MLSD61779.2024.10739611

18. Tsyganov, V.V. and Fedyanin, D.N., Mentoring Mechanisms in Organizational Systems and Increasing Energy Efficiency of Railway Transport, *Herald of the Ural State University of Railway Transport*, 2019, no. 3 (43), pp. 55–66. (In Russian.)
19. Tsyganov, V.V., Basyrov, S.K., and Kovalev, S.M., An Energy Efficiency Learning Mechanism in Digital Railway Transport, *Trudy 8-oi nauchno-tekhnicheskoi konferentsii "Intellektual'nye sistemy upravleniya na zheleznodorozhnom transporte. Komp'yuternoye i matematicheskoye modelirovanie"* (Proceedings of the 8th Scientific and Technical Conference "Intelligent Control Systems in Railway Transport. Computer and Mathematical Modeling"), Moscow, 2019, pp. 185–190. (In Russian.)
20. Genkin, D., New Shunting Locomotives, *Oktyabr'skaya Main Line*. November 28, 2019. URL: https://company.rzd.ru/ru/9401/page/78314?accessible=true&id=182659 (Accessed April 17, 2025.) (In Russian.)

This paper was recommended for publication by RAS Academician D.A. Novikov, a member of the Editorial Board.

Received April 22, 2025, and revised June 20, 2025. Accepted June 25, 2025.

Author information

Tsyganov, Vladimir Viktorovich. Dr. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ bbc@ipu.ru
ORCID ID: 0000-0001-9274-0548

Cite this paper

Tsyganov, V.V., Sustainable Development Mechanisms for a Transport Corporation with Supervised Learning. *Control Sciences* **4,** 32–42 (2025).

Original Russian Text © Tsyganov, V.V., 2025, published in *Problemy Upravleniya*, 2025, no. 4, pp. 40–51.

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com

THE CORRELATION BETWEEN THE SOCIAL NETWORK ACTIVITY AND ACADEMIC PERFORMANCE OF TECHNICAL UNIVERSITY STUDENTS: A CASE STUDY OF *VKontakte*

V. V. Sych

Bauman Moscow State Technical University, Moscow, Russia
Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

✉ sycheslavovich@gmail.com

Abstract. The academic performance of students at Bauman Moscow State Technical University and its correlation to their activity in the *VKontakte* social network are considered. Machine learning methods are used to identify distinct performance paths reflecting the dynamics of educational achievement. The subscription lists of students are analyzed to identify marker communities characterizing the predominance of specific performance categories. Graph-theoretic clustering is applied to reveal structural groups of student interests. For each path, the stochastic vectors of interest shares across the community clusters are constructed and then used to identify the clusters having a statistically significant relation to particular performance paths. The results indicate a correlation between the digital behavior and academic outcomes of students, which contributes to the development of performance prediction models considering student interests in a social network.

Keywords: academic performance, *VKontakte*, machine learning, graph theory, performance prediction.

INTRODUCTION

Within the digitalization of the educational process, the data generated by students on social networks are becoming an important source of information for analyzing their academic and social activity. The possibility of differentiating students by their academic performance level based on their activity on *VKontakte* (VK), a Russian online social media and social networking service, was earlier demonstrated in several research works [1]. The dependence of career achievements on participation in professional social networks was also investigated [2], and academic success was considered to spread through friendly and educational links [3]. An inverted U-shaped relationship between the intensity of social network use and academic performance was identified [4]. As also established, mental health can act as a mediator in this relationship [5].

The vast majority of students have VK pages with a significant amount of information reflecting their interests and social links. Although social networks are rarely used to search for serious information and are more often launched “for leisure,” the long-term structure of subscriptions reflects the stable preferences and personal orientations of users. Therefore, their digital footprint can be associated with educational indicators and used as a reliable predictor of academic performance. This research is intended to identify the correlation between the academic performance and behavior of students in the digital space using machine learning methods and graph theory.

The topicality of this research is due to the need to develop effective tools for academic achievement monitoring and early identification of students at risk. Traditional performance assessment methods based on limited samples and surveys often neglect the multifaceted aspects of student behavior outside educational institutions. Analyzing digital footprints on social networks provides a more complete picture of students with their interests, social links, and even the interests of their friends, supplementing the existing assessment methods.

The aim of this research is to analyze the correlation between the academic performance paths and digital interests of students. Its uniqueness lies in the use of an extensive database: among 76 000 students of the Bauman Moscow State Technical University in 2014–2023, real VK pages were successfully matched for 48 000 students. The data include both academic performance indicators (end-of-semester examination grades) and information about subscriptions and friendships; therefore, a comprehensive digital portrait of each student can be created. Figure 1 shows an example of a friendship graph based on VK data.

The methodological foundation of this research includes clustering, standardization, and principal component analysis (PCA) algorithms to identify typical academic performance paths, as well as graph theory methods to analyze community structures. The research is based on VK data and university electronic records, anonymized after matching.

1. DATA DESCRIPTION

The analysis involves two main sources of information. The first source is an academic database obtained from the Electronic University system, covering end-of-semester examination grades for 2014–2023. For each student, grades for each semester are recorded, and the average grade for each of the four academic years can be calculated accordingly. The second source is VK data, collected using the VK API. For each user, the lists of community subscriptions and friendship information are extracted to analyze the interests and network environment of students. The profiles subscribed to university-related communities and all their friends were downloaded. A similar approach to data mining was used in [6].

The matching of academic data with VK profiles was carried out iteratively. At the first stage, the "core" of students with a high data fit degree was formed, where the comparison was carried out via the strict matching of full names and dates of birth, supplemented by the Levenshtein distance [7] calculated for non-matching name forms.

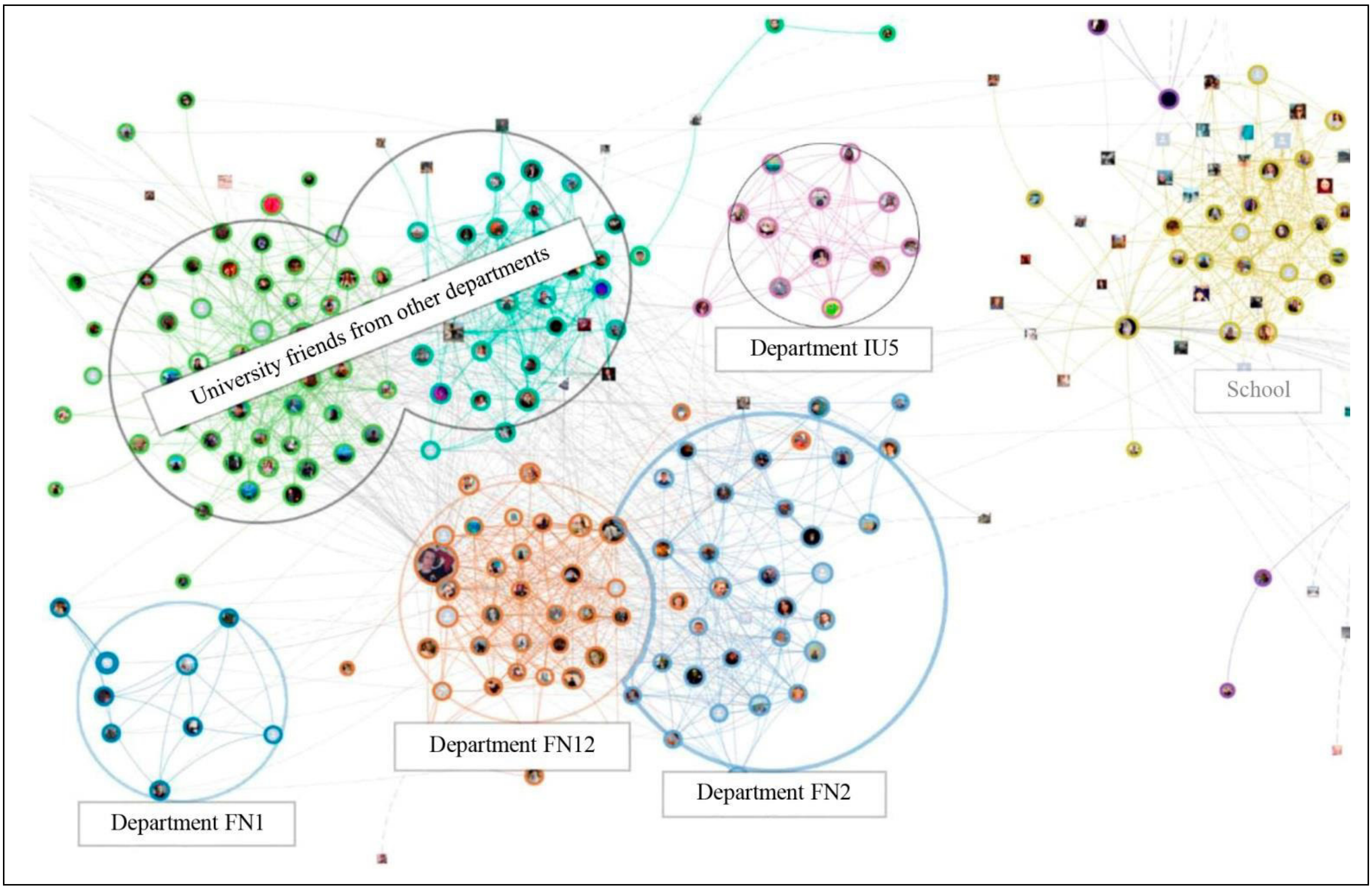


Fig. 1. The author's friendship graph based on VK links as an example (large clusters are labeled).

At the second stage, additional profiles were identified using friendship information: if a candidate matched only the basic parameters (e.g., name and year of birth) but was associated with the identified students from a particular study group, he/she was assigned to the same group with high probability. This process was based on the well-known principle: "tell me who your friends are, and I'll tell you who you are" (friendships as an additional marker of reliability). Following this approach, we managed to include students without enough initial matches for unambiguous identification.

The iterative repetition of the process (a core of 5000 students obtained at the first iteration, about 22 000 at the second, and 46 000 at the third) is based on the principle that the more friend profiles have already been confirmed for a student, the higher the probability of correctly matching his/her profile with academic data will be. For example, in cases where many users have the same name and the same birth date, an additional feature in the form of friendships with students with confirmed profiles significantly increases the reliability of identification. This approach considerably enlarges the sample and ensures its high representativeness. After the matching procedure, the data were anonymized and used for analysis only in this form. Similar methods for the extended matching of digital footprints with educational data demonstrated high effectiveness when investigating the academic performance of high school and college students [8, 9].

2. ACADEMIC PERFORMANCE ANALYSIS

To analyze academic performance paths, we selected students who had successfully completed four bachelor's or specialist program years without academic debt; the corresponding sample contains 15 495 males and 6285 females. A cluster approach to the segmentation of academic outcomes was earlier applied to assess the relationship between time spent on social networks and the grades of students specialized in biology [10] and analyze the use of social media on mobile devices in an international sample [11], confirming its suitability for identifying latent academic performance paths. For each student i, we calculated the average year grade vector

$$x_i = (x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4}) \in \mathbb{R}^4,$$

where $x_{i,c}$ denotes the average grade for academic year c (c = 1, 2, 3, 4). Next, the matrix $X \in R^{N\times 4}$ was standardized as follows:

$$z_{i,c} = \frac{x_{i,c} - \mu_c}{\sigma_c},$$

where μ_c and σ_c stand for the sample mean and standard deviation for academic year c, respectively.

We employed the k-means algorithm to partition students by academic performance paths. The optimal number k of clusters was selected based on the *silhouette score*, a measure of intra-cluster compactness (tightness) and inter-cluster separation proposed by P. Rousseeuw [12]. The maximum value of this measure (about 0.43) was achieved at $k = 2$, but this partition does not reflect the diversity of academic performance paths. Increasing k from 6 to 16 stabilized the silhouette score in the range of 0.21–0.27, indicating moderate but stable clustering quality.

To confirm the optimal number of clusters, we applied the *Gap Statistic* method proposed by R. Tibshirani et al. [13]. This method allows comparing the clustering quality of the original data with that of randomly generated data with the uniform distribution in the same range. The higher the value Gap(k) of this statistic is, the more distinct structure the data will have. (In other words, the clusters identified will be more significant than random ones.) For $k = 6$ and $k = 8$, the values of Gap(k) were close: 0.994 ± 0.006 and 0.911 ± 0.003, respectively. The slight decrease in the gap statistic value when passing from six to eight clusters is within the standard error, forming the so-called "plateau." This indicates that the detail of the clusters does not significantly deteriorate when increasing their number from six to eight.

Thus, despite the moderate values of the quality metrics, the choice of $k = 8$ is a reasonable trade-off between the statistical stability of the results and the sufficient detail of academic performance paths for the subsequent analysis of their correlation with the digital interests of students.

After the partition, each student was assigned a cluster number, in ascending order of the overall average academic performance (the simple arithmetic mean of all grades for eight semesters, see Table 1). Note the decreasing trend in the proportion of males when passing from clusters with low academic performance to those with high academic performance.

The academic performance paths are shown in Figs. 2 and 3.

The final partition is characterized by the following paths:

– **Path 0.** Stably low academic performance.

– **Path 1.** Low grades in the initial academic years, followed by an improvement.

– **Path 2.** Initially good outcomes, followed by a decline.

– **Path 3.** Low academic performance with noticeable growth.

– **Path 4.** Stably good grades with a slight dip.

– **Path 5.** Moderate grades transitioning to stably good outcomes.

Table 1

Statistics by path clusters

Cluster	The number of students	Average grade	The share of males, %
0	1910	3.48	90.0
1	2510	3.81	85.7
2	2120	3.92	82.8
3	2870	4.19	76.7
4	1950	4.21	71.6
5	2200	4.43	69.7
6	3500	4.57	52.9
7	4720	4.86	56.5

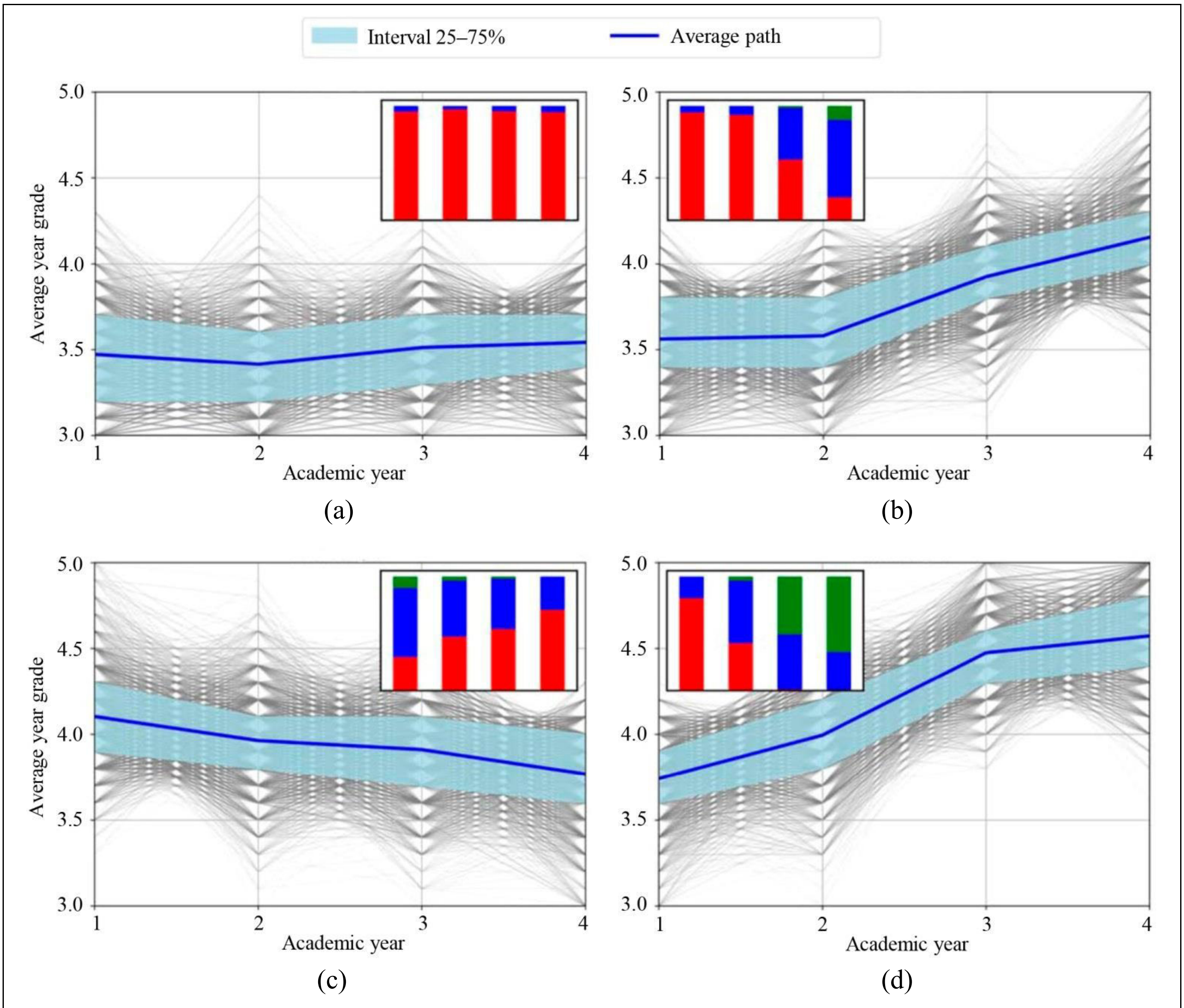


Fig. 2. Path clusters: (a) cluster 0, (b) cluster 1, (c) cluster 2, and (d) cluster 3.

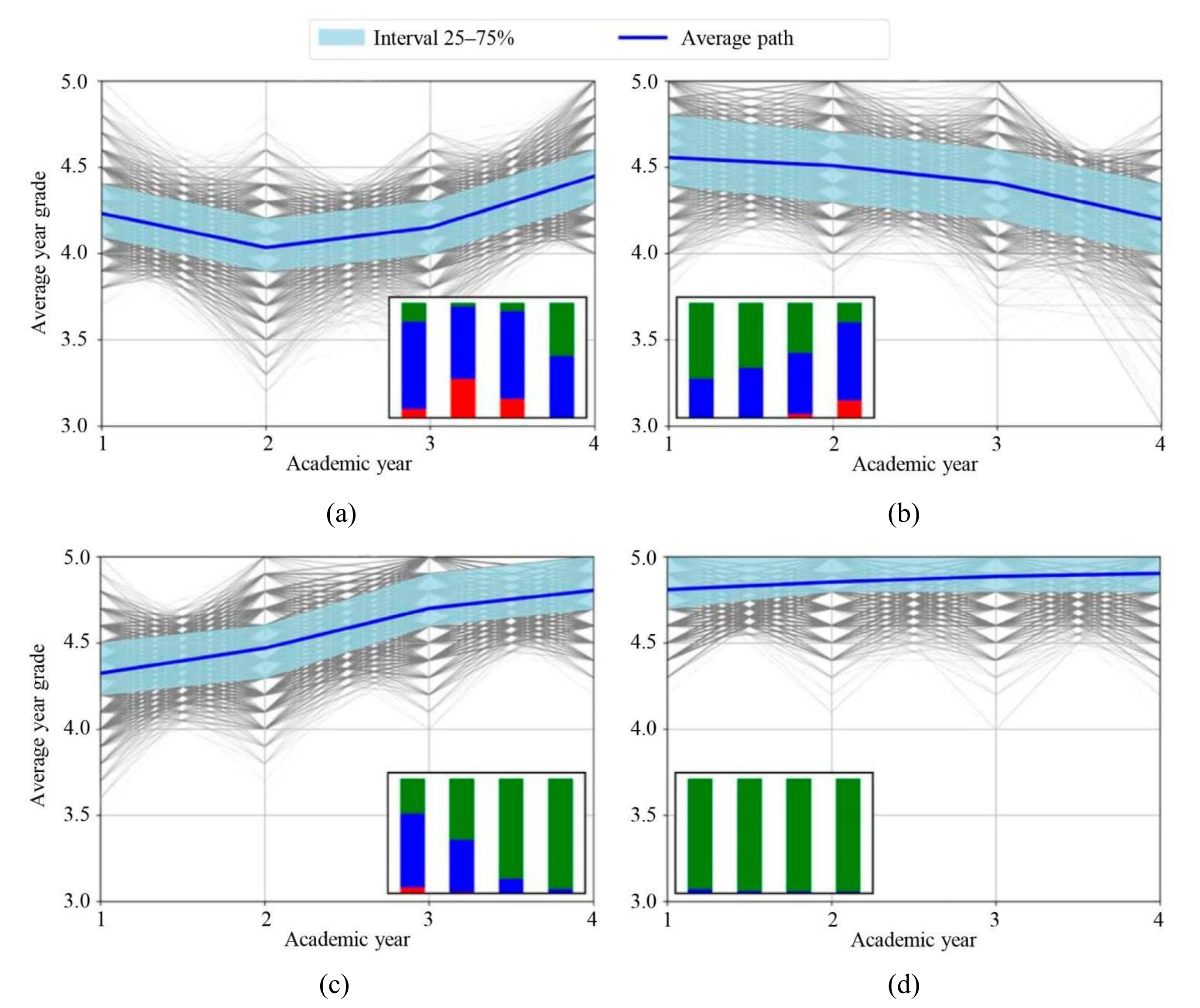


Fig. 3. Path clusters: (a) cluster 4, (b) cluster 5, (c) cluster 6, and (d) cluster 7.

For clarity, each cluster is represented by diagrams showing the proportions of students in different academic performance categories[1]: red, blue, and green bars indicate the shares of mediocre $(\bar{g} < 3.9)$, good $(3.9 \le \bar{g} < 4.5)$, and excellent $(\bar{g} \ge 4.5)$ students, respectively, where $\bar{g}$ is the student's average grade (the arithmetic mean of all his/her end-of-semester grades). These diagrams allow assessing the distribution of academic performance levels of different students across academic years in each path.

For additional visualization of the clustering results, we employed the *principal component analysis* (PCA) method. In Fig. 4, the points corresponding to students are displayed on the plane, with the horizontal axis reflecting the overall academic performance and the vertical one the dynamics of changes. The cross indicates the center of mass of the set.

The repeated runs of the *k*-means algorithm confirmed the stability of the resulting partition, and the visualization results demonstrate the separation of clusters, despite the partial overlap of their boundaries in the 2D projection. This structure provides a reliable foundation for further analysis of the correlation between the digital interests and academic performance of students since the average values of the paths well describe the main differences between the path clusters.

[1] In Russian higher education, the five-grade system is used: 1—"fail," 2—"unsatisfactory," 3—"satisfactory," 4—"good," and 5—"excellent." In the Western grade system, they are equivalent to F, D, C, B, and A, respectively.

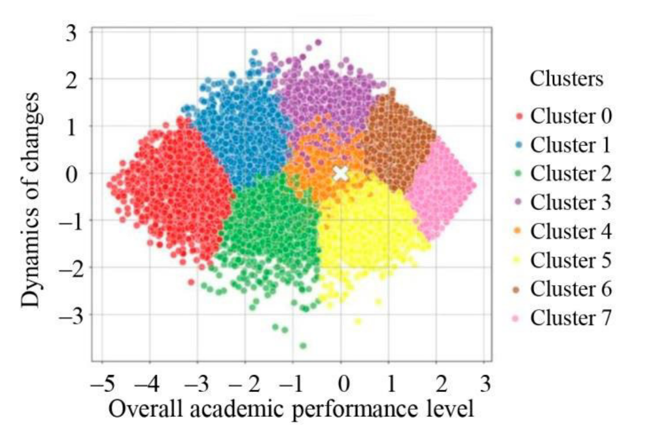


Fig. 4. Student clusters based on academic performance: PCA results.

3. THE GRAPH MODEL OF VK COMMUNITIES AND ITS ROLE IN ACADEMIC PERFORMANCE PATH ANALYSIS

3.1. Initial Data

As the initial data, we took all VK public pages (also called communities or groups) with at least 50 students subscribed; the resulting corpus contains approximately 4500 communities. The approach of using community subscriptions as academic success features has recently proven itself in the identification of the best students by NLP algorithms [14]. Before proceeding to theoretical graph analysis, we checked each community for correlation with academic performance paths using single-factor *analysis of variance (*ANOVA): for 760 communities, the differences between the paths were significant at $p < 0.05$; for 350, at $p < 0.01$.

Let $G = (V, E, W)$ be an undirected graph where V denotes the set of communities and an edge $(c_1, c_2) \in E$ exists if there is at least one common subscriber. The strength of a link between communities was estimated using the *weighted Jaccard coefficient*, first proposed for assessing the similarity of floral compositions [15]:

$$w(c_1, c_2) = \ln\left(1 + \frac{\left|S_{c_1} \cap S_{c_2}\right|}{\left|S_{c_1} \cup S_{c_2}\right|}\right),$$

where S_{c_i} is the set of students subscribed to community c_i. Taking the logarithm allows smoothing out extreme values of the coefficient when the communities overlap greatly or, conversely, are almost disjoint.

To detect communities, we applied the Louvain algorithm [16], which maximizes graph modularity. A single run on the entire graph with *resolution* = 1 produces several abnormally large clusters and many small clusters, so we employed a step-by-step approach. First, the entire graph was clustered at a low resolution value; then the overly large groups were recursively re-partitioned using the same method with *resolution* = 0.7. Additionally, 30% of the weakest links were eliminated, as their total number was very large. The iterative merging of the results provided a final partition into 21 clusters of comparable size.

3.2. The Visualization and Interpretation of Clusters

The final structure of the community graph is shown in Fig. 5: the complete graph with 21 clusters (on the left) and cluster 10 in detail (on the right),

Fig. 5. The visualization of the VK community graph.

Table 2

The characteristics of VK community clusters based on clustering results

Cluster	The description of communities	3/4/5
0	Large communities with universal entertainment content, covering a wide audience	**41**/32/27
1	Communities focused on humorous content, including memes related to e-sports and video games (CS2, PUBG)	**42**/33/25
2	Highly popular communities with memes targeted at a mass audience	**42**/31/27
3	Communities devoted to sports broadcasts and e-sports (soccer, UFC, Dota2)	**46**/32/22
4	Entertainment communities with simple and widespread memes	**42**/33/25
5	Communities with original, unique, or niche memes	**46**/34/20
6	Entertainment communities with repetitive trite memes	**46**/33/21
7	Communities spreading post-ironic memes targeted at a specific audience	**40**/35/25
8	Humorous communities focused on distributing video content	**46**/34/20
9	Communities representing post-irony and absurd humor	**42**/34/23
10	Communities related to educational and scientific topics: programming, internships, research	27/34/**39**
11	Student communities, including faculty public pages and groups like “Eavesdrop”	32/**35**/33
12	Communities focused on student life, including events and activities	27/35/**37**
13	Communities with friendly memes, often focused on pets (especially cats)	32/34/34
14	Communities devoted to aesthetics, fitness, nutrition, and self-care, primarily targeted at a female audience	26/33/**41**
15	Communities about fashion, the beauty industry, online stores, and female memes	25/34/**41**
16	Communities related to self-development, literature, foreign language learning, and aesthetic concepts	30/33/**37**
17	Foreign communities containing female memes and aesthetic content	29/34/**38**
18	Communities devoted to informal fashion, clothing, and footwear trade	**46**/35/19
19	Music communities focused on rap artists and the music industry	**45**/35/20
20	Gaming communities (Dota2, e-sports), as well as groups related to the trade of used cars	**46**/34/20

which includes mainly educational communities. The node colors correspond to the predominant academic performance level of subscribers, namely, red to mediocre, blue to good, and green to excellent students. This coding allows immediately assessing the homogeneity of the clusters in terms of academic performance.

Note that the path clusters ($n = 8$) reflect the dynamics of student academic performance changes over time, while the categories (mediocre, good, and excellent) are calculated based on the individual average grade for the entire study period and are introduced separately for a clear interpretation of the academic performance level in each community. They serve as an additional “projection” of the data.

The description of all clusters and the distribution of students by their academic performance categories are given in Table 2 (based on the author’s subjective generalizations).

4. THE CORRELATION BETWEEN THE DIGITAL INTERESTS AND ACADEMIC PERFORMANCE PATHS OF STUDENTS

4.1. The Formalization of Interests and Calculation of the Log-Odds Ratio

For each student i, we constructed the stochastic interest vector

$$p_i = \left(p_{i,0},\ p_{i,1}, \ldots,\ p_{i,20}\right)$$

with equal weights for all clusters (i.e., $\sum_k p_{i,k} = 1$), where $p_{i,k}$ is the ratio of subscriptions of student i in cluster k to all his/her subscriptions in total. This vector reflects the degree of concentration of the student's interests on particular community clusters.

To identify distinctions in the digital interests of students from different academic performance paths, we used a sample of students with known subscription data: after filtering, 4500 males and 1900 females remained. The *log-odds ratio* [17] was selected for the assessment, measuring the deviation of the share of subscriptions to cluster *k* in academic performance path *t* from the global average:

$$\log\_\text{odds}_{k,t} = \ln\left(\frac{p_{k,t}}{1-p_{k,t}}\right) - \ln\left(\frac{p_{k,\mathrm{g}}}{1-p_{k,\mathrm{g}}}\right).$$

In this formula:

$p_{k,t}$ is the *trimmed mean* share of subscriptions to cluster *k* among students in path *t*. First, for each student *i* from path *t*, the ratio of his/her subscriptions in cluster k to all his/her subscriptions is calculated. Then, the trimmed mean is taken from the ratios obtained for all students in path *t*;

$p_{k,\mathrm{g}}$ is the global trimmed mean share of subscriptions to cluster *k* across the entire sample of students. First, for each student, the ratio of his/her subscriptions in cluster *k* to all his/her subscriptions is calculated. Then, the trimmed mean is taken from all these ratios.

To reduce distortion due to atypically low or high individual values, we calculated the average shares for each cluster using trimmed mean: the 10% lowest and highest share values were excluded from the analysis. This approach yielded more reliable and informative estimates robust to outliers.

Positive values of $\log\_\text{odds}$ indicate increased interest in the cluster among students of a particular academic performance path, while negative values indicate a lower share of subscriptions compared to the global average $p_{k,\mathrm{g}}$ (i.e., "avoidance" of this cluster). Here are some illustrative examples of this metric:

- $\log\_\text{odds} = 0.5$ means that the odds of encountering subscriptions to this cluster among students of a given academic performance path are approximately 65% higher than the overall level since $e^{0.5} \approx 1.65$.
- $\log\_\text{odds} = -1$ means that the chances of encountering subscriptions to this cluster among students of a given academic performance path are approximately $e^{-1} \approx 0.37$, i.e., $1/0.37 \approx 2.7$ times lower than the average.

In further analysis, we calculated log_odds separately for males and females. The reasons are as follows:

- **The gender composition of paths is irregular.** In path cluster 0, males account for about 90%; in path cluster 7, only about 57%. During aggregation, this irregularity distorts the global averages $p_{k,\mathrm{g}}$ and, accordingly, the logarithmic coefficients.
- **Subscriptions have gender specifics.** Some communities (clusters 14–17) are targeted at a female audience; their "avoidance" by males is not informative for academic performance conclusions but increases the variance in the overall sample.

Figure 6 shows the heat map of $\log\_\text{odds}$ values for the male sample, revealing marked differences in the interests of such students depending on their academic performance path. The percentage of subscriptions to a given cluster among students of a given academic performance path is indicated below the corresponding log_odds value.

Here are some observations:

- Paths 0–2 (low and declining academic performance). These paths demonstrate a strong bias towards entertainment clusters 0–6 (log_odds ≈ 0.18–0.51) and a moderate bias towards street fashion and rap communities. Also, there is an avoidance of educational cluster 10 and university communities 11–12 (log_odds up to −0.41).
- Path 7 (stably excellent students). The picture is mirrored: negative coefficients for entertainment content and a noticeable jump in interest in clusters 10–12 (education and university communities).

Figure 7 shows a similar heat map for the female sample.

Similar conclusions can be drawn for female students from low academic performance paths: high interest in entertainment clusters and avoidance of educational ones. However, for the best paths (6 and 7), a more uniform distribution of interests is observed. Despite the general obviousness of the conclusions (entertainment prevails among weak students whereas education among strong ones), there are individual preferences within each path. These distinctions in digital interests can be used to build a predictive academic performance model.

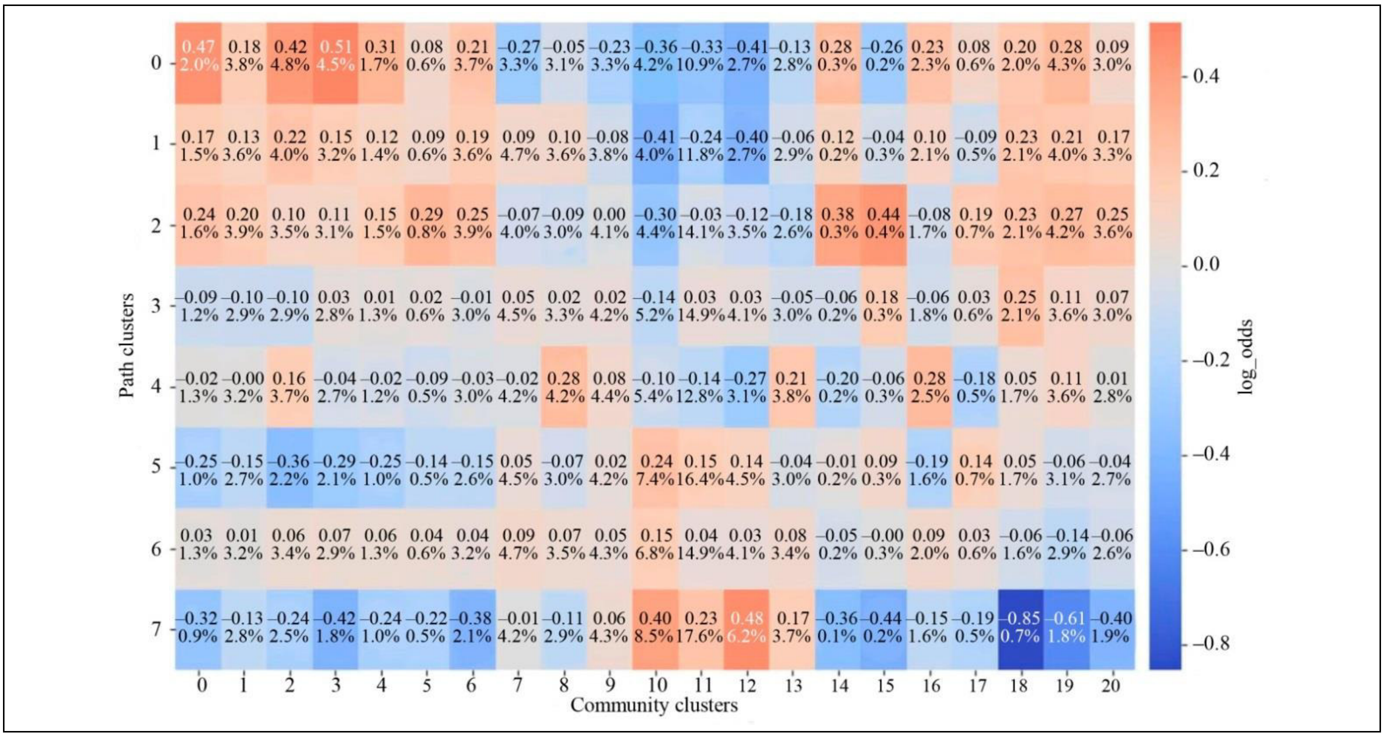


Fig. 6. The heat map of log_odds values for the male sample.

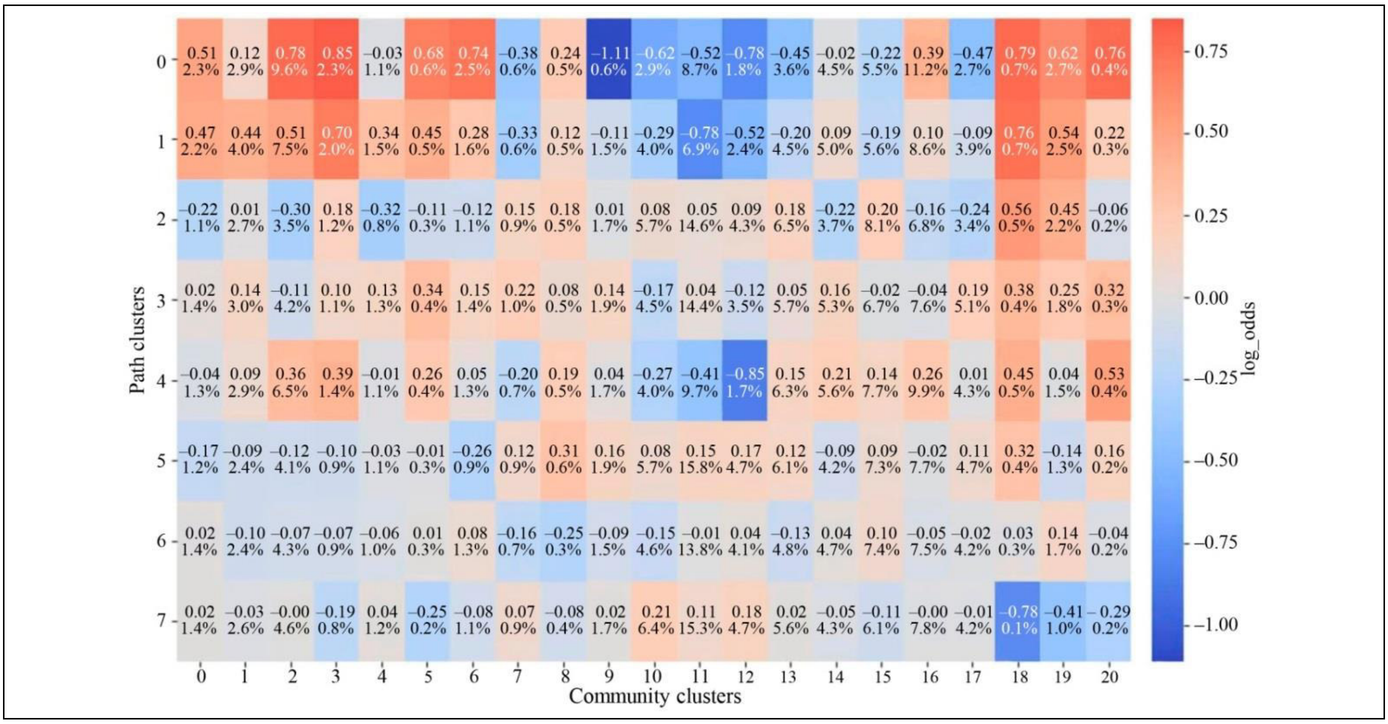


Fig. 7. The heat map of log_odds values for the female sample.

4.2. Testing the Hypothesis on the Independence of Academic Performance Paths from Digital Interests

To formally test the hypothesis on the statistical correlation between the academic performance and digital interests of students, we carried out a contingency table analysis using Pearson's χ^2 test, separately for the male and female samples (Table 3). The global χ^2 test confirmed the existence of a statistically significant correlation between belonging to a particular academic performance path and the subscription profile, among both male and female students.

The distinctions are illustrated in detail in Fig. 8. More specifically, Fig. 8a and 8b present the partial χ^2 test statistic for each of the 21 interest clusters; the χ^2-significant distinctions (α = 0.05) are set off in red whereas the insignificant ones in gray.

Clearly, correlations with academic performance paths became significant in 10 clusters for males and 15 for females. Figures 8c and 8d show estimates of the effect η^2, where the dotted lines indicate the thresholds of 0.01 (small effect), 0.06 (medium), and 0.14 (large) [18]. Most of the significant clusters demonstrate η^2 > 0.3, and especially among females, about half of the clusters exceed 0.5 (high practical significance of the distinctions revealed). The strongest correlation with academic performance is observed in clusters of educational and student communities (numbers 10–12), as well as entertainment and fashion/music communities (numbers 18–19). This logically fits the subscription patterns observed previously. The data obtained clearly reject the null hypothesis on the independence of academic performance paths from digital interests, confirming that students' digital preferences are representative and substantively related to their academic performance indicators.

Table 3

The global χ^2 test of the independence of academic performance paths from digital interests

Gender	Sample, students	χ^2 test statistic	The number of degrees of freedom	p-value	The critical χ^2 test value (α = 0.05)
Males	~ 4500	557.6	140	1e-51	168.613
Females	~ 1900	755.8		1e-85	

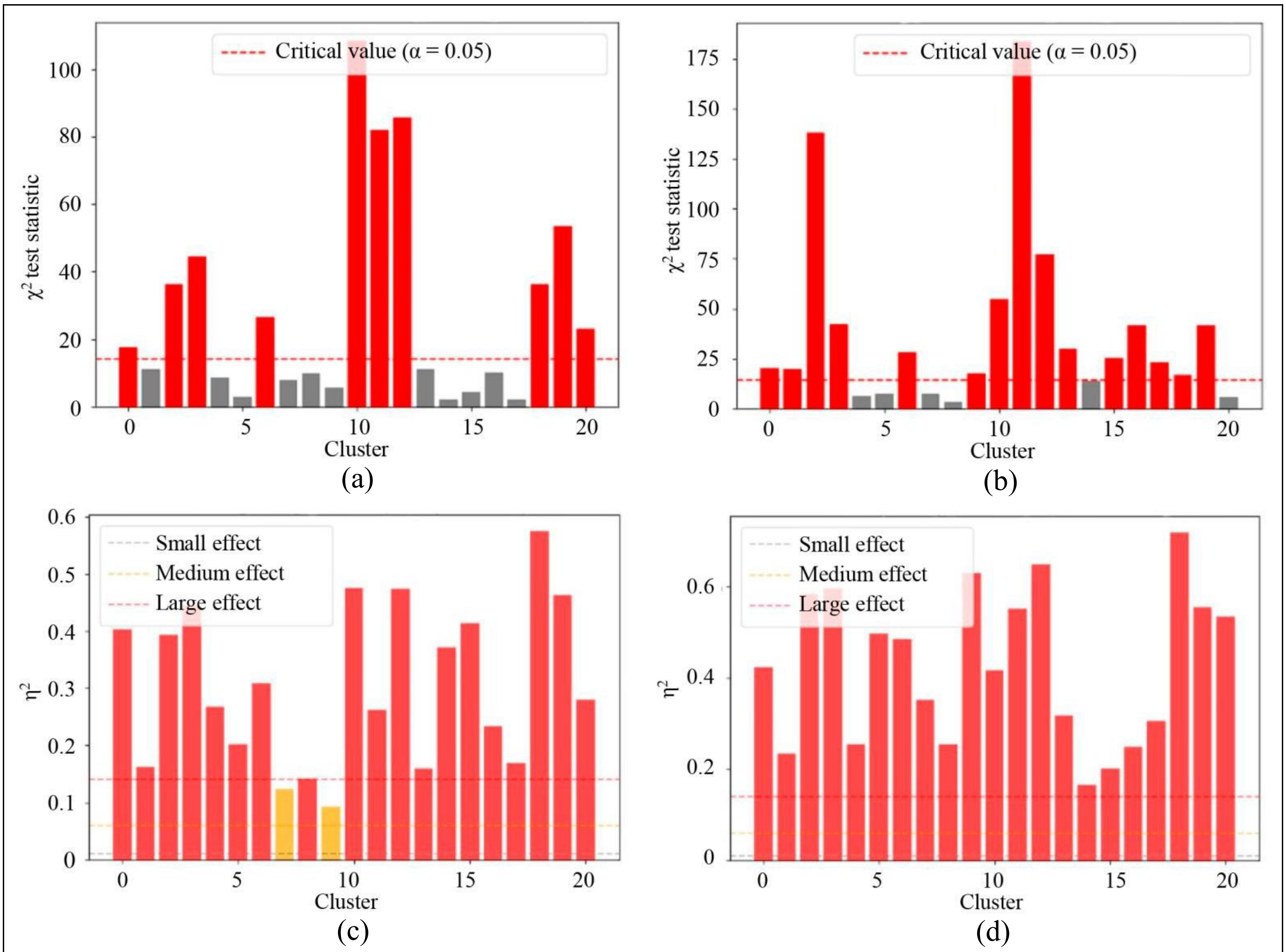


Fig. 8. Partial indicators of the relationship between the clusters of digital interests and academic performance paths: (a), (b) χ^2 statistic (red color – $p < 0.05$; dotted line – critical χ^2 test value) and (c), (d) estimates of the effect η^2 (dotted lines – thresholds). Bar charts (a) and (c) correspond to males whereas bar charts (b) and (d) to females.

CONCLUSIONS

In this research, we have established a statistical correlation between the topics of subscriptions of students at Bauman Moscow State Technical University on the *VKontakte* social network and their academic performance. According to the results, the digital footprint formed through interests and online activity can be a significant predictor of educational paths and, possibly, other offline indicators. A formal test of the hypothesis using Pearson's χ^2 test has shown high statistical significance ($p < 0.001$) of the identified correlations for both gender samples, and an analysis of the effect has confirmed the practical significance of the distinctions revealed.

Studies by other researchers also indicate a relationship between social media activity and academic performance, but most of them are limited to small samples and assessments of time spent on social media.

In the future, we plan to expand the analysis by adding the network structure of student friendships in order to build an integrated model for predicting academic performance based on both individual digital interests and their structural position in the student network (centrality measures, community membership, and broker roles). This approach will enable the more accurate diagnosis of academic risks and the formation of personalized recommendations for academic support, from course selection and consulting to inclusion in relevant educational communities.

REFERENCES

1. Smirnov, I.B., Differentiation of Students by Academic Performance in a Social Network, *Cand. Sci. (Ped.) Dissertation*, Moscow: National Research University Higher School of Economics, 2018, p. 210. (In Russian.)
2. Nikitkov, A. and Sainty, B., The Role of Social Media in Influencing Career Success, *International Journal of Accounting & Information Management*, 2014, vol. 22, no. 4, pp. 273–294. DOI 10.1108/IJAIM-02-2014-0009
3. Dokuka, S., Valeeva, D., and Yudkevich, M., How Academic Achievement Spreads: The Role of Distinct Social Networks in Academic Performance Diffusion, *PLOS ONE*, 2020, vol. 15, no. 7, art. no. e0236737. DOI: 10.1371/journal.pone.0236737
4. Tafesse, W., Social Networking Sites Use and College Students' Academic Performance: Testing for an Inverted U-shaped Relationship Using Automated Mobile App Usage Data, *International Journal of Educational Technology in Higher Education*, 2022, vol. 19, art. no. 16. DOI: 10.1186/s41239-022-00322-0
5. Al Mosharrafa, R., Akther, T., and Siddique, F.K., Impact of Social Media Usage on Academic Performance of University Students: Mediating Role of Mental Health under a Cross-sectional Study in Bangladesh, *Health Science Reports*, 2024, vol. 7, no. 1, art. no. e1788. DOI 10.1002/hsr2.1788
6. Chkhartishvili, A., Gubanov, D.A., Melnichuk, V.S., and Sych, V.V., Exploratory Data Analysis and Natural Language Processing Model for Analysis and Identification of the Dynamics of COVID-19 Vaccine Opinions on Small Datasets, *Advances in Systems Science and Applications*, 2023, vol. 23, no. 3, pp. 108–126. DOI: 10.25728/assa.2023.23.3.1381
7. Levenshtein, V.I., Binary Codes Capable of Correcting Deletions, Insertions and Reversals, *Soviet Physics Doklady*, 1966, vol. 10, no. 8, pp. 707–710.
8. Kashpur, V.V., Gubanov, A.V., Feshchenko, A.V., et al., Correlation Between Academic Achievements of High School Students and Their Digital Shadow in Social Network, *Pedagogy and Education*, 2020, no. 4, pp. 37–51. (In Russian.)
9. Smirnov, I.B., Estimating Educational Outcomes from Students' Short Texts on Social Media, *EPJ Data Science*, 2020, vol. 9, art. no. 27. DOI: 10.1140/epjds/s13688-020-00238-1
10. Dudina, V.A., Anisimovskaya, A.I., Gorchenko, A.L., et al., Assessing the Influence of Social Networks and Other Factors on the Academic Performance of Students at the Faculty of Biology, *Molodoi Uchenyi*, 2019, no. 48(286), pp. 438–440. (In Russian.)
11. Giunchiglia, F., Zeni, M., Gobbi, E., et al., Mobile Social Media Usage and Academic Performance, *Computers in Human Behavior*, 2018, vol. 82, pp. 177–185. DOI: 10.1016/j.chb.2017.12.041
12. Rousseeuw, P.J., Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis, *Journal of Computational and Applied Mathematics*, 1987, vol. 20, pp. 53–65. DOI: 10.1016/0377-0427(87)90125-7
13. Tibshirani, R., Walther, G., and Hastie, T., Estimating the Number of Clusters via the Gap Statistic, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2001, vol. 63, no. 2, pp. 411–423. DOI: 10.1111/1467-9868.00293
14. Gorshkov, S., Ignatov, D.I., Chernysheva, A.Yu., et al., Identifying Top-Performing Students via VKontakte Social Media Communities Using Advanced NLP Techniques, *IEEE Access*, 2025, vol. 13, pp. 962–979. DOI: 10.1109/ACCESS.2024.3521857
15. Jaccard, P., Étude Comparative de la distribution florale dans une portion des Alpes et du Jura, *Bulletin de la Société Vaudoise des Sciences Naturelles*, 1901, vol. 37, pp. 547–579. DOI: 10.5169/seals-266450
16. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E., Fast Unfolding of Communities in Large Networks, *Journal of Statistical Mechanics: Theory and Experiment*, 2008, no. 10, art. no. P10008. DOI: 10.1088/1742-5468/2008/10/P10008
17. Barnard, G.A., Statistical Inference, *Journal of the Royal Statistical Society: Series B (Methodological)*, 1949, vol. 11, no. 2, pp. 115–139. DOI: 10.1111/j.2517-6161.1949.tb00028.x
18. Cohen, J., *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed., Hillsdale, NJ: Lawrence Erlbaum Associates, 1988.

This paper was recommended for publication by RAS Academician D.A. Novikov, a member of the Editorial Board.

Received May 4, 2025, and revised July 27, 2025. Accepted July 28, 2025.

Author information

Sych, Vladislav Vital'evich. Student, Bauman Moscow State Technical University, Moscow, Russia; engineer, Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ sycheslavovich@gmail.com
ORCID iD: https://orcid.org/0009-0001-6787-8948

Cite this paper

Sych, V.V., The Correlation Between the Social Network Activity and Academic Performance of Technical University Students: A Case Study of *VKontakte*. *Control Sciences* **4**, 43–54 (2025).

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com

APPLICATION OF SUBDEFINITE MODELS IN THE GLOBAL LOCALIZATION OF A MOBILE ROBOT

A. D. Moscowsky

National Research Center "Kurchatov Institute," Moscow, Russia

✉ moscowskyad@yandex.ru

Abstract. This paper considers the application of subdefinite (SD) models, a variation of constraint programming, to the localization problem of a mobile robot. A complex technology with semantic maps and point cloud maps is proposed. The technology is intended to accelerate and increase the accuracy of global localization in large, symmetric, and periodic environments. The conventional localization approach is based on data from rangefinders generating point clouds; the idea proposed instead is, first, to match the objects observed by the robot to those on the semantic map (recognize the scene), and then apply SD computations to perform localization via visual landmarks. SD computations are used to determine interval constraints on the robot's positions, represented by several sets for each hypothesis obtained during the scene recognition. Within the interval constraints, the robot is localized using rangefinder data based on a particle filter initialized within these constraints. According to the experiments conducted on the open KITTI-360 dataset, localization based on SD computations can reduce the search space to 0.2% of the original map size. The complex technology shows a significant advantage compared to approaches involving point clouds or visual landmarks only, especially in scenarios with multiple hypotheses about the matches of observed objects and those on the semantic map.

Keywords: global localization, subdefinite models, constraint programming, scene recognition, semantic maps, mobile robot.

INTRODUCTION

Localization is crucial in modern mobile robotics since the vast majority of robot control approaches are based on the current position knowledge. When it is difficult to use satellite navigation (e.g., in urban or indoor environments), one applies localization methods building a 3D map in the form of a point cloud. The popularity of this approach is determined, among other things, by its versatility: there is no need to place additional labels (devices) in the robot's environment to aid localization. Such a detailed map can also be used in navigation, as it contains information about obstacles. It can be built by means of common sensors such as scanning laser rangefinders and depth cameras. The modern development of SLAM (*Simultaneous Localization and Mapping*) methods [1] allows building quite accurate maps of large areas, but the following global localization problem arises when using such maps further: it is necessary to determine the robot's starting position without knowledge of any initial conditions. This problem is solved during robot initialization (the start of operation) and also when the robot gets lost (its position differs from the expected one) due to errors during localization or, e.g., was moved by third parties. All these situations prevent the use of available data on the robot's previous position and displacement for localization. Finding the robot's position in the entire space (usually limited by the map size) becomes more difficult as the map size increases, as noted in the review [2] of the corresponding approaches. Besides the high volume of necessary computations, the additional problem is that the above urban and indoor environments often contain domains with quite similar geometry, especially in typical buildings. Such "periodicity" and "symmetry" of the environment lead to the so-called local minima on the map, in which the robot can erroneously globally localize itself. And the number of such local minima grows with the map size.

Researchers tackle this problem from different standpoints. One R&D direction involves only point clouds, and the task is to construct qualitative descriptors associated with point clouds in order to identify and compare geometric and other features of their elements, including semantic ones. Generally speaking, this approach includes three main stages: descriptor extraction, descriptor matching, and position refinement. Among descriptor extraction methods, we mention FPFH (*Fast Point Feature Histograms*) [3], NDT (*Normal Distribution Transform*) [4], NeRF (*Neural Radiance Fields*) [5], Minkloc3d [6], and others [7]. Different approaches are used for descriptor matching: RANSAC (*RANdom SAmple Consensus*) [8], graph-based approaches [9], optimization algorithms [10], and learning approaches [11] are widespread. ICP (*Iterative Closest Point*) [12] and its extensions (for example, see [13]) are used to further refine the robot's position. Also note probabilistic approaches, such as the histogram filter or particle filter [14], applied to point clouds [15]. However, these descriptors are primarily based on the geometric features of the mutual arrangement of the points, and the problem of local minima due to the periodicity and symmetry of the spaces cannot be eliminated accordingly. Despite that the descriptors narrows the search space, they remain at the point cloud level and, to some extent, are associated with the above problems, so other approaches pass to a "higher" map representation level, which will be discussed later.

Similar to descriptor construction, Place Recognition [16] encodes the entire environment of a robot as a vector of numbers. Usually, neural network models are applied for this purpose; they are trained so that the same environment, taken from different angles, at different times of day, etc., produces vectors close to each other. Such models often have multimodal input and consider point clouds, "raw" images, and the semantics of the scene. Thus, the map is provided with a set of "key frames" with encoding vectors, and localization is reduced to finding the frame best matching the robot's current observation from a set of sensors. This approach is good in the sense of utilizing all available information about the environment, but it requires dense coverage of the space with key frames. A strong change in camera angle can also deteriorate the frame search for different compositions of the robot's sensors. Such methods demonstrate the best performance for sensor compositions with full view, but this is not possible or reasonable for all robots.

This paper follows another R&D direction with the so-called semantic maps. On such maps, objects are assigned semantic labels (classes). Methods for obtaining semantic maps in an automatic set are being actively developed [17, 18]. A set of same-class objects is extracted in the environment directly observed by a robot. Then it is necessary to recognize the scene, i.e., match the objects of the map to the objects in the environment (scene). Matching at the object level allows narrowing the search space compared to searching in 3D maps. Semantics gives additional "uniqueness" to scenes with monotonous geometry. Once a match is found, localization methods based on visual landmarks can be used. However, one problem of this direction is that such scene recognition methods generate a number of hypotheses about object matching. Therefore, it is required to perform localization for each hypothesis and evaluate the resulting quality of object matching. The direction with semantic maps has several advantages: the methods are less sensitive to changes in camera angle and allow for the manual editing of semantic maps. The latter is an obvious benefit of the method because the environment may change over time and performing a complete mapping procedure often seems unreasonable.

In view of the aforesaid, the aim of this research is to accelerate and increase the accuracy of global localization of a mobile robot on point cloud maps. We choose the R&D direction involving point clouds jointly with semantic maps, which requires scene recognition. This paper focuses on the global localization problem of a mobile robot based on scene recognition results considering such features as high object positioning errors, object matching errors, and selection of the best result among a set of hypotheses.

1. LOCALIZATION METHODS FOR SCENE RECOGNITION

In modern works on localization using semantic maps addressing scene recognition, a prevalent method is an optimization-based approach for matching two sets of 3D points, known as SVD (*Singular Value Decomposition*) [19]. Localization by this method based on scene recognition results was performed in [20–22]. The approach minimizes the root-mean-square (RMS) error of point positions between two sets, but requires exact element-by-element correspondence of one set to another and neglects the position errors in computations. In [23], the ICP approach (see above) was applied not to point clouds but to two sets of objects: on the map and on the scene. In several publications (for example, [24, 25]), robot localization was performed by the 2D-3D *reprojection* technology: the positions of flat objects in the image were matched to their 3D positions. This approach eliminates the need for the 3D localization of the scene objects in space and, therefore, partially offsets the errors of determining this distance, which can be significant.

A seemingly promising direction is to combine localization by point clouds with data obtained by scene recognition methods. An interesting approach was proposed in [26]: the separate 3D registration of point clouds of selected objects, followed by a refinement using ICP.

At the same time, to the best of the author's knowledge, search probabilistic approaches based on a histogram filter or a particle filter have not been applied in the literature to perform localization via visual landmarks in the setting under consideration. An explanation is that these methods require restricting the search space, causing difficulties for the majority of the localization approaches discussed here: they output a point in space rather than a domain. Imposing interval constraints on the robot's position would settle this problem. The common approach of *constraint programming* [27] yields such data. Its generalization in the form of subdefinite (SD) computations [28] was applied to perform localization via visual landmarks and use a histogram filter within the resulting constraints [29]. Below, this approach will be applied to robot localization based on a set of hypotheses using a particle filter (PF), which determines the robot's position from point clouds within the resulting constraints.

2. THE COMPLEX GLOBAL LOCALIZATION TECHNOLOGY

The complex global localization technology proposed in this paper is within the direction using semantic maps. This technology requires both a map layer (in the form of a point cloud) and a semantic layer (with marked positions of objects and some semantic information about them). In addition to the two map layers, the input data are images and point clouds from the robot's sensors. The complex technology allows solving the localization problem in the following stages:

1. recognizing and localizing all objects in the robot's environment (forming the scene);
2. recognizing the scene (matching the objects recognized to those on the semantic map and obtaining several hypotheses);
3. applying the SD localization approach to these hypotheses (discarding contradictory hypotheses and determining interval constraints on the robot's position);
4. determining the robot's position by means of a PF in the resulting constraints for each hypothesis and calculating quality values;
5. selecting the best solution with the highest quality value.

The first stage is implemented either by the classical method for recognizing objects in images and further localizing them using depth maps or point clouds [30], or by 3D recognition approaches [31].

The second stage, including scene recognition, was described in detail in [32]. Graph theory algorithms were used therein to extract geometric features in the mutual arrangement of groups of objects. CLIP (*Contrastive Language-Image Pre-training*) [33], a foundational visual-language model, was applied to consider the visual similarity of objects in addition to the semantic label, as previously done by most researchers in the field of robot localization with semantic maps. According to the conclusions, the visual similarity criterion of objects used jointly with the geometric feature criterion significantly improves the accuracy of scene recognition compared to methods involving only one of the criteria. However, several problems of working with objects, including high localization errors, the presence of visually similar objects, errors of recognition systems, and the multiplicity of solutions, lead to many hypotheses at the output of such systems, and the correct solution does not always have the highest quality value.

In the third stage, the SD localization approach—the focus of this paper—is applied. The approach serves to perform robot localization via visual landmarks: it imposes interval constraints on the robot's positions and allows identifying possible input data inconsistencies due to the above features of scene recognition.

In the fourth stage, the robot's position is determined using a PF and a map layer (a point cloud). The rationale behind this approach is that a PF can be naturally initialized in interval constraints derived from SD localization. In addition, a PF can handle a wide set of input data, and a quality value of any particle can be obtained for these data; in turn, the value represents a hypothesis about the robot's position in the localization problem.

The fifth stage is to select the best solution for all hypotheses of the second stage (the one with the highest quality value).

The flowchart of the complex global localization technology is shown in Fig. 1.

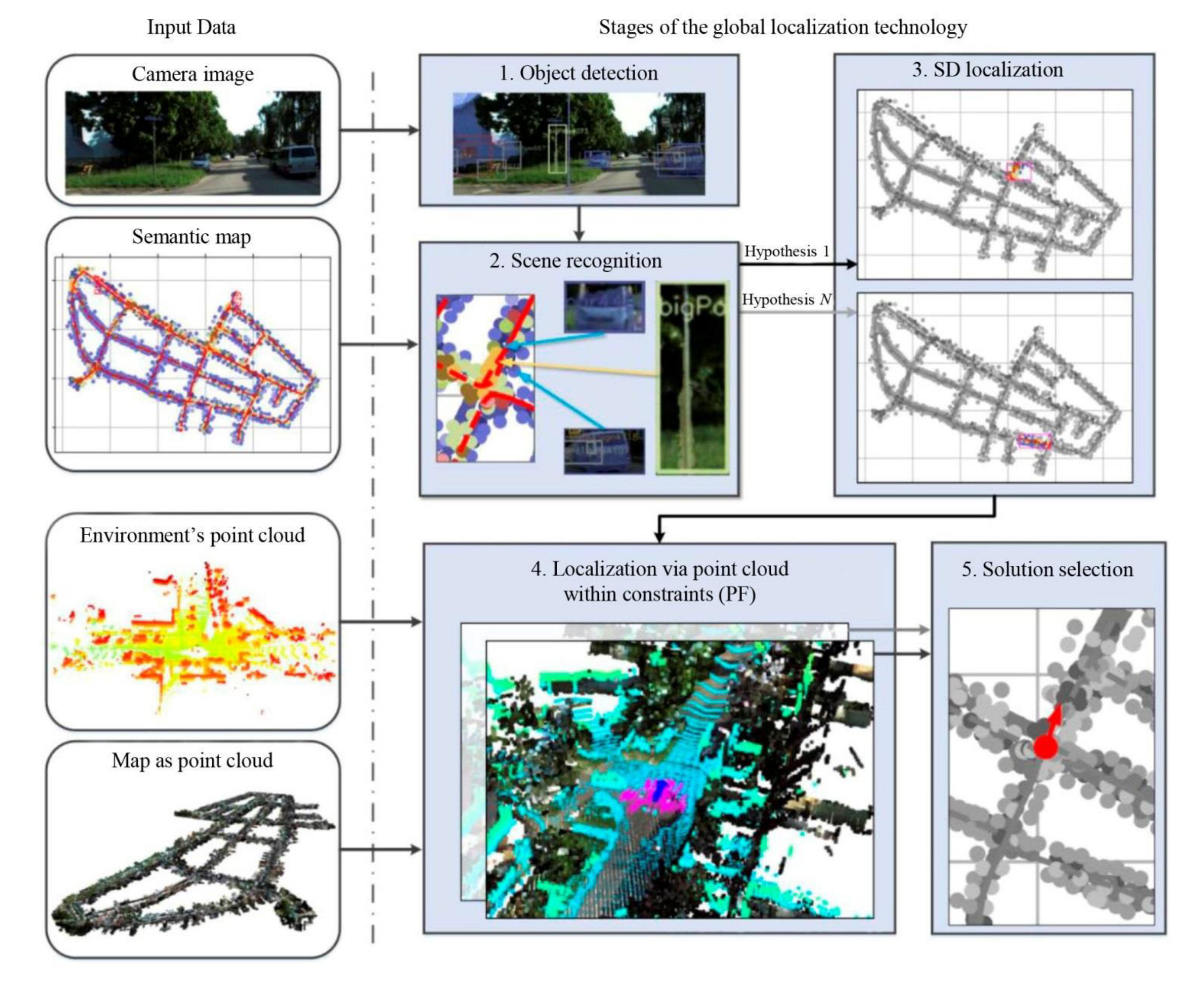


Fig. 1. The flowchart of the complex global localization technology.

2.1. Localization via Landmarks Based on SD Models

To apply SD computations, one has to describe the problem under consideration in the form of an SD model. Such a model represents the desired variables, their definitional domain, as well as interpretation and assignment functions to update them [28]. The desired variables—the robot's position—are specified in the form of multi-intervals, i.e., a non-intersecting set of intervals in ascending order:

$$^{*}a = \left[\left[a^1_{\text{low}}, a^1_{\text{high}}\right], \ldots, \left[a^N_{\text{low}}, a^N_{\text{high}}\right],\right.$$

$$\left. a^i_{\text{low}\backslash\text{high}} \in R,\ a^i_{\text{low}} \le a^i_{\text{high}},\ a^i_{\text{high}} < a^{i+1}_{\text{low}}\right],$$

where $^{*}a$ is some SD variable defined by a multi-interval; a^i_{low} and a^i_{high}, $i = \overline{1, N-1}$, are lower and upper limits, respectively, of intervals forming a multi-interval. Multi-intervals are a special case of SD variables, denoted by the left-hand superscript "*" throughout this paper.

Originally, these multi-intervals are initialized with all available values. Then, the so-called interpretation functions are defined to reduce the uncertainty (i.e., narrow the limits of the multi-intervals): the new value calculated at each iteration is the intersection of the old one with that obtained by applying the interpretation function. The absence of intersection means that the input data contain inconsistencies and should be excluded from consideration. Interpretation functions are defined based on the following relationship between the robot's position ($^{*}X$, $^{*}Y$, $^{*}\theta$), the position of the observed landmarks (X_l, Y_l), and their measurements, which include the distance r_l and angle α_l to the object:

$$^*X = \pm\sqrt{(r+\Delta r_l)^2 - (^*Y - Y_l)^2} + X_l, \quad (1)$$

$$^*Y = \pm\sqrt{(r+\Delta r_l)^2 - (^*X - X_l)^2} + Y_l, \quad (2)$$

$$^*X = \frac{^*Y - Y_l}{\tan(^*\theta + {^*\Delta a_l} + a_l)} + X_l, \quad (3)$$

$$^*Y = (^*X - X_l)\tan(^*\theta + {^*\Delta a_l} + a_l) + Y_l, \quad (4)$$

$$^*\theta = \arctan\left(\frac{Y_l - {^*Y}}{X_l - {^*X}}\right) - (a_l + {^*\Delta a_l}), \quad (5)$$

where $^*\Delta r_l$ and $^*\Delta\alpha_l$ are the measurement errors of r_l and α_l, respectively, expressed in intervals (they can be obtained from the RMS errors by the N sigma rule: $^*\Delta r_l = [-N\sigma r_l, N\sigma r_l]$. In formulas (1)–(5), some variables are represented as multi-intervals and are calculated using interval arithmetic [34]. For each observed landmark obtained during scene recognition, a different set of the interpretation functions (1)–(5) is formed. The procedure of SD computations is reduced to the iterative application of functions to a set of SD variables. Once a function has been applied, it is removed from the list of active functions; however, if an input variable for some assignment function outside the active list has been updated (its value has changed), this function will be returned to the list of active ones. The computational procedure continues until the list of active functions is empty. In practice, one may also limit the maximum number of iterations or set a desired accuracy for the resulting values of the variables.

At the same time, several situations cannot be resolved by scene recognition methods. These include symmetry situations: algorithms on graphs are unable to distinguish the order of appearance of objects when increasing the angle to them, since the graph contains information only about the distance between two objects. This situation can be resolved at the level of SD models by introducing additional correctness checking functions. Based on the current constraints ($^*X$, $^*Y$, $^*\theta$), it is necessary to calculate the angles to the leftmost and rightmost objects in the scene and check the nonnegativity of their difference:

$$\neg\left(\arctan\left(\frac{^*Y - Y_{\text{left}}}{^*X - X_{\text{left}}}\right) - \arctan\left(\frac{^*Y - Y_{\text{right}}}{^*X - X_{\text{right}}}\right) < 0\right), \quad (6)$$

where the subscripts "left" and "right" indicate the leftmost and rightmost, respectively, landmarks by the angle. One should apply additional correctness checking functions not to the full set ($^*X$, $^*Y$, $^*\theta$) but to the Cartesian product where each element is a tuple of three intervals for each variable. The tuples not satisfying condition (6) are excluded from the total set.

The output is the values of the SD variables represented by the set of tuples of intervals. Such tuples describe the union of rectangular domains (with sides parallel to the x- and y-axes, sometimes referred to as bounding rectangles [34]) specifying an angle constraint. However, the answer in this form may be somewhat redundant and contain unnecessary values (Fig. 2a).

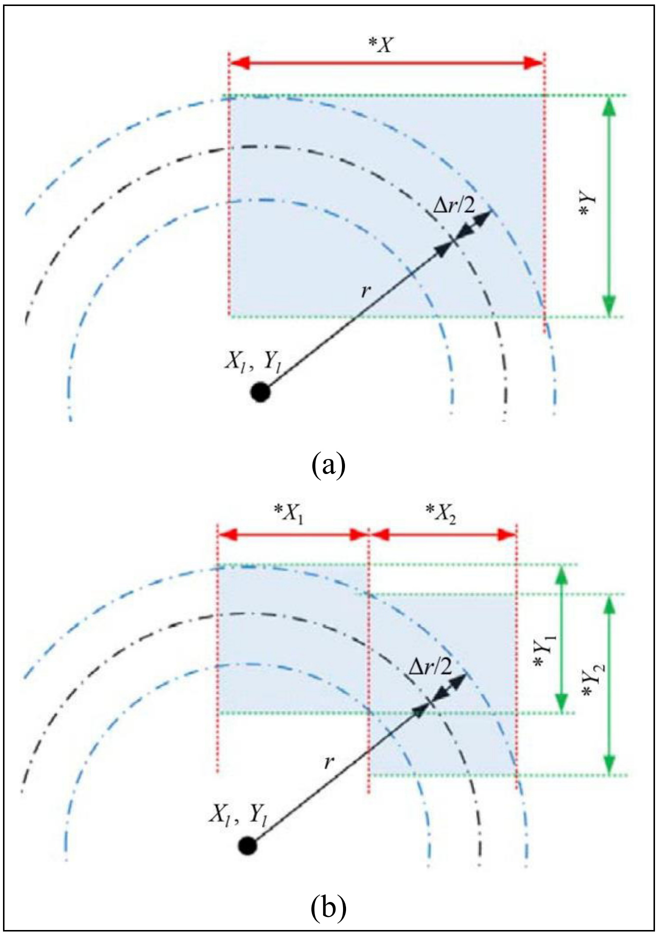


Fig. 2. Exhaustive estimation: (a) the redundant estimation of the robot's location bounded by blue dash-and-dot lines, where $^*X$ is the input SD variable and $^*Y$ is the calculated one; (b) the exhaustive estimation of its location by splitting the input variable.

For such situations, exhaustive estimation algorithms are used in interval analysis [34]: the intervals exceeding a given constraint are iteratively split, and the computational procedure is reapplied to the result. Figure 2b shows an example of such splitting after the first iteration of the algorithm on the variable $^*X$ and the computation of the new values of $^*Y_1$ and $^*Y_2$. Thus, the general global localization algorithm based on SD computations (SD localization algorithm) can be represented as the following pseudocode.

Global SD localization algorithm via visual landmarks

```
  Algorithm sdm_global_localization({<X1, Y1, r1, α1, σr, σα>}, Nσ, xy_lim):
     -- X1, Y1 – the positions of objects on the map
     -- r1, α1 – the measurements of objects in the scene
     -- σr, σα – the measurement errors of objects
     -- Nσ – the factor to convert the errors into intervals
     -- xy_lim – the limit size of intervals for x and y
     V = ∅
     *X = *Y = [[-∞, ∞]]
     *θ = [[-π, π]]
     R = init_R({<X1, Y1, r1, α1, σr, σα>}, Nσ) -- (1)-(5)
     Status, *X, *Y, *θ = sdm_process(*X, *Y, *θ, R)
     if !Status
        return V
     C = init_C({<X1, Y1, r1, α1, σr, σα>}, Nσ) -- (6)
     function process_leaf(*x, *y, *γ)
        Status,*x, *y, *γ = sdm_process(*x, *y, *γ, R)
        if Status
           if |*x| > xy_lim:
              *x1, *x2 = divide(*x)
              process_leaf(*x1, *y, *γ)
              process_leaf(*x2, *y, *γ)
           elif |*y| > xy_lim:
              *y1, *y2 = divide(*y)
              process_leaf(*x, *y1, *γ)
              process_leaf(*x, *y2, *γ)
           else
              add (*x, *y, *γ) to V
     for (*x, *y, *γ) in *X×*Y×*θ do
        if C(*x, *y, *γ) then
           process_leaf(*x, *y, *γ)
     return V
```

The **sdm_global_localization** algorithm takes as input a set of landmark positions and their measurements, the factor *N*σ to convert the RMS errors into intervals, and the limit *xy_lim* for the intervals. The **init_R** auxiliary function initializes the parameters of the assignment function (1)–(5) based on landmark data. The **sdm_process** function computes the SD variables as described above (also, see [28]) and returns the status (success or failure) and the updated values of the SD variables. The **init_C** function initializes the parameters of the additional correctness checking functions (6). The **divide** function splits an interval into two parts at its center, and the algorithm returns a set of interval tuples that do not exceed the limit *xy_lim* in the *x*- and *y*-axis and have passed all additional checks.

2.2 Localization Using a Particle Filter within the Constraints

The constraints yielded by the **sdm_global_localization** algorithm describe a domain where the robot can be located. With these constraints available, one can carry out a search using a PF. Each particle in this context is a hypothesis about the robot's position. It is required to initialize the initial distribution within the constraints uniformly. For the operation of the PF, it is necessary to evaluate the quality of each particle. The idea is to use dense map data in the form of a point cloud and a point cloud obtained from the robot. To compare the two point clouds, one should calculate the shortest distances between the points in the robot's cloud and those in the map cloud.

The arithmetic mean of these distances will show the overlapping quality of the clouds. However, it is recommended to use a more "sensitive" estimate to accelerate the convergence of the PF:

$$p(d, \sigma) = \frac{e^{-(d/\sigma)^2}}{\sqrt{2\pi}\sigma}, \quad (7)$$

where p is the desired particle estimate; d is the mean of the shortest distances; finally, σ is the desired sensitivity threshold. At each iteration of the PF operation, the weights of all particles are calculated by transferring the point cloud from the robot to the particle coordinates and calculating the estimate (7). Next, a *resampling* procedure is applied to discard low-weight particles and multiply high-weight ones; the probability with which a particle will fall into the new sample is proportional to its weight. Because some particles are represented in multiple instances in the resampling process, it makes sense to add some random noise to each particle at the beginning of each iteration in order to separate the same hypotheses and cover the search space more densely. The procedure consisting of the above steps can be performed either for a fixed number of iterations or until reaching target values of some criteria, e.g., the mean weight of the particles, the values of the covariance matrix constructed from the particle distribution, etc. There are different practices to select the desired position: the weighted average of all particles, the particle with the maximum weight, and initial clustering of particles with the subsequent application of the above methods to local clusters. Once the desired robot's position is computed, its quality can be evaluated using the same metric (7). The global localization algorithm using a PF is as follows.

Global localization algorithm via point clouds using a particle filter within constraints

```
Algorithm pf_global_localization(V, σ, N, iter, Δ, PCD_map, PCD_robot):
   -- V – constraints on the robot's position
   -- σ – sensitivity
   -- N – the number of particles
   -- iter – the number of iterations
   -- Δ – noise parameters
   -- PCD_map – map point cloud
   -- PCD_robot – robot's point cloud
   P = ∅
   for n in N do
      v = sample_element(V)
      p = sample_particle(v)
      add p to P
   for i in iter do
      W = ∅
      P' = ∅
      for p in P do
         p = random_shift(p, Δ)
         add p to P'
         PCD = transform_cloud(PCD_robot, p)
         d = get_mean_p2p(PCD, PCD_map)
         w = get_w(d, σ) -- (7)
         add w to W
      P = resampling(P', W)
   pose_final = get_pose(P, W)
   PCD_final = transform_cloud(PCD_robot, pose_final)
   d_final = get_mean_p2p(PCD_final, PCD_map)
   w_final = get_w(d_final, σ) -- (7)
   return pose_final, w_final
```

The **pf_global_localization** algorithm takes as input a set of constraints yielded by the **sdm_global_localization** algorithm, a series of parameter values, and point clouds from the robot and map. The **sample_element** function randomly selects one of the interval tuples in proportion to the total domain size (the product of the sizes of all intervals). The **sample_particle** function equiprobably generates a particle including the position on the *x*- and *y*-axis and the angle so that all values will belong to the selected intervals. The **random_shift** function adds random white noise with the specified parameters to the particles. The **transform_cloud** function performs a geometric transfer of the point cloud to the particle's coordinates. The **get_mean_p2p** function computes the shortest distances between the points of the first and second clouds. The **get_w** function computes the particle weight by formula (7). The **resampling** function resamples the particles, and the **get_pose** function computes the robot's position in one of the above ways (e.g., by taking the weighted average, by multiplying the particle matrix by a weight vector). The algorithm returns the robot's position and the weight of this position as a quality criterion.

Thus, the **sdm_global_localization** and **pf_global_localization** algorithms are sequentially applied to each hypothesis selected for matching scene objects to map objects; in the resulting set of solutions with calculated quality criteria, the best one is chosen in terms of the introduced metric.

3. AN EXPERIMENTAL STUDY

The complex technology was experimentally studied on KITTI-360 [35], an adapted open dataset, for the scene recognition task proposed in [32]. The prepared data[1] contain descriptions of semantic maps and scenes in the form of object sets with an indication of matches between these sets. As input data for the **sdm_global_localization** algorithm, we chose the results of the scene recognition method based on the search of isomorphic subgraph *sig_lite_clip* [32], which showed the best results in terms of the metric of finding a completely correct answer. The answer for the scene recognition task is a match between observed objects and objects on the map; this match is used to form a localization task based on landmark data. Nevertheless, a significant part of the answers did not contain the correct answer (when all objects are matched correctly) in the first place in the list of hypotheses ordered by the certainty factor (a quality value of scene recognition results). To test the effectiveness of the technology in dealing with such input data, the ten answers with the maximum certainty factors were selected for each input scene.

The selected scenes were divided into several groups: *super_true*, *has_true*, and *wrong*. The *super_true* group corresponds to those scene recognition results in which the correct answer of matching scene objects to map objects is present and has the highest certainty factor among the selected ten answers. The *has_true* group contains the answers in which the correct answer is present, but the certainty factor is not the highest among the other hypotheses. The *wrong* group contains results without completely correct answers. The size of the groups was correlated as 68%, 27%, and 4% to the total number of scenes, and the remaining number was not processed by recognition methods. All experiments were carried out on a computer with an AMD Ryzen7 2700X Eight-Core 3.70GHz processor and 32GB RAM, without involving GPUs.

For the ten hypotheses obtained, the SD localization procedure (SDM, *Subdefinite Models*) with $N\sigma$ = 4, in the variants with (xy_lim = 10 m) and without (xy_lim = ∞ m) additional interval splitting was performed for each scene (see Fig. 2). The resulting interval constraints on the robot's position for all hypotheses of the same scene were evaluated for interval accuracy and interval coverage (Table 1). The interval accuracy was calculated by determining whether the true robot's position, known from the dataset, belongs to the calculated interval constraints (the larger the value is, the better the result will be). The interval coverage reflects the reduction of the search space. It was evaluated by distributing a million particles (robot's positions) uniformly over the entire map area, determining the number of particles belonging to any of the resulting constraint sets, and taking the ratio of the resulting number to the entire set (the smaller its value is, the better the result will be).

According to Table 1, the *wrong* group (no completely correct answers) is far behind the other groups in interval accuracy. Also, there is an order-of-magnitude difference in interval coverage for the variant of the algorithm with splitting, albeit with a small loss of interval accuracy, compared to the variant without splitting. However, an order-of-magnitude decrease in interval coverage causes an order-of-magnitude increase in the running time of the algorithm.

[1] URL: https://github.com/MoscowskyAnton/scene_recognition_kitti_360

Table 1

Testing results for the SD localization approach

Method	Interval accuracy, %, ↑			Interval coverage, %, ↓			$\|t\|$, s
	Group			Group			All groups
	super_true	*has_true*	*wrong*	*super_true*	*has_true*	*wrong*	
SDM with splitting	97.1	98	17.6	**0.23**	**0.41**	0.65	0.69
SDM without splitting	**99.8**	**100**	**37**	3.22	4.44	**0.37**	**0.03**

The proposed technique was compared with other global localization methods to evaluate the accuracy of determining the robot's position. All approaches under study were implemented by the author or taken from open source libraries. The methods working with the scene recognition result received all ten variants for each scene as input and chose the best one by the quality value; for the best solution, the position and angle errors and the running time were calculated accordingly. The errors were described by the mean (|.|) and the median (*M*). The running time of the scene recognition method was not considered; for the method and sequence selected, it was 41.55 ± 46.5 s [32] on the same computational hardware. Note that in the KITTI-360 set, the point clouds for the map are presented not in a monolithic version but as several overlapping domains (sub-maps); where possible, one of the sub-maps best fitting the search space was selected. The following approaches were considered in the study:

– **RANSAC+ICP**, a classical localization approach based on point clouds only, with the calculation of FPFH descriptors [3], searching for their matches using the RANSAC method [8], and further refinement using ICP [12]. This approach was chosen for consideration because of the available open implementations. RANSAC matching was performed for each sub-map, and ICP refinement was performed for the position with the highest *fitness* value.

– **SVD**, a localization method based on scene recognition data [19]. For different scene recognition results, the quality of the solution was determined by the average position error of each object between the scene and the map.

– **SVD+ICP**, localization from scene recognition data using the SVD method and further position refinement using ICP via point clouds. The *fitness* parameter of the ICP method was taken as a quality value.

– **SDM+PF**, the SD localization technology via landmarks (proposed in this paper) with the **sdm_global_localization** algorithm ($N\sigma = 4$, xy_lim = 10 m) and a PF used within the constraints by the **pf_global_localization** algorithm ($N = 150$; $\sigma = 0.1$; $iter = 5$; $\Delta = (0.5; 0.1)$). The particle with the highest weight was selected as the solution; the weight also served as a criterion for selecting solutions for different scene recognition results.

– **SDM+RANSAC+ICP**, the **RANSAC+ICP** approach applied to a point cloud derived from the constraints calculated by the **sdm_global_localization** algorithm. The computed constraints were extended by the range of the rangefinder. The *fitness* value of the ICP method was also taken as a measure of solution evaluation. The variant of the SDM method without splitting (xy_lim = ∞) was used, and the other parameters corresponded to those in **SDM+PF**.

Additionally, the *recall@k* (*R@k*) metric, often encountered in global localization tasks, was calculated for the results. It shows the percentage of the answers falling within some certainty region for *k* answers. In the literature on global localization [36, 37], the certainty region is usually chosen to be 20 m: from a practical viewpoint, it is no longer important how much the error exceeds this value.

Table 2 presents the numerical results of the methods obtained for the *super_true* group.

According to the results in Table 2, the widespread SVD approach without reference to point clouds significantly outperforms the other methods in almost all parameters. The matter is that the certainty factor used for normalizing answers in scene recognition methods ideologically coincides with the quality metric of the SVD method; therefore, for the *super_true* group, this method makes almost no mistake, selecting the hypothesis with the highest certainty factor (actually, the correct one). This fact can also be observed when comparing SVD with its extension SVD+ICP, where the mean error increases significantly due to the appearance of scenes with a different hypothesis preferred, but the median error is reduced compared to SVD as ICP improves the robot's position on the point cloud data in over 50% of the cases. Other methods using scene recognition results show a comparable median but a strongly higher mean due to incorrectly chosen hypotheses. (In some cases, they are at other ends of the map, thereby significantly affecting the mean.) The impact of errors "distant" from the correct answer is also indirectly confirmed by the superior *R@k* values for SDM+PF, since the *R@k* metric describes a threshold estimate.

The results for the *has_true* group are of greater interest within the study because they reflect the ability of different methods to find the correct answer in a set of similar hypotheses. Table 3 summarizes the numerical results for this group.

In Table 3, SVD (the best method of Table 2) has significant accuracy losses and shows poor performance. The reason is that it neglects additional point cloud data, in contrast to the other methods. This fact confirms the importance of the complex approach proposed above. However, additional correction using ICP significantly improves the accuracy and provides almost better results in all parameters and good time performance of the algorithm. Generally speaking, the values of the accuracy measures of the methods based on the results of scene recognition and position refinement via point clouds insignificantly differ from those for the *super_true* group (see Table 2). Hence, it is possible to find a solution among many similar hypotheses. The strong decrease in the $R@1$ value is due to the incorrect best answer in the *has_true* group; the decrease in the $R@5$ value is because the correct answer may fall outside the top five solutions.

The group of *wrong* solutions (without a completely correct answer) is also of interest in localization via visual landmarks: the problem statement contains knowingly false data. The numerical results are given in Table 4.

According to the experimental results in Table 4, the lack of a correct answer from the scene recognition system has a negative impact on the results (all accuracy measures decrease significantly compared to Tables 2 and 3). Therefore, it is important to develop robust scene recognition methods. The SD localization approach based on correctness checking functions allows discarding some hypotheses: on average, 17.5% of all processed hypotheses are discarded, and this value varies insignificantly within the groups under study (16.6, 20.1, and 18.5%, respectively).

Figure 3 provides the values of *recall*@1 and *recall*@5 without division into groups.

The proposed methods based on search space reduction (SDM+PF and SDM+RANSAC+ICP) show comparable results, in terms of accuracy, with the SVD-based approach, even excelling them in some cases (see Fig. 3). Hence, this R&D direction is

Table 2

Experimental results for the *super_true* group of solutions

Method	$\|r\|$, m	$M(r)$, m	$\|\alpha\|$, rad	$M(\alpha)$, rad	$\|t\|$, s	$M(t)$, s	$R@1$, %	$R@5$, %
RANSAC+ICP	233.3	190.1	0.71	0.12	4418.1	4171.5	37	-
SVD	**4.7**	2.3	**0.26**	**0.06**	**0.005**	**0.004**	94.4	95.6
SVD+ICP	39.5	1.6	0.29	**0.06**	18.6	17.7	94.9	96.1
SDM+PF	83.7	2.1	0.39	0.06	95.1	97.1	**98.4**	**99.3**
SDM+RANSAC+ICP	100.6	**1.4**	0.28	0.05	1009	976	69.4	83.7

Table 3

Experimental results for the *has_true* group of solutions

Method	$\|r\|$, m	$M(r)$, m	$\|\alpha\|$, rad	$M(\alpha)$, rad	$\|t\|$, s	$M(t)$, s	$R@1$, %	$R@5$, %
RANSAC+ICP	207.5	158.9	1	0.46	4038.3	4002	27.7	–
SVD	191.1	28.9	1.15	0.83	**0.005**	**0.004**	44.3	85.8
SVD+ICP	**38.4**	2	0.35	0.04	19.6	19.6	44.5	**85.9**
SDM+PF	80.4	**1.5**	0.29	0.05	94.8	95.6	**62.2**	63.9
SDM+RANSAC+ ICP	71.5	6.7	**0.22**	**0.03**	911.2	878.3	61.5	84.6

Table 4

Experimental results for the *wrong* group of solutions

Method	$\|r\|$, m	$M(r)$, m	$\|\alpha\|$, rad	$M(\alpha)$, rad	$\|t\|$, s	$M(t)$, s	$R@1$, %	$R@5$, %
RANSAC+ICP	289	262.3	0.87	**0.06**	4501	4255	**33.3**	–
SVD	351	298.3	1.22	0.92	**0.004**	**0.004**	24.3	**30.7**
SVD+ICP	206.7	**95.2**	1.13	0.86	17.8	17.2	24.5	**30.7**
SDM+PF	**203.4**	150.8	1.34	1	79.1	83.4	25.7	25.7
SDM+RANSAC+ICP	210.5	285.5	**0.59**	0.07	1076.6	1030.9	13.3	20

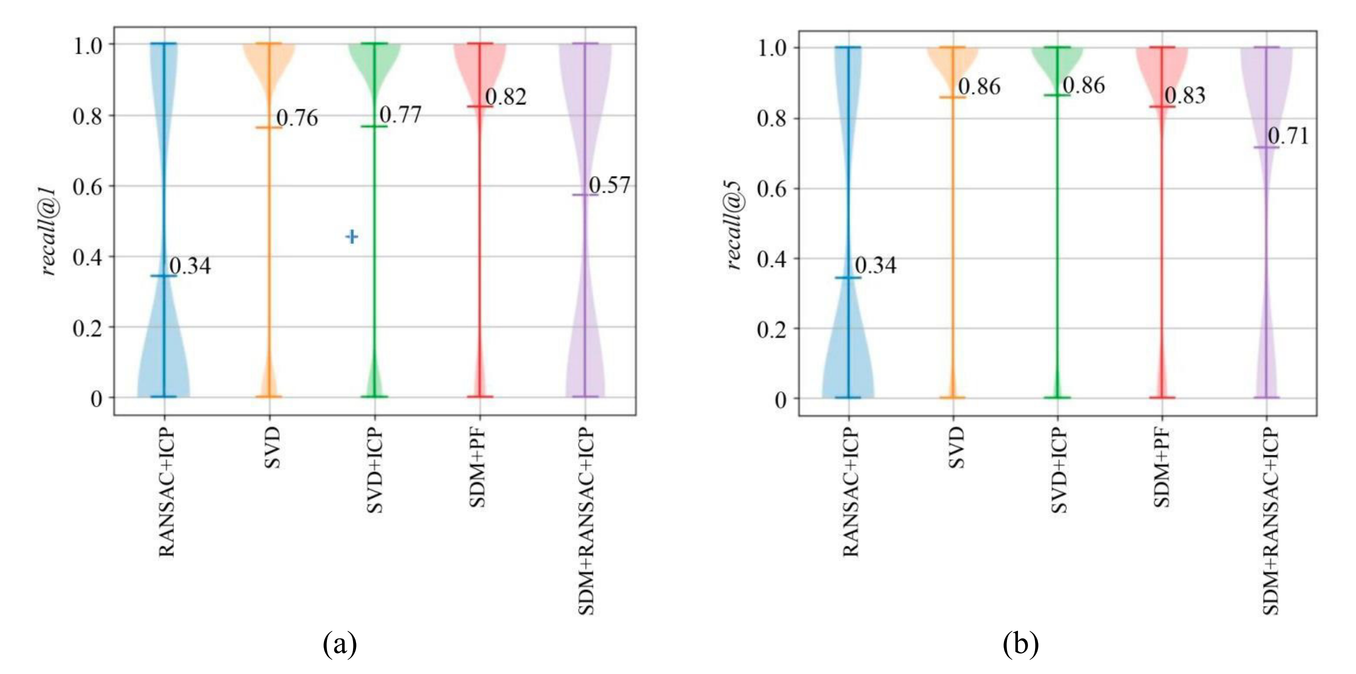


Fig. 3. The distributions and means of *recall@k* for different methods: (a) *recall*@1 and (b) *recall*@5.

promising for solving global localization problems. The values of the operating parameters of the algorithms used in this study are rather for orientation, since no software optimization issues arose during the implementation of the methods. The PF-based approach using the point-to-point distance for cloud comparison has a limitation: it performs well only near the correct answer and with small deviations in angle. Investigating and applying more sensitive metrics for comparing point clouds, including their color characteristics, would potentially improve the result as well. Note separately that SDM+RANSAC+ICP has demonstrated both a significant increase in accuracy and a reduction in running time compared to the classical RANSAC+ICP approach. This finding suggests that the search space reduction technology based on SD localization can be used for other global localization methods with point clouds.

CONCLUSIONS

This paper has described a subdefinite (SD) localization approach as part of a complex global localization technology for a robot on 3D maps equipped with a semantic layer. By recognizing and localizing the objects in the robot's field of view, one can solve the scene recognition problem by matching them with the map objects. The SD localization approach is applied to the resulting data to determine constraints on the robot's position, which in turn can be used to reduce the search area for a localization method based on rangefinder data.

According to the experiments, the point cloud localization methods applied within the constraints obtained by the complex technology significantly improve the accuracy and time performance of the algorithm compared to the basic global localization approach without semantics. The reduction of the search space to 0.2% of the map size confirms the applicability of this technology for any localization methods based on rangefinder data. The experiments have also demonstrated that combining localization via landmarks with localization via point clouds is effective when the correct scene recognition answer "hides" among several similar hypotheses. Thus, based on the experimental results, the aim of this research—improving the quality of global localization—has been achieved. The quality of the algorithm has been improved owing to SD computations, which are used within the integrated localization technology with semantic maps.

Further research will analyze and apply more sensitive point cloud registration methods to increase the accuracy of robot positioning and hypothesis estimation. In addition, it is reasonable to consider the proposed approach jointly with Place Recognition to reduce the search space among the available key frames: they are often provided with a coordinate label, and the latter can be used to determine compliance with the constraints yielded by SD localization. Note that a promising task is the multicriteria optimization of the parameters of the methods used, which requires significant computational resources. However, this task goes beyond the scope of the paper, as the parameters have been chosen empirically.

Acknowledgments. *This work was carried out under the state assignment of the National Research Center "Kurchatov Institute."*

REFERENCES

1. Chen, P., Zhao, X., Zeng, L., et al., A Review of Research on SLAM Technology Based on the Fusion of LiDAR and Vision, *Sensors*, 2025, vol. 25, no. 5, art. no. 1447.
2. Yin, H., Xu, X., Lu, S., et al., A Survey on Global LiDAR Localization: Challenges, Advances and Open Problems, *Int. J. Comput. Vis*., 2024, vol. 132, no. 8, pp. 3139–3171.
3. Rusu, R.B., Blodow, N., and Beetz, M., Fast Point Feature Histograms (FPFH) for 3D Registration, *Proceedings of 2009 IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009, pp. 3212–3217.
4. Saarinen, J., Andreasson, H., Stoyanov, T., et al., Normal Distributions Transform Monte-Carlo Localization (NDT-MCL), *Proceedings of 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 2013, pp. 382–389.
5. Mildenhall, B., Srinivasan, P.P., Tancik, M., et al., NeRF: Representing Scenes As Neural Radiance Fields for View Synthesis, *Commun. ACM*, 2022, vol. 65, no. 1, pp. 99–106.
6. Komorowski, J., MinkLoc3D: Point Cloud Based Large-Scale Place Recognition, *Proceedings of 2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA, 2021, pp. 1789–1798.
7. Rosen, D.M., Doherty, K.J., Terán Espinoza, A., et al., Advances in Inference and Representation for Simultaneous Localization and Mapping, *Annu. Rev. Control. Robot. Auton. Syst*., 2021, vol. 4, no. 1, pp. 215–242.
8. Fishler, M. and Bolles, R., Random Sample Consensus: A Paradigm for Model Fitting Applications to Image Analysis and Automated Cartography, *Communication of the ACM*, 1981, vol. 24, pp. 381–395.
9. Yang, H., Shi, J., and Carlone, L., TEASER: Fast and Certifiable Point Cloud Registration, *IEEE Trans. Robot*., 2021, vol. 37, no. 2, pp. 314–333.
10. Lusk, P.C. and How, J.P., CLIPPER: Robust Data Association without an Initial Guess, *IEEE Robot. Autom. Lett.*, 2024, vol. 9, no. 4, pp. 3092–3099.
11. Choy, C., Dong, W., and Koltun, V., Deep Global Registration, *Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020, pp. 2511–2520.
12. Rusinkiewicz, S. and Levoy, M., Efficient Variants of the ICP Algorithm, *Proceedings of the 3rd International Conference on 3-D Digital Imaging and Modeling*, Quebec City, Canada, 2001, pp. 145–152.
13. Vizzo, I., Guadagnino, T., Mersch, B., et al., KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way, *IEEE Robot. Autom. Lett.*, 2023, vol. 8, no. 2, pp. 1029–1036.
14. Barfoot, T.D., *State Estimation for Robotics*, Cambridge: Cambridge University Press, 2017.
15. Chen, R., Yin, H., Jiao, Y., et al., Deep Samplable Observation Model for Global Localization and Kidnapping, *IEEE Robot. Autom. Lett*., 2021, vol. 6, no. 2, pp. 2296–2303.
16. Zhang, X., Wang, L., and Su, Y., Visual Place Recognition: A Survey from Deep Learning Perspective, *Pattern Recognit.*, 2021, vol. 113, art. no. 107760.
17. Xia, L., Cui, J., Shen, R., et al., A Survey of Image Semantics-Based Visual Simultaneous Localization and Mapping: Application-Oriented Solutions to Autonomous Navigation of Mobile Robots, *Int. J. Adv. Robot. Syst.*, 2020, vol. 17, no. 3, art. no. 172988142091918.
18. Jin, X., Li, X., Zhu, Y., et al., A Survey of Visual Semantic Mapping, *Proceedings of 2022 3rd Asia-Pacific Conference on Image Processing, Electronics and Computers*, New York, NY, USA, 2022, pp. 295–301.
19. Arun, K.S., Huang, T.S., and Blostein, S.D., Least-Squares Fitting of Two 3-D Point Sets, *IEEE Trans. Pattern Anal. Mach. Intell*., 1987, vol. PAMI-9, no. 5, pp. 698–700.
20. Pramatarov, G., De Martini, D., Gadd, M., et al., BoxGraph: Semantic Place Recognition and Pose Estimation from 3D LiDAR, *arXiv:2206.15154v1*, 2022. DOI: 10.48550/arXiv.2206.15154
21. Wu, Y., Zhang, Y., Zhu, D., et al., An Object SLAM Framework for Association, Mapping, and High-Level Tasks, *arXiv:2305.07299*, 2023. DOI: https://doi.org/10.48550/arXiv.2305.07299
22. Ankenbauer, J., Lusk, P.C., Thomas, A., et al., Global Localization in Unstructured Environments Using Semantic Object Maps Built from Various Viewpoints, *Proceedings of 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Detroit, MI, USA, 2023, pp. 1358–1365.
23. Guo, X., Hu, J., Chen, J., et al., Semantic Histogram Based Graph Matching for Real-Time Multi-Robot Global Localization in Large Scale Environment, *IEEE Robot. Autom. Lett.*, 2021, vol. 6, no. 4, pp. 8349–8356.
24. Wang, Y., Jiang, C., and Chen, X., GOReloc: Graph-Based Object-Level Relocalization for Visual SLAM, *IEEE Robot. Autom. Lett*., 2024, vol. 9, no. 10, pp. 8234–8241.
25. Matsuzaki, S., Sugino, T., Tanaka, K., et al., CLIP-Loc: Multi-modal Landmark Association for Global Localization in Object-Based Maps, *Proceedings of 2024 IEEE International Conference on Robotics and Automation (ICRA)*, Yokohama, Japan, 2024, pp. 13673–13679.
26. Liu, Y., Petillot, Y., Lane, D., et al., Global Localization with Object-Level Semantics and Topology, *Proceedings of 2019 International Conference on Robotics and Automation (ICRA)*, Montreal, Canada, 2019, pp. 4909–4915.
27. *Handbook of Constraint Programming*, Rossi, F., van Beek, P., and Walsh, T., Eds., Amsterdam: Elsevier, 2006.
28. Narin'yani, A.S., Introduction to Subdefiniteness, *Information Technologies*, 2007, no. 4, pp. 1–32. (In Russian.)
29. Moscowsky, A., Subdefinite Computations for Reducing the Search Space in Mobile Robot Localization Task, in *Artificial Intelligence. RCAI 2021*, Kovalev, S.M., Kuznetsov, S.O., and Panov, A.I., Eds., Lecture Notes in Computer Science, Cham: Springer, 2021, vol. 12948, pp. 180–196.
30. Moscowsky, A.D., Extended Object Detection: Flexible Object Description System for Detection in Robotic Tasks, in *Smart Electromechanical Systems*, Gorodetskiy, A.E. and Tarasova, I.L., Eds., Studies in Systems, Decision and Control, Cham: Springer, 2022, vol. 419, pp. 27–43.
31. Contreras, M., Jain, A., Bhatt, N.P., et al., A Survey on 3D Object Detection in Real Time for Autonomous Driving, *Front. Robot. AI*, 2024, vol. 6, no. 11, art. no. 1212070

32. Moscowsky, A.D., Scene Recognition for the Mobile Robot Global Localization Problem Based on Image Vectorization and Graphs Approaches, *Large-Scale Systems Control*, 2025, no. 114, pp. 307–344. (In Russian.)
33. Radford, A., Kim, J.W., Hallacy, C., et al., Learning Transferable Visual Models from Natural Language Supervision, *Proceedings of the International Conference on Machine Learning*, Vienna, Austria, 2021, pp. 8748–8763.
34. Bazhenov, A.N., Zhilin, S.I., Kumkov, S.I., et. al., *Obrabotka i analiz interval'nykh dannykh* (Processing and Analysis of Interval Data), Moscow–Izhevsk: Institute of Computer Studies, 2024. (In Russian.)
35. Liao, Y., Xie, J., and Geiger, A., KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023, vol. 45, no. 3, pp. 3292–3310.
36. Keetha, N., Mishra, A., Karhade, J., et al., Anyloc: Towards Universal Visual Place Recognition, *arXiv:2308.00688*, 2023, DOI: https://doi.org/10.48550/arXiv.2308.00688
37. Puligilla, S., Omama, M., Zaidi, H., et al., LIP-Loc: LiDAR Image Pretraining for Cross-Modal Localization, *Proceedings of 2024 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, Waikoloa, HI, USA, 2024, pp. 939–948.

This paper was recommended for publication by R. V. Meshcheryakov, a member of the Editorial Board.

Received May 5, 2025,
and revised May 12, 2025.
Accepted July 16, 2025

Author information

Moscowsky, Anton Dmitrievich. Research group head, National Research Center "Kurchatov Institute," Moscow, Russia
✉ moscowskyad@yandex.ru
ORCID iD: https://orcid.org/0000-0002-6546-8697

Cite this paper

Moscowsky, A.D., Application of Subdefinite Models in the Global Localization of a Mobile Robot, *Control Sciences* **4,** 55–67 (2025).

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com