ISSN 2782-2427

CONTROL SCIENCES

1/2026

[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

Editor-in-Chief
D.A. Novikov, RAS Academician

Deputy Editor-in-Chief
F.F. Pashchenko

Executive Editor-in-Chief
N.E. Maximova

Editor
L.V. Petrakova

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: March 16, 2026

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

1.2026

CONTENTS

CONSTRUCTIVE D-PARTITION FOR TWO PARAMETERS ENTERING A POLYNOMIAL LINEARLY. PART II: Approximation of Stability Regions and Robustness Analysis

A. A. Tremba

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
Moscow Institute of Physics and Technology, Dolgoprudny, Russia

✉ atremba@ipu.ru

Abstract. For a polynomial linearly dependent on two parameters, several methods are proposed to approximate its stability region with respect to a given root localization region (also called a root clustering set in the literature). The first method is to apply a sufficiently uniform grid to the stability region boundary that ensures its complete coverage with a given accuracy. The second (semi-grid) method yields an internal approximation of the stability region using line segments or curve arcs bounded by the stability region. The third method is to cover the stability region boundary with simple sets (cells) in order to obtain piecewise linear internal and external approximations of the stability region. All methods are based on the constructive D-partition (constructive D-decomposition) method, which describes the stability region boundary as a set of line segments and rational curve arcs. The exact stability radius and its simple estimate are derived in the parameter plane. Implementation of all methods and algorithms is reduced to finding the real roots of polynomials.

Keywords: constructive D-partition, rational curves, approximation of the stability region, sufficiently uniform grid, grid methods, semi-grid methods, support function, stability radius.

INTRODUCTION

We analyze the location of the roots of a polynomial of degree n with two parameters entering it linearly:

$$G(s, k_1, k_2) = k_1 P(s) + k_2 Q(s) + R(s). \qquad (1)$$

For a given root localization region $\mathbf{D} \subset \mathbb{C}$, it is necessary to determine a set, called the *stability region*, on the parameter plane (k_1, k_2), each point of which corresponds to a stable polynomial:

$$D_n = \{(k_1, k_2) : \text{all } n \text{ roots of } G(s, k_1, k_2) \text{ lie in } \mathbf{D}\}. \qquad (2)$$

Recall that the stability (D-stability[1]) of roots and a polynomial is defined relative to a given root localization region $\mathbf{D}$ and generalizes the concept of Hurwitz and Schur polynomials corresponding to stable (in the classical sense) continuous- and discrete-time systems. The problem of finding the stability region arises in the controller design of linear control systems in the case when two controller parameters enter the characteristic polynomial of degree n linearly, or in the stability analysis of dynamics systems depending on two parameters (or one parameter, see subsection 2.1).

By selecting a root localization region $\mathbf{D}$, it is possible to ensure a desired stability degree or damping ratio of a closed-loop system, etc. [2–4]. By assumption, $\mathbf{D}$ is a regular open set. In this case, the stability region is also an open set.

This paper is the second part of the study [5], where the constructive D-partition was proposed, i.e., a description of the stability region boundary using the D-partition of the parameter plane. Recall the idea of this method; for details, see [6–10].

The first set is defined by a mapping of the boundary $\partial\mathbf{D}$ of the root localization region onto the parameter plane using the *main equation*

[1] There exists the concept of *D-stable matrices* with a significantly different meaning [1].

$$K_{bnd} = \{(k_1, k_2) : G(s,\ k_1, k_2) = 0_{\mathbb{C}},\ s \in \partial\mathbf{D}\}. \quad (3)$$

The second set is defined by the *degree drop condition* for the coefficient at s^n:

$$K_{deg} = \{(k_1, k_2) : G_n(k_1, k_2) = 0_{\mathbb{C}}\}. \quad (4)$$

The sets (3) and (4) define a D-partition, i.e., a division of the parameter plane into simply connected regions, some forming the stability region. In each D-partition region, the number of stable roots remains the same when continuously varying the parameters within this region, provided that the coefficients of the polynomial depend continuously on the parameters. The equation in formula (3) defines the marginal case in which at least one of the polynomial roots lies on the boundary $\partial\mathbf{D}$.

Let us present several properties of the D-partition for the polynomial (1), together with the results from part I of the study [5].

First, the set K_{bnd} consists of the so-called main curve and a set of (straight) lines, called *singular lines*, defined by the equations

$$a_i k_1 + b_i k_2 + c_i = 0,\ \ i = 1, \ldots,\ M, \quad (5)$$

and K_{deg} is a singular line (possibly an empty set) defined by the equation

$$P_n k_1 + Q_n k_2 + R_n = 0.$$

The stability region boundary belongs to the union of these sets:

$$\partial D_n \subset K_{bnd} \cup K_{deg}.$$

Second, if the boundary $\Gamma = \partial\mathbf{D}$ of a root localization region consists of a finite set of rational curve arcs,

$$\Gamma = \bigcup\nolimits_{\ell} \Gamma_{\ell},\ \ \Gamma_{\ell} = \{s_{\ell}(w) \in \mathbb{C} : w \in W_{\ell}\},$$
$$\ell = 1, \ldots,\ L,$$

then the main equation in (3) is equivalent to L systems of two polynomial equations with some polynomials $P_{\ell,1}(w)$, $P_{\ell,2}(w)$, $Q_{\ell,1}(w)$, $Q_{\ell,2}(w)$, $R_{\ell,1}(w)$, and $R_{\ell,2}(w)$, depending on the initial polynomials P, Q, R and the rational functions $s_{\ell}(w)$:

$$\begin{cases} k_1 P_{\ell,1}(w) + k_2 Q_{\ell,1}(w) + R_{\ell,1}(w) = 0 \\ k_1 P_{\ell,2}(w) + k_2 Q_{\ell,2}(w) + R_{\ell,2}(w) = 0 \\ w \in W_{\ell},\ \ell = 1, \ldots, L. \end{cases} \quad (6)$$

The solution of each system is a rational curve (*main curve*) of the form

$$k_{\ell,1}(w) = \frac{1}{\det T_{\ell}(w)}\big(R_{\ell,2}(w)\ Q_{\ell,1}(w) - R_{\ell,1}(w)\ Q_{\ell,2}(w)\big),$$
$$k_{\ell,2}(w) = \frac{1}{\det T_{\ell}(w)}\big(R_{\ell,1}(w)\ P_{\ell,2}(w) - R_{\ell,2}(w)\ P_{\ell,1}(w)\big), \quad (7)$$
$$w \in W_{\ell},$$

and a set of singular lines of the form (5) corresponding, for each $\ell = 1, \ldots, L$, to such real roots (critical frequencies) of the equation

$$\det T_{\ell}(w) = P_{\ell,1}(w) Q_{\ell,2}(w) - Q_{\ell,1}(w) P_{\ell,2}(w) = 0,\ \ w \in W_{\ell},$$

for which system (6) has a solution. Hereinafter, the subscript ℓ will be occasionally omitted for brevity.

Third, it has been proven that for a localized D-partition (bounded to a compact set $\boldsymbol{K}$), the above rational curves are defined (or redefined) on closed intervals (segments), and singular lines become segments. Moreover, if the boundary of $\boldsymbol{K}$ consists of a finite set of rational curve arcs, the D-partition will also consist of a finite number of rational curve arcs and segments. The constructive D-partition method proposed allows determining the parameterization intervals of these rational curve arcs and segments as sections of singular lines by finding the real roots of polynomials.

In particular, the boundary of the stability region (2) within the set $\boldsymbol{K}$ consists of a finite set of rational functions of the form (7) and segments. It is convenient to represent the segments in parametric form as

$$k(t) = t\ d + p,\ \ t \in [t_1, t_2],\ p, d \in \mathbb{R}^2,\ d \neq 0,$$

where the vectors $d = (-b,\ a)^{\mathrm{T}}$ and $p = -\left(\frac{ac}{a^2 + b^2}, \frac{bc}{a^2 + b^2}\right)$ are obtained from the equations of the lines (5); hereinafter, the subscripts are omitted for brevity.

The parameter and intersection point of the line initially defined as $k(t) = t\ d + p$, $t \in (-\infty, +\infty)$, and the line (5) are given by

$$t^* = -\frac{c + ap_1 + bp_2}{ad_1 + bd_2} = -\frac{c + (a,\ b)p}{(a,\ b)d},$$
$$k^* = t^* d + p = -\frac{c + (a,\ b)p}{(a,\ b)d}\ d + p. \quad (8)$$

We emphasize one feature of constructing a localized D-partition. The rational curve arcs of the boundaries of the stability region (7) are defined on closed intervals, but there may exist an infinite interval, e.g., $(-\infty, +\infty)$ or $[w_1, +\infty)$. The domain is reduced to a finite closed interval by an appropriate change of the parameter, see subsection 1.2.

The Intersection of Main Curves, Lines, and Localized D-Partition

Recall some results concerning the intersection of rational curves and lines. In particular, the intersection of the singular lines (5) and the main curve (7) of the D-partition is given by the equations

$$\begin{gathered} a_i\left(R_2(w)Q_1(w)-R_1(w)Q_2(w)\right) \\ +b_i\left(R_1(w)P_2(w)-R_2(w)P_1(w)\right) \\ +c_i\left(P_1(w)Q_2(w)-Q_1(w)P_2(w)\right)=0, \\ i=0,\ldots,K. \end{gathered} \tag{9}$$

Here, the subscript 0 corresponds to the singular line K_{deg}. The equations are polynomial with respect to w, and their real roots w_m, $m=1,\ldots$, can be calculated explicitly. These roots, together with the critical frequencies w_i, split the interval W into segments and intervals corresponding to the simple continuous parts of the D-partition boundaries, the arcs of the main curve. In fact, these arcs form the curved part of the D-partition. The intersection points themselves are determined from equation (7) as $k(w_m)=\left(k_1(w_m), k_2(w_m)\right)$. Together with the limit points $\lim\limits_{w\to w_m} k(w)$ (if any), they divide the main curve into arcs and, moreover, singular lines into segments or infinite intervals (rays). In the case of a localized D-partition, there are no infinite intervals. Similarly, one can find the self-intersection and intersection points of several main curves; see part I of the study [5].

It is convenient to perform a localized D-partition in a rectangle $\boldsymbol{K}=\left[\underline{k}_1, \overline{k}_1\right]\times\left[\underline{k}_2, \overline{k}_2\right]$, whose boundaries are vertical and horizontal segments. For these segments, equation (9) gets simplified since it suffices to consider each component of the rational curve (7) separately:

$$\begin{gathered} R_2(w)Q_1(w)-R_1(w)Q_2(w) \\ =x\left(P_1(w)Q_2(w)-Q_1(w)P_2(w)\right), \\ x=\underline{k}_1, \overline{k}_1, \end{gathered} \tag{10}$$

$$\begin{gathered} R_1(w)P_2(w)-R_2(w)P_1(w) \\ =y\left(P_1(w)Q_2(w)-Q_1(w)P_2(w)\right), \\ y=\underline{k}_2, \overline{k}_2. \end{gathered} \tag{11}$$

Having solved each of these equations, one should check that the second coordinate is within the desired interval (i.e., the main curve intersects a rectangle's side). For example, if the roots of equation (10) are w_m, then only those are selected for which $k_2(w_m)\in\left[\underline{k}_2, \overline{k}_2\right]$, and vice versa.

In what follows, based on the constructive description of the stability region boundary, we propose several approximations for the stability region boundary or the region itself. In addition, the method proposed is applied to perform stability region analysis (subsection 4.3) and robust analysis (subsection 4.4). The resulting parameterization of the stability region boundary can be used for subsequent optimization within the stability region and other problems related to the analysis and design of controllers with performance characteristics determined by the root localization region.

1. POINTWISE APPROXIMATION OF THE D-PARTITION BOUNDARY

Consider a localized D-partition on a bounded set $\boldsymbol{K}$, e.g., on a rectangle $\left[\underline{k}_1, \overline{k}_1\right]\times\left[\underline{k}_2, \overline{k}_2\right]$. After applying the constructive D-partition algorithm (8), (10), (11), it is possible to determine the intersection of the singular lines (5) and main curves (7) of the D-partition with the boundaries of the set $\boldsymbol{K}$. The resulting stability region D_n will be bounded by a finite number of segments $k_m(t)$ and main curve arcs $k_\ell(w)$:

$$\partial D_n=\cup_m k_m(t)\cup_\ell k_\ell(w).$$

According to Lemmas 1 and 4 from part I of the study [5], each of the boundary parts $k_m(t)$, $k_\ell(w)$ is parameterized by a segment $W_\ell=\left[w_{\ell,1}, w_{\ell,2}\right]$ or a closed infinite interval. Moreover, by Theorem 1 [5], it is possible to replace an infinite parameterization interval with a segment (or segments). This replacement, in the context of curve approximation, will be considered separately in subsection 1.2.

The problem is to select a finite set of points (*nodes*) forming a grid $K_{grid}=\{k_r, r=1,\ldots\}$ on these curves and segments in order to approximate the stability region boundary with a given fineness $\rho>0$.

The grid fineness relative to a set is defined by the Hausdorff distance between the set of nodes and the given set (in the case under consideration, the stability region boundary). Let us select the nodes lying on the boundary: $k_r \in \partial D_n$. They satisfy the condition

$$\operatorname{dist}_{\mathrm{H}}\left(K_{grid},\, \partial D_n\right) = \max_{k\in\partial D_n} \min_{r} \|k - k_r\| \le \rho. \quad (12)$$

Thus, an appropriate grid is built if and only if circles of radius ρ, centered at k_r, cover the boundary. Such an approximation and the corresponding grid will be called *sufficiently uniform*.

The boundary is easily approximated for each of its arcs separately, by explicitly adding nodes at the junction points of the boundary arcs. For the segments $k_m(t) = t\,d_m + p_m,\ t \in \left[t_{m,1}, t_{m,2}\right]$, we propose an obvious uniform approximation with N intervals, including both endpoints of the segment:

$$k_{m,r} = \left(t_{m,1} + r\delta\right)d_m + p_m,\ r = 0,1,\ldots,N,$$
$$\delta = \frac{t_{m,2} - t_{m,1}}{N},\ N = \left\lceil \frac{t_{m,2} - t_{m,1}}{2\rho} \right\rceil. \quad (13)$$

The distance between two nodes does not exceed 2ρ. This grid is optimal among all grids containing endpoints, in the sense of its uniformity and a minimum number of nodes.

The situation is more complicated for boundary arcs $k_\ell(w)$ defined on closed intervals. Recall that on these intervals, the function $k_\ell(w)$ is continuous. Consider one arc, $k(w), w \in W$, omitting the subscript.

Formally, a smooth curve arc can be uniformly approximated, with any given accuracy, by the so-called natural parameterization of a curve arc using the curve length function from the point $k(w_1)$ in the direction of increasing the parameter value. The length function can be defined by the "speed" of a point along the curve, $v(w) = \|k'(w)\| = \sqrt{k_1'(w)^2 + k_2'(w)^2}$, which will be called the parametric speed (or simply the speed). The length function is given by

$$\lambda_{w_1}(w) = \int_{w_1}^{w} v(\tau)\,d\tau,\ w \ge w_1,$$
and (14)
$$\lambda_{w_1}(w) = -\int_{w_1}^{w} v(\tau)\,d\tau,\ w < w_1.$$

For a monotonic function $\lambda(w), w \ge w_1$, the inverse function $w(\lambda)$ is built. Then, the segment $\left[0,\, \lambda(w_2)\right]$ is evenly split by the numbers λ_r into segments of a maximum length of 2ρ, similar to formula (13). The resulting set $k_r = k\left(w(\lambda_r)\right)$ divides the curve $k(w), w \in W$, into arcs of equal length, each no longer than 2ρ. Due to the continuity of the curve and the triangle rule, the distance from each arc point to one of the neighboring nodes, including the endpoints $k(w_1)$ and $k(w_2)$, does not exceed ρ. Unfortunately, the integral (and then the inverse function) cannot usually be calculated in analytic form since the speed vector components $k_1'(w)$, $k_2'(w)$ are defined by rational functions.

1.1. Algorithms for Building a Sufficiently Uniform Grid

We propose two algorithms with an upper estimate of the speed on subintervals and sequential addition of grid nodes. A feature of the algorithms is non-uniform speed estimation for more efficient use of grid nodes. The resulting grid satisfying condition (12) will be sufficiently uniform; in the first algorithm, the nodes on the curve will be located regularly (with respect to w).

Let us calculate the set of points containing the extremum values of the speed within the segment W_ℓ using the necessary optimality condition:

$$\begin{aligned} &\operatorname{Arg\,extr}_{w\in W_\ell} v(w) \\ &\in \operatorname{Arg\,extr}_{w\in W_\ell} v(w)^2 \in \left\{w : \left(v(w)^2\right)' = 0\right\} \\ &= \left\{w : k_{\ell,1}'(w)k_{\ell,1}''(w) + k_{\ell,2}'(w)k_{\ell,2}''(w) = 0\right\}. \end{aligned} \quad (15)$$

According to the last expression, due to the rationality of the functions, stationary points are found by calculating the roots of the corresponding polynomial. Denoting by $\{w_s\}$ the set of stationary points, we define a function $v_{\max}(w_a, w_b)$ returning the maximum value of the speed on an arbitrary segment $\left[w_a, w_b\right] \in W_\ell$, $w_a < w_b$:

$$\begin{aligned} v_{\max}(w_a, w_b) = \max\big\{&v(w_a), v(w_b), \\ &\max\left\{v(w_s) : w_s \in (w_a, w_b)\right\}\big\}. \end{aligned} \quad (16)$$

Formula (16) employs Fermat's theorem, i.e., the fact that on a segment (closed interval), the maximum value of a continuous differentiable function is achieved either at the ends of or within this segment.

For inner maximum points, the necessary optimality condition holds, which is valid for the points from the set $\{w_s\}$. The function $v_{\max}(w_a, w_b)$ can be calculated more easily by considering, among the stationary points, only those of (local) maximum defined by the second-order optimality condition $v''(w_s)<0$. If the second derivative is equal to zero, it is necessary to analyze higher-order derivatives or treat such points as candidates.

Thus, according to formula (14), on the segment $[w_a, w_b]$, the length of the curve from the endpoint $k_\ell(w_a)$ to the intermediate point $k_\ell(w)$ is estimated as

$$\lambda_{w_a}(w) \le v_{\max}(w_a, w_b)\cdot(w - w_a),\ w \in [w_a, w_b]. \quad (17)$$

A similar estimate is valid for the curve length in the opposite direction, from w_b to w. Based on these estimates and the triangle rule, for each half of the curve before and after $w_c = (w_a + w_b)/2$, the curve lengths satisfy the following upper and lower bounds:

$$\|k_\ell(w_a) - k_\ell(w)\| \le \lambda_{w_a}(w) \le \lambda_{w_a}(w_c)$$

$$\le v_{\max}(w_a, w_b)\cdot\frac{w_b - w_a}{2},\ w \in [w_a, w_c],$$

$$\|k_\ell(w_b) - k_\ell(w)\| \le \lambda_{w_b}(w) \le \lambda_{w_b}(w_c)$$

$$\le v_{\max}(w_a, w_b)\cdot\frac{w_b - w_a}{2},\ w \in [w_c, w_b].$$

Therefore, one arrives at the sufficient criterion: if $v_{\max}(w_a, w_b)\cdot(w_b - w_a) \le 2\rho$, then the arc of the curve $k_\ell(w)$, $w \in [w_a, w_b]$, lies in the union of two circles of radius ρ with centers at $k(w_a)$ and $k(w_b)$.

We propose the following iterative algorithm for finding a sufficiently uniform grid K_{set} that covers the arc of the curve $k_\ell(w)$, $w \in W_\ell$, using the function $v_{\max}(w_a, w_b)$ on subintervals.

Algorithm 1. A sufficiently uniform grid for a rational curve arc.

Input: a rational curve $k_\ell(w)$, $w \in W_\ell = [w_1, w_2]$, whose denominator does not vanish on the interval W_ℓ, and a fineness parameter $\rho > 0$.

1. Compute the stationary points $\{w_s\}$ for the speed by formula (15) (or the local maximum points) belonging to W_ℓ, and determine the upper bound function $v_{\max}(w_a, w_b)$.

2. Set the initial list of nodes as a list of two nodes $\{k(w_1), k(w_2)\}$. Set the initial list of intervals as a list containing one element, i.e., the interval $[w_1, w_2]$.

3. If the list of intervals is empty, terminate the algorithm and return the list of nodes. Otherwise, select any interval (e.g., the first or leftmost one), and designate it as $[w_a, w_b]$; remove this interval from the list.

4. If $v_{\max}(w_a, w_b)\cdot(w_b - w_a) \le 2\rho$, go to Step 3.

5. Split the interval $[w_a, w_b]$ in half at the point $w_c = (w_a + w_b)/2 \in (w_a, w_b)$; add $k(w_c)$ to the list of nodes, and add the intervals $[w_a, w_c]$ and $[w_c, w_b]$ to the list of intervals; get back to Step 3.

Output: a set of parameters from the interval $[w_1, w_2]$ and the corresponding set of nodes (a sufficiently uniform grid).

Algorithm 1 is finite, as the parametric speed is bounded above on the entire interval W_ℓ; see the estimate (18) below. By construction, the grid obtained using Algorithm 1 covers the curve with circles of radius ρ and, moreover, divides the curve into arcs of a maximum length of 2ρ. Algorithm 1 can be trivially generalized to any curves with a speed estimate of the form $\bar{v}(w) \ge v(w)$. In addition, the algorithm is effectively implemented if this speed estimate is a polynomial or other function with simple or precomputed extrema (maxima). Figure 1 shows the parametric speed and a sufficiently uniform grid for the arc of the stability region boundary in Example 1 (see Section 5).

In the particular case of using the maximum speed $\bar{v}(w) = v_{\max}(w_1, w_2) = \text{const}$ as the upper estimate, Algorithm 1 outputs a uniform (binary) grid of the form

$$k_r = k(w_r),\ w_r = w_1 + r\,\delta_2,\ \delta_2 = \frac{w_2 - w_1}{2^M},$$

$$M = \max\left\{0, \left\lceil \log_2 \frac{v_{\max}(w_1, w_2)(w_2 - w_1)}{\rho} \right\rceil - 1\right\}, \quad (18)$$

$$r = 0, \ldots, 2^M.$$

In terms of the number of nodes, the uniform binary grid is generally worse than (but at most twice as bad as) the uniform grid based on the maximum speed

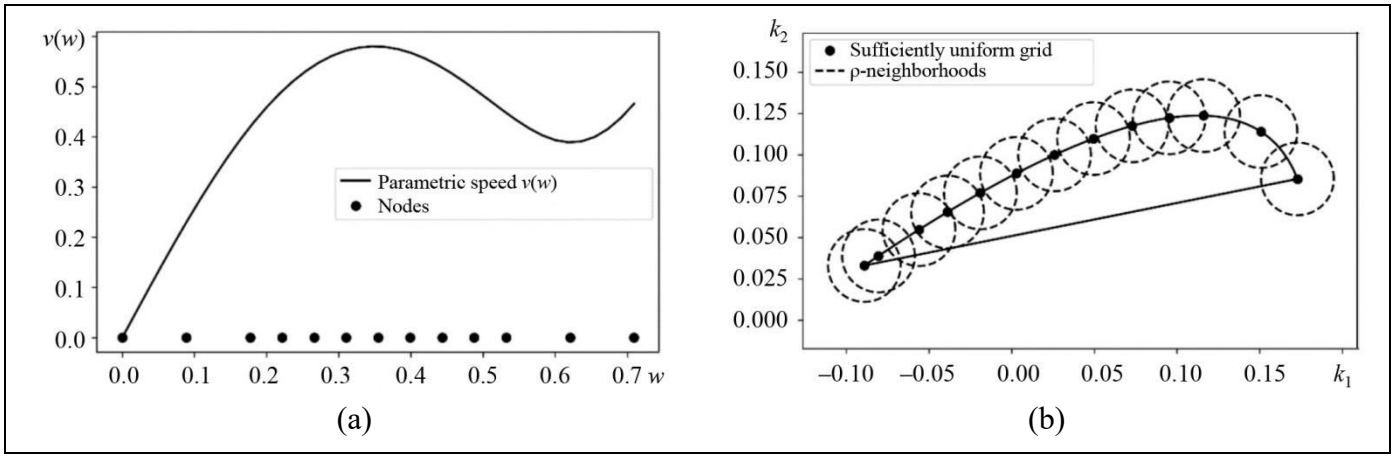

Fig. 1. (a) The parametric speed for the curved part of the stability region boundary in Example 1 and (b) the sufficiently uniform grid obtained by Algorithm 1, together with the set of covering circles.

estimate $v_{\max}(w_1, w_2)$, built similarly to formula (13) as

$$w_r = w_1 + r\delta,\ r = 0, 1, \ldots, N,\ \delta = \frac{w_2 - w_1}{N},$$
$$N = \left\lceil \frac{v_{\max}(w_1, w_2)(w_2 - w_1)}{2\rho} \right\rceil. \qquad (19)$$

In Algorithm 1, the distance between the grid nodes $k_r = k(w_r)$ is adapted to the maximum speed in each subinterval, and this algorithm has higher efficiency in practice (significantly fewer nodes compared to the uniform grid (19)). The reason consists in that the speed function is related to rational functions, therefore being significantly nonuniform (especially at the edges of the definitional intervals).

Note that the segment is split by taking its middle point $w_c = (w_a + w_b)/2$, and the intervals yielded by Algorithm 1 will be multiples of the smallest interval $\delta_2 = 2^{-M}(w_2 - w_1)$ for the same M. Indeed, the maximum speed value is reached on at least one of the subintervals.

The function $v_{\max}(w_a, w_b)$ in formula (17) is calculated trivially if the interval contains no extrema:

$$v_{\max}(w_a, w_b) = \max\{v(w_a), v(w_b)\}$$
$$\text{if } \nexists w_s \in (w_a, w_b).$$

Using this fact, Algorithm 1 can be simplified by immediately splitting the interval $[w_1, w_2]$ by stationary points (or maximum points). Then, on each subinterval, the maximum speed will be determined exclusively by the speeds at the endpoints of the segment, and the same applies to the subsequent division into smaller segments.

Algorithm 2. A sufficiently uniform grid for a piecewise rational curve arc (the simplified version).

Input: a rational curve $k_\ell(w)$, $w \in W_\ell = [w_1, w_2]$, whose denominator does not vanish on the interval W_ℓ, and a fineness parameter $\rho > 0$.

1. Compute the stationary points $\{w_s\}$ for the speed by formula (15) (or the local maximum points) belonging to W_ℓ; split the interval $[w_1, w_2]$ by these points.

2. Set the initial list of nodes as $\{k(w_1), k(w_2)\} \cup \{k(w_s)\}$. Set the initial list of intervals obtained by splitting the interval $[w_1, w_2]$ by the set $\{w_s\}$.

3. If the list of intervals is empty, terminate the algorithm and return the list of nodes. Otherwise, select any interval (e.g., the first or leftmost one), and designate it as $[w_a, w_b]$; remove this interval from the list.

4. If $\max\{v(w_a), v(w_b)\} \cdot (w_b - w_a) \le 2\rho$, get back to Step 3.

5. Split the interval $[w_a, w_b]$ in half at the point $w_c = (w_a + w_b)/2 \in (w_a, w_b)$; add $k(w_c)$ to the list of nodes, and add the intervals $[w_a, w_c]$ and $[w_c, w_b]$ to the list of intervals; get back to Step 3.

Output: a set of parameters from the interval $[w_1, w_2]$ and the corresponding set of nodes (a sufficiently uniform grid). Algorithms 1 and 2 have the same output.

Compared to Algorithm 1, Algorithm 2 requires fewer computations (no need to check the belonging of the stationary parameters $\{w_s\}$ to the current interval), but the grid turns out to be irregular.

1.2. Infinite Intervals and Reparameterization

Algorithms 1 and 2 are supplemented by the following considerations. First, for a localized D-partition, the finite (possibly redefined) domain interval of an arc of the rational curve $k_\ell(w)$ (7) within the set $\boldsymbol{K}$ is always closed; see Lemmas 1–3 in part I of the study [5]. To apply approximation on infinite intervals with respect to w, a special change of variables is introduced [11]. Without loss of generality, let a rational curve $k_\ell(w)$ be defined and continuous on the interval $W_\ell = [w_1, \infty)$. We choose an endpoint $w_0 < w_1$, i.e., some finite value $w_0 \neq -\infty$, e.g., $w_0 = w_1 - 1$. Since the D-partition is localized, there exists the limit point $k_\infty = \lim\limits_{w\to\infty} k_\ell(w) \in \boldsymbol{K}$; otherwise, Lemma 3 [5] is valid, and it suffices to consider a finite interval on which the curve lies inside the set $\boldsymbol{K}$. With the change of variable $w = w_0 + 1/u$, $u \in U = \left(0, 1/(w_1 - w_0)\right] \subset (0, 1]$, we obtain another parameterization of the same curve arc $k_{\ell,u}(u) = k_\ell(w_0 + 1/u)$, $u \in U$; the node $k_{\ell,u}(0)$ is replaced by the same limit point $k_\infty = \lim\limits_{u\to+0} k_{\ell,u}(u)$, and the speed value at this point is considered to be zero. This change allows obtaining a parameterization of the finite boundary arcs on a finite interval. The case $W_\ell = (-\infty, +\infty)$ leads to a division into two finite intervals after replacing the parameter.

A similar redefinition of the speed with a zero value is applied when redefining the rational function at the endpoint of the interval $[w_1,\ldots)$ if w_1 is a root of its denominator but the limit point $\lim\limits_{w\to w_1} k_\ell(w)$ exists.

Note that for a nonlocalized D-partition, parts of the boundary $k_\ell(w)$ may be unbounded due to an unbounded or open interval W_ℓ, including the cases where the boundary of the interval W_ℓ is zero of the denominator for one component of $k_\ell(w)$. In these cases, the limit of $k_\ell(w)$ does not exist, and an unbounded curve cannot be approximated by a finite number of points with a given accuracy. In such cases, the rational curve (7) has asymptotes with the ratio

$$k_{\ell,1}(w) : k_{\ell,2}(w) = R_{\ell,2}(w)\ Q_{\ell,1}(w) - R_{\ell,1}(w)\ Q_{\ell,2}(w) : R_{\ell,1}(w)\ P_{\ell,2}(w) - R_{\ell,2}(w)\ P_{\ell,1}(w) \quad \text{as } w \to \infty,$$

and a set of points combined with rays can be considered a set approximating the D-partition. Analysis of the quality and accuracy of such an approximation is a separate problem going beyond the scope of this paper.

2. SEMI-GRID METHODS

In addition to the pointwise approximation method (Section 1), several numerical methods can be proposed for estimating both the boundaries and regions of a D-partition. They involve two approaches as follows.

The idea of the first approach is to parameterize the parameter plane k_1, k_2 using two auxiliary parameters and form a discrete grid based on an auxiliary parameter. Next, it is necessary to take the resulting set of lines (continuous with respect to the second auxiliary parameter, e.g., straight lines or curves) and find the intersection of the D-partition boundary and this grid. First, the intersections will provide a pointwise approximation of the boundary. Second, the stability intervals determined on these lines (the latter's parts falling within the stability region) will provide an internal approximation of the stability region. The corresponding methods, based on the stratification of the parameter space, will be called semi-grid ones. The simplest example is a set of horizontal (or vertical) lines.

Essentially, the one-dimensional parameterization (decomposition, slicing, or stratification) of the parameter plane/space and a grid for this parameterization are used. In the case of three or more parameters, a similar idea is employed to visualize a three-dimensional D-partition; see subsection 2.5.

The second approach is to divide the parameter plane into simple cell sets K_i and select those intersecting the stability region boundary. These sets form a covering of the stability region boundary, and, therefore, the complement to their union contains an internal approximation of the stability region on the parameter plane. In this case, an internal approximation of the stability region is formed on one part of the boundary of the covering set union, and an external approximation is formed on the other part (Fig. 3). We propose more effective checks for the intersection of the D-partition boundary and the sets K_i by selecting the latter as elements of a regular grid.

Both approaches use the constructive D-partition in the form of a set of arcs of stability region boundaries or the one-dimensional D-partition. For the second approach, the constructive D-partition is necessary not to "skip" the stability region components lying entirely in one of the cells.

We will present several semi-grid methods, and then, in Section 3, several grid methods, preceded by a description of the one-dimensional D-partition.

2.1. One-Dimensional D-Partition and Its Connection to Constructive D-Partition

The one-dimensional D-partition refers to a polynomial $G(s, t)$ whose coefficients depend on a scalar parameter $t \in \mathbb{R}$. In this case, the real line of the parameter is divided, by certain points t_i, into segments and rays corresponding to the D-partition regions. The points t_i defining the boundaries of the one-dimensional D-partition will be called critical points. As in the case of two parameters, they are defined by the main equation

$$\begin{aligned} G(s, t) = G_n(t)s^n + G_{n-1}(t)s^{n-1} + \dots \\ + G_1(t)s + G_0(t) = 0_{\mathbb{C}},\ s \in \Gamma, \end{aligned} \tag{20}$$

and the degree drop condition $G_n(t) = 0$. If the boundary Γ of the root localization region is described by a piecewise rational curve with respect to the parameter w, and the polynomial G depends on t polynomially, then the main equation can be reduced to a system of two polynomial equations with two unknowns.

The situation is simplified if the polynomial linearly depends on t; then the main equation takes the form

$$G(s, t) = t\,P_t(s) + R_t(s) = 0_{\mathbb{C}}. \tag{21}$$

It can be solved explicitly, assuming the absence of common roots[2] of the polynomials $P_t(s)$ and $R_t(s)$ on Γ. The critical points t_i splitting the parameter line into segments (and rays) with a constant number of stable roots satisfy the equation

$$t = -\frac{R_t(s(w))}{P_t(s(w))}. \tag{22}$$

[2] Otherwise, the polynomial is obviously unstable since the root on the boundary does not belong to the open root localization region. This case is analogous to the case of two parameters with no common roots for the polynomials P, Q, and R. If we consider non-open sets **D** and the common root on the boundary is supposed to be stable, it can be reduced.

In view of the complex-valued right-hand side, we can eliminate the variable t by solving the equation with respect to w:

$$\operatorname{Im}\frac{R_t(s(w))}{P_t(s(w))} = 0.$$

This equation is equivalent to

$$\begin{aligned} &\operatorname{Im} R_t(s(w))\ \operatorname{Re} P_t(s(w)) \\ &= \operatorname{Re} R_t(s(w))\ \operatorname{Im} P_t(s(w)),\ \ s \in \Gamma, \end{aligned} \tag{23}$$

except those points where $P_t(s(w)) = 0$. Such points are not the solutions of the original equation (21) since $R_t(s(w)) \neq 0$ at them due to the absence of common roots. Equation (23) is reduced to a polynomial one with respect to w for rational functions $s(w)$. Next, the roots w_i found are substituted into equation (22) to get t_i. They are supplemented with the root of the degree drop equation, $t_0 = R_{t,n} / P_{t,n}$, if it is real and $P_{t,n} \neq 0$. Finally, for the intervals (t_i, t_{i+1}) obtained by splitting the parameter line division, the number of stable roots is determined, e.g., by the polynomial's roots at the midpoints of the intervals, $G(s, (t_i + t_{i+1})/2)$, and at two points outside the interval $[\min_i t_i, \max_i t_i]$.

Similar to the two-dimensional D-partition, for a rational boundary function $s(w)$, one can first reduce equation (21) to a polynomial dependence on w and then proceed to the relations (22) and (23). If the boundary Γ of the root localization region consists of several arcs, equation (21) shall be solved for each arc.

In the general case, for a polynomial linearly dependent on m parameters, the one-dimensional D-partition allows explicitly finding the intersection of the stability region and an arbitrary line (or segment)

$$k(t) = t\,d + p,\ t \in [t_1, t_2],\ p, d \in \mathbb{R}^m,$$

where the boundaries can be either bounded or unbounded. In the two-dimensional case, the polynomial (1) takes the form (21):

$$\begin{aligned} &G(s, t) = G(s, k_1(t), k_2(t)) \\ &= (d_1 P(s) + d_2 Q(s))t + p_1 P(s) + p_2 Q(s) + R(s). \end{aligned} \tag{24}$$

The critical points t_i are the roots of a certain polynomial according to formula (23). Among them, only those belonging to the interval $[t_1, t_2]$ are selected. This method is used to analyze arbitrary lines and

segments in the parameter space of any dimension, particularly as an auxiliary method for analyzing implicitly defined sets, i.e., as an oracle determining the intersection of a line and a stability region [12].

Note that if, for a polynomial depending on two parameters, the constructive D-partition is used to characterize the stability region boundaries in the form of a set of curves and lines, then the intersection of the stability region and lines in the parameter space can be easily obtained from the intersection of the D-partition boundaries and a line. As a result, we get the same stability segments on the line under consideration as with the one-dimensional D-partition.

It is difficult to compare the effectiveness of the two approaches (the intersection with boundary curve arcs and the one-dimensional D-partition) a priori, since their complexity depends on the number of arcs of the stability region boundary and the degrees of the auxiliary polynomials. One should keep in mind that the coordinates of the arcs of the stability region boundary are easily estimated on the plane, see subsection 4.3. With the estimated location of the boundary arcs, the absence of intersection of segments (and lines) and the D-partition regions can be checked effectively, e.g., by comparing rectangles containing the boundary arcs of the D-partition regions. On the other hand, the one-dimensional D-partition for the polynomial (24), which is linear with respect to the parameter, is implemented by solving equation (21) directly, without the need to divide the D-partition boundary into the main curve and singular lines.

2.2. The Semi-Grid Method with Parallel Lines

As mentioned above, it is easiest to apply the method by fixing one parameter. In the case of two parameters, the fixed first (second) parameter on the plane k_1, k_2 corresponds to vertical (horizontal, respectively) lines. If a localization region $\boldsymbol{K} = K_1 \times K_2 = \left[\underline{k}_1, \overline{k}_1\right] \times \left[\underline{k}_2, \overline{k}_2\right]$ is defined, it is easiest to take a uniform grid, based either on fineness or on the number of lines $N+1$; see examples in Figs. 2a and 2b. For instance, vertical lines $\boldsymbol{K}_i$ are defined in explicit and parametric form as

$$\begin{gathered} k_1 - k_{1,i} = 0,\ k_{1,i} = \underline{k}_1 + i\frac{\overline{k}_1 - \underline{k}_1}{N},\ \ i = 0,\ldots,N, \\ \begin{pmatrix} 0 \\ \overline{k}_2 - \underline{k}_2 \end{pmatrix} t + \begin{pmatrix} k_{1,i} \\ \underline{k}_2 \end{pmatrix},\ t \in [0,1],\ \ i = 0,\ldots,N. \end{gathered} \tag{25}$$

The intersection of the D-partition boundaries is given by equations (10) and (11) for the main curve and (8) for lines; alternatively, one can apply the one-dimensional D-partition of the polynomial (24) with respect to t using the parameterization (25). On each line, segments are selected for which the polynomial is stable. The set of such segments on all lines $\boldsymbol{K}_i$ corresponding to the stability regions forms an internal approximation of the stability region.

2.3. The Semi-Grid Method with Rays (Angular Grid)

Consider a point k_0 on the parameter plane, e.g., corresponding to a stable polynomial. We select several rays passing through this point and analyze the intersection of these rays and the D-partition boundaries. For construction purposes, it is more convenient to use lines (rather than rays) whose inclination angle relative to the axis is uniformly distributed on a semicircle. For example, for N lines, a grid is described by the equation

$$\begin{gathered} \left(k_1 - k_{0,1}\right)\sin\frac{i\pi}{N} - \left(k_2 - k_{0,2}\right)\cos\frac{i\pi}{N} = 0, \\ i = 0,\ \ldots,\ N-1, \end{gathered} \tag{26}$$

or, in the parametric form, $k_0 + t\left(\cos\frac{i\pi}{N},\ \ \sin\frac{i\pi}{N}\right)^{\mathrm{T}}$. In contrast to the case of lines parallel to the coordinate axes, the intersection points with the boundary arcs of the stability region shall be found using formulas (8) and (9). An alternative is to use the one-dimensional D-partition. If the point k_0 passed by the lines corresponds to a stable polynomial, then the minimum positive and negative parameters of the line shall be taken: $\max_{m:t_m<0} t_m \le t \le \min_{m:t_m>0} t_m$, where all t_m correspond to the intersection points with the main curve and singular curves. In this case, we find the intersection of the line and the component of the stability region containing the point k_0. An example of an angular grid is shown in Fig. 2c.

Note that when building a grid by angle, a set of randomly directed lines can be taken as well, and the point k_0 does not necessarily correspond to a stable polynomial.

2.4. The Semi-Grid Method with Concentric Circles (Radial Grid)

Finally, consider a parameterization using a set of concentric circles $\boldsymbol{K}_i = \left\{k : \|k - k_0\| = r_i\right\}$. The intersection points of the boundaries of the D-partition and the circles are found explicitly: for singular lines in the parametric form $t\,d + p$, from the quadratic equation

$$\left(t\,d_1 + p_1 - k_{0,1}\right)^2 + \left(t\,d_2 + p_2 - k_{0,2}\right)^2 = r_i^2; \tag{27}$$

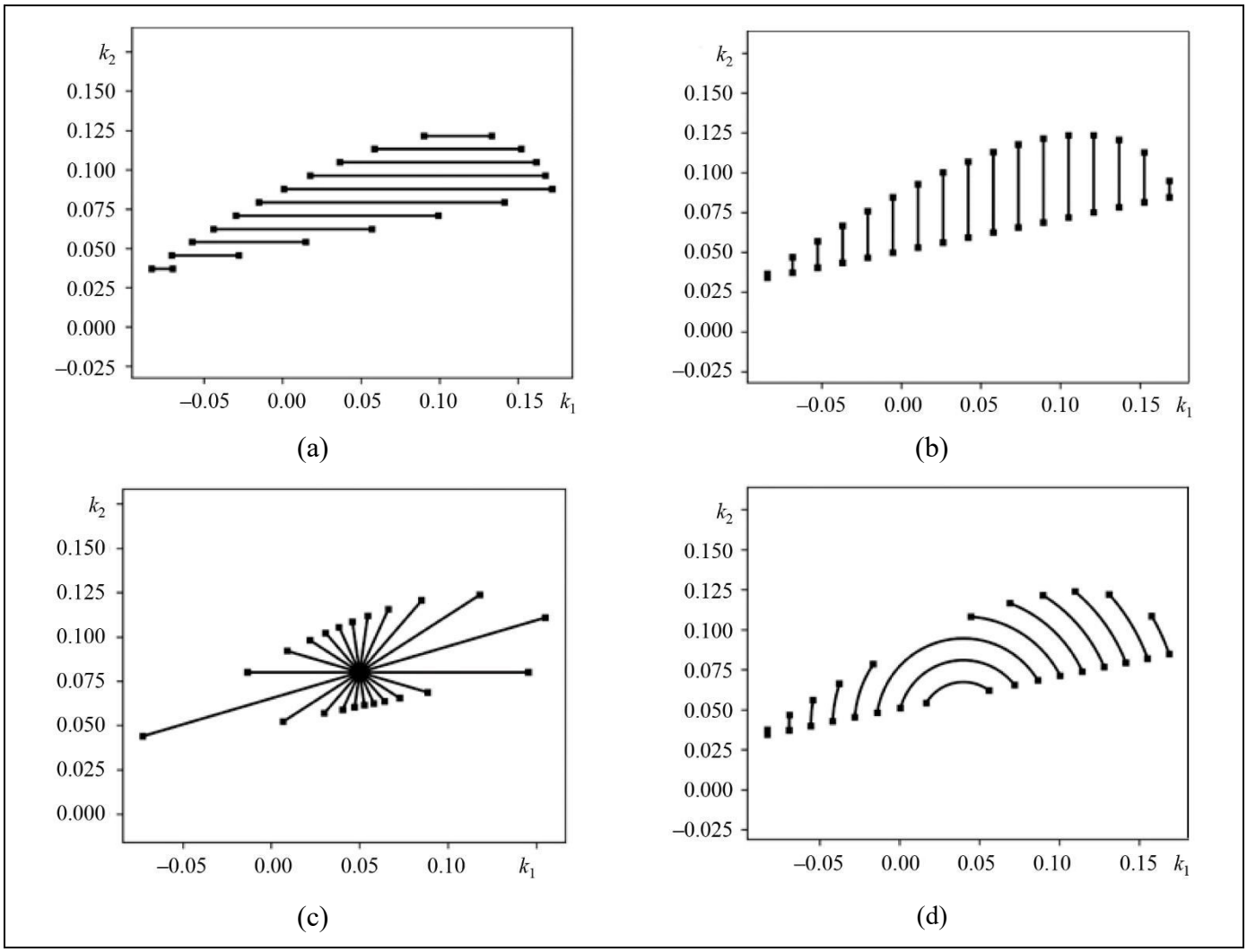

Fig. 2. Internal approximations of the stability region on grids: (a) horizontal lines, (b) vertical lines, (c) concentric circles, and (d) an angular grid.

and for the arcs of the main curve $k_\ell(w)$, $w \in W_\ell$, from equations reducible to polynomial ones with respect to w:

$$\left(k_{\ell,1}(w) - k_{0,1}\right)^2 + \left(k_{\ell,2}(w) - k_{0,2}\right)^2 = r_i^2 . \qquad (28)$$

Fig. 2d presents an example of a radial grid centered outside the set under study.

2.5. The Semi-Grid Method with Parallel (Hyper-)planes

For a polynomial dependent on three parameters (arising, in particular, in the analysis of a closed-loop system with a PID controller), a grid is taken for one parameter. For each fixed value of this parameter, the D-partition is constructed for the remaining two parameters, and the resulting planes are "joined" together. Thus, the construction and visualization of a three-dimensional region is reduced to the construction of a series of two-dimensional stability regions. This classical technique is used to design stabilizing controllers, H_∞-controllers, etc. [3].

In the general case, fixing one parameter reduces the number of free parameters, and it is possible to construct (or approximate) a D-partition for the remaining parameters.

3. GRID METHODS (PARAMETER PLANE PARTITION INTO CELLS)

In semi-grid methods, there is no estimate of how the stability region boundary behaves outside the sets forming the slicing of the parameter space. Only its intersection points with the lines (circles, hyperplanes) that form the slicing boundaries are known. Of course, if the localization region $\boldsymbol{K}$ is defined, the slicing generate a series of bounded sets (segments or circle arcs), but their size is comparable to that of the region $\boldsymbol{K}$.

In grid methods, a finer partition of the parameter space or localization region into sets K_i is used, controlled by *two* grids. Let us emphasize the difference between the method proposed in this paper and the one described in [13]. The common idea is to search for regions for which all polynomials are stable. In this paper, we determine the stability region boundary explicitly using the constructive D-partition and then check its intersection with a *finite* set of the boundaries of the sets K_i. In other words, we explicitly search for the intersection of cell boundaries and the boundaries of the stability region.

Also, we use a cellular grid, particularly formed by a finite set of lines, and explicitly find the cells K_i containing the boundary, based on the intersections of the stability region boundaries and the lines forming the grid. These lines are simultaneously the boundaries of the cells. Clearly, the other cells either entirely belong to some D-partition region, including the stability region, or entirely contain some D-partition region.[3] In this sense, the approach is close to the ideas of adaptive partition of the parameter plane: *quadtree* [14] or the partition by intersecting arcs [11]. The method proposed differs from tracking methods along the boundary of a set using intersecting segments (e.g., orthogonal, simplex, caterpillar, and other methods [2]), as it surely finds all sets covering the boundary.

3.1. An Orthogonal Linear Grid

An obvious combination of one-dimensional grids for each of the parameters k_1, k_2 (subsection 2.2) suggests itself, leading to the analysis of a set of rectangles. This approach is convenient in the sense that it suffices to study the intersection of the D-partition boundary and vertical and horizontal lines (see formulas (28) and (29)), and the intersection points of the grid lines with each other are obvious. And one automatically obtains the rectangular cells K_i containing the stability region boundary; see an example in Fig. 3a.

3.2. A Polar Grid

A similar approach works when combining angular and radial grids. The resulting polar grid consists of concentric circles and radial lines passing through the center of the circles. The radii of the circles are not necessarily uniform, as is the case with the grid at the angles of the lines. If the selected center lies inside the stability region, the polar grid allows finding a simple

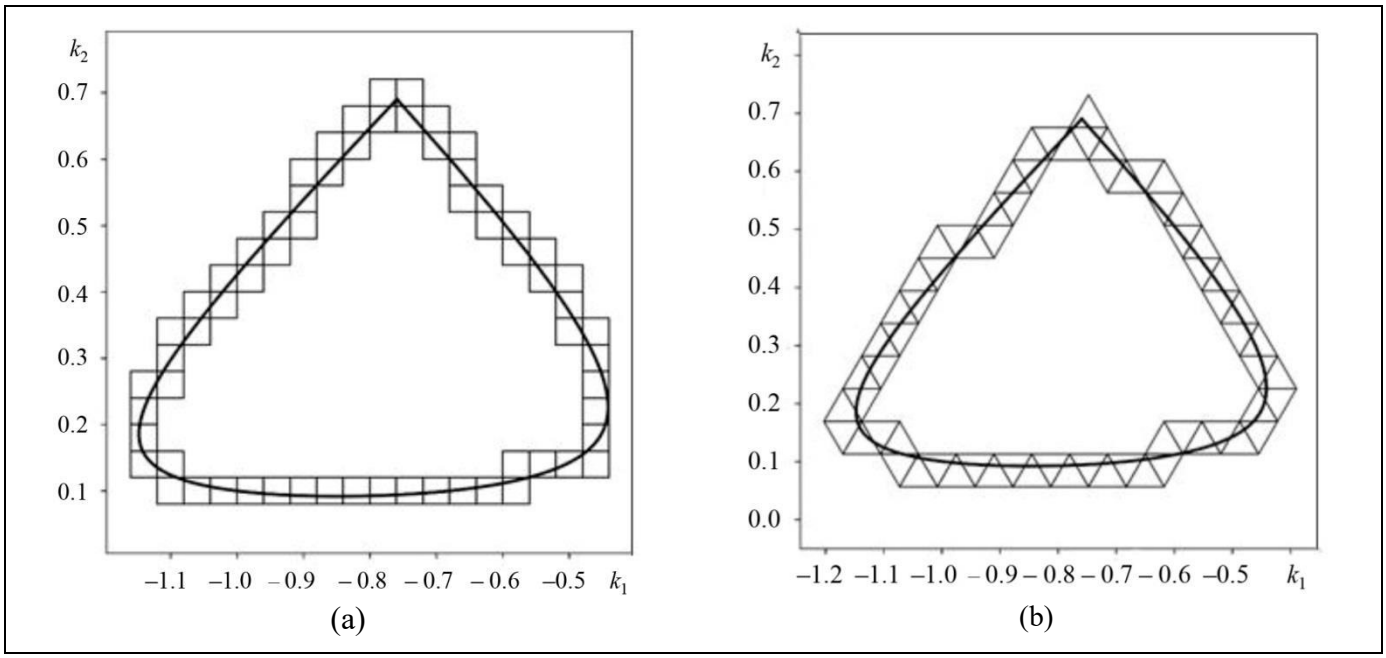

Fig. 3. The boundary of the stability region component in Example 2, approximated by (a) an orthogonal grid and (b) a triangular grid.

[3] In fact, this is the main drawback of the traditional grid approach—it fails to detect explicitly the sets lying inside the cell K_i. This fundamental feature cannot be directly circumvented by reducing the size of cells. However, under the above assumptions, Theorem 1 from part I of the study [5] is valid, and the stable region boundary consists of a finite number of arcs, all explicitly listed. Thus, it is possible to check the localization of each of these arcs (see subsection 4.3) as well as their position relative to the cells K_i.

sectoral internal (and external) approximation of the stability region.

3.3. A Triangular Grid

Here is another type of grids formed by lines—a triangular grid—which requires 1.5 times more computations but yields a "smoother" approximation of the boundary. It is necessary to take a series of horizontal lines and a series of inclined lines; see an example for uniform triangles in Fig. 3b. A triangular grid is convenient because the boundary of the set covering the D-partition boundary has higher "uniformity" and "smoothness." Also, it consists of segments of multiple lengths (with angles of 60°, 120°, and 240°) if the triangles are equilateral.

4. APPLICATION OF CONSTRUCTIVE D-PARTITION

Without loss of generality, the results in this section are also considered relative to the stability region D_n rather than an arbitrary D-partition region D_d.

4.1. Stability Regions on a Curve

Constructive D-partition allows finding stability regions along a curve in the parameter space. Let a curve $\boldsymbol{K}$ on the parameter plane be explicitly defined as

$$F(k_1,\ k_2)=0. \tag{29}$$

Its intersection points with the boundaries of the D-partition regions divide the curve into arcs where the number of stable roots is constant. In particular, to find the "stability arcs" corresponding to stable polynomials, it suffices to consider the intersection of the curve and the stability region boundaries $k_\ell(w)$. These points satisfy the equations

$$F\left(k_{\ell,1}(w),\ k_{\ell,2}(w)\right)=0,\ \ w\in W_\ell, \tag{30}$$

for the intersection with the boundaries representing arcs of the main curve $k_\ell(w)$, and the equations

$$F(d_1t+p_1,\ d_2t+p_2)=0 \tag{31}$$

for the intersection with the boundaries representing singular lines defined in the parametric form $td+p$.

If the curve $\boldsymbol{K}$ is algebraic, i.e., the function $F(k_1, k_2)$ is a polynomial, then the solution of both equations reduces to calculating the roots of a certain polynomial. This technique has been adopted in subsection 2.4 to analyze the intersections of the D-partition boundary and circles using equations (27) and (28) instead of the circle representation (26). Note also that equations (30) and (31) can be treated as boundary equations for the curve $k_\ell(w)$ [15].

Moreover, for the algebraic curves of the form (29), we can obtain the parameterization $\boldsymbol{K}=\{\boldsymbol{k}(v): v\in\mathbb{R}\}$, and from the intersection points with the D-partition boundaries, we can restore the parameters v_i [16]. These points split the curve into arcs with a constant number of roots, and some of the roots correspond to stability arcs. Also, with a known curve parameterization $\boldsymbol{K}=\{\boldsymbol{k}(v): v\in\mathbb{R}\}$, the curve equation can be immediately substituted into the main equation of the D-partition, $G(s, \boldsymbol{k}_1(v), \boldsymbol{k}_2(v))=0_{\mathbb{C}}$, $s\in\partial\boldsymbol{D}$, and then the one-dimensional D-partition with respect to the parameter v can be performed. If the curve parameterization is rational, this partition can be reduced to the one-dimensional D-partition for a polynomial of the form (20).

The above approach serves to analyze polynomials with coefficients nonlinearly dependent on one parameter, e.g., $G(s, t)=f(t)P(s)+g(t)Q(s)+R(s)$. First, the constructive D-partition of the polynomial $G(s, f, g)=fP(s)+gQ(s)+R(s)$ is found with respect to f and g as free parameters. Then, the curve $(f(t), g(t))^{\mathrm{T}}$ is parameterized in the form (29), and equations (30) and (31) are solved. A similar technique can be employed to analyze polynomials with several parameters entering the coefficients nonlinearly.

4.2. The Intersection with Sets of Additional Constraints

The stability region description as a set of boundary arcs obtained by the constructive D-partition makes it easy to consider additional constraints on the coefficients k_1, k_2. Let these constraints be defined by sets $\boldsymbol{K}_i$. In this case, first, the parts of the boundary $\partial\boldsymbol{K}_i$ lying inside the stability region are determined, e.g., as indicated in subsection 4.1. Then the stability region boundaries are updated by taking these (new) boundaries into account and excluding the arcs of the stability region boundary outside the sets $\boldsymbol{K}_i$. This procedure is repeated for all regions of the additional constraints $\boldsymbol{K}_i$. In essence, each time the main curves and singular lines are cut off, by analogy with the localized D-partition.

4.3. Localization of D-Partition Regions on the Parameter Plane and the Support Function

Besides limiting the set of parameters of interest a priori, the term "localization" can be used for the opposite purpose, i.e., identifying the position of the stability region or its components on the plane.

Let a set (in particular, a stability region) be described by a set of boundary arcs $k_i(w),\ w \in W_i = [w_{i,a}, w_{i,b}]$, including both main curve arcs and segments on finite intervals W_i. Then it is not difficult to describe the position of this set. For instance, endpoints can be found using the fact that boundary arcs are defined by rational curves. For example, the minimum and maximum values of the components (i.e., the rectangle containing the boundary arc $k_i(w)$) are determined by the extreme points in the same way as for the speed in subsection 1.1. To this end, we find the roots $w_{i,1,m}$, $m = 1,\ldots$, $w_{i,2,\ell}$, $\ell = 1,\ldots$, of the two equations

$$w_{i,1,m} : k'_{i,1}(w) = 0,\ w \in [w_{i,a}, w_{i,b}],$$

$$w_{i,2,\ell} : k'_{i,2}(w) = 0,\ \ w \in [w_{i,a}, w_{i,b}].$$

As for the speed estimate (16), there are inequalities, and the extreme values are reached at the endpoints of the curve arcs or at the stationary points $w_{i,1,m}$, $w_{i,1,\ell}$:

$$\min\left\{k_{i,1}(w_{i,a}),\ k_{i,1}(w_{i,b}),\ \min_m k_{i,1}(w_{i,1,m})\right\} \le k_{i,1}(w)$$
$$\le \max\left\{k_{i,1}(w_{i,a}),\ k_{i,1}(w_{i,b}),\ \max_m k_{i,1}(w_{i,1,m})\right\},$$

$$\min\left\{k_{i,2}(w_{i,a}), k_{i,2}(w_{i,b}),\ \min_\ell k_{i,2}(w_{i,2,\ell})\right\} \le k_{i,2}(w)$$
$$\le \max\left\{k_{i,2}(w_{i,a}), k_{i,2}(w_{i,b}),\ \max_\ell k_{i,2}(w_{i,2,\ell})\right\}.$$

Extreme points can be replaced by only minima (maxima) of the lower (upper, respectively) bounds.

For segments, the intervals are determined by the endpoints: for example, for k_1,

$$\min\left\{k_{i,1}(w_{i,a}),\ k_{i,1}(w_{i,b})\right\}$$
$$\le k_{i,1}(w) \le \max\left\{k_{i,1}(w_{i,a}),\ k_{i,1}(w_{i,b})\right\},$$

and analogously for the second coordinate $k_{i,2}(w)$.

Fig. 4 shows an example of the localization of individual components of the stability region.

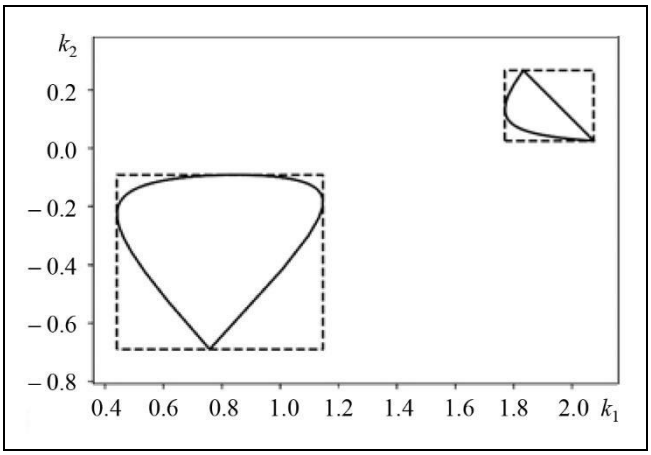

Fig. 4. The localization of two stability region components in Example 2 (see Section 5).

Similarly, it is easy to obtain the support function $\mathrm{supp}_{D_n}(d) = \max_{k \in D_n} d^{\mathrm{T}} k$ and its support element $\mathrm{suppel}(d)$ for the stability region in the direction of the vector $d \in \mathbb{R}^2$. For this purpose, the support function for each arc of the boundary $k_i(w)$ is used:

$$\mathrm{supp}_{D_n}(d) = \max_i \max_{w \in W_i} d_1 k_{i,1}(w) + d_2 k_{i,2}(w),$$

$$\mathrm{suppel}(d) = k_j(w_j),$$

$$w_i = \arg\max_{w \in W_i} d_1 k_{i,1}(w) + d_2 k_{i,2}(w),$$

$$j = \arg\max_i d_1 k_{i,1}(w_i) + d_2 k_{i,2}(w_i).$$

The maximum of a rational function on an interval where its denominator does not vanish is found by calculating the roots of the polynomial in the numerator of its derivative. Moreover, a support function can be built not only for the entire stability region but also for its individual component, by considering only the boundaries of this component.

In the general case, we propose a three-stage approach for the primary analysis of the stability region.

1. Obtain a constructive description of the stability region boundary.

2. Select a sufficiently large localization region $\boldsymbol{K}$ to get finite and closed parameterization intervals W_i, also applying the results from subsection 3.2 on the parameterization change.

3. Localize the stability region by the boundary arcs inside the set $\boldsymbol{K}$.

For the initial localization with a large set $\boldsymbol{K}$, equations (8), (10), and (11) are used for a rectangle, equations (27) and (28) for a circle, etc.

In addition, the stability region can be localized numerically, using a sufficiently uniform grid of fineness ρ. Assume that, according to subsection 1.1, such a grid is obtained by applying Algorithm 1 or 2 to all arcs of the stability region boundary: $K_{grid} = \{k_r,\ r = 1,\ldots\}$. By sequentially connecting the grid nodes, we get a polygon approximating the stability region with an accuracy no worse than ρ (the grid parameter). For this polygon, we can also explicitly express the support function and the support element through its vertices:

$$\mathrm{supp}_{D_n}(d) = \max_r d^{\mathrm{T}} k_r,$$

$$\mathrm{suppel}_{D_n}(d) = k_r,\ \ r = \arg\max_r d^{\mathrm{T}} k_r.$$

The above expressions are approximations to the support function and element, and the stability region itself surely lies inside the rectangle

$$\left[-\rho + \min_r k_{r,1},\ \max_r k_{r,1} + \rho\right] \times \left[-\rho + \min_r k_{r,2},\ \max_r k_{r,2} + \rho\right],$$

where ρ denotes the fineness of a sufficiently uniform grid.

Analogously, for a set of points, it is possible to calculate the minimum covering circle with center k_0 and radius R. A circle with the same center and radius $R + \rho$ will contain the stability region.

4.4. The Distance to the Stability/Instability Region

For a point k_0 characterizing a stable polynomial $G(s, k_{0,1}, k_{0,2})$, the natural problem is to find the distance to the nearest unstable point, i.e., to determine the circle of maximum radius $\|k - k_0\| \le R$ lying entirely in the stability region. This radius is called the stability radius [3].

Given the constructive D-partition, the stability radius can be calculated exactly or estimated using a sufficiently uniform grid on the boundary. The exact solution is determined by a minimization problem considering all stability region parts $k_i(w)$:

$$R = \min_i \min_{w \in W_i} \|k_i(w) - k_0\|.$$

This problem is decomposed into a set of subproblems of minimizing rational functions on the interval $\min_{w \in W_i} \|k_i(w) - k_0{}^2\|$. In turn, each of the subproblems is reduced to calculating the roots of a certain polynomial, similar to the parametric speed (15), and checking the endpoints of the segments.

Analogously, by maximizing the distance from the point k_0 to the boundary points, we can find the minimum circle containing the stability region, see Fig. 5.

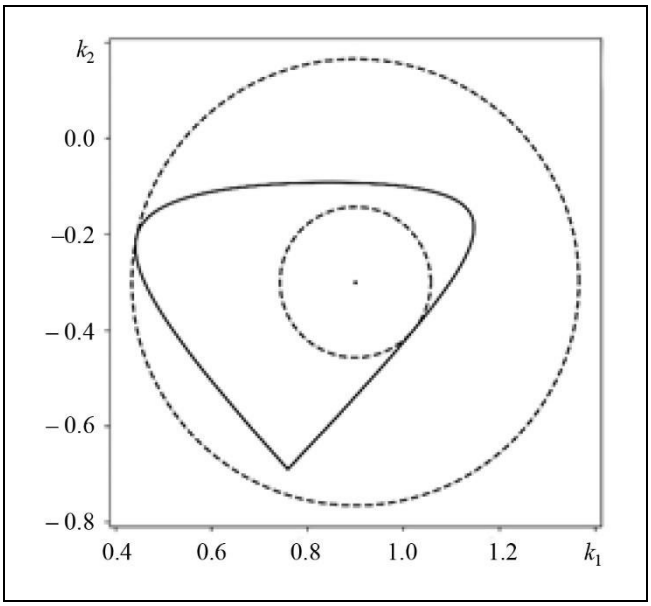

Fig. 5. The circle of maximum radius with a given center contained in the stability region and the circle of minimum radius containing the stability region component, in Example 2 (see Section 5).

The distance to the stability region boundary can also be estimated by the points of a sufficiently uniform grid on the boundary as

$$\min_i \min_r \|k_r - k_0\| - \rho \le R \le \min_i \min_r \|k_r - k_0\| + \rho.$$

Here, k_r are the nodes of the grid covering the stability region boundary. It consists of the union of the grid nodes of each boundary part $k_i(w)$, $w \in W_i$ (without repetitions).

4.5. Application to Robust D-Partition

Robust analysis problems involve polynomials depending on not only "control" but also uncertain parameters, further denoted by q:

$$G(s, k_1, k_2, q) = k_1 P(s) + k_2 Q(s) + R_0(s) + \sum_i q_i R_i(s),\ \ q \in Q. \tag{32}$$

Here, the uncertainty in the polynomial is characterized by a finite set Q. Thus, for each fixed k, a family of polynomials is considered. The polynomial for particular (chosen) parameter values $q = q_0$ is called

the nominal polynomial. In this case, the polynomial (32) takes the form (1) with $R(s)=R_0(s)+\sum_i q_{0,i}R_i(s)$.

The problem is to determine robust stability regions, i.e., such k under which all roots of the polynomial are stable *for all* parameters $q\in Q$. The construction of such regions, albeit being conceptually the same as that of a D-partition, is significantly more complex since the D-partition regions with a constant (for any $q\in Q$) number of stable roots are separated by two sets instead of one-dimensional lines. These sets are defined by the zero exclusion principle, which generalizes equations (3) and (4):

$$K_{bnd}=\{k_1,\,k_2:\,G(s,k_1,k_2,q)=0_{\mathbb{C}},\,s\in\partial\mathbf{D},\,q\in Q\}$$

and

$$K_{deg}=\{G_n(k_1,\,k_2,\,q)=0_{\mathbb{C}},\,q\in Q\}.$$

Accordingly, the boundaries of these two sets are the boundaries of robust D-partition regions. The boundaries of a robust stability region always lie inside the stability region of the nominal polynomial. This follows from the fact that each polynomial of the family (32) is stable, including the nominal one $G(s,\,k_1,\,k_2,\,q_0)$. Thus, when describing the boundaries of the robust stability region, it suffices to consider only those boundaries of the sets K_{bnd} and K_{deg} that lie inside the stability region of the nominal polynomial.

In addition, one can repeat the constructive D-partition many times for the polynomials $G(s,\,k_1,\,k_2,\,q_r)$, $r=1,\,2,\ldots,$ where each polynomial is defined by randomly selected parameters $q_r\in Q$. The robust stability region lies in the intersection of the stability regions of all selected polynomials.

Let the boundary $\partial\mathbf{D}$ of a root localization region have a parameterization $s(w)$, and let the boundary of the nominal polynomial be parameterized by a set of curves and lines $k_i(w)$, $w\in W_i$. As it turns out, the boundary of robust D-partition regions is characterized similarly: K_{bnd} is the envelope of the family of sets $K_{bnd}(w)=\{k_1,\,k_2:\,G(s(w),k_1,k_2,q)=0_{\mathbb{C}},\;q\in Q,\;s(w)\in\Gamma\}$ [3, 17].

One might expect that the intervals of the sets $K_{bnd}(w)$ generating the boundary of the robust stability region (with respect to w) are within the intervals W_i of the stability region boundaries without uncertain parameters. Unfortunately, in the general case, the hypothesis of nested intervals is incorrect, since the boundary of the set $K_{bnd}(w)$ with $w\notin W_i$ can affect the final robust stability region. This depends on the values of the main curve function $k(w)$ for the nominal D-partition and the *size* of the set Q. The analysis and design of the constructive robust D-partition are an open research problem even for the case of parameters entering the characteristic polynomial linearly.

5. EXAMPLES

The figures use stability regions or their components from two examples discussed in detail in part I of the study [5].

Example 1 [18 , *p. 77*]. Consider a closed-loop continuous-time system with the characteristic polynomial

$$G(s,\,k_1,\,k_2)=k_1s(s-1)(s-2)+k_2(s-1)(s-2)+s(s+1)(s^2+s+1). \tag{33}$$

It is required to analyze its stability with respect to the root localization region $\boldsymbol{D}=\{s:\mathrm{Re}\,s<0.2\}$ with the boundary parameterization $s(w)=-0.2+jw$. Since the polynomial (33) has real coefficients, it suffices to take the upper part of the boundary, $W=[0,\,\infty)$.

There is one main curve, $w\in[0,\,\infty)$, with the components

$$k_1(w)=\frac{4.6\,w^4-7.112\,w^2+0.62016}{-w^4-6.28\,w^2-6.9696},$$

$$k_2(w)=\frac{-w^6+8.68\,w^4-5.4208\,w^2-0.230784}{-w^4-6.28\,w^2-6.9696}.$$

The critical frequency $w_0=0$ is associated with the unique singular line $-0.528k_1+2.64k_2-0.1344=0$, with the parameterization $p+td$, where $p=(-0.00979021;0.04895105)^{\mathrm{T}}$ and $d=(-2.64;-0.528)^{\mathrm{T}}$. The stability region is bounded by one arc of the main curve $k(w)$, $w\in[0,\,0.70951628]$, and the segment $p+td$, $t\in[-0.06916025,\,0.02999640]$ (Fig. 6).

Figure 2 shows a series of internal approximations of the stability region by one-dimensional lines. Next, Fig. 3a presents the parametric speed $v(w)$ of the curve arc specified and the sufficiently uniform parameter values yielded by Algorithm 1 for $\rho=0.02$; Fig. 3b, the nodes and their neighborhoods of radius ρ covering this curve arc.

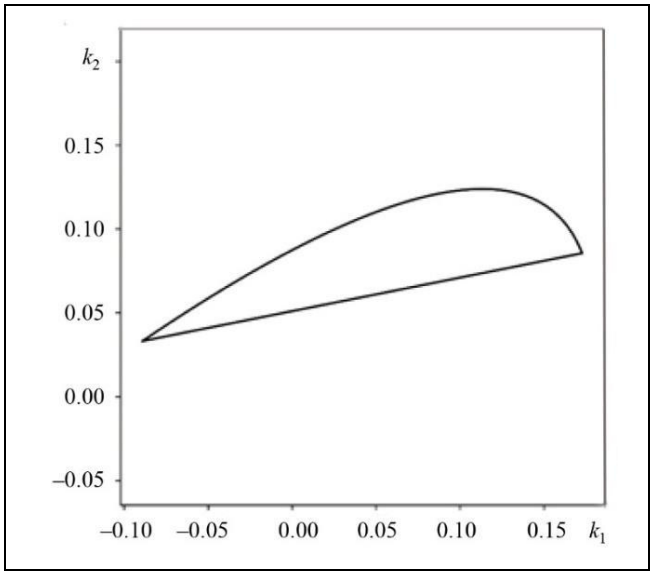

Fig. 6. The stability region in Example 1.

Example 2 [19]. Consider the characteristic polynomial $G_0(z,\, k_1,\, k_2) = z^n + k_1 z^{n-1} + (1+\varepsilon) z^{n-2} + k_2$, $n = 5$, $\varepsilon = 0.1$, of a discrete-time system. Its stability is equivalent to the Hurwitz property of the polynomial

$$G(s,\, k_1,\, k_2) = (s+1)^5 + (1+\varepsilon)(s-1)^2 (s+1)^3 + k_1 (s-1)(s+1)^4 + k_2 (s-1)^5.$$

The boundary of the D-partition regions consists of the one main curve

$$k_1(w) = \frac{-16.6\, w^8 + 128.8\, w^6 - 221.2\, w^4 + 128.8\, w^2 - 16.6}{8\left(w^8 - 6\, w^6 + 6\, w^2 - 1\right)},$$

$$k_2(w) = \frac{-0.2\, w^8 - 0.8\, w^6 - 1.2\, w^4 - 0.8\, w^2 - 0.2}{8\left(w^8 - 6\, w^6 + 6\, w^2 - 1\right)}$$

and two singular lines. One of them corresponds to $w = 0$ and has the parameterization $td + p$, where $p = (1.05;\, 1.05)$ and $d = (1, -1)$. The other singular line is defined by the degree drop condition (4) with the parameterization $td + p$, where $p = (-1.05;\, -1.05)$ and $d = (-1;\, 1)$.

The stability region consists of four components (Fig. 7):

1) the segment of the first singular line for $t \in [1.025, 1.45]$ and the arc of the main curve for $w \in [0,\, 0.37796447]$;

2) the arc of the main curve for $w \in [0.42972375,\, 0.96431209]$;

3) the arc of the main curve for $w \in [1.03700867,\, 2.32707640]$;

4) the segment of the second singular line for $t \in [1.025,\, 1.45]$ and the arc of the main curve for $w \in [2.64575131,\ \infty)$.

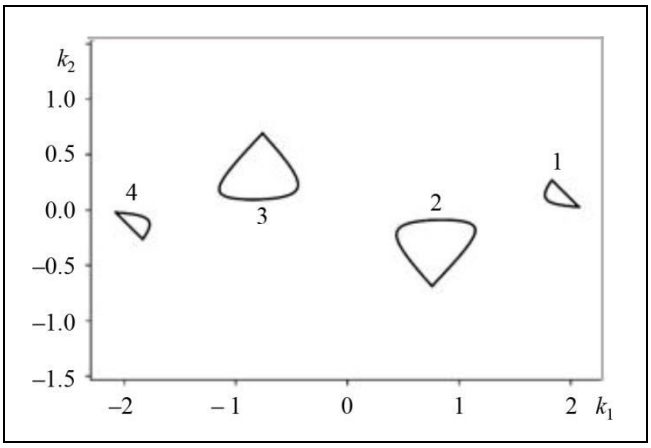

Fig. 7. The stability region components in Example 2.

The last arc of the curve can be written as $k_u(u) = k_\ell(1/u)$, $u \in [0,\, 0.37796447]$ (see subsection 3.2), where the value at $u = 0$ is defined and coincides with $k_{(\infty)}$:

$$k_{u,1}(u) = \frac{16.6\, u^8 - 128.8\, u^6 + 221.2\, u^4 - 128.8\, u^2 + 16.6}{8\left(u^8 - 6\, u^6 + 6\, u^2 - 1\right)},$$

$$k_{u,2}(u) = \frac{0.2\, u^8 + 0.8\, u^6 + 1.2\, u^4 + 0.8\, u^2 + 0.2}{8\left(u^8 - 6\, u^6 + 6\, u^2 - 1\right)}.$$

Note that the resulting parameterization coincides with the original one in w up to the sign, and the interval coincides with the interval of the first component. This is due to the symmetry of the original root localization region of the discrete system (the unit circle) and its parameterization.

Figure 3 shows the approximations of the second component of the stability region using orthogonal and triangular grids. In addition, Fig. 4 presents the bounding regions $[1.7706,\, 2.075] \times [0.025,\, 0.2667]$ for the first component and $[0.4414,\, 1.1472] \times [-0.6901,\, -0.09185]$ for the second component of the stability region; the third and fourth components are symmetric to the second and first, respectively. Finally, Fig. 5 demonstrates two circles with centers $k_1 = 0.9$, $k_2 = -0.3$: the smaller lies entirely within the stability region and determines a stability radius of 0.4417 for the above controller; the larger circle with radius 0.9538 contains the entire second component of the stability region.

CONCLUSIONS

Based on the constructive D-partition method (see part I of the study [5]), two algorithms have been pro-

posed to build a sufficiently uniform grid on a stability region boundary with a given fineness. The algorithms involve an estimate of the parametric speed of rational curves. Several semi-grid internal approximations of a stability region have been proposed, together with a regular covering method for the boundary of a stability region with rectangles and triangles. The above methods have been applied in stability analysis problems, including the localization of stability region components on a plane, the construction of their support functions and their approximations, as well as the calculation of the stability radius.

Acknowledgments. *The research presented in Sections 2 and 3 was supported by the Russian Science Foundation, project no. 21-71-30005-П, https://rscf.ru/project/21-71-30005/. The author is grateful to the anonymous reviewers and the editorial proofreader for careful reading of the manuscript and helpful remarks.*

REFERENCES

1. Kushel, O.Y., Unifying Matrix Stability Concepts with a View to Applications, *SIAM Review*, 2019, vol. 61, no. 4, pp. 643–729.
2. Diduk, G.A., *Mashinnye metody issledovaniya avtomaticheskikh sistem* (Machine Methods for Studying Automatic Systems), Leningrad: Energoatomizdat, 1983. (In Russian.)
3. Ackermann, J., *Robust Control: The Parameter Space Approach*, 2nd ed., London: Springer, 2002.
4. Violet, G., Continuity Argument Revisited: Geometry of Root Clustering via Symmetric Products, *arXiv:1512.08645*, 2016, pp. 1–45. DOI: https://doi.org/10.48550/arXiv.1512.08645
5. Tremba, A.A., Constructive D-Partition for Two Parameters Entering Polynomial Linearly: Part I. Description of the Boundaries of the D-Partition Regions, *Control Sciences*, 2025, no. 6, pp. 33–49.
6. Gryazina, E.N., Polyak, B.T., and Tremba, A.A., D-Partition Technique State-Of-The-Art, *Automation and Remote Control*, 2008, vol. 69, no. 12, pp. 1991–2026.
7. Neimark, Yu.I., On the Determining Parameter Values for Which an Automatic Control System Is Stable, *Avtomatika i Telemekhanika*, 1948, vol. 9, no. 3, pp. 190–203. (In Russian.)
8. Neimark, Yu.I., *Dinamicheskie sistemy i upravlyaemye protsessy* (Dynamic Systems and Controllable Processes), Moscow: Nauka, 1978. (In Russian.)
9. Gryazina, E.N. and Polyak, B.T., Stability Regions in the Parameter Space: D-Partition Revisited, *Automatica*, 2006, vol. 42, no. 1, pp. 13–26.
10. Šiljak, D.D., Analysis and Synthesis of Feedback Control Systems in the Parameter Plane I–Linear Continuous Systems, *IEEE Transactions on Applications and Industry*, 1964, vol. 83, no. 75, pp. 449–458.
11. Tan, L., Li, B., Zhang, B., and Cheng, J.S., An Algorithm for the Intersection Problem of Planar Parametric Curves, in *Lecture Notes in Computer Science*, Cham: Springer, 2023, vol. 14139, pp. 312–329. (Proceedings of International Workshop on Computer Algebra in Scientific Computing.)
12. Polyak, B.T. and Gryazina, E.N., Randomized Methods Based on New Monte Carlo Schemes for Control and Optimization, *Annals of Operations Research*, 2011, vol. 189, pp. 343–356.
13. Pryashnikova, P.F., D-Partition in the Case of Polynomial Dependence of the Coefficients of a Polynomial on Two Parameters, *Automation and Remote Control*, 2021, vol. 82, no. 3, pp. 398–409.
14. Farouki, R.T., Reduced Difference Polynomials and Self-Intersection Computations, *Applied Mathematics and Computation*, 2018, vol. 324, pp. 174–190.
15. Saydy, L., Tits, A.L., and Abed, E.H., Guardian Maps and the Generalized Stability of Parametrized Families of Matrices and Polynomials, *Mathematics of Control, Signals and Systems*, 1990, vol. 3, no. 4, pp. 345–371.
16. Goldman, R.N., Sederberg, T.W., and Anderson, D.C., Vector Elimination: A Technique for the Implicitization, Inversion, and Intersection of Planar Parametric Rational Polynomial Curves, *Computer Aided Geometric Design*, 1984, vol. 1, no. 4, pp. 327–356.
17. Tremba, A.A., Robust D-Partition under l_p-Bounded Parametric Uncertainties, *Automation and Remote Control*, 2006, vol. 67, no. 12, pp. 1878–1892.
18. Francis, B., *A Course in H_∞ Control Theory*, Lecture Notes in Control and Information Sciences, Berlin: Springer-Verlag, 1987, vol. 88.
19. Gryazina, E.N., The D-Partition Theory, *Automation and Remote Control*, 2004, vol. 65, no. 12, pp. 1872–1884.

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

Received August 6, 2025, and revised October 8, 2025. Accepted October 23, 2025.

Author information

Tremba, Andrey Aleksandrovich. Cand. Sci. (Phys.–Math.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia; Moscow Institute of Physics and Technology, Dolgoprudny, Russia
✉ atremba@ipu.ru
ORCID ID: https://orcid.org/0000-0001-5783-7600

Cite this paper

Tremba, A.A., Constructive D-Partition for Two Parameters Entering a Polynomial Linearly. Part II: Approximation of Stability Regions and Robustness Analysis. *Control Sciences* **1,** 2–18 (2026).

Original Russian Text © Tremba, A.A., 2026, published in *Problemy Upravleniya*, 2026, no. 1, pp. 3–21.

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 VICTORY FUNCTION AND ITS APPLICATION IN CONFLICT MODELING

V. V. Shumov

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

✉ v.v.shumov@yandex.ru

Abstract. The victory function (VF) in combat and special operations is a separate class of contest success functions (CSFs) describing the probability of a participant's success in a competition or conflict. This paper briefly characterizes aggregate functions, which include production functions, utility functions, CSFs, power (might) indices of countries, etc. Substantive (postulates) and formal (axioms, properties) requirements for the VF are given. Probabilistic (based on A.N. Kolmogorov's law of target destruction and the Weibull distribution) and substantive (based on the framework of military science and practice) justifications of the VF are presented. Economic models of conflict and appropriation are overviewed. Using the security model and the VF, three problems of distributing a disputed resource (territory and population) between countries are formulated. A promising line of further research is to develop conflict theory at several levels, namely, in the theater of military operations, the intercountry, and geopolitical levels.

Keywords: mathematical model, security, combat and special operations, victory function, postulates of conflict technology, models of conflict and appropriation, military cybernetics.

INTRODUCTION

Aggregate functions are used in many fields. They represent mathematical relationships between the expected result (opportunities) and several factors (efforts, resources, technologies, etc.). An aggregate function maps a vector of quantitative indicators (resource types) to a real number. This number characterizes the potential or latent capability of a system rather than the result observed.

In computer science and statistics, aggregation conventionally refers to operations like summation, averaging, or computing extremal values (minimum, maximum, median, etc.). In systems analysis, aggregation—combining several elements into a single whole—is closely related to the emergence property (the presence of some properties in a system that are not inherent in its individual components) [1] and uncertainty (the category of uncertainty may be sufficient to consider the factors of complexity and emergent behavior [2]).

Among the most extensively studied aggregate functions we note production functions [3], utility functions [4–6], contest success functions (CSFs, see the review [7]), the indices (models) of geopolitical power and might of countries (see the paper [8] and the comprehensive review in the monograph [9]), etc. Despite criticism of aggregate production functions (the so-called "Cambridge debates" [10, 11]), they remain the foundation of applied statistical studies and economic growth theories.

In the 1950s, the scope of traditional economics was expanded: in addition to production and trade, researchers began considering the problems of appropriation (seizure of foreign products or protection of domestic ones) [12]. In conflict theory, CSFs, which describe the probability of a participant's success in a competition (conflict), are analogous to production functions in production theory and utility functions in consumption theory. Conflict was usually considered to be the result of incomplete and asymmetric information or even the result of irrationality. It was shown that military operations change the strategic positions of opponents in the long term. In this case, the party (to conflict) considering short- and long-term effects may choose war, and this choice will correspond to rational and far-sighted behavior based on complete information. If one or more parties consider the future to be sufficiently important, the outbreak of war should be expected [7].

The objective of this paper is to study the victory function (VF) in combat and special operations as a distinct subclass of CSFs and to formulate problems related to its application in conflict and appropriation models.

1. REQUIREMENTS FOR CONTEST SUCCESS FUNCTIONS

Let us categorize the requirements for the aggregate functions of contest success (victory in combat and special operations) as *substantive* (implied by the postulates of military science and systems analysis) and *formal* (axiomatic requirements).

1.1. Substantive Requirements

Figure 1 shows a selection of works that have significantly influenced the development of military science and its practical application.

J. Bernoulli's treatise on the art of conjectures [13] was the first systematic book of probability theory. It presented combinatorics, binomial distribution, and the first version of the law of large numbers.

The "classical stage" in the development of probability theory was completed by P. Laplace's work on the analytical theory of probability [14]. The author examined discrete and continuous random variables[1], introduced the concepts of probability density and characteristic function, gave the formula for total probability, and proved the convergence of the binomial distribution to the normal distribution (the Moivre–Laplace theorem) as the number of trials tends to infinity. As often been the case in later periods (e.g., the 20th century, with the successes of cybernetics and disappointment in it [15]), the number of works on probability theory continued to grow in the 19th century, and attempts were undertaken to extend its methods far beyond reasonable limits: to ethics, psychology, law, theology, etc., compromising the science.

C. Clausewitz undoubtedly belongs to the classics of European and world science of the modern era. His fundamental *Von Krieg* [16] had a profound influence on the development of military science, being still relevant nowadays. This study does not aim to provide a comprehensive exposition of Clausewitz's intellectual legacy—a task better suited to dedicated military historians and practitioners. Instead, following the established scholarly tradition [17, 18], we will comparatively analyze the perspectives on the role of uncertainty in war and combat as held by C. Clausewitz and L. Tolstoy [19].

Adhering to the approach described in [20, 2], we define the *uncertainty* of military (combat) operations as the possibility of certain events accompanying these operations that affect their implementation and result, but may/may not occur. Due to the uncertainty of military operations, it is impossible to predict a priori the characteristics of its result, as well as the time and effort (resources) to achieve the result.

The *measurable uncertainty* of military operations is defined as the possibility of events described by certain regularities that may/may not occur. Such events can be analyzed using quantitative methods (e.g., probabilistic and statistical) based on previous measurements or fundamental laws (together with the assumption of invariable conditions and regularities).

The *true uncertainty* of military operations is the possibility of unique (or rarely recurring) events that cannot be explained by known regularities.

The fundamental distinction between true uncertainty and measurable uncertainty is that the former's events occur due to unknown factors (a frequent and important, albeit special, case is the active choice of an individual), whereas the latter's events, although unpredictable, are described by known regularities [2].

As famously observed by Clausewitz [16], war is a realm of uncertainty; three-quarters of war action's foundation lies in the fog of the unknown. His way of eliminating uncertainty is a subtle, flexible, and penetrating mind. The unreliability of information and assumptions leads to the fact that those fighting in reality face a completely different situation than they expected. Many reports received during war contradict each other; there are even more false reports, and the vast majority of them are unreliable. Another recipe to eliminate uncertainty at the strategic level is to provide a decisive numerical superiority (one and a half to two times) over the enemy in terms of forces and means and take the moral factor into account. Uncertainty decreases when passing from the tactical to strategic level. The third approach to deal with uncertainty is to have reserves. Reserves are a means to counter

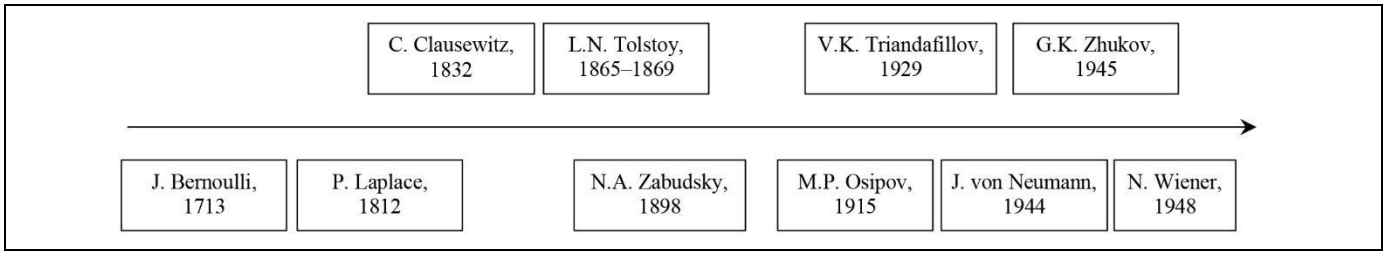

Fig. 1. Key influential works of military science and practice.

[1] The concepts of a "random variable" and "distribution function" appeared in the 20th century in the works of the Russian probability school.

the enemy's unforeseen actions and, moreover, to correct the unpredictable outcome of a battle under an unfavorable turn of events.

The moral factor decisively influences on the outcome of a battle. At the tactical level, morale gradually recovers. The moral influence of victory grows exponentially depending on the scale of military operations. This may also indicate a lesser degree of uncertainty at the strategic level. Another condition determining the moral weight of victory is the numerical ratio of the forces fighting each other. Defeating the enemy with small forces is evidence of overall superiority over it. However, in reality, the actual ratio of forces at the tactical level is usually unknown (with rare exceptions when the battlefield is clearly defined), and at the strategic level, it becomes known many years later. Therefore, such information about the moral ratio of the parties has no effect on current events.

In conclusion, Clausewitz argued that the bravery and spirit of armed forces have always increased their physical strength, and this will continue to be the case. However, in some periods in history, a sharp superiority in the organization and armament of forces gave a significant moral advantage; in other periods, superiority was provided by the mobility of forces[1]; later on, newly introduced tactical systems had an impact, then the art of war was dominated by the desire to use the terrain skillfully, etc. Nowadays, armies have become so similar in terms of weapons, equipment, and training that there is no noticeable difference between the best and worst of them in this respect. The degree of the development of scientific forces still significantly differs between armies, but this difference only makes some armies the initiators and inventors of various improvements and the others their quick imitators. As the above factors tend to equilibrium, Clausewitz concluded, the numerical preponderance of forces regains its status as the most decisive element in strategic calculations.

Tolstoy considered wars from a broader perspective (philosophical, civilizational, political, cultural, ethical, etc.) and highlighted the true uncertainty of military activity: "Napoleon, who seems to us to have been the leader of all this movement (as the figurehead on the prow of a ship may seem to a savage to guide the vessel), Napoleon during all that time of his activity was like a child who, holding on to the straps inside a carriage, imagines that he is driving it." [19]. Countless free forces (for nowhere is a person freer than in a battle, where it is a matter of life and death) influence the direction of the battle, and this direction can never be known in advance and never coincides with the direction of any force. Historians write after the fact: *chance* created the situation, and *genius* took advantage of it. But the words "chance" and "genius" mean nothing that really exists and therefore cannot be defined. The officers—the heroes of Tolstoy's novel—wander across the battlefield, blinded by gun smoke, without the slightest idea of what is happening. The author explained the true uncertainty of war by the fact that the human mind cannot comprehend the causes of events in their entirety, but the desire to find them is inherent in the human soul.

Today, Tolstoy's ideas are particularly relevant and significant for military experts and researchers, particularly in connection with the adoption of artificial intelligence (AI) systems. Undoubtedly, AI will be used to solve simple and typical tasks (machine vision, sensor data processing in near real time, etc.). According to T. Lipsky (the United States Military Academy, West Point) , given its interest in adopting AI, the army risks forgetting the importance of keeping the boundary between man and machine, not only for ethical reasons but also because, as Tolstoy showed[2], a commander cannot obey algorithms and templates in a battle. Command remains an "artistic and psychological" task [18]. AI is ill-suited to rapidly changing situations, where actions depend largely on context and require human judgment. This is how any war can be characterized. Therefore, it is important to include healthy skepticism about AI in army command instructions. Military personnel must learn to make plans and manage their implementation without external assistance before incorporating AI into the process, and they must undergo regular recertification in this area throughout their military careers [18]. As shown in the paper [21], machine learning does not lead to general AI, and the myth of AI weakens belief in human potential. O.P. Kuznetsov noted that understanding is interpretation in terms of a person's worldview; our brain constructs a worldview, and it is structured through the categorization of human experience; meanings (senses) are formed earlier than conceptual structures; meanings are based on biological and social

[1] Superiority in mobility was observed in the 1930s–1950s (the transition of armies to continuous motorization of forces, the creation of motorized rifle and mechanized units and formations): motorized groups (divisions) were created for deep operations and deep encirclement of enemy's infantry units and formations.

[2] In *War and Peace*, the role of M.I. Kutuzov was described as follows: "By long years of military experience he knew, and with the wisdom of age understood, that it is impossible for one man to direct hundreds of thousands of others struggling with death, and he knew that the result of a battle is decided not by the orders of a commander in chief, nor the place where the troops are stationed, nor by the number of cannon or of slaughtered men, but by that intangible force called the spirit of the army, and he watched this force and guided it in as far as that was in his power." [19].

goals; cognitive processes involve not only the brain but also the body, and understanding is connected with actions in the environment, and knowledge of the latter is contained in the worldview.

Starting from the second half of the 19th century, in Russia and European countries, the achievements of natural science and probability theory were actively adopted in the theory and practice of artillery and firing control [22]. During World War I, the first works on modeling military and combat operations were published (the Osipov–Lanchester models [23, 24]).

Based on military statistics on the largest battles of regular armies, M.P. Osipov developed the so-called quadratic model of a battle, laid the foundations of the theory of combat potential, and quantitatively assessed the role of the moral factor and the art of command. (According to his approach, victory depends not on the duration of a battle but mainly on the losses suffered by the parties; therefore, it would be more accurate to assume that a battle lasts until the losses of one party reach a certain percentage. This percentage can be considered to be 20% on average [23].)

V. K. Triandafilov's book on the nature of operations of modern armies is noteworthy in two respects. First, the author developed a new branch of military art, i.e., operational art (in addition to tactics and strategy); second, the book gives every reason to consider Triandafilov the founder of systems analysis. In the preface to the first edition of the book, the author outlined the methodology of his research as follows. First, the material basis of military operations is examined—the armament of forces, their number, organization, and other important data about the situation affecting the nature of military operations. Then, based on all the data above, the issues of modern tactics, individual operations, and a series of consecutive operations are examined. All postulates are expressed in terms of particular numbers as well as tactical and operational norms. This is the only way to show the difference between the present and the past and the direction in which military science is evolving. Moreover, all the numerical data are *indicative*; norms may change for each particular situation. The task of a commander (practical leader) is to determine the nature and extent of such changes [25, *pp. 13, 14*]. Triandafilov's contribution to military science and systems analysis (as a branch of cybernetics) was analyzed in the monograph [9].

In the 1930s–1940s, there was a rare case in history when both game theory (a branch of operations research) and military practice solved the same problem in parallel: increasing efficiency by eliminating patterns (using the so-called "mixed strategies"). In 1944, the seminal work *Game Theory and Economic Behavior* was published [5]. The cornerstone of modern non-cooperative game theory is the concept of Nash equilibrium, which exists for all finite games [26]. Game-theoretic problem statements were reviewed in [9].

In the legacy of outstanding Soviet commander G.K. Zhukov, we mention only two results (see the publication [27]). First, Zhukov identified the same factors affecting the success of any battle (combat and military operation), i.e., those valid for all levels: tactical, operational, and strategic (operational-strategic). Second, starting from 1943, he utilized the ideas of "mixed strategies" when planning strategic offensive operations (breaking through the enemy's prepared defense).

Operations research, as an applied mathematical discipline, appeared during World War II. Its purpose was defined as providing commanders with quantitative grounds for decision-making. Subsequently, this discipline became a branch of cybernetics [28], which is now understood as the science of organizing and managing systems [15]. Today, the network-centric approach is extremely fashionable and productive. It includes the principles of organizing and analyzing any networks in general and, in particular, those "assembled" for a combat mission, at the right time and in the right place [15].

Based on the above considerations, we draw the following conclusions (substantive requirements for the VF).

- *Zhukov's postulate:* the VF shall have the same form for describing military operations at all levels (tactical, operational, and operational-strategic), including special operations (guerrilla, anti-guerrilla, sabotage, reconnaissance and diversion, counterterrorism, etc.);
- *Clausewitz –Tolstoy's postulate:* the VF shall reflect both true and measurable uncertainty;
- *Osipov–Triandafilov's postulate:* the VF shall consider the number of armed forces of each party, their morale and technological characteristics, situational specifics, and the existing and prospective weapon systems.

1.2. Formal Requirements

Historically, the first VF stems from Osipov's battle model. In the absence of operational losses and reserves, a typical battle is described by the system of differential equations

$$\frac{dx(t)}{dt} = -a_y y(t), \quad \frac{dy(t)}{dt} = -a_x x(t),$$

where $x(t)$ and $y(t)$ denote the number of armed forces of the first and second parties, respectively, at a time instant t; a_x and a_y are their striking efficiencies. From

the equal forces condition $y_0 = x_0\sqrt{a_x / a_y}$, we obtain Osipov's indicator VF (the probability of the first party's victory):

$$p_x(x,\, y) = \begin{cases} 1, & x\sqrt{a_x} > y\sqrt{a_y}, \\ 0.5, & x\sqrt{a_x} = y\sqrt{a_y}, \\ 0, & x\sqrt{a_x} < y\sqrt{a_y}. \end{cases} \quad (1)$$

One model of a guerrilla war [29] has the form

$$\frac{dx(t)}{dt} = -a_y y(t),\ \frac{dy(t)}{dt} = -a_x x(t)\frac{y(t)}{y_0},$$

where $x(t)$ is the number of regular armed forces, and $y(t)$ is the number of guerrillas. From the equal forces condition $y_0\sqrt{2a_y} = x_0\sqrt{a_x}$, we obtain S. Deitchman's VF

$$p_x(x,\, y) = \begin{cases} 1, & x\sqrt{a_x} > y\sqrt{2a_y}, \\ 0.5, & x\sqrt{a_x} = y\sqrt{2a_y}, \\ 0, & x\sqrt{a_x} < y\sqrt{2a_y}. \end{cases}$$

(As a matter of fact, it follows from the guerrilla war model [29].)

In other words, *ceteris paribus*, parity is achieved if the number of regular armed forces exceeds, by $\sqrt{2} \approx 1.4$ times, that of guerrillas (cf. the expression (1)).

The following class of VFs (those of success in a competition or auction) has been fairly well investigated by now:

$$p_x(x,\, y) = \frac{f_x(x)}{f_x(x) + f_y(y)}, \quad (2)$$

where $f_x(\cdot)$ and $f_y(\cdot)$ are nonnegative strictly increasing functions. Here are some of the most common functional forms of model (2) [7, 9].

G. Tullock's model

$$p_x(x,\, y) = \frac{x^\mu}{x^\mu + y^\mu} = \frac{(x/y)^\mu}{1 + (x/y)^\mu}, \quad (3)$$

where $0 < \mu \le 1$ is the determination parameter of the parties, belongs to the class of models based on the ratio of the forces (power) of the parties involved.

The model proposed by D. McFadden and J. Hirshleifer,

$$p_x(x,\, y) = \frac{e^{\mu x}}{e^{\mu x} + e^{\mu y}} = \frac{1}{1 + e^{\mu(x-y)}}, \quad (4)$$

belongs to the class of models based on the difference in the forces of the parties. The probit model $p_x(x,\, y) = \Phi(x - y)$, where Φ indicates the Laplace function, is another representative of this class.

The probabilistic justification of conflict functions proceeds from an analysis of the influence of neglected factors (random errors) on the result. In the general case, regression functions are of the form $Y_x = h(x, \varepsilon_x)$ and $Y_y = h(y, \varepsilon_y)$, where the error functions ε_x and ε_y have zero mean. Then the probability of the first party's victory in a conflict is given by

$$p_x(x,\, y) = P(Y_x > Y_y) = P\left(h(x, \varepsilon_x) > h(y, \varepsilon_y)\right).$$

Conflict functions have been axiomatized, in particular, by R. Luce [30] and S. Skaperdas [31]. The axiomatic system is based on the *Independence of Irrelevant Alternatives*: in the context of conflict, this property requires that the outcome of a conflict between any two parties depends only on the amount of armaments possessed by them, and not on the amount of armaments possessed by third parties. The next important requirement for conflict functions is *zero-degree homogeneity*, i.e., $p_x(tx, ty) = p_x(x, y)$ for all $t > 0$. Models (3) and (4) have *symmetry or anonymity*: if the efforts of the parties interchange places, the probabilities of their victory will also do so.

Skaperdas et al. noted that despite the rich literature on the modeling of conflicts, contests, and auctions, only a small number of publications have addressed the verification of conflict functions based on real data [32].

2. PROBABILISTIC AND SUBSTANTIVE JUSTIFICATION OF THE VICTORY FUNCTION

2.1. Probabilistic Justification of the Victory Function

In 1945, A.N. Kolmogorov proposed a firing efficiency criterion based on the law of target destruction, i.e., the probability of destroying a single or group target (the expected number of targets destroyed) depending on the number of shots fired at this target. The probability of destroying a target (event A) with x hits is given by [33]

$$P(A \mid x) = 1 - e^{-\alpha x}, \quad (5)$$

where $\alpha > 0$ is a parameter. The expression (5) describes the exponential distribution, a special case of the Weibull distribution. It has a wide range of applications (the reliability of technical systems, queuing systems, etc.) and is closely related to the concept of a Poisson process. For such a process, the intervals between successive events are independent random variables with the exponential distribution, and α is the mean number of events per unit time.

Let the random variables X and Y be the numbers of hits required to destroy the enemy's targets by the first and second parties, respectively. Using the Weibull distribution, we find the probabilities of successful task completion (destroying the enemy's targets) by the parties:

$$F_x(x) = 1 - e^{-(\alpha_x x)^m},\ \alpha_x = \beta_x r_x,$$
$$F_y(y) = 1 - e^{-(\alpha_y y)^m},\ \alpha_y = \beta_y r_y,$$

where $m > 0$ is the scale parameter of combat operations; r_x and r_y are the numbers of combat units at the disposal of the first and second parties, respectively; finally, $\beta_x > 0$ and $\beta_y > 0$ are the combat efficiencies of the units of the parties.

The densities of the random variables X and Y are

$$f_x(x) = \alpha_x m(\alpha_x x)^{m-1} e^{-(\alpha_x x)^m},$$
$$f_y(y) = \alpha_y m(\alpha_y y)^{m-1} e^{-(\alpha_y y)^m}.$$

The first party will defeat the opponent in combat and special operations with the probability

$$P_x = P(x < y) = 1 - P(x > y)$$
$$= 1 - \int_0^\infty f_x(x)\left[\int_0^x f_y(y)dy\right]dx. \quad (6)$$

(The fewer hits are required for a victory, the more efficient the combat units will be.)

With intermediate calculations omitted, formula (6) yields

$$P_x = \frac{(\beta_x r_x)^m}{(\beta_x r_x)^m + (\beta_y r_y)^m}.$$

A similar approach was used in the paper [34], but with a different interpretation of the model parameters and without the specifics of combat operations.

By denoting $x = r_x$, $y = r_y$, and $\beta = \beta_x/\beta_y$, we obtain the following VF in combat and special operations:

$$p_x(x, y) = \frac{(\beta x)^m}{(\beta x)^m + (y)^m},\ \beta = \varphi\rho, \quad (7)$$

where β is the combat superiority parameter of the first party over the second; φ is the moral superiority parameter; finally, ρ is the technological superiority parameter (superiority in coordination of actions, reconnaissance, firepower, and maneuverability; for details, see the paper [35]).

Let $q = \beta x/y$ be the ratio of the forces of the parties. Then

$$p_x = \frac{q^m}{q^m + 1} = 1 - s^{-m},\ s^m = q^m + 1,\ s > 1, \quad (8)$$

which is the Pareto distribution. A random variable with the distribution function (8) has the density

$$f(s) = ms^{-m-1},\ s > 1,$$

and the mean

$$M[S] = \int_1^\infty ms^{-m}ds = \left.\frac{ms^{1-m}}{1-m}\right|_1^\infty.$$

If $m \le 1$, the mean of the distribution (8) is infinite. Consequently, the VF (8) with $m \le 1$ reflects true uncertainty whereas the VF (8) with $m > 1$ measurable uncertainty.

From formula (8), we find the required ratio of forces to achieve victory with a given probability:

$$q = \sqrt[m]{\frac{p_x}{1 - p_x}}. \quad (9)$$

In the paper [36], the scale parameter was statistically evaluated, and the hypothesis on the conformity of model (7) to statistical data was verified using Pearson's χ^2 test. Based on a sufficiently large volume of statistical data for the 19th–20th centuries, it was shown that the parameter m has the following values:

– for special operations, $m \approx 0.5$;

– for battles (the tactical level), $m \approx 1$;

– for combat operations (the operational level), $m \approx 2$;

– for military operations (the strategic level), $m \approx 3$.

In applied sciences, the most important question is: at what probability may an event be considered almost true?

There are four levels of combat readiness for formations, units, and subunits[3]: combat-ready (at least 75% of organizational structures are combat-ready); boundedly combat-ready (50–75%); partially combat-ready (30–50%); and non-combat-ready (less than 30% of organizational structures are combat-ready).

Concerning preparation for combat and special operations, the following confidence degrees of the first party's victory can be assigned:

– an almost confident victory (the probability of victory ranges from 0.85 to 0.95);

– a sufficient confidence degree of victory (the corresponding probability ranges from 0.80 to 0.85);

– an acceptable confidence degree of victory (the corresponding probability ranges from 0.7 to 0.8).

The confidence degrees are assigned considering the current situation. Assigning high confidence degrees of victory is not always advisable, as this re-

[3] Combat readiness. URL: https://encyclopedia.mil.ru/encyclopedia/dictionary/details.htm?id=3465@morfDictionary (Accessed August 10, 2023.)

quires a sufficiently great concentration of armed forces and, consequently, increases the risk of group defeat by the enemy forces.

According to the results of computations by formulas (8) and (9), as the scale of combat operations increases, uncertainty shifts, more and more, from true to measurable. For instance, at the strategic level, an almost confident victory is achieved with a twofold superiority over the enemy. The greatest uncertainty is characteristic of special operations, where even an acceptable confidence degree of victory over the enemy (guerrillas, sabotage and reconnaissance groups, and terrorist groups) requires nine-fold superiority.

2.2. Substantive Justification of the Victory Function

Based on the experience of the Great Patriotic War, the combat order of a Soviet rifle division was built in two echelons, in a strip 8–12 km wide and 8–10 km deep over the front. The defense strips of the army and the front were relatively narrow but covered a large area. In other words, at the operational and strategic levels of defense, a relatively small share of the front (army) forces could be deployed in a timely manner to hold the strip and launch a counterattack against the invading enemy; see [9, *subsection 3.2.3*]. This corresponds to a higher value of the scale parameter m of model (7).

After World War II, several generations of weapons changed, and the motorization of armed forces was almost completed; moreover, unmanned systems, high-precision weapons, and automated control systems for forces and weapons appeared. The capabilities of divisions and brigades to defeat the enemy in the tactical and operational depth grew significantly.

The impact of modern weapon systems (and, consequently, new tactical tricks and action methods for armed forces) on the significance of the scale parameter of the VF at the operational and strategic levels was assessed in [9]. Due to the greater effective range of enemy reconnaissance and the defeat of its combat units, the values of the scale parameter at the operational and strategic levels decreased to $m \approx 1.5$–2 for the operational level and $m \approx 2$–3 for the strategic level. Estimating the value of the parameter m for various theaters and conditions of warfare is a topical scientific problem.

3. MODELS OF CONFLICT AND APPROPRIATION: A REVIEW

In conventional economics, appropriation is understood as a nonviolent process guaranteed by perfect property rights and their unimpeded enforcement. Conflict economics is based on a model of competition between two or more players (agents), each choosing between producing resources (consumer goods) and producing weapons (tools intended to appropriate the resources produced by other players, individually or jointly). Recall the well-known *free rider problem*: individual consumers of public goods contribute nothing to their provision, hoping that others will do so for them. In view of this problem, it has been established that group structures are less stable the more participants they have; see the reviews in [7, 37].

The following environment was considered in [7]. There are two identical and risk-neutral agents (countries) competing for R units of a resource that can be *consumed directly*. Due to the imperfection of governance and enforcement institutions, the dispute may be resolved by a conflict (threat of conflict). The objective functions of the parties are

$$V_i(G_1, G_2) = p_i(G_1, G_2)R - G_i,$$

$$p_i(G_1, G_2) = \frac{G_i^{\mu}}{G_1^{\mu} + G_2^{\mu}}, \ i = 1, 2,$$

where $0 < \mu \le 1$ is the determination parameter; G_i is the amount of resources spent by the ith party on weapons production.

The optimal values of the objective functions are

$$V_i(G^*) = V^* = \frac{1 - \mu/2}{2}R, \ i = 1, 2.$$

Since $\mu \le 1$, the players benefit by allocating funds for armaments. Next, the authors of [7] considered the costs of conflict and two-stage games: at the first stage, the parties simultaneously and independently allocate resources for weapons production; at the second stage, the parties start negotiations on the distribution of the disputed resource. If they reach an agreement, the resource will be distributed between the parties. Otherwise, the negotiations end in conflict, and the winner will take all of the disputed resource.

A. Alesina and E. Spolaore studied a more general problem, i.e., the relationship between a conflict and the distribution of country sizes in a model where both peaceful negotiations and military conflicts are possible [8]. As is known, when the size of a country and its population grow, the costs of public goods (defense, security, education, etc.) for its citizens decrease; however, the costs of coordinating the interests and preferences of different ethnic and social groups increase accordingly. In the case of high heterogeneity of these groups and attempts to impose certain actions on all groups, the costs may manifest themselves in the form of interethnic conflicts and civil wars. The

authors demonstrated the existence of optimal sizes for countries, which can be established through negotiations or conflicts.

4. APPLICATION OF THE VICTORY FUNCTION IN CONFLICT MODELING

Consider two countries, $i = 1, 2$. (Without loss of generality, these can be two blocs of countries governed by two centers.) For the sake of clarity, assume that there is a disputed territory ($i = 3$), e.g., previously under the joint control of the first and second countries (Fig. 2).

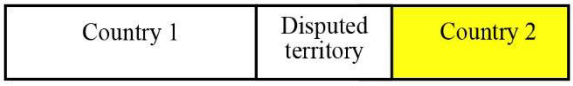

Fig. 2. Two countries and a disputed resource.

Let us introduce the following notation: s_i is the area of country/disputed territory i; z_i is the population of country i; $\mu_{ij} \geq 1$ is the parameter of ethnic diversity between the populations of countries i and j; δ_i is the parameter of attraction of country i; finally, A_i is the socio-technological development index of country i. Suppose that $A_1 = A_2 = A_3 = A$ and $z_1 > z_2 > z_3$.

Following the established tradition (see the paper [8]), we will assign each country an analog of a production function, i.e., a security function of the form [38]

$$u_i = w_i q_i, \quad w_i = A\left(\frac{z_i}{z_{\max}}\right)^{\omega}\left(\frac{s_i}{s_{\max}}\right)^{1-\omega},$$
$$q_i = \left(\frac{\xi_i}{z_i}\right)^{\frac{\xi_i + \mu_i(z_i - \xi_i)}{\delta_i z_i}},$$

where $z_{\max}$ is the population of the country with the largest population (currently India); $s_{\max}$ is the area of the country with the largest territory (Russia); $\omega \approx 0.5$ is the population elasticity parameter; ξ_i is the population of the main ethnic group in country i; μ_i is the parameter of ethnic diversity in country i (between the main ethnic group and the others); δ_i is the parameter of attraction of the main ethnic group in country i; w_i is the sovereignty function of country i; finally, q_i is the preservation function of country i.

If $\mu_i = 1$, there are no ethnic differences in the country (in terms of the participation of ethnic groups in socially significant activities). As the value of μ_i increases, these differences grow. If $\delta_i > 1$, an ethnic group is capable of effectively integrating other nationalities into society. Small values of the parameter ($\delta_i < 0.5$–0.6) are characteristic of peoples without established statehood [38].

The entry of new countries (regions, territories) into a country (union, bloc) increases the value of the sovereignty function (and, consequently, reduces the costs of individuals for the production of public goods), but at the same time reduces the value of the preservation function (an increase in costs associated with interethnic conflicts).[4] Figure 3 shows an example of the dynamics of the functions of sovereignty, preservation, and security.

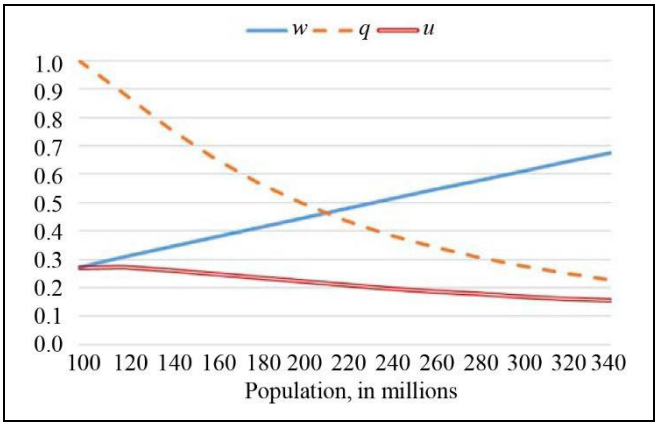

Fig. 3. The functions of sovereignty (w), preservation (q), and security (u).

At ***the first stage*** (*negotiations*), we calculate the security functions of the new unions of countries (territories). The security function of country 1 and the disputed territory is

$$U_{13} = A\left(\frac{z_1 + z_3}{z_{\max}}\right)^{\omega}\left(\frac{s_1 + s_3}{s_{\max}}\right)^{1-\omega}\left(\frac{\xi_1}{z_1 + z_3}\right)^{\frac{\mu_1 z_1 + \mu_{13} z_3}{\delta_1(z_1 + z_3)}}.$$

The security function of country 2 and the disputed territory is

$$U_{23} = A\left(\frac{z_2 + z_3}{z_{\max}}\right)^{\omega}\left(\frac{s_2 + s_3}{s_{\max}}\right)^{1-\omega}\left(\frac{\xi_2}{z_2 + z_3}\right)^{\frac{\mu_2 z_2 + \mu_{23} z_3}{\delta_2(z_2 + z_3)}}.$$

The security function of countries 1 and 2 and the disputed territory is

$$U_{123} = A\left(\frac{z_1 + z_2 + z_3}{z_{\max}}\right)^{\omega}\left(\frac{s_1 + s_2 + s_3}{s_{\max}}\right)^{1-\omega}$$
$$\times\left(\frac{\xi_1}{z_1 + z_2 + z_3}\right)^{\frac{\mu_1 z_1 + \mu_{12} z_2 + \mu_{13} z_3}{\delta_1(z_1 + z_2 + z_3)}}.$$

[4] The ethnic composition of a country (hence, the value of its security function) can change as a result of uncontrolled migration.

In the absence of conflict, the following equilibria (payoffs from negotiations) are possible depending on the values of the security functions of individual countries and their unions:

– The disputed territory declares sovereignty.
– Country 1 and the disputed territory unite.
– Country 2 and the disputed territory unite.
– Countries 1, 2, and the disputed territory unite.

In the event of disagreement by one or more agents (countries, governments), ***the second stage*** (conflict, combat operations) may begin. The subject of the conflict may be the struggle between countries 1 and 2 for the disputed territory. We define the payoff functions of these countries as follows:

$$H_i = V_i\pi_i - C_i x_i,\ \pi_1 = \frac{(\beta x_1)^m}{(\beta x_1)^m + (x_2)^m},$$

$$\pi_2 = 1 - \pi_1,\ V_i = \frac{(z_i + z_3)\delta_i}{z_i \mu_{i3}},\ i = 1, 2,$$

where V_i is the value of the disputed territory for country i; π_i is the probability of country i's victory; x_1 (x_2) is the resource allocated by the first (second, respectively) country for combat operations; β is the parameter of the combat superiority of the first country's armed forces over those of the second country; finally, C_i is the cost of acquiring and maintaining the resources of country i. The parameter μ_{i3} of the ethnic diversity between country i and the disputed territory reflects the former's costs of conducting combat operations in the disputed territory, and the attraction parameter δ_i reflects the capability to reduce these costs. Thus, the object's value is proportional to the degree of increase in the country's population and the attraction parameter and inversely proportional to the diversity parameter.

Let $C_1 = C_2 = C$. To find the Nash equilibrium, we apply the first-order necessary optimality conditions. Omitting the intermediate calculations of the partial derivatives, we obtain

$$x_1^* = \frac{m(V_1V_2)^m}{C[(V_1)^m + (V_2)^m]^2}\frac{V_1}{\beta},\ x_2^* = \frac{m(V_1V_2)^m}{C[(V_1)^m + (V_2)^m]^2}V_2.$$

In this conflict, the probability of the first country's victory over the opponent is

$$\pi_1 = \frac{(V_1)^m}{(V_1)^m + (V_2)^m} = \frac{1}{1 + \left(\frac{\delta_2(1 + z_3/z_2)\mu_{13}}{\delta_1(1 + z_3/z_1)\mu_{23}}\right)^m}.$$

Having determined the expected outcomes of the conflict, the parties proceed to ***the third stage***—estimating the costs of integrating the disputed territory. We define the objective functions of individuals (citizens) in the first and second countries as follows:

$$G_i = Y_i - T_i - (\mu_{i3} - 1)S_i\frac{\pi_i z_3}{z_i},\ i = 1, 2,$$

where Y_i is the citizen's income in country i; T_i is the citizen's taxes; finally, S_i is the citizen's costs due to counterterrorism and special operations.

The last expression shows that the citizen's costs depend significantly on the parameter of the ethnic diversity between the country and the population of the disputed territory. These costs also depend on the ratio of the population of the disputed territory to the population of the country.

Based on the security indicator, conflict costs, and the impact of integrating a new population into the country on the citizens' welfare, the government can make well-grounded decisions and gain public support.

CONCLUSIONS

In this study, we have formulated a comprehensive set of substantive and formal requirements for the victory function (VF) in combat and special operations as a variation of contest success functions (CSFs).

Using A.N. Kolmogorov's law of target destruction and the Weibull distribution, a particular type of VFs has been justified in probabilistic terms. Also, the function and its parameters have been substantively justified based on the postulates of military science and operational practice. With the above justification, the VF in combat and special operations should be treated as a separate class of CSFs.

In recent decades, the scope of political economy and economics has expanded. In addition to the conventional problems of production and distribution of goods, the issues of appropriation as a result of conflict between agents (countries) and the establishment of optimal boundaries between countries have begun to be investigated. Using a security model and the VF, three tasks (stages) have been set for the distribution of disputed resources (territory and population) between countries. At the first stage, to justify its negotiating position, each country calculates the security function under the condition that the disputed territory will become part of it. If the issue is not resolved peacefully, the governments of the countries threaten conflict (combat operations), allocating the appropriate resources to the armed forces. The second stage is to determine the expected outcome of the conflict. At the third stage, the costs of integrating the disputed territory into one of the countries are estimated. As a

result, the government can gain the support of the public during negotiations or in the course of the conflict.

A promising line of further research is to develop conflict theory at several levels, namely, in the theater of military operations, the intercountry, and geopolitical levels.

Acknowledgements. *The author is grateful to A.A. Galyaev and D.A. Novikov for careful reading of the manuscript and helpful remarks.*

REFERENCES

1. Peregudov, F.I. and Tarasenko, F.P., *Osnovy sistemnogo analiza* (Fundamentals of Systems Analysis), Tomsk: NTL, 1997. (In Russian.)
2. Belov, M.V. and Novikov, D.A., *Methodology of Complex Activity. Foundations of Understanding and Modeling*, Cham: Springer, 2020.
3. Cobb, C.W. and Douglas, P.H., A Theory of Production, *The American Economic Review*, 1928, vol. 18, no. 1, pp. 139–165.
4. Bernoulli, D., Specimen theoriae novae de mensura sortis, *Commentarii Academiae Scientiarum Imperialis Petropolitanae. Petropoli*, 1738, vol. V, pp. 175–192. (In Latin.)
5. von Neumann, J. and Morgenstern, O., *Theory of Games and Economic Behavior*, Princeton: Princeton University Press, 1944.
6. Schoemaker, P.J.H., The Expected Utility Model: Its Variants, Purposes, Evidence and Limitations, *Journal of Economic Literature*, 1982, vol. XX, no.2, pp. 529–563.
7. Garfinkel, M.R. and Skaperdas, S., Economics of Conflict: An Overview, *Handbook of Defense Economics*, 2007, no. 2, pp. 649–709.
8. Alesina, A. and Spolaore, E., Conflict, Defense Spending, and the Number of Nations, *European Economic Review*, 2006, no. 50, pp. 91–120.
9. *Modeli voennykh, boevykh i special'nykh deistvii* (Models of Military, Combat and Special Operations), Novikov, D.A., Ed., Moscow: LENAND, 2025. (In Russian.)
10. Robinson, J., The Production Function and the Theory of Capital, *The Review of Economic Studies*, 1953, vol. 21, no. 2, pp. 81–106.
11. Solow, R.M., The Production Function and the Theory of Capital, *The Review of Economic Studies*, 1955, vol. 23, no. 2, pp. 101–108.
12. Haavelmo, T., *A Study in the Theory of Economic Evolution*, Amsterdam: North-Holland, 1954.
13. Bernoulli, J., *Ars conjectandi, opus posthumum. Accedit Tractatus de seriebus infinitis, et epistola Gallice scripta De ludo pilae reticularis*, Basileae: impensis Thurnisiorum, fratrum, 1713. (In Latin.)
14. Laplace, P.S., *Théorie analytique des probabilities*, Paris: Ve. Courcier, 1812. (In French.)
15. Novikov, D.A., *Cybernetics: From Past to Future*, Cham: Springer, 2016.
16. Clausewitz, K., *Vom Krieg*, 1832. (In German.)
17. Grodetskaya, A.G., On War and Peace on the Coast of Hudson: The War and Peace Conference at West Point, *Russian Literature*, 2010, no. 4, pp. 115–119. (In Russian.)
18. Lipsky, T., Tolstoy's Complaint: Mission Command in the Age of AI, Modern War Institute. URL: https://mwi.westpoint.edu/tolstoys-complaint-mission-command-in-the-age-of-artificial-intelligence/ (Accessed September 8, 2025.)
19. Tolstoy, L.N., *War and Peace*, translated by A. and L. Maude, Oxford University Press, 2010.
20. Knight, F., *Risk, Uncertainty and Profit*, Hart Schaffner and Marx Prize Essays, no. 31, Boston–New York: Houghton Mifflin, 1921.
21. Kuznetsov, O.P., On Machine Learning, Myths About General AI, and What Understanding Is, *Ontology of Designing*, 2024, vol. 14, no. 4 (54), pp. 466–482. (In Russian.)
22. Zabudsky, N.A., *Teoriya veroyatnostei i primenenie ee k strel'be i pristrelke* (Probability Theory and Its Application to Shooting and Sighting), St. Petersburg: Printing House of the Imperial Academy of Sciences, 1898. (In Russian.)
23. Osipov, M.P. The Influence of the Number of Combatants on Their Losses, *Voennyi Sbornik*, 1915, no. 6, pp. 59–74; no. 7, pp. 25–36; no. 8, pp. 31–40; no. 9, pp. 25–37. (In Russian.)
24. Lanchester, F., *Aircraft in Warfare: the Dawn of the Fourth Arm*, London: Constable and Co., 1916.
25. Triandafilov, V.K., *Kharakter operatsii sovremennykh armii* (The Nature of Operations of Modern Armies), 3rd ed., Moscow: Voenizdat, 1936. (In Russian.)
26. Nash, J., Non-cooperative Games, *Annals of Mathematics*, 1951, vol. 54, no. 2, pp. 286–295.
27. G.K. Zhukov's Speech at a Military and Scientific Conference, December 1945, *Voennaya Mysl'*, 1985, special issue (February), pp. 3, 17–33. (In Russian.)
28. Wiener, N., *Cybernetics or Control and Communication in the Animal and the Machine*, New York: John Wiley & Sons; Paris: Hermann et cie, 1948.
29. Deitchman, S., A Lanchester Model of Guerrilla War, *Operational Research*, 1962, no. 10, pp. 818–827.
30. Luce, R.D., *Individual Choice Behavior: A Theoretical Analysis*, New York: Wiley, 1959.
31. Skaperdas, S., Contest Success Functions, *Economic Theory*, 1996, no. 7, pp. 283–290.
32. Jia, H., Skaperdas, S., and Vaidya, S., Contest Functions: Theoretical Foundations and Issues in Estimation, *International Journal of Industrial Organization*, 2013, no. 31, pp. 211–222.
33. Kolmogorov, A.N., The Number of Hits with Multiple Shots and General Principles for Evaluating the Effectiveness of a Firing System, *Trudy Mat. Inst. im. V.A. Steklova*, 1945, vol. 12, pp. 7–25. (In Russian.)
34. Jia, H.A, Stochastic Derivation of the Ratio Form of Contest Success Functions, *Public Choice*, 2008, no. 135 (3), pp. 125–130.
35. Korepanov, V.O. and Shumov, V.V., Modeling Military, Combat, and Specialized Actions, *Voennaya Mysl'*, 2023, no. 1, pp. 28–41. (In Russian.)
36. Shumov, V.V., A Study of Contest Success Function for Battles (Combats, Operations), *Control Sciences*, 2020, no. 6, pp. 19–30. (In Russian.)
37. Elagin, D.P., Political Economy of Violent Conflict: A Review of Theoretical Explanations of Conflict Onset (Part I), *World and National Economy*, 2021, no. 1 (54), art. no. 6. (In Russian.)
38. Shumov, V.V., *Natsional'naya bezopasnost': modelirovanie i prognozirovanie* (National Security: Modeling and Prediction), Moscow: LENAND, 2023. (In Russian.)

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

Received September 27, 2025,
Accepted October 23, 2025.

Author information
Shumov, Vladislav Vyacheslavovich. Dr. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ v.v.shumov@yandex.ru
ORCID ID: https://orcid.org/0000-0002-5722-7770

Cite this paper
Shumov, V.V., The Victory Function and Its Application in Conflict Modeling. *Control Sciences* **1,** 19–29 (2026).

Original Russian Text © Shumov, V.V., 2026, published in *Problemy Upravleniya*, 2026, no. 1, pp. 22–33.

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

DESIGN OF SELF-CHECKING DISCRETE DEVICES BASED ON BOOLEAN SIGNALS CORRECTION AND COMPOSITION OF CONSTANT-WEIGHT CODES OF THE "1-OUT-OF-4" AND "3-OUT-OF-4" TYPES. PART I: The Design Method with Conversion of All Signals from the Object under Diagnosis

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

*,**Peter the Great Saint Petersburg Polytechnic University, St. Petersburg, Russia
*Solomenko Institute of Transport Problems, Russian Academy of Sciences, St. Petersburg, Russia

*✉ TrES-4b@yandex.ru, **✉ eseniya-elina@mail.ru

Abstract. It is proposed to use the composition of constant-weight codes of the "1-out-of-4" and "3-out-of-4" types in the design of self-checking discrete devices based on Boolean signals correction. For this composition of constant-weight codes, a simple and compact self-checking checker can be implemented, requiring only four test combinations for a complete check. The structure of a self-checking discrete device is described. The conversion of all four signals from an object under diagnosis is considered when designing a concurrent error-detection circuit for the object. The simplest algorithm for building a self-checking concurrent error-detection circuit based on the Boolean signals correction and the composition of 1-out-of-4 and 3-out-of-4 codes is developed. The results of experiments with combinational benchmarks are provided. They demonstrate the advantage of the proposed approach over the well-known duplication method and the one involving the 1-out-of-4 code together with Boolean signals correction, in terms of structural redundancy of the final self-checking discrete device. The method developed in this paper is promising for self-checking discrete device design with different element bases in various fields of application.

Keywords: self-checking discrete device, concurrent error-detection circuit, Boolean signals correction, composition of constant-weight codes, 1-out-of-4 and 3-out-of-4 codes, structural redundancy, controllable structure.

INTRODUCTION

When designing self-checking discrete devices, common methods are based on introducing redundancy not into the original object itself (the object under diagnosis) but into a special concurrent error-detection (CED) circuit [1–3]. Such a circuit is built taking into account the characteristics of the object under diagnosis and the selected diagnostic attribute for detecting faults and errors in computations. In essence, a working diagnosis system is implemented for a given object: based on the results of computing its functions, this system indirectly identifies the presence/absence of faults [4]. CED circuits are implemented as self-checking, which eliminates the problem of "watch dog" when the developer intends to ensure operability control for the CED circuit itself. Totally self-checking CED circuits are self-testing and protected from faults [5]. According to the first property, for all faults of a CED circuit from a given class, there exists at least one set of argument values applied to its inputs to identify a control error signal. The second property states that when any fault occurs in a CED circuit from a given class, either correct values are computed at its outputs, or an error signal is identified.

There are two main approaches to building CED circuits, each generating a variety of methods. The first (classical) approach has been known since the

middle of the last century [6]. Within this approach, data signals are formed at the parallel outputs of an object under diagnosis; when applying each set of argument values to the inputs, these signals are supplemented by check signals in the CED circuit to establish a certain correspondence between the data and check signals if the object is operable. This is often done using the diagnostic attribute of the belonging of codewords, formed by data and check signals, to a given uniform block code [7–9]. Object's faults during its operation cause errors in computations, which violate the correspondence between the values of the data and check signals and, in turn, are identified by the CED circuit. The second approach has been known since the 1990s–2000s [10–13]. In contrast to supplementing data signals with check signals, this approach converts all or part of the signals so that the required codeword with specified diagnostic properties is identified for each set of argument values.

Both approaches have advantages and drawbacks. For example, the approach based on signal supplementation directly considers the error detection properties of uniform block codes and makes it possible to use special circuitry methods surely detect certain types of errors in CED circuits [14, 15]. However, for a given element basis and method for building CED circuit components, its structure is unique, and the developer cannot regulate the performance characteristics of the final self-checking discrete device (e.g., structural redundancy, controllability, power consumption, etc.). For each set of argument values, the second approach (based on Boolean signals correction, BSC) allows selecting a codeword from a certain set to convert the binary vector from the outputs of an object under diagnosis; this feature provides wide possibilities for building various CED circuits even for a given element base and implementation methods [16]. In addition, optimization problems can be solved for a particular performance indicator of CED circuits. However, due to BSC for the vector at the object's outputs, well-known circuit engineering techniques cannot be applied to cover certain types of errors in CED circuits. Alternative approaches are required, such as the selection of subsets of outputs with special control properties (e.g., those on which errors with a given multiplicity d (single, double, triple, etc.) cannot occur) [17]. In all these cases, there are two options for building CED circuits: the first is the complete coverage of errors at the object's outputs, caused by its internal structure faults; and the second is the manifestation of faults at least on one set of argument values [18, 19].

The simplest way to implement CED circuits using the second approach is based on the use of inseparable block codes. They include constant-weight codes [20], Plotkin (Hadamard) codes [21], Borden codes [22], and various other compositions of constant-weight codes [23]. A distinctive feature of methods with such codes is that when organizing CED circuits, it is easy to use circuit engineering techniques to eliminate certain types of errors in codewords, whereas for separable codes, it is also necessary to consider potential simultaneous distortions of both data and check symbols [24].

Research by the authors has shown that, in addition to traditional constant-weight codes, there is a special composition of these codes formed by combining codewords belonging to the "1-out-of-4" and "3-out-of-4" types [23] (further referred to as 1-out-of-4 and 3-out-of-4 codes, respectively). A checker with a simple and compact structure can be built for it, requiring only four test combinations for a complete check. This is the minimum cardinality of the set of codewords forming a test for self-checking checkers [5].

This study is devoted to describing CED circuit design methods based on BSC with compositions of 1-out-of-4 and 3-out-of-4 codes. It consists of two parts: in the first, we present the CED circuit structure based on BSC with conversion of signals from all outputs of an object under diagnosis and the simplest algorithm for building a self-checking CED circuit; the second part will provide methods for reducing structural redundancy by utilizing the properties of the composition of 1-out-of-4 and 3-out-of-4 codes and decreasing the number of conversion elements in the CED circuit.

1. CED CIRCUIT BASED ON THE COMPOSITION OF 1-OUT-OF-4 AND 3-OUT-OF-4 CODES

The structure of a CED circuit based on the composition of 1-out-of-4 and 3-out-of-4 codes is shown in Fig. 1. It is built for a set of four outputs of an object under diagnosis. The output set of cardinality 4 is chosen due to the length of the code vector of the composition of 1-out-of-4 and 3-out-of-4 codes.

An object under diagnosis is the combinational part of a discrete device, or simply a combinational discrete device $F(X)$ that computes the values of Boolean functions $f_1(X)$, $f_2(X)$, $f_3(X)$, and $f_4(X)$ under a given set $\langle X\rangle = \langle x_t\ x_{t-1}\ \dots\ x_2\ x_1\rangle$ of argument values supplied to the inputs. (Hereinafter, this set is supposed to be complete.) Thus, a Boolean vector $\langle f_4(X)\ f_3(X)\ f_2(X)\ f_1(X)\rangle$ is formed for each set of argument values. A special CED circuit is organized to check the correctness of computations. It consists of three functional blocks: the block $G(X)$ for computing the values of correction functions, the signal correction block (SCB) itself, and a *totally self-checking checker* (TSC) for the composition of 1-out-of-4 and 3-out-of-4 codes.

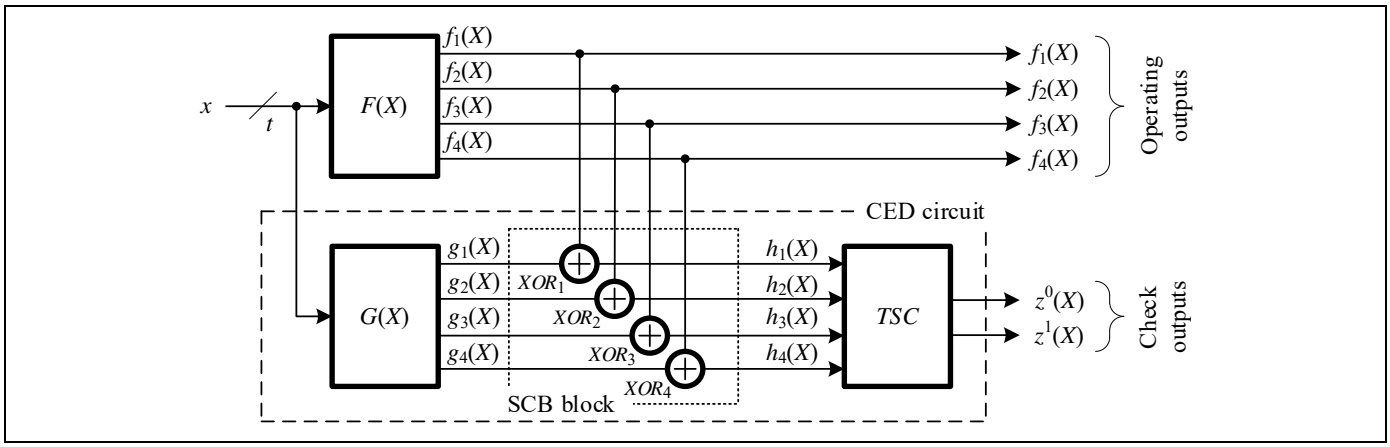

Fig. 1. The structural diagram of a CED circuit.

The block $G(X)$ is intended to compute the values of the functions $g_1(X)$, $g_2(X)$, $g_3(X)$, and $g_4(X)$ for correcting signals from the object $F(X)$ when identical sets of argument values are supplied to the inputs of both blocks. Each signal correction function implements the following conversion in the CED circuit:

$$h_i(X) = f_i(X) \oplus g_i(X),\ i = \overline{1,4}, \qquad (1)$$

where $h_i(X)$ is the corrected value at the SCB output obtained under a particular set of argument values supplied to the inputs.

Two-input XORs are used to correct signals. They allow any value 0 (1) to be converted to any of the values 0 (1).[1] Thus, for each set of argument values, the code vector $<f_4(X)\ f_3(X)\ f_2(X)\ f_1(X)>$ can be converted into the code vector $<h_4(X)\ h_3(X)\ h_2(X)\ h_1(X)>$ with specified diagnostic properties. The structure shown in Fig. 1 performs conversion to a codeword belonging to the composition of 1-out-of-4 and 3-out-of-4 codes. The TSC block is installed to verify the belonging of the codeword $<h_4(X)\ h_3(X)\ h_2(X)\ h_1(X)>$ to the given composition. It is equipped with two outputs, $z^0(X)$ and $z^1(X)$, which operate in two-rail logic: the combinations <01> and <10> indicate the absence of errors in computations whereas the combinations <00> and <11> the presence of an error (and, indirectly, the occurrence of a fault in the object under diagnosis).

Figure 2 demonstrates the structure of the simplest checker for the composition of 1-out-of-4 and 3-out-of-4 codes.

[1] The equivalence function implemented by an XNOR gate has a similar property. It could also be used in the CED circuit design based on BSC. Other elementary Boolean functions can be applied to BSC only with several restrictions.

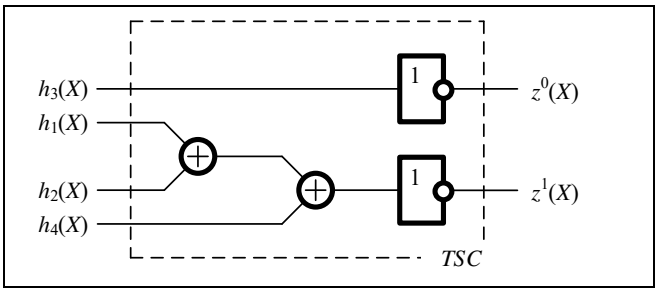

Fig. 2. The structural diagram of a checker for the composition of 1-out-of-4 and 3-out-of-4 codes.

The checking test T_{che} for the checker in Fig. 2 contains combinations from the following set:

$$T_{che} = \{1000, 0010, 1101, 0111\} \cup \{0001, 1011, 0100, 1110\}. \qquad (2)$$

When designing CED circuits, it suffices to form at least one subset of the set (2). In this case, a complete check of TSC can be performed during the operation of the self-checking device.

Thus, the structure in Fig. 1 is self-checking.

2. THE SIMPLEST ALGORITHM FOR BUILDING A CED CIRCUIT BASED ON THE COMPOSITION OF 1-OUT-OF-4 AND 3-OUT-OF-4 CODES

The conversion (1) is performed on each set of argument values. The vector $<f_4(X)\ f_3(X)\ f_2(X)\ f_1(X)>$ can be converted into the code vector $<h_4(X)\ h_3(X)\ h_2(X)\ h_1(X)>$ in eight different variants since the set of codewords for the composition of the 1-out-of-4 and 3-out-of-4 codes has cardinality $C_4^1 + C_4^3 = 8$. Hence, for t arguments, the conversion (1) can be performed in 8^{2^t} variants. Due to the need

to form the set T_{che} for the checker, we have $8^{2^t} - 3$ variants. Even in the case $t = 3$, this gives 16 777 213 variants to design self-checking CED circuits. Among them, the best variant can be selected, e.g., by the minimum structural redundancy criterion of a self-checking discrete device.

Meanwhile, let us propose a simple algorithm for building a CED circuit based on BSC and the composition of 1-out-of-4 and 3-out-of-4 codes, which forms the set T_{che} for the checker, but requires post-verification of the test combinations formed for XOR gates. Assume that all functions of an object under diagnosis are completely defined, and all 2^t sets of argument values are supplied to the inputs. Note also that at the initial stage, the structures of the BSC and TSC blocks are known, and the logic of $F(X)$ is specified; however, the values of the functions implemented by the block $G(X)$ are not set. The main goal of design is to obtain the values of these functions for each set of argument values and to form checking tests for the BSC and TSC blocks.

Algorithm 1. *CED circuit design for four outputs of the object under diagnosis:*

1. Determine $\delta = \dfrac{2^t}{8} = 2^{t-3}$, which is the number of each of the words used in the composition of constant-weight codes $<h_4(X)\ h_3(X)\ h_2(X)\ h_1(X)>$. As a result, one builds a CED circuit in which the codewords of the composition of 1-out-of-4 and 3-out-of-4 codes will be formed uniformly at the TSC inputs.

2. Consider the sets of argument values in lexicographic order, from the set with decimal equivalent 0 to the set with decimal equivalent $2^t - 1$. On the sets of argument values with decimal equivalents $0\ldots(2^{t-3} - 1)$, redefine the bits of the vectors $<h_4(X)\ h_3(X)\ h_2(X)\ h_1(X)>$ to the codeword <0001>; on the sets with numbers $2^{t-3}\ldots(2\cdot 2^{t-3} - 1)$, to the codeword <0010>; on the sets with numbers $2\cdot 2^{t-3}\ldots(3\cdot 2^{t-3} - 1)$, to the codeword <0100>; on the sets with numbers $3\cdot 2^{t-3}\ldots(4\cdot 2^{t-3} - 1)$, to the codeword <1000>; on the sets with numbers $4\cdot 2^{t-3}\ldots(5\cdot 2^{t-3} - 1)$, to the codeword <1110>; on the sets with numbers $5\cdot 2^{t-3}\ldots(6\cdot 2^{t-3} - 1)$, to the codeword <1101>; on the sets with numbers $6\cdot 2^{t-3}\ldots(7\cdot 2^{t-3} - 1)$, to the codeword <1011>; finally, on the sets with numbers $7\cdot 2^{t-3}\ldots(8\cdot 2^{t-3} - 1)$, the codeword <0111>. This step of the algorithm forms each of the codewords for the composition of 1-out-of-4 and 3-out-of-4 codes at the TSC inputs exactly δ times.

3. Determine the values of the functions $g_i(X)$ based on the known values of the functions $h_i(X)$ for each set of argument values:

$$\begin{gathered} g_i(X) = f_i(X) \oplus h_i(X) \\ \Leftrightarrow h_i(X) = f_i(X) \oplus g_i(X),\ i = \overline{1,\ 4}. \end{gathered} \qquad (3)$$

4. Verify the formation of the checking test for each XOR gate in the CED circuit: under the canonical realization of XOR, the checking test includes all four working combinations {00, 01, 10, 11} [25], and each of them shall be formed on at least one set of argument values. If tests are formed for all SCB gates, proceed to Step 5 of the algorithm; otherwise, redefine the bit values in the vector $<h_4(X)\ h_3(X)\ h_2(X)\ h_1(X)>$ on a selected number of the sets of argument values in accordance with the method described in [26].

5. Optimize the functions $g_4(X)$, $g_3(X)$, $g_2(X)$, and $g_1(X)$ [27].

6. Design a self-checking discrete device in the selected element basis.

We pay the reader's attention to Step 2 of the algorithm, which involves all eight codewords belonging to the given composition. However, in view of the expression (2), this step can be changed: the vectors $<h_4(X)\ h_3(X)\ h_2(X)\ h_1(X)>$ can be redefined to only four codewords of the given composition, which will ensure the complete check of the checker.

Example 1. Let us build a CED circuit for a device whose operating logic is described by the first nine columns of Table 1.

Following Step 1 of Algorithm 1, we determine the number $\delta = 2^{4-3} = 2$, characterizing the number of each of the words used in the composition of 1-out-of-4 and 3-out-of-4 codes. Then, according to Step 2 of Algorithm 1, we sequentially consider the sets of argument values and fix the following codewords of the composition of 1-out-of-4 and 3-out-of-4 codes in the vector $<h_4(X)\ h_3(X)\ h_2(X)\ h_1(X)>$: on sets 0 and 1, the word <0001>; on sets 2 and 3, the word <0010>; on sets 4 and 5, the word <0100>; on sets 6 and 7, the word <1000>; on sets 8 and 9, the word <1110>; on sets 10 and 11, the word <1101>; on sets 12 and 13, the word <1011>; finally, on sets 14 and 15, the word <0111> (see the columns corresponding to the bits of the vector $<h_4(X)\ h_3(X)\ h_2(X)\ h_1(X)>$ in Table 1). Using Step 3 of Algorithm 1 and formula (3), we obtain the values of the functions implemented at the outputs of the block $G(X)$ for each set of argument values (see the columns corresponding to the bits of the vector $<g_4(X)\ g_3(X)\ g_2(X)\ g_1(X)>$ in Table 1). Following Step 4 of Algorithm 1, we determine whether checking tests are formed for each of the gates XOR_4, XOR_3, XOR_2, and XOR_1. All combinations formed at their inputs are given in the corresponding columns of Table 1. Analysis of the combinations arriving at the inputs of each correction element shows that a checking test is formed for each of them.

Table 1

The operation of a device with four outputs and the CED circuit for it

No.	The sets of argument values				The values at the outputs of the device $F(X)$				The values at the outputs of the SCB block				The values at the outputs of the block $G(X)$				Test combinations of the SCB block			
	x_4	x_3	x_2	x_1	$f_4(X)$	$f_3(X)$	$f_2(X)$	$f_1(X)$	$h_4(X)$	$h_3(X)$	$h_2(X)$	$h_1(X)$	$g_4(X)$	$g_3(X)$	$g_2(X)$	$g_1(X)$	XOR_4	XOR_3	XOR_2	XOR_1
0	0	0	0	0	1	1	0	0	0	0	0	1	1	1	0	1	11	11	00	01
1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	1	00	00	11	01
2	0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	10	00	11	00
3	0	0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	01	00	00	00
4	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	00	10	00	0
5	0	1	0	1	0	0	1	0	0	1	0	0	0	1	1	0	00	01	11	00
6	0	1	1	0	0	1	0	0	1	0	0	0	0	1	1	0	00	11	01	00
7	0	1	1	1	0	0	0	1	1	0	0	0	0	0	1	1	00	00	01	11
8	1	0	0	0	1	0	1	0	1	1	1	0	1	1	1	0	11	01	11	00
9	1	0	0	1	1	0	0	0	1	1	1	0	1	1	0	0	11	01	00	00
10	1	0	1	0	0	1	1	0	1	1	0	1	1	1	1	0	01	11	11	00
11	1	0	1	1	0	0	0	1	1	1	0	1	1	1	0	0	01	01	00	10
12	1	1	0	0	1	0	1	1	1	0	1	1	0	0	0	0	10	00	10	10
13	1	1	0	1	1	0	1	0	1	0	1	1	0	0	0	1	10	00	10	01
14	1	1	1	0	1	1	1	1	0	1	1	1	1	0	0	0	11	10	10	10
15	1	1	1	1	0	1	0	1	0	1	1	1	0	0	1	0	00	10	01	10

Next, Steps 5 and 6 of Algorithm 1 should be carried out, but they appear to be trivial: the functions $g_4(X)$, $g_3(X)$, $g_2(X)$, and $g_1(X)$ are optimized, and a self-checking discrete device is implemented in the selected element basis. ♦

A CED circuit for a device with $n > 4$ outputs is built as follows.

Algorithm 2. *CED circuit design for an object under diagnosis with $n > 4$ outputs:*

1. Order the device outputs and divide them into $q = \left\lceil \frac{n}{4} \right\rceil$ subsets. (The symbol $\lceil ... \rceil$ denotes the ceiling of a corresponding value.) If n(mod4) = 0, each of the q subsets will contain unique outputs; otherwise, the last subset will contain n(mod4) unique outputs and 4 – n(mod4) ones already used in q – 1 subsets.

2. Design the CED circuit according to Algorithm 1 for each of the q subsets of outputs. To reduce redundancy, the individual blocks $G(X)$ of each CED circuit shall be built together.

3. Connect the outputs of q CED circuits to the inputs of the self-checking comparator qTRC1, which compresses q two-rail signals into one and is based on q – 1 elementary *two-rail checkers* (TRC) [28–30].

Example 2. Provide the structure of a self-checking discrete device for organizing a CED circuit for the initial device with $n = 10$ outputs.

According to Step 1 of Algorithm 2, we order the outputs and divide them into the following number of subsets: $q = \left\lceil \frac{10}{4} \right\rceil = 3$. The first two subsets are disjoint: I – $\{f_1(X), f_2(X), f_3(X), f_4(X)\}$, and II – $\{f_5(X), f_6(X), f_7(X), f_8(X)\}$. Since 10(mod4) = 2, the last subset includes two outputs from the second subset and two previously unused outputs: III – $\{f_7(X), f_8(X), f_9(X), f_{10}(X)\}$.

Figure 3 shows the structure of a self-checking discrete device in Example 2. To implement it, three CED circuits are allocated with separate SCB blocks and checkers. The check outputs of individual CED circuits are connected to the inputs of a self-checking comparator implemented on two TRC blocks. Since the outputs $f_7(X)$ and $f_8(X)$ are used twice, the corresponding functions in the CED circuit are marked by "*" and "**," respectively. ♦

The main advantage is that Algorithms 1 and 2 provide the simplest possible design of CED circuits for initial objects under diagnosis, and the procedures implemented in these algorithms can be easily automated for further use in computer-aided design systems for self-checking discrete devices.

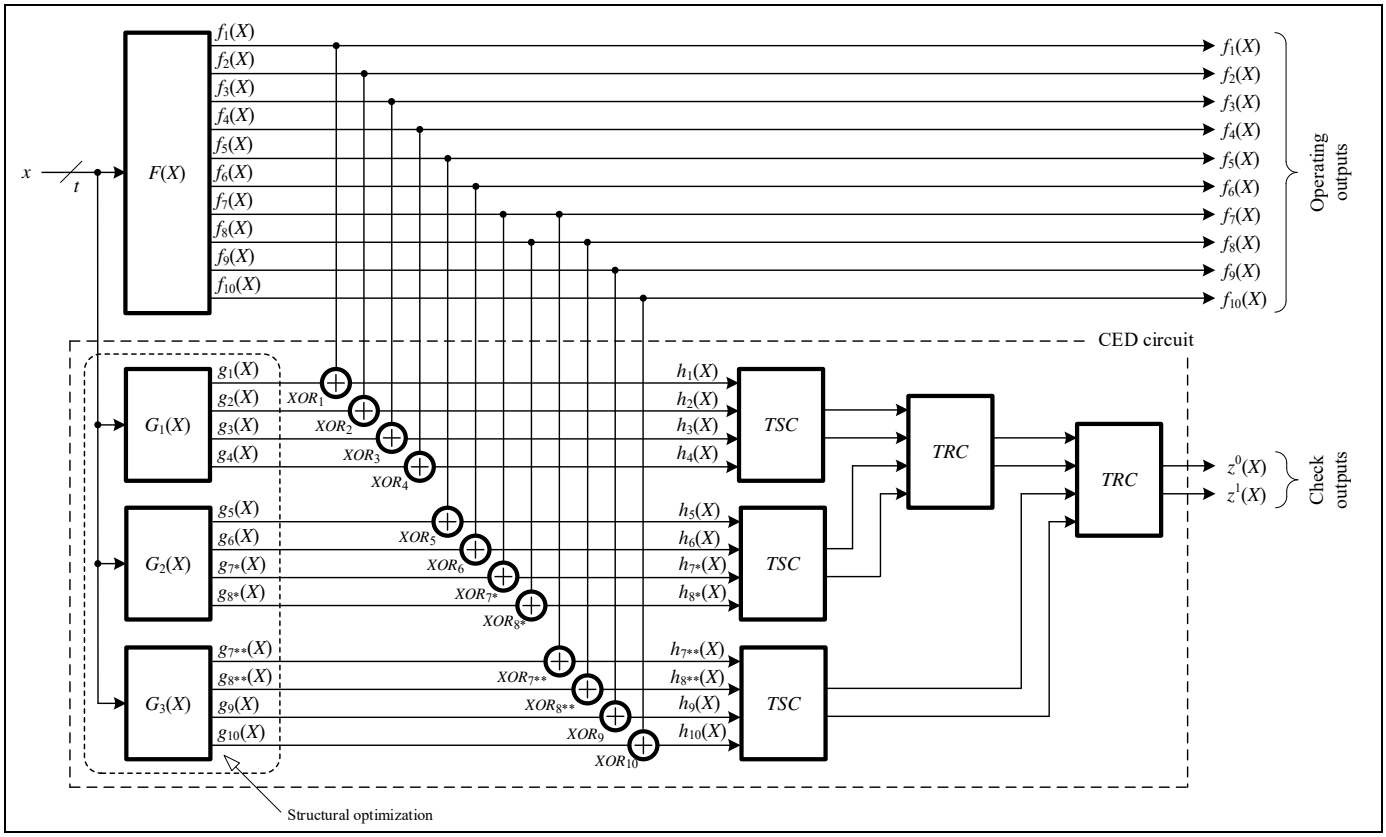

Fig. 3. The structural diagram of the self-checking discrete device in Example 2.

3. EXPERIMENTAL RESULTS

An important task of this study is to confirm experimentally the efficiency of the structure proposed (Fig. 1), as well as Algorithms 1 and 2, in the design of self-checking discrete devices for test combinational circuits (benchmarks) from MCNC Benchmarks [31, 32]. During the experiments, structural redundancy indicators were estimated for self-checking discrete devices designed for each benchmark. For comparison, two methods were also used with the same total number of outputs for the blocks $G(X)$ as for the object under diagnosis. The first method was classical duplication: all blocks $G(X)$ were actually replaced by a single copy of the object or by a device computing equivalent Boolean functions [6]. (Nowadays, this method is widely used to build self-checking implementations of discrete devices [33].) The second method was the one based on BSC and the use of the 1-out-of-4 code with correction of all signals from the object [34].

CED circuits were designed for the selected benchmarks by three methods. Combinational benchmarks in the *.pla format were taken. In this format, a circuit is specified by an interval description of Boolean functions (as a compressed truth table representing a ternary matrix with elementary conjunctions) [35]. The processing techniques for benchmarks, as well as the algorithms developed, were automated and realized in *DM Coding*[2] to obtain descriptions in the *.pla format for the functional blocks of CED circuits. Next, the well-known SIS interpreter [36, 37] and the *stdcell2_2.genlib* library of functional elements were used to design self-checking devices and determine some of their parameters. During the design, the *full_simplify* optimization procedure was applied for the blocks $G(X)$. (The other blocks of CED circuits were typical.) This procedure serves to optimize a system of Boolean functions using binary decision diagrams (BDDs) [38]. Then, the *map -s* procedure was carried out to perform the structural synthesis of devices in the given library of functional elements and obtain their main parameters. For comparison, an additional parameter—the area occupied by the device on a chip—was considered to characterize the implemen-

[2] DM Coding is a software module developed by the authors of this paper jointly with Cand Sci. (Eng.) V.V. Dmitriev. This module is expandable and intended for setting up experiments with novel fault diagnosis methods. Written in C#, it implements certain functions for analyzing benchmarks with the algorithms being developed, including the functional description of CED circuit blocks built by various methods.

tation complexity of a device (in conditional units of the element library). For each CED circuit component, the implementation complexity indicators were determined, including the final values of the implementation complexity indicator of a self-checking device (by analogy with the method described in [39]). For duplication, this indicator is denoted by L_D; for the method based on BSC and the 1-out-of-4 code, by $L_{1/4}$; for the novel method (proposed above), by $L_{1/4+3/4}$. The following relative indicators were calculated for comparing the three methods with each other:

$$\mu = \frac{L_{1/4+3/4}}{L_D} \cdot 100\%, \quad (4)$$

$$\eta = \frac{L_{1/4+3/4}}{L_{1/4}} \cdot 100\%. \quad (5)$$

The indicators (4) and (5) characterize the self-checking device designed by the novel method in comparison with those designed by duplication and BSC with the 1-out-of-4 code. If $\mu > 100\%$ ($\eta > 100\%$), then the novel method is more efficient than duplication (BSC with the 1-out-of-4 code, respectively). Also, for a convenient efficiency analysis of the novel method, the following indicators (the relative gain in its complexity) were calculated:

$$\Delta\mu = \left(1 - \frac{L_{1/4+3/4}}{L_D}\right) \cdot 100\,\%,$$

$$\Delta\eta = \left(1 - \frac{L_{1/4+3/4}}{L_{1/4}}\right) \cdot 100\,\%.$$

A value $\Delta\mu > 0\%$ ($\Delta\eta > 0\%$) indicates a gain in the complexity of the novel method compared to duplication (BSC with the 1-out-of-4 code, respectively).

Table 2 summarizes the results of experiments for several benchmarks. For each benchmark, the values of the implementation complexity indicators $L_{F(X)}$, L_D, $L_{1/4}$, and $L_{1/4+3/4}$ are provided. For the methods based on BSC, the values of the implementation complexity indicator $L_{G(X)}$ of the block $G(X)$ are also given. For circuits with $n \geq 5$ outputs, the block $G(X)$ was obtained by combining separate blocks for computing the values of signal correction functions for each of the check and design subcircuits into a single circuit. These indicators can be compared for different methods based on BSC in order to assess the impact of the complexity indicator of the block for computing correction functions on the final value of the complexity indicator of the self-checking discrete device. Separately, the bar charts in Fig. 4 show the values of $\Delta\mu$ and $\Delta\eta$.

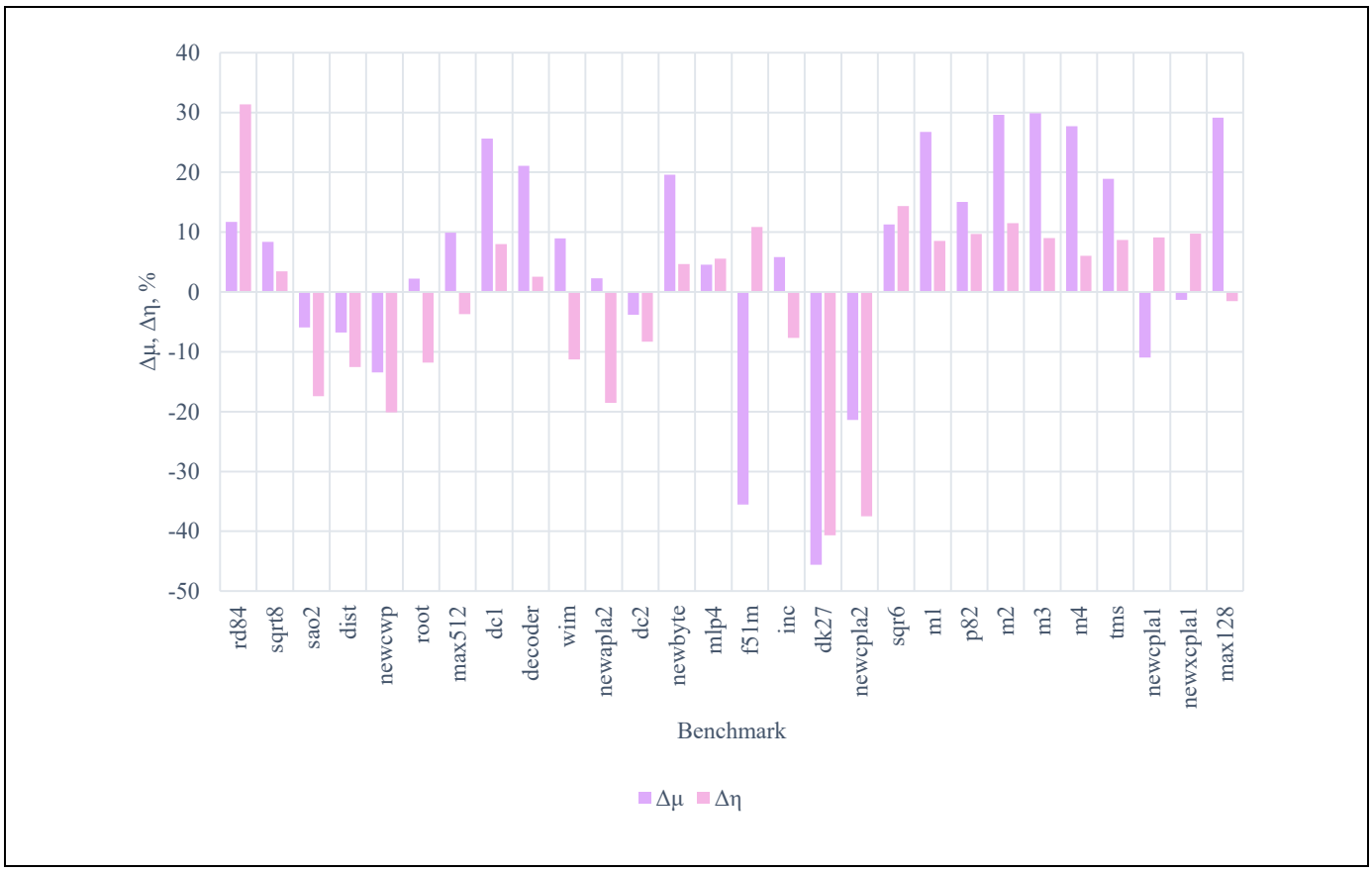

Fig. 4. The values of Δμ and Δη for different benchmarks.

Table 2

Experimental results

Benchmark	n	q	$L_{F(X)}$	L_D	The novel method		The method from [34]		μ	η	Δμ	Δη
					$L_{G(X)}$	$L_{1/4+3/4}$	$L_{G(X)}$	$L_{1/4}$				
rd84	4	1	4912	10464	4056	9240	8128	13456	88.303	68.668	11.697	31.332
sqrt8	4	1	1160	2960	1280	2712	1232	2808	91.622	96.581	8.378	3.419
sao2	4	1	2992	6624	3752	7016	2568	5976	105.918	117.403	–5.918	–17.403
dist	5	2	6968	14784	8080	15784	6448	14024	106.764	112.55	–6.764	–12.55
newcwp	5	2	440	1728	784	1960	584	1632	113.426	120.098	–13.426	–20.098
root	5	2	3496	7840	3432	7664	2752	6856	97.755	111.785	2.245	–11.785
max512	6	2	9632	20320	7944	18312	7224	17656	90.118	103.715	9.882	–3.715
dc1	7	2	976	3216	680	2392	632	2600	74.378	92	25.622	8
dekoder	7	2	736	2736	688	2160	488	2216	78.947	97.473	21.053	2.527
wim	7	2	712	2688	1000	2448	496	2200	91.071	111.273	8.929	–11.273
newapla2	7	2	600	2464	1072	2408	480	2032	97.727	118.504	2.273	–18.504
dc2	7	2	2424	6112	3184	6344	2440	5856	103.796	108.333	–3.796	–8.333
newbyte	8	2	592	2656	808	2136	384	2240	80.422	95.357	19.578	4.643
mlp4	8	2	7224	15920	7232	15192	7520	16088	95.427	94.431	4.573	5.569
f51m	8	2	2272	6016	5144	8152	5528	9144	135.505	89.151	–35.505	10.849
Inc	9	3	2376	6432	2480	6056	1712	5624	94.154	107.681	5.846	–7.681
dk27	9	3	528	2736	2256	3984	768	2832	145.614	140.678	–45.614	–40.678
newcpla2	10	3	1896	5680	3800	6896	1392	5016	121.408	137.48	–21.408	–37.48
sqr6	12	3	2648	7600	2896	6744	2672	7872	88.737	85.671	11.263	14.329
m1	12	3	3064	8432	1912	6176	1096	6752	73.245	91.469	26.755	8.531
p82	14	4	2368	7456	2304	6336	1712	7016	84.979	90.308	15.021	9.692
m2	16	4	10096	23328	4656	16416	4288	18544	70.37	88.525	29.63	11.475
m3	16	4	13464	30064	5952	21080	5536	23160	70.117	91.019	29.883	8.981
m4	16	4	18704	40544	8952	29320	8344	31208	72.316	93.95	27.684	6.05
tms	16	4	6784	16704	5096	13544	3928	14832	81.082	91.316	18.918	8.684
newcpla1	16	4	2520	8176	4888	9072	3304	9984	110.959	90.865	–10.959	9.135
newxcpla1	23	6	3760	12112	5920	12272	3216	13600	101.321	90.235	–1.321	9.765
max128	24	6	20192	45184	9232	32016	4528	31536	70.857	101.522	29.143	–1.522

According to Table 2 and Fig. 4, we draw the following conclusions. For 9 of the 28 benchmarks, duplication is more efficient. However, for most benchmarks (19, approximately 68%), the novel method yields self-checking discrete devices with smaller implementation complexity. Moreover, the gain for some benchmarks is $\Delta\mu > 20\%$. Compared to the method described in [34], the novel one is more effective for 16 benchmarks (approximately 57%). The improvement in the indicator $\Delta\eta$ does not exceed 10% for most benchmarks. This is explained by the similarity between the novel method and the one from [34] in terms of the number of signals corrected. The effect is achieved mainly due to the gain in the indicator $L_{G(X)}$ when applying one or another method. For three benchmarks (see Fig. 4), a significant loss is obtained in the implementation complexity indicator: over 20% for both $\Delta\mu$ and $\Delta\eta$. This result is due to the complexity of the Boolean functions implemented by the blocks $F(X)$ and $G(X)$ [40]. In some cases, the complexity of Boolean functions implemented by the block $G(X)$ is significantly higher than that of the ones implemented by the block $F(X)$. (The reader can compare the data in the columns $L_{F(X)}$ and $L_{G(X)}$ row by row.) Nevertheless, for most benchmarks, the novel method demonstrates a positive effect, which suggests its practical applicability.

CONCLUSIONS

When designing self-checking discrete devices based on BSC, the composition of 1-out-of-4 and 3-out-of-4 codes can be efficiently used. The checker for this code has a simple and compact structure, and four combinations are sufficient for a complete check. As a result, it is rather easy to build self-checking CED circuits for discrete devices, e.g., using the algorithms presented above.

According to the structural redundancy analysis of the experimental results obtained using the software modules and products, the method proposed in this paper has advantages over some known design methods of self-checking discrete devices. Moreover, as will be demonstrated in part II of the study, the properties of the composition of 1-out-of-4 and 3-out-of-4 codes can be utilized to reduce the number of signals converted from the object under diagnosis and, thereby, decrease the number of outputs of the blocks $G(X)$. Such a structural modification procedure for CED circuits shown in Fig. 1 will provide an even greater gain in terms of the implementation complexity of self-checking discrete devices compared to well-known methods. Note that other metrics, synthesizers, and sets of functional elements can also be used to determine the features and characteristics of the novel method.

The advantage of using BSC and the composition of 1-out-of-4 and 3-out-of-4 codes is a large number of variants to design CED circuits: one can solve related optimization problems and build self-checking discrete devices, e.g., with the lowest hardware cost. A drawback of the novel method is the need to consider the number of combinations of distortions at the outputs of an object under diagnosis when selecting quadruples among them. There are two approaches for building a CED circuit: the classical one with full coverage of any errors at the object's outputs, and the one with coverage of faults occurring at least on one set of argument values. Also, it is possible to use the first approach for implementing a CED circuit by covering any errors at the object's outputs with an increased number of the checked subsets of outputs (as compared to the minimum required for the novel method), but with an advantage over other methods in terms of implementation complexity. The possibility of such an implementation has been demonstrated by some experimental results, particularly by the above analysis of the gains in complexity indicators.

REFERENCES

1. Sogomonyan, E.S. and Slabakov, E.V., *Samoproveryaemye ustroistva i otkazoustoichivye sistemy* (Self-Checking Devices and Fault-Tolerant Systems), Moscow: Radio i Svyaz', 1989. (In Russian.)
2. Mitra, S. and McCluskey, E.J., Which Concurrent Error Detection Scheme to Choose?, *Proceedings of International Test Conference*, Atlantic City, 2000, pp. 985–994. DOI: 10.1109/TEST.2000.894311
3. Göessel, M., Ocheretny, V., Sogomonyan, E., and Marienfeld, D., *New Methods of Concurrent Checking*, 1st ed., Dordrecht: Springer Science+Business Media, 2008.
4. Drozd, A.V., An Untraditional View on Operational Diagnostics of Computing Devices, *Control Sciences*, 2008, no. 2, pp. 48–56. (In Russian.)
5. Sapozhnikov, V.V. and Sapozhnikov, Vl.V., *Samoproveryaemye diskretnye ustroistva* (Self-Checking Discrete Devices), St. Petersburg: Energoatomizdat, 1992. (In Russian.)
6. Goessel, M. and Graf, S., *Error Detection Circuits*, London: McGraw-Hill, 1994.
7. Sapozhnikov, V.V., Sapozhnikov, Vl.V., and Efanov, D.V., *Kody Khemminga v sistemakh funktsional'nogo kontrolya logicheskikh ustroistv* (Hamming Codes in Functional Control Systems of Logical Devices), St. Petersburg: Nauka, 2018. (In Russian.)
8. Sapozhnikov, V.V., Sapozhnikov, Vl.V., and Efanov, D.V., *Kody s summirovaniem dlya sistem tekhnicheskogo diagnostirovaniya. Tom 1: Klassicheskie kody Bergera i ikh modifikatsii* (Sum Codes for Technical Diagnosis Systems. Vol. 1: Classical Berger Codes and Their Modifications), Moscow: Nauka, 2020. (In Russian.)
9. Sapozhnikov, V.V., Sapozhnikov, Vl.V., and Efanov, D.V., *Kody s summirovaniem dlya sistem tekhnicheskogo diag-*

nostirovaniya. Tom 2: Vzveshennye kody s summirovaniem, (Sum Codes for Technical Diagnosis Systems. Vol. 2: Weight-Based Codes), Moscow: Nauka, 2021. (In Russian.)
10. Goessel, M., Saposhnikov, V., Saposhnikov, Vl., and Dmitriev, A., A New Method for Concurrent Checking by Use of a 1-Out-Of-4 Code, *Proceedings of 6th IEEE International On-Line Testing Workshop*, Palma de Mallorca, Spain, 2000, pp. 147–152. DOI: 10.1109/OLT.2000.856627
11. Dmitriev, A., Saposhnikov, V., Saposhnikov, V., and Goessel, M., New Self-Dual Circuits for Error Detection and Testing, *VLSI Design*, 2000, vol. 11, no. 1, pp. 1–21. DOI: 10.1155/2000/84720
12. Sapozhnikov, V.V., Sapozhnikov, Vl.V., Dmitriev, A.V., et al., Organization of Functional Control of Combinational Circuits Using the Boolean Complement Method, *Electronic Modeling*, 2002, vol. 24, no. 6, pp. 52–66. (In Russian.)
13. Gessel, M., Morozov, A.V., Sapozhnikov, V.V., and Sapozhnikov, Vl.V., Logic Complement, a New Method of Checking the Combinational Circuits, *Automation and Remote Control*, 2003, vol. 64, no. 1, pp. 153–161.
14. Saposhnikov, V.V., Morosov, A., Saposhnikov, Vl.V., and Göessel, M., A New Design Method for Self-Checking Unidirectional Combinational Circuits, *Journal of Electronic Testing: Theory and Applications*, 1998, vol. 12, no. 1-2, pp. 41–53. DOI: 10.1023/A:1008257118423
15. Matrosova, A.Yu. and Mitrofanov, E.V., Delay Testable Sequential Circuit Design, *Tomsk State University Journal of Control and Computer Science*, 2013, no. 2 (23), pp. 140–147. (In Russian.)
16. Efanov, D.V. and Yelina, Y.I., Investigation of Ways of Synthesizing Concurrent Error-Detection Circuits Based on Boolean Signal Correction Using Uniform Separable Codes, *Russian Microelectronics*, 2024, vol. 53, no. 5, pp. 471–482. DOI: 10.1134/S1063739724600456
17. Efanov, D.V., Synthesis of Self-Checking Computing Devices Based on a Complete System of Special Groups of the Diagnostic Object Outputs, *Journal of Instrument Engineering*, 2023, vol. 66, no. 5, pp. 355–372. DOI: 10.17586/0021-3454-2023-66-5-355-372 (In Russian.)
18. Goessel, M. and Sogomonyan, E.S., Design of Self-Testing and Self-Checking Combinational Circuits with Weakly Independent Outputs, *Automation and Remote Control*, 1992, vol. 53, no. 8, pt. 2, pp. 1264–1272.
19. Sogomonyan, E.S. and Gössel, M., Design of Self-Testing and On-Line Fault Detection Combinational Circuits with Weakly Independent Outputs, *Journal of Electronic Testing: Theory and Applications*, 1993, vol. 4, no. 4, pp. 267–281. DOI: 10.1007/BF00971975
20. Freiman, C.V., Optimal Error Detection Codes for Completely Asymmetric Binary Channels, *Information and Control*, 1962, vol. 5, no. 1, pp. 64–71. DOI: 10.1016/S0019-9958(62)90223-1
21. MacWilliams, F.J. and Sloane, N.J.A., *The Theory of Error-Correcting Codes*, Amsterdam: North-Holland, 1977.
22. Borden, J.M., Optimal Asymmetric Error Detecting Codes, *Information and Control*, 1982, vol. 53, no. 1-2, pp. 66–73. DOI: 10.1016/S0019-9958(82)91125-1
23. Efanov, D.V., Compositions of Two Constant-Weight Codes with Orthogonal Combinations over All Bits for Self-Checking Discrete Device Design, *Control Sciences*, 2025, no. 3, pp. 41–61.
24. Efanov, D., Osadchy, G., and Zueva, M., Special Aspects of Errors Definition via Sum Codes within Embedded Control Schemas Being Realized by Means of Boolean Complement Method, *Proceedings of 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2021)*, Cracow, Poland, 2021, vol. 1, pp. 424–431. DOI: 10.1109/IDAACS53288.2021.9660837
25. Aksenova, G.P., Necessary and Sufficient Conditions for the Synthesis of Completely Testable Modulo 2 Convolution Circuits, *Automation and Remote Control*, 1979, vol. 40, no. 9, pp. 1362–1369.
26. Efanov, D.V., Self-Checking Combinational Devices Synthesis Based on the Boolean Signal Correction Method Using Bose—Lin Codes, *Information Technologies*, 2023, vol. 29, no. 10, pp. 503–511. DOI: 10.17587/it.29.503-511 (In Russian.)
27. Zakrevskii, A.D., Pottosin, Yu.V., and Cheremisinova, L.D., *Logicheskie osnovy proektirovaniya diskretnykh ustroistv* (Logical Foundations of Discrete Device Design), Moscow: Fizmatlit, 2007. (In Russian.)
28. Parkhomenko, P.P. and Sogomonyan, E.S., *Osnovy tekhnicheskoi diagnostiki. Tom 2: Optimizatsiya algoritmov diagnostirovaniya, apparaturnye sredstva* (Foundations of Technical Diagnosis. Vol. 2: Optimization of Diagnostic Algorithms, Hardware Means), Moscow: Energoatomizdat, 1981. (In Russian.)
29. Lala, P.K., *Self-Checking and Fault-Tolerant Digital Design*, San Francisco: Morgan Kaufmann Publishers, 2001.
30. Chioktour, V. and Kakarountas, A., Adaptive BIST for Concurrent On-Line Testing on Combinational Circuits, *Electronics*, 2022, vol. 19, no. 11, pp. 1–20. DOI: 10.3390/electronics11193193
31. McElvain, K., *IWLS'93 Benchmark Set. Version 4.0*. Distributed as a Part of IWLS'93 Benchmark Set, 1993.
32. *Collection of Digital Design Benchmarks*. URL: https://ddd.fit.cvut.cz/www/prj/Benchmarks/ (Accessed August 26, 2025.)
33. Hahanov, V., Litvinova, E., Chumachenko, S., et al., Vector Logic Modeling Self-Tested Circuits, *Proceedings of the 21st IEEE East-West Design & Test Symposium (EWDTS'2025)*, Tbilisi, Georgia, 2025. DOI: 10.1109/EWDTS67441.2025.11303706
34. Efanov, D.V., Using the "1-Out-Of-4" Constant-Weight Code in the Synthesis of Self-Checking Concurrent Error-Detection Circuit Based on Boolean Signal Correction, *Tomsk State University Journal of Control and Computer Science*, 2025, no. 72, pp. 114–133. DOI: 10.17223/19988605/72/12 (In Russian.)
35. Zakrevskij, A.D. and Toropov, N.R., Minimization of Boolean Functions of Many Variables – Iterative Method and Program Realization, *Applied Discrete Mathematics*, 2009, no. 1, pp. 5–14. (In Russian.)
36. Sentovich, E.M., Singh, K.J., and Moon, C., Sequential Circuit Design Using Synthesis and Optimization, *Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers & Processors*, Cambridge, 1992, pp. 328–333. DOI: 10.1109/ICCD.1992.276282
37. *SIS: A System for Sequential Circuit Synthesis*, Sentovich, E.M., Singh, K.J., Lavagno, L., Moon, C., Murgai, R., Saldanha, A., Savoj, H., Stephan, P.R., Brayton, R.K., and Sangiovanni-Vincentelli, A., Berkeley: Electronics Research Laboratory, Department of Electrical Engineering and Computer Science, University of California, 1992.
38. Bibilo, P.N., *Binarnye diagrammy reshenii v logicheskom proektirovanii* (Binary Decision Diagrams in Logical Design), Moscow: LENAND, 2024. (In Russian.)
39. Efanov, D.V. and Yelina, Y.I., Design of Self-Checking Digital Devices with Boolean Signals Correction Using Weight-Based Bose–Lin Codes, *Control Sciences*, 2024, no. 4, pp. 22–36. DOI: http://doi.org/10.25728/pu.2024.4.3

40.Korshunov, A.D., Computational Complexity of Boolean Functions, *Russian Math. Surveys*, 2012, vol. 67, no. 1, pp. 93–165. https://doi.org/10.1070/RM2012v067n01ABEH004777

This paper was recommended for publication by L.Yu. Filimonyuk, a member of the Editorial Board.

Received September 14, 2025, and revised November 27, 2025. Accepted November 27, 2025.

Author information

Efanov, Dmitry Viktorovich. Dr. Sci. (Eng.), Peter the Great Saint Petersburg Polytechnic University, St. Petersburg, Russia; Solomenko Institute of Transport Problems, Russian Academy of Sciences, St. Petersburg, Russia
✉ TrES-4b@yandex.ru
ORCID iD: https://orcid.org/0000-0002-4563-6411

Yelina, Yeseniya Igorevna. Postgraduate, Peter the Great Saint Petersburg Polytechnic University, St. Petersburg, Russia
✉ eseniya-elina@mail.ru
ORCID iD: https://orcid.org/0009-0004-4167-3591

Cite this paper
Efanov, D.V. and Yelina, Y.I., Design of Self-Checking Discrete Devices Based on Boolean Signals Correction and Composition of Constant-Weight Codes of the "1-Out-Of-4" and "3-Out-Of-4" Types. Part I: The Design Method with Conversion of All Signals from the Object under Diagnosis. *Control Sciences* 1, 30–40 (2026).

Original Russian Text © Efanov, D.V., Yelina, Y.I., 2026, published in *Problemy Upravleniya*, 2026, no. 1, pp. 34–46.

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 LARGE LANGUAGE MODELS IN DECISION SUPPORT SYSTEMS. PART I: Explanation Models and Large Language Models

A. A. Kulinich

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

✉ alexkul@rambler.ru

Abstract. Large language models (LLMs) significantly influence many spheres of life: education, creativity, science, and business. This paper considers the use of LLMs to explain alternative solutions obtained by a decision support system under uncertainty. Classical and pragmatic models of explanation proposed by philosophers are discussed. The goals and tasks of explanation in decision support processes under uncertainty are formulated. The operation of LLMs is conceptually analyzed, and their current capabilities in solving typical test tasks are assessed. The main techniques of prompting (a system of queries to a language model) are considered; with these techniques, a language model can be tuned to generate explanations for alternative solutions to particular tasks in a subject area. Finally, prompt techniques for supporting pragmatic and classical theories of explaining alternative solutions are considered.

Keywords: decision support, explanation models, explanation goal, explanation tasks, large language model (LLM), prompt techniques.

INTRODUCTION

To support decision-making in social, political, economic, and organizational systems under uncertainty, "soft" system analysis is used. It is based on the principle of bounded rationality [1]: due to the limited cognitive resources of human model developers, an object's simplified system model can be formed with a hypothetical structure, and its parameters may have linguistic values [2]. Within soft system analysis, model building involves literature examination and expert assessments, i.e., the source data are unstructured data (free text). A model describes an object in a limited natural language proposed by an expert.

Cognitive maps are a well-known mathematical tool of soft system analysis [3, 4]. The results of such a model are presented in an expert's limited natural language and interpreted by an expert in terms of his/her knowledge, which may be insufficient to obtain a new solution. An interpretation should include relevant knowledge of a subject area that, however, lies outside the expert's limited language and a simplified model of the situation. By interpreting the results of modeling, an expert attempts to explain and link the model processes with real-world ones. In a situation model, causal chains of inference can be formally constructed to explain the result [5]. However, these chains of reasoning are expressed in a limited natural language and can ensure the rigor of inference within the accepted assumptions and restrictions, but not the practical applicability of the solution. The real world is much more diverse and richer.

In psychology, decision-making is believed to be carried out by human intelligence in a psychological environment called a mental space. A mental space reflects the knowledge and life experience of an expert gained throughout his/her life. All mental operations—reasoning, generalization, interpretation, explanation, and assessment of possible solutions—take place in a mental space. These operations of decision-making under uncertainty can be supported by artificial intelligence (AI) systems, particularly large language models (LLMs).

Nowadays, both the number and variety of LLMs are growing rapidly. LLMs GPT-2 [6], GPT-3 [7], InstructGPT [8], and GPT-4 [9] from OpenAI, as well as BERT [10], RoBERTa, and ALBERT [11] from Google DeepMind, have become widespread and are

intensively used. Developments from Meta AI*, such as LLaMA (Large Language Model Meta AI) [12], are also actively applied. DeepSeek [13], the Chinese language model, is very popular. In Russia, LLMs GigaChat-2 from Sber [14] and Yandex GPT 5.1 from Yandex [15] are widely used. The state-of-the-art advances in the field of LLMs were comprehensively reviewed in [16].

LLMs are applied to solve various tasks in economics, the social sphere, medicine, education, public administration (at the federal, regional, and municipal levels), etc. We mention some reviews of the application of such models: in scientific research [17], for forecasting complex economic systems in modern conditions [18], in public administration [19], in healthcare [20], in organizations and the banking sector [21, 22], as digital assistants [23], and in business analytics and decision-making [24].

A methodology for using LLMs to reference and generate texts in public administration tasks was proposed in [25]. According to the strategic direction in the field of digital transformation of public administration in the Russian Federation [26], it is necessary to introduce AI systems, big data, and the Internet of Things into the sphere of public administration.

Within the Strategic Foresight Session on Fundamental Research in the Field of AI [27], the following priority areas in this field were identified: machine learning architectures and algorithms, computing and data for AI, fundamental and generative models, human–AI interaction, and applied research for science, education, and the social sphere.

This multi-part study is focused on current issues of human–AI interaction in decision-making under uncertainty.

The goal of this study is to develop methods and approaches (models and algorithms) to support decision-making under uncertainty using LLMs.

The main tasks to be solved for achieving this goal are as follows.

First, it is necessary to understand whether the explanatory texts generated by a language model are explanations in the sense of philosophical theories of explanation, which form the fundamental basis of research methodology. To do this, we will analyze the existing explanation models and reveal their core essence.

The second task is to tune an LLM to generate accurate explanations of situations for decision support tasks under uncertainty in tandem with a decision-maker (DM).

The third task is to assess the quality of a decision support system (DSS) that includes a DM and an LLM as an assistant.

The study consists of two parts. In part I, we describe the first two tasks. In particular, three classes of philosophical explanation models accepted in research methodology are identified. With these classes of explanation models, it is possible to understand whether the text generated by an LLM represents an explanation from a scientific viewpoint and to attribute the text to one of the classes. The role, purpose, and tasks of explanation in decision processes under uncertainty are also defined here. Information on the current performance of LLMs from leading companies is provided. Based on this information, it is possible to judge the applicability of language models as assistants to DMs in decision processes. The possibilities of tuning an LLM using the techniques of prompting (a system of queries to a language model) to generate explanations in terms of the selected classes of explanation models and explanation tasks solved under conditions are determined.

Part I of the study presents the basic concepts and definitions that will be used in part II. The former will also be helpful for those trying to apply an LLM in a DSS for the first time.

Part II of the study will deal with measuring the quality of a cognitive DSS composed of a human (DM) and a language model. To determine the quality of such a system, it is necessary to measure a latent variable, i.e., the DM's satisfaction with an explanation given by a language model. Criteria for assessing the DM's satisfaction degree when solving the explanation tasks outlined in part I of the study will be formulated. A reasonable respondent model will be proposed. An example will be provided where the reasonable respondent model will be used to assess explanations of two Russian LLMs, namely, Sber's GigaChat 2.0 and Yandex GPT 5 Pro.

1. EXPLANATION MODELS IN RESEARCH METHODOLOGY

Philosophical, sociological, and mathematical dictionaries and encyclopedias offer various definitions of an explanation, ranging from its properties and structure to its role in scientific cognition. Here are some generalized definitions.

In the philosophical encyclopedia [28], an explanation is a line of reasoning whose premises contain sufficient information to deduce a description of a phenomenon being explained. An explanation is an answer to the following question: why does this phenomenon occur?

* Meta Platforms, Inc. has been recognized as extremist in Russia, and its activities are prohibited within the Russian Federation.

The function of an explanation is defined as that of scientific cognition, the revelation of the essence of an object under study; it is realized by comprehending the law the object obeys, or by establishing the links and relations that determine its essential traits [29].

In research methodology, an explanation is a cognitive procedure aimed at enriching and deepening knowledge of real-world phenomena by incorporating them into a structure of definite links, relations, and dependencies to reveal the essential traits of a given phenomenon [30].

In philosophy and research methodology, the challenge is to create a unified universal theory of explanation applicable in various fields of human activity, e.g., during knowledge acquisition in scientific disciplines such as physics, chemistry, biology, sociology, etc. Variants of the general theory of explanation being created are different explanation models proposed by different researchers or groups of researchers. The explanation models being developed are based on numerous examples of theories of the real world; in general, these models reflect the mechanisms of human intelligence in the processes of research, comprehension, and justification of empirical observations and facts [31].

It is important to understand the extent to which artificial LLMs can explain real objects or events in decision processes under uncertainty.

Note that there are many test programs (benchmarks) to verify and assess the capabilities of language models in the areas of general knowledge, logical reasoning, common sense, etc. It is topical to investigate the capabilities of a language model and assess its quality as an explanation system in decision support processes.

Let us consider the main explanation models proposed in the philosophy of science.

1.1. The Deductive-Nomological Model of Explanation

According to this model, a scientific explanation consists of two main elements: the *explanandum*, which is a statement "describing the phenomenon to be explained," and the *explanans*, which is "statements given to explain the phenomenon" [32]. For explanations (explanans) to explain the explained (explanandum), several conditions must be satisfied.

• The explanandum must be a logical consequence of the explanans, and the statements making up the explanans must be true [32]; i.e., the explanation must take the form of a deductive argument in which the explanandum is derived from the premises making up the explanans. This is the "deductive" component of the deductive-nomological model (DN model).

• The explanans must contain at least one "law of nature," representing a necessary premise for the inference in the sense that if this premise disappears, the inference of the explanandum will be invalid. This is the nomological component of the model [32]. ("Nomological" is a philosophical term that essentially means "lawful, legal") [32].

Formally, the explanation problem for this model (and for all models considered below) has the following most general statement.

An explanation model is a tuple $<\Lambda, Ex1, O, Ex2>$, where:

• Λ is a set of laws; C. Hempel [32] distinguished several classes of laws:

  ○ universal laws L_c that apply to all areas of knowledge, such as the laws of logic and mathematics;

  ○ particular laws L_s of separate sciences, e.g., the laws of chemistry, physics, biology, sociology, political science, etc.;

  ○ individual facts L_f, which obey more general laws.

Thus, in the DN model, the set of laws Λ includes three subsets of the laws defined above, i.e., $\Lambda = \{L_c, L_s, L_f\}$.

The concept of a law is ambiguous (law of nature, state law, etc.), so we will provide several definitions for further reasoning. In philosophy, a law is a necessary connection (interrelation, relationship) between events, phenomena, and also between the internal states of objects, determining their stability, development, stagnation, or destruction. In a philosophical sense, a law is understood to mean objective connections between phenomena and events that exist regardless of whether they are known to anyone or not [33].

A law is a statement, expressed verbally or mathematically, describing objectively existing relationships and connections between various scientific phenomena and objects [28]. A law is proposed as an explanation of facts and is recognized as consistent with them by the scientific community at a certain stage. A law whose validity has been established not from theoretical considerations but from experimental data is called an empirical law.

• $Ex1$ is the explainable, i.e., the situation or phenomenon that needs to be explained.

• $O = \{o_j\}$ is a set of facts that characterize the explainable situation $Ex1$.

• and finally, Ex2 is an explanation.

Definition 1. A DN explanation of a certain phenomenon or situation $Ex1$ is a mapping DN: $L_1(o_1),\ldots, L_k(o_k) \rightarrow Ex2$, where $L_i()$ is a law, $L_i(\mathbf{.}) \in \Lambda$; o_j are facts, variables of the law $L_i(\mathbf{.})$, all facts are true,

i.e., the probability $P(o_j) = 1$; and finally, *DN* is a procedure of deductive inference of the explained phenomenon $Ex1$ from the laws L_i and facts o_j. ♦

An explanation that satisfies Definition 1 is called a causal-nomological explanation. This definition implies the application of a deterministic law in the explanation. However, laws from the fields of quantum physics, biology (genetics), as well as sociology and other humanities, are probabilistic in nature. Therefore, the framework of DN explanations defines deductive-statistical and inductive-statistical explanations.

Definition 2. If, in a DN explanation, a general statistical law $L_i^*(.)\in\Lambda$ is used to define explanatory facts, such an explanation is called deductive-statistical (SN explanation). ♦

In this case, the explained object or situation $Ex1$ may include statistical characteristics of a random process (samples, etc.), the mean $M(o_j)$ and the standard deviation $\sigma(o_j)$, and the nomological probability of the explanation $Ex2(M(o_j), \sigma(o_j))$ must be defined.

In this case, the mapping of an explanation from Definition 1 is written as follows:

$$SN\text{: } L_1(o_1),\ldots, L_i^*(M(o_j), \sigma(o_j)),\ldots,$$
$$L_k(o_k) \rightarrow Ex2(M(o_j), \sigma(o_j)), L_i^*(.)\in\Lambda.$$

Definition 3. If at least one fact in a DN explanation is obtained by inductive reasoning (the subjective probability of this fact is defined), then the explanation is called inductive-statistical (IN explanation). ♦

It is believed that an inductive fact must have a certain subjective probability. Usually, the probability threshold is set at 0.5, i.e., $o_j | Pr(o_j) > 0.5$.

In this case, the mapping of an explanation from Definition 1 is written as follows:

$$IN\text{: } L_1(o_1),\ldots, o_j \mid Pr(o_j) > 0.5,\ldots,$$
$$L_k(o_k)\rightarrow Pr(Ex1), L_i(.)\in\Lambda$$

Unlike deductive-statistical explanations, which are based on a large number of observations of random variables and can be considered a real statistical law, inductive conclusions are formed from a small number of observations and are subjective. In the DN explanation model, inductive facts are not reliable without specifying their context and estimating their probability. Hempel [32] called deductive-statistical and inductive-statistical explanations statistically relevant and inductively relevant, respectively.

In the DN model, explanation validity criteria are proposed. The first is the explanation generalization criterion, and the second is the explanation symmetry criterion. Staying within the mathematical formulations of the DN model, we will define the generalization of explanation elements. In the most general form, generalization is the replacement of an explained concept $Ex1$ and explaining facts $(o_1,\ldots, o_k)$ with the names of the classes or categories they belong to. Let $Kl(Ex1)$ and $Kl(o_j)$ denote the names of the classes of the explained concept and facts, respectively.

Definition 4. A generalization of the DN explanation model is an explanation obtained by replacing the name of the concept $Ex1$ with the name of its class $Kl(Ex1)$ and replacing the names of the facts $(o_1,\ldots, o_k)$ with the names of their classes $(Kl(o_1),\ldots, Kl(o_k))$. ♦

Explanation validity criterion 1: a DN explanation *DN*: $L_1(o_1),\ldots, L_k(o_k)\rightarrow Ex2$ (see Definition 1) is valid (true) if its generalized DN explanation *DN*: $Kl(L_1(o_1),\ldots, Kl(L_k(o_k))\rightarrow Kl(Ex2)$ is true. ♦

It is possible to prove the validity of an explanation formally based on generalization if its underlying law refers to the general laws L_c and there is a mathematical model of the object or situation. In this case, there may exist a rigorous mathematical or logical proof of the DN explanation, e.g., in predicate logic, etc. However, in many cases (for instance, in the humanities), such models are absent or too abstract. Then this explanation validity criterion can be confirmed or rejected by an expert, via substituting the names of the classes of concepts of the explained and the facts into the explanation. This criterion allows one to assert the truth and universality of the law chosen for the explanation.

Another explanation validity criterion is the inference symmetry criterion.

Explanation validity criterion 2: A DN explanation *DN*: $L_1(o_1),..., L_k(o_k)\rightarrow Ex2$ (see Definition 1) is valid (true) if the converse DN explanation *DN*: $Ex2\rightarrow L_1(o_1),\ldots, L_k(o_k)$ is true. ♦

The existence of a converse explanation inference indicates its consistency. This validity criterion can be formally verified if a known universal law is used for the explanation, and the model of the object or situation is known as well. For this purpose, one can involve the mathematical apparatus of proof theory [34]. In other cases, the consistency of the deductive inference given in an explanation can be verified by an expert.

The DN explanation model is considered a classical model of scientific explanation of objects or situations. However, under the explanation validity criteria, this model turns out to be rather cumbersome and difficult to comprehend. A complete and detailed explanation obtained using the DN model is considered an ideal scientific explanation and is called the "hidden structure" of the explanation.

According to some philosophers, a detailed explanation is excessive and therefore so-called "explanation sketches" are acceptable [32]. By assumption, an incomplete explanation obtained, e.g., by generalizing an ideal explanation, will contain a component that explains the ideal explanation. In this case, the matter concerns explaining an ideal explanation in understandable terms, such as "common sense." *Common sense* is a set of skills, ways of thinking, and views of the surrounding reality that are developed, used, and shared by people in their everyday practical activities.

An explanation is commonly considered to be something that provides understanding. In this regard, one task of the theory of explanation is to identify the structural features of explanations that provide understanding. Understanding is a universal mental operation associated with the assimilation of new content and its incorporation into a system of established ideas and concepts [35]. It is believed that explanation sketches in commonsense terms can provide an understanding of a phenomenon.

Let common sense be certain empirical laws accepted by society. We add commonsense laws L_{cs} to the set of laws Λ in the DN model: $\Lambda = \{L_c, L_s, L_f, L_{cs}\}$.

Within the hidden structure strategy, an explanation can then be represented as follows.

Definition 5. A hidden structure explanation $Ex2^*$ of a DN explanation $Ex2$ of a certain object or situation is a mapping DN^*: $L_{cs}^{1}(L_1(o_1)),\ldots, L_{cs}^{k}(L_k(o_k))\rightarrow Ex2^*$, where $L_{cs}^{j}()$ are commonsense laws $L_{cs}^{j}(.)\in\Lambda$; $Ex2^*$ is a commonsense explanation; and finally, DN^* is a deductive inference procedure of the phenomenon being explained from commonsense laws and facts. ♦

A hidden structure explanation can be viewed as an interpretation of an ideal explanation of the DN model. The hidden structure explanation validity can be verified if a homomorphism relation is defined between the basic L_c or particular L_s laws and the commonsense laws L_{cs}, i.e., Φ: $L_c\rightarrow L_{cs}$ or Φ: $L_s\rightarrow L_{cs}$. Under a homomorphism, the elements and relations of the laws L_c, L_s are mapped to those of the commonsense laws L_{cs} while preserving causal relations. Under a homomorphism, the inverse mapping is not unique.

In this case, validity criteria 1 and 2 may fail. Such an explanation will be called empirical.

Philosophers believe that explanations in terms of commonsense laws (explanations based on similarities, analogies, metaphors, etc.) play an important supporting role when formulating an ideal scientific explanation. This fact is due to the ambiguity of the inverse mapping Φ^{-1}: $L_{cs} \rightarrow \{L_{ci}\}$, i.e., a set of rigorous DN explanations based on general laws $\{L_{ci}\}$ can be obtained from an explanation based on the commonsense laws (L_{cs}). This creates the possibility of choosing the best explanation model and stimulates the researcher's thinking.

1.2. The Unificationist Theory of Explanation

The following explanation model is based on the unificationist theory, which states that scientific explanations can be represented as a uniform description of various objects and phenomena. This theory defines a schematic proposition of the following form [36]: "For all X, if X is O and A, then X is P." Here, X is a variable (the explained); O, A, and $P$are variables and facts, e.g., some properties of the explained.

The left part of this proposition is the premise ($Prm = (\forall X(O, A))$), and the right part is the inference, i.e., explanation ($Inf = (X\rightarrow P)$). P. Kitcher introduced the concept of schematic arguments (sequences of schematic propositions) and a classifier to determine which arguments are premises/conclusions/inference rules [36].

Within his approach, a rule for filling (replacing) abstract variables (X, O, A, P) with subject variables (concepts from a subject area) is defined. A definition of an argument pattern is given, including arguments, their classification, and filling instructions. A pattern filled with subject variables is considered to be an explanation. Of course, it is possible to obtain many explanation patterns, but which one is correct?

Kitcher introduced the concept of an explanatory reserve [36], i.e., a set of argument patterns containing a set of beliefs shared by scientists at a particular time. To prove that a particular inference—an explanation pattern—is a sound or acceptable explanation, one needs to show its belonging to an explanatory reserve.

The inference procedure in this case is to substitute the values of the subject variable into the argument pattern and check the pattern's consistency with the explanatory reserve. If consistency holds, then the explanation is obtained; otherwise, the pattern is filled with new data. Strictly speaking, the schematic proposition of this model describes inference in first-order logic. In such logic, a premise explicitly contains a certain law used for inference. For example, in the statement "All humans are mortal" (this is a law), "Socrates is a human" (this is a fact), and "Socrates is mortal" (this is a conclusion), the inference is obtained based on the law.

In the unificationist explanation model, there is no explicit reference to a certain law. A reference to a law in an explanation provides its important property, i.e., the causality of the explanation. This led to philosophical debates about the correctness of such an explanation model. However, according to Kitcher, first, there

are examples of explanations without causal relationships, and second, the argument patterns of premises and conclusions should be a generalization of the historical experience of humans and society in various spheres of activity, reflecting their cultural traditions [36]. The long evolutionary process in society forms beliefs, particularly including causal relationships. All these social beliefs are added to the explanatory reserve of the model and are used to validate the explanation pattern. In general, this explanation model suffers from some drawbacks; but, agreeing with its author, we can consider social beliefs to be a certain empirical social law L_{so} and add them to the set of laws Λ of the classical explanation model to get its significant extension with $\Lambda = \{L_c, L_s, L_f, L_{cs}, L_{so}\}$.

Definition 6. An empirical explanation $Ex1^{emp}$ of a certain object or situation is a mapping DN^{emp}: $L_{so}^{1}(o_1),\ldots, L_{so}^{k}(o_k) \rightarrow Ex2^{emp}$, where $L_{so}^{i}()$ is an empirical social law $L_{so}^{i}(\mathbf{.}) \in \Lambda$; o_j are facts, variables of the law $L_{so}^{i}(\mathbf{.})$, all facts are true, i.e., the probability $P(o_j) = 1$; $Ex2^{emp}$ is an explanation based on empirical laws; and finally, DN^{emp} is a procedure for substituting an empirical explanation $Ex2^{emp}$ from the set of explanatory reserves ER, $Ex2^{emp} \in ER$. ♦

In this case, validity criteria 1 and 2 for an empirical social explanation can be applied by an expert. However, due to the empiricism of a law L_{so}, nothing can be said about the reliability of the explanation (or even its probabilistic assessment). The fulfillment of these criteria will indicate the correct and consistent inference of the explanation, but not its reliability. Various reference books and encyclopedias can serve as an explanatory reserve.

1.3. The Pragmatic Theories of Explanation

The explanation models discussed above neglect the role of humans in the explanatory process. There are pragmatic theories of explanation directly involving humans both in the process of generating explanations and in the process of consuming explanations (as end users).

An explanation theory contains "pragmatic" elements if they require a mandatory reference to facts about the interests, beliefs, or other psychological characteristics of those who give or receive the explanation, and a mandatory reference to the context in which the explanation arises [37].

The authors of the classical DN explanation model agree that pragmatic elements play some role in the process of providing or receiving explanations; but they believe that there is a non-pragmatic core of explanation, and its description is the main task of explanation theory [32].

Contemporary pragmatic theories of explanation are based on constructive empiricism, as outlined by American philosopher B. van Fraassen. According to his viewpoint [37], the goal of science is to construct "empirically adequate" theories yielding true or accurate descriptions of observable phenomena, and research is more about construction than discovery, i.e., building models that must be adequate to the phenomenon rather than discovering the truth related to the unobservable. "Science aims to give us theories that are empirically adequate; and acceptance of a theory involves as belief only that it is empirically adequate." [37]. Van Fraassen's "empirical adequacy" means the coincidence of the empirical manifestations of the theoretical model of a phenomenon and the phenomenon itself.

The theory of constructive empiricism complements the theory of scientific realism, which is based on rigorous inference, laws, and proofs. However, as claimed by van Fraassen, the rigor of inference in the theory of scientific realism is based on unproven postulates and axioms, which limits its application in the humanities. While elaborating the theory of constructive empiricism, the author referred to the model of semiotics—the science of signs and sign systems, in which a phenomenon observed is represented at three levels: syntactic, semantic, and pragmatic [38].

A rigorous logical explanation obtained within the theory of scientific realism can be represented as a syntactic abstract structure. In constructive empiricism, such logical structures are interpreted in semantic (meaningful) terms.

In constructive empiricism, a human is an observer of manifestations of reality, a constructor of an empirical model of reality, and its validator. Consequently, "accepting" a theory means only believing in its empirical adequacy [37]. The issues of applying the constructive approach to the study of social systems were considered in [39].

The explanation model proposed within the theory of constructive empiricism is a triple of the form

$$Q = <T_k, X, RL>,$$

where Q is a query; T_k is a topic or context; $X = \{X_1,\ldots, X_n\}$ is a set of contrasting responses, essentially alternative responses (classes of responses); and finally, RL is the relevance relation between the topic and the alternative response.

Let us comment on this model. An explanation is not just a set of judgments but a response to the query: "Why?" It always arises in a definite context and is defined by three factors. The first factor is the topic, i.e., what is being asked about (T_k). The second factor is the contrast class, consisting of a set of statements

alternative to the topic. The third factor is the relevance relation between the topic of the query and the contrast class, which defines what can serve as an explanation.

Suppose that for a particular topic T_k and a given set of alternative responses X, a set of pairs—topic and alternative response—can be formed by the direct product $T_k \times X$.

Definition 7. For a pair $(T_k, X_j) \in T_k \times X$, $X_j \in X$, there exists a relevance relation $RL \subseteq T_k \times X$ if the subject believes and is convinced that X_j responds to a query Q in the context T_k. In this case, the explained Q and the explanation X_j are empirically adequate. ♦

A statement X_j is relevant to a query Q if and only if there is a relation RL for the pair (T_k, X_j). A relevance relation cannot be established unambiguously. For example, consider the query Q = "Why does blood circulate in the body?" and two possible responses, X_1 = "Because the contraction of the heart causes blood to move through the arteries" and X_2 = "In order to deliver oxygen to all tissues of the body."

Clearly, there are two alternative responses (two contrasting classes), but it is impossible to establish a relevance relation between the topic of Q and the contrasting class X beyond the context of this query [37].

The query "Why?" identifies a certain problem and sets a definite context. At the same time, the response to it includes a theoretical context—a scientific explanation. For this context of the query, the empirically adequate response is X_1 = "Because the contraction of the heart causes blood to move through the arteries." Here, a causal relevance relation holds. However, if we change the context of the query, Q = "Why does blood circulate in the body?", the empirically adequate response will be X_2 = "In order to deliver oxygen to all tissues of the body," with a functional relevance relation.

It is believed that "…A scientific explanation is not a piece of pure science but an application of science. It is a use of science to satisfy certain of our desires; these desires are always specific to a certain context, but they are always desires for descriptive information. ... The exact content of the desire, and the appraisal of how well it is satisfied, varies from context to context." [37].

Summarizing the explanation models discussed, we emphasize the main differences between pragmatic explanation models and the family of DN models. Recall that the family of DN explanation models (called classical in the literature) was developed within the theory of scientific realism, and such models are characterized by the following properties:

– An explanation is based on laws (deterministic, statistical, or empirical).

– The inference of the explained through observable facts is based on laws and must be correct and true (reliable).

– An explanation reflects the objective laws of nature and/or society and is independent of the psychological characteristics of a subject (his/her interests, beliefs, desires, assessments, etc.).

The pragmatic theories of explanation are based on the theory of constructive empiricism and are characterized by the following properties:

• A subject is included in an explanation model. An explanation is formed considering the subject's psychological characteristics and the context formulated in his/her query.

• A response is formed based on the subjective empirical adequacy of the explanation. The statement regarding the empirical adequacy of an explanation is much weaker than that regarding the truth (reliability) of the explanation in DN explanation models. However, the empirical adequacy of an explanation satisfies the subject's research needs.

• An explanation contains scientifically grounded statements at the level of semantics—the meaning of the explained—rather than at the level of inference in terms accessible to a limited circle of narrow experts, as is done within the theory of scientific realism.

• An explanation is aimed at satisfying the needs and desires of a particular subject in his/her context of interest in order to obtain additional relevant information and expand his/her worldview (mental space).

The above two classes of models form competing theories of explanation, but they can complement each other in decision support processes.

Note that the literature provides explanation models for different subject areas, such as explanation models in sociology, engineering, and mathematics, explanation models of consciousness, etc. Some philosophical explanation models from research methodology have been considered above. Other models of scientific explanation can be found in philosophical literature, e.g., in the review [9]. Here, we briefly overview the well-known, qualitatively different classes of explanation models so that texts generated by an LLM can be expertly assigned one of the classes of scientific explanation. As a result, it becomes possible to assess the quality of the language model's explanations and, therefore, its utility for decision support tasks.

1.4. The Goal of Explanation in Decision Support

The literature on various competing explanation models is quite extensive. Here, researchers interpret the very concept of explanation in different ways, pay-

ing little attention to the goals of explanation—what explanations are used for [31]. Obviously, the goal of explanation is determined by the goal of scientific research, which depends on the object of study. The most commonly proposed goals of research are proof, structural analysis, the principle of operation (functioning), development prediction, control of a complex object, etc. The goal of explanation also depends on the field of research; e.g., in the practice of teaching mathematics, the following types of explanation are distinguished: explanation to justify, explanation of a problem solution, explanation to reveal meaning, and explanation to verify.

However, the core of different explanation models in different subject areas is the connection between explanation and understanding. It is believed that explanation provides understanding. Understanding is a universal mental operation associated with the assimilation of new content (new content is included in the explanation) and its inclusion in a system of established ideas and beliefs (in essence, in the subject's knowledge system) [35].

In fact, an explanation is a description of the main properties of an object or situation under study (possibly linking them to known laws) that triggers a universal mental operation—understanding—to integrate this new information into the subject's current knowledge.

According to A.M. Sokhor's connection between understanding and explanation [40], those who understand the explanation constantly and easily move from the real object to its "ideal" model and back can repeat all the cognitive operations of the explainer and understand why these particular operations have been performed in the corresponding sequence. (It is the "ideal" model of an object that is created by abstractions.) Thus, explanation is a bridge linking the real world and the mental processes of human thinking and is intended to form an abstract, idealized mental model of an object or situation. The mental model can then be expressed in a rigorous mathematical language and become a law.

In this case, the goal of explanation can be formulated somewhat differently as applied to decision support. First, an explanation is aimed at forming an information environment of an object or situation, including the parameters of the object or phenomenon, its structure, belonging to a certain class (classification), etc., in the mental space of the subject. In this case, there are various relevance relations of an explanation that meet the subject's needs and the nonrigorous empirical adequacy of the explanation.

Second, an explanation is aimed at forming or identifying causal relations, rigorous inference based on known or newly discovered laws or regularities, for making decisions to control an object or situation.

Thus, the goal of explanation in decision support processes has two components:

– The research component is aimed at creating an information environment for decision-making and developing alternative solutions.

– The practical component is aimed at adopting, justifying, and implementing alternative solutions to control an object or situation.

In this case, competing explanation models—the classical (DN model) and pragmatic ones—complement each other. When treating decision-making as a research process, we can identify two stages of this process. The first stage—research—is a series of sequential queries and responses/explanations to form the information environment necessary for elaborating alternative solutions. At this stage, pragmatic explanation models are appropriate, as they yield empirically adequate explanations of alternative solutions without restricting the researcher's desires and needs.

The second stage is to select the best alternative using the DN explanation model, which justifies alternatives based on known laws and rigorous inference.

2. LARGE LANGUAGE MODELS AND ASSESSMENT OF THEIR CAPABILITIES

LLMs are deep learning neural networks trained on huge arrays of text data. They are based on the transformer architecture, which includes a set of neural networks consisting of an encoder and a decoder. Transformer is a neural network architecture designed for natural language processing and many other machine learning tasks [41]. An encoder is an element of the transformer architecture that converts a text corpus into vector representations: each word is described by a vector of probabilities of its joint use with other words in the text corpus, and the information about the structure and relationships between words is preserved. A decoder uses the encoded information to generate a response or make predictions. It "decodes" the received data, creating new text sequences considering previous words, context, and the probabilities of the joint use of words in that context.

The neural network of an LLM is a multilayer structure where each layer consists of a set of artificial neurons connected to neurons of neighboring layers. The number of layers in different LLMs varies and can reach several dozen or more. The neural network is trained with a large text corpus to form a dictionary of

tokens, i.e., individual letters or their groups, words, phrases, and sentences with a defined probability of their joint use in a given context. When encoding input information and transferring it between layers, an attention mechanism is used to reveal important elements of the language structure. Attention mechanisms can identify different types of syntactic relations between words, actually separating the syntactic structure of a sentence [42]. During training, a vector space of contextual token vectors is formed, and the (cosine) measure of proximity is determined for tokens. When decoding, tokens (words, phrases, and sentences) close to the query's topic are selected from the vector space to form sentences responding to this query. Selecting tokens from the vector space that are close to the query tokens allows generating a response from the words that frequently occur in the context of this query.

Thus, an LLM produces a statistical response to a query without working at the semantic level or analyzing the meaning of the query and response. The meaning of a response depends on the quality of a corresponding text corpus. If the corpus includes texts from verified sources, the response will be meaningful and understandable.

If a language model does not understand a query or has been trained with incomplete or incorrect data, it tries to guess the response based on existing syntactic patterns, which may lead to false responses. This behavior of an LLM is called hallucination. According to the hallucination phenomenon, LLMs generate factually incorrect or fictitious data that are not based on real information. This behavior is also characteristic of people when, under deficient information or time constraints for problem analysis, they form random or fictitious responses that are far from reality.

The problem of hallucinations in LLMs is crucial in applications with critical requirements for reliable information. Currently, various metrics have been introduced to assess the tendency of language models to hallucinate. Also, several methods for reducing language model hallucinations have been developed: the creation of correct text corpora (manual verification and text cleaning); the formation of correct queries to a language model, prompting the correct response; retraining a language model; and searching for accurate and relevant information in external sources.

Note that language model hallucinations in research and creative tasks can stimulate human intuition and lead to original solutions/decisions. Recall that heuristic methods for solving creative tasks (such as brainstorming and synectics) do not reject absurd and counterproductive alternative solutions as potential intuition impulsion to generate new and original solutions.

Despite this drawback, language models are useful owing to their capability to solve many natural language-written tasks that could previously be settled only by humans.

LLMs are assessed by special programs (benchmarks), which measure their basic qualities: knowledge volume, response accuracy, reliability, etc.

A benchmark uses definite datasets, metrics, and assessment tasks to test a language model, allowing one to compare different models and measure their accuracy. There are benchmarks to test knowledge, logical thinking, reading comprehension, common sense, etc. [43]. Many benchmarks have been developed; let us consider some of them.

Knowledge benchmarks test models in various fields. They assess how effectively a model can recall information from different areas, such as physics, geography, etc. MMLU (*Measuring Massive Multitask Language Understanding*) is a well-known benchmark created to check the model's factual knowledge on various topics, such as the humanities, social sciences, history, computer science, and even law. It includes 57 queries and 15 000 tasks designed to verify the high capabilities of a language model. On this benchmark, GPT-4-omni correctly responded to 88.7% of the queries asked.

Logical thinking benchmarks test the model's capability to "think" step by step and make logical conclusions (inference).

Benchmarks for assessing the mathematical capabilities of language models are used as well. For example, the GSM8K test consists of 8500 middle-school math tasks. Solving them requires the model to perform several steps of elementary calculations. Language models pre-trained for mathematical reasoning perform well on this benchmark; for example, GPT-4 achieves an accuracy of 96.5%.

GPQA (*Graduate-Level Google-Proof Q&A Benchmark*) assesses the logical thinking of a language model using a dataset of only 448 queries. This difficult test, developed by experts in biology, physics, and chemistry, was passed by GPT-4-omni with an accuracy of 53.6% only, while graduate students achieve 65%.

Reasoning language models are being developed by many leading companies. In 2024, OpenAI released a new language model, OpenAI-o1, which demonstrates excellent results in complex reasoning, outperforming humans in tests in mathematics, coding, and natural sciences. In the qualifying round for *the International Mathematical Olympiad* (IMO), this model succeeded in solving 83% of the tasks, while its predecessor (GPT-4o) gave only 13% correct responses. According to the developers from OpenAI, when solv-

ing complex test tasks in physics, chemistry, and biology, the model demonstrates results comparable to those of postgraduates.

Reading comprehension benchmarks test the model's capability to interpret natural language and generate appropriate responses. The test is to respond to queries about texts, which allows assessing comprehension and the capability to make conclusions, grasp, and remember important details.

DROP (*Discrete Reasoning Over Paragraphs*) is a benchmark for testing reading comprehension. It challenges models to reason based on their analysis of paragraph content. This benchmark includes 96 000 queries to test the reasoning capabilities of a language model. DROP's queries contain information that requires models to perform mathematical operations such as addition, subtraction, and comparison based on information scattered throughout the text. When responding to these complex queries, GPT-4 achieved an accuracy of 80%, while humans give 96% correct responses on the DROP dataset.

Commonsense benchmarks assess the model's capability to generalize knowledge of the world. Such test sets typically include queries that require extensive encyclopedic knowledge to respond correctly. Commonsense testing in language models assesses the model's capability to make judgments and conclusions consistent with human thinking. Humans form a holistic view of the world through practical experience, while language models are trained on huge datasets without understanding context.

HellaSwag (*Harder Endings, Longer Contexts, and Low-Shot Activities for Situations with Adversarial Generations*) is a benchmark for checking the model's capability to predict a plausible continuation of a given scenario. According to HellaSwag testing results, modern models such as GPT-4 have achieved near-human levels of accuracy.

IFEval (*Inference and Fidelity Evaluation*) is a benchmark for assessing both the accuracy and quality of the generated text. First, a model is assessed by the Inference metric—the capability to generate text on a large volume of data. Then, the quality (Fidelity) of the generated text is assessed. The assessment includes checking the compliance of the generated text with the expected result and assessing the degree of preserving the meaning and structure of the text. Next, the final IFEval score is calculated, which reflects both the model's capability to generate text and the quality of that text. The higher the IFEval score is, the better the model will succeed in text generation.

The **HumanEval benchmark** is a reference dataset designed for objectively assessing the quality of code generated by AI models based on a text description of the task. The benchmark consists of 164 programming tasks manually written for this dataset to ensure their absence in the model's training samples. All tasks are formulated in Python and presented as code snippets with descriptions.

Table 1 presents performance estimates for LLMs from leading developers: GigaChat 2 MAX from the Russian company Sber; Qwen 2.5 72B from leading e-commerce platform Alibaba; Llama 3.3 70B developed by Meta AI; GPT-4o developed by OpenAI; DeepSeek-V3 from Chinese company DeepSeek, owned by the High-Flyer fund; and Yandex GPT5 Lite Instruct from Russian company Yandex. Testing was conducted on benchmarks in the following categories: general knowledge, mathematics, code work, and text generation quality. The data are current [44, 45] as of early 2025. The figures in the table show the percentage of tasks solved.

Testing the capabilities of language models using benchmarks allows objectively assessing and comparing the quality of models from different suppliers and understanding the current level and dynamics of language model development. Language model developers believe that the capabilities of these models are approaching those of humans; hence, they can assist in decision support systems.

However, practical tasks, such as making decisions in complex economic, political, or social situations, require decision-makers to have a comprehensive combination of all possible human intelligence capabilities. Therefore, automated metrics alone cannot cover the entire spectrum of language model assessment, especially when it comes to the subjective aspects of language comprehension and generation. Here, human assessment is much more accurate.

In this case, it is advisable to involve experts or a group of experts to provide an accurate and reliable assessment of the capabilities of LLMs [43], e.g., based on a survey.

Some problems arise in the expert assessment of language model explanations. Here, it is necessary to assess the interaction between a human and an LLM. A language model gives a statistically plausible explanation without understanding its meaning, while a human tries to understand the explanation and integrate it into his/her knowledge system. A human understands a language model's explanation depending on his/her level of knowledge; and therefore, different people, receiving the same response from a language model, may give it different assessments.

In this case, it is required to develop a method for assessing the individual's (non-group) satisfaction with the explanations of a decision situation that he/she receives from an LLM. The solution to this task will be discussed in part II of the study.

Table 1

Performance of LLMs from leading developers

Category	Benchmark name	GigaChat 2 MAX	Qwen 2.5 72B	Llama 3.3 70B	GPT-4o	DeepSeek-V3	Yandex GPT5 Lite Instruct
General knowledge	MMLU (RU)	80.46	78.30	65.08	80.00	73.74	70
	MMLU (EN)	86.00	83.85	78.57	88.7	85.24	75.8
Mathematics	GSM8K	95.68	95.07	92.87	95.00	94.99	87.9
	MATH	77.26	78.74	62.80	76.60	85.48	82.0
Code work	HumanEval	87.20	86.60	86.0	84.00	91.46	71.8
Quality of text generation	IFEVAL (RU)	83.62	84.27	75.12	80.24	84.37	76.9
	IFEVAL (EN)	89.99	90.43	90.83	88.51	92.21	72.6

3. EXPLANATION MODELS AND LARGE LANGUAGE MODELS

The explanation models discussed above form two classes of models: classical (deductive-nomological) models, based on rigorous inference and known objective laws or regularities, and pragmatic models, which consider the goals and desires of the researcher or explainer and the context of the query to be explained.

In these models, a human formulates an explanation based on observable facts and knowledge. Obviously, the knowledge of a human researcher necessary to formulate a response-explanation will be different in each class of explanation models. Benchmarks show the good capabilities of language models in different areas. Can a language model provide an explanation satisfiable for a DM?

The knowledge of a language model is determined by the content of the text corpus used to train it. Therefore, a language model will operate in the class of classical explanation models if it has been trained in the known laws of logic, mathematics, physics, chemistry, biology, etc. In this case, the model is likely to generate rigorous logical explanations based on known laws. The reliability of such inference can be verified, and the logical conclusions and explanations themselves can be used to control a situation.

A language model will operate in the class of pragmatic explanation models if it has been trained in general knowledge, commonsense laws, traditions, etc. According to the theory of constructive empiricism, the resulting empirically adequate explanations are based on the researcher's beliefs. Such explanations are useful for expanding the mental space of a DM, increasing his/her awareness, and stimulating his/her intuition to generate non-trivial creative decisions.

Training language models to obtain classical explanations or pragmatic explanations from scratch is a very costly procedure. First, it involves collecting, preparing, and cleaning a very large training text corpus relevant to a given task. As a rule, text corpora are prepared by special outsourced teams. The dictionaries of modern language models contain billions of tokens. As is believed, only very large language models exhibit emergent properties, such as the capability to reason. Second, training a neural network requires very high computing power—supercomputers—and, accordingly, significant time and energy. Training a language model from scratch is only affordable for large companies, which allow other developers to use trained models in their tasks.

Pre-trained language models that are publicly available can be retrained with particular texts, focusing them on the solution to particular tasks. In this case, the stages of preparing the text corpus and training the model are also present, but at a lower cost.

An LLM is a huge neural network with a huge number of connections between tokens. Each query to a network activates a definite chain of tokens, forming a response. Also, there are a huge number of options for activating token chains, and therefore, the number of plausible alternative responses is also large.

Currently, *Prompt Engineering* (PE) is being actively developed as an alternative tuning method for an LLM for particular tasks using prompts [46, 47]. A prompt is a hint or sequence of hints for a language model. Prompts are represented as an algorithm or scenario for solving a user task; they are created by the user and can be embedded in a language model. Thus, prompts control the operation of a language model to obtain the best result.

3.1. Techniques of Prompting and Explanation Models

Consider some of the most popular techniques of prompting.

Zero-shot. This technique of prompting implies no input or output examples to retrain a language model. It is a simple query to a language model, and the latter can respond by providing information that does not require multi-step instructions.

Few-shot. This technique of prompting involves examples: several examples of input and output data are used to retrain a language model. With these examples, a language model learns to generate queries following the patterns. For example, a model can be retrained to detect the tone (emotional "color") of texts accurately. Few-shot prompting can be employed as a technique for the contextual retraining of a language model. Examples specify the context of an expected response.

Role-based. This technique of prompting is to assign a role to a language model when it responds to a given query. For example, a model is asked to respond to a query from the standpoint of an economist, financier, lawyer, etc. The query is the same, but the responses will be generated in the contexts of different subject areas [48].

Chain-Of-Thought (CoT). This technique of prompting forces a language model to think step by step. In this technique, a user gives a language model an algorithm for solving a task. It makes the model's thinking closer to human thinking. As a rule, a complex task is decomposed into subtasks, and their sequential solution makes the overall solution more accurate. The application of this technique in LLMs significantly improves the accuracy of solving mathematical and logical problems of the GSM8K benchmark.

According to the authors, CoT leads to higher-quality solutions to linguistic tasks as well as general knowledge and commonsense tasks. This popular technique of prompting, in various modifications, is widely used in reasoning language models [49].

Chain-of-Verification (CoV). It complements CoT by forcing a model to verify all previous steps before taking the next step, making reasoning language models more reliable [50].

Chain-of-Note (CoN). With this technique of prompting, a language model is forced to make "notes" in the process of solving a task, thereby explaining each successive step. As a result, the hallucinations of a language model can be detected [51].

Chain-of-Knowledge (CoK). In this technique, a prompt necessarily includes verified knowledge of a subject area, and a language model can use it to solve a given task. Unlike CoT, a language model relies on known facts, laws, or regularities provided in a user's prompt and then builds a logical chain from them, leading to a specific and reasonable answer. For example, a user includes the laws of physics in the prompt to solve a physics problem, and a language model will generate the solution using these laws. The correctness of the response will be higher than in (simple) CoT [52].

Tree of Thoughts (ToT) [53]. This technique is used to solve complex research tasks or strategic planning tasks when conventional or simple methods of prompting are insufficient. ToT was proposed in [53] as a generalization of CoT. In this technique, attention is paid to the study of responses, which serve as intermediate steps for solving tasks by language models. ToT supports the construction of a response tree (tree of thoughts) whose nodes contain intermediate response texts. Then search algorithms, such as breadth-first search and depth-first search, are applied to the tree to obtain a solution and analyze it.

ReAct Prompting. In this technique of prompting [54], a language model is used in interactive mode with a human to generate chains of reasoning in order to form relevant actions for a given task. With generated chains of reasoning, a language model is able to create, track, and update action plans, as well as to handle error situations. Actions implement interaction with external information sources, e.g., knowledge bases.

ReAct allows a language model to interact with external sources to obtain additional information, resulting in more reliable and accurate responses. ReAct improves the interpretability and reliability of a language model. According to the authors, the best approach is to apply ReAct in combination with CoT; in this case, both internal knowledge and external information obtained by reasoning are used.

Commonsense prompts. Currently, prompts for supporting deductive reasoning based on commonsense knowledge are of great research interest. The fact is that LLMs are trained on large text corpora, and when answering queries, they give statistically better responses. Such responses are correct, but they usually have no elements of deductive reasoning (obvious to humans) and are represented as implicit knowledge. Prompts are developed to obtain a language model's response containing an explanation with deductive reasoning; they are hints for a language model to extract a meaningful answer. Such hints for a language model are generated using external commonsense knowledge bases. These are static knowledge bases ConceptNet [55] and ATOMIC [56], as well as dynamic commonsense knowledge base COMeT [54], in which commonsense knowledge is formed from the context of a query to a language model and is represented in the form of a reasoning graph. Experiments with this dynamic knowledge base [57] have shown an increase in the performance of language models on commonsense benchmarks. In the paper [58], a method was described to generate knowledge from a language model itself and provide this knowledge as additional input data (hints) when answering a query. This method does not require access to a structured commonsense knowledge base but improves the performance of modern LLMs. The is-

sues of generating commonsense responses by a language model based on internal dialog with a language model were considered in [59]. The corresponding method involves no external knowledge bases, but the performance of a language model on commonsense benchmarks with such a prompt has demonstrated an increase. However, according to the authors of [59], the disadvantage of LLMs is the lack of introspection, i.e., knowledge of their knowledge.

Only some techniques of prompting have been discussed above. Currently, many developers offer custom prompts that can be embedded in pre-trained language models and do not require additional training in a subject area. This seems to be an inexpensive option for tuning a language model to solve user tasks.

In decision support, there are heuristic techniques of analyzing situations under uncertainty, which are methods for finding decisions in complex situations. For example, they include Five Whys, aimed at identifying the causes of a problem; the Ishikawa (fishbone) diagram, intended to build a hierarchical model of cause-and-effect relations; the garland and association method, serving to generate creative decisions; SWOT analysis, allowing one to determine the development strategy of an organization, etc. All these techniques have been tested in decision practice, and their algorithms can be considered a prompting algorithm for an LLM.

A prompt can be represented as a control program that extracts the information necessary for decision-making from a language model via sequential steps.

The above techniques of prompting can help a language model implement the explanation models. Let us consider this issue in detail.

For pragmatic explanation theories focused on satisfying the user's desires and interests, suitable techniques are those organizing a dialog with a language model to expand the user's mental space and awareness. Recall that in pragmatic explanation models, there shall exist a relevance relation between the topic of a query and the language model's explanation, and the DM shall be convinced that the explanation is empirically adequate.

The techniques of prompting from Table 2 can be applied to support pragmatic explanation models.

Note that Few-shot allows implementing the unificationist explanation model [36]. Recall that in this model, a pattern is defined, and an explanation selected from an explanatory resource is substituted into this pattern. An explanatory resource is an opinion of prominent scientists, customs, traditions, and information from reference books, encyclopedias, etc.

For classical explanation theories (DN models [32]), aimed at obtaining and justifying an explanation in the form of decision inference based on known laws and regularities, suitable techniques of prompting are provided in Table 3.

We have mentioned two goals of explanation in decision support. These are the research goal, which is achieved by using a language model with prompts implementing pragmatic explanations to increase expert awareness, and the practical goal, which is achieved by using a language model with prompts for classical explanation theories that help obtain and justify rigorous decision inference and its application to control a situation.

The application of different combinations of the existing techniques of prompting or the development of a custom prompt allows an LLM to be tuned for explaining alternative solutions in both the classical and pragmatic theories of explanation.

The issues related to the development and testing of user prompts for solving research and practical decision support tasks are not addressed in this work.

Table 2

The role of prompting in the pragmatic explanation model

The technique of prompting	The role of prompting
Zero-shot	Allows getting a response to any query
Few-shot	Allows getting a response in the form of a user-defined pattern
Role based	Allows getting responses to the same query in different contexts
Chain-Of-Thought (CoT)	Allows representing a complex task as a series of related subtasks and solving them sequentially
Chain-of-Verification	Extends CoT; allows verifying the reasoning of a language model and detecting hallucinations
Chain-of-Note	Extends CoT; allows improving the reasoning quality of a language model by supplementing explanations
Tree of Thoughts (ToT)	Allows building a tree of reasoning instead of a chain of thoughts in CoT; as a result, allows finding a better alternative explanation
ReAct Prompting	Enables an interactive mode between a user and a language model. The user can utilize external data to solve research and planning tasks

Table 3

The role of prompting in the classical explanation model

The technique of prompting	The role of prompting
Chain-of-Knowledge	Allows retraining a language model with a knowledge model that includes a description of the laws or patterns of a particular subject area. As a result, allows increasing the inference reliability of a language model in the subject area under study.
ReAct Prompting	Organizes an interactive mode of operation between a user and a language model with access to external sources of verified data. As a result, allows the use of descriptions of laws from external data sources, which can increase inference reliability
Commonsense	Allows getting a "sketch" of a classical explanation in commonsense terms. Such an explanation has been called a hidden-structure explanation of a deductive-nomological explanation (see above).

CONCLUSIONS

We have considered the application of LLMs for generating and explaining alternative solutions obtained by a decision support system under uncertainty. We have discussed classical (deductive-nomological) and pragmatic models of explanation proposed by philosophers. The goals and tasks of explanation in decision support processes under uncertainty have been formulated. The goal of explanation in a decision support system is to form an information environment for decision-making; it reduces to solving a research problem to formulate alternative solutions and a practical problem to justify and implement the best alternative. The operation of LLMs has been conceptually analyzed, and their current capabilities in solving typical test tasks have been assessed. The main techniques of prompting (a system of queries to a language model) have been considered; with these techniques, a language model can be tuned to generate explanations for alternative solutions to research and practical tasks of decision support under uncertainty.

In part I of the study, we have formulated the basic concepts and definitions that will be used in part II. In turn, part II of the study will be devoted to issues of measuring and assessing the satisfaction of decision-makers with explanations of LLMs. Satisfaction assessments will be conducted for research and practical tasks of explanation under uncertainty. The explanations of two Russian LLMs will be analyzed and assigned to the above classes of explanation models.

REFERENCES

1. Simon, H.A., Rationality as Process and as Product of Thought, in *Decision Making: Descriptive, Normative, and Prescriptive Interactions*, Bell, D.E., Raiffa, H., and Tversky, A., Eds., Cambridge: Cambridge University Press, 1988, pp. 58–77.
2. Checkland, P.B., *Systems Thinking, Systems Practice*, New York: Wiley, 1981.
3. Axelrod, R., *The Structure of Decision: Cognitive Maps of Political Elites*, Princeton: Princeton University Press, 1976.
4. Kosko, B., *Fuzzy Thinking: T*he New Science of Fuzzy Logic, New York: Hyperion, 1993.
5. Kulinich, A.A., Cognitive Maps Verification Based on Processes Explanation, *Large-Scale System Control*, 2010, no. 30.1, pp. 453–469. (In Russian.)
6. *Neural Network GPT-2 by OpenAI. Fast Start*. URL: https://habr.com/ru/articles/440564/ (Accessed February 16, 2026; in Russian.)
7. Brown, T., Mann, B., Ryder, N., et al., Language Models Are Few-Shot Learners, *arXiv:2005.14165*, 2020. DOI: https://doi.org/10.48550/arXiv.2005.14165
8. Ouyang, L., Wu, J., Jiang, X., et al., Training Language Models to Follow Instructions with Human Feedback, *arxiv.org/abs/2203.02155*, 2022. DOI: https://doi.org/10.48550/arXiv.2203.02155
9. Brown, A., GPT-4 Is OpenAI's Most Advanced System, Producing Safer and More Useful Responses, *International Journal of Architectural Computing*, 2024, vol. 22, no. 3, pp. 275–276. DOI:10.1177/14780771241280148
10. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K., Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding, *arXiv:1810.04805*, 2018. DOI: https://doi.org/10.48550/arXiv.1810.04805
11. Lan, Z., Chen, M., Goodman, S., et al., Albert: A Lite Bert for Self-Supervised Learning of Language Representations, *arXiv:1909.11942*, 2019. DOI: https://doi.org/10.48550/arXiv.1909.11942
12. Touvron, H., Lavril, T., Izacard, G., et al., Llama: Open and Efficient Foundation Language Models, *arXiv:2302.13971*, 2023. DOI: https://doi.org/10.48550/arXiv.2302.13971
13. Bi, X., Chen, D., Chen, G., et al., Deepseek LLM: Scaling Open-Source Language Models with Longtermism, *arXiv:2401.02954*, 2024. DOI: https://doi.org/10.48550/arXiv.2401.02954
14. *GigaChat-2*. URL: https://giga.chat/ (Accessed February 14, 2026; in Russian.)
15. *Yandex GPT 5.1*. URL: https://ya.ru/ai/gpt (Accessed February 14, 2026; in Russian.)
16. Vizil'ter, Yu.V., Priority Research Areas and Key Trends in the Development of AI Technologies, *Trudy 22-oi Natsional'noi konferentsii po iskusstvennomu intellektu s mezhdunarodnym uchastiem* (Proceedings of the 22nd Russian Conference on Artificial Intelligence with International Participation (RCAI-2025)), St. Petersburg, 2025, vol. 1, pp. 7–25. DOI: 10.15622/rcai.2025.001 (In Russian.)

17.Bragin, A., Bakhtizin, A., and Makarov, V., Large Fourth-Generation Language Models as a New Tool in Scientific Research, *Artificial Societies*, 2023, vol. 18, no. 1. DOI: 10.18254/S207751800025046-9 (In Russian.)
18.Bakhtizin, A.R., The Challenges of Forecasting under Current Conditions, *The Economic Revival of Russia*, 2023, no. 2 (76), pp. 53–62. (In Russian.)
19.Kommersant Learned about the Testing of Russian Language Models for Gosuslugi, *Forbes*, February 2, 2024. URL: https://www.forbes.ru/tekhnologii/505447-kommersant-uznal-o-testirovanii-rossijskih-azykovyh-modelej-dla-gosuslug?ysclid=lzfs8wut7z80399139 (Accessed February 14, 2026; in Russian.)
20.Gusev, A., A Review of Russian Artificial Intelligence Systems for Healthcare. URL: https://webiomed.ru/blog/obzor-rossiiskikh-sistem-iskusstvennogo-intellekta-dlia-zdravookhraneniia/ (Accessed February 14, 2026; in Russian.)
21.Shrestha, Y.R., Ben-Menahem, S., and von Krogh, G., Organizational Decision Making Structures in the Age of Artificial Intelligence, *California Management Review*, 2019, vol. 61, no. 4, pp. 66–83.
22.Gasanov, E., Decision Intelligence: Artificial Intelligence with a Human Face, *IT World*, March 22, 2022. URL: https://www.it–world.ru/cionews/1q746pou69z40kokwskckk4gkg4c0og.html (Accessed February 14, 2026; in Russian.)
23.Virtual Assistants, *TAdviser*, August 6, 2022. URL: https://www.tadviser.ru/index.php (Accessed February 15, 2026; in Russian.)
24.AI in Analytics: What is Beyond BI?, *TAdviser*, August 4, 2022. URL: https://www.tadviser.ru/index.php (Accessed February 15, 2026; in Russian.)]
25.Dudikhin, V.V. and Kondrashov, P.E., Methodology of Using Large Language Models to Solve Tasks of State and Municipal Government for Intelligent Abstracting and Automatic Generation of Text Content, *Public Administration*, 2024, no. 105, pp. 169–179. (In Russian.)
26.*Strategic Direction in the Field of Digital Transformation of Public Administration*. Approved by Decree no. 2998-r of the Government of the Russian Federation on October 22, 2021. (In Russian.)
27.Dmitry Chernyshenko Held a Strategic Foresight Session on Fundamental Research in the Field of Artificial Intelligence, *Official Website of the Government of the Russian Federation*, May 30, 2024. URL: http://government.ru/news/51726/ (Accessed February 16, 2026; in Russian.)
28.*Filosofiya: Entsiklopedicheskii slovar'* (Philosophy: Encyclopedic Dictionary), Ivin, A.A., Ed., Moscow: Gardariki, 2004. (In Russian.)
29.*Filosofskii entsiklopedicheskii slovar'* (Philosophical Encyclopedic Dictionary), Il'ichev, L.F., Fedoseev, P.N., Kovalev, S.M., and Panov, V.G., Eds., Moscow: Sovetskaya Entsiklopediya, 1983. (In Russian.)
30.*Novaya filosofskaya entsiklopediya* (New Philosophical Encyclopedia), Stepin, V.S., Ed., Moscow: Mysl', 2001. (In Russian.)
31.Woodward, J., Scientific Explanation, in *The Stanford Encyclopedia of Philosophy*, Zalta, E.N., Ed., Stanford: Stanford University, 2017. URL: https://plato.stanford.edu/archives/fall2023/entries/scientific-explanation/ (Accessed February 15, 2026.)
32.Hempel, C., *Aspects of Scientific Explanation and Other Essays in the Philosophy of Science*, New York: Free Press, 1965.
33.Regularity (law), in *Bol'shaya rossiiskaya entsiklopediya* (The Great Russian Encyclopedia), Osipov, Yu.S., Ed., Moscow: Great Russian Encyclopedia, 2004–2017. (In Russian.)
34.Dragalin, A.G., *Mathematical Intuitionism. Introduction to Proof Theory*, Translated by E. Mendelson, Providence, Rhode Island: AMS, 1988.
35.Understanding, in *Slovar' po logike* (Dictionary of Logic), Ivin, A.A. and Nikiforov, A.L., Eds., Moscow: Vlados, 1997. (In Russian.)
36.Kitcher, P., Explanatory Unification and the Causal Structure of the World, in *Scientific Explanation*, Kitcher, P. and Salmon, W., Eds., Minneapolis: University of Minnesota Press, 1989, pp. 410–505.
37.Van Fraassen, B., *The Scientific Image*, Oxford: Oxford University Press, 1980.
38.Morris, C., Foundations of the Theory of Signs, in *Writings on the General Theory of Signs*, Sebeok, T., Ed., The Hague: Mouton, 1971.
39.Plotinskii, Yu.M., *Modeli sotsial'nykh protsessov* (Models of Social Processes), Moscow: Logos, 2001. (In Russian.)
40.Sokhor, A.M., *Ob"yasnenie v protsesse obucheniya: Elementy didakticheskoi kontseptsii* (Explanation in the Learning Process: Elements of the Didactic Concept), Moscow: Pedagogika, 1988. (In Russian.)
41.Vaswani, A., Shazeer, N., Parmar, N., et al., Attention is All You Need, *arXiv:1706.03762*, 2017. DOI: https://doi.org/10.48550/arXiv.1706.03762
42.Marecek, D. and Rosa, R., Extracting Syntactic Trees from Transformer Encoder Self-attentions, *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Brussels, Belgium, 2018, pp. 347–349. URL: https://aclanthology.org/W18–5444.pdf/ (Accessed February 15, 2026.)
43.Barskaya, I., Benchmarks for LLMs, *Unite.AI*, August 28, 2024. URL: https://www.unite.ai/benchmarks-for-llms/ (Accessed February 15, 2026.)
44.*Sber Presented GigaChat 2.0, the New Generation of Its Neural Network*. URL: https://t-j.ru/news/sber-gigachat-2/ (Accessed February 15, 2026; in Russian.)
45.*YandexGPT-5-Lite-8B-instruct*. URL: https://huggingface.co/yandex/YandexGPT-5-Lite-8B-instruct (Accessed February 15, 2026.)
46.Sahoo, P., Singh, A.K., Saha, S., et al., A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications, *arXiv:2402.07927*, 2024. DOI: https://doi.org/10.48550/arXiv.2402.07927
47.*Prompt Engineering Guide*. URL: https://www.promptingguide.ai/ (Accessed December 16, 2023.)
48.Wang, Z., Peng, Z., Que, H., et al., RoleLLM: Benchmarking, Eliciting, and Enhancing Role-Playing Abilities of Large Language Models, *arXiv:2310.00746*, 2024. DOI: https://doi.org/10.48550/arXiv.2310.00746
49.Wei, J., Wang, X., Schuurmans, D., et al., Chain of Thought Prompting Elicits Reasoning in Large Language Models, *arXiv:2201.11903*, 2022. DOI: https://doi.org/10.48550/arXiv.2201.11903
50.Dhuliawala, S., Komeili, M., Xu, J., et al., Chain-of-Verification Reduces Hallucination in Large Language Models, *arXiv:2309.11495*, 2023. DOI: https://doi.org/10.48550/arXiv.2309.11495
51.Yu, W., Zhang, H., Pan, X., et al., Chain-of-Note: Enhancing Robustness in Retrieval-Augmented Language Models,

arXiv:2311.09210, 2023. DOI: https://doi.org/10.48550/arXiv.2311.09210
52. Liu, J., Lui, A., Lu, X., et al., Generated Knowledge Prompting for Commonsense Reasoning, *arXiv:2110.08387*, 2021. DOI: https://doi.org/10.48550/arXiv.2110.08387
53. Yao, S., Yu, D., Zhao, J., et al., Tree of Thoughts: Deliberate Problem Solving with Large Language Models, *arXiv:2305.10601*, 2023. DOI: https://doi.org/10.48550/arXiv.2305.10601
54. Yao, S., Zhao, J., Yu, D., et al., React: Synergizing Reasoning and Acting in Language Models, *arXiv:2210.03629*, 2023. DOI: https://doi.org/10.48550/arXiv.2210.03629
55. Speer, R., Chin, J., and Havasi, C., Conceptnet 5.5: An Open Multilingual Graph of General Knowledge, *arxiv.org/abs/1612.03975*, 2017. DOI: https://doi.org/10.48550/arXiv.1612.03975
56. Sap, M., Le Bras, R., Allaway, E., et al., Atomic: An Atlas of Machine Commonsense for If-Then Reasoning, *Proceedings of the AAAI Conference on Artificial Intelligence*, Honolulu, 2019, vol. 33, pp. 3027–3035.
57. Bosselut, A., Rashkin, H., Sap, M., et al., COMET: Commonsense Transformers for Automatic Knowledge Graph Construction, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019, pp. 4762–4779.
58. Bosselut, A., Le Bras, R., and Choi, Y., Dynamic Neuro-Symbolic Knowledge Graph Construction for Zero-Shot Commonsense Question Answering, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vancouver, Canada, 2021, pp. 4923–4931.
59. Shwartz, V., West, P., Le Bras, R., et al., Unsupervised Commonsense Question Answering with Self-talk, *arXiv:2004.05483*, 2020. DOI: https://doi.org/10.48550/arXiv.2004.05483

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

Received July 8, 2025, and revised November 17, 2025. Accepted December 16, 2025.

Author information

Kulinich, Aleksandr Alekseevich. Cand. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ alexkul@rambler.ru
ORCID ID: https://orcid.org/0000-0002-4751-205X

Cite this paper
Kulinich, A.A., Application of Large Language Models in Decision Support Systems. Part I: Explanation Models and Large Language Models. *Control Sciences* **1,** 41–56 (2026).

Original Russian Text © Kulinich, A.A., 2026, published in *Problemy Upravleniya*, 2026, no. 1, pp. 47–65.

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

METHODS FOR SOLVING THE AIRCRAFT LANDING OPTIMIZATION PROBLEM

E. L. Kulida*, V. G. Lebedev**, and N. A. Egorov***

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

*✉ elena-kulida@yandex.ru, **✉ lebedev-valentin@yandex.ru, ***✉ negorov@bk.ru

Abstract. This paper considers the problem of optimizing the sequence and time of aircraft landings, which is topical for increasing the capacity of airport runways. The main approaches to solving this problem are briefly overviewed, and its mathematical statement is provided, including the key constraints and objective functions. The exact solution method and approximate ones using genetic algorithms with two different crossover operators and a heuristic algorithm are described. The time to obtain an exact solution grows exponentially with increasing the problem dimension (the number of aircraft), which makes it impractical. Approximate methods yield a suboptimal, albeit quite good, solution in real time during aircraft landing. An original simulation software complex is presented and applied to compare the efficiency of the main methods implemented. It is proposed to use deep reinforcement learning to solve the problem.

Keywords: airport runway capacity, separation criteria between aircraft landings, optimality criteria, library of methods.

INTRODUCTION

Airport runway capacity constraints are a well-known bottleneck in air traffic management and a main cause of flight delays. Flight delays, especially before landing, lead to additional costs for airlines, increased fuel consumption, air pollution, and passenger dissatisfaction. To increase the capacity of existing airport runways, a key task is to optimize the flow of aircraft landings. The separation criteria between aircraft landings depend on the types (weights) of aircraft. Airport runway capacity depends on the sequence of aircraft landings since the minimum safe landing time for a heavy aircraft after a light one is significantly smaller than that for a light aircraft after a heavy one due to the turbulent vortex trail created by the leading aircraft. The *First Come First Served* (FCFS) principle, widely used in practice, rarely provides an optimal sequence in terms of runway capacity or average aircraft delay.

The problem of optimizing the flow of aircraft landings (further called the aircraft landing optimization problem) is to plan the sequence and time of aircraft landings in a terminal area, taking into account operational constraints, in order to minimize given criteria. Solving this problem is relevant for decision support tools that help air traffic controllers manage the flow of arriving aircraft, such as CenterTRACON (CTAS), developed by NASA and FAA [1], and its European counterpart, ArrivalMANager (AMAN) [2]. Extended AMAN is the version covering an area up to 500 miles from an airport, for early optimization of aircraft sequencing to reduce delays, noise, and fuel consumption in the airspace near airports [3].

Being long known, the aircraft landing optimization problem has been solved for several decades since the 1980s. However, the search for new solutions continues to the present day: the successful application of solutions in real time, when aircraft are preparing to land in an airport area, requires the highest possible speed of obtaining such solutions [4–6].

There are two different statements of the aircraft landing optimization problem: static and dynamic. In the static one, information about the set of aircraft approaching for landing is assumed to be known [7–10]. In the dynamic statement, the set of aircraft changes over time, as some aircraft land and new aircraft arrive in an airport area [11, 12]. However, the vast majority of studies have been devoted to the static statement since the dynamic counterpart is addressed based on the solution of the static one by using a sliding horizon and updating the solutions when the set of aircraft changes [12–15].

Safety is a major requirement in air traffic management. Mandatory safety requirements in aircraft landing optimization include constraints on a safe sep-

aration between sequential aircraft, in accordance with the International Civil Aviation Organization rules, and the requirement for aircraft to land within a specified time window, determined by its flight characteristics and available fuel. In practice, weather conditions, runway configuration, departure and approach routes, etc., can affect echeloning conditions. Additional constraints may be imposed, such as a limited shift relative to the aircraft position in the registration order of an airport area [7, 16–18], priority constraints, runway closure periods [8], etc. Various statements of the problem were investigated in the literature, considering one [11, 19, 20] or several runways [8, 9, 21–24], used only for landing or for both landing and takeoff [10, 25, 26]. Recently, there have been studies on the integrated planning of interrelated arrival-departure processes and aircraft movements on taxiways [27].

Objective functions vary as well. Commonly used objective functions are to minimize the deviation from a target landing time, i.e., the average, maximum, or total flight delay [3, 21, 23, 28–30], maximize runway capacity [31], minimize airline costs [32–34], and minimize environmental losses by optimizing fuel consumption and minimizing the emissions of harmful substances into the atmosphere [32, 35, 36]. Often, several conflicting objectives are considered in the problem statement, i.e., a multicriteria optimization problem is posed [37–39]. For multiple runways, it is important to balance the number of aircraft landings on different runways.

The scope of research into this problem is evolving dynamically, from exact solution methods to modern approaches based on reinforcement learning. The aircraft landing optimization problem is formalized as an NP-hard problem based on mixed-integer programming [14, 34, 40–42]. However, finding an exact solution is possible only for a problem of small dimension (for a small number of aircraft): the time required to calculate the solution grows exponentially when increasing the number of aircraft and is not satisfactory for practical application. The main methods for solving the problem include dynamic programming, the branch-and-bound method, and heuristic and metaheuristic methods [43]. Many authors use dynamic programming [15–18] and the branch-and-bound method [7, 25] to solve this problem. However, an important disadvantage of these approaches is that they require too much computational effort to solve high-dimensional problems. Many diverse heuristic and metaheuristic algorithms [44] have been developed to obtain good-quality approximate solutions in a reasonable time, such as genetic algorithms [45, 46], simulation annealing [20], search with prohibitions [14], ant colony optimization [38, 47–49], particle swarm optimization [50], gray wolf optimization [51], sparrow search algorithm [52], etc. Integer programming models and metaheuristic methods that use an initial solution to improve computational efficiency were compared in [24].

Comparing the efficiency of such approaches is difficult because the main analysis tool is simulation, and researchers use different datasets available to them, obtained from different airports around the world, e.g., Malpensa [13], Linate [14, 17], Beijing Capital Airport [11], Carthage [9], Fiumicino [13, 14], Gatwick [26], Doha [10], Kunming Changshui [52], etc. In this regard, the proposed solutions are commonly compared with the FCFS approach [28, 29, 42, 47–49].

Recently, reinforcement learning methods have shown great potential for obtaining quick solutions [53–55]. However, due to the low level of confidence in the results, artificial intelligence (AI) methods are not widely applied in air traffic management systems, where safety, traditionally ensured by humans, is critical. As concluded in the paper [56], such methods need further research, and explainable AI methods should be developed for their acceptance by end users.

1. ALGORITHMS FOR SOLVING THE PROBLEM

1.1. Problem Statement and Exact Solution

The problem of forming an optimal aircraft landing sequence is stated as a linear or quadratic mixed-integer programming problem, depending on the choice of the objective function.

Let us introduce the following notation: P is the number of aircraft waiting to land; E_i is the earliest possible landing time of aircraft i, $i = \overline{1, P}$; L_i is the latest possible landing time of aircraft i, $i = \overline{1, P}$; T_i is the optimal landing time of aircraft i, $i = \overline{1, P}$; C_i is the turbulence class of aircraft i, $i = \overline{1, P}$ (aircraft are classified by turbulence, and minimum permissible delays are assigned to aircraft classes); $S_{C_iC_j}$ is the minimum delay between the landing of an aircraft of class c_j after the landing of an aircraft of class c_i, $i, j = \overline{1, P}$, $i \neq j$; finally, x_i is the landing time assigned to aircraft i, $i = \overline{1, P}$.

Table 1 shows an example of the matrix $S_{C_iC_j}$ for four aircraft types: 1—Boeing 747, 2—Boeing 727, 3—Boeing 707, and 4—McDonnel Douglas DC-9.

Table 1

The minimum permissible delay between sequential aircraft landings (s)

		The type of next aircraft			
		1	2	3	4
The type of previous aircraft	1	96	200	181	228
	2	72	80	70	110
	3	72	100	70	130
	4	72	80	70	90

The problem is stated with $\sim P^2$ additional Boolean variables δ_{ij} determining the order of aircraft landings:

$$\delta_{ij} = \begin{cases} 1 \text{ if aircraft } i \text{ is landing before aircraft } j \\ 0 \text{ otherwise,} \end{cases}$$

$$i,\ j = \overline{1, P},\ i \neq j.$$

The main constraints are as follows:

• The landing time window constraint $[E_i, L_i]$ is given by

$$E_i \le x_i \le L_i,\ i = \overline{1,P}. \quad (1)$$

• The landing order constraint: either aircraft i is landing before aircraft j ($\delta_{ij} = 1$), or vice versa ($\delta_{ji} = 1$):

$$\delta_{ij} + \delta_{ji} = 1,\quad i,\ j = \overline{1, P},\ i \neq j. \quad (2)$$

• The minimum aircraft separation constraint can be written as

$$x_j \ge x_i + S_{C_iC_j}\delta_{ij} - (L_i - E_j)\delta_{ji},\ \forall (i,\ j) \in U. \quad (3)$$

The problem is to minimize an objective function under given constraints. The commonly used objective functions are:

• minimizing the cost of a deviation from the optimal landing time, i.e., a piecewise linear objective function of the form

$$\min_X \sum_{i=1}^{P} (g_i\alpha_i + h_i\beta_i),\ X = \{x_i,\},$$

where g_i and h_i are the penalties of aircraft i for landing before and after, respectively, the optimal time; the variables α_i and β_i are given by

$$\alpha_i = \max(0, T_i - x_i),\ i = \overline{1, P},$$

$$\beta_i = \max(0, x_i - T_i),\ i = \overline{1, P};$$

• minimizing the sum of the squared deviations of the landing times assigned from the optimal ones, i.e., a quadratic objective function of the form

$$\min_X \sum_{i=1}^{P} (T_i - x_i)^2,\ X = \{x_i,\ i = \overline{1, P}\};$$

(the maximum and average deviations from the optimal landing time are also minimized);

• minimizing the landing time for the entire group of aircraft:

$$\min\ (\max[x_1, \ldots, x_N] - \min[x_1, \ldots, x_N]).$$

The exact solution of the problem can be obtained using the CPLEX software package. Exact solutions are impractical due to the considerable computation time, but they are used to estimate the efficiency of approximate methods.

1.2. Approximate Solution of the Problem Using Genetic Algorithms

The solution of the problem under consideration is represented as two synchronized vectors $R = \{Y, X\}$, where $Y = \{y_1, \ldots, y_P\}$ is a permutation of aircraft numbers determining the landing sequence; $X = \{x_{y_1}, \ldots, x_{y_P}\}$ is a vector of landing times sorted in ascending order according to the landing sequence given by the vector Y.

The implementation of a genetic algorithm to solve the aircraft landing optimization problem begins with the creation of an initial population containing N random solutions.

The aircraft permutation vectors Y are formed randomly; then, for a known vector $Y = \{y_1, \ldots, y_P\}$, the landing time vector $X = \{x_{y_1}, \ldots, x_{y_P}\}$ with the minimum separations between sequential aircraft is calculated by the formulas

$$x_{y_1} = \max(T_1, E_{y_1}),$$

$$x_{y_i} = \min\left(\max\left(x_{y_{i-1}} + S_{C_{y_i}, C_{y_{i-1}}}, E_{y_i}\right), L_{y_i}\right), \quad (4)$$

$$i = \overline{2, P}.$$

The necessary and sufficient condition for satisfying the problem constraints (1)–(3) is

$$W(R) = \sum_{i=1}^{P} \sum_{\substack{j=1, \\ j \neq i,\ x_j > x_i}}^{P} \max\left(0, S_{C_iC_j} - (x_j - x_i)\right) = 0. \quad (5)$$

If the condition fails, the corresponding solution is inadmissible.

Solutions $R^1 = \{Y^1, X^1\}$ and $R^2 = \{Y^2, X^2\}$ are compared based on the value of the function $W(R)$ and the value of the objective function $F(R)$ as follows:

– If $W(R^1) < W(R^2)$, then the solution R^1 is better than R^2.

– If $W(R^1) = W(R^2) = 0$ and $F(R^1) < F(R^2)$, then the solution R^1 is better than R^2.

– If $W(R^1) = W(R^2) = 0$ and $F(R^1) = F(R^2)$, then the solution R^1 is equivalent to R^2.

Next, an iterative search process is executed to obtain a solution.

The step of this process consists in forming a new population using genetic operators. All solutions of the current population are divided into pairs, and genetic operators of crossover and mutation are applied to each pair. As a result, a new pair of daughter solutions is created for each pair of solutions, i.e., the number of solutions doubles. The values of $W(R)$ and $F(R)$ are calculated for all solutions, and then solutions are selected for a new population consisting of *N* solutions.

The process ends after a given number of iterations, and the solution with the minimum value of the objective function is selected from the last population.

The crossover operator is crucial for the successful solution of the problem using a genetic algorithm. Here, a challenge is that standard crossover operators cannot be applied to permutations: the resulting vector will not necessarily be a permutation. Special crossover operators have been developed for permutations. However, they are significantly more complex and require considerable computations.

Since the solution of the problem consists of two vectors $R = \{Y, X\}$, the following artificial technique can be used to settle this challenge [57]. For solutions

$$R^1 = \{Y^1, X^1\} \text{ and } R^2 = \{Y^2, X^2\},$$

the single-point crossover operator with the parameter k is applied to the vectors

$$X^1 = \{x^1_{y_1}, \ldots, x^1_{y_P}\} \text{ and } X^2 = \{x^2_{y_1}, \ldots, x^2_{y_P}\}.$$

The resulting vectors

$$Z^1 = \{x^1_{y_1}, \ldots, x^1_{y_k}, x^2_{y_{k+1}}, \ldots, x^2_{y_P}\},$$

$$Z^2 = \{x^2_{y_1}, \ldots, x^2_{y_k}, x^1_{y_{k+1}}, \ldots, x^1_{y_P}\}$$

are sorted in ascending order. After rearranging the aircraft landing sequence, the corresponding vectors

$$\hat{Z}^1 = \{z^1_{y'_1}, \ldots, z^1_{y'_P}\} \text{ and } \hat{Z}^2 = \{z^2_{y''_1}, \ldots, z^2_{y''_P}\}$$

are mapped to the vectors

$$\hat{Y}_1 = \{y'_1, \ldots, y'_P\} \text{ and } \hat{Y}_2 = \{y''_1, \ldots, y''_P\}.$$

According to formulas (4), the vectors $\hat{Y}_1$ and $\hat{Y}_2$ are matched with the vectors $\hat{X}_1$ and $\hat{X}_2$, and the two daughter solutions are obtained:

$$\hat{R}_1 = \{\hat{Y}_1, \hat{X}_1\} \text{ and } \hat{R}_2 = \{\hat{Y}_2, \hat{X}_2\}.$$

To assess the effectiveness of the proposed crossover operator, a more complex matrix crossover operator was also implemented [45]. The genetic algorithm based on this operator requires significantly more computing time to perform many matrix operations, but there is no gain in the efficiency of the resulting solution.

1.3. The Heuristic Algorithm for Improving the Solution

A well-known property of genetic algorithms is the rapid improvement of the initial (usually random) solution in the first iterations and a slowdown in the convergence rate when approaching the optimal solution. The heuristic algorithm for the iterative improvement of the approximate solution [58] has demonstrated good convergence acceleration near the optimal solution.

The initial solution becomes the current one: $R_1 = \{Y_1, X_1\}$.

The *first step* of the iterative algorithm is to perform $P-1$ sequential comparisons of solutions $R_j = \{Y_j, X_j\}$ and $R_{j+1} = \{Y_{j+1}, X_{j+1}\}$ for two aircraft permutations differing in the order of two adjacent aircraft, in accordance with the formulas

$$Y_j = \{y_1, \ldots, y_j, y_{j+1}, \ldots, y_P\},$$

$$Y_{j+1} = \{y_1, \ldots, y_{j+1}, y_j, \ldots, y_P\},$$

$$j = \overline{1, P-1}.$$

The corresponding vectors X_j and X_{j+1} are used to compare the solutions. In each comparison, the best of the solutions R_j and R_{j+1} becomes the current solution $\hat{R}$.

The *second step* of the iterative algorithm is to perform $P-2$ sequential comparisons of solutions

$R_j = \{Y_j, X_j\}$, $R_{j+1} = \{Y_{j+1}, X_{j+1}\}$, $R_{j+2} = \{Y_{j+2}, X_{j+2}\}$ for three aircraft permutations differing in the order of three adjacent aircraft, in accordance with the formulas

$$Y_j = \{y_1, \ldots, y_j, y_{j+1}, \ y_{j+2}, \ldots, y_P\},$$

$$Y_{j+1} = \{y_1, \ldots, y_{j+1}, y_{j+2}, y_j, \ldots, y_P\},$$

$$Y_{j+2} = \{y_1, \ldots, y_{j+2}, y_j, y_{j+1}, \ldots, y_P\},$$

$$j = \overline{1, P-2};$$

the best of the three solutions is selected as the current solution $\hat{R}$.

As shown by computational experiments, this heuristic algorithm appreciably improves the approximate solution obtained after a small number of steps of the genetic algorithm.

2. ALGORITHMS FOR SOLVING THE PROBLEM: AN EFFICIENCY ANALYSIS

The efficiency of different approaches (solution methods) for a particular problem is complex and ambiguous. The choice of an appropriate method depends on the intensity of aircraft landing flows, the time available for solving the problem, as well as the goals and constraints of decision-making. No known method can solve all optimization problems equally successfully. Different methods may work better for different groups of problems [5].

To analyze the efficiency of different algorithms for constructing optimal aircraft landing sequences, we developed a simulation software complex. The complex includes a simulation system for aircraft landing flows [59], mixed-integer programming algorithms for the exact solution of the problem with different objective functions using the CPLEX method [60], two genetic algorithms with different crossover operators, and heuristic and hybrid algorithms [61]. The user of this complex can:

– simulate problems with a given number of aircraft landings;

– apply various optimization algorithms and analyze the results;

– visualize the results;

– generate statistical data based on a given number of tests.

Figure 1 shows the main window of the complex, used to specify the problem parameters, select the solution methods, and visualize the results.

Four types of aircraft are considered, denoted by U (ultralight), L (light), M (medium), and H (heavy).

Three different solutions for a test are presented in Fig. 1. The first solution corresponds to the FCFS principle (Method: SortT); the second solution was obtained using the genetic algorithm (Method: GA); and the third is the exact solution yielded by CPLEX (Method: CPLEX).

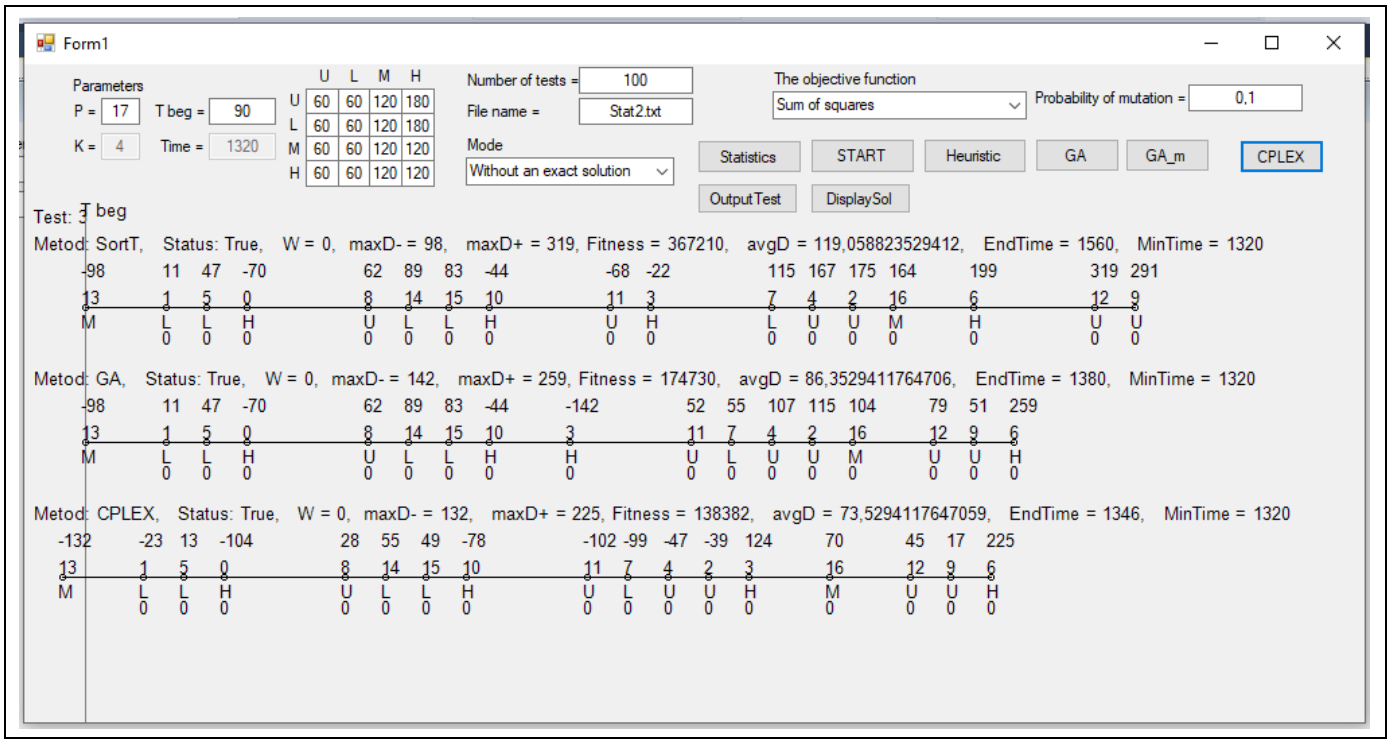

Fig. 1. The main window of the complex. The problem solution obtained by different algorithms, with visualization of the results.

For each solution R, the following parameters are calculated:

– $Status(R) = \begin{cases} \text{true if } W(R) = 0 \\ \text{false otherwise;} \end{cases}$

– the maximum advance relative to the optimal landing time,

$$\max D-(R) = \max_{i=\overline{1,P}}(0, T_i - x_i);$$

– the maximum delay from the optimal landing time:

$$\max D+(R) = \max_{i=\overline{1,P}}(0, x_i - T_i);$$

– the value of the objective function

$$F(R) = \sum_{i=1}^{P}(T_i - x_i)^2;$$

– the average deviation from the optimal landing time,

$$\text{avg}D(R) = \sum_{i=1}^{P}|T_i - x_i| / P;$$

– the runway occupancy time

$$EndTime(R) = \max[x_1,\ldots, x_N] - \min[x_1,\ldots, x_N].$$

During computational experiments, tests were generated, i.e., aircraft landing sequences with different numbers of aircraft. Then, the resulting aircraft landing optimization problem was solved, and the efficiency of the solutions by various methods was calculated on a large number of tests.

The tests were generated as follows.

The aircraft type (an integer) was randomly generated from the range $\overline{1, K}$: $C_i = rnd.Next(K)$, $i = \overline{1, P}$.

The times $0 \le E_i \le T_i \le L_i \le Time$, $i = \overline{1, P}$, for all aircraft were formed using a random real number generator. Here:

- $T_i = rnd.Next(T_0, T_0 + Time)$ is the optimal arrival time of aircraft i, a random number in the range $(\overline{T_0, T_0 + Time})$, $i = \overline{1, P}$. The aircraft flow intensity is varied with the parameter *Time*, which is selected close to the minimum time required for aircraft landing in accordance with formulas (4) and (5).
- $E_i = T_0 - t - rnd.Next(2t)$ is the earliest possible landing time of aircraft i, $i = \overline{1, P}$; t is an approximate minimum length of the landing window, e.g., t = 300 s; $rnd.Next(2t)$ is the random variation of the landing window length in the range $\overline{0, 2t}$.
- $L_i = T_0 + t + rnd.Next(2t)$ is the latest possible landing time for aircraft i, $i = \overline{1, P}$.

The simulation software was described in [62].

Based on the comparison of the obtained results, for all minimized parameters, the approximate solution significantly outperforms, on average, the solution with the initial aircraft landing sequence by the FCFS principle, but is worse than the optimal one.

The values of the objective function for the FCFS solution R and the generated new solution $\bar{R}$ are compared. Let the efficiency gain be the percentage decrease in the value of the objective function of the new solution $\bar{R}$ relative to that of the solution R:

$$\lambda = \frac{F(R) - F(\bar{R})}{F(R)} \cdot 100\ \%.$$

Table 2 shows the average efficiency gains λ, based on 500 tests, for the solutions of problems with 17, 50, and 100 aircraft, obtained using the genetic algorithm (GA) and then improved using the heuristic algorithm (GA + EA). For the low-dimension problem (with 17 aircraft), the maximum possible efficiency gain for the optimal solution is also given.

Table 2

The efficiency gains when using different algorithms

The number of aircraft	Optimal solution	GA	GA + EA
17	≈ 67.04%	≈ 46.2%	≈ 56.4%
50	–	≈ 51.5%	≈ 70.1%
100	–	≈ 53.5%	≈ 72.4%

3. REINFORCEMENT LEARNING – A PROMISING APPROACH TO SOLVING THE PROBLEM

In recent years, significant results for decision problems have been obtained in many areas, including aviation, using reinforcement learning algorithms [63, 64]. Reinforcement learning is searching for optimal control of a Markov decision process by the trial-and-error method when an agent interacts with an environment. The agent's choice of actions in different environment states is evaluated using instantaneous rewards. The agent's goal is to maximize long-term rewards [65].

We propose to use this approach to solve the aircraft landing optimization problem, where the decision time is critical: in reinforcement learning, a significant part of the computation time is transferred to the training stage, and the trained agent (program) generates

solutions very quickly. Computational experiments were carried out to study the possibility of using deep reinforcement learning to optimize the flow of aircraft landings.

The environment state at a time step t is described by

$$S_t = \{Y(t),\ X(t), C(t),\ D(t)\},$$

where $Y(t) = (y_1(t),\ldots,\ y_N(t))$ is the aircraft landing sequence (the aircraft numbers for landing); $X(t) = (x_1(t),\ldots,\ x_N(t))$ is the landing times in ascending order; $C(t) = \left(c_{y_1(t)}, \ldots, c_{y_N(t)}\right)$ is the vector of aircraft types in the order of landings; finally, $D(t) = \left(d_{y_1(t)}, \ldots, d_{y_N(t)}\right)$ is the vector of deviations from the optimal landing times.

The optimization problem is to determine the sequence $Y = (y_1,\ldots,\ y_N)$ and landing times $X = (x_1,\ldots,\ x_N)$ for a group of N aircraft during a given time interval $[T_0, T_k]$ that minimize an objective function under the constraints (1)–(3).

The objective function to be minimized has the form

$$F(X) = \sum_{i=1}^{N} (T_i - x_i)^2.$$

Figure 2 shows an example of the initial information for the problem. The separation time matrix is presented in two forms: the separation time matrix between aircraft of different classes and the separation time matrix between particular aircraft types.

A training episode begins with some arbitrary state S_0. At each time step t, given an environment state S_t, the agent performs some action $A_t \in A$; as a result, the environment passes to a state S_{t+1}, and the agent receives an instantaneous reward R_t.

The instantaneous reward for the episode is a large penalty when failing to build a landing schedule for all aircraft that satisfies the above constraints, or a penalty

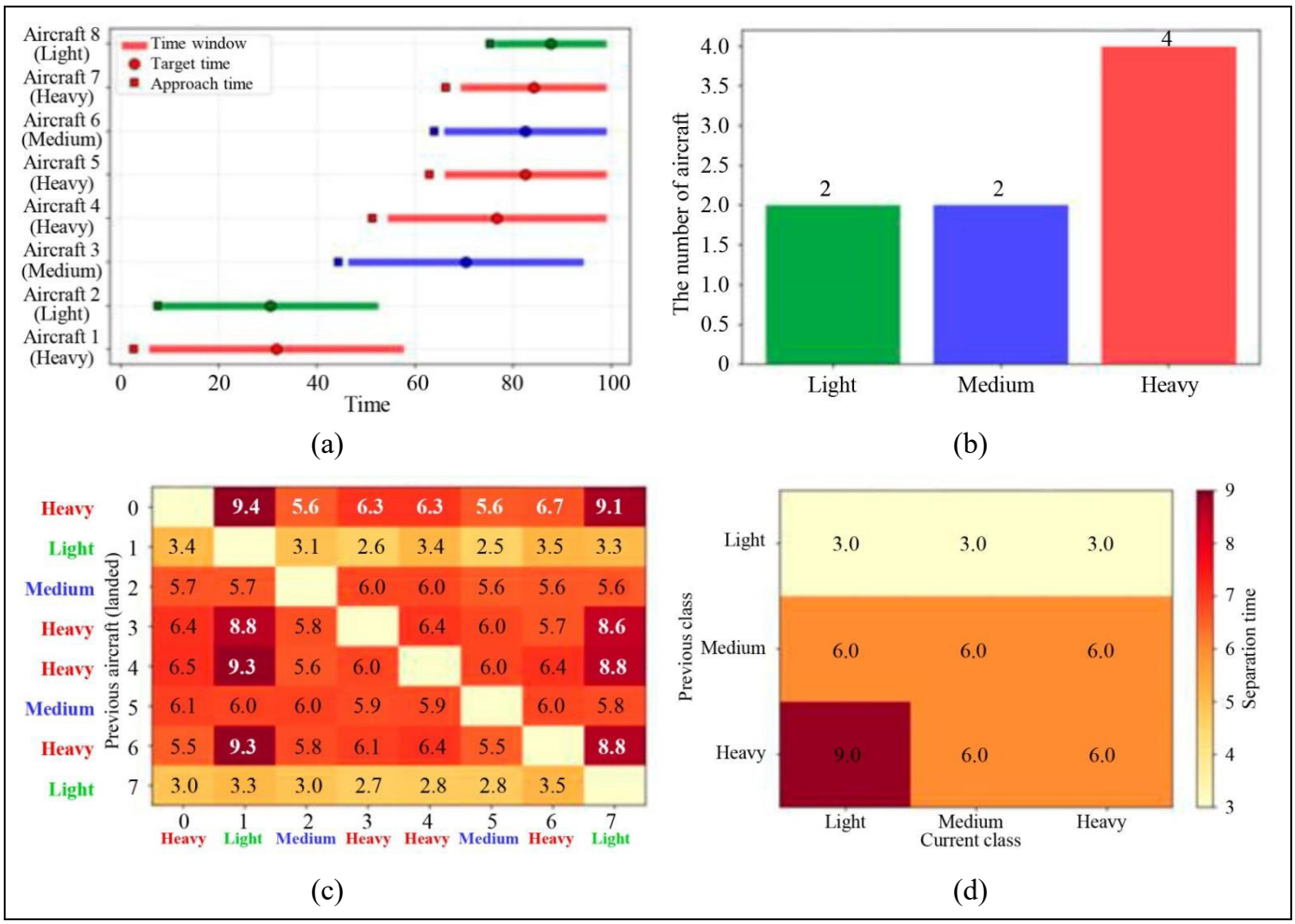

Fig. 2. Initial data for the problem: (a) aircraft landing time windows, (b) the aircraft distribution by classes, (c) the separation time matrix between particular aircraft types (row → column: previous → current), and (d) the separation time matrix between aircraft of different classes (previous → current).

equal to the sum of the squared deviations from the target landing times otherwise:

$$R_t = \begin{cases} -5000 \text{ if a correct schedule is not obtained} \\ -\sum_{i=1}^{N} \left(T_i - x_i\right)^2 \text{ otherwise.} \end{cases}$$

After the episode is completed, the sets $\left[S_t, A_t, R_t, S_{t+1}\right]$ are stored in memory (replay pool, see Fig. 3). When training the agent, mini-packets of accumulated data are selected from the replay pool (Fig. 4).

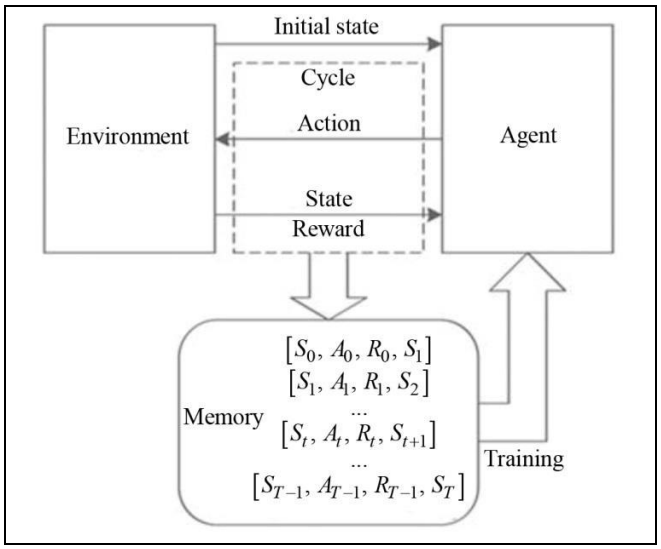

Fig. 3. Data formation for agent's training.

In the state S_t, the agent selects an action $A_t \in A$ according to an action selection function a in a state s, i.e., a strategy $\pi(s, a)$; the goal of the strategy is to maximize the long-term reward.

In Q-learning [66], the action value function $Q(S_t, A_t)$ approximates the optimal action value function regardless of the strategy used. At each time step, the values of the former function are updated according to the formula

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t)$$

$$+\alpha\left[R_{t+1} + \gamma \max_{a \in A} Q(S_{t+1}, a) - Q(S_t, A_t)\right],$$

where α is the learning rate and γ is the discount factor.

After updating the action value function, the strategy $\pi(s, a)$ is updated. The Q-learning agent performs a ϵ-greedy action selection. The action A_t maximizing the value of $Q(S_t, A_t)$ in a state S_t is selected with probability $1-\epsilon$, and a random action is selected with probability ϵ. First, a maximum value $\epsilon = \epsilon_{\max}$ is set; then, during the learning process, ϵ is reduced until reaching a minimum value $\epsilon = \epsilon_{\min}$. Thus, an iterative process is implemented to refine the function $Q(s, a)$ sequentially and improve the strategy $\pi(s, a)$ constantly relative to the refined function $Q(s, a)$.

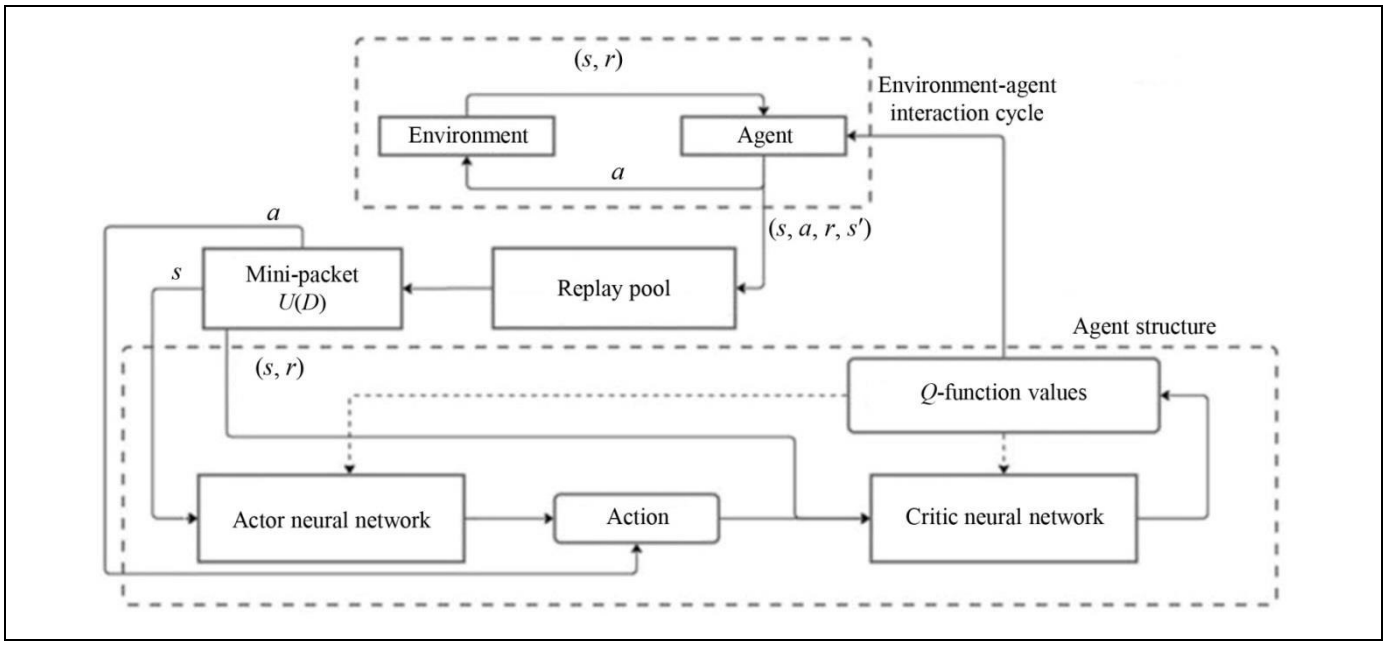

Fig. 4. The flowchart of the deep Q-learning algorithm.

In deep Q-learning (Fig. 4), a deep neural network is used to approximate the function Q by considering the function $Q(s, a, \theta)$, where θ are neural network parameters [67]. The goal of network training is to determine the parameters θ_t':

$$Y_t' = R_{t+1} + \max_{a_{t+1}} Q(S_{t+1}, A_{t+1}; \theta_t').$$

To find the best action according to the function $Q(s, a, \theta)$, a second neural network is used: it is trained to determine the parameters θ_t'':

$$Y_t'' = R_{t+1} + \gamma Q(s_{t+1}, \arg\max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta_t'); Q_t'').$$

The reward graph for successive agent's training episodes with the parameters $\alpha = 0.0005$, $\gamma = 0.98$, $\epsilon_{\max} = 1.0$, $\epsilon_{\min} = 0.05$, and $\epsilon_{decay} = 0.999$ is shown in Fig. 5.

Figure 6 shows the deviations of landing times from the target time in the test example (Fig. 2) in the schedule built by the trained agent.

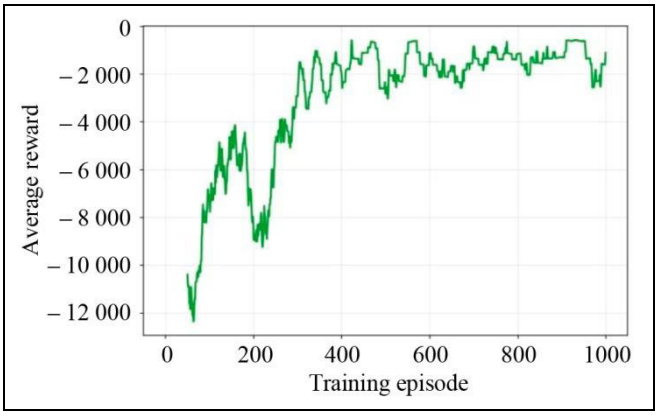

Fig. 5. The reward graph for agent's training.

The use of this method to solve the aircraft landing optimization problem has been mentioned in the literature, but no descriptions of particular algorithms can be found in open sources. The reinforcement learning approach has several undeniable advantages over other algorithms:

– The solution is obtained significantly faster by a trained agent.

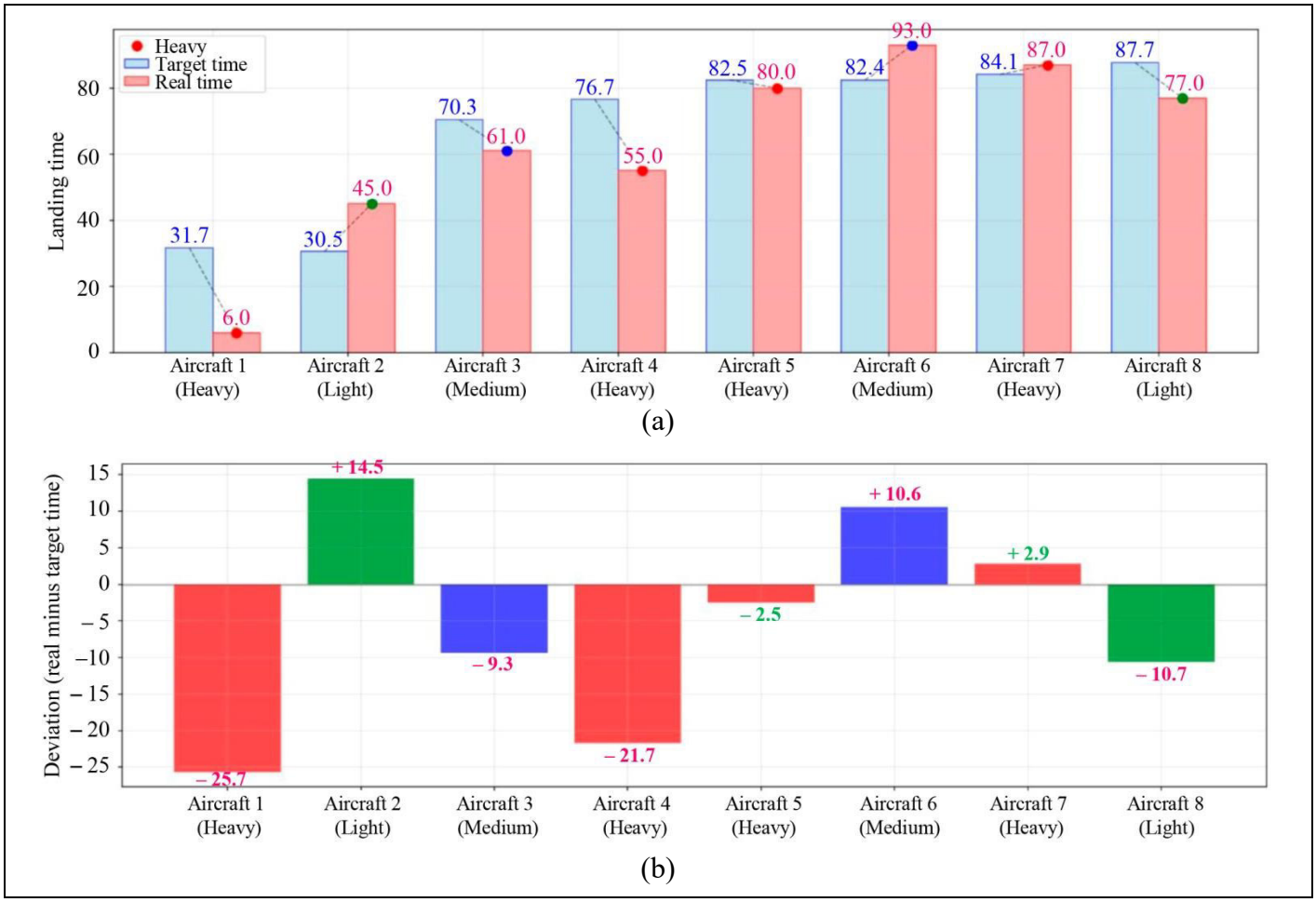

Fig. 6. The deviations of landing times from the target time: the solution obtained by a trained reinforcement learning agent.

– It is possible to use the necessary objective function by retraining the agent with a new reward function.

– It is possible to solve the problem without dividing aircraft into 3 or 4 classes by setting a separation time matrix between landings for specific aircraft types.

We plan to investigate this approach further. According to the experimental results with one of the first reinforcement learning algorithms (deep *Q*-learning), reinforcement learning can be applied to optimize the flow of aircraft landings. They demonstrate that a program agent can be trained to build an aircraft landing schedule under the necessary constraints, without using a pre-coded algorithm, through repeated episodes of interaction with the simulation environment. This repetitive interaction process is based on the reward received at each interaction step, and the environment uses the reward to evaluate the success of the agent's actions.

CONCLUSIONS

Optimizing the flow of aircraft landings is one of the most important problems in air traffic management since optimal landing planning allows increasing the efficiency of runways and reducing flight delays. Several methods have been developed to solve this problem based on mixed-integer programming, dynamic programming, and metaheuristic approaches such as genetic algorithms, ant colony optimization, gray wolf optimization, etc. In this study, the efficiency of genetic and hybrid (genetic and heuristic) algorithms has been compared on a large number of tests; according to the results, the algorithms can significantly improve the value of the objective function, even yielding its optimal value in many cases. These methods are not as computationally expensive as exact counterparts, but they still require a significant amount of computation while aircraft approach the airport, and the solution must be constantly recalculated. The situation is complicated by the fact that simulation is the main way to assess the efficiency of heuristic algorithms; in addition to equipment with different parameters and different levels of programmer skills, simulation involves available datasets obtained in different conditions at airports around the world.

Reinforcement learning, which has become widespread in recent years, seems a promising approach for a faster solution of the problem. A feature of this approach is the transfer of a significant part of computations to the training stage of neural networks (agents); then the trained agent (program) generates a solution fairly quickly. Further research is needed to assess the efficiency of this approach.

Despite many years of intensive effort by numerous scientists, effective algorithms for optimizing the sequence and times of aircraft landings have not yet been found. The best algorithms developed so far have both undoubted advantages and obvious drawbacks, demonstrating different efficiency in different situations with different objective functions and constraints. One recipe could be a library of methods to apply the best available solution in particular conditions of the aircraft landing optimization problem.

REFERENCES

1. Erzberger, H., Davis, T.J., and Green, S., Design of Center-TRACON Automation System, *Proc. of the AGARD Guidance and Control Symposium on Machine Intelligence in Air Traffic Management*, Berlin, Germany, 1993.
2. Liang, M., Aircraft Route Network Optimization in Terminal Maneuvering Area, *PhD Thesis*, Toulouse, France: Universit´e Paul Sabatier, 2018. URL: https://theses.hal.science/tel-01703861/document.
3. Montlaur, A. and Delgado, L., Delay Assignment Optimization Strategies at Pre-Tactical and Tactical Levels, *Proc. of the Fifth SESAR Innovation Days*, Bologna, Italy, 2015, pp. 1–8.
4. Ikli, S., Mancel, C., Mongeau, M., et al., The Aircraft Runway Scheduling Problem: A Survey, *Computers & Operations Research*, 2021, vol. 132, no. 3. DOI: 10.1016/j.cor.2021.105336
5. Shirini, K., Aghdasi, H.S., and Saeedvand, S., A Comprehensive Survey on Multiple-Runway Aircraft Landing Optimization Problem, *International Journal of Aeronautical and Space Sciences*, 2024, no. 25, pp. 1574–1602. DOI: 10.1007/s42405-024-00747-z
6. Knyazhsky, A.Yu. and Baushev, S.V., Current State and Prospects for Development of Systems for Planning Airspace Management. Part 2, *Crede Experto: Transport, Society, Education, Language*, 2025, no. 3 (46). DOI 10.51955/2312-1327_2025_3_87 (In Russian.)
7. Eun, Y., Hwang, I., and Bang, H.-C., Optimal Arrival Flight Sequencing and Scheduling Using Discrete Airborne Delays, *IEEE Transactions on Intelligent Transportation Systems*, 2010, vol. 11, no. 2, pp. 359–373. DOI: 10.1109/TITS.2010.2044791
8. Phirouzabadi, A.M., Aminnayeri, M., and Mahmoudian, M., Aircraft Landing Scheduling Based on Unavailability of Runway Constraint Through a Time Segment Heuristic Method, *International Journal of Informatics and Communication Technology*, 2013, vol. 2, no. 3, pp. 175–182. DOI: 10.11591/ijict.v2i3.5284
9. Messaoud, M.B., Ghedira, K., and Harizi, R., The Multiple Runway Aircraft Landing Problem: A Case Study for Tunis Carthage Airport, *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Banff, Canada, 2017. DOI: 10.1109/SMC.2017.8123051
10. Ghoniem, A., Sherali, H.D., and Baik, H., Enhanced Models for a Mixed Arrival-Departure Aircraft Sequencing Problem, *INFORMS Journal on Computing*, 2014, vol. 26, no. 3, pp. 514–530. DOI: 10.1287/ijoc.2013.0581
11. Ji, X.-P., Cao, X.-B., Du, W., and Tang, K., An Evolutionary Approach for Dynamic Single Runway Arrival Sequencing and Scheduling Problem, *Soft Computing*, 2017, vol. 21, pp. 7021–7037. DOI: 10.1007/s00500-016-2241-8

12.Bennell, J.A., Mesgarpour, M., and Potts, C.N., Dynamic Scheduling of Aircraft Landing, *European Journal of Operational Research*, 2017, vol. 258, no. 1, pp. 315–327. DOI: 10.106/j.ejor.206.08.05
13.Samà, M., D'Ariano, A., and Pacciarelli, D., Rolling Horizon Approach for Aircraft Scheduling in the Terminal Control Area of Busy Airports, *Procedia Social and Behavioral Sciences*, 2013, no. 80, pp. 531–552. DOI: 10.1016/j.sbspro.2013.05.029
14.Furini, F., Kidd, M.P., Persiani, C.A., and Toth, P., Improved Rolling Horizon Approaches to the Aircraft Sequencing Problem, *Journal of Scheduling*, 2015, vol. 18, no. 5, pp. 435–447. DOI: 10.1007/s10951-014-0415-8
15.Lieder, A. and Stolletz, R., Scheduling Aircraft Take-offs and Landings on Interdependent and Heterogeneous Runways, *Transportation Research. Part E: Logistics and Transportation Review*, 2016, no. 88, pp. 167–188. DOI: 10.1016/j.tre.2016.01.015
16.Balakrishnan, H. and Chandran, B.G., Algorithms for Scheduling Runway Operations under Constrained Position Shifting, *Operations Research*, 2010, vol. 58, no. 6, pp. 1650–1665. DOI: 10.1287/opre.1100.0869
17.Furini, F., Kidd, M.P., Persiani, C.A., and Toth, P., State Space Reduced Dynamic Programming for the Aircraft Sequencing Problem with Constrained Position Shifting, in *Lecture Notes in Computer Science*, vol. 8596, Fouilhoux, P., Neves Gouveia, L.E., Mahjoub, A.R., and Paschos, V.T., Eds., pp. 267–279. DOI: 10.1007/978-3-319-09174-7-23. (Proceedings of International Symposium on Combinatorial Optimization, Lisbon, Portugal, 2014.)
18.Chandran, B. and Balakrishnan, H., A Dynamic Programming Algorithm for Robust Runway Scheduling, *Proceedings of 2007 American Control Conference*, New York, USA, 2007. DOI: 0.09/ACC.2007.4282922
19.Vadlamani, S. and Hosseini, S., A Novel Heuristic Approach for Solving Aircraft Landing Problem with Single Runway, *Journal of Air Transport Management*, 2014, no. 40, pp. 144–148. DOI: 0.1016/j.jairtraman.2014.06.009
20.Faye, A., A Quadratic Time Algorithm for Computing the Optimal Landing Times of a Fixed Sequence of Planes, *European Journal of Operational Research*, 2018, vol. 270, no. 3, pp. 1148–1157. DOI: 10.1016/j.ejor.2018.04.021
21.Hancerliogullari, G., Rabadi, G., AI-Salem, A.H., and Kharbeche, M., Greedy Algorithms and Metaheuristics for a Multiple Runway Combined Arrival-Departure Aircraft Sequencing Problem, *Journal of Air Transport Management*, 2013, no. 32, pp. 39–48. DOI: 10.1016/j.jairtraman.2013.06.001
22.Ghoniem, A.F., Farhadi, F., and Reihaneh, M., An Accelerated Branch-and-Price Algorithm for Multiple-runway Aircraft Sequencing Problems, *European Journal of Operational Research*, 2015, vol. 246, no. 1, pp. 34–43. DOI: 10.1016/j.ejor.2015.04.019
23.Salehipour, A., An Algorithm for Single-and Multiple-Runway Aircraft Landing Problem, *Mathematics and Computers in Simulation*, 2020, vol. 175, no. 6, pp. 179–191. DOI: 10.1016/j.matcom.2019.10.006
24.Santos, L. and Fuchigami, H., An Optimization Study of the Time-Index and Arc-Time-Indexed Models and the Iga Metaheuristic for the Aircraft Sequencing Problem, *Revista de Gestão Social e Ambiental*, 2025, vol. 19, no. 4. DOI: 10.24857/rgsa.v19n4-032
25.D'Ariano, A. and D'Urgolo, P., Optimal Sequencing of Aircrafts Take-off and Landing at a Busy Airport, *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems*, Funchal, Portugal, 2010, pp. 1569–1574. DOI: 10.109/ITSC.2010.5625114
26.Rodríguez-Díaz, A., Adenso-Díaz, B., and González-Torre, P.L., Minimizing Deviation from Scheduled Times in a Single Mixed Operation Runway, *Computers Operations Research*, 2016, no. 78, pp. 193–202. DOI: 10.1016/j.cor.2016.09.014
27.Xia, C., Wen, Y., Hu, M., et al., Microscopic-Level Collaborative Optimization Framework for Integrated Arrival-Departure and Surface Operations: Integrated Runway and Taxiway Aircraft Sequencing and Scheduling, *Aerospace*, 2024, vol. 11, no. 12. DOI: 10.3390/aerospace11121042
28.Salehipour, A. and Ahmadian, M.M., Heuristics for Flights Arrival Scheduling at Airports, *International Transactions in Operational Research*, 2020, vol. 29, no. 1. DOI: 10.1111/itor.12901
29.Awasthi, A., Kramer, O., and Lässig, J., Aircraft Landing Problem: an Efficient Algorithm for a Given Landing Sequence, *Proceedings of the 2013 IEEE 16th International Conference on Computational Science and Engineering*, Sydney, Australia, 2013. DOI: 0.109/CSE.2013.14
30.Mokhtarimousavi, S., Rahami, H., and Kaveh, A., Multi-Objective Mathematical Modeling of Aircraft Landing Problem on a Runway in Static Mode, Scheduling and Sequence Determination Using NSGA-II, *International Journal of Optimization in Civil Engineering*, 2015, vol. 5, no. 1, pp. 21–36.
31.Prakash, R., Piplani, R., and Desai, J., An Optimal Data-Splitting Algorithm for Aircraft Scheduling on a Single Runway to Maximize Throughput, *Transportation Research. Part C: Emerging Technologies*, 2018, no. 95, pp. 570–581. DOI: 10.1016/j.trc.208.07.031
32.Chen, H. and Solak, S., Lower Cost Arrivals for Airlines: Optimal Policies for Managing Runway Operations under Optimized Profile Descent, *Production and Operations Management*, 2015, vol. 24, no. 3, pp. 402–420. DOI: 10.1111/poms.12244
33.Rezaei, H., An Iterative Bidding Approach Applied to Cost Reduction in the Context of Aircraft Landing Problem, *Proceedings of 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Nanjing, China, 2018. DOI: 10.1109/cscwd.2018.8465389
34.Serhan, D., Lee, H., and Yoon, S.W., Minimizing Airline and Passenger Delay Cost in Airport Surface and Terminal Airspace Operations, *Journal of Air Transport Management*, 2018, no. 73, pp. 120–133. DOI: 10.1016/j.jairtraman.2018.07.001
35.Sölveling, G., Solak, S., Clarke, J.-P., and Johnson, E.L., Scheduling of Runway Operations for Reduced Environmental Impact, *Transportation Research. Part D: Transport and Environment*, 2011, vol. 16, no. 2, pp. 110–120. DOI: 10.1016/j.trd.2010.09.004
36.Tian, Y., Wan, L., Han, K., and Ye, B., Optimization of Terminal Airspace Operation with Environmental Considerations, *Transportation Research. Part D: Transport and Environment*, 2018, no. 63, pp. 872–889. DOI: 10.1016/j.trd.2018.06.018
37.Zhang, J., Zhao, P., Zhang, Y., et al., Criteria Selection and Multi-Objective Optimization of Aircraft Landing Problem, *Journal of Air Transport Management*, 2020, no. 82. DOI: 10.1016/j.jairtraman.2019.101734
38.Tang, K., Wang, Z., Cao, X., et.al., A Multi-Objective Evolutionary Approach to Aircraft Landing Scheduling Problems, *Proceedings of 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, Hong Kong, China, 2008. DOI: 10.1109/cec13288.2008

39. Wang, S., Yue, Z., Zhang, Z.-H., and Yu, H., Multi-Objectives Optimization on Flights Landing Sequence at Busy Airport, *Journal of Transportation Systems Engineering and Information Technology*, 2012, vol. 12, no. 4, pp. 135–142. DOI: 10.1016/s1570-6672(11)60218-3
40. Beasley, J.E., Krishnamoorthy, M., Sharaiha, Y.M., and Abramson, D., Scheduling Aircraft Landings – the Static Case, *Transportation Science*, 2000, vol. 34, no. 2, pp. 180–197. DOI: 10.1287/trsc.34.2.80.12302
41. Spiridonov, A.A. and Kumkov, S.S., Keeping Order of Vessels in Problem of Safe Merging Aircraft Flows, *Vestnik Udmurt. Univ. Mat. Mekh. Komp. Nauki*, 2022, vol. 32, no. 3, pp. 433–446. DOI: 35634/vm220306
42. Rogovs, S., Nikitina, V., and Gerdts, M., A Novel Mixed-Integer Programming Approach for the Aircraft Landing Problem, *Frontiers in Future Transportation*, 2022, no. 3. DOI: 10.3389/ffutr.2022.968957
43. Bennell, J.A., Mesgarpour, M., and Potts, C.N., Airport Runway Scheduling, *Annals of Operations Research*, 2011, vol. 9, no. 2, pp. 115–138. DOI: 10.1007/s10288-011-0172-x
44. Veresnikov, G.S., Egorov, N.A., Kulida, E.L., and Lebedev, V.G., Methods for Solving of the Aircraft Landing Problem. II. Approximate Solution Methods, *Automation and Remote Control*, 2019, vol. 80, no. 8, pp. 1502–1518.
45. Hu, X.B. and Di Paolo, E., Binary-Representation-Based Genetic Algorithm for Aircraft Arrival Sequencing and Scheduling, *IEEE Transactions on Intelligent Transportation Systems*, 2008, vol. 9, no. 2, pp. 301–310. DOI: 10.1109/TITS.2008.922884
46. Hu, X.B. and Paolo, E.D., An Efficient Genetic Algorithm with Uniform Crossover for Air Traffic Control, *Computers & Operations Research*, 2009, vol. 36, no. 1, pp. 245–259. DOI: 10.1016/j.cor.2007.09.005
47. Hu, B., An Efficient Ant Colony Algorithm Based on Wake-Vortex Modeling Method for Aircraft Scheduling Problem, *Journal of Computational and Applied Mathematics*, 2016, vol. 317, no. 2, pp. 157–170. DOI: 10.1016/j.cam.2016.11.043
48. Zhan, Z.-H., Zhang, J., Li, Y., et al., An Efficient Ant Colony System Based on Receding Horizon Control for the Aircraft Arrival Sequencing and Scheduling Problem, *IEEE Transactions on Intelligent Transportation Systems*, 2010, vol. 11, no. 2, pp. 399–412. DOI: 10.1109/TITS.2010.2044793
49. Feng, X.R., Feng, X.J., and Liu, D., The Application of Ant Colony Optimization Algorithm in the Flight Landing Scheduling Problem, *Applied Mechanics and Materials*, 2013, vols. 411–414, pp. 2698–2703. DOI: 10.4028/www.scientific.net/AMM.411-414.2698
50. Girish, B., An Efficient Hybrid Particle Swarm Optimization Algorithm in a Rolling Horizon Framework for the Aircraft Landing Problem, *Applied Soft Computing*, 2016, no. 44, pp. 200–221. DOI: 10.1016/j.asoc.2016.04.011
51. Teymori, M., Taghizadeh, H., Honarmand, M.A., and Pourmahmond, J., A New Improved Gray Wolf Optimization Algorithm to Solve the Aircraft Landing Problem at Mashhad Shahid Hasheminejad International Airport, *International Journal of Nonlinear Analysis and Application*, 2022, vol. 13, no. 2, pp. 435–445. DOI: 10.22075/ijnaa.2021.23255.2510
52. Zhao, W. and Liang, T., Optimization of Terminal Area Arrival Flight Sorting Based on an Improved Sparrow Search Algorithm, *Science Progress*, 2024, vol. 107, no. 1, pp. 1–17. DOI: 10.1177/00368504241238078
53. Kulida, E.L. and Lebedev, V.G., Methods for Solving Some Problems of Air Traffic Planning and Regulation. Part II: Application of Deep Reinforcement Learning, *Control Sciences*, 2023, no. 2, pp. 2–14. DOI: 10.25728/cs.2023.2.1
54. Sosedov, V., Makarevich, M., and Kulida, E., Solving the Aircraft Landing Problem Using Deep Q-Networks Approach, *Proceedings of 2024 17th International Conference on Management of Large-Scale System Development (MLSD)*, Moscow, Russian Federation, 2024, pp. 1-5. DOI: 10.1109/MLSD61779.2024.10739636.
55. Brittain, M. and Wei, P., Autonomous Aircraft Sequencing and Separation with Hierarchical Deep Reinforcement Learning, *Proceedings of the International Conference for Research in Air Transportation (ICRAT2018)*, Barcelona, Spain, 2018.
56. Degas, A., Islam, M.R., Hurter, C., et al., A Survey on Artificial Intelligence (AI) and eXplainable AI in Air Traffic Management: Current Trends and Development with Future Research Trajectory, *Applied Sciences*, 2022, vol. 12, no. 3, art. no. 1295. DOI: 10.3390/ app12031295
57. Kulida, E.L., Genetic Algorithm for Solving the Problem of Optimizing Aircraft Landing Sequence and Times, *Automation and Remote Control*, 2022, vol. 83, no. 3, pp. 426–436.
58. Kulida, E.L., Lebedev, V.G., and Egorov, N.A., Study of the Effectiveness of the Algorithm to Optimize the Flow of Aircraft on Landing, *Control Sciences*, 2019, no. 6, pp. 63–69. DOI: 10.25728/pu.2019.6.7 (In Russian.)
59. Kulida, E.L., An Instrumental Tool for Studying Algorithms for Constructing Optimal Aircraft Landing Sequences, *Certificate of State Registration of a Computer Program no. 2018615221*, Russian Federation. Registered May 3, 2018. (In Russian.)
60. Egorov, N.A., A Library of Precise Methods for Constructing Optimal Aircraft Landing Sequences by Four Optimality Criteria, Implemented as a .Net Assembly, *Certificate of State Registration of a Computer Program no. 2019616970*, Russian Federation. Registered June 3, 2019. (In Russian.)
61. Kulida, E.L., Implementation of a Genetic Algorithm for Optimizing the Sequence and Timing of Aircraft Landings, *Certificate of State Registration of a Computer Program no. 2022616075*, Russian Federation. Registered April 18, 2022. (In Russian.)
62. Kulida, E.L., Analysis of Algorithms for Solving the Aircraft Landing Problem, *Proceedings of 2020 13th International Conference on Management of Large-Scale System Development (MLSD)*, Moscow, Russia, 2020, pp. 1-4. DOI: 10.1109/MLSD49919.2020.9247839.
63. Razzaghi, P., Tabrizian, A., Guo, W., et al., A Survey on Reinforcement Learning in Aviation Application, *Engineering Applications of Artificial Intelligence*, 2024, vol. 136, no. 3. DOI: 10.1016/j.engappai.20244.108911
64. Ivanova, P.I., Pechenejskiy, V.K., and Chuvikovskaya, E.K., Application of Machine Learning and Neural Network Technologies for Selecting Optimal Aircraft Flight Trajectories in Air Traffic Control, *Automation. Modern Technologies*, 2024, no. 12. DOI: 10.36652/0869-4931-2024-78-12-553-557 (In Russian.)
65. Sutton, R.S. and Barto, A.G., *Reinforcement Learning: An Introduction*, 2nd ed., Cambridge: MIT Press, 2014, 2015.
66. Watkins, C.J. and Dayan, P., Q-learning, *Machine Learning*, 1992, vol. 8, pp. 279–292.
67. Morales, M., *Grokking Deep Reinforcement Learning*, London: Dimensions, 2020.

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

Received September 18, 2025, and revised November 27, 2025. Accepted December 16, 2025.

Author information
Kulida, Elena L'vovna. Cand. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ elena-kulida@yandex.ru
ORCID iD: https://orcid.org/0009-0003-0226-9708

Lebedev, Valentin Grigor'evich. Dr. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ lebedev-valentin@yandex.ru
ORCID iD: https://orcid.org/0000-0003-3206-9558

Egorov, Nikolay Aleksandrovich. Cand. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ negorov@bk.ru
ORCID iD: https://orcid.org/0009-0006-5141-1207

Cite this paper
Kulida, E.L., Lebedev, V.G., and Egorov, N.A., Methods for Solving the Aircraft Landing Optimization Problem. *Control Sciences* **1**, 57–69 (2026).

Original Russian Text © Kulida, E.L., Lebedev, V.G., Egorov, N.A., 2026, published in *Problemy Upravleniya*, 2026, no. 1, pp. 66–80.

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

A UNIFIED DETECTION PROBABILITY FIELD FOR A GROUP OF STATIONARY OBSERVERS

I. M. Rudko

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

✉ igor-rudko@mail.ru

Abstract. This paper addresses the problem of calculating a unified detection probability field for a group of stationary observers monitoring a given area and operating in active mode. Stationary observers are located on a plane equipped with the Cartesian coordinate system, and their coordinates are known. For an object with given coordinates, a unified detection probability field is calculated for all stationary observers in the area. As shown, the greatest complexity arises for the objects necessitating the consideration of the terrain relief. For such objects, it is required to calculate a multilayer map with the altitude of the object's movement, in contrast to a single-layer (flat) detection probability map, which is sufficient for detecting objects without taking the terrain relief into account. Examples are provided to demonstrate that the probability of object detection depends on the location and altitude of the observers, the altitude of the object's movement, and the terrain relief. With a unified detection probability field calculated for a given area in the form of a multilayer map, it is much easier to optimize the location of observers (on the one hand) and control moving objects in a conflict environment (on the other).

Keywords: probability of object detection, detection probability field for an object, trajectory of an object, terrain relief.

INTRODUCTION

When detecting an object in active mode, the laws of radio wave propagation in space are used. The main laws are as follows: the constancy of the signal propagation speed; the rectilinear nature of signal propagation; the directionality of signal radiation and reception, which is based on the phenomenon of radio wave interference; and the Doppler effect. In this case, the probing and reflected signals propagate along a rectilinear trajectory without distorting their shape [1]. Considering the curvature of the Earth's surface, for an object at an altitude of 50 m, the line-of-sight range of an observer located on a perfectly flat surface is approximately 39 km and will increase with the object's altitude. The detection range of an object moving at an altitude of 100 m is approximately 52–55 km. A moving object is successfully detected only within the line of sight. For objects whose detection is not affected by the terrain relief, it is possible to create a so-called continuous detection field. This is much more difficult to do when taking the terrain relief into account. Note that in the USSR, a continuous detection field was never created, even in the European part of the country. If an object moves at very low altitudes (VLA), from several tens to hundreds of meters [2], then terrain, forests, buildings, and structures create so-called clutter notches. Today, 99% of objects move at altitudes of 200–300 m, and even higher above rugged terrain, since flight altitude is measured either by satellite global positioning systems or by barometric altimeters. The adoption of laser altimeters will reduce this altitude to 30–50 m, making them even more difficult to detect.

An observer processes the signals reflected from an object in three stages [1]:

- *Primary processing* includes operations for detecting and measuring (estimating) the parameters of received signals. Primary processing is performed directly by the observer. The set of signal parameter estimates forms an object's mark.
- *Secondary processing* is performed on the set of marks and provides trajectory information.

• *Tertiary processing* is intended to combine and identify information from individual observers within the system or information from individual systems.

This paper addresses the problem of object detection considering the terrain relief in an area monitored by several stationary observers (i.e., primary and tertiary processing).

The region of potential object detection depends on the terrain relief, i.e., it has a contrasting structure, with alternating line-of-sight (LOS) regions (object in direct visibility) and shadow regions (object outside direct visibility). The contrasting structure of the observation region can be described by any indicators: either as the intensity of the useful signal or as the probability of detecting the useful signal, as long as they represent a correct convolution of all the main influencing factors and are available for estimation. This possibility is directly characterized by a **probabilistic criterion**, i.e., the probability of an event that, during the object's movement along the route, it will be detected by at least one of $L \geq 1$ observers located in the area. Let this probability of object detection be denoted by P_{det}. In the case of independent observers, the probability of object detection by at least one observer is estimated using the integral (cumulative) probability formula [3]:

$$P_{det} = 1 - \prod_{i=1}^{L} (1 - P_{det\,i}), \qquad (1)$$

where L is the number of observers, and $P_{det\,i}$ is the probability of detection by the ith observer. This task belongs to tertiary processing. With known coordinates of the observers, one can use formula (1) to calculate the integral probability $P_{det}(x, y)$ for all points in the area monitored by these observers.

In [4], stationary hydroacoustic observation devices operating in passive mode, with known coordinates, were considered on a plane equipped with the Cartesian coordinate system. A unified detection probability field of an object, $P_{det}(x, y)$, was calculated for the above devices over the area, taking the anisotropy of the hydroacoustic field into account.

This paper deals with stationary observers operating in active mode, with known coordinates, on a plane equipped with the Cartesian coordinate system *XOY*. A unified detection probability field of an object, $P_{det}(x, y)$, is calculated for all observers over the area depending on the altitude of the observers, the altitude of the object, and the terrain relief.

1. BASIC RELATIONSHIPS

Consider the maximum detection range [5]

$$R_{\max} = \sqrt[4]{\frac{E_{tra} G A \delta_c}{E_{rec\,\max} (4\pi)^2}},$$

where E_{tra} is the transmitted power; $E_{rec\,\max}$ is the power of the signal received by the observer; G is the directivity of the transmitting antenna; δ_c is the radar cross section (RCS) of the object; A is the effective aperture of the receiving antenna; finally, $R_{\max}$ is the maximum detection range, i.e., for a distance $R \leq R_{\max}$, the target will be detected with a probability of correct detection (P_{det}) not smaller than the permissible one for the given false alarm probability P_{fal} and minimum permissible detection probability $P_{det\,\min}$. This definition generally refers to a single (instantaneous) observation under established dependencies between the observation time T_0, the signal-to-noise ratio ρ, P_{det}, and P_{fal}.

The parameters E_{tra}, G, and A are associated with the observer and, therefore, do not depend on R. The parameter δ_c is also independent of the distance to the object R [1]. The energy of the received signal, like its power, is inversely proportional to the fourth power of the distance to the object [5]. Thus, the power of the signal received by the observer can be written as

$$E_{rec} = \beta / R^4, \qquad (2)$$

where β combines all the parameters independent of R.

The criterion for the possibility of detecting a signal against a noisy background is the signal-to-noise ratio ρ, defined as [6]

$$\rho = 2 \int_{\tau} P_{rec} dt / P_{noi}, \qquad (3)$$

where P_{rec} is the power of the useful signal at the input of the observer's receiving device, and P_{noi} is the spectral power density of the noise. Integration is performed over the signal transmission time τ. That is, the problem is reduced to a classical energy receiver.

Thus, the detection problem is to test two hypotheses: H_0 (only noise is received) and H_1 (a mixture of a useful signal and noise is received). The decision on the presence of a useful signal is made according to the Neumann–Pearson criterion: the optimal detection system shall maximize the probability of correct detection P_{det} under a fixed value of the false alarm probability P_{fal}, i.e., $P_{det} \Rightarrow \max$ with $P_{fal} = \text{const}$ [1].

2. CALCULATING A UNIFIED DETECTION PROBABILITY FIELD FOR MULTIPLE OBSERVERS

In active mode, detection is performed based on the processing results of the signal transmitted and received by an observer in the presence of interference. The decision on the presence or absence of a signal from an object is made periodically, after the preliminary processing of the realization of a Gaussian random process $X(t)$ with zero mean received during an observation (accumulation) interval of duration T_0. (This can be a single pulse or a pulse train.) In the absence of a signal from the object, the random process $X(t)$ has variance σ_{noi}^2; in the presence of a useful signal from the object, variance $\sigma_{sig}^2 + \sigma_{noi}^2$.

As is known [7], the density of a statistic y for the energy receiver (3), as the sum of squares of Gaussian random variables with zero mean and variance σ^2, is described by the central χ^2-distribution:

$$f(y) = \frac{1}{(2\sigma^2)^{n/2}\Gamma(n/2)} y^{n/2-1} e^{-y/2\sigma^2},\ y \geq 0,$$

where $\Gamma(\cdot)$ denotes the gamma function, and n is the number of degrees of freedom. In the case of digital analysis, it is determined by the number N of averaged sample energy estimates: $n = 2N$; and in the case of analog processing, $n = 2T_0\Delta f$, where T_0 is the duration of the transmitted signal (pulse), and Δf is the filter bandwidth.

The numerical characteristics (the first two moments) of the χ^2-distribution are given by [8]

$$m_\chi = n\sigma^2,\ \sigma_\chi^2 = 2n\sigma^4.$$

For the hypothesis H_0, we have $m_0 = n\sigma_{noi}^2$ and $\sigma_0^2 = 2n\sigma_{noi}^4$; for the hypothesis H_1, $m_1 = n(\sigma_{noi}^2+\sigma_{sig}^2)$ and $\sigma_1^2 = 2n(\sigma_{noi}^2+\sigma_{sig}^2)^2$. Let us define the signal-to-noise ratio as $\rho = \sigma_{sig}^2 / \sigma_{noi}^2$, then

$$m_1 = n\sigma_{noi}^2(1+\rho),$$
$$\sigma_1^2 = 2n[\sigma_{noi}^2(1+\rho)]^2 = 2n\sigma_{noi}^4(1+\rho)^2. \quad (4)$$

For sufficiently large n, the χ^2-distribution is well approximated by the Gaussian distribution $N(m_\chi, \sigma_\chi^2)$.

Based on formulas (2)–(4), under a fixed P_{fal}, it is possible to construct the dependence of P_{det} on the distance R to the target, i.e., $P_{det}(R)$, or the dependence of P_{det} on the target's RCS, i.e., $P_{det}(\delta_c)$. The qualitative nature of these dependencies for two fixed values δ_c is shown in Fig. 1, where $\delta_{c1} > \delta_{c2}$. With regard to surveillance observers, by conventional assumption, reliable detection is ensured at $\rho > 25$ [6].

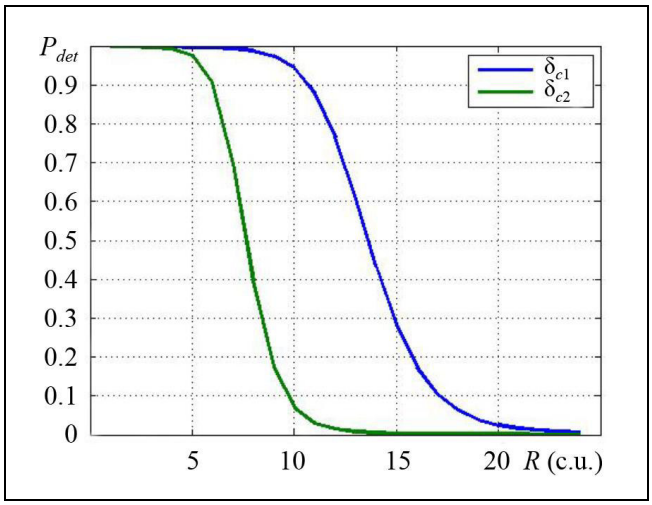

Fig. 1. The dependence of P_{det} on the distance to the target under fixed P_{fal} for different δ_c.

This dependence describes the primary processing stage. For a given false alarm probability $P_{fal} = \alpha$, the probability of correct detection P_{det} grows with increasing ρ, and ρ decreases monotonically with increasing the distance R to the target.

When solving the detection problem, it is necessary to consider the influence of the terrain relief. In other words, $P_{det}(R)$ is the conditional density

$$\{P_{det}(R) \mid P_{los}\}, \quad (5)$$

where $P_{los} = 1$ in the LOS region and $P_{los} = 0$ outside it (in the shadow region).

As a rule, the observer's receiving system operates periodically, accumulating and processing the signal for a fixed time T (a single observation period), during which the position of the detected object and its speed remain almost invariable. After this time, a final decision is made on the presence/absence of the object.

Since each observer generally operates with a particular accumulation time, the integral probability formula (1) cannot be applied. If the accumulation times of all observers were reduced to the same (basic) value T_{bas} = const, it would be possible to construct a unified probability map $P_{det}(x, y)$ in XOY coordinates for all observers located in a given area.

According to the paper [4], for a group of observers with different single observation times T_i, the prob-

abilities of non-detection during the time T_i, $P_{ndet\,i}(T_i)$, can be recalculated into the probabilities of non-detection during the basic time T_b, $P_{ndet\,i}(T_b)$, as follows:

$$P_{ndet\,i}(T_{bas}) = \left[P_{ndet\,i}(T_i)\right]^{T_{bas}/T_i}. \quad (6)$$

Consequently, the probabilities of non-detection for two observers with different observation times can be combined using formula (6) into the probability of non-detection with a unified time (recalculated into the observation time of one observer):

$$P_{ndet\Sigma}(T_2) = P_{ndet2}(T_2)P_{ndet1}(T_2) = P_{ndet2}(T_2)\left[P_{ndet1}(T_1)\right]^{T_2/T_1}.$$

Let an observer in the area have coordinates (x_l, y_l, H_l), where H_l is the altitude of the observer's location. Then, using the above dependence $P_{det}(R)$ for a given altitude h and the expression (5), it is possible to calculate the probability of non-detection for all area points (x, y), i.e., to form a field (matrix) of the conditional probabilities of non-detection $P_{ndet}(x, y \mid h)$.

With the probabilities of non-detection obtained for each observer by formula (6), one can calculate a unified (total) field for all observers:

$$P_{ndet}(x, y) = \prod_{l=1}^{L} P_{ndet\,l}(x_l, y_l, H_l)^{T_M/T_l}, \quad (7)$$

where L is the number of observers; T_l is the observation duration for the lth observer; $T_M = \max[T_l]$; (x_l, y_l) are the coordinates of the lth observer; finally, $P_{ndet\,l}$ is the probability of non-detection for the lth observer. Thus,

$$P_{det}(x, y) = 1 - P_{ndet}(x, y).$$

In addition to setting a single observation time for all observers, there is another obvious requirement to calculate a unified detection probability field: all observers shall have **the same probability of false alarms**, $P_{fal} = \alpha$.

3. SIMULATION RESULTS

The simulation was performed in MATLAB.

Figure 2 shows the map used for the simulation (the right-hand side is a color scale of altitudes; hereinafter, the scales of altitudes and distances are specified in conventional units (c.u.)).

Two stationary observers with coordinates (x, y) and altitude H (5 c.u. above the ground) were considered:

– the first observer with ($x_1 = 30$, $y_1 = 60$, $H_1 = 263$);

– the second observer with three possible coordinates (see the label "+" in Fig. 4b), i.e., ($x_2 = 85$, $y_2 = 56$, $H_2 = 193$), ($x_2 = 83$, $y_2 = 75$, $H_2 = 104$), and ($x_2 = 76$, $y_2 = 92$, $H_2 = 119$).

The distance between the first and second observers is the same in all three variants.

Two variants of the object's flight altitude were considered: $h_1 = 600$ c.u. (a high-altitude object) and $h_2 = 200$ c.u. (a VLA object).

Figure 3 shows an example of calculating the unified detection probability field (DPF) for two observers monitoring a high-altitude object (h_1). The right-hand side of the figure is the color scale of the detection probabilities.

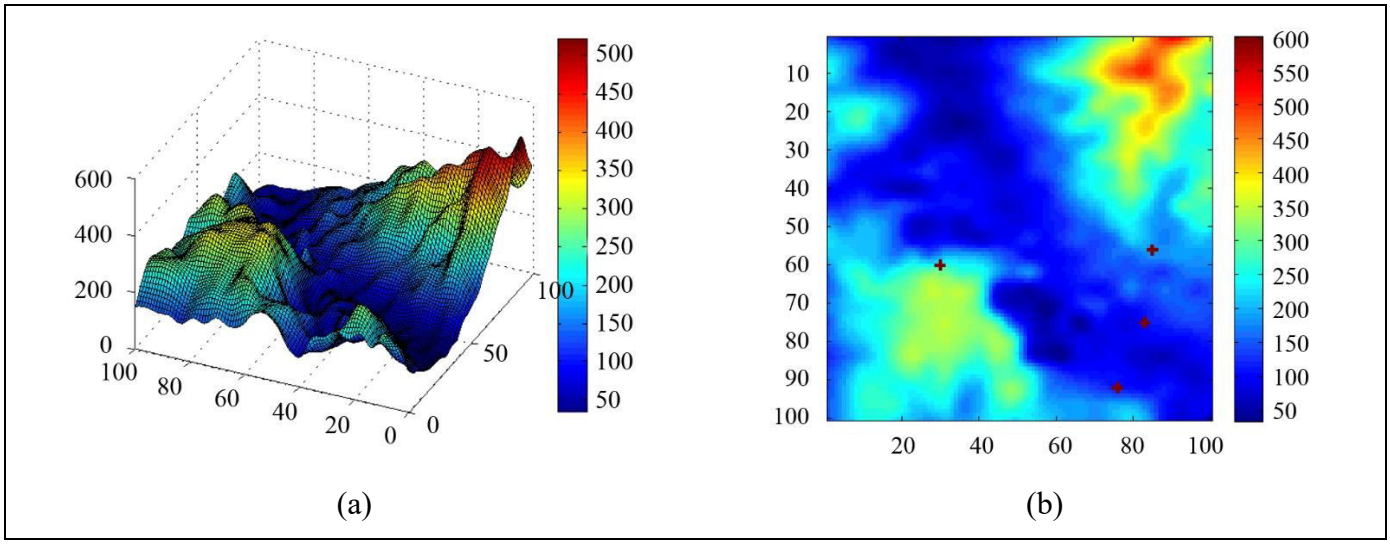

Fig. 2. The map used for simulation: (a) the 3D projection and (b) the 2D projection.

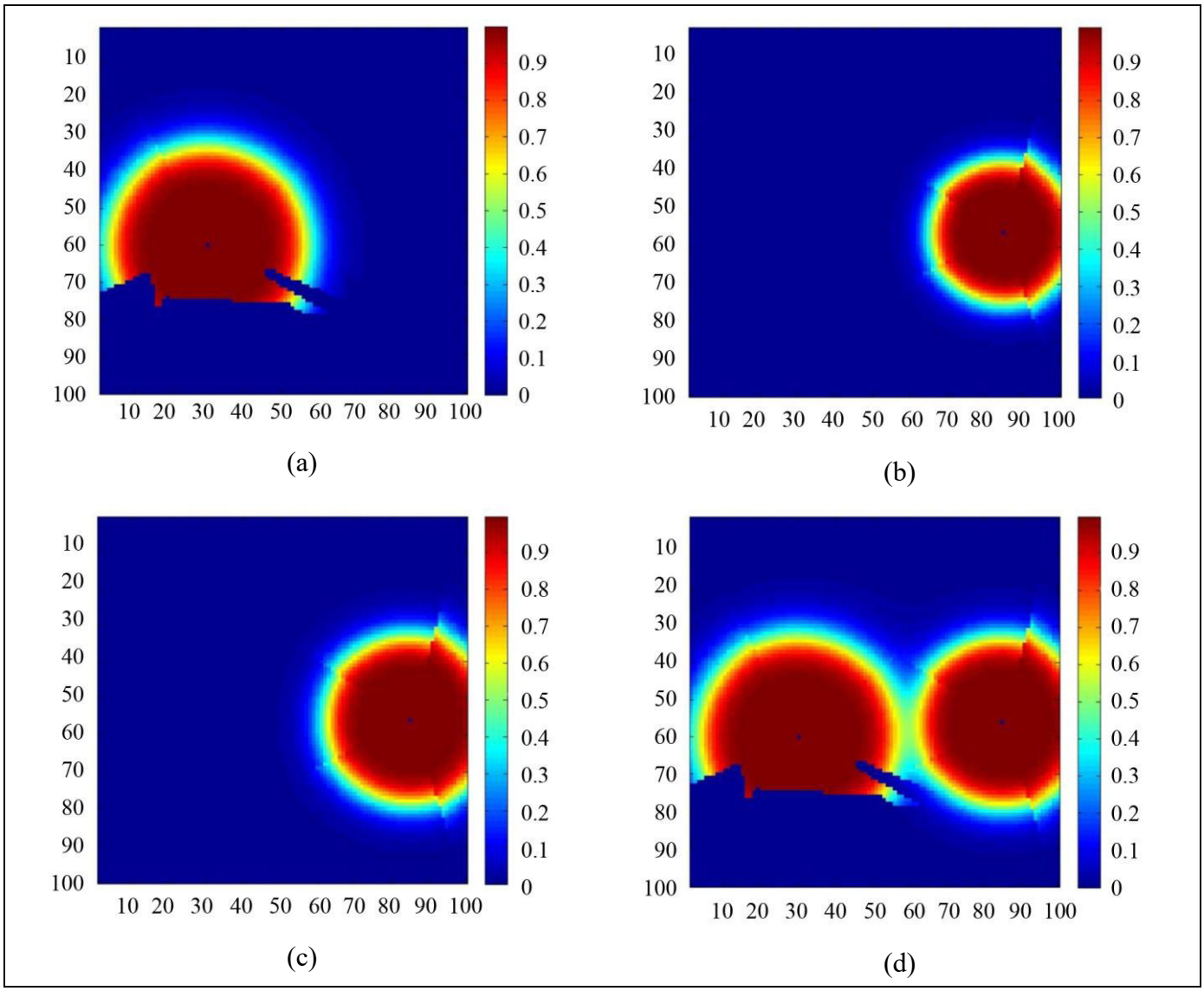

Fig. 3. The unified DPF for two radars monitoring a high-altitude object (h_1): (a) the DPF P_{det1} for the first observer (the observation time T_1); (b) the DPF P_{det2} for the second observer (the observation time T_2); (c) the DPF P_{det3} for the second observer recalculated by (6) (the observation time T_1); and (d) the unified DPF $P_{det} = 1 - P_{ndet}$ for two observers calculated by (7).

According to Fig. 3, both of the observers have a circular directional pattern, but the first observer is located on a hillside (see Fig. 2); therefore, the slope creates a shadow region for it (see the lower part of Fig. 3a). The detection probability fields for the first and second observers do not overlap; in view of $T_1 > T_2$, after the reduction to a single observation time by formula (6), their unified DPF almost forms a continuous detection probability field (Fig. 3d).

Figure 4 presents an example of calculating the unified DPF for two observers when working with the terrain relief (h_2). The right-hand side of Fig. 4a is the color scale of altitudes, and the right-hand sides of Figs. 4b–4e are the color scales of the detection probability.

Figure 4a shows the geographical map sectioned by the object's altitude and the relief exceeding h_2, which is the region where the object cannot appear. This relief defines the "mask of prohibited regions" for the object since the relief altitude in the regions is greater than the object's altitude h. The DPF on the maps is shaded in green.

The shadow region clearly seen in Fig. 4b is formed by the slope of the hill on which the first observer is located. The shadow appears due to $H_1 > h_2$: the slope of the hill forms a clutter notch completely covering the field of view of the first observer in this sector at the altitude h_2.

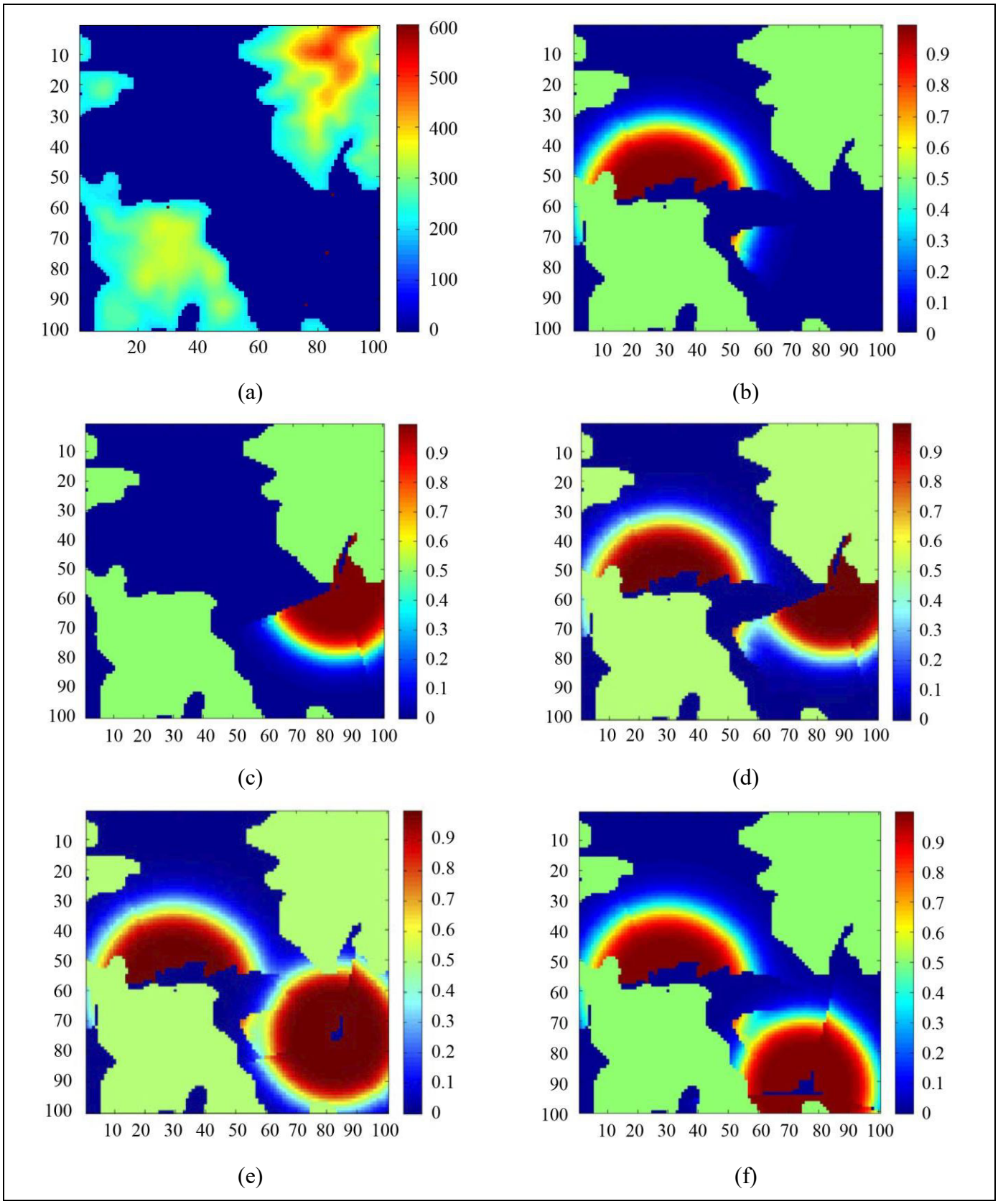

Fig. 4. The unified DPF for two observers when working with the terrain relief (h_2): (a) the map section at altitude h_2; (b) the DPF P_{det1} for the first observer (the observation time T_1); (c) the DPF P_{det3} for the second observer recalculated by (6) (the observation time T_1); (d) the unified DPF $P_{det} = 1 - P_{ndet}$ for two observers calculated by (7) (the coordinates of the second radar are ($x_2 = 85$, $y_2 = 56$, $H_2 = 193$)); (e) the unified DPF $P_{det} = 1 - P_{ndet}$ for two observers calculated by (7) (the coordinates of the second radar are ($x_2 = 83$, $y_2 = 75$, $H_2 = 104$)); (f) the unified DPF $P_{det} = 1 - P_{ndet}$ for two observers calculated by (7) (the coordinates of the second radar are ($x_2 = 76$, $y_2 = 92$, $H_2 = 119$)).

Figure 5 presents an example of forming LOS regions (green) and shadow regions (red) in a given direction for the map coordinates (x_1 = 30, y_1 = 60, H_1 = 266) and (x_2 = 100, y_2 = 40) in the case h = 200.

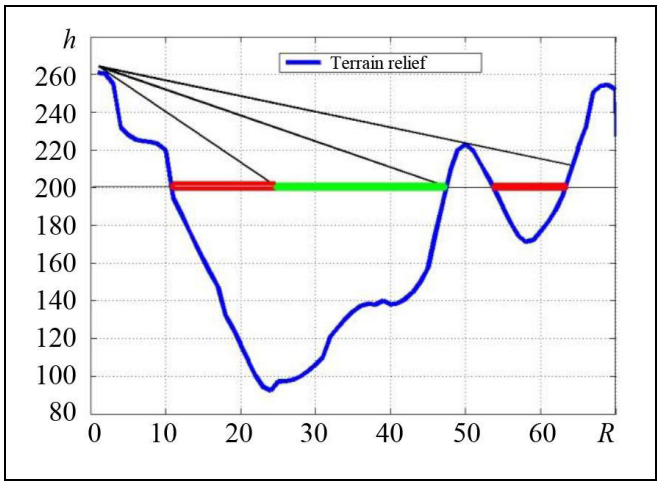

Fig. 5. LOS and shadow regions.

Direct comparison of Figs. 4 and 5 indicates that for object detection considering the terrain relief, the mutual position of the observers is crucial when forming a unified DPF.

Taking the terrain relief into account, the circular diagram of the DPF is significantly distorted and depends on four parameters:

– the distance R between the object and the observer,

– the object's altitude h,

– the observer's antenna altitude H, and

– the direction to the object.

Nevertheless, under a fixed altitude h of the object, it is also possible to calculate the probability of detection for all points in the area, $P_{det}(x, y)$, i.e., construct a detection probability map for h = const. Such a set of probability maps can be described by a three-dimensional matrix in the coordinates (x, y, h) for a series of fixed altitudes h_j ($1 \le j \le J$) under fixed coordinates and altitudes H_l ($1 \le l \le L$) of each observer. This procedure yields a so-called multilayer map.

Obviously, the above calculation of a unified DPF using formulas (5)–(7) can be easily generalized to an arbitrary number of observers.

4. APPLICATION OF A UNIFIED DETECTION PROBABILITY FIELD

There exists a fairly wide range of problems where the calculation of a unified detection probability field for several observers is of interest. These problems can be divided into two groups.

The first group. When detecting a high-altitude object, calculating the unified probability field $P_{\text{det}}(x, y)$ for the entire area allows combining information from observers with different technical characteristics (see Fig. 3). In this case, the map is one-dimensional.

A unified probability field $P_{\text{det}}(x, y, h)$ calculated for the entire area can be superimposed on a geographical map. Then a human operator can visually identify weak spots (the regions of small P_{det}) and, e.g., send additional resources there or change the location of observers to cover the weak spots. (As is well known, the human eye copes very well with such a task.) Calculation of a unified DPF in the form of a multilayer map simplifies the solution of optimization problems. For example, in the case of object detection, it simplifies the optimal placement of a fixed number of radars in the area according to a given criterion. Such a problem has not been considered so far.

The second group includes problems of penetrating through an area monitored by several stationary observers or evading detection by these observers. In the literature, this class of problems is referred to as control problems for mobile objects in a conflict environment. Thus, the problem is reduced to route planning by a probabilistic criterion. Route planning on a single-layer map with obstacles is a well-studied problem, both theoretically and practically. Classical graph search algorithms, such as Dijkstra's algorithm, are typically used to solve it. They are widely and successfully applied to a single layer of the map. However, it is challenging to extend these algorithms to multilayer maps [9].

In the case of a probabilistic criterion, route selection consists in minimizing the accumulated probability according to formula (1) for a single-layer map [10–12], but a unified DPF is not used. In [13], the problem of selecting the trajectory of a maneuvering object and the law of its velocity in a three-dimensional anisotropic signal propagation environment was considered, provided that several observers located in a given area are trying to detect it. (For each observer, P_{det} was calculated separately.) In the problems mentioned, the preliminary calculation of a unified DPF would significantly facilitate dynamic programming.

For example, when an object avoids the terrain at a given altitude (a nap-of-the-earth flight), an available multilayer map of the unified DPF of the area can be

easily reduced to a single-layer one; on the latter, a route is planned much more easily. In addition, on a single-layer map, the operator can visually select a route, as shown in Fig. 6 (the map of Fig. 4d is used).

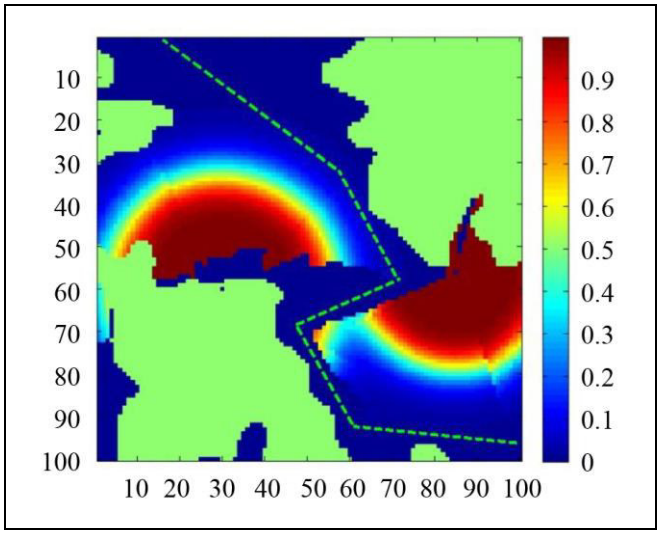

Fig. 6. The route planned by the operator.

CONCLUSIONS

The availability of a unified detection probability field for all observers monitoring a given area provides the following capabilities:

– It can be generalized to the detection of a moving object, taking the terrain relief into account, by adding two coordinates (the observer's altitude and the object's altitude).

– It can be used for visualization, i.e., presented to a human operator.

– It can be selected as a basis for combining detection probabilities from other sources of information (e.g., from observers located on moving objects [14]). Passive location systems can also be applied to detect and track moving objects [14]. In addition to location-based detection methods, other groups of methods for detecting moving objects are currently being investigated [15], in particular, infrared (thermal) detection, radio frequency scanning, detection using optical cameras, detection of acoustic signals, and approaches involving classical machine learning methods with feature extraction as preprocessing (artificial intelligence). DPFs from such information sources can be included in a unified detection probability field if there are models for calculating the probabilities of object detection by these means.

REFERENCES

1. *Radiolokatsionnye sistemy* (Radar Systems), Berdyshev, V.P., Ed., Krasnoyarsk: Siberian Federal University, 2011. (In Russian.)
2. Manuilenko, V.G. and Udin, E.G., *Teoreticheskie osnovy krylatykh upravlyaemykh raket* (Theoretical Foundations of Winged Guided Missiles), St. Petersburg: ITMO University, 2020. (In Russian.)
3. Abchuk, V.A. and Suzdal', V.G., *Poisk ob"ektov* (Search for Objects), Moscow: Sovetskoe Radio, 1977. (In Russian.)
4. Rudko, I.M., Unified Field of Detection Probabilities for Heterogeneous Means of Observations, *Tomsk State University Journal of Control and Computer Science*, 2020, no. 53, pp. 93–101. (In Russian.)
5. *Teoreticheskie osnovy radiolokatsii* (Theoretical Foundations of Radar), Shirman, Ya.D., Ed., Moscow: Sovetskoe Radio, 1970. (In Russian.)
6. Zalogin, N.N., Kalinin, V.I., and Sknarya, A.V., The Active Location with the Use of Ultrawide-Band Chaotic Signals, *RENSIT*, 2011, vol. 3, no. 1, pp 3–17. (In Russian.)
7. Levin, B.R., *Teoreticheskie osnovy statisticheskoi radiotekhniki. T. 2* (Theoretical Foundations of Statistical Radio Engineering. Vol. 2), Moscow: Sovetskoe Radio, 1968. (In Russian.)
8. Cramer, H., *Mathematical Methods of Statistics*, Princeton: Princeton University Press, 1946.
9. Motorin, D.E. and Popov, S.G., Multi-Criteria Path Planning Algorithm for a Robot on a Multilayer Map, *Information and Control Systems*, 2018, no. 3, pp. 45–53. DOI: 10.15217/issn1684-8853.2018.3.45 (In Russian.)
10. Dogan, A., Probabilistic Path Planning for UAV, *Proceedings of 2nd AIAA "Unmanned Unlimited" Conf. and Workshop & Exhibit.*, San Diego, CA, 2003, pp. 1–7.
11. Abramyants, T.G., Galyaev, A.A., Maslov, E.P., et al., Evasion of a Moving Object from Detection by a System of Heterogeneous Observers in the Threat Environment, *Automation and Remote Control*, 2017, vol. 72, no. 5, pp. 345–354.
12. Dobrovidov, A.V., Kulida, E.L., and Rudko, I.M., Control of Object Movement in Threat Environment, *Control Sciences*, 2011, no. 3, pp. 64–75. (In Russian.)
13. Dobrovidov, A.V., Kulida, E.L., and Rudko, I.M., Optimization of the Object Movement Path on the Probabilistic Criterion in the Mode of Passive Sonar in an Anisotropic Medium, *Control Sciences*, 2014, no. 4, pp. 31–37. (In Russian.)
14. Khawaja, W., Ezuma, M., Semkin, V., et al., A Survey on Detection, Classification, and Tracking of UAVs Using Radar and Communications Systems, *IEEE Communications Surveys and Tutorials*, 2025, vol. 28, pp. 3272–3310. DOI: https://doi.org/10.1109/COMST.2025.3554613
15. Zitar, R.A., Al-Betar, M., Ryalat, M., and Kassaymeh, S. A Review of UAV Visual Detection and Tracking Methods, *Proc. 9th Annual Conf. on Computational Science & Computational Intelligence (CSCI'22)*, Las Vegas, NV, 2022, paper no. ffhal-04108638f.

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

Received June 30, 2025,
and revised December 8, 2025.
Accepted December 16, 2025

Author information

Rudko, Igor Mikhailovich. Cand. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Science, Moscow, Russia
✉ igor-rudko@mail.ru
ORCID iD: https://orcid.org/0000-0001-6146-9177

Cite this paper

Rudko, I.M., A Unified Detection Probability Field for a Group of Stationary Observers. *Control Sciences* **1,** 70–78 (2026).

Original Russian Text © Rudko, I.M., 2026, published in *Problemy Upravleniya*, 2026, no. 1, pp. 81–89.

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

33RD INTERNATIONAL CONFERENCE ON PROBLEMS OF COMPLEX SYSTEMS SECURITY CONTROL

In December 2025, the 33rd International Conference on Problems of Complex Systems Security Control took place at the Trapeznikov Institute of Control Sciences, the Russian Academy of Sciences (ICS RAS), Moscow.

The conference was attended by 86 authors from 28 organizations, who presented 63 papers. The conference was divided into the following sections:

1. General theoretical and methodological issues of security support;
2. Problems of economic and sociopolitical security support;
3. Problems of information security support;
4. Cybersecurity. Security aspects in social networks;
5. Ecological and technogenic security;
6. Modeling and decision-making for complex systems security control;
7. Automated systems and means of complex systems security support.

I.V. Chernov, the conference opener and chair of the plenary session, gave the floor to *D.I. Pravikov*, who dedicated his speech to the memory of *Sergey P. Rastorguev* (1958–2017), an outstanding Russian scientist, Dr. Sci. (Eng.), Prof., and author of over 100 research and scientific-popular publications of special and interdisciplinary nature. Rastorguev's fundamental and applied research interests were very broad to cover various fields of computer science, programming, cryptography, antivirus protection, avatarization (the transfer of the basics of biological life and consciousness to a computer system), philosophy and sociology, learning theory, and pedagogy. He is a founder of the Russian scientific school of information confrontation, the author of many scientific works on the theory and practice of information warfare, which have become classics of this fundamentally new scientific direction of the recent past. The term "information warfare" itself came to be understood in the Russian literature in Rastorguev's interpretation after the release of the same-name monograph in 1999.

The basis of his philosophical views was formed at a time when neural networks became a topical and rapidly developing field of science and engineering. As emphasized in the speech, Rastorguev's postulate about one of the fundamental properties of networks arose at that time: learning takes place through not only the creation but also the destruction of connections and the elimination of elements. This postulate, seemingly unobvious at first glance, underlies the description of the main properties of information weapons: during (self-)training, complex systems can (self-)restrict or suppress themselves.

According to the speaker, Rastorguev's scientific legacy and heritage are so profound and multifaceted that the research community has yet to comprehend and understand them. Operating intuitively understandable basic meanings ("life," "death," "information," "knowledge"), Rastorguev went far beyond the ordinary in his conclusions and philosophical constructs and aimed at a critical understanding of the surrounding reality. The scientific results obtained by S.P. Rastorguev remain extremely relevant today.

The program part of the conference was opened with a comprehensive review by *T.S. Akhromeeva*, *G.G. Malinetskii*, *and S.A. Toropygina*, entitled "New Approaches to the System Analysis of Large-Scale Projects." As stated in the first part of the review, in modern conditions, conventional interdisciplinary system analysis-based approaches to the problems of analysis, structuring, and management of the development of large-scale systems in various fields are undergoing a deep crisis, breaking down into a set of weakly interconnected methodological directions.

According to the authors, the main practice-suggested reason is that the conventional approaches fail to cope with the main contemporary problems of designing the development of large-scale systems, namely, analysis, prediction of their development dynamics, and risk management. Here, Akhromeeva–Malinetskii–Toropygina's recipe is to create new tools for pressing state development tasks based on the ex-

perience of applying methods and models of self-organization theory (synergetics) in the analysis of complex problems and the development of large-scale projects. This methodology can be effectively used to describe the properties and characteristics of competing large-scale projects, as well as the rivalry between countries, blocs, civilizations, and ethnic groups in the economy, military, or other areas of confrontation.

In particular, while considering the advantages of synergetics and analyzing the methodology for solving the above tasks, the authors noted that this approach eliminates the "curse of dimensionality" by identifying order parameters (primary variables and degrees of freedom) that gradually—over time—start determining the dynamics of other characteristics of the complex system under study. In other cases, where the key factors are so-called ultra-fast processes (compared to them, all others get "frozen"), there is an opportunity for simplification associated with the use of order parameters, which (in this case, on the contrary) describe fast variables.

The second part of the review was devoted to publications reflecting the use of various fragments of this approach to solve technological development tasks of the military-industrial complex of Russia's geopolitical opponents over a twenty-year horizon. The authors concluded that the high-level tension in international relations convincingly demonstrates the need to develop system analysis methods and technologies in order to manage the interaction (confrontation) of competing agents.

A distinctive feature of the conference was many interesting and diverse papers devoted to solving a wide range of problems related to the secure and sustainable socio-economic development of Russia in the current (extremely difficult) conditions.

V.V. Tsyganov's paper "Secure Sustainable Development Mechanisms in a Multipolar World," considered the problems of the country's economic development in the context of intensifying contradictions and the negative impact of globalization processes. As noted therein, the Concept of Sustainable Development (put forward at the UN Conference on Environment and Development in Rio de Janeiro, 1992) pays central attention to the interests of not only the current but also future generations; however, this concept is now being exploited by the global capital center in its own interests as a tool to influence and manipulate environmental standards by applying environmental taxes to imports from developing countries. With such methods, globalists impose excessive environmental requirements on the production and goods of developing countries, which cannot be satisfied without expensive Western technologies. In this regard, it is necessary to modify sustainable development mechanisms in order to make their application secure for developing countries. This primarily concerns mechanisms for the secure and sustainable development of key economic agents in these countries, ensuring their independence, as well as the ability to adapt and self-organize in the face of regional or global changes.

The author presented a robust mechanism for solving this problem, including procedures for forming security indicators, algorithms for calculating the norms of the indicators to categorize them, as well as the convolutions of the resulting categories for an integrated assessment of sustainable development security to incentivize decision-makers. The practical application of the robust mechanism was illustrated by a detailed example of ensuring the secure and sustainable development of Russian rail transport under environmental protection requirements (the task of managing the overhaul of the diesel locomotive fleet within the maintenance program of JSC Russian Railways).

In the paper "Problems of the Development of the Russian Financial Market and Its Strategic Planning," *A.E. Abramov*, *M.I. Chernova*, *and F.S. Levin* comprehensively assessed the intermediate implementation results of the federal project "Development of the Financial Market" and analyzed the existing problems of financial regulation and strategic planning hindering its success. As stated in the paper, a necessary condition for the successful implementation of the Russian financial market development strategy based on domestic savings and investments is its integration into the strategic planning system for the development of the Russian economy as a whole. This necessitates greater engagement of the financial regulator in market development, which should largely be reduced to creating conditions for the growth of innovative business activity through the regulator's traditional functions.

According to Federal Law no. 172-FZ "On Strategic Planning in the Russian Federation," the Bank of Russia is a participant in the state long-term planning system. However, nowadays, a serious problem is that Federal Law no. 86-FZ "On the Central Bank of the Russian Federation (Bank of Russia)" describes the functions of the mega-regulator in a very fragmented manner, limiting them to the development and stability of the financial market. As a result, several pressing tasks of financial regulation and supervision (over-

coming "market failures," supporting competition, protecting the rights of investors and consumers of financial services, ensuring the stability of financial institutions, etc.) have almost no intersection with the objectives of the Bank of Russia as defined in the latter federal law.

The authors emphasized a series of challenges to be settled, notwithstanding the progress in recent years (significant in several areas) in integrating the financial market development policy into the strategic planning system, which was largely implemented by overcoming a definite gap between policy documents on financial market development and programs for the development of the Russian economy as a whole. Based on the comprehensive and detailed analysis results of the main target-setting and regulatory documents, as well as the key development indicators of the Russian financial market for the period up to 2030 (see the paper), Abramov et al. identified as the main problem the insufficiently justified quantitative targets reflecting the dynamics of financial market development, as well as the fragmentary nature of measures aimed at stimulating domestic long-term savings. To solve this problem in the long term, it is necessary to specify completely the key functions of the financial regulator at the legislative level, as well as to optimize the structure of strategic financial planning documents (in particular, to eliminate duplication and unify the system of predictive indicators for the development of the financial market for 2030, contained in the Strategy for the Development of the Financial Market of the Russian Federation until 2030 and the federal project "Development of the Financial Market").

The paper "Risks in a Financial Market and Their Assessment" by *A.D. Kozlov and N.L. Noga* presented an original methodology for assessing risks when conducting operations in a financial market. This methodology is based on the combined use of econometric and fuzzy logic methods; structurally, it consists of a sequence of interconnected stages as follows.

The first stage is to examine a financial market and determine a set of parameters characterizing both systematic and unsystematic risks. The resulting set is divided into groups of financial, internal, and external economic parameters, as well as other parameters (if necessary). In the next stage, expert procedures are used to select from the initial set the subsets of parameters with the greatest impact on the risks of financial losses, and a fuzzy knowledge base is formed accordingly. These parameters are written as linguistic variables normalized in the range from 0 to 1. Then, a table of production rules is formed, where each row is assigned a specific risk level. In the third stage, a linear multiple regression model is constructed based on this table, and standardized equation coefficients are determined for their ranking. The fourth (final) stage is to identify the variables with the greatest impact on the risks of financial losses and to check their interdependence. The quality of the resulting model is assessed by computing the multiple determination coefficients. The statistical significance of the regression coefficients and the regression equation as a whole is verified using Fisher's F-test and Student's t-test. In conclusion, the authors provided an example illustrating the practical possibilities of using the methodology by investors to predict financial losses under uncertainty and risk.

Note a large group of conference participants who considered a wide range of methodological and applied problems of managing the socio-economic development of Russia, its regions, and economic agents: *N.I. Komkov*, *V.V. Sutyagin*, *and N.N. Volodina* ("Possible Coordination of the Effectiveness of Economic and Social Development Management"); *V.A. Irikov and D.R. Gonchar* ("The Breakthrough Socio-Economic Development of the Country in 2025–2030: New Directions, Features, Opportunities, and Examples of Multiple Growth"); *Z.K. Avdeeva and S.V. Kovriga* ("The Strategic Planning Graph in the Russian Federation's National Security System"); *V.V. Shumov* ("Conflict Modeling Using Methods of Military Cybernetics and Security Studies"); *V.V. Nicheporchuk* ("Intelligent Services for Territorial Security Management"); *A.V. Rozhnov* ("Justifying the Development of an Information and Analytical System When Implementing Hybrid Analysis Technology Models for the Environment in Predictive Modeling"); *V.I. Medennikov* ("Digital Tools Providing a More Environmentally Friendly and Safer Path for Humanity Development"); *O.B. Bairamov* ("Combined Application of Penalty Models and Environmental Insurance for Managing the Risks of Water Basin Degradation"); *T.Kh. Usmanova and O.V. Dem'anova* ("Formation of the Socio-Economic Security of the North–South International Transport Corridor"); *L.E. Mistrov* ("A Method for Distributing Heterogeneous Resources to Ensure Conflict Resilience in the Interaction of Organizational and Technical Systems"); *D.R. Gonchar* ("Population Preservation as a New Target Indicator for the More Successful Socio-Economic Development of the Country in 2025–2030"); *V.O. Sirotyuk and L.V. Bogatyreva* ("Compre-

hensive Security of the Subjects of an Intellectual Property Management System"); and *N.N. Lanter* ("Features of Arctic Concepts of Foreign Countries in 2025–2030").

A series of interesting papers were devoted to theoretical and practical problems of developing a scenario analysis methodology and simulation modeling technologies for managing the development of complex socio-economic systems at the governmental, industrial, regional, and object levels. Among them, let us mention the following: "The Vulnerability of a Complex System: A Hierarchy of Concepts" (*D.A. Kononov and I.V. Chernov*); "The Sustainable and Secure Development of Complex Systems, Southern Russia, Cognitive Modeling" (*G.V. Gorelova*); "Analysis of Verification Methods and Technologies for Scenario Management Models" (*V.L. Shul'ts*, *I.V. Chernov*, and *A.B. Shelkov*); "Support for the Management of Complex Socio-Economic Systems Using Situational Scenario Analysis Methods" (*M.Yu. Dmitrieva*, *I.D. Butusov*, and *Yu.A. Gogoladze*); "The Use of Scenario Analysis in DSSs for Regional Security: A Brief Review" (*N.V. Komanich*); "A Scenario Model for Studying Threats to the Secure Development of Urban Infrastructure" (*M.Yu. Dmitrieva* and *L.V. Bogatyreva*); and "Prospects for Using Scenario Analysis to Ensure AI Information Security" (*E.D. Ermolaeva*).

The paper "Justifying Comprehensive Assessment Indicators for the Security of Critical Information Infrastructure of Oil and Gas Companies" by *D.I. Pravikov and V.A. Burkin* was devoted to developing security assurance methods for the production, information, and technology infrastructure of companies extracting hydrocarbon minerals. As noted therein, oil and gas complex facilities are complex sociotechnical systems in which automated process control subsystems, information resources, personnel, and regulatory documents form a single interdependent structure. Managing their security requires passing from regulatory supervision to a quantitatively verifiable and comprehensive (integrated) security assessment to compare facilities, prioritize organizational and technical measures, and justify managerial decisions based on measurable indicators.

The authors presented an original methodology for calculating a comprehensive security indicator for the critical information infrastructure of oil and gas companies (a weighted sum of ten standardized partial metrics). These metrics reflect a wide range of security indicators: critical segments of the information structure are covered by centralized collection and correlation of security events, and nodes are covered by the minimum necessary set of protective measures; the timeliness of eliminating critical vulnerabilities of protected facilities is assessed, and the unscheduled downtimes of industrial systems and the rate of recovery after an incident are estimated; the level of personnel's readiness for actions in abnormal and critical situations is assessed, etc. Appropriate metrics are selected considering industry specifics, and each metric has a direct connection to the actions stipulated by the relevant regulatory, normative, and other organizational documents and requirements. As a consequence, the indicators are comparable, the results can be audited, and operational risk management in the loop of industrial and information security becomes more efficient. Generally speaking, the approach proposed is a practice-oriented tool for managing the security of critical information infrastructure, linking target security levels to particular actions and resources; moreover, it ensures the reproducibility of assessments and the transparency of decisions.

Traditionally, many conference papers deal with various information security management problems. In this thematic group, note the following: "An Information Security Model Considering Sanctions" (*N.G. Kereselidze*); "Improving Video Data Security by Using a Noise-Resistant Video Steganography Method Based on Deep Neural Networks" (*S.A. Shustov and R.V. Meshcheryakov*); "Approaches and Tools for Collecting Information from Open Sources to Monitor and Identify Information Security Threats" (*L.N. Loginova and A.D. Drozdov*); "REST Services Based on the C# Language to Provide Information Protection in Windows–Linux Environments" (*R.E. Asratyan*, *S.S. Vladimirova*, *E.A. Kurako*, and *V.L. Orlov*); "An Online Reputation Monitoring Algorithm Based on Search Queries" (*D.S. Ignatov*); "Architectural Features and Specifics of Ensuring Information Security in High-Load Information Systems" (*A.D. Domashkin and L.N. Loginova*); "Risk Management for Computer Networks with the Tree Topology" (*A.A. Shiroky*); "Modeling Security Threats to Critical Information Infrastructure Facilities in the Republic of Angola" (*I.F. Mikhalevich and A.M. Fransisco Nelson*); "Using the Principles of Lean Management to Ensure the Information Security of an Industrial Enterprise" (*V.V. Vedishchev and R.V. Batishchev*); "Using Regular Expressions to Manage the Information Security of Intelligent Transport Systems" (*I.F. Mikhalevich and*

D.I. Pchelintsev); "Formalization of Information Security Risk in Intelligent Water Transport Systems as a Fuzzy Linguistic Assessment Based on Decision Theory" (*L.A. Baranov*, *N.D. Ivanova*, and *I.F. Mikhalevich*); and "Modeling a Smart Home Security System" (*Yu.A. Klimenko*, *A.P. Preobrazhenskii*, *and I.A. Tikhonov*).

A number of interesting application-oriented papers were devoted to the security of industrial and transport systems and facilities: "Cybernetics of the Security of Energy Systems with a Nuclear Reactors" (*V.V. Leshchenko*); "Assessment of the Effectiveness of Industrial Safety Management System Audits" (*E.V. Klovach*, *I.A. Kruchinina*, and *V.A. Tkachenko*); "Security and Reliability of Complex Technical Systems" (*S.K. Somov*); "An Approach to Assessing Changes in the Critical Characteristics of Robotic Systems over Time" (*M.V. Mamchenko*); "Reliability of Conductive Elements of Power Equipment with Pulse Modulation" (*O.B. Skvortsov and V.I. Stashenko*); "Mathematical Modeling of the Security of Ground Protective Structures under an External Seismic Wave Impact" (*V.K. Musaev*); "A Simulation Model of Operational Fire Response Phases at Fuel and Energy Complex Facilities with Robust Optimization of Time, Risk, and Resources" (*R.Sh. Khabibulin*); "General Theoretical and Methodological Issues of Security Assurance in the Development of a Flight Controller for UAVs: Control in the Process of Setting Up Experiments" (*D.A. Vol'f and R.R. Galin*); "Forming the Structure of a Flight Safety Management System for Unmanned Aerial Systems Based on Interoperability" (*D.M. Mel'nik*); "Security Assessment for the "Unmanned Aerial Complex–Personnel–Environment" System Based on Scenario Analysis" (*A.G. Davydovskii*); "A Mathematical Model of the Influence of Space Flight Factors on the Quality of Astronaut Performance and the Formation of Ergonomic Risk" (*E.A. Timme*); "Safety of Transport Infrastructure and Vehicle Control Systems" (*L.A. Baranov*, *S.E. Ikonnikov*, and *A.E. Ermakova*); "A Threat Model for a Wheeled Platform Control System with a Multi-Agent Multi-Level Neural Network Implementation" (*O.A. Tel'minov*); "Organizational Features of the Transportation Process on the Moscow–St. Petersburg High-Speed Railway under Construction" (*A.I. Isakova and A.S. Meshcheryakova*); "Development of a Microservice for Analyzing and Predicting the Wear of Railway Contact Network Elements" (*A.S. Ikonnikov*); and "Application of Artificial Intelligence Technologies to Ensure the Information Security of Medical Systems and Devices" (*V.A. Zorin*).

The conference proceedings are published electronically[1] and are also available at the official website: https://iccss2025.ipu.ru/conf_proceedings.

The 34th International Conference on Problems of Complex Systems Security Control is scheduled for November–December 2026 at ICS RAS. The date and time of the conference will be announced in the information letter of the Organizing Committee, which will be published on the official website (https://iccss2026.ipu.ru/) as well as distributed to potential participants, interested parties, and specialized organizations. Also, please contact the Organizing Committee via phone + 7 495 198-17-20 (ext. 1407) or e-mail iccss@ipu.ru. The Technical Secretary of the conference is *Al'fiya Farissovna Ibragimova*.

Academic Secretary of the Organizing Committee
A.B. Shelkov
Event coordinator of the Organizing Committee
L.V. Bogatyreva

Author information

Shelkov, Alexey Borisovich. Cand. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ abshelkov@gmail.com
ORCID iD: https://orcid.org/0000-0003-1408-5212

Bogatyreva, Larisa Vladimirovna. Cand. Sci. (Hist.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ lbogat@mail.ru
ORCID iD: https://orcid.org/0000-0003-2744-0404

Cite this paper

Shelkov, A.B. and Bogatyreva, L.V., 33rd International Conference on Problems of Complex Systems Security Control. *Control Sciences* **1**, 79–83 (2026).

Original Russian Text © Shelkov, A.B., Bogatyreva, L.V., 2026, published in *Problemy Upravleniya*, 2026, no. 1, pp. 90–96.

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

[1] *Materialy 33-ei Mezhdunarodnoi konferentsii "Problemy upravleniya bezopasnost'yu slozhnykh sistem"* (Proceedings of the 33rd International Conference on Problems of Complex Systems Security Control), December 17, 2025, Moscow, Kalashnikov, A.O. and Chernov, I.V., Eds., Moscow: Trapeznikov Institute of Control Sciences RAS, 2025. (In Russian.)