

ISSN 3083-6573 Print
ISSN 3083-6581 Online

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

IT&S INFORMATION TECHNOLOGIES SYSTEMS

3⁽³⁾
2025

TOPICS

- DEEP REINFORCED LEARNING OF DRONES
- EVOLUTION OF VIDEO SURVEILLANCE SYSTEMS
- KNOWLEDGE CONVERGENCE IN DIGITAL HEALTH

Editor-in-Chief: O.M. KHIMICH, (Kyiv, Ukraine)

Deputies Editor-in-Chief: O.Ye. VOLKOV, (Kyiv, Ukraine),
Ye.A. SAVCHENKO-SYNIKOVA, (Kyiv, Ukraine)

Editorial Board: Ali Abbasov Mammad oglu (Baku, Azerbaijan); O.Yu. Azarkhov (Dnipro, Ukraine); I.Ye. Andrushchak (Lutsk, Ukraine); A.V. Anisimov (Kyiv, Ukraine); I. Vlahavas (Thessaloniki, Greece); W. Wojcik (Lublin, Poland); D.O. Volosheniuk (Kyiv, Ukraine); A.M. Hlibovets (Kyiv, Ukraine), O. Gorbunovs (Riga, Latvia); V.F. Gubarev (Kyiv, Ukraine); L.F. Gulyanytskyi (Kyiv, Ukraine); A.M. Gupal (Kyiv, Ukraine); V.V. Zosimov (Odesa, Ukraine); P.I. Kohut (Dnipro, Ukraine); L.M. Kozak (Kyiv, Ukraine); S.L. Kryvyy (Kyiv, Ukraine); V.I. Lytvynenko (Kherson, Ukraine); R. Martínez Béjar (Murcia, Spain); O.V. Palagin (Kyiv, Ukraine); S.L. Pogorilyy (Kyiv, Ukraine); N. Prokofyeva (Riga, Latvia); B. Savchynskyy (Heidelberg, Germany); A.-B. M. Salem (Cairo, Egypt); K.M. Synytsa (Kyiv, Ukraine); V.S. Stepashko (Kyiv, Ukraine); I.V. Surovtsev (Kyiv, Ukraine); L.S. Fainzilberg (Kyiv, Ukraine); A.O. Chykriy (Kyiv, Ukraine); M.I. Schlesinger (Kyiv, Ukraine)

Responsible Executor: H.O. Pezentsali

Editors: N.A. Charchiyan, A.Yu. Vitchenko, O.O. Lysenko

Computer Group: O.V. Tupalskiy, N.S. Stashkova

Media ID R30-05899

Editorial address: Institute of Information Technologies and Systems
of the National Academy of Sciences of Ukraine,
40, Hlushkova Akd. ave., Kyiv, 03187
phone: +380 (44) 526-00-09, e-mail: its.journal.ua@gmail.com,
<https://nasu-periodicals.org.ua/index.php/its>

Головний редактор: О.М. ХІМІЧ (Київ, Україна)

Заступники головного редактора: О.Є. ВОЛКОВ (Київ, Україна),
Є.А. САВЧЕНКО-СИНЯКОВА (Київ, Україна)

Редакційна колегія: Алі Аббасов Маммед огли (Баку, Азербайджан); О.Ю. Азархов (Дніпро, Україна); І.Є. Андрушак (Луцьк, Україна); А.В. Анісімов (Київ, Україна); І. Влахавас (Салоніки, Греція); В. Войчик (Люблін, Польща); Д.О. Волошенюк (Київ, Україна); А.М. Глібовець (Київ, Україна); О. Горбуновс (Рига, Латвія); В.Ф. Губарев (Київ, Україна); Л.Ф. Гуляницький (Київ, Україна); А.М. Гупал (Київ, Україна); В.В. Зосімов (Одеса, Україна); П.І. Когут (Дніпро, Україна); Л.М. Козак (Київ, Україна); С.Л. Кривий (Київ, Україна); В.І. Литвиненко (Херсон, Україна); Р. Мартінес Бежар (Мурсія, Іспанія); О.В. Палагін (Київ, Україна); С.Л. Погорілий (Київ, Україна); Н. Прокоф'єва (Рига, Латвія); Б. Савчинський (Гейдельберг, Німеччина); А.-Б. М. Салем (Каїр, Єгипет); К.М. Синиця (Київ, Україна); В.С. Степашко (Київ, Україна); І.В. Суровцев (Київ, Україна); Л.С. Файнзільберг (Київ, Україна); А.О. Чикрій (Київ, Україна); М.І. Шлезінгер (Київ, Україна)

Відповідальний виконавець: Г.О. Пезенцалі

Редактори: Н.А. Чарчян, А.Ю. Вітченко, О.О. Лисенко

Комп'ютерна група: О.В. Тупальський, Н.С. Сташкова

Ідентифікатор медіа: R30-05899

Адреса: Інститут інформаційних технологій та систем НАН України, м. Київ,
Просп. Акад. Глушкова, 40, 03187
Телефон: 526-00-09, e-mail: its.journal.ua@gmail.com,
Сайт: <https://nasu-periodicals.org.ua/index.php/its/>

Підп. до друку 22.09.2025 р. Формат 70 × 108/16. Гарн. Book Antiqua.
Ум. друк. арк. 8,75. Обл.-вид. арк. 8,88. Тираж 83 пр. Зам. № 7794

Видавець і виготовлювач ВД «Академперіодика» НАН України
01024, Київ, вул. Терещенківська, 4

Свідоцтво про внесення до Державного реєстру суб'єктів
видавничої справи серії ДК № 544 від 27.07.2001



NATIONAL ACADEMY OF SCIENCES OF UKRAINE
INSTITUTE OF INFORMATION TECHNOLOGIES AND SYSTEMS
OF THE NAS OF UKRAINE

INFORMATION TECHNOLOGIES & SYSTEMS

3 (3)
2025

ACADEMIC AND RESEARCH JOURNAL
FOUNDED IN JANUARY 2025
PUBLISHED 6 TIMES PER YEAR
KYIV

CONTENTS

Theory of Information Technologies and Systems Construction

Rytsar B.Ye. New Method for Generating Test Codes to Detect Multiple Stuck-at-Faults in Combinational Circuits. Part 2. 3

Intellectual Information Technologies

Oursatyev O.A., Volkov O.Ye. Approaches to Creating Multiagent Systems and Deep Reinforced Learning of Drones 30

Computer Vision and Pattern Recognition

Golovin O.M., Sapunova N.O. Evolution of Video Surveillance Systems: from Analog Cameras to Intelligent Video Analytics Systems Based on Edge Computing 56

Digital Medicine

Petrenko M.G., Malakhov K.S. Methodology and Practice of Interdisciplinary Knowledge Convergence in Digital Health 76

Author Guidelines 97

ЗМІСТ

Теорія побудови інформаційних технологій та системи

Rytsar B. Ye. New Method for Generating Test Codes to Detect Multiple Stuck-at-faults in Combinational Circuits. Part 2 3

Інтелектуальні інформаційні технології

Урсатсьєв О.А., Волков О.Є. Підходи до створення мультиагентних систем і глибокого посиленого навчання дронів 30

Комп'ютерний зір та розпізнавання образів

Головін О.М., Сапунова Н.О. Еволюція систем відеоспостереження: від аналогових камер до інтелектуальних систем відеоаналітики на основі граничних обчислень 56

Цифрова медицина

Петренко М.Г., Малахов К.С. Методологія і практика конвергенції міждисциплінарних знань у сфері цифрової охорони здоров'я 76

Керівництво для авторів 97

THEORY OF INFORMATION TECHNOLOGIES AND SYSTEMS CONSTRUCTION

ТЕОРІЯ ПОБУДОВИ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА СИСТЕМ

<https://doi.org/10.15407/intechsys.2025.03.003>
UDC 519.7

B.Ye. RYTSAR, DSc (Engineering), Professor,
Department of Radioelectronic Technologies of Information Systems,
Institute of Information and Communication Technologies and Electronic Engineering,
L'viv Polytechnic National University
12, Stepan Bandery str., L'viv, 79013, Ukraine
<http://orcid.org/0000-0002-2929-2954>
bohdanrytsar@gmail.com

A NEW METHOD FOR GENERATING TEST CODES TO DETECT MULTIPLE STUCK-AT-FAULTS IN COMBINATIONAL CIRCUITS. Part 2

The article is devoted to a new method of generating test codes to detect multiple stuck-at-faults (0/1) type damages in digital combinational PIPO-circuits, which is based on the artificial introduction of non-essential variables and the application of the procedure of q -partition of the system minterms of a given system of functions. Due to the use of the numerical set-theoretic approach, the proposed method differs from the known ones in a relatively simpler practical implementation to detect stuck-at-faults (0/1) type damages both at one point and simultaneously at several points simultaneously of the studied circuit.

Keywords: combinational PIPO-circuit, single and multiple stuck-at-faults (0/1) type damages, procedure of q -partition of system minterms, non-essential variables, vector of test codes.

Introduction

The method of generating test codes for detecting multiple stuck-at-faults (0/1) in combinational circuits, proposed in the previous article (Part 1), is of particularly practical importance for digital microelectronics. In digital devices of the PIPO type that are described by the system of functions $F(X) = \{f_1(X), f_2(X), \dots, f_s(X)\}$, $X = \{x_1, x_2, \dots, x_n\}$, and compared to devices of the PISO type, the interconnectedness of system functions makes the detection of multiple stuck-at faults (0/1) considerably more difficult

Cite: Rytsar B.Ye. Automated Sentence Alignment in Ukrainian-German Parallel Texts. *Information Technologies and Systems*, Kyiv, 2025, Vol. 3 (3), 03–29. <https://doi.org/10.15407/intechsys.2025.03.003>

© Publisher PH "Akadempriodyka" of the NAS of Ukraine, 2025. The article is published under an open access license CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

than detecting a single one. Multiple damage of the stuck-at-faults (0/1) type in the studied circuit manifests itself as the simultaneous appearance of an arbitrary combination of faults s-a-0 and s-a-1 at any of its points, and therefore, the detected damage in any one function can simultaneously occur in other functions of the system. Therefore, the usage of the search for such damage in logic networks that are based on single fault methods and the determination of the code test vectors does not provide a reliable result for the diagnosis of digital devices of the PIPO type.

A number of analytical and heuristic methods (path sensitizing, Equivalent-Normal-Form (ENF) method, Karnaugh map and tabular method, the ENF K-map method, the Boolean difference method, the SPOOF and ILP method and Genetic Algorithm method) that have appeared in the literature over the past decades is presented in [3]. There are also methods for diagnosing multiple stuck-at-faults in combinational circuits that work on the basis of modeling single faults with the procedure of repeated addition and elimination of faults [4–6] and on the analytical approach basis [7–14]. However, these methods have restrictive assumptions about the final result and they are quite complex and time-consuming in terms of finding test code vectors. Moreover, the proposed structural approaches additionally complicate the correction of design errors and impose certain practical limitations in the process of detecting multiple damage in combinational PIPO-circuits.

The paper considers a method for generating test code vectors to determine the location and of stuck-at-faults (0/1) type of single and multiple faults in combinational PIPO-circuits. It is based on the artificial introduction of one or more non-essential variables into the studied circuit and the application of the well-known procedure of q -partition [15, 16] of the system minterms of a given system of Boolean functions.

Theoretical Part

To determine the location and type of multiple stuck-at-faults (0/1) in combinational circuits described by a system of Boolean functions [2], the generation of test code vectors is performed using the same algorithm as for a single function f (Part 1), but taking into account that the objects of operations and procedures will now be system minterms $m_{s_1}, m_{s_2}, \dots, m_{s_k}$ of the set $Y_{1,2,\dots,s}^1 = \{m_{s_1}, m_{s_2}, \dots, m_{s_k}\}^1$ with indices $s_i \in \{1, 2, \dots, s\}$ of functions of the given system $F(X)$, the total number of which is equal to $s \times 2^n$.

The q -partition procedure (operator \Rightarrow) is performed over the system minterms $m_{s_1}, m_{s_2}, \dots, m_{s_k}$ of a given system $F(X)$ for an arbitrary literal mask $\{l_{\alpha_1} l_{\alpha_2} \dots l_{\alpha_{n-q}} | l_{\beta_1} l_{\beta_2} \dots l_{\beta_q}\}$, where $l_i \in \{\bar{x}_i, x_i\}$ and the dash $|$ is the q -partition symbol. As a result, a set of partitioned system minterms is formed, which consist of system subminterms of the $(n - q)$ -class and q -class. As noted in Part 1, binary subminterms of the $(n - q)$ -class are chosen to introduce “non-essential” variables. Therefore, the simplifica-

tion of the set of partitioned system minterms is performed first by the procedure of “right union” (operator $\overset{\cup}{\Rightarrow}$) and then (if necessary) by the procedure of “left union” (operator $\overset{\cup}{\Leftarrow}$). Then, “non-essential” variables will be introduced for any partition mask provided that the values of the system subminterms of the $(n-q)$ -class form the complete set \mathbf{E}'_2 , where $r = 1, 2, \dots, (n-1)$ is the number of introduced “non-essential” variables. Therefore, to introduce one ($r = 1$) “non-essential” variable x_i into the system $F(X)$, it is necessary to impose a literal mask $\{l_{\alpha_i} | l_{\beta_1} l_{\beta_2} \dots l_{\beta_{n-1}}\}$ on the set of system minterms $m_{s_1}, m_{s_2}, \dots, m_{s_k}$ and to obtain (after the simplification procedure) two subsets ($\mathbf{E}'_2 = 2$) of partitioned system minterms $\{0_{\alpha_i} | l_{\beta_1} l_{\beta_2} \dots l_{\beta_{n-1}}\}$ and $\{1_{\alpha_i} | l_{\beta_1} l_{\beta_2} \dots l_{\beta_{n-1}}\}$. The latter, similarly as in [2], will form pseudo-perfect STFs $Y_{s_i(x_i/0)}^1$ and $Y_{s_i(x_i/1)}^1$ after introducing into the values of their subminterms of the $(n-q)$ -class (here 1-class) $\log 1(x_i/0)$, forming $\{(0_{\alpha_i}, 1) | l_{\beta_1} l_{\beta_2} \dots l_{\beta_{n-1}}\}$, and $\log 0(x_i/1)$, forming $\{(0, 1_{\alpha_i}) | l_{\beta_1} l_{\beta_2} \dots l_{\beta_{n-1}}\}$, and applying the concatenation procedure (operator $\overset{con}{\Rightarrow}$). The elements of the pseudo-perfect STFs $Y_{s_i(x_i/0)}^1$ and $Y_{s_i(x_i/1)}^1$ of the “damaged” system $F(X)$ are then combined with the system minterms of the perfect STF Y_s^1 of the given system $F(X)$ in a polynomial format, where a simplification procedure is performed by eliminating pairs of elements that have the same values and the same function indices, as well as by eliminating those elements of the perfect STF Y_s^1 that have a pair with the same value but different indices. After that, the test code vectors in the case of single stuck-at-faults (0/1) type damages will be formed in the form (4) and (5) (Part 1).

In the case of introducing of two ($r = 2$) “non-essential” variables x_i and x_j , that is necessary to determine the desired vectors for multiple damage at two points of the circuit, the algorithm is implemented in a similar way, starting with the imposition of a literal mask $\{l_{\alpha_i} l_{\beta_j} | l_{\beta_1} l_{\beta_2} \dots l_{\beta_{n-2}}\}$ on the set of system minterms of the given system $F(X)$. At the same time, in order to form four pseudo-perfect STFs $Y_{s_i(x_i, x_j/00)}^1, Y_{s_i(x_i, x_j/01)}^1, Y_{s_i(x_i, x_j/10)}^1, Y_{s_i(x_i, x_j/11)}^1$, it is necessary to introduce certain logical values from the set $\{00, 01, 10, 11\}$ into the values of their subminterms of the $(n-q)$ -class (here of 2-class) so that the complete set $\mathbf{E}'_2 = 4$ is formed there and so on.

We also note, that similarly to the previous article (Part 1), when $|Y_s^1| > |Y_s^0|$, it is appropriate to execute all procedures and operations taking into account the set Y^0 , as well as that the operations of combining and simplifying the elements of the pseudo-perfect STF $Y_{x_i/\sim}^0$ and the elements of the perfect STF Y_s^0 will now be performed in the inverse polynomial format; in particular, in the case of one “non-essential” variable x_i this is the set $\left\{ \begin{matrix} Y_{x_i/\sim}^0 \\ Y_s^0 \end{matrix} \right\}^{\oplus}$. In order to determine the vectors of test codes of the form (4) and (5) in this simplified set (Part 1), it is necessary to

perform certain operations on the elements of these sets, taking into account the priority of the elements of the pseudo-perfect STF $Y_{x_i/\sim}^0$ over the elements of the perfect STF Y_s^0 . To implement the proposed approach, the initial requirement is to cancel first all the function indices in the elements of the set Y_s^0 , that is, to form a set of elements without indices Y^{0*} . Next, it is necessary to perform procedural transformations

on the elements of the sets $\begin{cases} Y_{x_i/\sim}^0 \\ Y^{0*} \end{cases}$ into vectors of test codes of the form (4)

and (5) $Y_{x_i/\sim}^0 \rightarrow C_{x_i/\sim}^1$ and $Y^{0*} \rightarrow C_{x_i/\sim}^0$, that is, to perform transformations

$\begin{cases} Y_{x_i/\sim}^0 \\ Y^{0*} \end{cases} \xRightarrow{Y \rightarrow C} \begin{cases} C_{x_i/\sim}^1 \\ C_{x_i/\sim}^0 \end{cases}$ (where $\xRightarrow{Y \rightarrow C}$ is the transformation operator), according to

the following rule:

1) move the elements of the set without indices Y^{0*} into the set $C_{x_i/\sim}^1$, writing down all the indices s of the given system $F(X)$;

2) move the elements of the set $Y_{x_i/\sim}^0$ that have all the indices s of the given system $F(X)$ to the set $C_{x_i/\sim}^0$ by canceling all their indices, and move the elements of the set $Y_{x_i/\sim}^0$ with indices $s' < s$ to the set $C_{x_i/\sim}^1$ with indices $s - s'$.

Thus, the procedure for converting (operator $\xRightarrow{Y \rightarrow C}$) a set from the inverse polynomial format into a set of vectors of test codes of the form (4) and (5) (see Part 1) looks like this (using the example of one “non-essential” variable x_i):

$$\begin{cases} Y_{x_i/0}^0 \\ Y^0 \end{cases} \xRightarrow{\oplus} \begin{cases} Y_{x_i/0}^0 \\ Y^{0*} \end{cases} \xRightarrow{Y \rightarrow C} \begin{cases} C_{x_i/0}^1 \\ C_{x_i/0}^0 \end{cases}, \quad (1)$$

$$\begin{cases} Y_{x_i/1}^0 \\ Y^0 \end{cases} \xRightarrow{\oplus} \begin{cases} Y_{x_i/1}^0 \\ Y^{0*} \end{cases} \xRightarrow{Y \rightarrow C} \begin{cases} C_{x_i/1}^1 \\ C_{x_i/1}^0 \end{cases}. \quad (2)$$

The transformation procedures (1) and (2) are illustrated by an example:

$$\Rightarrow \begin{cases} 100_{1,2}, 101_1 \\ 000_{1,2}, 001_1, 110_2 \end{cases} \xRightarrow{\oplus} \begin{cases} \{100_{1,2}, 101_1\}^0 \\ \{000, 001, 110\}^0 \end{cases} \xRightarrow{Y \rightarrow C} \begin{cases} \{000_{1,2}, 001_{1,2}, 101_2, 110_{1,2}\}^1 \\ \{100\}^0 \end{cases}.$$

It is obvious that procedures (1) and (2) will be performed in the same way for a larger number of “non-essential” variables, taking into account the elements of the sets of indices of the functions of the given system.

Practical Part

Let us consider the proposed method of generating and determining test code vectors based on the introduction of one or more “non-essential”

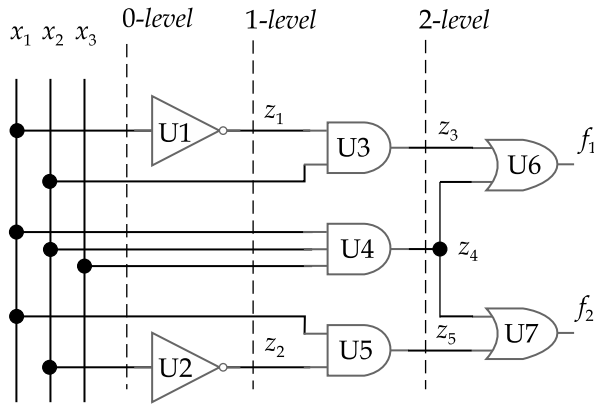


Fig. 1

variables for detecting multiple stuck-at-fault (0/1) type faults on examples of circuits described by a system of functions.

Example 1. Determine test code vectors for detecting single and multiple stuck-at-fault (0/1) type faults at different points on the 0-, 1- and 2-levels of the circuit shown in Fig. 1 (the circuit is borrowed from [2, 17, p. 87]).

Solution. The scheme in Fig. 1 is described by a system of functions

$$\begin{cases} f_1(x_1, x_2, x_3) = \bar{x}_1 x_2 \vee x_1 x_2 x_3 \\ f_2(x_1, x_2, x_3) = x_1 \bar{x}_2 \vee x_1 x_2 x_3 \end{cases}, \text{ to which corresponds a system of perfect}$$

$$\text{STFs } \{Y_1^1, Y_2^1\} \begin{cases} Y_1^1 = \{(01-), (111)\}^1 \equiv \{2, 3, 7\}^1 \\ Y_2^1 = \{(10-), (111)\}^1 \equiv \{4, 5, 7\}^1 \end{cases} \text{ and a set of system minterms}$$

$$Y_{1,2}^1 = \{010_1, 011_1, 100_2, 101_2, 111_{1,2}\}^1.$$

To detect stuck-at-fault (0/1) type “damages” at individual points of the circuit, at the 0-level of the circuit, we perform a 2-partition of the system minterms of the set $Y_{1,2}^1$:

$$\begin{aligned} Y_{1,2}^1 &= \{010_1, 011_1, 100_2, 101_2, 111_{1,2}\}^1 \xRightarrow{p^2} \\ &\Rightarrow \begin{cases} \{l_1 | l_2 l_3\} = \{(0 | 10, 11)_1, (1 | 00_2, 01_2, 11_{1,2})\}^1 \\ \{l_2 | l_1 l_3\} = \{(0 | 10, 11)_2, (1 | 00_1, 01_1, 11_{1,2})\}^1 \\ \{l_3 | l_1 l_2\} = \{(0 | 01_1, 10_2), (1 | 01_1, 10_2, 11_{1,2})\}^1 \end{cases} \end{aligned}$$

From here we define the vectors of test codes at the 0-level of the scheme, introducing into the studied scheme the “non-essential” variables x_1, x_2 and x_3 . As mentioned above, after the procedure of concatenation of the partitioned system minterms, the next procedure is the simplification in the polynomial format of the combined sets – pseudo-perfect STF $Y_{1,2(x_i/\sim)}^1$ and perfect STF $Y_{1,2}^1$. It is performed by eliminating pairs of

elements with the same values and indices, as well as elements of perfect STF $Y_{1,2}^1$ having the same values but different indices, compared to the elements of the set $Y_{1,2(x_i/\sim)}^1$ (we will denote them by a dash with the opposite slope). Therefore, the required vectors for the “non-essential” variable x_1 are obtained as follows:

$$Y_{1,2(x_1/0)}^1 = \{((\mathbf{0}, \mathbf{1}) | 10, 11)_1\}^1 \stackrel{con}{\Rightarrow} \{(010, 011, 110, 111)_1\}^1 \Rightarrow \\ \Rightarrow \left\{ \begin{array}{l} \overline{010}_1, \overline{011}_1, 110_1, 111_1 \\ \overline{010}_1, \overline{011}_1, 100_2, 101_2, \overline{111}_{1,2} \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{110_1, 111_1\}^1 \\ \{100_2, 101_2\}^{0'} \end{array} \right.$$

$$Y_{1,2(x_1/1)}^1 = \{(0, \mathbf{1}) | 00_2, 01_2, 11, 1_2\}^1 \stackrel{con}{\Rightarrow} \{000_2, 001_2, 011, 1_2, 100_2, 101_2, 111, 1_2\}^1 \Rightarrow \\ \Rightarrow \left\{ \begin{array}{l} 000_2, 001_2, 011, 1_2, \overline{100}_2, \overline{101}_2, \overline{111}_{1,2} \\ 010_1, \overline{011}_1, \overline{100}_2, \overline{101}_2, \overline{111}_{1,2} \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{000_2, 001_2, 011, 1_2\}^1; \\ \{010_1\}^0 \end{array} \right.;$$

for “non-essential” variables x_2 and x_3 we get the same vectors as in ([2] Table 7):

$$Y_{1,2(x_2/0)}^1 = \left\{ \begin{array}{l} \{110_2, 111_2\}^1 \\ \{010_1, 011_1\}^0 \end{array} \right. , \quad Y_{1,2(x_2/1)}^1 = \left\{ \begin{array}{l} \{000_1, 001_1, 101, 1_2\}^1 \\ \{100_2\}^0 \end{array} \right. ; \\ Y_{1,2(x_3/0)}^1 = \left\{ \begin{array}{l} \{\emptyset\}^1 \\ \{111, 1_2\}^0 \end{array} \right. , \quad Y_{1,2(x_3/1)}^1 = \left\{ \begin{array}{l} \{110, 1_2\}^1 \\ \{\emptyset\}^0 \end{array} \right. .$$

To determine the vectors of test codes with simultaneous “damage” at two points of 0-level of the scheme, we apply the 1-partition of the system minterms of the set $Y_{1,2}^1$:

$$Y_{1,2}^1 = \{010_1, 011_1, 100_2, 101_2, 111, 1_2\}^1 \stackrel{p^1}{\Rightarrow} \\ \stackrel{p^1}{\Rightarrow} \left\{ \begin{array}{l} \{l_1 l_2 | l_3\} = \{(00 | \emptyset), (01 | 0, 1)_1, (10 | 0, 1)_2, (11 | 1)_1, 1_2\}^1 \\ \{l_1 l_3 | l_2\} = \{(00, 01 | 1)_1, (10 | 0)_2, (11 | 0_2, 1, 1_2)\}^1 \\ \{l_2 l_3 | l_1\} = \{(00, 01 | 1)_2, (10 | 0)_1, (11 | 0_1, 1, 1_2)\}^1 \end{array} \right. ;$$

$$Y_{1,2(x_1 x_2/00)}^1 = \{(\mathbf{00}, \mathbf{01}, 10, 11) | \emptyset\}^1 \stackrel{con}{\Rightarrow} \{\emptyset\}^1 \Rightarrow \\ \Rightarrow \left\{ \begin{array}{l} \emptyset \\ 010_1, 011_1, 100_2, 101_2, 111, 1_2 \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{\emptyset\}^1 \\ \{010_1, 011_1, 100_2, 101_2, 111, 1_2\}^{0'} \end{array} \right.$$

$$Y_{1,2(x_1 x_2/01)}^1 = \{(00, \mathbf{01}, 10, 11) | 0, 1\}^1 \stackrel{con}{\Rightarrow} \\ \stackrel{con}{\Rightarrow} \{(000, 001, 010, 011, 100, 101, 110, 111)_1\}^1 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} 000_1, 001_1, \cancel{010_1}, \cancel{011_1}, 100_1, 101_1, 110_1, 111_1 \\ \cancel{010_1}, \cancel{011_1}, 100_2, 101_2, 111_{1,2} \end{array} \right\}^{\oplus} \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} \{000_1, 001_1, 100_1, 101_1, 110_1, 111_1\}^1 \\ \{\emptyset\}^0 \end{array} \right. ,$$

$$Y_{1,2(x_1x_2/10)}^1 = \{(00, 01, \mathbf{10}, 11) | 0, 1\}_2^{\text{con}} \Rightarrow$$

$$\Rightarrow \{(000, 001, 010, 011, 100, 101, 110, 111)_2\}^1 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} 000_2, 001_2, 010_2, 011_2, \cancel{100_2}, \cancel{101_2}, 110_2, 111_2 \\ 010_1, 011_1, \cancel{100_2}, \cancel{101_2}, 111_{1,2} \end{array} \right\}^{\oplus} \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} \{000_2, 001_2, 010_2, 011_2, 110_2, 111_2\}^1 \\ \{\emptyset\}^0 \end{array} \right. ,$$

$$Y_{1,2(x_1x_2/11)}^1 = \{(00, 01, 10, \mathbf{11}) | 1\}_{1,2}^{\text{con}} \Rightarrow \{(001, 011, 101, 111)_{1,2}\}^1 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} 001_{1,2}, 011_{1,2}, 101_{1,2}, \cancel{111_{1,2}} \\ 010_1, \cancel{011_1}, 100_2, \cancel{101_2}, \cancel{111_{1,2}} \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{001_{1,2}, 011_{1,2}, 101_{1,2}\}^1 \\ \{010_1, 100_2\}^0 \end{array} \right. ;$$

$$Y_{1,2(x_1x_3/00,01)}^1 = \{((\mathbf{00}, 01, 10, 11) | 1)_1\}^{\text{con}} \Rightarrow \{(010, 011, 110, 111)_1\}^1 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} \cancel{010_1}, \cancel{011_1}, 110_1, 111_1 \\ \cancel{010_1}, \cancel{011_1}, 100_2, 101_2, \cancel{111_{1,2}} \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{110_1, 111_1\}^1 \\ \{100_2, 101_2\}^0 \end{array} \right. ' ,$$

$$Y_{1,2(x_1x_3/10)}^1 = \{(00, 01, \mathbf{10}, 11) | 0\}_2^{\text{con}} \Rightarrow \{(000, 001, 100, 101)_2\}^1 \Rightarrow$$

$$\left\{ \begin{array}{l} 000_2, 001_2, \cancel{100_2}, \cancel{101_2} \\ 010_1, 011_1, \cancel{100_2}, \cancel{101_2}, 111_{1,2} \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{000_2, 001_2\}^1 \\ \{010_1, 011_1, 111_{1,2}\}^0 \end{array} \right. ' ,$$

$$Y_{x_1x_3/11}^1 = \{(00, 01, 10, \mathbf{11}) | 0_2, 1_{1,2}\}^{\text{con}} \Rightarrow$$

$$\{000_2, 001_2, 010_{1,2}, 011_{1,2}, 100_2, 101_2, 110_{1,2}, 111_{1,2}\}^1 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} 000_2, 001_2, 010_{1,2}, 011_{1,2}, \cancel{100_2}, \cancel{101_2}, 110_{1,2}, \cancel{111_{1,2}} \\ 010_1, 011_1, \cancel{100_2}, \cancel{101_2}, 111_{1,2} \end{array} \right\}^{\oplus} \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} \{000_2, 001_2, 010_{1,2}, 011_{1,2}, 110_{1,2}\}^1 \\ \{\emptyset\}^0 \end{array} \right. ;$$

$$Y_{1,2(x_2x_3/00,01)}^1 = \{((\mathbf{00}, 01, 10, 11) | 1)_2\}^{\text{con}} \Rightarrow \{(100, 101, 110, 111)_2\}^1 \Rightarrow$$

$$\left\{ \begin{array}{l} \cancel{100_2}, \cancel{101_2}, 110_2, 111_2 \\ 010_1, 011_1, \cancel{100_2}, \cancel{101_2}, \cancel{111_{1,2}} \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{110_2, 111_2\}^1 \\ \{010_1, 011_1\}^0 \end{array} \right. ' ,$$

$$\begin{aligned}
 Y_{1,2(x_2x_3/10)}^1 &= \{(00,01,10,11) | 0\}_1^1 \xRightarrow{con} \{(000,001,010,011)_1\}^1 \Rightarrow \\
 &\left\{ \begin{array}{l} 000_1, 001_1, \cancel{010_1}, \cancel{011_1} \\ \cancel{010_1}, \cancel{011_1}, 100_2, 101_2, 111_{1,2} \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{000_1, 001_1\}^1 \\ \{100_2, 101_2, 111_{1,2}\}^0 \end{array} \right\}' \\
 Y_{1,2(x_2x_3/11)}^1 &= \{(00,01,10,11) | 0, 1, 1, 2\}^1 \xRightarrow{con} \\
 &\xRightarrow{con} \{000_1, 001_1, 010_1, 011_1, 100_{1,2}, 101_{1,2}, 110_{1,2}, 111_{1,2}\}^1 \Rightarrow \\
 &\Rightarrow \left\{ \begin{array}{l} 000_1, 001_1, \cancel{010_1}, \cancel{011_1}, 100_{1,2}, 101_{1,2}, 110_{1,2}, \cancel{111_{1,2}} \\ \cancel{010_1}, \cancel{011_1}, \cancel{100_2}, \cancel{101_2}, \cancel{111_{1,2}} \end{array} \right\}^{\oplus} \Rightarrow \\
 &\Rightarrow \left\{ \begin{array}{l} \{000_1, 001_1, 100_{1,2}, 101_{1,2}, 110_{1,2}\}^1 \\ \{\emptyset\}^0 \end{array} \right\}.
 \end{aligned}$$

Table 1 contains test code vectors for detecting possible stuck-at-fault (0/1) type damages at two points of the 0-level of the circuit.

Table 1

Stuck-at-fault in $f_{1,2}$	$x_1 x_2 / \sim \sim$	$x_1 x_3 / \sim \sim$	$x_2 x_3 / \sim \sim$
s-a-00	$\begin{pmatrix} 010_1 \\ 011_1 \\ 100_2 \\ 101_2 \\ 111_{1,2} \end{pmatrix}^0$	$\begin{pmatrix} 110_1 \\ 111_1 \end{pmatrix}^1, \begin{pmatrix} 100_2 \\ 101_2 \end{pmatrix}^0$	$\begin{pmatrix} 110_2 \\ 111_2 \end{pmatrix}^1, \begin{pmatrix} 010_1 \\ 011_1 \end{pmatrix}^0$
s-a-01	$\begin{pmatrix} 000_1 \\ 001_1 \\ 100_1 \\ 101_1 \\ 110_1 \\ 111_1 \end{pmatrix}^1$	$\begin{pmatrix} 110_1 \\ 111_1 \end{pmatrix}^1, \begin{pmatrix} 100_2 \\ 101_2 \end{pmatrix}^0$	$\begin{pmatrix} 110_2 \\ 111_2 \end{pmatrix}^1, \begin{pmatrix} 010_1 \\ 011_1 \end{pmatrix}^0$
s-a-10	$\begin{pmatrix} 000_2 \\ 001_2 \\ 010_2 \\ 011_2 \\ 110_2 \\ 111_2 \end{pmatrix}^1$	$\begin{pmatrix} 000_2 \\ 001_2 \end{pmatrix}^1, \begin{pmatrix} 010_1 \\ 011_1 \\ 111_{1,2} \end{pmatrix}^0$	$\begin{pmatrix} 000_1 \\ 001_1 \end{pmatrix}^1, \begin{pmatrix} 100_2 \\ 101_2 \\ 111_{1,2} \end{pmatrix}^0$
s-a-11	$\begin{pmatrix} 001_{1,2} \\ 011_{1,2} \\ 101_{1,2} \end{pmatrix}^1, \begin{pmatrix} 010_1 \\ 100_2 \end{pmatrix}^0$	$\begin{pmatrix} 000_2 \\ 001_2 \\ 010_{1,2} \\ 011_{1,2} \\ 110_{1,2} \end{pmatrix}^1$	$\begin{pmatrix} 000_1 \\ 001_1 \\ 100_{1,2} \\ 101_{1,2} \\ 110_{1,2} \end{pmatrix}^1$

Similarly, we define the vectors of test codes on the 1-level of the scheme (see Fig. 1) for the system of functions $\begin{cases} f_1(z_1, z_2, x_3) = z_1 \bar{z}_2 \vee \bar{z}_1 \bar{z}_2 x_3 \\ f_2(z_1, z_2, x_3) = \bar{z}_1 z_2 \vee \bar{z}_1 \bar{z}_2 x_3 \end{cases}'$

where $z_1 = \bar{x}_1$, and $z_2 = \bar{x}_2$, to which corresponds the system of perfect STFs $\{Y_1^1, Y_2^1\} \begin{cases} Y_1^1 = \{(10-), (001)\}^1 \\ Y_2^1 = \{(01-), (001)\}^1 \end{cases}'$ and hence $Y_{1,2}^1 = \{001_{1,2}, 010_2, 011_2, 100_1, 101_1\}^1$ or $Y_{1,2}^0 = \{000_{1,2}, 010_1, 011_1, 100_2, 101_2, 110_{1,2}, 111_{1,2}\}^0$.

As a result of the 2-partition of the system minterms of the set $Y_{1,2}^1$, we obtain the following sets:

$$\begin{aligned} Y_{1,2}^1 &= \{001_{1,2}, 010_2, 011_2, 100_1, 101_1\}^1 \xrightarrow{p^2} \\ &\Rightarrow \begin{cases} \{l_1 | l_2 l_3\} = \{(0 | 01_{1,2}, 10_2, 11_2), (1 | 00_1, 01_1)\}^1 \\ \{l_2 | l_1 l_3\} = \{(0 | 01_{1,2}, 10_1, 11_1), (1 | 00_2, 01_2)\}^1 \\ \{l_3 | l_1 l_2\} = \{(0 | 01_2, 10_1), (1 | 00_{1,2}, 01_2, 10_1)\}^1 \end{cases} \end{aligned}$$

Let's define the vectors of test codes for the "non-essential" variable z_1 :

$$\begin{aligned} Y_{1,2(z_1/0)}^1 &= \{(0, 1) | 01_{1,2}, 10_2, 11_2\}^1 \xrightarrow{con} \{001_{1,2}, 010_2, 011_2, 101_{1,2}, 110_2, 111_2\}^1 \Rightarrow \\ &\Rightarrow \left\{ \begin{array}{l} \{001_{1,2}, 010_2, 011_2, 101_{1,2}, 110_2, 111_2\}^\oplus \\ \{001_{1,2}, 010_2, 011_2, 100_1, 101_1\}^0 \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \{101_{1,2}, 110_2, 111_2\}^1 \\ \{100_1\}^0 \end{array} \right\}, \\ Y_{1,2(z_1/1)}^1 &= \{(0, 1) | 00_1, 01_1\}^1 \xrightarrow{con} \{000_1, 001_1, 100_1, 101_1\}^1 \Rightarrow \\ &\Rightarrow \left\{ \begin{array}{l} \{000_1, 001_1, 100_1, 101_1\}^\oplus \\ \{001_{1,2}, 010_2, 011_2, 100_1, 101_1\}^0 \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \{000_1, 001_1\}^1 \\ \{010_2, 011_2\}^0 \end{array} \right\}. \end{aligned}$$

The test code vectors for z_1 , as well as for the remaining "non-essential" variables z_2 and x_3 , are the same as in (in [2] Table 9), namely:

$$\begin{aligned} Y_{1,2(z_2/0)}^1 &= \left\{ \begin{array}{l} \{011_{1,2}, 110_1, 111_1\}^1 \\ \{010_2\}^0 \end{array} \right\}, \quad Y_{1,2(z_2/1)}^1 = \left\{ \begin{array}{l} \{000_2, 001_2\}^1 \\ \{100_2, 101_2\}^0 \end{array} \right\}; \\ Y_{1,2(x_3/0)}^1 &= \left\{ \begin{array}{l} \{\emptyset\}^1 \\ \{001_{1,2}\}^0 \end{array} \right\}, \quad Y_{1,2(x_3/1)}^1 = \left\{ \begin{array}{l} \{000_{1,2}\}^1 \\ \{\emptyset\}^0 \end{array} \right\}. \end{aligned}$$

In the case of simultaneous "damage" of the stuck-at-fault type (0/1) at two points of the 1-level scheme of test code vectors, we retain after 1-split the system minterms of the set:

$$\begin{aligned}
 Y_{1,2}^1 &= \{001_{1,2}, 010_2, 011_2, 100_1, 101_1\}^{P^1} \Rightarrow \\
 &\left\{ \begin{aligned} \{l_1 l_2 | l_3\} &= \{(00 | 1_{1,2}), (01 | 0_2, 1_2), (10 | 0_1, 1_1), (11 | \emptyset)\}^1 \\ \{l_1 l_3 | l_2\} &= \{(00 | 1_2), (01 | 0_{1,2}, 1_2), (10, 11 | 0_1)\}^1 \\ \{l_2 l_3 | l_1\} &= \{(00 | 1_1), (01 | 0_{1,2}, 1_1), (10, 11 | 0_2)\}^1 \end{aligned} \right. ; \\
 Y_{1,2(z_1 z_2 / 00)}^1 &= \{(00, 01, 10, 11) | 1_{1,2}\}^1 \xRightarrow{con} \{001_{1,2}, 011_{1,2}, 101_{1,2}, 111_{1,2}\}^1 \Rightarrow \\
 &\left\{ \begin{aligned} \{001_{1,2}, 011_{1,2}, 101_{1,2}, 111_{1,2}\}^\oplus \\ \{001_{1,2}, 010_2, 011_2, 100_1, 101_1\} \end{aligned} \right\} \Rightarrow \left\{ \begin{aligned} \{011_{1,2}, 101_{1,2}, 111_{1,2}\}^1 \\ \{010_2, 100_1\}^0 \end{aligned} \right. , \\
 Y_{1,2(z_1 z_2 / 01)}^1 &= \{(00, 01, 10, 11) | 0_2, 1_2\}^1 \xRightarrow{con} \\
 &\xRightarrow{con} \{000_2, 001_2, 010_2, 011_2, 100_2, 101_2, 110_2, 111_2\}^1 \Rightarrow \\
 &\Rightarrow \left\{ \begin{aligned} \{000_2, 001_2, 010_2, 011_2, 100_2, 101_2, 110_2, 111_2\}^\oplus \\ \{001_{1,2}, 010_2, 011_2, 100_1, 101_1\} \end{aligned} \right\} \Rightarrow \\
 &\Rightarrow \left\{ \begin{aligned} \{000_2, 001_2, 100_2, 101_2, 110_2, 111_2\}^1 \\ \{\emptyset\}^0 \end{aligned} \right. , \\
 Y_{1,2(z_1 z_2 / 10)}^1 &= \{(00, 01, 10, 11) | 0_1, 1_1\}^1 \xRightarrow{con} \\
 &\xRightarrow{con} \{000_1, 001_1, 010_1, 011_1, 100_1, 101_1, 110_1, 111_1\}^1 \Rightarrow \\
 &\Rightarrow \left\{ \begin{aligned} \{000_1, 001_1, 010_1, 011_1, 100_1, 101_1, 110_1, 111_1\}^\oplus \\ \{001_{1,2}, 010_2, 011_2, 100_1, 101_1\} \end{aligned} \right\} \Rightarrow \\
 &\Rightarrow \left\{ \begin{aligned} \{000_1, 001_1, 010_1, 011_1, 110_1, 111_1\}^1 \\ \{\emptyset\}^0 \end{aligned} \right. , \\
 Y_{1,2(z_1 z_2 / 11)}^1 &= \{(00, 01, 10, 11) | \emptyset\}^1 \xRightarrow{con} \{\emptyset\}^1 \Rightarrow \\
 &\Rightarrow \left\{ \begin{aligned} \{\emptyset\} \\ \{001_{1,2}, 010_2, 011_2, 100_1, 101_1\} \end{aligned} \right\}^\oplus \Rightarrow \left\{ \begin{aligned} \{\emptyset\}^1 \\ \{001_{1,2}, 010_2, 011_2, 100_1, 101_1\}^0 \end{aligned} \right. ; \\
 Y_{1,2(z_1 x_3 / 00)}^1 &= \{(00, 01, 10, 11) | 1_2\}^1 \xRightarrow{con} \{010_2, 011_2, 110_2, 111_2\}^1 \Rightarrow \\
 &\Rightarrow \left\{ \begin{aligned} \{010_2, 011_2, 110_2, 111_2\}^\oplus \\ \{001_{1,2}, 010_2, 011_2, 100_1, 101_1\} \end{aligned} \right\} \Rightarrow \left\{ \begin{aligned} \{110_2, 111_2\}^1 \\ \{001_{1,2}, 100_1, 101_1\}^0 \end{aligned} \right. , \\
 Y_{1,2(z_1 x_3 / 01)}^1 &= \{(00, 01, 10, 11) | 0_{1,2}, 1_2\}^1 \xRightarrow{con} \\
 &\xRightarrow{con} \{000_{1,2}, 001_{1,2}, 010_2, 011_2, 100_{1,2}, 101_{1,2}, 110_2, 111_2\}^1 \Rightarrow
 \end{aligned}$$

$$\Rightarrow \left\{ \begin{array}{l} 000_{1,2}, \cancel{001_{1,2}}, \cancel{010_2}, \cancel{011_2}, 100_{1,2}, 101_{1,2}, 110_2, 111_2 \end{array} \right\}^{\oplus} \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} 001_{1,2}, 010_2, 011_2, 100_1, \cancel{101_1} \\ \{000_{1,2}, 100_{1,2}, 101_{1,2}, 110_2, 111_2\}^1, \\ \{\emptyset\}^0 \end{array} \right\}$$

$$Y_{1,2(z_1x_3/10,11)}^1 = \{(00, 01, \mathbf{10}, \mathbf{11}) | 0_1\}^1 \xRightarrow{con} \{000_1, 001_1, 100_1, 101_1\}^1 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} 000_1, 001_1, \cancel{100_1}, \cancel{101_1} \\ \cancel{001_{1,2}}, 010_2, 011_2, \cancel{100_1}, \cancel{101_1} \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{000_1, 001_1\}^1 \\ \{010_2, 011_2\}^0 \end{array} \right\};$$

$$Y_{1,2(z_2x_3/00)}^1 = \{(\mathbf{00}, 01, 10, 11) | 1_1\}^1 \xRightarrow{con} \{100_1, 101_1, 110_1, 111_1\}^1 \Rightarrow$$

$$\left\{ \begin{array}{l} \cancel{100_1}, \cancel{101_1}, 110_1, 111_1 \\ 001_{1,2}, 010_2, 011_2, \cancel{100_1}, \cancel{101_1} \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{110_1, 111_1\}^1 \\ \{001_{1,2}, 010_2, 011_2\}^0 \end{array} \right\}'$$

$$Y_{1,2(z_2x_3/01)}^1 = \{(00, \mathbf{01}, 10, 11) | 0_{1,2}, 1_1\}^1 \xRightarrow{con}$$

$$\xRightarrow{con} \{000_{1,2}, 001_{1,2}, 010_{1,2}, 011_{1,2}, 100_1, 101_1, 110_1, 111_1\}^1 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} 000_{1,2}, 001_{1,2}, 010_{1,2}, 011_{1,2}, \cancel{100_1}, \cancel{101_1}, 110_1, 111_1 \\ \cancel{001_{1,2}}, \cancel{010_2}, \cancel{011_2}, \cancel{100_1}, \cancel{101_1} \end{array} \right\}^{\oplus} \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} \{000_{1,2}, 010_{1,2}, 011_{1,2}, 110_1, 111_1\}^1 \\ \{\emptyset\}^0 \end{array} \right\},$$

$$Y_{1,2(z_2x_3/10,11)}^1 = \{(00, 01, \mathbf{10}, \mathbf{11}) | 0_2\}^1 \xRightarrow{con} \{000_2, 001_2, 010_2, 011_2\}^1 \Rightarrow$$

$$\left\{ \begin{array}{l} 000_2, 001_2, \cancel{010_2}, \cancel{011_2} \\ \cancel{001_{1,2}}, \cancel{010_2}, \cancel{011_2}, 100_1, 101_1 \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{000_2, 001_2\}^1 \\ \{100_1, 101_1\}^0 \end{array} \right\}.$$

The determined vectors of test codes at 1-level to detect “damages” at two points of the circuit simultaneously, obtained by introducing two non-essential variables of the system of functions $\left\{ \begin{array}{l} f_1(z_1, z_2, x_3) \\ f_2(z_1, z_2, x_3) \end{array} \right\}$, are presented in Table 2.

At the 2-level of the scheme (see Fig. 1) we have a system of functions $\left\{ \begin{array}{l} f_1(z_3, z_4, z_5) = z_3 \vee z_4 \\ f_2(z_3, z_4, z_5) = z_4 \vee z_5 \end{array} \right.$, where $z_3 = \bar{x}_1x_2$, $z_4 = x_1x_2x_3$, $z_5 = x_1\bar{x}_2$, to which

corresponds the system of perfect STFs $\{Y_1^1, Y_2^1\}$ $\left\{ \begin{array}{l} Y_1^1 = \{(1--), (-1-)\}^1 \\ Y_2^1 = \{(-1-), (--1)\}^1 \end{array} \right.$ as well as the sets of system minterms

$Y_{1,2}^1 = \{001_2, 010_{1,2}, 011_{1,2}, 100_1, 101_{1,2}, 110_{1,2}, 111_{1,2}\}^1$ or $Y_{1,2}^0 = \{000_{1,2}, 001_1, 100_2\}^0$.

Since $|Y_{1,2}^1| > |Y_{1,2}^0|$, then on the 2-level of the scheme it is advisable to apply the perfect STF $Y_{1,2}^0$.

Over the system minterms of the set $Y_{1,2}^0$, we perform the 2-partition procedure:

$$Y_{1,2}^0 = \{000_{1,2}, 001_1, 100_2\}^0 \xrightarrow{p^2} \begin{cases} \{l_1 | l_2 l_3\} = \{(0 | 00_{1,2}, 01_1), (1 | 00_2)\}^0 \\ \{l_2 | l_1 l_3\} = \{(0 | 00_{1,2}, 01_1, 10_2), (1 | \emptyset)\}^0 \\ \{l_3 | l_1 l_2\} = \{(0 | 00_{1,2}, 10_2), (1 | 00_1)\}^0 \end{cases}$$

From here we define the vectors of test codes at individual points of the 2-level scheme taking into account the transformation procedure (operator $\xrightarrow{Y \rightarrow C} \Rightarrow$):

$$\begin{aligned} Y_{1,2(z_3/0)}^0 = \{(\mathbf{0}, 1) | 00_{1,2}, 01_1\}^0 &\xrightarrow{con} \{000_{1,2}, 001_1, 100_{1,2}, 101_1\}^0 \Rightarrow \\ \Rightarrow \left\{ \begin{array}{l} \overline{000}_{1,2}, \overline{001}_1, 100_{1,2}, 101_1 \\ \overline{000}_{1,2}, \overline{001}_1, 100_2 \end{array} \right\}^{\oplus} &\Rightarrow \left\{ \begin{array}{l} \{100_{1,2}, 101_1\}^0 \\ \{\emptyset\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{101_2\}^1 \\ \{100\}^0 \end{array} \right\}, \end{aligned}$$

Table 2

$f_{1,2}$	$x_1 x_2 / \sim \sim$	$x_1 x_3 / \sim \sim$	$x_2 x_3 / \sim \sim$
s-a-00	$\begin{pmatrix} 011 \\ 101 \\ 111 \end{pmatrix}_{1,2}^1, \begin{pmatrix} 010_2 \\ 100_1 \end{pmatrix}^0$	$\begin{pmatrix} 110 \\ 111 \end{pmatrix}_2^1, \begin{pmatrix} 001_{1,2} \\ 100_1 \\ 101_1 \end{pmatrix}^0$	$\begin{pmatrix} 110 \\ 111 \end{pmatrix}_1^1, \begin{pmatrix} 001_{1,2} \\ 010_2 \\ 011_2 \end{pmatrix}^0$
s-a-01	$\begin{pmatrix} 000 \\ 001 \\ 100 \\ 101 \\ 110 \\ 111 \end{pmatrix}_2^1$	$\begin{pmatrix} 000_{1,2} \\ 100_{1,2} \\ 101_{1,2} \\ 110_2 \\ 111_2 \end{pmatrix}^1$	$\begin{pmatrix} 000_{1,2} \\ 010_{1,2} \\ 011_{1,2} \\ 110_1 \\ 111_1 \end{pmatrix}^1$
s-a-10	$\begin{pmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 110 \\ 111 \end{pmatrix}_1^1$	$\begin{pmatrix} 000 \\ 001 \\ 010 \\ 011 \end{pmatrix}_1^1, \begin{pmatrix} 010 \\ 011 \end{pmatrix}_2^0$	$\begin{pmatrix} 000 \\ 001 \\ 100 \\ 101 \end{pmatrix}_2^1, \begin{pmatrix} 100 \\ 101 \end{pmatrix}_1^0$
s-a-11	$\begin{pmatrix} 001_{1,2} \\ 010_2 \\ 011_2 \\ 100_1 \\ 101_1 \end{pmatrix}^0$	$\begin{pmatrix} 000 \\ 001 \\ 010 \\ 011 \end{pmatrix}_1^1, \begin{pmatrix} 010 \\ 011 \end{pmatrix}_2^0$	$\begin{pmatrix} 000 \\ 001 \\ 100 \\ 101 \end{pmatrix}_2^1, \begin{pmatrix} 100 \\ 101 \end{pmatrix}_1^0$

$$Y_{1,2(z_3/1)}^0 = \{(0, \mathbf{1}) | 00_2\}^0 \Rightarrow \{000_2, 100_2\}^0 \Rightarrow \left\{ \begin{array}{l} 000_2, 100_2 \\ 000_{1,2}, 001_1, 100_2 \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{000_2\}^0 \\ \{001\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{000_1, 001_{1,2}\}^1 \\ \{\emptyset\}^0 \end{array} \right\};$$

$$Y_{1,2(z_4/0)}^0 = \{(0, \mathbf{1}) | 00_{1,2}, 01_1, 10_2\}^0 \Rightarrow \{000_{1,2}, 001_1, 010_{1,2}, 011_1, 100_2, 110_2\}^0 \Rightarrow \left\{ \begin{array}{l} 000_{1,2}, 001_1, 010_{1,2}, 011_1, 100_2, 110_2 \\ 000_{1,2}, 001_1, 100_2 \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{010_{1,2}, 011_1, 110_2\}^0 \\ \{\emptyset\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{011_2, 110_1\}^1 \\ \{010\}^0 \end{array} \right\},$$

$$Y_{1,2(z_4/1)}^0 = \{(0, \mathbf{1}) | \emptyset\}^0 \Rightarrow \{\emptyset\}^0 \Rightarrow \left\{ \begin{array}{l} \emptyset \\ 000_{1,2}, 001_1, 100_2 \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{\emptyset\}^0 \\ \{000, 001, 100\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{000_{1,2}, 001_{1,2}, 100_{1,2}\}^1 \\ \{\emptyset\}^0 \end{array} \right\};$$

$$Y_{1,2(z_5/0)}^0 = \{(0, \mathbf{1}) | 00_{1,2}, 10_2\}^0 \Rightarrow \{000_{1,2}, 001_{1,2}, 100_2, 101_2\}^0 \Rightarrow \left\{ \begin{array}{l} 000_{1,2}, 001_{1,2}, 100_2, 101_2 \\ 000_{1,2}, 001_1, 100_2 \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{001_{1,2}, 101_2\}^0 \\ \{\emptyset\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{101_1\}^1 \\ \{001\}^0 \end{array} \right\},$$

$$Y_{1,2(z_5/1)}^0 = \{(0, \mathbf{1}) | 00_1\}^0 \Rightarrow \{000_1, 001_1\}^0 \Rightarrow \left\{ \begin{array}{l} 000_1, 001_1 \\ 000_{1,2}, 001_1, 100_2 \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{000_1\}^0 \\ \{100\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{000_2, 100_{1,2}\}^1 \\ \{\emptyset\}^0 \end{array} \right\}.$$

Table 3 contains the truth table of a given system of functions at 2-level of the scheme and its “damaged” variants in the case of stuck-at-

Table 3

«10»	$z_3z_4z_5$	$f_{1,2}$	$f_{1,2z_3}/\sim$		$f_{1,2z_4}/\sim$		$f_{1,2z_5}/\sim$	
			$z_3/0$	$z_3/1$	$z_4/0$	$z_4/1$	$z_5/0$	$z_5/1$
0	000	0	0	1 ₁	0	1 _{1,2}	0	1 ₂
1	001	1 ₂	1 ₂	1 _{1,2}	1 ₂	1 _{1,2}	0	1 ₂
2	010	1 _{1,2}	1 _{1,2}	1 _{1,2}	0	1 _{1,2}	1 _{1,2}	1 _{1,2}
3	011	1 _{1,2}	1 _{1,2}	1 _{1,2}	1 ₂	1 _{1,2}	1 _{1,2}	1 _{1,2}
4	100	1 ₁	0	1 ₁	1 ₁	1 _{1,2}	1 ₁	1 _{1,2}
5	101	1 _{1,2}	1 ₂	1 _{1,2}	1 _{1,2}	1 _{1,2}	1 ₁	1 _{1,2}
6	110	1 _{1,2}	1 _{1,2}	1 _{1,2}	1 ₁	1 _{1,2}	1 _{1,2}	1 _{1,2}
7	111	1 _{1,2}	1 _{1,2}	1 _{1,2}	1 _{1,2}	1 _{1,2}	1 _{1,2}	1 _{1,2}

faults (0/1) at one point, caused by the introduction of one specific “non-essential” variable, where the values of the “damaged” functions $f_{1,2z_3/\sim}$, $f_{1,2z_4/\sim}$ and $f_{1,2z_5/\sim}$, which differ from the values of the functions $f_{1,2}$ of the given system, are highlighted in bold.

This table also illustrates how the proposed method “works” in the event of damage at a single point on the 2-level of the studied circuit, which is shown in Fig. 1. For example, if at the input code $\langle 0,1,0 \rangle$, both outputs of the circuit are observed to be log. 0, at the input code $\langle 0,1,1 \rangle$ we have log 1 only at the output f_2 , and at the input code $\langle 1,1,0 \rangle$ we have log 1 only at the output f_1 , then this indicates that the circuit has damage at a point z_4 of the stuck-at-faults type $z_4 / 0$ (the 6th column in Table 3).

In the case of “damage” stuck-at-faults (0/1) on the 2-level at two points of the scheme, we perform the procedure of 1-partition of the system minterms of the set $Y_{1,2}^0$:

$$Y_{1,2}^0 = \{000_{1,2}, 001_{1,2}, 100_2\}^0 \Rightarrow^{P1} \begin{cases} \{L_1 L_2 | L_3\} = \{(00 | 0_{1,2}, 1_1), (01, 11 | \emptyset), (10 | 0_2)\}^0 \\ \{L_1 L_3 | L_2\} = \{(00 | 0_{1,2}), (01 | 0_1), (10 | 0_2), (11 | \emptyset)\}^0 \\ \{L_2 L_3 | L_1\} = \{(00 | 0_{1,2}, 1_2), (01 | 0_1), (10, 11 | \emptyset)\}^0 \end{cases}$$

The vectors of test codes in the case of introducing two “non-essential” variables are obtained as follows:

$$\begin{aligned} Y_{1,2(z_3z_4/00)}^0 &= \{(00, 01, 10, 11) | 0_{1,2}, 1_1\}^0 \Rightarrow^{con} \\ &\Rightarrow \{000_{1,2}, 001_{1,2}, 010_{1,2}, 011_{1,2}, 100_{1,2}, 101_{1,2}, 110_{1,2}, 111_{1,2}\}^0 \Rightarrow \\ &\Rightarrow \left\{ \overline{000}_{1,2}, \overline{001}_{1,2}, 010_{1,2}, 011_{1,2}, 100_{1,2}, 101_{1,2}, 110_{1,2}, 111_{1,2} \right\}^{\oplus} \Rightarrow \\ &\Rightarrow \left\{ \begin{array}{l} \{010_{1,2}, 011_{1,2}, 100_{1,2}, 101_{1,2}, 110_{1,2}, 111_{1,2}\}^0 \\ \{\emptyset\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{011_2, 101_2, 111_2\}^1 \\ \{010, 100, 110\}^0 \end{array} \right\}^0, \\ Y_{1,2(z_3z_4/01,11)}^0 &= \{(00, \mathbf{01}, 10, \mathbf{11}) | \emptyset\}^0 \Rightarrow^{con} \{\emptyset\}^0 \Rightarrow \left\{ \emptyset, \overline{000}_{1,2}, \overline{001}_{1,2}, \overline{100}_2 \right\}^{\oplus} \Rightarrow \\ &\Rightarrow \left\{ \begin{array}{l} \{\emptyset\}^0 \\ \{000, 001, 100\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{000_{1,2}, 001_{1,2}, 100_{1,2}\}^1 \\ \{\emptyset\}^0 \end{array} \right\}, \\ Y_{1,2(z_3z_4/10)}^0 &= \{(00, 01, \mathbf{10}, 11) | 0_2\}^0 \Rightarrow^{con} \{000_2, 010_2, 100_2, 110_2\}^0 \Rightarrow \\ &\Rightarrow \left\{ \overline{000}_2, \overline{010}_2, \overline{100}_2, \overline{110}_2 \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{000_2, 010_2, 110_2\}^0 \\ \{001\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{000_1, 001_{1,2}, 010_1, 110_1\}^1 \\ \{\emptyset\}^0 \end{array} \right\}; \end{aligned}$$

$$Y_{1,2(z_3z_5/00)}^0 = \{(\mathbf{00}, \mathbf{01}, \mathbf{10}, \mathbf{11}) | 0_{1,2}\}^{\text{con}} \Rightarrow \{000_{1,2}, 001_{1,2}, 100_{1,2}, 101_{1,2}\}^0 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} 000_{1,2}, 001_{1,2}, 100_{1,2}, 101_{1,2} \\ 000_{1,2}, 001_{1,2}, 100_2 \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{001_{1,2}, 100_{1,2}, 101_{1,2}\}^0 \\ \{\emptyset\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{\emptyset\}^1 \\ \{001, 100, 101\}^{0'} \end{array} \right\}$$

$$Y_{1,2(z_3z_5/01)}^0 = \{(00, \mathbf{01}, \mathbf{10}, \mathbf{11}) | 0_1\}^{\text{con}} \Rightarrow \{000_1, 001_1, 100_1, 101_1\}^0 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} 000_1, 001_1, 100_1, 101_1 \\ 000_{1,2}, 001_1, 100_2 \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{001_1, 100_1, 101_1\}^0 \\ \{\emptyset\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{001_2, 100_2, 101_2\}^1 \\ \{\emptyset\}^0 \end{array} \right\},$$

$$Y_{1,2(z_3z_5/10)}^0 = \{(00, \mathbf{01}, \mathbf{10}, \mathbf{11}) | 0_2\}^{\text{con}} \Rightarrow \{000_2, 001_2, 100_2, 101_2\}^0 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} 000_2, 001_2, 100_2, 101_2 \\ 000_{1,2}, 001_1, 100_2 \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{000_2, 001_2, 101_2\}^0 \\ \{\emptyset\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{000_1, 001_1, 101_1\}^1 \\ \{\emptyset\}^0 \end{array} \right\},$$

$$Y_{1,2(z_3z_5/11)}^0 = \{(00, \mathbf{01}, \mathbf{10}, \mathbf{11}) | \emptyset\}^{\text{con}} \Rightarrow \{\emptyset\}^0 \Rightarrow \left\{ \begin{array}{l} \emptyset \\ 000_{1,2}, 001_1, 100_2 \end{array} \right\}^{\oplus} \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} \{\emptyset\}^0 \\ \{000, 001, 100\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{000_{1,2}, 001_{1,2}, 100_{1,2}\}^1 \\ \{\emptyset\}^0 \end{array} \right\};$$

$$Y_{1,2(z_4z_5/00)}^0 = \{(\mathbf{00}, \mathbf{01}, \mathbf{10}, \mathbf{11}) | 0_{1,2}, 1_2\}^{\text{con}} \Rightarrow$$

$$\xrightarrow{\text{con}} \{000_{1,2}, 001_{1,2}, 010_{1,2}, 011_{1,2}, 100_2, 101_2, 110_2, 111_2\}^0 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} 000_{1,2}, 001_{1,2}, 010_{1,2}, 011_{1,2}, 100_2, 101_2, 110_2, 111_2 \\ 000_{1,2}, 001_1, 100_2 \end{array} \right\}^{\oplus} \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} \{001_{1,2}, 010_{1,2}, 011_{1,2}, 101_2, 110_2, 111_2\}^0 \\ \{\emptyset\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{101_1, 110_1, 111_1\}^1 \\ \{001, 010, 011\}^0 \end{array} \right\},$$

$$Y_{1,2(z_4z_5/01)}^0 = \{(00, \mathbf{01}, \mathbf{10}, \mathbf{11}) | 0_1\}^{\text{con}} \Rightarrow \{000_1, 001_1, 010_1, 011_1\}^0 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} 000_1, 001_1, 010_1, 011_1 \\ 000_{1,2}, 001_1, 100_2 \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{000_1, 010_1, 011_1\}^0 \\ \{100\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{000_2, 010_2, 011_2, 100_{1,2}\}^1 \\ \{\emptyset\}^0 \end{array} \right\},$$

$$Y_{1,2(z_4z_5/10,11)}^0 = \{(00, \mathbf{01}, \mathbf{10}, \mathbf{11}) | \emptyset\}^{\text{con}} \Rightarrow \{\emptyset\}^0 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} \emptyset \\ 000_{1,2}, 001_1, 100_2 \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{\emptyset\}^0 \\ \{000, 001, 100\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{000_{1,2}, 001_{1,2}, 100_{1,2}\}^1 \\ \{\emptyset\}^0 \end{array} \right\}.$$

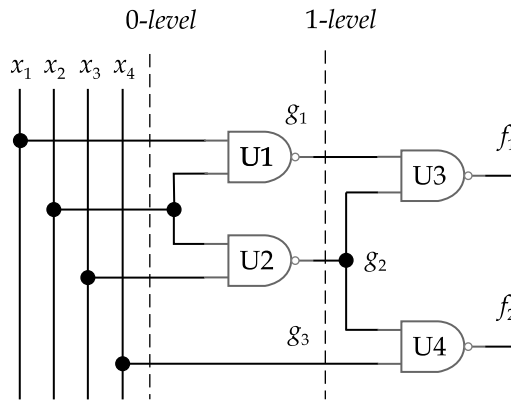


Fig. 2

Table 4

Stuck-at-fault in $f_{1,2}$	$x_3x_4/\sim\sim$	$x_3x_5/\sim\sim$	$x_4x_5/\sim\sim$
s-a-00	$\begin{pmatrix} 011 \\ 101 \\ 111 \end{pmatrix}_2^1, \begin{pmatrix} 010 \\ 100 \\ 110 \end{pmatrix}_2^0$	$\begin{pmatrix} 001 \\ 100 \\ 101 \end{pmatrix}_2^0$	$\begin{pmatrix} 101 \\ 110 \\ 111 \end{pmatrix}_1^1, \begin{pmatrix} 001 \\ 010 \\ 011 \end{pmatrix}_1^0$
s-a-01	$\begin{pmatrix} 000 \\ 001 \\ 100 \end{pmatrix}_{1,2}^1$	$\begin{pmatrix} 001 \\ 100 \\ 101 \end{pmatrix}_2^1$	$\begin{pmatrix} 000_2 \\ 010_2 \\ 011_2 \\ 100_{1,2} \end{pmatrix}_2^1$
s-a-10	$\begin{pmatrix} 000_1 \\ 001_{1,2} \\ 010_1 \\ 110_1 \end{pmatrix}_1^1$	$\begin{pmatrix} 000 \\ 001 \\ 101 \end{pmatrix}_1^1$	$\begin{pmatrix} 000 \\ 001 \\ 100 \end{pmatrix}_{1,2}^1$
s-a-11	$\begin{pmatrix} 000 \\ 001 \\ 100 \end{pmatrix}_{1,2}^1$	$\begin{pmatrix} 000 \\ 001 \\ 100 \end{pmatrix}_{1,2}^1$	$\begin{pmatrix} 000 \\ 001 \\ 100 \end{pmatrix}_{1,2}^1$

Table 4 contains vectors of test codes determined by the proposed method for the case of stuck-at-fault (0/1) type “damages” at arbitrary two points of the 2-level of the circuit.

Example 2. Determine the vectors of test codes to detect multiple stuck-at-faults (0/1) type in the logic circuit shown in Fig. 2, which implements the system of functions (the circuit is borrowed from [18, p.217]).

Solution. In the set-theoretic format, the given system

$$\begin{cases} f_1(x_1, x_2, x_3, x_4) = \overline{\overline{x_1 x_2 x_3}} \\ f_2(x_1, x_2, x_3, x_4) = \overline{\overline{x_2 x_3 x_4}} \end{cases}$$

corresponds to a system of perfect STFs

$$\{Y_1^1, Y_2^1\} \begin{cases} Y_1^1 = \{6, 7, 12, 13, 14, 15\}^1 \\ Y_2^1 = \{0, 2, 4, 6, 7, 8, 10, 12, 14, 15\}^1 \end{cases}$$

which can also be represented by a set of system minterms

$$Y_{1,2}^1 = \{0_2, 2_2, 4_2, 6_{1,2}, 7_{1,2}, 8_2, 10_2, 12_{1,2}, 13_1, 14_{1,2}, 15_{1,2}\}^1$$

or

$$Y_{1,2}^0 = \{0_1, 1_{1,2}, 2_1, 3_{1,2}, 4_1, 5_{1,2}, 8_1, 9_{1,2}, 10_1, 11_{1,2}, 13_2\}^0.$$

Since both sets have the same power, we will use the set $Y_{1,2}^1$ to solve the problem at the 0-level of the scheme.

First, we define vectors of test codes for detecting stuck-at-fault (0/1) damage at separate points on the 0-level of the circuit (Fig. 2), introducing one “non-essential” variable into the given system. To do this, we use the procedure of 3-partition to system minterms of the set $Y_{1,2}^1$:

$$Y_{1,2}^1 = \{0000_2, 0010_2, 0100_2, 0110_{1,2}, 0111_{1,2}, 1000_2, 1010_2, 1100_{1,2}, 1101_1, 1110_{1,2}, 1111_{1,2}\}^1 \Rightarrow$$

$$\begin{cases} \{l_1 | l_2 l_3 l_4\} = \{(0 | 000_2, 010_2, 100_2, 110_{1,2}, 111_{1,2}), (1 | 000_2, 010_2, 100_{1,2}, 101_1, 110_{1,2}, 111_{1,2})\}^1 \\ \{l_2 | l_1 l_3 l_4\} = \{0 | 000_2, 010_2, 100_2, 110_2\}, (1 | 000_2, 010_{1,2}, 011_{1,2}, 100_{1,2}, 101_1, 110_{1,2}, 111_{1,2})\}^1 \\ \{l_3 | l_1 l_2 l_4\} = \{0 | 000_2, 010_2, 100_2, 110_{1,2}, 111_1\}, (1 | 000_2, 010_{1,2}, 011_{1,2}, 100_2, 110_{1,2}, 111_{1,2})\}^1 \\ \{l_4 | l_1 l_2 l_3\} = \{0 | 000_2, 001_2, 010_2, 011_{1,2}, 100_2, 101_2, 110_{1,2}, 111_{1,2}\}, (1 | 011_{1,2}, 110_1, 111_{1,2})\}^1 \end{cases}$$

For the introduced “non-essential” variable x_1 , we obtain the following vectors of test codes:

$$Y_{1,2(x_1/0)}^1 = \{(0, 1) | 000_2, 010_2, 100_2, 110_{1,2}, 111_{1,2}\}^1 \xrightarrow{con} \Rightarrow \{0000_2, 0010_2, 0100_2, 0110_{1,2}, 0111_{1,2}, 1000_2, 1010_2, 1100_2, 1110_{1,2}, 1111_{1,2}\}^1 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} \overline{0000_2}, \overline{0010_2}, \overline{0100_2}, \overline{0110_{1,2}}, \overline{0111_{1,2}}, \overline{1000_2}, \overline{1010_2}, \overline{1100_2}, \overline{1110_{1,2}}, \overline{1111_{1,2}} \\ \overline{0000_2}, \overline{0010_2}, \overline{0100_2}, \overline{0110_{1,2}}, \overline{0111_{1,2}}, \overline{1000_2}, \overline{1010_2}, \overline{1100_2}, \overline{1101_1}, \overline{1110_{1,2}}, \overline{1111_{1,2}} \end{array} \right\}^{\oplus} \Rightarrow$$

$$\Rightarrow \begin{cases} \{1100_2\}^1, \\ \{1101_1\}^0 \end{cases}$$

$$Y_{1,2(x_1/1)}^1 = \{(0, 1) | 000_2, 010_2, 100_{1,2}, 101_1, 110_{1,2}, 111_{1,2}\}^1 \xrightarrow{con} \Rightarrow \{0000_2, 0010_2, 0100_{1,2}, 0101_1, 0110_{1,2}, 0111_{1,2}, 1000_2, 1010_2, 1100_{1,2}, 1101_1, 1110_{1,2}, 1111_{1,2}\}^1 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} \overline{0000_2}, \overline{0010_2}, \overline{0100_{1,2}}, \overline{0101_1}, \overline{0110_{1,2}}, \overline{0111_{1,2}}, \overline{1000_2}, \overline{1010_2}, \overline{1100_{1,2}}, \overline{1101_1}, \overline{1110_{1,2}}, \overline{1111_{1,2}} \\ \overline{0000_2}, \overline{0010_2}, \overline{0100_2}, \overline{0110_{1,2}}, \overline{0111_{1,2}}, \overline{1000_2}, \overline{1010_2}, \overline{1100_{1,2}}, \overline{1101_1}, \overline{1110_{1,2}}, \overline{1111_{1,2}} \end{array} \right\}^{\oplus} \Rightarrow$$

$$\Rightarrow \begin{cases} \{0100_{1,2}, 0101_1\}^1, \\ \{\emptyset\}^0 \end{cases}$$

Performing similar procedures and operations on the system minterms of the set $Y_{1,2}^1$, after the introducing of “non-essential” variables x_2, x_3, x_4 , we obtain the following vectors of test codes:

$$\begin{aligned}
 Y_{1,2(x_2/0)}^1 &\Rightarrow \left\{ \begin{array}{l} \{0110_2, 1100_2, 1110_2\}^1 \\ \{0111_{1,2}, 1101_1, 1111_{1,2}\}^0 \end{array} \right\} \\
 Y_{1,2(x_2/1)}^1 &\Rightarrow \left\{ \begin{array}{l} \{0010_{1,2}, 0011_{1,2}, 1000_{1,2}, 1001_1, 1010_{1,2}, 1011_{1,2}\}^1 \\ \{\emptyset\}^0 \end{array} \right\} \\
 Y_{1,2(x_3/0)}^1 &\Rightarrow \left\{ \begin{array}{l} \{0110_2, 1111_1\}^1 \\ \{0111_{1,2}\}^0 \end{array} \right\} , \quad Y_{1,2(x_3/1)}^1 \Rightarrow \left\{ \begin{array}{l} \{0100_{1,2}, 0101_{1,2}, 1101_{1,2}\}^1 \\ \{\emptyset\}^0 \end{array} \right\} \\
 Y_{1,2(x_4/0)}^1 &\Rightarrow \left\{ \begin{array}{l} \{0001_2, 0011_2, 0101_2, 1001_2, 1011_2, 1101_{1,2}\}^1 \\ \{\emptyset\}^0 \end{array} \right\} \\
 Y_{1,2(x_4/1)}^1 &\Rightarrow \left\{ \begin{array}{l} \{1100_1\}^1 \\ \{0000_2, 0010_2, 0100_2, 1000_2, 1010_2\}^0 \end{array} \right\} .
 \end{aligned}$$

The test code vectors obtained at the 0-level of the scheme are listed in Table 5.

By introducing two “non-essential” variables at the 0-level of the scheme, we will be able to determine the stuck-at-fault damage (0/1) in it at two of its points and the values of the test code vectors. To do this, we will apply the 2-partition procedure to the system minterms of the set $Y_{1,2}^1$:

Table 5

Stuck-at-fault in $f_{1,2}$	x_1/\sim	x_2/\sim	x_3/\sim	x_4/\sim
s-a-0	$(1100_2)^1$ $(1101_1)^0$	$\begin{pmatrix} 0110_2 \\ 1100_2 \\ 1110_2 \end{pmatrix}^1$, $\begin{pmatrix} 0111_{1,2} \\ 1101_1 \\ 1111_{1,2} \end{pmatrix}^0$	$\begin{pmatrix} 0110_2 \\ 1111_1 \end{pmatrix}^1$ $(0111_{1,2})^0$	$\begin{pmatrix} 0001_2 \\ 0011_2 \\ 0101_2 \\ 1001_2 \\ 1011_2 \\ 1101_{1,2} \end{pmatrix}^1$
s-a-1	$\begin{pmatrix} 0100_{1,2} \\ 0101_1 \end{pmatrix}^1$	$\begin{pmatrix} 0010_{1,2} \\ 0011_{1,2} \\ 1000_{1,2} \\ 1001_1 \\ 1010_{1,2} \\ 1011_{1,2} \end{pmatrix}^1$	$\begin{pmatrix} 0100_{1,2} \\ 0101_{1,2} \\ 1101_{1,2} \end{pmatrix}^1$	$(1100_1)^1$ $\begin{pmatrix} 0000_2 \\ 0010_2 \\ 0100_2 \\ 1000_2 \\ 1010_2 \end{pmatrix}^0$

$$\begin{aligned}
 Y_{1,2}^1 &= \{0000_2, 0010_2, 0100_2, 0110_{1,2}, 0111_{1,2}, 1000_2, 1010_2, 1100_{1,2}, 1101_1, 1110_{1,2}, 1111_{1,2}\}^1 \xRightarrow{p^2} \\
 &\left\{ \begin{aligned}
 \{l_1 l_2 | l_3 l_4\} &= \{(00 | 00_2, 10_2), (01 | 00_2, 10_{1,2}, 11_{1,2}), (10 | 00_2, 10_2), (11 | 00_{1,2}, 01_1, 10_{1,2}, 11_{1,2})\}^1 \\
 \{l_1 l_3 | l_2 l_4\} &= \{(00 | 00_2, 10_2), (01 | 00_2, 10_{1,2}, 11_{1,2}), (10 | 00_2, 10_{1,2}, 11_1), (11 | 00_2, 10_{1,2}, 11_{1,2})\}^1 \\
 \{l_1 l_4 | l_2 l_3\} &= \{(00 | 00_2, 01_2, 10_2, 11_{1,2}), (01 | 11_{1,2}), (10 | 00_2, 01_2, 10_{1,2}, 11_{1,2}), (11 | 10_1, 11_{1,2})\}^1 \\
 \{l_2 l_3 | l_1 l_4\} &= \{00 | 00_2, 10_2), (01 | 00_2, 10_2), (10 | 00_2, 10_{1,2}, 11_1), (11 | 00_{1,2}, 01_{1,2}, 10_{1,2}, 11_{1,2})\}^1 \\
 \{l_2 l_4 | l_1 l_3\} &= \{00 | 00_2, 01_2, 10_2, 11_2), (01 | \emptyset), (10 | 00_2, 01_{1,2}, 10_{1,2}, 11_{1,2}), (11 | 01_{1,2}, 10_1, 11_{1,2})\}^1 \\
 \{l_3 l_4 | l_1 l_2\} &= \{00 | 00_2, 01_2, 10_2, 11_{1,2}), (01 | 11_1), (10 | 00_2, 01_{1,2}, 10_2, 11_{1,2}), (11 | 01_{1,2}, 11_{1,2})\}^1
 \end{aligned} \right. \xRightarrow{p^2} \\
 &\left\{ \begin{aligned}
 \{l_1 l_2 | l_3 l_4\} &= \{(00, 10 | 00_2, 10_2), (01 | 00_2, 10_{1,2}, 11_{1,2}), (11 | 00_{1,2}, 01_1, 10_{1,2}, 11_{1,2})\}^1 \\
 \{l_1 l_3 | l_2 l_4\} &= \{(00 | 00_2, 10_2), (01, 11 | 00_2, 10_{1,2}, 11_{1,2}), (10 | 00_2, 10_{1,2}, 11_1)\}^1 \\
 \{l_1 l_4 | l_2 l_3\} &= \{(00 | 00_2, 01_2, 10_2, 11_{1,2}), (01 | 11_{1,2}), (10 | 00_2, 01_2, 10_{1,2}, 11_{1,2}), (11 | 10_1, 11_{1,2})\}^1 \\
 \{l_2 l_3 | l_1 l_4\} &= \{00, 01 | 00_2, 10_2), (10 | 00_2, 10_{1,2}, 11_1), (11 | 00_{1,2}, 01_{1,2}, 10_{1,2}, 11_{1,2})\}^1 \\
 \{l_2 l_4 | l_1 l_3\} &= \{00 | 00_2, 01_2, 10_2, 11_2), (01 | \emptyset), (10 | 00_2, 01_{1,2}, 10_{1,2}, 11_{1,2}), (11 | 01_{1,2}, 10_1, 11_{1,2})\}^1 \\
 \{l_3 l_4 | l_1 l_2\} &= \{00 | 00_2, 01_2, 10_2, 11_{1,2}), (01 | 11_1), (10 | 00_2, 01_{1,2}, 10_2, 11_{1,2}), (11 | 01_{1,2}, 11_{1,2})\}^1
 \end{aligned} \right. \xRightarrow{p^2}
 \end{aligned}$$

Using the example of introducing “non-essential” variables x_1 and x_2 , which reflects the literal mask $\{l_1 l_2 | l_3 l_4\}$, we will show the procedure for determining test code vectors for the stuck-at-fault (0/1) damage case at any two points on the 0-level of the scheme:

$$\begin{aligned}
 Y_{1,2(x_1 x_2 / 00, 10)}^1 &= \{(00, 01, 10, 11) | 00_2, 10_2\}^1 \xRightarrow{con} \\
 &\xRightarrow{con} \{0000_2, 0010_2, 0100_2, 0110_2, 1000_2, 1010_2, 1100_2, 1110_2\}^1 \Rightarrow \\
 &\Rightarrow \left\{ \begin{aligned}
 &\{0000_2, 0010_2, 0100_2, 0110_2, 1000_2, 1010_2, 1100_2, 1110_2 \\
 &0000_2, 0010_2, 0100_2, 0110_{T,2}, 0111_{1,2}, 1000_2, 1010_2, 1100_{T,2}, 1101_1, 1110_{T,2}, 1111_{1,2}\}^{\oplus} \\
 &\{0110_2, 1100_2, 1110_2\}^1 \\
 &\{0111_{1,2}, 1101_1, 1111_{1,2}\}^0
 \end{aligned} \right\} \Rightarrow \\
 Y_{1,2(x_1 x_2 / 01)}^1 &= \{(00, 01, 10, 11) | 00_2, 10_{1,2}, 11_{1,2}\}^1 \xRightarrow{con} \\
 &\xRightarrow{con} \{0000_2, 0010_{1,2}, 0011_{1,2}, 0100_2, 0110_{1,2}, 0111_{1,2}, 1000_2, 1010_{1,2}, 1011_{1,2}, 1100_2, 1110_{1,2}, 1111_{1,2}\}^1 \Rightarrow \\
 &\Rightarrow \left\{ \begin{aligned}
 &\{0000_2, 0010_{1,2}, 0011_{1,2}, 0100_2, 0110_{1,2}, 0111_{1,2}, 1000_2, 1010_{1,2}, 1011_{1,2}, 1100_2, 1110_{1,2}, 1111_{1,2}\}^{\oplus} \\
 &0000_2, 0010_2, 0100_2, 0110_{1,2}, 0111_{1,2}, 1000_2, 1010_2, 1100_{T,2}, 1101_1, 1110_{1,2}, 1111_{1,2} \\
 &\{0010_{1,2}, 0011_{1,2}, 1010_{1,2}, 1011_{1,2}, 1100_2\}^1, \\
 &\{1101_1\}^0
 \end{aligned} \right\} \Rightarrow \\
 Y_{1,2(x_1 x_2 / 11)}^1 &= \{(00, 01, 10, 11) | 00_{1,2}, 01_1, 10_{1,2}, 11_{1,2}\}^1 \xRightarrow{con} \\
 &\xRightarrow{con} \{0000_{1,2}, 0001_1, 0010_{1,2}, 0011_{1,2}, 0100_{1,2}, 0101_1, 0110_{1,2}, 0111_{1,2}, \\
 &1000_{1,2}, 1001_1, 1010_{1,2}, 1011_{1,2}, 1100_{1,2}, 1101_1, 1110_{1,2}, 1111_{1,2}\}^1 \Rightarrow
 \end{aligned}$$

$$\Rightarrow \left\{ \begin{array}{l} 0000_{1,2}, 0001_1, 0010_{1,2}, 0011_{1,2}, 0100_{1,2}, 0101_1, \cancel{0110_{1,2}}, \cancel{0111_{1,2}}, 1000_{1,2}, 1001_1, \\ 1010_{1,2}, 1011_{1,2}, \cancel{1100_{1,2}}, \cancel{1101_1}, \cancel{1110_{1,2}}, \cancel{1111_{1,2}} \\ 0000_2, 0010_2, 0100_2, 0110_{1,2}, 0111_{1,2}, \cancel{1000_2}, \cancel{1010_2}, \cancel{1100_{1,2}}, \cancel{1101_1}, \cancel{1110_{1,2}}, \cancel{1111_{1,2}} \end{array} \right\}^{\oplus} \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} \{0000_{1,2}, 0001_1, 0010_{1,2}, 0011_{1,2}, 0100_{1,2}, 0101_1, 1000_{1,2}, 1001_1, 1010_{1,2}, 1011_{1,2}\}^1 \\ \{\emptyset\}^0 \end{array} \right.$$

The vectors of test codes for the remaining pairs of introduced “non-essential” variables, namely $\langle x_1, x_4 \rangle$, $\langle x_2, x_3 \rangle$, $\langle x_2, x_4 \rangle$ and $\langle x_3, x_4 \rangle$, are easily determined in a similar way, but not given here due to the cumbersome nature of the notation.

To determine the vectors of test codes in the case of three introduced “non-essential” variables at the 0-level of the scheme, which, according to the proposed method, allow to recognize the type and location of damage simultaneously at three points of the studied scheme, we apply the 1-partition procedure to the system minterms of the set $Y_{1,2}^1$:

$$Y_{1,2}^1 = \{0000_2, 0010_2, 0100_2, 0110_{1,2}, 0111_{1,2}, 1000_2, 1010_2, 1100_{1,2}, 1101_1, 1110_{1,2}, 1111_{1,2}\}^1 \xRightarrow{P^1}$$

$$\xRightarrow{P^1} \left\{ \begin{array}{l} \{l_1 l_2 l_3 | l_4\} = \{(000, 001, 010, 100, 101 | 0_2), (011, 111 | 0_{1,2}, 1_{1,2}), (110 | 0_{1,2}, 1_1)\}^1 \\ \{l_1 l_2 l_4 | l_3\} = \{(000, 100 | 0_2, 1_2), (001, 101 | \emptyset), (010 | 0_2, 1_{1,2}), (011 | 1_{1,2}), (110 | 0_{1,2}, 1_{1,2}), (111 | 0_1, 1_{1,2})\}^1 \cup \\ \{l_1 l_3 l_4 | l_2\} = \{(000 | 0_2, 1_2), (001 | \emptyset), (010, 100, 110 | 0_2, 1_{1,2}), (011, 111 | 1_{1,2}), (101 | 1_1)\}^1 \\ \{l_2 l_3 l_4 | l_1\} = \{(000, 010 | 0_2, 1_2), (001, 011 | \emptyset), (100 | 0_2, 1_{1,2}), (101 | 1_1), (110, 111 | 0_{1,2}, 1_{1,2})\}^1 \end{array} \right. \xRightarrow{P^1}$$

Let us show the procedure for determining test code vectors using the example of introducing three “non-essential” variables x_1, x_2 and x_3 , by using a literal mask $\{l_1 l_2 l_3 | l_4\}$:

$$Y_{1,2(x_1, x_2, x_3 / 000, 001, 010, 100, 101)}^1 = \{(000, 001, 010, 011, 100, 101, 110, 111) | 0_2\}^1 \xRightarrow{con}$$

$$\xRightarrow{con} \{0000_2, 0010_2, 0100_2, 0110_2, 1000_2, 1010_2, 1100_2, 1110_2\}^1 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} 0000_2, 0010_2, 0100_2, 0110_2, \cancel{1000_2}, \cancel{1010_2}, 1100_2, 1110_2 \\ 0000_2, 0010_2, 0100_2, 0110_{1,2}, 0111_{1,2}, \cancel{1000_2}, \cancel{1010_2}, \cancel{1100_{1,2}}, 1101_1, \cancel{1110_{1,2}}, 1111_{1,2} \end{array} \right\}^{\oplus} \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} \{0110_2, 1100_2, 1110_2\}^1 \\ \{0111_{1,2}, 1101_1, 1111_{1,2}\}^0 \end{array} \right.$$

$$Y_{1,2(x_1, x_2, x_3 / 011, 111)}^1 = \{(000, 001, 010, 011, 100, 101, 110, 111) | 0_{1,2}, 1_{1,2}\}^1 \xRightarrow{con}$$

$$\xRightarrow{con} \{0000_{1,2}, 0001_{1,2}, 0010_{1,2}, 0011_{1,2}, 0100_{1,2}, 0101_{1,2}, 0110_{1,2}, 0111_{1,2}, 1000_{1,2},$$

$$1001_{1,2}, 1010_{1,2}, 1011_{1,2}, 1100_{1,2}, 1101_{1,2}, 1110_{1,2}, 1111_{1,2}\}^1 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} 0000_{1,2}, 0001_{1,2}, 0010_{1,2}, 0011_{1,2}, 0100_{1,2}, 0101_{1,2}, \cancel{0110_{1,2}}, \cancel{0111_{1,2}}, 1000_{1,2}, 1001_{1,2}, \\ 1010_{1,2}, 1011_{1,2}, \cancel{1100_{1,2}}, 1101_{1,2}, \cancel{1110_{1,2}}, \cancel{1111_{1,2}} \\ 0000_2, 0010_2, 0100_2, 0110_{1,2}, 0111_{1,2}, \cancel{1000_2}, \cancel{1010_2}, \cancel{1100_{1,2}}, \cancel{1101_1}, \cancel{1110_{1,2}}, \cancel{1111_{1,2}} \end{array} \right\}^{\oplus} \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} \{0000_{1,2}, 0001_{1,2}, 0010_{1,2}, 0011_{1,2}, 0100_{1,2}, 0101_{1,2}, 1000_{1,2}, 1001_{1,2}, 1010_{1,2}, 1011_{1,2}, 1101_{1,2}\}^1 \\ \{\emptyset\}^0 \end{array} \right.,$$

$$\begin{aligned} Y_{1,2}^1(x_1, x_2, x_3, /110) &= \{(000, 001, 010, 011, 100, 101, \mathbf{110}, 111) | 0_{1,2}, 1_1\}^1 \xRightarrow{con} \{0000_{1,2}, 0001_1, 0010_{1,2}, 0011_1, \\ &0100_{1,2}, 0101_1, 0110_{1,2}, 0111_1, 1000_{1,2}, 1001_1, 1010_{1,2}, 1011_1, 1100_{1,2}, 1101_1, 1110_{1,2}, 1111_1\}^1 \Rightarrow \\ &\Rightarrow \left\{ \begin{array}{l} 0000_{1,2}, 0001_1, 0010_{1,2}, 0011_1, 0100_{1,2}, 0101_1, \mathbf{0110}_{1,2}, 0111_1, 1000_{1,2}, 1001_1, \\ 1010_{1,2}, 1011_1, \mathbf{1100}_{1,2}, \mathbf{1101}_1, \mathbf{1110}_{1,2}, 1111_1 \\ 0000_2, 0010_2, 0100_2, 0110_2, \mathbf{0111}_{1,2}, \mathbf{1000}_2, \mathbf{1010}_2, \mathbf{1100}_{1,2}, \mathbf{1101}_1, \mathbf{1110}_{1,2}, \mathbf{1111}_{1,2} \end{array} \right\}^{\oplus} \Rightarrow \\ &\Rightarrow \left\{ \begin{array}{l} \{0000_{1,2}, 0001_1, 0010_{1,2}, 0011_1, 0100_{1,2}, 0101_1, 0111_1, 1000_{1,2}, 1001_1, 1010_{1,2}, 1011_1, 1111_1\}^1 \\ \{\emptyset\}^0 \end{array} \right. \end{aligned}$$

Introducing in a similar way the remaining “non-essential” variables for $\langle x_1, x_2, x_4 \rangle$, $\langle x_1, x_3, x_4 \rangle$ and $\langle x_2, x_3, x_4 \rangle$, we obtain vectors of test codes, which are not given here due to the cumbersomeness of the notation.

At the 1-level of the scheme in Fig. 2, we have a system of functions

$$\left\{ \begin{array}{l} f_1(g_1, g_2, g_3) = \overline{g_1 g_2} \\ f_2(g_1, g_2, g_3) = \overline{g_2 g_3} \end{array} \right., \text{ where } g_1 = \overline{x_1 x_2}, g_2 = \overline{x_2 x_3}, g_3 = x_4. \text{ In the set-theo-}$$

retic format, the given system corresponds to the system of perfect STFs

$$\{Y_1^1, Y_2^1\} \left\{ \begin{array}{l} Y_1^1 = \{(0--), (-0-)\}^1 = \{0, 1, 2, 3, 4, 5\}^1 \\ Y_2^1 = \{(-0-), (--0)\}^1 = \{0, 1, 2, 4, 5, 6\}^1 \end{array} \right., \text{ as well as the sets of system}$$

minterms $Y_{1,2}^1 = \{0_{1,2}, 1_{1,2}, 2_{1,2}, 3_1, 4_{1,2}, 5_{1,2}, 6_2\}^1$ and $Y_{1,2}^0 = \{3_2, 6_1, 7_{1,2}\}^1$. Since $|Y_{1,2}^1| > |Y_{1,2}^0|$, it is advisable to use the set of system minterms $Y_{1,2}^0$ at 1-level of the scheme.

We define the vectors of test codes at individual points of the 1-level of the scheme by applying the 2-partition of the system minterms of the set $Y_{1,2}^0$:

$$Y_{1,2}^0 = \{011_2, 110_1, 111_{1,2}\}^0 \xRightarrow{p^2} \left\{ \begin{array}{l} \{l_1 | l_2 l_3\} = \{(0 | 11_2), (1 | 10_1, 11_{1,2})\}^0 \\ \{l_2 | l_1 l_3\} = \{(0 | \emptyset), (1 | 01_2, 10_1, 11_{1,2})\}^0; \\ \{l_3 | l_1 l_2\} = \{(0 | 11_1), (1 | 01_2, 11_{1,2})\}^0 \end{array} \right.;$$

$$\begin{aligned} Y_{1,2(g_1/0)}^0 &= \{(0, 1) | 11_2\}^0 \xRightarrow{con} \{011_2, 111_2\}^0 \Rightarrow \\ &\Rightarrow \left\{ \begin{array}{l} \mathbf{011}_2, 111_2 \\ \mathbf{011}_2, 110_1, \mathbf{111}_{1,2} \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{111_2\}^0 \xrightarrow{Y \rightarrow C} \{110_{1,2}, 111_1\}^1 \\ \{110\}^0 \Rightarrow \{\emptyset\}^0 \end{array} \right., \end{aligned}$$

$$\begin{aligned} Y_{1,2(g_1/1)}^0 &= \{(0, 1) | 10_1, 11_{1,2}\}^0 \xRightarrow{con} \{010_1, 011_{1,2}, 110_1, 111_{1,2}\}^0 \Rightarrow \\ &\Rightarrow \left\{ \begin{array}{l} 010_1, 011_{1,2}, \mathbf{110}_1, \mathbf{111}_{1,2} \\ \mathbf{011}_2, \mathbf{110}_1, \mathbf{111}_{1,2} \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{010_1, 011_{1,2}\}^0 \xrightarrow{Y \rightarrow C} \{010_2\}^1 \\ \{\emptyset\}^0 \Rightarrow \{011\}^0 \end{array} \right., \end{aligned}$$

$$Y_{1,2(g_2/0)}^0 = \{(0,1) | \emptyset\}^0 \Rightarrow \{\emptyset\}^0 \Rightarrow \left\{ \begin{array}{l} \emptyset \\ 011_2, 110_1, 111_{1,2} \end{array} \right\}^{\oplus} \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} \{\emptyset\}^0 \\ \{011, 110, 111\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{011_{1,2}, 110_{1,2}, 111_{1,2}\}^1 \\ \{\emptyset\}^0 \end{array} \right\},$$

$$Y_{1,2(g_2/1)}^0 = \{(0,1) | 01_2, 10_1, 11_{1,2}\}^0 \Rightarrow \{001_2, 011_2, 100_1, 110_1, 101_{1,2}, 111_{1,2}\}^0 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} 001_2, 011_2, 100_1, 110_1, 101_{1,2}, 111_{1,2} \\ 011_2, 110_1, 111_{1,2} \end{array} \right\}^{\oplus} \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} \{001_2, 100_1, 101_{1,2}\}^0 \\ \{\emptyset\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{001_1, 100_2\}^1 \\ \{101\}^0 \end{array} \right\},$$

$$Y_{1,2(g_3/0)}^0 = \{(0,1) | 11_1\}^0 \Rightarrow \{110_1, 111_1\}^0 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} 110_1, 111_1 \\ 011_2, 110_1, 111_{1,2} \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{111_1\}^0 \\ \{011\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{011_{1,2}, 111_2\}^1 \\ \{\emptyset\}^0 \end{array} \right\},$$

$$Y_{1,2(g_3/1)}^0 = \{(0,1) | 01_2, 11_{1,2}\}^0 \Rightarrow \{010_2, 011_2, 110_{1,2}, 111_{1,2}\}^0 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} 010_2, 011_2, 110_{1,2}, 111_{1,2} \\ 011_2, 110_1, 111_{1,2} \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{010_2, 110_{1,2}\}^0 \\ \{\emptyset\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{010_1\}^1 \\ \{110\}^0 \end{array} \right\}.$$

To determine the vectors of test codes in the case of stuck-at-fault (0/1) damage at two points on the 1-level of the circuit, we apply the procedure of 1-partition of the system minterms of the set $Y_{1,2}^0$:

$$Y_{1,2}^0 = \{011_2, 110_1, 111_{1,2}\}^0 \Rightarrow^{p^1} \left\{ \begin{array}{l} \{l_1 l_2 | l_3\} = \{(00, 10 | \emptyset), (01 | 1_2), (11 | 0_1, 1_{1,2})\}^0 \\ \{l_1 l_3 | l_2\} = \{(00 | \emptyset), (01 | 1_2), (10 | 1_1), (11 | 1_{1,2})\}^0 \\ \{l_2 l_3 | l_1\} = \{(00, 01 | \emptyset), (10 | 1_1), (11 | 0_2, 1_{1,2})\}^0 \end{array} \right.$$

Thus, we obtain the vectors of test codes at two “damaged” points on the 1-level of the scheme:

$$Y_{1,2(g_1 g_2 / 00, 10)}^0 = \{(00, 01, 10, 11) | \emptyset\}^0 \Rightarrow \{\emptyset\}^0 \Rightarrow \left\{ \begin{array}{l} \emptyset \\ 011_2, 110_1, 111_{1,2} \end{array} \right\}^{\oplus}$$

$$\Rightarrow \left\{ \begin{array}{l} \{\emptyset\}^0 \\ \{011, 110, 111\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{011_{1,2}, 110_{1,2}, 111_{1,2}\}^1 \\ \{\emptyset\}^0 \end{array} \right\},$$

$$\begin{aligned}
 Y_{1,2(g_1g_2/01)}^0 &= \{(00, \mathbf{01}, 10, 11) | 1_2\}^{\text{con}} \Rightarrow \{001_2, 011_2, 101_2, 111_2\}^0 \Rightarrow \\
 &\Rightarrow \left\{ \begin{array}{l} 001_2, \cancel{011_2}, 101_2, 111_2 \\ \cancel{011_2}, 110_1, \cancel{111}_{1,2} \end{array} \right\}^{\oplus} \Rightarrow \\
 &\Rightarrow \left\{ \begin{array}{l} \{001_2, 101_2, 111_2\}^0 \\ \{110\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{001_1, 101_1, 110_{1,2}, 111_1\}^1 \\ \{\emptyset\}^0 \end{array} \right\}, \\
 \\
 Y_{1,2(g_1g_2/11)}^0 &= \{(00, 01, 10, \mathbf{11}) | 0_1, 1_{1,2}\}^{\text{con}} \Rightarrow \\
 &\xrightarrow{\text{con}} \{000_1, 001_{1,2}, 010_1, 011_{1,2}, 100_1, 101_{1,2}, 110_1, 111_{1,2}\}^0 \Rightarrow \\
 &\Rightarrow \left\{ \begin{array}{l} 000_1, 001_{1,2}, 010_1, 011_{1,2}, 100_1, 101_{1,2}, \cancel{110}_1, \cancel{111}_{1,2} \\ \cancel{011}_2, \cancel{110}_1, \cancel{111}_{1,2} \end{array} \right\}^{\oplus} \Rightarrow \\
 &\Rightarrow \left\{ \begin{array}{l} \{000_1, 001_{1,2}, 010_1, 011_{1,2}, 100_1, 101_{1,2}\}^0 \\ \{\emptyset\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{000_2, 010_2, 100_2\}^1 \\ \{001, 011, 101\}^0 \end{array} \right\}; \\
 \\
 Y_{1,2(g_1g_3/00)}^0 &= \{(\mathbf{00}, 01, 10, 11) | \emptyset\}^{\text{con}} \Rightarrow \{\emptyset\}^0 \Rightarrow \\
 &\Rightarrow \left\{ \begin{array}{l} \emptyset \\ 011_2, 110_1, 111_{1,2} \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{\emptyset\}^0 \\ \{011, 110, 111\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{011_{1,2}, 110_{1,2}, 111_{1,2}\}^1 \\ \{\emptyset\}^0 \end{array} \right\}, \\
 \\
 Y_{1,2(g_1g_3/01)}^0 &= \{(00, \mathbf{01}, 10, 11) | 1_2\}^{\text{con}} \Rightarrow \{010_2, 011_2, 110_2, 111_2\}^0 \Rightarrow \\
 &\Rightarrow \left\{ \begin{array}{l} 010_2, \cancel{011}_2, 110_2, 111_2 \\ \cancel{011}_2, \cancel{110}_1, \cancel{111}_{1,2} \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{010_2, 110_2, 111_2\}^0 \\ \{\emptyset\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{010_1, 110_1, 111_1\}^1 \\ \{\emptyset\}^0 \end{array} \right\}, \\
 \\
 Y_{1,2(g_1g_3/10)}^0 &= \{(00, 01, \mathbf{10}, 11) | 1_1\}^{\text{con}} \Rightarrow \{010_1, 011_1, 110_1, 111_1\}^0 \Rightarrow \\
 &\Rightarrow \left\{ \begin{array}{l} 010_1, 011_1, \cancel{110}_1, 111_1 \\ \cancel{011}_2, \cancel{110}_1, \cancel{111}_{1,2} \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{010_1, 011_1, 111_1\}^0 \\ \{\emptyset\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{010_2, 011_2, 111_2\}^1 \\ \{\emptyset\}^0 \end{array} \right\}, \\
 \\
 Y_{1,2(g_1g_3/11)}^0 &= \{(00, 01, 10, \mathbf{11}) | 1_{1,2}\}^{\text{con}} \Rightarrow \{010_{1,2}, 011_{1,2}, 110_{1,2}, 111_{1,2}\}^0 \Rightarrow \\
 &\Rightarrow \left\{ \begin{array}{l} 010_{1,2}, 011_{1,2}, 110_{1,2}, \cancel{111}_{1,2} \\ \cancel{011}_2, \cancel{110}_1, \cancel{111}_{1,2} \end{array} \right\}^{\oplus} \Rightarrow \\
 &\Rightarrow \left\{ \begin{array}{l} \{010_{1,2}, 011_{1,2}, 110_{1,2}\}^0 \\ \{\emptyset\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{\emptyset\}^1 \\ \{010, 011, 110\}^0 \end{array} \right\};
 \end{aligned}$$

$$Y_{1,2(g_2g_3/00,01)}^0 = \{(00, \mathbf{01}, 10, 11) | \emptyset\}^0 \Rightarrow \{\emptyset\}^0 \Rightarrow$$

$$\Rightarrow \{\emptyset\}^0 \Rightarrow \left\{ \begin{array}{l} \emptyset \\ 011_2, 110_1, 111_{1,2} \end{array} \right\}^{\oplus} \Rightarrow \left\{ \begin{array}{l} \{\emptyset\}^0 \\ \{011, 110, 111\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{011_{1,2}, 110_{1,2}, 111_{1,2}\}^1 \\ \{\emptyset\}^0 \end{array} \right\},$$

$$Y_{1,2(g_2g_3/10)}^0 = \{(00, 01, \mathbf{10}, 11) | 1_1\}^0 \Rightarrow \{100_1, 101_1, 110_1, 111_1\}^0 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} 100_1, 101_1, \cancel{110}_1, 111_1 \\ 011_2, \cancel{110}_1, \cancel{111}_{1,2} \end{array} \right\}^{\oplus} \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} \{100_1, 101_1, 111_1\}^0 \\ \{011\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{011_{1,2}, 100_2, 101_2, 111_2\}^1 \\ \{\emptyset\}^0 \end{array} \right\},$$

$$Y_{1,2(g_2g_3/11)}^0 = \{(00, 01, 10, \mathbf{11}) | 0_2, 1_{1,2}\}^0 \Rightarrow$$

$$\Rightarrow \{000_2, 001_2, 010_2, 011_2, 100_{1,2}, 101_{1,2}, 110_{1,2}, 111_{1,2}\}^0 \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} 000_2, 001_2, 010_2, \cancel{011}_2, 100_{1,2}, 101_{1,2}, 110_{1,2}, \cancel{111}_{1,2} \\ \cancel{011}_2, \cancel{110}_1, \cancel{111}_{1,2} \end{array} \right\}^{\oplus} \Rightarrow$$

$$\Rightarrow \left\{ \begin{array}{l} \{000_2, 001_2, 010_2, 100_{1,2}, 101_{1,2}, 110_{1,2}\}^0 \\ \{\emptyset\}^0 \end{array} \right\} \xrightarrow{Y \rightarrow C} \left\{ \begin{array}{l} \{000_1, 001_1, 010_1\}^1 \\ \{100, 101, 110\}^0 \end{array} \right\}.$$

Table 6

Stuck-at-fault in $f_{1,2}$	$g_1g_2/\sim\sim$	$g_1g_3/\sim\sim$	$g_2g_3/\sim\sim$
s-a-00	$\begin{pmatrix} 011_{1,2} \\ 110_{1,2} \\ 111_{1,2} \end{pmatrix}^1$	$\begin{pmatrix} 011_{1,2} \\ 110_{1,2} \\ 111_{1,2} \end{pmatrix}^1$	$\begin{pmatrix} 011_{1,2} \\ 110_{1,2} \\ 111_{1,2} \end{pmatrix}^1$
s-a-01	$\begin{pmatrix} 001_1 \\ 101_1 \\ 110_{1,2} \\ 111_1 \end{pmatrix}^1$	$\begin{pmatrix} 010_1 \\ 110_1 \\ 111_1 \end{pmatrix}^1$	$\begin{pmatrix} 011_{1,2} \\ 110_{1,2} \\ 111_{1,2} \end{pmatrix}^1$
s-a-10	$\begin{pmatrix} 011_{1,2} \\ 110_{1,2} \\ 111_{1,2} \end{pmatrix}^1$	$\begin{pmatrix} 010_2 \\ 011_2 \\ 111_2 \end{pmatrix}^1$	$\begin{pmatrix} 011_{1,2} \\ 100_2 \\ 101_2 \\ 111_2 \end{pmatrix}^1$
s-a-11	$\begin{pmatrix} 000_2 \\ 010_2 \\ 100_2 \end{pmatrix}^1, \begin{pmatrix} 001 \\ 011 \\ 101 \end{pmatrix}^0$	$\begin{pmatrix} 010 \\ 011 \\ 110 \end{pmatrix}^0$	$\begin{pmatrix} 000_1 \\ 001_1 \\ 010_1 \end{pmatrix}^1, \begin{pmatrix} 100 \\ 101 \\ 110 \end{pmatrix}^0$

Table 6 contains vectors of test codes to detect stuck-at-faults at arbitrary two points on the 1-level of the circuit.

Conclusion

A new method of generating test code vectors for determining the location and type of stuck-at-faults (0/1) of single and multiple faults in combinational PIPO circuits is proposed. It is based on the artificial introduction of one or more non-essential variables into the studied circuit and the application of the procedure of q -partition of system minterms of a given system of Boolean functions. Compared to known methods and algorithms, the proposed method is characterized by a relatively simpler implementation and reliability of the final results of practical diagnostics without the use of additional tools and restrictions. The examples given in the article of determining the location and type of stuck-at-faults (0/1) of single and multiple damages illustrate the effectiveness of the proposed method.

REFERENCES

1. Rytsar B.Ye. A Simple Stuck-at-faults Detection Method in Digital Combinational Circuits. *Control Systems and Computers*, 2023, Vol. 301 (1), 5–17. <https://doi.org/10.15407/csc.2023.01.005>
2. Rytsar B.Ye. A Simple Stuck-at-faults Detection Method in Digital Combinational Circuits. II. *Control Systems and Computers*, 2024, Vol. 301 (1), 3–17. <https://doi.org/10.15407/csc.2024.01.003>
3. Thamarai S.M., Kuppasamy R., Meyyappan T. Fault Detection and Test Minimization Methods for Comnational Circuits – A Survey. *Inter. J. Comp. Trends and Technology*, 2011 ,Vol. 21 (2), 140–146.
4. Takahashi H., Boateng K.O., Saluja K.K., Taramatsu Y. On diagnosing multiple stuck-at faults using multiple and single fault simulation in combinational circuits. in *IEEE Trans. Comp.-Aid. Design of Integrated Circuits and Systems*, 2002, Vol. 21 (3), 362–368. <https://doi.org/10.1109/43.986429>
5. Jutman A., Ubar R. Design error diagnostic in digital circuits with stuck-at-fault model. *Microelectronics Reliability*, 2000, Vol. 40 (2), 307–320. [https://doi.org/10.1016/S0026-2714\(99\)00203-6](https://doi.org/10.1016/S0026-2714(99)00203-6)
6. Huang S.Y, Chen K.C, Cheng K.T. Incremental Logic Rectification. In: *Proc. VLSI Test Symposium*, 1997, 134B139 p.
7. Sarang S. Samangadkar, Shridhar S. Dudam, Amit Kumar Sinha. *Fault Diagnosis in Combinational Logic Circuits: A Survey*. *IJSRD*, 2015, Vol. 3 (2), 2051–2054.
8. Liu J., Veneris A., Abadir M.S. Efficient and Exact Diagnosis of Multiple Stuck-At Faults. 2002. URL: <https://www.researchgate.net/publication/228964338>
9. Dall’Occo F., Dalpasso M., Favalli M. Techniques for SAT-Based Boolean Reasoning on Multiple Faults Affecting Logic Cells and Interconnects in Digital ICs. *Electronics*, 2022, Vol. 11 (3), Article 382. <https://doi.org/10.3390/electronics11030382>
10. Lin C.C, Chen K.C, Chang S.C, Marek-Sadowska M, Cheng K.T. Logic Synthesis for Engineering Change. *Proc. Design Automation Conference*, Jun. 1995, 647–652. <https://doi.org/10.1145/217474.217604>
11. Karkouri Y., Aboulhamid M. Multiple Stuck-at Fault in Logic Circuits. URL: http://www.iro.umontreal.ca/~aboulham/pdfs_sources/KCCVLSI.pdf

12. Liu Y., Xu Q. On modeling faults in FinFET logic circuits. *IEEE Inter. Test Conf.*, 2012, Paper 11.3. <https://doi.org/10.1109/TEST.2012.6401565>
13. Parag K. Lala. *An Introduction to Logic Circuit Testing*. Morgan & Claypool, 2009, 111 p.
14. Kim J.C., Vishvani D. Agraval, Kewal K. Saluja. Multiple Faults: Modeling, Simulation and Test. *7th ASPDAC and 15th Int'l Conf. on VLSI Design*, Jan. 2002.
15. Rytsar B.Ye. Decomposition of the Boolean Functions by the q -decomposition method. *USiM*, Kyiv, 1999, Issue 6, 29–42. [In Ukrainian: Рицар Б.Є. Декомпозиція булевих функцій методом q -розбиття. УСиМ.]
16. Rytsar B.Ye. *Theoretic-Set Optimization Methods for Logic Synthesis of Combinational Networks: dys. doktora tekhn. nauk*. Lviv, 2004, 348 p. [In Ukrainian: Рицар Б. Є. Теоретико-множинні оптимізаційні методи логікового синтезу комбінаційних мереж: дис. д-ра техн. наук. Рицар Б.Є. Львів, 2004. 348 с.]
17. Sasao T. *Switching Theory for Logic Synthesis*. Kluwer Academic Publishers, 1999, 361 p. <https://doi.org/10.1007/978-1-4615-5139-3>
18. Kohavi Z., Jha N. *Switching and Finite Automata Theory*. Cambridge University Press, 2010, 206–250. <https://doi.org/10.1017/CBO9780511816239>

Received 08.05.2025

Б.Є. РИЦАР, д-р техн. наук, професор,
Національний університет «Львівська політехніка»,
Інститут інформаційно-комунікаційних технологій та електронної техніки,
вул. Степана Бандери, 12, м. Львів, 79013, Україна
<https://orcid.org/0000-0002-2929-2954>
bohdanrytsar@gmail.com

НОВИЙ МЕТОД ГЕНЕРУВАННЯ ТЕСТОВИХ КОДІВ ДЛЯ ВИЯВЛЕННЯ МНОЖИННИХ ПОШКОДЖЕНЬ *STUCK-AT-FAULTS* У КОМБІНАЦІЙНИХ СХЕМАХ. Частина 2

Вступ. Діагностувати множинні несправності *stuck-at-faults* (0/1) у цифрових схемах типу *PIPO* значно складніше, ніж у пристроях типу *PISO* (частина 1). Функції системи переважно взаємопов'язані між собою, а отже, виявлене пошкодження в якійсь одній функції схеми може передатися інших функцій системи, що описує роботу досліджуваної схеми. Відповідно, методи генерування тестових кодів на основі одиночних несправностей не працюють для схем типу *PIPO*, а ті методи й алгоритми діагностики, що використовують моделювання одиночних несправностей, ускладнені додатковими процедурами, не дають надійного результату. Аналогічний висновок про певні практичні обмеження можна також зробити і до аналітичних підходів до розв'язання зазначеної проблеми виявлення множинних несправностей *stuck-at-faults* (0/1) у цифрових схемах типу *PIPO*.

Мета статті. Запропонувати метод генерування векторів тестових кодів для виявлення як одиночних, так і множинних пошкоджень типу *stuck-at-faults* (0/1) у комбінаційних пристроях типу *PIPO*, який порівняно з відомими методами й алгоритмами може забезпечувати достовірні результати з допомогою реалізації простих операцій і процедур.

Методи. Запропонований метод генерування тестових кодів ґрунтується на числовому теоретико-множинному підході до реалізації всіх операцій і процедур, а саме: штучного впровадження у буловий простір заданої системи повних функцій $F(X)$, $X = \{x_1, x_2, \dots, x_n\}$, що описує роботу досліджуваної *PIPO*-схеми, одної або більше (до $n - 1$) неістотних змінних та застосуванні процедури q -розбиття до системних мінтермів з урахуванням індексів функцій заданої системи $F(X)$.

Результати. Завдяки застосуванню процедури q -розбиття системних мінтермів впровадження «неістотних» змінних у буловий простір заданої системи забезпечує виявлення всіх можливих як одиночних, так і множинних пошкоджень типу *stuck-at-faults* (0/1) у досліджуваній схемі. Унаслідок цього формуються 2^r ($r = 1, 2, \dots, n-1$) псевдодосконалих ТМФ «пошкодженої» системи $F(X)$, на підставі яких після виконання простих операцій спрощення одержуються шукані вектори тестових кодів, з допомогою яких можна визначити в схемі як місце пошкодження, так і тип одиночного та множинного *stuck-at-faults* (0/1) пошкодження.

Висновки. Запропоновано новий метод генерування векторів тестових кодів для визначення місця і типу *stuck-at-faults* (0/1) одиночних і множинних пошкоджень у комбінаційних РІРО-схемах, що ґрунтується на штучному впровадженні в досліджувану схему одної і більше неістотних змінних та застосуванні процедури q -розбиття системних мінтермів заданої системи булових функцій. Порівняно з відомими методами та алгоритмами метод відрізняється відносно простішою реалізацією та надійністю отриманих остаточних результатів практичної діагностики без застосування додаткових засобів та обмежень. Наведені в статті приклади визначення місця і типу *stuck-at-faults* (0/1) одиночних і множинних пошкоджень ілюструють ефективність запропонованого методу.

Ключові слова: комбінаційна РІРО-схема, одиночне та множинне пошкодження типу *stuck-at-faults* (0/1), процедура q -розбиття системних мінтермів, неістотні змінні, вектор тестових кодів.

<https://doi.org/10.15407/intechsys.2025.03.030>
УДК 004.8 + 004.032.26

О.А. УРСАТЬЄВ, канд. техн. наук, старш. наук. співроб., пров. наук. співроб.,
Інститут інформаційних технологій та систем НАН України,
просп. Акад. Глушкова, 40, м. Київ, 03187, Україна
<https://org/0009-0009-8323-0525>
aleksei@irtc.org.ua

О.Є. ВОЛКОВ, канд. техн. наук., старш. дослідник, директор,
Інститут інформаційних технологій та систем НАН України,
просп. Акад. Глушкова, 40, м. Київ, 03187, Україна
<https://orcid.org/0000-0002-5418-6723>
alexvolk@ukr.net

ПІДХОДИ ДО СТВОРЕННЯ МУЛЬТИАГЕНТНИХ СИСТЕМ І ГЛИБОКОГО ПОСИЛЕНОГО НАВЧАННЯ

Розглянуто зарубіжний досвід розробки та застосування штучного інтелекту за допомоги глибокого посиленого навчання нейромереж для розв'язання проблем з якими стикаються рухливі об'єкти у невідомих, можливо частково спостережуваних середовищах, для опису яких не існує математичної моделі. Надано таксономію різних завдань, що виникають при управлінні БПЛА чи роєм дронів, і наведено запропоновані безмодельні алгоритми глибокого посиленого навчання для розв'язання кожного з них. Виконано математичну формалізацію завдань у сфері управління БПЛА при посиленому навчанні, зокрема розглянуто парадигму навчання у багатоагентному середовищі. Розглянуто рішення деяких завдань щодо використання БПЛА.

Ключові слова: *безпілотні рухомі об'єкти, безпілотні літальні апарати БПЛА, управління роєм БПЛА, глибоке посилене навчання, ментальна модель світу, навчання нейромережі в уяві з застосуванням моделі світу, парадигми навчання агентів та схеми виконання завдань у багатоагентному середовищі.*

Цитування: Урсатьєв О.А., Волков О.Є. Підходи до створення мультиагентних систем і глибокого посиленого навчання дронів. *Information Technologies and Systems*, Київ, 2025, Том 3 (3), 30–55. <https://doi.org/10.15407/intechsys.2025.03.030>

© Видавець ВД «Академперіодика» НАН України, 2025. Стаття опублікована на умовах відкритого доступу за ліцензією CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Вступ

Безпілотні літальні апарати (БПЛА, *Unmanned Aerial Vehicles (UAVs)*) все частіше використовуються у багатьох складних та різноманітних завданнях, що належать до цивільних та військових галузей. БПЛА — це клас літальних апаратів, які можуть літати без присутності на борту людини-пілота. Їх зазвичай називають дронами. Вони можуть функціонувати з різним ступенем автономності: під дистанційним керуванням оператора-людини або автономно за допомоги бортових комп'ютерів. Вимога значного рівня автономності диктується необхідністю виконувати заплановані завдання у несподіваних ситуаціях без втручання людини. В огляді наведено стиснену класифікацію БПЛА і сформульовано мету, а саме: забезпечити бажані рівні досяжності автономності та функціонування БПЛА на кожному з етапів польотного завдання, та обґрунтовано стан використовуваних на той час методів рівня розвитку *state-of-the-art* глибокого посиленого навчання (навчання з підкріпленням, *Reinforcement Learning, DRL*) [1]. Наведені методи *DRL* були застосовані для забезпечення стабільної та плавної навігації БПЛА шляхом навчання *середовищ, змодельованих на комп'ютері*.

Класифікацію БПЛА було зроблено за багатьма важливими аспектами, такими як конфігурація верхнього рівня, яка передбачає виконання крила фіксованим, гвинтокрила та гібридного, гранична висота, середня злітна вага, рівень автономності тощо. Тип БПЛА із фіксованим крилом має жорстке крило з аеродинамічним профілем, що працює, збільшуючи поступальну швидкість польоту як звичайні літаки. Ці БПЛА підтримують тривалі польоти на витривалість та баражування, забезпечує високошвидкісний рух та підтримує високе корисне навантаження порівняно з конфігурацією гвинтокрилих БПЛА. Деякі з проблем полягають у необхідності злітно-посадкової смуги внаслідок вимоги поступальної швидкості повітря, завдання зависання не виконуються, оскільки ці БПЛА повинні продовжувати безперервний політ до посадки наприкінці будь-якого польоту.

Безпілотний вертоліт. Його лопасті, що обертаються, можуть створювати аеродинамічну силу тяги, достатню для польоту без необхідності відносної швидкості повітря. Ця конфігурація БПЛА має перевагу маневровості: може виконувати вертикальний зліт / посадку, літати на малих висотах, наприклад, у складних міських умовах, виконувати завдання зависання. Однак такий БПЛА не може зберігати те ж корисне навантаження, яке підтримує конфігурація з фіксованим крилом. Технічно ця конфігурація далі поділяється на підкатегорії залежно від кількості задіяних роторів; вона охоплює однороторні (вертольоти) і багатороторні (трироторні дрони, квадрокоптери тощо). Однороторні механічно складні і високовартісні, оскільки вони можуть злітати або приземлятися вертикально, підтримувати відносно високе корисне навантаження, тоді як багато-

роторні набагато швидші і здатні зависати або переміщатися навколо мети дуже плавно.

Гібридна конфігурація – це особливий тип повітряної платформи, що класифікуються як конвертоплани та хвостові літальні апарати. Гібридні БПЛА поєднують у собі переваги мультикоптерів та літаків з фіксованим крилом: вертикальний зліт, посадка та низьке енергоспоживання. Мають складну, змішану динаміку.

У 2005 році було введено рівні автономного управління (*Autonomous Control Levels, ACL*) або рівня незалежності від участі людини у пілотуванні. Кожен рівень спирається на три обставини: рівень незалежності, складність виконуваного завдання та рівень конфігурації середовища [1]. Перший – дистанційно пілотований. Цей рівень повністю контролює сертифікований експерт, який отримує візуальний зворотний зв'язок або сенсорні дані. Другий – дистанційно керований, званий напіваавтономним. Цей рівень дозволяє транспортному засобу керувати собою на основі рішень, які приймаються під час спостереження за пілотом. Останній – повністю автономний рівень. На цьому рівні транспортному засобу надаються загальні завдання для виконання, і він стає здатним робити це, не уточнюючи, як виконати завдання.

У [1] розглянуто три основні проблеми, з якими стикаються БПЛА: планування шляху, навігація та керування (*path planning of drones, navigation, and control*). У складних системах, до яких, безумовно, відноситься БПЛА, із застосуванням штучного інтелекту (AI) для розв'язання цих проблем, наразі використовується посилене навчання (RL). Для забезпечення бажаного функціонування БПЛА було адаптовано безліч наявних та розроблені алгоритми DRL і згенероване на їх основі AI, що реалізує допомогу у невідомих середовищах, для опису яких немає математичної моделі.

Статтю присвячено аналізу сучасного зарубіжного досвіду застосування аналітичних платформ керування рухомими об'єктами, зокрема безпілотними автомобілями та БПЛА, у завданнях планування шляху, навігації та керування за допомогою штучного інтелекту, який генерується глибокими нейронними мережами, а саме посиленим навчанням у складному середовищі.

Мета статті – ознайомити фахівців з предметної області, чия основна робоча функція перебуває поза цариною машинного навчання, із проблемами, пов'язаними із застосуванням штучного інтелекту для розв'язання цих завдань, надійних та складних глибоких нейронних мереж а також їх навчання, що залишається складним завданням і потребує великої кількості даних та практичного досвіду і знань. Це може бути формою громадянської науки (*citizen science*) і сприятиме відтворенню досліджень та демократизації штучного інтелекту [2].

Огляд глибокого посиленого навчання (Deep Reinforcement Learning, DRL)

Навчання через взаємодію є основоположною ідеєю, що лежить в основі майже всього навчання та інтелекту. Найважливішою особливістю, яка відрізняє посилене навчання від інших типів навчання, є те, що воно використовує навчальну інформацію, яка оцінює виконані дії, а не інструктує, надаючи правильні дії. Глибоке навчання (DL) в нейронних мережах (NN) є актуальним для контрольованого навчання (SL) і самонавчання (UL) (рис. 1). Такі NN вчаться приймати / кодувати / передбачати / класифікувати шаблони або послідовності шаблонів, але вони не навчаються діяти в більш загальному сенсі посиленого навчання, RL в невідомих середовищах [3, 4].

Supervised and Unsupervised – контрольоване навчання і самонавчання відповідно. Самонавчання можна розділити на два підтипи: кластеризація та скорочення розмірності (*Dimension Reduction*). Кластеризація організовує дані в групи на основі критеріїв подібності. Скорочення розмірності займається відображенням даних із простору з високою розмірністю в простір з низькою розмірністю зі збереженням основних характеристик даних. *Reinforcement Learning* також можна розділити на два підтипи: Поведінка агентів та недиференційовані алгоритми, такі що не належать до жодного класу. Принцип навчання методом спроб і помилок при взаємодії агента з середовищем можна перефразувати для роботи з алгоритмами, які не можуть бути математично визначені. Їх можна навчити за допомоги посиленого навчання.

Посилене навчання (RL) – це парадигма для навчання задач прийняття рішень, яка може дати змогу агентам навчатися та адаптуватися до ситуацій у режимі онлайн. RL (див. рис. 1) – це форма навчання методом спроб і помилок, яка працює з винагородами та покараннями. Тобто, RL навчання – це навчання з того, що необхідно зробити і як зіставляти ситуації з діями, щоб максимізувати винагороду. Цей підтип навчання («Поведінка агентів», див. рис. 1)

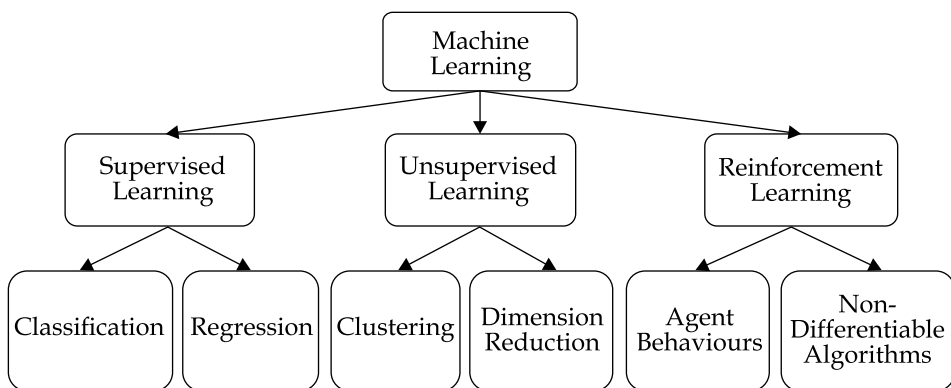


Рис. 1. Типи машинного навчання

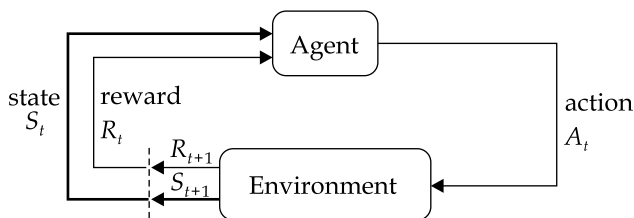


Рис. 2. Взаємодія агента та середовища у Марківському процесі прийняття рішень [4, 7]

часто застосовується для агентних *симуляцій*¹, в яких агенту необхідно розробити стратегію успішної взаємодії з навколишнім середовищем. Рівень успіху вимірюється як винагорода, яку агент отримує від навколишнього середовища після виконання дії (рис. 2), що може бути отримана одразу або згодом. Одна з основних проблем *RL* стосується розуміння щодо присвоєння винагород за дії, які виконувалися в далекому минулому [5, 6].

Інформація, що циркулює між агентом та середовищем, у системі *RL*: *Environment* (Середовище), *Agent* (Агент), *State* (Стан), *Action* (Дія), *Reward* (Нагорода). Середовище — це світ, в якому діє агент. Середовище і агент в кожен момент часу мають певний стан. Агент може спостерігати за середовищем. Аналізуючи дані спостережень, агент вибирає дію, що призводить до зміни стану середовища. Після виконання дії, середовище реагує на нього, повертаючи агенту значення нагороди, та набуває нового стану.

Агент у контексті посиленого навчання — це сутність, яка інтерактивна зі середовищем, інтелектуальна, намагається досягти своєї мети і максимізувати свою нагороду, враховуючи поточний стан та досвід. Політика, винагорода, цінність та модель середовища є основними елементами *RL*. *Політика* визначає план дій агента (тобто, як агент реагує на різні ситуації навколишнього середовища та як він переводить стани в дії), або ймовірність того, що агент вдасться до певної дії, коли середовище перебуває в певному стані. *Винагороди* — це числові значення, які середовище дає агентові у відповідь на пару стан-дія. Ці значення винагороди описують негайну внутрішню бажаність станів навколишнього середовища. *Функція цінності* — це довгострокова версія функції винагороди, що обчислює дисконтовану дохідність, починаючи з певного стану, дотримуючись певної політики. Модель середовища надає поведінку середовища, що допомагає підвищити продуктивність алгоритму за допомогою розуміння довкілля. Основна мета будь-якого алгоритму посиленого навчання — дозволити агенту швидко вивчити оптимальну політику, яка точно досягає поставленого завдання і, таким чином, призводить

¹ Симулятор імітує керування, в загальному випадку, будь-яким процесом, і містить засоби, що відображають частину реальних явищ і властивостей у віртуальному середовищі, що створює враження дійсності.

до найбільшого значення винагороди. Глибоке посилене навчання готове зробити революцію в галузі штучного інтелекту і є кроком до створення автономних систем з більш високим рівнем розуміння візуального світу [1, 4–7].

Політика. Те, що агент намагається вивчити у сценарії *RL*, це політика, яка максимізує винагороду. Політика визначає ймовірність того, що агент вдасться до певної дії, коли середовище знаходиться в певному стані. Політика, яка максимізує загальну винагороду, називається оптимальною політикою. Вважається, що методи, засновані на політиці, більш стабільні, а методи, не засновані на політиці (*off-Policy*), більш ефективні та забезпечують кращий баланс між дослідженням нових політик та використанням найкращих на даний момент політик [3].

Підходи до *DRL* поділяють три основні категорії: засновані на цінностях (*Value-based approach*), на політиці (*Policy-based*), і підхід, заснований на моделі (*Model-based*) або без моделей (*Model-Free*). У навчанні на основі цінностей агент має на меті знайти політику, яка максимізує функцію цінності у довгостроковій перспективі у послідовності дій. Потім агент повинен знайти політику, що веде до оптимального значення для цільової функції. Ця категорія далі поділяється на детермінований та стохастичний підходи. Перша політика застосовує одну й ту саму дію у будь-якому стані, тоді як останній включає варіації дій, засновані на ймовірнісних оцінках. Нарешті, *DRL* з моделлю залежить від надання агенту моделі середовища з вимогою до агента вивчити її для виконання завдань у цьому конкретному середовищі.

Незважаючи на відмінності, реалізовані в трьох підходах, вони мають деякі важливі характеристики, успадковані від концепцій *DRL*. Керування, реалізоване за допомоги *DRL*, діє як керування із замкнутим циклом, тоді як винагорода є зворотним зв'язком системи. Ця винагорода затримується, хоча багато алгоритмів намагаються зменшити цю затримку. Більше того, алгоритми, реалізовані в *DRL*, мають послідовну поведінку прийняття рішень, з довгостроковими винагородами, що залежать від послідовності дій. Концепція, яка називається проблемою призначення кредиту, описує залежність реалізації *DRL* від часу, оскільки деякі дії показують свої наслідки через деякий час і багато проміжних станів системи прийняття рішень [7].

На рис. 3 подано огляд типових та популярних алгоритмів посиленого навчання у структурному вигляді [7, 8]. Алгоритми класифіковано з різних точок зору, включно з описаними раніше методами, та наведено їх комбінацію, засновану на цінностях та політиці, методи Монте-Карло (*MC*), тимчасової різниці (*TD*) та на основі політик і поза політикою.

***Model-Free* і *Model-Based* методи *RL*.** Існує два основних типи алгоритмів *RL*, які відрізняються один від одного тим, як вони працюють із ймовірностями переходу станів. Алгоритм *RL* з урахуванням

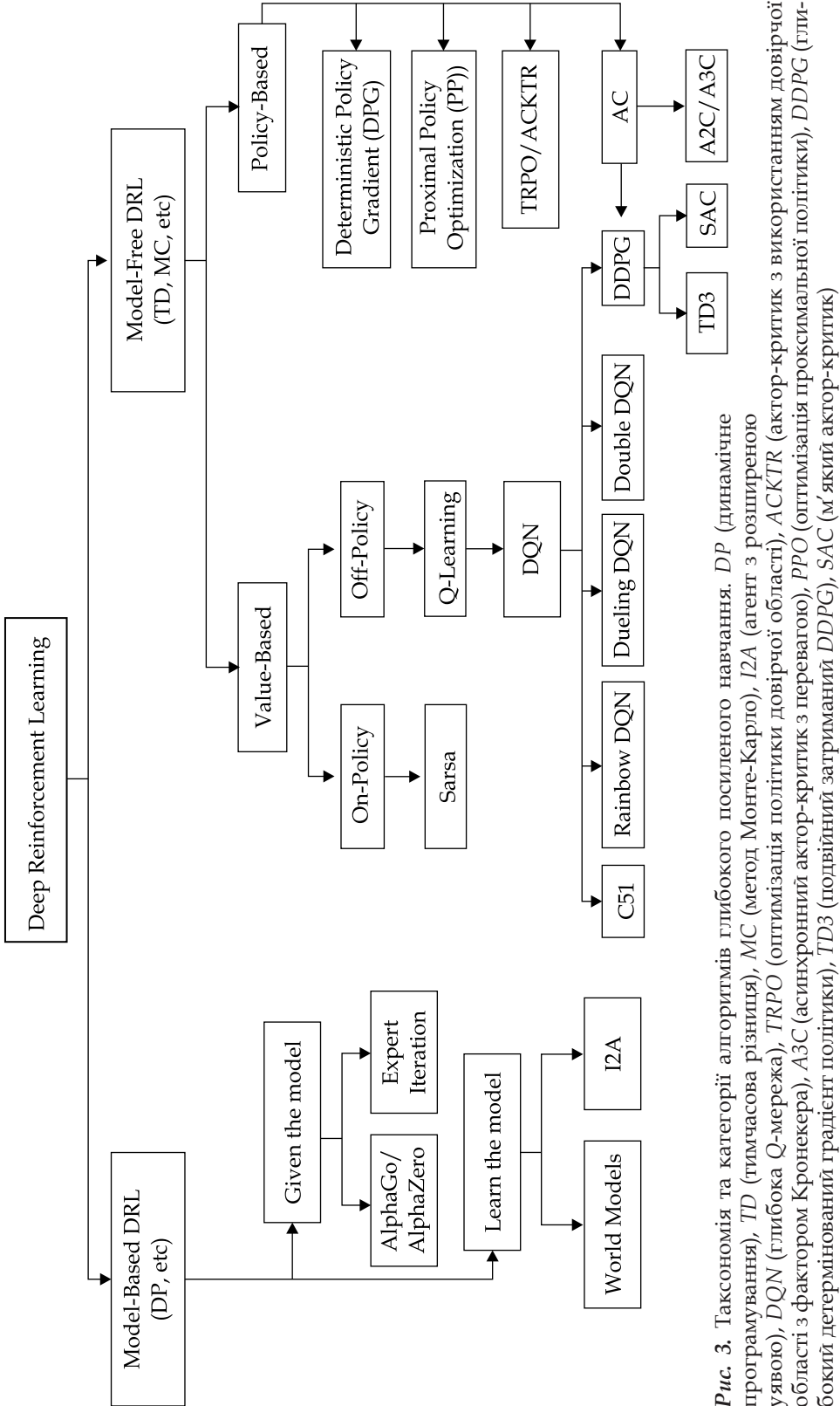


Рис. 3. Таксономія та категорії алгоритмів глибокого посиленого навчання. DP (динамічне програмування), TD (тимчасова різниця), MC (метод Монте-Карло), I2A (агент з розширеною увагою), DQN (глибока Q-мережа), TRPO (оптимізація політики довірчої області), ACKTR (актор-критик з використанням довірчої області з фактором Кронекера), A3C (асинхронний актор-критик з перевагою), PPO (оптимізація проксимальної політики), DDPG (глибокий детермінований градієнт політики), TD3 (подвійний загрузаний DDPG), SAC (м'який актор-критик)

моделей намагається моделювати ці можливості переходу, тоді як алгоритм *RL* без моделей цього не робить. Таким чином, алгоритм *RL* без моделей повністю слідує підходу проб і помилок, тоді як алгоритм *RL* на основі моделей робить це меншою мірою. Це загальна сутність всіх *Model-Free* алгоритмів у *RL*. Але саме вони досі є *state-of-the-art* (рівень розвитку: — найновіший, сучасний, тобто досягає найсучасніших результатів у цій задачі і є ефективнішим).

Навчання без моделі (*Model-Free*), лише на основі даних, що відбувається за допомоги механізму зворотного розповсюдження помилок, практично витіснило інші підходи в багатьох завданнях розпізнавання та оцінювання параметрів. Потребує багато навчальних прикладів. Перевагою є те, що оптимальні дії шукають градієнтним спуском. Недоліки. Основний з них — навчання з підкріпленням погано, а точніше, зовсім не працює з високими розмірностями. *Model-Free* методи можуть зійтися на ключових факторах, ігноруючи інші, але якщо алгоритму відразу не вдасться їх виявити, він швидше за все взагалі не навчиться. Також вони можуть зацикловатися на не оптимальній дії за умови, якщо до нього зійшовся градієнтний спуск, ігноруючи інші фактори. Навіть для незначно відмінних завдань нейромережу доводиться навчати заново.

Model-Based методи [5–6, 9–10] докорінно відрізняються від описаного підходу. *Model-Based* нейромережа тільки передбачає, що буде далі, не пропонуючи жодних дій, тобто є просто моделлю реальності (звідси «*Model-Based*»), а не системою прийняття рішень. *Model-Based* нейромережі легко навчаються внаслідок того, що вони просто проорокують як змінюватиметься середовище, не роблячи при цьому жодних пропозицій щодо оптимальності дій для збільшення нагороди. Нейромережа використовує для навчання всі наявні варіанти без винятку, а не тільки ті, що ведуть до зміни нагороди, як це відбувається у *Model-Free*. Разом з тим нейромережа повинна вивчити реальну динаміку системи, а отже, повинна бути достатньо ємною. Нейромережа у цьому разі є лише моделлю реальності, а оптимальні дії вибираються за допомоги зовнішнього планування завдань. Для здійснення необхідної дії необхідно змоделювати безліч випадкових дій і вибрати найкраще за цих обставин. Це класичний *Model-Based RL*. Однак за великих розмірностей і довгих ланцюжків, число можливих дій виходить занадто великим для перебирання. Тому *Model-Based* методи зазвичай поступаються *Model-Free*, які градієнтним спуском безпосередньо сходяться до найбільш оптимальних дій.

В [11] стверджується, що навчання на основі моделей зазвичай є ефективнішим з точки зору досвіду, оскільки для хорошого набування досвіду потрібно набагато менше затрат, а методи без моделей часто не приносять користі, оскільки час обчислень втрачається марно. За самостійного вивчення моделі середовища агент зіткнеться з деякими неточностями та невизначеністю, які можуть додатково вплинути на його політику та необхідні завдання. Таким чином, було

запропоновано низку підходів для інтеграції методів без моделей та на основі моделей [12]. Також і в [13] висловлено думку, згідно з якою у багатьох завданнях посиленого навчання [4] агент зі штучним інтелектом виграє від наявності хорошого уявлення минулих і дійсних станів, а також хорошій прогнозній моделі майбутнього^{2,3}, переважно потужній прогнозній моделі, такої як рекурентна нейронна мережа *RNN* [5, 6, 9]. Великі *RNN* — це дуже виразні моделі, які можуть вивчати багаті просторові та часові подання даних. Однак багато *Model-Free* методів *RL* змушені використовувати тільки малі нейронні мережі з невеликою кількістю параметрів. Алгоритм *RL* часто стикається з проблемою⁴ присвоєння кредитів, через що традиційним алгоритмам *RL* важко вивчити мільйони ваги великої моделі, тому на практиці використовуються мережі меншого розміру, оскільки вони швидше переходять до хорошої політики під час навчання.

Оскільки вся система БПЛА заснована на взаємодії агента та середовища, досягнення оптимальної стратегії потребує високо розмірних вхідних даних та точного представлення середовища. Середовище, у свою чергу, може бути повністю або частково спостережуване. Це є двома основними парадигмами штучного інтелекту, кожна з яких має свої відмінні характеристики та виклики. Відсутність повної інформації створює труднощі у прийнятті рішень, оскільки агент може не повністю зрозуміти поточну ситуацію або вважатиме складним передбачити майбутні ситуації. Спираючись тільки на звичайні алгоритми *RL*, потрібні високоякісні та багатовимірні вхідні дані для створення точного опису станів БПЛА. Ці вимоги утворюють прогалину через можливості звичайного *RL*. Інтеграція між глибокою нейронною мережею та алгоритмами *RL* створена для заповнення цього пробілу [1].

Ментальна модель світу. Наразі діє і все більше поширюється така концепція посиленого навчання — навчання нейромережі в уяві з використанням моделі світу [13]. У своїй доповіді *Recurrent World*

² Werbos P.J., 1987. *Learning How the World Works: Specifications for Predictive Networks in Robots and Brains. Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, N.Y.

³ David Silver's Lecture on Integrating Learning and Planning [PDF] Silver D., 2017.

⁴ У багатьох завданнях *RL* зворотний зв'язок (позитивна чи негативна винагорода) дається в кінці послідовності кроків. Завдання присвоєння кредитів вирішує проблему з'ясування того, які кроки викликали результуючий зворотний зв'язок — які кроки мають отримати оцінку чи осуд за кінцевий результат. Керування, яке реалізується навчанням з підкріпленням, діє як замкнутий цикл, тоді як винагорода є зворотним зв'язком системи. Ця винагорода затримується, тоді як багато алгоритмів намагаються зменшити цю затримку. Більш того, алгоритми, реалізовані в *RL*, характеризуються послідовною поведінкою прийняття рішень із довгостроковими винагородами залежно від послідовності дій. Концепція, звана проблемою присвоєння кредиту, визначає залежність реалізації *RL* від часу, оскільки деякі дії виявляють свої наслідки через деякий час та безліч проміжних станів, пройдених системою [7].

Models Facilitate Policy Evolution (Рекурентні моделі світу сприяють еволюції політики навчання) на *NIPS 2018* автори [13] запропонували рішення, яке поєднало в собі елементи навчання в уяві, що стали тепер класичними. Це дає змогу, застосовуючи універсальну модель світу, використовувати єдину *Model-Based* нейромережу для вирішення будь-якої кількості завдань. Модель світу — це система штучного інтелекту, яка створює внутрішнє уявлення середовища та використовує його для моделювання майбутніх подій у цьому середовищі [14]. Модель світу є ключем до інтелектуальних систем. Люди застосовують модель світу як симулятор у своєму мозку. Модель виходить за допомоги навчання на великих обсягах сенсомоторних даних у взаємодію з середовищем. Вивчити модель світу можна, використовуючи глибокі генеративні моделі. В основі вибору моделі навколишнього середовища — моделі світу — покладені відомі міркування. Люди розробляють ментальну модель світу, засновану на тому, що вони здатні сприймати своїми обмеженими органами чуття — це лише образ навколишнього світу, доступний нашому сприйняттю та опису з позицій вирішення поставленого завдання. Рішення та дії, які ми приймаємо і виконуємо, ґрунтуються на цій внутрішній моделі. Ніхто не уявляє собі весь світ у всьому його різноманітті. Вибираємо лише концепції та відношення між ними і використовуємо їх для подання реальної системи [13].

В [13] звернули увагу на те, щоб упоратися з величезною кількістю інформації, яка проходить через наше повсякденне життя. Мозок людини вивчає абстрактне подання як просторових так і тимчасових аспектів цієї інформації. Ми здатні спостерігати сцену та запам'ятовувати її абстрактний опис. Наявні дані також свідчать що те, що саме ми сприймаємо в будь-який момент, визначається пророкуванням нашим мозком такого майбутнього, яке засноване на нашій внутрішній моделі.

Один із способів зрозуміти прогностичну модель, яка існує всередині нашого мозку, полягає в тому, що вона може існувати не просто у пророкуванні майбутнього в цілому, а у пророкуванні майбутніх сенсорних даних з урахуванням наших поточних рухових дій. Ми здатні інстинктивно діяти відповідно до цієї прогностичної моделі і діяти швидко та рефлексивно, коли стикаємося з небезпекою, без потреби свідомо розгортати можливі сценарії майбутнього для формування плану відповідно до своїх передбачень майбутнього [13].

Математична формалізація завдань БПЛА за посиленого навчання

Multi-Agent RL (MARL). Багато мультиагентних систем природним чином містять кілька агентів, що приймають рішення, і які взаємодіють одночасно. Завдяки досягненням одноагентного глибокого *RL*, посилене багатоагентне навчання перейшло до багатоагентного

посиленого глибокого навчання (*MADRL*) з реальною складністю [15]. Посилене навчання формалізовано у Марківських процесах прийняття рішень (*MDP*) як основу для навчання з одним агентом, і у разі багатоагентного навчання – Марківську гру [15]. Традиційне завдання посиленого навчання [16] пов'язане з навчанням політики управління, що оптимізує числову продуктивність шляхом ухвалення рішень поетапно. Адаптивний агент, що приймає рішення, взаємодіє з середовищем невідомої динаміки методом проб і помилок з метою поліпшити свою роботу, що визначається імовірнісною функцією переходу. Стандартним формулюванням для такого послідовного ухвалення рішень є *MDP*. Взаємодія адаптивного агента із середовищем визначається імовірнісною функцією переходу станів. У цьому соліпсичному⁵ поданні вторинні агенти можуть бути лише частиною середовища і, отже, є фіксованими у своїй поведінці. Структура Марківських ігор дозволила розширити це уявлення, долучивши до нього кілька адаптивних агентів із взаємодіючими чи конкуруючими цілями [17]. Тому, коли послідовне прийняття рішень поширюється на кількох агентів, як структуру зазвичай застосовують Марківські ігри⁶. Подібно до проблеми з одним агентом, мета кожного агента полягає в тому, щоб змінити свою політику таким чином, щоб оптимізувати отримані винагороди в довгостроковій перспективі. В одноагентному формалізмі агент є єдиним екземпляром, що приймає рішення, який впливає на стан середовища. Переходи станів можна чітко приписати агенту, тоді як усе, що знаходиться за межами поля впливу агента, сприймається як частина базової динаміки системи. Незважаючи на те, що середовище може бути стохастичним, проблема навчання залишається стаціонарною. Навпаки, одна з фундаментальних проблем у багатоагентній галузі полягає в тому, що агенти оновлюють свої політики під час процесу навчання одночасно, тому середовище здається нестационарним з точки зору одного агента. Отже, припущення Маркова про *MDP* більше не виконується, і агенти мають справу з проблемою цілі, яка рухається.

Також є зауваження щодо середовища. Відсутність повної інформації про середовище створює труднощі у прийнятті рішень, оскільки агент може не зрозуміти поточну ситуацію або вважатиме складним передбачити майбутні ситуації. Частковість спостереження може бути наслідком різних причин, таких як неточні датчики, обмежений діапазон датчиків або складність середовища.

⁵ **Соліпсизм** (лат. *solus ipse* – тільки сам) – філософська доктрина і позиція, крізь призму якої визнається власна індивідуальна свідомість як єдина і безперечна реальність навколишнього світу. Розглядається як крайня форма суб'єктивного ідеалізму. Трактуються як заперечення реальності всього, крім власної свідомості. «Мій розум – єдина річ, яка існує.»

⁶ Марківські ігри також відомі як стохастичні ігри (Shapley 1953). Термін «гра Маркова» використовується, щоб провести чітку різницю між детермінованими та стохастичними Марківськими іграми.

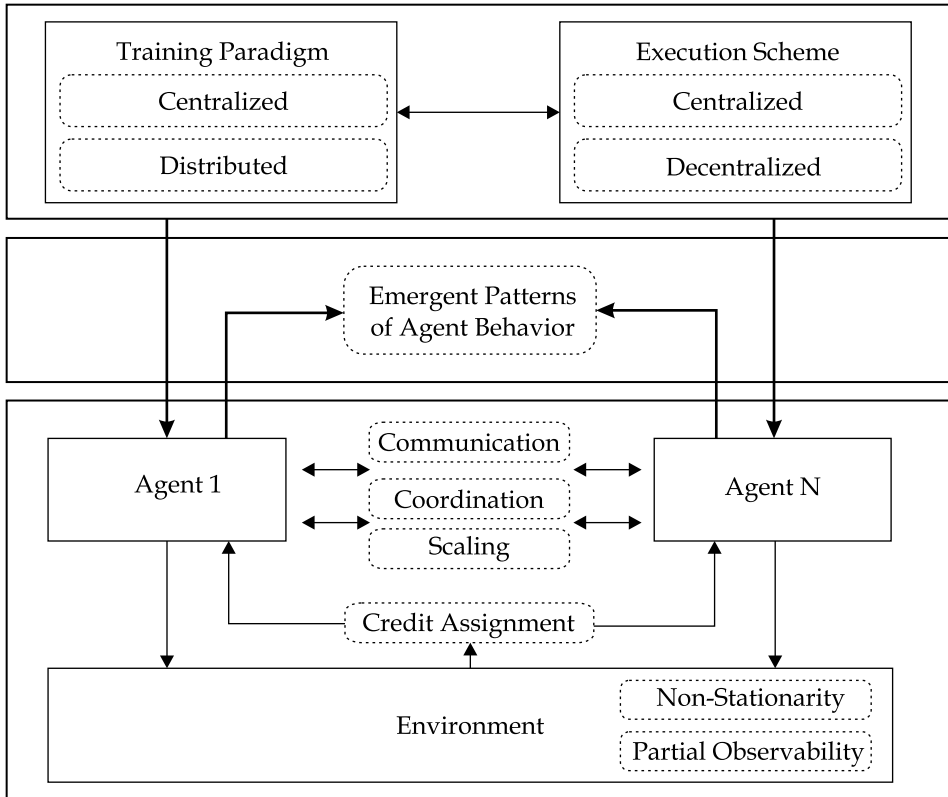


Рис. 4. Схематична структура матеріалу, викладеного в роботі [15]. Парадигми навчання (*Training Paradigms*) поведінки агентів у багатоагентному середовищі: централізована (*Centralized*) та розподілена (*Distributed*). Схема виконання (*Execution scheme*): централізована (*Centralized*) та децентралізована (*Decentralized*), коли агенти визначають дії відповідно до своєї індивідуальної політики. Емерджентні моделі поведінки агента (*Emergent Patterns of Agent Behavior*), обумовлені взаємодією сутностей, можуть викликати системну поведінку, яка відноситься до несподіваних та нелегко передбачуваних результатів. Таким збудником можуть бути структури винагороди, мови та соціального контексту.

У частково спостережуваному середовищі агенти повинні використовувати стратегії для вирішення проблеми частковості спостереження, включаючи оцінку стану, імовірнісні міркування та використання пам'яті. Формальні структури, такі як частково спостережувані процеси прийняття рішень Маркова (*Partially Observable Markov Decision Processes, POMDPs*), зазвичай використовуються для моделювання та вирішення проблем у таких середовищах, що дозволяє агентам розробляти складні стратегії, які збалансовують розвідку та розроблення [18].

В [15] подано критичний огляд поточних розробок в галузі MADRL з аналізом структури схем навчання для декількох агентів, а саме закономірності поведінки агентів у кооперативних, конкурентних та змішаних сценаріях. Систематизовано проблеми, які виникають виключно в багатоагентній сфері глибокого RL (рис. 4.),

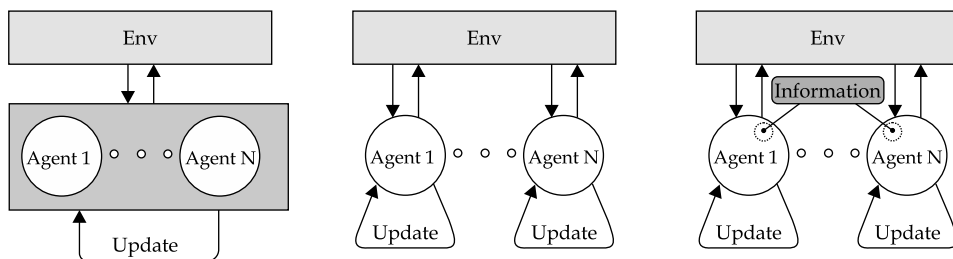


Рис. 5. Схеми навчання у багатоагентному середовищі. Зліва направо: *CTCE* підтримує загальну політику всім агентам; *DTDE* – кожен агент оновлює власну індивідуальну політику; *CTDE* дає змогу агентам обмінюватися під час навчання додатковою інформацією, яка потім відкидається під час виконання

та розглянуто сучасні рішення, які були запропоновані для розв'язання проблем. Обговорено досягнення, визначено тенденції та позначено можливі напрями робіт у цій галузі досліджень.

Парадигма навчання у багатоагентному середовищі. За централізованого навчання агентів *CTCE* (див. рис. 4) політики оновлюються з урахуванням взаємного обміну інформацією під час навчання. Навчання у розподілений спосіб *DTDE* відбувається у разі, коли кожен агент виконує оновлення самостійно та навчається індивідуальній політиці без використання зовнішньої інформації. Централізоване виконання передбачає, що всім агентам обчислюються спільні дії. Навпаки, агенти визначають дії відповідно до своєї індивідуальної політики для децентралізованого виконання *CTDE*.

Поточні проблеми *MADRL* охоплюють нестаціонарність середовища, що виникає через спільне адаптування агентів; навчання комунікації; необхідність узгодженої координації дій; проблему присвоєння кредиту; можливість масштабування до довільної кількості агентів, які приймають рішення; та неМарківські середовища через частковість їхніх спостережень (*Partial Observability*).

За централізованого навчання агентів *CTCE* (рис. 4) політики оновлюються з урахуванням взаємного обміну інформацією під час навчання. Навчання розподіленим чином *DTDE* відбувається у разі, коли кожен агент виконує оновлення самостійно та навчається індивідуальній політиці без використання зовнішньої інформації. Централізоване виконання передбачає, що агенти керуються з централізованого блоку, який обчислює спільні дії всім агентам. Навпаки, агенти визначають дії відповідно до своєї індивідуальної політики для децентралізованого виконання *CTDE*. Огляд схем навчання у багатоагентному середовищі зображено на рис. 5.

Парадигма централізованого навчання із децентралізованим виконанням⁷ в одному і тому ж середовищі (*Centralized Training with*

⁷ MathWorks. MATLAB Answers. Centralized vs Decentralized Training for Multi Agent Reinforcement Learning. Commented: Lin on 22 Jul 2024. <https://uk.mathworks.com/>

Decentralized Execution; CTDE). *CTDE* — це домінуючий підхід як для кооперативного, так і для змішаного середовища, завдяки своїй здатності ефективно навчати децентралізованої політики, є комбінацією незалежного і централізованого *MARL* [15]. У той час як у змішаних середовищах повна автономія агентів може бути бажаним результатом, кооперативне середовище дозволяє агентам обмінюватися інформацією для полегшення координації.

Повністю кооперативна обстановка означає, що агенти отримують однакову винагороду за переходи станів. У такій рівноправній обстановці винагороди агенти мотивовані до співпраці та намагаються уникнути невдачі окремого агента, щоб максимізувати продуктивність команди. У більш загальному плані йдеться про кооперативні умови, коли агенти заохочуються до співпраці, але не мають рівноправної нагороди.

Встановлено, що розглянуті алгоритми багатоагентного навчання *RL* показали, що основна складність полягає в узагальненні на безперервні дії та простір станів, а також масштабуванні на велику кількість агентів [15]. Навчання кількох агентів тривалий час було обчислюваним завданням. Оскільки складність у просторі станів та дій зростає експоненційно з кількістю агентів, навіть сучасні підходи глибокого навчання можуть досягати своїх меж. Визначено такі проблеми координації незалежних агентів, що навчаються, які виникають у повністю кооперативних Марківських іграх, зокрема нестаціонарність, стохастичність та тіньова рівновага [15].

Посилене навчання для керування БПЛА (*Reinforcement Learning for UAV Control*)

Керування БПЛА, окрім завдань навігації та планування шляху, як і раніше є широкою концепцією, яка охоплює безліч різних завдань починаючи від керування орієнтацією одного БПЛА до керування каналами зв'язку між декількома БПЛА. Також глибоке посилене навчання сприяє вирішенню нових складних завдань у цієї сфері, таких як безпечне керування гібридним БПЛА та керування роєм дронів (*UAV Swarm control or swarm of UAVs*). Ця сфера керування до виконання певних завдань з мінімальними ресурсами (мінімальними переміщеннями, часом, енергією тощо) дуже складна. Складність полягає у самій її природі: складна, змішана динаміка, взаємодія з експлуатаційними та екологічними умовами, що змінюються, потребою бути енергетично ефективною та її вразливістю до перешкод, шумів датчиків і не модельованої динаміки. Поза тим, існує багато завдань з помітними варіаціями, в яких потрібне стабільне та точне керування БПЛА. На рис. 6 надано таксономію різних завдань, що

matlabcentral/answers/2002007-centralized-vs-decentralized-training-for-multi-agent-reinforcement-learning

виникають в разі керування БПЛА, і вказано запропоновані *Model-Free DRL* алгоритми для кожного з них — рис. 1 [1].

У цьому контексті *DRL* було протестовано в режимі реального часу із застосуванням різноманітних *Model-Free* алгоритмів на основі базових політик, таких як, *DDPG*⁸, *PPO*, *TRPO*⁹ тощо (рис. 3). Кожен алгоритм показав свої переваги та недоліки, що робить їх легкими для одних застосувань і незручними для інших [1].

Основною темою були реальні застосування БПЛА, які використовуються для утримання дронів у бажаному положенні та орієнтації для виконання певного завдання. Йдеться про формування зовнішнього контуру, що утворюється роєм дронів, які можуть підтримувати комунікації між собою. Враховуючи складність завдання, високу швидкість переміщення дронів та точність, яка має бути гарантована для керування роєм, застосовувалося посилене глибоке навчання. Було застосовано декілька підходів, та дослідження у цій галузі ще не завершено [1]. Наприклад, політика керування глибокою згортковою нейронною мережею (*DCNN*¹⁰ *P-policy*) дала змогу керування п'ятьма БПЛА [19]. Завдання роя полягали в тому, щоб автономно вишикуватися у певну форму та захистити територію від нападників. Мотивоване необхідністю кооперативної поведінки дронів, багатоагентне посилене навчання, *MARL* застосовувалося для вирішення проблеми відсутності інформації про всі змінні стани для всіх БПЛА в рої [15]. У цій проблемі керування в алгоритмі *DCNNP* застосовувалися централізовані підходи до навчання та централізованого виконання. Тільки одному БПЛА, названому агентом, було дозволено вивчати модель, планувати поведінку для всіх інших БПЛА та спілкуватися з ними про їхні завдання. Поведінка керуван-

⁸ *DDPG* (*Deep Deterministic Policy Gradients*) — це алгоритм, що поєднує у собі ідеї з двох сфер: *DPG* (*Deterministic Policy Gradients*) та *DQN* (*Deep Q-Network*), алгоритм посиленого навчання, який поєднує в собі елементи *Q*-навчання (*Q*-функція оцінки довгострокової нагороди) та методу актор-критик (*Actor-Critic*). У *DDPG* використовується нейронна мережа для параметризації політики (актор) та критика (критика) для оцінки *Q*-функції стану-дії. Підходить для завдань з безперервною дією, став основою, наприклад, для керування роботами та автономного водіння.

PPO (*Proximal Policy Optimization*) — це алгоритм посиленого навчання, який використовує методи оптимізації політики, щоб навчати агентів приймати дії у середовищі. У ньому використовується актор-критик, де актор (політика) оновлює поліпшення вибору дій, а критик (оцінювач) оцінює, наскільки добре дії агента відповідають очікуванім нагородам. *PPO* є популярним вибором завдяки своїй стабільності та хорошій продуктивності в різних середовищах — <https://habr.com/ru/companies/otus/articles/771412/>

⁹ *TRPO* (*Trust Region Policy Optimization*, Оптимізація політики довірчої області) — це алгоритм посиленого навчання, який ґрунтується на методі максимізації функції корисності з обмеженням (*trust region*). Відомий своєю здатністю забезпечувати стабільне навчання та гарантувати покращення політики.

¹⁰ *DCNN* — Глибока згорткова нейронна мережа (*CNN*) з кількома шарами, яка зазвичай використовується для аналізу та розпізнавання зображень — <https://builtin.com/articles/dcnn>

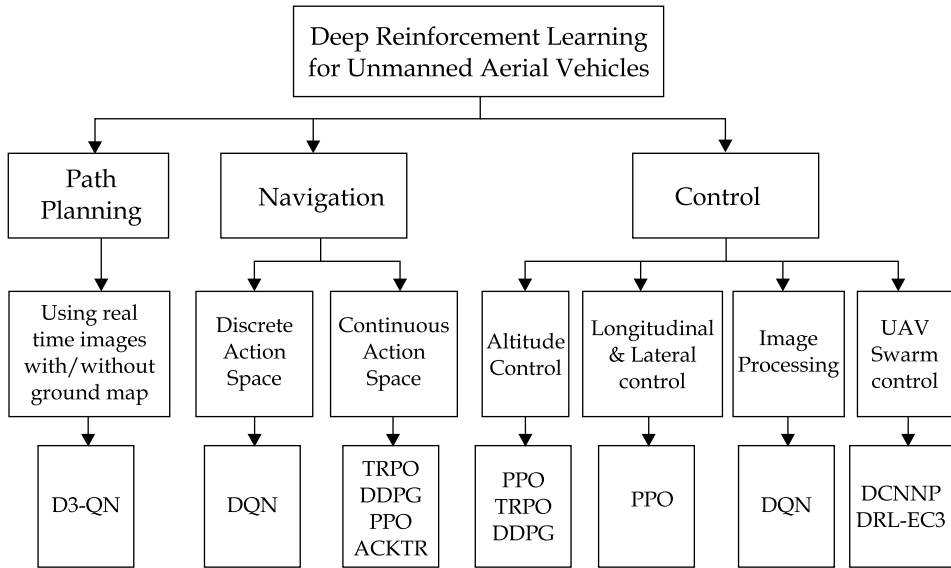


Рис. 6. Таксономія алгоритмів глибокого посиленого навчання у завданнях БПЛА, що наведені у третьому зверху ряду: *Using real time images with / without ground map* – використання зображень у реальному часі з картою місцевості / без неї, *Discrete or Continuous Action Space* – дискретний або безперервний простір дій, *Altitude control* – контроль висоти, *Longitudinal & Lateral control* – поздовжній і бічний контроль, *Image Processing* – обробка зображень; *UAV Swarm control* – керування роєм БПЛА

Algorithm	Agent Type	Policy	Policy Type	MC or TD	Action Space	State Space
State action reward state action (SARSA) SARSA Lambda	Value-based	On-policy	Pseudo-deterministic ($\epsilon - greedy$)	TD	Discrete only	Discrete only
Deep Q Network (DQN) Double DQN Noisy DQN Prioritized Replay DQN Dueling DQN Categorical DQN Disturbed DQN (C51)	Value-based	Off-policy	Pseudo-deterministic ($\epsilon - greedy$)		Discrete only	Discrete or Continuous
Normalized Advantage Functions (NAF) = Continuous DQN	Value-based				Continuous	Continuous
REINFORCE (Vanilla policy gradient)	Policy-based	On-policy	Stochastic	MC		
Policy Gradient	Policy-based		Stochastic			
TRPO	Actor-critic	On-policy	Stochastic		Discrete or Continuous	Discrete or Continuous
PPO	Actor-critic	On-policy	Stochastic		Discrete or Continuous	Discrete or Continuous
A2C/A3C	Actor-critic	On-policy	Stochastic	TD	Discrete or Continuous	Discrete or Continuous
DDPG	Actor-critic	Off-policy	Deterministic		Continuous	Discrete or Continuous
TD3	Actor-critic				Continuous	Discrete or Continuous
SAC	Actor-critic	Off-policy			Continuous	Discrete or Continuous
ACER	Actor-critic				Discrete	Discrete or Continuous
ACKTR	Actor-critic				Discrete or Continuous	Discrete or Continuous

Рис. 7. Зведена таблиця безмодельних алгоритмів RL у завданнях БПЛА

ня *DCNNP* порівнювалася з поведінкою імовірнісної політики, поданої випадковим переміщенням БПЛА до досягнення місії, та ідеальної / периметральної політики, що досягається рівномірним переміщенням БПЛА в одному напрямку. Результати показали, що *DCNNP* значно перевершив випадкову політику обох задач; спостерігалися невеликі відмінності у продуктивності порівняно з підходом периметра для оборони об'єкта, але зі зниженою продуктивністю порівняно з ідеальною політикою, особливо коли кількість БПЛА невелика у задачі спостереження.

Інше дослідження, пов'язане з керуванням роєм, було запропоновано в [20]. БПЛА можуть застосовуватися як повітряні базові станції для покращення покриття та продуктивності комунікаційних мереж, зокрема як екстрений зв'язок та доступ до мережі для віддалених районів. Вони можуть встановлювати канали зв'язку для наземних користувачів у разі доставки пакетів. Однак БПЛА мають обмежені радіус дії зв'язку та енергетичні ресурси. Зокрема, для великого регіону вони не можуть постійно покривати всю територію або продовжувати політ протягом тривалого часу. Таким чином, складно керувати групою БПЛА для досягнення певного покриття зв'язку в довгостроковій перспективі, зберігаючи при цьому їх зв'язок та мінімізуючи споживання енергії. З цією метою було запропоновано енергоефективне керування БПЛА на основі *DRL* за новими методом і алгоритмом, що створено, який здійснює підключення до зони покриття мережі (*DRL Based Energy Efficient Control for Coverage Connectivity (DRL – EC3)*) [20]. Він застосовує метод глибокого детермінованого градієнта політики (*Deep Deterministic Policy Gradient (DDPG)*) з деякими модифікаціями та політикою актор-критик, що використовує двошарову глибоку нейронну мережу для кожного з них. Цей алгоритм реалізовано на *TensorFlow*¹¹ з 400 нейронами в першому шарі та 300 у другому в кожній мережі. Основною метою цього дослідження була економія енергоспоживання роєвих БПЛА під час роботи та зв'язку без зниження точності виконання завдань. Винагорода, що використовувалася в навчанні, ґрунтувалася на покритті, індексі справедливості покриття, споживанні енергії та безперервному зв'язку для всіх БПЛА. Це дослідження показало, що запропонована модель керування *DRL-EC3* перевершує як відомі випадкові, так і жадібні (*Greedy algorithm*¹²) політики. Понад те, дослідження

¹¹ *TensorFlow* – відкрита програмна бібліотека для машинного навчання, розроблена *Google* для вирішення завдань побудови та тренування нейронної мережі, дозволяє швидко створювати нейромережі.

¹² Жадібні алгоритми – це клас алгоритмів, які роблять локально оптимальний вибір на кожному кроці з надією знайти глобально оптимальне рішення, тобто алгоритм є простим і скоріше евристичним, який приймає найкраще рішення, виходячи з наявних на кожному етапі даних. Легкий в реалізації і часто дуже ефективний за часом виконання, але багато задач не можуть бути розв'язані за його допомоги – <https://www.geeksforgeeks.org/greedy-algorithms/>

показало, що зі збільшенням кількості БПЛА новий алгоритм керування виявився успішним [20].

Ще одне дослідження вирішує проблему роєм (кластером) БПЛА для виконання планування завдань у реальному часі [21]. Останнім часом рої БПЛА широко застосовуються завдяки їхній високій гнучкості, широкому охопленню зони покриття зв'язку та надійній ефективності передачі інформації. Для досягнення співпраці БПЛА в разі виконання кількох завдань кластеру запропоновано алгоритм планування завдань на основі посиленого глибокого навчання, який дає змогу БПЛА автоматично і динамічно коригувати свою стратегію виконання завдань, використовуючи власні обчислення ефективності їх виконання. Оскільки БПЛА необхідно виконувати завдання, працюючи в динамічному середовищі без централізованого керування, йому необхідно навчатися завданням відповідно до даних теж у реальному часі. Посилене навчання здатне навчатися та приймати рішення в реальному часі на основі інформації у середовищі, що є дочерним та здійсненим методом планування завдань кластерів БПЛА.

З цієї точки зору, розглядалось посилене навчання, яке вирішує проблему розподілу каналів, що існує в плануванні завдань кластерів БПЛА. Цей підхід полягає у виділенні каналів для підтримки зв'язку для завдань керування БПЛА, таких як оцінювання стану та відстеження траєкторії. Таке завдання передбачає величезну кількість даних, отриманих в результаті навчання, тренування та обробки цих даних у реальному часі. Враховуючи це, Q -навчання було виключене з варіантів через його складність оброблення великих наборів даних. Натомість використовувалася глибока мережа посиленого навчання через її здатність подавати дані в нижчій розмірності та отримувати значення Q більш ефективно. Архітектуру керування можна описати як багатоагентну з децентралізованим керуванням, при цьому не потрібні ніякі моделі для системи, а поведінка агентів розглядається з погляду кожного агента як спостереження за станом навколишнього середовища. Алгоритм глибокого посиленого навчання розглядає чотири випадки стану каналу. По-перше, якщо канал зайнятий, система чекає, доки дані не будуть надіслані знову. По-друге, якщо передача не вдалася, дані мають бути надіслані повторно. По-третє, якщо передача успішна, наступні завдання можуть бути виконані. По-четверте, якщо немає завдання, ліниво чекайте. Оскільки головна мета полягає в якості передачі, винагорода заснована на максимальній кількості переданих даних та меншій затримці передачі. Запропонована стратегія успішно забезпечує точну та швидку робочу поведінку для зв'язку між БПЛА [1].

Порівняльна оцінка застосованих алгоритмів RL у вирішенні завдань БПЛА

Складна проблема реального застосування дронів потребує вирішення невизначеностей та адаптації до динамічних невідомих середовищ. *DRL* пропонує гнучку основу для вирішення цих

проблем без застосування моделі (*Model-Free*) та може навчатися на досвіді й передбачати правильне рішення для майбутніх спостережень. Із двох гілок алгоритмів *Model-Based RL* намагаються вибрати оптимальну політику на основі вивченої моделі середовища, тоді як у безмоделних алгоритмах оптимальна політика вибирається на основі помилок, з якими стикається агент (див. рис. 7). Стратегія керування агентом (функція прийняття рішень), яка є відображенням ситуацій у дії, називається політикою, й існує її два типи, тобто політика відповідності та політика поза політикою. Методи, що базуються на політиці (*SARSA*, *PPO*, *TRPO*), намагаються оцінити або покращити політику, яка використовується для прийняття рішень, тоді як методи, що не базуються на політиці (*Q*-навчання, *DQN*, *DDPG*), оцінюють або покращують політику, відмінну від тієї, що використовується для генерації даних [1].

Усі алгоритми досягають певного стаціонарного стану, однак, *TRPO* та *DDPG* дають екстремальні коливання як по осі крену, так і по осі ризику, що може спричинити нестабільність під час польоту. Однак, загалом кажучи, з точки зору похибки, *PPO* показав себе як більш точний контролер положення завдяки його методу градієнта політики першого порядку, який особливо популярний завдяки чудовим характеристикам у тестах та простоті реалізації. Для навчання завдань керування польотом дрона проведено експерименти з використанням *PPO*, *DDPG* та *TRPO* для зависання, посадки, випадкових точок маршруту та супроводження цілі [1]. Виявлено, що *PPO* загалом був найбільш послідовним алгоритмом для посадки, за ним йшов *TRPO*. Для зависання *DDPG* забезпечував дуже плавний політ порівняно з іншими алгоритмами. У завданні випадкової навігації за точкою маршруту *PPO* забезпечував найплавнішу навігацію порівняно з іншими алгоритмами. Для завдання слідування за ціллю *DDPG* забезпечує кращі середні винагороди та найплавнішу навігацію за точкою маршруту, за ним йшли *PPO* та *TRPO*.

Для великих складних невідомих середовищ деякі ідеї спрямовано на застосування методів *DRL* для просування до своїх завдань, покладаючись тільки на сенсорні дані і сигнали *GPS*. Не має необхідності планувати невизначену *3D*-модель для невідомого середовища та планувати шляхи для проходження транспортних засобів, так стверджують автори, — БПЛА зможе розумно переміщатися з довільних місць відправлення у довільні цільові позиції [22]. В цій роботі, пов'язаній з доставою вантажів за допомогою БПЛА, що нині актуально, моделюється автономна навігація БПЛА у масштабному невідомому складному середовищі як безперервне завдання керування у дискретному часі, та вирішується ця задача за допомоги *DRL*. Без планування шляху або побудови карти пропозицій запропонований метод дозволяє БПЛА переміщатися до пунктів призначення, використовуючи тільки сенсорну інформацію локального середовища та сигнал *GPS*. Прийнято, що завдання навігації є

Марківським процесом, який частково спостерігається, прийняттям рішень (*POMDP*), а також зауважуємо, що рекурентний детермінований алгоритм градієнта політики менш ефективний. Отже, створено швидший алгоритм навчання політиці для *POMDP* з урахуванням архітектури актор-критик. Для перевірки змодельовано п'ять віртуальних середовищ та віртуальний БПЛА, який летить на фіксованій висоті з постійною швидкістю. Пізнання локального середовища досягається шляхом вимірювання відстаней від БПЛА до перешкод у кількох напрямках. Результати моделювання показали ефективність такого підходу.

До слова, рішення для уникнення об'єктів може бути досягнуто лише шляхом зміни висоти БПЛА без переміщення праворуч або ліворуч. Застосований метод *DRL* вивчає значення Q і оптимальну політику для уникнення об'єктів. Можна також застосовувати інші політики та методи, такі як асинхронний актор-критик з перевагою (*A3C*), глибокий детермінований градієнт політики (*DDPG*) та архітектуру дуельної мережі для мереж Q з подвійною глибиною (*D3QN*)¹³. Варто лише змінити функцію втрат [1], [23].

Модний алгоритм *DQN* страждає від суттєвого переоцінювання дії-стану в задачах посиленого навчання, таких як ігри в домені *Atari 2600* та домені планування шляху. Щоб зменшити переоцінювання значень дій під час навчання, представлено нову комбінацію алгоритму подвійного Q -навчання та дуельного *DQN*, а також розроблено алгоритм під назвою «Варіант подвійного дуельного *DQN*» (*V-D D3QN*) [23]. Автори зосередились на ідеї, що лежить в основі алгоритму *V-D D3QN*, та запропонували можливу ідею застосування двох дуельних *DQN*-мереж для зменшення переоцінки значень дій під час навчання. Конкретний підхід полягає у випадковому виборі однієї дуельної *DQN*-мережі на кожному кроці часу для оновлення її параметрів, використовуючи решту дуельної *DQN*-мережі для визначення цілей оновлення. Потім здійснили експерименти в налаштованому віртуальному середовищі-мережі. Результати експериментів показали, що запропонований алгоритм не тільки зменшує переоцінки ефективніше, ніж алгоритм *Double DQN (DDQN)*, але й призводить до набагато кращої продуктивності в області планування маршруту з великою здатністю до узагальнення нових та швидкозмінних середовищ [23].

Що стосується загального поліпшення систем, слід взяти до уваги, що умови навколишнього середовища, такі як швидкість вітру, дощі та пил, вносять невизначеність в результати. Тому їх слід розглядати як системні перешкоди і потім професійно обробляти [1].

¹³ Архітектура дуелі складається з двох потоків, які представляють функції цінності та переваги, при цьому розділяючи загальний модуль навчання згорткових ознак. Два потоки об'єднуються за допомогою спеціального шару, що агрегує, для отримання оцінки функції цінності стану-дії Q [23].

Навігація та керування БПЛА стали новою сферою через широкий спектр їх застосувань. БПЛА використовують в різних аспектах життя, таких як цивільні завдання, військові місії, відстеження об'єктів та пошуково-рятувальні операції [24]. Кожна з цих програм вимагає точної навігації, щоб уникати зіткнень під час вибору оптимального маршруту для досягнення мети. Навігація в невідомому середовищі з перешкодами, що змінюються, є однією з проблем керування БПЛА [1]. Для вирішення цієї проблеми було розроблено та оцінено різні алгоритми керування та методи штучного інтелекту.

DRL є одним з перспективних рішень навігації БПЛА. Тоді як іншим алгоритмам машинного навчання потрібні марковані дані для початку процесу навчання, що недоступно у випадку з БПЛА, оскільки вони мають справу з новими середовищами під час кожної місії [22]. Використання *RL* у навігації БПЛА переважно спирається на функцію винагороди, що визначається діями БПЛА. Одним з алгоритмів, що використовуються в разі *RL*-навчання, є мережа *Q*, яку можна об'єднати з різними нейронними мережами для визначення оптимального значення дії. Алгоритм спирається на безперервне оновлення станів БПЛА на основі даних, отриманих від бортових датчиків, а потім на визначення оптимальної дії, яку необхідно зробити, і відповідного значення винагороди.

Загалом автономна навігація у невідомому чи невизначеному середовищі є одним із складних завдань для БПЛА. Щоб вирішити цю проблему, необхідно мати методи керування високого рівня, які можуть здійснювати навчання і адаптуватися до умов, що змінюються. Одним із найперспективніших фреймворків для досягнення такої мети є посилене навчання. Описано новий *Model-Based* алгоритм посиленого навчання з моделлю *TEXPLORE*¹⁴, розроблений як метод керування високого рівня для автономної навігації БПЛА [24].

TEXPLORE [25, 26] – ефективний за вибіркою алгоритм з *Model-Based RL* посиленого навчання в реальному часі на основі динамічної моделі середовища, яку можна адаптувати стосовно її функційності, що поєднує інтеграцію планування та навчання. Інтеграція цих двох етапів більш відома як *RL* навчання з моделлю. Нагадаємо, традиційне завдання посиленого навчання [16] пов'язане з навчанням політики керування, що оптимізує числову продуктивність шляхом прийняття рішень поетапно. Поєднання планування та навчання провадиться для оптимізації процесу послідовного прийняття рішень Маркова, *MDP*. Тобто, *TEXPLORE* – це метод навчання агентів виконанню послідовних завдань прийняття рішень за допомоги взаємодії з довкіллям. Підходи до навчання динамічних моделей охоплюють вирішення таких проблем, як робота зі стохастичністю, не-

¹⁴ *TEXPLORE* було випущено публічно як пакет *ROS* за адресою – <http://www.ros.org/wiki/rl-texplore-ros-pkg>

визначеністю, частковістю спостереження та тимчасовою абстракцією [25, 26].

TEXPLORE застосовує факторизовану¹⁵ модель, роблячи окремий прогноз про наступне значення кожної функції стану та винагороди. Він будує дерева рішень для моделювання кожної функції, що дозволяє узагальнювати ефекти дій за станами та динаміку на невидимі стани. Алгоритм *RL* вивчає модель переходу станів та динаміки винагороди, а потім використовує її для планування політики, що дає змогу йому навчатися за меншу кількість дій, ніж у багатьох підходах без моделі. Він також використовує архітектуру реального часу, яка виконує навчання та планування моделі в паралельних потоках, тому агент може діяти в реальному часі. Агент націлюється на дослідження станів, які є багатообіцяльними для остаточної політики, так і невизначеними в моделі. З плануванням на основі вибірок та паралельною архітектурою *TEXPLORE* може безперервно вибирати дії у реальному часі, коли це необхідно. Високої ефективності вибірки він досягає як внаслідок використання властивостей узагальнення дерев рішень при побудові своєї моделі *MDP* та застосування випадкових лісів для обмеження дослідження станами, які є перспективними для швидкого навчання гарної (але не обов'язково оптимальної) політики, замість більш вичерпного аналізу для гарантії оптимальності. Оскільки алгоритм повинен виконувати обмежене дослідження, то дослідження має бути ефективним і націленим на дії станів, які можуть бути перспективними для отримання винагороди.

Ключові ідеї *TEXPLORE*: 1) вивчення кількох доменних моделей, що узагальнюють ефекти дій у різних станах, та цільове дослідження невизначених і перспективних станів; 2) об'єднання пошуку по дереву Монте-Карло та паралельної архітектури для безперервного виконання дій у режимі реального часу.

Як стверджують автори [24], це перший випадок адаптації моделі *TEXPLORE* для автономної навігації квадаторного БПЛА (*Quadrotor UAVs*), який є одним із привабливих типів малих безпілотних літальних апаратів через дуже просту механічну конструкцію та принцип руху. Розроблений алгоритм ретельно протестували з квадрокоптером у середовищі *ROS-Gazebo*. Експериментальні результати показали, що метод здатний вивчити ефективну траєкторію за кілька ітерацій та виконувати дії у реальному часі. Більше того, підхід значно перевершує метод на основі *Q*-навчання.

Відповідно, глибоке посилене навчання, *DRL* призвело до справжнього зсуву в керуванні БПЛА. Ніхто не може заперечувати, що вища точність подання інформації мотивує дослідників докладати додаткових зусиль у використанні цих алгоритмів, які потребують біль-

¹⁵ Факторизація або розкладання на множники — це декомпозиція об'єкта у добуток інших об'єктів.

ше обчислювального часу, ніж класичне керування. Не кажучи вже про те, що *DRL* дає змогу виконувати нові завдання, які раніше були майже неможливими. Той факт, що розроблення нових методів і стратегій *DRL* все ще є постійним процесом з багатьма можливостями, є багатообіцяльною новиною для досліджень у сфері керування БПЛА [1].

В останні роки досягнуто значного прогресу у вирішенні складних проблем у різних галузях за допомогою посиленого глибокого навчання. Відтворення наявних робіт та точна оцінка покращень, що пропонуються новими методами, є життєво важливими для підтримки цього прогресу. На жаль, відтворення результатів для сучасних методів посиленого глибокого навчання не часто буває простим. У роботі [28] надано такі пропозиції щодо того, як забезпечити постійний прогрес у цій галузі, мінімізуючи марні зусилля, які забезпечують через результати, які є невідтворюваними та легко неpravильно інтерпретуються.

Висновки

З погляду класифікації підходів до створення мультиагентних систем із застосуванням глибокого посиленого навчання, етапів польотного завдання та основних проблем, з якими стикаються БПЛА: планування шляху, навігація та керування, кожен з цих елементів охоплює ряд підзадач, які вимагають високого рівня керування для бажаного функціонування. Сфера керування БПЛА чи роєм дронів для виконання певних завдань дуже складна. Складність полягає в самій її природі: змішаній динаміці, потребі бути енергетично ефективною та її вразливості до перешкод, шумів датчиків, атмосферних завад, і не модельованої динаміки, взаємодії з невідомими середовищами, що змінюються, тощо. Підходи до навчання динамічних моделей повинні вирішувати такі проблеми як робота зі стохастичністю, невизначеністю, частковістю спостереження тощо. Модель БПЛА важко описати, враховуючи нелінійності в системі. З урахуванням цього найбільш прийнятним для розв'язання проблем керування БПЛА або рою дронів і забезпечення стабільної та плавної навігації БПЛА, запропоновано *DRL* нейронних мереж з підходом без моделей (*Model-Free*) та є декілька робіт зі застосуванням моделі (*Model-Based*).

Останній підхід здійснює оптимізацію процесу послідовного прийняття рішень Маркова за допомоги взаємодії з середовищем і поєднує планування дій агента та його навчання. В цьому разі досягається висока ефективність вибірки як внаслідок вивчення моделі переходу станів та динаміки винагороди, з подальшим її використанням для планування політики, що дозволяє навчатися за меншу кількість дій, ніж у багатьох підходах без моделі та цільового дослідження невизначених та перспективних станів.

На жаль, майже всі роботи мають лабораторний характер, їм бракує перевірки у реальних чи наближених до них умов навколишнього середовища.

Подальші дослідження, на наш погляд, треба зосередити на рішеннях із застосуванням моделі (*Model-Based*) та приділити увагу проєктування типових середовищ, які відповідають тим чи іншим умовам виконання завдання. Такі моделі, можливо, буде простіше і швидше адаптувати до реального оточення.

ЛІТЕРАТУРА / REFERENCES

1. Azar AT, Koubaa A., Ali Mohamed N. et al. Drone Deep Reinforcement Learning: A Review. *Electronics* 2021, Vol. 10(9), 1–30. <https://doi.org/10.3390/electronics10090999>
2. Oursatyev O.A, Volkov, O.Ye, Tkalya V.H. Automated Machine Learning. State and Prospects Development. *Information Technologies and Systems*, Vol. 2(2), 3–33. [In Ukrainian: Автоматизоване машинне навчання. Стан та перспективи розвитку. *Information Technologies and Systems (Інформаційні технології та системи)* <https://doi.org/10.15407/intechsys.2025.02.003>
3. Schmidhuber J. Deep Learning in Neural Networks: An Overview. *Neural Networks*, 2015, Vol. 61, 85–117. <https://doi.org/10.1016/j.neunet.2014.09.003>
4. Sutton, R.S., Barto A.G. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, Massachusetts, London, England, 2018, 526 p. URL: <https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.andrew.cmu.edu/course/10-703/textbook/BartoSutton.pdf>
5. Schmidhuber J. Making the World Differentiable: On Using Self-Supervised Fully Recurrent Neural Networks for Dynamic Reinforcement Learning and Planning in Non-Stationary Environments. *IDSIA*, 1990. URL: [https://people.idsia.ch/~juergen/FKI-126-90_\(revised\)bw_ocr.pdf](https://people.idsia.ch/~juergen/FKI-126-90_(revised)bw_ocr.pdf)
6. Schmidhuber J. Reinforcement Learning in Markovian and Non-Markovian Environments. *IDSIA*, 1991. URL: <https://sferics.idsia.ch/pub/juergen/nipsnonmarkov.pdf>
7. Arulkumaran K. et al. Deep reinforcement learning: A brief survey, 2017. <https://doi.org/10.1109/MSP.2017.2743240>
8. Zhang H., Yu T. Taxonomy of Reinforcement Learning Algorithms. *Deep Reinforcement Learning*, Springer Singapore, 2020. https://doi.org/10.1007/978-981-15-4095-0_3
9. Schmidhuber J. An on-line algorithm for dynamic reinforcement learning and planning in reactive environments. *IJCNN International Joint Conference on Neural Networks*, 1990, Vol. 2, 253–258. <https://doi.org/10.1109/IJCNN.1990.137723>
10. Nagabandi A. et al. PDDM: Planing with Deep Dynamics Models for Learning Dexterous Manipulation. *Conference on Robot Learning (CoRL)*, 2019. URL: <https://sites.google.com/view/pddm>
11. Poole D.L., Mackworth A.K. *AI Foundations of Computational Agents*. University of British Columbia, TJ Books Limited, Padstow, Cornwall, 2023, 870 p. <https://doi.org/10.1017/9781009258227>
12. Francois-Lavet V., Henderson P. et al. *An Introduction to Deep Reinforcement Learning*. 2018. <https://doi.org/10.1561/9781680835397>
13. Ha D., Schmidhuber J. World Models. Can agents learn inside of their own dreams? *NIPS 2018*, March 27 2018, Oral Presentation. <https://doi.org/10.5281/zenodo.1207631>
14. Anastasis Germanidis. Introducing General World Models. URL: <https://research.runwayml.com/introducing-general-world-models> [Accessed Dec. 2023]

15. Gronauer S., Diepold K. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, 2021, 1–49. URL: <https://link.springer.com/article/10.1007/s10462-021-09996-w>
16. Sutton R. S., Barto A. G. *Reinforcement Learning: An Introduction*. A Bradford Book, The MIT Press Cambridge, Massachusetts, London, England, 2015, 1–337. URL: <https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf>
17. Littman M.L. Markov games as a framework for multi-agent reinforcement learning. *Machine Learning Proceedings*, 11-th International Conference, Rutgers University, New Brunswick, NJ, 10–13 Jul. 1994, 157–163. <https://doi.org/10.1016/B978-1-55860-335-6.50027-1>
18. Fully Observable vs. Partially Observable Environment in AI. URL: <https://www.geeksforsgeeks.org/fully-observable-vs-partially-observable-environment-in-ai/> [Accessed May. 2024]
19. Tožička J., Szulyovszky B., de Chambrier G. et al. Application of deep reinforcement learning to UAV fleet control. *SAI Intelligent Systems Conference*, London, UK, 5–6 Sept. 2018, 1169–1177. https://doi.org/10.1007/978-3-030-01057-7_85
20. Liu C.H. et al. Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach. *IEEE J. Sel. Areas Commun.*, Aug. 2018, Vol. 36 (9), 2059–2070. <https://doi.org/10.1109/JSAC.2018.2864373>
21. Yang J. et al. Application of reinforcement learning in UAV cluster task scheduling. *Future Gener. Comput. Syst.* 2019, Vol. 95, 140–148. <https://doi.org/10.1016/j.future.2018.11.014>
22. Wang C. et al. Autonomous navigation of UAV in large-scale unknown complex environment with deep reinforcement learning. *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Montreal, QC, Canada, 14–16 Nov. 2017, 858–862. <https://doi.org/10.1109/GlobalSIP.2017.8309082>
23. Huang Y., Wei G., Wang, Y. V-D D3QN: the Variant of Double Deep Q-Learning Network with Dueling Architecture. *37th Chinese Control Conference (CCC)*, Wuhan, China, 2018, 9130–9135. <https://doi.org/10.23919/ChiCC.2018.8483478>
24. Imanberdiyev N., Fu C., Kayacan E., Chen I.M. Autonomous navigation of UAV by using real-time model-based reinforcement learning. *14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Phuket, Thailand, 13–15 Nov. 2016, 1–6. <https://doi.org/10.1109/ICARCV.2016.7838739>
25. Hester T., Stone P. TEXPLORE: Real-Time Sample-Efficient Reinforcement Learning for Robots. AAI Technical Report SS-12-02. *Designing Intelligent Robots: Reintegrating AI*. Department of Computer Science, The University of Texas at Austin. URL: <https://cdn.aaii.org/ocs/4271/4271-19461-1-PB.pdf>
26. Moerland T.M. *Model-based Reinforcement Learning: A Survey*. 2020, 67 p. URL: <http://arxiv.org/pdf/2006.16712>
27. Bou-Ammar H., Voos H., Ertel W. Controller Design for Quadrotor UAVs using Reinforcement Learning. *IEEE International Conference on Control Applications*, Yokohama, Japan, 8–10 Sept. 2010, 2130–2135. <https://doi.org/10.1109/CCA.2010.5611206>
28. Henderson P. et al. Deep Reinforcement Learning that Matters. Thirtieth-Second AAAI Conference On Artificial Intelligence (AAAI), 2018. <https://doi.org/10.1609/aaai.v32i1.11694>

Received 17.06.2025

O.A. OURSATYEV, PhD (Engineering), Senior Researcher,
Institute of Information Technologies and Systems of the NAS of Ukraine,
40, Hlushkova Akad. ave, Kyiv, 03187, Ukraine
<https://orcid.org/0009-0009-8323-0525>
aleksei@irtc.org.ua

O.Ye. VOLKOV, PhD (Engineering), Senior Researcher, Director,
Institute of Information Technologies and Systems of the NAS of Ukraine,
40, Hlushkova Akad. ave, Kyiv, 03187, Ukraine
<https://orcid.org/0000-0002-5418-6723>
alexvolk@ukr.net

APPROACHES TO CREATING MULTIAGENT SYSTEMS AND DEEP REINFORCEMENT LEARNING OF DRONES

Introduction. Unmanned aerial vehicles (UAVs) are increasingly used in many complex and diverse tasks related to civil and military spheres. UAVs are a class of aircraft, commonly referred to as drones. They can fly without the presence of a human pilot on board. However, there are a number of unsolved problems with UAVs development: flight path planning, navigation and control. In complex systems, which certainly include UAVs, artificial intelligence (AI) is usually used to solve these problems and ensure the required of its functioning, implemented by the method of deep learning with reinforcement. Modern foreign experience in the use of analytical platforms for controlling mobile objects, in particular UAVs, allows for the use of deep neural networks for the above tasks.

The purpose of the paper is to introduce domain experts whose primary job function is outside of machine learning to the challenges of applying AI to these problems, robust and complex deep neural networks and their training, which remains challenging and requires large amounts of data and practical experience. This can be a form of citizen science and will contribute to the replication of research and the democratization of AI.

Results. An analysis of solutions to these problems using deep reinforcement learning is performed, in particular, control of a swarm of UAVs etc. and a taxonomy of Model-Free deep reinforcement learning algorithms applied in UAV tasks is given. The first experience of solutions using the environment model is considered. Unfortunately, almost all works are of a nature, they lack verification in real or close to them environmental conditions. This paper presents a brief overview of approaches to solving problems of reinforcement learning - interactions between agents and the environment in the process of step-by-step decision making. This approach is applied to solving problems of moving objects and complex and partially observable environments; model-free and model-based learning; mathematical formalization of solving UAV problems under reinforcement learning, including paradigms for learning agents in a multi-agent environment Multi-Agent reinforcement learning. Problems arising in the multi-agent field, such as non-stationarity of the environment from the point of view of a single agent, relative overgeneralization and the problem of assigning credits are discussed. Formal concepts underlying these Multi-Agent reinforcement learning are presented.

Conclusions. An overview of methods for solving the problem of reinforced learning is presented, as a result of which the authors conclude that further research should focus on solutions using a model (Model-Based) and pay attention to the design of typical environments that meet certain conditions for performing the task. Such models may be easier and faster to adapt to the real environment.

Keywords: *unmanned moving objects, UAVs, UAV swarm control, swarm of UAVs, deep reinforcement learning, DRL, world models, world models introduces a model-based approach to RL, training paradigms execution scheme.*

COMPUTER VISION AND PATTERN RECOGNITION

КОМП'ЮТЕРНИЙ ЗІР ТА РОЗПІЗНАВАННЯ ОБРАЗІВ

<https://doi.org/10.15407/intechsys.2025.03.056>
УДК 681.5:004.8

О.М. ГОЛОВІН, канд. техн. наук, пров. наук. співроб.,
Інститут кібернетики імені В.М. Глушкова НАН України,
просп. Акад. Глушкова, 40, Київ, 03187, Україна
<https://orcid.org/0000-0002-0279-812X>
o.m.golovin.1@gmail.com

Н.О. САПУНОВА, молодш. наук. співроб.,
Інститут кібернетики імені В.М. Глушкова НАН України,
просп. Акад. Глушкова, 40, Київ, 03187, Україна
<https://orcid.org/0009-0009-1553-1369>
sapunova2008@gmail.com

ЕВОЛЮЦІЯ СИСТЕМ ВІДЕОСПОСТЕРЕЖЕННЯ: ВІД АНАЛОГОВИХ КАМЕР ДО ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ ВІДЕОАНАЛІТИКИ НА ОСНОВІ ГРАНИЧНИХ ОБЧИСЛЕНЬ

Подано комплексний аналіз еволюційного розвитку систем відеоспостереження від первинних аналогових рішень до сучасних інтелектуальних платформ із застосуванням граничних обчислень. Досліджено фундаментальні етапи технологічної трансформації галузі, архітектурні модифікації та функційні можливості відеоаналітичних систем. Виконано глибокий аналіз переваг граничних обчислень як прогресивної парадигми, що забезпечує децентралізоване оброблення даних безпосередньо на граничних пристроях. Обґрунтовано технологічну перевагу граничних обчислень над традиційними централізованими підходами у контексті швидкодії, енергоефективності, захисту персональних даних та відмовостійкості. Окремо розглянуто еволюцію алгоритмів штучного інтелекту в системах відеоспостереження, а також архітектуру сучасних інтелектуальних систем відеоспостереження з граничними обчисленнями, її апаратні та програмні складові.

Ключові слова: відеоспостереження, відеоаналітика, граничні обчислення, штучний інтелект, комп'ютерний зір, глибоке навчання, нейронні мережі, розподілені системи, децентралізоване оброблення даних, ефективність обчислень.

Цитування: Головін О.М., Сапунова Н.О. Еволюція систем відеоспостереження: від аналогових камер до інтелектуальних систем відеоаналітики на основі граничних обчислень. *Information Technologies and Systems*, Київ, 2025, Том 3 (3), 56–75. <https://doi.org/10.15407/intechsys.2025.03.056>

© Видавець ВД «Академперіодика» НАН України, 2025. Стаття опублікована на умовах відкритого доступу за ліцензією CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Вступ

Системи відеоспостереження стали невід'ємним компонентом сучасної інфраструктури безпеки, технологічного ландшафту та цифрової екосистеми загалом. Еволюційний розвиток цих систем протягом останніх десятиліть відображає фундаментальні зміни у парадигмах оброблення даних, обчислювальних можливостях та інтелектуальних технологіях аналізу інформації. Шлях від примітивних аналогових пристроїв до високоінтелектуальних мереж із розподіленими обчисленнями та відеоаналітикою характеризується комплексними технологічними трансформаціями, що суттєво розширили функційні можливості та сфери застосування таких систем [1, 2].

Актуальність наукового дослідження еволюційних процесів у галузі систем відеоспостереження зумовлена не лише історичною цінністю такого аналізу, але й необхідністю розуміння фундаментальних тенденцій, що визначають вектор розвитку сучасних рішень. Особливої значущості набуває концепція граничних обчислень (*edge computing*), що фундаментально змінює архітектурні підходи до побудови інтелектуальних систем відеоаналітики та забезпечує якісно новий рівень ефективності, продуктивності та функційності [3, 4].

На відміну від традиційного підходу, де оброблення відеоданих здійснюється централізовано на віддалених серверах чи хмарних платформах, парадигма граничних обчислень передбачає перенесення значної частини обчислювальних процесів безпосередньо до місця генерації даних — на камери спостереження та локальні граничні пристрої. Така децентралізація оброблення створює передумови для мінімізації мережевого трафіку, зниження затримок, підвищення енергоефективності та забезпечення конфіденційності даних [5].

Метою цього дослідження є комплексний аналіз еволюції систем відеоспостереження від аналогових технологій до сучасних інтелектуальних рішень на основі граничних обчислень, виявлення ключових технологічних інновацій, що формували розвиток галузі, та прогнозування перспективних напрямків подальшого розвитку. Особливу увагу приділено порівняльному аналізу традиційних централізованих підходів та технології граничних обчислень з метою визначення їх оптимальної інтеграції для створення високоефективних систем відеоаналітики. Дослідження спрямоване на формування цілісного розуміння технологічного ландшафту сучасних інтелектуальних систем відеоспостереження та обґрунтування перспективності архітектурних рішень на основі граничних обчислень для подальшого розвитку галузі [6].

1. Історична еволюція систем відеоспостереження

Комплексний аналіз історичного розвитку систем відеоспостереження дає змогу ідентифікувати ключові етапи технологічної трансформації галузі та фундаментальні фактори, що визначали характер еволюційних змін.

1.1. Ера аналогових систем (1950–1990). Фундаментальні основи систем відеоспостереження були закладені в середині ХХ століття з появою перших комерційних телевізійних систем замкнутого контуру (ССТV). Історичні документи свідчать, що перша документально зафіксована система ССТV була розроблена компанією *Siemens AG* у 1942 році для спостереження за запуском ракет V-2 у Німеччині [7]. Комерційне застосування таких систем розпочалося в 1950-х роках, зокрема, для безпеки банківських установ та контролю виробничих процесів.

Технологічний фундамент цієї ери становили аналогові відеокамери на основі електронно-променевих трубок (ЕПТ), коаксіальні системи передавання відеосигналу та апаратура записування на магнітну стрічку. Системи характеризувалися централізованою архітектурою, де аналоговий сигнал від камер передавався безпосередньо на монітори спостереження та пристрої записування [8] (рис. 1).

Науково-технічний аналіз систем цього періоду виявляє низку принципових обмежень:

- низьку роздільну здатність зображення (технічний максимум становив 400–500 телевізійних ліній за фактичної чіткості часто нижче 300 ліній);
- домінування монохромних зображень до 1980-х років через технологічну складність та високу вартість кольорових систем;
- лімітовані можливості реєстрації (тривалість записування на одну касету типу *VHS* не перевищувала 8 годин);
- відсутність автоматизованих засобів аналізу відео контенту;
- високу залежність від людського фактору під час моніторингу [9].

У процесі еволюційного розвитку аналогових систем відеоспостереження відбувалося поступове вдосконалення їхніх компонентів: впровадження твердотільних матриць *CCD* замість ЕПТ, підвищення чутливості камер в умовах низької освітленості, розвиток систем механічного позиціонування камер (*PTZ*-систем). Проте архітектурна парадигма залишалася незмінною – централізована система з прямим передаванням аналогового сигналу від сенсорів до пунктів моніторингу та записування [10].

1.2. Цифрова революція (1990–2010). 1990-ті роки ознаменували початок фундаментальної трансформації галузі відеоспостереження на основі цифрових технологій. Науковий аналіз цього періоду демонструє, що впровадження цифрових методів оброблення, збері-

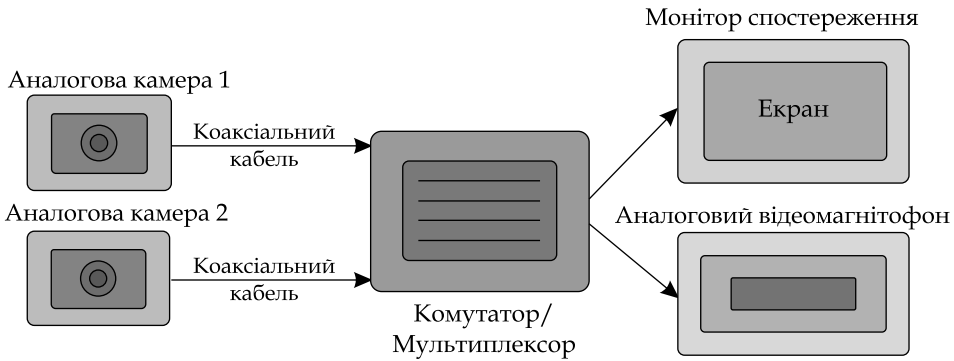


Рис. 1. Архітектура аналогової системи відеоспостереження [8]

гання та передавання інформації спричинило парадигмальні зміни в архітектурі та функційності систем відеоспостереження [11, 12].

Серед ключових технологічних інновацій цього періоду слід відзначити:

- цифрові відеореєстратори (*DVR*), що інтегрували функції аналого-цифрового перетворення, стиснення, індексації та тривалого зберігання відеоданих. Дослідження, результати яких наведено в [13], продемонструвало, що впровадження *DVR* збільшило ефективну ємність систем зберігання у 8-10 разів порівняно з аналоговими відеомагнітофонами;

- мережеві IP-камери, перша з яких була представлена компанією *Axis Communications* у 1996 році, що інтегрували функції захоплення зображення, оброблення та передавання даних через комп'ютерні мережі на базі протоколу *TCP/IP* [14];

- ефективні алгоритми компресії відеоданих (*MJPEG*, *MPEG-2*, *MPEG-4*, *H.264*), що забезпечили зниження вимог до пропускної здатності мережі та ємності систем зберігання при збереженні високої якості зображення [15];

- програмне забезпечення для керування відеоданими (*VMS*), що трансформувало підхід до організації інтерфейсу користувача, забезпечивши інтелектуальний пошук, аналіз та експорт відеоматеріалів [16] (рис. 2).

Цифрова трансформація суттєво розширила функціональні можливості систем відеоспостереження, особливо в аспектах якості зображення, тривалості зберігання та аналітичних функцій. Зокрема, автори роботи [17] зафіксували підвищення ефективної роздільної здатності камер у 3-4 рази (до 1-2 мегапікселів), збільшення періоду безперервного записування до кількох місяців та впровадження початкових форм відеоаналітики, таких як детекція руху на основі піксельного аналізу та цифрове масштабування зображення.

Принципове значення для еволюції систем відеоспостереження мала поява мережевої архітектури на основі *IP*-протоколу, що забезпечила можливість створення розподілених систем з інтеграцією до

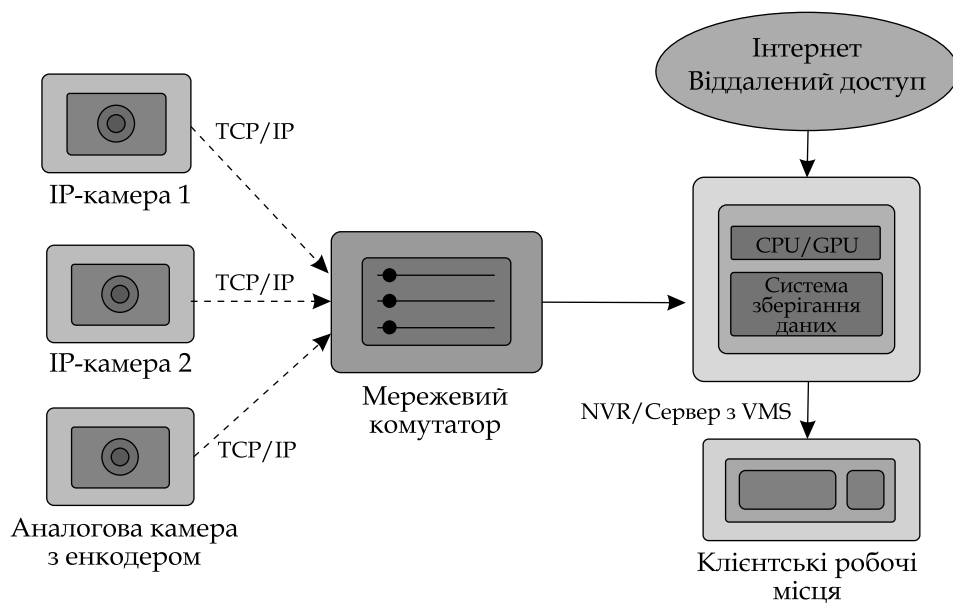


Рис. 2. Архітектура цифрової системи відеоспостереження [16]

корпоративної ІТ-інфраструктури. Це створило фундамент для віддаленого доступу до систем спостереження, централізованого керування та масштабування систем до сотень і тисяч камер [18].

1.3. Хмарна ера та початок відеоаналітики (2010–2018). Цей період характеризувався активним впровадженням хмарних технологій у сферу відеоспостереження та формуванням початкових систем відеоаналітики на основі алгоритмів комп'ютерного зору та машинного навчання. Науковий аналіз цього етапу еволюції демонструє фундаментальні зміни в архітектурних підходах та функційних можливостях систем [19, 20].

Хмарні технології трансформували модель розгортання та використання систем відеоспостереження, забезпечивши:

- перенесення функцій зберігання та оброблення даних на віддалені серверні платформи, доступні через Інтернет;
- формування моделі «відеоспостереження як послуги» (*VSaaS*), що мінімізувала капітальні витрати на розгортання системи;
- впровадження масштабованих систем зберігання та розподілених обчислень для оброблення великих обсягів відеоданих;
- універсальний доступ до систем через веб-інтерфейси та мобільні застосунки [21] (рис. 3).

Дослідження свідчать, що хмарні рішення дали змогу знизити сукупну вартість володіння системами відеоспостереження на 30–40 % завдяки зменшенню витрат на локальне обладнання та технічне обслуговування [22].

Паралельно з розвитком хмарних платформ відбувалося впровадження відеоаналітичних функцій, що базувалися на алгоритмах

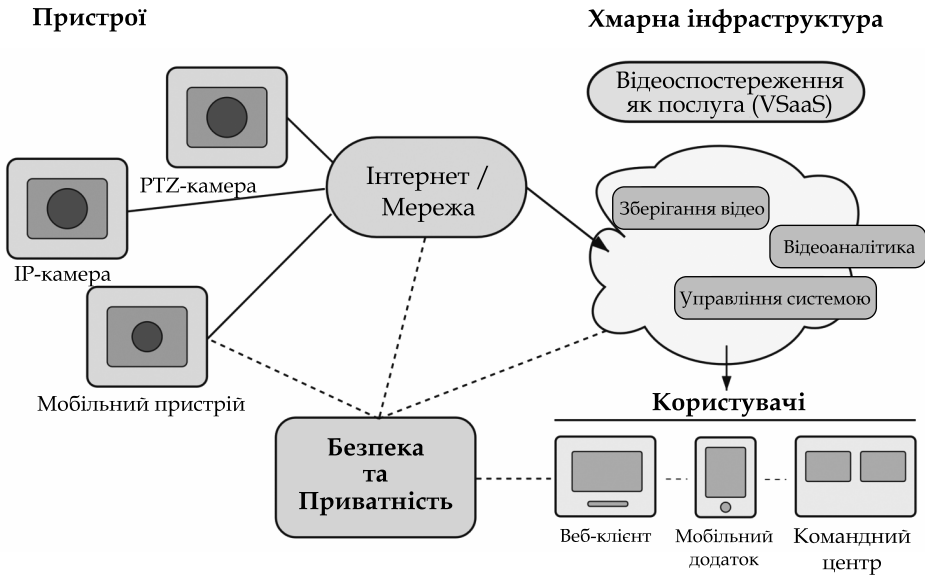


Рис. 3. Хмарна архітектура системи відеоспостереження [21]

комп'ютерного зору. На початку цього періоду основними методами були [23]:

- алгоритми виявлення та відстеження об'єктів на основі видобування та класифікації візуальних ознак (*HOG*, *SIFT*, *SURF*);
- методи виявлення аномальної поведінки на основі статистичного аналізу траєкторій руху;
- технології розпізнавання обличчя на основі геометричних ознак та зіставлення шаблонів [23].

Значним проривом у розвитку відеоаналітики став перехід від традиційних методів комп'ютерного зору до технологій глибокого навчання, особливо після демонстрації ефективності згорткових нейронних мереж (*CNN*) у задачах аналізу зображень на конкурсі *ImageNet* у 2012 році. Дослідження в роботі [24] продемонструвало, що введення процедури автоматичного вибору оптимального параметра гамма-корекції у відеоаналітичних системах значно підвищує ефективність процесів сегментації та розпізнавання об'єктів на відео.

Технологічний розвиток апаратної складової систем відеоспостереження характеризувався:

- суттєвим підвищенням роздільної здатності камер (до рівня 4K та 8K);
- впровадженням багатосенсорних та панорамних камер з кутом огляду 180–360 градусів;
- інтеграцією додаткових сенсорів (тепловізійних, акустичних, радарних) для підвищення ефективності в складних умовах спостереження [25].

Незважаючи на значний прогрес, системи відеоаналітики хмарної ери мали суттєві обмеження, зокрема:

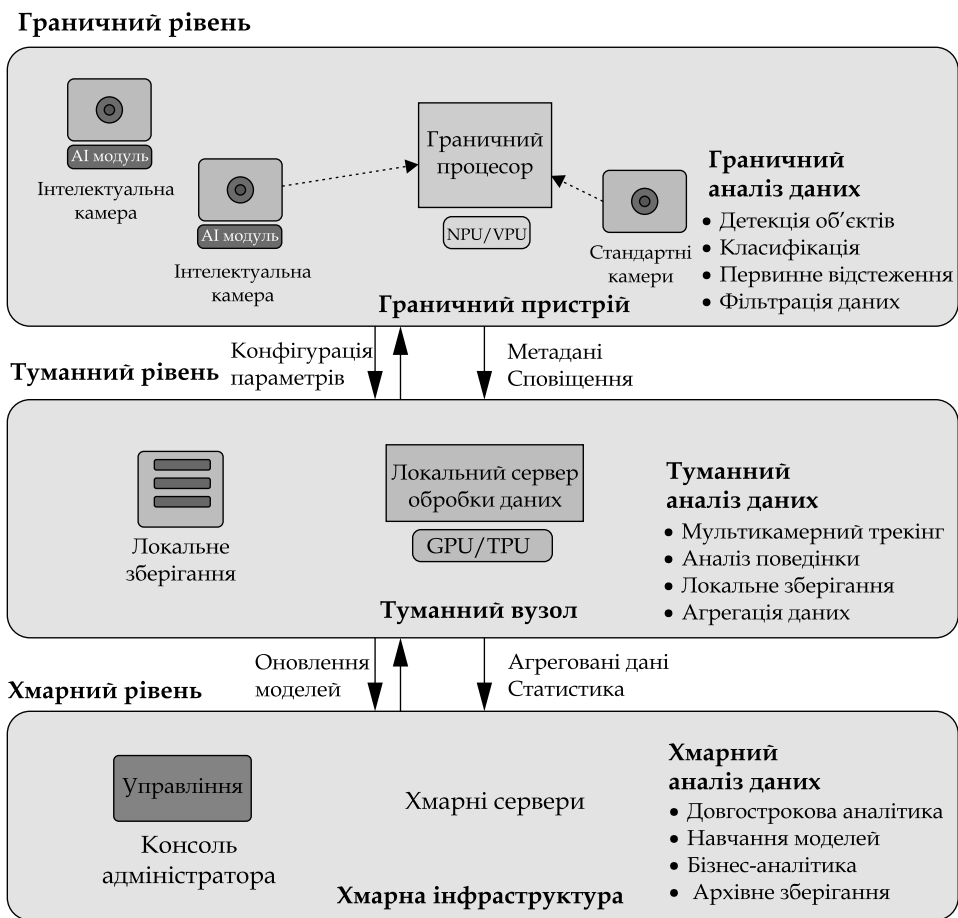


Рис. 4. Модель граничних обчислень у системах відеоспостереження [28]

- високі вимоги до пропускнув здатності мережі для передавання відеопотоків високої роздільної здатності;
- значні затримки при обробленні даних через необхідність передавання відеопотоку до хмарних серверів;
- проблеми з масштабованістю в умовах збільшення кількості камер та підвищення роздільної здатності відео;
- ризики порушення приватності при передаванні та зберіганні чутливих даних у хмарних системах [26, 27].

Ці обмеження стали стимулом для розвитку нової архітектурної парадигми – граничних обчислень.

1.4. Ера граничних обчислень (2018 – теперішній час)

Починаючи з 2018 року галузь відеоспостереження вступила в нову фазу розвитку, пов'язану з масовим впровадженням концепції граничних обчислень. Ця парадигма передбачає перенесення частини або всіх аналітичних функцій безпосередньо на периферійні пристрої – інтелектуальні камери, спеціалізовані граничні сервери та шлюзи [28] (рис. 4).

Ключовими драйверами розвитку граничних обчислень у сфері відеоспостереження стали:

- удосконалення енергоефективних обчислювальних платформ для граничних пристроїв, таких як системи на чипі (SoC) з інтегрованими нейронними прискорювачами;
- розвиток оптимізованих алгоритмів штучного інтелекту, здатних ефективно функціонувати на пристроях з обмеженими ресурсами;
- зростаючі вимоги до приватності та захисту персональних даних, що стимулювали локальне оброблення чутливої інформації;
- потреба в мінімізації затримок при прийнятті рішень у критичних сценаріях застосування [29].

Аналіз еволюційного розвитку систем відеоспостереження демонструє поступове зміщення балансу між централізованими та децентралізованими підходами до оброблення даних, яке досягло певної кульмінації у парадигмі граничних обчислень. Цей підхід забезпечує оптимальний компроміс між функціональними можливостями та обмеженнями, що характерні для систем з різним розподілом обчислювальних ресурсів [30].

2. Еволюція штучного інтелекту в системах відеоспостереження

2.1. Ранні алгоритми аналізу відео та їх обмеження. Перші спроби автоматизації аналізу відеоданих датуються 1990-ми роками і базувались на традиційних методах комп'ютерного зору. Ці підходи охоплювали:

- детектори руху на основі різниці кадрів — прості алгоритми, що порівнювали послідовні кадри для виявлення змін у пікселях, які могли свідчити про рух. Такі методи були вкрай чутливими до зміни освітлення, тіней та шуму в зображенні, що призводило до численних помилкових спрацьовувань;

- методи на основі віднімання фону (*background subtraction*) — виділяли рухомі об'єкти шляхом віднімання поточного кадру від моделі статичного фону. Хоча ці методи були стійкішими за прості детектори руху, вони все ще мали проблеми з адаптацією до зміни умов освітлення, тіней та повільних змін у сцені (рис. 5);

- алгоритми виявлення ознак (*feature detection*), такі як детектор кутів Гарріса, *SIFT (Scale-Invariant Feature Transform)*, та алгоритми їх відстеження на основі оптичного потоку. Ці методи — забезпечували більш надійне відстеження об'єктів, проте не мали здатності до семантичного розуміння сцени;

- статистичні моделі для виявлення аномалій — наприклад, метод Гауссової суміші (*GMM*) для моделювання нормальної поведінки та виявлення відхилень. Такі моделі могли виявляти незвичні події навіть за наявності певних обмежень [31].

Основним обмеженням цих ранніх підходів була відсутність здатності до узагальнення та семантичного розуміння сцени. Вони

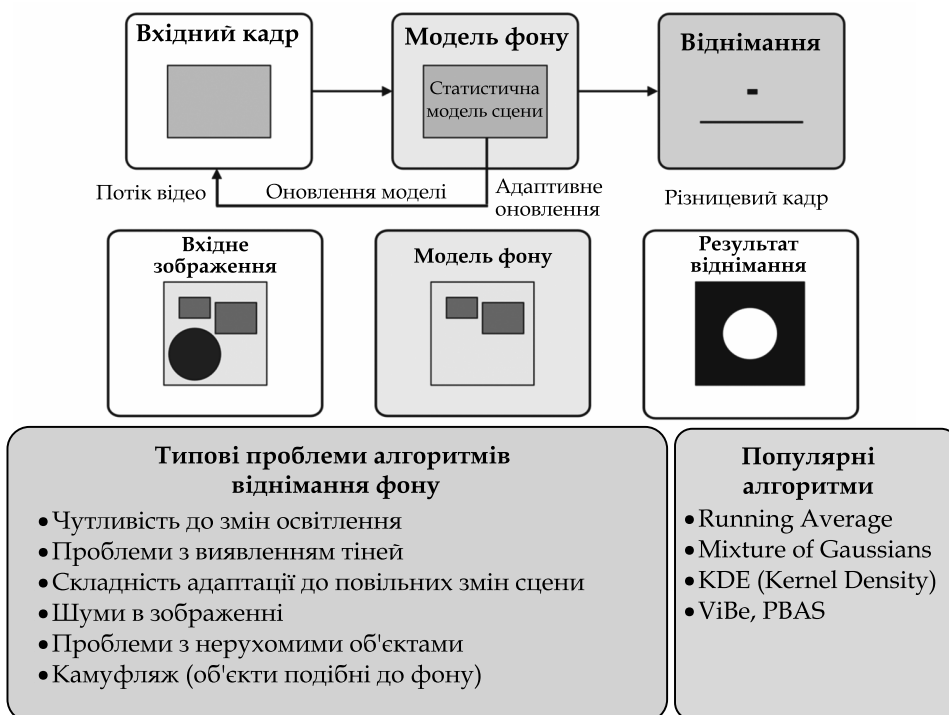


Рис. 5. Методи віднімання фону для виявлення руху [31]

ефективно працювали лише в контрольованих умовах і не могли адаптуватися до різноманіття реальних сценаріїв.

2.2. Прорив глибокого навчання та згорткових нейронних мереж. Революційним моментом для відеоаналітики став 2012 рік, коли модель *AlexNet* продемонструвала вражаючі результати в класифікації зображень на конкурсі *ImageNet*. Це ознаменувало початок ери глибокого навчання в комп'ютерному зорі [32] (рис. 6).

Згорткові нейронні мережі (*CNN*) принесли низку переваг:

- Ієрархічне навчання ознак — від простих елементів (краї, кути) до складних абстрактних концепцій, що дозволило моделям ефективно розпізнавати об'єкти незалежно від змін у їхньому положенні, розмірі та освітленні.

- Здатність до навчання з даних — замість ручного проектування ознак, моделі автоматично вивчали найважливіші характеристики з навчальних прикладів, що суттєво підвищило їхню точність та універсальність.

- Масштабованість — з кожним поколінням мереж відбувалось покращення точності завдяки збільшенню глибини (кількості шарів) та ширини (кількості параметрів) моделей [33].

Першими *CNN*-архітектурами для задач детекції об'єктів були *R-CNN* (*Regions with CNN*), *Fast R-CNN* та *Faster R-CNN*. Вони вперше забезпечили точність розпізнавання, достатню для практичного застосування в системах відеоспостереження.

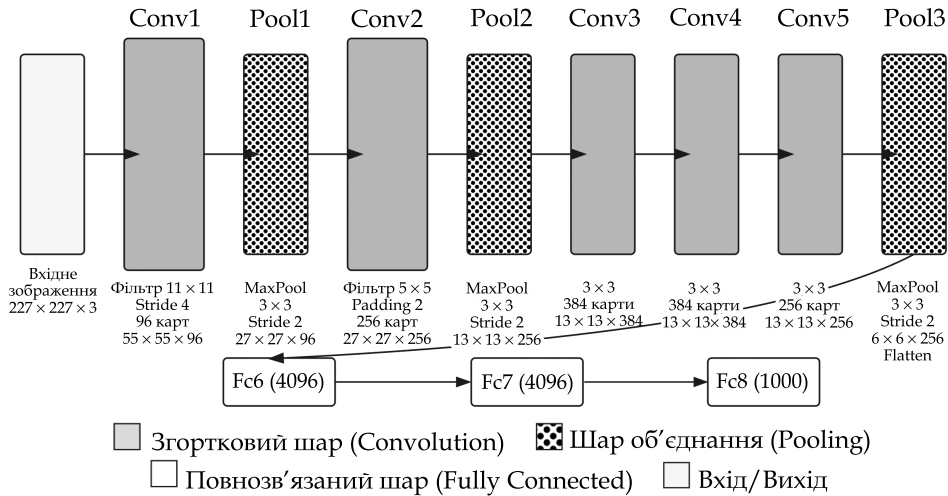


Рис. 6. Архітектура згорткової нейронної мережі AlexNet [32]

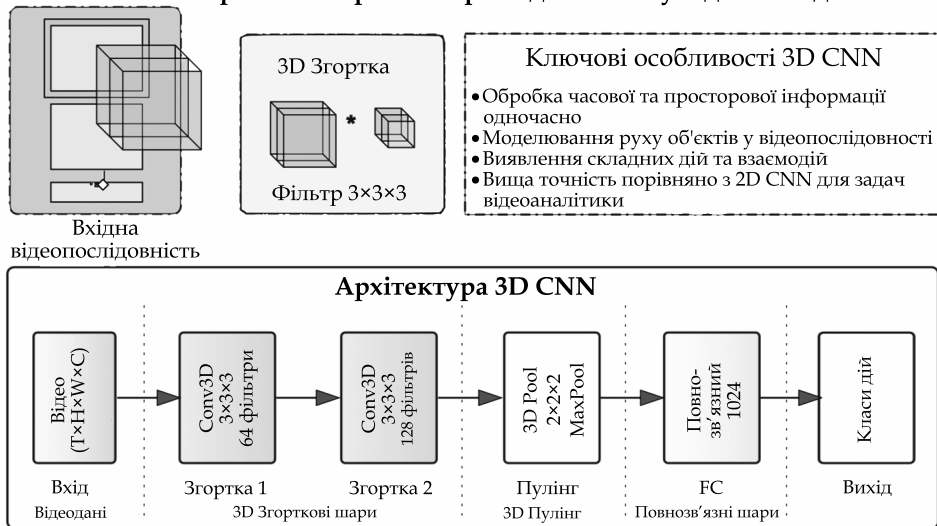
2.3. Спеціалізовані архітектури для відеоаналітики. Від 2015 року почали з'являтися спеціалізовані архітектури, оптимізовані для конкретних задач відеоаналітики:

- однопрохідні детектори об'єктів — такі як YOLO (You Only Look Once) та SSD (Single Shot MultiBox Detector), які забезпечили баланс між точністю та швидкістю, необхідний для аналізу відео в реальному часі;
- мережі для аналізу дій та поведінки — такі як C3D (3D Convolutional Networks), I3D (Inflated 3D ConvNet) та TimeSformer, що обробляють відео як просторово-часові дані, даючи змогу розпізнавати складні дії та взаємодії (рис. 7);
- архітектури для відстеження об'єктів — такі як SORT (Simple Online and Realtime Tracking), DeepSORT, що поєднували традиційні алгоритми відстеження з глибоким навчанням, забезпечуючи стійке відстеження об'єктів у часі;
- моделі для виявлення аномалій — включно автоенкодерами, генеративно-змагальними мережами (GAN) та моделями на основі прогнозування, що дають змогу виявляти незвичні події без явного програмування всіх можливих сценаріїв [34].

Особливо важливим досягненням стала розробка легких архітектур, оптимізованих для роботи на граничних пристроях з обмеженими обчислювальними ресурсами: MobileNet, EfficientNet, ShuffleNet. Ці моделі забезпечили можливість виконання складної відеоаналітики безпосередньо на камерах та інших периферійних пристроях [35].

2.4. Трансформери та мультимодальні моделі в аналізі відео. Останні роки (2020–2024) ознаменувалися впровадженням архітек-

3D згорткові нейронні мережі для аналізу відеопослідовностей



Застосування 3D CNN у відеоаналітиці

- Розпізнавання дій
- Виявлення аномалій
- Відстеження об'єктів
- Аналіз поведінки
- Контроль зони

Рис. 7. 3D згорткові нейронні мережі для аналізу відеопослідовностей [34]

тури трансформерів у відеоаналітику, що раніше революціонізувала оброблення природної мови:

- трансформерні архітектури для зображень (*Vision Transformers, ViT*) та відео (*Video Vision Transformers, ViViT*) забезпечують більш ефективне моделювання довгострокових залежностей у відеопослідовностях порівняно з *CNN*;

- мультимодальні моделі – такі як *CLIP (Contrastive Language-Image Pre-training)*, що поєднують розуміння зображень із текстовими описами, дозволяючи здійснювати пошук та класифікацію об'єктів на основі природномовних запитів;

- фундаментальні моделі (*Foundation Models*) – великі попередньо навчені моделі, здатні виконувати широкий спектр задач без специфічного донавчання для кожної задачі, подібно до великих мовних моделей;

- генеративні моделі для відеоаналітики – такі як *Diffusion Models*, що використовуються для покращення якості відео, реконструкції зображення в умовах низької видимості та генерації синтетичних даних для навчання інших моделей [36].

Ці новітні підходи суттєво розширюють можливості відеоаналітики, забезпечуючи більш глибоке семантичне розуміння сцени, здатність до узагальнення та адаптації до нових умов та сценаріїв.

3. Архітектура сучасних інтелектуальних систем відеоспостереження

3.1. Загальна архітектура систем з граничними обчисленнями.

Сучасні інтелектуальні системи відеоспостереження базуються на багаторівневій архітектурі, що поєднує граничні, туманні та хмарні обчислення для забезпечення оптимального балансу між продуктивністю, затримкою та масштабованістю. Ця архітектура охоплює:

Граничний рівень:

- інтелектуальні камери з вбудованими обчислювальними модулями;
- спеціалізовані граничні пристрої для оброблення потоків від кількох камер;
- локальні NVR з функціями відеоаналітики.

На цьому рівні здійснюється первинне оброблення відеоданих, а саме: детекція об'єктів, класифікація, відстеження та розпізнавання простих подій. Це дає змогу відфільтрувати нерелевантну інформацію та зменшити обсяг даних, що передаються на вищі рівні [37] (рис. 8).

Туманний рівень:

- локальні сервери оброблення даних;
- шлюзи для агрегації даних від груп камер;
- локальні системи зберігання.

Цей проміжний рівень забезпечує складнішу аналітику (що вимагає контексту від кількох камер), таку як відстеження об'єктів між

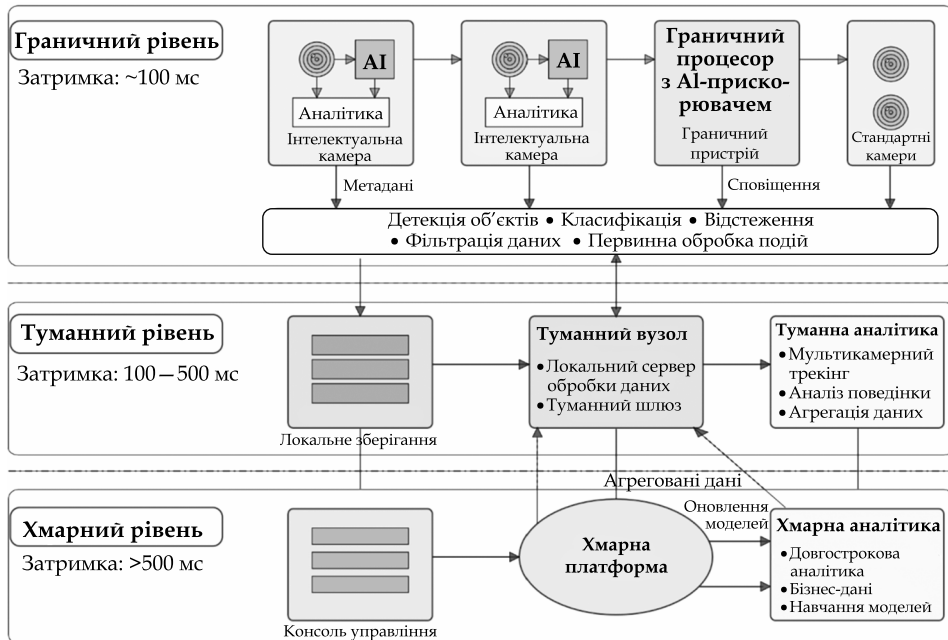


Рис. 8. Багаторівнева архітектура системи відеоаналітики з граничними обчисленнями [37]

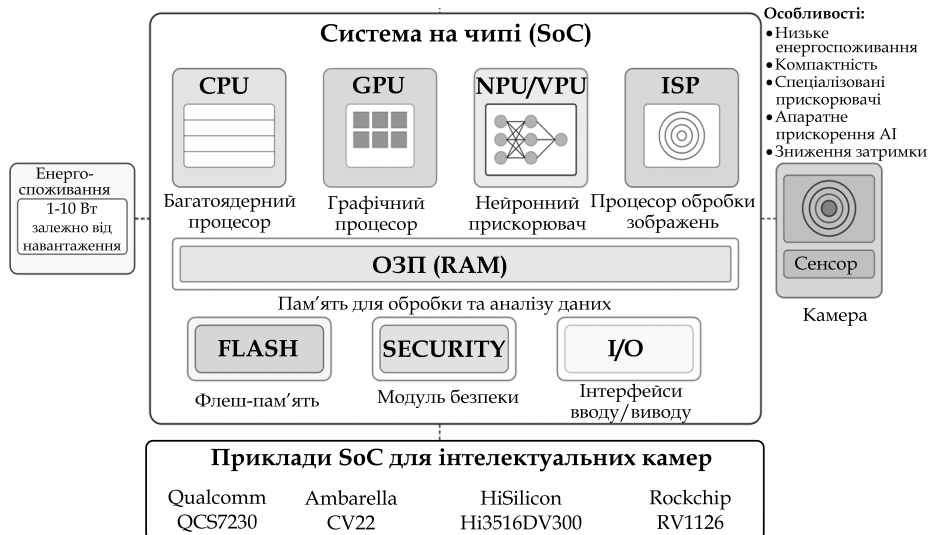


Рис. 9. Системи на чипі (SoC) для інтелектуальних камер [40]

Таблиця 1. Спеціалізовані апаратні платформи для граничних обчислень у системах відеоспостереження

Апаратні платформи	Призначеність
Системи на чипі (SoC) для інтелектуальних камер	Інтегрують CPU, GPU та спеціалізовані прискорювачі для комп'ютерного зору, забезпечуючи потужне оброблення відео при низькому енергоспоживанні. Вони дають змогу камерам виконувати попередню аналітику безпосередньо на пристрої [40] (рис. 9)
Спеціалізовані процесори для прискорення AI	Ці прискорювачі оптимізовані для ефективного виконання операцій нейронних мереж, таких як згортки, що дає їм змогу досягати високої продуктивності при нижчому енергоспоживанні порівняно з пристроями загального призначення (CPU або GPU)
Граничні сервери та шлюзи	Забезпечують оброблення даних від кількох камер одночасно, виконуючи більш складну аналітику та координуючи роботу розподіленої системи [41]
Модульні системи та карти розширення	Дають змогу гнучко розширювати обчислювальні можливості наявних платформ та легко масштабувати системи відповідно до зростання вимог

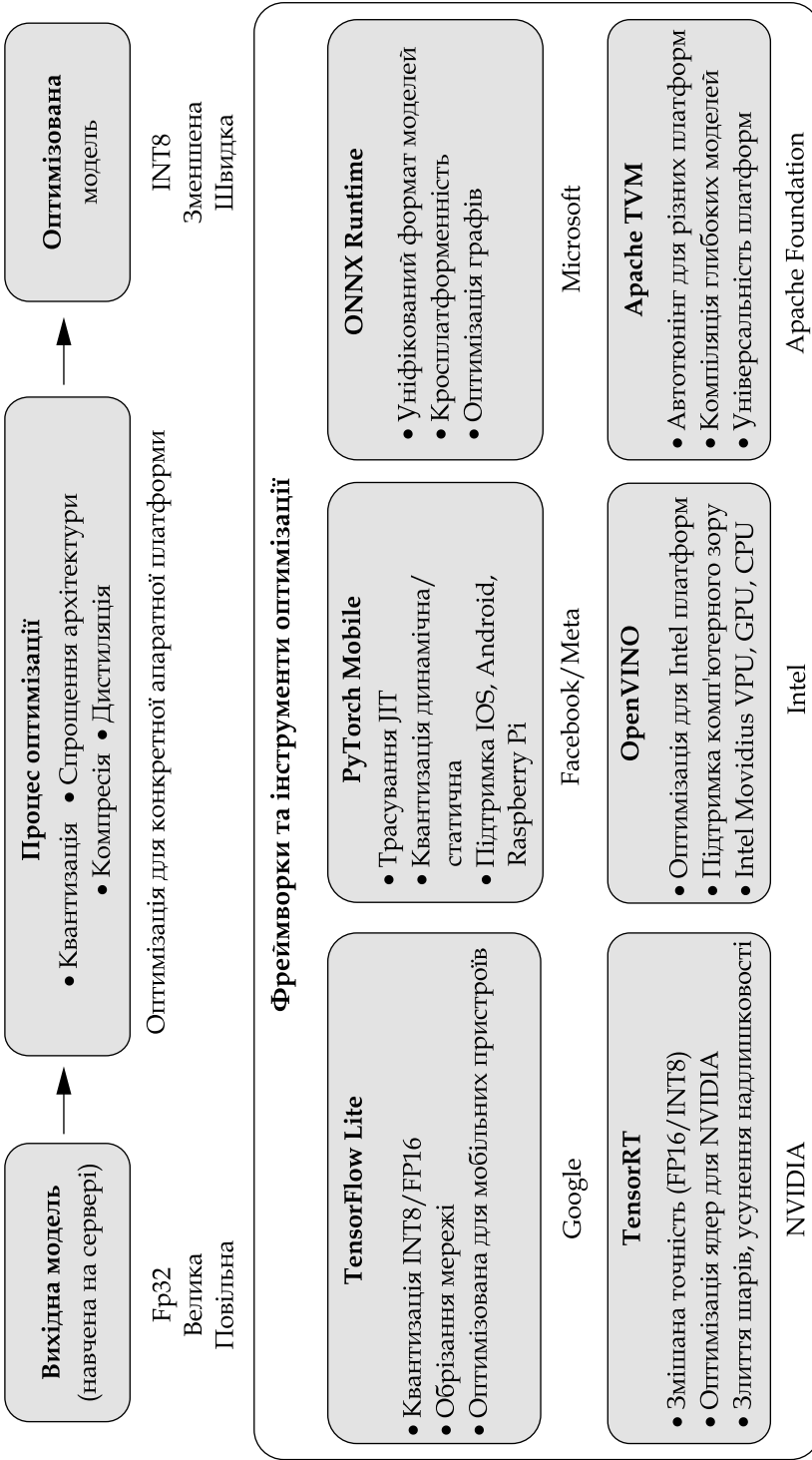
Таблиця 2. Програмні платформи та фреймворки для граничних обчислень у системах відеоспостереження

Спеціалізовані програмні платформи та фреймворки	Призначеність	
Фреймворки для розробки моделей	<p><i>TensorFlow Lite</i> для граничних пристроїв; <i>PyTorch Mobile</i>; <i>ONNX Runtime</i>; <i>Apache TVM</i>; <i>Glow</i></p>	<p>Забезпечують інструменти оптимізації моделей для роботи на пристроях з обмеженими ресурсами, включно з квантизацією, спрощенням архітектури (<i>Architecture Simplification</i>) або стисненням моделі (<i>Model Compression</i>), компресією та апаратно-специфічною оптимізацією [43] (рис. 10)</p>
Інструменти оптимізації для граничних обчислень	<p><i>NVIDIA TensorRT</i>; <i>Intel OpenVINO</i>; <i>TensorFlow Model Optimization Toolkit</i>; <i>Qualcomm AI Engine</i>; <i>MediaTek NeuroPilot</i></p>	<p>Дають змогу адаптувати моделі, навчені на потужних серверах, для ефективного виконання на граничних пристроях з обмеженими ресурсами, зі збереженням високої точності</p>
Програмні стеки для керування граничними пристроями	<p><i>AWS IoT Greengrass</i>; <i>Azure IoT Edge</i>; <i>Google Cloud IoT</i>; <i>EdgeX Foundry</i>; <i>KubeEdge</i></p>	<p>Забезпечують інфраструктуру для розгортання, моніторингу та оновлення програмного забезпечення на розподілених граничних пристроях, а також координацію між пристроями та хмарною інфраструктурою [44]</p>
Програмне забезпечення для відеоменеджменту та аналітики	<p><i>Milestone XProtect</i>; <i>Genetec Security Center</i>; <i>Axis AXIS Camera Station</i>; <i>Avigilon Control Center</i>; OpenSource рішення: <i>ZoneMinder</i>, <i>MotionEye</i></p>	<p>Інтегрують традиційні функції відеоменеджменту з можливостями сучасної відеоаналітики, забезпечуючи єдиний інтерфейс для керування фізичними пристроями, відеопотоками та результатами</p>
API та стандарти для інтеграції	<p><i>ONVIF Profile M</i> для метаданих відеоаналітики; <i>Open Security & Safety Alliance (OSSA) App Framework</i>; <i>NVIDIA DeepStream SDK</i>; Відкриті API для AI-моделей (<i>Common API</i>); <i>RESTful</i> та <i>MQTT</i> інтерфейси для IoT-інтеграції</p>	<p>Забезпечують взаємодію між компонентами різних виробників та можливість інтеграції з іншими системами, такими як системи контролю доступу, системи автоматизації будівель та бізнес-аналітичні платформи [45]</p>

камерами, аналіз потоків, а також поведінкову аналітику. Туманні обчислення також забезпечують буферизацію даних у разі проблем зв'язку з хмарою [38].

Хмарний рівень:

- центральні сервери оброблення та аналізу даних;
- тривале зберігання;



Квантизація: Зниження точності (FP32-INT8) **Прорідження:** Видалення надлишкових ваг
Дистиляція: Передача знань малій моделі

Рис. 10. Фреймворки оптимізації моделей для граничних обчислень [43]

- платформи для керування системою та візуалізації даних.

На хмарному рівні здійснюється глобальна аналітика, що потребує даних від усієї системи, навчання моделей, зберігання історичних даних та надання інтерфейсів для користувачів.

Ключовими принципами архітектури є:

- розподіл обчислювальних задач за принципом близькості до джерела даних;
- каскадна фільтрація даних для зменшення навантаження на мережу;
- адаптивний розподіл задач залежно від доступних ресурсів та потреб;
- забезпечення локальної автономності в разі проблем зі зв'язком;
- гнучка масштабованість як горизонтально, так і вертикально [39].

3.2. Апаратні та програмні засоби для граничних обчислень.

Розвиток граничних обчислень у системах відеоспостереження став можливим завдяки появі енергоефективних і спеціалізованих апаратних платформ для оброблення даних, а також спеціалізованих програмних платформ та фреймворків, що забезпечують ефективне розроблення, розгортання та керування AI-моделями на граничних пристроях (табл. 1 і 2).

Ключові характеристики сучасних апаратних платформ:

- низьке енергоспоживання (від кількох ват до кількох десятків ват);
- оптимізація для конкретних задач відеоаналітики;
- підтримка різних форматів кількісного подання (*INT8*, *FP16*, *BF16*);
- вбудовані функції безпеки та шифрування;
- стійкість до роботи в складних умовах (широкий температурний діапазон, вібрації, тощо) [42].

Висновки

Проведене дослідження еволюції систем відеоспостереження демонструє фундаментальну трансформацію галузі від примітивних аналогових рішень до складних інтелектуальних екосистем з інтегрованими можливостями відеоаналітики на основі граничних обчислень. Цей еволюційний шлях характеризується не лише технологічним прогресом, але й парадигмальними змінами в архітектурних підходах та концептуальному розумінні ролі систем відеоспостереження в сучасному суспільстві.

Аналіз розвитку технології граничних обчислень у контексті систем відеоспостереження виявив їхні суттєві переваги порівняно з традиційними централізованими підходами. Децентралізоване оброблення відеоданих безпосередньо на периферійних пристроях забезпечує значне зниження мережевого трафіку, мінімізацію затримок при прийнятті рішень, підвищення енергоефективності та формування потужного фундаменту для забезпечення приватності та конфіденційності даних.

Інтелектуальні алгоритми відеоаналітики пройшли шлях від простих методів детекції руху до складних нейромережових моделей, здатних до семантичного розуміння сцени та розпізнавання складних поведінкових патернів. Особливого значення набувають технології оптимізації нейромережових моделей для ефективної роботи на граничних пристроях з обмеженими обчислювальними ресурсами, включно з квантизацією, спрощенням архітектури або стисненням моделі мереж, дистиляцією знань та нейроархітектурним пошуком.

Сучасні апаратні платформи для граничних обчислень, такі як спеціалізовані системи на чипі, нейроморфні процесори та тензорні прискорювачі, забезпечують безпрецедентний рівень обчислювальної потужності при низькому енергоспоживанні, що є критично важливим для розгортання інтелектуальних систем у різноманітних сценаріях застосування.

Майбутній розвиток систем відеоспостереження пов'язаний з конвергенцією технологій, інтеграцією з 5G / 6G-мережами, системами доповненої реальності, робототехнікою та платформами інтернету речей. Паралельно відбуватиметься вдосконалення алгоритмів штучного інтелекту в напрямі підвищення їх ефективності, адаптивності, інтерпретованості та етичності.

У підсумку, еволюція систем відеоспостереження від аналогових камер до інтелектуальних систем відеоаналітики на основі граничних обчислень відображає глобальні технологічні тренди цифрової трансформації та формує основу для нового покоління рішень, що гармонійно поєднують високу функційність, енергоефективність, захист приватності та етичне використання. Подальший розвиток цих технологій потребує міждисциплінарного підходу та співпраці між науковцями, інженерами, бізнесом, регуляторними органами та суспільством для максимізації потенційних переваг за мінімізації ризиків.

ЛІТЕРАТУРА / REFERENCES

1. Myagmar-Ochir Y., Kim W. A Survey of Video Surveillance Systems in Smart Cities. *Electronics*, 2023, Vol. 12 (17), Article 3567. <https://doi.org/10.3390/electronics12173567>
2. Patrikar D.R., Parate M.R. Anomaly Detection Using Edge Computing in Video Surveillance System: Review. *International Journal of Multimedia Information Retrieval*, 2022, Vol. 11 (2), 85–110. <https://doi.org/10.1007/s13735-022-00227-8>
3. Khan W.Z., Ahmed E., Hakak S., Yaqoob I., Ahmed A. Edge Computing: A Survey. *Future Generation Computer Systems*, 2019, Vol. 97, 219–235. <https://doi.org/10.1016/j.future.2019.02.050>
4. Wang X., Han Y., Leung V.C., Niyato D., Yan X., Chen X. Convergence of Edge Computing and Deep Learning: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*, 2020, Vol. 22 (2), 869–904. <https://doi.org/10.1109/COMST.2020.2970550>
5. Dianlei Xu, Tong Li, Yong Li, Xiang Su, Sasu Tarkoma, Tao Jiang, Jon Crowcroft, Pan Hui. Edge intelligence: Architectures, challenges, and applications. *arXiv preprint arXiv*, 2020, Vol. 2003, Article 12172. <https://doi.org/10.48550/arXiv.2003.12172>
6. Ezzat M.A., Abd El Ghany M.A., Almotairi S., Salem M.A.M. Horizontal Review on Video Surveillance for Smart Cities: Edge Devices, Applications, Datasets, and Future Trends. *Sensors*, 2021, Vol. 21 (9), 3222. <https://doi.org/10.3390/s21093222>

7. Kruegle H. *CCTV Surveillance: Video practices and technology*. Elsevier, 2011.
8. Alexandria G. Surveillance Cameras and Crime: A Review of Randomized and Natural Experiments. *Journal of Scandinavian Studies in Criminology and Crime Prevention*, 2017, Vol. 18 (2), 210–222. <https://doi.org/10.1080/14043858.2017.1387410>
9. Piza E.L., Welsh B.C., Farrington D.P., Thomas A.L. CCTV Surveillance for Crime Prevention: A 40-Year Systematic Review with Meta-Analysis. *Criminology & Public Policy*, 2019, Vol. 18 (1), 135–159. <https://doi.org/10.1111/1745-9133.12419>
10. Vennam P., Pramod T.C., Thippeswamy B.M., Kim Y.-G., Pavan Kumar B.N. Attacks and Preventive Measures on Video Surveillance Systems: A Review. *Applied Sciences*, 2021, Vol. 11 (12), Article 5571. <https://doi.org/10.3390/app11125571>
11. Shi W., Cao J., Zhang Q., Li Y., Xu, L. Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, 2016, Vol. 3 (5), 637–646. <https://doi.org/10.1109/JIOT.2016.2579198>
12. Abbas N., Zhang Y., Taherkordi A., Skeie T. Mobile Edge Computing: A Survey. *IEEE Internet of Things Journal*, 2018, Vol. 5 (1), 450–465. <https://doi.org/10.1109/JIOT.2017.2750180>
13. Chen Y., Zhang S., Xu S., Li G.Y. Fundamental Trade-offs on Green Wireless Networks. *IEEE Communications Magazine*, 2011, Vol. 49 (6), 30–37. <https://doi.org/10.1109/MCOM.2011.5783982>
14. Nilsson F. *Intelligent network video: Understanding modern video surveillance systems*. crc Press, 2023. <https://doi.org/10.4324/9781003412205>
15. Sullivan G.J., Ohm J.R., Han W.J., Wiegand T. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, Vol. 22 (12), 1649–1668. <https://doi.org/10.1109/TCSVT.2012.2221191>
16. Kolekar M.H. *Intelligent video surveillance systems: an algorithmic approach*. Chapman and Hall/CRC, 2018. <https://doi.org/10.1201/9781315153865>
17. Elharrouss O., Almaadeed N., Al-Maadeed S. A review of video surveillance systems. *Journal of Visual Communication and Image Representation*, 2021, Vol. 77, Article 103116. <https://doi.org/10.1016/j.jvcir.2021.103116>
18. Wang C, He Y., Yu F.R., Chen Q., Tang L. Integration of Networking, Caching, and Computing in Wireless Systems: A Survey, Some Research Issues, and Challenges. *IEEE Communications Surveys & Tutorials*, 2018, Vol. 20 (1), 7–38. <https://doi.org/10.1109/COMST.2017.2758763>
19. Silva B.N., Khan M., Han K. Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. *Sustainable cities and society*, 2018, Vol. 38, 697–713. <https://doi.org/10.1016/j.scs.2018.01.053>
20. LeCun Y., Bengio Y., Hinton G. Deep Learning. *Nature*, 2015, Vol. 521, 436–444. <https://doi.org/10.1038/nature14539>
21. Roman R., Lopez J., Mambo M. Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. *Future Generation Computer Systems*, 2018, Vol. 78, 680–698. <https://doi.org/10.1016/j.future.2016.11.009>
22. Bilal K., Khalid O., Erbad A., Khan S.U. Potentials, Trends, and Prospects in Edge Technologies: Fog, Cloudlet, Mobile Edge, and Micro Data Centers. *Computer Networks*, 2018, Vol. 130, 94–120. <https://doi.org/10.1016/j.comnet.2017.10.002>
23. Gawande, U., Hajari, K., & Golhar, Y. (2020). Pedestrian detection and tracking in video surveillance system: issues, comprehensive review, and challenges. Recent Trends in Computational Intelligence, 1–24. DOI: <http://dx.doi.org/10.5772/intechopen.90810>
24. Golovin O. Computer-Aided Selection of the Optimal Gamma Correction Parameter for Image Enhancement in Video Analytics Systems. *11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems*, 2021, 1–6. <https://doi.org/10.1109/IDAACS53288.2021.9660869>
25. Abbasi A., Queirós S., da Costa N.M.C., Fonseca J.C., Borges J. Sensor Fusion Approach for Multiple Human Motion Detection for Indoor Surveillance Use-Case. *Sensors*, 2023, Vol. 23 (8), Article 3993. <https://doi.org/10.3390/s23083993>

26. Liu J., Wang H. Scalability Issues in Large-Scale Cloud-Based Video Analytics Systems. *Future Generation Computer Systems*, 2022, Vol. 131, 234–248. <https://doi.org/10.1016/j.future.2022.01.018>
27. Hassan M.M., Gumaiei A., Alsanad A., Alrubaian M., Fortino G. A Hybrid Deep Learning Model for Efficient Intrusion Detection in Big Data Environment. *Information Sciences*, 2020, Vol. 513, 386–396. <https://doi.org/10.1016/j.ins.2019.10.069>
28. Li H., Ota K., Dong M. Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing. *IEEE Network*, 2018, Vol. 32 (1), 96–101. <https://doi.org/10.1109/MNET.2018.1700202>
29. Chen J., Ran X. Deep Learning with Edge Computing: A Review. *Proceedings of the IEEE*, 2019, Vol. 107 (8), 1655–1674. <https://doi.org/10.1109/JPROC.2019.2921977>
30. Satyanarayanan M. The Emergence of Edge Computing. *Computer*, 2017, Vol. 50 (1), 30–39. <https://doi.org/10.1109/MC.2017.9>
31. Bouwmans T., Silva C., Marghes C., Zitouni M.S., Bhaskar H., Frelicot C. On the Role and the Importance of Features for Background Modeling and Foreground Detection. *Computer Science Review*, 2017, Vol. 28, 26–91. <https://doi.org/10.1016/j.cosrev.2018.01.004>
32. Krizhevsky A., Sutskever I., Hinton G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*, 2017, Vol. 60 (6), 84–90. <https://doi.org/10.1145/3065386>
33. SzeV., Chen Y.-H., Yang T.-J., Emer J.S. Efficient Processing of Deep Neural Networks: A Tutorial and Survey. *Proceedings of the IEEE*, 2017, Vol. 105 (12), 2295–2329. <https://doi.org/10.1109/JPROC.2017.2761740>
34. Kumari, S., Prabha, C., Karim, A., Hassan, M. M., & Azam, S. (2024). A Comprehensive Investigation of Anomaly Detection Methods in Deep Learning and Machine Learning: 2019–2023. *IET Information Security*, 2024(1), 8821891. <https://doi.org/10.1049/2024/8821891>
35. Howard et al. Searching for MobileNetV3. *IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 2019, 1314–1324. <https://doi.org/10.1109/ICCV.2019.00140>
36. Dosovitskiy A., Beyer L., Kolesnikov A., Weissenborn D., Zhai X., et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *International Conference on Learning Representations (ICLR)*, 2021. URL: <https://openreview.net/forum?id=YicbFdNTTy>
37. Yu W. et al. A Survey on the Edge Computing for the Internet of Things. *IEEE Access*, 2018, Vol. 6, 6900–6919. <https://doi.org/10.1109/ACCESS.2017.2778504>
38. Bonomi F., Milito R., Zhu J., Addepalli S. Fog Computing and Its Role in the Internet of Things. *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, 2012, 13–16. <https://doi.org/10.1145/2342509.2342513>
39. Satyanarayanan M., Bahl P., Caceres R., Davies N. The Case for VM-Based Cloudlets in Mobile Computing. *IEEE Pervasive Computing*, 2009, Vol. 8 (4), 14–23. <https://doi.org/10.1109/MPRV.2009.82>
40. Chen S., Xu H., Liu D., Hu B., Wang H. A Vision of IoT: Applications, Challenges, and Opportunities with China Perspective. *IEEE Internet of Things Journal*, 2014, Vol. 1 (4), 349–359. <https://doi.org/10.1109/JIOT.2014.2337336>
41. Mao Y., You C., Zhang J., Huang K., Letaief K.B. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Communications Surveys & Tutorials*, 2017, Vol. 19 (4), 2322–2358. <https://doi.org/10.1109/COMST.2017.2745201>
42. Liu Y., Yang C., Jiang L., Xie S., Zhang Y. Intelligent Edge Computing for IoT-Based Energy Management in Smart Cities. *IEEE Network*, 2019, Vol. 33 (2), 111–117. <https://doi.org/10.1109/MNET.2019.1800254>
43. Zhang X., Wang Y., Lu S., Liu L., Xu L., Shi W. OpenEI: An Open Framework for Edge Intelligence. *IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, Dallas, TX, USA, 2019, 1840–1851. <https://doi.org/10.1109/ICDCS.2019.00182>

44. Hong C.H., Varghese B. Resource Management in Fog/Edge Computing: A Survey on Architectures, Infrastructure, and Algorithms. *ACM Computing Surveys*, 2019, Vol. 52 (5), 1–37. <https://doi.org/10.1145/3326066>
45. Premsankar G., Di Francesco M., Taleb T. Edge Computing for the Internet of Things: A Case Study. *IEEE Internet of Things Journal*, 2018, Vol. 5 (2), 1275–1284. <https://doi.org/10.1109/JIOT.2018.2805263>

Отримано / Received: 16.06.2025

O.M. GOLOVIN, PhD (Engineering), Leading Researcher,
Glushkov Institute of Cybernetics of the NAS of Ukraine,
40, Hlushkova Akad. ave., Kyiv, 03187, Ukraine
<https://orcid.org/0000-0002-0279-812X>
o.m.golovin.1@gmail.com

N.O. SAPUNOVA, Junior Researcher,
Glushkov Institute of Cybernetics of the NAS of Ukraine,
40, Hlushkova Akad. ave., Kyiv, 03187, Ukraine
<https://orcid.org/0009-0009-1553-1369>
sapunova2008@gmail.com

EVOLUTION OF VIDEO SURVEILLANCE SYSTEMS: FROM ANALOG CAMERAS TO INTELLIGENT VIDEO ANALYTICS SYSTEMS BASED ON EDGE COMPUTING

Introduction. Video surveillance systems have undergone significant evolution from primitive analog devices to highly intelligent networks with distributed computing and video analytics. Current industry development is characterized by the implementation of edge computing concept, which fundamentally changes architectural approaches to building intelligent video analytics systems. Unlike traditional centralized video data processing on remote servers, the edge computing paradigm involves transferring computational processes directly to surveillance cameras and local edge devices.

The purpose of the paper is a comprehensive analysis of the evolution of video surveillance systems from analog technologies to modern intelligent solutions based on edge computing, identification of key technological innovations, and forecasting promising development directions with particular attention to comparative analysis of traditional centralized approaches and edge computing technology.

Methods. The research is based on systematic analysis of historical stages of video surveillance systems development, comparative analysis of architectural solutions and functional capabilities of different generations of video analytics systems, study of artificial intelligence algorithms evolution and assessment of edge computing advantages over centralized approaches.

Results. Four main evolutionary stages were identified: analog systems era (1950–1990), digital revolution (1990–2010), cloud era (2010–2018) and edge computing era (2018–present). It was demonstrated that edge computing provides significant network traffic reduction, latency minimization, energy efficiency improvement and personal data protection compared to traditional approaches.

Conclusions. The evolution of video surveillance systems is characterized by paradigmatic changes from centralized to decentralized data processing. Edge computing forms the foundation for a new generation of solutions that combine high functionality, energy efficiency, and privacy protection. Future development is associated with convergence of 5G/6G technologies, augmented reality and Internet of Things.

Keywords: *video surveillance, video analytics, edge computing, artificial intelligence, computer vision, deep learning, neural networks, distributed systems, decentralized data processing, computational efficiency.*

DIGITAL MEDICINE

ЦИФРОВА МЕДИЦИНА

<https://doi.org/10.15407/intechsys.2025.03.076>
УДК 004.318

М.Г. ПЕТРЕНКО, д-р техн. наук, старш. наук. співроб., провідн. наук. співроб,
Інститут кібернетики імені В.М. Глушкова НАН України,
просп. Акад. Глушкова, 40, Київ, 03187, Україна
<https://orcid.org/0000-0001-6440-0706>
petrng@ukr.net

К.С. МАЛАХОВ, наук. співроб.,
Інститут кібернетики імені В.М. Глушкова НАН України,
просп. Акад. Глушкова, 40, Київ 03187, Україна
<https://orcid.org/0000-0003-3223-9844>
malakhovks@outlook.com

МЕТОДОЛОГІЯ І ПРАКТИКА КОНВЕРГЕНЦІЇ МІЖДИСЦИПЛІНАРНИХ ЗНАНЬ У СФЕРІ ЦИФРОВОЇ ОХОРОНИ ЗДОРОВ'Я¹

У статті обґрунтовано методологічні засади трансдисциплінарного підходу та показано його прикладне значення для цифрової охорони здоров'я. Інтеграція знань медицини, інформатики, інженерії та соціально-поведінкових наук розглядається як передумова

¹ Дослідження виконано за договором з ТНМУ ім. І.Я. Горбачевського МОЗ України – НДР «Розроблення дослідного зразка експертної системи підтримки прийняття рішень на основі ШІ» (<https://prozorro.gov.ua/uk/contract/UA-2025-05-13-012069-a-c1>), у межах виконання НТР «Розроблення персоналізованої теледіагностичної платформи з ШІ для лікаря та пацієнта (ТД+AI)» згідно з договором між МОН України та ТНМУ імені І.Я. Горбачевського МОЗ України від 12.02.2025 №ДЗ/164-2025. Реєстраційна картка НДДКР 0125U001036 від 13.02.2025.

Дослідження також проведено в межах НДР «Розробити теоретичні засади і функціональну модель комп'ютера для обробки складних інформаційних структур» (державний реєстраційний номер: 0124U002317, <https://nrat.ukrintei.ua/searchdoc/0124U002317/>) та НДР «Розробити засоби підтримки технологій віртуалізації та їх використання в комп'ютерній інженерії та інших застосуваннях» (державний реєстраційний номер: 0124U001826, <https://nrat.ukrintei.ua/searchdoc/0124U001826/>)

Усі проєкти реалізовано на базі Інституту кібернетики імені В.М. Глушкова НАН України (Київ, Україна).

Цитування: Петренко М.Г., Малахов К.С. Методологія і практика конвергенції міждисциплінарних знань у сфері цифрової охорони здоров'я. *Information Technologies and Systems*, Київ, 2025, Том 3 (3), 76–96. <https://doi.org/10.15407/intechsys.2025.03.076>

© Видавець ВД «Академперіодика» НАН України, 2025. Стаття опублікована на умовах відкритого доступу за ліцензією CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

створення ефективних дистанційних реабілітаційних сервісів. Окреслено концептуальні межі дисциплінарного, між-, мульти- та трансдисциплінарного підходів і показано, як їхня конвергенція генерує інноваційні рішення. На підтвердження теоретичних висновків виконано умовно-емпіричне дослідження у гібридному хмарному середовищі *Hybrid cloud environment for telerehabilitation*. Отримані результати свідчать про потенційну клінічну й економічну ефективність трансдисциплінарно-орієнтованих цифрових платформ, особливо для пацієнтів із обмеженим доступом до очної реабілітаційної медицини

Ключові слова: наукова картина світу, трансдисциплінарність, міждисциплінарність, конвергентні технології, цифрова охорона здоров'я, умовно-емпіричне дослідження.

Вступ

У сучасній науковій картині світу (НКС), що ґрунтується на принципах загальної еволюції, самоорганізації, коеволюції та нелінійності, передбачається, що суб'єкт — маєтєся на увазі суспільство — входить у систему, яку він прагне зрозуміти як активний компонент цієї процес-системи. Ця перспектива підкреслює динамічну та колективну роль суспільства у формуванні систем, які воно вивчає [1–3].

Відомою філософською концепцією, що виникла в цьому контексті, є трансгуманізм, який досліджує можливості та наслідки науково-технічного прогресу, зважаючи на потенційні небезпеки та переваги його застосування. На противагу цьому, ідеї постгуманізму пропонують центральну тезу, яка підкреслює взаємозалежну еволюцію людини, суспільства і природи як трьох взаємопов'язаних систем [4].

Технонаука суттєво змінила умови життя людини. Ці зміни зумовили необхідність розширення наукового світогляду, вимагаючи більш глибокого та інтенсивного проникнення в сутність законів, що керують природою і суспільством, — за межі того, що можна було досягти в рамках дисциплінарних або навіть міждисциплінарних підходів. Жан Піаже, як відомо, вважав, що «після етапу міждисциплінарних досліджень слід очікувати вищого етапу — трансдисциплінарних досліджень, які не обмежуватимуться міждисциплінарними зв'язками, а помістять ці зв'язки в глобальну систему, без жорстких кордонів між дисциплінами. Трансдисциплінарність слід розглядати як нову галузь знань, відмінну від мультидисциплінарності та міждисциплінарності» [5]. Історичний розвиток трансдисциплінарних досліджень досліджувався в різних роботах [6–9].

Метою дослідження є вивчення теоретичних і практичних аспектів трансдисциплінарних досліджень з акцентом на їх застосуванні у *Digital Health* [1]. Їх дослідження має на меті зробити посильний внесок у постійний розвиток цифрової охорони здоров'я (ЦОЗ) як інтегрованої та цілісної галузі дослідження.

Постановка проблеми

У класифікації наукових підходів корисним критерієм є ступінь повноти знань про навколишній світ. Виходячи з цього критерію, наукові підходи можна розділити на чотири основні типи: дисциплінарний (Д), міждисциплінарний (МД), мультидисциплінарний (МтД) та трансдисциплінарний (ТД).

В статті зазначені терміни розуміють у зв'язку з «розподілом» концептів і наукових дисциплін на різних онтологічних рівнях ієрархії, що передбачає різні моделі взаємодії. Такий розподіл важливий при розгляді методології ТД взаємодії, системної інтеграції знань з предметних дисциплін, формування «кластерів конвергенції» [7, 10] при реалізації трансдисциплінарних проєктів, включно з їх інформаційно-технологічною підтримкою.

До категоріального рівня належать такі ключові поняття, як Ноосфера, Об'єкт, Процес, Система, Інформація, Природа, Суспільство, Людина, Сфера знань, Наука, Наукова діяльність, Наукова картина світу, Техніка, Технологія. Такі поняття, як Філософія, Фізика, Математика, Біологія, Хімія, Медицина, Гуманітарні та Соціальні науки, Інформатика, Нанотехнології відповідають рівню доменів наукових дисциплін. Багато субдисциплін і спеціалізацій у цих галузях підпадають під рівень наукових дисциплін.

Схему категоріального та доменного рівнів знань у НКС та взаємозв'язки між відповідними концептами наведено на рис. 1. Повний опис усіх елементів виходить за рамки цього дослідження і розглянуто в [9].

Дисциплінарний підхід дає змогу науці розвиватися в межах конкретних предметних галузей, розділяючи світ на окремі сфери. Однак, коли проблема виходить за межі однієї дисципліни, її часто вважають такою, що перебуває на перетині наукових дисциплін. У міру розвитку Д підходу виникає природна невизначеність: з одного боку, він сприяє накопиченню дисциплінарного знання, а з іншого — досягає природної межі в повному розумінні навколишнього світу. Ця ситуація свідчить про те, що коли стає неможливим вирішити проблеми в межах дисциплінарного напрямку, сфера застосування дисциплінарної методології може бути розширена [6]. Таке розширення призвело до появи МД та МтД наукових підходів, які формують наступні рівні в класифікації наукових підходів. Розвиток цих підходів трансформував метафору «стику дисциплін» у сформовані міждисциплінарні та мультидисциплінарні напрямки, кожен з яких має свої методи вирішення дослідницьких проблем [6].

Міждисциплінарність передбачає інтеграцію кількох наукових дисциплін, причому одна з них часто відіграє провідну роль. Результати міждисциплінарних досліджень зазвичай інтерпретуються в рамках провідної дисципліни. Ключовою особливістю міждисциплінарного підходу є сприяння прямому перенесенню методів дослі-

дження з однієї дисципліни в іншу на основі виявлення схожості між предметними областями. МД підхід в першу чергу спрямований на вирішення конкретних дисциплінарних проблем, особливо коли виникають концептуальні та методологічні труднощі в межах певної дисципліни.

Синергетична парадигма, як відгалуження МД підходу в ієрархії знань, посідає особливе місце. З одного боку, вона апелює до цілісного уявлення, системно визначаючи ефекти взаємодії між об'єктами, процесами та суб'єктами. З іншого боку, вона фокусується на нелінійностях, нестабільності та появи атракторів, які в кінцевому підсумку змінюють багаторівневу організацію та поведінку системи. Ця парадигма виражається через формальні моделі самоорганізації і відіграє ключову роль у відтворенні НКС, особливо під час переходу до ТД підходу та реалізації парадигми глобального еволюціонізму. НКС можна подати як ТД онтологію [7], що включає в себе не тільки онтології окремих дисциплін, але й їхні методи. ТД підхід дає змогу створити єдину методологію аналізу та синтезу, інтегруючи її в загальну НКС. Виклики та потенціал синергетики обговорюються в [7, 8].

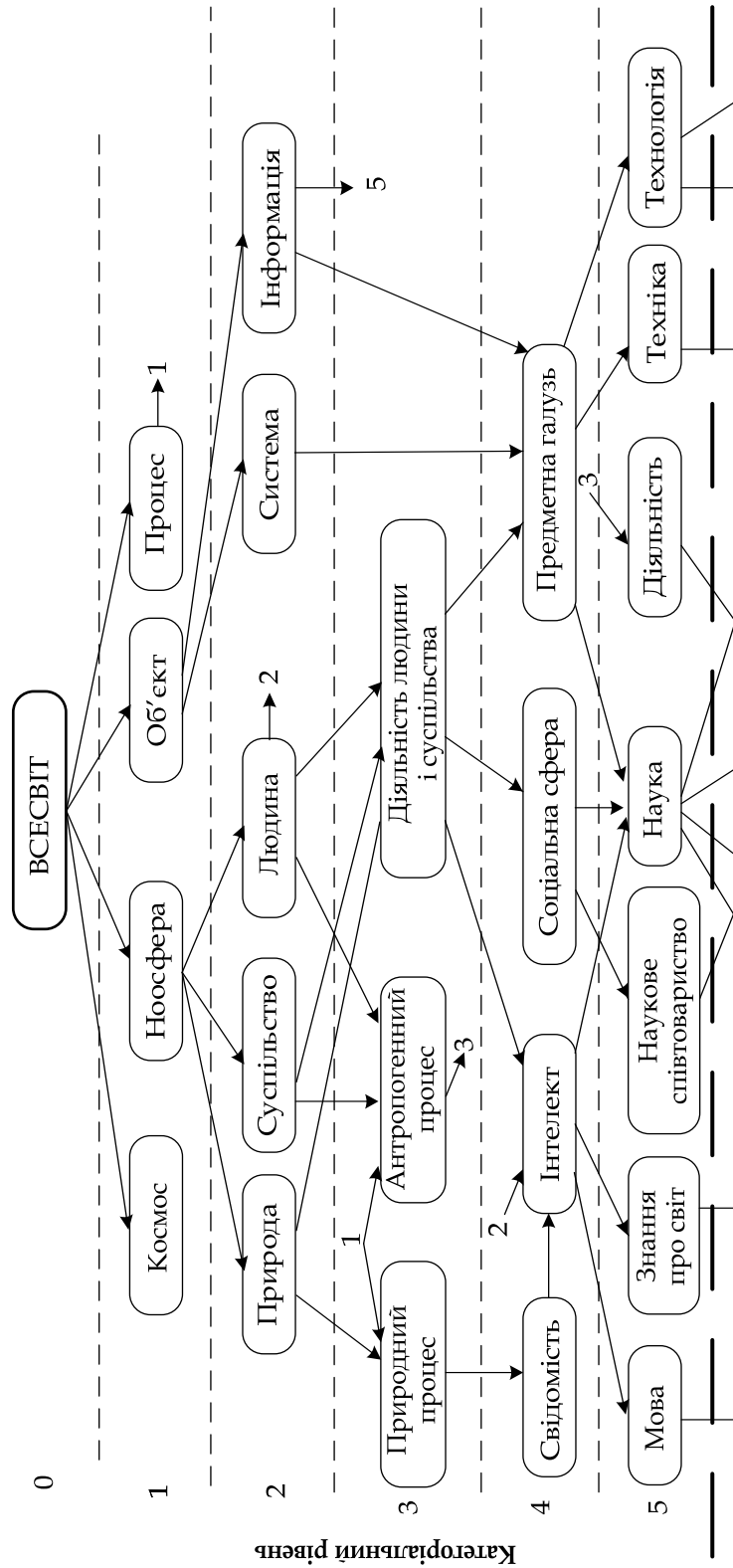
В рамках *мультидисциплінарного підходу* дослідники розробляють узагальнену картину предмета дослідження, причому всі дисциплінарні методи і моделі є однаково вагомими. Накопичення результатів міждисциплінарних досліджень у схожих галузях дисциплінарного знання призводить до появи нових міждисциплінарних дисциплін, таких як фізико-хімічна біологія, екологія тощо. Мультидисциплінарний підхід знайшов практичне застосування, зокрема в роботі експертних груп.

Мультидисциплінарність не передбачає перенесення методів дослідження між дисциплінами, натомість кожна дисципліна зберігає свій предметний фокус.

ТД системний підхід використовує знання, отримані та накопичені дисциплінарними, міждисциплінарними та мультидисциплінарними підходами. ТД має на меті забезпечити координацію та інтеграцію дисциплінарних знань на основі єдиного аксіоматичного підходу (загальних ТД систем). Таке бачення трансдисциплінарності було вперше концептуалізовано Ж. Піаже та Е. Янчем [5, 6].

Трансдисциплінарність є дослідницькою стратегією, яка виходить за межі дисциплінарних кордонів, сприяючи формуванню цілісного погляду (пріоритет цілого над його частинами). У вузькому сенсі ТД передбачає інтеграцію різних форм і методів дослідження, включно зі спеціалізованими методами наукового пошуку, для вирішення складних наукових проблем. У ширшому розумінні ТД означає єдність знань, що виходять за межі конкретних дисциплін.

Трансформація структури науки, пов'язана з переходом від дисциплінарних до ТД досліджень, характеризується кількома ознаками постнеокласичного етапу: зміною характеру наукової діяльності внаслідок прогресу в отриманні та зберіганні знань (наприклад,



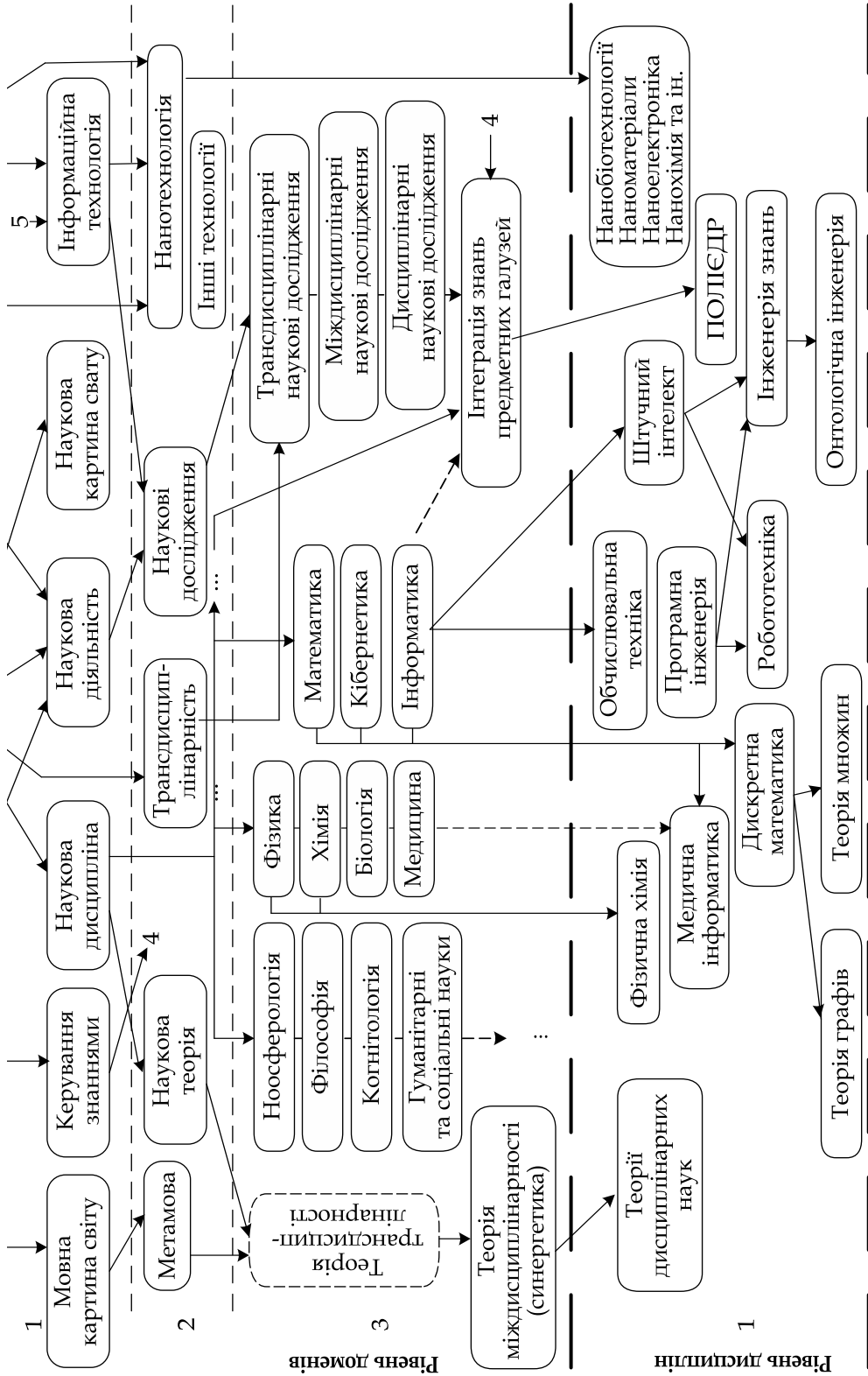


Рис. 1. Схема концептів категоріального та доменного рівнів знань у НКС

комп'ютеризація науки, зрощення науки з промисловим виробництвом), зростанням значення економічних і соціально-політичних чинників і цілей, а також зміною самого об'єкта дослідження – відкритих систем, що саморозвиваються (наприклад, об'єкти біотехнологій, екологічні системи, біосфера тощо).

ТД дослідження, захоплюючи прикордонні зони (демаркаційні лінії) наукових дисциплін, інтегрують їх сутнісні основи, формуючи «кластери конвергенції» [7, 8]. Ці кластери сприяють потужній синергетичній взаємодії завдяки взаємопроникненню парадигм і конкретних поточних результатів кожної залученої дисципліни. Ця взаємодія відображає цілісність реального світу, слугуючи одночасно і стимулом, і гарантією успіху для ТД досліджень та пов'язаних з ними практичних проєктів, нетривіальність і значущість яких підкреслюється їхніми результатами.

Основи методології

Єдність і системна складність світу як об'єкта наукового дослідження свідчать про те, що поряд з процесом диференціації не менш важливо розглядати інтеграцію наукових дисциплін і відповідних технологій. Цей інтегративний процес узгоджується з фундаментальною парадигмою еволюційної теорії, запропонованої акад. В.І. Вернадським [11]. Хоча інтеграція наукових дисциплін почалася дещо спонтанно та інтуїтивно, зараз вона стала більш свідомим і цілеспрямованим процесом. Цей процес дедалі більше керується еволюційною теорією, спрямованою на створення єдиного масиву загальних знань (ЗЗ) [12]. Суть цього прагнення полягає в побудові загального знання на основі трансдисциплінарної концепції наукового розвитку.

Початковий етап цього інтеграційного процесу можна охарактеризувати як етап кластеризації. Навіть поверхневий аналіз феномену кластеризації показує, що в його основі лежить глибока взаємодія методів, інструментів і можливостей компонентів кластера. Синтез цих компонентів породжує синергетичні ефекти за рахунок інтеграції їх функційних властивостей, тим самим відкриваючи широкі перспективи для створення раніше невідомих наукових теорій, нової техніки і технологій. Процес інтеграції, що триває, ставить численні питання, відповіді на які допоможуть окреслити нові, перспективні шляхи еволюції знань і науки в цілому, відповідно до ноосферної теорії В.І. Вернадського [10, 11].

Одним із завдань на цьому еволюційному шляху є аналіз структури знань на концептуальному рівні. ЗЗ, як і кожна з їх дисциплін, можна подати як всеосяжний набір понять-термінів, що становлять онтологічну основу для опису знань, як формальних, так і неформальних. Кожен термін у цій структурі має своє загальноприйняте визначення, яке пояснюється за допомогою термінів або понять нижчого рівня. Розроблено технологічний багаторівневий онтологічний

опис як для ЗЗ, так і для їх окремих дисциплін, розділів, теорій тощо, що дає змогу формально подати знання за допомоги уніфікованого інструментарію онтологічної інженерії [13–15]. Цей інструментарій відкриває широкі перспективи для розвитку когнітивних технологій та їх продуктивного застосування. Нижче наведено узагальнений формальний опис сукупності предметних знань

$$O^{33} = \sum_{i=1}^n O_i^{D3},$$

де $D3$ – дисциплінарні знання i -ої предметної галузі, які входять в загальну ТД онтологію, $i = \overline{1, N}$, N – множина предметних галузей.

Наразі спостерігається тенденція до інтенсифікації наукових досліджень як на перетині різних предметних дисциплін (МД дослідження), так і в межах конвергентних кластерів (ТД-дослідження). Для підтримки цих досліджень визначальними є кілька ключових факторів, серед яких побудова інформаційних систем, орієнтованих на знання, вдосконалення процесів організації досліджень, вдосконалення методів та інструментів онтологічного аналізу об'єктів природної мови з використанням генеративних мовних моделей [16, 17] для вилучення знань, а також прикладне використання онтологій, мета-онтологій та систем інтеграції знань [18] у трансдисциплінарних кластерах конвергенції.

Стрімкий розвиток конвергентних технологій має потенціал для значного покращення як людської діяльності, так і національних економік. Яскравими прикладами таких досягнень є підвищення ефективності роботи та навчання, розширення сенсорних та когнітивних можливостей, принципово нові виробничі процеси, революційні зміни в охороні здоров'я, підвищення індивідуальної та групової ефективності, передові методи комунікації (включно з взаємодією «мозок-мозок»), вдосконалення людино-машинних інтерфейсів [19] (наприклад, нейроморфної інженерії для промислового та особистого використання), розширення можливостей людини в оборонних цілях, досягнення сталого розвитку за допомоги інструментів NBIC кластера (N – Nano, B – Bio, I – Info, C – Cogno) [20]. Крім того, ці технології є перспективними для подолання фізичного та когнітивного занепаду, пов'язаного зі старінням.

Яскравим прикладом технонауки є описаний NBIC кластер конвергенції (Roco, 2016), де інформатика виступає як системоутворюючою, так і комп'ютерно-технологічною складовою. Основними проривними напрямками в рамках цих кластерів є розмивання меж між живими і неживими системами, досягнення в наноробототехніці з численними застосуваннями, а також розвиток глобальних суперкомп'ютерних агломерацій з розвиненими можливостями штучного інтелекту (ШІ). До цього можна додати єдину розподілену систему ТД знань, що є глобальною комунікативною версією загальної НКС і

є наступним етапом в еволюції наявного інтернету та семантичного павутиння.

Можлива схема прогнозування та цілеспрямованого формування перспективних кластерів конвергенції, а також їх синергетичної взаємодії та результуючої сукупної ефективності для людства потребує подальшого вивчення в окремому дослідженні.

Застосування трансдисциплінарного підходу до телереабілітації

Принципи ТД досліджень, особливо ті, що стосуються інтеграції та конвергенції знань з різних дисциплін, мають важливе значення для сфери цифрової охорони здоров'я. ЦОЗ за своєю суттю працює на перетині декількох областей, включно з медициною, фізичною терапією, цифровим здоров'ям, інформатикою та інженерією. На рис. 2 показано схему зв'язків між когнітивними технологіями, різними галузями науки і ЦОЗ та телереабілітацією.

Застосування трансдисциплінарного підходу у цьому контексті уможливорює розробку цілісних реабілітаційних програм, які є більш персоналізованими, адаптивними та чутливими до унікальних потреб пацієнтів. Зокрема, можна зазначити ось що.

1. Кластеризація дисциплін і технологій. У ЦОЗ етап кластеризації передбачає інтеграцію медичних знань, методів фізичної терапії та цифрових технологій, таких як телекомунікаційні платформи, переносні датчики і штучний інтелект. Такий синтез створює нові можливості для надання дистанційних реабілітаційних послуг, які є одночасно ефективними та результативними. Наприклад, інтеграція аналітики, керованої ШІ з даними про пацієнта в реальному часі [21] може забезпечити індивідуальні реабілітаційні вправи та динамічно коригувати протоколи терапії залежно від прогресу пацієнта.

2. Синергетичні ефекти та інновації. Синергетичний ефект від інтеграції різноманітних дисциплін може призвести до інновацій у реабілітаційних практиках. Наприклад, поєднання знань з нейробиології з передовою робототехнікою і машинним навчанням може призвести до розроблення інтелектуальних протезів [22, 23], які не тільки підвищують мобільність, але й адаптуються до неврологічних сигналів користувача, тим самим покращуючи результати в умовах ЦОЗ.

3. Онтологічна інженерія в цифровій охороні здоров'я. Застосування технологічних багаторівневих онтологічних описів у ЦОЗ [24–26], дає змогу формально подавати дані про пацієнтів [27], протоколи лікування та результати реабілітації. Ці структуровані знання можуть бути використані для створення комплексних, інтероперабельних інформаційних систем [12], які полегшують комунікацію та обмін даними між різними постачальниками медичних послуг, підвищуючи безперервність надання допомоги пацієнтам, які проходять реабілітацію.

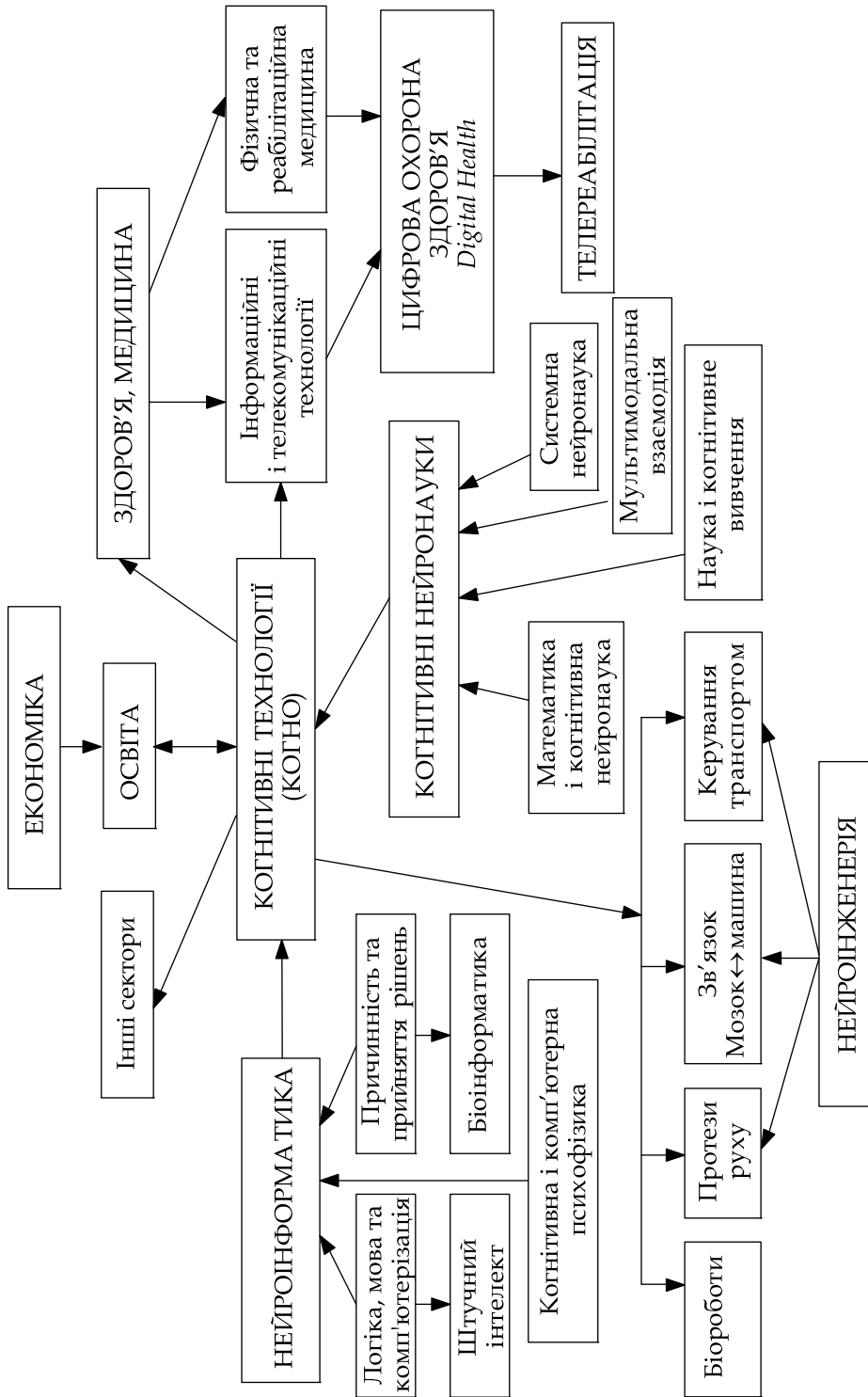


Рис. 2. Зв'язок між когнітивними технологіями та різними галузями науки і техніки

4. Кластери конвергенції на практиці. NBIC-кластер конвергенції в ЦОЗ сприяє розвитку інтегрованих систем, в яких нанотехнології, біотехнології, інформаційні технології та когнітивні науки об'єднуються для створення передових засобів реабілітації. Наприклад, інтерфейси «мозок-комп'ютер» [19] можуть бути використані в ЦОЗ для допомоги пацієнтам з важкими порушеннями рухливості, даючи змогу їм керувати протезами або спілкуватися з допомогою думки, що значно покращує якість їхнього життя.

5. Виклики та майбутні напрями. Хоча застосування трансдисциплінарного підходу до ЦОЗ є багатообіцяльним, воно також пов'язане з певними проблемами, які потребують вирішення. Серед них можливо відзначити необхідність стандартизованих протоколів для інтеграції даних [28], забезпечення конфіденційності та безпеки даних пацієнтів (з використанням сучасних технологій, таких як багаторівневе розпізнавання облич [29, 30], а також розроблення масштабованих моделей, які можуть бути застосовані в різних медичних установах. Майбутні дослідження повинні бути зосереджені на вирішенні цих проблем, продовжуючи досліджувати нові кордони інтеграції технологій і дисциплін в рамках ЦОЗ.

ТД дослідження являють якісно новий етап в інтеграції науки і суспільства. Щоб повністю реалізувати свій потенціал, особливо в таких галузях як ЦОЗ, науковій спільноті все ще необхідно розробити і розвинути кілька ключових компонентів:

1. Загальну НКС, що охоплює предметні дисципліни та відповідну глобальну мережу ТД знань.

2. Метатеорію та метамову [31, 32] опису ТД знань.

3. Системологію ТД взаємодії, яка містить образний понятійний апарат і моделі, здатні охопити всі фактори, що формують і впливають на складні проблеми, а також виявити і врахувати механізми, з допомогою яких ці впливи відбуваються.

4. Метод (або сукупність методів) системного дослідження, що забезпечує доступ до всієї дисциплінарної інформації та її аналіз, зрозумілий і доступний фахівцям з будь-якої наукової дисципліни.

5. Перспективні та самодостатні кластери конвергенції, які складуть ядро шостого технологічного укладу.

6. Експериментальні методи, що дозволяють досліджувати багатофакторні впливи на об'єкти пізнання та оцінювати їх результати. Сюди відносяться різні інструменти інформаційних технологій, покликані підвищити ефективність і прискорити отримання результатів.

7. Підходи до постановки та розв'язання складних багатофакторних задач у науці, техніці та технологіях.

Ці розробки стануть ключовими в просуванні інтеграції знань і досягненні цілей трансдисциплінарних досліджень в майбутньому, зокрема в сфері ЦОЗ, що розвивається і має великий вплив.

Інтеграція та її ключові дисципліни для цифрової охорони здоров'я

ЦОЗ – це галузь, яка за своєю суттю потребує інтеграції кількох дисциплін для надання ефективної, орієнтованої на пацієнта допомоги дистанційно. Оскільки охорона здоров'я продовжує розвиватися разом із технологічним прогресом, інтеграція дисциплін у процес ЦОЗ (і телереабілітації зокрема) стає все більш важливою. Нижче коротко розглядаються основні дисципліни, пов'язані з ЦОЗ, і те, як їх перетин та інтеграція сприяють розробці та впровадженню міждисциплінарних підходів у цій галузі.

Медицина та фізична реабілітація. ЦОЗ глибоко інкорпорована в принципах медицини та фізичної реабілітації. Лікарі та фізіотерапевти несуть відповідальність за діагностику, розроблення та впровадження протоколів реабілітації, адаптованих до індивідуальних потреб пацієнтів. Ці медичні працівники повинні гарантувати, що терапевтичне втручання, яке здійснюється дистанційно, є таким же ефективним, як і особисте лікування, враховуючи нюанси стану кожного пацієнта.

ЦОЗ та медична інформатика. Поява ЦОЗ зробила революцію в медицині, уможлививши дистанційне надання медичної допомоги за допомогою телекомунікаційних технологій [33–35]. Інформатика охорони здоров'я відіграє важливу роль в управлінні даними пацієнтів, сприянні телеконсультаціям і забезпеченні повної інтеграції електронних медичних записів (ЕМЗ) із протоколами реабілітації. Ці технології підтримують безперервний моніторинг, залучення пацієнтів та ефективне керування великими обсягами даних, що є критично важливим для успіху програм ЦОЗ.

Комп'ютерні науки та інженерія. Розроблення та застосування цифрових платформ переносних мобільних додатків для здоров'я та технологій, обумовлені досягненнями в комп'ютерних науках та техніці. Ці дисципліни сприяють створенню зручних інтерфейсів, безпечних протоколів передачі даних та інтеграції алгоритмів машинного навчання в системі ЦОЗ. Інженери та комп'ютерні фахівці тісно співпрацюють із постачальниками медичних послуг, щоб переконатися, що ці технології не тільки функційні, але й підвищують якість медичної допомоги, яка надається віддалено.

Когнітивні та поведінкові науки. Когнітивні та поведінкові науки є невід'ємною частиною розуміння поведінки пацієнтів, мотивації та дотримання протоколів реабілітації. Ці дисципліни інформують про розроблення втручань, які є психологічно підтримувальними та заохочують постійне залучення пацієнтів. Застосовуючи принципи когнітивно-поведінкової терапії та інші психологічні рамки, програми ЦОЗ можна адаптувати до психічних та емоційних аспектів відновлення, які часто є критичними для успішних результатів реабілітації.

Гуманітарні та соціальні науки. Гуманітарні та соціальні науки пропонують важливу інформацію про соціальні детермінанти здоров'я, культурні міркування та етичні наслідки ЦОЗ [36]. Ці дисципліни гарантують, що програми ЦОЗ є інклюзивними, культурними та етичними. Вони також розглядають ширший суспільний вплив дистанційної реабілітації, включно з доступом до догляду, справедливості і цифровим розривом, який може суттєво вплинути на ефективність системи ЦОЗ та її послуг.

Роль міждисциплінарної та трансдисциплінарної інтеграції

Інтеграція наведених дисциплін у ЦОЗ відбувається як на рівнях МД, так і на рівнях ТД.

Рівень МД інтеграції. В цифровій охороні здоров'я МД інтеграція передбачає співпрацю різних дисциплін, де кожна дисципліна зберігає свої специфічні методології та перспективи. Наприклад, програма реабілітації може охоплювати фізіотерапевта, який працює з інженером над розробленням переносного пристрою, що відстежує рухи пацієнта (інтернет медичних речей). Дані, зібрані пристроєм, потім аналізують фахівці з медичної інформатики для коригування протоколу реабілітації, а вчені-когнітологи можуть оцінити психологічну реакцію пацієнта на лікування. Кожна дисципліна вносить свій досвід, але фокус залишається на вирішенні певного аспекту процесу реабілітації.

Рівень ТД інтеграції. ТД інтеграція йде ще далі, виходячи за рамки традиційних дисциплін, щоб створити єдину структуру для ЦОЗ. У такому підході відмінності між дисциплінами стають менш вираженими, суттєво проявляються спільні зусилля для інтеграції знань, теорій і методів, а також для цілісного вирішення складних проблем реабілітації. Наприклад, розроблення комплексної програми ЦОЗ, яка охоплює адаптивну терапію на основі штучного інтелекту, потребує повної інтеграції медичних знань, цифрових платформ охорони здоров'я, алгоритмів ШІ та розуміння поведінки пацієнтів. Цей цілісний підхід дає змогу створювати персоналізовані плани реабілітації, які постійно оптимізуються на основі даних у реальному часі та відгуків пацієнтів [19].

Інтеграція трансдисциплінарних знань у ЦОЗ, яка підтверджена симуляційним експериментом (детально описаним у наступному розділі), відкриває низку практичних переваг. По-перше, персоналізовані плани реабілітації, які формуються на основі медичних, когнітивних та технічних даних, адаптуються в реальному часі, що підвищує залученість та дотримання пацієнтів. По-друге, цифрові платформи долають географічні бар'єри, полегшуючи доступ пацієнтів із віддалених регіонів і зменшуючи потребу у фізичних візитах. По-третє, міждисциплінарна співпраця клініцистів, інженерів та фа-

хівців з даних забезпечує безперервне вдосконалення інструментів телереабілітації та сприяє спільному прийняттю рішень. Нарешті, автоматизований моніторинг і аналітика підвищують економічну ефективність, оптимізують використання ресурсів і підтримують доказову практику.

Умовно-емпіричне дослідження оцінки ефективності ТД підходу при розробленні персоналізованих програм телереабілітації

У сфері ЦОЗ та біомедичної інформатики український термін «умовно-емпіричне дослідження» (УЕД) стосується робіт, у яких методи комп'ютерного моделювання або синтетичні дані поєднуються з емпіричними підходами. В англійській літературі [37–40] такі роботи описують за допомогою термінів: *in silico experiment / study, simulation-based evaluation, synthetic patient data evaluation, scenario-based evaluation* чи *model-driven evaluation*. Кожен із них має власний контекст і акцент [37, 38, 40]:

- *in silico trial* – повністю комп'ютерно змодельоване клінічне випробування; поширене у обчислювальній фармакології та оцінюванні медичних виробів.

- *Simulation study / simulated experiment* – контрольований експеримент із віртуальними пацієнтами (наприклад, сценарій катастрофи з акторами або манекенами) для тестування цифрових втручань.

- *Synthetic patient data* – використання віртуальних електронних медичних записів або цифрових двійників (англ. *Digital twins*) пацієнтів для розроблення та валідації алгоритмів без ризику порушення конфіденційності.

- *Scenario-based evaluation* – застосовують у дослідженнях продуктивності та навчальних симуляціях; система випробується за заздалегідь підготовленими клінічними сценаріями.

- *Model-driven evaluation* – базується на формальних обчислювальних моделях (зокрема *digital-twin models*) й прогнозує поведінку системи до її фактичного впровадження.

У кожному випадку дослідження є емпіричним лише умовно – воно спирається на комп'ютерні або сценарні проксі, а не на безпосереднє спостереження за реальними пацієнтами. Отже, *умовно-емпіричне дослідження* – це оцінювання клінічних інновацій з допомогою *simulation, synthetic data* або *model-based scenarios* у ситуаціях, коли традиційні випробування є надто складними чи затратними; у такому форматі вивчення спирається на комп'ютерні або сценарні проксі, а не на безпосереднє спостереження за реальними пацієнтами, що робить його оптимальним підходом для реалізації нашого експерименту.

Ми проводили УЕД з метою оцінювання ефективності ТД підходу під час розроблення персоналізованих програм телереабілітації.

Умови УЕД. 60 пацієнтів після інсульту (із залишковими функційними порушеннями) випадковим чином розподілені на дві групи – контрольну та експериментальну (по 30 осіб у кожній). Контрольна група отримувала стандартну програму телереабілітації без ТД-компонентів, тоді як експериментальна – індивідуалізовану програму з використанням ТД-моделі. Програма експериментальної групи передбачала попередню оцінку когнітивного статусу пацієнта та адаптацію вправ за допомоги алгоритмів, методів та засобів гібридної хмарної платформи HCET (англ. *Hybrid Cloud Environment for Telerehabilitation*) [21], а також застосування мультисенсорного біологічного зворотного зв'язку (БЗЗ) [41] (елементів технологій віртуальної та доповненої реальності, сенсорів руху, тощо). Враховано, що такі системи, як HCET можуть аналізувати дані пацієнта у реальному часі, персоналізуючи терапію і коригуючи план занять [42], а віддалений моніторинг дає змогу контролювати дотримання пацієнтом вправ та оперативно реагувати на потребу змінення програм [21, 42].

Учасники УЕД. Вибірка дослідження складалася із 60 умовних пацієнтів старшого віку (середній вік 65 років) зі стійкими моторними й когнітивними порушеннями після ішемічного інсульту. Всі пацієнти мали подібний ступінь початкового дисфункціонування. Експеримент моделюється, як рандомізоване контрольоване дослідження з попереднім і кінцевим вимірами. *Зазначимо, що наведені результати є умовною (змодельованою) ілюстрацією гіпотетичного експерименту.*

Метрики ефективності. Оцінку результатів здійснювали за такими показниками ефективності реабілітації:

Приріст за шкалою FIM (англ. *Functional Independence Measure*) – сумарне підвищення балів незалежності пацієнта, що відображає зміну в потребі допомоги при виконанні повсякденних дій.

Рівень дотримання плану реабілітації (адерентність, англ. *adherence rate*) – відсоток виконаних пацієнтом призначених сеансів чи вправ. Дотримання лікування є ключовим чинником успіху реабілітації, і попередні дослідження відзначали його недостатню оптимальність при стандартних підходах.

Кількість завершених сеансів (запланованих до певного терміну) – практичне продовження метрики адерентності.

Зниження інтенсивності болю за візуально-аналоговою шкалою (ВАШ) – різниця між початковими й кінцевими балами за 10-бальною шкалою болю (чим більший приріст, тим більше зниження болю). ВАШ є загальноприйнятим інструментом для оцінювання болю у пацієнтів з неврологічними порушеннями.

Для кожного показника розраховано середні значення (M) і стандартні відхилення (SD) у контрольній та експериментальній групах. Статистичний аналіз – порівняння двох незалежних вибірок – здійснювався з допомогою непарного *t*-критерію Стьюдента ($\alpha = 0,05$).

Таблиця 1 узагальнює середні результати за вибраними метриками. Як видно, експериментальна група продемонструвала вищі показники за всіма метриками порівняно з контрольною групою.

Опис результатів. Приріст за *FIM* у контрольній групі становив $11,1 \pm 4,5$ балів, тоді як у експериментальній – $17,4 \pm 4,7$. Різниця між групами є статистично значущою ($t = 5,36, p < 0,001$), що свідчить про більший функційний приріст пацієнтів ТД-програми. Відсоток дотримання реабілітаційної програми в контрольній групі склав у середньому $75,1 \pm 9,9\%$, а в експериментальній – $84,7 \pm 6,9\%$ ($t = 4,34, p < 0,001$). Тобто пацієнти з ТД-підтримкою охочіше виконували призначені завдання. Аналогічна тенденція спостерігалася за кількістю завершених сеансів: середнє число сеансів в експериментальній групі ($18,4 \pm 1,5$ з 20 запланованих) значно перевищувало показник контрольної групи ($14,6 \pm 3,0$) ($t = 6,10, p < 0,001$). Щодо зниження болю, то експериментальна група мала середнє зниження $4,1 \pm 1,6$ бала проти $2,1 \pm 1,6$ у контрольній ($t = 4,87, p < 0,001$), що теж підтверджує кращий результат на ТД-програмі.

Ці результати узгоджуються з теоретичними уявленнями про те, що системи телереабілітації на кшталт *HCET*, здатні забезпечувати адаптивний (пацієнт-центричний) під пацієнта підхід. Так, попередні дослідження показали, що ТД-підсилена телереабілітація дає змогу віддалено відстежувати прогрес пацієнта і вносити зміни в програму в реальному часі, що значно покращує результативність втручання [21, 42]. Наведені умовні дані УЕД ілюструють, що ТД підхід (індивідуалізована програма з когнітивним аналізом, та мультисенсорним БЗЗ) потенційно може суттєво підвищити ефективність телереабілітації.

Змодельований експеримент показує, що застосування ТД-моделі в телереабілітації дає вищі середні значення поліпшення функційного стану, кращу адерентність до програми та більшу кількість виконаних сеансів порівняно зі звичайною дистанційною реабілітацією. Усі наведені відмінності є статистично значущими. Отримані дані – умовні й ілюстративні – свідчать про потенційну перевагу ТД підходу: він дозволяє максимально враховувати індивідуальні

Таблиця 1. Порівняльні результати зміни ключових показників реабілітації в контрольній та експериментальній групах (умовні дані УЕД)

Метрика	Контрольна група (M ± SD)	Експериментальна група (M ± SD)	t-критерій (p)
Приріст <i>FIM</i> , бали	11,1 ± 4,5	17,4 ± 4,7	$t = 5,36, p < 0,001$
Рівень дотримання програми, %	75,1 ± 9,9	84,7 ± 6,9	$t = 4,34, p < 0,001$
Завершених сеансів	14,6 ± 3,0	18,4 ± 1,5	$t = 6,10, p < 0,001$
ВАШ	2,1 ± 1,6	4,1 ± 1,6	$t = 4,87, p < 0,001$

потреби пацієнтів та оперативно коригувати план реабілітації. Проте слід мати на увазі гіпотетичний характер експерименту, результати потребують верифікації на реальних клінічних дослідженнях.

Наведене УЕД подано у стислому виді, оскільки його повний опис виходить за межі цієї статті. Докладна інформація про організацію та симуляцію експерименту в хмарному середовищі *HCET*, разом із повним набором даних, доступна в [43].

Виклики та пріоритети подальших досліджень

Результати змодельованого експерименту у *HCET* підтвердили ефективність ТД персоналізованих програм. Наступним кроком має стати тестування комбінованих методик, зокрема інтеграції *VR / AR* із когнітивно-поведінковою терапією та нейромодуляцією, що потенційно підвищить мотивацію пацієнтів і стимулюватиме нейропластичність.

Масове впровадження *Digital health* та телереабілітації потребує досліджень, які гарантуватимуть рівний доступ до сервісів, незалежно від соціально-економічного статусу, та забезпечуватимуть збереження людяності взаємодії пацієнта і фахівця у цифровому середовищі.

Подальший розвиток *5G / 6G*, *edge*-обчислень і блокчейн-реєстрів відкриває можливості для мінімізації затримок, прозорого керування даними та безпечної інтеграції між клініками; ефективність цих технологій слід перевірити в умовах реального навантаження *HCET*.

Конвергенція міждисциплінарних знань, підсилена хмарними архітектурами та елементами штучного інтелекту, залишається ключовим чинником на шляху до перетворення персоналізованої ЦОЗ (телереабілітації зокрема) на стандарт практики, підтверджений етичними, соціальними й технологічними критеріями якості.

Висновки

Трансдисциплінарний підхід продемонстрував як методологічну обґрунтованість, так і практичну ефективність для ЦОЗ. УЕД у гібридному хмарному середовищі *HCET* підтвердив суттєві клінічні переваги персоналізованої ШІ-адаптованої телереабілітації: приріст за шкалою *FIM* зріс на 57 %, адерентність – на 13 %, кількість завершених сеансів – на 26 %, а інтенсивність болю зменшилася удвічі порівняно зі стандартною програмою.

Отримані показники свідчать, що ТД-орієнтовані цифрові платформи здатні забезпечити доступний, персоналізований і економічно доцільний догляд, особливо для пацієнтів, які не мають регулярного доступу до очної реабілітації.

Наступні кроки передбачають перспективні клінічні випробування для довгострокової валідації результатів, стандартизацію протоколів інтеграції та детальне опрацювання етичних і соціальних наслідків масштабного впровадження таких систем.

ЛІТЕРАТУРА/REFERENCES

1. Malakhov K.S. Insight into the Digital Health System of Ukraine (eHealth): Trends, Definitions, Standards, and Legislative Revisions. *International Journal of Telerehabilitation*, 2023, Vol. 15 (2). <https://doi.org/10.5195/ijt.2023.6599>
2. Palagin O.V. Information technology tools for controlled evolution. *Problems of Control and Informatics*, 2021, Vol. 66 (5), 104–123. <https://doi.org/10.34229/1028-0979-2021-5-9>
3. Shyrovkov V. Evolution as universal natural law (prolegomena to the future general evolution theory). *Bionics of Intelligence*, 2017, Vol. 88 (1), 3–14. [https://doi.org/10.30837/bi.2018.1\(90\).01](https://doi.org/10.30837/bi.2018.1(90).01)
4. Bostrom N. A History of Transhumanist Thought. *Journal of Evolution and Technology*, 2005, Vol. 14(1), 1–25. URL: <http://jetpress.org/volume14/freitas.html>
5. Piaget J. L'Épistémologie des Relations Interdisciplinaires. In R. Schwarz (Ed.). *Wissenschaft als interdisziplinäres Problem, Teil 1*. De Gruyter, 1974, 154–172. <https://doi.org/10.1515/9783112415504-006>
6. Jantsch E. *Technological Planning and Social Futures*. Littlehampton Book Services Ltd, 1972. URL: <https://archive.org/details/technologicalpla0000jant>
7. Palagin O.V. Transdisciplinarity Problems and the Role of Informatics. *Cybernetics and Systems Analysis*, 2013, Vol. 49 (5), 643–651. <https://doi.org/10.1007/s10559-013-9551-y>
8. Palagin O.V., Petrenko M.G., Kryvyi S., Boyko M., Malakhov K.S. *Ontology-Driven Processing of Transdisciplinary Domain Knowledge*. Iowa State University Digital Press, 2023. <https://doi.org/10.31274/isudp.2023.140>
9. Palagin O.V., Petrenko M.G. Methodological Foundations for Development, Formation and IT-support of Transdisciplinary Research. *Journal of Automation and Information Sciences*, 2018, Vol. 50 (10), 1–17. <https://doi.org/10.1615/JAutomatInfScien.v50.i10.10>
10. Palagin O.V., Kurgaev O.P., Shevchenko A.I. The Noosphere Paradigm of the Development of Science and Artificial Intelligence. *Cybernetics and Systems Analysis*, 2017, Vol. 53 (4), 503–511. <https://doi.org/10.1007/s10559-017-9952-4>
11. Vernadsky V.I. The biosphere and the noosphere. *American Scientist*, 1945, Vol. 33 (1), 1–12. URL: https://monoskop.org/images/5/59/Vernadsky_WI_1945_The_Biosphere_and_the_Noosphere.pdf
12. Palagin O.V., Petrenko M.G., Malakhov K.S. Challenges and Role of Ontology Engineering in Creating the Knowledge Industry: A Research-Related Design Perspective. *Cybernetics and Systems Analysis*, 2024, Vol. 60 (4), 633–645. <https://doi.org/10.1007/s10559-024-00702-6>
13. Malakhov K.S., Petrenko M.G., Cohn E. Developing an ontology-based system for semantic processing of scientific digital libraries. *South African Computer Journal*, 2023, Vol. 35 (1), 19–36. <https://doi.org/10.18489/sacj.v35i1.1219>
14. Palagin O.V., Petrenko M.G., Velychko V.Yu., Malakhov K.S. Development of formal models, algorithms, procedures, engineering and functioning of the software system “Instrumental complex for ontological engineering purpose.” *CEUR Workshop Proceedings*, 2014, Vol. 1843, 221–232. URL: <http://ceur-ws.org/Vol-1843/221-232.pdf>
15. Petrenko M.G., Cohn E., Shchurov O., Malakhov K.S. Ontology-Driven Computer Systems: Elementary Senses in Domain Knowledge Processing. *South African Computer Journal*, 2023, Vol. 35 (2), 127–144. <https://doi.org/10.18489/sacj.v35i2.17445>
16. Palagin O.V., Kaverinskiy V.V., Malakhov K.S., Petrenko M.G. Fundamentals of the Integrated Use of Neural Network and Ontolinguistic Paradigms: A Comprehensive Approach. *Cybernetics and Systems Analysis*, 2024, Vol. 60 (1), 111–123. <https://doi.org/10.1007/s10559-024-00652-z>
17. Ramesh S. *Will LLMs make Structured Healthcare Data Obsolete?* (18 Oct. 2023). URL: <https://www.youtube.com/watch?v=KCKYdR7Otg4>

18. Palagin O.V. An Ontological Conception of Informatization of Scientific Investigations. *Cybernetics and Systems Analysis*, 2016, Vol. 52 (1), 1–7. <https://doi.org/10.1007/s10559-016-9793-6>
19. Kurgaev O.P., Palagin O.V. Rehabilitation According to the Biological Feedback. *11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, 2021, 1170–1175. <https://doi.org/10.1109/IDAACS53288.2021.9660953>
20. Roco M.C. NBIC. In Bainbridge W.S., Roco M.C. (Eds.), *Handbook of Science and Technology Convergence*, 209–226. Springer International Publishing. https://doi.org/10.1007/978-3-319-07052-0_16
21. Malakhov K.S. Innovative Hybrid Cloud Solutions For Physical Medicine and Telerehabilitation Research. *International Journal of Telerehabilitation*, 2024, Vol. 16 (1), Article 1. <https://doi.org/10.5195/ijt.2024.6635>
22. Nasr A., Laschowski B., McPhee J. Myoelectric Control of Robotic Leg Prostheses and Exoskeletons: A Review. *45th Mechanisms and Robotics Conference (MR)*, 2021, Vol. 8A, V08AT08A043. <https://doi.org/10.1115/DETC2021-69203>
23. Sanchez A., Rossos T., Mihailidis A., Laschowski B. Preliminary Development of a Robotic Hip-Knee Exoskeleton With 3D-Printed Backdrivable Actuators. *47th Mechanisms and Robotics Conference (MR)*, 2023, Vol. 8, V008T08A072. <https://doi.org/10.1115/DETC2023-116406>
24. Kaverinsky V.V., Malakhov K.S. Natural Language-Driven Dialogue Systems for Support in Physical Medicine and Rehabilitation. *South African Computer Journal*, 2023, Vol. 35 (2), 119–126. <https://doi.org/10.18489/sacj.v35i2.17444>
25. Palagin O.V., Kaverinsky V.V., Petrenko M.G., Malakhov K.S. Digital Health Systems: Ontology-Based Universal Dialog Service for Hybrid E-Rehabilitation Activities Support. *IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, 2023, Vol. 1, 84–89. <https://doi.org/10.1109/IDAACS58523.2023.10348639>
26. Palagin O.V., Kaverinskiy V.V., Litvin A., Malakhov K.S. OntoChatGPT Information System: Ontology-Driven Structured Prompts for ChatGPT Meta-Learning. *International Journal of Computing*, 2023, Vol. 22 (2), 170–183. <https://doi.org/10.47839/ijc.22.2.3086>
27. *Conceptual Models for Clinical Data Repository Implementation*. 2022. URL: <https://www.youtube.com/watch?v=uw0a4HAX7mc>
28. Ramesh S. *Generative AI, Structured Data & openEHR*. (10 Jun. 2024). URL: <https://www.youtube.com/watch?v=XUp99f8C5us>
29. Opanasenko V.M., Fazilov S.K., Mirzaev O.N., Kakharov S.S. ugli. An Ensemble Approach To Face Recognition In Access Control Systems. *Journal of Mobile Multimedia*, 2024, Vol. 20 (3), 749–768. <https://doi.org/10.13052/jmm1550-4646.20310>
30. Opanasenko V.M., Fazilov Sh.Kh., Radjabov S.S., Kakharov Sh.S. Multilevel Face Recognition System. *Cybernetics and Systems Analysis*, 2024, Vol. 60 (1), 146–151. <https://doi.org/10.1007/s10559-024-00655-w>
31. Kurgaev O.P. Extension of the Metalanguage of Normal Forms of Knowledge. *Cybernetics and Systems Analysis*, 2020, Vol. 56 (6), 1021–1028. <https://doi.org/10.1007/s10559-020-00322-w>
32. Kurgaev O.P., Grigoriev S.N. Metalanguage of Normal Forms of Knowledge. *Cybernetics and Systems Analysis*, 2016, Vol. 52 (6), 839–848. <https://doi.org/10.1007/s10559-016-9885-3>
33. Busti C., Gamboni A., Calabrò G., Zampolini M., Zedde M., Caso V., Corea F. Telesroke: Barriers to the Transition. *Frontiers in Neurology*, 2021, Vol. 12. <https://doi.org/10.3389/fneur.2021.689191>
34. *Remote Patient Monitoring Playbook*. American Medical Association. 2022. URL: <https://www.ama-assn.org/system/files/ama-remote-patient-monitoring-playbook.pdf>

35. *Telehealth Implementation Playbook*. American Medical Association. 2022. URL: <https://www.ama-assn.org/system/files/ama-telehealth-playbook.pdf>
36. Stucki G., Zampolini M., Juocevicius A., Negrini S., Christodoulou N. Practice, science and governance in interaction: European effort for the system-wide implementation of the International Classification of Functioning, Disability and Health (ICF) in Physical and Rehabilitation Medicine. *European Journal of Physical and Rehabilitation Medicine*, 2017, Vol. 53 (2), 299–307. <https://doi.org/10.23736/S1973-9087.16.04436-1>
37. Pammi M., Shah P.S., Yang L.K., Hagan J., Aghaeepour N., Neu J. Digital twins, synthetic patient data, and in-silico trials: Can they empower paediatric clinical trials? *The Lancet Digital Health*, 2025, Vol. 7 (5), 100851. <https://doi.org/10.1016/j.landig.2025.01.007>
38. Pathmanathan P., Aycok K., Badal A., Bighamian R., Bodner J., Craven B.A., Niederer S. Credibility assessment of in silico clinical trials for medical devices. *PLOS Computational Biology*, 2024, Vol. 20 (8), e1012289. <https://doi.org/10.1371/journal.pcbi.1012289>
39. Müller A., Kraus S., Arimond R., Kunczik J., Rossaint R., Czaplík M., Follmann A. Telemedicine in civil protection: A controlled simulation study for the analysis of patient care. *Digital Health*, 2024, Vol. 10, 20552076241272662. <https://doi.org/10.1177/20552076241272662>
40. Ringeval M., Etindele Sosso F.A., Cousineau M., Paré G. Advancing Health Care With Digital Twins: Meta-Review of Applications and Implementation Challenges. *Journal of Medical Internet Research*, 2025, Vol. 27, e69544. <https://doi.org/10.2196/69544>
41. Kurgaev O.P., Palagin O.V., Malakhov K.S., Semykopna T.V. *Device for the Rehabilitation of People With Disabilities*, 2022, (Patent No. 151356). URL: <https://sis.nipo.gov.ua/en/search/detail/1697507/>
42. Calabrò R.S., Mojdehdehbaheer S. AI-Driven Telerehabilitation: Benefits and Challenges of a Transformative Healthcare Approach. *AI*, 2025, Vol. 6 (3), 62 p. <https://doi.org/10.3390/ai6030062>
43. Malahov K.S. *Simulation-based study (in-silico trials) on the effectiveness of a transdisciplinary approach to developing personalized telerehabilitation programs*. [Dataset]. Zenodo, 2025. <https://doi.org/10.5281/zenodo.15785402>

Отримано / Received: 19.12.2024

M.G. PETRENKO, DSc (Engineering), Leading Researcher,
Microprocessor Technology Department of the
V.M. Hlushkov Institute of Cybernetics of the NAS of Ukraine,
40, Hlushkova Akad. ave., Kyiv, 03187, Ukraine
<https://orcid.org/0000-0001-6440-0706>
petrng@ukr.net

K.S. MALAKHOV, Researcher,
Microprocessor Technology Department of the
V.M. Hlushkov Institute of Cybernetics of the NAS of Ukraine,
40, Hlushkova Akad. ave., Kyiv, 03187, Ukraine
<https://orcid.org/0000-0003-3223-9844>
k.malakhov@outlook.com

METHODOLOGY AND PRACTICE OF INTERDISCIPLINARY KNOWLEDGE CONVERGENCE IN DIGITAL HEALTH

Introduction. Transdisciplinary integration of medicine, informatics, engineering and social-behavioral sciences is increasingly viewed as essential for scalable, patient-centered digital health solutions. This study builds on a transdisciplinary framework

to explore how converging disciplinary knowledge can advance remote rehabilitation services.

Methods. A simulation study was conducted in the Hybrid Cloud Environment for Telerehabilitation HCET. Sixty virtual post-stroke patients were randomly assigned to a control group (standard telerehabilitation) or an experimental group (HCET-driven, AI-adapted program). Key end points included change on the Functional Independence Measure (FIM), session adherence, number of completed sessions (max = 20), and self-reported pain reduction.

Results. The HCET-adapted program produced a 57% greater mean FIM gain (17.4 ± 4.7 vs 11.1 ± 4.5 points), a 13% higher adherence rate, a 26% increase in completed sessions, and a two-fold reduction in pain scores compared with controls.

Discussion and conclusions. Findings suggest that transdisciplinary-oriented digital platforms can deliver clinically and economically meaningful benefits, particularly for patients with limited access to in-person rehabilitation. Although based on simulated data, the results validate the proposed methodological framework and justify prospective real-world trials to confirm long-term effectiveness.

Keywords: *scientific worldview, transdisciplinarity, interdisciplinarity, convergent technologies, digital health, simulation study.*

AUTHOR GUIDELINES

The journal publishes the results of research in the field of computer science, information technologies and systems and their applications in various fields of activity, system analysis, intelligent control, cyber security, biological and medical cybernetics, digital economy and learning in the digital age, etc.

Target audience – scientists, engineers, graduate students and students of higher educational institutions of the relevant specialty.

Requirements for manuscripts

1. The manuscript is accepted in electronic form, if possible – on paper in one copy (language of the article – English (*in priority*) or Ukrainian, manuscript up to 30 pages). The manuscript should contain:

- information about the Authors in English and Ukrainian: Full name, Academic Degree, Academic Title, Position, Affiliations, Postal address of the organization, Direct links to author's ORCID (if necessary, Researcher ID) and E-mail, Author-correspondent and their telephone number (*for contacting the editor*);

- Short Abstract with keywords in paper language, and Extended Abstract with keywords in either Ukrainian for English-language paper or English for Ukrainian-language paper. The text of the Extended Abstract is not less than 1800 characters with spaces, by headings: *Introduction, The purpose of the paper, Methods, Results, Conclusions, Keywords* (5–8 words);

- *REFERENCES* – a list of sources in English in the order of mention in the text. For Non-English-language sources, citation are translated, the original language is indicated in square brackets, for Ukrainian-language papers information about authors and the title in the original language is given. Examples for design of the References is given below;

- *LITERATURE* – a list of sources in Ukrainian do not use for English-language papers;

- if desired, the authors provide information about the grant or financial support of the research;

- the license agreement is signed automatically when the submission is created in the electronic editorial system.

2. The text of the article should be submitted with the following mandatory headings: *Introduction, Problem Statement / Problem Definition, Objective, Results, and clearly formulated Conclusions*.

Requirements for the text file

File format: *.doc, *.rtf. Applicable styles: Times New Roman font, 12 pt, without a hyphen for a line break, line spacing – 1.5. Paper size: A4, all sides – 2 cm.

Formulas are typed in Formula Editors (preferably Microsoft Equation Editor 3.0. and MathType 6.9b.) Formula Editor options are (10.5; 8.5; 7.5; 14; 10). **The width of formulas is up to 12 cm.**

Figures must be of high quality, they are provided in separate files of appropriate formats (*.png, *.jpg, *.tiff, etc.). **The width of the figures is up to 12 cm.**

Tables are created using a standard text editor built into the Table toolkit. **The width of the table is up to 12 cm.**

КЕРІВНИЦТВО ДЛЯ АВТОРІВ

В журналі друкуємо результати досліджень у сфері інформатики, інформаційних технологій та систем і їх застосувань у різних сферах діяльності, системного аналізу, інтелектуального керування, кібербезпеки, біологічної та медичної кібернетики, цифрової економіки та навчання в цифрову епоху тощо.

Цільова аудиторія — науковці, інженери, аспіранти та студенти вищих навчальних закладів відповідного фаху.

Вимоги до рукописів статей

1. Рукопис приймаємо в електронному виді, за можливості — на папері в одному примірнику (мова статті — англійська (*в пріоритеті*) або українська, рукопис до 30 стор.). Рукопис має містити:

- відомості про авторів англійською та українською мовами: ПІБ, науковий ступінь, вчене звання, посаду, місце роботи, поштову адресу організації, прямі посилання авторських ідентифікаторів ORCID (за потреби *Researcher ID*) та *E-mail*, автор-кореспондент із номером телефону (*для зв'язку з редактором*);

- коротку анотацію з ключовими словами мовою статті та розширену анотацію з ключовими словами українською для англійської статті або ж англійською для українськомовної статті. Текст розширеної анотації не менше 1800 знаків з пробілами, за рубриками: *Introduction / Вступ, The purpose of the paper / Мета роботи, Methods / Методи, Results / Результати, Conclusions / Висновки, Keywords / Ключові слова* (5–8 слів);

- ЛІТЕРАТУРА — перелік джерел для українськомовних статей українською мовою оригіналу в порядку згадування в тексті, для неукраїнськомовних джерел посилання даються англійською, в квадратних дужках вказується мова оригіналу;

- REFERENCES — перелік джерел англійською мовою в порядку згадування в тексті. Для неанглійськомовних джерел в квадратних дужках вказується мова оригіналу, для українськомовних після двокрапки наводиться інформація про авторів та назва джерела мовою оригіналу. Приклади оформлення переліку посилань наведено далі;

- за бажанням автори надають інформацію про грант або фінансову підтримку дослідження;

- ліцензійний договір підписується автоматично при створенні подання в системі електронної редакції.

2. Текст статті подають з обов'язковими рубриками: *Вступ, Постановка завдання/Окреслення проблеми, Мета, Результати, чітко сформульовані Висновки.*

Вимоги до текстового файлу

Формат файлу *.doc, *.rtf. Застосовні стилі: шрифт Times New Roman, 12 пт, без переносів, міжрядковий інтервал — 1,5. Формат паперу А4, всі береги — 2 см.

Формули набирають у редакторах формул (бажано Microsoft Equation Editor 3.0 та MathType 6.9b.) Опції редактора формул — (10,5; 8,5; 7,5; 14; 10).

Ширина формул — до 12 см.

Рисунки мають бути якісними, їх надають окремими файлами відповідних форматів (*.png, *.jpg, *.tiff тощо). **Ширина рисунків — до 12 см.**

Таблиці виконують стандартним вбудованим у інструментарієм «Таблиця» текстового редактора. **Ширина таблиці — до 12 см.**

Examples for design of the References

REFERENCES

The Books

1. De Vooght E. *Learning to think critically*. Academia Press, Gent , 2025, 168 p. [In Dutch]
2. Ustymenko V.A. *Adaptation of national legislation to EU law: foundations, criteria, sustainability degree*. Akademperiodyka, Kyiv, 2025, 452 p. [In Ukrainian: Устименко В.А. *Адаптація національного законодавства до права Європейського Союзу: основи, критерії, виміри стійкості*]
3. Divan M. J., Johri P., Guim F., Shchemelinin D., Carranza M. *Advances in Image Processing, Reliability, and Artificial Intelligence*. Elsevier, 2025.
4. *Department of Informatics of NAS of Ukraine. Historical and Biographical Directory*. Akademperiodyka, Kyiv, 2017, 286 p. [In Ukrainian: Відділення інформатики НАН України. Історико-біографічний довідник. Академперіодика]

Papers in Periodicals / Статті в періодичних виданнях

5. Zhuoqun Xia, Longfei Huang, Jingjing Tan, Yongbin Yu, Wei Hao, Kejun Long. A lightweight intrusion detection system for connected autonomous vehicles based on ECANet and image encoding. *Journal of Information Security and Applications*, Elsevier, 2025, Vol. 92 (7), Article 104082.
<https://doi.org/10.1016/j.jisa.2025.104082>
6. ZagorodnyA. G., Khimich O. M., Andon F. I., et al. Implementation of European principles of open science in the National Academy of Sciences of Ukraine. *Visnik Nacionalnoi Akademii Nauk Ukrainy*, 2025, Vol. 1, 11–33.

Conferences Materials

7. Neumannova A. Organizational Culture and Digital Resilience: Competing Values Perspective. *17th IADIS International Conference Information Systems*, Porto, Portugal, 2024, 158-162. URL: <https://www.iadisportal.org/digital-library/organizational-culture-and-digital-resilience-competing-values-perspective> [Accessed 20 May. 2025]
8. Husna B. A., Munir R. 3D Traffic Scenes Reconstruction for Autonomous Vehicles Using Gaussian Process Latent Variable Model (GPLVM). *11th International Conference on Advanced Informatics: Concept, Theory and Application (ICAICTA)*, Singapore, Issue 1, 2024, 1-6.
<https://doi.org/10.1109/ICAICTA63815.2024.10763077>

Electronic Sources

9. Information for Authors of Springer Computer Science Proceedings: Instructions for proceedings authors (pdf). URL: <https://resource-cms.springernature.com/springer-cms/rest/v1/content/19242230/data/v17> [Accessed Mar. 2025]
10. Cyrillic Gap Analysis. W3C Group Draft Note 02 April 2025. URI: <https://www.w3.org/TR/cyrl-gap/> [Accessed 26 Jun. 2025]

Приклади оформлення

ЛІТЕРАТУРИ за ДСТУ 8302:2015 для українськомовних статей

Книги

1. Єгоров І.Ю., Никифорок О.І. Цифрові технології в інноваційній трансформації економіки України: колективна монографія. Київ : НАН України, ДУ «Інститут економіки та прогнозування НАН України», 2020. 308 с.
2. Апанасенко Г.Л., Попова Л.А. Медична валеологія. Київ: Здоров'я, 1998. 246 с.
3. Людський розвиток в Україні: трансформація рівня життя та регіональні диспропорції: у 2-х т.: кол. моногр. Київ, 2012. 436 с.

Розділ книги

4. Корнацький В.М. Проблема здоров'я в умовах війни. В кн.: Стрес і серцево-судинні захворювання в умовах воєнного стану. Київ: ДУ «Національний науковий центр «Інститут кардіології, клінічної та регенеративної медицини імені академіка М.Д. Стражеска НАМК України», 2022. С. 5–23.

Статті у періодичних виданнях

5. Сидоренко Ю.В., Шалденко О.В. Вагові політочкові перетворення при моделюванні можливих результатів пластичної хірургії. Сучасні проблеми моделювання, 2019. Вип. 15. С. 151–161.
6. Методологічні підходи до оцінки рівня економічної безпеки країни / Під ред. Ю.М. Харазішвілі. Наука та наукознавство. 2014. № 4. С. 44–58.
7. Манако А.Ф. Системне дослідження безперервного електронного навчання як складної інформаційної системи. Control Systems and Computers. 2022. № 3. С. 53–62. <https://doi.org/10.15407/csc.2022.03.053>
8. Мороз О.Г. Побудова авторегресійних моделей на основі комбінаторно-генетичного алгоритму. Індуктивне моделювання складних систем: Зб. наук. пр. Київ: МННЦ ІТС НАН та МОН України, 2019. Вип. 11. С. 53–60.

Матеріали конференцій

9. Суровцев І.В., Степашко В.С., Галімова В.М., Савченко-Синякова Є.А., Єфіменко С.М., Мороз О.Г. Математичне моделювання нижньої огинаючої диференціального сигналу для задачі інверсної хронопотенціометрії. Міжнародна науково-практична конференція «Математичне та імітаційне моделювання систем» МОДС 2024, Чернігів, Україна, НУ «Чернігівська політехніка», 11-13 листопада 2024 р.
10. Chavarriga, O., Florian-Gaviria, B., Solarte, O. A Recommender System for Students Based on Social Knowledge and Assessment Data of Competences. In: Rensing, C., de Freitas, S., Ley, T., Muñoz-Merino, P.J. (eds) Open Learning and Teaching in Educational Communities. EC-TEL 2014. Lecture Notes in Computer Science, 2014. Vol. 8719. Springer, Cham. https://doi.org/10.1007/978-3-319-11200-8_5

Електронні ресурси

11. Закон України від 21.06.2018 № 2469-VIII Про національну безпеку України. URL: <https://ips.ligazakon.net/document/T182469> (дата звернення: 15.09.2023).
12. OpenAerialMap. URL: <https://openaerialmap.org/>
13. 24 Канал. Якою мовою українці спілкуються вдома: опитування. URL: 24tv.ua/yakoyu-movoyu-ukrayintsi-spilkuuyutsya-vdoma-opituvannya-ukrayina-novini_n1715078 (дата звернення 10.07.2024)
14. Наукові публікації і видавнича діяльність НАН України. Київ, 2007. URL: <http://www.nas.gov.ua/publications> (дата звернення: 19.03.2014)

Патенти

15. Пат. КМ 98077 Україна. Спосіб одержання йодиду цезію або йодиду натрію для вирощування монокристалів. Опубл. 10.04.2015.

Journal “*Information Technologies and Systems*” that is a merger of two academic journals with a long history – Control Systems and Computers journal ISSN (Print) 2706-8145, ISSN (Online) 2706-8153 (published since 1972) and Cybernetics and Computer Engineering journal ISSN (Print) – 2663-2578, ISSN (Online) – 2663-2586 (published since 1965). The organization responsible for publishing the journal is Institute of Information Technologies and Systems of National Academy of Sciences of Ukraine.

The journal publishes original scientific and review papers about fundamental and applied research results of informatics and information technologies, intelligent control and systems, methods and means of information technology support of knowledge, application of the mentioned technologies in various fields of life. Number of Issues is 6 per year. Currently, the journal does not charge any fees to the authors from submission to publication. The papers are an Open Access under the CC BY-NC-ND 4.0 license.



Журнал «Інформаційні технології та системи» є результатом об'єднання двох академічних журналів з багаторічною історією – журналу «Системи керування та обчислювальна техніка» ISSN (друковане видання) 2706-8145, ISSN (онлайн) 2706-8153 (видається з 1972 року) та журналу «Кібернетика та обчислювальна техніка» ISSN (друковане видання) – 2663-2578, ISSN (онлайн) – 2663-2586 (видається з 1965 року). Організацією, відповідальною за видання журналу, є Інститут інформаційних технологій та систем Національної академії наук України.

Журнал публікує оригінальні наукові та оглядові статті про фундаментальні та прикладні результати досліджень інформатики та інформаційних технологій, інтелектуального керування та систем, методів та засобів інформаційно-технологічної підтримки знань, застосування згаданих технологій у різних сферах життя. Кількість випусків – 6 на рік. Наразі журнал не стягує жодних гонорарів з авторів від моменту подання до публікації. Статті знаходяться у відкритому доступі за ліцензією CC BY-NC-ND 4.0.

The background of the page is a vibrant blue with abstract, flowing white and light purple lines that create a sense of motion and depth. These lines are layered and overlap, giving the impression of light trails or digital data paths.

ISSN 3083-6573 INFORMATION TECHNOLOGIES AND SYSTEMS 2025, №3. 1-100