

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Дніпровський національний університет імені Олеся Гончара

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Дніпровський національний університет імені Олеся Гончара

Кваліфікаційна наукова
праця на правах рукопису

ДУБОВИК ВІТАЛІЙ ВАЛЕНТИНОВИЧ

УДК 004.9:629.8:004.89

ДИСЕРТАЦІЯ
РОЗРОБЛЕННЯ ПРОГРАМНОГО КОМПЛЕКСУ НАВЧАННЯ
РОБОТЕХНІЧНИХ СИСТЕМ ЛІКВІДАЦІЇ НАСЛІДКІВ
ТЕХНОГЕННОГО ПОХОДЖЕННЯ

12 Інформаційні технології

121 Інженерія програмного забезпечення

Подається на здобуття ступеня доктора філософії. Дисертація містить результати власних досліджень. Використання ідей, результатів та текстів інших авторів мають посилання на відповідне джерело

_____ В.В. Дубовик

Науковий керівник:
Кузенков Олександр Олександрович
кандидат фізико-математичних наук, доцент

Дніпро – 2026

АНОТАЦІЯ

Дубовик В.В. Розроблення програмного комплексу навчання роботехнічних систем ліквідації наслідків техногенного походження. – Кваліфікаційна наукова робота на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 121 Інженерія програмного забезпечення. – Дніпровський національний університет імені Олеся Гончара, м. Дніпро, 2026.

Дисертаційна робота присвячена проектуванню та розробленню програмного забезпечення для навчання роботехнічних систем ліквідації наслідків техногенного походження.

Метою дисертаційної роботи є проектування, розроблення та тестування програмного забезпечення для навчання роботехнічних систем ліквідації наслідків техногенного походження.

Проведений аналіз показав актуальність використання роботизованих систем для розв’язання питань ліквідації наслідків техногенного походження. Використання таких систем вимагає їх проектування, навчання, адаптації та налаштування під конкретні умови використання та задачі. У зв’язку з цим *актуальною науковою задачею* є саме проектування, розробка та тестування системи навчання роботехнічних систем ліквідації наслідків техногенного походження.

Об’єктом дослідження є роботехнічні системи, що використовуються для ліквідації наслідків техногенного походження.

Предметом дослідження є програмний комплекс збору інформації, прийняття рішення та реагування на аварійні ситуації, що виникають унаслідок подій техногенного походження.

Для досягнення поставленої мети були поставлені та розв’язані такі наукові задачі:

– здійснено класифікацію роботехнічних систем, що використовуються під час ліквідації наслідків техногенного походження;

– здійснено порівняльний аналіз підходів, що використовуються для автоматизації процесу навчання роботехнічних систем різної конструкції та призначення;

– сформульовано постановку технічної задачі для розробки програмного комплексу навчання роботехнічних систем ліквідації наслідків техногенного походження;

– спроектовано програмний комплекс навчання роботехнічних систем ліквідації наслідків техногенного походження;

– проаналізовано існуючі інструменти та сформовано набір технологій для розробки програмного комплексу навчання роботехнічних систем ліквідації наслідків техногенного походження;

– розроблено програмний комплекс навчання роботехнічних систем ліквідації наслідків техногенного походження;

– здійснено тестування програмного комплексу навчання роботехнічних систем ліквідації наслідків техногенного походження.

У сучасних умовах розвитку суспільства постає значна кількість проблем техногенного та воєнного характеру, пов'язаних із функціонуванням складних промислових, енергетичних, транспортних і інфраструктурних об'єктів, а також із наслідками бойових дій та руйнування критичної інфраструктури. Аварії на потенційно небезпечних виробництвах, пожежі, вибухи, витoki токсичних речовин, обвали конструкцій, мінування територій, наявність нерозірваних боєприпасів і завалів створюють підвищену загрозу для життя і здоров'я людей та унеможливають безпечне виконання робіт безпосередньо в зоні ураження. Проведення розвідки, пошуку постраждалих, локалізації джерел небезпеки, розмінування та ліквідації наслідків руйнувань вимагає застосування технічних засобів, здатних діяти автономно в умовах ризику, обмеженої видимості, нестабільного середовища та дефіциту часу.

За таких обставин важливу роль відіграють роботехнічні системи, використання яких дозволяє мінімізувати залучення людини до небезпечних операцій, підвищити оперативність реагування та забезпечити безперервність виконання завдань у складних умовах. Мобільні платформи, безпілотні наземні та повітряні комплекси, маніпуляційні роботи здатні здійснювати моніторинг територій, транспортування вантажів, проведення технічних робіт і спеціалізованих операцій там, де присутність людини є небажаною або неможливою. Водночас ефективність їх застосування значною мірою залежить від рівня автономності, здатності адаптуватися до змін обстановки та швидко приймати рішення на основі даних сенсорів.

Існуючі підходи до програмування та підготовки роботехнічних засобів часто передбачають ручне налаштування алгоритмів і сценаріїв роботи, що потребує значних часових витрат і не забезпечує достатньої гнучкості при зміні умов експлуатації. Реальні середовища характеризуються невизначеністю, неповнотою інформації, динамічними перешкодами та складною структурою простору, що ускладнює використання жорстко заданих методів керування. Це зумовлює необхідність створення спеціалізованих інструментів, які дозволяють організувати процес навчання роботів, відпрацювання стратегій поведінки та тестування алгоритмів у безпечному керованому середовищі.

У роботі розглянуто розроблення програмного комплексу, призначеного для моделювання умов надзвичайних ситуацій, формування навчальних сценаріїв і підготовки роботехнічних систем до виконання задач ліквідації наслідків техногенних аварій і воєнних руйнувань. Комплекс забезпечує інтеграцію засобів імітаційного моделювання, обробки сенсорної інформації, планування руху, координації дій та інтелектуального прийняття рішень. Реалізація механізмів машинного навчання та адаптивного керування дозволяє формувати стійкі моделі поведінки роботів, здатних ефективно діяти в умовах невизначеності, змінювати стратегію залежно від поточної ситуації та взаємодіяти між собою під час виконання спільних завдань.

Застосування розробленого програмного комплексу сприяє підвищенню якості підготовки роботехнічних систем, скороченню часу розроблення алгоритмів, зниженню витрат на експериментальні дослідження та підвищенню безпеки випробувань. Отримані результати можуть бути використані під час створення інтелектуальних роботизованих засобів для аварійно-рятувальних служб, підрозділів цивільного захисту та робіт із відновлення пошкоджених територій і об'єктів інфраструктури.

У **вступі** обґрунтовано актуальність розроблення програмного комплексу навчання роботехнічних систем для задач ліквідації наслідків техногенного походження, сформульовано мету, наукові задачі, об'єкт і предмет дослідження, визначено методи дослідження, наукову новизну та практичне значення отриманих результатів.

У **першому розділі** проаналізовано сучасний стан і перспективи розвитку роботехнічних систем, що застосовуються в умовах надзвичайних і техногенних ситуацій. Розглянуто архітектури та програмні платформи роботехнічних систем, методи навігації, локалізації, машинного навчання, навчання з підкріпленням, а також питання людино-роботної взаємодії та безпечного керування в аварійних сценаріях. На основі проведеного аналізу сформульовано наукову задачу дослідження.

Другий розділ присвячено проектуванню архітектури програмного комплексу навчання роботехнічних систем. У розділі визначено вимоги до програмного комплексу, запропоновано концепцію гібридної архітектури, обґрунтовано використання цифрового двійника як основи навчального процесу, описано взаємодію основних програмних модулів та визначено переваги запропонованого архітектурного підходу.

У **третьому розділі** розроблено моделі та методи адаптивного навчання роботехнічних систем на основі Digital Twin. Наведено формалізацію процесу адаптивного навчання, модель навчального циклу цифрового двійника, підходи до використання методів машинного навчання та навчання з підкріпленням у програмному комплексі, а також програмний модуль

комп'ютерного зору для візуальної ідентифікації та локалізації об'єктів. Окрему увагу приділено методу оцінювання ефективності адаптивного навчання роботехнічних систем.

Четвертий розділ присвячено застосуванню цифрового двійника для моделювання, прогнозування та аналізу кризових процесів, пов'язаних із наслідками техногенного походження. У ньому розглянуто автоматизовану систему збору та аналізу інформації про виявлені повітряні об'єкти як компонент ситуаційної обізнаності, а також систему аналізу та прийняття рішень щодо оперативних заходів протидії поширенню інфекційних захворювань як приклад моделювання складних динамічних процесів у цифровому середовищі.

У **висновках** узагальнено основні результати дисертаційного дослідження, показано їх відповідність поставленій меті та науковим задачам, сформульовано положення наукової новизни, практичне значення отриманих результатів та окреслено напрями подальшого розвитку теми.

Наукова новизна одержаних результатів полягає у наступному:

1. **Вперше** розроблено концептуальну та програмну модель комплексної системи навчання роботехнічних систем ліквідації наслідків техногенного походження, що інтегрує засоби імітаційного моделювання аварійних сценаріїв, механізми обробки сенсорної інформації, алгоритми автономної навігації та прийняття рішень в єдиному програмному середовищі, орієнтованому на підготовку роботів до роботи в умовах невизначеності та динамічних змін середовища.

2. **Вперше** запропоновано архітектурний підхід до побудови програмного комплексу навчання роботехнічних систем для задач техногенного реагування, який базується на модульному принципі організації компонентів, підтримує масштабування, інтеграцію різномірних алгоритмів керування та забезпечує можливість адаптації до різних типів роботехнічних платформ.

3. Удосконалено метод організації процесу симуляційного навчання роботехнічних систем шляхом формалізації навчальних сценаріїв надзвичайних ситуацій із варіативністю параметрів середовища, що дозволяє підвищити стійкість алгоритмів керування до змін зовнішніх умов та зменшити залежність від натурних експериментів.

4. Удосконалено підхід до інтеграції алгоритмів машинного навчання та класичних методів керування у програмному середовищі навчання роботехнічних систем, що забезпечує поєднання адаптивності, керованості та підвищеної надійності функціонування в аварійних сценаріях.

5. Набули подальшого розвитку методи програмної підготовки роботехнічних систем до роботи в умовах техногенних аварій за рахунок розширення функціональності симуляційного середовища та впровадження механізмів оцінювання ефективності навчання на основі системи кількісних показників.

6. Набули подальшого розвитку підходи до проектування програмних комплексів спеціалізованого призначення для задач цивільного захисту та аварійно-рятувальних операцій шляхом адаптації принципів інженерії програмного забезпечення до специфіки роботехнічних систем з підвищеними вимогами до надійності та безпеки.

Практичне значення одержаних у дисертаційній роботі результатів полягає в тому, що розроблений програмний комплекс навчання роботехнічних систем ліквідації наслідків техногенного походження може бути безпосередньо використаний для підготовки та тестування роботехнічних платформ, призначених для роботи в умовах надзвичайних ситуацій.

1. Розроблений програмний комплекс забезпечує можливість моделювання сценаріїв техногенних аварій, воєнних руйнувань та інших кризових ситуацій у контрольованому симуляційному середовищі, що дозволяє здійснювати навчання роботів без ризику пошкодження обладнання та без загрози для персоналу.

2. Використання комплексу дозволяє суттєво скоротити витрати на натурні експериментальні випробування, зменшити часові витрати на розробку та налагодження алгоритмів керування, а також підвищити ефективність підготовки роботехнічних систем до реальних умов експлуатації.

3. Запропонована архітектура програмного забезпечення забезпечує масштабованість та можливість інтеграції з різними типами роботехнічних платформ (мобільними наземними, повітряними та маніпуляційними системами), що розширює сферу практичного застосування результатів дослідження.

4. Реалізовані механізми формування навчальних сценаріїв та оцінювання ефективності навчання дозволяють використовувати програмний комплекс як інструмент підтримки прийняття технічних рішень при проєктуванні та модернізації роботехнічних систем спеціального призначення.

5. Результати дослідження можуть бути використані у діяльності аварійно-рятувальних служб, підрозділів цивільного захисту, підприємств критичної інфраструктури, а також організацій, що здійснюють роботи з ліквідації наслідків техногенних аварій та відновлення пошкоджених об'єктів.

6. Матеріали дисертаційної роботи можуть бути використані у навчальному процесі закладів вищої освіти при підготовці фахівців за спеціальностями, пов'язаними з інженерією програмного забезпечення, роботехнікою, штучним інтелектом та системами автоматизованого керування.

Таким чином, отримані результати мають прикладний характер і спрямовані на підвищення рівня автономності, ефективності та безпеки використання роботехнічних систем у задачах ліквідації наслідків техногенного походження.

Ключові слова: програмне забезпечення, програмний комплекс, роботехнічні системи, інформаційні технології, цифровий двійник, адаптивне навчання, машинне навчання, навчання з підкріпленням, імітаційне

модельовання, комп'ютерний зір, розпізнавання образів, архітектура програмного забезпечення, гібридна архітектура, модель керування, надзвичайні ситуації техногенного характеру.

ANNOTATION

Dubovyk V.V. Development of a Software System for Training Robotic Systems for the Elimination of the Consequences of Man-Made Disasters. – Qualification scientific work submitted as a manuscript.

Dissertation for the degree of Doctor of Philosophy in Specialty 121 Software Engineering. – Oles Honchar Dnipro National University, Dnipro, 2026.

The dissertation is devoted to the design and development of software for training robotic systems intended for the mitigation of the consequences of man-made emergencies.

The aim of the dissertation is the design, development, and testing of software for training robotic systems for the mitigation of the consequences of man-made emergencies.

The conducted analysis demonstrated the relevance of using robotic systems to address the problems of mitigating the consequences of man-made incidents. The use of such systems requires their design, training, adaptation, and configuration for specific operating conditions and tasks. In this regard, an important scientific task is the design, development, and testing of a training system for robotic platforms used in the mitigation of the consequences of man-made emergencies.

The object of research is robotic systems used for mitigating the consequences of man-made incidents.

The subject of research is a software suite for information acquisition, decision-making, and response to emergency situations caused by man-made events.

To achieve the stated goal, the following scientific tasks were formulated and solved:

- classification of robotic systems used during the mitigation of the consequences of man-made emergencies was carried out;

- a comparative analysis of approaches used to automate the training process of robotic systems of various designs and purposes was performed;

- the technical problem statement for the development of a software suite for training robotic systems was formulated;
- the software suite for training robotic systems was designed;
- existing tools were analyzed and a set of technologies for the development of the software suite was selected;
- the software suite for training robotic systems was developed;
- testing of the developed software suite was conducted.

Under current conditions of societal development, a significant number of man-made and military-related challenges arise, associated with the operation of complex industrial, energy, transportation, and infrastructure facilities, as well as with the consequences of hostilities and the destruction of critical infrastructure. Accidents at hazardous industrial sites, fires, explosions, toxic leaks, structural collapses, landmines, unexploded ordnance, and debris pose increased risks to human life and health and make it impossible to safely perform operations directly within affected areas. Reconnaissance, victim search and rescue, hazard localization, demining, and damage mitigation require the use of technical means capable of operating autonomously under risk, limited visibility, unstable environments, and time constraints.

In such conditions, robotic systems play an important role, as their use minimizes human involvement in hazardous operations, increases response efficiency, and ensures the continuity of task execution in difficult environments. Mobile platforms, unmanned ground and aerial vehicles, and manipulation robots are capable of monitoring territories, transporting loads, performing technical operations, and conducting specialized missions where human presence is undesirable or impossible. At the same time, the effectiveness of their application largely depends on the level of autonomy, adaptability to changing conditions, and the ability to make rapid decisions based on sensor data.

Existing approaches to programming and preparing robotic platforms often involve manual configuration of algorithms and operational scenarios, which

requires considerable time and does not provide sufficient flexibility when operating conditions change. Real environments are characterized by uncertainty, incomplete information, dynamic obstacles, and complex spatial structures, which complicates the use of rigid control methods. This necessitates the creation of specialized tools that enable the organization of robot training processes, development of behavioral strategies, and testing of algorithms within a safe and controlled environment.

The work considers the development of a software suite intended for modeling emergency conditions, forming training scenarios, and preparing robotic systems to perform tasks related to mitigating the consequences of man-made accidents and wartime destruction. The suite integrates simulation tools, sensor data processing, motion planning, coordination mechanisms, and intelligent decision-making. The implementation of machine learning and adaptive control mechanisms enables the formation of stable behavioral models for robots capable of effectively operating under uncertainty, adapting strategies according to the current situation, and cooperating with each other while performing joint tasks.

The application of the developed software suite improves the quality of training of robotic systems, reduces the time required for algorithm development, lowers experimental costs, and increases testing safety. The obtained results can be used in the creation of intelligent robotic tools for emergency rescue services, civil protection units, and recovery operations in damaged areas and infrastructure facilities.

The **Introduction** substantiates the relevance of developing a software suite for training robotic systems intended for tasks related to the mitigation of consequences of man-made emergencies. It formulates the aim, research objectives, object and subject of the study, defines the research methods, and outlines the scientific novelty and practical significance of the obtained results.

Chapter 1 analyzes the current state and development prospects of robotic systems used in emergency and man-made disaster scenarios. It considers the architectures and software platforms of robotic systems, methods of navigation and localization, machine learning and reinforcement learning approaches, as well as

issues of human–robot interaction and safe control in emergency scenarios. Based on the conducted analysis, the scientific problem of the study is formulated.

Chapter 2 is devoted to the design of the architecture of the software suite for training robotic systems. The chapter defines the requirements for the software suite, proposes the concept of a hybrid architecture, substantiates the use of a digital twin as the basis of the training process, describes the interaction between the main software modules, and identifies the advantages of the proposed architectural approach.

Chapter 3 presents the development of models and methods for adaptive training of robotic systems based on Digital Twin technology. It provides a formalization of the adaptive training process, a model of the digital twin training cycle, approaches to the use of machine learning and reinforcement learning methods within the software suite, as well as a computer vision software module for visual identification and localization of objects. Particular attention is paid to the method for evaluating the effectiveness of adaptive training of robotic systems. The module also supports pattern recognition tasks as part of the computer vision subsystem.

Chapter 4 is dedicated to the application of a digital twin for modeling, forecasting, and analyzing crisis processes associated with the consequences of man-made emergencies. It examines an automated system for collecting and analyzing information about detected aerial objects as a component of situational awareness, as well as a system for analyzing and supporting decision-making regarding operational measures to counter the spread of infectious diseases as an example of modeling complex dynamic processes in a digital environment.

The Conclusions summarize the main results of the dissertation research, demonstrate their correspondence to the stated aim and research objectives, formulate the provisions of scientific novelty and the practical significance of the obtained results, and outline directions for further development of the research topic.

Scientific novelty of the obtained results is as follows:

1. **For the first time**, a conceptual and software model of a comprehensive training system for robotic systems intended for mitigation of man-made emergency consequences has been developed. The proposed model integrates tools for simulation of emergency scenarios, sensor data processing mechanisms, autonomous navigation algorithms, and decision-making modules within a unified software environment designed to prepare robots for operation under uncertainty and dynamic environmental changes.

2. **For the first time**, an architectural approach to the development of a software suite for training robotic systems in technogenic response tasks has been proposed. The approach is based on a modular organization of components, supports scalability, enables integration of heterogeneous control algorithms, and ensures adaptability to various types of robotic platforms.

3. **The method** for organizing the simulation-based training process of robotic systems **has been improved** through the formalization of emergency training scenarios with variable environmental parameters. This improvement enhances the robustness of control algorithms to external changes and reduces dependence on full-scale physical experiments.

4. **The approach** to integrating machine learning algorithms with classical control methods within a robotic training software environment **has been improved**, ensuring a balanced combination of adaptability, controllability, and increased operational reliability in emergency scenarios.

5. **The methods** of software-based preparation of robotic systems for operation in man-made emergency conditions **have been further developed** by extending the functionality of the simulation environment and implementing mechanisms for evaluating training effectiveness based on a system of quantitative performance indicators.

6. **The approaches** to the design of specialized software systems for civil protection and emergency rescue operations **have been further developed** through

the adaptation of software engineering principles to the specific requirements of robotic systems characterized by increased reliability and safety constraints.

The practical significance of the results obtained in the dissertation lies in the fact that the developed software suite for training robotic systems intended for the mitigation of man-made emergency consequences can be directly applied to the preparation and testing of robotic platforms operating under emergency conditions.

1. The developed software suite enables the simulation of man-made accident scenarios, wartime destruction, and other crisis situations within a controlled virtual environment, allowing robotic systems to be trained without risk of equipment damage and without endangering personnel.

2. The use of the proposed suite significantly reduces the cost of full-scale experimental testing, decreases the time required for the development and tuning of control algorithms, and increases the efficiency of preparing robotic systems for real-world operational conditions.

3. The proposed software architecture ensures scalability and integration with various types of robotic platforms (mobile ground, aerial, and manipulation systems), thereby expanding the scope of practical application of the research results.

4. The implemented mechanisms for generating training scenarios and evaluating training effectiveness make it possible to use the software suite as a decision-support tool in the design and modernization of specialized robotic systems.

5. The research results can be applied in the activities of emergency rescue services, civil protection units, critical infrastructure enterprises, as well as organizations engaged in mitigation of man-made accidents and restoration of damaged facilities.

6. The materials of the dissertation may also be used in the educational process of higher education institutions for training specialists in software engineering, robotics, artificial intelligence, and automated control systems.

Thus, the obtained results have a practical orientation and are aimed at increasing the autonomy, efficiency, and safety of robotic systems used for the mitigation of man-made emergencies.

Keywords: software, software suite, robotic systems, information technologies, digital twin, adaptive learning, machine learning, reinforcement learning, simulation modeling, computer vision, pattern recognition, software architecture, hybrid architecture, control model, man-made emergencies.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Статті у наукових фахових виданнях України:

1. Kuzenkov O., Dubovyk V. Software implementation of the COVID-2019 spread simulation algorithm using the theory of optimal set partitioning. *Актуальні проблеми автоматизації та інформаційних технологій*, т. 24, Дніпро, 2020, с. 96-103. DOI: <https://dx.doi.org/10.15421/432011> (особистий внесок Дубовика В.В.: участь у програмній реалізації алгоритму моделювання поширення COVID-2019, підготовка вхідних даних для чисельного експерименту, аналіз та інтерпретація отриманих результатів; внесок Кузенкова О.О.: постановка наукової задачі, формалізація математичної моделі, методичне керівництво та узагальнення результатів).

2. Kiseleva E., Kuzenkov O., Dubovyk V. Software implementation of the algorithm for solving the dynamic problem of optimal placement of fire groups. *Актуальні проблеми автоматизації та інформаційних технологій*, т. 26, Дніпро, 2022, с. 137–145. DOI: <https://dx.doi.org/10.15421/432217> (особистий внесок Дубовика В.В.: участь у програмній реалізації алгоритму розв'язання динамічної задачі оптимального розміщення вогневих груп, проведення чисельних експериментів, аналіз та інтерпретація результатів; внесок Кісельової О.М.: постановка оптимізаційної задачі, обґрунтування математичної моделі та методологічний супровід; внесок Кузенкова О.О.: наукове консультування, участь у формалізації алгоритмічного підходу та узагальнення результатів).

3. Kuzenkov O., Dubovyk V. Design and implementation of a modular software architecture for dynamic image segmentation in robotic vision systems. *Актуальні проблеми автоматизації та інформаційних технологій*, т. 28, Дніпро, 2024, с. 318–326. DOI: <http://dx.doi.org/10.15421/432430> (особистий внесок Дубовика В.В.: участь у проектуванні модульної архітектури програмної системи динамічної сегментації зображень, визначення структури програмних модулів, аналіз підходів до комп'ютерного зору та розпізнавання образів, аналіз отриманих результатів; внесок Кузенкова О.О.: постановка наукової задачі, методичне консультування, коригування архітектурних рішень та узагальнення результатів).

4. Kuzenkov O., Dubovyk V. Development of a dynamic partitioning model and a software system for image segmentation in the implementation of computer

vision for robotic systems. *Сучасні проблеми моделювання*, т. 27, Мелітополь, 2025, с. 123–135. DOI: <http://dx.doi.org/10.33842/2313-125X-2025-19-123-135>. (особистий внесок Дубовика В.В.: розроблення динамічної моделі розбиття та програмної системи для сегментації зображень, постановка чисельного експерименту, аналіз результатів роботи програмної реалізації; внесок Кузенкова О.О.: постановка наукового завдання, методологічне консультування, уточнення математичної моделі та узагальнення результатів).

5. Kuzenkov O.O., Dubovyk V.V. Development of a software system for training robotic systems for mitigation of technogenic disaster consequences. *Актуальні проблеми автоматизації та інформаційних технологій*, т. 29, Дніпро, 2025, с. 425–436. DOI: <http://dx.doi.org/10.15421/432538> (особистий внесок Дубовика В.В.: обґрунтування концепції програмного комплексу навчання роботехнічних систем, визначення його архітектури й функціональних модулів, участь у розробленні програмної реалізації, тестуванні та аналізі результатів функціонування системи; внесок Кузенкова О.О.: постановка наукового завдання, наукове консультування, коригування архітектурних рішень і узагальнення результатів).

Наукові праці, які засвідчують апробацію матеріалів дисертації:

6. Кузенков О.О., Дубовик В.В. Особливості проектування програмного комплексу навчання роботехнічних систем. *Нотатки сучасної науки: матеріали II Міжнародної науково-практичної конференції «Актуальні питання сучасної науки: історія, теорія, практика»*, № 33, Україна – США, 2026, с. 52–53. URL: <https://www.newroute.org.ua/rep/>; <https://mega.nz/file/Sn5TSIwY#KnGnQoMU-pwC3RKfsf5X4M7Fv58fNZxfKTmhh0dIB2s> (особистий внесок Дубовика В.В.: аналіз вимог до програмного комплексу навчання роботехнічних систем, обґрунтування модульної архітектури, визначення основних функціональних компонентів системи, формування підходу до використання симуляційного середовища та цифрового двійника для підготовки роботехнічних систем до роботи в умовах техногенного походження, аналіз та узагальнення отриманих результатів; внесок Кузенкова О.О.: постановка наукового завдання, наукове консультування, уточнення архітектурної концепції та узагальнення отриманих результатів).

7. Кузенков О.О., Дубовик В.В. Програмна реалізація комплексу навчання роботехнічних систем. *Advanced top technology: матеріали II Міжнародної науково-практичної конференції «Новітні технології сучасного суспільства»*, № 18, Україна – Польща, 2026, с. 49–50. URL: <https://www.newroute.org.ua/rep/>; <https://mega.nz/file/m2YUkSoA#2jD2hS8rOf4DV8p7Jx-iH2IAtZXwxNjQrrItiFT-WEk> (особистий внесок Дубовика В.В.: розроблення програмної реалізації комплексу навчання роботехнічних систем, опис структури основних програмних модулів, обґрунтування логіки взаємодії компонентів, підготовка матеріалів щодо експериментальної реалізації, аналіз результатів функціонування системи; внесок Кузенкова О.О.: постановка наукового завдання, наукове консультування, коригування архітектурних рішень і узагальнення отриманих результатів).

8. Кузенков О.О., Дубовик В.В. Навчання систем керування роботехнічних систем, призначених для ліквідації наслідків техногенного походження. Соціально-гуманітарний вісник: збірник наукових праць за матеріалами IV Міжнародної науково-практичної конференції «Сучасні тенденції соціально-гуманітарного розвитку суспільства», вип. 66, Україна – Німеччина, 2026, с. 77–78. URL: <https://www.newroute.org.ua/rep/>; https://mega.nz/file/vuA2XAKI#Jvbzo3YHOdtQMy4sRCmskeyt_rYr5fOZwx_Lz4nXn6k (особистий внесок Дубовика В.В.: аналіз підходів до навчання систем керування роботехнічних систем, обґрунтування застосування методів машинного навчання та адаптивного керування для задач ліквідації наслідків техногенного походження, формування логіки використання цифрового двійника та симуляційного середовища для підготовки роботехнічних систем, аналіз та узагальнення отриманих результатів; внесок Кузенкова О.О.: постановка наукового завдання, наукове консультування, уточнення методологічних положень і узагальнення отриманих результатів).

ЗМІСТ

ВСТУП	22
РОЗДІЛ 1. ОГЛЯД СУЧАСНОГО СТАНУ, ПЕРСПЕКТИВ РОЗВИТКУ ТА МЕТОДІВ ДОСЛІДЖЕННЯ СИСТЕМ НАВЧАННЯ РОБОТЕХНІЧНИХ СИСТЕМ ПРИ ЛІКВІДАЦІЇ НАСЛІДКІВ ТЕХНОГЕННОГО ПОХОДЖЕННЯ	31
1.1. Роботехнічні системи для роботи в умовах надзвичайних та техногенних ситуацій	31
1.2. Архітектури та програмні платформи роботехнічних систем	36
1.3. Методи навігації, локалізації та взаємодії роботехнічних систем із середовищем	40
1.4. Методи машинного навчання і навчання з підкріпленням у роботехніці	44
1.5. Людино-роботна взаємодія та безпечне керування в аварійних сценаріях	48
1.6. Аналіз існуючих підходів та постановка наукової задачі	51
1.7. Висновки до розділу 1	55
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБЛЕННЯ АРХІТЕКТУРИ ПРОГРАМНОГО КОМПЛЕКСУ НАВЧАННЯ РОБОТЕХНІЧНИХ СИСТЕМ	59
2.1. Аналіз вимог до програмного комплексу навчання роботехнічних систем	59
2.2. Концепція гібридної архітектури програмного комплексу навчання роботехнічних систем	63
2.3. Цифровий двійник роботехнічної системи як основа навчання	68
2.4. Архітектура програмного комплексу та взаємодія модулів	73
2.5. Обґрунтування архітектурних рішень та переваги запропонованого підходу	78
2.6. Висновки до розділу 2	82

РОЗДІЛ 3. МОДЕЛІ ТА МЕТОДИ АДАПТИВНОГО НАВЧАННЯ РОБОТЕХНІЧНИХ СИСТЕМ НА ОСНОВІ DIGITAL TWIN	85
3.1. Формалізація процесу адаптивного навчання роботехнічних систем ..	85
3.2. Модель навчального циклу Digital Twin	88
3.3. Методи адаптивного навчання в програмному комплексі	91
3.4. Розроблення динамічної моделі сегментації та програмної системи для сегментації зображень при реалізації комп'ютерного зору для роботизованих систем	95
3.5. Метод оцінювання ефективності адаптивного навчання роботехнічних систем	102
3.6. Висновки до розділу 3	106
РОЗДІЛ 4. ЗАСТОСУВАННЯ ЦИФРОВОГО ДВІЙНИКА В ПРОЦЕСІ МОДЕЛЮВАННЯ, ПРОГНОЗУВАННЯ ТА АНАЛІЗУ НАСЛІДКІВ ТЕХНОГЕННОГО ПОХОДЖЕННЯ.....	109
4.1. Автоматизована система збору та аналізу інформації про виявлені повітряні об'єкти.....	109
4.2. Система аналізу та прийняття рішень про оперативні заходи протидії поширенню інфекційних захворювань	124
4.3 Висновки до розділу 4	136
ВИСНОВКИ.....	139
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	143
ДОДАТКИ.....	149
Додаток А Список публікацій здобувача за темою дисертації	149
Додаток Б Програмний код системи візуальної ідентифікації.....	152
Додаток В Програмний код реалізації альтернативної системи оповіщення	156
ДОДАТОК Г Програмний код розв'язання задачі розбиття території на зони відповідальності медичних закладів	162

ВСТУП

Актуальність теми. Сучасний етап розвитку суспільства характеризується високим рівнем технологізації виробничих, енергетичних, транспортних та інфраструктурних процесів. Функціонування складних технічних систем, використання небезпечних речовин, експлуатація об'єктів підвищеної небезпеки та зростання щільності урбанізованих територій зумовлюють підвищення ризику виникнення надзвичайних ситуацій техногенного характеру. Аварії на промислових підприємствах, вибухи, пожежі, витоки токсичних або радіоактивних речовин, руйнування будівель і споруд можуть спричиняти масштабні матеріальні збитки, негативні екологічні наслідки та загрозу життю і здоров'ю людей.

Особливої актуальності зазначена проблематика набула в умовах воєнних дій, що супроводжуються руйнуванням критичної інфраструктури, мінуванням територій, наявністю нерозірваних боєприпасів, утворенням завалів та пошкодженням промислових об'єктів. У таких умовах ліквідація наслідків техногенних і воєнних подій потребує виконання робіт у середовищах з підвищеним рівнем небезпеки, обмеженою видимістю, нестабільними конструкціями та високою ймовірністю вторинних уражень. Безпосередня участь людини в подібних операціях пов'язана зі значними ризиками та не завжди є можливою з огляду на складність і динамічність обстановки.

У зв'язку з цим зростає потреба у застосуванні роботехнічних систем, здатних виконувати розвідку, моніторинг, пошук і евакуацію постраждалих, транспортування вантажів, локалізацію джерел небезпеки, розмінування та інші спеціалізовані операції. Використання мобільних наземних і повітряних платформ, маніпуляційних комплексів та автономних роботизованих засобів дозволяє мінімізувати залучення персоналу до небезпечних зон, підвищити оперативність реагування та забезпечити безперервність виконання критично важливих завдань.

Разом із тим ефективність застосування роботехнічних систем значною мірою визначається рівнем їх автономності, здатністю адаптуватися до змін умов середовища та приймати обґрунтовані рішення на основі сенсорної інформації. Реальні умови функціонування характеризуються невизначеністю, неповнотою та зашумленістю даних, наявністю динамічних перешкод, складною геометрією простору та необхідністю швидкого реагування. Традиційні підходи до програмування роботів, що базуються на жорстко заданих алгоритмах і сценаріях, не забезпечують достатньої гнучкості та потребують значних часових і людських ресурсів для налаштування під конкретні задачі.

Суттєвою проблемою є також підготовка роботехнічних систем до роботи в нових або змінених умовах. Проведення натурних випробувань у реальних небезпечних середовищах пов'язане з високими витратами, технічними обмеженнями та ризиками пошкодження обладнання. Це зумовлює необхідність створення програмних інструментів, які дозволяють здійснювати моделювання сценаріїв надзвичайних ситуацій, відпрацювання алгоритмів керування та навчання роботів у контрольованому та безпечному віртуальному середовищі. У цьому аспекті інформаційні технології забезпечують інструментальну основу для формалізації сценаріїв, збору даних і перевірки поведінкових моделей у безпечному цифровому середовищі.

У цьому контексті особливого значення набуває розробка програмного комплексу навчання роботехнічних систем ліквідації наслідків техногенного походження. Такий комплекс має забезпечувати інтеграцію засобів моделювання робочого середовища, обробки сенсорних даних, планування руху, координації дій та інтелектуального прийняття рішень. Використання методів машинного навчання та адаптивного керування дозволяє формувати стійкі моделі поведінки роботів, здатних ефективно діяти в умовах невизначеності та змінної обстановки. Саме інформаційні технології дають змогу поєднати моделювання середовища, аналіз сенсорних потоків і

підтримку прийняття рішень у єдиному контурі підготовки роботехнічних систем.

Отже, зростання кількості техногенних і воєнних загроз, необхідність підвищення безпеки виконання аварійно-рятувальних робіт, а також потреба в удосконаленні методів підготовки роботехнічних систем обумовлюють актуальність розробки спеціалізованого програмного комплексу їх навчання. Реалізація такого підходу сприятиме підвищенню рівня автономності роботів, ефективності їх застосування та зменшенню ризиків для людини під час ліквідації наслідків надзвичайних ситуацій.

Зв'язок роботи з науковими програмами, планами, темами.

Дисертаційна робота виконувалась у відповідності з індивідуальним планом підготовки аспіранта кафедри інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара. Дослідження здійснювалось в рамках науково-дослідної роботи № ФПМ-2-22 «Розроблення програмного забезпечення аналізу та кластеризації часових рядів» 2022-2024 рр. номер держреєстрації 0122U001465 та № 58 – ФПМ-2-25 «Розроблення інформаційної технології обробки статистичних даних» 2025-27рр. номер держреєстрації 0125U002280.

Мета і задачі дослідження. *Метою дисертаційної роботи є проєктування, розробка та тестування програмного забезпечення для навчання роботехнічних систем ліквідації наслідків техногенного походження.*

Для досягнення поставленої мети були поставлені та розв'язані такі наукові задачі:

- здійснено класифікацію роботехнічних систем, що використовуються під час ліквідації наслідків техногенного походження;

- здійснено порівняльний аналіз підходів, що використовуються для автоматизації процесу навчання роботехнічних систем різної конструкції та призначення;

– сформульовано постановку технічної задачі для розробки програмного комплексу навчання роботехнічних систем ліквідації наслідків техногенного походження;

– спроектовано програмний комплекс навчання роботехнічних систем ліквідації наслідків техногенного походження;

– проаналізовано існуючі інструменти та сформовано набір технологій для розробки програмного комплексу навчання роботехнічних систем ліквідації наслідків техногенного походження;

– розроблено програмний комплекс навчання роботехнічних систем ліквідації наслідків техногенного походження;

– здійснено тестування програмного комплексу навчання роботехнічних систем ліквідації наслідків техногенного походження.

Об'єктом дослідження є роботехнічні системи, що використовуються для ліквідації наслідків техногенного походження.

Предметом дослідження є програмний комплекс збору інформації, прийняття рішення та реагування на аварійні ситуації, що виникають унаслідок подій техногенного походження.

Методами дослідження є методи проектування, розробки та тестування програмного забезпечення, методи збору і візуалізації даних та обрані методи системного аналізу. У межах дослідження інформаційні технології розглядаються як методична основа поєднання програмної інженерії, аналізу даних, комп'ютерного зору та імітаційного моделювання.

Достовірність отриманих результатів забезпечується використанням добре апробованих моделей, коректністю математичних та технічних постановок задач, використання методів та алгоритмів, що є теоретично обґрунтованими та апробованими на великій кількості практичних задач. Достовірність результатів забезпечується, у тому числі, узгодженістю між собою числових та експериментальних результатів, несуперечністю

отриманих в роботі результатів відповідним опублікованим результатам інших авторів.

Наукова новизна одержаних результатів полягає у наступному:

1. **Вперше** розроблено концептуальну та програмну модель комплексної системи навчання роботехнічних систем ліквідації наслідків техногенного походження, що інтегрує засоби імітаційного моделювання аварійних сценаріїв, механізми обробки сенсорної інформації, алгоритми автономної навігації та прийняття рішень в єдиному програмному середовищі, орієнтованому на підготовку роботів до роботи в умовах невизначеності та динамічних змін середовища.

2. **Вперше** запропоновано архітектурний підхід до побудови програмного комплексу навчання роботехнічних систем для задач техногенного реагування, який базується на модульному принципі організації компонентів, підтримує масштабування, інтеграцію різнорідних алгоритмів керування та забезпечує можливість адаптації до різних типів роботехнічних платформ.

3. **Удосконалено** метод організації процесу симуляційного навчання роботехнічних систем шляхом формалізації навчальних сценаріїв надзвичайних ситуацій із варіативністю параметрів середовища, що дозволяє підвищити стійкість алгоритмів керування до змін зовнішніх умов та зменшити залежність від натурних експериментів.

4. **Удосконалено** підхід до інтеграції алгоритмів машинного навчання та класичних методів керування у програмному середовищі навчання роботехнічних систем, що забезпечує поєднання адаптивності, керованості та підвищеної надійності функціонування в аварійних сценаріях.

5. **Набули подальшого розвитку** методи програмної підготовки роботехнічних систем до роботи в умовах техногенних аварій за рахунок розширення функціональності симуляційного середовища та впровадження механізмів оцінювання ефективності навчання на основі системи кількісних показників.

6. Набули подальшого розвитку підходи до проектування програмних комплексів спеціалізованого призначення для задач цивільного захисту та аварійно-рятувальних операцій шляхом адаптації принципів інженерії програмного забезпечення до специфіки роботехнічних систем з підвищеними вимогами до надійності та безпеки.

Практичне значення одержаних у дисертаційній роботі результатів полягає в тому, що розроблений програмний комплекс навчання роботехнічних систем ліквідації наслідків техногенного походження може бути безпосередньо використаний для підготовки та тестування роботехнічних платформ, призначених для роботи в умовах надзвичайних ситуацій.

1. Розроблений програмний комплекс забезпечує можливість моделювання сценаріїв техногенних аварій, воєнних руйнувань та інших кризових ситуацій у контрольованому симуляційному середовищі, що дозволяє здійснювати навчання роботів без ризику пошкодження обладнання та без загрози для персоналу.

2. Використання комплексу дозволяє суттєво скоротити витрати на натурні експериментальні випробування, зменшити часові витрати на розробку та налагодження алгоритмів керування, а також підвищити ефективність підготовки роботехнічних систем до реальних умов експлуатації.

3. Запропонована архітектура програмного забезпечення забезпечує масштабованість та можливість інтеграції з різними типами роботехнічних платформ (мобільними наземними, повітряними та маніпуляційними системами), що розширює сферу практичного застосування результатів дослідження.

4. Реалізовані механізми формування навчальних сценаріїв та оцінювання ефективності навчання дозволяють використовувати програмний комплекс як інструмент підтримки прийняття технічних рішень при проектуванні та модернізації роботехнічних систем спеціального призначення.

5. Результати дослідження можуть бути використані у діяльності аварійно-рятувальних служб, підрозділів цивільного захисту, підприємств критичної інфраструктури, а також організацій, що здійснюють роботи з ліквідації наслідків техногенних аварій та відновлення пошкоджених об'єктів.

6. Матеріали дисертаційної роботи можуть бути використані у навчальному процесі закладів вищої освіти при підготовці фахівців за спеціальностями, пов'язаними з інженерією програмного забезпечення, роботехнікою, штучним інтелектом та системами автоматизованого керування.

Таким чином, отримані результати мають прикладний характер і спрямовані на підвищення рівня автономності, ефективності та безпеки використання роботехнічних систем у задачах ліквідації наслідків техногенного походження.

Особистий внесок здобувача. Результати дисертаційної роботи відображено у 8 наукових працях. Усі основні наукові результати, що становлять зміст дисертації та виносяться на захист, отримані здобувачем особисто. Здобувачем здійснено аналіз предметної області, проектування архітектурних рішень, розроблення програмних модулів, постановку чисельних та експериментальних досліджень, а також аналіз, інтерпретацію й узагальнення отриманих результатів. У працях, опублікованих у співавторстві, здобувачу належать: у [1] – участь у програмній реалізації алгоритму моделювання поширення COVID-2019, постановці чисельного експерименту, аналізі та інтерпретації отриманих результатів; у [2] – участь у програмній реалізації алгоритму розв'язання динамічної задачі оптимального розміщення вогневих груп, постановці чисельного експерименту, аналізі та інтерпретації отриманих результатів; у [3] – проектування модульної архітектури програмної системи для динамічної сегментації зображень у системах роботехнічного зору, аналіз підходів до комп'ютерного зору та розпізнавання образів, а також аналіз результатів роботи запропонованої програмної архітектури; у [4] – розроблення динамічної моделі розбиття та програмної

системи для сегментації зображень при реалізації комп'ютерного зору для роботехнічних систем, постановка чисельного експерименту, аналіз та узагальнення отриманих результатів; у [5] – обґрунтування концепції програмного комплексу навчання роботехнічних систем, визначення його архітектури, основних функціональних модулів, підходів до формування навчальних сценаріїв, роботи з даними та тестування програмного комплексу, а також аналіз результатів функціонування програмної реалізації.

У працях, що засвідчують апробацію матеріалів дисертації, здобувачу належать: у [6] – аналіз вимог до програмного комплексу навчання роботехнічних систем, обґрунтування модульної архітектури, визначення основних функціональних компонентів та узагальнення результатів аналізу; у [7] – розроблення програмної реалізації комплексу навчання роботехнічних систем, опис структури програмних модулів, логіки їх взаємодії та аналіз результатів експериментальної реалізації; у [8] – аналіз підходів до навчання систем керування роботехнічних систем, обґрунтування застосування методів машинного навчання, адаптивного навчання та цифрового двійника для підготовки роботехнічних систем до роботи в умовах техногенного походження, а також аналіз і узагальнення отриманих результатів.

Апробація результатів дисертації. Результати дисертаційної роботи доповідались і обговорювались на наукових семінарах «Актуальні питання оптимізації та дискретної математики» при Науковій раді НАН України з проблеми «Кібернетика», який функціонує при Дніпровському національному університеті імені Олеся Гончара (науковий керівник семінару член-кореспондент НАНУ, доктор фіз.-мат. наук, професор Кісельова О.М.); на Міжнародних науково-практичних конференціях, а саме II-й Міжнародній науково-практичній конференції «Актуальні питання сучасної науки: історія, теорія, практика» (2026), II-й Міжнародній науково-практичній конференції «Новітні технології сучасного суспільства» (2026), IV Міжнародної науково-практичної конференції «Сучасні тенденції соціально-гуманітарного розвитку суспільства» (2026).

Публікації. Основні результати дисертаційної роботи опубліковано у 8 наукових працях: 5 статей [1–5] у фахових наукових виданнях України категорії Б, 3 тези доповідей у збірниках матеріалів міжнародних наукових конференцій [6–8].

Структура та обсяг дисертації. Дисертаційна робота складається із вступу, чотирьох розділів, висновків, переліку використаних джерел, що містить 70 найменувань, загальний обсяг дисертації – 166 сторінок, обсяг основного тексту – 148 сторінок. Робота містить 29 рисунків та 3 таблиці.

РОЗДІЛ 1. ОГЛЯД СУЧАСНОГО СТАНУ, ПЕРСПЕКТИВ РОЗВИТКУ ТА МЕТОДІВ ДОСЛІДЖЕННЯ СИСТЕМ НАВЧАННЯ РОБОТЕХНІЧНИХ СИСТЕМ ПРИ ЛІКВІДАЦІЇ НАСЛІДКІВ ТЕХНОГЕННОГО ПОХОДЖЕННЯ

1.1. Роботехнічні системи для роботи в умовах надзвичайних та техногенних ситуацій

Сучасний розвиток роботехніки нерозривно пов'язаний із потребою автоматизації робіт у середовищах, небезпечних для людини. Особливої актуальності це набуває в контексті ліквідації наслідків надзвичайних ситуацій техногенного походження, що характеризуються високим рівнем ризику, невизначеністю умов функціонування та обмеженим часом на прийняття рішень. Техногенні аварії, пожежі, руйнування промислових об'єктів, витіки небезпечних речовин або інфраструктурні катастрофи вимагають залучення спеціалізованих технічних засобів, здатних діяти без безпосередньої участі людини. У цьому контексті роботехнічні системи розглядаються як ключовий інструмент підвищення ефективності аварійно-рятувальних робіт і зниження ризику для особового складу служб реагування [6; 10].

Теоретичні основи побудови роботехнічних комплексів формуються на стику мехатроніки, автоматизованого керування, комп'ютерних технологій та штучного інтелекту. У навчальній і науковій літературі робот розглядається як інтегрована кіберфізична система, що поєднує сенсорні засоби, механічну платформу, виконавчі пристрої та програмне забезпечення для прийняття рішень [1; 5; 11; 21]. Вітчизняні дослідження підкреслюють важливість модульності архітектури роботів, що забезпечує адаптацію платформ до різних умов експлуатації, зокрема до сценаріїв надзвичайних ситуацій [3; 4; 14]. У сучасних умовах саме програмна складова визначає рівень автономності системи та здатність адаптуватися до динамічних змін середовища.

Надзвичайні ситуації техногенного характеру мають низку специфічних особливостей, що визначають вимоги до роботехнічних систем. До них належать невизначеність простору, наявність перешкод, пошкодження інфраструктури, нестабільні поверхні, зашумленість сенсорних даних, а також необхідність роботи в умовах обмеженої видимості та високих температур [22; 23]. У таких умовах традиційні механізми автоматизації виявляються недостатньо ефективними, що стимулює розвиток мобільних роботизованих платформ із розширеними можливостями автономної навігації та дистанційного управління [18; 24].

У міжнародній науковій практиці напрям disaster robotics сформувався як окрема дослідницька галузь, орієнтована на застосування роботів у пошуково-рятувальних операціях. Фундаментальні дослідження у цій сфері представлені роботами R. Murphy, де систематизовано досвід використання роботехнічних систем під час реальних катастроф та визначено основні класи задач для роботів: розвідка, картографування, пошук потерпілих, моніторинг середовища та виконання маніпуляцій [46]. Автор підкреслює, що успішність застосування роботів у надзвичайних ситуаціях суттєво залежить від програмних механізмів адаптації, оскільки жоден сценарій реальної катастрофи не може бути повністю передбачений наперед.

Практичні напрацювання у сфері disaster robotics були значно розширені в межах міжнародних експериментальних програм, зокрема DARPA Robotics Challenge. У межах цього проєкту було продемонстровано можливості антропоморфних та мобільних роботів виконувати складні завдання у зруйнованому середовищі, включаючи пересування пересіченою місцевістю, відкривання дверей, використання інструментів та роботу з елементами інфраструктури [29; 42]. Аналіз результатів конкурсу засвідчив, що попри значний прогрес у розвитку апаратної частини, ключовими обмеженнями залишаються недостатня автономність і складність навчання роботів діям у невизначених умовах.

Одним із важливих аспектів є питання мобільності роботехнічних систем. У роботах, присвячених мобільним роботам, наголошується, що вибір конструкції платформи безпосередньо визначає здатність системи діяти в умовах руйнувань [54; 68; 69]. Колісні платформи характеризуються простотою реалізації та енергоефективністю, однак мають обмежену прохідність. Гусеничні системи краще адаптовані до подолання перешкод, але потребують складніших систем стабілізації. Крокуючі та гібридні платформи демонструють високу маневровість, проте вимагають складних алгоритмів керування. Усі ці підходи об'єднує необхідність використання потужного програмного забезпечення, що забезпечує адаптацію поведінки робота до конкретних умов.

Ключовою тенденцією сучасних досліджень є зміщення акценту з апаратних характеристик на інтелектуалізацію роботехнічних систем. У роботах з інтелектуальної роботехніки підкреслюється, що ефективне функціонування робота визначається здатністю обробляти великі обсяги даних, формувати модель середовища та приймати рішення в реальному часі [17; 24]. Ця тенденція особливо важлива для надзвичайних ситуацій, де середовище часто є хаотичним, а традиційні алгоритми не забезпечують достатньої гнучкості.

Суттєвий вплив на розвиток роботехнічних систем для надзвичайних ситуацій мав сучасний воєнний стан, який сформував нові сценарії використання роботів. Руйнування промислових об'єктів, інфраструктури та енергетичних систем унаслідок бойових дій спричиняють виникнення складних техногенних аварій, що потребують оперативного реагування із залученням роботизованих комплексів [22; 23]. Роботи у таких умовах виконують задачі дистанційної розвідки, обстеження небезпечних зон, моніторингу критичних об'єктів і підтримки рятувальних операцій. Особливість воєнних умов полягає у високій динамічності середовища, що підвищує вимоги до адаптивності програмних систем керування.

Не менш значущим фактором, який вплинув на розвиток роботехніки для надзвичайних ситуацій, стала глобальна пандемія COVID-19. Пандемія продемонструвала необхідність використання автономних систем для мінімізації контактів між людьми та підтримання функціонування критичних процесів. Роботизовані платформи застосовувалися для транспортування медичних матеріалів, дезінфекції приміщень, дистанційного моніторингу та логістичних задач [36; 48]. Хоча ці сценарії відрізняються від класичних техногенних катастроф, вони показали важливість гнучких програмних рішень, здатних швидко адаптуватися до нових умов експлуатації. У результаті пандемія сприяла активному розвитку програмних платформ моделювання та дистанційного навчання роботів, що пізніше стало основою для сучасних підходів до підготовки роботехнічних систем.

Важливим компонентом досліджень є питання взаємодії людини та робота. У надзвичайних ситуаціях повністю автономні системи поки що не можуть забезпечити необхідний рівень надійності, тому широко застосовується модель спільної роботи оператора та робота [56; 60]. Дослідження показують, що ефективність такої взаємодії залежить від інтуїтивності інтерфейсів, швидкості передачі інформації та рівня автоматизації окремих функцій. Це формує нові вимоги до програмного забезпечення, яке має поєднувати автономність із можливістю оперативного втручання оператора.

У роботах щодо застосування роботів у пошуково-рятувальних операціях підкреслюється важливість збору даних про середовище та побудови його цифрових моделей [61; 63]. Роботи виконують роль мобільних сенсорних платформ, що забезпечують отримання актуальної інформації для прийняття управлінських рішень. Саме тому сучасні системи increasingly орієнтовані на інтеграцію засобів картографування, аналізу даних та дистанційної координації декількох роботів у межах єдиної операції.

Аналіз наукових джерел свідчить, що значна частина існуючих рішень розроблялася під конкретні сценарії, що знижує універсальність таких систем.

Спеціалізовані роботи демонструють високу ефективність у вузьких задачах, однак потребують суттєвої адаптації при зміні умов експлуатації [46; 62]. Це особливо критично для техногенних надзвичайних ситуацій, де характер середовища може швидко змінюватися. У зв'язку з цим у сучасних дослідженнях дедалі більша увага приділяється створенню універсальних програмних комплексів, здатних забезпечувати навчання роботехнічних систем у різних сценаріях.

Систематизація наукових підходів дозволяє виділити кілька основних тенденцій розвитку роботехнічних систем для надзвичайних ситуацій. По-перше, відбувається перехід від дистанційно керованих платформ до систем із частковою автономністю. По-друге, зростає роль програмних засобів симуляції та моделювання, які дозволяють готувати роботи до роботи у складних умовах без ризику для обладнання. По-третє, розширюється спектр задач, які охоплюють не лише класичні техногенні аварії, але й нові типи криз, зокрема воєнні дії та пандемічні обмеження.

Таким чином, проведений аналіз літератури свідчить, що роботехнічні системи стають невід'ємним елементом сучасних технологій реагування на надзвичайні ситуації. Разом із тим існуючі дослідження демонструють наявність низки проблем, серед яких ключовими є недостатня універсальність рішень, складність навчання роботів для роботи у невизначених умовах та обмеженість існуючих програмних платформ. Це обґрунтовує необхідність розроблення спеціалізованих програмних комплексів навчання роботехнічних систем, що забезпечують підготовку роботів до виконання задач ліквідації наслідків техногенного походження. Саме ці питання стають логічним підґрунтям для подальшого аналізу архітектур і програмних платформ, що розглядається у наступному підрозділі.

1.2. Архітектури та програмні платформи роботехнічних систем

Ефективність сучасних роботехнічних систем значною мірою визначається не лише конструктивними або апаратними характеристиками, а й програмними архітектурами, які забезпечують інтеграцію сенсорів, виконавчих механізмів, алгоритмів керування та засобів взаємодії з оператором. З розвитком роботехніки відбулося поступове зміщення акценту від автономних механічних пристроїв до складних програмно-апаратних комплексів, у яких програмне забезпечення фактично виступає центральним елементом системи [1; 11; 21]. Такий підхід особливо важливий для роботехнічних платформ, що застосовуються в умовах надзвичайних ситуацій техногенного походження, де необхідна висока адаптивність до змін зовнішнього середовища та швидка інтеграція нових функцій. У цьому контексті інформаційні технології виступають засобом інтеграції апаратних компонентів, алгоритмів керування та інтерфейсів взаємодії з оператором.

У загальному розумінні архітектура роботехнічної системи визначає спосіб організації її компонентів, структуру обміну даними, рівні управління та принципи взаємодії між програмними модулями. Базові моделі архітектур описані у класичних роботах з роботехніки, де виділяються ієрархічні, реактивні та гібридні підходи [32; 67]. Ієрархічні архітектури передбачають чіткий поділ на рівні стратегічного планування, тактичного управління та низькорівневого контролю, що забезпечує структурованість системи, але може обмежувати швидкість реакції на зміни середовища. Реактивні архітектури орієнтовані на безпосередню взаємодію з середовищем і забезпечують високу швидкість реагування, однак часто мають обмежені можливості довгострокового планування. Гібридні підходи поєднують переваги обох моделей і сьогодні є найбільш поширеними у практиці побудови роботехнічних систем.

Розвиток програмних платформ у роботехніці суттєво прискорився завдяки появі відкритих модульних фреймворків. Найбільш відомою

платформою є Robot Operating System (ROS), яка забезпечує стандартизований підхід до розроблення програмних компонентів для роботів [47]. ROS дозволяє організовувати програмне забезпечення у вигляді незалежних вузлів, що обмінюються повідомленнями через стандартизовані інтерфейси. Така модульність значно спрощує масштабування систем і дозволяє швидко інтегрувати нові алгоритми або апаратні компоненти. У наукових дослідженнях ROS розглядається як де-факто стандарт для експериментальних та прикладних роботехнічних систем, зокрема у сферах автономної навігації, маніпулювання об'єктами та багатороботних систем [31; 48].

Важливим елементом сучасних програмних архітектур є використання симуляційних середовищ, які дозволяють тестувати алгоритми керування без ризику пошкодження обладнання. У роботах Koenig та Howard описано можливості симулятора Gazebo, що забезпечує фізичне моделювання руху роботів, взаємодію з об'єктами та імітацію сенсорних даних [40]. Інші середовища, такі як MuJoCo, спрямовані на високоточне моделювання динаміки складних механічних систем і широко використовуються в задачах навчання роботів [53]. Порівняльний аналіз симуляторів показує, що вибір програмної платформи залежить від цілей дослідження: для задач навігації і багатороботної взаємодії ефективними є універсальні середовища, тоді як для навчання політик керування перевагу мають фізично точні симулятори [34].

У контексті ліквідації наслідків надзвичайних ситуацій симуляційні платформи набувають особливого значення. Реальні експерименти у небезпечних умовах часто є дорогими, складними та потенційно ризикованими, тому попереднє навчання роботів у віртуальному середовищі стає необхідною умовою розроблення ефективних систем [46; 62]. Моделювання дозволяє відтворювати сценарії руйнувань, пожеж, обмеженої видимості або нестабільних поверхонь, що створює передумови для перевірки алгоритмів до їх впровадження у реальних умовах.

Архітектурні рішення значною мірою визначають рівень автономності роботехнічних систем. У класичних підходах керування більшість рішень приймалася оператором, тоді як сучасні системи дедалі частіше використовують алгоритми автономного планування та адаптивної поведінки [43; 51]. Це обумовлює необхідність створення програмних платформ, які забезпечують інтеграцію різнорівневих алгоритмів — від базового управління приводами до високорівневих систем прийняття рішень на основі штучного інтелекту [36; 50]. У літературі підкреслюється, що саме програмна архітектура визначає можливість масштабування автономності систем без повної перебудови апаратної частини.

Суттєвий внесок у розвиток архітектур роботехнічних систем зробили дослідження в області багатороботних систем. Координація декількох роботів у межах однієї операції дозволяє підвищити ефективність виконання задач пошуку та моніторингу, однак потребує складної програмної інфраструктури для обміну даними та синхронізації дій [37; 55]. У таких системах особливо важливими є механізми розподіленого прийняття рішень і забезпечення стійкості до втрати зв'язку, що є типовим для умов надзвичайних ситуацій.

Питання архітектур програмного забезпечення набули особливої актуальності в умовах воєнного стану, коли роботехнічні системи повинні швидко адаптуватися до нових завдань. Практика застосування роботизованих платформ у зонах бойових дій показала необхідність використання модульних програмних рішень, які дозволяють оперативно змінювати функціональність систем залежно від поточних потреб [22; 23]. У таких умовах програмна архітектура повинна підтримувати швидке оновлення алгоритмів, інтеграцію нових сенсорів та роботу в умовах нестабільної комунікаційної інфраструктури.

Окремої уваги заслуговує вплив пандемії COVID-19 на розвиток програмних платформ роботехнічних систем. У період пандемії різко зросла потреба у дистанційному керуванні роботами, телеприсутності та автоматизованих логістичних системах. Це стимулювало розвиток хмарних

архітектур та розподілених систем керування, що дозволяють операторам взаємодіяти з роботами на значній відстані [36; 48]. Таким чином, програмні платформи почали орієнтуватися не лише на локальну автономність, а й на інтеграцію із мережевими сервісами та віддаленими обчислювальними ресурсами.

Аналіз наукових джерел показує, що сучасні архітектури роботехнічних систем поступово еволюціонують у напрямі сервісно-орієнтованих моделей. У таких системах окремі функції роботів реалізуються як незалежні програмні сервіси, що можуть повторно використовуватися у різних сценаріях [31; 47]. Це особливо важливо для задач навчання роботів, оскільки дозволяє відокремити процес формування навичок від конкретної апаратної реалізації. У перспективі такі архітектури створюють основу для формування універсальних програмних комплексів навчання.

Незважаючи на значний прогрес, сучасні програмні платформи мають низку обмежень. По-перше, більшість із них орієнтована на дослідницькі або лабораторні умови, тоді як робота у надзвичайних ситуаціях вимагає підвищеної надійності та стійкості до відмов. По-друге, інтеграція алгоритмів машинного навчання у реальні системи часто супроводжується проблемами сумісності та високими вимогами до обчислювальних ресурсів [49; 52]. По-третє, існуючі платформи не завжди забезпечують зручні засоби для формалізації навчальних сценаріїв, що ускладнює підготовку роботів до роботи у специфічних умовах техногенних катастроф.

Важливим аспектом є також питання стандартизації програмних архітектур. Відсутність єдиних підходів до опису функціональних модулів, форматів даних та інтерфейсів взаємодії ускладнює інтеграцію рішень, розроблених різними дослідницькими групами. У результаті багато систем залишаються ізольованими та не можуть бути ефективно використані в комплексних аварійно-рятувальних операціях [48; 54].

Отже, аналіз літератури свідчить, що розвиток архітектур і програмних платформ роботехнічних систем є одним із ключових напрямів сучасних

досліджень. Перехід до модульних, сервісно-орієнтованих та симуляційно-орієнтованих архітектур створює передумови для розроблення програмних комплексів навчання роботехнічних систем, здатних адаптуватися до різноманітних сценаріїв надзвичайних ситуацій. Водночас наявні рішення не забезпечують повною мірою інтеграцію навчальних процесів, моделювання та реального застосування роботів у динамічних умовах техногенних катастроф, що визначає необхідність подальших досліджень у цьому напрямі.

1.3. Методи навігації, локалізації та взаємодії роботехнічних систем із середовищем

Однією з ключових складових функціонування роботехнічних систем, призначених для роботи в умовах надзвичайних ситуацій техногенного походження, є здатність автономно або напівавтономно орієнтуватися у складному, динамічному та часто частково невизначеному середовищі. Саме системи навігації, локалізації та взаємодії із зовнішнім простором визначають ефективність виконання задач розвідки, пошуку постраждалих, моніторингу небезпечних зон і підтримки рятувальних операцій. У сучасній роботехніці ці функції базуються на поєднанні математичних моделей руху, сенсорних даних, алгоритмів картографування та методів планування траєкторій [32; 51; 68].

Класичні підходи до навігації мобільних роботів ґрунтуються на побудові внутрішньої моделі середовища та визначенні положення робота відносно цієї моделі. У роботах із теорії автономних мобільних роботів підкреслюється, що навігація передбачає вирішення трьох взаємопов'язаних задач: локалізація, побудова карти та планування маршруту [54; 69]. Для структурованих середовищ ці задачі можуть вирішуватися окремо, однак у реальних умовах надзвичайних ситуацій вони взаємозалежні та часто виконуються одночасно в режимі реального часу.

Однією з фундаментальних проблем роботехніки є задача одночасної локалізації та картографування (Simultaneous Localization and Mapping,

SLAM). У класичних роботах Bailey та Durrant-Whyte показано, що SLAM є базовим механізмом забезпечення автономності мобільних роботів у невідомому середовищі [26; 33]. Алгоритми SLAM дозволяють роботу одночасно визначати власне положення та формувати карту простору на основі сенсорних вимірювань. У подальших дослідженнях концепція SLAM була значно розширена: сучасні підходи включають використання лазерних сканерів, камер, інерціальних систем та багатосенсорної інтеграції [30; 58; 59].

Для роботів, що працюють у техногенних надзвичайних ситуаціях, задача SLAM ускладнюється через високий рівень завад, динамічність середовища та можливу відсутність стабільних орієнтирів. У роботах Cadena та співавторів зазначено, що традиційні алгоритми SLAM часто демонструють знижену стійкість у середовищах із руйнуваннями або великою кількістю рухомих об'єктів [30]. Це стимулювало розвиток більш робастних методів, здатних працювати в умовах часткової втрати даних або зміни структури простору.

Важливе значення для забезпечення навігації мають методи локалізації. Одним із класичних підходів є Монте-Карло локалізація, що базується на ймовірнісному представленні положення робота [38]. Такий підхід дозволяє враховувати невизначеність сенсорних даних та забезпечує стійкість до помилок вимірювань. У книзі «Probabilistic Robotics» детально описані байєсівські методи оцінювання стану робота, які сьогодні становлять основу багатьох сучасних систем навігації [51]. Ймовірнісні моделі особливо ефективні у сценаріях аварійно-рятувальних робіт, де середовище є непередбачуваним, а дані сенсорів часто зашумлені.

Не менш важливою складовою є планування руху робототехнічних систем. Теоретичні основи цієї області представлені у роботах LaValle, де запропоновано універсальні алгоритмічні підходи до побудови траєкторій у просторі станів [43]. У практичних реалізаціях широко застосовуються методи потенціальних полів, зокрема підхід Khatib, який дозволяє роботам уникати перешкод у реальному часі [39]. Подальший розвиток отримали алгоритми

динамічного уникнення перешкод, наприклад *dynamic window approach*, що ефективно працює для мобільних платформ у змінному середовищі [35].

У контексті надзвичайних ситуацій питання навігації набуває додаткової складності через обмеження комунікації та нестабільність середовища. Роботи часто повинні діяти в умовах неповної карти, обмеженої видимості або порушеної інфраструктури, що робить традиційні методи планування недостатніми. У таких умовах актуальними стають адаптивні алгоритми, здатні перебудовувати маршрути в процесі руху та враховувати нові дані від сенсорів [54; 68].

Особливе місце посідають сучасні методи, що базуються на використанні машинного навчання для навігації. Дослідження показують, що глибинні нейронні мережі можуть навчатися формувати політики руху без явного програмування правил, що відкриває нові можливості для адаптації роботів до складних середовищ [44; 45]. Разом із тим зазначається, що використання таких підходів у реальних аварійно-рятувальних сценаріях потребує ретельної перевірки через ризики непередбачуваної поведінки систем.

Значний прогрес у навігації роботів забезпечило використання тривимірного картографування на основі LiDAR-сенсорів. Метод LOAM, запропонований Zhang та Singh, демонструє високу точність побудови карт у режимі реального часу та широко застосовується у мобільній роботехніці [58]. Для задач ліквідації наслідків техногенних аварій такі підходи дозволяють створювати детальні моделі зруйнованого середовища та покращувати ситуаційну обізнаність операторів.

Поряд із автономною навігацією важливим напрямом є взаємодія роботехнічної системи з людиною-оператором. У працях, присвячених людино-роботній взаємодії, зазначено, що оператор залишається ключовим елементом системи в умовах високої невизначеності [56; 60]. Ефективність взаємодії залежить від якості візуалізації даних навігації, точності відображення карти середовища та рівня автоматизації прийняття рішень.

Тому сучасні системи навігації мають поєднувати автономні алгоритми з можливістю оперативного втручання людини.

В умовах воєнного стану проблематика навігації роботів набуває нових аспектів. Руйнування інфраструктури, поява небезпечних зон та нестабільність навколишнього середовища вимагають від роботів здатності працювати без попередньо підготовлених карт [22; 23]. У таких сценаріях важливого значення набуває швидке формування локальної карти та адаптивне планування маршрутів, що забезпечує можливість виконання задач розвідки або моніторингу без ризику для персоналу.

Досвід пандемії COVID-19 також вплинув на розвиток навігаційних систем. Роботи, що використовувалися для логістичних або сервісних задач, повинні були безпечно пересуватися в середовищах із людьми, враховуючи соціальні обмеження та динамічні перешкоди [36; 48]. Це сприяло розвитку алгоритмів навігації, орієнтованих на прогнозування поведінки людей та безпечну взаємодію в спільному просторі.

Варто зазначити, що сучасні дослідження дедалі частіше розглядають навігацію як компонент комплексної системи навчання роботів. Застосування reinforcement learning та симуляційного навчання дозволяє формувати навігаційні стратегії у віртуальних середовищах з подальшим перенесенням у реальні умови [41; 52]. Однак проблема sim-to-real transfer залишається однією з ключових, оскільки моделі, навчені в симуляції, не завжди демонструють стабільну роботу в реальних аварійних умовах.

Аналіз літератури дозволяє зробити висновок, що сучасні методи навігації та локалізації пройшли значну еволюцію — від детермінованих алгоритмів до ймовірнісних моделей і методів машинного навчання. Проте для задач ліквідації наслідків техногенних надзвичайних ситуацій залишається низка невирішених проблем: недостатня стійкість алгоритмів до хаотичних змін середовища, обмежена адаптивність до нових сценаріїв та складність інтеграції різномірних сенсорних даних у єдину систему прийняття рішень.

Таким чином, ефективна навігація роботехнічних систем у надзвичайних ситуаціях потребує комплексного підходу, що поєднує класичні алгоритми локалізації, сучасні методи машинного навчання та розвинуті механізми взаємодії з оператором. Наявні дослідження створюють потужну теоретичну основу, проте не забезпечують повністю інтегрованих рішень для навчання роботів у різноманітних сценаріях техногенних аварій.

1.4. Методи машинного навчання і навчання з підкріпленням у роботехніці

Сучасний етап розвитку роботехнічних систем характеризується активним впровадженням методів машинного навчання, що забезпечують здатність роботів адаптуватися до складних і непередбачуваних умов функціонування. На відміну від класичних алгоритмічних підходів, які передбачають жорстко визначені правила поведінки, машинне навчання дозволяє формувати політики керування на основі даних, що особливо важливо для застосування роботехнічних систем у надзвичайних ситуаціях техногенного походження. У таких умовах середовище часто є частково невідомим, динамічним та погано формалізованим, що обмежує ефективність традиційних методів керування [36; 50].

Фундаментальні основи використання машинного навчання в роботехніці закладені у працях, присвячених глибинному навчанню та навчанню з підкріпленням. У дослідженнях Goodfellow, Bengio та Courville описані принципи побудови нейронних мереж, які здатні обробляти великі обсяги сенсорних даних, формувати складні ознаки та використовувати їх для прийняття рішень [36]. Впровадження таких підходів у роботехніку дозволило перейти від ручного програмування поведінки до автоматичного набуття навичок на основі досвіду.

Одним із найбільш перспективних напрямів є навчання з підкріпленням (reinforcement learning, RL), що розглядає процес навчання як послідовність взаємодій агента із середовищем з метою максимізації довгострокової

винагороди. Теоретичні основи RL детально викладені у класичній праці Sutton і Barto [50], яка визначає ключові концепції — політику дій, функцію цінності та процес дослідження середовища. У контексті роботехніки ці методи дозволяють формувати складну поведінку роботів без необхідності ручного визначення всіх сценаріїв функціонування.

Важливим етапом розвитку RL стало застосування глибинних нейронних мереж для апроксимації функцій керування. Робота Mnih та співавторів продемонструвала можливість досягнення високих результатів шляхом поєднання нейронних мереж і навчання з підкріпленням [45]. Надалі ці ідеї були перенесені у роботехніку, де вони дали змогу роботам навчатися маніпуляціям, навігації та ухваленню рішень у складних середовищах [44].

У роботехнічних задачах важливе місце займає imitation learning — навчання шляхом наслідування дій експерта. У роботі Abbeel та Ng запропоновано підхід інверсного навчання з підкріпленням, який дозволяє відтворювати поведінку людини-оператора на основі спостережуваних дій [25]. Такий підхід є особливо актуальним для задач ліквідації наслідків техногенних аварій, де накопичений досвід рятувальників може бути використаний для формування початкових стратегій роботів.

Застосування машинного навчання у роботехніці має низку специфічних особливостей. На відміну від віртуальних агентів, робот працює у фізичному середовищі, де помилки можуть призводити до пошкодження обладнання або небезпечних ситуацій. Саме тому значна частина досліджень зосереджена на використанні симуляцій для навчання роботів. Симуляційне навчання дозволяє генерувати великі обсяги даних без ризику та суттєво прискорює процес формування навичок [34; 53].

Однією з ключових проблем є перенесення навчених моделей із симуляційного середовища у реальний світ, відоме як sim-to-real transfer. У роботі Tobin та співавторів запропоновано підхід domain randomization, що передбачає випадкову зміну параметрів симуляції під час навчання для підвищення стійкості моделей у реальних умовах [52]. Цей метод виявився

особливо корисним для роботів, що працюють у непередбачуваних середовищах, зокрема у сценаріях техногенних аварій.

Огляд сучасних досліджень свідчить, що машинне навчання активно використовується для вирішення задач навігації, маніпулювання об'єктами та взаємодії з середовищем [41; 49]. Навчання з підкріпленням дозволяє роботам формувати політики руху, які враховують складну динаміку системи та обмеження навколишнього середовища. При цьому підкреслюється, що ефективність алгоритмів значною мірою залежить від архітектури програмної платформи та якості навчального середовища.

Особливої актуальності методи машинного навчання набувають у задачах disaster robotics. У роботах Murphy та інших дослідників зазначається, що традиційні алгоритми часто не здатні забезпечити достатню гнучкість у хаотичних середовищах, характерних для надзвичайних ситуацій [46; 62]. Навчальні підходи дозволяють роботам адаптувати поведінку до нових умов, враховуючи досвід попередніх дій. Це відкриває можливості для створення систем, здатних ефективно діяти в умовах часткової невизначеності.

Важливим напрямом є використання машинного навчання для багатороботних систем. У таких системах алгоритми навчання застосовуються для координації дій декількох роботів, розподілу задач і забезпечення колективної поведінки [37; 55]. Це особливо актуально для аварійно-рятувальних операцій, де одночасне використання декількох платформ дозволяє значно скоротити час виконання завдань.

Події воєнного стану продемонстрували необхідність швидкої адаптації робототехнічних систем до нових умов експлуатації. Сценарії, пов'язані з руйнуванням інфраструктури та високою динамічністю середовища, вимагають алгоритмів, здатних до самоадаптації та перенавчання без повної реконфігурації системи [22; 23]. Методи машинного навчання забезпечують таку гнучкість, проте ставлять питання щодо надійності та передбачуваності поведінки роботів у критичних ситуаціях.

Пандемія COVID-19 також стала каталізатором розвитку навчальних підходів у роботехніці. Під час пандемії зросла потреба в автономних системах, здатних виконувати логістичні та сервісні задачі в умовах обмеженої участі людини. Це сприяло активному розвитку алгоритмів, що дозволяють роботам працювати в середовищах із людьми, адаптуючись до змін поведінки та динамічних обмежень [36; 48]. Отриманий досвід підтвердив важливість гнучких програмних комплексів навчання, здатних швидко адаптуватися до нових умов.

Попри значний прогрес, застосування машинного навчання у роботехніці має низку обмежень. По-перше, навчання моделей потребує значних обчислювальних ресурсів і великих масивів даних. По-друге, результати навчання часто є складними для інтерпретації, що ускладнює верифікацію безпеки поведінки роботів [49]. По-третє, існує проблема стабільності навчання, коли алгоритми демонструють нестійку поведінку в нових сценаріях.

У сучасних дослідженнях активно розглядаються гібридні підходи, які поєднують класичні алгоритми керування з методами машинного навчання. Такий підхід дозволяє забезпечити базову безпечну поведінку роботів, одночасно надаючи їм можливість адаптації до нових умов [41; 65]. Для задач ліквідації наслідків техногенних аварій це є особливо важливим, оскільки помилки автономних систем можуть мати критичні наслідки.

Таким чином, аналіз літератури свідчить, що машинне навчання та навчання з підкріпленням відіграють ключову роль у сучасному розвитку роботехнічних систем. Вони забезпечують можливість адаптації роботів до складних умов, формування поведінки на основі досвіду та підвищення рівня автономності. Разом із тим залишаються невирішені питання щодо надійності, безпеки та інтеграції навчальних методів у реальні роботехнічні комплекси для надзвичайних ситуацій. Це обґрунтовує необхідність розгляду аспектів людино-роботної взаємодії та безпечного керування, що становить зміст наступного підрозділу.

1.5. Людино-роботна взаємодія та безпечне керування в аварійних сценаріях

Одним із визначальних чинників ефективності роботехнічних систем, що застосовуються у надзвичайних ситуаціях техногенного походження, є організація взаємодії між людиною-оператором і роботехнічним комплексом. Попри стрімкий розвиток автономних алгоритмів та методів машинного навчання, більшість сучасних систем, які використовуються в аварійно-рятувальних операціях, залишаються напівавтономними та потребують участі людини у процесі прийняття рішень. Це обумовлено високим рівнем невизначеності середовища, ризиками для безпеки та необхідністю оперативної адаптації до непередбачуваних умов [46; 56].

Людино-роботна взаємодія (Human–Robot Interaction, HRI) розглядається як міждисциплінарна область, що поєднує роботехніку, когнітивні науки, ергономіку та системи управління. У роботах Yanco та Drury запропоновано класифікацію взаємодії людини і робота залежно від рівня автономності, типу задачі та умов середовища [56]. Для надзвичайних ситуацій характерною є модель спільного керування, коли робот виконує частину функцій автономно, а оператор здійснює стратегічний контроль та приймає критичні рішення.

Практика застосування роботів у пошуково-рятувальних операціях показує, що ефективність HRI значною мірою залежить від якості інтерфейсів керування. У дослідженнях Murphy та співавторів зазначено, що оператори часто стикаються з інформаційним перевантаженням через велику кількість сенсорних даних, які надходять у реальному часі [46; 63]. Тому сучасні системи мають забезпечувати адаптивну візуалізацію інформації та пріоритезацію критично важливих даних, що дозволяє знизити когнітивне навантаження на оператора.

Безпечне керування роботехнічними системами є ключовою вимогою для застосування в аварійних сценаріях. У класичних роботах із керування

маніпуляторами досліджуються методи забезпечення стабільності руху та безпечної взаємодії з середовищем [57]. Підходи, засновані на імпедансному та пасивному керуванні, дозволяють обмежувати сили взаємодії між роботом і об'єктами, що особливо важливо під час роботи в умовах руйнувань або у присутності людей.

У контексті мобільних роботів безпека пов'язана насамперед із навігацією та уникненням перешкод. Алгоритми динамічного планування траєкторії, поєднані з системами автономного контролю, дозволяють мінімізувати ризик зіткнень навіть у складних середовищах [35; 39]. Проте у реальних надзвичайних ситуаціях повна автономність часто є недосяжною, тому в більшості випадків реалізується концепція *shared control* — розподіленого керування між людиною та системою [60].

Особливої актуальності питання безпеки набувають у сценаріях, пов'язаних із воєнним станом. Роботи можуть застосовуватися для обстеження зруйнованих об'єктів, дистанційного моніторингу небезпечних зон або виконання операцій у середовищах із підвищеною загрозою. У таких умовах оператори часто працюють під значним психологічним навантаженням, що підвищує ризик помилок управління [22; 23]. Це підкреслює важливість розроблення інтелектуальних інтерфейсів, здатних підтримувати оператора та автоматично запобігати небезпечним діям.

Досвід використання роботизованих систем під час пандемії COVID-19 також сприяв розвитку підходів до людино-роботної взаємодії. Роботи застосовувалися для транспортування матеріалів, дезінфекції приміщень та дистанційної взаємодії з пацієнтами, що вимагало безпечного співіснування роботів і людей у спільному просторі [36; 48]. У результаті були розроблені нові механізми адаптивної навігації та поведінки роботів у середовищах із непередбачуваною людською активністю.

У сучасних дослідженнях значна увага приділяється ролі автономності в HRI. З одного боку, підвищення автономності дозволяє зменшити навантаження на оператора та прискорити виконання задач. З іншого —

надмірна автономність може знижувати рівень контролю людини над системою та викликати недовіру до рішень робота [49]. Це породжує концепцію адаптивної автономності, коли рівень незалежності робота змінюється залежно від складності ситуації та стану оператора.

Важливим аспектом є також питання навчання операторів робототехнічних систем. У літературі зазначається, що ефективність використання роботів у надзвичайних ситуаціях значною мірою залежить від підготовки персоналу та наявності тренувальних середовищ [1; 20]. Симуляційні платформи дозволяють відпрацьовувати сценарії взаємодії без ризику для техніки або людей, що є важливою складовою сучасних програмних комплексів навчання.

З розвитком методів машинного навчання та штучного інтелекту виникає питання довіри до автономних систем. Дослідники підкреслюють, що оператори повинні розуміти логіку прийняття рішень роботом, інакше ефективність взаємодії різко знижується [49; 60]. Це стимулює розвиток підходів explainable AI, спрямованих на підвищення прозорості алгоритмів і покращення взаєморозуміння між людиною та машиною.

У багатороботних системах HRI ускладнюється необхідністю одночасного керування кількома платформами. Дослідження показують, що ефективно управління групою роботів можливе лише за умови високого рівня автоматизації рутинних дій та інтелектуального розподілу задач між агентами [37; 55]. У надзвичайних ситуаціях це дозволяє оператору концентруватися на стратегічних рішеннях, делегуючи виконання локальних дій автономним алгоритмам.

Аналіз літератури також свідчить, що безпека взаємодії визначається не лише технічними аспектами, а й організаційними факторами. Під час аварійно-рятувальних операцій роботи інтегруються у вже існуючі структури управління, тому програмні комплекси повинні підтримувати сумісність із прийнятими процедурами роботи служб реагування [7; 46]. Це висуває додаткові вимоги до інтерфейсів та протоколів обміну даними.

Важливим напрямом розвитку HRI є створення інтелектуальних асистивних систем, які допомагають оператору приймати рішення на основі аналізу даних із сенсорів. Такі системи можуть автоматично пропонувати альтернативні маршрути, попереджати про потенційні ризики або виконувати частину задач без прямої участі людини [41; 52]. У перспективі це дозволяє перейти до більш ефективної моделі співпраці між людиною і роботом.

Отже, людино-роботна взаємодія та безпечне керування є невід’ємними складовими сучасних роботехнічних систем для роботи в умовах надзвичайних ситуацій. Аналіз наукових джерел показує, що попри зростання рівня автономності, роль людини залишається ключовою у прийнятті рішень та забезпеченні безпеки операцій. Разом із тим існуючі підходи часто не забезпечують достатньої інтеграції між навчанням роботів, моделями взаємодії та реальними сценаріями застосування.

1.6. Аналіз існуючих підходів та постановка наукової задачі

Проведений аналіз літературних джерел засвідчив, що сучасний розвиток роботехнічних систем відбувається в напрямі зростання рівня автономності, інтелектуалізації процесів керування та інтеграції машинного навчання у практичні сценарії експлуатації. Роботи дедалі активніше застосовуються для виконання завдань у середовищах із підвищеним рівнем небезпеки, зокрема під час ліквідації наслідків техногенних аварій, проведення пошуково-рятувальних операцій та моніторингу критичних об’єктів [46; 62]. Разом із тим огляд наукових публікацій показує, що незважаючи на значний прогрес, існуючі рішення не формують цілісної методологічної бази для навчання роботехнічних систем саме у контексті надзвичайних ситуацій техногенного походження.

Аналіз базових підходів до побудови роботехнічних систем демонструє, що значна частина досліджень зосереджена на конструктивних і мехатронних аспектах платформ [1; 11; 21], тоді як питання інтегрованого програмного забезпечення часто розглядаються фрагментарно. Навіть у сучасних роботах,

присвячених інтелектуальним системам керування, акцент переважно робиться на окремих алгоритмах або підсистемах, а не на комплексній організації процесу навчання роботів [3; 17; 24]. Це призводить до того, що розроблені рішення мають високий рівень спеціалізації та складно адаптуються до нових сценаріїв.

Дослідження в галузі disaster robotics доводять ефективність робототехнічних систем у реальних аварійних ситуаціях, проте одночасно виявляють низку фундаментальних проблем. Зокрема, R. Murphy зазначає, що більшість роботів, застосованих під час реальних катастроф, потребували значної участі операторів та мали обмежену автономність [46]. Це пояснюється складністю роботи у непередбачуваному середовищі, де традиційні алгоритми не здатні забезпечити стабільну поведінку системи.

Розгляд сучасних програмних платформ показує, що існуючі рішення, такі як ROS, Gazebo або MuJoCo, створюють потужну основу для розробки робототехнічних систем [40; 47; 53]. Однак ці платформи орієнтовані насамперед на дослідницькі або навчальні задачі та не завжди підтримують повний цикл підготовки роботів до експлуатації у надзвичайних ситуаціях. Зокрема, бракує стандартизованих механізмів створення сценаріїв аварійних подій, інтеграції навчальних алгоритмів із системами моніторингу ризиків та оцінювання ефективності поведінки роботів у складних умовах.

Аналіз методів навігації і локалізації свідчить про значний прогрес у розвитку алгоритмів SLAM, ймовірнісної локалізації та планування руху [26; 30; 51]. Проте більшість алгоритмів розроблялися для відносно стабільних або напівструктурованих середовищ. У реальних техногенних аваріях середовище часто змінюється в процесі виконання завдань, що знижує ефективність традиційних підходів [58; 59]. Це підтверджує необхідність навчання роботів працювати в умовах невизначеності та неповноти інформації.

Розвиток машинного навчання та навчання з підкріпленням відкрив нові можливості для формування адаптивної поведінки роботів [36; 50]. Дослідження показують, що такі методи здатні забезпечувати автоматичне

набуття навичок на основі досвіду [25; 45]. Проте застосування цих підходів у реальних аварійно-рятувальних операціях залишається обмеженим через проблеми безпеки, складність валідації алгоритмів та труднощі перенесення моделей із симуляції у реальне середовище [41; 52]. Таким чином, існує розрив між теоретичними можливостями машинного навчання і практичними потребами систем реагування на надзвичайні ситуації.

Аналіз досліджень у сфері людино-роботної взаємодії показує, що навіть найбільш сучасні автономні системи потребують участі оператора, особливо у критичних сценаріях [56; 60]. Це обумовлено необхідністю швидкого прийняття рішень у ситуаціях, де помилки можуть мати катастрофічні наслідки. Водночас у літературі підкреслюється, що ефективність взаємодії значною мірою залежить від якості програмних інтерфейсів та рівня автоматизації рутинних операцій [63]. Наявні рішення часто не забезпечують достатньої інтеграції між навчальними процесами та реальними умовами роботи оператора.

Суттєвий вплив на формування нових вимог до роботехнічних систем здійснили глобальні події останніх років. Воєнний стан та пов'язані з ним техногенні загрози продемонстрували необхідність використання роботів у середовищах із високою динамікою та нестабільною інфраструктурою [22; 23]. У таких умовах традиційні підходи до програмування роботів виявляються недостатньо гнучкими, що підсилює потребу у системах навчання, здатних швидко адаптувати поведінку платформ.

Пандемія COVID-19, у свою чергу, підкреслила важливість дистанційного керування та автономних рішень для мінімізації ризику контактів між людьми [36; 48]. Роботи почали активно використовуватися в медичних і логістичних сценаріях, що продемонструвало необхідність універсальних програмних платформ, здатних швидко змінювати функціональність залежно від нових умов. Отриманий досвід став каталізатором розвитку підходів до дистанційного навчання та симуляційної підготовки роботів.

Систематизація літературних джерел дозволяє виділити основні проблеми сучасних підходів:

- фрагментарність існуючих програмних рішень та відсутність єдиної інтегрованої платформи для навчання роботехнічних систем [47; 53];
- обмежена адаптивність роботів до динамічних умов надзвичайних ситуацій [46; 62];
- складність перенесення алгоритмів машинного навчання із симуляцій у реальні сценарії [41; 52];
- недостатня інтеграція людино-роботної взаємодії у процес навчання систем [56; 60];
- відсутність універсальних моделей формування навчальних сценаріїв для техногенних аварій.

Порівняльний аналіз наукових праць свідчить, що більшість дослідників розглядають окремі аспекти проблеми — навігацію, машинне навчання, архітектуру ПЗ або безпечне керування — без створення єдиної концептуальної моделі. У результаті виникає наукова проблема, пов'язана з необхідністю інтеграції різнорідних підходів у межах єдиного програмного комплексу навчання роботехнічних систем.

Таким чином, на основі проведеного огляду формулюється наукова задача дослідження: розроблення програмного комплексу навчання роботехнічних систем ліквідації наслідків техногенного походження, що забезпечує інтеграцію симуляційного моделювання, алгоритмів машинного навчання, засобів навігації та механізмів людино-роботної взаємодії в єдиному середовищі підготовки роботів до роботи у складних і динамічних умовах.

Вирішення цієї задачі передбачає:

- аналіз і синтез архітектури програмного комплексу, придатного для моделювання різних сценаріїв надзвичайних ситуацій;
- розроблення методів навчання роботів із використанням симуляційного середовища;

- інтеграцію механізмів адаптивного керування та взаємодії з оператором;
- забезпечення можливості перенесення результатів навчання у реальні роботехнічні системи.

Отже, результати огляду літератури підтверджують актуальність дослідження та обґрунтовують необхідність створення нового підходу до навчання роботехнічних систем, який поєднуватиме переваги сучасних алгоритмів і програмних платформ із вимогами практичного застосування у сфері ліквідації наслідків техногенних надзвичайних ситуацій. Це визначає логіку подальших розділів дисертаційної роботи, присвячених розробленню концепції та реалізації відповідного програмного комплексу.

1.7. Висновки до розділу 1

Проведений огляд літературних джерел дозволив комплексно проаналізувати сучасний стан досліджень у галузі роботехнічних систем, орієнтованих на виконання завдань у умовах надзвичайних ситуацій техногенного походження, а також визначити ключові тенденції розвитку, наявні обмеження та напрями подальших досліджень. Аналіз охопив як фундаментальні теоретичні засади роботехніки, так і сучасні підходи до програмного забезпечення, навігації, машинного навчання, людино-роботної взаємодії та практичного використання роботів у кризових сценаріях.

У результаті аналізу встановлено, що роботехнічні системи поступово стають невід'ємною частиною технологій реагування на надзвичайні ситуації. Використання роботів дозволяє знизити ризики для персоналу, забезпечити доступ до небезпечних зон і підвищити оперативність виконання аварійно-рятувальних робіт. Водночас більшість існуючих рішень мають спеціалізований характер і розроблялися під конкретні сценарії застосування, що обмежує їх універсальність та адаптивність до нових умов експлуатації. Особливо це проявляється у техногенних аваріях, де структура середовища

може швидко змінюватися, а типи задач часто виходять за межі заздалегідь визначених моделей.

Аналіз архітектур і програмних платформ показав, що сучасні роботехнічні системи ґрунтуються переважно на модульних підходах, які забезпечують можливість інтеграції різнорідних програмних компонентів. Поширення відкритих платформ і симуляційних середовищ суттєво прискорило розроблення нових алгоритмів і дозволило підвищити доступність роботехнічних досліджень. Разом із тим існуючі програмні середовища переважно орієнтовані на дослідницькі або лабораторні задачі й не забезпечують повного циклу підготовки роботів до роботи у реальних аварійних умовах. Виявлено, що відсутність інтегрованих засобів формування навчальних сценаріїв і оцінювання готовності роботів є однією з ключових прогалин сучасних підходів.

Розгляд методів навігації, локалізації та взаємодії із середовищем засвідчив значний прогрес у розвитку алгоритмів SLAM, ймовірного моделювання та планування руху. Сучасні рішення забезпечують високу точність позиціонування та побудову карт у реальному часі, проте їх ефективність знижується в умовах динамічних змін середовища, характерних для техногенних катастроф. Невизначеність середовища, обмежена доступність сенсорних даних та порушення комунікаційних каналів вимагають використання більш адаптивних підходів, здатних працювати в умовах неповної інформації.

Аналіз застосування машинного навчання та навчання з підкріпленням у роботехніці показав, що ці методи створюють принципово нові можливості для формування адаптивної поведінки роботів. Використання симуляційного навчання, імітаційних підходів та перенесення моделей із віртуальних середовищ у реальні системи дозволяє значно скоротити час розроблення алгоритмів і підвищити рівень автономності роботів. Водночас залишаються суттєві проблеми, пов'язані з безпечністю навчання, інтерпретованістю моделей та складністю перенесення отриманих результатів у реальні умови

експлуатації. Це особливо актуально для сценаріїв надзвичайних ситуацій, де помилки системи можуть призвести до критичних наслідків.

Окрему увагу приділено питанням людино-роботної взаємодії та безпечного керування. Встановлено, що навіть за високого рівня автономності роль людини-оператора залишається ключовою, особливо в умовах підвищеної невизначеності та ризику. Ефективність роботехнічних систем значною мірою визначається якістю інтерфейсів, рівнем автоматизації рутинних дій та здатністю системи підтримувати оператора у процесі прийняття рішень. Таким чином, створення програмних комплексів навчання має враховувати не лише технічні аспекти керування роботами, а й особливості людського фактора.

Проведений огляд також показав, що сучасні глобальні виклики — зокрема воєнний стан та наслідки пандемії COVID-19 — суттєво розширили спектр застосування роботехнічних систем. Воєнні дії спричиняють виникнення техногенних аварій нового типу, що потребують швидкої адаптації роботів до нестабільних умов та роботи у зруйнованій інфраструктурі. Пандемія, у свою чергу, актуалізувала потребу в дистанційних та автономних рішеннях, що підтвердило необхідність гнучких програмних платформ навчання, здатних оперативно адаптуватися до нових сценаріїв використання.

Узагальнення результатів аналізу дозволило виявити низку системних проблем сучасного стану досліджень. По-перше, більшість існуючих рішень орієнтована на окремі аспекти функціонування роботів і не формує цілісного підходу до навчання систем у комплексних аварійних сценаріях. По-друге, існує розрив між алгоритмічними досягненнями машинного навчання та їх практичним впровадженням у реальні роботехнічні комплекси. По-третє, відсутність інтегрованих програмних середовищ, що поєднують симуляцію, навчання, моделювання сценаріїв і засоби взаємодії з оператором, обмежує можливості підготовки роботів до роботи у складних умовах.

Таким чином, результати огляду літератури підтверджують актуальність наукової проблеми, що полягає у необхідності розроблення програмного комплексу навчання роботехнічних систем ліквідації наслідків техногенного походження. Такий комплекс повинен забезпечувати інтеграцію симуляційного моделювання, методів машинного навчання, алгоритмів навігації та засобів людино-роботної взаємодії в єдиному середовищі, орієнтованому на підготовку роботів до реальних сценаріїв надзвичайних ситуацій.

Отримані висновки формують наукове підґрунтя для подальших розділів дисертаційної роботи, у яких буде обґрунтовано концепцію програмного комплексу навчання, визначено його архітектуру, методи реалізації та критерії оцінювання ефективності.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБЛЕННЯ АРХІТЕКТУРИ ПРОГРАМНОГО КОМПЛЕКСУ НАВЧАННЯ РОБОТЕХНІЧНИХ СИСТЕМ

2.1. Аналіз вимог до програмного комплексу навчання роботехнічних систем

Проектування програмного комплексу навчання роботехнічних систем, призначених для ліквідації наслідків надзвичайних ситуацій техногенного походження, потребує формалізації комплексу вимог, що впливають із особливостей середовища експлуатації, функціональних задач роботехнічних платформ та сучасних технологічних тенденцій у галузі роботехніки. На відміну від універсальних роботехнічних платформ загального призначення, системи аварійно-рятувального спрямування функціонують у середовищах із високим рівнем невизначеності, де критичними є питання безпеки, адаптивності та надійності [6; 46; 62].

Аналіз літератури показує, що основними викликами під час застосування роботехнічних систем у техногенних надзвичайних ситуаціях є нестабільність середовища, часткова або повна відсутність структурованої інформації, ризик руйнування інфраструктури та високий рівень завад для сенсорних систем [22; 23]. У зв'язку з цим програмний комплекс навчання має забезпечувати можливість підготовки роботів до роботи у широкому спектрі сценаріїв, включаючи непередбачувані зміни умов функціонування. Це обумовлює потребу у використанні гнучких архітектурних рішень і механізмів адаптивного навчання.

Першою групою є функціональні вимоги до програмного комплексу. До них належить підтримка формування навчальних сценаріїв, які відображають реальні умови техногенних катастроф, зокрема пожежі, руйнування будівель, витoki небезпечних речовин або обмеження видимості. Важливим аспектом є можливість моделювання поведінки роботів у середовищах із різною

структурою простору, що відповідає сучасним підходам до симуляційного навчання [34; 53]. Програмний комплекс має забезпечувати відтворення як детермінованих, так і стохастичних факторів, що впливають на поведінку роботехнічної системи.

Однією з ключових функціональних характеристик є підтримка інтеграції різних підходів до навчання роботів. У сучасній роботехніці застосовуються як класичні алгоритми керування, так і методи машинного навчання та навчання з підкріпленням [36; 50]. Тому програмний комплекс має бути орієнтований на гібридне використання алгоритмів, що дозволяє поєднувати передбачуваність традиційних методів із адаптивністю інтелектуальних моделей. Це особливо важливо у задачах ліквідації наслідків техногенних аварій, де зміна сценарію може вимагати негайної адаптації поведінки робота.

Наступною групою є нефункціональні вимоги, серед яких ключову роль відіграють надійність, масштабованість та модульність системи. Програмний комплекс повинен забезпечувати можливість додавання нових типів роботехнічних платформ без суттєвої зміни базової архітектури [47]. Модульний підхід дозволяє повторно використовувати компоненти системи, що значно знижує витрати на адаптацію програмного забезпечення до нових задач. Крім того, модульність сприяє інтеграції різнорідних сенсорів та алгоритмів, що є характерною потребою сучасних роботехнічних систем.

Важливою вимогою є підтримка симуляційного середовища як основного інструмента навчання. Реальні експерименти у небезпечних умовах часто є неможливими або економічно недоцільними, тому значна частина навчального процесу переноситься у віртуальний простір [40; 53]. Програмний комплекс повинен забезпечувати достовірне фізичне моделювання поведінки роботів, а також можливість генерації великої кількості навчальних сценаріїв для формування стійких моделей поведінки.

Аналіз сучасних досліджень показує, що ключовою проблемою симуляційного навчання є розрив між віртуальним та реальним середовищем

(sim-to-real gap) [52]. У зв'язку з цим до вимог програмного комплексу слід віднести підтримку механізмів адаптації навчальних моделей до реальних умов експлуатації. Це передбачає використання випадкових варіацій параметрів середовища, різних моделей шуму сенсорів та інших методів підвищення робастності алгоритмів.

Особливої уваги потребують вимоги, зумовлені сучасними соціальними та геополітичними викликами. Воєнний стан створює нові типи техногенних загроз, пов'язаних із руйнуванням критичної інфраструктури та виникненням складних аварійних сценаріїв [22; 23]. У таких умовах роботехнічні системи мають бути здатними швидко адаптуватися до нових задач без повного перепроєктування програмного забезпечення. Отже, програмний комплекс навчання повинен підтримувати гнучке налаштування сценаріїв та швидке перенавчання моделей.

Досвід пандемії COVID-19 також сформував додаткові вимоги до роботехнічних систем. Використання роботів у медичній та логістичній сферах показало необхідність дистанційного керування, віддаленого моніторингу та швидкого оновлення функціональності систем [36; 48]. Це підкреслює важливість мережевої взаємодії між компонентами програмного комплексу та підтримки розподілених обчислень, що дозволяє організовувати навчання роботів із використанням зовнішніх обчислювальних ресурсів.

Ще однією критичною вимогою є забезпечення ефективної взаємодії людини-оператора з навчальним процесом. Як показують дослідження у сфері HRI, оператор залишається ключовим елементом системи в умовах підвищеної невизначеності [56; 60]. Тому програмний комплекс має включати засоби візуалізації навчальних процесів, моніторингу стану роботів та інструменти коригування поведінки під час навчання. Це дозволяє поєднати автоматичне формування стратегій із експертними знаннями людини.

Важливою вимогою є також підтримка багатороботних сценаріїв навчання. У сучасних аварійно-рятувальних операціях дедалі частіше застосовуються групи роботів, що виконують різні функції — розвідку,

моніторинг або транспортування [37; 55]. Програмний комплекс повинен забезпечувати координацію дій декількох агентів, синхронізацію даних і можливість колективного навчання, що підвищує ефективність виконання комплексних задач.

На основі аналізу літератури можна сформулювати вимоги до інформаційної складової системи. Програмний комплекс має забезпечувати накопичення, зберігання та повторне використання навчальних даних, що дозволяє формувати базу сценаріїв і прискорювати процес підготовки робіт. Наявність централізованого сховища даних також створює передумови для подальшого використання технології цифрового двійника, яка забезпечує безперервний зв'язок між реальним роботом і його віртуальною моделлю. З позицій побудови програмного комплексу інформаційні технології забезпечують організацію життєвого циклу даних — від їх збирання та зберігання до повторного використання в навчальних сценаріях.

Узагальнюючи результати аналізу, можна виділити основні групи вимог до програмного комплексу навчання роботехнічних систем: функціональні (формування сценаріїв, підтримка навчальних алгоритмів, моделювання середовища), нефункціональні (надійність, модульність, масштабованість), експлуатаційні (адаптивність до НС, підтримка дистанційної роботи), а також вимоги безпеки та взаємодії з оператором. Сукупність цих вимог обумовлює необхідність використання гібридної архітектури, здатної інтегрувати різноманітні компоненти в єдину систему.

Таким чином, проведений аналіз підтверджує, що існуючі програмні рішення не повністю відповідають сукупності вимог, які виникають у задачах навчання роботехнічних систем для роботи в умовах надзвичайних ситуацій техногенного походження. Це створює передумови для формування нової концепції програмного комплексу, що поєднуватиме модульну гібридну архітектуру з технологією цифрового двійника. Результати цього аналізу слугують основою для побудови концептуальної моделі системи, яка розглядається у наступному підрозділі.

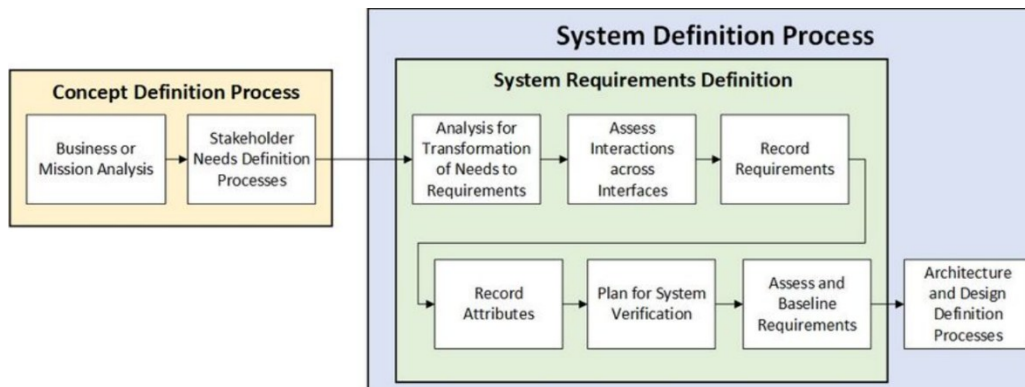


Рисунок 2.1 — Модель вимог до програмного комплексу навчання роботехнічних систем

2.2. Концепція гібридної архітектури програмного комплексу навчання роботехнічних систем

Сучасні роботехнічні системи, що призначені для роботи в умовах надзвичайних ситуацій техногенного походження, характеризуються складною структурою керування та високими вимогами до адаптивності. Аналіз сучасних досліджень свідчить, що використання лише класичних алгоритмічних підходів або виключно методів машинного навчання не забезпечує необхідного рівня надійності в умовах невизначеності [46; 49]. Тому для програмного комплексу навчання роботехнічних систем доцільним є застосування гібридної архітектури, що поєднує детерміновані алгоритми, адаптивні моделі навчання та симуляційне середовище.

Концептуальні основи гібридної архітектури. Гібридна архітектура базується на поєднанні двох підходів:

- класичного керування (rule-based control);
- інтелектуальних алгоритмів навчання (learning-based control).

Класичні алгоритми забезпечують стабільність та безпеку руху, тоді як ML/RL моделі відповідають за адаптацію до нових сценаріїв та оптимізацію поведінки [50; 41].

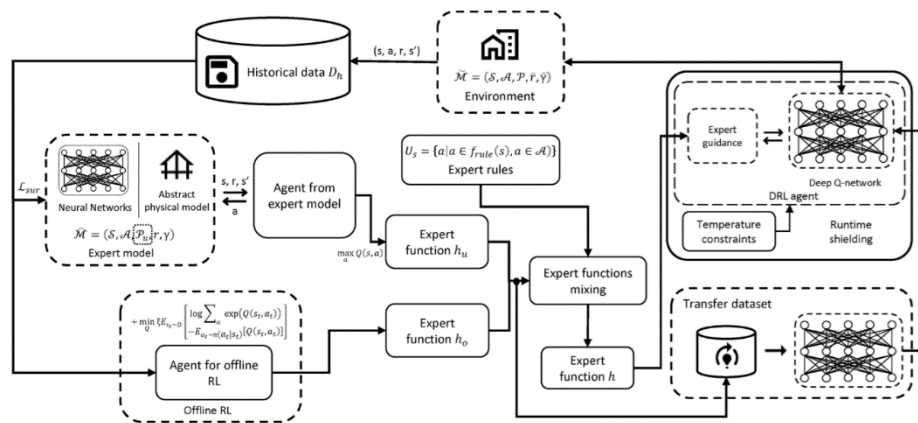


Рисунок 2.2 — Логічна концепція гібридної архітектури програмного комплексу

У межах такої концепції робот розглядається як система, поведінка якої формується двома паралельними контурами: алгоритмічним та навчальним. Це дозволяє поєднувати передбачуваність із здатністю до адаптації.

Рівнева структура гібридної системи. Для організації взаємодії модулів доцільно використовувати багаторівневу модель.

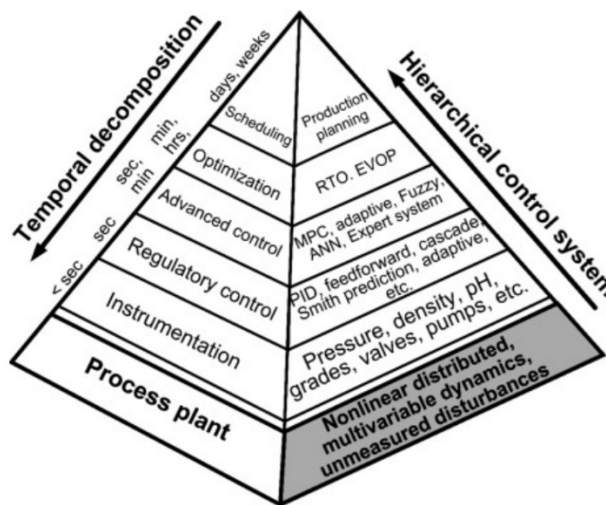


Рисунок 2.3 — Рівні керування гібридної архітектури

Такий підхід відповідає сучасним принципам архітектури роботехнічних систем [32; 67] та дозволяє інтегрувати алгоритми машинного навчання без порушення стабільності керування.

Digital Twin як центральний елемент концепції. Ключовою особливістю запропонованої архітектури є використання цифрового двійника (Digital Twin), який виступає динамічною віртуальною копією реальної роботехнічної системи.

Digital Twin забезпечує:

- моделювання поведінки робота;
- генерацію навчальних сценаріїв;
- тестування алгоритмів до їх запуску;
- накопичення досвіду експлуатації.

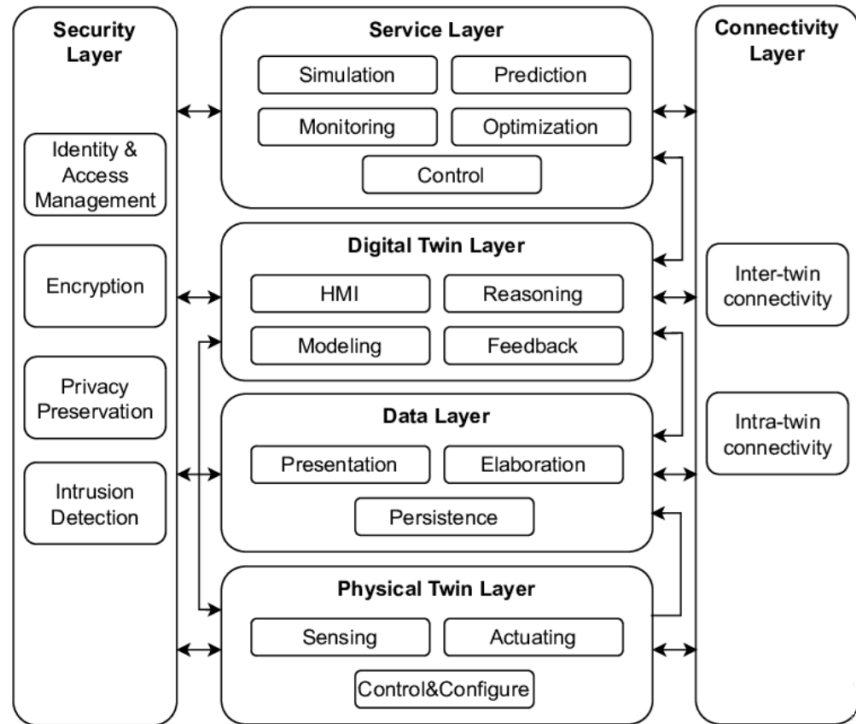


Рисунок 2.4 — Контур Digital Twin у програмному комплексі

Такий контур дозволяє реалізувати безперервне навчання системи та зменшити розрив між симуляцією і реальним середовищем [52; 53].

Потоки даних у гібридній архітектурі. Однією з ключових переваг запропонованої концепції є чітка організація потоків даних.

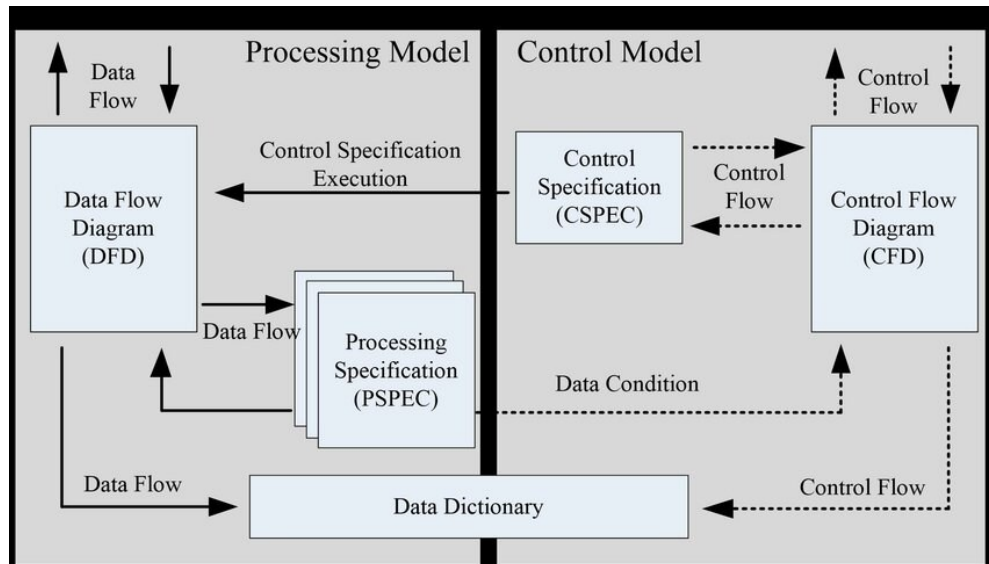


Рисунок 2.5 — Потоки даних між модулями програмного комплексу

Такий data-centric підхід дозволяє забезпечити прозорість процесів навчання і подальшу масштабованість системи. Такий підхід відповідає сучасному розумінню поняття «інформаційні технології» як основи для побудови масштабованих, даноцентричних і повторно використовуваних програмних рішень.

Роль оператора в гібридній архітектурі. Умови надзвичайних ситуацій характеризуються високим рівнем ризику, тому повна автономність системи наразі є недоцільною [56; 60]. У запропонованій концепції оператор виконує роль стратегічного керівника, який може:

- задавати сценарії навчання;
- контролювати процес адаптації;
- втручатися у критичних ситуаціях.

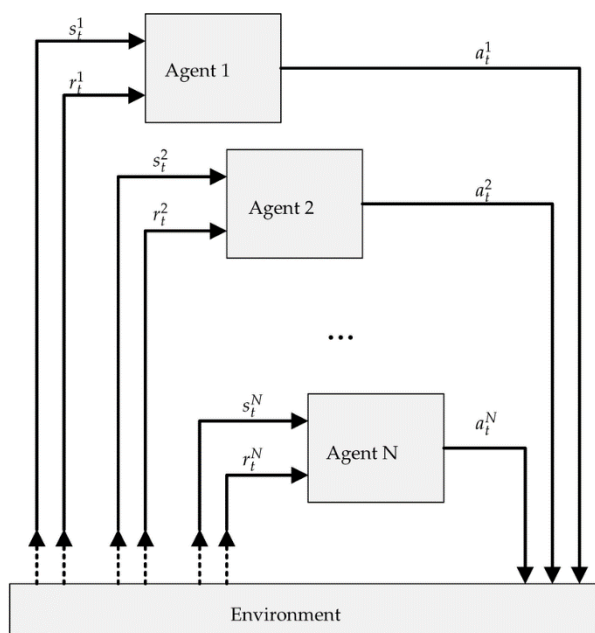


Рисунок 2.6 — Модель взаємодії оператора та гібридної системи

Переваги запропонованої концепції. Порівняно з традиційними підходами гібридна архітектура забезпечує:

- адаптацію до невідомих умов;
- безпечну інтеграцію ML алгоритмів;
- безперервне навчання через Digital Twin;
- масштабованість системи для різних типів роботів.

Це особливо важливо для сценаріїв, пов'язаних із техногенними аваріями, воєнними діями та кризами типу COVID-2019, де середовище може швидко змінюватися.

Таким чином, концепція гібридної архітектури формує основу програмного комплексу навчання роботехнічних систем, у якому детерміноване керування поєднується з адаптивними алгоритмами та цифровим двійником. Це забезпечує необхідний баланс між безпечністю, адаптивністю та ефективністю, що є критичним для застосування роботів у складних надзвичайних сценаріях. Запропонована концепція виступає базою для подальшого розгляду цифрового двійника як центрального елемента архітектури, що буде детально проаналізовано в наступному підрозділі.

2.3. Цифровий двійник роботехнічної системи як основа навчання

Однією з ключових тенденцій сучасного розвитку інтелектуальних технічних систем є використання концепції цифрового двійника (Digital Twin), яка забезпечує створення віртуальної динамічної моделі фізичного об'єкта з можливістю двостороннього обміну даними між реальним та віртуальним середовищем. Для роботехнічних систем, що функціонують у складних умовах надзвичайних ситуацій техногенного походження, технологія цифрового двійника відкриває нові можливості щодо навчання, тестування та адаптації поведінки роботів без ризику для обладнання та персоналу [46; 53]. У межах запропонованої дисертаційної роботи цифровий двійник розглядається як базовий елемент гібридної архітектури програмного комплексу навчання.

Концептуальні засади цифрового двійника в роботехніці. Цифровий двійник є не просто симуляційною моделлю, а системою, яка динамічно відображає стан фізичного об'єкта на основі телеметричних даних та забезпечує можливість прогнозування його поведінки. У класичному розумінні цифровий двійник включає три ключові компоненти: фізичну систему, віртуальну модель та канал двостороннього обміну даними між ними. У роботехніці це означає, що реальний робот і його цифровий аналог постійно синхронізуються, що дозволяє відтворювати реальні сценарії в симуляційному середовищі [34; 53].

У контексті навчання роботехнічних систем цифровий двійник виконує роль інструмента безпечного експериментування. Замість проведення ризикованих випробувань у реальному середовищі алгоритми можуть навчатися та перевірятися у віртуальній копії, що значно скорочує витрати часу та ресурсів [52].

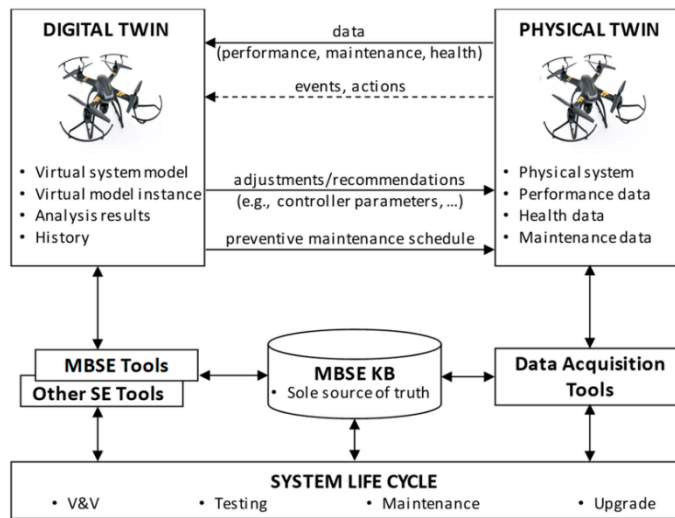


Рисунок 2.7 — Загальна концепція цифрового двійника роботехнічної системи

Роль цифрового двійника у процесі навчання роботів. Основною функцією цифрового двійника в межах програмного комплексу є організація замкненого циклу навчання (learning loop), у якому результати реальної експлуатації використовуються для покращення моделей поведінки.

Процес функціонування такого циклу включає:

1. отримання даних із реального робота;
2. оновлення цифрової моделі;
3. моделювання нових сценаріїв;
4. навчання або перенавчання алгоритмів;
5. передавання оновлених моделей у реальну систему.

Застосування такого підходу особливо актуальне для роботів, що працюють у надзвичайних ситуаціях, де кожна місія може генерувати унікальні дані, які мають використовуватись для подальшого покращення системи [46; 62].

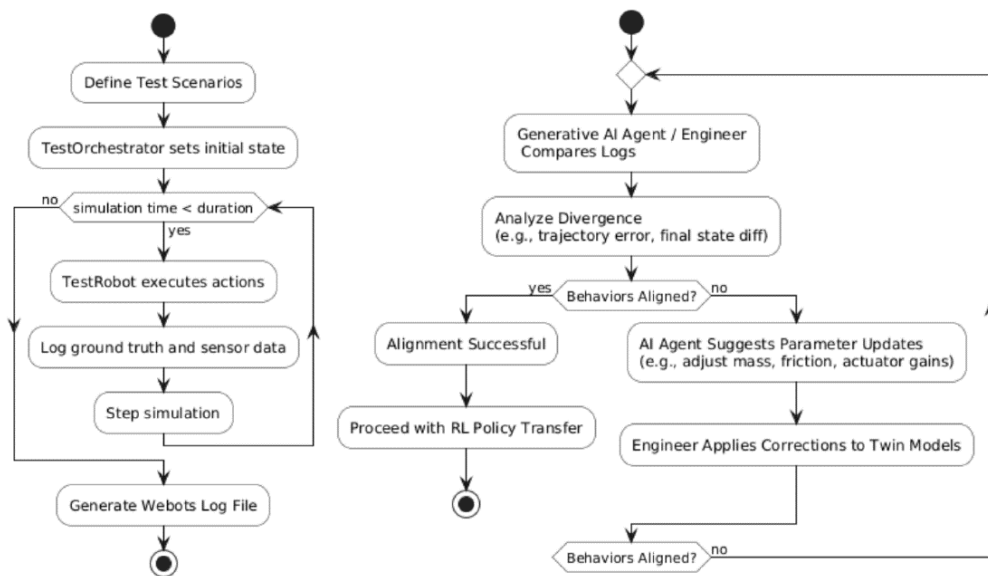


Рисунок 2.8 — Замкнений цикл навчання на основі Digital Twin

Моделювання сценаріїв надзвичайних ситуацій. Однією з найбільших переваг цифрового двійника є можливість формування великої кількості навчальних сценаріїв, які відтворюють реальні умови техногенних аварій. У межах програмного комплексу можуть моделюватися:

- руйнування інфраструктури;
- обмеження видимості;
- складні рельєфи;
- наявність рухомих об’єктів;
- порушення комунікаційних каналів.

Завдяки цьому робот здатний накопичувати досвід у середовищах, які фізично неможливо або небезпечно відтворити в реальності [40; 53].

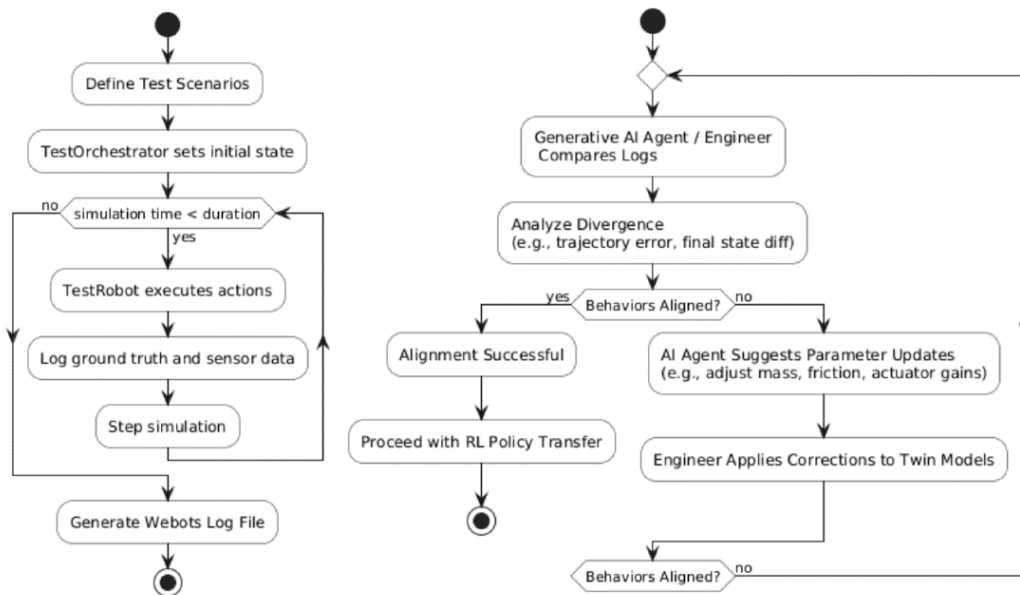


Рисунок 2.9 — Модель генерації навчальних сценаріїв у Digital Twin

Використання стохастичних сценаріїв дозволяє підвищити робастність навчальних моделей та зменшити залежність алгоритмів від конкретних умов.

Digital Twin і проблема sim-to-real transfer. Однією з ключових проблем сучасного роботехнічного навчання є розрив між симуляційним та реальним середовищем [52]. Традиційні симуляції часто не враховують усіх фізичних особливостей, що призводить до зниження ефективності моделей після перенесення в реальний світ. Цифровий двійник частково вирішує цю проблему за рахунок постійного оновлення параметрів віртуальної моделі на основі реальних даних.

У запропонованій концепції цифровий двійник виконує функції:

- калібрування фізичних параметрів;
- адаптації сенсорних моделей;
- корекції поведінкових політик.

Завдяки цьому навчання стає безперервним процесом, у якому віртуальна модель поступово наближається до реальних умов експлуатації.

Інтеграція цифрового двійника з гібридною архітектурою. У рамках розроблюваного програмного комплексу цифровий двійник інтегрується з двома основними компонентами гібридної архітектури:

- класичним модулем керування;
- інтелектуальним модулем навчання.

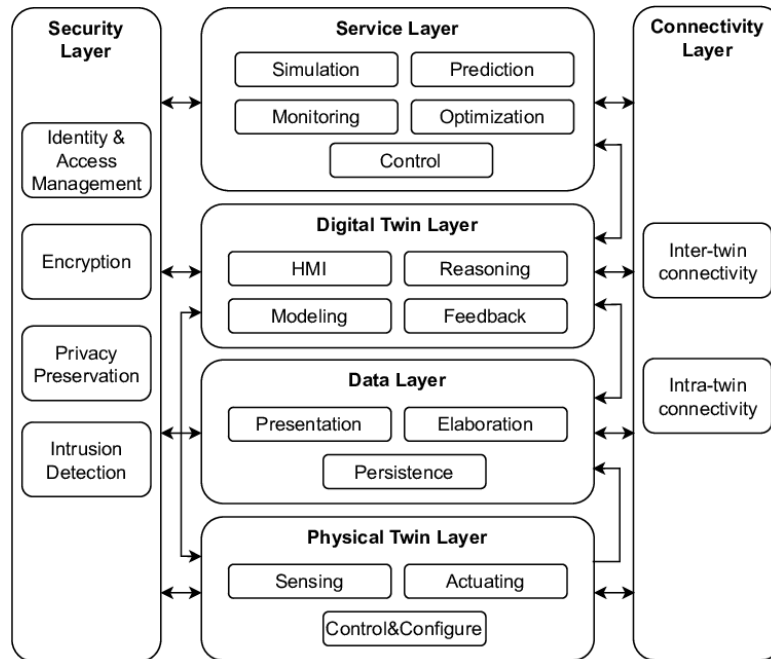


Рисунок 2.10 — Місце Digital Twin у гібридній архітектурі

Така інтеграція дозволяє забезпечити узгодженість між алгоритмами керування та навчальними моделями, знижуючи ризик неконтрольованої поведінки роботів.

Роль цифрового двійника в умовах воєнного стану та COVID-2019.

Сучасні виклики, пов'язані з воєнними діями та глобальними кризами, підкреслюють необхідність швидкої адаптації роботехнічних систем до нових сценаріїв. Умови бойових дій часто характеризуються високою динамічністю та неможливістю проведення повномасштабних натурних випробувань [22; 23]. Використання цифрового двійника дозволяє швидко моделювати нові сценарії та перенавчати роботів без ризику для техніки.

Пандемія COVID-2019 також продемонструвала необхідність дистанційного тестування та підготовки автономних систем [36; 48]. Цифрові двійники дали змогу розробникам проводити навчання роботів у віртуальному середовищі та оперативно оновлювати алгоритми без фізичного доступу до обладнання.

Переваги використання цифрового двійника. Узагальнюючи результати аналізу, можна виділити ключові переваги використання цифрового двійника в програмному комплексі навчання:

- безпечне моделювання небезпечних сценаріїв;
- зменшення витрат на експерименти;
- безперервне навчання системи;
- зниження sim-to-real gap;
- можливість накопичення досвіду експлуатації.

Ці переваги роблять Digital Twin ключовим компонентом сучасних роботехнічних систем, орієнтованих на роботу у складних середовищах.

Таким чином, цифровий двійник виступає центральною складовою концепції гібридної архітектури програмного комплексу навчання роботехнічних систем. Його використання забезпечує зв'язок між симуляційним середовищем і реальними роботами, дозволяє організувати безперервний цикл навчання та підвищує адаптивність систем у умовах надзвичайних ситуацій техногенного походження. Запропонований підхід створює основу для побудови цілісної архітектури програмного комплексу, яка буде детально розглянута у наступному підрозділі.

2.4. Архітектура програмного комплексу та взаємодія модулів

На основі сформованих у попередніх підрозділах вимог, концепції гібридної архітектури та використання цифрового двійника формується узагальнена архітектура програмного комплексу навчання роботехнічних систем. Основною метою проектування архітектури є забезпечення цілісного середовища, у якому поєднуються симуляційне моделювання, навчання алгоритмів, керування роботами та взаємодія з оператором. У контексті задач ліквідації наслідків техногенних надзвичайних ситуацій архітектура повинна забезпечувати високу адаптивність, масштабованість та можливість інтеграції різномірних роботехнічних платформ [47; 53].

Запропонована архітектура ґрунтується на гібридному принципі організації, де функціональні модулі мають чітко визначені ролі, але взаємодіють через єдиний інформаційний простір. Такий підхід забезпечує незалежність окремих компонентів та можливість поступового розширення системи без зміни її базової структури.

Загальна структура програмного комплексу. Архітектура програмного комплексу включає п'ять ключових підсистем:

1. модуль симуляційного середовища;
2. модуль цифрового двійника;
3. модуль навчання та адаптації;
4. модуль керування роботехнічними системами;
5. модуль взаємодії з оператором та моніторингу.

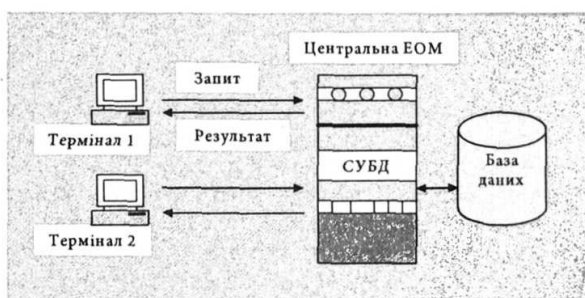


Рисунок 2.11 — Загальна архітектура програмного комплексу навчання роботехнічних систем

Центральне місце архітектури займає цифровий двійник, який забезпечує синхронізацію між віртуальною та фізичною системами. Така структура дозволяє реалізувати замкнений цикл навчання та безперервного вдосконалення поведінки роботехнічних систем.

Модуль симуляційного середовища. Симуляційне середовище виконує функцію генерації навчальних сценаріїв та тестування алгоритмів без ризику для реального обладнання. На основі сучасних платформ моделювання забезпечується фізично коректна імітація руху роботів, взаємодії з об'єктами та сенсорного сприйняття [40; 53].

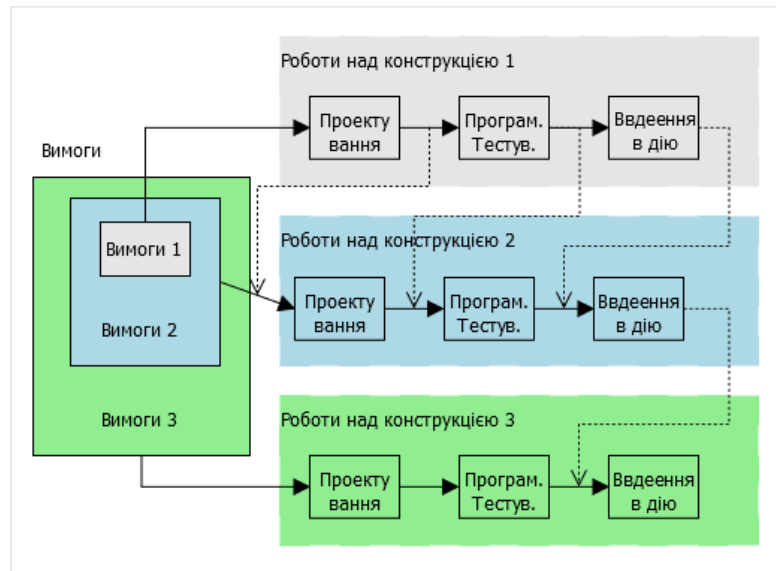


Рисунок 2.12 — Архітектура симуляційного модуля

Особливістю модуля є можливість параметризації сценаріїв, що дозволяє моделювати різні типи надзвичайних ситуацій, включаючи руйнування інфраструктури або обмеження видимості.

Модуль цифрового двійника. Цифровий двійник виконує функцію інтеграційного ядра архітектури. Він накопичує телеметричні дані, оновлює параметри моделі робота та забезпечує актуальність симуляції відносно реального середовища [34; 52].

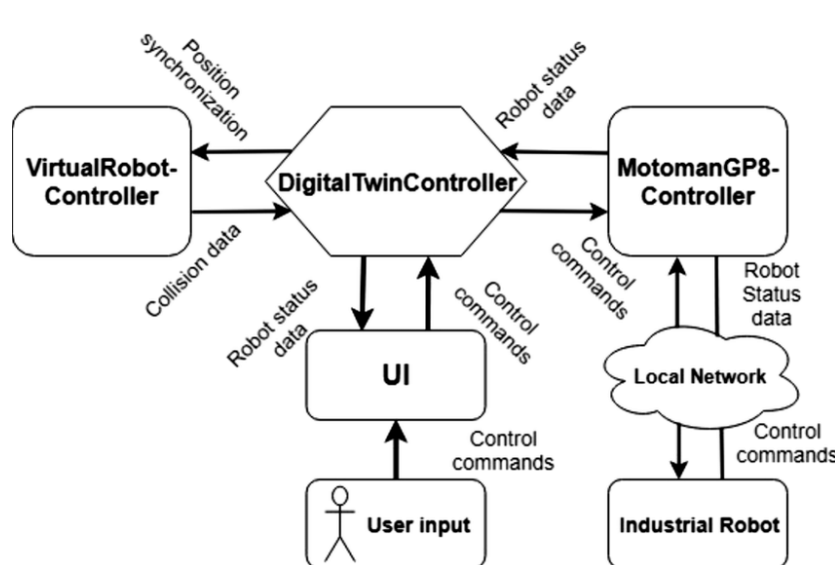


Рисунок 2.13 — Внутрішня структура Digital Twin

У межах програмного комплексу цифровий двійник дозволяє відтворювати реальні аварійні сценарії та передбачати поведінку роботів у нових умовах.

Модуль навчання та адаптації. Модуль навчання реалізує алгоритми машинного навчання, включаючи навчання з підкріпленням, imitation learning та адаптивні стратегії керування [25; 50]. Його основною функцією є формування моделей поведінки роботів на основі симуляційних і реальних даних.

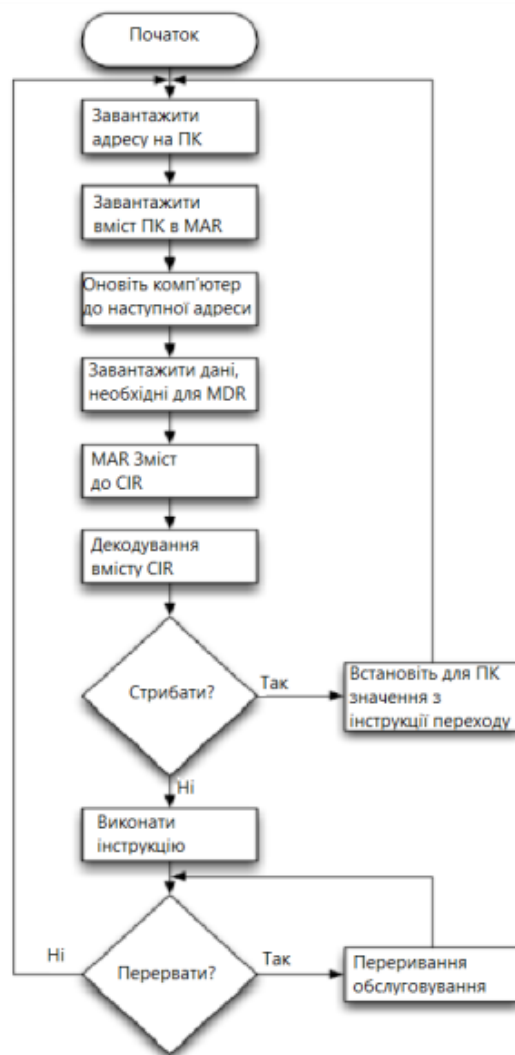


Рисунок 2.14 — Pipeline навчання роботехнічної системи

Модуль навчання інтегрується з цифровим двійником, що дозволяє використовувати накопичений досвід експлуатації для постійного оновлення

моделей. Особливу роль відіграють механізми контролю безпеки, які забезпечують перевірку моделей перед їх застосуванням у фізичній системі.

Модуль керування роботехнічними системами. Цей модуль відповідає за виконання команд у реальному середовищі та реалізує гібридний підхід до керування. Базові функції руху та стабілізації забезпечуються класичними алгоритмами навігації та локалізації [26; 35; 51], тоді як адаптивна поведінка формується за допомогою моделей машинного навчання.

Схема має включати два контури:

- детермінований контролер;
- адаптивний ML-модуль, що коригує дії.

Такий підхід дозволяє гарантувати базову безпеку руху навіть у випадках нестабільної роботи інтелектуальних алгоритмів.

Модуль взаємодії з оператором. Операторська підсистема забезпечує моніторинг стану роботів, контроль процесів навчання та можливість втручання у критичних ситуаціях [56; 60]. Інтерфейс повинен забезпечувати узагальнене представлення даних, знижуючи когнітивне навантаження на оператора.

Схема має відображати потоки інформації від роботів і Digital Twin до оператора та зворотний канал управління.

В умовах воєнного стану та під час дистанційної роботи, характерної для сценаріїв COVID-2019, модуль підтримує віддалений доступ і розподілену взаємодію [22; 48].

Організація потоків даних у системі. Однією з ключових переваг запропонованої архітектури є уніфікована система обміну даними між модулями. Усі компоненти функціонують у межах єдиного інформаційного простору, що забезпечує цілісність навчального процесу.

Рекомендовано показати потоки сенсорних даних, навчальних параметрів, команд керування та даних моніторингу.

Централізований підхід до управління даними дозволяє реалізувати масштабованість системи та підтримку багатороботних сценаріїв [37; 55].

Адаптивність архітектури до різних сценаріїв застосування.

Запропонована архітектура дозволяє використовувати програмний комплекс у різних прикладних задачах, зокрема:

- роботизовані системи евакуації у бойових умовах;
- автономні логістичні платформи;
- моніторингові роботи;
- системи дистанційного обстеження небезпечних зон.

Гнучкість досягається завдяки модульному принципу та використанню цифрового двійника, який забезпечує швидке налаштування системи під нові сценарії.

Таким чином, запропонована архітектура програмного комплексу навчання роботехнічних систем базується на гібридному підході та інтеграції технології цифрового двійника. Вона забезпечує поєднання симуляційного навчання, адаптивного керування та людино-роботної взаємодії в межах єдиного середовища. Архітектура є масштабованою, адаптивною та придатною для застосування у складних умовах надзвичайних ситуацій техногенного походження. Результати проектування створюють основу для обґрунтування прийнятих архітектурних рішень і порівняння запропонованого підходу з існуючими системами, що розглядається у наступному підрозділі.

2.5. Обґрунтування архітектурних рішень та переваги запропонованого підходу

Побудова програмного комплексу навчання роботехнічних систем для ліквідації наслідків надзвичайних ситуацій техногенного походження потребує не лише формального опису архітектури, але й наукового обґрунтування обраних структурних рішень. У попередніх підрозділах було сформовано вимоги до системи, визначено концепцію гібридної архітектури та обґрунтовано використання цифрового двійника як центрального елемента

навчального процесу. У цьому підрозділі здійснюється узагальнений аналіз прийнятих рішень, їх порівняння з існуючими підходами та визначення переваг запропонованої архітектури.

Обґрунтування вибору гібридної архітектури. Аналіз сучасних роботехнічних систем свідчить, що використання виключно класичних алгоритмів керування не забезпечує достатнього рівня адаптивності в умовах невизначеності [32; 67]. Водночас повністю навчальні системи, побудовані лише на основі машинного навчання, часто демонструють нестабільну поведінку в критичних сценаріях та потребують значних обчислювальних ресурсів [49; 50]. У зв'язку з цим обраний гібридний підхід дозволяє поєднати переваги обох напрямів.

Гібридна архітектура забезпечує розподіл функцій між детермінованими та адаптивними компонентами. Детерміновані модулі відповідають за безпеку руху, стабільність керування та базові реакції на зовнішні впливи, тоді як інтелектуальні алгоритми формують поведінкові стратегії в умовах, які не були передбачені заздалегідь [41; 52]. Такий баланс є особливо важливим для систем, що працюють у зонах техногенних аварій або в умовах воєнного стану, де сценарії можуть змінюватися у реальному часі [22; 23].

Обґрунтування інтеграції Digital Twin. Використання цифрового двійника як центрального елемента архітектури обумовлено необхідністю мінімізації ризиків під час навчання роботехнічних систем. На відміну від традиційних симуляторів, цифровий двійник забезпечує постійну синхронізацію між реальною системою та її віртуальною моделлю [34; 53]. Це дозволяє використовувати реальні дані для корекції моделей і поступово зменшувати розрив між симуляційним та фізичним середовищем.

Особливої актуальності такий підхід набуває у задачах, де натурні експерименти є небезпечними або економічно недоцільними. У сценаріях ліквідації наслідків техногенних аварій або під час бойових дій можливість відпрацювання поведінки роботів у цифровому середовищі забезпечує значне підвищення рівня безпеки [46; 62].

На схемі варто відобразити відсутність зворотного зв'язку у звичайній симуляції та наявність двонапрямної синхронізації у Digital Twin.

Обґрунтування модульної структури системи. Модульність архітектури обрана з метою забезпечення масштабованості та можливості адаптації програмного комплексу до різних типів робіт. Сучасні дослідження підкреслюють, що відкриті модульні платформи значно спрощують інтеграцію нових алгоритмів та апаратних компонентів [47; 48]. У запропонованому підході кожен модуль виконує незалежну функцію, а взаємодія відбувається через уніфіковані інтерфейси обміну даними.

Це дозволяє:

- оновлювати окремі компоненти без перебудови системи;
- адаптувати архітектуру під нові сценарії;
- підтримувати багатороботні системи.

Модульність також сприяє повторному використанню програмних компонентів, що має суттєве значення для практичного впровадження.

Порівняння із існуючими програмними платформами. У сучасній роботехніці широко використовуються системи типу ROS, Gazebo та інші симуляційні середовища [40; 47]. Проте аналіз показує, що вони переважно орієнтовані на розроблення окремих модулів або дослідницькі експерименти. На відміну від них, запропонований програмний комплекс орієнтований на повний цикл навчання роботехнічних систем.

Рекомендується показати критерії: симуляція, навчання, адаптація, Digital Twin, підтримка операторів, інтеграція сценаріїв НС.

Порівняння дозволяє зробити висновок, що основною відмінністю запропонованого підходу є інтеграція навчальних процесів із експлуатаційними даними в межах єдиного середовища.

Переваги запропонованої архітектури для реальних сценаріїв застосування. Запропонована архітектура демонструє високу універсальність і може бути використана для різних типів завдань:

- роботизовані евакуатори для роботи в зоні бойових дій;
- автономні системи доставки;
- роботи для моніторингу небезпечних середовищ;
- системи акустичного або візуального моніторингу.

Завдяки Digital Twin система може швидко адаптуватися до нових сценаріїв, що є особливо важливим в умовах динамічних змін середовища [22; 23].

Досвід пандемії COVID-2019 підтвердив ефективність підходів, що передбачають дистанційне навчання та переналаштування роботів без фізичного доступу до обладнання [36; 48]. Запропонований комплекс забезпечує такі можливості завдяки гібридній архітектурі та централізованому управлінню даними.

Наукова новизна запропонованого рішення полягає у:

- інтеграції Digital Twin у структуру програмного комплексу навчання;
- поєднанні класичних алгоритмів керування з ML-моделями в єдиному архітектурному середовищі;
- реалізації замкненого циклу навчання «симуляція – реальна система – адаптація»;
- підтримці багатосценарійного навчання в умовах надзвичайних ситуацій.

Отже, проведене обґрунтування підтверджує доцільність використання гібридної модульної архітектури із цифровим двійником як центральним елементом програмного комплексу навчання роботехнічних систем. Запропоновані рішення забезпечують підвищення адаптивності, безпеки та ефективності навчання роботів у складних умовах експлуатації. Порівняння з існуючими платформами показало, що розроблений підхід дозволяє інтегрувати симуляційне навчання, реальні дані та операторське керування в єдину систему, що створює основу для подальшої реалізації алгоритмів

навчання та експериментальної перевірки, які будуть розглянуті у наступних розділах дисертаційної роботи.

2.6. Висновки до розділу 2

У другому розділі дисертаційної роботи було здійснено проектування та обґрунтування архітектури програмного комплексу навчання роботехнічних систем, призначених для роботи в умовах надзвичайних ситуацій техногенного походження. На відміну від традиційних підходів, які розглядають навчання роботів як окремий етап розроблення, запропонований підхід орієнтований на формування цілісного програмного середовища, що інтегрує симуляційне моделювання, алгоритми машинного навчання, класичні засоби керування та механізми взаємодії з оператором.

На основі аналізу сучасних наукових джерел і вимог до роботехнічних систем сформовано комплекс функціональних і нефункціональних вимог до програмного комплексу. Встановлено, що специфіка надзвичайних ситуацій техногенного походження, а також сучасні виклики, пов'язані з воєнним станом і кризовими явищами типу COVID-2019, висувають підвищені вимоги до адаптивності, масштабованості та безпечності роботехнічних систем. Це обумовлює необхідність створення гнучкої програмної архітектури, здатної працювати з різними типами сценаріїв без повного перепроєктування системи.

У результаті дослідження обґрунтовано доцільність використання гібридної архітектури програмного комплексу, яка поєднує детерміновані алгоритми керування з інтелектуальними методами машинного навчання. Такий підхід забезпечує баланс між стабільністю функціонування системи та її здатністю до адаптації в умовах невизначеного середовища. Показано, що гібридна структура дозволяє реалізувати багаторівневе керування, у межах якого базова безпечність забезпечується класичними методами, тоді як адаптивна поведінка формується на основі навчальних алгоритмів.

Ключовим результатом розділу стало введення концепції цифрового двійника роботехнічної системи як центрального елемента програмного

комплексу навчання. Встановлено, що використання Digital Twin забезпечує безперервний цикл навчання «реальна система — віртуальна модель — адаптація», що дозволяє суттєво зменшити розрив між симуляційним і реальним середовищем. Це створює передумови для безпечного тестування алгоритмів, накопичення досвіду експлуатації та швидкої адаптації роботів до нових сценаріїв.

У межах розділу розроблено узагальнену архітектуру програмного комплексу, яка включає модуль симуляційного середовища, модуль цифрового двійника, модуль навчання та адаптації, модуль керування роботехнічними системами та модуль взаємодії з оператором. Показано, що взаємодія модулів через єдиний інформаційний простір забезпечує цілісність навчального процесу, підтримує масштабованість системи та дозволяє інтегрувати різні типи роботехнічних платформ.

Порівняльний аналіз запропонованої архітектури з існуючими платформами показав, що її основною перевагою є інтеграція навчальних процесів із реальними даними експлуатації в межах єдиного середовища. На відміну від традиційних підходів, де симуляція, навчання та керування існують окремо, запропонована модель забезпечує замкнений цикл адаптації системи, що особливо важливо для роботи в умовах динамічних надзвичайних ситуацій.

Узагальнюючи результати проведеного дослідження, можна зробити такі основні висновки:

1. Сформовано систему вимог до програмного комплексу навчання роботехнічних систем, що враховує специфіку техногенних надзвичайних ситуацій, воєнного стану та сценаріїв кризового реагування.

2. Обґрунтовано використання гібридної архітектури як найбільш ефективного підходу для поєднання безпечності та адаптивності роботехнічних систем.

3. Запропоновано використання цифрового двійника як інтеграційного ядра програмного комплексу, що забезпечує безперервне навчання та адаптацію алгоритмів.

4. Розроблено архітектурну модель програмного комплексу, яка забезпечує взаємодію симуляційного навчання, реального керування та людино-роботної взаємодії.

5. Показано переваги запропонованого підходу порівняно з існуючими програмними платформами з точки зору універсальності, масштабованості та придатності до роботи у складних сценаріях.

Таким чином, результати другого розділу формують теоретичне та методологічне підґрунтя для подальших досліджень, присвячених розробленню методів та алгоритмів навчання роботехнічних систем у межах запропонованого програмного комплексу. Це визначає логіку наступного розділу дисертаційної роботи, у якому буде розглянуто математичні моделі та алгоритмічне забезпечення процесу навчання.

РОЗДІЛ 3. МОДЕЛІ ТА МЕТОДИ АДАПТИВНОГО НАВЧАННЯ РОБОТЕХНІЧНИХ СИСТЕМ НА ОСНОВІ DIGITAL TWIN

3.1. Формалізація процесу адаптивного навчання роботехнічних систем

Адаптивне навчання роботехнічних систем у межах програмного комплексу на основі Digital Twin розглядається як безперервний процес удосконалення поведінкових стратегій робота на основі взаємодії з реальним та віртуальним середовищем. На відміну від класичних підходів, у яких навчання завершується до початку експлуатації системи, запропонована концепція передбачає постійну адаптацію моделей у процесі роботи. Це особливо важливо для роботів, що використовуються в умовах надзвичайних ситуацій техногенного походження, де структура середовища та задачі можуть змінюватися непередбачувано [46; 62].

Загальна модель адаптивного навчання. Процес адаптації базується на циклічному принципі: робот отримує дані із середовища, виконує дії, аналізує результати та коригує поведінку. У запропонованому підході цей цикл розширюється за рахунок цифрового двійника, який дозволяє моделювати нові сценарії та тестувати оновлені стратегії без ризику для фізичної системи.

На схемі слід показати послідовність: фізичний робот → збір даних → Digital Twin → навчальний модуль → оновлена стратегія → робот.

Такий підхід дозволяє забезпечити безперервне вдосконалення системи й накопичення досвіду під час експлуатації.

Складові процесу навчання. Процес адаптивного навчання включає чотири ключові складові:

- сприйняття середовища — збір сенсорних даних;
- аналіз і моделювання — побудова внутрішньої моделі середовища;
- прийняття рішень — формування керуючих дій;

– оцінювання результатів — перевірка ефективності дій.

У традиційних роботехнічних системах ці етапи реалізуються за фіксованими алгоритмами. У запропонованому підході моделі прийняття рішень можуть адаптуватися відповідно до нової інформації.

Рекомендовано зобразити блоки: сенсори → аналіз → policy module → контроль → середовище.

Простір станів та поведінкових сценаріїв. Особливістю задач роботів у надзвичайних ситуаціях є висока розмірність інформації. Система повинна одночасно враховувати дані камер, лідарів, акустичних сенсорів, карти середовища та інформацію про ризики.

У зв'язку з цим процес навчання розглядається як пошук оптимальної поведінки у багатовимірному просторі можливих станів.

Слід показати багатовимірний простір, у якому траєкторія навчання поступово наближається до оптимальної області.

Такий підхід дозволяє пояснити необхідність використання Digital Twin для генерації великої кількості навчальних сценаріїв.

Контур адаптації поведінки. Ключовим елементом запропонованої моделі є адаптація поведінкової стратегії на основі нового досвіду. Після кожної місії результати роботи аналізуються, після чого оновлюються параметри навчальної моделі.

Схема повинна включати: результати місії → аналіз → оновлення моделі → тестування в Digital Twin → впровадження.

Такий контур забезпечує еволюційний розвиток системи, що дозволяє знижувати кількість помилок у подальших місіях.

Роль Digital Twin у формалізованій моделі. Цифровий двійник у процесі навчання виконує функції:

- безпечного експериментального середовища;
- генератора альтернативних сценаріїв;
- інструмента перевірки нових стратегій;

- накопичувача історичних даних експлуатації.

На схемі показати двонапрямний обмін даними: телеметрія → цифровий двійник → оновлені параметри → робот.

Наявність такого контуру суттєво зменшує ризики експериментального навчання у реальних умовах.

Безпечне адаптивне навчання. Оскільки роботехнічні системи працюють у небезпечному середовищі, адаптація моделей не може відбуватися без контролю. Тому у моделі вводиться механізм безпечного навчання, що передбачає:

- попередню перевірку стратегій у цифровому двійнику;
- обмеження небезпечних дій;
- повернення до базових алгоритмів у критичних ситуаціях.

Слід зобразити safety filter між модулем навчання та модулем керування.

Людина-оператор у циклі адаптації. Попри високий рівень автономності, оператор залишається важливою частиною навчального циклу. У запропонованій моделі оператор:

- задає стратегічні цілі навчання;
- коригує сценарії;
- контролює впровадження нових моделей.

На схемі показати, що оператор взаємодіє із рівнем планування, а не з низькорівневими командами робота.

Динаміка ефективності навчання. Процес адаптивного навчання має накопичувальний характер: із зростанням кількості ітерацій система демонструє покращення показників безпечності та ефективності.

На графіку — швидке покращення на початкових етапах та поступове насичення.

У підрозділі сформовано концептуальну модель процесу адаптивного навчання роботехнічних систем у межах програмного комплексу на основі Digital Twin. Показано, що ключовими елементами моделі є циклічний

характер навчання, безперервне оновлення стратегій та інтеграція фізичної системи з цифровим середовищем. Запропонована структура забезпечує можливість безпечної адаптації роботів до нових сценаріїв та створює основу для розгляду конкретних методів адаптивного навчання, що буде подано у наступному підрозділі.

3.2. Модель навчального циклу Digital Twin

Однією з ключових ідей запропонованого підходу є використання Digital Twin не лише як симуляційного середовища, а як активного елемента процесу адаптивного навчання роботехнічних систем. На відміну від класичних симуляторів, цифровий двійник забезпечує безперервну синхронізацію між фізичною платформою та її віртуальною моделлю, що дозволяє реалізувати замкнений цикл навчання, де досвід реальної експлуатації безпосередньо впливає на подальший розвиток алгоритмів керування [52; 53].

У задачах ліквідації наслідків надзвичайних ситуацій та роботи у складних умовах (техногенні аварії, військові дії, кризові сценарії на кшталт COVID-2019) можливість безпечного тестування нових стратегій має критичне значення. Саме тому модель навчального циклу Digital Twin виступає центральним елементом запропонованої концепції.

Загальна структура навчального циклу. Навчальний цикл Digital Twin ґрунтується на безперервному обміні інформацією між фізичною системою та її віртуальною копією. Основні етапи циклу включають:

1. збір телеметричних даних із реальної роботехнічної системи;
2. синхронізацію параметрів цифрового двійника;
3. моделювання альтернативних сценаріїв;
4. навчання або перенавчання моделей керування;
5. тестування стратегій у віртуальному середовищі;
6. впровадження оновлених моделей у фізичну систему.

На схемі необхідно показати замкнений контур: Physical Robot → Data Acquisition → Digital Twin → Learning Module → Validation → Deployment → Physical Robot.

У межах цього циклу цифровий двійник виконує роль середовища безпечної перевірки, що дозволяє мінімізувати ризики експлуатаційних помилок.

Потоки даних у навчальному циклі. Центральною особливістю моделі є data-driven підхід: усі рішення щодо адаптації алгоритмів базуються на накопичених даних експлуатації. Потоки даних включають:

- сенсорні дані реального часу;
- журнали виконаних дій;
- показники ефективності місій;
- інформацію про помилки або ризикові ситуації.

Схема повинна відображати два основні потоки:
– реальні дані → цифровий двійник;
– оновлені моделі → реальна система.

Такий підхід дозволяє забезпечити поступове наближення поведінки віртуальної моделі до реальної динаміки системи.

Генерація навчальних сценаріїв. Однією з найважливіших функцій Digital Twin є генерація сценаріїв, які можуть бути складними або небезпечними для реальних випробувань. У межах програмного комплексу моделюються:

- руйнування інфраструктури;
- динамічні перешкоди;
- нестабільні маршрути;
- обмеження комунікації;
- випадкові зміни середовища.

Рекомендується відобразити блок генератора сценаріїв із параметрами випадковості та рівнями складності.

Такий механізм дозволяє формувати широкий спектр навчальних ситуацій і підвищувати стійкість алгоритмів до непередбачуваних умов.

Механізм адаптації моделей. Після завершення кожної симуляційної або реальної місії результати аналізуються та використовуються для оновлення моделей керування. Процес адаптації включає:

- виявлення помилок поведінки;
- аналіз критичних подій;
- корекцію параметрів моделей;
- повторне тестування в Digital Twin.

На схемі показати: Evaluation → Model Update → Re-simulation → Validation.

Таким чином реалізується принцип безперервного вдосконалення, де кожна місія підвищує ефективність системи.

Контур безпечного впровадження моделей. Перед перенесенням нової стратегії у фізичний робот виконується багаторівнева перевірка безпеки. Це включає:

- тестування в екстремальних симуляційних сценаріях;
- порівняння з базовими алгоритмами;
- перевірку стабільності поведінки.

Схема повинна містити safety gate між модулем навчання та реальним роботом.

Такий механізм особливо важливий для систем, що працюють у бойових умовах або під час ліквідації техногенних катастроф.

Роль оператора у навчальному циклі. Оператор у запропонованій моделі виконує стратегічну функцію управління процесом навчання. Його роль полягає у:

- виборі типів сценаріїв;
- затвердженні нових стратегій;
- контролі ризиків під час впровадження.

Такий підхід дозволяє поєднати автономність системи та експертні знання людини.

Динаміка ефективності навчання. Використання Digital Twin забезпечує накопичувальний ефект: із кожним циклом система демонструє покращення показників ефективності.

На графіку — швидке початкове зростання, подальша стабілізація та поступове покращення.

Це підтверджує, що адаптивне навчання набуває характеру еволюційного процесу.

Переваги моделі навчального циклу Digital Twin. Запропонована модель забезпечує:

- безпечне тестування алгоритмів;
- постійне оновлення моделей на основі реальних даних;
- зменшення розриву між симуляцією і реальним середовищем;
- швидку адаптацію до нових сценаріїв;
- можливість масштабування для різних типів роботів.

Ці характеристики особливо важливі для систем, що працюють у динамічних умовах та повинні швидко переналаштовуватися під нові задачі.

3.3. Методи адаптивного навчання в програмному комплексі

Реалізація запропонованого програмного комплексу навчання роботехнічних систем на основі Digital Twin потребує використання методів, здатних забезпечити одночасно адаптивність, безпечність та стабільність поведінки роботів у складних умовах. Аналіз сучасних досліджень показує, що жоден окремий підхід до навчання не може повністю задовольнити вимоги систем, які функціонують у надзвичайних ситуаціях техногенного походження [46; 49]. У зв'язку з цим у межах роботи застосовується гібридний підхід, що поєднує декілька методів адаптивного навчання.

Загальна класифікація методів адаптивного навчання. У структурі програмного комплексу виділяються три основні групи методів:

- навчання з підкріпленням (Reinforcement Learning);
- навчання шляхом наслідування (Imitation Learning);
- гібридне навчання (Hybrid Learning), що поєднує класичні алгоритми керування з ML-моделями.

На схемі слід показати три гілки методів і їх інтеграцію через Digital Twin.

Такий розподіл дозволяє забезпечити баланс між швидкістю навчання, передбачуваністю поведінки та здатністю до адаптації.

Навчання з підкріпленням (Reinforcement Learning). Навчання з підкріпленням використовується для формування поведінкових стратегій у ситуаціях, де заздалегідь неможливо описати всі правила взаємодії з середовищем. Робот навчається шляхом повторних ітерацій, отримуючи оцінку результатів власних дій.

У межах Digital Twin RL використовується переважно для:

- автономної навігації в складному середовищі;
- вибору оптимальних траєкторій;
- адаптації до нових умов середовища.

Слід відобразити цикл: стан → дія → середовище → винагорода → оновлення політики.

Використання RL у віртуальному середовищі дозволяє виконувати тисячі ітерацій без ризику пошкодження роботехнічної системи.

Навчання шляхом наслідування (Imitation Learning). Для задач, де існують експертні знання людини-оператора, доцільним є використання imitation learning. У цьому випадку робот формує модель поведінки на основі записаних дій оператора.

Такий підхід застосовується для:

- маневрування у складних умовах;

- логістичних операцій;
- сценаріїв евакуації у небезпечному середовищі.

На схемі потрібно показати: оператор → демонстрації → модель → тестування в Digital Twin.

Перевагою методу є швидкий початок навчання без необхідності тривалих ітерацій пошуку оптимальної політики.

Гібридне навчання (Hybrid Learning). Основною особливістю запропонованого програмного комплексу є використання гібридного навчання, яке поєднує:

- базові детерміновані алгоритми керування;
- адаптивні моделі ML;
- механізми корекції через Digital Twin.

У цьому підході базові алгоритми забезпечують безпечну поведінку, тоді як навчальні моделі оптимізують рішення у нестандартних ситуаціях.

На схемі слід показати два паралельні контури керування: rule-based і learning-based із блоком інтеграції.

Такий підхід дозволяє уникати критичних помилок, характерних для повністю автономних моделей.

Сценарій-орієнтоване навчання. Важливою особливістю програмного комплексу є навчання, орієнтоване на сценарії. Digital Twin генерує різні типи середовищ, що відповідають реальним умовам експлуатації:

- техногенні аварії;
- порушення інфраструктури;
- логістичні сценарії;
- кризові ситуації типу пандемії.

Схема повинна показати генератор сценаріїв → модулі навчання → оцінювання.

Такий підхід підвищує здатність моделей до узагальнення та адаптації.

Safe Learning у процесі адаптації. Для роботехнічних систем, що працюють у небезпечних середовищах, критичною є проблема безпечного навчання. У програмному комплексі це реалізується через:

- обмеження допустимих дій;
- перевірку стратегій у Digital Twin;
- fallback до базових алгоритмів.

На схемі має бути safety layer між ML-модулем і виконавчою системою.

Це дозволяє уникати небезпечних дій навіть за умов активного навчання.

Адаптивне перенавчання моделей. Особливістю запропонованого підходу є можливість регулярного перенавчання моделей на основі даних реальної експлуатації. Після кожної місії:

- дані збираються та аналізуються;
- визначаються нові патерни середовища;
- проводиться додаткове навчання в Digital Twin.

Такий механізм забезпечує довготривалу еволюцію системи.

Порівняння методів адаптивного навчання. Для обґрунтування вибору гібридного підходу доцільно виконати порівняння основних методів за критеріями:

- швидкість навчання;
- безпечність;
- здатність до адаптації;
- обчислювальна складність.

Рекомендується подати радарну діаграму або стовпчиковий графік.

Результати показують, що гібридний підхід забезпечує найкращий баланс характеристик.

У підрозділі розглянуто методи адаптивного навчання, що застосовуються в програмному комплексі на основі Digital Twin. Визначено роль reinforcement learning, imitation learning та гібридних підходів у формуванні поведінкових стратегій роботехнічних систем. Показано, що

поєднання різних методів дозволяє забезпечити адаптивність системи при збереженні необхідного рівня безпеки. Отримані результати створюють основу для формування алгоритмів навчання та побудови поведінкових стратегій, які розглядаються у наступному підрозділі.

3.4. Розроблення динамічної моделі сегментації та програмної системи для сегментації зображень при реалізації комп'ютерного зору для роботизованих систем

У сучасному світі роботехніка відіграє ключову роль у вирішенні завдань, пов'язаних із безпекою, автоматизацією та оперативним реагуванням у надзвичайних ситуаціях. Роботи використовуються для дослідження небезпечних територій, виявлення хворих та постраждалих, маніпулювання об'єктами в недоступних умовах і моніторингу довкілля під час техногенних, епідеміологічних або природних катастроф та надзвичайних ситуацій. У таких сценаріях комп'ютерний зір є критично важливою складовою — він дозволяє системам автономно аналізувати навколишнє середовище, приймати рішення та виконувати дії з високою точністю. У таких умовах розпізнавання образів є однією з базових функцій комп'ютерного зору, оскільки дозволяє роботизованій системі співвідносити візуальні ознаки середовища з об'єктами, що мають значення для виконання місії.

Одним із найважливіших завдань комп'ютерного зору є сегментація зображень — процес виявлення та відокремлення об'єктів або областей на зображенні, що належать до одного класу. Для роботизованих систем, які працюють у складних умовах (погане освітлення, шум, непередбачуване середовище), висока точність і адаптивність алгоритмів сегментації є критично необхідними. У цьому контексті дослідження та порівняння як класичних підходів (таких як Watershed і GrabCut), так і сучасних методів на основі штучного інтелекту (таких як U-Net і DeepLab) залишаються надзвичайно актуальними. Отже, сегментація зображень доповнює

розпізнавання образів, оскільки не лише визначає наявність об'єкта, але й уточнює його просторові межі на зображенні.

Сегментація зображень є ключовим завданням комп'ютерного зору, особливо в контексті роботехніки та реагування на надзвичайні ситуації. Існує широкий спектр методів сегментації, які умовно поділяються на класичні (традиційні) методи та сучасні підходи на основі глибокого навчання.

Класичні алгоритми, такі як **Watershed** і **GrabCut**, широко застосовуються завдяки своїй простоті та ефективності в умовах обмежених обчислювальних ресурсів.

– **Watershed** ґрунтується на морфологічному аналізі зображення, розглядаючи його як топографічну поверхню. Метод є ефективним для розділення зображень із чітко визначеними межами, проте чутливий до шуму і може призводити до надмірної сегментації.

– **GrabCut** використовує мінімізацію енергії та гаусові сумішеві моделі (GMM) для відокремлення переднього плану від фону. Він потребує мінімальної взаємодії користувача для ініціалізації, що робить його зручним для інтерактивних застосувань.

Названі методи не потребують великих обсягів навчальних даних, що робить їх придатними для використання в умовах обмежених ресурсів або за відсутності розмічених наборів даних.

Сучасні підходи до сегментації активно використовують глибокі згорткові нейронні мережі (CNN), які демонструють високу точність і здатність до узагальнення.

– **U-Net**, запропонована Ronneberger та співавторами у 2015 році, поєднує шляхи кодування та декодування зі з'єднаннями пропуску (skip connections), що дозволяє точно локалізувати об'єкти навіть за обмеженої кількості навчальних даних. U-Net широко застосовується в медичній сегментації зображень і демонструє високу ефективність у задачах, що потребують піксельної точності.

– **DeepLab**, розроблена Chen та співавторами у 2016 році, використовує атруозні (розріджені) згортки та просторове пірамідальне об'єднання (ASPP) для захоплення багатомасштабної контекстної інформації. Додаткова інтеграція з умовними випадковими полями (CRF) дозволяє покращити локалізацію меж об'єктів. DeepLab продемонструвала високу точність сегментації на наборах даних PASCAL VOC і Cityscapes.

Порівняння класичних і сучасних методів показує, що вибір підходу залежить від конкретних вимог задачі. Класичні методи, такі як Watershed і GrabCut, є менш обчислювально затратними та не потребують навчання, що робить їх придатними для використання в умовах обмежених обчислювальних ресурсів або відсутності розмічених даних. Натомість методи на основі глибокого навчання, такі як U-Net і DeepLab, забезпечують вищу точність і кращу здатність до узагальнення, але потребують великих наборів даних і потужних обчислювальних ресурсів. У задачах розпізнавання образів та візуальної ідентифікації вибір між класичними і нейромережевими методами залежить від вимог до точності, швидкодії та доступності обчислювальних ресурсів.

У контексті надзвичайних ситуацій — таких як пошуково-рятувальні операції, гасіння пожеж, епідеміологічні спалахи або гуманітарне розмінування — точна сегментація зображень є критично важливою для ідентифікації об'єктів і прийняття рішень у реальному часі. Наприклад, U-Net і DeepLab можуть використовуватися для автоматичного виявлення хворих та постраждалих або небезпечних об'єктів на зображеннях, отриманих з дронів чи роботизованих систем, що значно підвищує ефективність рятувальних операцій. Для аварійних сценаріїв розпізнавання образів дозволяє своєчасно виявляти людей, транспортні засоби, уламки, потенційно небезпечні предмети та інші об'єкти, що впливають на рішення системи керування.

У сучасних умовах завдання ефективного застосування роботизованих систем у надзвичайних ситуаціях набуває дедалі більшої актуальності. Роботи активно використовуються для обстеження територій після обстрілів,

гуманітарного розмінування, розбору завалів, пошуку постраждалих, інспекції пошкоджених будівель, підвалів, замкнених просторів, а також для моніторингу зон, небезпечних для перебування людини.

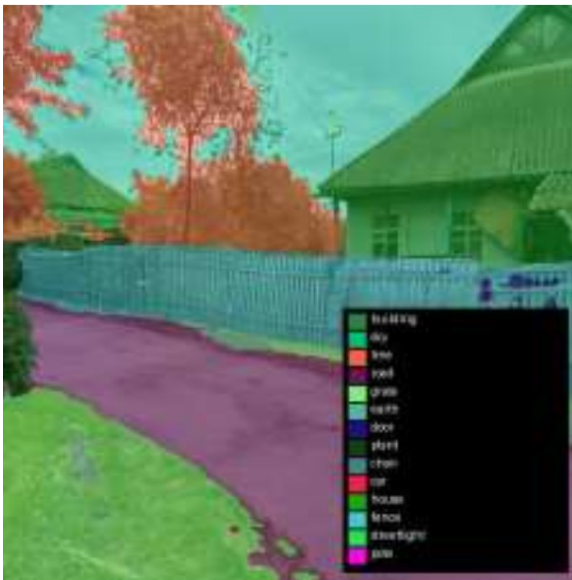
Якість функціонування таких систем значною мірою залежить від здатності комп'ютерного зору точно сегментувати об'єкти на зображеннях — зокрема, ідентифікувати людей, уламки, міни, прохідні зони та вибухонебезпечні предмети. Це завдання є особливо складним у реальних умовах, що характеризуються поганим освітленням, димом, пилом, шумами, динамічними сценами та непередбачуваними факторами. Завдання розпізнавання образів у таких умовах ускладнюється варіативністю ракурсів, частковим перекриттям об'єктів, шумами сенсорів та нестабільністю освітлення.

Крім того, обчислювальні та енергетичні ресурси таких систем часто є обмеженими, а підключення до хмарних сервісів може бути недоступним. За таких умов надзвичайно важливо правильно обрати метод сегментації. Традиційні алгоритми (наприклад, GrabCut, Watershed) не потребують навчання та споживають менше ресурсів, тоді як методи на основі нейронних мереж (U-Net, DeepLab) забезпечують вищу точність і кращу здатність до узагальнення.

Розглянута модель разом із відповідним алгоритмом і програмною реалізацією була застосована до 12 тестових зображень різного змісту та композиції (рис. 3.1–3.5).

На рисунку 3.1 наведено результати застосування розробленого програмного забезпечення до зображень: а) архітектурного пейзажу та б) природного пейзажу. Програма продемонструвала достатньо точну ідентифікацію будівель, неба, дерев, доріг, трави, ґрунту, архітектурних елементів, квітів, парканів, освітлювальних приладів і стовпів.

Візуалізація сегментації виконується за допомогою динамічної кольорової палітри, яка адаптується незалежно від кількості виявлених класів об'єктів для забезпечення чіткої візуальної диференціації.



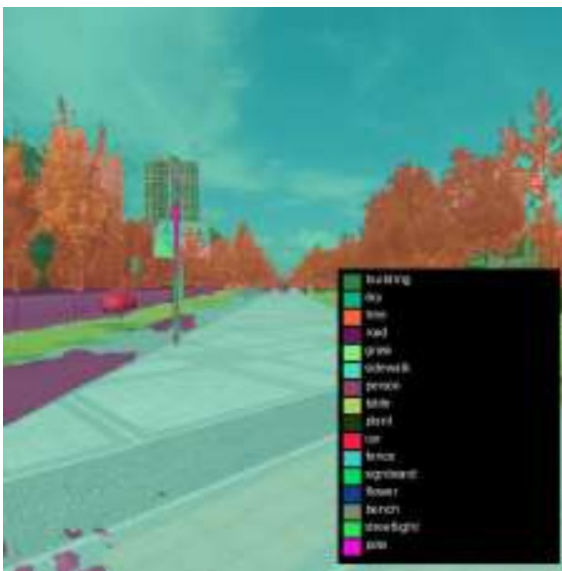
а)



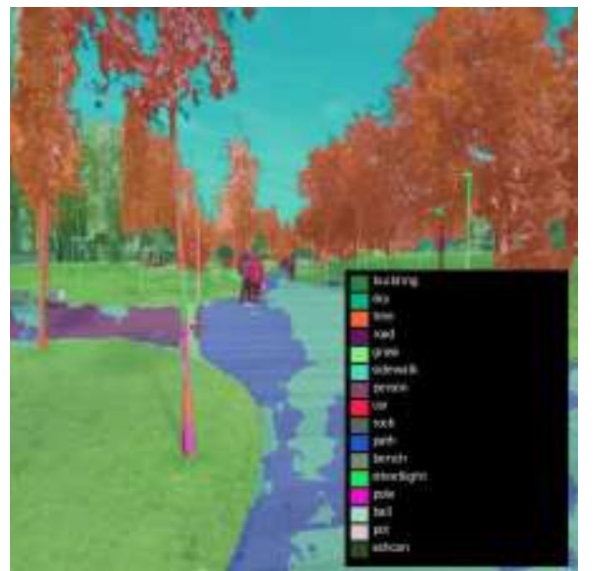
б)

Рисунок 3.1 – Результати роботи програмного застосунку сегментації зображень: а) архітектурний пейзаж; б) природний пейзаж

Було проведено порівняння результатів програми із середніми результатами трьох AI-застосунків: ChatGPT, Midjourney та Runway. Для прикладу на рисунку 1а відносна похибка становила 2,1%, а для випадку на рисунку 1б — 2,7% відповідно.



а)



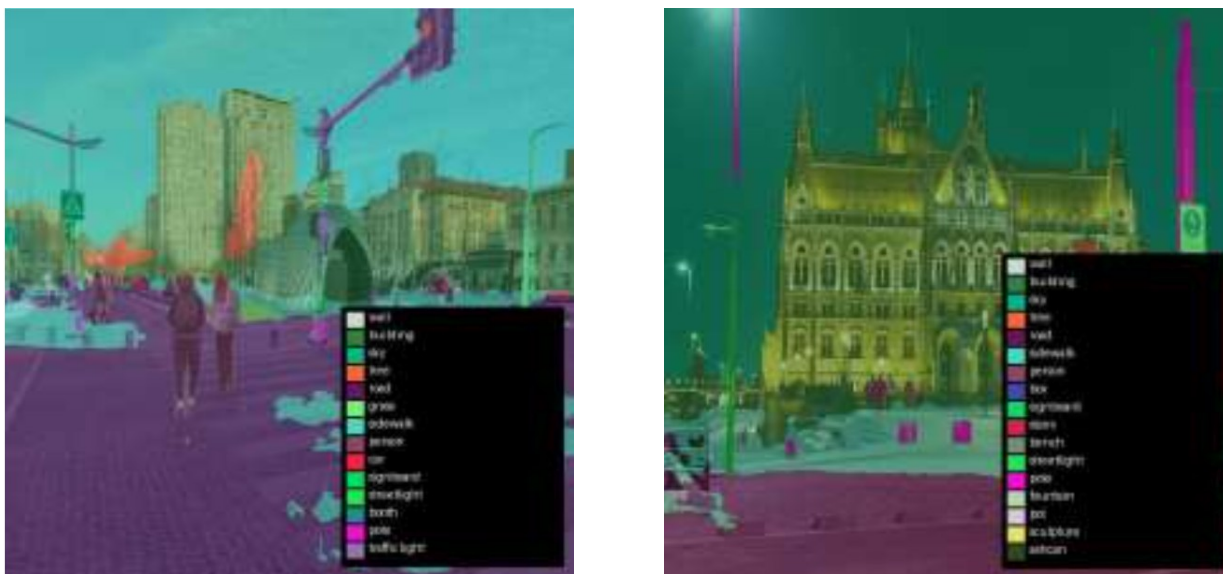
б)

Рисунок 3.2 – Результати сегментації для зображень зі змішаним вмістом

На рисунку 3.2 представлено результати сегментації зображень міського паркового ландшафту. Ці зображення містять велику кількість різних типів і

окремих об'єктів. Програма успішно ідентифікувала будівлі, небо, дерева, дороги, траву, людей, рослини, автомобілі та багато інших елементів.

Порівняння з усередненими результатами ChatGPT, Midjourney та Runway продемонструвало високий ступінь узгодженості. Відносна похибка для рисунка 3.2а становила 3,1%, а для рисунка 3.2б — 4,2%. З огляду на складність і насиченість зображень деталями, точність алгоритму можна вважати задовільною.



а)

б)

Рисунок 3.3 – Результати сегментації для типових міських зображень

На рисунку 3.3 наведено результати сегментації типового міського ландшафту, що містить понад десять типів об'єктів. Усі вони були успішно ідентифіковані. Будівлі, небо, рослинність, люди, дороги, транспортні засоби та інші елементи міського середовища були розпізнані з високою точністю. Отримані результати демонструють, що розпізнавання образів у міському середовищі може бути використане як допоміжний етап для побудови карти сцени та подальшого планування траєкторії руху.

Порівняльний аналіз підтвердив адекватність моделі. Відносна похибка становила 2,8% для рисунка 3.3а та 3,6% для рисунка 3.3б.



а)

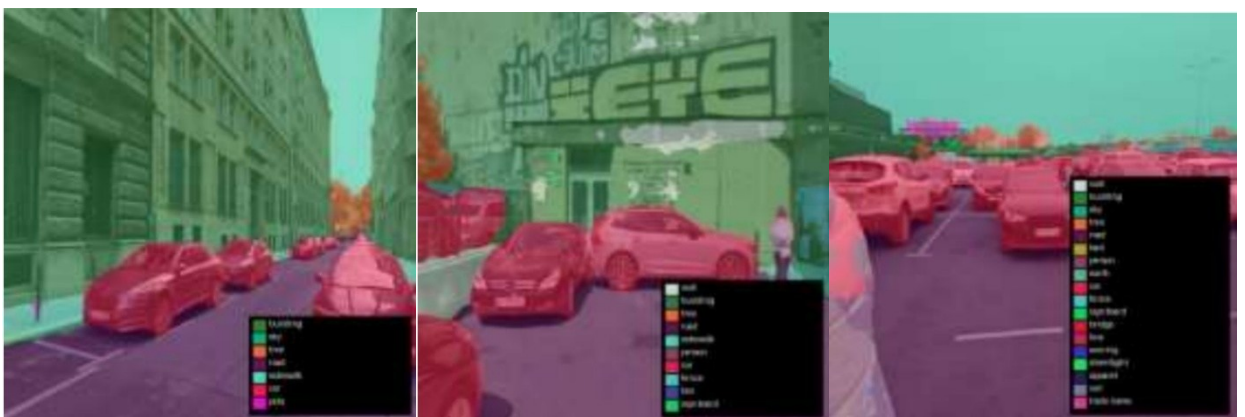
б)

в)

Рисунок 3.4 – Результати сегментації для зображень із колісними та повітряними транспортними засобами

На рисунку 3.4 представлено результати сегментації зображень із поодинокими транспортними засобами. Цей приклад є простішим порівняно з рисунком 3.5, де зображено велику кількість транспортних засобів. Метою порівняння є визначення різниці в точності сегментації між зображеннями з ізольованими об'єктами та з великою кількістю однотипних об'єктів.

Відносна похибка для рисунків 3.4а, 3.4б і 3.4в становить 2,3%, 2,8% і 3,1% відповідно. Для зображень із великою кількістю транспортних засобів похибка майже вдвічі більша: для рисунків 3.5а, 3.5б і 3.5в — 4,6%, 5,1% і 5,1% відповідно.



а)

б)

в)

Рисунок 3.5 – Результати сегментації для зображень із великою кількістю колісних транспортних засобів

Це свідчить про те, що складність і щільність об'єктів на зображенні суттєво впливають на точність сегментації, хоча результати залишаються в межах прийнятних значень.

Таким чином модель динамічної сегментації, алгоритм і його реалізацію, що охоплює як класичні математичні методи, так і сучасні методи на основі штучного інтелекту. Усі методи реалізовано мовою Python із використанням відкритих бібліотек.

Існуючі та розроблені нейронні мережі демонструють вищу точність, однак потребують значних обчислювальних ресурсів для проведення чисельних експериментів. У простіших сценаріях запропонований інструментарій може забезпечити результати, прийнятні для практичного застосування.

Отримані результати підтверджують доцільність комбінування різних підходів залежно від контексту застосування.

3.5. Метод оцінювання ефективності адаптивного навчання роботехнічних систем

Ефективність адаптивного навчання роботехнічних систем визначається не лише якістю сформованих поведінкових стратегій, але й здатністю системи стабільно функціонувати в умовах змінного середовища, характерного для надзвичайних ситуацій техногенного походження. У зв'язку з цим виникає необхідність розроблення методики оцінювання, яка дозволяє кількісно порівнювати результати навчання, аналізувати динаміку розвитку моделей та визначати доцільність їх впровадження у реальні роботехнічні системи.

У запропонованому програмному комплексі оцінювання виконує три основні функції:

- контроль якості процесу навчання;
- перевірка безпечності сформованих стратегій;
- порівняння ефективності різних методів адаптації.

Загальні принципи оцінювання. Метод оцінювання базується на багатокритеріальному підході, оскільки поведінка роботехнічної системи характеризується декількома взаємопов'язаними параметрами. Основні принципи оцінювання:

- комплексність (аналіз декількох критеріїв одночасно);
- порівнянність результатів між різними сценаріями;
- урахування реальних експлуатаційних даних;
- можливість автоматичного збору метрик у Digital Twin.

На схемі слід показати блоки: дані місії → обчислення метрик → інтегральна оцінка → прийняття рішення про оновлення моделі.

Основні критерії ефективності. У межах дослідження запропоновано використовувати такі групи критеріїв:

1. ефективність виконання завдання — час виконання, довжина маршруту, кількість досягнутих цілей;
2. безпека функціонування — кількість небезпечних ситуацій, зіткнень або критичних дій;
3. стійкість до змін середовища — здатність підтримувати продуктивність у нових сценаріях;
4. адаптивність — швидкість перенавчання після появи нових умов;
5. ресурсна ефективність — використання енергії та обчислювальних ресурсів.

Схема повинна мати деревоподібну структуру критеріїв.

Такий підхід дозволяє оцінювати систему комплексно, а не лише за одним показником продуктивності.

Метрики оцінювання безпечності. Оскільки роботехнічні системи орієнтовані на роботу у небезпечних середовищах, безпека є пріоритетним критерієм.

Основні метрики:

- кількість аварійних ситуацій за місію;

- відстань до перешкод;
- частота втручання оператора;
- кількість переходів у fallback-режим.

Метрики адаптивності. Для оцінювання здатності системи адаптуватися до нових умов вводяться такі показники:

- час стабілізації після зміни сценарію;
- кількість ітерацій перенавчання;
- швидкість відновлення продуктивності.

На графіку показати різке зниження performance після зміни умов і подальше відновлення.

Ці метрики дозволяють оцінити ефективність використання Digital Twin у процесі адаптації.

Інтегральна оцінка ефективності. Для порівняння різних моделей використовується інтегральний показник ефективності, який об'єднує кілька критеріїв у єдину оцінку.

На схемі показати вагове об'єднання критеріїв: безпека, ефективність, стабільність, адаптивність.

Такий підхід дозволяє об'єктивно порівнювати результати різних методів навчання.

Оцінювання в Digital Twin та реальному середовищі. Особливістю запропонованої методики є подвійне оцінювання:

- у симуляційному середовищі (Digital Twin);
- у реальних експлуатаційних умовах.

На графіку слід показати зближення результатів у процесі адаптації цифрового двійника.

Це дозволяє контролювати рівень достовірності симуляцій і зменшувати sim-to-real gap.

Візуалізація динаміки навчання. Для аналізу процесу адаптації доцільно використовувати графіки зміни ключових метрик у часі:

- продуктивність;
- стабільність;
- рівень ризику;
- ефективність використання ресурсів.

Рекомендовано побудувати декілька кривих на одному графіку для демонстрації взаємозв'язку метрик.

Порівняння методів адаптивного навчання. Метод оцінювання дозволяє виконувати порівняння різних алгоритмів:

- RL;
- imitation learning;
- hybrid learning.

Можна використати радарну діаграму або стовпчикове представлення.

Очікувано, гібридний підхід демонструє найкращий баланс між ефективністю та безпечністю.

Узагальнений алгоритм оцінювання. Процес оцінювання включає:

1. збір результатів місії;
2. обчислення локальних метрик;
3. формування інтегрального показника;
4. прийняття рішення щодо оновлення моделі;
5. передавання даних у Digital Twin для подальшого навчання.

У підрозділі розроблено метод оцінювання ефективності адаптивного навчання роботехнічних систем у межах програмного комплексу на основі Digital Twin. Запропоновано систему критеріїв і метрик, що дозволяє комплексно аналізувати продуктивність, безпечність і адаптивність моделей. Розроблений підхід забезпечує можливість об'єктивного порівняння алгоритмів навчання та створює основу для експериментальної перевірки ефективності запропонованих рішень у наступному розділі дисертаційної роботи.

3.6. Висновки до розділу 3

У третьому розділі дисертаційної роботи було розроблено та систематизовано моделі й методи адаптивного навчання роботехнічних систем на основі Digital Twin, що функціонують у межах запропонованого програмного комплексу. На відміну від традиційних підходів, орієнтованих на одноразове навчання або статичне налаштування алгоритмів, у розділі сформовано концепцію безперервного адаптивного навчання, яка забезпечує постійне оновлення поведінкових стратегій роботів у процесі експлуатації.

Виконано формалізацію процесу адаптивного навчання роботехнічних систем. Визначено основні складові навчального циклу: сприйняття середовища, аналіз стану системи, формування керуючих дій та оцінювання результатів. Показано, що процес навчання має циклічний характер і базується на безперервній взаємодії фізичної системи із цифровим двійником. Запропоновано концептуальну модель adaptive learning loop, яка забезпечує накопичення досвіду та поступове вдосконалення поведінки робота.

Розглянуто модель навчального циклу Digital Twin як ключового елемента адаптивного навчання. Встановлено, що цифровий двійник виконує функції синхронізації станів, генерації сценаріїв, тестування нових стратегій та безпечного перенесення моделей у фізичну систему. Показано, що застосування замкненого циклу «реальна система — цифровий двійник — навчання — впровадження» дозволяє суттєво зменшити розрив між симуляційним та реальним середовищами і підвищити ефективність підготовки роботехнічних систем до складних умов експлуатації.

Проаналізовано методи адаптивного навчання, що використовуються в програмному комплексі. Визначено роль навчання з підкріпленням, навчання шляхом наслідування та гібридних підходів у формуванні поведінкових стратегій. Обґрунтовано, що саме поєднання різних методів у межах єдиної архітектури забезпечує баланс між адаптивністю системи та її безпечністю. Особливу увагу приділено сценарій-орієнтованому навчанню та механізмам

safe learning, що є критично важливими для роботів, які працюють у небезпечних середовищах.

Розроблено алгоритмічні основи формування поведінкових стратегій роботехнічних систем. Запропоновано багаторівневий pipeline навчання, який включає підготовку даних, генерацію сценаріїв, навчання моделей, контроль безпеки та адаптивне оновлення політик керування. Визначено механізми інтеграції rule-based алгоритмів із навчальними моделями, що забезпечує стабільність роботи системи навіть у разі змін середовища. Показано, що алгоритмічна структура підтримує еволюційне покращення поведінки роботів у процесі багаторазових ітерацій навчання.

Розроблено метод оцінювання ефективності адаптивного навчання, який базується на багатокритеріальному підході. Запропоновано систему метрик, що охоплює продуктивність, безпечність, адаптивність та стійкість моделей. Показано, що використання інтегральної оцінки дозволяє об'єктивно порівнювати різні методи навчання та приймати рішення щодо оновлення моделей. Також обґрунтовано доцільність подвійного оцінювання результатів — у Digital Twin і в реальних умовах експлуатації.

Узагальнення результатів третього розділу дозволило сформулювати такі основні висновки:

1. Розроблено концептуальну модель адаптивного навчання роботехнічних систем, що базується на безперервному циклі взаємодії фізичної системи та Digital Twin.

2. Обґрунтовано використання цифрового двійника як центрального механізму безпечного навчання та адаптації моделей керування.

3. Визначено доцільність застосування гібридного підходу до навчання, який поєднує reinforcement learning, imitation learning і класичні алгоритми керування.

4. Запропоновано алгоритмічний pipeline формування поведінкових стратегій, що забезпечує адаптивність системи при збереженні вимог безпеки.

5. Розроблено метод оцінювання ефективності навчання, який дозволяє кількісно аналізувати результати адаптації та порівнювати різні навчальні методи.

Таким чином, у розділі сформовано теоретико-алгоритмічну основу програмного комплексу навчання роботехнічних систем на основі Digital Twin. Отримані результати створюють методологічне підґрунтя для переходу до наступного розділу дисертації, у якому буде здійснено реалізацію програмного комплексу та експериментальну перевірку запропонованих моделей і методів у прикладних сценаріях.

РОЗДІЛ 4. ЗАСТОСУВАННЯ ЦИФРОВОГО ДВІЙНИКА В ПРОЦЕСІ МОДЕЛЮВАННЯ, ПРОГНОЗУВАННЯ ТА АНАЛІЗУ НАСЛІДКІВ ТЕХНОГЕННОГО ПОХОДЖЕННЯ

4.1. Автоматизована система збору та аналізу інформації про виявлені повітряні об'єкти

Огляд прикладної проблеми. Вирішальним чинником результативного перехоплення повітряних цілей є оперативність їх виявлення. Проте реалізація цього завдання істотно ускладнюється масштабністю території України, обмеженою кількістю технічних засобів контролю повітряного простору та специфікою тактики противника. Активне застосування ракет і безпілотних літальних апаратів із наднизькими висотами польоту та значними швидкостями суттєво знижує ефективність традиційних засобів протиповітряної оборони, оскільки навіть незначна затримка у передачі інформації може унеможливити своєчасне перехоплення.

Наявні на початку широкомасштабної агресії технічні засоби спостереження — зокрема наземні радіолокаційні станції П-18 «Терек», П-19 «Дуга», 36Д6М1-1, 80К6КС1, мобільні РЛС TRML-4D та IRIS-T SLM, а також тепловізійні, оптичні й акустичні системи, пасивні сенсори та засоби радіоелектронної розвідки — продемонстрували належну ефективність у бойових умовах. Однак їх кількісні можливості були недостатніми для забезпечення безперервного й суцільного моніторингу повітряного простору в межах усієї території держави.

У цих умовах особливого значення набув потенціал громадянського суспільства. З огляду на низьковисотну траєкторію руху багатьох повітряних цілей цивільні особи нерідко здатні фіксувати їх візуально або за характерними акустичними ознаками. Така обставина створює передумови для організованого залучення населення до процесу оперативного

інформування підрозділів ППО з подальшим накопиченням і аналітичним опрацюванням отриманих даних.

Практика перших тижнів повномасштабного вторгнення засвідчила наявність поодиноких випадків, коли оперативна передача інформації про повітряну ціль дозволяла забезпечити її успішне ураження. Водночас відсутність системності знижувала загальну ефективність такого механізму. Для досягнення стабільного результату необхідне одночасне виконання трьох умов: своєчасна фіксація об'єкта, наявність надійного каналу комунікації з підрозділами ППО та передача максимально точної інформації про місце спостереження й напрям руху цілі.

Саме сукупність зазначених факторів зумовила потребу у створенні програмного рішення, здатного автоматизувати процес збору повідомлень, їх обробку та негайну передачу операторам протиповітряної оборони, а також формувати статистичні масиви даних для подальшого прогнозування та стратегічного планування.

У відповідь на ці виклики було розроблено спеціалізований програмний комплекс, що поєднав функцію оперативного сповіщення з розвиненим аналітичним інструментарієм. У його структурі реалізовано математичні моделі прогнозування та алгоритми адаптивного зонування території Дніпропетровської області з метою оптимального розміщення стаціонарних засобів ППО з урахуванням їх обмеженої кількості та складності передислокації. Рішення відповідало вимогам інформаційної безпеки, включно із захистом персональних даних користувачів, та було впроваджене під назвою «Система альтернативного оповіщення “Світло”».

Система була створена та інтегрована у березні 2022 року, після чого пройшла етапи тестування й доопрацювання безпосередньо в умовах активної фази бойових дій. У період з березня по грудень 2022 року вона активно застосовувалася та фактично не мала повноцінних аналогів. За даними Повітряного командування «Схід», попри альтернативний статус, її ефективність підтверджена практикою: із використанням САО «Світло» було

виявлено понад два десятки повітряних цілей противника, серед яких — балістична ракета типу «Іскандер-М». Навіть після появи інших рішень зі схожим функціоналом система продовжує використовуватися окремими регіональними підрозділами ППО завдяки доведеній результативності.

Предметна постановка задачі. Зона відповідальності Повітряного командування «Схід» охоплює східні та частково центральні області України, які характеризуються підвищеною інтенсивністю повітряної активності противника. До цієї зони належать Харківська, Дніпропетровська, Донецька (разом із тимчасово окупованими територіями), Луганська (разом із ГОТ), а також окремі райони Запорізької, Кіровоградської, Полтавської та Сумської областей.

З урахуванням масштабності території та оперативних потреб було визначено архітектуру програмного комплексу, що складалася з трьох взаємопов'язаних складових: клієнтського модуля, адміністративної панелі та серверної інфраструктури.

Клієнтська частина реалізовувалася у форматі мобільного застосунку, призначеного для цивільних осіб, представників територіальної оборони та учасників громадських формувань. Інтерфейс додатку був максимально спрощений з метою забезпечення швидкої передачі ключових даних — координат фіксації повітряної цілі та орієнтовного напрямку її руху — із мінімальними витратами часу з боку користувача.

Адміністративний модуль функціонував у вебформаті та використовувався представниками підрозділів ППО з дотриманням встановлених вимог інформаційної безпеки. У межах цієї панелі відображалася інтерактивна карта України з нанесеними позначками виявлених повітряних об'єктів, їх просторовими координатами та вектором переміщення. Функціонал передбачав можливість налаштування принципів візуалізації міток, контролю часу, що минув із моменту фіксації цілі, а також редагування, відновлення чи видалення відповідних записів.

Окремою функціональною можливістю адміністративної панелі було відстеження геолокації зареєстрованих користувачів клієнтського модуля навіть у випадку відсутності активних повідомлень про повітряні загрози. Такий механізм дозволяв визначати фактичні рубежі спостереження в межах зони відповідальності Повітряного командування «Схід» та оцінювати просторове покриття мережі спостерігачів.

Серверна складова комплексу забезпечувала технічну реалізацію процесів приймання, структуризації, зберігання та передавання даних, а також їх подальше відображення в адміністративному інтерфейсі, об'єднуючи програмні й апаратні ресурси в єдину систему обробки інформації.

Математична постановка задачі. Нехай Ω – просторово обмежена сукупність, до складу якої входять території Харківської, Дніпропетровської, Донецької (разом із тимчасово окупованими територіями), Луганської (разом із ТОТ), а також окремі частини Запорізької, Кіровоградської, Полтавської та Сумської областей України (рис. 4.1).



Рисунок 4.1 – Частина території України, яка входить до зони відповідальності Повітряного командування «Схід»

Об'єднання підмножин $\Omega_1, \dots, \Omega_N$ множини $\Omega \subset E_n$ позначимо як допустиме розбиття цієї множини за умови

$\bigcup_{i=1}^N \Omega_i = \Omega, \text{mes}(\Omega_i \cap \Omega_j) = 0, i, j = 1, \dots, N, (i \neq j),$ де $N > 0$ – визначене натуральне число. Надалі вважатимемо $N = 100$.

Позначимо через Σ_{Ω}^N сукупність усіх допустимих розбиттів множини $\Omega \subset E_n$ на фіксовану кількість N її підмножин, що є вимірними за Лебегом:

$$\Sigma_{\Omega}^N = \{\bar{\omega} \equiv \{\Omega_1, \dots, \Omega_N\} \in \Omega^N : \bigcup_{i=1}^N \Omega_i = \Omega, \text{mes}(\Omega_i \cap \Omega_j) = 0, i, j = 1, \dots, N, i \neq j\}$$

Далі, через τ_1, \dots, τ_N позначимо множину модельних центрів дислокації засобів ППО для відповідних підмножин $\Omega_1, \dots, \Omega_N$ відповідно, які надалі визначатимемо як центри цих підмножин: $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega_i, i = 1, \dots, N,$ і припускатимемо, що координати всіх центрів є заданими. Для визначеності розмістимо виділену на рис. 4.1 зону відповідальності ПвК «Схід» у межах квадрата зі стороною 1 та розташуємо центри τ_1, \dots, τ_N рівномірно в межах цього квадрата. Таким чином

$$\begin{aligned} &\tau_1(0,05;0,05), \tau_2(0,05;0,15), \dots, \tau_{10}(0,05;0,95), \\ &\tau_{11}(0,15;0,05), \tau_{12}(0,15;0,15), \dots, \tau_{20}(0,15;0,95), \\ &\dots \\ &\tau_{91}(0,95;0,05), \tau_{92}(0,95;0,15), \dots, \tau_{100}(0,95;0,95) \end{aligned} \tag{4.1}$$

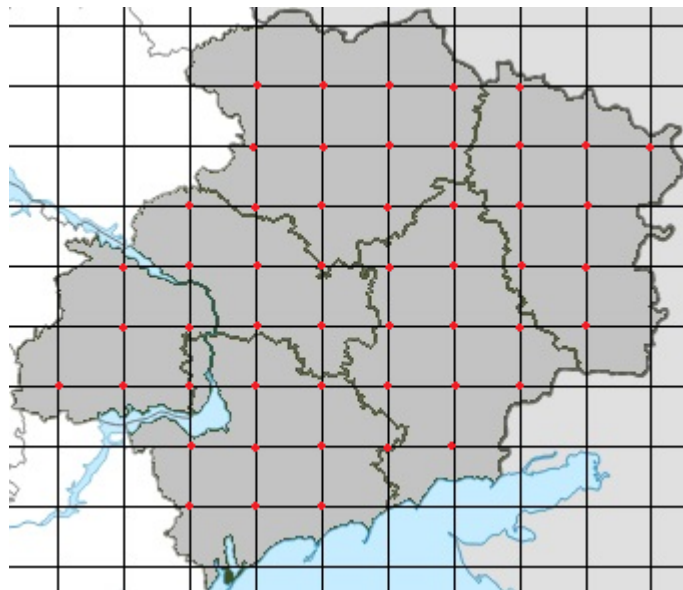


Рисунок 4.2 – Область Ω що вписана у квадрат зі стороною 1 із сіткою з кроком h , у комірках якої розміщені центри (4.1)

На рис. 4.2 зображено область Ω вписану в квадрат зі стороною 1 із нанесеною сіткою з кроком h , у вузлах якої розміщено центри (4.1).

Як впливає з рис. 4.2, центри з порядковими номерами 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 19, 20, 21, 22, 23, 31, 32, 40, 41, 50, 51, 60, 69, 70, 71, 72, 78, 79, 80, 81, 82, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100 розташовані за межами досліджуваної області. Отже, 49 центрів виключаються з подальшого розгляду через неможливість їх використання для розміщення відповідних об'єктів. τ_i .

Постановка задачі. Визначити розбиття $\bar{\omega} = \{\Omega_1, \dots, \Omega_N\} \in \sum_{\Omega}^N$ множини $\Omega \subset E_n$ і векторну функцію $c(x, \tau, t) = (c_1(x, \tau_1, t), \dots, c_N(x, \tau_N, t))$, визначену м.в. для $x \in \Omega$ при заданому фіксованому наборі центрів $\tau = \{\tau_1, \dots, \tau_N\} \subset \Omega^N$ та всіх $t \in [0, T]$, які забезпечують $\inf_{\bar{\omega} \in \sum_{\Omega}^N; c(\cdot, \cdot, \cdot) \in L_2^N(\Omega \times \Omega \times [0, T])} F(\bar{\omega}, c(\cdot, \cdot, \cdot))$ де

$$F(\bar{\omega}, c(\cdot, \cdot, \cdot)) = \int_0^T \sum_{i=1}^N \int_{\Omega_i} (c_i(x, \tau_i, t) \cdot m(x, \tau_i) + a_i) \rho(x) dx dt \quad \text{за наявності умов}$$

$$\frac{dc_i(x, \tau_i, t)}{dt} = \sum_{j=1}^N A_{ij} \cdot d_j \cdot c_i(x, \tau_i, t), 0 \leq t \leq T, \quad c_i(x, \tau_i, t_0) = c_{0i}(x, \tau_i), i = 1, \dots, N \quad \text{м.в.}$$

для $x \in \Omega$, при фіксованих $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega_i, i = 1, \dots, N$. Тут $c_i(x, \tau_i, t), i = 1, \dots, N$, – невідомі дійснозначні функції, задані на $\Omega \times \Omega \times [0, T]$, що для кожного фіксованого $\tau_i = (\tau_i^{(1)}, \dots, \tau_i^{(n)}) \in \Omega_i$ є неперервно диференційовними за аргументом t на відрізку $[0, T]$ м.в. для $x = (x^{(1)}, \dots, x^{(n)}) \in \Omega$, є обмеженими та вимірними за відповідним аргументом x на Ω для всіх $t \in [0, T]$; $m(x, \tau_i)$, $c_{0i}(x, \tau_i)$ – визначені наперед дійснозначні функції, задані на $\Omega \times \Omega$, є обмеженими та вимірними відносно відповідного аргументу $x \in \Omega$ при будь-якому фіксованому $\tau_i \in \Omega_i$ для всіх $i = 1, \dots, N$; $\rho(x)$ – визначена невід'ємна функція, що є обмеженою та вимірною на Ω ; $a_i, i = 1, \dots, N$, – задані, як правило, невід'ємні числа; $0 \leq A_{ij} \leq 1, i, j = 1, \dots, N$, – задані числові параметри; $T > 0$ та $t_0 \in [0, T]$ задані. Далі для простоти будемо вважати $\rho(x) = 1$, $d_i = 1$, $a_i = 0$.

Результати чисельного експерименту. Система автоматизованого оповіщення «Світло» була розроблена з урахуванням потреб функціонування в умовах воєнного стану. Її ключове призначення полягає в оперативному повідомленні про ракетні удари та інші загрози, зумовлені збройною агресією противника. Висока швидкість обробки та передавання інформації забезпечує своєчасне отримання даних про потенційні ризики та можливість негайного реагування.

На відміну від міжнародних платформ, зокрема OnSolve MIR3 чи AlertHub Pro, CAO «Світло» орієнтована безпосередньо на виявлення та супровід воєнних загроз. Така спеціалізація підвищує її прикладну ефективність у національному контексті, де фактор часу при інформуванні про небезпеку має критичне значення.

Однією з суттєвих переваг системи є інтеграція з підрозділами протиповітряної оборони, що забезпечує підвищену достовірність і швидкість формування сповіщень. Це, своєю чергою, сприяє зростанню рівня захисту цивільного населення, надаючи додатковий часовий ресурс для реагування на можливу загрозу.

Функціональні можливості інтерфейсу передбачають відображення інтерактивної карти як у масштабі всієї держави, так і окремих регіонів. За відсутності активних повідомлень карта не містить позначок і відображається в порожньому вигляді (рис. 4.3).

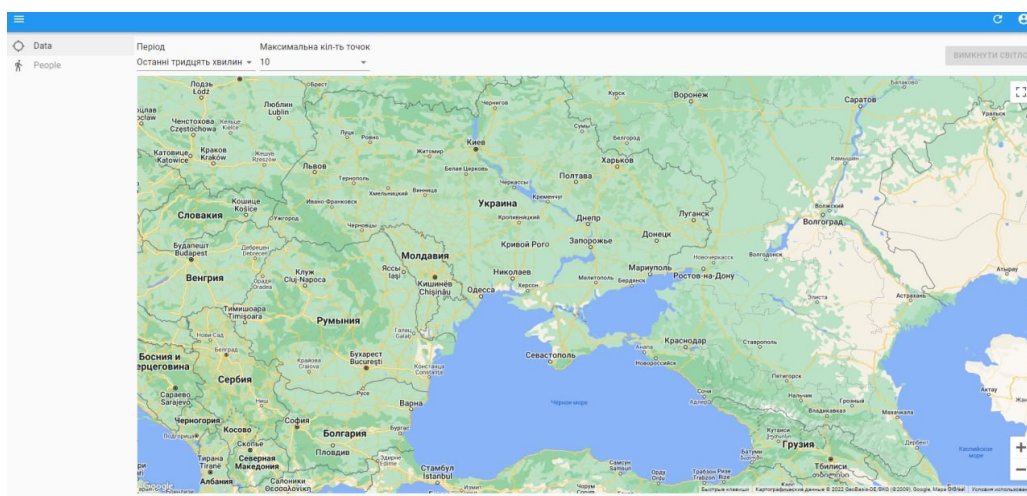


Рисунок 4.3 – Загальний інтерфейс адміністративної панелі

Слід зазначити, що після надходження повідомлення про фіксацію ворожої повітряної цілі в інтерфейсі адміністративної панелі автоматично відображаються маркери, колірна насиченість яких відображає час, що минув із моменту отримання відповідної інформації. З плином часу інтенсивність забарвлення поступово зменшується, а тривалість їх відображення визначається налаштуваннями головного меню панелі. Найбільш нові повідомлення позначаються маркерами з максимально яскравим кольором, тоді як попередні — відображаються більш блідо.

На рис. 4.4 продемонстровано приклад руху «ворожої повітряної цілі» за нелінійною траєкторією: а) у напрямку з Харкова до Дніпра; б) у межах міста Дніпро.

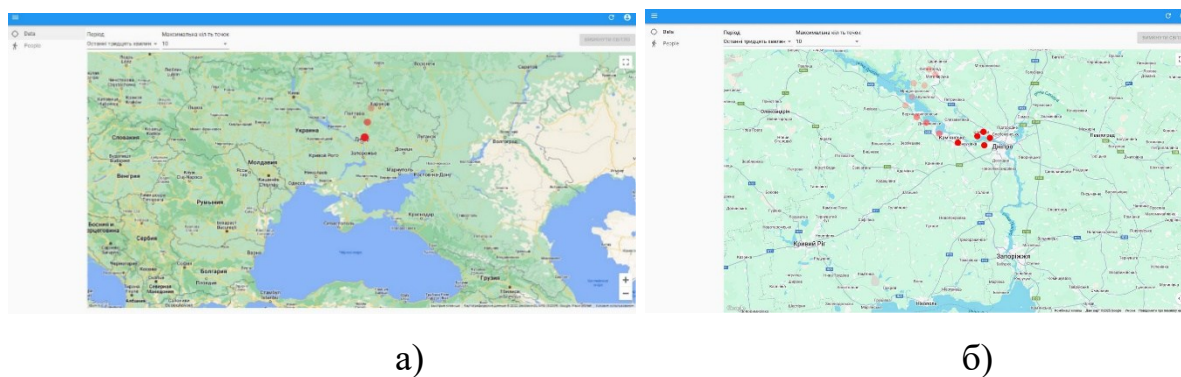


Рисунок 4.4 – Масштабоване представлення інтерфейсу адміністративної панелі з відображенням отриманих вхідних даних, диференційованих за часом їх надходження

З огляду на можливість масштабування карти в межах САО «Світло» забезпечується детальне опрацювання як окремих регіонів, так і території держави загалом. На рис. 4.5 подано модельне відображення нелінійної траєкторії переміщення ворожої повітряної цілі в межах міста Кам'янське. Перші шість точок, позначені градієнтом червоного кольору, відповідають фактично зафіксованим координатам спостереження об'єкта. Дві наступні точки відображають прогнозоване продовження його руху, отримане шляхом екстраполяції з урахуванням нелінійного характеру траєкторії.

З метою прогнозування подальшого переміщення ракети до функціоналу було інтегровано модуль екстраполяції, реалізований на основі методу найменших квадратів. Активація цієї опції здійснюється через кнопку «Припущення», розміщену у вкладці Data адміністративної панелі САО «Світло». Водночас з огляду на варіативність та часті зміни траєкторій руху ворожих повітряних цілей застосування екстраполяційного алгоритму продемонструвало результативність приблизно у 15 % випадків.

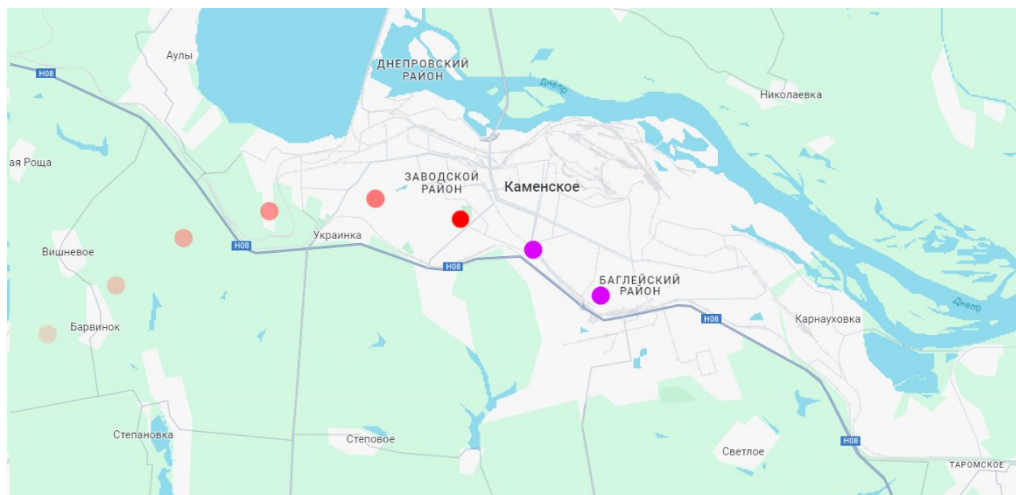


Рисунок 4.5 – Масштабоване відображення інтерфейсу адміністративної панелі з прогнозованою (екстрапольованою) траєкторією переміщення ворожої повітряної цілі

Графічне відображення на адміністративній панелі АСО «Світло» масованих атак із застосуванням безпілотників, за яких у межах обмеженої території одночасно фіксується переміщення значної кількості ворожих повітряних об'єктів, наведено на рис. 4.6.

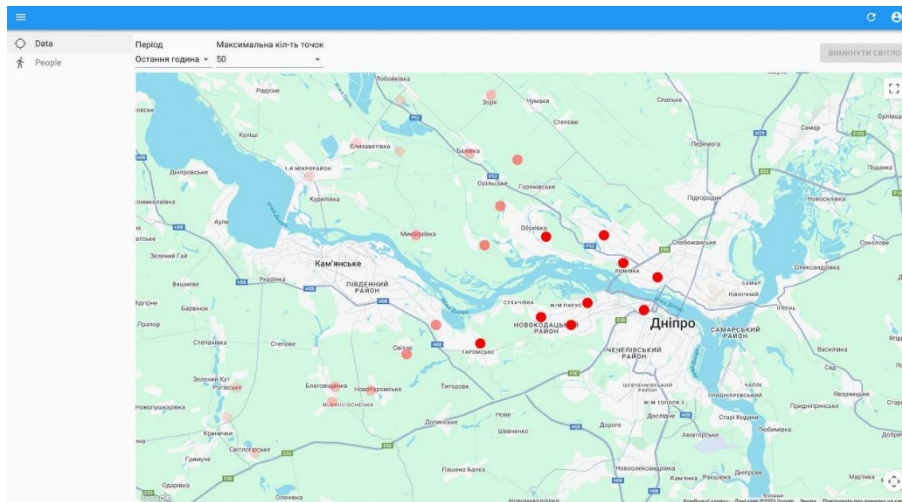


Рисунок 4.6 – Інтерфейс адміністративної панелі АСО «Світло» під час масованих атак безпілотників у межах окремого населеного пункту

У такому випадку часові деталізації прольоту забезпечує можливість визначення не лише напрямку переміщення ворожих повітряних цілей, а й їх кількісних характеристик, що дозволяє ідентифікувати та розмежовувати їх між собою.

У таблиці 4.1 наведено результати чисельного експерименту для різної кількості вогневих груп, що були застосовані для Дніпропетровського регіону. У дослідженні було враховано, зокрема, загальну площу, яку зазначена кількість вогневих груп покриває, частку цієї площі від площі Дніпропетровської області, середній час відгуку та відповідне значення цільового функціоналу.

Як свідчать дані таблиці 4.1, збільшення кількості вогневих груп у межах фіксованої території не забезпечує пропорційного зростання цільового показника. Водночас спостерігається істотне скорочення середнього часу реагування, що якісно підвищує загальну ефективність застосування вогневих підрозділів.

Результати чисельного моделювання за умови варіювання кількості вогневих груп у межах Дніпропетровської області

Кількість вогневих груп	Загальна площа покриття (км ²)	Частина площі від загальної площі Дніпропетровського регіону (%)	Середній час відгуку (хв.)	Значення цільового функціоналу
20	23,8	62,4	10,4	0,61
30	28,4	72,2	8,4	0,73
40	31,85	81,0	8,1	0,82

На основі сформульованої в цьому розділі математичної моделі динамічної задачі оптимального розбиття множини та інформації про ворожі повітряні цілі, отриманої за допомогою АСО «Світло» під час дев'яти масованих атак, здійснених Росією упродовж 2022–2024 років, було визначено оптимальні варіанти розміщення вогневих груп на території Дніпропетровської області. Результати чисельного розв'язання задачі для зазначеної території наведено нижче у вигляді дев'яти прикладів.

У кожному з прикладів застосовуються такі умовні позначення. Дата й час проведення чисельного експерименту подаються у форматі рік/місяць/день–година/хвилина. Характеристика повітряних цілей містить відомості про напрям їх руху (N — з півночі, S — з півдня, W — із заходу, E — зі сходу) та координати точки перетину межі Дніпропетровської області. Для визначення цих координат картографічне зображення області було вписано в прямокутник розміром 500×300 умовних одиниць.

1. Дата-час: 22/10/04-12:04. Кількість вогневих груп: 6. Характеристика ворожих повітряних цілей (15 шт.): S:53:112 N:63:209 E:32:-13 E:113:-85 S:24:141 N:89:304 N:110:224 E:10:-18 S:115:163 W:84:219 E:190:44 E:163:-76 N:13:207 N:36:230 E:111:24. Координати оптимального

розміщення вогневих груп: (170;115), (196;187), (120;67), (113;25), (293;27), (90;46).

2. Дата-час: 22/11/18-08:01. Кількість вогневих груп: 5. Характеристика ворожих повітряних цілей (13 шт.): N:146:353 N:135:193 E:144:36 S:141:75 W:184:136 S:93:28 W:85:268 E:65:-63 W:79:150 N:110:213 W:0:185 N:45:352 S:108:41. Координати оптимального розміщення вогневих груп: (122;158), (97;21), (195;162), (180;191), (272;52).

3. Дата-час: 22/12/16-14:12. Кількість вогневих груп: 8. Характеристика ворожих повітряних цілей (17 шт.): W:9:122 W:11:217 N:124:305 N:95:335 E:58:-79 E:10:-75 N:145:223 N:138:259 S:100:118 E:133:47 W:198:138 N:138:327 W:159:245 S:28:34 S:9:14 S:10:121 W:175:260. Координати оптимального розміщення вогневих груп: (196;27), (40;175), (42;44), (108;77), (174;121), (174;106), (217;173), (89;111).

4. Дата-час: 23/05/12-03:12. Кількість вогневих груп: 6. Характеристика ворожих повітряних цілей (15 шт.): W:102:233 S:53:131 W:167:96 E:174:10 S:30:36 W:161:182 N:143:241 S:38:114 E:51:-8 E:196:85 N:3:254 W:18:26 W:153:111 S:119:134 N:133:352. Координати оптимального розміщення вогневих груп: (103;2), (99;189), (242;29), (37;197), (43;130), (189;34).

5. Дата-час: 23/06/21-23:05. Кількість вогневих груп: 6. Характеристика ворожих повітряних цілей (15 шт.): W:92:129 W:6:168 S:115:10 N:128:184 N:71:223 N:50:214 S:54:151 S:26:74 S:50:166 N:132:181 W:82:142 S:20:72 W:17:142 E:169:52 S:11:99. Координати оптимального розміщення вогневих груп: (185;47), (187;105), (90;35), (277;90), (62;165), (268;54).

6. Дата-час: 23/08/03-16:13. Кількість вогневих груп: 4. Характеристика ворожих повітряних цілей (15 шт.): S:58:34 E:3:32 W:179:257 S:47:128 S:80:110 N:117:315 E:36:17 S:92:70 W:129:198 E:65:69 E:43:13 E:142:-82 N:17:331 S:72:100 N:107:256. Координати оптимального розміщення вогневих груп: (149;152), (204;147), (145;199), (163;113).

7. Дата-час: 24/01/08-18:00. Кількість вогневих груп: 6.
Характеристика ворожих повітряних цілей (15 шт.): S:110:81 E:94:-20 E:59:-67
E:139:77 S:43:153 S:42:164 N:39:332 E:199:72 E:167:-14 E:147:29 W:160:229
N:50:356 S:48:142 E:47:51 N:13:221. Координати оптимального розміщення
вогневих груп: (9;153), (231;14), (82;168), (160;198), (125;155), (171;54).

8. Дата-час: 24/03/14-16:42. Кількість вогневих груп: 4.
Характеристика ворожих повітряних цілей (15 шт.): S:127:142 S:46:51
W:180:184 W:88:197 W:1:100 N:18:183 W:171:176 W:46:132 S:77:40 E:138:82
N:78:238 S:124:33 S:29:172 W:14:195 N:86:244. Координати оптимального
розміщення вогневих груп: (165;122), (222;160), (146;114), (118;11).

9. Дата-час: 24/08/20-10:19. Кількість вогневих груп: 6.
Характеристика ворожих повітряних цілей (14 шт.): E:111:24 S:11:99 S:109:36
E:101:-4 W:46:254 E:58:71 N:138:352 E:121:-60 W:133:105 W:189:129 E:103:49
N:146:329 S:55:83 S:72:153. Координати оптимального розміщення вогневих
груп: (277;182), (217;64), (80;143), (62;133), (289;104), (154;40).

Оцінка ефективності роботи програмного комплексу. Для визначення результативності функціонування програмного комплексу було сформовано перелік основних показників ефективності, до яких віднесено:

- час реагування системи;
- надійність роботи (частоту відмов та тривалість простоїв);
- рівень безпеки.

З метою оцінювання масштабованості рішення здійснено навантажувальне тестування та проаналізовано продуктивність системи в умовах пікових звернень. На рис. 4.7 подано графічне відображення результатів тестування САО «Світло» під час максимального навантаження.

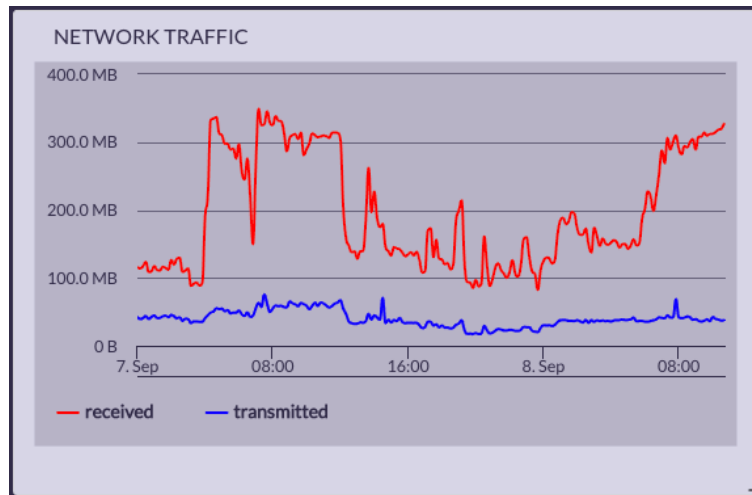


Рисунок 4.7 – Графік завантаження системи

Базові модулі програмного комплексу САО «Світло» розроблено із застосуванням мови програмування JavaScript. На рис. 4.8 продемонстровано адаптивність та кросплатформену сумісність інтерфейсу, що забезпечує коректне відображення на різних типах екранів і пристроїв. Такий підхід є принципово важливим для гарантування зручності користування та доступності системи, зважаючи на те, що мобільний застосунок САО «Світло» функціонує на різних платформах мобільних пристроїв.

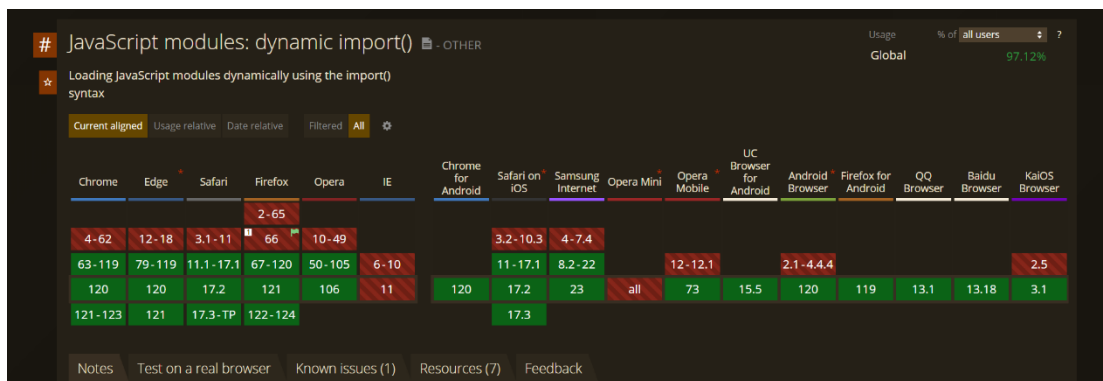


Рисунок 4.8 – Підтримка JavaScript різними системми

У таблиці 4.2 подано узагальнені результати оцінювання комплексної ефективності програмного комплексу з акцентом переважно на його технічні характеристики та перспективи експлуатації. Такий аналіз дає змогу оцінити фактичний рівень результативності системи, а також окреслити потенційні напрями її подальшої модернізації й адаптації до майбутніх викликів.

Результати комплексної оцінки програмного комплексу САО «Світло»

Метрика	Оцінка та коментар
Час Реакції Системи (Response Time)	Середній рівень. Обумовлений пріоритетністю швидкої передачі інформації до адміністративної панелі та зменшенням навантаження на вхідний мережевий трафік.
Надійність (Reliability)	Високий рівень. Досягається завдяки застосуванню оптимізованих алгоритмів, доопрацьованих на основі даних, отриманих у процесі активної експлуатації системи.
Масштабованість (Scalability)	Висока. Пріоритетом застосунку є забезпечення його ефективного використання значною кількістю користувачів.
Безпека (Security)	Висока. Такий рівень визначався ще на етапі проектування з огляду, зокрема, на необхідність захисту даних щодо місць концентрації цивільного населення.

Таким чином, у підрозділі розглянуто розроблення та експериментальну апробацію системи альтернативного оповіщення «Світло» як прикладу програмного модуля ситуаційної обізнаності, орієнтованого на підвищення ефективності реагування на загрози, пов'язані з ракетними та іншими повітряними ударами в умовах воєнного стану. Основну увагу було зосереджено на аналізі її функціональних можливостей, зокрема роботі з повідомленнями, візуалізації геоінформаційних даних, підтримці операторського аналізу та потенційному використанні результатів для формування цифрової моделі кризової ситуації.

Окрему увагу приділено динамічній задачі оптимального розбиття території області, зокрема зони відповідальності Повітряного командування «Схід», з метою розміщення 51 вогневої групи, координати центрів яких задаються вручну з урахуванням тактичної доцільності. Розроблена математична модель дозволяє гнучко адаптуватися до змін у розстановці сил,

напрямах загроз і структурі місцевості. Такий підхід має суттєве значення в умовах мобільного театру бойових дій, де швидкість прийняття рішень визначає ефективність оборони.

Завдяки поєднанню геоінформаційної платформи з динамічними алгоритмами розбиття, досягається можливість майже в реальному часі перераховувати оптимальні зони відповідальності, оперативно переорієнтовуючи ППО у разі зміни конфігурації повітряної загрози. Цей підхід закладає основу для побудови адаптивної оборонної системи, яка здатна не лише реагувати на факти ворожих дій, але й діяти на випередження.

Було здійснено порівняльний аналіз системи «Світло» з низкою міжнародних рішень, таких як OnSolve MIR3 та AlertHub Pro. Такий аналіз дав змогу визначити особливості та потенційні переваги запропонованого програмного рішення в контексті її адаптації до специфіки кризових і воєнно-техногенних загроз на території України. Завдяки вузькій спеціалізації та гнучкій архітектурі система дозволяє враховувати не лише координати зафіксованих подій, але й динаміку їх розвитку, що створює передумови для більш своєчасного інформування відповідальних операторів.

Таким чином, система альтернативного оповіщення «Світло» може розглядатися як інструмент інформування та аналітичної підтримки прийняття рішень у кризових умовах. Використання геоінформаційних інструментів у поєднанні з алгоритмами динамічного розбиття території демонструє можливість формування програмних компонентів, здатних забезпечувати зворотний зв'язок, аналіз просторових даних і підготовку інформаційної основи для подальшого реагування.

4.2. Система аналізу та прийняття рішень про оперативні заходи протидії поширенню інфекційних захворювань

Огляд прикладної проблеми. Наприкінці 2019 року поширення вірусу SARS-CoV-2 спричинило масштабний спалах COVID-19, який швидко набув глобального характеру. Уже на початку 2020 року ситуацію було визнано

надзвичайною у сфері охорони здоров'я міжнародного рівня, а згодом — пандемією. Темпи зростання кількості інфікованих у різних країнах суттєво відрізнялися та залежали від стану медичної системи, забезпеченості лікарень, наявності кваліфікованого персоналу, нормативної бази щодо запровадження обмежень та інших соціально-економічних чинників. Пандемія продемонструвала, що навіть розвинені держави не завжди здатні оперативно реагувати на кризу такого масштабу.

Попри зменшення гостроти ситуації, ризик поширення інфекції зберігається. Водночас накопичені статистичні дані дозволяють будувати математичні моделі для прогнозування динаміки захворюваності як на локальному, так і на глобальному рівнях. Практика показала, що організаційні заходи боротьби з інфекцією за своєю результативністю не поступаються медичним втручанням. Швидке зростання кількості хворих у великих містах, дефіцит медичного обладнання та перевантаження лікарень актуалізували потребу в прогнозуванні чисельності інфікованих на певний період для прийняття управлінських рішень. Математичне моделювання покликане не лише оцінювати швидкість і фази поширення захворювання, а й аналізувати ефективність обмежувальних заходів, враховуючи їх соціальні та економічні наслідки.

Сьогодні розроблено значну кількість моделей оцінювання епідемічної динаміки, більшість із яких орієнтована на конкретні регіони. Відмінності у віковій структурі населення, поширеності хронічних захворювань і рівні соціальної активності суттєво впливають на перебіг епідемії. Оскільки зростання захворюваності зазвичай має експоненційний характер, навіть незначні демографічні особливості з часом призводять до істотних розбіжностей у статистичних показниках. Прикладом є різна динаміка захворюваності серед людей похилого віку в окремих країнах, що спричинило перевантаження систем охорони здоров'я.

Хоча за останні роки накопичено великий обсяг даних щодо перебігу пандемії, механізми впливу інфекції на організм залишаються складними для

повного опису. Найчастіше для моделювання застосовують моделі типу SIR та їх модифікації. Водночас обмеженість достовірних статистичних даних може призводити до накопичення похибок і спотворення результатів прогнозування. Крім того, класичні моделі враховують обмежену кількість параметрів і не завжди дозволяють оцінити вплив різноманітних карантинних заходів.

Динаміка поширення інфекції істотно різниться між країнами й визначається густотою населення, мобільністю, станом медичної системи та характером обмежень. Хоча моделі SIS, SIR і SEIR широко застосовуються у практиці, їх традиційне використання без урахування просторової структури населення знижує точність прогнозів. Такі підходи не повною мірою враховують транспортні потоки, концентрацію населення та регіональні особливості.

Інфекційні захворювання вже понад століття становлять серйозну загрозу для громадського здоров'я, спричиняючи як локальні спалахи, так і глобальні пандемії. Історичні дані свідчать, що кількість інфікованих у різні періоди коливалася від тисяч до сотень мільйонів, а смертність — від сотень до десятків мільйонів осіб. Попри досягнення сучасної медицини, точне прогнозування пікових значень захворюваності та часової еволюції епідемій залишається ключовим елементом ефективного управління кризовими ситуаціями. Саме якісні прогнози дозволяють своєчасно впроваджувати обґрунтовані управлінські рішення та мінімізувати соціально-економічні втрати.

Пандемія COVID-19, що розвивалася хвилями різних штамів вірусу, ще раз підтвердила важливість надійного епідеміологічного моделювання. За кілька років кількість інфікованих у світі обчислювалася сотнями мільйонів, а рівень смертності сягнув багатомільйонних значень. У цих умовах математичні моделі стали важливим інструментом підтримки управлінських рішень.

Основою більшості підходів залишаються компартментальні моделі, які поділяють населення на групи сприйнятливих, інфікованих та імунних осіб. Вони забезпечують аналітичну структуру для оцінювання порогів епідемії та масштабів спалахів. Однак їх класична форма передбачає однорідне змішування населення і не враховує просторову та поведінкову неоднорідність. Тому сучасні дослідження розширюють базові схеми, додаючи часові затримки, вікову структуру, мобільність населення та локальні обмежувальні заходи.

Розвиток математичного моделювання епідемій має глибоке історичне підґрунтя. Перші систематичні підходи були запропоновані на початку ХХ століття, коли було сформульовано моделі міграції носіїв інфекції та механізми її передачі. Надалі ці ідеї були доповнені врахуванням демографічних і вікових характеристик популяції, що дозволило створити більш реалістичні диференціальні системи для опису динаміки поширення інфекцій.

У подальших дослідженнях було запропоновано модифікації SEI та SEIR, які деталізують структуру популяції за стадіями перебігу хвороби. Такі моделі дозволяють враховувати вікові відмінності, нерівномірність ризиків зараження та різний ступінь тяжкості перебігу захворювання. Окремі роботи також включали постановку задач оптимального керування з метою мінімізації поширення інфекції.

Подальший розвиток моделей пов'язаний із використанням систем автономних диференціальних рівнянь та збільшенням кількості змінних для врахування територіальних особливостей. Це дозволило частково подолати обмеження класичних схем щодо просторової однорідності та відобразити специфіку розподілу інфекції в межах різних регіонів.

Сучасні дослідження дедалі більше орієнтовані на врахування просторової структури епідемічних процесів. Просторово-розподілені моделі дозволяють аналізувати вплив густоти населення, мобільності, доступності медичних послуг і локальних обмежень на швидкість передачі інфекції. На

відміну від гомогенних підходів, такі моделі дають змогу враховувати регіональні особливості та застосовувати цільові протиепідемічні заходи, включаючи локальні карантинні обмеження чи кампанії вакцинації.

У межах цього напрямку сформувався окремий клас регіонально-орієнтованих моделей, де аналіз проводиться на рівні адміністративних одиниць із використанням наявних статистичних та організаційних даних. Подібні підходи поєднують динаміку інфекції з плануванням ресурсів системи охорони здоров'я та дозволяють виконувати сценарний аналіз навіть за обмеженої повноти інформації. Це особливо важливо для територій із нерівномірною інфраструктурою спостереження.

Окрім просторового аспекту, сучасні моделі активно інтегрують часову мінливість параметрів, стохастичні ефекти та мережеві механізми передачі інфекції. Впровадження методів машинного навчання та байєсівських підходів забезпечує роботу з невизначеністю даних і дозволяє уточнювати параметри в режимі реального часу. Гібридні моделі, що поєднують компартментальні схеми з регресійними або виживальними підходами, підвищують глибину аналізу, особливо за наявності вікових і контактних даних.

Практична цінність епідемічного моделювання визначається якістю калібрування та перевірки моделей на основі реальної статистики. Порівняння прогнозів із фактичними даними вимагає постійного вдосконалення алгоритмів і обережності в інтерпретації результатів. Включення інформації про мобільність населення, соціальні контакти та зміну поведінки підвищує адаптивність моделей до нових умов, зокрема появи нових штамів вірусу.

Попри ускладнення інструментарію, прості парсимоніозні моделі залишаються корисними для аналізу окремих фаз епідемії та оцінки впливу обмежувальних заходів. У багатьох випадках математичні підходи демонструють вищу прогностичну здатність порівняно з експертними оцінками, що підкреслює їх значення у прийнятті рішень на основі даних.

Загалом сучасний розвиток епідемічного моделювання характеризується переходом до просторово-структурованих, поведінково

орієнтованих та адаптивних до даних моделей. Інтеграція механізмів просторового поширення з класичними компартментальними підходами та використання методів асиміляції даних формують нову парадигму прогнозування та контролю епідемій.

У цьому розділі дисертаційного дослідження запропоновано просторово-регіональне розширення моделей SIS/SIR/SEIR, адаптоване для аналізу епідемічної ситуації в Дніпропетровській та Харківській областях. Розроблений підхід поєднує класичні математичні засади з урахуванням мобільності населення, регіональних особливостей і поведінкових чинників, що дозволяє підвищити точність прогнозів і забезпечити практичну підтримку управлінських рішень на регіональному рівні.

Предметна постановка задачі. Більшість математичних підходів, що застосовуються для аналізу та прогнозування перебігу інфекційних захворювань, спрямовані на оцінювання чисельності трьох основних груп населення: сприйнятливих, інфікованих і вилучених осіб. До категорії «вилучених» зазвичай відносять як тих, хто одужав і сформував імунітет, так і померлих. Теоретичною основою таких моделей є системи звичайних диференціальних рівнянь, для розв'язання яких використовуються аналітичні або чисельні методи. Проте подібний підхід переважно базується на так званому точковому припущенні, коли все населення розглядається як зосереджене в одній умовній просторовій точці без урахування географічного розподілу. Оскільки передача інфекції відбувається через контакти між людьми, просторовий фактор істотно впливає на результати моделювання і не може ігноруватися. Важливу роль відіграють також транспортні переміщення між регіонами, які сприяють поширенню інфекції та формують зони підвищеного ризику, зокрема у громадському транспорті. У класичних точкових моделях ці аспекти не враховуються.

Традиційна модель SIR поділяє популяцію на три групи:

– S (Susceptible) — особи, сприйнятливі до інфекції та здатні заразитися в разі контакту з інфікованими;

- I (Infected) — інфіковані, які можуть передавати збудника іншим;
- R (Removed) — вилучені з процесу поширення, тобто ті, хто одужав із формуванням імунітету або помер.

Одним із базових припущень моделі SIR є відсутність повторного зараження. Однак практичні спостереження підтверджують можливість реінфекції, у тому числі тим самим штамом вірусу, що суперечить класичному формулюванню. У зв'язку з цим сучасні модифікації моделі враховують імовірність повторного інфікування та застосовують підходи із часовими затримками для відображення періоду формування імунної відповіді.

Незважаючи на поширеність і зручність застосування, модель SIR має низку обмежень. Серед них — нехтування просторовою структурою, відсутність урахування чинників інтенсивного поширення захворювання (зокрема транспортних потоків і густоти населення), обмеженість часових горизонтів прогнозування та жорстка трьохстанова структура переходів. Модель може адекватно описувати короткочасну динаміку епідемії, однак її точність знижується при довгострокових прогнозах.

У загальному вигляді модель SIR описується системою звичайних диференціальних рівнянь.

$$\begin{cases} \dot{S} = -\beta \cdot \frac{S \cdot I}{N} \\ \dot{I} = \beta \cdot \frac{S \cdot I}{N} - \gamma \cdot I \\ \dot{R} = \gamma \cdot I \end{cases} \quad (4.1)$$

де:

S — кількість сприйнятливих осіб;

I — кількість інфікованих осіб;

R — кількість вилучених осіб (тих, хто одужав з імунітетом або помер);

$N = S + I + R$ — загальна чисельність популяції (метапопуляції);

β — коефіцієнт інфікування, який відображає ймовірність того, що сприйнятлива особа заразиться при контакті з інфікованою;

γ — коефіцієнт вибування, який визначає середню ймовірність того, що інфікована особа одужає або помре.

Модель SEIR описується наступною системою звичайних диференціальних рівнянь:

$$\begin{cases} \dot{S} = -\beta \cdot \frac{S \cdot I}{N} \\ \dot{E} = \beta \cdot \frac{S \cdot I}{N} - \alpha \cdot E \\ \dot{I} = \alpha \cdot E - \gamma \cdot I \\ \dot{R} = \gamma \cdot I \end{cases} \quad (4.2)$$

де:

E — кількість осіб у періоді інкубації (піддані, але ще не заразні);

α — швидкість переходу від стану підданості до інфікованості, яка є оберненою до середньої тривалості інкубаційного періоду;

інші позначення ($S, U, R, \beta, \gamma, N$) відповідають моделі SIR.

В основу роботи покладено уявлення про популяцію як сукупність індивідів, яку можна умовно структурувати за кількома ознаками. Зокрема, виділялися такі категорії:

- неінфіковані (здорові) особи;
- інфіковані;
- ті, що вибули з аналізу (померлі або такі, що одужали);
- мешканці окремих територіальних секторів;
- розподіл за статтю (чоловіки та жінки);
- вікові групи (до 15 років, 15–55 років, понад 55 років);
- медичні працівники та допоміжний персонал, залучені до лікування інфікованих.

У межах дослідження припускалося, що ймовірність повторного зараження є низькою. Хоча наявність стійкого імунітету після одужання

остаточно не доведена, статистичні дані свідчать, що випадки реінфекції або відсутні, або трапляються поодинокі. Це може пояснюватися як неповним курсом лікування, так і хибнонегативними результатами тестів на момент виписки пацієнтів із медичних закладів.

Математична постановка задачі. Сукупність інфікованих розглядається як обмежений набір підгруп, кожна з яких відрізняється рівнем чутливості до збудника та характерною тривалістю перебігу хвороби. Для кожної такої підгрупи ймовірність інфікування задається окремими параметрами, що відображають її специфіку.

За цих припущень компартментну модель розвитку інфекційного процесу можна подати в аналітичному вигляді таким чином:

$$\left\{ \begin{array}{l} \dot{S} = -\sum_{i=1}^p \beta_i \cdot \frac{1}{N} \cdot S \cdot \sum_{j=1}^p I_j \\ \dot{I}_1 = \beta_1 \cdot \frac{1}{N} \cdot S \cdot \sum_{j=1}^p I_j - \gamma_1 \cdot I_1 \\ \dots \\ \dot{I}_p = \beta_p \cdot \frac{1}{N} \cdot S \cdot \sum_{j=1}^p I_j - \gamma_p \cdot I_p \\ \dot{R} = \sum_{i=1}^p \gamma_i \cdot I_i \end{array} \right. \quad (4.3)$$

У моделі (4.3) використовуються такі позначення:

S — кількість осіб у субпопуляції, які не інфіковані (сприйнятливі);

I_i — кількість інфікованих осіб в i -тій субпопуляції, яка характеризується окремою ймовірністю зараження та тривалістю перебігу хвороби;

p — кількість субпопуляцій інфікованих осіб, кожна з яких має унікальні значення ймовірності інфікування та тривалості захворювання;

N — загальна кількість осіб у мета популяції. При цьому вважається,

що $N = S + \sum_{i=1}^p I_i + R$;

β_i — ймовірність зараження для осіб з i -тої субпопуляції;

γ_i — ймовірність одужання для осіб з i -тої субпопуляції;

R — кількість осіб, які вибули з популяції (тобто тих, що одужали або померли).

Важливо зазначити, що принципова відмінність запропонованої моделі полягає у деталізованому розподілі інфікованих осіб на окремі підгрупи з істотно різними характеристиками перебігу хвороби. Частина пацієнтів може мати короткий інкубаційний період і, відповідно, не справляти значного впливу на загальну епідемічну динаміку. Натомість інші, з огляду на певні індивідуальні чинники, здатні залишатися заразними тривалий час, що суттєво позначається як на кількісних показниках розвитку інфекції, так і на темпах її поширення в межах метапопуляції.

Математичне моделювання розповсюдження COVID-2019 на території Дніпропетровської області. На рисунку 4.9 наведено результати чисельного моделювання поширення COVID-2019 для Дніпропетровської області України за період 2020–2024 років. Рисунок 4.9-а демонструє динаміку поширення COVID-19 відповідно до моделі SIS, а рисунок 4.9-б - відповідні результати для моделі SIR.

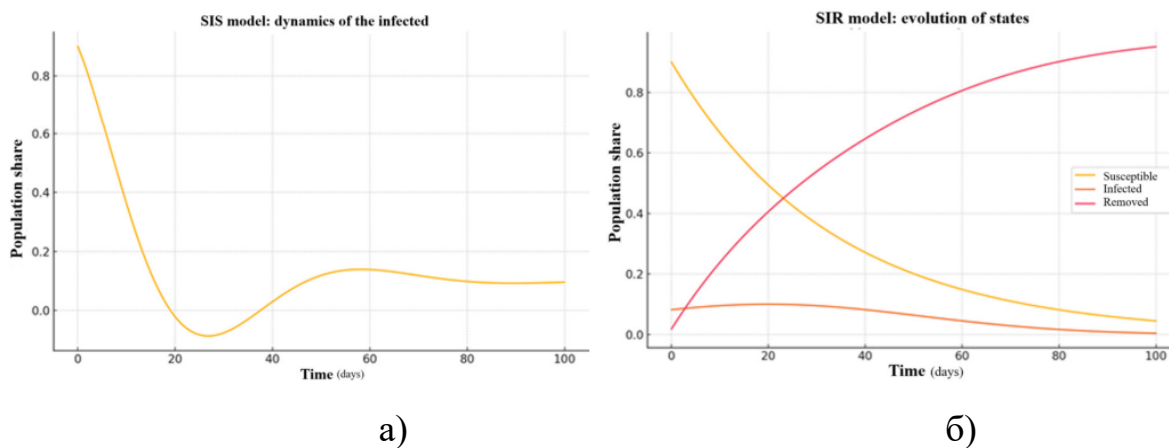


Рисунок 4.9 – Результати чисельної реалізації моделей: а) SIS, б) SIR для Дніпропетровської області за період 2020–2024 років

Результати, представлені на рисунку 4.9, свідчать про те, що обидві моделі адекватно відтворюють динаміку захворюваності в абсолютних величинах. Вісь ординат відображає частку інфікованого населення відносно

загальної чисельності населення регіону (3,1 мільйона осіб). Максимальна відносна похибка для всіх моделей не перевищує 8%, що вважається прийнятним рівнем для епідеміологічного прогнозування. Відтак запропонована модель була використана для прогнозування кількості інфікованих осіб і оцінки потреб у ліжко-місцях у регіоні.

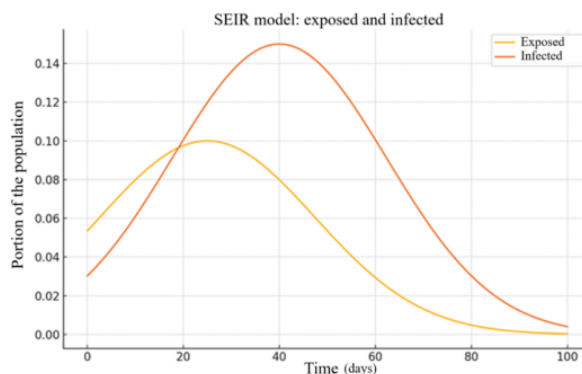


Рисунок 4.10 – Результати чисельної реалізації моделі SEIR для Дніпропетровської області за період 2020–2024 років

Рисунок 4.10 ілюструє результати застосування моделі SEIR на тому ж наборі даних, що і для чисельного експерименту, наведеного на рис. 4.9.

Як видно з рисунку 4.10, модель SEIR дає більш детальне представлення процесу розповсюдження інфекції ніж SIS та SIR моделі. Втім, її довгострокова точність обмежена через труднощі ідентифікації параметрів, що не повністю враховують такі фактори, як нереєстровані випадки (наприклад, особи, які лікуються вдома без ПЛР тестування). Це сприяє збільшенню рівня непередбачуваності захворювання, що, у свою чергу, впливає на коректність та ефективність обмежувальних заходів, а також зумовлює інші негативні наслідки. Незважаючи на ці обмеження, модель SEIR достатньо точно прогнозує кількість позитивних ПЛР-тестів і необхідну кількість ліжко-місць, з максимальною відносною похибкою 5,6%, що є кращим за відповідний показник моделей SIS та SIR, похибка яких, як було показано вище, становить близько 8%.

У таблиці 4.3 наведено результати аналізу похибки використаних моделей для Дніпропетровської області.

Відносна похибка результатів застосування SIS, SIR та SEIR моделей
на прикладі Дніпропетровської області (2020–2024 роки)

Період	SIS	SIR	SEIR	Період	SIS	SIR	SEIR
04.2020-06.2020	6,54%	2,88%	4,69%	10.2022-12.2022	6,93%	6,42%	2,6%
07.2020-09.2020	6,92%	3,60%	3,79%	01.2023-03.2023	6,15%	7,89%	3,51%
10.2020-12.2020	4,16%	7,34%	2,02%	04.2023-06.2023	2,16%	4,13%	4,2%
01.2021-03.2021	3,57%	7,49%	3,61%	07.2023-09.2023	5,87%	5,09%	3,27%
04.2021-06.2021	7,64%	6,23%	3,51%	10.2023-12.2023	4,59%	5,66%	2,94%
07.2021-09.2021	2,71%	4,74%	3,74%	01.2024-03.2024	5,32%	6%	5,6%
10.2021-12.2021	4,94%	2,54%	3,94%	04.2024-06.2024	6,22%	2,92%	2,06%
02.2022-03.2022	2,67%	7,07%	2,89%	07.2024-09.2024	2,81%	3,81%	4,68%
04.2022-06.2022	4,38%	2,89%	3,43%	10.2024-12.2024	5,78%	5,86%	4,3%
02.2022-03.2022	2,67%	7,07%	2,89%	07.2024-09.2024	2,81%	3,81%	4,68%
04.2022-06.2022	4,38%	2,89%	3,43%	10.2024-12.2024	5,78%	5,86%	4,3%
07.2022-09.2022	4,82%	5,36%	3,22%				
Максимальна відносна похибка	7,64%	7,49%	4,69%	Максимальна відносна похибка	6,93%	7,89%	5,6%

Запропоновано та обґрунтовано нову компартментальну математичну модель для прогнозування поширення COVID-19 та подібних інфекцій. Проведено аналіз наукових джерел, визначено переваги й обмеження існуючих підходів. Модель побудована на основі систем автономних диференціальних рівнянь і описує часову зміну чисельності основних груп населення — сприйнятливих, інфікованих, тих, хто одужує, та інших класів залежно від конфігурації. Її параметри можуть оперативнo налаштовуватися за офіційними статистичними даними, що забезпечує адаптивність до різних регіональних умов.

Перевірку адекватності здійснено на прикладі Дніпропетровської області. Побудовано шість прогнозів на 90 днів, при цьому середня похибка не перевищила 8,6%, що підтверджує надійність підходу. Результати моделювання дали змогу сформулювати практичні рекомендації щодо впровадження карантинних заходів і регулювання мобільності населення, а також оцінювати вплив різних сценаріїв втручання на розвиток епідемії.

Отримані висновки поглиблюють розуміння механізмів поширення інфекцій та ролі ключових параметрів, зокрема швидкості передачі, тривалості інкубаційного періоду й ефективності обмежень. Практична цінність моделей полягає у можливості їх застосування для реального прогнозування та управління епідемічними процесами. Вони можуть бути інтегровані в системи моніторингу й слугувати інструментом науково обґрунтованого планування протиепідемічних заходів на регіональному рівні.

Отримані результати дозволяють отримувати прогнозні значення чисельних показників поширення інфекції для підтримки прийняття оперативних рішень, у тому числі із застосуванням автоматизованих і роботизованих рішень.

4.3 Висновки до розділу 4

У четвертому розділі дисертаційної роботи досліджено можливості застосування цифрового двійника як програмного середовища для

моделювання, прогнозування та аналізу кризових процесів, пов'язаних із наслідками техногенного походження. На відміну від попередніх розділів, у яких основну увагу приділено архітектурі програмного комплексу та методам адаптивного навчання роботехнічних систем, у цьому розділі розглянуто прикладні сценарії використання запропонованих інформаційних технологій для збору даних, формування цифрової моделі ситуації, оцінювання стану середовища та підтримки прийняття рішень.

Показано, що цифровий двійник у межах програмного комплексу може виконувати не лише функцію симуляційного середовища для навчання роботехнічних систем, але й роль інтеграційного ядра для опрацювання різнорідних даних, побудови сценаріїв розвитку подій та аналізу можливих наслідків. Такий підхід дозволяє поєднати моделювання середовища, обробку сенсорної та користувачької інформації, прогнозування динаміки процесів і формування рекомендацій для подальшого реагування.

Розроблено прототип автоматизованої системи збору та аналізу інформації «Світло», призначеної для приймання, структурування, візуалізації та передачі повідомлень про виявлені повітряні об'єкти. У межах дисертаційного дослідження ця система розглядається як приклад програмного модуля ситуаційної обізнаності, що може використовуватися для наповнення цифрового двійника актуальними даними про стан середовища. Запропонована архітектура системи демонструє можливість побудови модульних програмних рішень, орієнтованих на роботу з розподіленими джерелами даних, операторськими інтерфейсами та механізмами підтримки прийняття рішень.

У розділі також розглянуто задачу динамічного розбиття території на зони відповідальності як приклад просторового моделювання та розподілу ресурсів у кризових умовах. Запропонована модель може бути використана для формування навчальних сценаріїв, у яких роботехнічні системи або автоматизовані засоби реагування діють у середовищі з обмеженими

ресурсами, змінною конфігурацією зон відповідальності та необхідністю адаптації до поточного стану ситуації.

Окремо проаналізовано систему моделювання поширення інфекційних захворювань на основі моделей SIS, SIR та SEIR. У контексті дисертаційної роботи ці моделі розглядаються не як самостійне медичне дослідження, а як приклад формального опису складного динамічного процесу в цифровому середовищі. Отримані результати демонструють можливість використання математичного та імітаційного моделювання для прогнозування розвитку кризових явищ, оцінювання можливих наслідків та підтримки вибору оперативних заходів реагування.

Таким чином, у четвертому розділі показано, що запропонований програмний комплекс може бути розширений за рахунок модулів збору даних, ситуаційного аналізу, просторового моделювання, прогнозування та підтримки прийняття рішень. Результати розділу підтверджують доцільність використання цифрового двійника як основи для формування навчальних і аналітичних сценаріїв підготовки роботехнічних систем до роботи в умовах техногенних надзвичайних ситуацій. Це посилює практичну спрямованість дисертаційного дослідження та узгоджує розглянуті прикладні задачі з предметною областю спеціальності 121 «Інженерія програмного забезпечення».

ВИСНОВКИ

У дисертаційній роботі вирішено актуальну науково-прикладну задачу, що полягає у проектуванні, розробці та тестуванні програмного комплексу навчання роботехнічних систем, призначених для ліквідації наслідків техногенного походження. Актуальність проведеного дослідження зумовлена зростанням кількості техногенних аварій, воєнних руйнувань критичної інфраструктури, підвищенням рівня ризику для персоналу аварійно-рятувальних служб та необхідністю підвищення автономності роботехнічних засобів.

У процесі виконання дисертаційного дослідження отримано такі основні наукові та практичні результати.

1. Проведено системний аналіз сучасного стану роботехнічних систем, що застосовуються для ліквідації наслідків техногенних аварій та надзвичайних ситуацій. Здійснено класифікацію роботехнічних платформ за конструктивними особливостями, рівнем автономності та функціональним призначенням. Встановлено, що ефективність застосування роботів у складних аварійних сценаріях значною мірою залежить від наявності спеціалізованих програмних засобів навчання та адаптації до змін середовища.

2. Виконано порівняльний аналіз існуючих підходів до автоматизації процесу навчання роботехнічних систем, включаючи класичні алгоритмічні методи, симуляційно-орієнтовані підходи, методи машинного навчання та навчання з підкріпленням. Визначено їх переваги, обмеження та придатність до використання у задачах ліквідації наслідків техногенного походження. Обґрунтовано необхідність створення інтегрованого програмного комплексу, який поєднує засоби моделювання, обробки сенсорних даних, планування руху та інтелектуального прийняття рішень.

3. Сформульовано постановку технічної задачі розробки програмного комплексу навчання роботехнічних систем. Визначено вимоги до архітектури програмного забезпечення, модульної структури, масштабованості,

можливості інтеграції різнорідних сенсорних систем та підтримки алгоритмів адаптивного керування. Запропоновано концептуальну модель програмного комплексу, яка забезпечує взаємодію між модулями симуляції, аналізу даних, формування навчальних сценаріїв та оцінювання результатів навчання.

4. Спроектовано архітектуру програмного комплексу навчання роботехнічних систем ліквідації наслідків техногенного походження. Архітектурне рішення базується на модульному принципі побудови з чітким розмежуванням функціональних компонентів: моделювання середовища, обробки сенсорної інформації, планування траєкторій руху, прийняття рішень та координації дій. Забезпечено можливість розширення функціональності комплексу шляхом підключення нових алгоритмів і типів роботехнічних платформ.

5. Проаналізовано сучасні програмні інструменти та технології, придатні для реалізації комплексу, та сформовано обґрунтований набір технологічних рішень. Реалізація програмного комплексу виконана з використанням сучасних мов програмування та інструментів розробки, що забезпечують кросплатформеність, адаптивність інтерфейсу та ефективну обробку даних у режимі реального часу. Особливу увагу приділено масштабованості системи та можливості її інтеграції з різними типами роботехнічних пристроїв.

6. Розроблено програмний комплекс навчання роботехнічних систем ліквідації наслідків техногенного походження, який забезпечує:

- моделювання сценаріїв надзвичайних ситуацій техногенного характеру;
- формування навчальних середовищ із можливістю варіювання параметрів;
- обробку та аналіз сенсорної інформації;
- реалізацію алгоритмів автономної навігації та адаптивного керування;
- оцінювання ефективності навчання роботехнічних систем.

7. Проведено тестування розробленого програмного комплексу, що включало функціональну перевірку, аналіз продуктивності та оцінювання стійкості роботи системи при різних сценаріях навантаження. Отримані

результати підтвердили коректність реалізації архітектурних рішень, стабільність роботи комплексу та його придатність до використання у задачах підготовки роботехнічних систем до роботи в умовах техногенних аварій.

Наукова новизна одержаних результатів полягає у тому, що:

- вперше запропоновано комплексний підхід до побудови програмного комплексу навчання роботехнічних систем ліквідації наслідків техногенного походження, який інтегрує засоби симуляційного моделювання, алгоритми адаптивного керування та механізми формування навчальних сценаріїв;

- удосконалено архітектурну модель програмного забезпечення для навчання роботехнічних систем шляхом впровадження модульної структури з підтримкою масштабування та інтеграції різнорідних алгоритмів навігації і прийняття рішень;

- набули подальшого розвитку підходи до організації процесу навчання роботів у симуляційному середовищі з можливістю перенесення отриманих результатів у реальні експлуатаційні умови.

Практичне значення отриманих результатів полягає в тому, що розроблений програмний комплекс може бути використаний для підготовки роботехнічних систем аварійно-рятувальних служб, підрозділів цивільного захисту, а також організацій, що здійснюють роботи з ліквідації наслідків техногенних аварій і воєнних руйнувань. Використання комплексу дозволяє скоротити витрати на натурні експерименти, підвищити безпеку випробувань, зменшити час розробки алгоритмів керування та підвищити рівень автономності роботехнічних систем.

Результати дисертаційної роботи апробовано на міжнародних науково-практичних конференціях та опубліковано у фахових наукових виданнях, що підтверджує їх наукову значущість та відповідність сучасному рівню досліджень у галузі інженерії програмного забезпечення.

Таким чином, поставлену мету дисертаційного дослідження досягнуто, усі сформульовані наукові задачі розв'язано. Розроблений програмний комплекс навчання роботехнічних систем ліквідації наслідків техногенного

походження є цілісним, функціонально завершеним інструментом, який поєднує сучасні методи моделювання, машинного навчання та програмної інженерії й може слугувати основою для подальшого розвитку інтелектуальних роботизованих систем реагування на надзвичайні ситуації.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Артюх О. М., Дударенко О. В., Кузьмін В. В., Сосик А. Ю. Основи мехатроніки: навч. посіб. Запоріжжя : НУ «Запорізька політехніка», 2021. 372 с.
2. Барабан С. В., Белзецький Р. С., Арсенюк І. Р. Комп'ютерна інженерія та основи роботехніки. Вінниця : ВНТУ, 2024. 155 с.
3. Бондаренко М. Ф., Шевченко В. В. Інтелектуальні роботехнічні системи. Харків : ХНУРЕ, 2017. 248 с.
4. Гладуш В. П., Семенюк М. П. Основи автоматизованого управління технічними системами. Київ : НУБіП України, 2015. 286 с.
5. Гуржій А. М. Основи автоматики та роботехніки : навч. посіб. Київ : НПУ ім. М. П. Драгоманова, 2018. 256 с.
6. Зеркалов Д. В. Техногенна безпека. Київ : Основа, 2009. 337 с.
7. Кодекс цивільного захисту України : Закон України від 02.10.2012 № 5403-VI.
8. Козак Л. М. Програмування роботехнічних систем. Львів : Львівська політехніка, 2018. 210 с.
9. Кузьменко О. Ю. Системи дистанційного керування мобільними роботами. Харків : ХНУРЕ, 2019. 198 с.
10. Латишева О. В., Степаненко І. В. Природно-техногенна безпека та екологічна безпека. Київ : Центр учбової літератури, 2016. 304 с.
11. Ловейкін В. С., Ромасевич Ю. О., Човнюк Ю. В. Мехатроніка : навч. посіб. Київ : Компринт, 2012. 357 с.
12. Морзе Н. В., Варченко-Троценко Л. О., Гладун М. А. Основи роботехніки. Кам'янець-Подільський : ПП Буйницький О. А., 2016. 184 с.
13. Ніколайчук В. М. Основи роботехніки. Рівне : НУВГП, 2008. 76 с.
14. Остапченко К. Б. Роботехнічні системи та комплекси: алгоритмізація і верифікація управління. Київ : КПІ ім. Ігоря Сікорського, 2020. 214 с.

15. Остапченко К. Б., Ткач М. М. Роботехнічні системи та комплекси. Курсовий проєкт. Київ : КПІ ім. Ігоря Сікорського, 2022. 156 с.
16. Павленко І. І., Мажара В. А. Роботизовані технологічні комплекси. Кропивницький : КНТУ, 2010. 392 с.
17. Петренко В. О. Інтелектуальні системи керування роботами. Одеса : ОНПУ, 2019. 230 с.
18. Савчук В. І. Мобільні роботизовані платформи для спеціальних умов експлуатації. Київ : НАУ, 2021. 198 с.
19. Струтинська О. В. Теоретико-методичні засади підготовки майбутніх учителів до навчання освітньої роботехніки. Київ : НПУ ім. М. П. Драгоманова, 2020. 356 с.
20. Трифонова О. М., Хомутенко М. В., Садовий М. І. Автоматизовані системи програмних навчальних комплексів. Кропивницький : ПП «Ексклюзив-Систем», 2019. 120 с.
21. Цвіркун Л. І., Грулер Г. Роботехніка та мехатроніка. Дніпро : НГУ, 2017. 224 с.
22. Шевчук Р. О. Роботехнічні системи в умовах надзвичайних ситуацій. Львів : ЛДУБЖД, 2023. 180 с.
23. Білокінь О. В. Безпілотні наземні роботизовані комплекси: сучасний стан і перспективи. Київ, 2022. 160 с.
24. Лисенко С. П., Яковенко І. В. Моделювання роботехнічних систем у MATLAB/Simulink. Київ : КПІ ім. Ігоря Сікорського, 2020. 210 с.
25. Abbeel P., Ng A. Y. Apprenticeship learning via inverse reinforcement learning // *Proceedings of ICML*. 2004. DOI: 10.1145/1015330.1015430.
26. Bailey T., Durrant-Whyte H. Simultaneous localization and mapping (SLAM): Part II // *IEEE Robotics & Automation Magazine*. 2006. Vol. 13, No. 3. P. 108–117. DOI: 10.1109/MRA.2006.1678144.
27. Bicchi A., Kumar V. Robotic grasping and contact: A review // *Proceedings of ICRA*. 2000.

28. Bohg J. et al. Data-driven grasp synthesis — A survey // *IEEE Transactions on Robotics*. 2014. Vol. 30, No. 2. DOI: 10.1109/TRO.2013.2289018.
29. Brock O. et al. (eds.) *The DARPA Robotics Challenge Finals*. Cham : Springer, 2018.
30. Cadena C. et al. Past, Present, and Future of SLAM // *IEEE Transactions on Robotics*. 2016. Vol. 32, No. 6. DOI: 10.1109/TRO.2016.2624754.
31. Corke P. *Robotics, Vision and Control*. Cham : Springer, 2017.
32. Craig J. J. *Introduction to Robotics: Mechanics and Control*. 3rd ed. Upper Saddle River : Pearson, 2004.
33. Durrant-Whyte H., Bailey T. Simultaneous localization and mapping: Part I // *IEEE Robotics & Automation Magazine*. 2006. Vol. 13, No. 2. DOI: 10.1109/MRA.2006.1638022.
34. Erez T., Tassa Y., Todorov E. Simulation tools for model-based robotics // *Proceedings of ICRA*. 2015. DOI: 10.1109/ICRA.2015.7139912.
35. Fox D., Burgard W., Thrun S. The dynamic window approach to collision avoidance // *IEEE Robotics & Automation Magazine*. 1997. Vol. 4, No. 1. DOI: 10.1109/100.580977.
36. Goodfellow I., Bengio Y. *Deep Learning*. Cambridge : MIT Press, 2016.
37. Hollinger G., Singh S. Multi-robot coordination with periodic connectivity // *Proceedings of ICRA*. 2010.
38. Howard A. et al. Mobile robot localization by Monte Carlo localization // *Proceedings of ICRA*. 1999.
39. Khatib O. Real-time obstacle avoidance for manipulators and mobile robots // *International Journal of Robotics Research*. 1986. Vol. 5, No. 1. DOI: 10.1177/027836498600500106.
40. Koenig N., Howard A. Design and use paradigms for Gazebo // *Proceedings of IROS*. 2004. DOI: 10.1109/IROS.2004.1389727.

41. Kober J., Bagnell J. A., Peters J. Reinforcement learning in robotics: A survey // *International Journal of Robotics Research*. 2013. Vol. 32, No. 11. P. 1238–1274. DOI: 10.1177/0278364913495721.
42. Krotkov E. et al. The DARPA Robotics Challenge Finals // *Journal of Field Robotics*. 2017. Vol. 34, No. 2. DOI: 10.1002/rob.21683.
43. LaValle S. *Planning Algorithms*. Cambridge:Cambridge University Press,2006.
44. Levine S. et al. End-to-end training of deep visuomotor policies // *Journal of Machine Learning Research*. 2016. Vol. 17.
45. Mnih V. et al. Human-level control through deep reinforcement learning // *Nature*. 2015. Vol. 518. P. 529–533. DOI: 10.1038/nature14236.
46. Murphy R. R. *Disaster Robotics*. Cambridge : MIT Press, 2014. DOI: 10.7551/mitpress/9407.001.0001.
47. Quigley M. et al. ROS: an open-source Robot Operating System // *ICRA Workshop*. 2009.
48. Siciliano B., Khatib O. (eds.) *Springer Handbook of Robotics*. 2nd ed. Cham : Springer, 2016.
49. Sünderhauf N. et al. The limits and potentials of deep learning for robotics // *International Journal of Robotics Research*. 2018. DOI: 10.1177/0278364918770733.
50. Sutton R. S., Barto A. G. *Reinforcement Learning: An Introduction*. 2nd ed. Cambridge : MIT Press, 2018.
51. Thrun S., Burgard W., Fox D. *Probabilistic Robotics*. Cambridge:MITPress,2005.
52. Tobin J. et al. Domain randomization for transferring deep neural networks // *Proceedings of IROS*. 2017. DOI: 10.1109/IROS.2017.8202133.
53. Todorov E., Erez T., Tassa Y. MuJoCo: A physics engine for model-based control // *Proceedings of IROS*. 2012. DOI: 10.1109/IROS.2012.6386109.
54. Tzafestas S. *Introduction to Mobile Robot Control*. Oxford : Elsevier, 2014.

55. Wurman P. R., D'Andrea R., Mountz M. Coordinating hundreds of cooperative autonomous vehicles // *AI Magazine*. 2008. Vol. 29, No. 1. DOI: 10.1609/aimag.v29i1.2082.
56. Yanco H. A., Drury J. L. A taxonomy for human-robot interaction // *AAAI Fall Symposium*. 2002.
57. Yokokohji Y., Yoshikawa T. Bilateral control of master–slave manipulators // *IEEE Transactions on Robotics and Automation*. 1994. Vol. 10, No. 5. DOI: 10.1109/70.326576.
58. Zhang J., Singh S. LOAM: Lidar Odometry and Mapping in Real-time // *Robotics: Science and Systems*. 2014.
59. Zou D., Tan P. CoSLAM: Collaborative visual SLAM in dynamic environments // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2013. Vol. 35, No. 2. DOI: 10.1109/TPAMI.2012.59.
60. Casper J., Murphy R. Human-robot interactions during disaster response // *IEEE Transactions on Systems, Man, and Cybernetics*. 2003.
61. Nourbakhsh I. et al. Human-robot teaming for search and rescue // *IEEE Pervasive Computing*. 2005.
62. Tadokoro S. (ed.) *Rescue Robotics*. Berlin : Springer, 2009.
63. Murphy R. R., Burke J. Up from the Rubble: Lessons learned about HRI from search and rescue // *Proceedings of HRI*. 2005.
64. Choset H. et al. *Principles of Robot Motion*. Cambridge : MIT Press, 2005.
65. Siciliano B., Sciavicco L., Villani L., Oriolo G. *Robotics: Modelling, Planning and Control*. London : Springer, 2009.
66. Lynch K. M., Park F. C. *Modern Robotics*. Cambridge : Cambridge University Press, 2017.
67. Spong M. W., Hutchinson S., Vidyasagar M. *Robot Modeling and Control*. Hoboken : Wiley, 2006.
68. Siegwart R., Nourbakhsh I. R., Scaramuzza D. *Introduction to Autonomous Mobile Robots*. Cambridge : MIT Press, 2011.

69. Dudek G., Jenkin M. *Computational Principles of Mobile Robotics*. Cambridge : Cambridge University Press, 2010.
70. Borenstein J., Everett H. R., Feng L. *Where Am I? Sensors and Methods for Mobile Robot Positioning*. Ann Arbor : University of Michigan Press, 1996.

ДОДАТКИ

Додаток А

Список публікацій здобувача за темою дисертації

Статті у наукових фахових виданнях України:

1. Kuzenkov O., Dubovyk V. Software implementation of the COVID-2019 spread simulation algorithm using the theory of optimal set partitioning. *Актуальні проблеми автоматизації та інформаційних технологій*, т. 24, Дніпро, 2020, с. 96-103. DOI: <https://dx.doi.org/10.15421/432011> (особистий внесок Дубовика В.В.: участь у програмній реалізації алгоритму моделювання поширення COVID-2019, підготовка вхідних даних для чисельного експерименту, аналіз та інтерпретація отриманих результатів; внесок Кузенкова О.О.: постановка наукової задачі, формалізація математичної моделі, методичне керівництво та узагальнення результатів).

2. Kiseleva E., Kuzenkov O., Dubovyk V. Software implementation of the algorithm for solving the dynamic problem of optimal placement of fire groups. *Актуальні проблеми автоматизації та інформаційних технологій*, т. 26, Дніпро, 2022, с. 137–145. DOI: <https://dx.doi.org/10.15421/432217> (особистий внесок Дубовика В.В.: участь у програмній реалізації алгоритму розв'язання динамічної задачі оптимального розміщення вогневих груп, проведення чисельних експериментів, аналіз та інтерпретація результатів; внесок Кісельової О.М.: постановка оптимізаційної задачі, обґрунтування математичної моделі та методологічний супровід; внесок Кузенкова О.О.: наукове консультування, участь у формалізації алгоритмічного підходу та узагальнення результатів).

3. Kuzenkov O., Dubovyk V. Design and implementation of a modular software architecture for dynamic image segmentation in robotic vision systems. *Актуальні проблеми автоматизації та інформаційних технологій*, т. 28, Дніпро, 2024, с. 318–326. DOI: <http://dx.doi.org/10.15421/432430> (особистий внесок Дубовика В.В.: участь у проектуванні модульної архітектури програмної системи динамічної сегментації зображень, визначення структури програмних модулів, аналіз підходів до комп'ютерного зору та розпізнавання образів, аналіз отриманих результатів;

внесок Кузенкова О.О.: постановка наукової задачі, методичне консультування, коригування архітектурних рішень та узагальнення результатів).

4. Kuzenkov O., Dubovyk V. Development of a dynamic partitioning model and a software system for image segmentation in the implementation of computer vision for robotic systems. *Сучасні проблеми моделювання*, т. 27, Мелітополь, 2025, с. 123–135. DOI: <http://dx.doi.org/10.33842/2313-125X-2025-19-123-135>.

(особистий внесок Дубовика В.В.: розроблення динамічної моделі розбиття та програмної системи для сегментації зображень, постановка чисельного експерименту, аналіз результатів роботи програмної реалізації; внесок Кузенкова О.О.: постановка наукового завдання, методологічне консультування, уточнення математичної моделі та узагальнення результатів).

5. Kuzenkov O.O., Dubovyk V.V. Development of a software system for training robotic systems for mitigation of technogenic disaster consequences.

Актуальні проблеми автоматизації та інформаційних технологій, т. 29, Дніпро, 2025, с. 425–436. DOI: <http://dx.doi.org/10.15421/432538>

(особистий внесок Дубовика В.В.: обґрунтування концепції програмного комплексу навчання роботехнічних систем, визначення його архітектури й функціональних модулів, участь у розробленні програмної реалізації, тестуванні та аналізі результатів функціонування системи; внесок Кузенкова О.О.: постановка наукового завдання, наукове консультування, коригування архітектурних рішень і узагальнення результатів).

Наукові праці, які засвідчують апробацію матеріалів дисертації:

6. Кузенков О.О., Дубовик В.В. Особливості проєктування програмного комплексу навчання роботехнічних систем. *Нотатки сучасної науки: матеріали II Міжнародної науково-практичної конференції «Актуальні питання сучасної науки: історія, теорія, практика»*, № 33, Україна – США, 2026, с. 52–53. URL: <https://www.newroute.org.ua/rep/>;

<https://mega.nz/file/Sn5TSlwY#KnGnQoMU-pwC3RKfsf5X4M7Fv58fNZxfKTmhh0dIB2s>

(особистий внесок Дубовика В.В.: аналіз вимог до програмного комплексу навчання роботехнічних систем, обґрунтування модульної архітектури, визначення основних функціональних компонентів системи, формування підходу до використання симуляційного середовища та цифрового двійника

для підготовки роботехнічних систем до роботи в умовах техногенного походження, аналіз та узагальнення отриманих результатів; внесок Кузенкова О.О.: постановка наукового завдання, наукове консультування, уточнення архітектурної концепції та узагальнення отриманих результатів).

7. Кузенков О.О., Дубовик В.В. Програмна реалізація комплексу навчання роботехнічних систем. *Advanced top technology: матеріали II Міжнародної науково-практичної конференції «Новітні технології сучасного суспільства»*, № 18, Україна – Польща, 2026, с. 49–50. URL: <https://www.newroute.org.ua/rep/>; <https://mega.nz/file/m2YUkSoA#2jD2hS8rOf4DV8p7Jx-iH2IAtZXwxNjQrrItiFT-WEk> (особистий внесок Дубовика В.В.: розроблення програмної реалізації комплексу навчання роботехнічних систем, опис структури основних програмних модулів, обґрунтування логіки взаємодії компонентів, підготовка матеріалів щодо експериментальної реалізації, аналіз результатів функціонування системи; внесок Кузенкова О.О.: постановка наукового завдання, наукове консультування, коригування архітектурних рішень і узагальнення отриманих результатів).

8. Кузенков О.О., Дубовик В.В. Навчання систем керування роботехнічних систем, призначених для ліквідації наслідків техногенного походження. Соціально-гуманітарний вісник: збірник наукових праць за матеріалами IV Міжнародної науково-практичної конференції «Сучасні тенденції соціально-гуманітарного розвитку суспільства», вип. 66, Україна – Німеччина, 2026, с. 77–78. URL: <https://www.newroute.org.ua/rep/>; https://mega.nz/file/vuA2XAKI#Jvbzo3YHOdtQMy4sRCmskeyt_rYr5fOZwx_Lz4nXn6k (особистий внесок Дубовика В.В.: аналіз підходів до навчання систем керування роботехнічних систем, обґрунтування застосування методів машинного навчання та адаптивного керування для задач ліквідації наслідків техногенного походження, формування логіки використання цифрового двійника та симуляційного середовища для підготовки роботехнічних систем, аналіз та узагальнення отриманих результатів; внесок Кузенкова О.О.: постановка наукового завдання, наукове консультування, уточнення методологічних положень і узагальнення отриманих результатів).

Програмний код системи візуальної ідентифікації

```

import argparse
import json
import os
from dataclasses import dataclass
from typing import List, Optional, Tuple
import cv2
import numpy as np
@dataclass
class MatchResult:
    label: str
    score: float
    good_matches: int
    total_matches: int
    homography_ok: bool
    bbox: Optional[List[List[int]]] # 4 points (x,y) if available
    ref_path: str
def imread_color(path: str) -> np.ndarray:
    img = cv2.imread(path, cv2.IMREAD_COLOR)
    if img is None:
        raise FileNotFoundError(f"Не можу прочитати зображення: {path}")
    return img
def compute_orb(img_bgr: np.ndarray, nfeatures: int = 2000):
    gray = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2GRAY)
    orb = cv2.ORB_create(nfeatures=nfeatures)
    kps, des = orb.detectAndCompute(gray, None)
    return kps, des
def ensure_dir(path: str) -> None:
    os.makedirs(path, exist_ok=True)
def load_gallery(gallery_dir: str) -> List[Tuple[str, str, np.ndarray]]:
    """
    Returns list of (label, filepath, image_bgr)
    """
    if not os.path.isdir(gallery_dir):
        raise NotADirectoryError(f"Папка gallery не знайдена: {gallery_dir}")

    exts = {".jpg", ".jpeg", ".png", ".bmp", ".tif", ".tiff", ".webp"}
    items = []
    for fn in sorted(os.listdir(gallery_dir)):
        p = os.path.join(gallery_dir, fn)
        if not os.path.isfile(p):
            continue
        _, ext = os.path.splitext(fn.lower())
        if ext not in exts:
            continue
        label = os.path.splitext(fn)[0]
        img = imread_color(p)
        items.append((label, p, img))

```

```

if not items:
    raise ValueError(f"У папці {gallery_dir} немає зображень підтримуваних форматів.")
return items
def match_one(ref_img: np.ndarray, qry_img: np.ndarray) -> Tuple[float, int, int, List[cv2.DMatch],
Optional[np.ndarray]]:
    """
Returns:
score (0..1 approx),
good_matches_count,
total_matches_count,
good_matches list,
homography matrix H if computed else None
    """
    ref_kp, ref_des = compute_orb(ref_img)
    qry_kp, qry_des = compute_orb(qry_img)
    if ref_des is None or qry_des is None or len(ref_kp) < 8 or len(qry_kp) < 8:
        return 0.0, 0, 0, [], None
    bf = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=False)
    # KNN matching + Lowe ratio
    knn = bf.knnMatch(ref_des, qry_des, k=2)
    good = []
    for m_n in knn:
        if len(m_n) != 2:
            continue
        m, n = m_n
        if m.distance < 0.75 * n.distance:
            good.append(m)
    total = len(knn)
    good_n = len(good)
    # Score: частка "good" + легка нормалізація (щоб не завищувати при малих числах)
    score = 0.0
    if total > 0:
        ratio = good_n / total
        score = float(ratio) * float(min(1.0, good_n / 60.0)) # насичення після ~60 good matches
    H = None
    if good_n >= 12:
        ref_pts = np.float32([ref_kp[m.queryIdx].pt for m in good]).reshape(-1, 1, 2)
        qry_pts = np.float32([qry_kp[m.trainIdx].pt for m in good]).reshape(-1, 1, 2)
        H, mask = cv2.findHomography(ref_pts, qry_pts, cv2.RANSAC, 5.0)
        # Якщо гомографія "погана", не використовуємо
        if H is not None and mask is not None:
            inliers = int(mask.sum())
            # додаткова перевірка якості
            if inliers < max(10, int(0.35 * good_n)):
                H = None
    return score, good_n, total, good, H
def draw_match_vis(ref_img: np.ndarray, qry_img: np.ndarray, ref_label: str,
    good_matches: List[cv2.DMatch]) -> np.ndarray:
    ref_kp, ref_des = compute_orb(ref_img)
    qry_kp, qry_des = compute_orb(qry_img)
    # Якщо дескриптори не порахувались — просто склейка
    if ref_des is None or qry_des is None:

```

```

h = max(ref_img.shape[0], qry_img.shape[0])
w = ref_img.shape[1] + qry_img.shape[1]
canvas = np.zeros((h, w, 3), dtype=np.uint8)
canvas[:ref_img.shape[0], :ref_img.shape[1]] = ref_img
canvas[:qry_img.shape[0], ref_img.shape[1]:] = qry_img
cv2.putText(canvas, f"Best: {ref_label} (no descriptors)", (10, 30),
            cv2.FONT_HERSHEY_SIMPLEX, 1.0, (0, 255, 255), 2, cv2.LINE_AA)
return canvas
# Візуалізуємо лише топ-N good matches (щоб картинка не була "каша")
good_sorted = sorted(good_matches, key=lambda m: m.distance)[:60]
vis = cv2.drawMatches(
    ref_img, ref_kp, qry_img, qry_kp, good_sorted, None,
    flags=cv2.DrawMatchesFlags_NOT_DRAW_SINGLE_POINTS
)
cv2.putText(vis, f"Best match: {ref_label} | good={len(good_matches)}", (10, 30),
            cv2.FONT_HERSHEY_SIMPLEX, 1.0, (0, 255, 255), 2, cv2.LINE_AA)
return vis
def compute_bbox_from_homography(ref_img: np.ndarray, H: np.ndarray) ->
Optional[List[List[int]]]:
    if H is None:
        return None
    h, w = ref_img.shape[:2]
    corners = np.float32([[0, 0], [w - 1, 0], [w - 1, h - 1], [0, h - 1]]).reshape(-1, 1, 2)
    proj = cv2.perspectiveTransform(corners, H).reshape(-1, 2)
    proj_int = proj.astype(int).tolist()
    return proj_int
def annotate_detection(qry_img: np.ndarray, bbox: List[List[int]], label: str, score: float) -> np.ndarray:
    out = qry_img.copy()
    pts = np.array(bbox, dtype=np.int32).reshape((-1, 1, 2))
    cv2.polylines(out, [pts], isClosed=True, color=(0, 255, 0), thickness=3)
    x, y = bbox[0]
    cv2.putText(out, f"{label} ({{score:.3f}})", (max(5, x), max(25, y - 10)),
                cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 255, 0), 2, cv2.LINE_AA)
    return out
def main():
    ap = argparse.ArgumentParser(description="Система візуальної ідентифікації (ORB +
BFMatcher).")
    ap.add_argument("--gallery", required=True, help="Папка з еталонними зображеннями (база).")
    ap.add_argument("--query", required=True, help="Зображення-запит, яке треба ідентифікувати.")
    ap.add_argument("--out", default="out", help="Папка для результатів.")
    ap.add_argument("--min_score", type=float, default=0.03, help="Мінімальний score для прийняття
збігу.")
    args = ap.parse_args()
    ensure_dir(args.out)
    qry_img = imread_color(args.query)
    gallery = load_gallery(args.gallery)
    best: Optional[MatchResult] = None
    best_good = []
    best_H = None
    best_ref_img = None
    for label, path, ref_img in gallery:
        score, good_n, total_n, good_matches, H = match_one(ref_img, qry_img)

```

```

res = MatchResult(
    label=label,
    score=score,
    good_matches=good_n,
    total_matches=total_n,
    homography_ok=(H is not None),
    bbox=None,
    ref_path=path
)
if best is None or res.score > best.score:
    best = res
    best_good = good_matches
    best_H = H
    best_ref_img = ref_img
if best is None or best_ref_img is None:
    raise RuntimeError("Не вдалося отримати жодного результату (перевір gallery та query).")
# BBox (якщо гомографія вдалась)
bbox = compute_bbox_from_homography(best_ref_img, best_H) if best_H is not None else None
best.bbox = bbox
# Візуалізація match
vis = draw_match_vis(best_ref_img, qry_img, best.label, best_good)
vis_path = os.path.join(args.out, "match_vis.png")
cv2.imwrite(vis_path, vis)
detected_path = None
if bbox is not None:
    det = annotate_detection(qry_img, bbox, best.label, best.score)
    detected_path = os.path.join(args.out, "detected.png")
    cv2.imwrite(detected_path, det)
status = "identified" if best.score >= args.min_score else "unknown"
result = {
    "status": status,
    "best_label": best.label if status == "identified" else None,
    "best_score": best.score,
    "good_matches": best.good_matches,
    "total_matches": best.total_matches,
    "homography_ok": best.homography_ok,
    "bbox": best.bbox,
    "reference_image": best.ref_path,
    "outputs": {
        "match_vis": vis_path,
        "detected": detected_path
    }
}
}
}
json_path = os.path.join(args.out, "result.json")
with open(json_path, "w", encoding="utf-8") as f:
    json.dump(result, f, ensure_ascii=False, indent=2)
print("=== VISUAL IDENTIFICATION RESULT ===")
print(json.dumps(result, ensure_ascii=False, indent=2))
if __name__ == "__main__":
    main()

```

Програмний код реалізації альтернативної системи оповіщення

```

import argparse
import datetime as dt
import json
import os
import platform
import socket
import subprocess
import sys
import threading
import time
from dataclasses import dataclass
from typing import Dict, List, Optional
@dataclass
class Alert:
    level: str # "INFO" | "WARN" | "CRITICAL"
    title: str # короткий заголовок
    message: str # текст
    source: str # звідки прийшло (trigger name)
    ts_utc: str # timestamp UTC ISO
    meta: Dict # додаткові поля
def utc_now_iso() -> str:
    return dt.datetime.utcnow().replace(microsecond=0).isoformat() + "Z"
# =====
# Канали оповіщення
# =====
class Notifier:
    def send(self, alert: Alert) -> None:
        raise NotImplementedError
class ConsoleNotifier(Notifier):
    def send(self, alert: Alert) -> None:
        print(f"[{alert.ts_utc}] {alert.level}: {alert.title} | {alert.message} (src={alert.source})")
class FileNotifier(Notifier):
    def __init__(self, path: str):
        self.path = path
        os.makedirs(os.path.dirname(path) or ".", exist_ok=True)
    def send(self, alert: Alert) -> None:
        line = json.dumps(alert.__dict__, ensure_ascii=False)
        with open(self.path, "a", encoding="utf-8") as f:
            f.write(line + "\n")
class BeepNotifier(Notifier):
    """
    Альтернативний “офлайн” канал: звуковий сигнал.
    Працює:
    - Windows: winsound
    - інше: BEL (\a) + спроба 'beep' якщо є
    """
    def __init__(self, enabled: bool = True):

```

```

self.enabled = enabled
def send(self, alert: Alert) -> None:
    if not self.enabled:
        return
    if alert.level not in ("WARN", "CRITICAL"):
        return

    try:
        if platform.system().lower().startswith("win"):
            import winsound # type: ignore
            freq = 880 if alert.level == "CRITICAL" else 660
            dur = 600 if alert.level == "CRITICAL" else 250
            winsound.Beep(freq, dur)
        else:
            # BEL
            sys.stdout.write("\a")
            sys.stdout.flush()
            # Якщо є система утиліта beep — спробуємо
            if alert.level == "CRITICAL":
                subprocess.run(["beep", "-f", "880", "-l", "400"], check=False)
            else:
                subprocess.run(["beep", "-f", "660", "-l", "200"], check=False)
    except Exception:
        # Нічого страшного: просто тихо, але система працює
        pass

class UDPBroadcastNotifier(Notifier):
    """
    Оповіщення по локальній мережі через UDP broadcast.
    Це “альтернативний” канал, який працює навіть без інтернету,
    якщо є локальна мережа/радіоміст/mesh.
    """
    def __init__(self, broadcast_ip: str, port: int):
        self.addr = (broadcast_ip, port)
        self.sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        self.sock.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)

    def send(self, alert: Alert) -> None:
        payload = json.dumps(alert.__dict__, ensure_ascii=False).encode("utf-8")
        self.sock.sendto(payload, self.addr)
# =====
# Джерела подій (тригери)
# =====

class Trigger:
    name: str
    def poll(self) -> Optional[Alert]:
        raise NotImplementedError

class HeartbeatTrigger(Trigger):
    def __init__(self, interval_sec: int = 30):
        self.name = "heartbeat"
        self.interval_sec = interval_sec
        self._next = time.time()
    def poll(self) -> Optional[Alert]:

```

```

now = time.time()
if now < self._next:
    return None
self._next = now + self.interval_sec
return Alert(
    level="INFO",
    title="Система активна",
    message="Heartbeat: система оповіщення працює.",
    source=self.name,
    ts_utc=utc_now_iso(),
    meta={}
)
class FilePanicTrigger(Trigger):
    """
    Ручна тривога: якщо з'явився файл ALERT.txt — відправляємо CRITICAL.
    Вміст файлу = текст повідомлення. Після відправки файл перейменовується.
    """
    def __init__(self, watch_path: str = "ALERT.txt"):
        self.name = "file_panic"
        self.watch_path = watch_path
    def poll(self) -> Optional[Alert]:
        if not os.path.exists(self.watch_path):
            return None
        try:
            with open(self.watch_path, "r", encoding="utf-8") as f:
                msg = f.read().strip() or "Ручна тривога (файл створено без тексту)."
        except Exception:
            msg = "Ручна тривога (не вдалося прочитати файл)."
        # "поглинаємо" подію: перейменувати файл, щоб не спамити повторно
        ts = dt.datetime.utcnow().strftime("%Y%m%d_%H%M%S")
        new_name = f"ALERT_SENT_{ts}.txt"
        try:
            os.replace(self.watch_path, new_name)
        except Exception:
            # якщо не вдалось — спробуємо просто видалити
            try:
                os.remove(self.watch_path)
            except Exception:
                pass
        return Alert(
            level="CRITICAL",
            title="Тривога!",
            message=msg,
            source=self.name,
            ts_utc=utc_now_iso(),
            meta={"file": self.watch_path}
        )
class PingHostTrigger(Trigger):
    """
    Контроль доступності вузла.
    Якщо хост перестав відповідати — WARN/CRITICAL (налаштовується).
    """

```

```

def __init__(self, host: str, interval_sec: int = 5, fail_threshold: int = 3, critical: bool = True):
    self.name = "ping_watch"
    self.host = host
    self.interval_sec = interval_sec
    self.fail_threshold = fail_threshold
    self.critical = critical
    self._next = time.time()
    self._fails = 0
    self._last_state = "unknown" # "up" | "down" | "unknown"

def _ping(self) -> bool:
    sysname = platform.system().lower()
    if sysname.startswith("win"):
        cmd = ["ping", "-n", "1", "-w", "800", self.host]
    else:
        cmd = ["ping", "-c", "1", "-W", "1", self.host]
    try:
        r = subprocess.run(cmd, stdout=subprocess.DEVNULL, stderr=subprocess.DEVNULL)
        return r.returncode == 0
    except Exception:
        return False

def poll(self) -> Optional[Alert]:
    now = time.time()
    if now < self._next:
        return None
    self._next = now + self.interval_sec

    ok = self._ping()
    if ok:
        self._fails = 0
        if self._last_state == "down":
            self._last_state = "up"
            return Alert(
                level="INFO",
                title="Вузол відновився",
                message=f"Хост {self.host} знову доступний.",
                source=self.name,
                ts_utc=utc_now_iso(),
                meta={"host": self.host}
            )
        self._last_state = "up"
        return None
    # fail
    self._fails += 1
    if self._fails < self.fail_threshold:
        return None
    # перейшли поріг
    self._last_state = "down"
    level = "CRITICAL" if self.critical else "WARN"
    return Alert(
        level=level,
        title="Втрата зв'язку з вузлом",

```

```

    message=f"Хост {self.host} не відповідає (fails={self._fails}).",
    source=self.name,
    ts_utc=utc_now_iso(),
    meta={"host": self.host, "fails": self._fails, "threshold": self.fail_threshold}
)
# =====
# Ядро системи оповіщення
# =====
class AlertSystem:
    def __init__(self, triggers: List[Trigger], notifiers: List[Notifier], poll_hz: float = 5.0):
        self.triggers = triggers
        self.notifiers = notifiers
        self.poll_sleep = 1.0 / max(0.2, poll_hz)
        self._stop = threading.Event()
    def stop(self):
        self._stop.set()
    def dispatch(self, alert: Alert):
        for n in self.notifiers:
            try:
                n.send(alert)
            except Exception as e:
                # не "валимо" систему через один канал
                print(f"[WARN] notifier failed: {type(n).__name__}: {e}", file=sys.stderr)
    def run(self):
        while not self._stop.is_set():
            for t in self.triggers:
                try:
                    alert = t.poll()
                except Exception as e:
                    alert = Alert(
                        level="WARN",
                        title="Помилка тригера",
                        message=f"Trigger {getattr(t, 'name', 'unknown')} error: {e}",
                        source=getattr(t, "name", "unknown"),
                        ts_utc=utc_now_iso(),
                        meta={}
                    )
                if alert is not None:
                    self.dispatch(alert)
            time.sleep(self.poll_sleep)
    def main():
        ap = argparse.ArgumentParser(description="Альтернативна система оповіщення (offline-
first + LAN broadcast).")
        ap.add_argument("--log", default="out/alerts.log.jsonl", help="Шлях до лог-файлу (jsonl).")
        ap.add_argument("--heartbeat", type=int, default=30, help="Heartbeat інтервал (сек). 0 =
вимкнути.")
        ap.add_argument("--panic-file", default="ALERT.txt", help="Файл для ручної тривоги.")
        ap.add_argument("--udp-broadcast", default=None, help="Broadcast IP (наприклад
255.255.255.255).")
        ap.add_argument("--udp-port", type=int, default=5005, help="UDP порт для broadcast.")
        ap.add_argument("--watch-host", default=None, help="Хост/IP для контролю доступності
(ping).")

```

```

ap.add_argument("--ping-interval", type=int, default=5, help="Ping інтервал (сек).")
ap.add_argument("--ping-threshold", type=int, default=3, help="Скільки невдалих ping
підряд = тривога.")
ap.add_argument("--ping-critical", action="store_true", help="Якщо задано — рівень
CRITICAL, інакше WARN.")
ap.add_argument("--no-beep", action="store_true", help="Вимкнути звукові сигнали.")
args = ap.parse_args()
# Triggers
triggers: List[Trigger] = []
if args.heartbeat and args.heartbeat > 0:
    triggers.append(HeartbeatTrigger(interval_sec=args.heartbeat))
triggers.append(FilePanicTrigger(watch_path=args.panic_file))
if args.watch_host:
    triggers.append(PingHostTrigger(
        host=args.watch_host,
        interval_sec=args.ping_interval,
        fail_threshold=args.ping_threshold,
        critical=bool(args.ping_critical)
    ))
# Notifiers
notifiers: List[Notifier] = [
    ConsoleNotifier(),
    FileNotifier(args.log),
    BeepNotifier(enabled=not args.no_beep)
]
if args.udp_broadcast:
    notifiers.append(UDPBroadcastNotifier(args.udp_broadcast, args.udp_port))
print("=== ALT ALERT SYSTEM STARTED ===")
print(f"Triggers: {[getattr(t, 'name', type(t).__name__) for t in triggers]}")
print(f"Notifiers: {[type(n).__name__ for n in notifiers]}")
print("Stop: Ctrl+C")
system = AlertSystem(triggers=triggers, notifiers=notifiers, poll_hz=5.0)
try:
    system.run()
except KeyboardInterrupt:
    print("\nStopping...")
    system.stop()
if __name__ == "__main__":
    main()

```

**Програмний код розв’язання задачі розбиття території на зони
відповідальності медичних закладів**

```

import argparse
import os
from dataclasses import dataclass
from typing import Optional, Tuple
import numpy as np
import pandas as pd
import geopandas as gpd
from shapely.geometry import Point, Polygon, box
from shapely.ops import unary_union
import matplotlib.pyplot as plt
@dataclass
class Config:
    facilities_csv: str
    boundary_path: Optional[str]
    out_dir: str
    cell_size_m: float
    weight_mode: str # "none" | "capacity_sqrt"
    min_capacity: float
    crs_work: str = "EPSG:3857" # метрична проекція для відстаней
def ensure_dir(path: str) -> None:
    os.makedirs(path, exist_ok=True)
def read_facilities(csv_path: str) -> gpd.GeoDataFrame:
    df = pd.read_csv(csv_path)
    required = {"id", "name", "lat", "lon"}
    missing = required - set(df.columns)
    if missing:
        raise ValueError(f'facilities.csv missing columns: {sorted(missing)}')
    if "capacity" not in df.columns:
        df["capacity"] = 1.0
    df["capacity"] = pd.to_numeric(df["capacity"], errors="coerce").fillna(1.0)
    gdf = gpd.GeoDataFrame(
        df,
        geometry=gpd.points_from_xy(df["lon"], df["lat"]),
        crs="EPSG:4326",
    )
    return gdf
def read_boundary(boundary_path: Optional[str], facilities_wgs84: gpd.GeoDataFrame) ->
gpd.GeoDataFrame:
    if boundary_path:
        b = gpd.read_file(boundary_path)
        if b.empty:
            raise ValueError("boundary file is empty")
        # беремо першу геометрію
        b = b.iloc[[0]].copy()
        if b.crs is None:
            # якщо CRS не задано — припустимо WGS84

```

```

    b.set_crs("EPSG:4326", inplace=True)
    # якщо геометрія не полігон — пробуємо унію
    geom = unary_union(list(b.geometry))
    b = gpd.GeoDataFrame({"name": ["boundary"]}, geometry=[geom], crs=b.crs)
    return b
# Авто-межа: прямокутник навколо закладів з невеликим полем
fac = facilities_wgs84.to_crs("EPSG:4326")
minx, miny, maxx, maxy = fac.total_bounds
pad_x = (maxx - minx) * 0.35 if (maxx - minx) > 0 else 0.02
pad_y = (maxy - miny) * 0.35 if (maxy - miny) > 0 else 0.02
bbox = box(minx - pad_x, miny - pad_y, maxx + pad_x, maxy + pad_y)
return gpd.GeoDataFrame({"name": ["auto_boundary"]}, geometry=[bbox], crs="EPSG:4326")
def build_grid_within(boundary_m: Polygon, cell: float) -> gpd.GeoDataFrame:
    minx, miny, maxx, maxy = boundary_m.bounds
    xs = np.arange(minx, maxx + cell, cell)
    ys = np.arange(miny, maxy + cell, cell)
    cells = []
    centroids = []
    for x in xs[:-1]:
        for y in ys[:-1]:
            c = box(x, y, x + cell, y + cell)
            if not c.intersects(boundary_m):
                continue
            inter = c.intersection(boundary_m)
            if inter.is_empty:
                continue
            cells.append(inter)
            centroids.append(inter.centroid)
    gdf = gpd.GeoDataFrame({"cell_id": np.arange(len(cells))}, geometry=cells,
crs="EPSG:3857")
    gdf["centroid"] = centroids
    return gdf
def weighted_distance_matrix(centroids_xy: np.ndarray, facilities_xy: np.ndarray, capacities:
np.ndarray,
    mode: str, min_capacity: float) -> np.ndarray:
    """
    centroids_xy: (N,2), facilities_xy: (M,2)
    returns: (N,M) weighted distances
    """
    # Евклідова відстань (планарна)
    # (N,1,2) - (1,M,2) => (N,M,2)
    diff = centroids_xy[:, None, :] - facilities_xy[None, :, :]
    dist = np.sqrt(np.sum(diff * diff, axis=2)) # (N,M)
    if mode == "none":
        return dist
    # capacity_sqrt: dist / sqrt(capacity)
    cap = np.maximum(capacities.astype(float), float(min_capacity))
    w = np.sqrt(cap)[None, :] # (1,M)
    return dist / w
def assign_cells(grid: gpd.GeoDataFrame, facilities_m: gpd.GeoDataFrame, cfg: Config) ->
pd.DataFrame:
    centroids = np.array([(p.x, p.y) for p in grid["centroid"]], dtype=float)

```

```

fac_xy = np.array([(p.x, p.y) for p in facilities_m.geometry], dtype=float)
caps = facilities_m["capacity"].to_numpy(dtype=float)
wd = weighted_distance_matrix(centroids, fac_xy, caps, cfg.weight_mode, cfg.min_capacity)
best_idx = np.argmin(wd, axis=1)
assign = pd.DataFrame({
    "cell_id": grid["cell_id"].to_numpy(),
    "facility_idx": best_idx
})
return assign
def dissolve_zones(grid: gpd.GeoDataFrame, facilities_m: gpd.GeoDataFrame, assign:
pd.DataFrame) -> gpd.GeoDataFrame:
    g = grid.merge(assign, on="cell_id", how="left")
    # групуємо клітинки за закладом і об'єднуємо
    zones = []
    for idx, group in g.groupby("facility_idx"):
        geom = unary_union(list(group.geometry))
        row = facilities_m.iloc[int(idx)]
        zones.append({
            "facility_id": row["id"],
            "facility_name": row["name"],
            "capacity": float(row["capacity"]),
            "geometry": geom
        })
    zones_gdf = gpd.GeoDataFrame(zones, geometry="geometry", crs=grid.crs)
    return zones_gdf
def save_outputs(zones_m: gpd.GeoDataFrame, facilities_m: gpd.GeoDataFrame,
    boundary_m: gpd.GeoDataFrame, out_dir: str) -> None:
    ensure_dir(out_dir)
    # Експорт у WGS84 для сумісності
    zones_wgs = zones_m.to_crs("EPSG:4326")
    fac_wgs = facilities_m.to_crs("EPSG:4326")
    b_wgs = boundary_m.to_crs("EPSG:4326")
    zones_path = os.path.join(out_dir, "zones.geojson")
    fac_path = os.path.join(out_dir, "facilities.geojson")
    b_path = os.path.join(out_dir, "boundary.geojson")
    zones_wgs.to_file(zones_path, driver="GeoJSON")
    fac_wgs.to_file(fac_path, driver="GeoJSON")
    b_wgs.to_file(b_path, driver="GeoJSON")
    # Карта PNG (в робочій CRS, щоб виглядало "плоско")
    fig, ax = plt.subplots(figsize=(10, 10))
    boundary_m.boundary.plot(ax=ax, linewidth=2)
    zones_m.plot(ax=ax, alpha=0.5, edgecolor="black", linewidth=0.8)
    facilities_m.plot(ax=ax, marker="*", markersize=180)
    # підписи
    for _, r in facilities_m.iterrows():
        ax.annotate(
            text=f' {r["name"]} (cap={int(r["capacity"])})',
            xy=(r.geometry.x, r.geometry.y),
            xytext=(6, 6),
            textcoords="offset points",
            fontsize=9
        )
)

```

```

ax.set_title("Зони відповідальності медичних закладів (зважене розбиття території)")
ax.set_axis_off()
png_path = os.path.join(out_dir, "zones.png")
plt.tight_layout()
plt.savefig(png_path, dpi=200)
plt.close(fig)
print("Saved:")
print(" -", zones_path)
print(" -", fac_path)
print(" -", b_path)
print(" -", png_path)
def main():
    ap = argparse.ArgumentParser(
        description="Розбиття території на зони відповідальності медзакладів (зважене Voronoi-
        подібне через grid).")
    )
    ap.add_argument("--facilities", required=True, help="CSV з медзакладами:
    id,name,lat,lon,capacity")
    ap.add_argument("--boundary", default=None, help="GeoJSON/SHAPEFILE з межею
    території (полігон).")
    ap.add_argument("--out", default="out", help="Папка для результатів.")
    ap.add_argument("--cell", type=float, default=400.0, help="Розмір клітинки grid у метрах
    (точність/швидкість).")
    ap.add_argument("--weight", choices=["none", "capacity_sqrt"], default="capacity_sqrt",
        help="Як враховувати потужність: none або dist/sqrt(capacity).")
    ap.add_argument("--min-capacity", type=float, default=1.0, help="Мінімальна capacity, щоб
    уникнути ділення на 0.")
    args = ap.parse_args()
    cfg = Config(
        facilities_csv=args.facilities,
        boundary_path=args.boundary,
        out_dir=args.out,
        cell_size_m=float(args.cell),
        weight_mode=args.weight,
        min_capacity=float(args.min_capacity),
    )
    facilities = read_facilities(cfg.facilities_csv)
    boundary = read_boundary(cfg.boundary_path, facilities)
    # Приводимо до робочої CRS в метрах
    facilities_m = facilities.to_crs(cfg.crs_work)
    boundary_m = boundary.to_crs(cfg.crs_work)
    boundary_geom = boundary_m.geometry.iloc[0]
    if boundary_geom.is_empty:
        raise ValueError("Boundary geometry is empty after reprojection.")
    if boundary_geom.geom_type not in ("Polygon", "MultiPolygon"):
        raise ValueError(f"Boundary must be Polygon/MultiPolygon, got:
    {boundary_geom.geom_type}")

    # Побудова grid і призначення
    grid = build_grid_within(boundary_geom, cfg.cell_size_m)
    if grid.empty:
        raise RuntimeError("Grid is empty — збільш --cell або перевір boundary.")

```

```
assign = assign_cells(grid, facilities_m, cfg)
zones_m = dissolve_zones(grid, facilities_m, assign)
# Перетин з boundary (на випадок MultiPolygon/країв)
zones_m["geometry"] = zones_m.geometry.intersection(boundary_geom)
save_outputs(zones_m, facilities_m, boundary_m, cfg.out_dir)
if __name__ == "__main__":
    main()
```