МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЧЕРНІГІВСЬКА ПОЛІТЕХНІКА»

Кваліфікаційна наукова
праця на правах рукопису

СУСЛО МИКОЛА ЮРІЙОВИЧ

УДК 004.94:629.735.33:681.5(043.5)

ДИСЕРТАЦІЯ

МОДЕЛІ ТА МЕТОДИ НАПІВНАТУРНОГО МОДЕЛЮВАННЯ МУЛЬТИАГЕНТНИХ СИСТЕМ БПЛА

122 – Комп’ютерні науки

12 – Інформаційні технології
галузь знань

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів, текстів інших авторів супроводжується посиланнями на відповідне джерело.

_____ Сусло Микола Юрійович
підпис

Науковий керівник Казимир Володимир Вікторович
доктор технічних наук, професор

Чернігів – 2026

АНОТАЦІЯ

Сусло Микола Юрійович. Моделі та методи напівнатурного моделювання мультиагентних систем БПЛА. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 – «Комп'ютерні науки» (12 – «Інформаційні технології»). – НУ «Чернігівська політехніка», МОН України, Чернігів, 2026.

В ході дисертаційного дослідження виконано конкретне наукове завдання з розробки та експериментального дослідження моделей і методів напівнатурного моделювання, що забезпечують спільне використання апаратних і програмних компонентів БПЛА у складі мультиагентних систем при виконанні польотних завдань без використання GPS.

Об'єктом дослідження є процеси керування БПЛА-агентами по даним бортової системи позиціонування.

Предметом дослідження є методи моделювання та керування мультиагентними системами БПЛА та програмно-апаратні засоби їх реалізації.

Метою дисертаційної роботи є підвищення ефективності керування мультиагентними системами БПЛА при їх розгортанні для виконання місії за рахунок розроблення моделей і методів напівнатурного моделювання процесів керування, що забезпечують виконання маршрутних сценаріїв і стабілізацію положення БПЛА-агентів за відео даними в умовах відсутності GPS.

Основні результати дослідження та наукова новизна роботи полягають у розробленні моделей і методів напівнатурного моделювання мультиагентних систем БПЛА, формалізованого виконання маршрутних сценаріїв та локального керування положенням БПЛА за візуальним зміщенням.

Вперше:

- розроблено метод фазового адаптивного PID-керування положенням БПЛА за візуальним зміщенням, який, на відміну від існуючих, у кожному циклі керування роботою позиційного контуру формує по даним позиційної похибки

базову фазову конфігурацію керуючих впливів для польотного контролера із урахуванням заданих параметрів поточного етапу маршрутного сценарію, що забезпечує підвищення надійності проходження маршрутного сценарію до 92 відсотків.

Удосконалено:

- метод навігації БПЛА при його розгортанні в ході виконання місії, що базується на програмно визначених за допомогою середовища моделювання маршрутах переміщення, як послідовності просторових векторів, та моделі виконання маршрутного сценарію з визначенням зміщення позиції по відео зображенню у поєднанні з незалежним вертикальним контуром утримання висоти, що забезпечує просторову стабілізацію БПЛА без використання GPS із середньою абсолютною похибкою за кожною з горизонтальних осей до 8 пікселів у модельованих умовах.

Набули подальшого розвитку:

- моделі та програмно-апаратна архітектура напівнатурного моделювання мультиагентних систем БПЛА, яка, на відміну від існуючих, спільно використовує інтегроване віртуальне імітаційне середовище, апаратні засоби (Hardware-in-the-Loop, HIL) та програмні емулятори (Software-in-the-Loop, SIL) для відтворення процесів керуваннями БПЛА-агентами у складі рою, що забезпечує дослідження динаміки виконання польотних завдань у реальному часі.

Практичне значення отриманих результатів полягає у створенні програмних засобів напівнатурного моделювання для дослідження процесів функціонування та керування БПЛА у складі рою. Розроблене HIL/SIL середовище дає змогу спільно досліджувати програмні та апаратні реалізації польотних контролерів, моделі маршрутних сценаріїв, задавати та виконувати параметризовані послідовності маршрутних кроків, виконувати просторову стабілізацію, досліджувати локальне керування положенням в умовах відсутності глобальної координатної прив’язки та оцінювати розрахунки параметрів регулятора відповідно до поточних умов роботи

позиційного контуру.

Розроблені моделі маршрутних сценаріїв, методи просторової стабілізації та PID-керування роботою позиційного контуру отримали практичну реалізацію в діючих макетах квадрокоптерних дронів у вигляді алгоритмів та вбудованих програм на мові Python, які прошли верифікацію в ході модельних та натурних експериментів.

У **вступі** обґрунтовано актуальність дослідження моделей і методів напівнатурного моделювання та локального керування мультиагентними системами БПЛА, сформульовано мету і задачі дослідження, визначено його об’єкт, предмет і методи, наведено основні положення наукової новизни, практичне значення отриманих результатів та особистий внесок здобувача.

У першому розділі проведено аналіз сучасних моделей, методів і засобів моделювання та керування системами БПЛА. Розглянуто особливості мультиагентних систем БПЛА, підходи до програмного та напівнатурного моделювання, зокрема SIL- і HIL-конфігурації, методи маршрутного керування, позиціювання та стабілізації в умовах відсутності GPS, а також підходи до керування за візуальними даними й адаптації параметрів PID-регуляторів. За результатами аналізу визначено обмеження наявних підходів та сформульовано задачі дисертаційного дослідження.

У другому розділі розроблено архітектуру гібридного середовища напівнатурного моделювання мультиагентної системи БПЛА, що поєднує апаратну HIL- та програмні SIL-реалізації польотних контролерів у спільному модельованому середовищі. Визначено принципи взаємодії апаратних і програмних компонентів, організацію інформаційних потоків та механізми розмежування каналів окремих БПЛА-агентів. Експериментальне середовище реалізовано на основі Gazebo, ArduPilot, MAVLink/MAVROS і польотного контролера Pixhawk. Працездатність архітектури перевірено для конфігурації, що містила один HIL- та три SIL-агенти. Така комбінація апаратного Pixhawk і програмних екземплярів ArduPilot є також основою опублікованого дослідження з напівнатурного моделювання групи БПЛА.

У **третьому розділі** розроблено агентно-орієнтовану модель БПЛА та модель

маршрутного сценарію, в якій маршрут подано як послідовність параметризованих маршрутних кроків, а процес його виконання формалізовано на основі автомата станів. Розроблено метод просторової стабілізації БПЛА в умовах відсутності GPS, у якому горизонтальне керування здійснюється за відносним візуальним зміщенням у координатах зображення, а вертикальна складова забезпечується незалежним контуром підтримання висоти. Для виконання маршрутних кроків використано механізм віртуального дрейфу, який забезпечує зв'язок між маршрутним рівнем та контуром візуальної стабілізації. Працездатність маршрутного керування перевірено у SIL- та HIL-конфігураціях, а просторової стабілізації – у модельованих і натурних умовах. Агентно-орієнтована модель та перевірка маршрутного сценарію у програмній і напівнатурній конфігураціях описані також у відповідній публікації. Метод просторової стабілізації було перевірено як у Gazebo/ArduPilot, так і на реальному зразку.

У четвертому розділі розроблено фазовий метод адаптивного PID-керування положенням БПЛА за візуальним зміщенням. Метод передбачає формування та коригування коефіцієнтів PID-регулятора на кожному циклі керування з урахуванням поточної фази роботи позиційного контуру, характеристик позиційної похибки, нормованого показника якості візуального оцінювання, параметрів маршрутного кроку, поперечного відхилення та насичення керуючих каналів. Насичення враховується незалежно для двох координатних каналів.

Дисертація виконувалася в Національному університеті «Чернігівська політехніка». Результати наукових досліджень були впроваджені під час виконання за державним замовленням НДР «Мультиагентна система захисту об'єктів критичної інфраструктури на основі рою мультикоптерних дронів» (номер державної реєстрації: 0123U101819 (2023-2025)), у Державному науково-дослідному інституті випробувань і сертифікації озброєння та військової техніки для проведення натурних досліджень та відпрацювання програм і методик випробувань БПЛА, у навчальний процес з підготовки бакалаврів, магістрів та аспірантів НУ«Чернігівська політехніка» (Додаток Б).

Ключові слова: мультиагентна система БПЛА, напівнатурне моделювання, візуальне позиціонування, просторова стабілізація, маршрутний сценарій, GPS-denied, адаптивне PID-керування, Gazebo, ArduPilot, Pixhawk.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові праці з основними науковими результатами дисертації

1. Сусло М. Ю., Казимир В. В. Метод стабілізації квадрокоптера в заданій позиції з просторовим PID-регулюванням за відеоданими без GPS. Технічні науки та технології. 2026. № 2(44). С. 306–316. DOI: 10.25140/2411-5363-2026-2(44)-306-316. (Особистий внесок здобувача: розроблення та програмна реалізація методу просторової стабілізації БПЛА за візуальним зміщенням, інтеграція сигналів візуального зміщення в контур PID-керування, проведення модельованих і натурних експериментів, оброблення та аналіз отриманих результатів.)
2. Сусло М. Ю., Казимир В. В. Напівнатурне моделювання маршрутних сценаріїв для мультиагентної системи БПЛА. Наука і техніка сьогодні. 2026. № 5(59). С. 6049–6068. DOI: 10.52058/2786-6025-2026-5(59)-6049-6068. (Особистий внесок здобувача: розроблення агентно-орієнтованої моделі виконання маршрутних сценаріїв БПЛА, формалізація структури маршрутного сценарію та механізму його виконання, програмна реалізація маршрутного керування, проведення експериментів у SIL- та HIL-конфігураціях, аналіз отриманих результатів.)
3. Suslo M., Kazymyr V. Adaptive PID Control of UAV Position Using Visual Displacement. Computer Systems and Information Technologies. 2026. No. 3. P. 58–68. DOI: 10.31891/csit-2026-3-6 (Особистий внесок здобувача: розроблення фазового методу адаптивного налаштування коефіцієнтів PID-регулятора, програмна реалізація адаптивного регулятора, розроблення методики експериментального дослідження, проведення серії SIL-експериментів, статистичне опрацювання та порівняльний аналіз результатів.)

Наукові праці, які засвідчують апробацію матеріалів дисертації:

1. Сусло М. Ю., Олексієнко П. Д., Сігута А. В., Казимир В. В. Напів натурне моделювання процесів керування роєм дронів. Новітні технології сучасного суспільства (НТСС-2024) : V Міжнародна науково-практична конференція (м. Чернігів, 12 грудня 2024 р.) : тези доповідей. Чернігів : НУ «Чернігівська

політехніка», 2025. С. 183–184. (Особистий внесок здобувача: участь у розробленні та дослідженні засобів напівнатурного моделювання мультиагентної системи БПЛА, реалізація та експериментальна перевірка компонентів модельованого середовища.)

2. Suslo M., Kazymyr V., Oleksiienko P., Kagitin D. Semi-Physical Modeling of Drone Swarm Control Processes. 2025 15th International Conference on Advanced Computer Information Technologies (ACIT). IEEE, 2025. P. 102–108. DOI: 10.1109/ACIT65614.2025.11185750. (Особистий внесок здобувача: участь у розробленні та програмній реалізації гібридного HIL/SIL середовища моделювання, інтеграція програмних і апаратних компонентів системи, налаштування експериментального середовища та участь в аналізі результатів моделювання.)
3. Kahitin D., Kazymyr V., Suslo M., Yatchenko Y., Oleksiienko P. Drone Positioning in a Specified Location Using Visual Data Under GPS-denied Conditions. 2025 IEEE 13th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). Vol. 1. Gliwice, Poland, 2025. P. 318–323. DOI: 10.1109/IDAACS68557.2025.11322385. (Особистий внесок здобувача: налаштування камери та експериментального середовища, інтеграція модуля оцінювання візуального зміщення з програмним контуром керування БПЛА, оброблення отриманих компонент зміщення та участь в експериментальній перевірці системи.)

ABSTRACT

Mykola Yuriiovych Suslo. Models and Methods of Semi-Physical Modeling of Multi-Agent UAV Systems. – Qualifying scientific work submitted as a manuscript.

Dissertation submitted for the degree of Doctor of Philosophy in Specialty 122 – Computer Science (Field of Knowledge 12 – Information Technology). – Chernihiv Polytechnic National University, Ministry of Education and Science of Ukraine, Chernihiv, 2026.

The dissertation research addresses a specific scientific problem involving the development and experimental investigation of models and methods of semi-physical simulation that enable the joint use of hardware and software components of UAVs within multi-agent systems when performing flight missions without GPS.

The object of the research is the processes of controlling UAV agents based on data from onboard positioning systems.

The subject of the research is the methods of modeling and controlling multi-agent UAV systems and the hardware and software tools for their implementation.

The aim of the dissertation is to improve the efficiency of controlling multi-agent UAV systems during their deployment for mission execution through the development of models and methods of semi-physical simulation of control processes that ensure the execution of route scenarios and the stabilization of UAV agents' positions based on video data in GPS-denied environments.

The main research results and scientific novelty consist in the development of models and methods for semi-physical modeling of multi-agent UAV systems, formalized route scenario execution, and local UAV position control based on visual displacement.

For the first time:

- method of phase-based adaptive PID position control of UAVs based on visual displacement has been developed. Unlike existing methods, it generates, in each control cycle of the position control loop, a basic phase-specific configuration of control inputs for the flight controller based on positional error data, taking into

account the specified parameters of the current stage of the route scenario. This improves the reliability of route scenario completion to 92%.

The following has been improved:

- UAV navigation method during deployment and mission execution, which is based on software-defined movement routes generated within a simulation environment and represented as sequences of spatial vectors, together with a route scenario execution model that determines positional displacement from video images in combination with an independent vertical altitude-holding control loop. This enables local UAV position stabilization without GPS, achieving a mean absolute error of no more than 8 pixels along each horizontal axis under simulated conditions.

The following has been further developed:

- models and a hardware-software architecture for semi-physical simulation of multi-agent UAV systems. Unlike existing approaches, the proposed architecture jointly employs an integrated virtual simulation environment, hardware-in-the-loop (HIL) components, and software-in-the-loop (SIL) emulators to reproduce the control processes of individual UAV agents within a swarm, enabling the investigation of flight mission execution dynamics in real time.

The practical significance of the results obtained lies in the creation of software tools for semi-physical simulation aimed at investigating the operation and control processes of UAVs within a swarm. The developed HIL/SIL environment enables the joint investigation of software and hardware implementations of flight controllers and route scenario models, the definition and execution of parameterized sequences of route steps, spatial stabilization, the investigation of local position control in the absence of global coordinate referencing, and the evaluation of controller parameter calculations according to the current operating conditions of the position control loop.

The developed route scenario models, spatial stabilization methods, and PID control methods for the position control loop have been practically implemented in operational quadcopter prototypes in the form of algorithms and embedded software written in Python. These implementations were verified through simulation-based and real-world experiments.

The Introduction substantiates the relevance of research into models and methods of semi-physical modeling and local control of multi-agent UAV systems. It formulates the research aim and objectives, defines the object, subject, and methods of the research, and presents the main contributions to scientific novelty, the practical significance of the results obtained, and the author's personal contribution.

Chapter One analyzes modern models, methods, and tools for modeling and controlling UAV systems. It examines the characteristics of multi-agent UAV systems, approaches to software-based and semi-physical modeling, including SIL and HIL configurations, methods of route control, positioning and stabilization under GPS-denied conditions, as well as approaches to vision-based control and adaptive adjustment of PID controller parameters. Based on this analysis, the limitations of existing approaches are identified, and the research objectives are formulated.

Chapter Two develops the architecture of a hybrid semi-physical modeling environment for a multi-agent UAV system, combining a hardware-based HIL flight controller with software-based SIL implementations within a shared simulated environment. The principles of interaction between hardware and software components, the organization of information flows, and mechanisms for separating the communication channels of individual UAV agents are defined. The experimental environment is implemented using Gazebo, ArduPilot, MAVLink/MAVROS, and a Pixhawk flight controller. The operability of the architecture is verified in a configuration comprising one HIL agent and three SIL agents. This combination of a physical Pixhawk controller and software instances of ArduPilot also forms the basis of a published study on semi-physical modeling of a UAV group.

Chapter Three develops an agent-oriented UAV model and a route scenario model in which a route is represented as a sequence of parameterized route steps, while its execution is formalized using a finite-state machine. A method for spatial stabilization of a UAV under GPS-denied conditions is developed, in which horizontal control is performed using relative visual displacement in image coordinates, while the vertical component is maintained by an independent altitude control loop. A virtual drift mechanism is employed

to execute route steps, providing a connection between the route control level and the visual stabilization loop. The operability of route control is verified in SIL and HIL configurations, while spatial stabilization is evaluated under simulated and real-world conditions. The agent-oriented model and verification of route scenario execution in software-based and semi-physical configurations are also described in a corresponding publication. The spatial stabilization method was evaluated both in Gazebo/ArduPilot and on a physical UAV prototype.

Chapter Four develops a phase-based adaptive PID method for UAV position control using visual displacement. The method involves determining and adjusting the PID controller coefficients during each control cycle, taking into account the current operating phase of the position control loop, characteristics of positional error, a normalized indicator of visual estimation quality, route step parameters, cross-track deviation, and saturation of control channels. Saturation is considered independently for the two coordinate control channel.

The dissertation research was conducted at Chernihiv Polytechnic National University. The research results were implemented within the framework of the state-funded research project "Multi-Agent System for the Protection of Critical Infrastructure Facilities Based on a Swarm of Multicopter Drones" (State Registration No. 0123U101819, 2023–2025); at the State Scientific Research Institute of Armament and Military Equipment Testing and Certification for conducting field studies and developing UAV testing programs and methodologies; and in the educational process for bachelor's, master's, and doctoral students at Chernihiv Polytechnic National University (Appendix B).

Keywords: multi-agent UAV system, semi-physical simulation, visual positioning, spatial stabilization, route scenario, GPS-denied, adaptive PID control, Gazebo, ArduPilot, Pixhawk.

ЗМІСТ

АНОТАЦІЯ 2
СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ 7
ABSTRACT 9
ЗМІСТ 13
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ 16
ВСТУП 18
РОЗДІЛ І. АНАЛІЗ МЕТОДІВ І ЗАСОБІВ МОДЕЛЮВАННЯ ТА КЕРУВАННЯ МУЛЬТИАГЕНТНИМИ СИСТЕМАМИ БПЛА 27
1.1 Особливості мультиагентних систем БПЛА та задачі їх моделювання 27
1.2 Методи та засоби моделювання систем БПЛА 29
1.3 Підходи до програмного та напівнатурного моделювання БПЛА 33
1.4 Методи маршрутного керування БПЛА 37
1.5 Методи позиціювання та стабілізації БПЛА в умовах відсутності GPS 40
1.6 Методи керування БПЛА за візуальними даними та адаптивного PID-регулювання 45
1.7 Постановка задач дослідження 52
1.8 Висновки до розділу І 54
РОЗДІЛ ІІ. МОДЕЛІ ТА АРХІТЕКТУРА НАПІВНАТУРНОГО МОДЕЛЮВАННЯ МУЛЬТИАГЕНТНИХ СИСТЕМ БПЛА 56
2.1 Принципи побудови середовища напівнатурного моделювання мультиагентної системи БПЛА 56
2.2 Архітектура гібридного HIL/SIL середовища моделювання 57
2.3 Взаємодія програмних та апаратних компонентів середовища моделювання 59
2.4 Організація інформаційних потоків агентів у HIL/SIL середовищі 61
2.5 Реалізація експериментального HIL/SIL середовища 62
2.6 Експериментальне дослідження функціонування гібридної HIL/SIL архітектури 65
2.7 Висновки до розділу ІІ 69
РОЗДІЛ ІІІ. МЕТОД НАВІГАЦІЇ БПЛА В ХОДІ ВИКОНАННЯ МІСІЇ В УМОВАХ ВІДСУТНОСТІ GPS 71
3.1 Формальна модель БПЛА в системі напівнатурного моделювання 71

3.2 Модель маршрутного сценарію БПЛА 72
3.3 Модель виконання маршрутного сценарію на основі автомата станів 76
3.4 Отримання та оброблення сигналу візуального зміщення за даними камери 80
3.5 Метод просторової стабілізації БПЛА за візуальними даними без GPS 82
3.6 Реалізація маршрутного керування на основі візуальної стабілізації 86
3.7 Експериментальне дослідження маршрутного керування та просторової стабілізації 91
3.7.1 Експериментальне дослідження маршрутного керування у SIL- та HIL-конфігураціях 91
3.7.2 Експериментальне дослідження просторової стабілізації в модельованих та натурних умовах 96
3.8 Висновки до розділу III 99
РОЗДІЛ IV. МЕТОД ФАЗОВОГО АДАПТИВНОГО PID-КЕРУВАННЯ ПОЛОЖЕННЯМ БПЛА ЗА ВІЗУАЛЬНИМ ЗМІЩЕННЯМ 101
4.1 Постановка задачі адаптивного керування положенням БПЛА 101
4.2 Структура фазового методу адаптивного PID-керування 102
4.3 Визначення фаз та правил адаптації коефіцієнтів PID-регулятора 104
4.4 Урахування параметрів маршрутного кроку та насичення керуючих сигналів під час адаптації 107
4.5 Програмна реалізація адаптивного PID-керування 111
4.6 Методика експериментального дослідження 113
4.7 Порівняльне оцінювання PID-регуляторів із фіксованими та адаптивними коефіцієнтами 118
4.7.1 Оцінювання успішності проходження маршруту 118
4.7.2 Оцінювання позиційної похибки під час виконання маршруту 119
4.7.3 Оцінювання часу проходження маршруту 121
4.7.4 Оцінювання насичення керуючих сигналів 122
4.7.5 Статистичний аналіз результатів експерименту 123
4.7.6 Аналіз функціонування фазового механізму адаптації 124
4.7.7 Аналіз зміни коефіцієнтів PID-регулятора 127
4.7.8 Аналіз роботи керуючих каналів в умовах насичення 130
4.7.9 Аналіз характеристик керування за сегментами маршруту 133

4.8 Обговорення результатів та обмежень розробленого методу 137
4.9 Висновки до розділу IV 139
ВИСНОВКИ 142
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 147
ДОДАТКИ 158
Додаток А. Список публікацій здобувача за темою дисертації 158
Додаток Б. Довідки про впровадження 160
Додаток В. Сертифікати участі в конференціях 163
Додаток Г. Параметри та конфігурація гібридного HIL/SIL середовища моделювання. 164
Додаток Д. Приклади опису та програмної реалізації маршрутних сценаріїв БПЛА 168
Додаток Е. Фрагменти програмної реалізації фазового адаптивного PID-керування 172

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БПЛА – безпілотний літальний апарат

CSV – Comma-Separated Values (формат даних зі значеннями, розділеними комами)

GNSS – Global Navigation Satellite System (глобальна навігаційна супутникова система)

GPS – Global Positioning System (глобальна система позиціювання)

HIL – Hardware-in-the-Loop (напівнатурне моделювання з апаратним компонентом у контурі)

IBVS – Image-Based Visual Servoing (візуальне сервокерування за параметрами зображення)

IMU – Inertial Measurement Unit (інерціальний вимірювальний модуль)

JSON – JavaScript Object Notation (формат структурованого подання даних)

LiDAR – Light Detection and Ranging (лазерне виявлення та вимірювання відстані)

MAVLink – Micro Air Vehicle Link (протокол обміну повідомленнями з БПЛА)

MAVROS – пакет ROS для взаємодії з MAVLink-сумісними автопілотами

ORB – Oriented FAST and Rotated BRIEF (метод формування локальних візуальних ознак)

PBVS – Position-Based Visual Servoing (візуальне сервокерування за оціненим просторовим положенням)

PID – Proportional-Integral-Derivative (пропорційно-інтегрально-диференціальний регулятор)

RANSAC – Random Sample Consensus (метод узгодження моделі за випадковою вибіркою)

RC – Radio Control (канали дистанційного керування)

ROS – Robot Operating System (робототехнічна програмна платформа)

RTF – Real-Time Factor (коефіцієнт реального часу)

SDF – Simulation Description Format (формат опису моделей і модельованого середовища)

SIL – Software-in-the-Loop (програмне моделювання з програмною реалізацією контролера в контурі)

SITL – Software-in-the-Loop (позначення програмного режиму моделювання в ArduPilot та PX4)

SLAM – Simultaneous Localization and Mapping (одночасна локалізація та побудова карти)

UDP – User Datagram Protocol (протокол користувацьких датаграм)

UWB – Ultra-Wideband (надширокосмугова радіотехнологія)

VIO – Visual-Inertial Odometry (візуально-інерціальна одометрія)

VO – Visual Odometry (візуальна одометрія)

ВСТУП

Актуальність теми дослідження. Безпілотні літальні апарати застосовуються для виконання широкого кола завдань, пов'язаних із моніторингом, спостереженням, пошуком об'єктів, інспектуванням, картографуванням та виконанням автономних місій. Подальше розширення функціональних можливостей таких систем пов'язане з переходом від використання окремих БПЛА до мультиагентних систем, у яких декілька літальних апаратів функціонують у межах спільного сценарію та можуть здійснювати інформаційну взаємодію. Дослідження таких систем потребує врахування не лише динаміки окремого літального апарата, а й процесів локального керування, сенсорного забезпечення, виконання маршрутних сценаріїв та взаємодії між агентами.

Моделювання мультиагентної системи БПЛА характеризується необхідністю узгодження різних рівнів представлення. На рівні окремого апарата мають бути відтворені його динаміка, стан, сенсорна інформація та локальні контури керування. На рівні агента необхідно представити маршрутний сценарій, правила зміни станів і формування керуючих дій. На рівні мультиагентної системи додатково виникають задачі розмежування інформаційних потоків, організації взаємодії та спільного функціонування декількох БПЛА. Надмірне спрощення окремих складових може виключати суттєві для експерименту процеси, тоді як надмірна деталізація збільшує обчислювальні й апаратні витрати. Така обставина визначає необхідність вибору складу експериментальної моделі відповідно до задачі дослідження.

Одним із поширених підходів до дослідження алгоритмів керування БПЛА є програмне моделювання за схемою Software-in-the-Loop (SIL), за якої програмне забезпечення польотного контролера виконується на комп'ютері, а динаміка літального апарата та сенсорні дані формуються засобами модельованого середовища. Такий підхід спрощує побудову конфігурацій із декількома БПЛА та проведення повторюваних експериментів, однак не дає змоги врахувати особливості виконання польотного програмного забезпечення на реальному апаратному

контролері.

Напівнатурне моделювання за схемою Hardware-in-the-Loop (HIL) передбачає включення фізичного польотного контролера до замкненого модельованого контуру. Це дає змогу досліджувати взаємодію реального апаратного компонента з модельованою динамікою та сенсорним середовищем. Водночас використання окремого фізичного польотного контролера для кожного агента мультиагентної системи потребує відповідного збільшення кількості обладнання, каналів зв'язку та засобів синхронізації. Тому актуальним є дослідження гібридних конфігурацій, у яких HIL- та SIL-реалізації польотних контролерів функціонують у межах одного середовища моделювання. У роботі така конфігурація реалізована для одночасного функціонування одного HIL- та трьох SIL-агентів у спільному середовищі Gazebo.

Розвитку моделей і методів моделювання та керування мультиагентними системами БПЛА присвячено низку наукових досліджень, у яких розглянуто різні аспекти побудови таких систем, організації їх функціонування та експериментальної перевірки. Зокрема, Понніа (Ponniah) і Дантскер (Dantsker) розглядали особливості організації інформаційної взаємодії та координації дій БПЛА у складі мультиагентних систем [8]. У працях Амескіта-Брукса (Amezquita-Brooks) та співавторів досліджено математичні моделі динаміки квадрокоптерів і визначено особливості їх застосування залежно від умов моделювання [16]. Питання програмного моделювання та експериментальної перевірки алгоритмів БПЛА з урахуванням обмежень апаратних платформ висвітлено в роботах Ауджелло (Augello) та співавторів [30]. Особливості напівнатурного моделювання систем із декількома БПЛА досліджували Янг (Yang) та співавтори, які запропонували експериментальну платформу з використанням моделей динаміки, реальних польотних контролерів і засобів інформаційного обміну [32].

Незважаючи на отримані результати, подальшого дослідження потребують особливості побудови середовищ напівнатурного моделювання мультиагентних систем БПЛА, зокрема організація спільного функціонування апаратних і програмних реалізацій польотних контролерів у єдиному модельованому середовищі.

При цьому важливими залишаються питання розмежування інформаційних потоків окремих агентів, узгодження роботи компонентів системи та забезпечення можливості експериментальної перевірки процесів керування БПЛА під час виконання маршрутних сценаріїв. Використання таких середовищ потребує моделей, які дають змогу відтворювати процеси маршрутного переміщення окремих БПЛА-агентів та досліджувати їх функціонування за різних умов навігаційного забезпечення.

Таким чином, актуальним науковим завданням є розробка та експериментальне дослідження моделей і методів напівнатурного моделювання, що забезпечують спільне використання апаратних і програмних компонентів БПЛА у складі мультиагентних систем при виконанні польотних завдань без використання GPS.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота виконувалася в Національному університеті «Чернігівська політехніка» в межах науково-дослідної роботи за державним замовленням «Мультиагентна система захисту об'єктів критичної інфраструктури на основі рою мультикоптерних дронів» (номер державної реєстрації 0123U101819, 2023–2025 рр.). У межах зазначеної науково-дослідної роботи здобувачем виконано дослідження, пов'язані з розробленням моделей і методів напівнатурного моделювання мультиагентних систем БПЛА, організацією HIL/SIL середовища напівнатурного моделювання, формалізацією маршрутних сценаріїв та дослідженням локального керування положенням БПЛА за візуальними даними.

Результати дисертаційного дослідження впроваджено у Державному науково-дослідному інституті випробувань і сертифікації озброєння та військової техніки під час проведення натурних досліджень та відпрацювання програм і методик випробувань безпілотних авіаційних комплексів. Практичне застосування отримали програмно-апаратна архітектура напівнатурного HIL-моделювання, підхід до маршрутного переміщення БПЛА без використання GPS/GNSS та адаптивне керування стабілізацією положення БПЛА за даними спрямованої вниз відеокамери.

Мета і задачі дослідження. Метою дисертаційної роботи є підвищення

ефективності керування мультиагентними системами БПЛА при їх розгортанні для виконання місії за рахунок розроблення моделей і методів напівнатурного моделювання процесів керування, що забезпечують виконання маршрутних сценаріїв і стабілізацію положення БПЛА-агентів за відео даними в умовах відсутності GPS.

Для досягнення поставленої мети передбачається вирішення таких задач:

- провести аналіз існуючих моделей, методів і засобів моделювання та керування мультиагентними системами БПЛА, визначити їх особливості та обмеження;
- розробити архітектуру напівнатурного моделювання мультиагентної системи БПЛА, що забезпечує спільне функціонування апаратних HIL- та програмних SIL-реалізацій польотних контролерів;
- розробити агентно-орієнтовану модель виконання маршрутних сценаріїв БПЛА в середовищі моделювання;
- розробити метод керування положенням і просторової стабілізації БПЛА за візуальними даними в умовах відсутності GPS;
- розробити фазовий метод адаптивного PID-керування положенням БПЛА за візуальним зміщенням;
- провести експериментальну перевірку розроблених моделей і методів та оцінити отримані результати.

Об’єктом дослідження є процеси керування БПЛА-агентами по даним бортової системи позиціонування.

Предметом дослідження є методи моделювання та керування мультиагентними системами БПЛА та програмно-апаратні засоби їх реалізації.

Методи дослідження. Для вирішення поставлених задач використано методи системного та порівняльного аналізу для дослідження наявних моделей, методів і програмних засобів моделювання та керування системами БПЛА і визначення їхніх можливостей та обмежень.

Методи агентно-орієнтованого моделювання використано для формалізації окремого БПЛА як автономного агента з локальним станом, маршрутним сценарієм, контуром керування та правилами переходів між станами. Положення теорії

скінченних автоматів застосовано для представлення послідовності станів під час виконання маршрутного сценарію та формалізації умов переходу між ними.

Методи імітаційного та напівнатурного моделювання застосовано для побудови та експериментального дослідження програмних SIL- і апаратно-програмних HIL-конфігурацій БПЛА. Для реалізації експериментального середовища використано Gazebo, ArduPilot і реальний польотний контролер Pixhawk. Працездатність розробленої архітектури перевірено в конфігурації одного HIL- та трьох SIL-агентів.

Методи PID-регулювання використано для формування керуючих впливів у горизонтальному та вертикальному контурах просторової стабілізації, а також у контурі маршрутного керування. Для горизонтального керування як сигнал зворотного зв'язку застосовано оцінку візуального зміщення, сформовану зовнішнім функціональним модулем за даними спрямованої вниз камери. Внутрішні алгоритми визначення візуального зміщення в межах дисертаційної роботи не розроблялися та не модифікувалися.

Методи адаптивного керування використано для побудови фазового механізму зміни коефіцієнтів PID-регулятора залежно від поточних умов роботи позиційного контуру, характеристик позиційної похибки, нормованого показника якості візуального оцінювання, параметрів маршрутного кроку та ознак насичення окремих координатних каналів.

Для кількісного оцінювання експериментальних результатів використано методи статистичного аналізу та порівняння вибірок. Експериментальне оцінювання розробленого фазового методу виконано за однакових умов SIL-моделювання.

Достовірність одержаних результатів дослідження забезпечується експериментальною перевіркою розроблених моделей і методів у середовищах SIL і HIL, а для методу просторової стабілізації також шляхом порівняння з результатами, отриманими у натурних умовах, повторюваністю умов проведення експериментів, кількісним аналізом зареєстрованих даних та застосуванням статистичних методів під час порівняння конфігурацій керування. Результати дослідження апробовано на

наукових і науково-практичних конференціях. Практичну значущість отриманих результатів підтверджено їх впровадженням у Державному науково-дослідному інституті випробувань і сертифікації озброєння та військової техніки під час проведення натурних досліджень та відпрацювання програм і методик випробувань безпілотних авіаційних комплексів.

Основні результати дослідження та наукова новизна роботи полягають у розробленні моделей і методів напівнатурного моделювання мультиагентних систем БПЛА, формалізованого виконання маршрутних сценаріїв та локального керування положенням БПЛА за візуальним зміщенням.

Вперше:

- розроблено метод фазового адаптивного PID-керування положенням БПЛА за візуальним зміщенням, який, на відміну від існуючих, у кожному циклі керування роботою позиційного контуру формує по даним позиційної похибки базову фазову конфігурацію керуючих впливів для польотного контролера із урахуванням заданих параметрів поточного етапу маршрутного сценарію, що забезпечує підвищення надійності проходження маршрутного сценарію до 92 відсотків.

Удосконалено:

- метод навігації БПЛА при його розгортанні в ході виконання місії, що базується на програмно визначених за допомогою середовища моделювання маршрутах переміщення, як послідовності просторових векторів, та моделі виконання маршрутного сценарію з визначенням зміщення позиції по відео зображенню у поєднанні з незалежним вертикальним контуром утримання висоти, що забезпечує просторову стабілізацію БПЛА без використання GPS із середньою абсолютною похибкою за кожною з горизонтальних осей до 8 пікселів у модельованих умовах.

Набули подальшого розвитку:

- моделі та програмно-апаратна архітектура напівнатурного моделювання мультиагентних систем БПЛА, яка, на відміну від снуючих, спільно

використовує інтегроване віртуальне імітаційне середовище, апаратні засоби (Hardware-in-the-Loop, HIL) та програмні емулятори (Software-in-the-Loop, SIL) для відтворення процесів керуваннями дронами-агентами у складі рою, що забезпечує дослідження динаміки виконання польотних завдань у реальному часі.

Практичне значення отриманих результатів полягає у створенні програмних засобів напівнатурного моделювання для дослідження процесів функціонування та керування БПЛА у складі рою. Розроблене HIL/SIL середовище дає змогу спільно досліджувати програмні та апаратні реалізації польотних контролерів, моделі маршрутних сценаріїв, задавати та виконувати параметризовані послідовності маршрутних кроків, виконувати просторову стабілізацію, досліджувати локальне керування положенням в умовах відсутності глобальної координатної прив’язки та оцінювати розрахунки параметрів регулятора відповідно до поточних умов роботи позиційного контуру.

Розроблені моделі маршрутних сценаріїв, методи просторової стабілізації та PID-керування роботою позиційного контуру отримали практичну реалізацію в діючих макетах квадрокоптерних дронів у вигляді алгоритмів та вбудованих програм на мові Python, які пройшли верифікацію в ході модельних та натурних експериментів.

Результати дисертаційного дослідження впроваджені

Дисертація виконувалася в Національному університеті «Чернігівська політехніка». Результати наукових досліджень були впроваджені під час виконання за державним замовленням НДР «Мультиагентна система захисту об’єктів критичної інфраструктури на основі рою мультикоптерних дронів» (номер державної реєстрації: 0123U101819 (2023-2025)), у Державному науково-дослідному інституті випробувань і сертифікації озброєння та військової техніки для проведення натурних досліджень та відпрацювання програм і методик випробувань БПЛА, у навчальний процес з підготовки бакалаврів, магістрів та аспірантів НУ«Чернігівська політехніка» (Додаток Б).

Особистий внесок здобувача. Усі наукові результати, що становлять основний зміст дисертаційної роботи та виносяться на захист, отримані здобувачем особисто. У наукових працях, опублікованих у співавторстві, у дисертації використано лише результати, отримані за особистої участі здобувача. Особистий внесок здобувача у спільні публікації полягає у розробленні та програмній реалізації моделей і методів, створенні та налаштуванні експериментального середовища, проведенні SIL-, HIL- і натурних експериментів, статистичному опрацюванні та аналізі отриманих результатів. Конкретний особистий внесок у кожну спільну публікацію наведено у списку публікацій здобувача за темою дисертації.

Апробація результатів дисертації. Основні наукові результати дисертаційної роботи доповідалися та обговорювалися на таких наукових і науково-практичних конференціях**:**

1. V Міжнародній науково-практичній конференції «Новітні технології сучасного суспільства (НТСС-2024)» (м. Чернігів, Україна, 12 грудня 2024 р.)
2. 15th International Conference on Advanced Computer Information Technologies (ACIT 2025)
3. 13th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS 2025) (м. Глівіце, Польща, 2025 р.)

Матеріали апробації результатів дисертаційного дослідження відображено у відповідних публікаціях, а підтвердні матеріали наведено у додатку В.

Публікації. За темою дисертаційного дослідження опубліковано 6 наукових праць, з яких 3 наукові праці відображають основні наукові результати дисертації та 3 наукові праці засвідчують апробацію матеріалів дисертації. У працях, опублікованих у співавторстві, особистий внесок здобувача зазначено окремо.

Структура та обсяг дисертації. Дисертаційна робота складається з анотації, вступу, чотирьох розділів із висновками до кожного з них, загальних висновків, списку використаних джерел та шести додатків. Список використаних джерел містить 100 найменувань. Робота містить 29 рисунків та 25 таблиць.

РОЗДІЛ І. АНАЛІЗ МЕТОДІВ І ЗАСОБІВ МОДЕЛЮВАННЯ ТА КЕРУВАННЯ МУЛЬТИАГЕНТНИМИ СИСТЕМАМИ БПЛА

1.1 Особливості мультиагентних систем БПЛА та задачі їх моделювання

У межах цього дослідження мультиагентна система БПЛА розглядається як система з декількох автономних БПЛА-агентів, що функціонують у спільному середовищі, мають власний локальний стан і засоби інформаційної взаємодії [1]. Механізми координації та спільного виконання місії визначаються конкретним сценарієм функціонування системи. На відміну від сукупності незалежно керованих БПЛА, у такій системі виникають задачі інформаційного обміну, розподілу завдань, планування переміщення та узгодження дій агентів. В огляді [2] комунікація, розподіл завдань, формування траєкторій і координація польоту розглядаються як ключові складові автономної роботи повітряних груп. Специфіка повітряних мультиагентних систем додатково визначається рухом у тривимірному просторі та впливом динаміки окремого літального апарата на можливість реалізації колективної поведінки.

Модель мультиагентної системи не можна звести до сукупності незалежних моделей окремих БПЛА, оскільки результат її функціонування залежить від взаємодії між агентами. У [3] показано взаємозалежність між можливостями окремого БПЛА та поведінкою рою в цілому: характеристики локального керування, сприйняття, обчислювальних ресурсів і комунікації обмежують можливі режими колективної поведінки. Водночас вимоги до групового функціонування визначають вимоги до окремих агентів. Отже, під час моделювання необхідно враховувати не тільки стан кожного БПЛА, а й правила, за якими локальні дії впливають на систему в цілому.

У межах цієї роботи під локальним керуванням розуміється формування керуючих дій на рівні окремого БПЛА-агента на основі його стану, маршрутного завдання та доступних сигналів зворотного зв'язку.

Суттєвою характеристикою мультиагентної системи є архітектура прийняття рішень. У централізованій схемі інформація про стан агентів надходить до

центрального компонента системи, де формуються рішення для всієї групи. До характерних обмежень такого підходу належать зростання комунікаційного та обчислювального навантаження зі збільшенням кількості БПЛА, а також залежність функціонування системи від центрального компонента [4]. У децентралізованих і розподілених схемах рішення формуються на рівні окремих агентів із використанням локально доступної інформації, що потребує механізмів узгодження їхніх станів і дій [5; 6]. Тому модель повинна відтворювати не лише рух БПЛА, а й структуру прийняття рішень та склад інформації, доступної кожному агенту.

Іншою складовою є інформаційна взаємодія. У системах БПЛА застосовуються як безпосередні канали обміну між апаратами, так і зв'язки з наземною або іншою зовнішньою інфраструктурою. Через рух агентів структура такої мережі може змінюватися в часі. До факторів, що ускладнюють комунікацію в роях БПЛА, віднесено високу мобільність, нестабільність повітряних каналів, обмежені ресурси та змінні вимоги до якості обслуговування [7]. Отже, під час моделювання необхідно явно визначати припущення щодо доступності каналів, затримок і складу інформації, що передається між агентами.

Зі збільшенням кількості БПЛА ускладнюються як керування, так і організація обміну інформацією. Ponniah і Dantsker розглядають ці аспекти як взаємопов'язані задачі: агенти повинні колективно визначати свої дії та водночас поширювати локальну інформацію, необхідну для формування спільного уявлення про стан системи. Тому масштабованість мультиагентної моделі визначається не лише можливістю збільшити кількість модельованих БПЛА, а й здатністю відтворювати необхідні механізми координації та інформаційної взаємодії при зміні розміру групи [8].

Для реалізації колективної поведінки важливою є також інформація про взаємне положення агентів. Відносна локалізація є основою для низки задач вищого рівня в мультиагентних системах, зокрема для координації та підтримання необхідного взаємного розташування [9]. Тому в мультиагентній моделі необхідно визначати, яка інформація про положення сусідніх агентів доступна кожному БПЛА.

Аналогічно групове планування потребує врахування взаємних просторових обмежень: зі збільшенням кількості БПЛА складність задачі побудови узгоджених маршрутів зростає [10]. Отже, взаємне положення агентів і узгодження їхнього руху є важливими складовими моделі мультиагентної системи.

Відповідність моделі умовам практичного застосування залежить і від того, які фізичні та інформаційні обмеження включені до експерименту. У [11] показано, що обмеження руху і комунікації, часові затримки, збурення та перешкоди можуть істотно впливати на колективну поведінку. Запропонована авторами модель ураховувала ці фактори та була перевірена на групі з 30 автономних БПЛА. Отже, оцінювання алгоритму лише за ідеалізованих умов може бути недостатнім, якщо досліджувана властивість системи залежить від відповідних фізичних або інформаційних обмежень.

На підставі проведеного аналізу задачі моделювання мультиагентної системи БПЛА доцільно розглядати на трьох взаємопов'язаних рівнях. На рівні окремого агента відтворюються його стан, динаміка, локальне керування та сенсорна інформація [2; 3]. На рівні міжагентної взаємодії задаються доступні канали обміну, інформація про сусідні апарати та правила, за якими стан одного агента впливає на рішення інших [7; 9]. На груповому рівні досліджуються координація, розподіл завдань, узгодження руху та взаємні просторові обмеження [12].

Отже, основна складність моделювання мультиагентних систем БПЛА полягає у відтворенні взаємозв'язку між локальними процесами окремого апарата та колективною поведінкою системи. Необхідний рівень деталізації фізичних, сенсорних і комунікаційних процесів повинен визначатися завданням експерименту, оскільки припущення, прийняті на рівні окремого агента або каналу взаємодії, можуть впливати на результати дослідження всієї групи [11; 13].

1.2 Методи та засоби моделювання систем БПЛА

Для дослідження систем БПЛА використовуються моделі різного рівня деталізації, вибір яких визначається метою експерименту. Критеріями вибору середовища можуть бути точність представлення динаміки, підтримка сенсорів,

можливість одночасного моделювання декількох апаратів, обчислювальна продуктивність та засоби інтеграції з алгоритмами керування [14]. Аналіз засобів моделювання роїв БПЛА також показує, що окремі платформи орієнтовані на різні складові системи: фізичну динаміку, сенсорне сприйняття, колективну поведінку або інформаційну взаємодію [15].

Для дослідження динаміки БПЛА використовують математичні моделі поступального й обертального руху. Для мультироторних апаратів вони можуть включати динаміку твердого тіла, тягу та моменти роторів, аеродинамічний опір та інші фізичні ефекти. Amezquita-Brooks та співавтори показали, що моделі квадрокоптерів відрізняються складом урахованих компонентів, а придатність спрощених представлень залежить від режиму, для якого вони застосовуються [16]. Отже, ступінь деталізації математичної моделі доцільно визначати відповідно до досліджуваних процесів, а не максимізувати незалежно від мети експерименту.

Для задач автономного керування математичної моделі руху часто недостатньо, оскільки необхідно також відтворювати середовище, перешкоди та сенсорну інформацію. Для цього застосовуються робототехнічні симулятори, які поєднують модель рухомого об’єкта з тривимірною сценою та моделями сенсорів. В огляді [14] до таких засобів віднесено Gazebo, Webots, CoppeliaSim, ARGoS та інші платформи, які відрізняються набором фізичних моделей, сенсорів, підтримуваних мов програмування, продуктивністю та можливостями інтеграції.

Одним із таких середовищ є Gazebo [17]. Актуальна гілка Gazebo Sim забезпечує моделювання фізичної взаємодії об’єктів і підтримує широкий набір моделей сенсорів, зокрема камер, інерціальних сенсорів, LiDAR, магнітометра та навігаційних сенсорів [18]. Модульна структура середовища дає змогу поєднувати фізичну модель із зовнішніми програмними компонентами керування. При цьому функціональні можливості залежать від конкретної версії Gazebo, тому характеристики окремих поколінь платформи не слід автоматично переносити між ними.

На основі робототехнічних симуляторів створюються спеціалізовані

середовища для БПЛА. RotorS є модульною платформою на основі Gazebo, яка включає моделі мікролітальних апаратів, засоби керування та оцінювання стану [19]. XTDrone інтегрує ROS, Gazebo та PX4, а також моделі динаміки, сенсорів, алгоритми керування й оцінювання стану. Платформа підтримує одночасне моделювання декількох БПЛА та використовувалася авторами для перевірки алгоритмів візуальної локалізації й формаційного польоту [20]. Такі інтегровані засоби дозволяють досліджувати взаємодію між моделлю апарата, сенсорами та програмними компонентами керування в межах спільного середовища.

Інший підхід реалізовано в MATLAB/Simulink та UAV Toolbox, де математичні моделі, алгоритми керування та сценарії можуть розроблятися в єдиному інструментальному середовищі. UAV Toolbox підтримує як спрощене сценарне представлення середовища, придатне для швидкого формування експериментів, так і тривимірне середовище на основі Unreal Engine для моделювання даних камер, LiDAR та інших сенсорів [22]. Це дозволяє вибирати рівень деталізації середовища відповідно до задачі: від аналізу алгоритмів керування до досліджень, у яких суттєвими є властивості сенсорних даних.

Вимоги до фізичної точності, швидкості моделювання та деталізації візуального середовища можуть суперечити одна одній. Ця проблема безпосередньо врахована в архітектурі Flightmare, де модуль динаміки квадрокоптера відокремлений від модуля тривимірної візуалізації, що дозволяє виконувати фізичне моделювання без обов’язкового формування зображення [21]. Такий підхід демонструє, що деталізація графічного середовища повинна визначатися потребами конкретного експерименту, особливо якщо візуальні дані не є його основним об’єктом.

Для мультиагентних систем окремого представлення може потребувати поведінка агентів. Агентно-орієнтоване моделювання дозволяє описувати автономних учасників системи через їхні локальні стани та правила взаємодії, при цьому детальна фізична динаміка апарата може бути спрощена відповідно до мети експерименту. Прикладом застосування такого підходу до систем БПЛА є моделювання захисного рою в середовищі NetLogo [23]. Такий рівень представлення

орієнтований насамперед на дослідження взаємодії агентів і формування групової поведінки.

Інший спеціалізований напрям пов’язаний із моделюванням комунікаційних процесів. OMNeT++ є дискретно-подійним середовищем, орієнтованим на мережі та інформаційну взаємодію; у дослідженнях роїв БПЛА воно використовується для аналізу топологій, маршрутизації та передавання даних між агентами [15; 24]. Такі засоби доповнюють фізичні симулятори у випадках, коли характеристики мережевого обміну є окремим предметом дослідження.

Для систематизації розглянутих підходів їх порівняльні характеристики наведено в Таблиці 1.1.

Таблиця 1.1

Порівняння основних підходів і засобів моделювання систем БПЛА

Підхід	**Основний об’єкт моделювання**	**Приклад засобів**	**Типові задачі**	**Основні обмеження**
Математичне моделювання	Динаміка та стан БПЛА	MATLAB/Simulink, чисельні моделі	Аналіз динаміки, розроблення і перевірка законів керування	Результат залежить від прийнятих спрощень і складу фізичних ефектів
Робототехнічне тривимірне моделювання	Динаміка, середовище, сенсори	Gazebo, Webots, CoppeliaSim	Навігація, взаємодія із середовищем, сенсорне моделювання	Зростання деталізації підвищує обчислювальне навантаження
Інтегровані платформи БПЛА	Модель БПЛА, сенсори, автопілотні та керувальні компоненти	RotorS, XTDrone	Комплексна перевірка програмних компонентів, експерименти з декількома БПЛА	Складність конфігурації зростає зі складом інтегрованих компонентів
Агентно-орієнтоване моделювання	Локальні стани та правила взаємодії агентів	NetLogo	Дослідження групової поведінки та координації	Фізична динаміка БПЛА представлена спрощено
Мережеве дискретно-подійне моделювання	Передавання даних та мережева взаємодія	OMNeT++	Топології, маршрутизація, затримки та інформаційний обмін	Не призначене для детального моделювання динаміки польоту

Наведені підходи не є взаємовиключними. В одному експериментальному середовищі можуть поєднуватися математична модель динаміки, тривимірна сцена, моделі сенсорів, програмні компоненти керування та засоби міжагентної взаємодії. Прикладом такої інтеграції є XTDrone, де Gazebo, ROS і PX4 об'єднані в платформу для дослідження одного або декількох БПЛА [20].

Отже, вибір методу та програмного засобу моделювання визначається тим, які властивості системи є суттєвими для конкретного експерименту. Для аналізу динаміки достатнім може бути математичне представлення апарата, тоді як дослідження сенсорних алгоритмів потребує моделі середовища та відповідних сенсорів, а мультиагентні задачі додатково вимагають представлення взаємодії між агентами. Тому комплексне моделювання систем БПЛА може потребувати поєднання декількох засобів, кожен із яких відтворює необхідний аспект досліджуваної системи [25].

1.3 Підходи до програмного та напівнатурного моделювання БПЛА

Засоби моделювання БПЛА відрізняються не лише рівнем деталізації фізичної моделі, а й складом експериментального контуру та місцем виконання програмного забезпечення керування. Залежно від постановки експерименту польотне програмне забезпечення може виконуватися на персональному комп'ютері або на реальному польотному контролері, тоді як динаміка літального апарата залишається модельованою. Для таких конфігурацій застосовуються, зокрема, підходи Software-in-the-Loop та Hardware-in-the-Loop.

До поширених відкритих автопілотних платформ для БПЛА належать ArduPilot та PX4 [26]. У документації ArduPilot і PX4 програмний режим моделювання позначається SITL (Software-in-the-Loop) [27]. У межах цієї дисертації для термінологічної узгодженості використовується скорочення SIL, тоді як позначення SITL зберігається лише під час згадування власної назви режиму відповідної програмної платформи.

У SIL-конфігурації польотне програмне забезпечення виконується на комп'ютері без використання фізичного польотного контролера. Наприклад,

ArduPilot SITL дозволяє запускати програмне забезпечення автопілота як окремий процес, використовуючи модельовані дані сенсорів і модель динаміки транспортного засобу [27]. Така конфігурація дає змогу перевіряти програмну логіку керування, параметри автопілота та взаємодію із зовнішніми програмними компонентами. Виконання автопілота як програмного процесу також дозволяє використовувати стандартні засоби налагодження й аналізу програмного коду.

Обмеження SIL полягає в тому, що польотне програмне забезпечення не виконується на цільовій апаратній платформі. Через це часові характеристики виконання, ресурсні обмеження контролера та частина особливостей апаратно-програмної взаємодії можуть залишатися поза межами експерименту. Augello та співавтори зазначають, що виконання задач у програмному симуляційному середовищі не обов’язково відтворює їх часову поведінку на ресурсно обмеженій бортовій платформі [30].

Для включення реального польотного контролера до модельованого контуру застосовується Hardware-in-the-Loop (HIL). У такій конфігурації польотне програмне забезпечення виконується на фізичному контролері, тоді як модель динаміки БПЛА та модельовані сенсорні дані формуються зовнішнім симулятором [29]. Сформовані контролером керуючі сигнали повертаються до моделі літального апарата, утворюючи замкнений контур. Це дає змогу досліджувати виконання польотного програмного забезпечення на цільовій апаратній платформі та його взаємодію з модельованим середовищем.

Водночас HIL не є еквівалентом натурного випробування. Реальним компонентом такого контуру є польотний контролер, тоді як рух літального апарата, зовнішнє середовище та значна частина сенсорних сигналів залишаються модельованими [29]. Тому результати HIL-експерименту залежать від адекватності моделей динаміки, сенсорів і середовища. Фізичні ефекти або похибки вимірювань, які не представлені в моделі, не відтворюються автоматично лише завдяки використанню реального польотного контролера. Порівняльні характеристики SIL- та HIL-моделювання наведено в Таблиці 1.2

Таблиця 1.2

Порівняння SIL- та HIL-моделювання БПЛА

Підхід	**SIL**	**HIL**
Виконання польотного програмного забезпечення	На комп'ютері.	На польотному контролері.
Використання польотного контролера	Ні	Так
Модель динаміки	Програмна	У зовнішньому симуляторі
Основні можливості	Розроблення, налагодження та функціональна перевірка програмних алгоритмів	Перевірка виконання ПЗ на цільовій апаратній платформі та взаємодії з модельованим контуром
Основні обмеження	Не забезпечує перевірку виконання ПЗ на цільовому польотному контролері	Динаміка апарата й сенсорні сигнали залишаються модельованими; ускладняється організація апаратно-програмної взаємодії

Для мультиагентних систем застосування HIL додатково ускладнює організацію інформаційного обміну та синхронізацію декількох контурів керування [31]. Yang та співавтори розробили напівнатурну платформу для декількох БПЛА літакового типу, до складу якої входили моделі динаміки, польотні контролери, бортові обчислювальні компоненти та засоби комунікації [32]. У дослідженій ними конфігурації пряме підключення декількох фізичних польотних контролерів до одного середовища супроводжувалося значними втратами даних уже для двох апаратів, що зумовило зміну структури інформаційної взаємодії. Цей результат характеризує конкретну реалізацію, однак показує, що масштабованість багатоагентного HIL-середовища залежить не лише від кількості апаратних компонентів, а й від організації каналів обміну та їх синхронізації. Як приклад апаратної платформи польотного контролера для БПЛА може використовуватися Pixhawk, архітектура якого розроблялася для автономного польоту та бортової обробки сенсорних даних [33].

Зі збільшенням кількості агентів конфігурація з окремим фізичним польотним

контролером для кожного модельованого БПЛА також потребує відповідного збільшення кількості апаратних компонентів. Натомість повністю програмна конфігурація не забезпечує перевірку виконання польотного програмного забезпечення на реальному контролері. Тому для мультиагентних систем доцільно розглядати можливість поєднання програмних і апаратних компонентів відповідно до властивостей системи, які необхідно дослідити.

Подальший розвиток підходу до напівнатурного моделювання у межах цієї роботи полягає у формуванні спільного експериментального середовища, в якому апаратна HIL- та програмні SIL-реалізації польотних контролерів одночасно взаємодіють з окремими моделями БПЛА в єдиній модельованій сцені. Розмежування інформаційних потоків окремих агентів виконується за допомогою незалежних просторів імен, ідентифікаторів та каналів обміну. Така структура дає змогу досліджувати взаємодію програмних і апаратних компонентів у межах однієї мультиагентної конфігурації. У роботі працездатність цього підходу експериментально перевірено для конфігурації одного HIL- та трьох SIL-агентів.

Термін «напівнатурне моделювання» у науковій літературі використовується не повністю уніфіковано. Yang та співавтори застосовують поняття semi-physical simulation до експериментальної платформи, що поєднує модельовані фізичні процеси з реальними компонентами системи керування та комунікації [31]. У межах цієї дисертації під напівнатурним моделюванням розуміється моделювання, у якому до замкненого модельованого контуру включено реальний апаратний компонент системи керування, зокрема польотний контролер. Повністю програмні конфігурації розглядаються окремо як SIL-моделювання.

Отже, SIL і HIL забезпечують різний ступінь залучення реального обладнання до модельованого контуру. Для мультиагентних систем збільшення кількості апаратних компонентів підвищує вимоги до організації інформаційного обміну, синхронізації та побудови експериментальної інфраструктури [31]. Це обґрунтовує дослідження конфігурацій, у яких програмні та апаратні компоненти поєднуються відповідно до задач конкретного експерименту.

1.4 Методи маршрутного керування БПЛА

У широкому розумінні маршрутне керування БПЛА охоплює декілька взаємопов'язаних задач: формування місії, пошук допустимого шляху, побудову траєкторії та виконання сформованої послідовності руху системою керування. У літературі поняття маршруту, шляху та траєкторії використовуються в різних контекстах, тому їх доцільно розмежовувати. У роботі [34] path planning розглядається як пошук геометрично допустимого шляху між заданими станами, тоді як траєкторія додатково враховує часовий закон руху та кінематичні або динамічні обмеження. Подібне розмежування між пошуком шляху і подальшим формуванням неперервної траєкторії наведено в [35].

У межах цієї роботи під шляхом розуміється геометричне представлення переміщення між заданими положеннями, під траєкторією — рух уздовж шляху з його часовою параметризацією, а під маршрутом — впорядкована послідовність цільових етапів виконання місії. У розробленій далі моделі маршрут подається як послідовність параметризованих маршрутних кроків.

Таке представлення дозволяє розглядати маршрутне керування як багаторівневий процес. На верхньому рівні визначаються цілі місії та порядок їх досягнення, далі формується допустимий шлях і траєкторія, а безпосередній рух забезпечується локальними контурами керування БПЛА. При цьому планування руху спирається на інформацію про стан апарата, середовище та встановлені обмеження [35].

Пошук шляху може формулюватися як задача оптимізації за одним або декількома критеріями, серед яких розглядаються довжина шляху, час виконання, енергетичні витрати та ризик зіткнення. До обмежень належать перешкоди, допустимі області руху та характеристики БПЛА. У сучасних дослідженнях застосовуються графові, вибіркові, потенціальні, метаевристичні, навчальні та гібридні методи планування [36-38]. Їх порівняльні характеристики наведено в Таблиці 1.3.

Таблиця 1.3

Порівняння основних класів методів планування руху БПЛА

Клас методів	**Основний принцип**	**Назва методу**	**Характерні можливості**	**Основні обмеження**
Графові	Пошук у дискретизованому представленні простору	Dijkstra, A*[38; 39]	Побудова шляху у відомому середовищі	Зростання кількості станів зі збільшенням розмірності та деталізації
Вибіркові	Формування графа або дерева за вибіркою допустимих станів	RRT, PRM [40; 41]	Робота зі складними конфігураційними просторами	Початковий шлях може потребувати оптимізації або згладжування
Потенціальні	Формування притягувального впливу цілі та відштовхувального впливу перешкод	APF [42]	Локальне коригування напрямку руху	Можливість виникнення локальних мінімумів
Метаевристичні	Ітераційна оптимізація за заданою цільовою функцією	GA, PSO, ACO [43-45]	Урахування декількох критеріїв оптимізації	Обчислювальні витрати залежать від структури та розмірності задачі
Навчальні	Формування керувальної політики або рішення на основі навчання	Методи навчання з підкріпленням	Застосування в динамічних і частково невідомих середовищах	Потреба у навчанні та відповідних обчислювальних ресурсах
Гібридні	Поєднання декількох методів планування	Комбінації глобальних і локальних алгоритмів	Розділення глобального планування та локальної реакції	Характеристики визначаються конкретною комбінацією методів

Наведені класи не є взаємозамінними універсальними рішеннями. Вибір методу залежить від структури середовища, доступної інформації, динамічності перешкод, обчислювальних обмежень та вимог до якості отриманого шляху [36; 37]. Гібридні підходи дозволяють розділяти глобальне планування та локальне реагування на зміни середовища, використовуючи для цих задач різні алгоритми.

Для мультиагентних систем до задачі планування додається необхідність узгодження руху декількох БПЛА. Планування місії в такому випадку може включати

розподіл завдань, визначення порядку відвідування цілей, побудову шляхів та їх просторово-часову координацію [46]. Тому незалежно допустимі маршрути окремих агентів не обов’язково утворюють допустимий спільний план: необхідно також враховувати можливі просторові й часові конфлікти між ними.

Окремою складовою маршрутного керування є представлення та виконання вже сформованої місії. У сучасних польотних контролерах місія може задаватися як впорядкована послідовність елементів, які визначають навігаційні цілі та додаткові дії. У PX4 попередньо сформована місія завантажується до польотного контролера й автоматично виконується в режимі Mission [29]. В ArduPilot режим AUTO аналогічно забезпечує виконання послідовності навігаційних та допоміжних команд [47].

Таке представлення місії не обмежується лише набором географічних координат. Навігаційні команди можуть визначати перехід до заданої точки, зліт або посадку, а додаткові команди та умови використовуються для керування іншими діями під час виконання місії [47]. На рівні протоколу MAVLink місія також представлена послідовністю елементів, кожен з яких містить команду, параметри та інформацію про систему координат [48]. Отже, маршрутний сценарій може бути формалізований як впорядкована послідовність етапів або команд із відповідними параметрами й умовами виконання.

Формування такого представлення та фізичне виконання маршруту є різними задачами. Опис місії визначає цільову послідовність дій, тоді як локальні контури польотного контролера повинні забезпечити переміщення апарата відповідно до заданих цілей та встановлених обмежень. Наприклад, у режимі AUTO ArduPilot виконує переходи між елементами місії з урахуванням параметрів швидкості, прискорення та підтримуваних способів згладжування руху [47].

Для мультиагентного моделювання це розмежування дозволяє окремо представляти маршрутний сценарій кожного БПЛА та процес його виконання локальним контуром керування. Такий підхід не вимагає, щоб усі функції планування, навігації та стабілізації були реалізовані одним алгоритмом: кожен рівень може бути представлений окремим функціональним компонентом із

визначеними вхідними та вихідними даними.

Незалежно від способу формування маршруту його виконання потребує інформації про поточний стан БПЛА відносно цільового стану. Отже, маршрутне керування охоплює не лише планування та формальне представлення місії, а й контроль проходження її етапів на основі доступної оцінки положення або зміщення апарата.

Розглянуті підходи забезпечують формалізоване представлення місії у вигляді послідовності навігаційних команд або етапів, однак їх виконання зазвичай передбачає використання координатної або позиційної інформації, доступної штатному навігаційному контуру. У межах цієї роботи подальший розвиток такого представлення полягає у поєднанні параметризованого маршрутного кроку з локальним контуром керування за відносним візуальним зміщенням. Для цього параметри активного маршрутного кроку перетворюються за допомогою механізму віртуального дрейфу у керуюче завдання для контуру візуальної стабілізації без необхідності використання глобальних координат БПЛА.

Таким чином, аналіз методів маршрутного керування показує необхідність розмежування задач планування, представлення місії та її безпосереднього виконання. Для мультиагентної системи додатково необхідно враховувати просторово-часову взаємозалежність маршрутів агентів, тоді як на рівні окремого БПЛА важливим є формалізоване представлення послідовності етапів маршруту та умов їх виконання.

У межах цієї дисертації автоматичний пошук або оптимізація шляху та генерація траєкторії не є окремим предметом дослідження. Основна увага приділяється формалізованому представленню наперед заданого маршрутного сценарію та його виконанню локальним контуром керування БПЛА.

1.5 Методи позиціювання та стабілізації БПЛА в умовах відсутності GPS

Під позиціюванням розуміється оцінювання положення БПЛА або його відносного зміщення за доступними сенсорними даними. Стабілізація розглядається як задача замкненого керування, у якій така оцінка використовується для утримання

БПЛА поблизу заданого положення або стану.

Для глобального позиціювання БПЛА широко застосовуються глобальні навігаційні супутникові системи (GNSS), зокрема GPS. Однак доступність і якість супутникових сигналів можуть знижуватися в приміщеннях, щільній міській забудові, поблизу конструкцій, що перекривають видимість супутників, а також за умов радіоперешкод. У таких ситуаціях для оцінювання стану БПЛА застосовуються інерціальні, візуальні, лазерні, радіочастотні та комбіновані підходи [49; 50].

У сучасних оглядах навігації в умовах недоступності GNSS розрізняють абсолютне та відносне позиціювання. Абсолютні методи визначають положення відносно глобальної або попередньо відомої системи координат, наприклад шляхом зіставлення сенсорних даних із картою чи відомими орієнтирами. Відносні методи оцінюють зміну положення відносно локально визначеного стану та можуть використовувати інерціальні, візуальні або інші бортові вимірювання [50].

Для задач локальної стабілізації таке розмежування має принципове значення. Якщо необхідно компенсувати відхилення від початкового положення або відносно спостережуваного орієнтира, глобальна географічна прив’язка не є обов’язковою за умови наявності придатної оцінки відносного зміщення. Водночас така локальна оцінка сама по собі не забезпечує глобальної навігації та потребує додаткової прив’язки, якщо цього вимагає польотне завдання.

Одним із базових джерел інформації про рух є інерціальний вимірювальний модуль. Вимірювання акселерометрів і гіроскопів використовуються для оцінювання орієнтації, швидкості та переміщення без зовнішньої інфраструктури. Обмеженням інерціального позиціювання є накопичення похибки через інтегрування шумів і систематичних похибок вимірювань, тому для тривалого оцінювання положення інерціальні дані зазвичай поєднують з іншими сенсорними джерелами [50].

Візуальні методи використовують послідовність зображень для оцінювання руху БПЛА відносно спостережуваного середовища. Для локального оцінювання горизонтального руху може застосовуватися оптичний потік. Flores та співавтори використали сенсор оптичного потоку для стабілізації квадрокоптера в умовах

відсутності GPS і показали, що зміна орієнтації апарата впливає на отриману оцінку руху; для зменшення цього впливу камера була встановлена на стабілізованому підвісі [51].

Для оцінювання просторового руху застосовуються візуальна одометрія та візуально-інерціальна одометрія [52]. Візуальна одометрія визначає зміну положення камери за послідовністю зображень, тоді як VIO поєднує візуальні спостереження з інерціальними вимірюваннями. Прикладом такого підходу є VINS-Mono, у якому камера й IMU використовуються в єдиній системі оцінювання стану [53-56].

Окремий клас становлять SLAM-методи, які одночасно оцінюють положення платформи та формують представлення навколишнього середовища. Наприклад, ORB-SLAM3 підтримує візуальні та візуально-інерціальні режими роботи для різних конфігурацій камер [57; 58]. Застосування камерних методів залежить від властивостей вхідних зображень; зміни освітлення, розмиття від руху та недостатня кількість характерних елементів сцени можуть погіршувати якість візуального оцінювання [49; 59].

Альтернативою візуальним засобам є лазерне позиціювання та картографування. LiDAR формує геометричні вимірювання навколишнього середовища і може поєднуватися з IMU для оцінювання руху. У FAST-LIO2 лазерні та інерціальні вимірювання використовуються для одночасного оцінювання стану та оновлення тривимірної карти [59]. Такий підхід не залежить від видимого освітлення так, як звичайна камера, але висуває власні вимоги до сенсорного обладнання та оброблення просторових даних.

Позиціювання без GNSS може також використовувати зовнішню радіочастотну інфраструктуру, зокрема UWB, Wi-Fi або інші радіосистеми [60]. Положення в таких системах визначається на основі характеристик поширення сигналу від опорних вузлів. Їх використання дозволяє сформувати локальну координатну систему, однак залежить від наявності відповідної інфраструктури з відомими параметрами або положенням [50].

Для отримання глобальної прив’язки без GNSS можуть застосовуватися також

методи зіставлення поточних сенсорних спостережень із попередньо відомою картою або еталонним представленням місцевості. Такі методи забезпечують оцінювання положення відносно відомого середовища, але їх застосування залежить від наявності актуальних опорних даних і можливості надійного зіставлення спостережуваної сцени з ними [50]. Порівняльні характеристики основних підходів наведено в Таблиці 1.4

Таблиця 1.4

Порівняння основних підходів до позиціювання БПЛА за відсутності GNSS

Підхід	**Основне джерело даних**	**Отримувана інформація**	**Характерні можливості**	**Основні обмеження**
Інерціальне оцінювання	IMU	Орієнтація, швидкість, відносне переміщення	Автономне оцінювання руху без зовнішньої інфраструктури	Накопичення похибки при інтегруванні
Оптичний потік	Камера	Локальний відносний рух	Формування сигналу зміщення або швидкості відносно спостережуваної поверхні	Залежність від руху камери та властивостей сцени
VO / VIO	Камера; камера + IMU	Відносна траєкторія та стан	Оцінювання просторового руху; у VIO використовується взаємодоповнення візуальних та інерціальних вимірювань	Залежність візуальної складової від якості зображень; вища алгоритмічна складність
SLAM	Камери та/або інші просторові сенсори	Положення та карта	Одночасна локалізація й формування представлення середовища	Залежність від сенсорних умов та додаткові обчислювальні витрати
LiDAR-інерціальне оцінювання	LiDAR + IMU	Просторовий рух і геометричне представлення	Локалізація за геометрією середовища без використання видимого освітлення	Потреба у спеціалізованому сенсорі та оброблені хмар точок
Радіочастотне позиціювання	UWB, Wi-Fi та ін.	Положення в локальній системі	Позиціювання відносно зовнішніх опорних вузлів	Залежність від попередньо розгорнутої інфраструктури

Локалізація за картою	Камери, висотоміри та інші сенсори + попередня карта	Абсолютне положення	Глобальна прив’язка без безпосереднього використання GNSS	Залежність від наявності та актуальності опорних даних

Наведені методи мають різні джерела похибок і вимоги до обладнання, тому в практичних навігаційних системах поширеним є поєднання декількох сенсорних модальностей. VINS-Mono поєднує камеру та IMU [53], ORB-SLAM3 підтримує візуально-інерціальне оцінювання [58], а FAST-LIO2 використовує LiDAR та IMU [59]. Огляд [50] також показує, що один тип сенсорів не забезпечує однакових характеристик у навігації в умовах недоступності GNSS.

Водночас для локального утримання положення не завжди необхідно відновлювати повну карту або визначати глобальні координати. Якщо система формує оцінку зміщення відносно опорного стану або спостережуваної області, така оцінка може використовуватися як інформаційний сигнал для замкненого контуру стабілізації. Практичну можливість використання камерної оцінки положення для керування БПЛА без GPS продемонстровано, зокрема, у роботах [51; 61]. У дослідженні Firdaus та співавторів координати, отримані засобами позиційного відстеження стереокамери ZED 2, використовувалися для проходження заданих маршрутів та утримання положення квадрокоптера в приміщенні [61].

На відміну від підходу [51], у якому для локальної стабілізації використовується оптичний потік із додатковою стабілізацією камери, та підходу [61], де для визначення положення застосовується просторове позиційне відстеження стереокамери, у запропонованому методі горизонтальні коригувальні впливи формуються безпосередньо за відносним візуальним зміщенням у площині зображення без відновлення глобальних координат БПЛА, тоді як вертикальне положення підтримується незалежним контуром висоти. Така структура дає змогу використовувати локальну візуальну оцінку як сигнал замкненого горизонтального контуру без побудови глобальної траєкторії або карти середовища.

Отже, вибір методу позиціювання БПЛА за відсутності GNSS визначається

необхідним типом координатної прив’язки, умовами середовища, доступними сенсорами та вимогами до обчислювальних ресурсів. Для задач локальної стабілізації окремий інтерес становлять відносні візуальні вимірювання, які дозволяють формувати сигнал відхилення від заданого стану без обов’язкового відновлення глобальних координат або повної карти середовища.

1.6 Методи керування БПЛА за візуальними даними та адаптивного PID-регулювання

Оцінка положення або зміщення БПЛА є інформаційною основою контуру керування, однак сама по собі не визначає спосіб формування керуючого впливу. Одним із класів методів, у яких інформація із зображення безпосередньо використовується у замкненому контурі, є візуальне сервокерування (visual servoing). У класичній класифікації розрізняють керування безпосередньо за параметрами зображення (image-based visual servoing, IBVS) та керування за оціненим просторовим положенням (position-based visual servoing, PBVS) [62].

У PBVS візуальні дані використовуються для оцінювання положення та орієнтації об’єкта або камери, після чого похибка формується у просторових координатах. В IBVS керуюча похибка визначається безпосередньо за характеристиками зображення, наприклад координатами візуальних ознак [62]. Для недопривідних літальних апаратів закон візуального керування повинен також враховувати особливості динаміки об’єкта, оскільки горизонтальний рух мультироторного БПЛА пов’язаний зі зміною його орієнтації [63].

Одним із напрямів є використання гомографічного представлення відносного положення камери та спостережуваної площини. У роботах [64; 65] інформація про гомографію між зображеннями використовується для формування візуальної похибки та побудови закону керування недопривідним БПЛА. Ці дослідження демонструють можливість використання параметрів зображення безпосередньо в контурі керування. При цьому алгоритм отримання візуальної оцінки та закон формування керуючого впливу залишаються різними функціональними задачами.

Нечітке адаптивне візуальне сервокерування квадрокоптером розглянуто Labed

та співавторами [66]. У запропонованому підході нечіткий регулятор динамічно змінює коефіцієнт підсилення IBVS у реальному часі з використанням похибки візуальних ознак та оціненої глибини. Такий підхід демонструє можливість адаптації параметрів закону керування безпосередньо у візуальному контурі за характеристиками сигналу зображення.

Поряд зі спеціалізованими законами візуального сервокерування в системах БПЛА застосовуються PID-регулятори. В огляді [67; 68] PID віднесено до основних класів методів керування квадрокоптерами поряд із лінійно-квадратичними, нелінійними, адаптивними та робастними підходами. Перевагами PID-регулювання є відносно проста структура та можливість реалізації в системах реального часу [69].

Для окремого каналу PID-закон формує керуючий вплив на основі поточної похибки, її накопиченого значення та швидкості зміни [76]. У контурі візуального керування така похибка не обов'язково має бути задана в метричних координатах: у разі IBVS вона може визначатися безпосередньо у просторі зображення [62].

Фіксовані коефіцієнти PID самі по собі не є недоліком. Для визначеного режиму роботи належним чином налаштований регулятор може забезпечувати необхідні характеристики керування. Однак за зміни режимів руху один набір коефіцієнтів може не забезпечувати однакових характеристик перехідного процесу в усіх станах системи. У роботі [69] ця проблема розглядається шляхом нечіткого планування коефіцієнтів PID залежно від похибки, швидкості її зміни та стану взаємопов'язаних контурів.

Планування коефіцієнтів є лише одним із механізмів адаптації [75]. Boubakir та співавтори запропонували адаптивний PID-подібний регулятор, параметри якого змінюються в процесі роботи відповідно до адаптивного закону [70]. Spencer та співавтори розробили автоматичний настроювач PID для мультироторів, у якому коефіцієнти коригуються за фактичною динамічною реакцією апарата під час польоту [71]. У розглянутих роботах термін "адаптивне PID-регулювання" використовується для підходів із різними механізмами зміни параметрів регулятора, які можуть ґрунтуватися на режимних змінних, оціненій динаміці або результатах автоматичного

налаштування.

Для керування за візуальними даними додаткового значення набувають характеристики самого сигналу зворотного зв'язку. У роботі [73] оцінка достовірності візуального спостереження використовується для зміни параметрів оцінювання та керування під час автономної посадки. Цей підхід не є адаптивним PID-регулюванням за візуальним зміщенням, однак демонструє можливість безпосереднього врахування якості вхідної візуальної інформації під час формування реакції системи керування.

Іншим фактором є режим виконуваного маневру. У сучасних системах візуального керування застосовуються перемикання режимів і зміна параметрів залежно від поточного стану системи. Зокрема, у роботі [72] параметри або структура візуального керування змінюються відповідно до умов виконання маневру та характеристик окремих каналів. Такі підходи показують, що однакові параметри керування не обов'язково повинні використовуватися для всіх станів візуально керованого руху.

Окремим напрямом адаптивного PID-керування БПЛА є відстеження заданої траєкторії зі зміною параметрів регулятора відповідно до поточного стану системи. Jia та співавтори [74] запропонували каскадну систему позиційного та орієнтаційного PID-керування квадрокоптером, у якій нечітка логіка використовується для адаптивного налаштування коефіцієнтів PID-регулятора в реальному часі. Вхідними величинами механізму адаптації є похибка між заданим і фактичним положенням БПЛА та швидкість її зміни. Додатково RBF-нейронна мережа використовується для оперативного коригування коефіцієнтів PID-регулятора та компенсації впливу зовнішніх збурень. Працездатність підходу перевірено під час моделювання відстеження заданих траєкторій.

Основні механізми зміни параметрів керування, представлені в розглянутих роботах, систематизовано в Таблиці 1.5.

Таблиця 1.5

Підходи до адаптації керування БПЛА та врахування характеристик візуального зворотного зв'язку

Підхід	**Адаптовані параметри**	**Ознаки, що враховуються**	**Призначення**
Планування PID-коефіцієнтів	K_p, K_i, K_d	Похибка, її зміна, стан пов'язаних каналів	Узгодження параметрів регулятора з режимом роботи
Адаптивний PID-подібний закон	Параметри регулятора	Динамічна реакція системи	Компенсація невизначеності параметрів
Автоматичне налаштування PID	Коефіцієнти регулятора	Реакція мультиротора під час роботи	Автоматизація процедури налаштування
Адаптивне візуальне сервокерування	Параметри закону керування	Візуальні ознаки та параметрична невизначеність	Адаптація безпосередньо у візуальному контурі
Урахування достовірності візуальних даних	Параметри оцінювання та керування	Оцінка якості візуального спостереження	Зменшення впливу недостовірних вимірювань
Урахування обмежень керування	Параметри оцінювання та керування	Стан насичення керуючих каналів	Забезпечення роботи в допустимому діапазоні

Розглянуті підходи вирішують різні задачі й не є безпосередньо взаємозамінними. Одні методи змінюють PID-коефіцієнти за величиною похибки або динамічною реакцією апарата, інші адаптують закон візуального сервокерування, враховують достовірність спостережень або фізичні обмеження керуючих сигналів. Тому результати таких робіт не можуть безпосередньо порівнюватися як альтернативні реалізації одного регулятора. Порівняння виконується за структурними ознаками механізмів керування, а не за кількісними результатами експериментів, оскільки розглянуті роботи відрізняються постановкою задач, експериментальними умовами та критеріями оцінювання.

Для конкретизації відмінностей між розглянутими підходами та методом, що розробляється в дисертаційній роботі, доцільно зіставити найближчі аналоги не лише

за загальним типом керування, а й за способом зміни параметрів регулятора, використанням візуального зворотного зв’язку, урахуванням режиму руху, характеристик маршрутного завдання та обмежень керуючих каналів. Таке зіставлення наведено в таблиці 1.6.

Таблиця 1.6

Порівняння розглянутих підходів, найближчих за окремими ознаками до запропонованого методу

Підхід	**Принцип адаптації**	**Фазова/режимна зміна**	**Якість візуального оцінювання**	**Параметри маршрутного кроку та cross-track**	**Урахування насичення**
Labed et al. [66]	Нечітке динамічне налаштування коефіцієнта підсилення IBVS за похибкою візуальних ознак та оціненою глибиною	-	-	-	-
Melo et al. [69]	Нечітке планування PID-коефіцієнтів за похибкою, її зміною та станом пов’язаних контурів	без явної фазової класифікації	-	-	-
Boubakir et al. [70]	Безперервна адаптація параметрів PID-подібного регулятора відносно	-	-	-	-

	еталонної поведінки				
Spencer et al. [71]	Автоматичне налаштування PID за динамічною реакцією БПЛА під час навчального польоту	-	-	Траєкторія використовується для налаштування, але не як змінна фазового scheduler-а	-
Hazelaar et al. [72]	Адаптивне візуальне сервокерування з перемиканням режимів під час посадки	+	Використовуються характеристики візуального сигналу	-	-
Zhou et al. [73]	Зміна параметрів оцінювання та predictive control за показником надійності спостереження	+	+	-	не передбачено незалежного врахування насичення каналів PID-регулятора
Jia et al. [74]	Нечітке online-налаштування PID за позиційною похибкою та швидкістю її зміни з RBF-компенсацією	без явної фазової класифікації	-	задана траєкторія використовується для формування позиційної похибки; маршрутні кроки та cross-track не враховуються	-
Запропонований метод	Фазове формування K_p, K_i, K_d на кожному циклі з подальшим осьовим коригуванням	+	c_k	номер/напрям маршрутного кроку та cross-track	незалежно для двох координатних каналів

Порівняльний аналіз показує, що окремі складові запропонованого підходу мають аналоги в існуючих методах керування БПЛА. Зокрема, у відомих роботах застосовуються адаптивна зміна параметрів PID-регуляторів [69–71], нечітко-нейромережеве адаптивне PID-керування під час відстеження заданої траєкторії [74], адаптивне та режимне візуальне сервокерування [66; 72], а також урахування якості візуального спостереження [73]. Отже, наукова новизна розроблюваного методу не пов'язується з використанням кожного із зазначених механізмів окремо.

Відмінною особливістю запропонованого методу є узгоджене поєднання зазначених механізмів у єдиному механізмі формування коефіцієнтів позиційного PID-регулятора. На кожному циклі керування виконується повторне визначення поточної фази позиційного контуру за характеристиками позиційної похибки та станом керуючих каналів, після чого формується базова фазова конфігурація коефіцієнтів K_p, K_i та K_d. Отримані коефіцієнти додатково коригуються окремо для координатних каналів з урахуванням параметрів поточного маршрутного кроку, співвідношення складових позиційної похибки та поперечного відхилення від маршруту. Насичення керуючих впливів враховується незалежно за двома координатними каналами. Метод також передбачає використання нормованого показника якості візуального оцінювання c_k для формування обмежувальної фази LOW_CONFIDENCE.

У межах проведеного аналізу джерел [66; 69–74] не виявлено підходу до адаптивного PID-керування положенням БПЛА за візуальним зміщенням, у якому одночасно поєднувалися б повторна фазова класифікація стану позиційного контуру на кожному циклі керування, фазове формування коефіцієнтів PID-регулятора, їх подальше маршрутне та осьове коригування з урахуванням поперечного відхилення, а також незалежне врахування насичення керуючих каналів. Саме зазначена сукупність взаємопов'язаних механізмів визначає відмінність розробленого методу від розглянутих аналогів.

Отже, аналіз існуючих методів показав, що адаптація параметрів регулятора,

використання візуального зворотного зв’язку, урахування якості спостережень, перемикання режимів та обмеження керуючих впливів досліджувалися в попередніх роботах переважно як окремі або частково поєднані механізми. За результатами проведеного аналізу обґрунтовано доцільність розроблення єдиного механізму адаптації PID-коефіцієнтів, що враховує поточний режим позиційного контуру, характеристики маршрутного кроку, поперечне відхилення та стан окремих керуючих каналів. Розробленню такого методу присвячено четвертий розділ дисертаційної роботи.

1.7 Постановка задач дослідження

Проведений аналіз показав, що дослідження мультиагентних систем БПЛА потребує узгодженого врахування процесів на рівні окремого апарата, його локального керування, виконання маршрутного сценарію та взаємодії з іншими агентами. Склад експериментальної моделі при цьому має визначатися властивостями системи, що підлягають дослідженню, оскільки надмірна деталізація збільшує обчислювальні витрати, а недостатня може виключати суттєві для експерименту процеси.

Аналіз програмного та напівнатурного моделювання показав, що SIL і HIL забезпечують різний ступінь залучення реального обладнання до експериментального контуру. SIL спрощує формування конфігурацій із декількома модельованими БПЛА, для яких використовуються програмні реалізації польотних контролерів, тоді як HIL дає змогу перевіряти виконання польотного програмного забезпечення на реальній апаратній платформі. Для мультиагентної системи використання окремого фізичного польотного контролера для кожного агента збільшує кількість необхідного обладнання та вимоги до організації інформаційного обміну й синхронізації [31]. Це обґрунтовує дослідження архітектури, у якій програмні та апаратні компоненти можуть спільно функціонувати в одному середовищі моделювання.

Аналіз методів маршрутного керування показав необхідність розмежування задач планування, представлення місії та її безпосереднього виконання. Для

мультиагентної системи маршрут окремого БПЛА має бути представлений у формі, придатній для автономного послідовного виконання, із визначеними маршрутними етапами, їх параметрами та умовами завершення. При цьому взаємодія декількох агентів додатково потребує врахування їхніх просторово-часових зв’язків і можливості координації маршрутів [46].

За відсутності GNSS локальне керування БПЛА може спиратися на відносні сенсорні оцінки. Зокрема, візуальне зміщення може використовуватися як сигнал зворотного зв’язку для локального керування положенням без обов’язкового визначення глобальних координат. Водночас характеристики такого сигналу можуть змінюватися в процесі руху, а керуючі канали мають фізичні обмеження. Аналіз методів адаптивного керування показав, що параметри PID-регулятора можуть змінюватися залежно від позиційної похибки, швидкості її зміни та інших характеристик стану системи [69; 74], тоді як у візуальних контурах адаптація може виконуватися безпосередньо за характеристиками візуальної похибки [66].

Отже, результати проведеного аналізу визначають необхідність дослідження моделей і методів, які поєднують напівнатурне моделювання мультиагентної системи, формалізоване виконання маршрутних сценаріїв та локальне керування окремими БПЛА-агентами за візуальними даними.

Метою дисертаційної роботи є підвищення ефективності керування мультиагентними системами БПЛА при їх розгортанні для виконання місії за рахунок розроблення моделей і методів напівнатурного моделювання процесів керування, що забезпечують виконання маршрутних сценаріїв і стабілізацію положення БПЛА-агентів за відео даними в умовах відсутності GPS.

Для досягнення мети необхідно вирішити такі задачі дослідження:

- провести аналіз існуючих моделей, методів і засобів моделювання та керування мультиагентними системами БПЛА, визначити їх особливості та обмеження;
- розробити архітектуру напівнатурного моделювання мультиагентної системи БПЛА, що забезпечує спільне функціонування програмних моделей БПЛА та реальних апаратних компонентів в єдиному середовищі моделювання;

- розробити агентно-орієнтовану модель виконання маршрутних сценаріїв БПЛА, призначену для використання в мультиагентній системі;
- розробити метод просторової стабілізації та керування положенням БПЛА в умовах відсутності GPS з використанням оцінки візуального зміщення;
- розробити фазовий метод адаптивного PID-керування положенням БПЛА за візуальним зміщенням;
- провести експериментальну перевірку розроблених моделей і методів та оцінити їх характеристики за визначеними показниками.

1.8 Висновки до розділу I

За результатами аналізу методів і засобів моделювання та керування мультиагентними системами БПЛА сформульовано такі висновки.

1. Моделювання мультиагентної системи БПЛА потребує врахування взаємозв'язку між процесами на рівні окремого апарата, локальним керуванням, сенсорним забезпеченням, виконанням маршрутних сценаріїв та взаємодією агентів. Існуючі засоби моделювання орієнтовані на різні складові цих процесів, тому склад і рівень деталізації моделі доцільно визначати відповідно до мети конкретного експерименту.

2. Програмне SIL-моделювання та напівнатурне HIL-моделювання забезпечують різний ступінь залучення реального обладнання до експериментального контуру. Програмна конфігурація полегшує формування систем із декількома модельованими агентами без відповідного збільшення кількості фізичних польотних контролерів, тоді як HIL дає змогу включати реальні апаратні компоненти до модельованого контуру. Для мультиагентних систем це обґрунтовує дослідження конфігурацій, у яких програмні та апаратні компоненти функціонують у спільному середовищі моделювання.

3. У широкому розумінні маршрутне керування БПЛА охоплює не лише побудову шляху та траєкторії, а й формалізоване представлення та виконання місії. Для мультиагентної системи важливим є представлення окремого БПЛА як автономного агента з власним станом і маршрутним сценарієм, а також визначення

послідовності етапів маршруту й умов їх завершення.

4. За відсутності GNSS для оцінювання стану БПЛА можуть використовуватися інерціальні, візуальні, лазерні, радіочастотні та комбіновані підходи. Для локальної стабілізації глобальна координатна прив'язка не завжди є необхідною, якщо доступна оцінка відносного зміщення. Візуальні дані можуть забезпечувати такий сигнал зворотного зв'язку, при цьому формування оцінки зміщення та формування керуючого впливу є окремими функціональними задачами.

5. Аналіз методів візуального сервокерування та PID-регулювання показав, що за зміни режимів руху один фіксований набір коефіцієнтів може не забезпечувати однакових характеристик керування в усіх станах системи. У наявних підходах параметри керування змінюються залежно від похибки, динамічного режиму та інших характеристик системи, а для візуального керування додаткового значення набувають характеристики сигналу зворотного зв'язку й обмеження керуючих каналів. Це створює передумови для дослідження адаптивного PID-регулювання положення БПЛА за візуальним зміщенням.

Результати досліджень, приведених в розділі, опубліковані в роботах [81; 82; 91; 94; 95; 100].

РОЗДІЛ ІІ. МОДЕЛІ ТА АРХІТЕКТУРА НАПІВНАТУРНОГО МОДЕЛЮВАННЯ МУЛЬТИАГЕНТНИХ СИСТЕМ БПЛА

2.1 Принципи побудови середовища напівнатурного моделювання мультиагентної системи БПЛА

Дослідження алгоритмів керування мультиагентними системами БПЛА потребує середовища, яке дає змогу відтворювати одночасне функціонування декількох літальних апаратів, їх взаємодію із зовнішнім середовищем та роботу бортових засобів керування. Проведення таких досліджень безпосередньо на реальних БПЛА пов’язане зі значними матеріальними витратами та ризиком пошкодження обладнання, особливо на етапах розроблення і налагодження алгоритмів. Тому доцільним є використання програмного та напівнатурного моделювання, які дають змогу перевіряти функціонування системи керування до переходу до натурних випробувань.

При програмному моделюванні всі складові модулю керування, включаючи автопілот і модель динаміки БПЛА, реалізуються програмно. Такий підхід спрощує відтворення експериментів, дає змогу одночасно моделювати декілька агентів та не потребує окремого апаратного обладнання для кожного БПЛА [77]. Разом із тим програмна модель не відтворює повною мірою особливості функціонування реального польотного контролера, пов’язані з апаратним виконанням алгоритмів, обміном даними та часовими характеристиками контуру керування.

Напівнатурне моделювання за схемою Hardware-in-the-Loop передбачає включення реального апаратного компонента системи керування до замкненого контуру з програмною моделлю БПЛА та зовнішнього середовища. У такому контурі реальний польотний контролер виконує алгоритми керування, використовуючи дані віртуальних сенсорів, сформовані середовищем моделювання, а отримані від контролера керуючі впливи повертаються до моделі літального апарата. Таким чином забезпечується взаємодія реального апаратного контролера з моделлю БПЛА без необхідності проведення реального польоту.

Для мультиагентної системи використання окремого апаратного польотного контролера для кожного модельованого БПЛА підвищує складність та вартість експериментального стенда. Водночас використання лише програмних автопілотів спрощує масштабування системи, але не дає змоги досліджувати особливості роботи апаратного контролера в загальному контурі. З огляду на це в роботі використано комбінований підхід, у якому частина агентів може функціонувати із залученням реального польотного контролера, а інші агенти реалізуються програмно.

Основою такого підходу є спільне модельоване середовище, у якому кожен БПЛА представлений окремою моделлю, а усі агенти функціонують у спільному модельованому зовнішньому середовищі, а динаміка кожного БПЛА відтворюється його окремою моделлю. Для реалізації середовища використано симулятор Gazebo, що забезпечує моделювання динаміки БПЛА, зовнішньої сцени та формування даних окремих віртуальних сенсорів.

Отже, принцип побудови гібридного середовища полягає у спільному використанні моделей БПЛА та зовнішнього середовища з апаратними і програмними реалізаціями компонентів керування.

2.2 Архітектура гібридного HIL/SIL середовища моделювання

На основі визначених принципів побудовано гібридну HIL/SIL архітектуру середовища моделювання мультиагентної системи БПЛА. У документації ArduPilot програмна реалізація автопілота в середовищі моделювання також позначається абревіатурою SITL (Software-In-The-Loop); у цій роботі для узгодженості з позначенням HIL використовується скорочення SIL. Структуру розробленої архітектури наведено на рисунку 2.1.

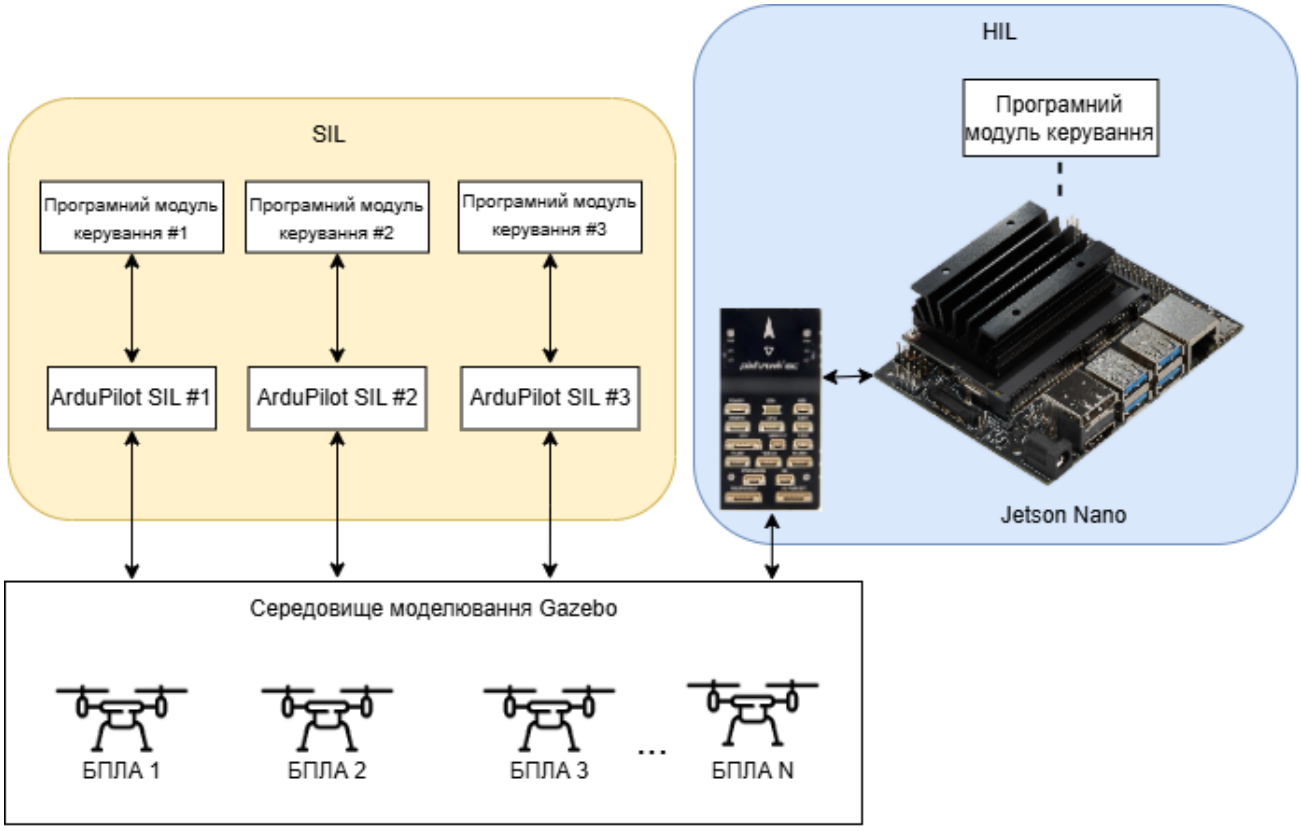


Рис. 2.1. Архітектура гібридного HIL/SIL середовища моделювання мультиагентної системи БПЛА.

Архітектура містить середовище моделювання, рівень польотних контролерів та засоби програмного керування. Середовище моделювання забезпечує формування моделей літальних апаратів, відтворення їх динаміки та зовнішньої сцени, а сенсорна інформація формується засобами модельованого середовища і симуляційної підсистеми автопілота. Рівень польотних контролерів представлений апаратною HIL- та програмними SIL-реалізаціями.

У HIL-реалізації функції польотного контролера виконує фізичний Pixhawk із програмним забезпеченням ArduPilot. На цьому рівні виконуються низькорівневі функції стабілізації та керування положенням БПЛА.

До апаратної конфігурації також входить одноплатний обчислювальний модуль Jetson Nano, на якому виконується розроблений програмний модуль керування, який формує команди керування рухом БПЛА, програмно відтворюючи керуючі дії

оператора, тоді як низькорівневе виконання команд забезпечується польотним контролером. Такий розподіл забезпечує відокремлення формування високорівневих команд від їх виконання засобами польотного контролера.

Для SIL-реалізації аналогічні високорівневі команди формуються програмним модулем, що виконується на комп'ютері моделювання. Функції польотного контролера для цих агентів виконують окремі програмні екземпляри ArduPilot. Апаратна HIL- та програмні SIL-реалізації взаємодіють із відповідними моделями БПЛА у спільному середовищі Gazebo.

Архітектура допускає зміну співвідношення HIL- та SIL-агентів залежно від конфігурації експериментального середовища та наявних апаратних засобів.

2.3 Взаємодія програмних та апаратних компонентів середовища моделювання

Функціонування гібридного середовища моделювання ґрунтується на двонапрямленому обміні даними між моделями БПЛА та відповідними польотними контролерами. Загальний цикл взаємодії передбачає формування стану БПЛА в середовищі моделювання, передавання сенсорних даних до автопілота, обчислення керуючих впливів та їх повернення до моделі літального апарата [80].

У HIL-конфігурації середовище Gazebo забезпечує моделювання динамічного стану БПЛА та формування модельованих даних, необхідних для роботи польотного контролера. Сформовані дані передаються до реального польотного контролера Pixhawk, на якому виконується програмне забезпечення ArduPilot. Польотний контролер обробляє отриману інформацію та формує керуючі впливи відповідно до реалізованих алгоритмів стабілізації і керування. Отримані сигнали повертаються до Gazebo та використовуються для оновлення стану моделі БПЛА. Після перерахунку динаміки формується новий набір сенсорних даних, і цикл повторюється. Таким чином утворюється замкнений контур напівнатурного моделювання.

У SIL-конфігурації логіка інформаційного обміну зберігається, однак функції польотного контролера виконує програмний екземпляр ArduPilot на комп'ютері моделювання. Це дає змогу використовувати HIL- і SIL-реалізації в межах спільної

модельованої сцени.

У початковій реалізації експериментального середовища інтеграція компонентів здійснювалася із застосуванням ROS та MAVROS [78]. Gazebo передавав стан моделей БПЛА і дані віртуальних сенсорів через ROS-топіки. MAVROS використовувався як програмний міст між ROS та MAVLink-сумісними автопілотами і забезпечував двонапрямлене перетворення та передавання повідомлень між середовищем моделювання і польотними контролерами [79].

Для апаратного HIL-агента взаємодія з Pixhawk здійснювалася через послідовний інтерфейс, тоді як програмні екземпляри ArduPilot використовували мережеві з'єднання. Це забезпечувало одночасне підключення до середовища фізичного польотного контролера Pixhawk та програмних екземплярів ArduPilot.

Взаємодію компонентів контуру керування можна подати як послідовність чотирьох основних етапів: формування стану моделі БПЛА та модельованих даних, необхідних для роботи польотного контролера; передавання цих даних до польотного контролера; обчислення керуючих впливів засобами ArduPilot; застосування отриманих команд до моделі БПЛА та оновлення її стану. Схему інформаційної взаємодії компонентів середовища моделювання наведено на рисунку 2.2.

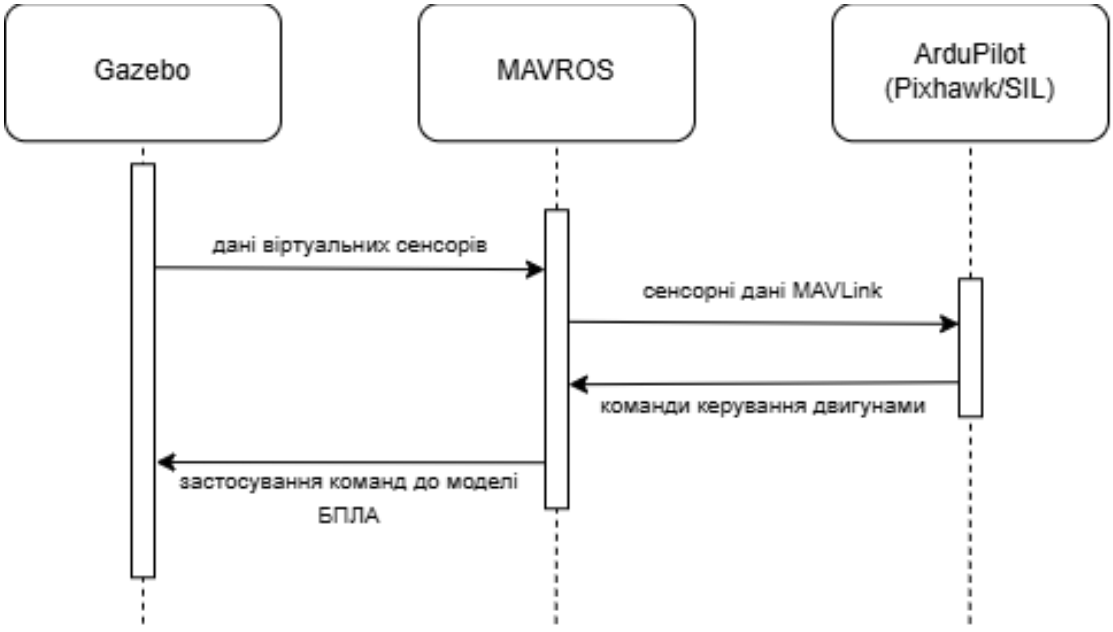


Рис. 2.2. Схема інформаційної взаємодії компонентів середовища моделювання.

Використання ROS і MAVROS у цій конфігурації було пов’язане з конкретною реалізацією інтеграційного шару та не визначало загальний принцип побудови запропонованої архітектури. Основною вимогою до взаємодії компонентів є забезпечення двонапрямленого обміну між середовищем моделювання та польотним контролером із передаванням сенсорних даних у прямому напрямку і керуючих впливів у зворотному.

2.4 Організація інформаційних потоків агентів у HIL/SIL середовищі

Функціонування декількох БПЛА в єдиному середовищі моделювання потребує розмежування інформаційних потоків окремих агентів. Для кожного БПЛА необхідно забезпечити незалежне передавання даних віртуальних сенсорів і телеметрії до відповідного польотного контролера, а також повернення сформованих керуючих команд до пов’язаної з ним моделі. Така організація унеможливлює змішування даних різних агентів під час їх одночасної роботи.

Розмежування інформаційних потоків агентів реалізовано в межах використовуваної ROS/MAVROS-інфраструктури шляхом призначення окремих просторів імен, каналів підключення та ідентифікаторів для кожного БПЛА.

Для розмежування інформаційних потоків окремих агентів використовувалися окремі простори імен MAVROS. У конфігурації з чотирма БПЛА моделям були призначені простори імен /uav1, /uav2, /uav3 та /uav4. Простір /uav1 відповідав БПЛА, керованому фізичним польотним контролером Pixhawk у режимі HIL, тоді як /uav2, /uav3 та /uav4 використовувалися для моделей із програмними екземплярами ArduPilot у режимі SIL. Після запуску середовища для кожного БПЛА окремо публікувалися дані одометрії модельованого БПЛА та дані віртуальних сенсорів.

Фізичний польотний контролер Pixhawk підключався до MAVROS через послідовний інтерфейс, тоді як програмні екземпляри ArduPilot використовували окремі мережеві UDP-з’єднання. Таким чином, для HIL- та SIL-агентів застосовувався спільний принцип інформаційної взаємодії, але відрізнявся спосіб підключення польотного контролера. Для програмних екземплярів ArduPilot також використовувалися окремі параметри підключення та ідентифікатори MAVLink, що

забезпечувало одночасне обслуговування декількох агентів у межах одного експериментального середовища.

Додавання програмного агента передбачає створення окремої моделі БПЛА, програмного екземпляра ArduPilot та незалежного каналу інформаційної взаємодії. У дослідженій конфігурації такий принцип розмежування забезпечив одночасне функціонування трьох SIL-агентів разом з одним HIL-агентом.

2.5 Реалізація експериментального HIL/SIL середовища

Для експериментальної перевірки розробленої архітектури було створено гібридне HIL/SIL середовище, у якому одночасно функціонували чотири моделі квадрокоптерів. Один БПЛА керувався апаратним польотним контролером Pixhawk у режимі HIL, а для трьох інших використовувалися окремі програмні екземпляри ArduPilot у режимі SIL. Така конфігурація дала змогу поєднати апаратну реалізацію польотного контролера з програмно реалізованими агентами в межах одного модельованого середовища.

Апаратна частина експериментального середовища включала персональний комп’ютер із процесором Intel Core i7 під керуванням Ubuntu 20.04 LTS, польотний контролер Pixhawk 6C із програмним забезпеченням ArduCopter 4.3 та одноплатний обчислювальний модуль Jetson Nano. Польотний контролер Pixhawk підключався до комп’ютера через USB.

Програмна частина експериментального середовища була побудована на основі Gazebo 11, ROS Noetic, MAVROS та ArduPilot. Для SIL-агентів запускалися незалежні програмні екземпляри ArduPilot з окремими UDP-з’єднаннями, тоді як апаратний Pixhawk підключався до MAVROS через послідовний інтерфейс.

Для представлення БПЛА у модельованому середовищі використано базову SDF-модель квадрокоптера Iris. Модель містить основну ланку корпуса, чотири ротори, інерціальний сенсор та допоміжні ланки, пов’язані між собою засобами опису моделі Gazebo. Геометричні, інерційні та аеродинамічні параметри базової моделі в межах проведеного дослідження окремо не визначалися та не оптимізувалися. Модель Iris використовувалася як готова фізична основа для реалізації та

дослідження HIL/SIL-конфігурацій БПЛА.

Основною ланкою SDF-моделі є корпус квадрокоптера base_link, відносно якого визначено положення інших компонентів моделі. Для основної ланки задано масу 1,5 кг, а її колізійну геометрію представлено прямокутним паралелепіпедом розміром 0,47 × 0,47 × 0,23 м. Чотири ротори реалізовано як окремі ланки, приєднані до корпуса моделі. Така структура забезпечує окремий опис геометричних та фізичних властивостей корпуса, роторів і сенсорних компонентів БПЛА.

Структурну організацію модельованої частини середовища деталізовано за наявною SDF-конфігурацією базової моделі БПЛА та модельованої сцени.

Структура модельованого середовища задавалася засобами формату SDF, який використовувався для опису складу сцени, фізичних параметрів середовища та моделей окремих об’єктів. У наявній конфігурації Gazebo для розрахунку динаміки використовується фізичний рушій ODE, а прискорення вільного падіння задається вектором $(0{,}0, -9{,}8)$м/с². Опис модельованого світу та опис БПЛА зберігаються окремо, що дає змогу включати необхідні екземпляри моделі літального апарата до складу конкретної експериментальної сцени.

Формування сенсорної інформації в програмній конфігурації розподілялося між середовищем Gazebo та симуляційними засобами ArduPilot. У SDF-моделі БПЛА безпосередньо визначено інерціальний сенсор, покази якого формуються відповідно до руху моделі та використовуються польотним контролером. Водночас конфігурація ArduPilot містить параметри програмного моделювання окремих бортових сенсорів, зокрема інерціальних вимірювальних модулів, барометрів, GPS-приймача, магнітометрів та дальноміра. Таким чином, сенсорна підсистема модельованого БПЛА не обмежується сенсорами, безпосередньо заданими у SDF-описі.

Керування HIL-агентом здійснювалося розробленим програмним модулем, який забезпечував формування команд керування БПЛА та обмін телеметричною інформацією з польотним контролером за протоколом MAVLink. Програмний модуль встановлював з’єднання з автопілотом, виконував контроль його стану та забезпечував передавання команд без безпосереднього формування низькорівневих

сигналів керування двигунами. Виконання алгоритмів стабілізації, перетворення отриманих команд у керуючі впливи та формування сигналів для двигунів залишалося функцією ArduPilot.

Для SIL-агентів аналогічний програмний модуль виконувався на комп'ютері моделювання та формував команди для відповідних програмних екземплярів ArduPilot.

До функцій програмного модуля належали переведення польотного контролера в стан готовності до польоту, зміна режиму роботи, формування команди зльоту та команд переміщення БПЛА, а також отримання телеметричних даних про положення, орієнтацію та стан апарата. Для задавання просторового переміщення використовувалися MAVLink-повідомлення з цільовими параметрами руху в локальній системі координат. Такий спосіб керування програмно відтворював функції, які під час ручного польоту виконує оператор, при цьому низькорівневі контури керування залишалися реалізованими засобами штатного автопілота.

Формування мультиагентної конфігурації передбачало використання окремої моделі БПЛА та відповідного екземпляра польотного контролера для кожного агента. Для програмних агентів запускалися незалежні екземпляри ArduPilot, а апаратний агент використовував фізичний контролер Pixhawk. Розмежування каналів інформаційного обміну забезпечувалося окремими параметрами підключення та ідентифікаторами агентів, що давало змогу одночасно використовувати декілька БПЛА в межах спільного модельованого середовища.

Експериментальна сцена, використана для перевірки гібридної HIL/SIL архітектури, містила чотири моделі квадрокоптерів та статичні об'єкти оточення. Моделі БПЛА мали різні початкові координати, що забезпечувало їх просторове розділення під час одночасного виконання експериментального сценарію. Для функціонування польотних контролерів у модельованому середовищі формувалися необхідні сенсорні дані про стан БПЛА. У сцені також були присутні статичні перешкоди, зокрема стіни та будівлі. Загальний вигляд модельованої сцени з чотирма БПЛА та статичними об'єктами оточення наведено на рисунку 2.3.

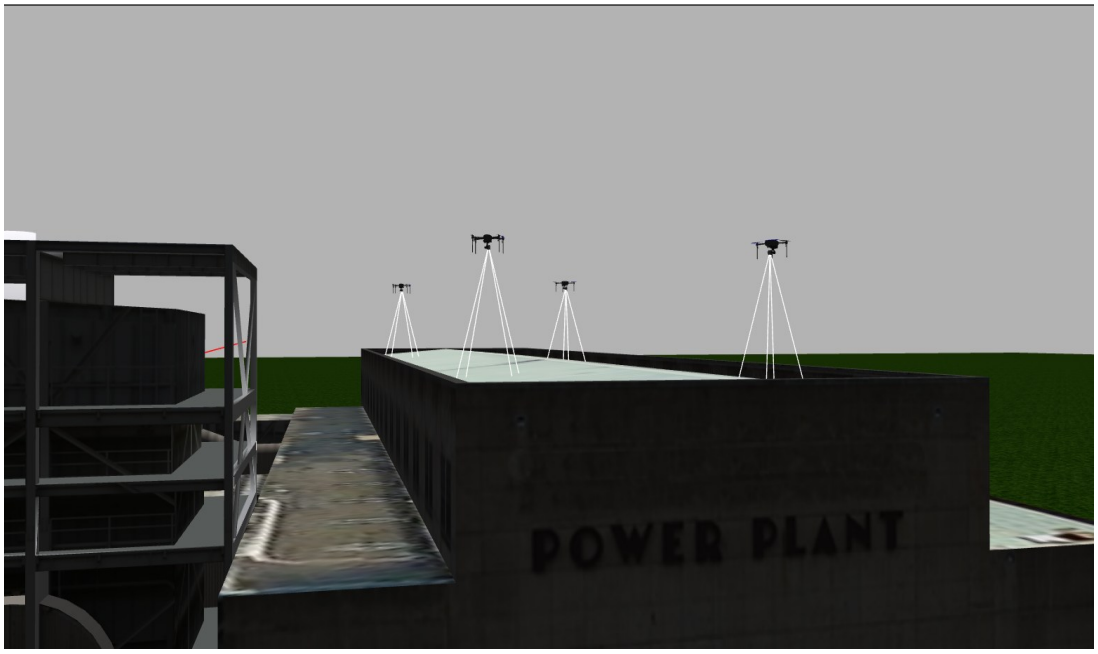


Рис. 2.3. Модельована сцена гібридного HIL/SIL середовища з чотирма БПЛА.

Конфігурація модельованого середовища та окремих БПЛА задавалася набором взаємопов'язаних файлів опису. Файл модельованого світу визначав склад сцени та підключення моделей, тоді як SDF-файли окремих БПЛА містили опис їх структури, фізичних компонентів і сенсорів. Параметри програмних екземплярів ArduPilot задавалися окремими файлами конфігурації. Такий розподіл давав змогу змінювати параметри моделей та склад експериментального середовища без модифікації програмної логіки керування. Фрагменти наявних конфігураційних описів базової моделі БПЛА та модельованого середовища наведено в додатку Г.

Перед початком експериментів перевірялися наявність обміну MAVLink-повідомленнями, коректність публікації ROS-топіків і доступність усіх чотирьох моделей БПЛА в середовищі Gazebo.

2.6 Експериментальне дослідження функціонування гібридної HIL/SIL архітектури

Для оцінювання працездатності реалізованого HIL/SIL середовища проведено експеримент з одночасним функціонуванням чотирьох БПЛА: одного HIL- та трьох SIL-агентів. Під час експерименту БПЛА виконували зліт, переміщення до заданих

положень та подальшу стабілізацію положення.

Задані положення БПЛА розташовувалися на відстані приблизно 3 м одна від одної. Після зльоту апарати виконували переміщення до заданих положень і переходили до режиму їх утримання на висоті близько 50 м. Статичні об’єкти модельованого середовища використовувалися як елементи сцени та не були предметом окремого дослідження, оскільки алгоритми активного уникнення перешкод у межах цього експерименту не застосовувалися. Оцінювання було спрямоване на перевірку працездатності гібридного HIL/SIL контуру та порівняння поведінки апаратної і програмної реалізацій польотного контролера під час переміщення та стабілізації.

Під час виконання експериментального сценарію реєструвалися координати БПЛА вздовж осей X, Y та Z. За зміною координат оцінювалися процес переміщення апарата до заданої позиції та подальший перехід до режиму її утримання. Характер зміни просторових координат одного з БПЛА під час виконання переміщення наведено на рисунку 2.4.

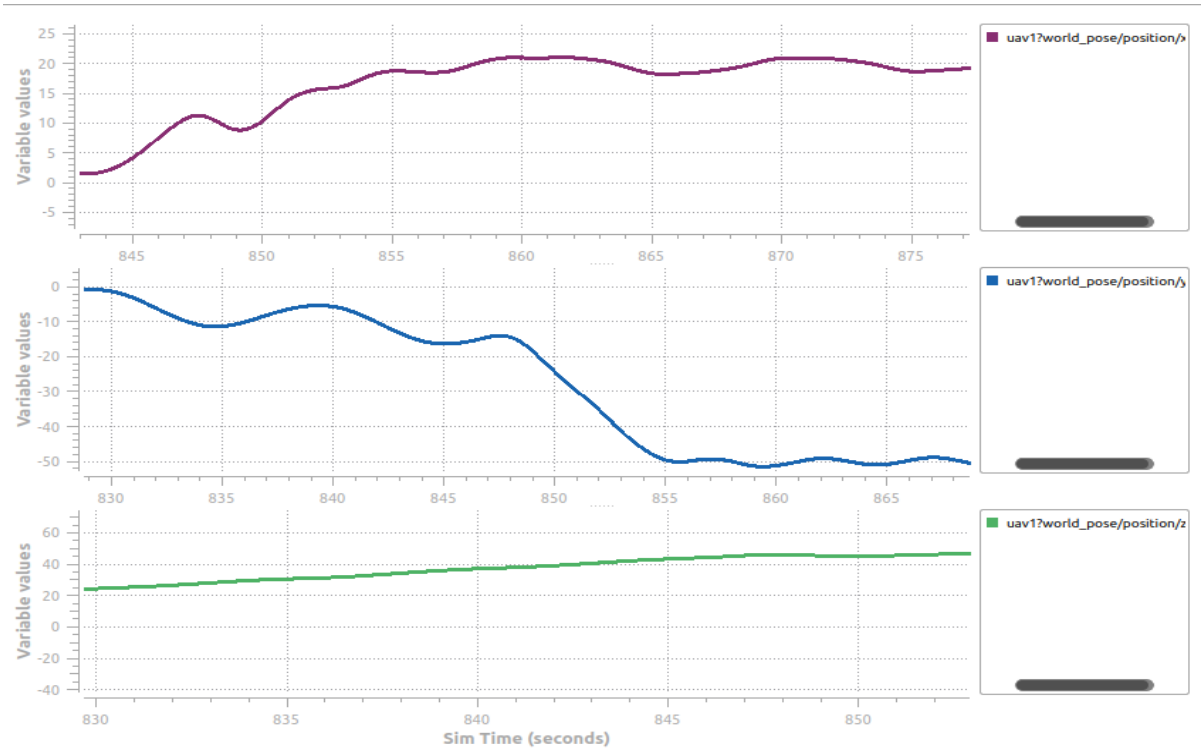


Рис. 2.4. Зміна координат БПЛА під час переміщення до заданої позиції.

Після досягнення заданого положення оцінювався характер її утримання. Для HIL- та SIL-реалізацій координати БПЛА залишалися в обмеженій області відносно заданого положення. При цьому для HIL-реалізації з фізичним польотним контролером Pixhawk спостерігався більший розкид координат порівняно з програмною реалізацією ArduPilot. Характер зміни координат у режимі стабілізації наведено на рисунку 2.5.

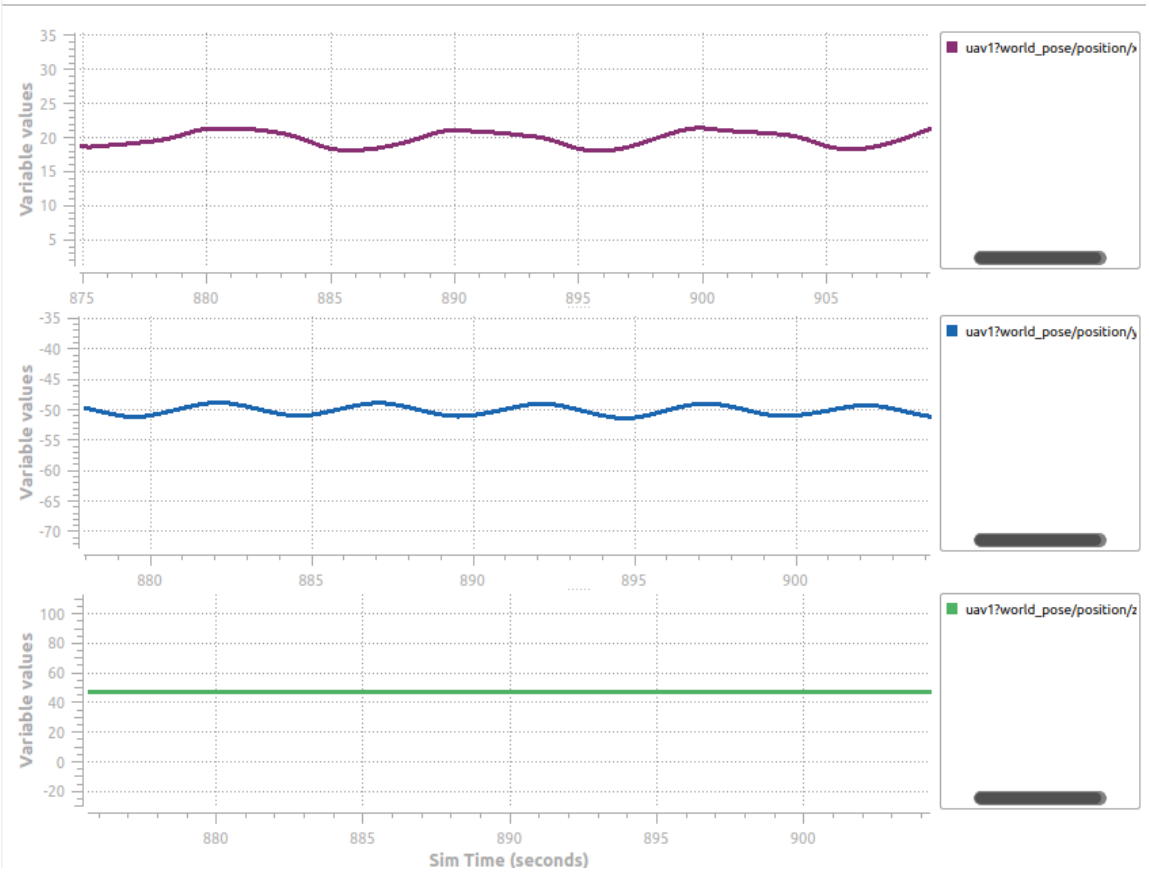


Рис. 2.5. Зміна координат БПЛА під час стабілізації в заданій позиції.

Для кількісного порівняння HIL- і SIL-реалізацій використовувалися локальні системи координат БПЛА, початок яких відповідав положенню кожного апарата в момент переведення польотного контролера в стан готовності до польоту (arming). Це дало змогу порівнювати характер утримання заданого положення незалежно від просторового розміщення БПЛА в модельованій сцені. У режимі стабілізації було опрацьовано вибірку зі 150 відліків координат. Результати порівняння наведено в Таблиці 2.1.

Таблиця 2.1

Порівняння положення БПЛА під час стабілізації для HIL- та SIL-реалізацій

Реалізація автопілота	**SIL**	**HIL**
Середнє X, м	21.30	21.32
Середнє Y, м	-50.09	-50.08
Стандартне відхилення X, м	1.09	1.77
Стандартне відхилення Y, м	0.69	2.40

Близькі середні значення координат X та Y характеризують положення БПЛА відносно локального початку відліку, а не їх глобальне розташування в модельованій сцені. Різниця середніх значень становила 0,02 м за координатою X та 0,01 м за координатою Y. Водночас стандартні відхилення для HIL-конфігурації були більшими: 1,77 м для HIL та 1,09 м для SIL за координатою X, а також 2,40 і 0,69 м відповідно за координатою Y. Отримані значення свідчать про близькість середніх положень HIL- та SIL-реалізацій у дослідженій конфігурації, при більшій варіативності координат HIL-агента.

Більший розкид координат HIL-агента може бути пов’язаний із додатковими апаратними затримками HIL-контуру, до складу якого входили фізичний контролер Pixhawk, послідовний канал зв’язку та засоби передавання повідомлень між компонентами середовища. Водночас внесок окремих складових контуру у збільшення розкиду координат у межах проведеного експерименту окремо не оцінювався. За результатами аналізу телеметрії частота обміну MAVLink-повідомленнями становила близько 50 Гц, втрат пакетів не зафіксовано, затримка передавання сенсорних даних становила приблизно 20–30 мс, а тривалість циклу керування була близькою до 4 мс.

Під час одночасного моделювання чотирьох БПЛА завантаження процесора експериментального комп’ютера досягало приблизно 70 %, а коефіцієнт реального часу (real-time factor, RTF) Gazebo дорівнював 1,0. Це свідчить про збереження режиму реального часу для дослідженої конфігурації з чотирма агентами.

Алгоритми активного уникнення перешкод у межах експерименту не досліджувалися. Крім того, у Gazebo не враховувалася аеродинамічна взаємодія між БПЛА, зокрема вплив повітряних потоків від гвинтів сусідніх апаратів. Ці фактори визначають межі застосовності отриманих експериментальних результатів.

Отримані результати підтвердили можливість одночасного функціонування HIL- та SIL-реалізацій польотних контролерів у спільному модельованому середовищі. Для дослідженої конфігурації з чотирма БПЛА експериментальний сценарій виконувався в режимі реального часу, а HIL- і SIL-реалізації забезпечували близькі середні значення координат під час утримання заданого положення.

2.7 Висновки до розділу II

Розроблено архітектуру гібридного HIL/SIL середовища напівнатурного моделювання мультиагентної системи БПЛА, яка передбачає спільне використання апаратних і програмних реалізацій польотних контролерів у єдиному модельованому середовищі. Організовано двонапрямлену інформаційну взаємодію між моделями БПЛА та польотними контролерами з розмежуванням інформаційних потоків окремих агентів за допомогою незалежних просторів імен, ідентифікаторів та каналів обміну.

Реалізовано експериментальне HIL/SIL середовище на основі Gazebo, ArduPilot, ROS і MAVROS із використанням апаратного польотного контролера Pixhawk та програмного модуля керування. Реалізація забезпечила одночасне підключення апаратного та програмних польотних контролерів до відповідних моделей БПЛА в спільному модельованому середовищі.

Експериментально підтверджено одночасне функціонування одного HIL- та трьох SIL-агентів під час зльоту, переміщення до заданих положень і стабілізації. HIL- і SIL-реалізації забезпечували близькі середні значення координат, при цьому для HIL-агента спостерігався більший розкид координат. Під час одночасного моделювання чотирьох БПЛА зберігався режим реального часу, а обмін MAVLink-повідомленнями відбувався без зафіксованих втрат пакетів. Отримані результати підтвердили працездатність реалізованої гібридної HIL/SIL архітектури в

досліджений конфігурації.

Результати досліджень, приведених в розділі, опубліковані в роботах [81, 82].

РОЗДІЛ ІІІ. МЕТОД НАВІГАЦІЇ БПЛА В ХОДІ ВИКОНАННЯ МІСІЇ В УМОВАХ ВІДСУТНОСТІ GPS

3.1 Формальна модель БПЛА в системі напівнатурного моделювання

Для формалізації виконання маршрутних сценаріїв у мультиагентній системі БПЛА розглядається як автономний агент із власним станом, маршрутним сценарієм, контуром керування та засобами інформаційної взаємодії.. Такий підхід забезпечує відокремлення процесів функціонування окремих БПЛА та дає змогу задавати для кожного агента власну послідовність дій у межах спільного модельованого середовища.

Мультиагентна система при цьому формується як сукупність агентів, кожен з яких виконує власний маршрутний сценарій, отримує сенсорну інформацію, формує керуючі впливи та підтримує інформаційну взаємодію з іншими компонентами системи. Така організація узгоджується з принципом розмежування моделей та інформаційних потоків окремих БПЛА, покладеним в основу гібридного HIL/SIL середовища моделювання.

Для формального подання і-го БПЛА-агента використано модель

$$A_i = (S_i, R_i, C_i, P_i),$$

де A_i – і-й агент; S_i – локальний стан агента; R_i – маршрутний сценарій; C_i – контур керування; P_i – правила переходів між станами.

Локальний стан агента S_i визначає поточний етап виконання маршрутного сценарію. Його використання дає змогу організувати функціонування окремого агента без формування єдиного глобального стану всієї мультиагентної системи. Поточний стан визначає етап виконання сценарію та використовується для вибору подальших дій відповідно до заданих правил.

Маршрутний сценарій R_i визначає послідовність маршрутних дій, які має виконати конкретний БПЛА. Відокремлення сценарію від програмної реалізації контуру керування дає змогу змінювати послідовність і параметри маршрутних дій без модифікації низькорівневих алгоритмів керування. Таким чином, маршрутний

сценарій задає необхідну поведінку агента на рівні послідовності дій, не визначаючи безпосередньо спосіб формування керуючих сигналів польотного контролера.

Контур керування C_i забезпечує формування керуючих впливів відповідно до поточного стану агента та параметрів маршрутного сценарію. Сценарний рівень і рівень безпосереднього керування функціонально розділені: сценарний рівень визначає послідовність дій та умови завершення окремих маршрутних кроків, тоді як контур керування забезпечує формування команд, необхідних для їх виконання. Такий розподіл дає змогу використовувати однаковий принцип опису маршрутного сценарію з різними реалізаціями польотного контуру, зокрема у програмній та напівнатурній конфігураціях.

Правила переходів P_i визначають умови зміни стану агента залежно від перебігу маршрутного сценарію та подій, що виникають під час його виконання. Послідовність дій БПЛА при цьому визначається поточним станом S_i, маршрутним сценарієм R_i та заданими правилами зміни станів. Таке подання дає змогу описати виконання місії як послідовність дискретних станів із визначеними умовами переходу між ними.

Кожному БПЛА відповідає власний набір компонентів моделі A_i, що забезпечує незалежне виконання маршрутного сценарію та локальне формування керуючих дій.

Запропонована модель використовується як формальна основа для представлення БПЛА-агента у програмній та напівнатурній конфігураціях. Вона забезпечує структурне відокремлення локального стану, маршрутного сценарію та контуру керування окремого агента. Експериментальна перевірка базового агентного контуру виконувалася у SIL- та HIL-конфігураціях, тоді як синхронна взаємодія декількох агентів не була предметом окремого експериментального дослідження.

3.2 Модель маршрутного сценарію БПЛА

Маршрутний сценарій R_i задає послідовність просторових переміщень, які має виконати і-й БПЛА-агент. Для програмного представлення сценарію використовується структурований опис у форматі JSON, у якому окремо задаються

загальні параметри виконання та послідовність маршрутних кроків [83]. Такий спосіб представлення дає змогу змінювати геометрію маршруту, кількість його кроків та умови їх завершення без зміни програмної реалізації контуру керування.

Формально маршрутний сценарій i-го агента задається у вигляді

$$R_i = \{r_{i1}, r_{i2}, \dots, r_{in}\},$$

де R_i – маршрутний сценарій i-го агента; r_{ij} – j-й маршрутний крок; n – кількість кроків маршруту. Порядок елементів послідовності визначає черговість виконання маршрутних кроків.

Окремий маршрутний крок визначається набором параметрів

$$r_{ij} = \left(dir, d_{px}, axis, e, e_{ct}, t_h\right),$$

де dir – напрям заданого переміщення; d_{px} – величина заданого переміщення у пікселях; $axis$ – вісь, за якою визначається завершення кроку; e – допустима похибка завершення кроку; e_{ct} – допустиме бокове відхилення від маршруту; t_h – час утримання положення після завершення сегмента.

Параметр dir визначає напрям заданого переміщення, а d_{px} характеризує його величину в координатах зображення. Параметр $axis$ задає координатну вісь, за якою контролюється виконання основного переміщення. Значення e визначає допустиму похибку досягнення цільового зміщення, за якої маршрутний крок може вважатися завершеним. Параметр e_{ct} обмежує допустиме відхилення за поперечною до основного напрямку віссю, а t_h задає тривалість утримання досягнутого положення перед переходом до наступного маршрутного кроку.

Формальні параметри маршрутного кроку мають безпосереднє представлення у файлі сценарію. Окрім них, у файлі задаються загальні параметри, які характеризують умови запуску та виконання маршруту. Структуру параметрів маршрутного сценарію наведено в Таблиці 3.1.

Таблиця 3.1

Структура параметрів маршрутного сценарію

Поле у файлі JSON	**Тип даних**	**Одиниця вимірювання**	**Призначення**
name	рядок	–	назва маршрутного сценарію
agent_id	рядок	–	ідентифікатор БПЛА-агента
controller_type	рядок	–	тип конфігурації керування
takeoff_altitude	дійсне число	м	задана висота зльоту
start_delay_seconds	дійсне число	с	затримка перед початком виконання маршруту
default_tolerance_pixels	дійсне число	пікселі	загальне допустиме відхилення під час завершення маршрутного кроку
default_cross_track_tolerance_pixels	дійсне число	пікселі	загальне допустиме бокове відхилення
default_hold_seconds	дійсне число	с	загальна тривалість утримання після завершення кроку
route / steps	послідовність	–	впорядкована сукупність маршрутних кроків

Загальні параметри застосовуються до маршрутного сценарію в цілому. Висота зльоту визначає початкову висоту виконання маршрутного завдання, а затримка перед початком руху задає часовий інтервал між підготовкою БПЛА та початком виконання першого маршрутного кроку. Ідентифікатор агента забезпечує прив'язку сценарію до відповідного БПЛА, а тип конфігурації керування визначає варіант польотного контуру, з яким виконується сценарій.

Для параметрів допустимої похибки, поперечного відхилення та часу утримання у файлі сценарію можуть бути задані загальні значення. Водночас структура маршрутного кроку передбачає можливість задавання окремих значень цих параметрів для конкретного переміщення. У такому разі параметри окремого кроку використовуються замість загальних значень сценарію. Це дає змогу в межах одного маршруту встановлювати різні умови завершення для окремих його ділянок без зміни структури програмного алгоритму.

Структура окремого маршрутного кроку наведена в Таблиці 3.2.

Таблиця 3.2

Параметри окремого маршрутного кроку

Поле у файлі JSON	**Відповідний параметр моделі**	**Одиниця вимірювання**	**Призначення**
name	–	–	ідентифікація маршрутного кроку
direction	dir	–	напрям заданого переміщення
distance_pixels	d_{px}	пікселі	величина заданого переміщення
completion_axis	$axis$	–	вісь контролю завершення маршрутного кроку
tolerance_pixels	e	пікселі	допустима похибка завершення кроку
cross_track_tolerance_pixels	e_{ct}	пікселі	допустиме бокове відхилення
hold_seconds	t_h	с	тривалість утримання після досягнення маршрутного кроку

Поле completion_axis визначає вісь, за якою контролюється виконання основного переміщення. У реалізації передбачено контроль за віссю x, віссю y або одночасно за обома координатами. Це дає змогу формувати маршрутні кроки з різною орієнтацією в площині зображення, зберігаючи однакову структуру їх опису. Допустиме поперечне відхилення використовується окремо від основної умови завершення та характеризує відхилення від заданого напрямку переміщення.

У реалізованій моделі координати зображення нижньої камери використовуються як система відліку маршрутного завдання. Тому довжина маршрутного кроку, допустима похибка та допустиме поперечне відхилення задаються у пікселях. Початкова точка нового маршрутного кроку використовується для визначення відносного переміщення, а цільове зміщення формується відповідно до параметрів цього кроку. Унаслідок цього маршрут являє собою послідовність відносних переміщень, а не набір абсолютних просторових координат.

Приклад виконання маршрутного сценарію як послідовності переміщень між опорними точками наведено на рисунку 3.1. Після досягнення чергової точки маршруту БПЛА отримує еталонне зображення місцевості, яке використовується для

визначення візуального зміщення під час подальшого переміщення. Послідовне виконання маршрутних кроків забезпечує перехід між заданими положеннями без використання глобальної координатної прив'язки.

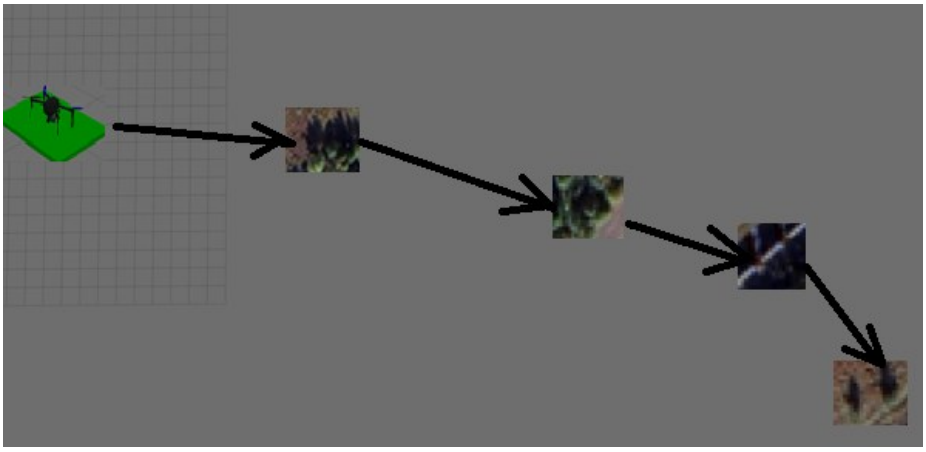


Рис. 3.1. Схематичне представлення виконання маршрутного сценарію БПЛА.

Таке представлення дає змогу описувати маршрут незалежно від глобальної системи позиціювання. Водночас величина переміщення у пікселях не є безпосереднім еквівалентом фізичної відстані. Відповідність між переміщенням у площині зображення та фактичним переміщенням БПЛА залежить від висоти польоту, параметрів камери та умов формування зображення. Тому без додаткового калібрування значення d_{px} характеризує відносне переміщення у прийнятій системі координат, а не відстань у метрах [84].

Подання маршруту як впорядкованої послідовності параметризованих кроків забезпечує незалежність структури сценарію від його конкретної геометрії. Зміна кількості кроків, їх напрямів, довжини та умов завершення здійснюється через параметри файлу JSON. Програмна логіка опрацювання маршрутного сценарію при цьому залишається незмінною, оскільки кожен його елемент має однакову структуру та обробляється за спільними правилами.

3.3 Модель виконання маршрутного сценарію на основі автомата станів

Виконання маршрутного сценарію організовано на основі діаграми станів, побудованого з використанням графічних нотацій, запропонованих в [85]. Дана

діаграма визначає послідовність етапів від підготовки системи до завершення польоту. Кожному стану відповідає окрема функціональна дія, а перехід до наступного стану здійснюється після виконання визначеної умови. Такий спосіб організації дає змогу відокремити послідовність виконання маршрутного завдання від безпосереднього формування керуючих впливів.

Діаграму станів виконання маршрутного сценарію БПЛА наведено на рисунку 3.2.

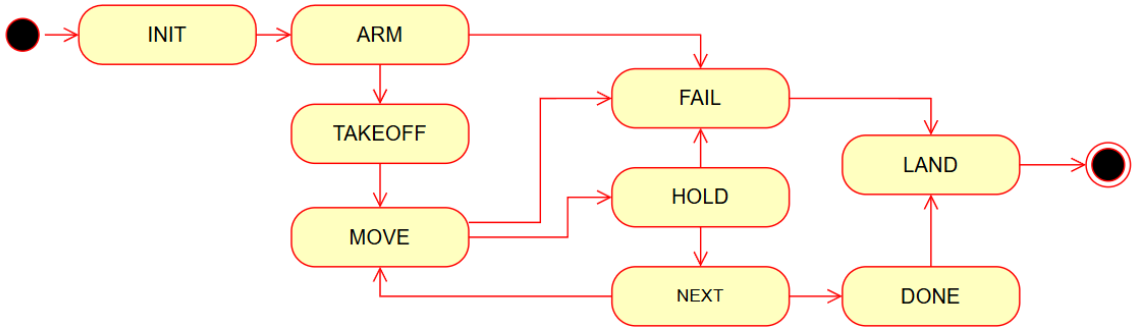


Рис. 3.2. Діаграма станів виконання маршрутного сценарію БПЛА.

Початковим станом є INIT, у якому виконуються підготовка параметрів маршрутного сценарію та початкова ініціалізація програмних компонентів. Після завершення підготовки здійснюється перехід у стан ARM, що відповідає підготовці польотного контролера до виконання польотних команд.

Після успішного завершення підготовки польотного контролера система переходить у стан TAKEOFF. У цьому стані формується команда зльоту до заданої висоти. Перехід до виконання маршрутного завдання здійснюється після досягнення необхідної висоти.

У стані MOVE виконується активний маршрутний крок. Його параметри визначають напрям і величину заданого переміщення, а також умови завершення. Поточне візуальне зміщення визначається у координатах зображення, після чого перевіряється його відповідність цільовому зміщенню поточного маршрутного кроку. Маршрутний крок вважається виконаним після досягнення заданого зміщення з урахуванням установленого допустимого відхилення.

Після виконання маршрутного кроку здійснюється перехід у стан HOLD. У цьому стані БПЛА утримує досягнуте положення протягом установленого інтервалу часу. Виділення окремого стану утримання дає змогу розмежувати процес активного переміщення та стабілізацію положення перед початком наступної маршрутної дії.

Після завершення заданого часу утримання система переходить у стан NEXT, у якому визначається наявність наступного маршрутного кроку. Якщо маршрут містить невиконані кроки, наступний крок встановлюється як активний, після чого система повертається у стан MOVE. Якщо всі маршрутні кроки виконано, здійснюється перехід у стан DONE, що відповідає завершенню маршрутної частини сценарію. Після цього виконується перехід у стан LAND, у якому формується команда завершення польоту та посадки.

Перелік станів процесу керування БПЛА під час виконання маршрутного сценарію та умов переходу між ними наведено в таблиці 3.3

Таблиця 3.3

Стани та умови переходів між ними під час виконання маршрутного сценарію

Поточний стан	**Виконувана дія**	**Умова переходу**	**Наступний стан**
INIT	початкова підготовка та завантаження параметрів сценарію	підготовку завершено	ARM
ARM	підготовка польотного контролера	підготовку завершено успішно	TAKEOFF
TAKEOFF	набір заданої висоти	заданої висоти досягнуто	MOVE
MOVE	виконання поточного маршрутного кроку	виконано умову завершення кроку	HOLD
HOLD	утримання досягнутого положення	завершено заданий інтервал утримання	NEXT
NEXT	перевірка наявності наступного маршрутного кроку	наступний крок існує	MOVE
NEXT	перевірка завершення маршруту	усі кроки виконано	DONE
DONE	завершення маршрутної частини сценарію	ініційовано завершення польоту	LAND

LAND	виконання посадки	посадку завершено	завершення сценарію
FAIL	припинення виконання маршрутних кроків та ініціювання завершення польоту	виникнення помилки, що унеможливлює подальше виконання сценарію	LAND

Основну повторювану частину автомата утворює послідовність MOVE–HOLD–NEXT. Під час першого з цих станів виконується переміщення відповідно до параметрів активного маршрутного кроку. Після досягнення цільового зміщення БПЛА переходить до короткочасного утримання положення. Після завершення утримання визначається наступний елемент маршрутної послідовності. Такий цикл повторюється доти, доки не буде виконано всі маршрутні кроки.

Використання окремого стану NEXT дає змогу відокремити перевірку структури маршруту від процесу безпосереднього переміщення. При цьому стан MOVE не визначає, чи є поточний крок останнім, а відповідає лише за виконання активного переміщення. Рішення про продовження або завершення маршруту приймається після закінчення відповідного етапу утримання.

Умови завершення стану MOVE визначаються параметрами активного маршрутного кроку. Для переміщення вздовж однієї з координатних осей контролюється досягнення заданого зміщення за відповідною віссю з урахуванням допустимої похибки. Поперечне відхилення при цьому контролюється окремо. Завдяки цьому однакова структура автомата може використовуватися для маршрутних кроків із різним напрямом і довжиною переміщення.

Структура автомата не залежить від кількості маршрутних кроків. Зміна довжини маршруту або послідовності переміщень не потребує зміни набору станів: змінюється лише кількість повторень циклу MOVE–HOLD–NEXT та параметри активного кроку. Такий принцип забезпечує використання єдиної логіки виконання для маршрутних сценаріїв різної структури.

Окремо в програмній реалізації передбачено перехід до стану завершення з помилкою у випадку, коли штатне продовження сценарію неможливе. Такий стан

відокремлює нормальне завершення маршрутної послідовності від переривання її виконання. Після виникнення помилки подальше виконання маршрутних кроків припиняється та ініціюється завершення польоту.

Таким чином, автомат станів забезпечує послідовне виконання підготовчих, маршрутних і завершальних етапів польоту. Циклічна обробка маршрутних кроків відокремлена від процедур початкової підготовки та завершення польоту, а параметри конкретного маршруту не змінюють загальної структури автомата.

3.4 Отримання та оброблення сигналу візуального зміщення за даними камери

Для керування положенням БПЛА в горизонтальній площині в умовах відсутності GPS використовується інформація про відносне зміщення зображення земної поверхні. Джерелом відеоданих є камера, спрямована вниз відносно БПЛА. У використовуваній конфігурації камера формує зображення розміром 640×480 пікселів із частотою 30 кадрів за секунду.

Для отримання оцінки горизонтального зміщення у площині зображення використано модуль оцінювання візуального зміщення, який обробляє відеопотік камери та формує дві складові зміщення d_x і d_y у піксельному просторі , а також показник matches_percent, що характеризує частку встановлених відповідностей між візуальними ознаками поточного та опорного кадрів. Компоненти d_x і d_y використовуються для формування керуючих впливів, тоді як matches_percent використовується для формування нормованого показника якості візуального оцінювання.

Використаний модуль визначає зміщення між поточним і опорним зображеннями на основі локальних візуальних ознак та оцінювання геометричного перетворення між кадрами. В опублікованому описі реалізації для цього застосовуються ознаки ORB, зіставлення відповідних точок та оцінювання гомографії з використанням RANSAC [86]. Результатом роботи цього етапу є компоненти d_x і d_y, виражені в пікселях, та показник matches_percent [87].

Детальний алгоритм виділення та зіставлення локальних ознак у цій роботі окремо не розглядається, оскільки предметом дослідження є використання сформованого сигналу візуального зміщення в контурі керування БПЛА. Таким чином, модуль оцінювання зміщення розглядається як функціональний компонент із відеопотоком на вході та оціненими значеннями d_x, d_y і matches_percent на виході [88-91].

Для інтеграції модуля оцінювання зміщення з системою керування було виконано налаштування відеоканалу та організовано передавання отриманих значень d_x і d_y до програмного модуля керування. При цьому враховувалася відповідність напрямів зміщення у просторі зображення напрямам формування керуючих впливів польотного контролера. Це забезпечувало коректну реакцію БПЛА на відхилення, зафіксовані нижньою камерою.

На дискретному кроці k сигнал візуального зміщення подається у вигляді

$$d(k) = \begin{bmatrix} d_x(k) \\ d_y(k) \end{bmatrix},$$

де $d_x(k)$ і $d_y(k)$ є горизонтальною та вертикальною складовими зміщення поточного кадру відносно опорного у координатах зображення.

Сформовані значення d_x і d_y характеризують відносне зміщення у координатах зображення і безпосередньо використовуються як вхідні сигнали контуру керування.

Перед використанням у контурі керування сигнал візуального зміщення додатково обробляється для зменшення впливу шуму та малих випадкових коливань. В опублікованій реалізації для цього застосовуються експоненційне згладжування та мертва зона, у межах якої невеликі значення d_x і d_y не спричиняють зміни керуючого впливу. Така обробка зменшує чутливість контуру до нестабільності оцінювання візуального зміщення та запобігає надмірній реакції польотного контролера на незначні коливання сигналу.

Після оброблення компоненти d_x і d_y використовуються як сигнали похибки для формування керуючих впливів у горизонтальній площині.

Кожному дискретному кроку k відповідає момент часу t, у якому формується чергова оцінка візуального зміщення. Тому компоненти дискретного сигналу можуть бути подані як $d_x(k) = d_x(t)$ та $d_y(k) = d_y(t)$. У подальшому під час опису закону керування використовується часове представлення цих сигналів $d_x(t)$ та $d_y(t)$.

3.5 Метод просторової стабілізації БПЛА за візуальними даними без GPS

Для утримання БПЛА в заданому положенні без використання GPS застосовано метод просторової стабілізації, що поєднує керування положенням у горизонтальній площині X-Y за сигналом візуального зміщення та окрему стабілізацію висоти за координатою Z. Необхідність такого розділення зумовлена різною природою вхідних даних: горизонтальне зміщення визначається у піксельному просторі за даними нижньої камери, тоді як вертикальне положення контролюється за показами висотного каналу.

Унаслідок зовнішніх збурень і похибок оцінювання стану БПЛА виникає зміщення від заданого положення, яке визначається за даними нижньої камери. Якщо величина зміщення перебуває в межах мертвої зони, коригування не здійснюється, що дає змогу уникнути зайвих керувальних впливів через незначні відхилення. У разі перевищення встановленого порогу система формує керувальний вплив для компенсації зміщення та повернення БПЛА до заданого положення. Принцип такого коригування схематично представлено на рисунку 3.3.

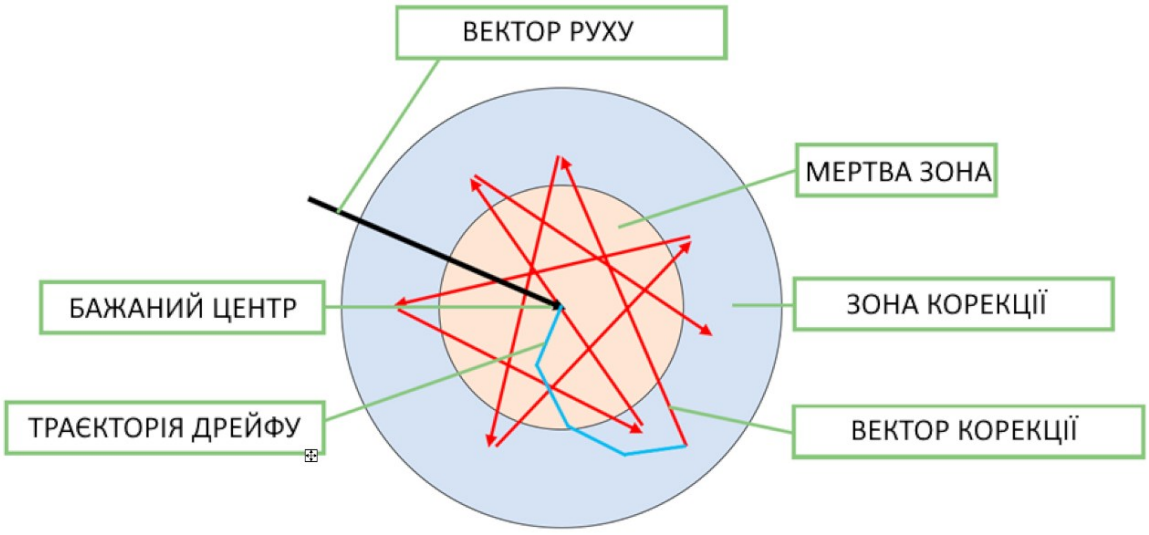


Рис. 3.3. Схема маневрування при стабілізації в заданій позиції БПЛА [91].

Структурно метод містить два паралельні контури: горизонтальний контур X-Y та вертикальний контур Z. У горизонтальному контурі вхідними величинами є складові візуального зміщення d_x і d_y, отримані та підготовлені відповідно до описаного способу. Вертикальний контур використовує поточне значення висоти $h(t)$. Вихідні керуючі впливи обох контурів передаються польотному контролеру, який виконує внутрішні низькорівневі алгоритми стабілізації БПЛА. Структурну схему методу наведено на рисунку 3.4.

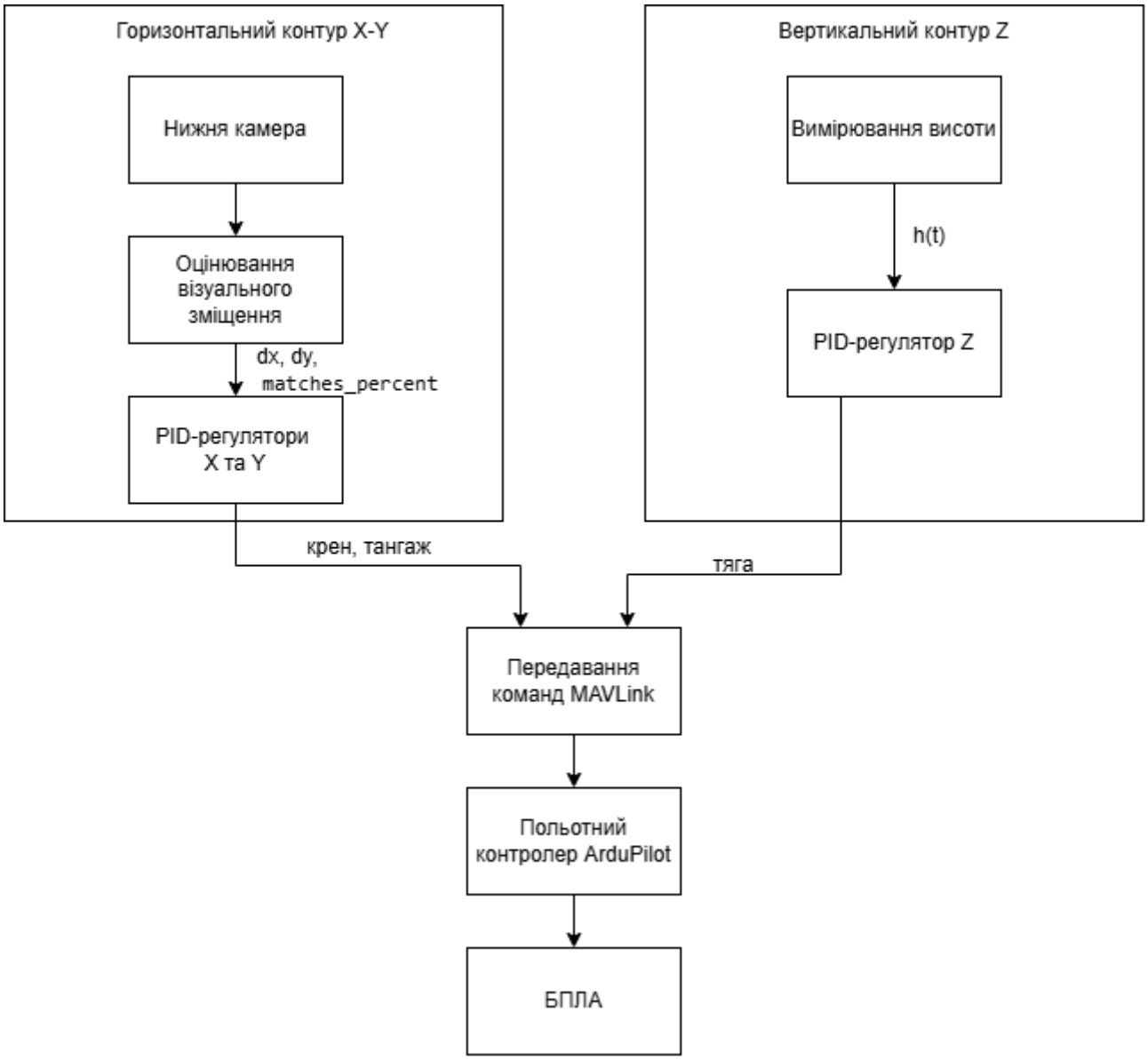


Рис. 3.4. Структурна схема двоконтурної просторової стабілізації БПЛА.

У горизонтальній площині заданому положенню БПЛА відповідає положення,

для якого компоненти візуального зміщення поточного кадру відносно опорного наближаються до нуля. Як вхідні сигнали горизонтального контуру використовуються піксельні відхилення $d_x(t)$ та $d_y(t)$, отримані за даними нижньої камери. Для кожної осі застосовується окремий PID-регулятор, а керуючі впливи визначаються як:

$$u_x(t) = PID_x\big(-d_x(t)\big),$$

$$u_y(t) = PID_y\left(-d_y(t)\right),$$

де $d_x(t)$ і $d_y(t)$ – піксельні відхилення поточного кадру відносно опорного; $u_x(t)$ та $u_y(t)$ – вихідні сигнали відповідних PID-регуляторів. Від'ємний знак забезпечує формування коригувального впливу в напрямі, протилежному зафіксованому візуальному зміщенню.

Для довільного каналу PID-закон має вигляд

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt},$$

де K_p, K_i та K_d є пропорційним, інтегральним і диференціальним коефіцієнтами регулятора відповідно.

Сформовані значення $u_x(t)$ та $u_y(t)$ перетворюються у коригувальні впливи для каналів крену та тангажу відповідно до прийнятого узгодження координат камери й системи керування БПЛА. Таким чином, зміщення зображення відносно опорного положення спричиняє керуючий вплив у напрямі, протилежному зафіксованому відхиленню. Після зменшення d_x і d_y до області допустимих значень коригувальний вплив відповідно зменшується.

Для обмеження надмірних керуючих впливів у горизонтальному контурі передбачено насичення вихідних сигналів та обмеження швидкості їх зміни. Мертва зона, застосована під час підготовки сигналу візуального зміщення, унеможливлює реакцію регуляторів на незначні коливання d_x і d_y поблизу заданого положення. Сукупне використання цих обмежень зменшує ймовірність виникнення надмірних корекцій та автоколивань під час утримання позиції.

Вертикальна складова методу реалізується окремим контуром стабілізації висоти. Похибка за висотою визначається як

$$e_z(t) = h_{ref} - h(t),$$

де h_{ref} є заданою висотою польоту, а $h(t)$ – її поточним виміряним значенням.

Керуючий вплив вертикального контуру визначається PID-регулятором

$$u_z(t) = PID_z\big(e_z(t)\big)$$

і використовується для корекції тяги БПЛА. Перед надходженням до вертикального PID-регулятора виміряне значення висоти згладжується за допомогою експоненційного ковзного середнього, що зменшує вплив високочастотних коливань і шуму вимірювання на формування керуючого впливу [92]. У програмній конфігурації джерелом даних про висоту були барометричні дані, а під час натурних випробувань використовувався дальномір.

Виділення незалежного вертикального контуру є принциповим для роботи горизонтальної візуальної стабілізації. Оскільки d_x і d_y визначаються у координатах зображення, зміна висоти БПЛА спричиняє зміну масштабу спостережуваної поверхні. Унаслідок цього однаковому фізичному переміщенню на різних висотах відповідають різні величини зміщення в пікселях. Підтримання висоти поблизу заданого значення зменшує зміну масштабу сцени та забезпечує більш сталі умови використання сигналу візуального зміщення в горизонтальному контурі.

Горизонтальний та вертикальний контури функціонують паралельно. На кожному циклі керування горизонтальний контур формує коригувальні впливи за d_x і d_y, а вертикальний контур – за похибкою висоти. Отримані складові керування об’єднуються у команди крену, тангажу та тяги й передаються до польотного контролера, який забезпечує їх виконання штатними внутрішніми контурами стабілізації. У реалізованій системі передавання керуючих впливів здійснювалося через MAVLink [93].

Таким чином, запропонований метод забезпечує просторову стабілізацію БПЛА без використання глобальних координат: положення у горизонтальній

площині підтримується за похибкою у просторі зображення, а вертикальне положення – за незалежним вимірюванням висоти. Такий поділ дає змогу незалежно формувати горизонтальні та вертикальні керуючі впливи та зберігати штатні низькорівневі функції стабілізації польотного контролера

3.6 Реалізація маршрутного керування на основі візуальної стабілізації

Програмна реалізація маршрутного керування забезпечує послідовне перетворення параметрів активного маршрутного кроку в керуючі впливи, що передаються польотному контролеру. До складу контуру входять засоби виконання маршрутного сценарію, навігаційної координації, модифікації керуючого сигналу, оцінювання візуального зміщення, PID-регулювання та інформаційної взаємодії з ArduPilot через протокол MAVLink. Така структура забезпечує розмежування між формуванням маршрутного завдання, визначенням позиційної похибки та безпосереднім формуванням керуючих команд.

Маршрутне завдання задається як послідовність параметризованих кроків. Після активації поточного кроку визначаються його напрям, величина заданого переміщення та умови завершення. На основі цих параметрів формується цільове зміщення в координатах зображення. Поточне візуальне зміщення визначається за даними зовнішнього модуля оцінювання візуального зміщення, який формує його складові за двома осями зображення. Внутрішні алгоритми цього модуля не є складовою розробленого методу; у контурі керування використовуються сформовані ним вихідні величини.

Основні функціональні складові реалізації наведено в таблиці 3.4

Таблиця 3.4

Функціональні складові програмної реалізації маршрутного керування

Функціональна складова	**Вхідні дані**	**Виконувана функція**	**Результат**
Засоби виконання маршрутного сценарію	параметри сценарію	послідовне виконання маршрутних кроків	активний маршрутний крок
Навігаційний координатор	активний крок, поточне зміщення	визначення цільового зміщення та перевірка умов завершення	навігаційна ціль

Модифікатор команд	цільове та поточне зміщення	визначення похибки та формування віртуального дрейфу	модифікований вхідний сигнал
Модуль оцінювання візуального зміщення	відеопотік нижньої камери	визначення відносного зміщення	складові d_x, d_y
PID-регулятори	похибки за координатними осями	формування коригувальних впливів	керуючі складові
Засоби взаємодії з ArduPilot	значення керуючих каналів	передавання команд через MAVLink	команди перевизначення RC-каналів
Засоби реєстрації даних	параметри поточного циклу керування	збереження експериментальних даних	файли результатів експерименту

У межах цієї роботи під віртуальним дрейфом розуміється програмно сформована складова вхідного сигналу контуру візуальної стабілізації, яка вводиться відповідно до похибки виконання активного маршрутного кроку та використовується для формування маршрутного переміщення БПЛА.

Для кожного маршрутного кроку формується вектор цільового зміщення

$$T = (t_x, t_y),$$

де t_x та t_y є складовими цільового зміщення у піксельному просторі. Поточне зміщення, отримане від модуля оцінювання візуального зміщення, подається у вигляді

$$S = (s_x, s_y),$$

де s_x та s_y є поточними складовими зміщення БПЛА відносно опорного положення.

На основі цільового та поточного зміщень визначається похибка виконання маршрутного кроку у координатах зображення

$$e_x = t_x - s_x,$$

$$e_y = t_y - s_y,$$

$$e_d = \sqrt{e_x^2 + e_y^2},$$

де e_x та e_y характеризують похибки за відповідними осями зображення, а e_d визначає модуль вектора похибки у піксельному просторі.

За складовими похибки e_x та e_y маршрутний PID-регулятор незалежно формує коригувальні складові для двох координатних осей. Величина сформованих керуючих впливів обмежується допустимим діапазоном вихідного сигналу. Незалежне регулювання за двома осями дає змогу одночасно враховувати похибку вздовж основного напряму поточного маршрутного кроку та поперечне відхилення від заданого напряму.

Маршрутна похибка не передається безпосередньо польотному контролеру. Для виконання маршрутного переміщення використовується механізм віртуального дрейфу. Його призначення полягає у формуванні штучного зміщення вхідного сигналу контуру візуальної стабілізації відповідно до поточного маршрутного завдання. Сформовані PID-регуляторами керуючі складові вводяться до сигналу візуального зміщення таким чином, щоб контур стабілізації сприймав їх як відхилення від положення утримання.

У реалізованому алгоритмі після визначення поточного положення обчислюються похибки за основною та поперечною координатними осями. За отриманими значеннями двоканальний PID-регулятор формує коригувальні складові, після чого модифікований сигнал передається до контуру візуальної стабілізації з протилежним знаком. Таким чином створюється віртуальне зміщення, компенсація якого приводить до фактичного переміщення БПЛА у напрямі, визначеному активним маршрутним кроком.

Такий спосіб керування дає змогу використовувати один і той самий контур візуальної стабілізації як для утримання положення, так і для виконання маршрутного переміщення. Різниця між цими режимами полягає у формуванні вхідного сигналу: під час утримання положення регулятор компенсує фактичне візуальне зміщення, а під час маршрутного руху до нього додатково вводиться штучно сформована складова, яка відповідає заданому напрямку переміщення.

Схему перетворення маршрутного кроку у керуючий вплив наведено на рисунку 3.5.

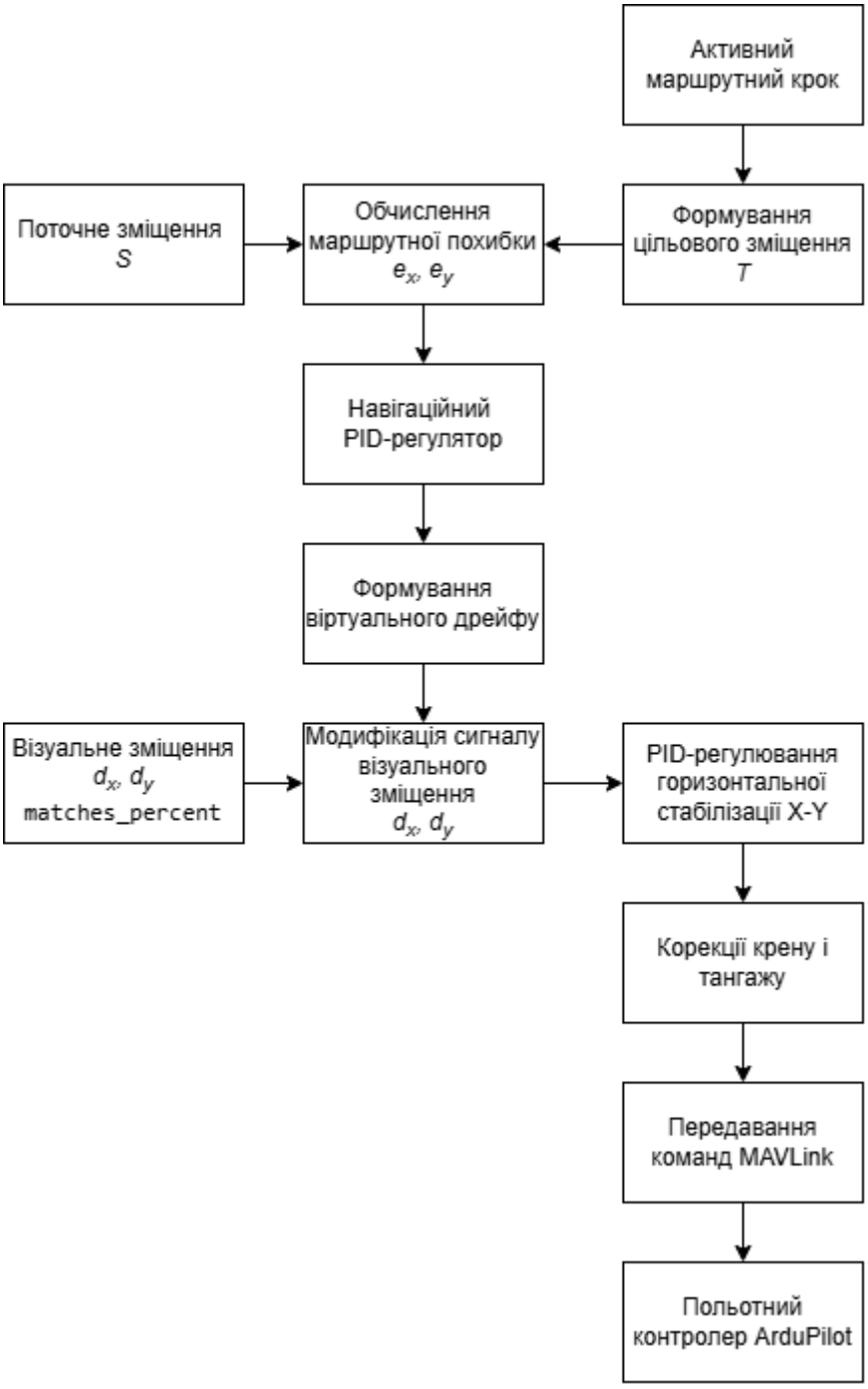


Рис. 3.5. Схема перетворення маршрутного кроку у керуючий вплив.

На рівні програмної реалізації послідовність оброблення даних включає визначення активного маршрутного кроку, формування цільового зміщення, отримання поточного візуального зміщення, обчислення позиційної похибки, формування коригувальних складових PID-регуляторами та введення відповідного віртуального дрейфу до контуру візуальної стабілізації. Результатом роботи цього

контуру є корекції керуючих каналів, які передаються до ArduPilot через MAVLink із використанням механізму перевизначення RC-каналів. Така послідовність відповідає фактичній реалізації програмного контуру маршрутного керування.

Для подальшого аналізу перебігу експерименту передбачено реєстрацію параметрів маршрутного та регуляторного контурів. До збережених даних належать ідентифікатор агента, тип конфігурації керування, назва маршрутного сценарію, номер поточного маршрутного кроку, його цільові параметри, поточне положення у піксельному просторі, складові позиційної та керуючої похибок, сформовані керуючі величини, параметри роботи PID-регуляторів і значення введеного віртуального зміщення.

Основні групи зареєстрованих параметрів наведено в табл. 3.5.

Таблиця 3.5

Основні групи параметрів, що реєструються під час виконання маршрутного сценарію

Група параметрів	**Зареєстровані дані**
Ідентифікаційні	ідентифікатор агента, тип конфігурації, назва маршрутного сценарію
Маршрутні	номер поточного кроку, загальна кількість кроків, цільове зміщення, вісь завершення, допустиме поперечне відхилення
Позиційні	поточне положення та відносне зміщення у піксельному просторі
Похибки керування	позиційна похибка та похибки за координатними осями
Регуляторні	сформовані керуючі величини, складові та коефіцієнти PID-регуляторів
Вихідні	складові введеного віртуального зміщення
Часові	момент реєстрації та час від початку виконання сценарію

Фактична структура файлів результатів містить параметри маршрутного кроку, цільового та поточного положення, похибки, сформовані керуючі величини, складові PID-регуляторів і введене зміщення, що дає змогу відновлювати перебіг окремого експериментального запуску.

Після завершення експерименту зареєстровані дані використовуються для побудови фактичної траєкторії у піксельному просторі та визначення показників

виконання маршрутного сценарію. Збереження первинних даних окремо від процедури їх подальшого опрацювання дає змогу повторно аналізувати результати окремих запусків без впливу на процес керування під час виконання маршруту.

Отже, програмна реалізація маршрутного керування поєднує формування маршрутної цілі у піксельному просторі, використання візуального зміщення як сигналу зворотного зв'язку, визначення позиційної похибки, незалежне PID-регулювання за двома координатними осями, формування віртуального дрейфу та передавання керуючих впливів польотному контролеру через MAVLink. Реєстрація параметрів поточного циклу забезпечує збереження даних, необхідних для подальшого кількісного аналізу результатів виконання маршрутних сценаріїв

3.7 Експериментальне дослідження маршрутного керування та просторової стабілізації

Експериментальне дослідження розроблених засобів керування виконувалося у двох окремих серіях. Перша серія була спрямована на перевірку виконання маршрутного сценарію у SIL- та HIL-конфігураціях. Друга серія була призначена для оцінювання просторової стабілізації БПЛА без GPS у модельованих та натурних умовах. Серії відрізнялися експериментальними конфігураціями, умовами проведення та показниками оцінювання.

3.7.1 Експериментальне дослідження маршрутного керування у SIL- та HIL-конфігураціях

Для перевірки маршрутного керування використовувався двосегментний сценарій, що передбачав послідовне переміщення БПЛА вперед та ліворуч у піксельному просторі нижньої камери. Маршрут складався з 28 елементарних кроків: 14 кроків у напрямку вперед та 14 кроків у напрямку ліворуч. Довжина кожного кроку становила 50 пікселів, а сумарна величина заданих маршрутних переміщень становила 1400 пікселів. Для завершення окремого кроку використовувалася допустима похибка 20 пікселів, допустиме бокове відхилення становило 60 пікселів, а тривалість утримання положення після виконання кроку становила 0,5 с. Висота зльоту в експериментальному сценарії була задана на рівні 2,0 м.

Вибір двосегментного маршруту давав змогу оцінити не лише виконання послідовного переміщення в одному напрямі, а й перехід між маршрутними ділянками зі зміною активної координатної осі. На першій ділянці основне переміщення виконувалося вздовж однієї осі з одночасним обмеженням поперечного відхилення, після чого для другої ділянки змінювався основний напрям руху. Таким чином, експеримент охоплював як виконання окремих маршрутних кроків, так і перехід між двома послідовними напрямами переміщення.

Для оцінювання кожного запуску використовували тривалість виконання маршрутного сценарію, кінцеву похибку, середню похибку протягом маршруту та мінімальне зафіксоване значення похибки. Похибка визначалася як модуль різниці між цільовим і поточним зміщеннями у координатах зображення. Кінцева похибка характеризувала стан на завершенні сценарію, середня похибка – загальний рівень відхилення протягом його виконання, а мінімальна похибка – найменше зафіксоване відхилення в межах відповідного запуску.

У програмній конфігурації було виконано п’ять запусків маршрутного сценарію. У кожному запуску БПЛА послідовно виконував обидва сегменти маршруту.

Для детального аналізу характеру руху було використано третій SIL-запуск, який характеризувався найменшою середньою похибкою серед п’яти проведених запусків. Порівняння заданої та фактичної траєкторій цього запуску наведено на рисунку 3.6.

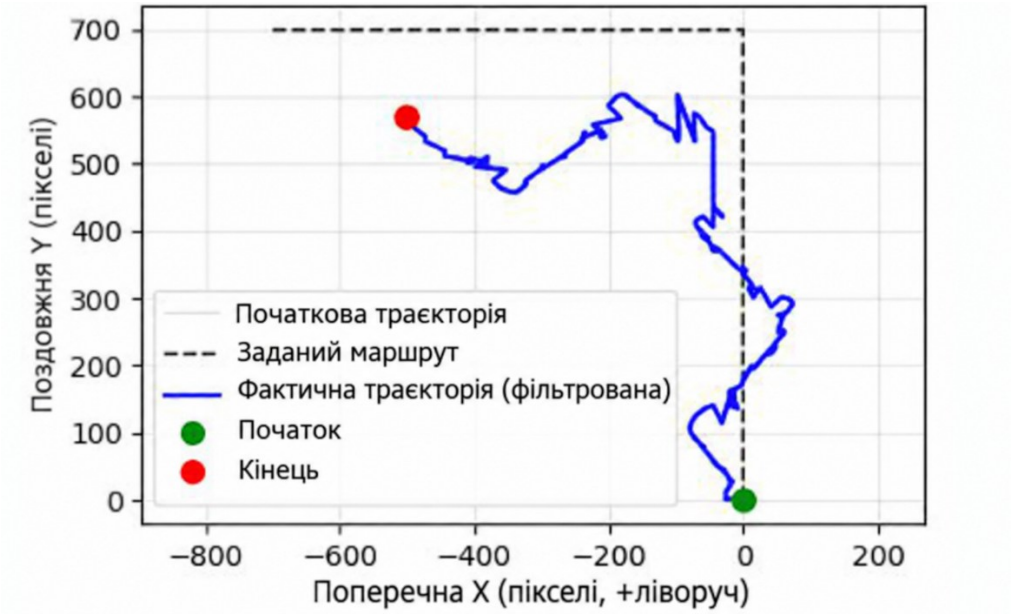


Рис. 3.6. Заданий маршрут та фактична траєкторія БПЛА під час виконання маршрутного сценарію в SIL-конфігурації.

Наведена траєкторія показує, що БПЛА відтворював загальну L-подібну структуру маршрутного завдання, проте під час руху спостерігалися бокові відхилення від заданого маршруту. Для оцінювання зміни похибки протягом виконання цього самого запуску було проаналізовано її часовий ряд. Зміну піксельної похибки під час проходження маршруту наведено на рисунку 3.7.

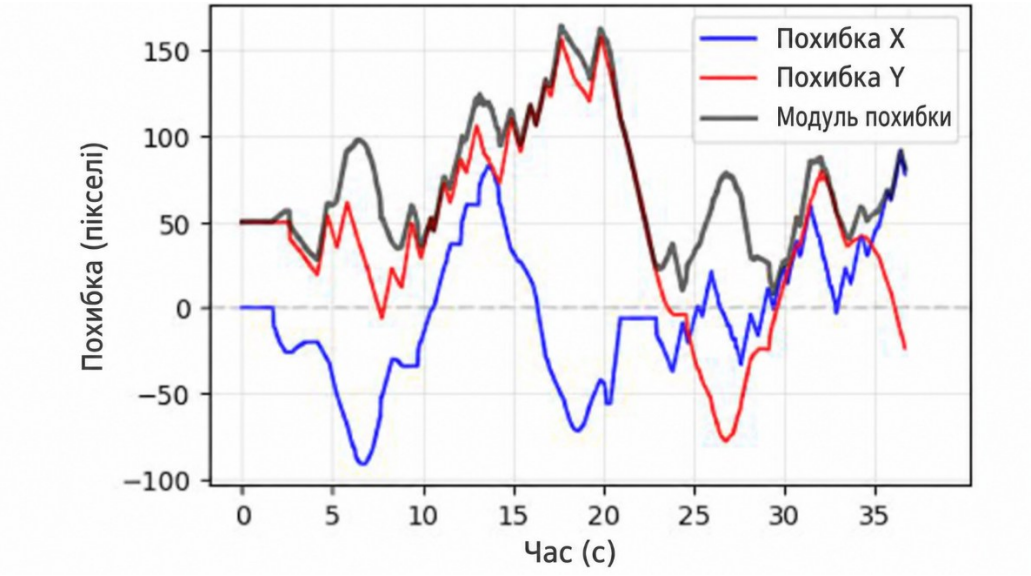


Рис. 3.7. Зміна піксельної похибки під час виконання маршрутного сценарію в SIL-конфігурації.

Аналіз часової залежності показує нерівномірний характер похибки та її локальне збільшення під час переходу між сегментами маршруту, коли змінювалася активна вісь переміщення.

Перед запуском маршрутного сценарію перевірялися MAVLink-з'єднання з контролером, отримання heartbeat-повідомлення, встановлення режиму STABILIZE та виконання команди армування. Для HIL-конфігурації було виконано три запуски того самого маршрутного сценарію. Для детального аналізу було використано другий HIL-запуск, що мав найменшу середню похибку серед трьох проведених запусків. Задану та фактичну траєкторії БПЛА для цього запуску наведено на рисунку 3.8.

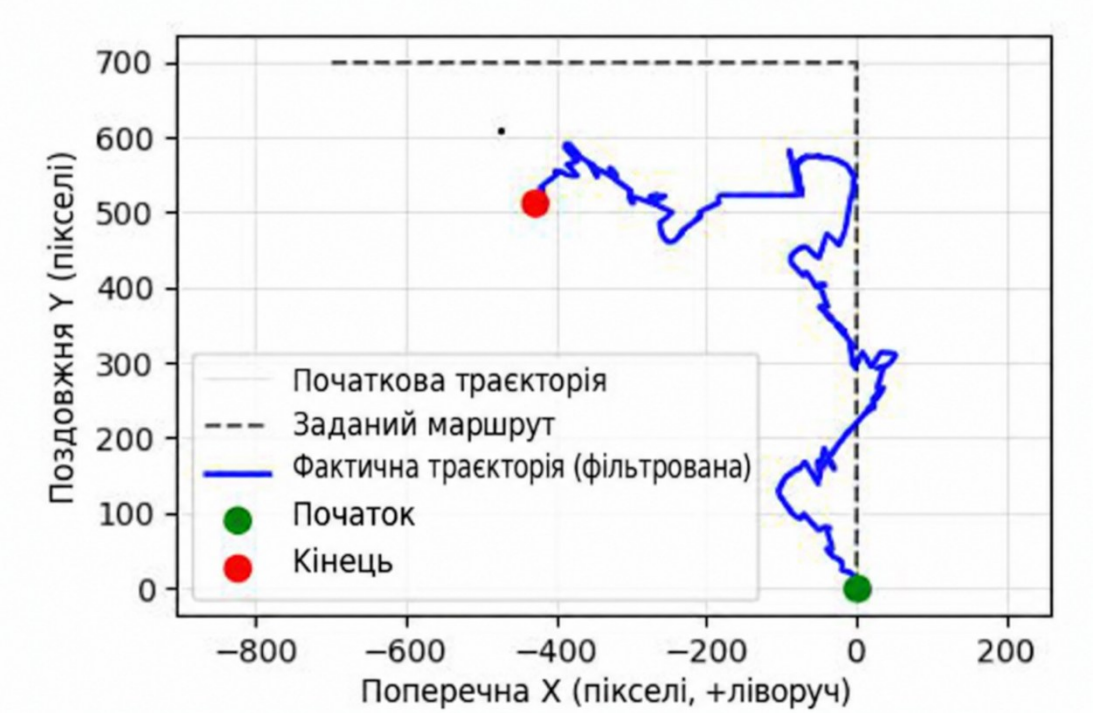


Рис. 3.8. Заданий маршрут та фактична траєкторія БПЛА під час виконання маршрутного сценарію в HIL-конфігурації.

У HIL-конфігурації фактична траєкторія також відтворювала загальну форму заданого маршруту, однак відхилення від нього були більшими порівняно з програмною реалізацією. Характер зміни піксельної похибки протягом HIL-запуску наведено на рисунку 3.9.

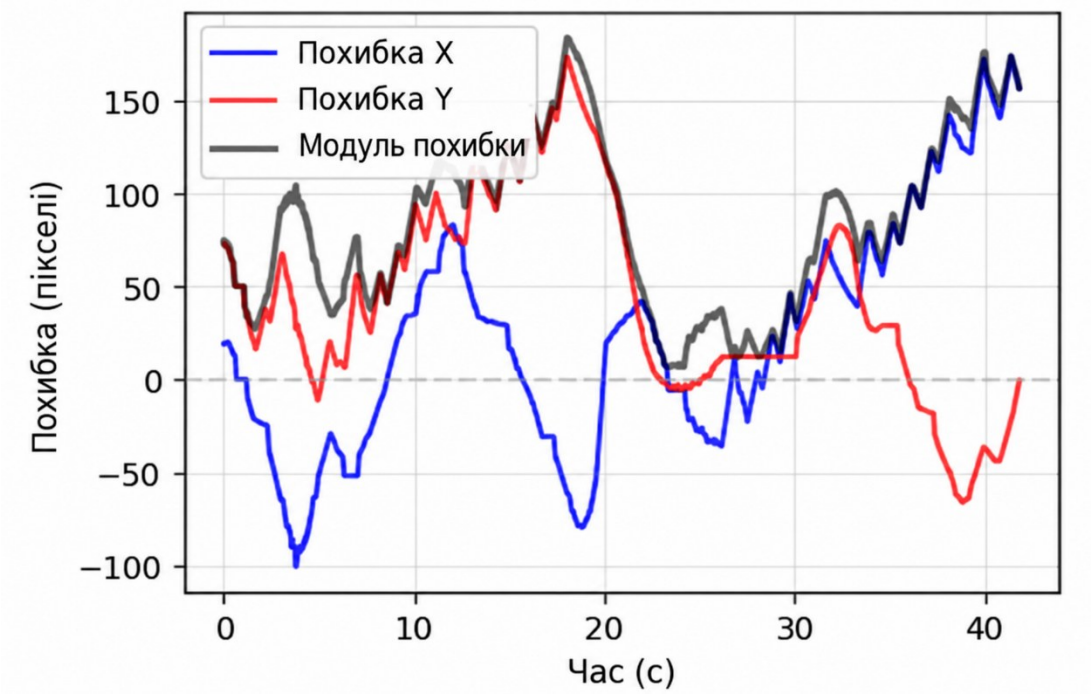


Рис. 3.9. Зміна піксельної похибки під час виконання маршрутного сценарію в HIL-конфігурації.

Для узагальненого порівняння програмної та напівнатурної конфігурацій використано середні показники всіх проведених запусків. Результати порівняння наведено в таблиці 3.6.

Таблиця 3.6

Порівняння виконання маршрутного сценарію в SIL- та HIL-конфігураціях

Показник	**SIL**	**HIL**
Кількість запусків	5	3
Середній час виконання, с	35.12	44.58
Середня кінцева похибка, пікс	93.24	169.63
Середня похибка, пікс.	91.46	106.60
Мінімальна похибка, пікс.	7.42	10.77

Порівняння показало, що однакова модель виконання маршрутного сценарію забезпечила проходження заданої послідовності дій як у програмній, так і в напівнатурній конфігурації. У дослідженій HIL-серії спостерігалися більші кінцеві та

середні похибки, а також більша тривалість виконання маршруту. Такі відмінності можуть бути пов’язані з особливостями роботи фізичного польотного контролера, обміну даними через MAVLink, часовими затримками в контурі керування та налаштуваннями регулятора. Вплив кожного із зазначених чинників окремо в межах проведеної експериментальної серії не оцінювався.

Експериментальна перевірка маршрутного керування стосувалася базового агентного контуру. Одночасне виконання маршрутних сценаріїв декількома агентами з подієвою взаємодією між ними в цій серії експериментів не досліджувалося.

3.7.2 Експериментальне дослідження просторової стабілізації в модельованих та натурних умовах

Експериментальне дослідження просторової стабілізації БПЛА без GPS виконувалося у двох умовах: у модельованому середовищі Gazebo/ArduPilot та під час натурного експерименту в приміщенні. Горизонтальний контур оцінювався за складовими візуального зміщення d_x та d_y, а вертикальний – за похибкою утримання заданої висоти. Метою експерименту було оцінювання роботи розробленого методу просторової стабілізації за модельованих і реальних умов функціонування.

Для програмної конфігурації використовувалися коефіцієнти PID K_p=1.20, K_i=0.05, K_d=0.80для обох горизонтальних каналів та deadband шириною 5 пікс. Для натурного експерименту застосовувалися менші коефіцієнти: для каналу X K_p=0.30, K_i=0.015, K_d=0.30; для каналу Y K_p=0.25, K_i=0.015, K_d=0.30. Ширина мертвої зони становила 10 пікселів.

Для оцінювання горизонтальної стабілізації аналізувалися часові ряди компонентів d_x та d_y. Результати для програмної конфігурації Gazebo/ArduPilot наведено на рисунку 3.10.

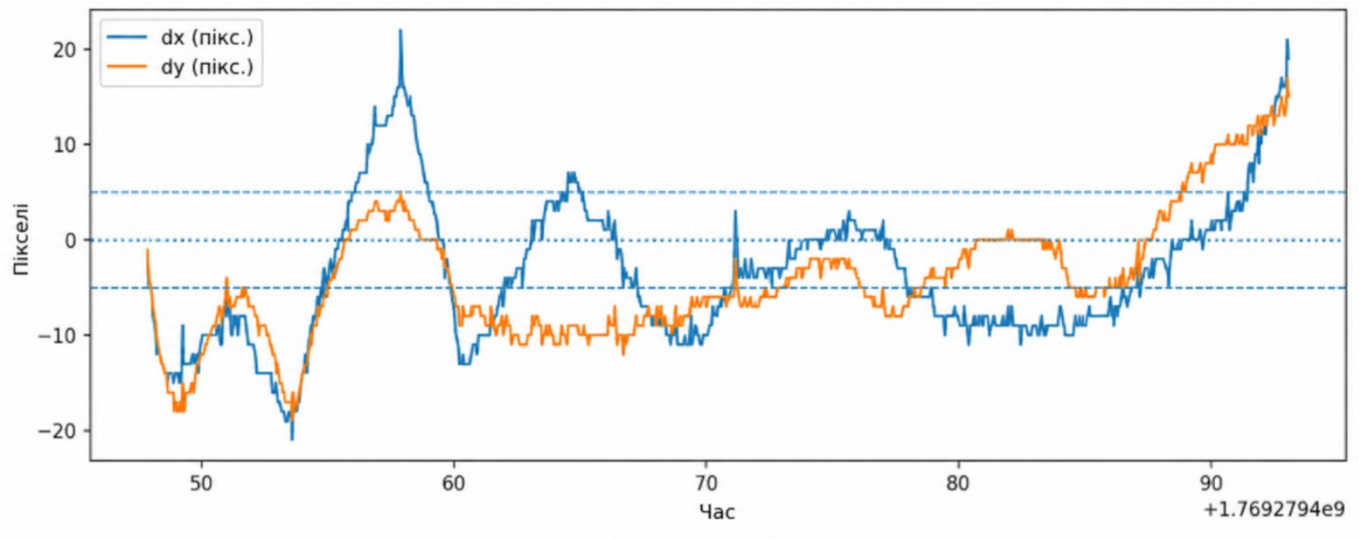


Рис. 3.10. Часові ряди піксельних відхилень d_x та d_y у програмній конфігурації.

У програмному середовищі значення d_x та d_y коливалися поблизу нуля з відносно невеликою амплітудою. Для порівняння з модельованими умовами було проведено натурний експеримент у приміщенні. Часові ряди піксельних відхилень під час натурного експерименту наведено на рисунку 3.11.

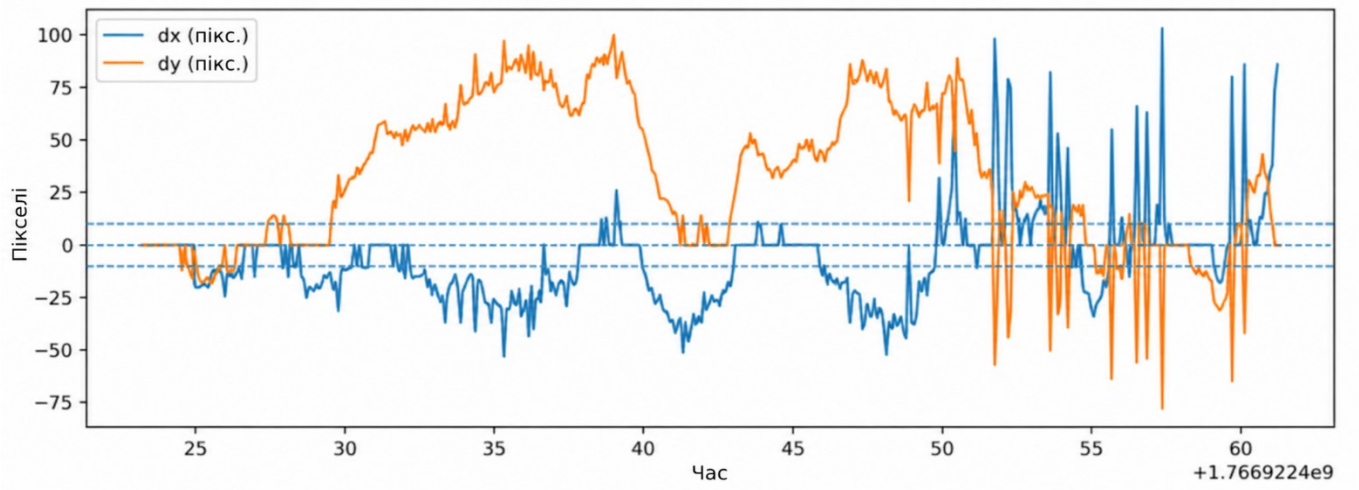


Рис. 3.11. Часові ряди піксельних відхилень d_x та d_y під час натурного експерименту.

У натурному експерименті спостерігалися більші та нерівномірніші відхилення. На характер візуального сигналу могли впливати вібрації БПЛА, неоднорідність спостережуваної поверхні та зміни освітлення. Вплив кожного з цих

чинників окремо в межах проведеного експерименту не оцінювався.

Для кількісного порівняння використовувалися середні абсолютні відхилення, середньоквадратичні похибки та 95-ті процентилі абсолютних відхилень. Узагальнені значення цих показників наведено в таблиці 3.7.

Таблиця 3.7

Показники горизонтальної стабілізації у програмній та натурній конфігураціях

Показник	**Модельований експеримент**	**Натурний експеримент**
Середнє абсолютне відхилення d_x, пікс.	6.78	15.60
Середнє абсолютне відхилення d_y, пікс.	6.55	37.26
Середньоквадратична похибка d_x, пікс.	8.21	23.21
Середньоквадратична похибка d_y, пікс.	7.80	47.68
95-й процентиль d_x, пікс.	15.00	42.00
95-й процентиль d_y, пікс.	14.00	85.75

У програмній конфігурації середні абсолютні відхилення становили 6,78 px для dx та 6,55 px для dy, тоді як у натурному експерименті вони збільшилися до 15,60 та 37,26 px відповідно. Аналогічне збільшення спостерігалося для RMSE та P95. Це свідчить про істотно більшу варіативність візуального сигналу в натурних умовах.

Вертикальний контур стабілізації оцінювався окремо. У програмній конфігурації задавалася висота 5 м. Зміну висоти та характер виходу контуру на задане значення наведено на рисунку 3.12.

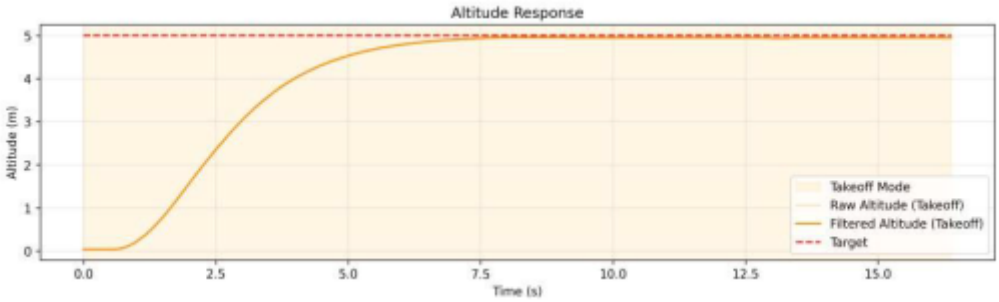


Рис. 3.12. Результати роботи контуру стабілізації висоти Z у програмній конфігурації.

Відгук мав аперіодичний характер та досягав заданої висоти без перерегулювання. Середня похибка становила 0,906 м, максимальна 4,960 м. Максимальне значення визначалося початковою фазою зльоту, коли поточна висота істотно відрізнялася від заданої.

У натурному експерименті задана висота становила 1,5 м. Результати роботи вертикального контуру в реальних умовах наведено на рисунку 3.13.

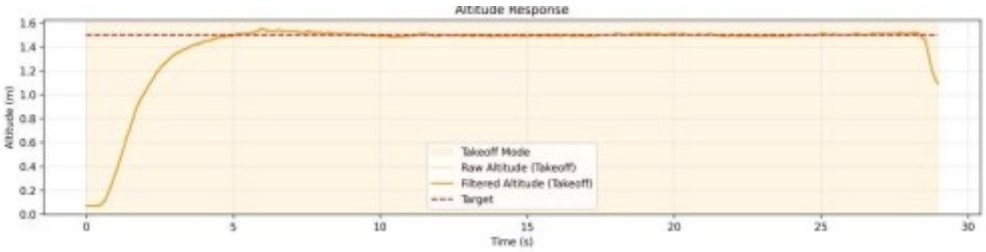


Рис. 3.13. Результати роботи контуру стабілізації висоти Z під час натурного експерименту.

У цьому випадку середня похибка становила 0,096 м, максимальна – 1,43 м, а перерегулювання – 3,3 %. У натурних умовах сигнал керування характеризувався більшою варіативністю. Такий характер сигналу може бути пов’язаний із шумом вимірювання та зовнішніми збуреннями, вплив яких окремо в межах проведеного експерименту не оцінювався

3.8 Висновки до розділу III

Розроблено агентно-орієнтовану модель БПЛА, у якій окремий агент представлено сукупністю локального стану, маршрутного сценарію, контуру керування та правил переходів між станами. Маршрутний сценарій формалізовано як послідовність параметризованих маршрутних кроків, а його виконання організовано на основі автомата станів. Для реалізації маршрутного переміщення використано механізм віртуального дрейфу, який забезпечує перетворення параметрів активного маршрутного кроку у вхідний вплив для контуру візуальної стабілізації без використання глобальних координат.

Для керування положенням БПЛА в умовах відсутності GPS використано сигнал візуального зміщення d_x та d_y, отриманий від модуля оцінювання візуального зміщення за даними нижньої камери. У межах дисертаційного дослідження виконано інтеграцію цього модуля з контуром керування та реалізовано оброблення отриманих значень для формування керуючих впливів польотного контролера.

Експериментальна перевірка маршрутного керування підтвердила можливість виконання однаково заданого маршрутного сценарію у SIL- та HIL-конфігураціях. У дослідженій HIL-серії спостерігалися більші значення похибки та триваліший час виконання маршруту порівняно із SIL-конфігурацією. Отримані результати характеризують досліджені експериментальні умови та вказують на доцільність окремого налаштування параметрів керування для програмної та напівнатурної конфігурацій [94; 95].

Працездатність просторової стабілізації без використання GPS перевірено в модельованих і натурних умовах. У модельованій конфігурації горизонтальні відхилення були меншими та близькими за двома координатними напрямами, тоді як у натурному експерименті спостерігалося їх збільшення та виражена нерівномірність між координатними складовими. Вертикальний контур в обох експериментальних конфігураціях забезпечував вихід на задану висоту та її подальше утримання. Натурні випробування також показали більшу варіативність сигналів за наявності реальних збурень і шуму вимірювання.

Синхронна взаємодія декількох БПЛА-агентів під час одночасного виконання маршрутних сценаріїв у межах цього етапу дослідження окремо не оцінювалася.

РОЗДІЛ IV. МЕТОД ФАЗОВОГО АДАПТИВНОГО PID-КЕРУВАННЯ ПОЛОЖЕННЯМ БПЛА ЗА ВІЗУАЛЬНИМ ЗМІЩЕННЯМ

4.1 Постановка задачі адаптивного керування положенням БПЛА

Керування положенням БПЛА за візуальним зміщенням виконується безпосередньо в координатах зображення. Вхідною інформацією позиційного контуру є поточне візуальне зміщення відносно опорного кадру та цільове зміщення, що відповідає поточному маршрутному завданню. Різниця між ними визначає позиційну похибку у координатах зображення та використовується для формування керуючих впливів у горизонтальній площині. Такий принцип визначення похибки є однаковим для регуляторів із фіксованими та адаптивними коефіцієнтами.

Використання PID-регулятора з постійними коефіцієнтами передбачає незмінність параметрів регулятора протягом усього маршруту. Водночас вимоги до керування змінюються залежно від поточного характеру руху. Під час активного наближення до цільового зміщення поточного маршрутного кроку необхідний достатній керуючий вплив, тоді як поблизу цілі важливішими стають зменшення перерегулювання, пригнічення коливань і стабілізація положення. Тому використання одного набору коефіцієнтів не дає змоги однаково враховувати різні умови функціонування позиційного контуру.

Додатковим фактором є зміна нормованого показника якості візуального оцінювання. За зниження показника якості візуального оцінювання незмінні параметри регулятора можуть спричиняти надмірну реакцію на похибку або небажане накопичення інтегральної складової.

Робота позиційного контуру також обмежується доступним діапазоном керуючих впливів. Під час проходження маршруту один або обидва координатні канали можуть досягати встановленої межі. Подальше збільшення розрахованого керуючого сигналу в такому стані не приводить до відповідного збільшення фактичного впливу на БПЛА. При цьому насичення одного каналу не обов’язково означає вичерпання доступного керуючого впливу за іншою віссю.

З огляду на зазначені особливості задача адаптивного керування полягає у зміні параметрів PID-регулятора відповідно до поточного стану позиційного контуру та умов виконання маршрутного завдання. Під час адаптації враховуються характер руху БПЛА, нормований показник якості візуального оцінювання, параметри маршрутного кроку, розподіл похибки між координатними осями та ознаки насичення керуючих каналів. При цьому спосіб визначення позиційної похибки та штатні низькорівневі контури польотного контролера залишаються незмінними.

Метою розроблення адаптивного методу є підвищення частки успішного завершення маршрутного сценарію з одночасним оцінюванням впливу адаптації на показники позиційної точності під час виконання маршруту. Результативність запропонованого підходу оцінюється шляхом порівняння регулятора з адаптивними коефіцієнтами та регулятора з фіксованими коефіцієнтами за однакових умов моделювання.

4.2 Структура фазового методу адаптивного PID-керування

Запропонований фазовий метод адаптивного PID-керування передбачає перерахунок параметрів позиційного регулятора під час кожного циклу керування. Значення пропорційного, інтегрального та диференціального коефіцієнтів формуються відповідно до поточних умов роботи контуру, тоді як принцип визначення позиційної похибки за візуальним зміщенням залишається незмінним.

Вхідними даними механізму адаптації є позиційна похибка у координатах зображення та її динаміка, нормований показник якості візуального оцінювання, параметри поточного маршрутного кроку, ознака наближення до цільового зміщення маршрутного кроку, поперечна похибка (cross-track error) та ознаки насичення керуючих каналів.

Адаптація охоплює послідовне визначення поточної фази, формування фазової конфігурації коефіцієнтів PID-регулятора, їх коригування за координатними осями, застосування необхідних обмежень та передавання отриманих параметрів позиційним регуляторам за осями x та y. Послідовність виконання фазового методу адаптивного PID-керування положенням БПЛА за візуальним зміщенням наведено на

рисунку 4.1.

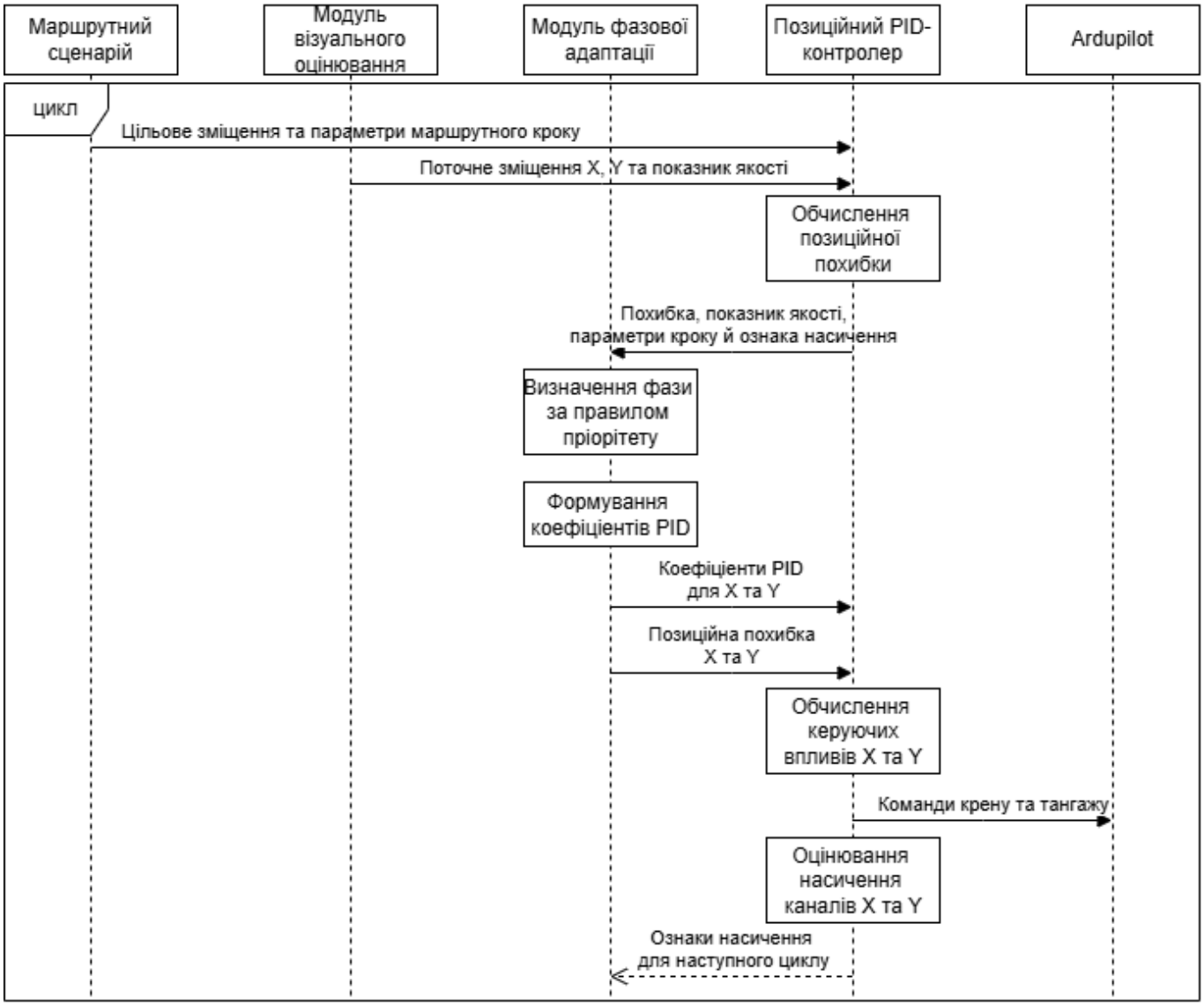


Рис. 4.1. Діаграма послідовності виконання фазового методу адаптивного PID-керування положенням БПЛА за візуальним зміщенням

Метод передбачає п’ять фаз: APPROACH, BRAKING, STABILIZATION, LOW_CONFIDENCE та SATURATION_LIMITED. Поточна фаза визначається повторно на кожному циклі керування за фіксованим пріоритетом умов. Тому фазовий механізм не є автоматом станів із наперед заданою послідовністю переходів, а виконує класифікацію поточного стану позиційного контуру.

У межах запропонованого методу під фазою роботи позиційного контуру розуміється категорія поточних умов його функціонування, яка повторно

визначається на кожному циклі керування та використовується для вибору базової конфігурації коефіцієнтів PID-регулятора.

Визначена фаза задає правило формування фазової конфігурації коефіцієнтів PID-регулятора. Після цього коефіцієнти коригуються окремо для координатних каналів з урахуванням параметрів маршрутного кроку, розподілу позиційної похибки між осями та поперечної похибки відносно маршруту. Отримані значення обмежуються допустимими діапазонами та використовуються позиційними PID-регуляторами протягом поточного циклу керування.

Ознаки насичення керуючих каналів формуються після обчислення керуючих впливів і використовуються як вхідні дані механізму адаптації на наступному циклі. Це утворює зворотний інформаційний зв'язок між використанням доступного діапазону керуючих сигналів і подальшим формуванням параметрів регулятора, що відображено на рис. 4.1.

4.3 Визначення фаз та правил адаптації коефіцієнтів PID-регулятора

Для формалізації умов вибору поточної фази використовується позиційна похибка в координатах зображення та характеристики її зміни. Результуюча величина позиційної похибки визначається евклідовою нормою.

$$e_k = \|e_k\|_2 = \sqrt{e_{x,k}^2 + e_{y,k}^2},$$

де $e_{x,k}$ та $e_{y,k}$– компоненти позиційної похибки за осями x та y на k-му кроці керування; e_k – результуюча величина позиційної похибки, виражена в пікселях.

Для характеристики зміни похибки в часі для кожної осі $j \in \{x, y\}$ обчислюється її дискретна похідна

$$\dot{e}_{j,k} = \frac{e_{j,k} - e_{j,k-1}}{t_k - t_{k-1}},$$

де $\dot{e}_{j,k}$ – швидкість зміни похибки за відповідною віссю; $e_{j,k}$ та $e_{j,k-1}$ – значення похибки на поточному та попередньому кроках керування; t_k та t_{k-1} – відповідні моменти часу.

Якість візуального оцінювання характеризується нормованим показником

$$c_k = \max\left(0, min\left(1, \frac{m_k}{100}\right)\right),$$

де c_k є нормованим показником якості візуального оцінювання на k-му кроці керування; m_k – значення matches_percent, сформоване модулем оцінювання візуального зміщення на k-му циклі керування. Більше значення c_k відповідає вищому рівню якості візуального оцінювання. Цей показник використовується як додаткова характеристика стану позиційного контуру та не змінює спосіб визначення самої похибки.

Поточна фаза ϕ_k визначається за правилом фіксованого пріоритету

$$\phi_k = \begin{cases} SATURATION_LIMITED, & s_{x,k} \vee s_{y,k}, \\ LOW_CONFIDENCE, & c_k < 0{,}55, \\ STABILIZATION, & n_k = 1 \vee e_k \leq 8, \\ BRAKING, & B_k = 1, \\ APPROACH, & \text{в інших випадках,} \end{cases} \qquad (4.1)$$

де $s_{x,k}$ та $s_{y,k}$ – ознаки насичення керуючих каналів відповідно за осями x та y; n_k – ознака наближення поточного візуального зміщення до цільового зміщення маршрутного кроку; B_k – ознака необхідності переходу до фази гальмування. Значення $n_k = 1$ відповідає наближенню поточного візуального зміщення до цільового зміщення маршрутного кроку.

Відповідно до виразу (4.1), найвищий пріоритет має умова насичення керуючих каналів. За її відсутності перевіряється значення показника якості візуального оцінювання, після чого умови стабілізації та гальмування. Якщо жодна з них не виконується, використовується фаза APPROACH. Порогове значення показника якості для LOW_CONFIDENCE становить 0,55, а поріг переходу до STABILIZATION за результуючою позиційною похибкою становить 8 пікселів.

Ознака B_k формується за динамікою похибки:

$$B_k = \begin{cases} 1, & |\dot{e}_{x,k}| \geq 40 \vee |\dot{e}_{y,k}| \geq 40 \vee \dot{e}_{x,k}\dot{e}_{y,k} < 0, \\ 0, & \text{в інших випадках.} \end{cases}$$

Фаза BRAKING активується, якщо абсолютне значення швидкості зміни похибки хоча б за однією з осей досягає порогового значення 40 пікс./с або якщо

похідні похибки за осями x та y мають різні знаки. Остання умова враховує різноспрямовану зміну похибки за двома координатними каналами.

Фаза APPROACH відповідає активному наближенню до цільового зміщення поточного маршрутного кроку. BRAKING відповідає умовам, за яких динаміка позиційної похибки потребує зменшення інтенсивності реакції регулятора. STABILIZATION відповідає роботі поблизу цілі або за малої результуючої похибки. LOW_CONFIDENCE відповідає зниженню нормованого показника якості візуального оцінювання нижче встановленого порогу, а SATURATION_LIMITED відповідає наявності ознаки насичення щонайменше одного керуючого каналу.

Як опорна конфігурація адаптивного механізму використовується набір коефіцієнтів

$$\left(K_{p,b}, K_{i,b}, K_{d,b}\right) = (1{,}2;\ 0{,}02;\ 0{,}60),$$

де $K_{p,b}$, $K_{i,b}$ та $K_{d,b}$ – опорні значення пропорційного, інтегрального та диференціального коефіцієнтів. Ці значення не використовуються як незмінні коефіцієнти позиційного регулятора, а є вихідними параметрами механізму адаптації.

Для основних фаз роботи позиційного контуру базові коефіцієнти визначаються як

$$\left(K_{p,\phi}, K_{i,\phi}, K_{d,\phi}\right) = \begin{cases} (1{,}32;\ 0{,}005;\ 0{,}60), & \phi_k = APPROACH, \\ (1{,}14;\ 0{,}007;\ 0{,}57), & \phi_k = BRAKING, \\ (0{,}72;\ 0{,}011;\ 0{,}54), & \phi_k = STABILIZATION. \end{cases}$$

де $K_{p,\phi}$, $K_{i,\phi}$ та $K_{d,\phi}$ – базові коефіцієнти, що відповідають поточній фазі ϕ_k.

Для APPROACH використовується найбільше серед трьох основних фаз значення пропорційного коефіцієнта. У BRAKING значення пропорційного та диференціального коефіцієнтів зменшуються, а інтегрального дещо збільшується. Для STABILIZATION пропорційний і диференціальний коефіцієнти додатково зменшуються, тоді як інтегральний коефіцієнт має найбільше значення серед трьох основних фаз. Таким чином, зміна фази приводить до дискретної зміни базової конфігурації позиційного PID-регулятора.

Для фази LOW_CONFIDENCE параметри регулятора змінюються залежно від

поточного значення показника якості c_k. За зниження показника якості візуального оцінювання пропорційний коефіцієнт зменшується, інтегральний коефіцієнт встановлюється рівним нулю та припиняється подальше накопичення інтегральної складової, а диференціальний коефіцієнт збільшується в межах установлених обмежень.

Отже, фази APPROACH, BRAKING і STABILIZATION визначають базову конфігурацію коефіцієнтів відповідно до характеру руху БПЛА, тоді як LOW_CONFIDENCE і SATURATION_LIMITED забезпечують зміну параметрів регулятора за обмежувальних умов роботи позиційного контуру.

4.4 Урахування параметрів маршрутного кроку та насичення керуючих сигналів під час адаптації

Після визначення поточної фази коефіцієнти PID-регулятора коригуються окремо для двох координатних каналів з урахуванням параметрів поточного маршрутного кроку, розподілу позиційної похибки між осями та поперечного відхилення від заданого маршруту. Ці параметри не впливають на вибір фази, а використовуються під час формування осьових множників для вже визначеної конфігурації регулятора. Такий порядок відповідає реалізованому механізму адаптації, у якому фазова конфігурація визначається до маршрутного та осьового коригування коефіцієнтів.

Урахування просування за маршрутом виконується за допомогою множника

$$\gamma_k = 1 + 0.02 \min(\max(m_k, 0), 4)$$

де γ_k – коефіцієнт, що враховує номер маршрутного кроку; m_k – індекс поточного маршрутного кроку. Значення γ_k збільшується на 0,02 зі зростанням індексу маршрутного кроку, причому для $m_k \geq 4$ подальше збільшення множника не відбувається.

Для врахування напрямку поточного маршрутного переміщення множник γ_k поєднується з позначенням осі або сукупності осей, за якими визначається завершення маршрутного кроку $q_k \in \{x, y, xy\}$. Масштабувальні коефіцієнти для

координатних каналів визначаються як

$$\left(A_{x,k}, A_{y,k}\right) = \begin{cases} (1.10\gamma_k, 1.20), q_k = \text{x}, \\ (1.20, 1.10\gamma_k), q_k = \text{y}, \\ (\gamma_k, \gamma_k), q_k = \text{xy}. \end{cases}$$

де $A_{x,k}$ та $A_{y,k}$ – масштабувальні коефіцієнти регуляторів за відповідними осями. Для основної осі поточного маршрутного кроку використовується множник 1,10, а для поперечної осі – 1,20.

Крім напрямку маршрутного кроку, під час осьового коригування враховується співвідношення між похибкою відповідної координатної осі, результуючою позиційною похибкою та поперечним відхиленням від маршруту. Для нормування складових похибки за віссю $j \in \{x, y\}$ визначається нормувальний знаменник

$$D_{j,k} = \max\left(\left|e_{j,k}\right|, |e_k|, \left|e_{ct,k}\right|, 1\right)$$

де $D_{j,k}$ – нормувальний знаменник; $e_{j,k}$ – компонента позиційної похибки за віссю j; e_k – результуюча позиційна похибка; $e_{ct,k}$ – поперечна похибка відносно поточного сегмента маршруту.

Нормовані внески позиційної та поперечної похибок визначаються відповідно як

$$r_{j,k} = \min\left(\frac{\left|e_{j,k}\right|}{D_{j,k}}, 1\right)$$

$$r_{\text{ct},k} = \min\left(\frac{\left|e_{\text{ct},k}\right|}{D_{j,k}}, 1\right)$$

На основі отриманих величин для кожного координатного каналу формується осьовий множник

$$M_{j,k} = A_{j,k}\left(0.8 + 0.2r_{j,k} + 0.1r_{ct,k}\right)$$

Отже, $M_{j,k}$ враховує напрямок і номер поточного маршрутного кроку, відносний внесок похибки відповідної осі та поперечної похибки відносно маршруту. При цьому поперечна похибка використовується для коригування коефіцієнтів після визначення поточної фази і не входить до правила її вибору.

Для фаз APPROACH, BRAKING та STABILIZATION пропорційний та інтегральний коефіцієнти коригуються за допомогою осьового множника:

$$K_{p,j,k} = K_{p,\phi} M_{j,k}$$

$$K_{i,j,k} = K_{i,\phi} M_{j,k}$$

тоді як диференціальний коефіцієнт визначається вибраною фазою:

$$K_{d,j,k} = K_{d,\phi}$$

Таким чином, K_p та K_i можуть відрізнятися для координатних каналів x і y, тоді як K_d для цих фаз визначається фазовою конфігурацією. Похідні $\dot{e}_{x,k}$ та $\dot{e}_{y,k}$ використовуються для визначення умови BRAKING, але після вибору фази не створюють додаткового безперервного множника для K_d.

Для фази LOW_CONFIDENCE осьове коригування також виконується з використанням множника $M_{j,k}$, однак результуючі коефіцієнти додатково залежать від поточного значення показника якості візуального оцінювання c_k. Для координатної осі j результуючі коефіцієнти визначаються як

$$K_{p,j,k} = \mathrm{clip}\big(K_{p,b}(0.55 + 0.25c_k)M_{j,k};\ 0;\ 1.68\big)$$

інтегральний коефіцієнт встановлюється рівним

$$K_{i,j,k} = 0,$$

а диференціальний коефіцієнт визначається як

$$K_{d,j,k} = \mathrm{clip}\big(K_{d,b}[1.10 + 0.10(1 - c_k)];\ 0;\ 0.85\big)$$

де функція $clip(z, l, u)$ обмежує значення z інтервалом $[l, u]$, а $M_{j,k}$ — осьовий множник, сформований з урахуванням параметрів поточного маршрутного кроку, розподілу похибки між координатними осями та поперечної похибки відносно маршруту.

Отримані коефіцієнти обмежуються допустимими діапазонами:

$$0 \le K_{p,j,k} \le 1.68$$

$$0 \le K_{i,j,k} \le 0.05$$

$$0 \le K_{d,j,k} \le 0.85$$

Обмеження застосовуються під час формування результуючих параметрів регулятора та запобігають виходу адаптивно сформованих коефіцієнтів за встановлені межі.

Окремо враховується насичення керуючих сигналів. Ознака насичення визначається незалежно для двох координатних каналів. Ознака насичення каналу встановлюється, якщо модуль сформованої для нього керуючої поправки досягає 98 % максимально доступного значення. У фінальній конфігурації гістерезис ознаки насичення не використовується, тому ознака насичення безпосередньо визначається поточним значенням керуючого сигналу відносно встановленого порогу.

Для координатного каналу, що перебуває у стані SATURATION_LIMITED, коефіцієнти визначаються як

$$K_{p,j,k} = 0.95\ \mathrm{clip}\big(0.90M_{j,k}, 0{,}1.68\big)$$

$$K_{i,j,k} = 0$$

$$K_{d,j,k} = 0.85$$

У цьому режимі пропорційний коефіцієнт зменшується з урахуванням осьового масштабування, інтегральний коефіцієнт встановлюється рівним нулю, а диференціальний – 0,85. Одночасно припиняється подальше накопичення інтегральної складової для каналу з активною ознакою насичення. При цьому накопичене раніше значення інтегральної складової не скидається, тобто блокування накопичення не є її обнуленням.

Насичення обробляється незалежно за координатними осями. Якщо межі керуючого впливу досягнуто лише за однією віссю, конфігурація SATURATION_LIMITED застосовується тільки до відповідного каналу. Для каналу без активної ознаки насичення визначається конфігурація, яка відповідала б поточному стану позиційного контуру без урахування ознаки насичення, і зберігається правило формування коефіцієнтів цієї фази. При цьому загальна поточна фаза позначається як SATURATION_LIMITED, якщо ознаку насичення зафіксовано хоча б для одного каналу. Такий підхід не обмежує керуючу дію за другою

координатною віссю лише через активну ознаку насичення першого каналу.

У результаті параметри маршрутного кроку, розподіл позиційної похибки та поперечне відхилення формують окремі осьові множники після визначення поточної фази, а ознака насичення додатково коригує лише той координатний канал, для якого вона активована. Це забезпечує незалежне формування параметрів PID-регуляторів за двома осями без зміни самого правила фазової класифікації

4.5 Програмна реалізація адаптивного PID-керування

Розроблений метод фазового адаптивного PID-керування реалізовано у програмному контурі позиційного керування БПЛА та перевірено в середовищі Software-in-the-Loop, що поєднує Gazebo та ArduPilot SIL.

Модуль оцінювання візуального зміщення використовується як зовнішній функціональний компонент програмного контуру. У межах цього дослідження алгоритми виділення та зіставлення візуальних ознак і визначення геометричного перетворення між зображеннями не розроблялися. Власна програмна реалізація розробленого методу охоплює інтеграцію сформованих сигналів із позиційним контуром та механізм адаптивного формування коефіцієнтів PID-регулятора.

Для обох конфігурацій керування використовується однаковий спосіб формування позиційної похибки: поточне візуальне зміщення порівнюється з цільовим зміщенням, сформованим для поточного маршрутного кроку. Тому режими FIXED_PID і ADAPTIVE_PID відрізняються не джерелом зворотного зв’язку або способом визначення похибки, а механізмом формування коефіцієнтів PID-регулятора. Такий принцип дає змогу змінювати алгоритм регулювання без зміни вхідних даних позиційного контуру.

Програмну реалізацію механізму адаптації винесено до окремого модуля AdaptivePositionGainScheduler. На кожному циклі позиційного керування він отримує компоненти позиційної похибки та їх похідні, результуючу похибку, нормований показник якості візуального оцінювання, поперечну похибку відносно маршруту, інформацію про поточний маршрутний крок і вісь його завершення, ознаку наближення до цільового зміщення поточного маршрутного кроку та ознаки

насичення двох каналів керування. На основі цих даних модуль формує окремі набори коефіцієнтів K_p, K_i та K_d для двох координатних осей, а також ознаки блокування накопичення інтегральної складової. Фактичний інтерфейс програмного модуля відповідає саме цьому набору вхідних і вихідних даних.

Розраховані для двох осей коефіцієнти передаються до позиційного PID-регулятора, який формує окремі керуючі поправки. Результати роботи регуляторів перетворюються на коригувальні значення каналів керування креном і тангажем та через MAVLink передаються до ArduPilot SIL. Автопілот формує низькорівневі керуючі впливи на модель БПЛА, після переміщення якої камера формує наступне зображення. У результаті утворюється замкнений контур, що охоплює модель БПЛА, отримання візуальних даних, оцінювання зміщення, позиційний регулятор та автопілот.

Після формування керуючих поправок у позиційному контролері незалежно оновлюються ознаки насичення каналів крену та тангажу. Отримані ознаки використовуються під час наступного циклу формування адаптивних коефіцієнтів.

Узагальнену структуру замкненого контуру позиційного керування в SIL-середовищі подано на рисунку 4.2.

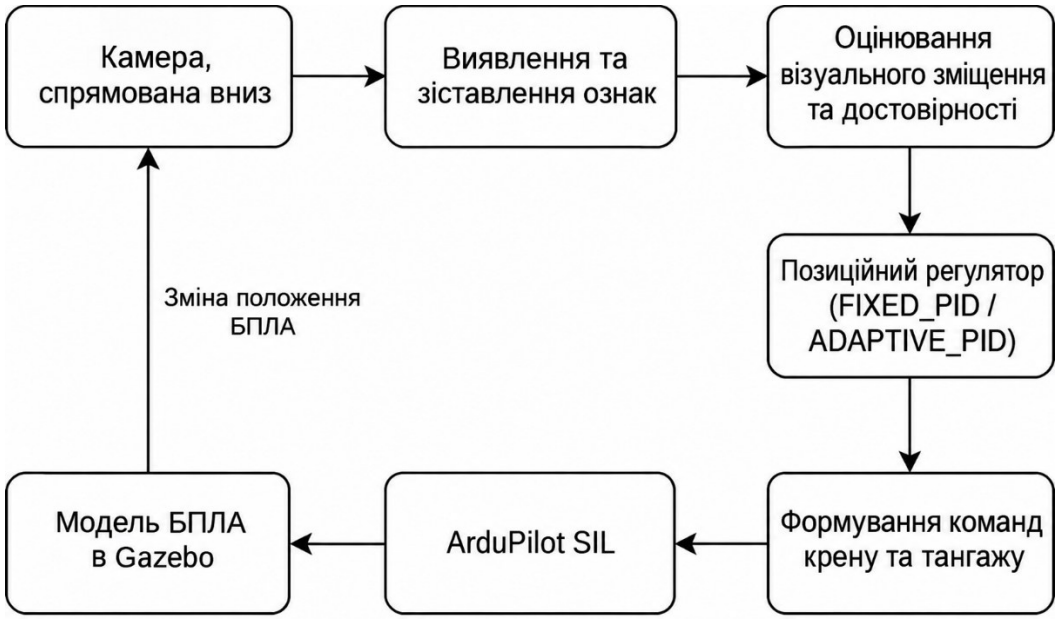


Рис. 4.2. Структура замкненого контуру керування положенням БПЛА за візуальним зміщенням у SIL-середовищі

Під час роботи програмного контуру додатково виконується реєстрація даних експерименту у файлах CSV. Записуються часові характеристики, режим роботи регулятора, дані поточного маршрутного кроку, компоненти візуального зміщення та позиційної похибки, показник якості візуального оцінювання, поточна фаза адаптації, сформовані коефіцієнти PID, керуючі величини, ознаки насичення та ознаки завершення маршруту. Ці дані використовуються для подальшого відновлення траєкторій і розрахунку показників експериментального оцінювання.

4.6 Методика експериментального дослідження

Експериментальне дослідження проведено з метою порівняння PID-регулятора з фіксованими коефіцієнтами та розробленого фазового адаптивного PID-регулятора за однакових умов моделювання. Для цього використовувалися дві конфігурації позиційного керування: FIXED_PID, у якій коефіцієнти регулятора залишалися незмінними протягом усього запуску, та ADAPTIVE_PID, у якій коефіцієнти формувалися відповідно до розробленого методу. В обох випадках використовувалися однакові середовище моделювання, геометрія маршруту, джерело сигналу зворотного зв’язку, спосіб визначення позиційної похибки та процедура реєстрації експериментальних даних. Відмінність між конфігураціями полягала в механізмі визначення коефіцієнтів PID-регулятора.

Експериментальна перевірка розробленого фазового методу адаптивного PID-керування виконувалася виключно в програмному середовищі Software-in-the-Loop на основі Gazebo та ArduPilot SIL. Апаратні компоненти HIL до цього експерименту не залучалися. Фінальна серія складалася зі 100 запусків: 50 для конфігурації FIXED_PID і 50 для ADAPTIVE_PID.

Для фінального порівняння використовувався один наперед заданий квадратний маршрут розміром 1000×1000 пікселів. Маршрут формувався з кроком 100 пікселів і містив 40 послідовних маршрутних кроків. Використання однакової геометрії маршруту для обох конфігурацій забезпечувало порівняння регуляторів за однакових маршрутних умов. Основні параметри експериментального дослідження наведено в Таблиці 4.1.

Таблиця 4.1

Основні параметри експериментального дослідження

Параметр	**Значення**
Середовище моделювання	Gazebo / ArduPilot SIL
Конфігурації регулятора	FIXED_PID, ADAPTIVE_PID
Кількість запусків для кожної конфігурації	50
Загальна кількість запусків	100
Геометрія маршруту	квадратна
Розмір маршруту	1000×1000 пікселів
Крок маршруту	100 пікселів
Кількість маршрутних кроків	40
Джерело позиційного зворотного зв'язку	візуальне зміщення за даними спрямованої вниз камери
Одиниці позиційної похибки	пікселі
Ідентифікатори псевдовипадкових послідовностей	10001–10050;
Експериментальна одиниця	один повний експериментальний запуск

Для організації експериментальних серій використовувалися наперед задані ідентифікатори псевдовипадкових послідовностей 10001–10050. Їх використання забезпечувало узгоджену організацію запусків, однак не розглядалося як умова повної детермінованості Gazebo та ArduPilot SIL. Тому окремі запуски за однакових заданих умов могли відрізнятися за фактичною часовою реалізацією процесу моделювання.

Параметри обох регуляторів визначалися під час попередніх налагоджувальних експериментів і фіксувалися до початку фінальної серії. Для FIXED_PID використовувалися коефіцієнти $K_p = 1{,}2$, $K_i = 0{,}05$ та $K_d = 0{,}8$, однакові для двох координатних каналів. Для ADAPTIVE_PID на цьому етапі уточнювалися фазові коефіцієнти, допустимі межі коефіцієнтів PID-регулятора, параметри осьового масштабування, порогове значення показника якості та пороги похибки, а також параметри, пов'язані з насиченням керуючих сигналів. Після вибору фінальних конфігурацій їх параметри не змінювалися протягом валідаційної серії. Попередні

запуски, використані для налаштування регуляторів, до фінальної вибірки не включалися.

Під час етапу розроблення додатково використовувалися трикутні та зигзагоподібні маршрутні конфігурації для перевірки працездатності регулятора за інших характерів руху. Ці запуски не входили до збалансованого порівняння FIXED_PID та ADAPTIVE_PID і не використовувалися для отримання кількісних результатів фінальної валідації.

Кожний валідаційний запуск являв собою повне виконання заданого маршруту однією з двох конфігурацій регулятора. Перед виконанням запуску ініціалізувалися процеси моделювання та ArduPilot SIL, після чого БПЛА виконував послідовність заданих маршрутних кроків. Запуск вважався успішно завершеним, якщо були виконані всі 40 маршрутних кроків і програмним модулем керування було зареєстровано подію завершення маршруту. Якщо повний маршрут не було завершено до припинення експерименту, запуск класифікувався як незавершений. Максимальна тривалість одного запуску становила 480 с.

Окремий експериментальний запуск приймався за одну експериментальну одиницю. Послідовні значення, зареєстровані в межах одного запуску на окремих циклах керування, утворювали один часовий ряд і не розглядалися як незалежні експериментальні спостереження. На основі цих значень для кожного запуску обчислювалися узагальнені показники, які використовувалися для подальшого порівняння конфігурацій FIXED_PID та ADAPTIVE_PID.

Для формалізації експериментального оцінювання окремий запуск позначається індексом r. Нехай N_r – кількість валідних відліків траєкторії в запуску r, а $e_{r,k}$ – величина позиційної похибки у координатах зображення для k-го відліку поточного запуску.

Середня позиційна похибка для окремого запуску визначається як

$$\bar{e}_r = \frac{1}{N_r} \sum_{k=1}^{N_r} e_{r,k}$$

Середньоквадратична похибка визначається як

$$\mathrm{RMSE}_r = \sqrt{\frac{1}{N_r}\sum_{k=1}^{N_r} e_{r,k}^2}$$

Для оцінювання верхньої частини розподілу похибки використовується 95-й процентиль

$$e_{95,r} = Q_{0.95}\{e_{r,1}, \dots, e_{r,N_r}\}$$

де $Q_{0.95}$ – квантиль рівня 0,95 для сукупності значень позиційної похибки відповідного запуску.

Кінцева позиційна похибка визначається за останнім валідним відліком запуску:

$$e_{\mathrm{final},r} = e_{r,N_r}$$

Середня позиційна похибка, RMSE, 95-й процентиль та кінцева похибка виражаються в пікселях.

Для окремого оцінювання відхилення від заданого маршруту використовується поперечна похибка. Нехай $d_{r,k}$ – мінімальна відстань між поточною точкою фактичної траєкторії та відповідним сегментом заданого маршруту для k-го відліку запуску r. Позначення $d_{r,k}$ використовується для експериментального оцінювання поперечної похибки в межах окремого запуску. Середня поперечна похибка визначається як

$$\bar{d}_r = \frac{1}{N_r}\sum_{k=1}^{N_r} d_{r,k}$$

Значення $d_{r,k}$ та $\bar{d}_r$виражаються в пікселях. Такий показник характеризує відхилення від заданого сегмента маршруту, а не відстань до поточної цільової точки.

Надійність проходження маршруту в межах цього дослідження оцінюється часткою успішно завершених запусків. Успішність окремого запуску розглядається як бінарний результат. Частка успішного завершення для сукупності з R запусків визначається як

$$C_{\text{rate}} = \frac{100}{R}\sum_{r=1}^{R} C_r$$

де C_{rate} – частка успішно завершених маршрутів, виражена у відсотках; R – загальна кількість запусків відповідної конфігурації; C_r – бінарний індикатор результату запуску, для якого значення 1 відповідає успішному завершенню маршруту, а 0 – незавершеному запуску.

Час проходження маршруту T_rвизначається в секундах лише для успішно завершених запусків і відповідає інтервалу часу, необхідному для виконання всіх заданих маршрутних кроків.

Окремо оцінюється частка циклів з активною ознакою насичення керуючих каналів. Для запуску r визначається як

$$S_r = \frac{100}{N_r^c}\sum_{k=1}^{N_r^c} I\left(s_{x,r,k} \vee s_{y,r,k}\right)$$

де S_r – частка циклів з активною ознакою насичення для запуску r, виражена у відсотках; N_r^c – кількість валідних циклів позиційного керування в цьому запуску; $s_{x,r,k}$ та $s_{y,r,k}$ – бінарні ознаки насичення керуючих каналів за осями x та yна k-му циклі; $I(\cdot)$ - індикаторна функція. Отже, S_r характеризує відсоток циклів позиційного керування, під час яких щонайменше для одного з двох каналів була активною ознака насичення.

Показники позиційної похибки, середня поперечна похибка, час проходження маршруту та частка циклів в насиченні аналізувалися лише для успішно завершених запусків. Частка успішного завершення маршруту визначалася за всіма 50 запусками кожної конфігурації. Частка циклів з активною ознакою насичення розглядалася окремо від показників позиційної похибки, оскільки вона характеризує режим роботи регулятора та використання доступного керуючого ресурсу, а не безпосередню величину позиційної похибки.

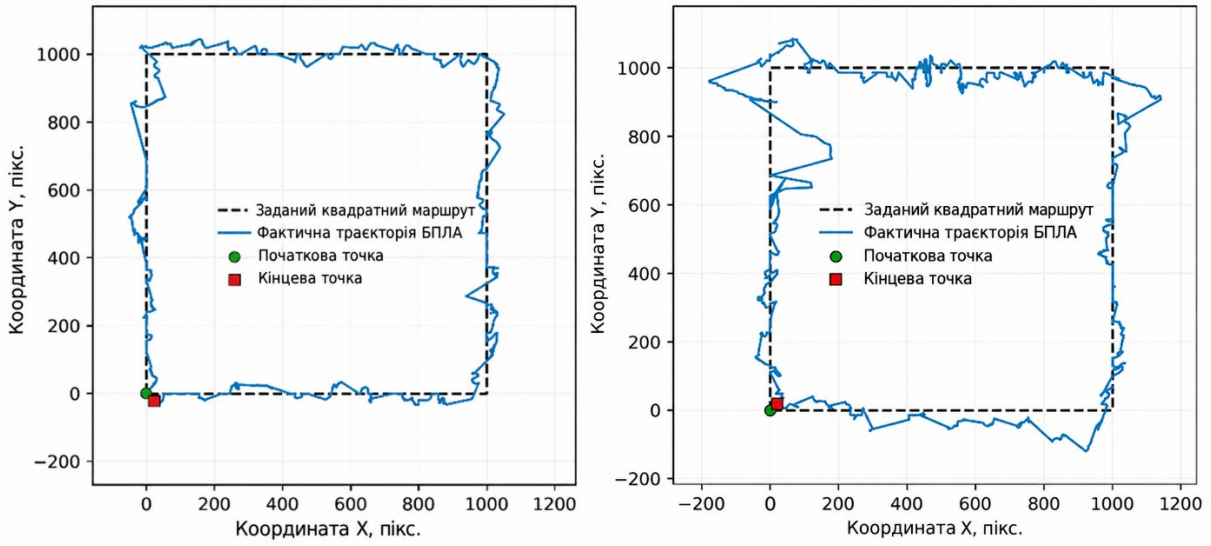
Для перевірки статистичної значущості відмінностей між конфігураціями FIXED_PID та ADAPTIVE_PID використовували рівень значущості $\alpha = 0.05$.

Частки успішного завершення маршруту порівнювали за допомогою двобічного точного критерію Фішера [96]. Неперервні показники окремих запусків порівнювали за допомогою двобічного критерію Манна–Вітні, що не потребує припущення про нормальність розподілів [97]. Для сукупності із семи неперервних показників застосовували поправку Холма на множинні порівняння [98]. Для подальшого аналізу зберігали як вихідні, так і скориговані за методом Холма значення p [99].

4.7 Порівняльне оцінювання PID-регуляторів із фіксованими та адаптивними коефіцієнтами

4.7.1 Оцінювання успішності проходження маршруту

Для ілюстрації характеру виконання заданого маршрутного сценарію на рисунку 4.3 наведено приклади успішно завершених траєкторій для конфігурацій ADAPTIVE_PID та FIXED_PID. В обох випадках БПЛА виконав повний квадратний маршрут, сформований послідовністю маршрутних кроків. Наведені траєкторії відображають окремі успішні запуски та використовуються для візуального представлення характеру руху, а не для кількісного порівняння точності двох конфігурацій.



а *б*

Рис. 4.3. Приклади успішно завершених траєкторій БПЛА: а – ADAPTIVE_PID; б – FIXED_PID

Наведені приклади показують, що обидві конфігурації здатні забезпечити виконання повного заданого маршруту. Водночас окремі траєкторії не характеризують загальну частку успішних запусків і не дають підстав для висновку про перевагу однієї конфігурації за точністю відстеження. Для оцінювання здатності регуляторів завершувати маршрут використано результати всієї валідаційної серії, наведені в Таблиці 4.2.

Таблиця 4.2

Результати завершення маршрутів

Результат запуску	**FIXED_PID**	**ADAPTIVE_PID**
Успішно завершено	35	46
Не завершено	15	4
Частка завершення, %	70	92

Таким чином, спостережувана частка успішного завершення маршруту для ADAPTIVE_PID була на 22 відсоткові пункти вищою.

Отриманий результат характеризує відмінність між конфігураціями за здатністю повністю завершувати заданий маршрут у досліджених умовах моделювання.

4.7.2 Оцінювання позиційної похибки під час виконання маршруту

Показники позиційної похибки для успішно завершених запусків FIXED_PID ($n = 35$) та ADAPTIVE_PID ($n = 46$) наведено в таблиці 4.3. Значення подано як середнє арифметичне та стандартне відхилення для відповідних сукупностей запусків.

Таблиця 4.3

Показники позиційної похибки

Показник	**FIXED_PID**	**ADAPTIVE_PID**	**Зміна, %**
Середня похибка, пікс.	150,11 ± 30,83	142,12 ± 23,03	−5,32
RMSE, пікс.	167,11 ± 32,39	158,99 ± 22,61	−4,86
P95 похибки, пікс.	278,57 ± 50,02	269,74 ± 38,93	−3,17

Кінцева похибка, пікс.	134,87 ± 64,76	129,72 ± 73,68	−3,82
Середня поперечна похибка, пікс	33,77 ± 8,40	33,85 ± 8,20	+0,24

Для ADAPTIVE_PID усі чотири показники позиційної похибки мали нижчі середні значення, ніж для FIXED_PID. Відносне зменшення становило від 3,17 до 5,32 %.

Середня поперечна похибка практично не відрізнялася між конфігураціями та становила 33,77 пікс. для FIXED_PID і 33,85 пікс. для ADAPTIVE_PID. Відносна різниця дорівнювала 0,24 %. Отже, за описовими показниками використання ADAPTIVE_PID супроводжувалося числовим зменшенням основних показників позиційної похибки без помітної зміни середнього поперечного відхилення від заданого маршруту.

Розподіл значень RMSE для успішно завершених запусків FIXED_PID та ADAPTIVE_PID наведено на рисунку 4.4. Розподіли значень RMSE для двох конфігурацій суттєво перекриваються, що візуально узгоджується з відсутністю статистично значущої відмінності за цим показником.

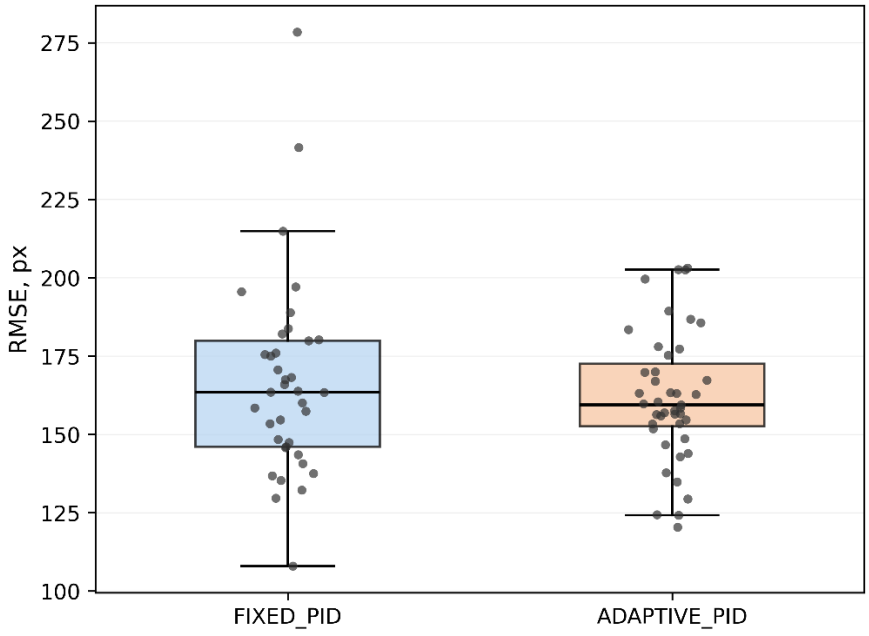


Рис. 4.4. Розподіл RMSE для успішно завершених запусків FIXED_PID та ADAPTIVE_PID

Отримані відмінності за показниками похибки мають невелику величину, тому

їх недостатньо для висновку про підвищення позиційної точності під час виконання маршруту лише на підставі середніх значень.

4.7.3 Оцінювання часу проходження маршруту

Розподіли часу проходження маршруту для успішно завершених запусків FIXED_PID та ADAPTIVE_PID наведено на рисунку 4.5. Для FIXED_PID середній час проходження маршруту становив 144,63±15,26с, тоді як для ADAPTIVE_PID — 141,35±31,00с. Таким чином, середнє значення для адаптивної конфігурації було нижчим на 3,28 с, що відповідає 2,27 %.

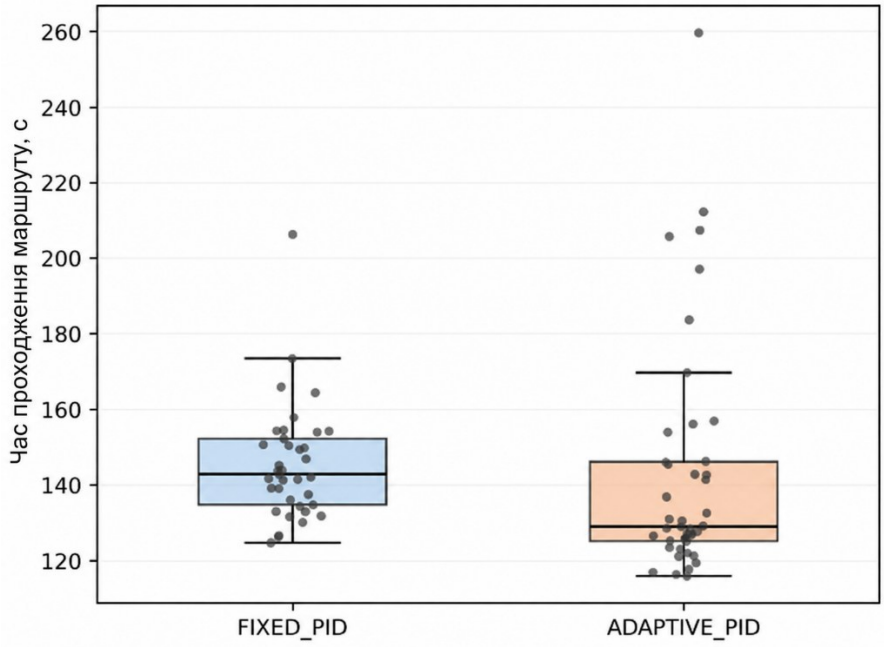


Рис. 4.5. Розподіл часу проходження маршруту для успішно завершених запусків FIXED_PID та ADAPTIVE_PID

Поряд із меншою середньою тривалістю для ADAPTIVE_PID спостерігався більший розкид результатів між окремими запусками. Стандартне відхилення часу становило 31,00 с для ADAPTIVE_PID порівняно з 15,26 с для FIXED_PID. Отже, зменшення середнього значення часу не супроводжувалося зменшенням міжзапускової варіативності.

Розподіли часу проходження маршруту для двох режимів відрізнялися не лише за середнім значенням, а й за характером розсіювання результатів. Це свідчить про

те, що оцінювання часу проходження маршруту лише за середнім значенням є недостатнім для характеристики поведінки адаптивного регулятора.

4.7.4 Оцінювання насичення керуючих сигналів

Для успішно завершених запусків середня частка циклів з ознакою насичення становила 52,60±4,66% для FIXED_PID та 48,00±4,84% для ADAPTIVE_PID. Таким чином, для адаптивної конфігурації середнє значення цього показника було нижчим на 4,60 відсоткового пункту.

Розподіли частки роботи керуючих каналів у стані насичення наведено на рисунку 4.6. Для ADAPTIVE_PID розподіл зміщений у бік нижчих значень порівняно з FIXED_PID, що відповідає зменшенню середньої частки циклів, виконаних на межі доступного керуючого впливу.

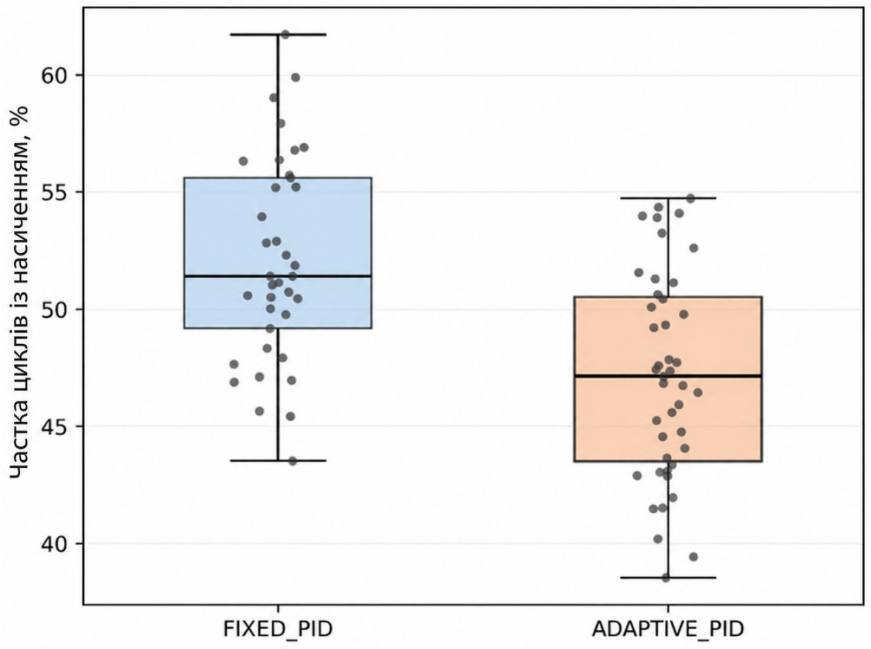


Рис. 4.6. Розподіл частки роботи керуючих каналів у стані насичення для успішно завершених запусків FIXED_PID та ADAPTIVE_PID

Отримані результати свідчать про те, що під час успішно завершених запусків ADAPTIVE_PID рідше працював на межі доступного керуючого сигналу. При цьому цей результат характеризує саме режим використання керуючого ресурсу і не є безпосереднім показником позиційної точності під час виконання маршруту.

4.7.5 Статистичний аналіз результатів експерименту

За результатами двобічного точного критерію Фішера відмінність між частками успішного завершення маршруту для FIXED_PID та ADAPTIVE_PID була статистично значущою.

Результати статистичного порівняння неперервних показників для успішно завершених запусків FIXED_PID (n=35) та ADAPTIVE_PID (n=46) наведено в Таблиці 4.4.

Таблиця 4.4

Результати статистичного порівняння неперервних показників

Показник	p	p **після поправки Холма**
Середня похибка	0,255	1,000
RMSE	0,328	1,000
P95 похибки	0,545	1,000
Кінцева похибка	0,584	1,000
Середня поперечна похибка	0,973	1,000
Час проходження маршруту	0,0046	0,027
Частка циклів з ознакою насичення	0,00013	< 0,001

Після застосування поправки Холма статистично значущих відмінностей для середньої похибки, RMSE, P95 похибки, кінцевої похибки та середньої поперечної похибки не виявлено. Для всіх п’яти показників скориговане значення pстановило 1,000. Отже, за результатами проведеного експерименту відмінності між конфігураціями за цими показниками не отримали статистичного підтвердження.

Для часу проходження маршруту скориговане значення становило $p = 0{,}027$, що вказує на статистично значущу відмінність між розподілами цього показника для FIXED_PID та ADAPTIVE_PID. Водночас статистична значущість не характеризує величину практичного ефекту, тому цей результат слід інтерпретувати з урахуванням абсолютної різниці значень та їх варіативності.

Для частки циклів з активною ознакою насичення керуючих каналів отримано

скориговане значення $p < 0{,}001$. Отже, відмінність між FIXED_PID та ADAPTIVE_PID за цим показником зберегла статистичну значущість після врахування множинних порівнянь.

Таким чином, статистично значущі відмінності між досліджуваними конфігураціями встановлено за часткою успішного завершення маршруту, часом його проходження та часткою циклів з активною ознакою насичення керуючих каналів. Для показників позиційної похибки статистично значущих відмінностей не виявлено.

4.7.6 Аналіз функціонування фазового механізму адаптації

Для додаткового аналізу внутрішньої роботи адаптивного регулятора було досліджено розподіл циклів позиційного керування між фазами, що визначалися під час виконання маршрутного сценарію. На відміну від показників похибки, часу проходження маршруту та насичення, такий аналіз характеризує не кінцевий результат запуску, а фактичні режими роботи механізму адаптації.

Для узгодженості з аналізом неперервних показників основну вибірку сформовано з 46 успішно завершених запусків ADAPTIVE_PID. У сукупності для них зареєстровано 147 446 циклів позиційного керування. Для кожного запуску окремо визначали частку циклів, протягом яких активною була кожна з фаз APPROACH, BRAKING, STABILIZATION, LOW_CONFIDENCE та SATURATION_LIMITED, після чого отримані значення узагальнювали за всією вибіркою. Результати наведено в Таблиці 4.5.

Таблиця 4.5

Розподіл циклів позиційного керування між фазами для успішних запусків ADAPTIVE_PID

Фаза	**Середня частка циклів, %**	**Стандартне відхилення, %**	**Медіана, %**
APPROACH	18,78	2,91	18,80
BRAKING	31,20	3,22	30,85
STABILIZATION	3,30	1,56	2,88
LOW_CONFIDENCE	0,00	0,00	0,00
SATURATION_LIMITED	46,73	4,59	46,60

У структурі циклів позиційного керування переважали фази SATURATION_LIMITED і BRAKING, тоді як STABILIZATION займала найменшу частку. Такий розподіл показує, що під час проходження дослідженого маршруту регулятор переважну частину часу працював у режимах активного руху, гальмування або врахування обмежень керуючого впливу.

У проведених експериментах фаза LOW_CONFIDENCE не активувалася, оскільки візуальний модуль у використаному модельованому середовищі стабільно забезпечував значення нормованого показника якості вище встановленого порогу $c_k = 0.55$. Це пов’язано з достатньою кількістю візуально розрізнюваних ознак сцени для стійкого оцінювання міжкадрового зміщення..

Оскільки поточна фаза визначалася повторно на кожному циклі позиційного керування, додатково було проаналізовано переходи між фазами. Під час 46 успішних запусків зафіксовано 50679 змін активної фази. Найхарактерніші напрямки переходів наведено на рисунку 4.7 у вигляді нормалізованої матриці переходів.

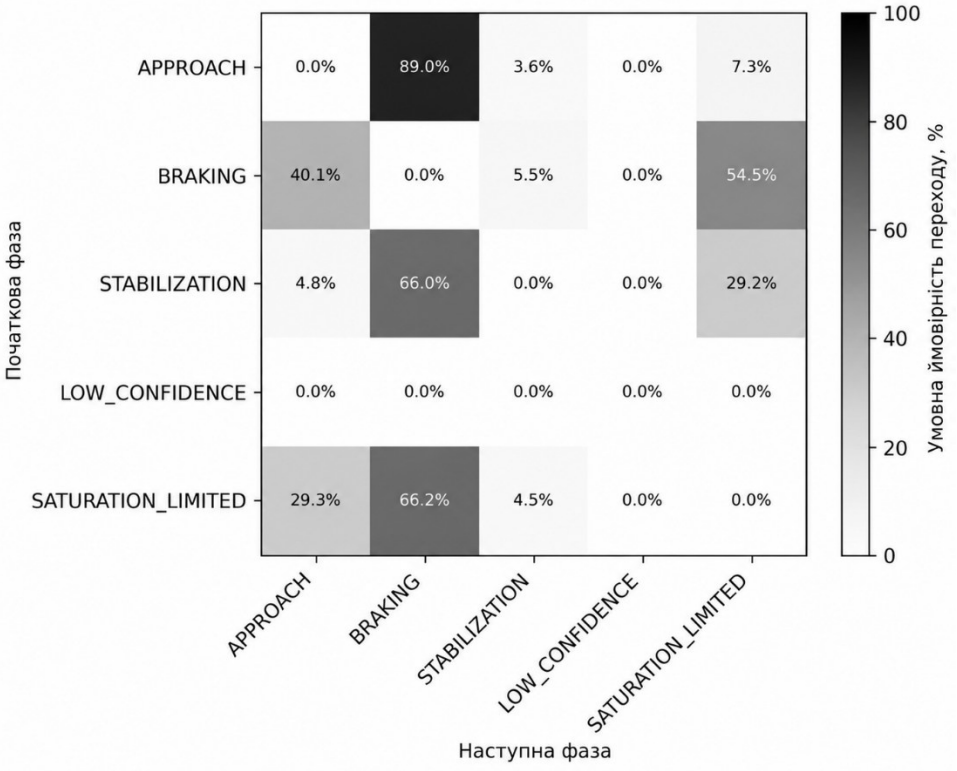


Рис. 4.7. Нормалізована матриця переходів між фазами для успішних запусків ADAPTIVE_PID

Серед усіх виходів із фази APPROACH у 89,03 % випадків наступною ставала BRAKING, тоді як перехід до SATURATION_LIMITED становив 7,34 %. Для BRAKING найбільш характерним був перехід до SATURATION_LIMITED – 54,48 % усіх змін із цієї фази; у 40,06 % випадків регулятор повертався до APPROACH. Після SATURATION_LIMITED у 66,20 % випадків активувалася BRAKING, а у 29,32 % – APPROACH. Для STABILIZATION також переважав перехід до BRAKING, частка якого становила 65,99 %.

Отримана структура переходів показує, що фази не утворювали наперед заданої послідовності, за якою регулятор обов'язково проходив би APPROACH, BRAKING і STABILIZATION. Натомість активний режим змінювався відповідно до поточного стану позиційного контуру. Зокрема, спостерігалися повторні переходи між APPROACH і BRAKING, а також між BRAKING і SATURATION_LIMITED. Така поведінка відповідає реалізованому принципу повторного визначення фази на кожному циклі керування на основі поточних ознак стану системи.

Висока частка SATURATION_LIMITED сама по собі не характеризує якість відстеження траєкторії та не означає, що протягом усіх відповідних циклів керуючий сигнал перебував безпосередньо на встановленій межі. Фаза відображає роботу фазового механізму за умов, пов'язаних з обмеженням доступного керуючого впливу, тоді як ознаки насичення координатних каналів реєструвалися окремо. Тому взаємозв'язок між вибором цієї фази та активністю ознак насичення потребує окремого аналізу.

Таким чином, аналіз експериментальних журналів підтвердив перемикання між фазами APPROACH, BRAKING, STABILIZATION та SATURATION_LIMITED під час проходження маршруту. Найбільша частка циклів припадала на фази SATURATION_LIMITED і BRAKING, тоді як STABILIZATION використовувалася протягом відносно невеликої частини часу. У проаналізованих циклах маршрутного позиційного керування фаза LOW_CONFIDENCE не активувалася, тому її використання під час проходження заданого маршруту за результатами цієї вибірки не оцінювалося. Отримані результати характеризують роботу фазового механізму в

досліджених умовах, але не дають змоги визначити окремий внесок кожної фази у відмінності між FIXED_PID та ADAPTIVE_PID.

4.7.7 Аналіз зміни коефіцієнтів PID-регулятора

Для оцінювання фактичної роботи механізму адаптації додатково проаналізовано значення коефіцієнтів PID-регулятора, що застосовувалися під час проходження маршруту в режимі ADAPTIVE_PID. Аналіз виконано для 46 успішно завершених запусків і тих самих циклів маршрутного позиційного керування, що використовувалися під час аналізу фазового механізму. Для кожного запуску та кожної активної фази окремо визначали середні значення пропорційного, інтегрального та диференціального коефіцієнтів за двома координатними каналами, після чого отримані значення узагальнювали за всією вибіркою.

На відміну від базових фазових конфігурацій, визначених структурою методу, фактично застосовані коефіцієнти додатково залежать від осьового масштабування, параметрів поточного маршрутного кроку, розподілу позиційної похибки між координатними каналами та поперечного відхилення від маршруту. Тому наведені далі значення в Таблиці 4.6 характеризують результуючі параметри регулятора, сформовані механізмом адаптації безпосередньо під час виконання маршрутного сценарію.

Таблиця 4.6

Фактично застосовані коефіцієнти PID-регулятора за фазами та координатними каналами

Фаза	**Вісь**	K_p	K_i	K_d
APPROACH	X	1,566 ± 0,017	0,00596 ± 0,00007	0,600
	Y	1,572 ± 0,017	0,00599 ± 0,00007	0,600
BRAKING	X	1,365 ± 0,010	0,00838 ± 0,00006	0,570
	Y	1,367 ± 0,009	0,00840 ± 0,00006	0,570
STABILIZATION	X	0,855 ± 0,020	0,01306 ± 0,00030	0,540
	Y	0,851 ± 0,023	0,01301 ± 0,00036	0,540
SATURATION_LIMITED	X	1,192 ± 0,039	0,00328 ± 0,00083	0,732 ± 0,031

	Y	1,157 ± 0,027	0,00250 ± 0,00061	0,760 ± 0,022

Примітка. Наведено середнє значення ± стандартне відхилення серед середніх значень відповідного коефіцієнта в окремих успішних запусках

Для наочного порівняння характеру зміни коефіцієнтів між фазами їх середні значення за двома координатними каналами наведено на рисунку 4.8.

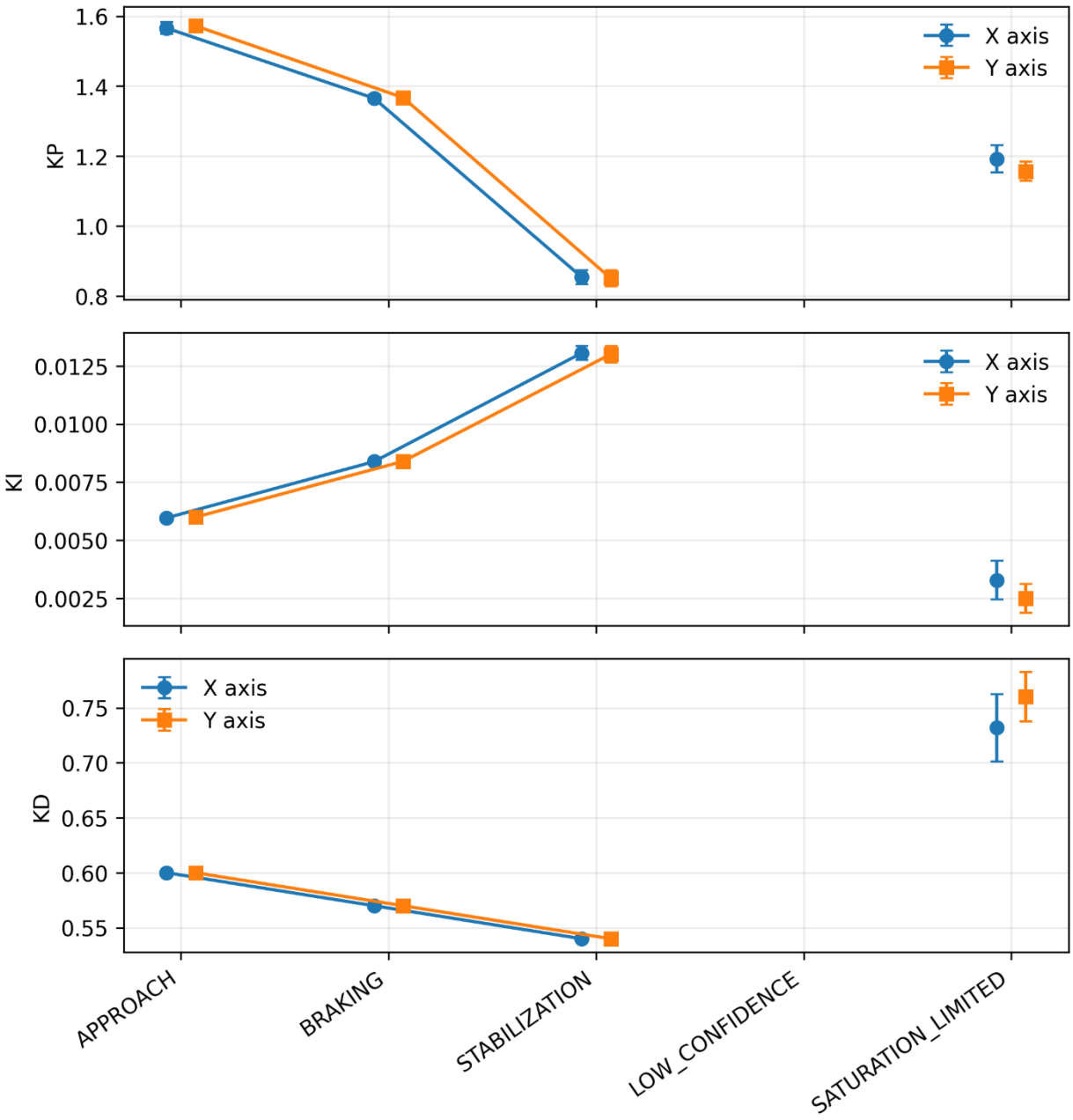


Рис. 4.8. Фактично застосовані коефіцієнти PID-регулятора за фазами адаптації для успішних запусків ADAPTIVE_PID

Як видно з таблиці 4.6 та рисунку 4.8, для основних фаз роботи позиційного контуру APPROACH, BRAKING і STABILIZATION зберігається передбачена структурою методу закономірність зміни параметрів регулятора. Під час переходу від активного наближення до цільового зміщення маршрутного кроку до режиму стабілізації зменшується пропорційна складова регулятора, тоді як інтегральна складова збільшується. Диференціальний коефіцієнт для цих трьох фаз визначається вибраною фазовою конфігурацією та не відрізняється між координатними каналами.

Водночас значення K_p та K_i навіть у межах однієї фази не є абсолютно сталими. Наявний розкид зумовлений тим, що після визначення базової фазової конфігурації коефіцієнти додатково коригуються окремо для координатних каналів. У цьому коригуванні враховуються параметри поточного маршрутного кроку, внесок похибки відповідної осі та поперечне відхилення від заданого маршруту. Отже, фактична конфігурація PID-регулятора визначається не лише назвою активної фази, а й поточним станом позиційного контуру.

Найбільш виражені відмінності між координатними каналами спостерігалися у фазі SATURATION_LIMITED. На відміну від APPROACH, BRAKING і STABILIZATION, у цьому режимі змінювався не лише пропорційний та інтегральний, а й результуючий диференціальний коефіцієнт. Крім того, для осей X та Y спостерігалися різні середні значення параметрів. Такий характер зміни узгоджується із закладеним у методі принципом незалежного врахування насичення за двома координатними каналами, за якого обмежувальна конфігурація може застосовуватися до кожної осі окремо.

Фаза LOW_CONFIDENCE до таблиці 4.6 та рисунку 4.8 не включена, оскільки в проаналізованих циклах маршрутного позиційного керування вона не активувалася. Відповідно, для цієї фази за результатами розглянутої вибірки відсутня сукупність фактично застосованих коефіцієнтів, придатна для аналогічного кількісного аналізу.

Таким чином, аналіз експериментальних журналів показав, що в режимі ADAPTIVE_PID фактично застосовані коефіцієнти змінювалися як унаслідок перемикання між фазами, так і внаслідок додаткового осьового коригування після

вибору фази. Найбільш виражена осьова відмінність параметрів спостерігалася в SATURATION_LIMITED, що узгоджується з реалізованим механізмом незалежного врахування насичення координатних каналів. Водночас цей аналіз характеризує фактичну зміну параметрів регулятора і сам по собі не визначає внесок окремих конфігурацій коефіцієнтів у підсумкові показники проходження маршруту.

4.7.8 Аналіз роботи керуючих каналів в умовах насичення

Загальне порівняння конфігурацій FIXED_PID та ADAPTIVE_PID показало статистично значуще зменшення частки циклів з активною ознакою насичення керуючих каналів для адаптивного регулятора. Для деталізації цього результату додатково проаналізовано розподіл циклів з активною ознакою насичення між фазами адаптивного керування. Аналіз виконано для 46 успішно завершених запусків ADAPTIVE_PID з використанням 147 446 циклів маршрутного позиційного керування, тобто тієї самої вибірки, що й під час аналізу фазового механізму та фактично застосованих коефіцієнтів PID-регулятора.

Для кожного запуску та кожної активної фази окремо визначали частку циклів, у яких ознака насичення була зафіксована щонайменше для одного з керуючих каналів. Додатково визначали частки насичення каналу крену, каналу тангажу та одночасного насичення обох каналів. Після цього показники узагальнювали за 46 успішними запусками. Результати наведено в таблиці 4.7.

Таблиця 4.7

Частка циклів з активною ознакою насичення керуючих каналів у різних фазах ADAPTIVE_PID

Фаза	**Середня частка циклів, %**	**Стандартне відхилення, %**	**Медіана, %**
APPROACH	3.81	1.50	3.53
BRAKING	27.21	5.02	26.31
STABILIZATION	15.59	6.01	15.44
SATURATION_LIMITED	81.19	1.21	81.07

Примітка. Для успішно завершених запусків ADAPTIVE_PID окремо визначено частку циклів відповідної фази, у яких була активною ознака насичення щонайменше одного керуючого каналу. Фазу LOW_CONFIDENCE не наведено, оскільки в проаналізованих циклах маршрутного позиційного керування вона не активувалася.

Як видно з таблиці 4.7, частка циклів з активною ознакою насичення керуючих каналів розподілялася між фазами нерівномірно. Найбільша його концентрація спостерігалася у фазі SATURATION_LIMITED, тоді як для APPROACH, BRAKING та STABILIZATION частка таких циклів була істотно нижчою. Для наочного представлення цієї відмінності розподіл частки циклів із насиченням наведено на рисунку 4.9.

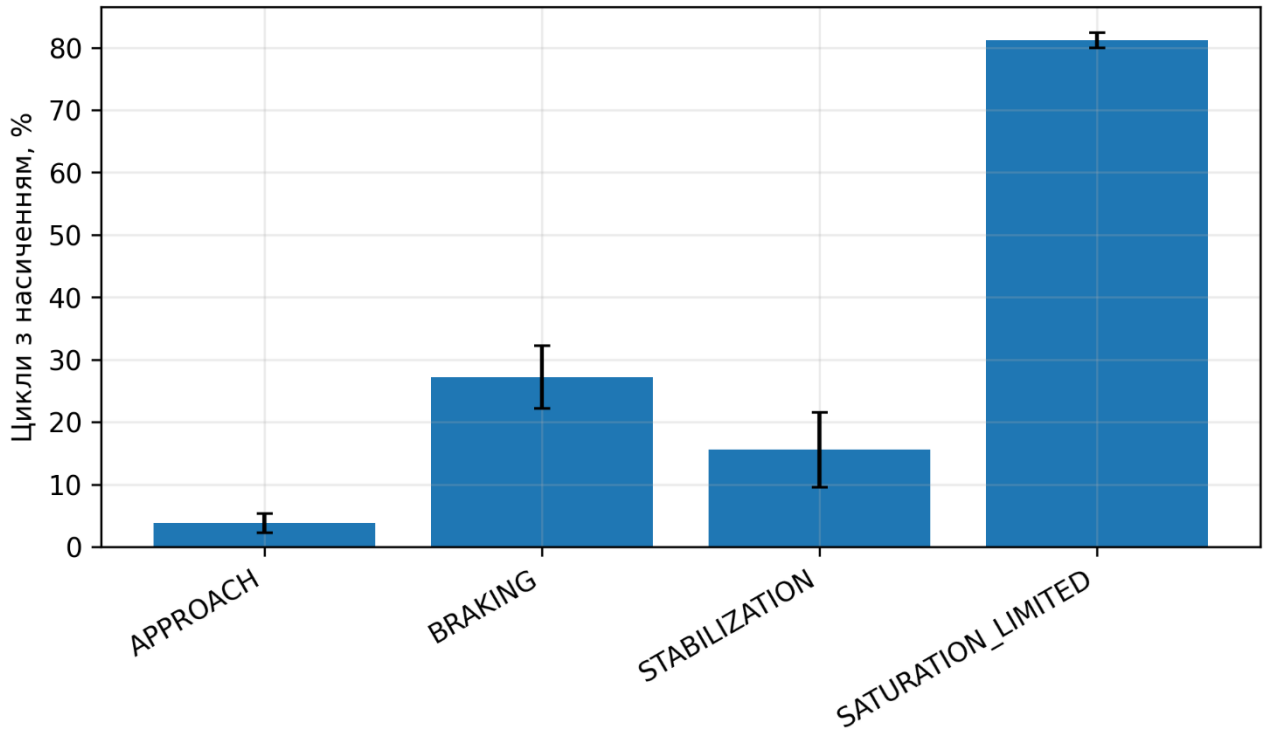


Рис. 4.9. Частка циклів із насиченням щонайменше одного керуючого каналу в різних фазах ADAPTIVE_PID

Отриманий розподіл узгоджується з призначенням SATURATION_LIMITED як фази роботи за умов обмеженого керуючого впливу. Водночас частка циклів з активною ознакою насичення у цій фазі не досягала 100 %, тому SATURATION_LIMITED не слід ототожнювати з безпосередньою бінарною ознакою насичення на поточному циклі. Фаза характеризує режим роботи механізму адаптації, тоді як ознака насичення окремого керуючого каналу реєструється незалежно.

Додатковий аналіз за координатними каналами показав, що насичення крену та тангажу виникало з різною частотою, а одночасне насичення обох каналів

спостерігалося рідше, ніж насичення хоча б одного з них. Розподіл насичення за окремими каналами наведено на рис. 4.10.

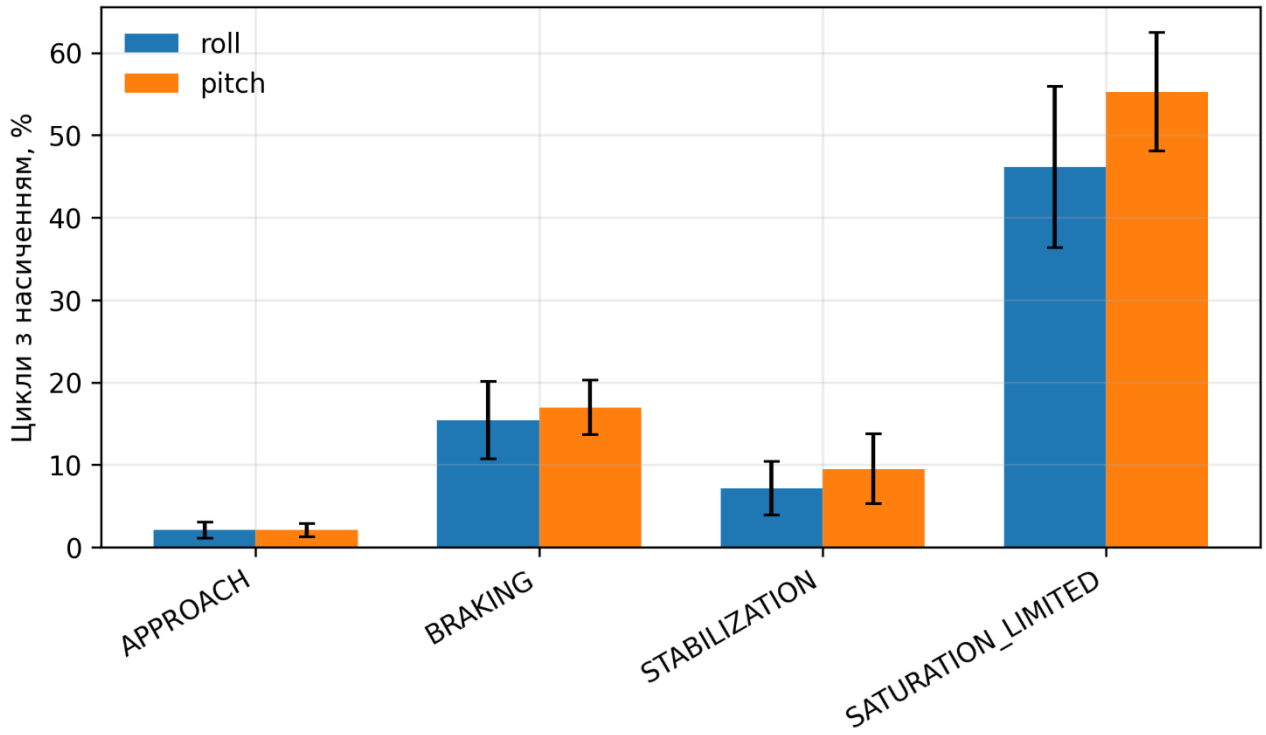


Рис. 4.10. Частка циклів із насиченням каналів крену та тангажу в різних фазах ADAPTIVE_PID

Найбільш виражена відмінність між каналами спостерігалася у фазі SATURATION_LIMITED. Такий характер роботи узгоджується з реалізованим принципом незалежного врахування насичення за двома координатними каналами, відповідно до якого обмежувальна конфігурація може застосовуватися до кожної осі окремо. Водночас наявність цієї відповідності не дає підстав пов'язувати з окремим механізмом насичення загальне зменшення частки роботи на межі керуючого впливу для ADAPTIVE_PID порівняно з FIXED_PID.

Таким чином, аналіз експериментальних журналів показав, що фактичне насичення керуючих каналів було найбільш сконцентроване у фазі SATURATION_LIMITED та мало осьово неоднорідний характер. Отримані результати деталізують роботу механізму незалежного врахування насичення, але не визначають його окремий внесок у підсумкові відмінності між FIXED_PID та

ADAPTIVE_PID.

4.7.9 Аналіз характеристик керування за сегментами маршруту

Для деталізації результатів порівняння FIXED_PID та ADAPTIVE_PID додатково проаналізовано характеристики керування на окремих сегментах заданого квадратного маршруту. Маршрут складався з 40 послідовних кроків і був поділений на чотири сторони по 10 кроків: перша сторона відповідала крокам 1–10, друга – 11–20, третя – 21–30, четверта – 31–40. Такий поділ дає змогу оцінити, чи спостережувані відмінності між конфігураціями однаково проявлялися на всьому маршруті або залежали від його окремої ділянки.

Сегментний аналіз виконано лише для успішно завершених запусків, тобто для 35 запусків FIXED_PID та 46 запусків ADAPTIVE_PID. Для уникнення непропорційного впливу запусків із більшою кількістю зареєстрованих циклів показники спочатку обчислювали окремо для кожного запуску та кожної сторони маршруту, після чого отримані значення узагальнювали за відповідною сукупністю запусків. Таким чином, кожний успішний запуск мав однакову вагу під час формування сегментних оцінок.

Основні значення RMSE та тривалості проходження окремих сторін маршруту наведено в таблиці 4.8.

Таблиця 4.8

Показники проходження окремих сторін маршруту

Сторона	**RMSE FIXED_PID, пікс.**	**RMSE ADAPTIVE_PID, пікс.**	**Час FIXED_PID, с**	**Час ADAPTIVE_PID, с**
1	178.79 ± 29.60	159.58 ± 27.59	34,07 ± 8,70	31,91 ± 10,97
2	158.00 ± 64.00	154.40 ± 56.51	44,21 ± 11,61	49,52 ± 29,31
3	162.52 ± 39.82	153.45 ± 22.09	31,32 ± 3,62	29,24 ± 3,22
4	148.79 ± 35.56	143.53 ± 34.87	32,49 ± 4,02	28,14 ± 3,97

За RMSE середні значення для ADAPTIVE_PID були нижчими на всіх чотирьох сторонах маршруту. Найбільша відносна відмінність спостерігалася на першій стороні, де RMSE зменшився з 178,79 ± 29,60 пікс. для FIXED_PID до 159,58 ± 27,59

пікс. для ADAPTIVE_PID. Після застосування поправки Холма статистично значущу відмінність за RMSE встановлено лише для першої сторони маршруту $p = 0.011$; для інших сторін статистично значущих відмінностей не встановлено. Отже, сегментний аналіз виявив локальну відмінність на окремій ділянці маршруту, але не дає підстав стверджувати про систематичне підвищення точності позиційного керування на маршруті в цілому.

Інший характер спостерігався для часу проходження окремих сторін. Відмінності були найбільш вираженими на третій і четвертій сторонах маршруту, для яких ADAPTIVE_PID забезпечував меншу середню тривалість проходження. На другій стороні, навпаки, середнє значення для адаптивної конфігурації було більшим і характеризувалося значною міжзапусковою варіативністю. Таким чином, встановлена раніше відмінність у загальному часі проходження маршруту не розподілялася рівномірно між його окремими частинами.

Окремо проаналізовано частку циклів роботи в стані насичення. Її розподіл за сторонами маршруту наведено на рисунку 4.11.

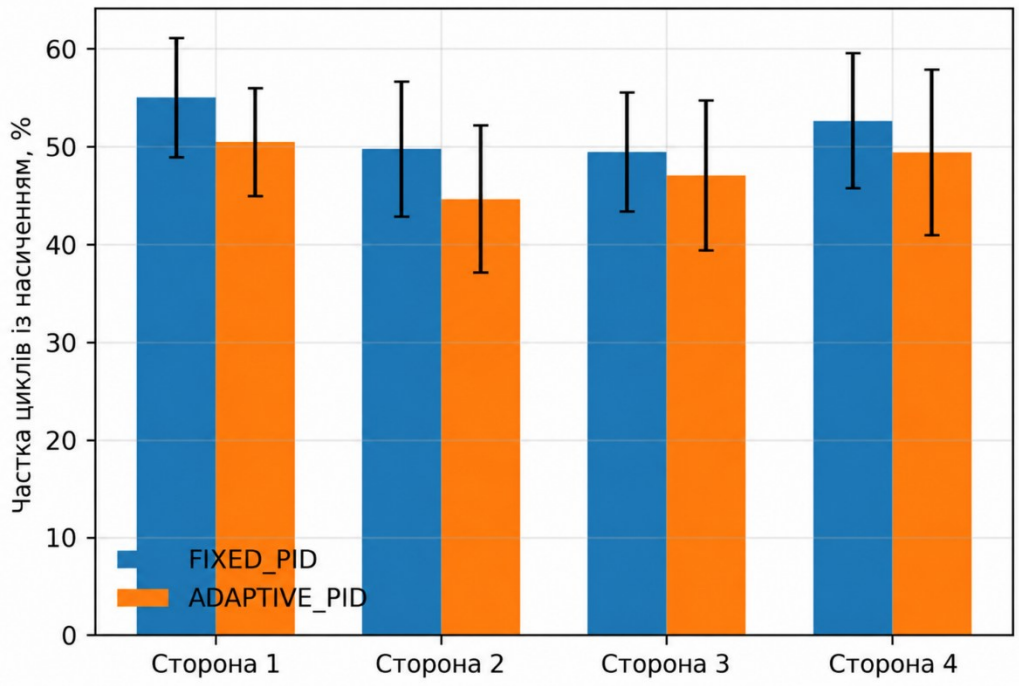


Рис. 4.11. Частка роботи керуючих каналів у стані насичення на окремих сторонах маршруту

Для ADAPTIVE_PID менші середні значення частки роботи в насиченні спостерігалися на всіх чотирьох сторонах маршруту. Отже, загальне зменшення цього показника, встановлене в підрозділі 4.7.4, не було пов'язане виключно з однією окремою ділянкою маршруту. Водночас величина відмінності змінювалася між сторонами, що вказує на неоднорідний характер використання доступного керуючого ресурсу під час виконання різних частин маршрутного сценарію.

Для перевірки статистичної значущості локальних відмінностей неперервні показники порівнювали за допомогою двобічного критерію Манна–Вітні на рівні окремих запусків. Для кожного показника поправку Холма застосовували до чотирьох порівнянь, що відповідали сторонам маршруту. Такий аналіз мав додатковий характер і не замінював загального статистичного порівняння конфігурацій, наведеного в підрозділі 4.7.5. Статистично значущі після поправки Холма сегментні відмінності наведено в таблиці 4.9.

Таблиця 4.9

Статистично значущі відмінності між FIXED_PID та ADAPTIVE_PID на окремих сторонах маршруту

Показник	**Сторона**	**FIXED_PID**	**ADAPTIVE_PID**	**Зміна**	p_{Holm}
Середня позиційна похибка, пікс.	1	164,12 ± 26,96	146,79 ± 23,33	−10,55 %	0,009
RMSE позиційної похибки, пікс.	1	178,79 ± 29,60	159,58 ± 27,59	−10,75 %	0,011
P95 позиційної похибки, пікс.	1	288,80 ± 47,57	257,66 ± 57,40	−10,78 %	0,009
Середня поперечна похибка, пікс.	2	35,38 ± 14,76	30,26 ± 17,12	−14,49 %	0,013
Час проходження, с	3	31,32 ± 3,62	29,24 ± 3,22	−6,65 %	0,012
	4	32,49 ± 4,02	28,14 ± 3,97	−13,39 %	< 0,001
Частка циклів з ознакою насичення, %	1	54,98 ± 6,06	50,46 ± 5,49	−4,52 п.п	0,005
	2	49,75 ± 6,87	44,64 ± 7,52	−5,12 п.п.	0,010

Результати таблиці 4.9 показують, що статистично підтверджені локальні відмінності стосувалися різних характеристик на окремих ділянках маршруту. Для першої сторони статистично значущі відмінності встановлено за середньою позиційною похибкою, RMSE та 95-м процентилем позиційної похибки, причому для

ADAPTIVE_PID значення цих показників були нижчими. Для другої сторони статистично значущою була відмінність за середньою поперечною похибкою. Для третьої та четвертої сторін відмінності встановлено за часом проходження, а для першої та другої сторін – за часткою циклів з ознакою насичення. Водночас локальний характер установлених відмінностей не дає підстав поширювати їх на маршрут у цілому.

Для ADAPTIVE_PID додатково проаналізовано розподіл фаз керування між окремими сторонами маршруту. Результати наведено на рисунку 4.12.

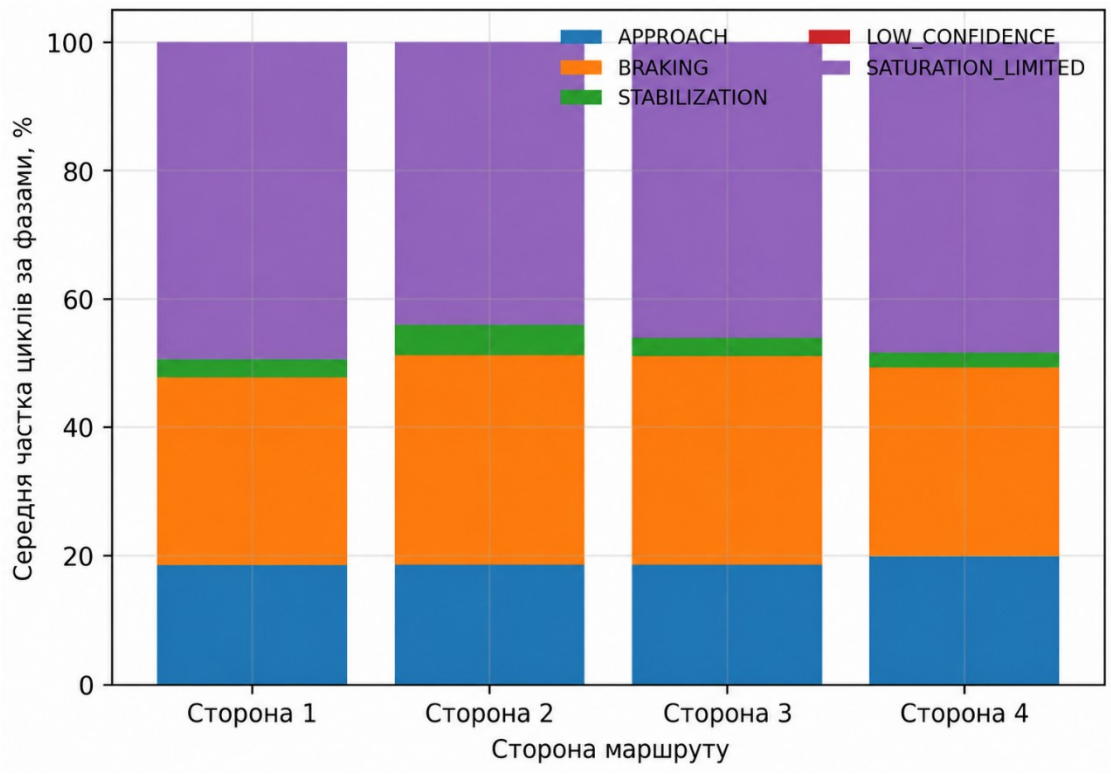


Рис. 4.12. Розподіл фаз адаптивного керування на окремих сторонах маршруту

Структура фаз залишалася загалом подібною на всіх чотирьох сторонах, однак їх співвідношення змінювалося залежно від ділянки маршруту. На першій та четвертій сторонах спостерігалася більша частка SATURATION_LIMITED, тоді як на другій стороні частка цієї фази була меншою, а STABILIZATION – більшою. Фази APPROACH і BRAKING були присутні на всіх сторонах із близькими за порядком

величини частками. Такий розподіл відображає зміну співвідношення фаз залежно від ділянки маршруту та характеризує неоднорідність режимів роботи адаптивного регулятора під час виконання маршрутного сценарію.

Аналіз насичення всередині окремих фаз також виявив залежність між напрямком руху та тим, який із двох керуючих каналів частіше досягав встановленої межі. Для першої та третьої сторін у фазі SATURATION_LIMITED частіше спостерігалося насичення каналу крену, тоді як для другої та четвертої – каналу тангажу. Цей результат узгоджується з осьовою структурою маршрутного керування, однак сам по собі не встановлює причинного зв'язку між напрямком руху та величиною підсумкових показників.

Таким чином, сегментний аналіз показав, що відмінності між FIXED_PID та ADAPTIVE_PID розподілялися вздовж маршруту нерівномірно. На першій стороні маршруту для ADAPTIVE_PID встановлено статистично значуще зменшення середньої позиційної похибки, RMSE та 95-го процентиля позиційної похибки, тоді як на другій стороні статистично значущим було зменшення середньої поперечної похибки. На третій і четвертій сторонах встановлено відмінності за часом проходження, а на першій і другій – за часткою циклів з ознакою насичення. При цьому менша середня частка циклів з ознакою насичення для ADAPTIVE_PID спостерігалася на всіх чотирьох сторонах маршруту. Отримані результати деталізують загальне порівняння двох конфігурацій, однак характеризують лише досліджений квадратний маршрут і не дають підстав узагальнювати локальні відмінності на маршрут у цілому.

4.8 Обговорення результатів та обмежень розробленого методу

Експериментальне порівняння показало, що на рівні маршруту в цілому відмінності між конфігураціями проявилися насамперед у частці успішного завершення маршруту, часі його проходження та частці роботи керуючих каналів у стані насичення, тоді як за агрегованими показниками позиційної похибки статистично значущих відмінностей не встановлено. Сегментний аналіз водночас виявив окремі локальні відмінності: на першій стороні маршруту для

ADAPTIVE_PID встановлено статистично значуще зменшення середньої позиційної похибки, RMSE та 95-го процентиля позиційної похибки, а на другій стороні – середньої поперечної похибки. Ці відмінності не спостерігалися систематично на всіх сегментах маршруту. Отриманий результат слід інтерпретувати як відмінність у здатності системи завершувати заданий маршрут та характері використання доступного керуючого ресурсу, а не як статистично підтверджене підвищення точності позиційного керування на маршруті в цілому. Серед успішно завершених запусків використання ADAPTIVE_PID не супроводжувалося статистично значущим погіршенням показників позиційної похибки.

Спостережуване зменшення частки роботи в насиченні узгоджується із закладеним у методі принципом незалежного врахування насичення за координатними каналами. Водночас фінальна конфігурація ADAPTIVE_PID оцінювалася як єдине ціле: в експерименті окремо не виключалися фазове налаштування коефіцієнтів, коригування за нормованим показником якості візуального оцінювання, маршрутне та осьове масштабування або оброблення насичення. Тому спостережувані відмінності не можна однозначно пов’язати з будь-яким окремим механізмом адаптації, а внесок кожного з них у отриманий результат не визначався.

Для часу проходження маршруту встановлено статистично значущу відмінність між конфігураціями, однак середні значення відрізнялися лише на 2,27 %, а для ADAPTIVE_PID спостерігалася більша міжзапускова варіативність. Крім того, цей показник визначався лише для успішно завершених маршрутів, тоді як частки завершення для двох конфігурацій були різними. Отже, отриману відмінність слід розглядати як характеристику успішних запусків, а не як підтвердження загального підвищення швидкості виконання маршруту адаптивним регулятором.

Межі узагальнення отриманих результатів визначаються умовами проведеної експериментальної перевірки. Кількісна валідація виконувалася для одного наперед заданого квадратного маршруту. Інші геометрії використовувалися лише під час розроблення та налагодження регулятора і не входили до збалансованої фінальної

вибірки. Тому отримані результати не підтверджують збереження встановлених експериментальних характеристик методу для довільної геометрії маршруту.

Експериментальна перевірка розробленого методу проводилася виключно в середовищі Gazebo та ArduPilot SIL. Натурні та HIL-експерименти для цього методу не виконувалися. Тому отримані результати характеризують поведінку регулятора в досліджених умовах програмного моделювання та не підтверджують збереження тих самих характеристик за фізичного польоту. Зокрема, у проведеній валідації окремо не досліджувалися вплив реального шуму камери, змін освітлення, розмиття зображення під час руху, зовнішніх збурень, затримок обміну даними та відмінностей між змодельованою і реальною динамікою БПЛА.

Позиційна похибка в експериментах визначалась у координатах зображення та виражалась у пікселях. Відповідно, наведені показники характеризують відхилення в просторі зображення і не є безпосередньою оцінкою просторової похибки в метричних одиницях. Це обмеження необхідно враховувати під час перенесення отриманих результатів на системи, у яких якість позиційного керування оцінюється за фізичними координатами.

Проведене експериментальне дослідження також не є формальним доведенням стійкості замкненого контуру керування. Воно характеризує спостережувану поведінку реалізованого методу за визначених умов моделювання, але не встановлює гарантій його поведінки для довільних маршрутів, зовнішніх збурень або інших умов функціонування. Подальша експериментальна перевірка має передбачати збалансоване дослідження декількох маршрутних сценаріїв, контрольовані зміни якості візуального оцінювання та зовнішніх збурень, а також перевірку із застосуванням фізичних компонентів БПЛА.

4.9 Висновки до розділу IV

Розроблено фазовий метод адаптивного PID-керування положенням БПЛА за візуальним зміщенням, у якому коефіцієнти регулятора перераховуються на кожному циклі позиційного керування. Метод поєднує визначення поточної фази, коригування коефіцієнтів залежно від нормованого показника якості візуального оцінювання,

параметрів маршрутного кроку, розподілу похибки між координатними осями та поперечної похибки відносно маршруту. Насичення керуючих сигналів враховується незалежно за двома координатними каналами, що дає змогу застосовувати обмежувальну конфігурацію лише до координатного каналу з активною ознакою насичення.

Фінальну конфігурацію розробленого методу реалізовано у програмному контурі позиційного керування та експериментально перевірено в середовищі Gazebo та ArduPilot SIL. Фінальна валідація складалася зі 100 запусків за узгоджених умов моделювання: 50 для регулятора з фіксованими коефіцієнтами FIXED_PID та 50 для фазового адаптивного регулятора ADAPTIVE_PID. HIL- та натурні експерименти під час перевірки розробленого методу не проводилися.

Для FIXED_PID успішно завершено 35 із 50 запусків, що відповідає 70 %, тоді як для ADAPTIVE_PID успішно завершено 46 із 50 запусків, або 92 %. Частка успішного завершення маршруту для адаптивного регулятора була вищою на 22 відсоткові пункти. Відмінність між частками завершення маршруту була статистично значущою за двобічним точним критерієм Фішера.

Для успішно завершених запусків середня частка циклів з активною ознакою насичення керуючих каналів становила 52,60±4,66% для FIXED_PID та 48,00±4,84% для ADAPTIVE_PID, тобто була нижчою для адаптивної конфігурації на 4,60 відсоткового пункту. Відмінність залишилася статистично значущою після поправки Холма ($p<0,001$). Водночас за агрегованими для маршруту в цілому показниками позиійної похибки статистично значущих відмінностей не встановлено. Сегментний аналіз водночас виявив окремі локальні відмінності: на першій стороні маршруту для ADAPTIVE_PID встановлено статистично значуще зменшення середньої позиційної похибки, RMSE та 95-го процентиля позиційної похибки, а на другій стороні – середньої поперечної похибки. Ці відмінності не спостерігалися систематично на всіх сегментах маршруту. Тому отримані результати не підтверджують статистично значущого зменшення позиційної похибки під час виконання маршруту в цілому.

Аналіз внутрішньої роботи адаптивного механізму показав фактичне

перемикання між фазами APPROACH, BRAKING, STABILIZATION та SATURATION_LIMITED під час виконання маршрутного сценарію. У проаналізованих циклах маршрутного позиційного керування найбільша частка припадала на SATURATION_LIMITED і BRAKING. Фактично застосовані коефіцієнти PID-регулятора змінювалися як унаслідок вибору поточної фази, так і внаслідок додаткового осьового коригування. Найбільша частка циклів з активною ознакою насичення керуючих каналів спостерігалася у фазі SATURATION_LIMITED, причому його розподіл відрізнявся між каналами крену та тангажу.

Середній час проходження успішно завершеного маршруту становив 144,63±15,26с для FIXED_PID та 141,35±31,00с для ADAPTIVE_PID. Разом із тим відносна різниця середніх значень становила лише 2,27 %, а для ADAPTIVE_PID спостерігалася більша міжзапускова варіативність, тому цей результат не слід трактувати як суттєве підвищення швидкості проходження маршруту. Сегментний аналіз показав, що відмінності між FIXED_PID та ADAPTIVE_PID розподілялися вздовж маршруту нерівномірно, а менша середня частка циклів з активною ознакою насичення для ADAPTIVE_PID спостерігалася на всіх чотирьох його сторонах. Отримані висновки обмежуються експериментальною перевіркою в SIL-середовищі для заданої геометрії маршруту та оцінюванням позиційної похибки у координатах зображення, тому не можуть бути безпосередньо поширені на довільні маршрути або фізичний БПЛА.

ВИСНОВКИ

У дисертаційній роботі вирішено актуальне наукове завдання щодо розроблення моделей і методів напівнатурного моделювання мультиагентних систем БПЛА та локального керування БПЛА-агентами, що забезпечують спільне функціонування програмних і апаратних компонентів у модельованому середовищі, формалізоване виконання маршрутних сценаріїв, просторову стабілізацію в умовах відсутності GPS та адаптацію параметрів позиційного PID-регулятора за візуальним зміщенням.

У результаті проведеного дослідження усі поставлені задачі вирішено, що підтверджується отриманими науковими та практичними результатами**:**

1. Проведено аналіз існуючих моделей, методів і засобів моделювання та керування мультиагентними системами БПЛА. Встановлено, що моделювання таких систем потребує узгодженого врахування процесів на рівні окремого літального апарата, локального керування, сенсорного забезпечення, виконання маршрутних сценаріїв та взаємодії агентів. Програмне SIL-моделювання та напівнатурне HIL-моделювання забезпечують різний ступінь залучення реального обладнання до експериментального контуру, що обґрунтовує використання гібридних конфігурацій із програмними та апаратними компонентами в єдиному середовищі. Встановлено також, що для локальної стабілізації БПЛА за відсутності GPS глобальна координатна прив'язка не є обов'язковою за наявності оцінки відносного зміщення, а зміна режимів руху, характеристик візуального сигналу та обмеження керуючих сигналів створюють передумови для адаптації параметрів PID-регулятора.

2. Розроблено архітектуру гібридного HIL/SIL середовища напівнатурного моделювання мультиагентної системи БПЛА. Сутність архітектури полягає у спільному використанні апаратних і програмних реалізацій польотних контролерів у єдиному модельованому середовищі з двонапрямленою інформаційною взаємодією та розмежуванням потоків даних окремих агентів за допомогою незалежних просторів імен, ідентифікаторів і каналів обміну. Реалізовано експериментальне

середовище на основі Gazebo, ArduPilot, ROS і MAVROS із використанням фізичного польотного контролера Pixhawk для HIL-агента. Експериментально підтверджено одночасне функціонування одного HIL- та трьох SIL-агентів під час зльоту, переміщення до заданих положень і стабілізації. У дослідженій конфігурації зберігався режим реального часу, обмін MAVLink-повідомленнями відбувався без зафіксованих втрат пакетів, а HIL- і SIL-реалізації забезпечували близькі середні значення координат при більшому розкиді координат HIL-агента.

3. Розроблено агентно-орієнтовану модель БПЛА та модель виконання маршрутного сценарію**.** Окремий БПЛА-агент представлено сукупністю локального стану, маршрутного сценарію, контуру керування та правил переходів між станами. Маршрут формалізовано як послідовність параметризованих маршрутних кроків, а його виконання організовано на основі автомата станів, що відокремлює опис маршрутної місії від безпосереднього формування керуючих впливів. Для реалізації маршрутного переміщення використано механізм віртуального дрейфу, який забезпечує перетворення параметрів активного маршрутного кроку у вплив на контур візуальної стабілізації без використання глобальних координат.

4. Розроблено метод просторової стабілізації та маршрутного керування положенням БПЛА в умовах відсутності GPS. Для горизонтального керування використовується сигнал візуального зміщення, сформований зовнішнім функціональним модулем за даними спрямованої вниз камери, а підтримання висоти здійснюється незалежним вертикальним контуром. У маршрутному контурі цільове і поточне зміщення використовуються для визначення похибки виконання маршрутного кроку, за якою маршрутний PID-регулятор формує коригувальні складові для механізму віртуального дрейфу; модифікований сигнал надходить до горизонтального контуру візуальної стабілізації, який формує корекції крену і тангажу. Внутрішній алгоритм визначення візуального зміщення в межах дисертаційного дослідження не розроблявся; власним результатом є його інтеграція з контуром керування та використання сформованих значень для керування БПЛА. Працездатність маршрутного керування перевірено за однаково заданого сценарію у

SIL- та HIL-конфігураціях. Для п'яти SIL-запусків середній час виконання маршруту становив 35,12 с, середня похибка — 91,46 пікс.; для трьох HIL-запусків відповідні значення становили 44,58 с і 106,60 пікс. Отримані результати характеризують досліджені конфігурації й не є підтвердженням синхронного виконання взаємопов'язаних маршрутних сценаріїв декількома агентами, оскільки така взаємодія окремо не досліджувалася. Працездатність просторової стабілізації перевірено в модельованих і натурних умовах. У модельованому експерименті середні абсолютні горизонтальні відхилення становили 6,78 і 6,55 пікс. за двома координатними напрямами, тоді як у натурному експерименті — 15,60 і 37,26 пікс. відповідно. Вертикальний контур в обох експериментальних конфігураціях забезпечував вихід на задану висоту та її подальше утримання. Натурний експеримент характеризувався більшою варіативністю візуального сигналу, однак вплив окремих чинників реального середовища в межах проведеної серії не оцінювався

5. Розроблено фазовий метод адаптивного PID-керування положенням БПЛА за візуальним зміщенням. Сутність методу полягає у перерахунку коефіцієнтів регулятора на кожному циклі позиційного керування відповідно до поточних умов роботи позиційного контуру. Під час адаптації враховуються фаза роботи позиційного контуру, характеристики позиційної похибки, нормований показник якості візуального оцінювання, параметри маршрутного кроку, співвідношення складових похибки за координатними осями та поперечне відхилення від заданого маршруту. Насичення керуючих сигналів визначається незалежно за двома координатними каналами, завдяки чому обмежувальна конфігурація застосовується лише до координатного каналу з активною ознакою насичення. Експериментальний аналіз внутрішньої роботи реалізованого регулятора підтвердив фактичне перемикання між фазами APPROACH, BRAKING, STABILIZATION та SATURATION_LIMITED і зміну коефіцієнтів відповідно до поточної фази роботи позиційного контуру та додаткового осьового коригування.

6. Проведено експериментальне оцінювання фазового методу адаптивного

PID-керування порівняно з регулятором із фіксованими коефіцієнтами. Фінальну валідацію виконано виключно в середовищі Gazebo та ArduPilot SIL за однакових умов моделювання для 100 запусків: 50 із використанням FIXED_PID і 50 із використанням ADAPTIVE_PID. Для FIXED_PID успішно завершено 35 із 50 маршрутів, що відповідає 70 %, тоді як для ADAPTIVE_PID — 46 із 50, або 92 %. Відмінність між частками завершення маршруту була статистично значущою за двобічним точним критерієм Фішера. Для успішно завершених запусків середні значення основних показників позиційної похибки ADAPTIVE_PID були чисельно нижчими на 3,17–5,32 %, однак після поправки Холма статистично значущих відмінностей за середньою похибкою, RMSE, 95-м процентилем похибки, кінцевою похибкою та середньою поперечною похибкою не встановлено. Отже, отримані результати не підтверджують статистично значущого зменшення позиційної похибки під час виконання маршруту в цілому. Середня частка циклів з активною ознакою насичення керуючих каналів для ADAPTIVE_PID була нижчою на 4,60 відсоткового пункту порівняно з FIXED_PID, а відмінність залишилася статистично значущою після поправки Холма $p < 0{,}001$. Для часу проходження успішно завершеного маршруту також встановлено статистично значущу відмінність $p = 0{,}027$ (після поправки Холма), однак різниця середніх значень становила лише 2,27 %, а для ADAPTIVE_PID спостерігалася більша міжзапускова варіативність. Тому цей результат характеризує успішно завершені запуски і не є підтвердженням загального підвищення швидкості виконання маршруту.

7. Практичне значення отриманих результатів полягає у можливості використання розроблених моделей, методів і програмно-апаратного середовища для підготовки та експериментальної перевірки алгоритмів керування БПЛА до переходу до повномасштабних натурних випробувань. Розроблене HIL/SIL середовище дає змогу спільно досліджувати програмні та апаратні реалізації польотних контролерів; модель маршрутного сценарію — задавати і виконувати параметризовані послідовності дій; метод просторової стабілізації — досліджувати локальне керування положенням без глобальної координатної прив'язки; фазовий метод

адаптивного PID-керування — оцінювати зміну параметрів позиційного регулятора залежно від поточних умов роботи позиційного контуру.

Межі застосування отриманих результатів визначаються умовами проведеної експериментальної перевірки. Гібридну архітектуру досліджено в конфігурації одного HIL- та трьох SIL-агентів; синхронне виконання взаємопов'язаних маршрутних сценаріїв декількома агентами окремо не перевірялося. Фінальну кількісну валідацію фазового адаптивного PID-керування проведено лише в SIL-середовищі для одного наперед заданого квадратного маршруту; HIL- і натурні експерименти для цього методу не виконувалися. Позиційна похибка оцінювалася у координатах зображення, тому наведені показники не є безпосередньою метричною оцінкою просторової похибки, а проведені експерименти не становлять формального доведення стійкості замкненого контуру для довільних умов функціонування

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Wooldridge M. An Introduction to MultiAgent Systems. 2nd ed. Chichester : Wiley, 2009. 488 p. ISBN 978-0-470-51946-2.

2. Chung S.-J., Paranjape A. A., Dames P., Shen S., Kumar V. A Survey on Aerial Swarm Robotics. IEEE Transactions on Robotics. 2018. Vol. 34, No. 4. P. 837–855. DOI: 10.1109/TRO.2018.2857475.

3. Coppola M., McGuire K. N., De Wagter C., de Croon G. C. H. E. A Survey on Swarming With Micro Air Vehicles: Fundamental Challenges and Constraints. Frontiers in Robotics and AI. 2020. Vol. 7. Art. 18. DOI: 10.3389/frobt.2020.00018.

4. Bu Y., Yan Y., Yang Y. Advancement Challenges in UAV Swarm Formation Control: A Comprehensive Review. Drones. 2024. Vol. 8, No. 7. Art. 320. DOI: 10.3390/drones8070320.

5. Lizzio F. F., Capello E., Guglieri G. A Review of Consensus-based Multi-agent UAV Implementations. Journal of Intelligent & Robotic Systems. 2022. Vol. 106. Art. 43. DOI: 10.1007/s10846-022-01743-9.

6. Ren W., Beard R. W. Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications. London : Springer, 2008. 319 p. ISBN 978-1-84800-014-8. DOI: 10.1007/978-1-84800-015-5.

7. Chen X., Tang J., Lao S. Review of Unmanned Aerial Vehicle Swarm Communication Architectures and Routing Protocols. Applied Sciences. 2020. Vol. 10, No. 10. Art. 3661. DOI: 10.3390/app10103661.

8. Ponniah J., Dantsker O. D. Strategies for Scaleable Communication and Coordination in Multi-Agent (UAV) Systems. Aerospace. 2022. Vol. 9, No. 9. Art. 488. DOI: 10.3390/aerospace9090488.

9. Wang S., Wang Y., Li D., Zhao Q. Distributed Relative Localization Algorithms for Multi-Robot Networks: A Survey. Sensors. 2023. Vol. 23, No. 5. Art. 2399. DOI: 10.3390/s23052399.

10. Aljalaud F., Kurdi H., Youcef-Toumi K. Bio-Inspired Multi-UAV Path Planning Heuristics: A Review. Mathematics. 2023. Vol. 11, No. 10. Art. 2356. DOI: 10.3390/math11102356.

11. Vásárhelyi G. et al. Optimized flocking of autonomous drones in confined environments. Science Robotics. 2018. Vol. 3, No. 20. Art. eaat3536. DOI: 10.1126/scirobotics.aat3536.

12. Adoni W. Y. H. et al. Intelligent Swarm: Concept, Design and Validation of Self-Organized UAVs Based on Leader–Followers Paradigm for Autonomous Mission Planning. Drones. 2024. Vol. 8, No. 10. Art. 575. DOI: 10.3390/drones8100575.

13. Alqudsi Y., Makaraci M. UAV swarms: research, challenges, and future directions. Journal of Engineering and Applied Science. 2025. Vol. 72. Art. 12. DOI: 10.1186/s44147-025-00582-3.

14. Chen Z., Yan J., Ma B., Shi K., Yu Q., Yuan W. A Survey on Open-Source Simulation Platforms for Multi-Copter UAV Swarms. Robotics. 2023. Vol. 12, No. 2. Art. 53. DOI: 10.3390/robotics12020053.

15. Phadke A., Medrano F. A., Sekharan C. N., Chu T. Designing UAV Swarm Experiments: A Simulator Selection and Experiment Design Process. Sensors. 2023. Vol. 23, No. 17. Art. 7359. DOI: 10.3390/s23177359.

16. Amezquita-Brooks L. A. et al. Towards a standard design model for quad-rotors: A review of current models, their accuracy and a novel simplified model. Progress in Aerospace Sciences. 2017. Vol. 95. P. 1–23. DOI: 10.1016/j.paerosci.2017.09.001.

17. Koenig N., Howard A. Design and Use Paradigms for Gazebo, an Open-Source Multi-Robot Simulator. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2004. Vol. 3. P. 2149–2154. DOI: 10.1109/IROS.2004.1389727.

18. Open Robotics. Gazebo Sensors Documentation. URL: https://gazebosim.org/libs/sensors/ (дата звернення: 01.07.2026).

19. Furrer F., Burri M., Achtelik M., Siegwart R. RotorS – A Modular Gazebo MAV Simulator Framework. In: Robot Operating System (ROS): The Complete Reference (Volume 1). Cham : Springer, 2016. P. 595–625. DOI: 10.1007/978-3-319-26054-9_23.

20. Xiao K., Tan S., Wang G., An X., Wang X., Wang X. XTDrone: A Customizable Multi-Rotor UAVs Simulation Platform. 2020 4th International Conference on Robotics and Automation Sciences (ICRAS). 2020. P. 55–61. DOI: 10.1109/ICRAS49812.2020.9134922.

21. Song Y., Naji S., Kaufmann E., Loquercio A., Scaramuzza D. Flightmare: A Flexible Quadrotor Simulator. Proceedings of the 2020 Conference on Robot Learning. PMLR. 2021. Vol. 155. P. 1147–1157.

22. MathWorks. UAV Toolbox Documentation. URL: https://www.mathworks.com/help/uav/ (дата звернення: 02.07.2026).

23. Kazymyr V., Oleksiienko P., Chornonoh O., Rohovenko A. Modeling of Defensive Drone Swarms with NetLogo. In: Mathematical Modeling and Simulation of Systems. Lecture Notes in Networks and Systems. Vol. 1091. Cham : Springer, 2024. P. 377–387. DOI: 10.1007/978-3-031-67348-1_28.

24. Varga A., Hornig R. An Overview of the OMNeT++ Simulation Environment. Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops (SIMUTools 2008), Marseille, France, 3–7 March 2008. Brussels : ICST, 2008. Art. 60. 10 p. DOI: 10.4108/ICST.SIMUTOOLS2008.3027.

25. Phadke A., Medrano F. A., Sekharan C. N., Chu T. An analysis of trends in UAV swarm implementations in current research: simulation versus hardware. Drone Systems and Applications. 2024. Vol. 12. P. 1–10. DOI: 10.1139/dsa-2023-0099.

26. Aliane N. A Survey of Open-Source UAV Autopilots. Electronics. 2024. Vol. 13, No. 23. Art. 4785. DOI: 10.3390/electronics13234785.

27. ArduPilot Dev Team. SITL Simulator (Software in the Loop). URL: https://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html (дата звернення: 04.07.2026).

28. Meier L., Honegger D., Pollefeys M. PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. 2015 IEEE International

Conference on Robotics and Automation (ICRA). 2015. P. 6235–6240. DOI: 10.1109/ICRA.2015.7140074.

29. PX4 Autopilot. Hardware in the Loop Simulation (HITL). URL: https://docs.px4.io/main/en/simulation/hitl (дата звернення: 05.07.2026).

30. Augello A., Gaglio S., Lo Re G., Peri D. Simulation and Test of UAV Tasks with Resource-Constrained Hardware in the Loop. 2021 IEEE International Conference on Smart Computing (SMARTCOMP). 2021. P. 347–352. DOI: 10.1109/SMARTCOMP52413.2021.00071.

31. Yang J., Thomas A. G., Singh S., Baldi S., Wang X. A Semi-Physical Platform for Guidance and Formations of Fixed-Wing Unmanned Aerial Vehicles. Sensors. 2020. Vol. 20, No. 4. Art. 1136. DOI: 10.3390/s20041136.

32. Zhang Z., Yang W., Shi Z., Zhong Y. Hardware-in-the-loop Simulation Platform for Unmanned Aerial Vehicle Swarm System: Architecture and Application. Proceedings of the 39th Chinese Control Conference (CCC), Shenyang, China, 27–29 July 2020. IEEE, 2020. P. 58–64. DOI: 10.23919/CCC50068.2020.9188541.

33. Meier L., Tanskanen P., Heng L., Lee G. H., Fraundorfer F., Pollefeys M. PIXHAWK: A micro aerial vehicle design for autonomous flight using onboard computer vision. Autonomous Robots. 2012. Vol. 33. P. 21–39. DOI: 10.1007/s10514-012-9281-4.

34. Coutinho W. P., Battarra M., Fliege J. The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review. Computers & Industrial Engineering. 2018. Vol. 120. P. 116–128. DOI: 10.1016/j.cie.2018.04.037.

35. Quan L., Han L., Zhou B., Shen S., Gao F. Survey of UAV motion planning. IET Cyber-Systems and Robotics. 2020. Vol. 2, No. 1. P. 14–21. DOI: 10.1049/iet-csr.2020.0004.

36. Meng W., Zhang X., Zhou L., Guo H., Hu X. Advances in UAV Path Planning: A Comprehensive Review of Methods, Challenges, and Future Directions. Drones. 2025. Vol. 9, No. 5. Art. 376. DOI: 10.3390/drones9050376.

37. Nafees M., Syeda T., Ali A., Farman H., Tayyab M. An in-depth analysis of UAV path planning, including procedures, algorithms, optimization models, and emerging challenges. MethodsX. 2026. Vol. 16. Art. 103826. DOI: 10.1016/j.mex.2026.103826.

38. Dijkstra E. W. A note on two problems in connexion with graphs. Numerische Mathematik. 1959. Vol. 1. P. 269–271. DOI: 10.1007/BF01386390.

39. Hart P. E., Nilsson N. J., Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics. 1968. Vol. 4, No. 2. P. 100–107. DOI: 10.1109/TSSC.1968.300136.

40. LaValle S. M. Rapidly-Exploring Random Trees: A New Tool for Path Planning. Technical Report TR 98-11. Ames, IA : Department of Computer Science, Iowa State University, 1998. 4 p. URL: https://msl.cs.illinois.edu/~lavalle/papers/Lav98c.pdf (дата звернення: 11.10.2026).

41. Kavraki L. E., Švestka P., Latombe J.-C., Overmars M. H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation. 1996. Vol. 12, No. 4. P. 566–580. DOI: 10.1109/70.508439.

42. Khatib O. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. The International Journal of Robotics Research. 1986. Vol. 5, No. 1. P. 90–98. DOI: 10.1177/027836498600500106.

43. Kennedy J., Eberhart R. Particle Swarm Optimization. Proceedings of ICNN'95 – International Conference on Neural Networks. 1995. Vol. 4. P. 1942–1948. DOI: 10.1109/ICNN.1995.488968.

44. Dorigo M., Gambardella L. M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computation. 1997. Vol. 1, No. 1. P. 53–66. DOI: 10.1109/4235.585892.

45. Holland J. H. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. Ann Arbor : University of Michigan Press, 1975. 183 p. ISBN 978-0-472-08460-9.

46. Song J., Zhao K., Liu Y. Survey on Mission Planning of Multiple Unmanned Aerial Vehicles. Aerospace. 2023. Vol. 10, No. 3. Art. 208. DOI: 10.3390/aerospace10030208.

47. ArduPilot Dev Team. AUTO Mode / Mission Commands. URL: https://ardupilot.org/copter/docs/auto-mode.html (дата звернення: 11.08.2026).

48. MAVLink. Mission Protocol. URL: https://mavlink.io/en/services/mission.html (дата звернення: 11.08.2026).

49. Arafat M. Y., Alam M. M., Moh S. Vision-Based Navigation Techniques for Unmanned Aerial Vehicles: Review and Challenges. Drones. 2023. Vol. 7, No. 2. Art. 89. DOI: 10.3390/drones7020089.

50. Jarraya I. et al. GNSS-denied unmanned aerial vehicle navigation: analyzing computational complexity, sensor fusion, and localization methodologies. Satellite Navigation. 2025. Vol. 6. Art. 9. DOI: 10.1186/s43020-025-00162-z.

51. Flores J., Gonzalez-Hernandez I., Salazar S., Lozano R., Reyes C. Improving Optical Flow Sensor Using a Gimbal for Quadrotor Navigation in GPS-Denied Environment. Sensors. 2024. Vol. 24, No. 7. Art. 2183. DOI: 10.3390/s24072183.

52. Miller A., Miller B., Popov A., Stepanyan K. UAV Landing Based on the Optical Flow Videonavigation. Sensors. 2019. Vol. 19, No. 6. Art. 1351. DOI: 10.3390/s19061351.

53. Qin T., Li P., Shen S. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. IEEE Transactions on Robotics. 2018. Vol. 34, No. 4. P. 1004–1020. DOI: 10.1109/TRO.2018.2853729.

54. Mourikis A. I., Roumeliotis S. I. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. 2007 IEEE International Conference on Robotics and Automation. 2007. P. 3565–3572. DOI: 10.1109/ROBOT.2007.364024.

55. Leutenegger S., Lynen S., Bosse M., Siegwart R., Furgale P. Keyframe-based visual–inertial odometry using nonlinear optimization. The International Journal of Robotics Research. 2015. Vol. 34, No. 3. P. 314–334. DOI: 10.1177/0278364914554813.

56. Scaramuzza D., Fraundorfer F. Visual Odometry [Tutorial]. IEEE Robotics & Automation Magazine. 2011. Vol. 18, No. 4. P. 80–92. DOI: 10.1109/MRA.2011.943233.

57. Fraundorfer F., Scaramuzza D. Visual Odometry: Part II: Matching, Robustness, Optimization, and Applications. IEEE Robotics & Automation Magazine. 2012. Vol. 19, No. 2. P. 78–90. DOI: 10.1109/MRA.2012.2182810.

58. Campos C. et al. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. IEEE Transactions on Robotics. 2021. Vol. 37, No. 6. P. 1874–1890. DOI: 10.1109/TRO.2021.3075644.

59. Xu W. et al. FAST-LIO2: Fast Direct LiDAR-Inertial Odometry. IEEE Transactions on Robotics. 2022. Vol. 38, No. 4. P. 2053–2073. DOI: 10.1109/TRO.2022.3141876.

60. Alarifi A. et al. Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances. Sensors. 2016. Vol. 16, No. 5. Art. 707. DOI: 10.3390/s16050707.

61. Firdaus A. R., Hutagalung A., Syahputra A., Analia R. Indoor Localization Using Positional Tracking Feature of Stereo Camera on Quadcopter. Electronics. 2023. Vol. 12, No. 2. Art. 406. DOI: 10.3390/electronics12020406.

62. Chaumette F., Hutchinson S. Visual Servo Control. Part I: Basic Approaches. IEEE Robotics & Automation Magazine. 2006. Vol. 13, No. 4. P. 82–90. DOI: 10.1109/MRA.2006.250573.

63. Chaumette F., Hutchinson S. Visual Servo Control. Part II: Advanced Approaches. IEEE Robotics & Automation Magazine. 2007. Vol. 14, No. 1. P. 109–118. DOI: 10.1109/MRA.2007.339609.

64. Zhong H., Wang Y., Miao Z., Li L., Fan S., Zhang H. A Homography-Based Visual Servo Control Approach for an Underactuated Unmanned Aerial Vehicle in GPS-Denied Environments. IEEE Transactions on Intelligent Vehicles. 2023. Vol. 8, No. 2. P. 1119–1129. DOI: 10.1109/TIV.2022.3163315.

65. Qian Z., Dong Y., Hou Y., Zhang H., Fan S., Zhong H. A geometric approach for homography-based visual servo control of underactuated UAVs. Measurement and Control. 2024. Vol. 57, No. 10. P. 1513–1523. DOI: 10.1177/00202940241238918.

66. Labed M., Mechgoug R., Hachani S. Fuzzy Logic-Enhanced Image-Based Visual Servoing for Quadrotor Systems. Journal of Robotics and Control. 2026. Vol. 7, No. 1. P. 3251–3258. DOI: 10.18196/jrc.v7i1.28153.

67. Kim J., Gadsden S. A., Wilkerson S. A. A Comprehensive Survey of Control Strategies for Autonomous Quadrotors. Canadian Journal of Electrical and Computer Engineering. 2020. Vol. 43, No. 1. P. 3–16. DOI: 10.1109/CJECE.2019.2920938.

68. Lopez-Sanchez I., Moreno-Valenzuela J. PID control of quadrotor UAVs: A survey. Annual Reviews in Control. 2023. Vol. 56. Art. 100900. DOI: 10.1016/j.arcontrol.2023.100900.

69. Melo A. G., Andrade F. A. A., Guedes I. P., Carvalho G. F., Zachi A. R. L., Pinto M. F. Fuzzy Gain-Scheduling PID for UAV Position and Altitude Controllers. Sensors. 2022. Vol. 22, No. 6. Art. 2173. DOI: 10.3390/s22062173.

70. Boubakir A., Souanef T., Labiod S., Whidborne J. F. A Robust Adaptive PID-like Controller for Quadrotor Unmanned Aerial Vehicle Systems. Aerospace. 2024. Vol. 11, No. 12. Art. 980. DOI: 10.3390/aerospace11120980.

71. Spencer J., Lee J., Paredes J. A., Goel A., Bernstein D. S. An Adaptive PID Autotuner for Multicopters with Experimental Results. 2022 IEEE International Conference on Robotics and Automation (ICRA). 2022. P. 7846–7853. DOI: 10.1109/ICRA46639.2022.9812065.

72. Hazelaar S. T., Wang C., de Wagter C., Muijres F. T., de Croon G. C. H. E., Yedutenko M. Bioinspired adaptive visual servoing control for quadrotors. Bioinspiration & Biomimetics. 2025. Vol. 20, No. 3. Art. 036014. DOI: 10.1088/1748-3190/adcdde.

73. Zhou W., Zhang Y., Ni L. Research on Autonomous UAV Shipboard Landing Control for Dynamic Ship Platforms. Machines. 2026. Vol. 14, No. 6. Art. 612. DOI: 10.3390/machines14060612.

74. Jia K., Lin S., Du Y., Zou C., Lu M. Research on Route Tracking Controller of Quadrotor UAV Based on Fuzzy Logic and RBF Neural Network. IEEE Access. 2023. Vol. 11. P. 111433–111447. DOI: 10.1109/ACCESS.2023.3322944.

75. Shamma J. S., Athans M. Gain Scheduling: Potential Hazards and Possible Remedies. IEEE Control Systems Magazine. 1992. Vol. 12, No. 3. P. 101–107. DOI: 10.1109/37.165527.

76. Åström K. J., Hägglund T. Advanced PID Control. Research Triangle Park, NC : ISA, 2006. 460 p. ISBN 978-1-55617-942-6.

77. Baidya S., Shaikh Z., Levorato M. FlyNetSim: An Open Source Synchronized UAV Network Simulator based on ns-3 and Ardupilot. Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM 2018). 2018. P. 37–45. DOI: 10.1145/3242102.3242118.

78. Quigley M., Gerkey B., Conley K., Faust J., Foote T., Leibs J., Berger E., Wheeler R., Ng A. Y. ROS: an open-source Robot Operating System. Open-Source Software Workshop at the 2009 IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan, 12–17 May 2009. 6 p. URL: https://ai.stanford.edu/~mquigley/papers/icra2009-ros.pdf (дата звернення: 20.08.2026).

79. MAVROS Project. MAVROS Documentation: MAVLink-to-ROS Bridge. URL: https://github.com/mavlink/mavros/blob/ros2/docs/index.md (дата звернення: 21.08.2026).

80. Koubaa A., Allouch A., Alajlan M., Javed Y., Belghith A., Khalgui M. Micro Air Vehicle Link (MAVLink) in a Nutshell: A Survey. IEEE Access. 2019. Vol. 7. P. 87658–87680. DOI: 10.1109/ACCESS.2019.2924410.

81. Suslo M., Kazymyr V., Oleksiienko P., Kagitin D. Semi-Physical Modeling of Drone Swarm Control Processes. 2025 15th International Conference on Advanced Computer Information Technologies (ACIT). IEEE, 2025. P. 102–108. DOI: 10.1109/ACIT65614.2025.11185750.

82. Сусло М. Ю., Олексієнко П. Д., Сігута А. В., Казимир В. В. Напів натурне моделювання процесів керування роєм дронів. Новітні технології сучасного суспільства (НТСС-2024) : V Міжнародна науково-практична конференція (м. Чернігів, 12 грудня 2024 р.) : тези доповідей. Чернігів : НУ «Чернігівська політехніка», 2025. С. 183–184.

83. Bray T. The JavaScript Object Notation (JSON) Data Interchange Format. RFC 8259. Internet Engineering Task Force, 2017. DOI: 10.17487/RFC8259.

84. Zhang Z. A Flexible New Technique for Camera Calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2000. Vol. 22, No. 11. P. 1330–1334. DOI: 10.1109/34.888718.

85. Harel D. Statecharts: A Visual Formalism for Complex Systems. Science of Computer Programming. 1987. Vol. 8, No. 3. P. 231–274. DOI: 10.1016/0167-6423(87)90035-9.

86. Rublee E., Rabaud V., Konolige K., Bradski G. ORB: An efficient alternative to SIFT or SURF. 2011 IEEE International Conference on Computer Vision. 2011. P. 2564–2571. DOI: 10.1109/ICCV.2011.6126544.

87. Fischler M. A., Bolles R. C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM. 1981. Vol. 24, No. 6. P. 381–395. DOI: 10.1145/358669.358692.

88. Hartley R., Zisserman A. Multiple View Geometry in Computer Vision. 2nd ed. Cambridge : Cambridge University Press, 2004. 655 p. ISBN 978-0-521-54051-3. DOI: 10.1017/CBO9780511811685.

89. Luo Y., Wang X., Liao Y., Fu Q., Shu C., Wu Y., He Y. A Review of Homography Estimation: Advances and Challenges. Electronics. 2023. Vol. 12, No. 24. Art. 4977. DOI: 10.3390/electronics12244977.

90. Sharma S. K., Jain K., Shukla A. K. A Comparative Analysis of Feature Detectors and Descriptors for Image Stitching. Applied Sciences. 2023. Vol. 13, No. 10. Art. 6015. DOI: 10.3390/app13106015.

91. Kahitin D., Kazymyr V., Suslo M., Yatchenko Y., Oleksiienko P. Drone Positioning in a Specified Location Using Visual Data Under GPS-denied Conditions. 2025 IEEE 13th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). Vol. 1. Gliwice, 2025. P. 318–323. DOI: 10.1109/IDAACS68557.2025.11322385.

92. Roberts S. W. Control Chart Tests Based on Geometric Moving Averages. Technometrics. 1959. Vol. 1, No. 3. P. 239–250. DOI: 10.1080/00401706.1959.10489860.

93. MAVLink. Common Message Set: RC_CHANNELS_OVERRIDE (Message 70). URL: https://mavlink.io/en/messages/common.html#RC_CHANNELS_OVERRIDE (дата звернення: 27.08.2026).

94. Сусло М. Ю., Казимир В. В. Напівнатурне моделювання маршрутних сценаріїв для мультиагентної системи БПЛА. Наука і техніка сьогодні. 2026. № 5(59). С. 6049–6068. DOI: 10.52058/2786-6025-2026-5(59)-6049-6068.

95. Сусло М. Ю., Казимир В. В. Метод стабілізації квадрокоптера в заданій позиції з просторовим PID-регулюванням за відеоданими без GPS. Технічні науки та технології. 2026. № 2(44). С. 306–316. DOI: 10.25140/2411-5363-2026-2(44)-306-316.

96. Fisher R. A. On the Interpretation of χ^2 from Contingency Tables, and the Calculation of P. Journal of the Royal Statistical Society. 1922. Vol. 85, No. 1. P. 87–94. DOI: 10.1111/j.2397-2335.1922.tb00768.x.

97. Mann H. B., Whitney D. R. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. The Annals of Mathematical Statistics. 1947. Vol. 18, No. 1. P. 50–60. DOI: 10.1214/aoms/1177730491.

98. Holm S. A Simple Sequentially Rejective Multiple Test Procedure. Scandinavian Journal of Statistics. 1979. Vol. 6, No. 2. P. 65–70. URL: https://www.jstor.org/stable/4615733 (дата звернення: 01.09.2026).

99. Hollander M., Wolfe D. A., Chicken E. Nonparametric Statistical Methods. 3rd ed. Hoboken, NJ : Wiley, 2014. 848 p. ISBN 978-0-470-38737-5. DOI: 10.1002/9781119196037.

100. Suslo M., Kazymyr V. Adaptive PID Control of UAV Position Using Visual Displacement. Computer Systems and Information Technologies. 2026. No. 3. P. 58–68. DOI: 10.31891/csit-2026-3-6.

ДОДАТКИ

Додаток А. *Список публікацій здобувача за темою дисертації*

1. Сусло М. Ю., Казимир В. В. Метод стабілізації квадрокоптера в заданій позиції з просторовим PID-регулюванням за відеоданими без GPS. Технічні науки та технології. 2026. № 2(44). С. 306–316. DOI: 10.25140/2411-5363-2026-2(44)-306-316 (розроблення та програмна реалізація методу просторової стабілізації БПЛА за візуальним зміщенням, інтеграція сигналів візуального зміщення в контур PID-керування, проведення модельованих і натурних експериментів, оброблення та аналіз отриманих результатів)
2. Сусло М. Ю., Казимир В. В. Напівнатурне моделювання маршрутних сценаріїв для мультиагентної системи БПЛА. Наука і техніка сьогодні. 2026. № 5(59). С. 6049–6068. DOI: 10.52058/2786-6025-2026-5(59)-6049-6068 (Особистий внесок здобувача: розроблення агентно-орієнтованої моделі виконання маршрутних сценаріїв БПЛА, формалізація структури маршрутного сценарію та механізму його виконання, програмна реалізація маршрутного керування, проведення експериментів у SIL- та HIL-конфігураціях, аналіз отриманих результатів)
3. Suslo M., Kazymyr V. Adaptive PID Control of UAV Position Using Visual Displacement. Computer Systems and Information Technologies. 2026. No. 3. P. 58–68. DOI: 10.31891/csit-2026-3-6 (Особистий внесок здобувача: розроблення фазового методу адаптивного налаштування коефіцієнтів PID-регулятора, програмна реалізація адаптивного регулятора, розроблення методики експериментального дослідження, проведення серії SIL-експериментів, статистичне опрацювання та порівняльний аналіз результатів.)
4. Сусло М. Ю., Олексієнко П. Д., Сігута А. В., Казимир В. В. Напів натурне моделювання процесів керування роєм дронів. Новітні технології сучасного суспільства (НТСС-2024) : V Міжнародна науково-практична конференція, м. Чернігів, 12 грудня 2024 р. : тези доповідей. Чернігів : НУ «Чернігівська

політехніка», 2025. С. 183–184. (Особистий внесок здобувача: участь у розробленні та дослідженні засобів напівнатурного моделювання мультиагентної системи БПЛА, реалізація та експериментальна перевірка компонентів модельованого середовища)

5. Suslo M., Kazymyr V., Oleksiienko P., Kagitin D. Semi-Physical Modeling of Drone Swarm Control Processes. 2025 15th International Conference on Advanced Computer Information Technologies (ACIT). IEEE, 2025. P. 102–108. DOI: 10.1109/ACIT65614.2025.11185750 (Особистий внесок здобувача: участь у розробленні та програмній реалізації гібридного HIL/SIL середовища моделювання, інтеграція програмних і апаратних компонентів системи, налаштування експериментального середовища та участь в аналізі результатів моделювання.)
6. Kahitin D., Kazymyr V., Suslo M., Yatchenko Y., Oleksiienko P. Drone Positioning in a Specified Location Using Visual Data Under GPS-denied Conditions. 2025 IEEE 13th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). Vol. 1. Gliwice, Poland, 2025. P. 318–323. DOI: 10.1109/IDAACS68557.2025.11322385. (Особистий внесок здобувача: налаштування камери та експериментального середовища, інтеграція модуля оцінювання візуального зміщення з програмним контуром керування БПЛА, оброблення отриманих компонент зміщення та участь в експериментальній перевірці системи.)

Додаток Б. *Довідки про впровадження*

Відкрита інформація

МІНІСТЕРСТВО ОБОРОНИ
УКРАЇНИ
ДЕРЖАВНИЙ
НАУКОВО-ДОСЛІДНИЙ
ІНСТИТУТ ВИПРОБУВАНЬ
І СЕРТИФІКАЦІЇ ОЗБРОЄННЯ
ТА ВІЙСЬКОВОЇ ТЕХНІКИ
«20» травня 2026 р.
№ 625/ 3162
18000, м. Черкаси

ДОВІДКА
про використання результатів
дисертаційної роботи Сусла Миколи Юрійовича на тему
«Моделі та методи напівнатурного моделювання мультиагентних систем БПЛА»,
представленої на здобуття ступеня доктора філософії
зі спеціальності F3 (122) - Комп'ютерні науки.

Результати дисертаційної роботи Сусла Миколи Юрійовича були використані у Державному науково-дослідному інституті випробувань і сертифікації озброєння та військової техніки під час проведення натурних досліджень та відпрацювання програм і методик випробувань безпілотних авіаційних комплексів (БпАК). Зокрема, практичне застосування знайшли:

1) програмно-апаратна архітектура напівнатурного моделювання (Hardware-in-the-Loop) поведінки рою БпЛА, що використовує інтегроване віртуальне імітаційне середовище, програмні емулятори та апаратні засоби для достовірного відтворення процесів керування безпілотними апаратами у складі групи при виконанні польотних завдань;

2) метод навігації БпЛА у складі рою під час їх розгортання в позиції чергування, що базується на програмно визначених маршрутах переміщення без використання супутникових систем навігації (GPS/GNSS), що є критично важливим в умовах радіоелектронної протидії;

3) метод адаптивного керування для стабілізації положення БпЛА у просторі, який використовує дані від спрямованої вниз відеокамери для динамічного корегування параметрів PID-регулятора відповідно до виявленого зміщення.

Використання наукових положень дисертації дозволило удосконалити чинні програми та методики випробувань БпАК у частині перевірки їхньої стійкості до інтенсивного впливу засобів РЕБ. Застосування запропонованої архітектури напівнатурного моделювання суттєво підвищило достовірність результатів стендових експериментів порівняно із суто програмною симуляцією, що забезпечило більш якісну підготовку до етапів натурних льотних випробувань.

Головний науковий співробітник науково-дослідного управління випробувань безпілотних та автоматизованих систем Державного науково-дослідного інституту випробувань і сертифікації озброєння та військової техніки
доктор технічних наук, професор — Володимир РУДНИЦЬКИЙ

Підпис Рудницького В.М. завіряю:
Учений секретар Державного науково-дослідного інституту випробувань і сертифікації озброєння та військової техніки, кандидат технічних наук, старший науковий співробітник — Олександр БУРСАЛА

ВІДКРИТА ІНФОРМАЦІЯ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЧЕРНІГІВСЬКА ПОЛІТЕХНІКА»

вул. Шевченка, 95, Чернігів, 14035, Україна

тел. +38(0462) 665-103
E-mail: cstu@stu.cn.ua
www.stu.cn.ua
Код ЄДРПОУ 05460798

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

CHERNIHIV POLYTECHNIC NATIONAL UNIVERSITY

95, Shevchenko str., Chernihiv, 14035, Ukraine

10.10.2026 № 202/07-2177/1

На № _____ від _____

ДОВІДКА ПРО ВПРОВАДЖЕННЯ

результатів дисертаційної роботи Сусла Миколи Юрійовича
на тему «Моделі та методи напівнатурного моделювання
мультиагентних систем БПЛА»,
представленої на здобуття вченого ступеня доктора філософії
зі спеціальності 122 – Комп'ютерні науки

Основні положення та результати дисертаційного дослідження Сусла Миколи Юрійовича на тему «Моделі та методи напівнатурного моделювання мультиагентних систем БПЛА» використовуються в навчальному процесі Національного університету «Чернігівська політехніка» на кафедрі інформаційних та комп'ютерних систем при проведенні лекцій та лабораторних робіт:

- з дисциплін «Статистичні основи комп'ютерної інженерії», «Моделювання систем» та «Методи досліджень» в процесі навчання бакалаврів та магістрів спеціальності F7 (123) – Комп'ютерна інженерія;
- з дисциплін «Статистичні методи обробки розподіленої інформації» та «Методи та технології математичного та комп'ютерного моделювання складних систем» в процесі навчання аспірантів спеціальності F3 (122) – Комп'ютерні науки.

Проректор з науково-методичної роботи
д.т.н., професор Володимир КАЛЬЧЕНКО

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ «Національний університет "Чернігівська політехніка"» 05460798

Войцеховська М.М. (0462) 665 115

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЧЕРНІГІВСЬКА ПОЛІТЕХНІКА»

вул. Шевченка, 95, Чернігів, 14035, Україна

тел. +38(0462) 665-103
E-mail: cstu@stu.cn.ua
www.stu.cn.ua
Код ЄДРПОУ 05460798

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

CHERNIHIV POLYTECHNIC NATIONAL UNIVERSITY

95, Shevchenko str., Chernihiv, 14035, Ukraine

19.10.2026 № 202/07-2177

На № _____ від _____

ДОВІДКА ПРО ВПРОВАДЖЕННЯ

результатів дисертаційної роботи Сусла Миколи Юрійовича
на тему «Моделі та методи напівнатурного моделювання
мультиагентних систем БПЛА»,
представленої на здобуття вченого ступеня доктора філософії
зі спеціальності 122 – Комп'ютерні науки

Результати дисертаційного дослідження Сусла М.Ю., а саме:

– моделі та програмно-апаратна архітектура напівнатурного моделювання мультиагентних систем БПЛА, що використовує інтегроване віртуальне імітаційне середовище, апаратні засоби (Hardware-in-the-Loop) та програмні емулятори (Software-in-the-Loop) для відтворення процесів керуваннями дронами-агентами у складі рою при виконанні польотних завдань;

– моделі та метод навігації дронів рою при їх розгортанні в позиції чергування, що базується на програмно визначених за допомогою середовища моделювання маршрутах переміщення без використання GPS;

– метод адаптивного керування дроном під час проходження маршруту та стабілізації його положення у просторі, який використовує дані від відео-камери, спрямованої вниз, для динамічного корегування параметрів PID-регулятора відповідно до виявленого зміщення;

– програмна реалізація розроблених методів та алгоритмів в діючих макетах дронів та їх верифікація в ході модельних та натурних експериментів –

були впроваджені в науково-дослідній роботі (прикладне дослідження) "Мультиагентна система захисту об'єктів критичної інфраструктури на основі рою мультикоптерних дронів" (номер державної реєстрації 0123U101819), що виконувалась в Національному університеті «Чернігівська політехніка» в період з 2023 по 2025 роки, що дозволило підвищити успішність виконання маршрутних сценаріїв БПЛА у складі рою при його розгортанні для виконання місії в умовах відсутності GPS на 32 відсотки.

Проректор з наукової роботи,
к.т.н., доцент Анатолій ПРИСТУПА

Войцеховська М.М. (0462) 665 115

Додаток В. *Сертифікати участі в конференціях*





Додаток Г. *Параметри та конфігурація гібридного HIL/SIL середовища моделювання.*

Таблиця Г.1

Апаратні та програмні компоненти експериментального середовища

Компонент	**Використана конфігурація**	**Призначення**
Персональний комп'ютер	Intel Core i7	виконання Gazebo, ROS, MAVROS і програмних екземплярів ArduPilot
Операційна система	Ubuntu 20.04 LTS	програмне середовище експериментального стенда
Середовище моделювання	Gazebo 11	моделювання БПЛА, сцени та сенсорних даних
Робототехнічне ПЗ	ROS Noetic	обмін даними між програмними компонентами
Інтерфейс автопілота	MAVROS	взаємодія ROS з MAVLink-сумісними автопілотами
Автопілот	ArduPilot / ArduCopter 4.3	низькорівневе керування БПЛА
Польотний контролер	Pixhawk 6C	апаратна HIL-реалізація
Одноплатний комп'ютер	Jetson Nano	виконання програмного модуля високорівневого керування
Модель БПЛА	Iris	базова модель квадрокоптера в Gazebo

Таблиця Г.2

Конфігурація агентів гібридного HIL/SIL середовища

Агент	**Режим**	**Польотний контролер**	**Простір імен ROS/MAVROS**	**Канал з'єднан**
UAV1	HIL	Pixhawk 6C, ArduCopter 4.3	/uav1	USB / послідовний інтерфейс
UAV2	SIL	ArduPilot SIL	/uav2	UDP
UAV3	SIL	ArduPilot SIL	/uav3	UDP
UAV4	SIL	ArduPilot SIL	/uav4	UDP

Таблиця Г.3

Основні параметри моделі БПЛА та модельованого середовища

Параметр	**Значення**
Базова модель	Iris
Формат опису моделі	SDF

Основна ланка	base_link
Маса основної ланки	1,5 кг
Колізійна геометрія корпуса	0,47 × 0,47 × 0,23 м
Кількість роторів	4
Фізичний рушій Gazebo	ODE
Прискорення вільного падіння	9,8 м/с²
Віртуальна камера	nadir_camera
Роздільна здатність камери	640 × 480 пікселів
Частота формування зображення	30 Гц

Лістинг Г.1

Фрагмент SDF-опису складу модельованої сцени

```
<include>
  <uri>model://ground_plane_unique</uri>
  <name>ground_plane</name>
  <pose>0 0 0 0 0 0</pose>
</include>

<include>
  <uri>model://iris_with_ardupilot</uri>
  <name>iris1</name>
  <pose>0 0 0 0 0 0</pose>
</include>

<include>
  <uri>model://down_cam</uri>
  <name>down_cam_1</name>
  <pose>0 0 0 0 0 0</pose>

  <plugin name="down_cam_follower_1"
          filename="libFollowerDownCamPlugin.so">
    <target_model>iris1</target_model>
    <target_link>iris::base_link</target_link>
    <offset>0 0 -0.05</offset>
  </plugin>
</include>
```

Лістинг Г.2

Фрагмент SDF-опису параметрів спрямованої вниз камери

```
<sensor name="nadir_camera" type="camera">
  <pose>0 0 0 0 1.57079632679 0</pose>

  <always_on>true</always_on>
  <update_rate>30</update_rate>
  <visualize>true</visualize>

  <camera>
    <horizontal_fov>1.1</horizontal_fov>
```

```
    <image>
      <width>640</width>
      <height>480</height>
      <format>R8G8B8</format>
    </image>

    <clip>
      <near>0.05</near>
      <far>100</far>
    </clip>
  </camera>
</sensor>
```

Лістинг Г.3

Для відтворення логічної послідовності запуску середовища нижче наведено запуск його основних програмних компонентів без урахування засобів контейнеризації. Контейнеризація використовувалася лише як засіб організації програмного середовища та не впливала на алгоритм взаємодії Gazebo, ArduPilot і розробленого модуля керування.

Послідовність запуску програмної SIL-конфігурації експериментального середовища

```
# Налаштування шляхів Gazebo
cd ~/Documents/DroneProject

export ARDUPILOT_GAZEBO_ROOT=/home/nano/ardupilot_gazebo
export
GAZEBO_MODEL_PATH="$PWD/gazebo/models:$ARDUPILOT_GAZEBO_ROOT/models:${GAZEBO_MODEL_PATH:-}"
export GAZEBO_PLUGIN_PATH="$ARDUPILOT_GAZEBO_ROOT/build:${GAZEBO_PLUGIN_PATH:-}"

# Запуск модельованого середовища
gzserver --verbose \
    gazebo/worlds/iris_arducopter_cmac_i1.world

# Запуск ArduPilot SITL
cd ~/ardupilot

sim_vehicle.py \
    -v ArduCopter \
    -f gazebo-iris \
    -I1 \
    --console \
    --out=tcpin:0.0.0.0:5764

# Запуск розробленого модуля маршрутного керування
cd ~/Documents/DroneProject

POSITION_CONTROL_MODE=ADAPTIVE_PID \
python3.10 -m src.mission.pixel_route_runner \
```

```
missions/pixel_route_demo.json
```

Наведені параметри та фрагменти конфігурації визначають основні компоненти гібридного HIL/SIL середовища, використаного для напівнатурного моделювання систем БПЛА. Спільне модельоване середовище Gazebo забезпечує уніфіковані умови взаємодії для програмно модельованих агентів ArduPilot SIL та апаратного польотного контролера, залученого в HIL-конфігурації. Поділ конфігурації за окремими агентами, каналами обміну та моделями БПЛА забезпечує можливість відтворення структури експериментального середовища та його подальшого розширення для дослідження мультиагентних сценаріїв.

Додаток Д. *Приклади опису та програмної реалізації маршрутних сценаріїв БПЛА*

Лістинг Д.1.

Скорочений фрагмент опису маршрутного сценарію у форматі JSON

```
{
  "name": "single_drone_square_1000px_step100",
  "agent_id": "uav_sitl_1",
  "controller_type": "SITL",
  "position_control_mode": "ADAPTIVE_PID",
  "takeoff_altitude": 2.0,
  "land_after_finish": true,
  "start_delay_seconds": 2.0,
  "default_tolerance_pixels": 25,
  "default_cross_track_tolerance_pixels": 250,
  "default_hold_seconds": 0.8,
  "route": [
    { "name": "forward_001", "direction": "forward",
      "distance_pixels": 100, "completion_axis": "y" },
    { "name": "forward_002", "direction": "forward",
      "distance_pixels": 100, "completion_axis": "y" },
    { "name": "left_001", "direction": "left",
      "distance_pixels": 100, "completion_axis": "x" },
    { "name": "back_001", "direction": "back",
      "distance_pixels": 100, "completion_axis": "y" },
    { "name": "right_010", "direction": "right",
      "distance_pixels": 100, "completion_axis": "x" }
  ]
}
```

У лістингу Д.1 наведено скорочений фрагмент опису маршрутного сценарію, що демонструє структуру параметрів окремих маршрутних кроків. Повний сценарій, використаний у фінальній експериментальній серії, задавав квадратний маршрут розміром 1000 × 1000 пікселів і складався з 40 послідовних відносних переміщень по 100 пікселів: по 10 кроків у напрямках forward, left, back і right. Завершення кроків у напрямках forward і back контролювалося за віссю y, а для left і right – за віссю x.

Лістинг Д.2.

Перетворення параметрів маршрутного кроку у цільове зміщення

```
def _parse_step(
    index: int,
    raw_step: Any,
    direction_map: Mapping[str, Mapping[str, float]],
    default_tolerance_pixels: float,
    default_cross_track_tolerance_pixels: float,
    default_hold_seconds: float,
) -> PixelRouteStep:
    tolerance = float(
        raw_step.get(
```

```
            "tolerance_pixels",
            raw_step.get("tolerance", default_tolerance_pixels),
        )
    )

    cross_track_tolerance = float(
        raw_step.get(
            "cross_track_tolerance_pixels",
            raw_step.get(
                "cross_track_tolerance",
                default_cross_track_tolerance_pixels,
            ),
        )
    )

    hold_seconds = float(
        raw_step.get(
            "hold_seconds",
            raw_step.get("hold_time", default_hold_seconds),
        )
    )

    name = str(
        raw_step.get(
            "name",
            raw_step.get("direction", f"step_{index + 1}"),
        )
    )

    direction = str(raw_step.get("direction", "")).lower()
    distance = float(
        raw_step.get(
            "distance_pixels",
            raw_step.get("pixels", raw_step.get("distance", 0.0)),
        )
    )

    vector = direction_map[direction]
    dx_pixels = float(vector["dx"]) * distance
    dy_pixels = float(vector["dy"]) * distance

    return PixelRouteStep(
        name=name,
        dx_pixels=dx_pixels,
        dy_pixels=dy_pixels,
        tolerance_pixels=tolerance,
        cross_track_tolerance_pixels=cross_track_tolerance,
        hold_seconds=hold_seconds,
        completion_axis=_parse_completion_axis(
            raw_step.get("completion_axis"),
            dx_pixels,
            dy_pixels,
        ),
    )
```

Під час завантаження сценарію напрям переміщення перетворюється у двовимірний вектор, який масштабується відповідно до distance_pixels. У результаті

формується об’єкт PixelRouteStep, що містить цільове зміщення за координатами зображення, параметри допустимої похибки, поперечного відхилення, утримання та вісь контролю завершення кроку.

Лістинг Д.3.

Перевірка завершення маршрутного кроку

```
def __check_if_target_reached(self) -> bool:
    if self.__target is None:
        return False

    if self.__target.completion_axis == "x":
        primary_error = abs(self.__relative_error_x)
        cross_track_error = abs(self.__error_y)
        result = self.__axis_progress_reached(
            self.__target.dx_pixels,
            self.__target.dx_pixels,
            self.__position_x,
            self.__target.tolerance_pixels,
        ) and (
            cross_track_error
            <= self.__target.cross_track_tolerance_pixels
        )

    elif self.__target.completion_axis == "y":
        primary_error = abs(self.__relative_error_y)
        cross_track_error = abs(self.__error_x)

        result = self.__axis_progress_reached(
            self.__target.dy_pixels,
            self.__target.dy_pixels,
            self.__position_y,
            self.__target.tolerance_pixels,
        ) and (
            cross_track_error
            <= self.__target.cross_track_tolerance_pixels
        )

    else:
        primary_error = math.sqrt(
            self.__relative_error_x ** 2
            + self.__relative_error_y ** 2
        )
        result = (
            primary_error
            <= self.__target.tolerance_pixels
        )

    return result
```

Лістинг Д.4.

Реалізація механізму віртуального дрейфу

```
@property
def __relative_error_x(self) -> float:
```

```
    return (
        self.__target.dx_pixels - self.__position_x
        if self.__target
        else 0.0
    )


@property
def __relative_error_y(self) -> float:
    return (
        self.__target.dy_pixels - self.__position_y
        if self.__target
        else 0.0
    )
def __navigation_control_errors(self) -> tuple[float, float]:
    if self.__target is None:
        return (0.0, 0.0)

    if self.__target.completion_axis == "x":
        primary_error = self.__relative_error_x
        cross_track_error = self.__error_y

        if (
            abs(cross_track_error)
            > self.__target.cross_track_tolerance_pixels
        ):
            primary_error = (
                0.0
                if self.__stop_primary_when_off_track
                else primary_error * self.__primary_axis_gain
            )

        return (
            primary_error,
            cross_track_error * self.__cross_track_gain,
        )

    if self.__target.completion_axis == "y":
        primary_error = self.__relative_error_y
        cross_track_error = self.__error_x

        if (
            abs(cross_track_error)
            > self.__target.cross_track_tolerance_pixels
        ):
            primary_error = (
                0.0
                if self.__stop_primary_when_off_track
                else primary_error * self.__primary_axis_gain
            )

        return (
            cross_track_error * self.__cross_track_gain,
            primary_error,
        )

    return (self.__error_x, self.__error_y)
control_error_x, control_error_y = \
    self.__navigation_control_errors()
```

```
    vel_x, vel_y = \
        self.__velocity_controller.calculate_command(
            control_error_x,
            control_error_y,
            dt,
        )

    interface_result = build_position_error_interface(
        interface=self.__position_error_interface,
        control_error_x=control_error_x,
        control_error_y=control_error_y,
        velocity_x=vel_x,
        velocity_y=vel_y,
        max_velocity_pixels=
            self.__velocity_controller.max_velocity,
        recovery_max_pixels=self.__recovery_max_pixels,
    )

    command_dx, command_dy = \
        command_input_for_interface(interface_result)

    command = ShiftCommand(
        dx=command_dx,
        dy=command_dy,
        angle_deg=0.0,
        matches_percent=shift_cmd.matches_percent,
        control_input_source=
            CONTROL_INPUT_SOURCE_NAVIGATION_MODIFIED,
        navigation_error_x=control_error_x,
        navigation_error_y=control_error_y,
        local_visual_error_x=shift_cmd.dx,
        local_visual_error_y=shift_cmd.dy,
    )
```

Реалізований механізм не передає маршрутну похибку безпосередньо польотному контролеру. За похибкою активного маршрутного кроку формується модифікований сигнал ShiftCommand із джерелом NAVIGATION_MODIFIED. Сигнал подається до наявного контуру позиційної стабілізації як штучно сформоване зміщення, компенсація якого приводить до переміщення БПЛА в напрямку поточного маршрутного завдання.

Додаток Е. Фрагменти програмної реалізації фазового адаптивного PID-керування

Лістинг Е.1.

Визначення поточної фази позиційного контуру

```
def _select_phase(
    self,
    *,
    confidence: float,
```

```
    error_magnitude: float,
    error_derivative_x: float,
    error_derivative_y: float,
    roll_saturated: bool,
    pitch_saturated: bool,
    is_near_target: bool,
) -> str:
    if roll_saturated or pitch_saturated:
        return "SATURATION_LIMITED"
    if confidence < self.low_confidence_threshold:
        return "LOW_CONFIDENCE"
    if (
        is_near_target
        or error_magnitude
        <= self.stabilization_error_threshold
    ):
        return "STABILIZATION"
    braking_indicator = (
        abs(error_derivative_x)
        >= self.braking_error_threshold
        or abs(error_derivative_y)
        >= self.braking_error_threshold
        or (error_derivative_x * error_derivative_y) < 0
    )
    if braking_indicator:
        return "BRAKING"

    return "APPROACH"
```

Поточна фаза визначається повторно на кожному циклі керування відповідно до фіксованого пріоритету умов. За відсутності ознак насичення, низької якості візуального оцінювання, стабілізації та гальмування використовується базова фаза APPROACH.

Лістинг Е.2.

Формування фазових і осьових коефіцієнтів

```
def _phase_gains(
    self,
    phase: str,
    confidence: float,
) -> tuple[float, float, float]:

    if phase == "LOW_CONFIDENCE":
        return (
            self.base_kp * (
                self.low_confidence_kp_base
                + self.low_confidence_kp_confidence_scale
                * confidence
            ),
            0.0,
            self.base_kd * (
                self.low_confidence_kd_base
                + self.low_confidence_kd_confidence_loss_scale
```

```
                * (1.0 - confidence)
            ),
        )

    if phase == "SATURATION_LIMITED":
        return (
            self.base_kp * self.saturation_kp_scale,
            0.0,
            self.base_kd * self.saturation_kd_scale,
        )

    if phase == "BRAKING":
        return (
            self.base_kp * self.braking_kp_scale,
            self.base_ki * self.braking_ki_scale,
            self.base_kd * self.braking_kd_scale,
        )

    if phase == "STABILIZATION":
        return (
            self.base_kp * self.stabilization_kp_scale,
            min(
                self.ki_max,
                self.base_ki
                * self.stabilization_ki_scale,
            ),
            self.base_kd
            * self.stabilization_kd_scale,
        )

    return (
        self.base_kp * self.approach_kp_scale,
        self.base_ki * self.approach_ki_scale,
        self.base_kd * self.approach_kd_scale,
    )
def _axis_scales(
    self,
    completion_axis: str,
    route_step_index: int,
) -> tuple[float, float]:

    axis = str(completion_axis or "xy").lower()

    route_factor = (
        1.0
        + min(
            max(route_step_index, 0),
            self.axis_route_step_cap,
        ) * self.axis_route_step_gain
    )

    if axis == "x":
        return (
            self.primary_axis_gain * route_factor,
            self.cross_track_gain,
        )

    if axis == "y":
        return (
```

```
            self.cross_track_gain,
            self.primary_axis_gain * route_factor,
        )

    return route_factor, route_factor
```

Лістинг Е.3.

Незалежне врахування насичення координатних каналів

```
if (
    self.saturation_logic == "AXIS_AWARE_V2"
    and (roll_saturated or pitch_saturated)
):
    base_phase = self._select_phase(
        confidence=confidence,
        error_magnitude=abs(error_magnitude),
        error_derivative_x=error_derivative_x,
        error_derivative_y=error_derivative_y,
        roll_saturated=False,
        pitch_saturated=False,
        is_near_target=is_near_target,
    )

    phase_x = (
        "SATURATION_LIMITED"
        if roll_saturated
        else base_phase
    )

    phase_y = (
        "SATURATION_LIMITED"
        if pitch_saturated
        else base_phase
    )

    phase = "SATURATION_LIMITED"

    integral_clamped_x = (
        roll_saturated
        or base_phase == "LOW_CONFIDENCE"
    )

    integral_clamped_y = (
        pitch_saturated
        or base_phase == "LOW_CONFIDENCE"
    )
```

У режимі AXIS_AWARE_V2 ознаки насичення обробляються незалежно для двох координатних каналів. Конфігурація SATURATION_LIMITED застосовується лише до каналу з активною ознакою насичення, тоді як для другого каналу зберігається фазова конфігурація, визначена без урахування насичення.

Лістинг Е.4.

Застосування адаптивних коефіцієнтів

```
if self.position_control_mode == 'ADAPTIVE_PID':
    adaptive_result = self.adaptive_scheduler.compute(
        error_x=error_x,
        error_y=error_y,
        error_magnitude=error_magnitude,
        error_derivative_x=error_derivative_x,
        error_derivative_y=error_derivative_y,
        matches_percent=matches_percent,
        visual_confidence=visual_confidence,
        cross_track_error=cross_track_error,
        completion_axis=self.navigation_completion_axis,
        route_step_index=route_step_index,
        roll_saturated=self.roll_saturated,
        pitch_saturated=self.pitch_saturated,
        is_near_target=is_near_target,
    )

    self.position_pid_x.set_gains(
        kp=adaptive_result.kp_x,
        ki=adaptive_result.ki_x,
        kd=adaptive_result.kd_x,
    )

    self.position_pid_y.set_gains(
        kp=adaptive_result.kp_y,
        ki=adaptive_result.ki_y,
        kd=adaptive_result.kd_y,
    )

    freeze_x = adaptive_result.integral_clamped_x
    freeze_y = adaptive_result.integral_clamped_y
```

Наведені лістинги підтверджують програмну реалізацію основних положень фазового методу адаптивного PID-керування, описаного в розділі 4. У кожному циклі керування визначається поточна фаза роботи позиційного контуру, після чого формуються та коригуються коефіцієнти PID-регуляторів окремо для координатних каналів із урахуванням поточного стану системи та ознак насичення. Результуючі коефіцієнти безпосередньо застосовуються позиційними регуляторами під час формування керуючих впливів.