

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Кваліфікаційна наукова
праця на правах рукопису

Біганський Богдан Миколайович

УДК 681.518:621.865.8

ДИСЕРТАЦІЯ

СИСТЕМА КЕРУВАННЯ МУЛЬТИРОТОРНИМ БПЛА НА ОСНОВІ НЕЙРОМЕРЕЖЕВОЇ ВІЗУАЛЬНО-ІНЕРЦІАЛЬНОЇ ОДОМЕТРІЇ

151 - Автоматизація та комп'ютерно-інтегровані технології

15 - Автоматизація та приладобудування

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей,
результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Б.М. Біганський

Науковий керівник Ковалюк Дмитро Олександрович, к. т. н., доцент

Київ - 2026

АНОТАЦІЯ

Біганський Б. М. Система керування мультироторним БПЛА на основі нейромережевої візуально-інерціальної одометрії. - Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 151 «Автоматизація та комп'ютерно-інтегровані технології» – Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» МОН України, Київ, 2026.

Дисертація присвячена розв'язанню актуальної науково-практичної задачі підвищення ефективності керування безпілотними літальними апаратами, які відіграють ключову роль у розвитку сучасних технологій, знаходячи застосування у широкому спектрі задач: від інспекції інженерних споруд та інфраструктури до пошукових або рятувальних робіт у небезпечних зонах. У таких системах необхідна надійна навігація та керування, особливо при відсутності сигналу глобальної супутникової системи. Тому, забезпечення точної оцінки траєкторії руху, пози та орієнтації в конкретний момент часу в невідомому середовищі є критично важливим для безпеки та ефективного виконання місії.

Еволюція алгоритмів навігації сьогодні нерозривно пов'язана з злиттям сенсорних даних, серед яких візуально-інерціальна одометрія відіграє роль базового джерела просторової інформації. Дослідження традиційних методів візуальної навігації виявило їхню критичну вразливість: вони ефективні лише за умови високої текстурності середовища. Зокрема, алгоритми, що оперують розрідженими ознаками (такі як ORB-SLAM3, VINS-Mono) різко втрачають надійність під час польотів у тунелях, індустриальних коридорах або приміщеннях з однотонними стінами. Втрата візуального відстеження у таких випадках провокує швидке накопичення навігаційного дрейфу, що неминуче викликає збої в контурі керування

БПЛА. Відповідно, створення алгоритмів щільної одометрії постає як актуальне практичне завдання.

В дисертаційному дослідженні детально розглянуто специфіку побудови щільних карт місцевості та способи оцінювання глибини за відеопотоком. З'ясовано, що точність і безперебійність навігаційного процесу прямо корелюють із тим, наскільки якісно реконструйовано геометрію сцени. Враховуючи суттєві обмеження класичних стереосистем і традиційних нейромереж для прогнозування глибини, у роботі доведено необхідність переходу до нових архітектур. Зокрема, обґрунтовано застосування сучасних геометрично-орієнтованих нейромереж. Їх головна перевага полягає у здатності генерувати інваріантну просторову структуру навіть за наявності лише одного монокулярного зображення. Встановлено ключову роль етапу геометричної реєстрації як основної стадії формування візуальних обмежень для корекції інерціального дрейфу. Також для підвищення ефективності процесу візуально-інерціальної одометрії необхідна реалізація оптимального алгоритму комплексування даних.

Описані вище етапи знайшли своє відображення у розробленні системи монокулярної візуально-інерціальної одометрії, яка поєднує нейромережеві оцінки щільної геометрії сцени та алгоритми прямої геометричної реєстрації у схему оптимізації на основі фактор-графа, що забезпечує низький дрейф навігації в умовах відсутності текстур.

У результаті проведеного аналізу сучасних інструментів глибинного навчання визначена модель MoGe (Monocular Geometry) як основа візуального фронтенду. Для математичного моделювання процесу оцінки стану розроблена модель, що включає фактори інерціальної преінтеграції на многовиді $SO(3)$ та щільні візуальні фактори, отримані в результаті реєстрації хмар точок.

Обґрунтовано математичне моделювання системи у вигляді імовірнісного фактор-графа, як найбільш ефективного способу врахування невизначеності вимірювань різнорідних сенсорів.

Сформовано методику та порядок проведення обробки даних у реальному часі. Основну увагу при розробці системи приділено: попередній фільтрації кінематичного стану БПЛА за допомогою фільтра ковзного вікна (SWF), який нівелює високочастотні шуми та формує стабільне початкове наближення для подальшої оптимізації; прямій регресії карт точок без використання матриці інтринсиків; адаптивній фільтрації шумів нейромережі за допомогою масок впевненості; використанню алгоритму ICP (Point-to-Point) з адаптивним порогом для швидкої реєстрації кадрів; а також розрахунку коваріації візуальних вимірювань на основі метрики відповідності (fitness score) для їх злиття у фінальному фактор-графі. Проте реалізації щільної одометрії вимагають великих обчислювальних ресурсів, що ускладнює їх застосування на бортових комп'ютерах мобільних роботів. Це зумовило подальші дослідження пов'язані з оптимізацією архітектури системи та параметрів алгоритмів.

Показано, що класичні підходи до оптимізації (Batch Optimization) потребують значного часу розрахунку при зростанні траєкторії і не можуть бути використані в системах керування реального часу. Обґрунтована необхідність використання інкрементального згладжування та механізму маргіналізації старих станів.

Аналіз існуючих алгоритмів виділення ключових кадрів виявив неефективність просторових евристик. Щоб уникнути надмірного зростання графа, у дослідженні застосовано інформаційно-теоретичний підхід, де критерієм відбору слугує ентропія. Завдяки цьому система додає нові змінні не за фактом проходження певної відстані, а виключно тоді, коли вони забезпечують вагомий інформаційний приріст.

Спільну роботу камери та інерціального вимірювального модуля (IMU) забезпечує спеціальний модуль преінтеграції. Його завдання полягає у тому, щоб агрегувати масив високочастотних показників IMU, перетворюючи їх на єдине компактне геометричне обмеження. Важливою складовою цього етапу є виведені у роботі аналітичні залежності. Вони необхідні для швидкого перерахунку

преінтегрованих величин щоразу, коли система оновлює оцінку зміщень для гіроскопа та акселерометра.

Експериментальна перевірка підтвердила дієвість запропонованого способу формування візуальних факторів. Завдяки застосуванню трансформерної моделі MoGe система здатна захоплювати глобальний контекст сцени. Це означає, що алгоритм зберігає стабільність навігації навіть за повної відсутності локальних текстур – у таких умовах він успішно орієнтується за загальною структурною геометрією простору (лінії перетину стін, площини підлоги та стелі).

Оцінювання точності та надійності створеної системи MoGe-VIO довело її перевагу над класичними рішеннями. Тестування алгоритму виконувалося на базі загальноприйнятих датасетів EuRoC MAV та TUM-VI, де розрахована траєкторія зіставлялася з істинною за критерієм абсолютної похибки траєкторії. Отримані дані підтверджують: під час польотів у зонах зі слабо вираженою текстурою (наприклад, локації TUM-VI Outdoors або звичайні коридори) запропонований метод генерує значно меншу похибку позиціонування порівняно з ORB-SLAM3 та VINS-Mono. Крім того, система продемонструвала високу стійкість до накопичення дрейфу масштабу.

Отримані результати підтверджують, що система здатна працювати на сучасних вбудованих платформах з частотою оновлення одометрії 10–15 Гц. Це дозволяє безпосередньо інтегрувати систему в контур керування БПЛА. Важливою перевагою створеного монокулярного рішення є його апаратна ефективність: на відміну від громіздких LiDAR-сенсорів чи стереокамер, воно потребує значно менше енергії та має компактні габарити, зберігаючи при цьому високу точність щільної реконструкції простору. Крім обчислення самої траєкторії польоту, алгоритм паралельно генерує щільну карту перешкод у режимі реального часу. Саме ця властивість відіграє ключову роль у безпечній навігації, оскільки забезпечує базу для планування шляху та своєчасного уникнення зіткнень.

З урахуванням розробленої системи візуально-інерціальної одометрії було розглянуто задачу стабілізації висоти БПЛА під дією зовнішніх збурень, зокрема поривів вітру. Використано стандартну математичну модель такого об'єкту, що базується на жорсткому тілі з шістьма ступенями вільності (6-DOF), описаними рівняннями Ньютона та Ейлера. Розглядаються рівняння динаміки, що описують результуючу силу та момент, а також швидкості та кутові швидкості у відповідній системі координат.

Для синтезу контурів керування було обрано три підходи: класичний PID-регулятор, регулятор на базі нечіткої логіки та лінійно-квадратичний регулятор (LQR). Щоб максимально наблизити експеримент до умов реальної експлуатації, параметри керування розраховувалися з урахуванням апаратних часових затримок надходженню даних від навігаційного модуля. Детально проаналізовано ключові показники: якість перехідних процесів, загальний запас стійкості системи, а також її здатність реагувати на зовнішні збурення та повертатися до стану рівноваги.

Результати демонструють відмінності у швидкості реакції, точності стабілізації та адаптивності методів керування, що дозволяє визначити оптимальні підходи для забезпечення надійного утримання висоти БПЛА в умовах змінного середовища. Моделювання систем керування БПЛА здійснено за допомогою MATLAB Simulink.

Наукова новизна отриманих результатів дисертаційної роботи полягає в наступному: *вперше розроблено* систему монокулярної візуально-інерціальної одометрії, що поєднує нейромережеві оцінки щільної геометрії сцени (MoGe) та алгоритми прямої геометричної реєстрації (ICP) у схему оптимізації на основі фактор-графа; *удосконалено* критерій оптимізації для оцінки траєкторії та станів БПЛА, який комплексно враховує преінтегровані пози, зміщення хмар точок між кадрами та апріорні геометричні знання від нейромережі, що дозволяє в рамках єдиної задачі мінімізації похибки поєднати візуальні обмеження від ICP та інерціальні дані, підвищуючи точність оцінки масштабу; *вперше розроблено* систему підтримки прийняття рішень для конфігурації параметрів алгоритмів

VIO/SLAM на бортових Edge-платформах, яка на відміну від існуючих підходів використовує виявлені закономірності «точок перегину» часової складності обчислень та результати порівняльного аналізу ефективності бібліотек нелінійної оптимізації (GTSAM, SymForce, Ceres); *дістали подальшого розвитку* математичні моделі систем керування БПЛА в умовах дії висотних збурень на основі нечіткого (Fuzzy) та лінійно-квадратичного (LQR) регуляторів, параметри яких адаптовано до динамічних характеристик часової затримки, що виникає при роботі VIO-алгоритмів;

Практичне значення отриманих результатів полягає у наступному: *розроблено робастну систему навігації* для автономного інспектування інженерних споруд (промислових об'єктів, фасадів, мостів). У складних індустріальних чи міських середовищах алгоритм гарантує високоточну локалізацію БПЛА, що є критично важливим етапом для подальшої просторової 3D-реконструкції; *Забезпечено апаратну оптимізацію та здешевлення комплексу*: перехід від масивних LiDAR-сканерів до монокулярної камери дає змогу ефективно інтегрувати розроблену систему на дрони, обладнані бортовими нейроприскорювачами (Edge GPU); *Розроблено методику адаптації обчислювального конвеєра спеціально для Edge-платформ*. Підхід дозволяє динамічно балансувати між деталізацією (щільністю) генерованих хмар точок та швидкістю оновлення навігаційних станів, гарантуючи роботу нейромереж в режимі реального часу; *підвищено стабільність контуру керування* під час виконання інспекційних польотів. Завдяки врахуванню апаратних часових затримок, притаманних обчисленням у глибоких нейромережах, апарат значно точніше утримує задану траєкторію та висоту навіть під впливом динамічних вітрових збурень.

Ключові слова: БПЛА, комп'ютерний зір, SLAM, візуально-інерціальна одометрія, орієнтація та навігація, IMU, нейронні мережі, моделювання, оптимізація, керування, стабілізація висоти, ПД-регулятор, нечітка логіка, LQR-регулятор.

ABSTRACT

Bihanskyi B. M. Control System for Multi-Rotor UAV Based on Neural Network Visual-Inertial Odometry. – Qualification scientific work with the status of a manuscript.

Dissertation for the degree of Doctor of Philosophy in specialty 151 «Automation and Computer-Integrated Technologies» – National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Ministry of Education and Science of Ukraine, Kyiv, 2026.

The thesis is devoted to solving the relevant scientific and practical problem of increasing the control efficiency of unmanned aerial vehicles, which play a key role in the development of modern technologies, finding application in a wide range of tasks: from infrastructure inspection and logistics to search and rescue operations in disaster zones. Strict requirements regarding navigation and control reliability are imposed on such systems, especially in conditions of absence or suppression of global satellite system signals. Ensuring accurate trajectory estimation in an unknown environment is critical for the safety and efficiency of mission execution.

Modern trends in the development of navigation algorithms determine the transition to sensor fusion methods, where visual-inertial odometry acts as the main source of information. The conducted analysis of existing visual navigation methods highlighted their dependence on the presence of a sufficient number of texture features in the surrounding environment. Such classical sparse feature-based algorithms as VINS-Mono and ORB-SLAM3 are vulnerable in conditions of low texture saturation (e.g., rooms with white walls, industrial corridors, tunnels). Loss of visual tracking in these conditions leads to rapid accumulation of navigation system drift and, as a consequence, to control errors; therefore, the development of dense odometry methods is particularly relevant.

The work analyzes methods for obtaining depth from images and the specifics of building dense terrain maps. It is shown that one of the determining factors in terms of navigation accuracy and stability is the quality of geometric scene reconstruction. The limitations of traditional stereo vision methods and classical depth estimation neural

networks are analyzed. Based on the analysis, the feasibility of using novel geometry-oriented neural networks capable of recovering the affine-invariant structure of a scene from a single image is substantiated. The key role of the geometric registration stage as the main stage of forming visual constraints for correcting inertial drift is established. Also, to increase the efficiency of the visual-inertial odometry process, it is necessary to implement an optimal data fusion algorithm.

The stages described above are reflected in the development of a monocular visual-inertial odometry system that combines neural network estimates of dense scene geometry and direct geometric registration algorithms into an optimization scheme based on a factor graph, ensuring low navigation drift in textureless conditions. As a result of the conducted analysis of modern deep learning tools, the MoGe (Monocular Geometry) model was defined as the basis of the visual Frontend. For mathematical modeling of the state estimation process, a model was developed that includes inertial pre-integration factors on the $SO(3)$ manifold and dense visual factors obtained as a result of point cloud registration. The mathematical modeling of the system in the form of a probabilistic factor graph is substantiated as the most effective way to account for the measurement uncertainty of heterogeneous sensors. A methodology and procedure for real-time data processing have been established. During the system development, the main focus was placed on: pre-filtering the UAV kinematic state using a Sliding Window Filter (SWF), which mitigates high-frequency noise and provides a stable initial guess for subsequent optimization; direct point map regression without utilizing the camera intrinsics matrix; adaptive filtering of neural network noise using confidence masks; the application of the Point-to-Point ICP algorithm with an adaptive threshold for fast frame registration; and the calculation of visual measurement covariance based on the fitness score for their fusion in the final factor graph. However, dense odometry implementations require significant computational resources, which complicates their application on the onboard computers of mobile robots. This necessitated further research related to the optimization of the system architecture and algorithm parameters.

It is shown that classical optimization approaches (Batch Optimization) require significant calculation time as the trajectory grows and cannot be used in real-time control systems. The necessity of using incremental smoothing and a mechanism for the marginalization of old states is substantiated. An analysis of keyframe selection methods was conducted. Based on the analysis, the feasibility of using an information-theoretic criterion based on entropy (the determinant of the information matrix) instead of simple distance heuristics is substantiated. This allowed for optimizing the graph size by adding new variables only when they make a significant information contribution.

To build a simplified but effective model of the interaction between the IMU and the camera, a pre-integration module was implemented, which allows collapsing hundreds of high-frequency inertial sensor measurements into a single geometric constraint. Analytical expressions were developed for correcting pre-integrated values when the accelerometer and gyroscope bias estimates change. Research results demonstrated the high efficiency of the proposed approach to forming visual factors. It was established that the use of the global scene context, provided by the MoGe transformer architecture, allows the system to function stably even in the complete absence of local texture features, relying on the structural geometry of the room (floor, ceiling, corners).

The conducted study of the accuracy and adequacy of the developed system showed its superiority over existing analogues. To verify the algorithm's effectiveness, a series of experiments was conducted on standard datasets EuRoC MAV and TUM-VI. Verification for adequacy was carried out by comparing the estimated trajectory with the ground truth using the Absolute Trajectory Error (ATE) metric. Analysis of the obtained results indicates that in low-texture conditions (TUM-VI Outdoors, corridors), the proposed MoGe-VIO method ensures a reduction in positioning error compared to VINS-Mono and ORB-SLAM3, demonstrating resistance to scale drift. The obtained results confirm that the system is capable of operating on modern embedded platforms with an odometry update rate of 10–15 Hz. This allows it to be used in the UAV control loop. Compared to systems based on stereo cameras or LiDAR, the developed monocular system has smaller

dimensions and power consumption, while providing comparable accuracy of dense environment reconstruction. The developed approach allows not only estimating the trajectory but also building a dense obstacle map in real-time, which is critical for path planning and collision avoidance tasks.

Taking into account the developed visual-inertial odometry system, the problem of quadcopter altitude stabilization under the influence of external disturbances, particularly wind gusts, was considered. A standard mathematical model of such an object based on a rigid body with six degrees of freedom (6-DOF), described by Newton's and Euler's equations, was used. The equations of dynamics describing the resultant force and moment, as well as velocities and angular velocities in the corresponding coordinate system, are considered.

The synthesis of control systems was carried out based on a classical PID regulator, fuzzy logic, and a Linear-Quadratic Regulator (LQR). The calculation of regulator parameters was performed based on data coming from the navigation system with certain time delays, reflecting functioning in real conditions. Simulation of control systems using each type of controller was implemented, and the quality of transient processes, stability, and the ability to restore equilibrium after the impact of disturbances were evaluated.

The results demonstrate differences in reaction speed, stabilization accuracy, and adaptability of control methods, which allows determining optimal approaches for ensuring reliable altitude holding of the quadcopter in a variable environment. Simulation of quadcopter control systems was carried out using MATLAB Simulink.

The scientific novelty of the results obtained in this dissertation is as follows: *for the first time*, a monocular visual-inertial odometry (VIO) system has been developed that integrates neural network-based estimates of dense scene geometry (MoGe) and direct geometric registration algorithms (ICP) into a factor-graph-based optimization scheme; *an optimization criterion* for estimating UAV trajectories and states has been improved. This criterion comprehensively accounts for pre-integrated poses, point cloud offsets between frames, and a priori geometric knowledge from the neural network. This allows

for the integration of visual constraints from ICP and inertial data within a single error minimization task, thereby increasing the accuracy of scale estimation; *for the first time*, a decision support system for configuring VIO/SLAM algorithm parameters on onboard Edge platforms has been developed. Unlike existing approaches, it utilizes identified patterns of "inflection points" in computational time complexity and the results of a comparative performance analysis of non-linear optimization libraries (GTSAM, SymForce, Ceres); mathematical models of UAV control systems operating under altitude disturbances *have been further developed*. These models are based on Fuzzy and Linear-Quadratic Regulator (LQR) controllers, with parameters adapted to the dynamic characteristics of time delays inherent in the operation of VIO algorithms.

The main practical results are: the development of a robust navigation system capable of stable operation in complex industrial and urban environments during the autonomous inspection of large-scale engineering structures (bridges, building facades, industrial facilities), where high-precision UAV localization is critical for subsequent spatial 3D reconstruction; hardware optimization and cost reduction achieved by replacing heavy laser scanners (LiDAR) with a monocular camera, thereby expanding the application potential of modern UAVs equipped with onboard neural accelerators (Edge GPUs); the formulation of a methodology for adapting the computational pipeline to Edge platforms, enabling dynamic balancing between the density of generated point clouds and the update frequency of navigation states to ensure the real-time execution of neural network algorithms; and the enhancement of control system stability during inspection flights by accounting for the processing latencies inherent to deep neural networks, which significantly improves the accuracy of trajectory tracking and altitude maintenance under dynamic wind disturbances.

Keywords: UAV, computer vision, SLAM, visual-inertial odometry, orientation and navigation, IMU, neural networks, modeling, optimization, control, altitude stabilization, PID controller, fuzzy logic, LQR controller.

Список публікацій здобувача, в яких опубліковані основні наукові результати дисертації:

1. Біганський, Б. М., & Ковалюк, Д. О. (2025). Порівняння програмних реалізацій алгоритму Bundle Adjustment для задачі SLAM. Вісник Національного технічного університету України “Київський політехнічний інститут імені Ігоря Сікорського”. Серія: Хімічна інженерія, екологія та ресурсозбереження, (1), 48–55 . <https://doi.org/10.20535/2617-9741.1.2025.325837>.

Науковим керівником Ковалюком Д.О. виконано постановку завдання дослідження, перевірку коректності та достовірності отриманих результатів.

Здобувачем виконано аналіз існуючих бібліотек, що реалізують алгоритм Bundle Adjustment. Запропоновано програмне рішення для порівняння бібліотек за швидкодією, кількістю ітерацій, значенням помилки на тестових наборах даних.

2. Біганський, Б. М., & Ковалюк, Д. О. (2025). Порівняння систем керування мультироторним БПЛА (укр.). Вісник Національного технічного університету України “Київський політехнічний інститут імені Ігоря Сікорського”. Серія: Хімічна інженерія, екологія та ресурсозбереження, (3), 52–60. <https://doi.org/10.20535/2617-9741.3.2025.340377>.

Науковим керівником Ковалюком Д.О. виконано постановку завдання дослідження, перевірку коректності та достовірності отриманих результатів.

Здобувачем виконано порівняльний аналіз систем керування БПЛА для задачі стабілізації висоти. Розраховано параметри систем на основі ПД-регулятора, нечіткої логіки та лінійно-квадратичного регулятора.

3. Біганський, Б. М., & Ковалюк, Д. О. (2025). Алгоритм візуально-інерційної одометрії з використанням геометрично-орієнтованої нейронної мережі (укр.). Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки, Том 36 (75), № 6 52–60. <https://doi.org/10.32782/2663-5941/2025.6.2/07>.

Науковим керівником Ковалюком Д.О. виконано постановку завдання дослідження. Проведені імітаційні дослідження для підтвердження адекватності запропонованого алгоритму.

Здобувачем запропоновано архітектуру системи візуально-інерціальної одометрії, що базується на інтеграції геометрично-орієнтованої нейронної мережі та методів ймовірнісної оптимізації. Створено її математичну модель в межах єдиного фактор-графа, здійснено програмну реалізацію.

Список публікацій здобувача, які засвідчують апробацію матеріалів дисертації:

4. Ковалюк Д. О., Біганський Б. М., Математична модель мультироторного БПЛА з 6-DOF – Автоматизація та комп'ютерно-інтегровані технології - 2025: матеріали XI Міжнародної науково-практичної конференції молодих учених, аспірантів і студентів (АКІТ – 2025) - Київ, 7 травня 2025 р. – К.: КПІ ім. Ігоря Сікорського, ISBN 978-966-990-144-6 – С. 41-42.

Науковим керівником Ковалюком Д.О. виконано постановку задачі.

Здобувачем виконано дослідження математичної моделі БПЛА.

5. Біганський Б. М., Ковалюк Д. О. Підвищення ефективності візуально-інерціальної одометрії з використанням геометрично-орієнтованої нейронної мережі – Матеріали XII Міжнародної науково-технічної Internet-конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами», 27 листопада 2025 [Електронний ресурс]. – К: НУХТ, 2025. – С.226-227 – ISBN 978-966-612-391-9.

Науковим керівником Ковалюком Д.О. виконано постановку задачі, перевірку коректності роботи алгоритму.

Здобувачем реалізовано алгоритм візуально-інерціальної одометрії з використанням нейронної мережі та проведено експериментальні дослідження.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	17
ВСТУП.....	18
РОЗДІЛ 1. Аналіз методів навігації та керування БПЛА для моніторингу інженерних конструкцій в умовах дефіциту візуальних ознак	25
1.1. Особливості задачі SLAM для БПЛА.....	26
1.2. Еволюція методів SLAM	30
1.3. Аналіз методів візуально-інерціальної одометрії.....	34
1.3.1. Аналіз методів VIO на основі розріджених ознак та їх обмеження	36
1.3.2. Прямі та напівпрямі методи для трекингу у VIO	38
1.3.3. Обмеження методів VIO для задач з реконструкцією середовища.....	41
1.3.4. Обґрунтування розробки гібридної архітектури VIO	43
1.4. Аналіз підходів та методів керування БПЛА.....	45
Висновки до розділу 1	53
РОЗДІЛ 2. Система монокулярної візуально-інерціальної одометрії на основі нейронної мережі MoGe.....	55
2.1. Архітектура системи	55
2.1.1. Концептуальна модель та потік даних	55
2.1.2. Функціональні модулі.....	57
2.2. Модуль обробки даних з камери.....	60
2.2.1. Архітектура нейронної мережі MoGe для генерації щільних карт глибини... 61	
2.2.2. Попередня обробка та підготовка хмар точок.....	63
2.3. Оцінка руху на основі геометричної реєстрації	65
2.4. Преінтеграції вимірювань IMU на основі кінетичної моделі із шумами.....	68
Висновки до розділу 2	72
РОЗДІЛ 3. Оцінка стану БПЛА на основі оптимізації фактор-графа	74
3.1. Критерій злиття даних на основі фактор-графа	74

3.2. Порівняння програмних реалізацій алгоритму Bundle Adjustment для задачі SLAM.....	79
3.3. Застосування нелінійної оптимізації	91
3.4. Керування ключовими кадрами та маргіналізація	93
Висновки до розділу 3	95
РОЗДІЛ 4. Математичне моделювання та керування БПЛА	97
4.1. Апаратна архітектура та компоненти БПЛА	97
4.2. Математичне моделювання динаміки БПЛА.....	106
4.3. Розробка систем керування на основі PID, Fuzzy Logic, LQR.....	110
Висновки до розділу 4	121
РОЗДІЛ 5. Експериментальні дослідження та валідація системи візуально-інерціальної одометрії.....	124
5.1. Апаратна реалізація та обчислювальні ресурси	124
5.2. Оцінка точності системи.....	129
Висновки до розділу 5	134
ЗАГАЛЬНІ ВИСНОВКИ	136
Список використаних джерел.....	139
Додатки.....	149
Додаток А. UML діаграма класів для розв'язання задачі Bundle Adjustment	149
Додаток Б. Моделювання системи керування в Simulink	152
Додаток В. Результати роботи MoGe-VIO (візуалізація в Rerun).....	154

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

SLAM – Simultaneous Localization and Mapping;

IMU – Inertial Measurement Unit;

VIO – Visual-Inertial Odometry;

VO – Visual Odometry;

SIFT – Scale-Invariant Feature Transform;

ORB – Oriented FAST and Rotated BRIEF;

LiDAR – Light Detection and Ranging;

БПЛА – Безпілотний літальний апарат;

ICP – Iterative Closest Point;

LQR – Linear Quadratic Regulator;

PID – Proportional-Integral-Derivative regulator;

ВСТУП

Актуальність теми. Сфера застосування безпілотних літальних апаратів (БПЛА) безперервно розширюється, проте ключовим бар'єром залишається проблема їхньої повної автономності. Особливої актуальності це питання набуває під час польотів у зонах, де сигнали супутникової навігації є недоступними: у закритих приміщеннях, підземних комунікаціях, тунелях тощо. За таких обставин виникає необхідність створення принципово нових навігаційних архітектур.

Розгляд наявних алгоритмів візуальної одометрії виявляє їхні суттєві обмеження. Наприклад, класичні рішення на базі розріджених ознак (зокрема VINS-Mono та ORB-SLAM) часто втрачають трекінг в однотонних, бідних на текстури локаціях, а також дають збій під час різких маневрів. Паралельно індустрія БПЛА вимагає мінімізації апаратної бази: замість масивних, енергоємних і дорогих лазерних сканерів (LiDAR) перевага віддається легким монокулярним камерам. Оскільки від точності та швидкодії навігаційного контуру залежить коректність роботи системи керування, то удосконалення цих методів є надзвичайно актуальним завданням.

Вагомий внесок у розробку теоретичних та прикладних основ систем SLAM та керування БПЛА зробили відомі закордонні та вітчизняні вчені. Фундаментальні підходи до побудови систем візуально-інерціальної одометрії (VIO) та методів нелінійної оптимізації закладені в роботах А. Мурікіса (A. Mourikis), С. Румеліотіса (S. Roumeliotis) та Ф. Деллаерта (F. Dellaert). Розробка найбільш ефективних практичних реалізацій VIO, що стали галузевими стандартами, пов'язана з іменами Т. Ціня (T. Qin) та Ш. Шеня (S. Shen) (автори системи VINS-Mono), а також В. Усенка (V. Usenko) та Д. Кремерса (D. Cremers) (розробники системи Basalt). Теоретичне узагальнення методів візуального SLAM та їх програмної реалізації детально висвітлено у фундаментальній праці Г. Гао (X. Gao) та Т. Чжана (T. Zhang)

(автори «SlamBook»). Питаннями робастного керування та стабілізації БПЛА займалися С. Агарвал (S. Agarwal) та А. Наджм (A. Najm).

Серед українських вчених значний внесок у вирішення задач навігації в умовах відсутності GPS та розробку контурів керування БПЛА внесли В. М. Синєглазов, С. Г. Стіренко, А. Ю. Сазонов, В. В. Бурнашев, О. А. Сущенко, В. Б. Ларін, О. В. Збруцький та інші. Їхні дослідження щодо навігації, керування, планування маршруту та стабілізації апаратів в умовах невизначеності стали основою для вдосконалення алгоритмів автономного польоту.

Великий стрибок у вирішенні цих завдань став можливим завдяки впровадженню технологій глибокого навчання. Сучасні нейромережі здатні генерувати щільну карту глибини на основі лише одного зображення, надаючи набагато більше просторових даних порівняно з класичною геометрією. Залучення таких щільних карт до контуру керування вирішує одразу дві проблеми: забезпечує високоточну локалізацію апарата та створює базу для своєчасного виявлення перешкод під час планування маршруту.

Таким чином, розробка системи навігації, яка поєднує переваги нейромережевої оцінки геометрії та точність інерціальних вимірювань у межах єдиної схеми, є актуальною задачею, що має важливе значення для підвищення автономності, надійності та ефективності керування БПЛА.

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

Дисертаційну роботу виконано на кафедрі технічних та програмних засобів автоматизації Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» відповідно до плану наукових досліджень за темою «Створення комп'ютерно-інтегрованих систем керування технологічними процесами та виробництвами для забезпечення якості продукції та ресурсозбереження» – державний реєстраційний номер 0121U114712.

Особисто автором в НДР запропоновано математичні моделі систем керування БПЛА в умовах дії збурень, а також математичне, алгоритмічне та програмне

забезпечення системи візуально-інерціальної одометрії на основі нейронної мережі MoGe.

Мета і завдання дослідження. Метою роботи є підвищення ефективності керування безпілотним літальним апаратом (БПЛА) в умовах відсутності сигналів GPS шляхом розробки системи монокулярної візуально-інерціальної одометрії, що поєднує нейромережеві оцінки щільної геометрії сцени (MoGe) та алгоритми прямої геометричної реєстрації (ICP) у схему оптимізації на основі фактор-графа.

Для досягнення поставленої мети було вирішено такі **завдання**:

1. Провести комплексний аналіз існуючих методів візуально-інерціальної одометрії (VIO), виявити переваги та недоліки систем на основі розріджених ознак (зокрема, їхню деградацію в середовищах з низькою текстурою) та дослідити доцільність розробки системи на основі щільних карт глибини.

2. Дослідити математичну модель динаміки руху БПЛА, що описує його кінематичні та динамічні характеристики, для забезпечення фізично коректної передінтеграції вимірювань інерціального вимірювального блоку (IMU).

3. Розробити модуль обробки даних з камери, що використовує нейронну мережу MoGe для перетворення вхідних монокулярних RGB зображень у щільні 3D хмари точок, що містять інформацію про геометрію сцени.

4. Розробити модуль оцінки відносного руху на основі ICP алгоритму для прямої геометричної реєстрації послідовних хмар точок.

5. Розробити модуль оптимізації на основі фактор-графа, що забезпечує об'єднання візуальних вимірювань (від ICP) та інерціальних вимірювань (отриманих на основі розробленої моделі БПЛА та передінтеграції IMU) для отримання оцінки траєкторії та станів.

6. Здійснити програмну реалізацію розробленої системи VIO та виконати експериментальне дослідження її ефективності. Оцінити точність та робастність системи на стандартних публічних наборах даних та провести порівняльний аналіз з провідними аналогами.

7. Дослідити систему керування БПЛА в умовах збурень на основі даних, отриманих із розробленої системи VIO.

Об'єктом дослідження є процеси автономної навігації та керування рухом безпілотних літальних апаратів мультироторного типу в умовах невизначеності зовнішнього середовища.

Предметом дослідження є математичні моделі систем керування БПЛА та методи гібридної візуально-інерціальної одометрії, що враховують інерціальні вимірювання із нейромережевою оцінкою щільної геометрії сцени.

Методи дослідження. Для розв'язання поставлених завдань у роботі використано комплекс взаємопов'язаних методів:

- теоретичного узагальнення та порівняльного аналізу - для визначення стану проблеми, аналізу існуючих алгоритмів VIO та обґрунтування вибору напрямку досліджень;

- математичного моделювання та теоретичної механіки - для опису динаміки 6-DOF моделі БПЛА та побудови моделі інерціальних вимірювань;

- глибокого навчання та комп'ютерного зору - для реалізації візуального фронтенду на основі нейромережі MoGe та генерації щільних хмар точок;

- обчислювальної геометрії - для реалізації алгоритмів реєстрації хмар точок (ICP) та оцінки відносного руху;

- теорії графів та нелінійної оптимізації - для розробки бекенду системи на основі фактор-графів та розв'язання задачі мінімізації глобальної похибки траєкторії;

- теорії автоматичного керування - для синтезу та аналізу алгоритмів керування (Fuzzy Logic, LQR) в умовах зовнішніх збурень та часових затримок вхідних даних;

- комп'ютерне моделювання - для верифікації розроблених алгоритмів на стандартних датасетах та оцінки точності позиціонування.

Наукова новизна одержаних результатів:

- *Вперше розроблено* систему монокулярної візуально-інерціальної одометрії, що поєднує нейромережеві оцінки щільної геометрії сцени (MoGe) та алгоритми прямої геометричної реєстрації (ICP) у схему оптимізації на основі фактор-графа, що, на відміну від існуючих аналогів, забезпечує стійку роботу та низький дрейф навігації в умовах відсутності текстур.

- *Удосконалено* критерій оптимізації для задачі оцінки траєкторії та станів БПЛА, який комплексно враховує преінтегровані пози, зміщення хмар точок між кадрами та апріорні геометричні знання від нейромережі, що дозволяє в рамках єдиної задачі мінімізації похибки поєднати візуальні обмеження від ICP та інерціальні дані, підвищуючи точність оцінки масштабу.

- *Вперше розроблено* систему підтримки прийняття рішень для конфігурації параметрів алгоритмів VIO/SLAM на бортових Edge-платформах, яка на відміну від існуючих підходів використовує виявлені закономірності «точок перегину» часової складності обчислень та результати порівняльного аналізу ефективності бібліотек нелінійної оптимізації (GTSAM, SymForce, Ceres). Це дозволяє за рахунок точності позиціювання, щільності геометричних даних та частоти оновлення станів забезпечити стабільність контурів автономного керування в умовах обмежених обчислювальних ресурсів.

- *Дістали подальшого розвитку* математичні моделі систем керування БПЛА в умовах дії збурень на основі нечіткого та лінійно-квадратичного регуляторів, параметри яких адаптовано до часової затримки, що виникає при роботі VIO-алгоритмів. Це дозволяє врахувати запізнення у контурі зворотного зв'язку та забезпечити стабілізацію висоти в умовах вітрових збурень.

Практичне значення одержаних результатів дисертаційного дослідження полягає у наступному:

- *Розробка системи візуально-інерціальної одометрії* на основі нейромережі MoGe та алгоритму ICP дозволяє безпілотним апаратам стабільно орієнтуватися в

приміщеннях з однотонними стінами, тунелях та зонах дії РЕБ.

- *Апаратна оптимізація та зниження вартості* за рахунок використання монокулярної камери замість дорогих і важких лазерних сканерів. Це дозволяє зменшити масу та енергоспоживання бортового обладнання, використовувати розроблені алгоритми на малогабаритних БПЛА обмеженої потужності.

- *Методика адаптації до Edge-платформ* дозволяє за рахунок пошуку «точок перегину» ефективно налаштовувати баланс між точністю обчислень та швидкістю роботи алгоритмів на реальному бортовому обладнанні в режимі реального часу.

- *Підвищення стабільності керування* на основі Fuzzy та LQR-регуляторів з урахуванням часових затримок обробки візуальних даних. Це дозволяє мінімізувати автоколивання БПЛА та підвищити точність утримання висоти в умовах вітрових збурень.

- *Інструментарій для розробників* у вигляді сформованих порівняльних характеристик бібліотек нелінійної оптимізації (GTSAM, SymForce, Ceres), що слугують практичним посібником для вибору архітектури майбутніх систем SLAM залежно від вимог до обчислювальної потужності.

Особистий внесок здобувача

Дисертаційна робота є самостійно виконаною науковою працею. Здобувач сформулював мету та завдання дослідження. Всі представлені наукові результати, приклади та експериментальні розрахунки, викладені у дисертації, одержані здобувачем одноосібно. З опублікованих наукових праць, які написані у співавторстві, використано лише результати розробок здобувача. Особистий внесок автора конкретизовано у списку публікацій.

Апробація результатів дисертації. Результати досліджень, викладених у дисертації, було висвітлено на таких конференціях.

- XI Міжнародна науково-практична конференція молодих учених, аспірантів і студентів «Автоматизація та комп'ютерно-інтегровані технології» - Київ, 7 травня 2025 р., КПІ ім. Ігоря Сікорського.

- XII Міжнародна науково-технічна Internet-конференція «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами» - Київ, 27 листопада 2025 р., Національний університет харчових технологій.

Публікації. Основні результати дисертаційної роботи опубліковано у 5 наукових роботах: 3 статті у наукових фахових виданнях України, 2 тез доповідей міжнародних конференцій.

Структура і обсяг роботи. Дисертація складається з анотації, вступу, п'яти розділів, висновків, списку використаних джерел та 3 додатків. Загальний обсяг дисертації складає 148 сторінок, з них основний текст роботи викладено на 121 сторінці друкованого тексту, у тому числі 32 рисунки, 10 таблиць, 3 додатків та списку використаної літератури з 88 найменувань.

РОЗДІЛ 1. АНАЛІЗ МЕТОДІВ НАВІГАЦІЇ ТА КЕРУВАННЯ БПЛА ДЛЯ МОНІТОРИНГУ ІНЖЕНЕРНИХ КОНСТРУКЦІЙ В УМОВАХ ДЕФІЦИТУ ВІЗУАЛЬНИХ ОЗНАК

Архітектура сучасних автономних БПЛА вимагає глибокого поєднання модулів навігації та керування в єдиний замкнений контур регулювання. Практика доводить, що розглядати їх ізольовано недоцільно. Виконання місії визначається двома рівноцінними факторами: наскільки точно навігатор оцінює стан апарата та наскільки швидко контур керування реагує на ці показники. Навіть найточніший SLAM-алгоритм виявиться неефективним, якщо базовий виконавчий рівень (Low-Level Control) не встигатиме нівелювати затримки оновлення координат або не зможе компенсувати динамічні збурення.

Під час об'єднання цих підсистем на перший план виходить проблема шумів вимірювань та апаратних затримок. Будь-яка візуально-інерціальна система (VIO) неминуче працює з певною затримкою через час експозиції камери та обчислювальну складність обробки кадрів. Ігнорування цієї специфіки під час налаштування контуру керування (наприклад, без залучення прогнозних моделей чи преінтеграції даних IMU) призводить до аварійних ситуацій. Саме тому синтез параметрів регуляторів має спиратися на динамічні характеристики конкретної навігаційної системи.

Як наслідок, архітектура надійної автономної платформи має базуватися на принципах наскрізного проектування. За такого підходу алгоритми сприйняття середовища оптимізуються спільно з алгоритмами стабілізації. Це забезпечує необхідний компроміс між ідеальним утриманням траєкторії та загальною робастністю БПЛА, гарантуючи його безпечну роботу в умовах невизначеності навколишнього середовища.

1.1. Особливості задачі SLAM для БПЛА

У сучасній робототехніці проблема повноцінної автономії БПЛА в умовах втрати (або нестабільності) супутникових сигналів залишається вкрай актуальною. Базовим інструментом для її розв'язання виступають алгоритми SLAM (Simultaneous Localization and Mapping), які дають змогу одночасно картографувати невідомий простір та локалізувати апарат у ньому.

Особливою складністю використання таких технологій є 3D-реконструкція відкритого простору та автономний моніторинг масштабних інфраструктурних об'єктів. До таких споруд належать дамби, мости, висотні фасади, вітрові турбіни та масивні промислові комплекси (резервуари, димові труби). Систематична інспекція цих поверхонь є необхідною для вчасного виявлення корозії, мікротріщин чи структурних деформацій, що гарантує запобігання техногенним аваріям. Застосування автономних БПЛА переводить цей процес у цифровий формат: зникає потреба в послугах промислових альпіністів, монтуванні дорогих риштувань чи зупинці роботи підприємства. Відповідно, нівелюються загрози для життя персоналу, а зібрані високоточні дані слугують основою для подальшого аналізу цифрової копії будівлі.

Попри ці переваги, практична організація інспекційних польотів стикається із серйозними технічними бар'єрами. Насамперед, масивність конструкцій та щільна міська забудова неминуче екранують GPS-сигнал. Застосування класичної візуальної одометрії як альтернативи ускладнюється специфікою індустріальних поверхонь: бетонні моноліти страждають від низької текстурності, а скло та метал створюють відблиски і повторювані патерни. Хоча можливість використання лазерних сканерів (LiDAR) здатна вирішити питання навігації, вона є неефективною для легких БПЛА через надмірну вагу, ціну та високе енергоспоживання. Комплекс цих проблем посилюється непередбачуваними вітровими потоками поблизу висотних споруд. Це формує жорстку вимогу: бортова навігація повинна

гарантувати максимальну точність із мінімальними часовими затримками для безпечного утримання БПЛА біля перешкод.

У загальному вигляді задачу SLAM можна сформулювати як проблему ймовірнісної оцінки [1]. Метою є знаходження апостеріорного розподілу ймовірностей траєкторії робота (БПЛА) $X_{0:k}$ та карти середовища m на основі послідовності вимірювань сенсорів $Z_{0:k}$ та керуючих впливів $U_{0:k}$:

$$P(X_{0:k}, m | Z_{0:k}, U_{0:k}) \quad (1.1)$$

де

- $X_{0:k} = \{x_0, x_1, \dots, x_k\}$ - траєкторія БПЛА (послідовність станів, що включають координати, орієнтацію, швидкості та зміщення сенсорів);
- m - карта навколишнього середовища (може бути представлена хмарою точок, сіткою зайнятості або розрідженими ознаками);
- $Z_{0:k}$ - дані, отримані від сенсорів (камери, IMU);
- $U_{0:k}$ - дані одометрії або керуючі команди.

На відміну від наземних колісних роботів, застосування SLAM на БПЛА пов'язане з низкою специфічних викликів, які визначають вибір архітектури системи:

- 6 ступенів вільності (6 DoF): БПЛА рухається у тривимірному просторі, змінюючи крен, тангаж та рискання (roll, pitch, yaw). Це вимагає складніших алгоритмів оцінки стану порівняно з планарним рухом (3 DoF).
- Обмеження SWaP (Size, Weight, and Power): бортові обчислювальні ресурси квадрокоптерів (особливо мікро- та міні-класу) суттєво обмежені. Використання "важких" 3D-лідарів часто неможливе через їхню вагу та енергоспоживання, що зміщує фокус досліджень у бік візуальних сенсорів.
- Швидка динаміка та вібрації: різкі маневри призводять до розмиття зображення (motion blur), а високочастотні вібрації від роторів вносять значний шум у дані інерціальних вимірювальних блоків (IMU).

- Проблема масштабу: при використанні монокулярних камер втрачається інформація про метричний масштаб сцени. Для відновлення масштабу необхідно використовувати додаткові сенсори (IMU, стереокамери) або апіорну інформацію.

Класифікація підходів та сенсорів. Сучасні методи SLAM для БПЛА класифікують за типом основних сенсорів:

- LiDAR SLAM: використовує лазерні сканери (наприклад, алгоритми LOAM, LIO-SAM). Забезпечує високу точність та незалежність від освітлення, але є обчислювально витратним та важким для малих платформ.

- Visual SLAM: використовує камери як основне джерело інформації.

- Монокулярний SLAM: легкий та компактний, але страждає від дрейфу масштабу.

- RGB-D SLAM: ефективний у приміщеннях, але часто не працює на відкритому сонці через IR-засвітку.

Архітектура сучасної системи SLAM. Більшість сучасних систем поділяються на дві основні складові[1]:

- Frontend (візуальна одометрія / трекінг): відповідає за обробку "сирих" даних сенсорів, виділення ознак, співставлення кадрів та попередню оцінку відносного руху між кадрами. Це швидкий процес, що працює в реальному часі.

- Backend (оптимізація): відповідає за мінімізацію накопиченої помилки (дрейфу) та побудову глобально узгодженої карти. Тут використовуються методи оптимізації графів факторів (Factor Graphs) або Bundle Adjustment (BA).

- Loop Closure (Замикання циклу): Окремий модуль, що розпізнає раніше відвідані місця. Це дозволяє скоригувати накопичену помилку траєкторії, "замикаючи" граф позувань.

Схема на рисунку 1.1, запропонована в роботі [2], демонструє основні функціональні блоки: сенсорний рівень (Sensors), візуальний фронтенд (Frontend), оптимізаційний бекенд (Backend), модуль замикавання циклів (Loop closure) та

підсистему картування (Mapping), а також ключові алгоритми, що реалізують ці етапи у хронологічному порядку.

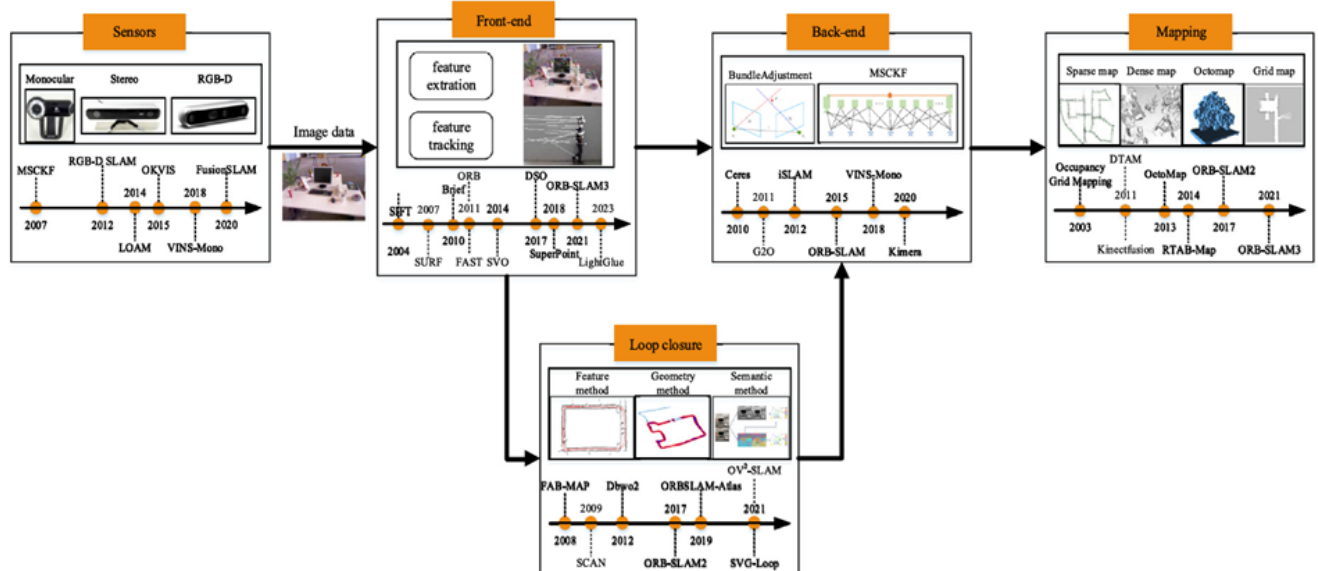


Рис. 1.1. Типова архітектура та хронологія розвитку систем візуального SLAM.

В роботі [3] представлено схему еволюції методів SLAM, наведена на рис. 1.2, де зображено перехід від класичних фільтраційних підходів (фільтр Калмана, EKF) до методів візуальної навігації (vSLAM) та сучасних візуально-інерціальних систем (viSLAM), що забезпечують високу робастність оцінки стану.

Для маневрених БПЛА чистий Visual SLAM часто виявляється недостатньо надійним через можливість втрати візуального контакту (текстури) або різких рухів. Тому стандартом де-факто стає Візуально-інерціальна одометрія (VIO), яка комбінує камеру та IMU.

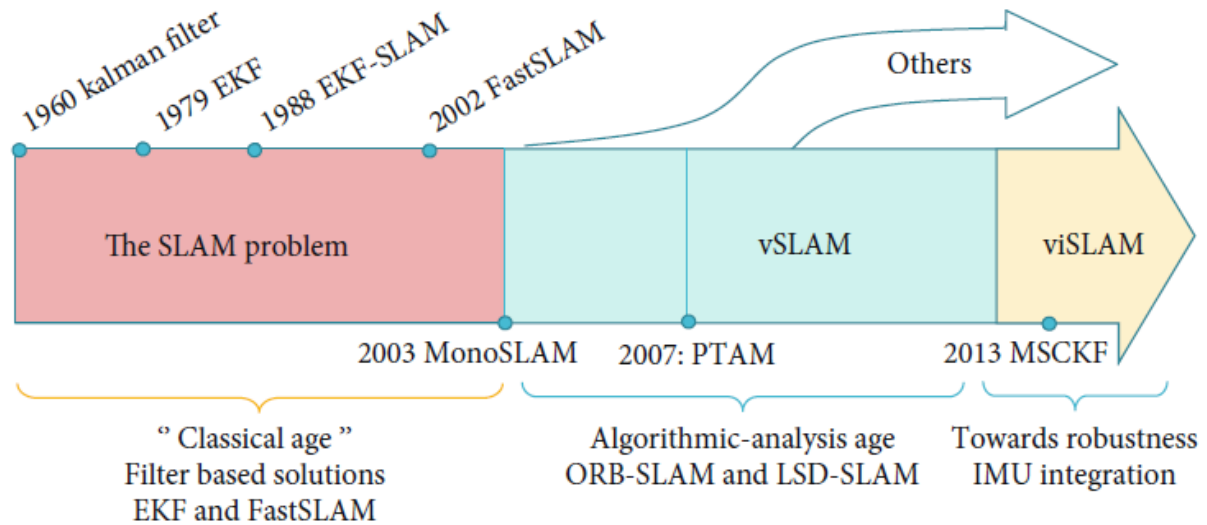


Рис. 1.2. Еволюція методів одночасної локалізації та картування

Саме аналіз методів VIO, їх різновидів та способів оптимізації є предметом подальшого розгляду, оскільки вони забезпечують найкращий баланс між точністю, вагою та обчислювальною складністю для автономних квадрокоптерів.

1.2. Еволюція методів SLAM

Еволюція методів SLAM пройшла шлях від класичних фільтраційних підходів до складних систем нелінійної оптимізації. Ранні рішення базувалися переважно на базі розширеного фільтра Калмана (EKF), де поточні координати апарата та просторові орієнтири об'єднувалися в загальний вектор стану. Однак такі EKF-системи мали критичну технічну вразливість, квадратичну залежність обчислювальної складності від кількості виявлених точок на карті. Цей фактор робив їхнє масштабування або тривале використання на мобільних роботах практично неможливим [4][5].

Розвиток методів візуального SLAM (Visual SLAM) дозволив розв'язувати задачу навігації, використовуючи лише візуальні ознаки. Точний трекінг у таких системах забезпечувався шляхом безперервного вилучення та зіставлення складних візуальних дескрипторів. Фундаментальну зміну в цій сфері здійснили Г. Клейн та

Д. Мюррей, презентувавши фреймворк PTAM (Parallel Tracking and Mapping) [6]. Ця система стала першою успішною реалізацією ідеї розділення задачі SLAM на два паралельні обчислювальні потоки: відстеження (tracking) положення камери в реальному часі та побудову карти (mapping) у фоновому режимі. Такий поділ дозволив застосувати важкий метод оптимізації Bundle Adjustment до карти, не гальмуючи при цьому роботу самої локалізації. Спершу PTAM створювався для індустрії доповненої реальності в обмежених просторах, проте його базова ідея виявилася ідеальною для БПЛА. Відмова від обробки кожного окремого зображення на користь лише вибраних ключових кадрів дозволила підтримувати високу точність навігації за мінімальних витрат процесорного часу порівняно з тогочасними EKF-підходами.

Формування сучасного візуального SLAM значною мірою відбулося завдяки системі ORB-SLAM від Mur-Artal R. та його співавторів [7]. Особливістю цього підходу є використання бінарних ознак ORB (Oriented FAST and Rotated BRIEF). Вони поєднують у собі стійкість до змін масштабу та обертань із високою обчислювальною швидкістю. Роботу системи поділили на три паралельні потоки: відстеження (tracking), побудова локальної карти (mapping) та замикання циклу (loop closing). Окремо варто виділити процес замикання циклу: під час повторного проходження відомої локації система ініціює глобальну оптимізацію графа, повністю списуючи накопичений навігаційний дрейф (drift).

Альтернативним вектором розвитку стали прямі методи (direct methods), серед яких особливе місце посідає LSD-SLAM (Large-Scale Direct Monocular SLAM) [8]. Замість пошуку конкретних дескрипторів, він оперує змінами яскравості пікселів уздовж високоградієнтних зон зображення. Результатом такої роботи є побудова напівщільної (semi-dense) карти середовища. Порівняно з розрідженою точковою хмарию, вона суттєво спрощує планування безпечних маршрутів для безпілотників. LSD-SLAM демонструє високу надійність там, де ключові точки зазвичай губляться: у слабо текстурованих приміщеннях або під час динамічного розмиття

кадру. Завдяки поєднанню прямого трекінгу з оптимізацією графа поз (pose-graph optimization), алгоритм успішно генерує масштабні точні карти. Додатковим інженерним плюсом є те, що всі розрахунки ефективно виконуються центральним процесором, без залучення графічних прискорювачів (GPU).

Окрему нішу в дослідженнях займає алгоритм SVO (Semi-direct Visual Odometry) [9]. Автори запропонували гібридний підхід, який поєднує точність прямих методів при відстеженні переміщення пікселів із ефективністю дескрипторних методів при побудові карти. Завдяки відсутності необхідності виділення ознак у кожному кадрі, SVO здатний працювати з надзвичайно високою частотою (понад 100 кадрів на секунду) на бортових комп'ютерах БПЛА. Це забезпечує мінімальну затримку у контурі керування, що є вирішальним фактором для стабілізації апарата під час виконання агресивних маневрів.

Проте для підвищення надійності в умовах динамічних маневрів БПЛА критично важливим є перехід до візуально-інерціальних систем (VI-SLAM). Інтеграція даних інерціальних вимірювань (кутової швидкості та прискорення) створює додаткові обмеження, що дозволяє зменшити кількість необхідних візуальних ознак для підтримання стабільної роботи системи. У роботах [10, 11] ранні методи SLAM на основі розширеного фільтра Калмана (EKF) було вдосконалено шляхом виключення 3D-орієнтирів із вектора стану. Завдяки цьому з'явилася можливість моделювати у стані декілька кадрів одночасно. Проте, як зазвичай властиво алгоритмам EKF, вони зберігають лише останній стан, а отже, є чутливими до похибок вимірювань, і їх важко відновити у разі втрати стабільності відстеження. У працях [12, 13, 14, 15] використовується часове ковзне вікно для вибору останніх візуальних та інерціальних вимірювань з метою оптимізації цільових функцій SLAM. Автори демонструють, що в багатьох випадках VI-SLAM на основі ковзного вікна є більш надійним та точним, ніж методи, засновані на фільтрації.

Важливим кроком у забезпеченні стійкості навігації БПЛА стала поява системи VINS-Mono [14]. У цій роботі реалізовано метод щільного поєднання даних монокулярної камери та інерціального вимірювального модуля (IMU). Ключовим внеском автора є використання процедури попередньої інтеграції IMU (pre-integration) у поєднанні з надійним алгоритмом ініціалізації стану.

Слід також відзначити алгоритм DM-VIO (Delayed Marginalization Visual-Inertial Odometry) [16]. Ключовою інновацією цієї системи є стратегія відкладеної маргіналізації, яка дозволяє системі повторно лінеаризувати минулі стани камери після того, як стають відомими нові оцінки параметрів інерціального модуля. Це ефективно розв'язує проблему неузгодженості фільтрів.

Ще одним знаковим рішенням є фреймворк BASALT [17], який використовує метод нелінійного відновлення факторів. Методи оптимізації зазвичай є спільними для різних реалізацій SLAM, незважаючи на те, що їх фронтенд частини та архітектури суттєво відрізняються. Відзначимо метод Bundle Adjustment (BA) [18] у візуальному SLAM, який використовує структуру розрідженості функції репроекції та матриці Гессе. У цій роботі термін BA узагальнено для позначення спільної оптимізації візуальних та інерціальних вимірювань у системах VI-SLAM. У працях [19, 20, 21] підвищено ефективність BA для масштабних сценаріїв. У дослідженні [22] показано, що блочний зумовлений метод спряжених градієнтів (PCG) можна застосовувати для обчислення доповнення Шура з метою підвищення продуктивності. Також існують високотехнологічні реалізації BA [23, 24], які зазвичай використовуються в найсучасніших SLAM-системах. Проте всі ці методи мають спільний недолік: їхня обчислювальна складність зростає квадратично відносно кількості камер. Як наслідок, системи SLAM, побудовані на базі таких методів, можуть використовувати лише дуже обмежену кількість нещодавніх вимірювань для оцінювання пози в режимі реального часу.

Сучасний етап розвитку візуальної навігації характеризується активною інтеграцією методів глибокого навчання у фронтенд-системи SLAM для заміни

класичних дескрипторів більш стійкими навченими ознаками. Фундаментальні роботи, такі як SuperPoint [25] та SuperGlue [26], продемонстрували перевагу нейронних мереж у задачах виділення ключових точок та їхнього інтелектуального співставлення за допомогою графових нейромереж, що забезпечує стійкість до різких змін освітлення. Розвиток цієї концепції в алгоритмах LoFTR [27] та LightGlue [28] дозволив реалізувати процес співставлення без детектування явних ознак або з суттєвим підвищенням обчислювальної ефективності.

Особливу увагу привертають підходи прямого оцінювання руху, такі як DeepVO [29] та UnDeerVO [30], що використовують рекурентні та згорткові мережі для прогнозування траєкторії безпосередньо з відеопотоку, мінімізуючи вплив помилок геометричних методів. Системи типу CNN-SLAM [31] та DeepFactors [32] інтегрують нейромережеве оцінювання глибини у традиційні оптимізаційні графі, що дозволяє будувати щільні карти навіть за монокулярних спостережень. Останні досягнення, представлені в роботах DROID-SLAM [33] та TartanVO [34], впроваджують диференційовану ітеративну оптимізацію та універсальні моделі візуальної одометрії, що забезпечують високу надійність навігації БПЛА в умовах розмиття зображень та складних текстур.

1.3. Аналіз методів візуально-інерціальної одометрії

Візуально-інерціальна одометрія являє собою метод інкрементальної оцінки положення та орієнтації рухомого об'єкта шляхом комплексування даних з двох різнорідних джерел: відеокамери та інерціального вимірювального модуля. Основною метою VIO є обчислення траєкторії руху апарата в шести ступенях свободи (6-DoF) в умовах відсутності зовнішніх навігаційних сигналів, таких як GNSS. На відміну від чистої візуальної одометрії (VO), яка покладається виключно на відстеження текстурних ознак, VIO інтегрує високочастотні дані акселерометра та гіроскопа, що дозволяє системі зберігати працездатність під час різких маневрів,

за умов розмиття зображення або при тимчасовій відсутності візуального контакту з середовищем [1].

Ключовою перевагою такого підходу є комплементарність сенсорів: камера забезпечує високу точність позиціонування на тривалих проміжках часу, компенсуючи дрейф інерціальних датчиків, тоді як IMU надає миттєву інформацію про динаміку руху та дозволяє відновити метричний масштаб сцени. Останнє є критично важливим для монокулярних систем, оскільки одна камера без додаткової інформації не здатна розрізнити розмір об'єктів та пройдену дистанцію.

Сучасні алгоритми VIO реалізують щільно-зв'язану архітектуру на основі нелінійної оптимізації або фільтрації (наприклад, EKF або Factor Graph). Такий підхід дозволяє одночасно оцінювати не лише траєкторію, але й зміщення акселерометра та гіроскопа, а також здійснювати зовнішню калібровку між камерою та IMU в режимі реального часу. Це робить VIO стандартом де-факто для навігації малих безпілотних літальних апаратів у складних умовах, таких як закриті приміщення, тунелі або міські каньйони. На Рис. 1.3. зображено класифікацію методів візуальної одометрії, яка представлена у роботі [2].

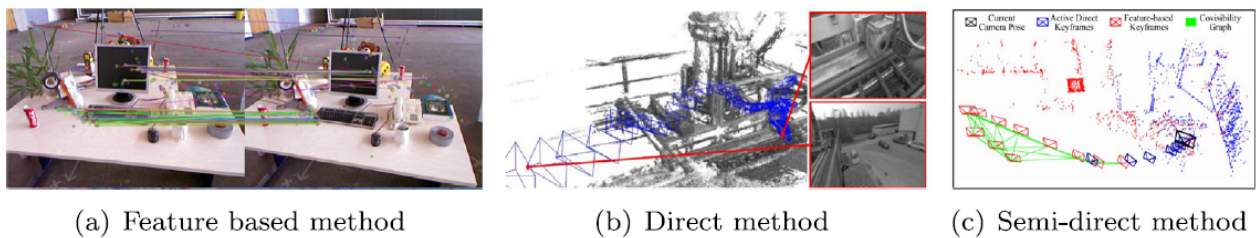


Рис. 1.3. Класифікація методів візуальної одометрії за типом обробки візуальної інформації: а) метод на основі ознак (Feature-based) - мінімізує геометричну помилку репроекції виділених ключових точок; б) прямий метод (Direct) - мінімізує фотометричну похибку, використовуючи інтенсивність усіх пікселів зображення; в) напівпрямий метод (Semi-direct) - комбінує відстеження розріджених ознак із прямим вирівнюванням кадрів для оптимізації швидкодії.

1.3.1. Аналіз методів VIO на основі розріджених ознак та їх обмеження

Серед різноманіття підходів до VIO, методи на основі розріджених ознак (Sparse Feature-based або Indirect methods) займають домінуюче положення в сучасних дослідженнях завдяки їхній стабільності та вивченості.

Суть непрямих (indirect) методів полягає в абстрагуванні від піксельної інтенсивності всього зображення. Замість обробки всього кадру, алгоритм виділяє на зображенні набір характерних точок (keypoints) – кутів, країв, контрастних плям, які можна надійно ідентифікувати на наступних кадрах.

Процес обробки зазвичай включає два етапи[1]:

- Frontend: виявлення та опис ознак за допомогою дескрипторів (SIFT, SURF, ORB, FAST/BRIEF) та їх співставлення між послідовними кадрами.
- Backend: мінімізація геометричної помилки репроекції.

Математично задачу оптимізації в таких методах можна представити як мінімізацію суми квадратів помилок репроекції ознак та помилок інерціальних залишків:

$$\min_X \left(\sum_{k \in F} \|r_I(z_I^k, X)\|_{\Sigma_I}^2 + \sum_{(i,j) \in V} \|r_V(z_{i,j}, X)\|_{\Sigma_V}^2 \right) \quad (1.2)$$

де

- X - вектор станів системи (положення, швидкість, зміщення IMU);
- r_I - залишок інерціальних вимірювань (IMU pre-integration);
- r_V - візуальний залишок (різниця між спостережуваною точкою на зображенні та проєкцією 3D-точки з карти).

У літературі виділяють два основні підходи до реалізації Feature-based VIO[1]:

1. Фільтраційні методи: зазвичай базуються на розширеному фільтрі Калмана. Найвідомішим представником є MSCKF (Multi-State Constraint Kalman Filter). Особливістю таких підходів є те, що стан системи оновлюється рекурсивно, старі стани маргіналізуються. Перевагою є висока обчислювальна ефективність ($O(N)$)

та мінімальна затримка. Недоліком - накопичення помилки лінеаризації та неможливість повторної лінеаризації минулих станів.

2. Оптимізаційні методи: використовують нелінійну оптимізацію (Bundle Adjustment) на ковзному вікні (Sliding Window) або на графі факторів. Яскраві представники: VINS-Mono [14], ORB-SLAM3 [7]. Особливістю є одночасна оптимізація всіх станів у вікні. Перевагами є вища точність порівняно з фільтрами та здатність обробляти замикання циклів. Недоліками - вищі вимоги до обчислювальних ресурсів. VINS-Mono та ORB-SLAM3 на сьогодні вважаються еталонними алгоритмами у цій категорії. VINS-Mono пропонує надійну ініціалізацію та щільну інтеграцію з IMU, тоді як ORB-SLAM3 демонструє виняткову здатність до повторного використання карти та роботи в мульти-сесійному режимі.

Обмеження методів на основі розріджених ознак. Попри широке розповсюдження, feature-based методи мають суттєві обмеження, які ускладнюють їх використання в повністю автономних системах БПЛА:

- Залежність від текстури: алгоритми критично залежать від наявності контрастних деталей. У середовищах з низькою текстурою (білі стіни, бетонна підлога, асфальт) детектори не знаходять достатньої кількості точок для відстеження руху, що призводить до зриву трекінгу.

- Чутливість до освітлення та розмиття: детектори ознак (особливо такі як FAST або Harris corners) розроблені для чітких зображень. При швидких маневрах БПЛА виникає розмиття зображення, яке змінює градієнти пікселів. Це призводить до того, що детектори не можуть "впізнати" ті самі точки на наступному кадрі.

- Обчислювальна вартість етапу Frontend: процес виділення та обчислення дескрипторів ознак (особливо складних, як SIFT або ORB) займає значну частину часу CPU (до 40-50% всього циклу обробки). Це обмежує частоту роботи системи, що є критичним для керування швидкісними БПЛА.

- Розрідженість карти: карта, побудована такими методами, складається з хмари розріджених точок. Така карта підходить для локалізації, але непридатна для навігації та уникнення перешкод. БПЛА може легко пролетіти крізь "стіну" на карті, якщо на цій стіні не було виявлено характерних точок. Для планування шляху потрібна додаткова підсистема для побудови щільної карти, що вимагає додаткових ресурсів.

Аналіз показує, що хоча методи VIO на основі розріджених ознак забезпечують високу точність локалізації в ідеальних умовах, їхня надійність знижується в однорідних середовищах, а отримана карта є недостатньою для безпечної автономної навігації. Це створює передумови для прямих методів, які використовують інформацію про інтенсивність усіх пікселів, що буде розглянуто в наступному підпункті.

1.3.2. Прямі та напівпрямі методи для трекингу у VIO

На відміну від методів, заснованих на ознаках, розглянутих у попередньому підпункті, прямі методи [1] являють собою альтернативний підхід до візуальної одометрії, який здобув значну популярність в останні роки. В основі цих методів лежить відмова від абстракції у вигляді дескрипторів та безпосереднє використання інформації про інтенсивність пікселів зображення.

Незважаючи на домінування feature-based підходів, дослідники виділяють три фундаментальні недоліки, які обмежують їх застосування в автономних системах БПЛА:

1. Обчислювальна складність: етапи виділення ключових точок та обчислення дескрипторів є ресурсомісткими. Наприклад, класичний дескриптор SIFT важко обчислювати в реальному часі на CPU, а ORB вимагає близько 20 мс на кадр. При роботі системи SLAM на частоті 30 Гц це означає, що понад 50% часу процесора витрачається лише на попередню обробку ознак.

2. Втрата інформації: зображення містить сотні тисяч пікселів, тоді як feature-based методи використовують лише кілька сотень точок. Це означає, що більша частина корисної інформації про структуру сцени відкидається.

3. Залежність від текстури: у середовищах з низькою текстурою (білі стіни, порожні коридори, бетонні покриття) кількість виявлених кутових точок різко зменшується, що унеможливорює надійний розрахунок руху камери.

Прямі методи вирішують ці проблеми шляхом оцінки руху камери на основі яскравості пікселів, що дозволяє уникнути обчислення дескрипторів та працювати з будь-якими пікселями, що мають градієнт (навіть якщо це не кути, а краї або плавні переходи).

Теоретичним фундаментом прямих методів є концепція оптичного потоку, зокрема метод Лукаса-Канаде (Lucas-Kanade). В основі лежить гіпотеза про постійність яскравості [1]: припускається, що інтенсивність пікселя, який відповідає одній і тій самій точці фізичного об'єкта, не змінюється між послідовними кадрами, незважаючи на його переміщення по площині зображення. Нехай $I(x, y, t)$ - функція яскравості пікселя з координатами (x, y) у момент часу t . Припустимо, що за час dt піксель перемістився на (dx, dy) . Згідно з гіпотезою постійності яскравості:

$$I(x + dx, y + dy, t + dt) = I(x, y, t) \quad (1.3)$$

Застосувавши розклад у ряд Тейлора для лівої частини рівняння і відкинувши члени вищих порядків, отримуємо:

$$I(x + dx, y + dy, t + dt) \approx I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt \quad (1.4)$$

Скоротивши рівняння на $I(x, y, t)$ та розділивши на dt , отримуємо основне рівняння оптичного потоку [1]:

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} = - \frac{\partial I}{\partial t} \quad (1.5)$$

Позначивши швидкість пікселя як $u = \frac{dx}{dt}$, $v = \frac{dy}{dt}$ градієнти зображення як I_x, I_y , а зміну яскравості у часі як I_t , рівняння можна записати у матричній формі [1]:

$$\begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -I_t \quad (1.6)$$

Оскільки рівняння з двома невідомими (u, v) неможливо розв'язати для одного пікселя, прямі методи використовують підхід, аналогічний методу Лукаса-Канаде: припускається, що рух пікселів у межах локального вікна (розміром $w \times w$) є однаковим. Це дозволяє сформувати перевизначену систему лінійних рівнянь [1]:

$$A \begin{bmatrix} u \\ v \end{bmatrix} = -b \quad (1.7)$$

де A - матриця градієнтів пікселів у вікні, а b - вектор часових похідних яскравості. Розв'язок знаходиться методом найменших квадратів:

$$\begin{bmatrix} u \\ v \end{bmatrix}^* = -(A^T A)^{-1} A^T b \quad (1.8)$$

У контексті VIO, замість простого відстеження пікселів (як у чистому оптичному потоці), ця фотометрична помилка мінімізується для оцінки параметрів жорсткого руху камери (обертання та трансляції) у 3D просторі. Залежно від кількості використовуваних пікселів, прямі методи класифікують на:

1. Розріджені (Sparse): використовують лише пікселі з сильним градієнтом.
2. Напівщільні (Semi-Dense): використовують усі пікселі, що належать до країв об'єктів.
3. Щільні (Dense): використовують усі пікселі зображення.

Важливо зазначити, що хоча гіпотеза постійності яскравості часто порушується в реальних умовах (через зміни експозиції камери, тіні, відблиски), прямі методи залишаються потужним інструментом, особливо при інтеграції з даними IMU, що дозволяє компенсувати недоліки візуальної частини.

1.3.3. Обмеження методів VIO для задач з реконструкцією середовища

Попри широке розповсюдження монокулярних систем візуально-інерціальної одометрії завдяки їхній низькій вартості, малій вазі та енергоефективності, їх застосування у реальних інженерних задачах стикається з низкою критичних обмежень. Більшість класичних алгоритмів (як на основі ознак, так і прямих методів) покладаються на припущення про наявність достатньої кількості статичних текстурних ознак у сцені. Порушення цієї умови призводить до нестабільності роботи системи або повної втрати локалізації.

У Таблиці 1.1. представлено порівняльну характеристику основних технологій візуального фронтенду VSLAM, яка була запропонована у роботі [2]. У таблиці наведено відмінності між методами на основі ознак, прямими та напівпрямими методами за критеріями щільності реконструкції сцени (Scene reconstruction) та робастності трекінгу (Tracking robustness).

Таблиця 1.1. Порівняльна характеристика основних технологій візуального фронтенду VSLAM.

Методи	Потрібні особливі точки	Оцінка положення (пози)	Реконструкція сцени	Стійкість відстеження	Сценарії застосування
Методи на основі ознак	Так	Зіставлення ознак	Розріджені / напівщільні карти	Залежить від якості ознак	Сцени, багаті на деталі (ознаки)
Прямі методи	Ні	Пряме зіставлення інтенсивності пікселів	Щільні карти	Можуть давати збої при складних текстурах	Використання повної інформації зображення
Напівпрямі методи	Частково	Комбінація ознак та інтенсивності пікселів	Розріджені / напівщільні карти	Баланс використання ознак та пікселів	Поєднують переваги методів на основі ознак та прямих методів

Можна виділити специфічний клас задач, де застосування класичних методів Mono-VIO є проблематичним або неможливим:

1. Навігація в умовах текстурної однорідності (Textureless Environments)

Класичні детектори ознак (FAST, ORB) не здатні виділити достатню кількість ключових точок на гладких поверхнях. Це робить навігацію вкрай ненадійною у таких середовищах, як:

- Внутрішні поверхні промислових резервуарів та бетонних тунелів;
- Вентиляційні шахти та «чисті кімнати» з однотонними стінами. У таких умовах розріджена карта ознак стає надто бідною для коректної оцінки руху, що призводить до швидкого накопичення похибки дрейфу.

2. Проблема уникнення перешкод (Obstacle Avoidance). Традиційні монокулярні системи SLAM генерують розріджену хмару точок, яка є недостатньою для безпечної навігації.

- Тонкі об'єкти (дроти, арматура, гілки) часто не містять виражених ознак і тому не потрапляють у карту.

- Відсутність щільної інформації про простір не дозволяє будувати карти зайнятості, необхідні для планування шляху. Без використання активних сенсорів БПЛА залишається «сліпим» до геометрії перешкод, які не мають текстури.

3. Робота зі складними поверхнями (Specular & Transparent Surfaces). Алгоритми VIO базуються на припущенні про фотометричну сталість - що одна й та сама точка простору виглядає однаково з різних ракурсів. Це припущення порушується при наявності:

- Дзеркальних поверхонь (скляні фасади, сонячні панелі, мокрі поверхні в колекторах);
- Металевих конструкцій з відблисками. У таких випадках алгоритм помилково інтерпретує відбиття як реальні об'єкти, що призводить до хибної оцінки глибини та траєкторії.

4. Задачі високоточної вертикальної стабілізації. Критичним недоліком монокулярних систем є невизначеність метричного масштабу та залежність від візуального потоку для оцінки висоти. Це створює ризики в наступних сценаріях:

- Польоти над однорідними підстилаючими поверхнями (сніг, пісок, наливні підлоги, вода). За відсутності текстур алгоритми оптичного потоку не можуть зафіксувати зміщення, що призводить до неконтрольованого дрейфу.

- Робота в замкнених об'ємах. У тунелях чи колекторах використання барометрів є неефективним через перепади тиску («ефект поршня»), а GPS-сигнал відсутній. Без надійної візуальної опори БПЛА не може утримувати стабільну висоту, що загрожує зіткненням зі стелею або підлогою.

Таким чином, існує невирішена науково-технічна проблема: відсутність методів, які б поєднували компактність монокулярних систем із здатністю до щільного метричного картування та робастністю LiDAR-систем. Це обґрунтовує необхідність розробки нових гібридних підходів, зокрема із залученням методів глибокого навчання для оцінки геометрії сцени.

1.3.4. Обґрунтування розробки гібридної архітектури VIO

Аналіз обмежень класичних методів VIO виявив ключову проблему: компроміс між надійністю та апаратними вимогами. LiDAR-системи є точними, але важкими для малих БПЛА; монокулярні системи є легкими, але страждають від дрейфу масштабу та втрати трекінгу в бестекстурних зонах.

Для вирішення цієї дилеми в даній дисертаційній роботі пропонується вибір гібридної архітектури, яка поєднує новітні підходи глибокого навчання для оцінки геометрії (MoGe) [35], алгоритми реєстрації хмар точок (ICP) [36] та імовірнісну оптимізацію (Factor Graph). Такий підхід дозволяє реалізувати високоточну геометричну навігацію, використовуючи лише одну камеру та IMU.

Ключовою інновацією пропонованої архітектури є інтеграція нейромережевого модуля глибини MoGe (Monocular Generalist Geometry). На відміну від класичних

CNN для оцінки глибини, моделі класу MoGe є foundation models, навченими на величезних наборах мультимодальних даних. Переваги використання MoGe у контурі VIO [35]:

- Відновлення метричного масштабу: MoGe здатна генерувати щільні карти глибини та карти нормалей з абсолютною метричною точністю (або з високою консистентністю відносного масштабу). Це вирішує фундаментальну проблему монокулярного SLAM – невизначеність масштабу.

- Робастність до домену (Zero-shot Generalization): модель ефективно працює як у приміщеннях (indoor), так і на вулиці (outdoor) без необхідності додаткового донавчання на специфічних датасетах, що критично важливо для універсальності БПЛА.

- Усунення геометричної неоднозначності: окрім глибини, MoGe передбачає нормалі поверхонь, що дозволяє фільтрувати шум та виділяти площинні структури (підлога, стіни) для більш точної навігації.

Отримавши щільну 3D-структуру сцени з 2D-зображення (завдяки MoGe), система отримує можливість застосувати алгоритми, які раніше були доступні лише для LiDAR-систем.

Алгоритм ICP (point-to-plane) використовується для вирівнювання послідовних "псевдо-хмар" точок [36]:

- Принцип дії: мінімізується відстань між точками поточної хмари (згенерованої MoGe) та площинами локальної карти.

- Функція в системі: ICP виступає як стабілізатор. Якщо візуальний трекер помиляється через зміни освітлення, геометричне вирівнювання коригує позу, оскільки геометрія кімнати не змінюється від того, увімкнено світло чи ні.

Для об'єднання різномірних даних - високочастотних, але шумних даних IMU, та низькочастотних, але точних геометричних оцінок MoGe+ICP - використовується оптимізація на графі факторів (Factor Graph Optimization). Ми формуємо

оптимізаційну задачу мінімізації, об'єднавши інерціальний фактор та фактор на основі залишку преінтегрованої пози IMU та пози з ICP.

Вибір гібридної архітектури MoGe + ICP + Factor Graph обґрунтовується необхідністю створення системи навігації, яка поєднує легкість пасивних сенсорів (камер) з надійністю активних сенсорів (LiDAR). Використання MoGe дозволяє подолати головний недолік монокулярних систем - відсутність інформації про глибину, а Factor Graph забезпечує математично строгу інтеграцію геометричних обмежень з динамікою польоту БПЛА.

1.4. Аналіз підходів та методів керування БПЛА

Ефективність функціонування БПЛА типу квадрокоптер безпосередньо залежить від обраної стратегії керування, оскільки такі системи є нелінійними, багатозв'язними та мають 6 ступенів вільності (6-DOF) при лише 4 виконавчих механізмах. Огляд сучасного стану технологій, представлений у роботі [37], підкреслює, що незважаючи на появу складних методів, класичні ПІД-регулятори залишаються базовим стандартом завдяки своїй простоті. Детальний аналіз та проектування таких класичних структур керування, проведений у [38], підтверджує, що вони складають надійну основу для стабілізації апарата, хоча й потребують ретельного налаштування коефіцієнтів. Проте для підвищення їхньої точності активно застосовуються методи оптимізації. Зокрема, у дослідженнях [39] та [40] продемонстровано ефективність використання генетичних алгоритмів (GA) для налаштування параметрів децентралізованих та нелінійних ПІД-контролерів, що дозволяє значно зменшити похибку відстеження траєкторії та витрати енергії.

Наступним кроком в еволюції систем керування стало впровадження методів простору станів та адаптивних алгоритмів. Використання лінійно-квадратичного регулятора (LQR) у поєднанні з фільтром Калмана для оцінки стану [41] забезпечує високу стабільність, тоді як L1-адаптивне керування [42] дозволяє системі швидко

реагувати на зовнішні збурення та зміну параметрів самого апарата. Окремий інтерес становлять гібридні підходи, такі як поєднання прогнозуючого керування моделями (MPC) з методом Backstepping [43], що дозволяє досягти оптимальної стабілізації в умовах динамічних обмежень.

Сучасний вектор досліджень спрямований на інтелектуалізацію керування. Роботи [44] та [45] описують використання нейронних мереж, нечіткої логіки (Fuzzy Logic) та нейроеволюційних алгоритмів для автоматичного налаштування Gain-параметрів. Такі «самонавчання» контролери демонструють здатність адаптуватися до складних умов польоту без необхідності точного математичного моделювання всіх фізичних процесів, що забезпечує безпрецедентну гнучкість при виконанні автономних місій.

Вагомий внесок у розвиток теорії керування та автономної навігації БПЛА зробили українські вчені, чії дослідження охоплюють широкий спектр задач – від синтезу робастних контурів керування до реалізації методів SLAM у GPS-негативному просторі. Зокрема, у працях А.Р. Кравчука та А.Г. Ткачука [46] систематизовано методи автономної навігації (Optical Flow, vSLAM, LiDAR), де наголошено на необхідності створення гібридних адаптивних систем. Розвиваючи цей напрям, В.М. Синєглазов та Р.В. Конюшенко [47] запропонували гібридний метод оптимізації покриття для задач точного землеробства, що поєднує глобальне планування на основі мурашиних алгоритмів із локальною корекцією маршруту. Дослідження В.В. Бурнашева [48] заклали теоретичний базис для стабілізації кутового руху БПЛА в умовах невизначеності. Розвиток цієї тематики продовжується у роботах вітчизняних фахівців, які фокусуються на робастному керуванні та оцінюванні стану (О.А. Сущенко [49], В.Б. Ларін [50]), інтелектуальних методах навігації та стабілізації (Ю.П. Кондратенко [51], О.С. Куценко [52], М.С. Павлюк [53]), підготовці спеціалізованих даних для навчання нейромережевих систем автоматизації польоту (С.Г. Стіренко [54]), а також на вдосконаленні алгоритмів комплексування даних для польотів у складних умовах

(О.В. Збруцький [55], О.І. Романов [56], О.М. Герасимов [57], В.В. Бондарчук [58]). Значний крок у напрямку безпечної навігації в неструктурованих 3D-середовищах зроблено в дослідженнях А. Сазонова, О. Кучкіна та І. Черепанської [59], де запропоновано ентропійно-регуляризований метод планування траєкторій (S3PM) безпосередньо у щільних хмарах точок з урахуванням динаміки перешкод та апаратних обмежень бортових Edge-платформ.

Ефективність автономної навігації БПЛА залежить не лише від точності оцінки власного стану, але й від здатності системи керування точно відпрацьовувати задану траєкторію в умовах зовнішніх збурень.

Структурна схема ієрархічної системи керування БПЛА, яка представлена на Рис. 1.4, запропонована в роботі [60], зображає взаємодію сенсорної підсистеми (Vision, IMU, Sonar) з каскадними регуляторами висоти, позиції та орієнтації, побудованими на основі методу інтегрального зворотного кроку, а також інтеграцію модуля уникнення перешкод у контур керування.

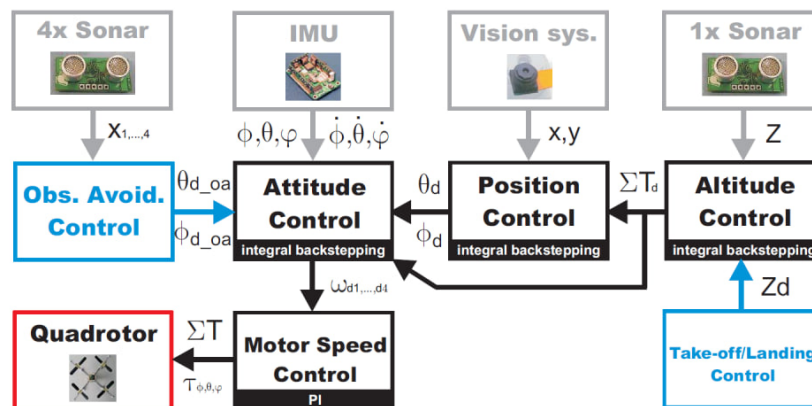


Рис. 1.4. Структурна схема ієрархічної системи керування БПЛА.

Сучасні методи керування БПЛА прийнято класифікувати за принципом їх математичного апарату та здатності компенсувати нелінійності моделі. У рамках даного дослідження виділимо такі основні групи: класичні лінійні регулятори (PID, LQR), методи інтелектуального керування (Fuzzy Logic), а також нелінійні та прогнозуючі стратегії (Geometric Control, MPC).

Класичне каскадне PID-регулювання. Найпоширенішим підходом у промисловості та відкритих платформах (ArduPilot, PX4) є архітектура каскадних PID-регуляторів. Система розбивається на контури з різною частотою оновлення:

- Контур позиції (Position Loop): Зовнішній контур (10-50 Гц), що мінімізує помилку координат (x, y, z) . Виходом є бажані кути нахилу та загальна тяга.
- Контур орієнтації (Attitude Loop): Проміжний контур (100-250 Гц), що відпрацьовує бажані кути Ейлера (roll, pitch, yaw).
- Контур кутової швидкості (Rate Loop): Внутрішній контур (400-1000 Гц), що керує кутовими швидкостями тіла.

$$u(t) = K_p e(t) + K_i \int e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad (1.9)$$

Переваги: Простота реалізації, низькі обчислювальні вимоги.

Недоліки: Оскільки PID є лінійним регулятором, він погано справляється з сильно нелінійною динамікою БПЛА при агресивних маневрах. Кожна вісь зазвичай регулюється незалежно, ігноруючи перехресні зв'язки.

Лінійно-квадратичний регулятор (LQR). Для подолання обмежень PID-регуляторів, які розглядають кожен канал керування ізольовано, застосовується лінійно-квадратичний регулятор (Linear Quadratic Regulator, LQR). Цей метод належить до класу оптимального керування і працює у просторі станів.

Модель динаміки БПЛА лінеаризується навколо робочої точки (зазвичай – режиму висіння) і подається у вигляді:

$$\dot{x} = Ax + Bu \quad (1.10)$$

де x – вектор станів, u – вектор керування.

Завдання LQR полягає у знаходженні такого закону керування $u = -Kx$, який мінімізує квадратичний функціонал якості J на нескінченному горизонті:

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (1.11)$$

де:

- Q – матриця ваг, що штрафує відхилення від цільового стану (точність);
- R - матриця ваг, що штрафує витрати енергії (величину керуючих сигналів).

Переваги LQR для БПЛА:

1. Багатовимірність: на відміну від PID, LQR автоматично враховує перехресні зв'язки між осями (наприклад, вплив зміни курсу на крен), що є критичним для стабільності мультироторних систем.

2. Оптимальність: алгоритм математично гарантує знаходження найкращих коефіцієнтів підсилення K для заданого компромісу між швидкістю реакції та витратою енергії.

Недоліки: LQR вимагає точної лінійної моделі. При значних відхиленнях від точки лінеаризації (наприклад, при агресивному маневруванні) якість керування різко погіршується.

Нечітке логічне керування (Fuzzy Logic Control). В умовах, коли точна математична модель БПЛА невідома або змінюється під час польоту (зміна маси корисного навантаження, вітрові пориви), ефективним є застосування методів інтелектуального керування, зокрема нечітких регуляторів.

Найпоширенішою архітектурою є Fuzzy-PD, який поєднує структуру класичного PD-регулятора з нечіткою логікою. Вхідними даними для нього є помилка позиції $e(t)$ та швидкість зміни помилки $\dot{e}(t)$.

Процес керування складається з трьох етапів:

1. Фазифікація: числові значення помилок перетворюються у лінгвістичні змінні (наприклад, "Негативна велика", "Нульова", "Позитивна мала") за допомогою функцій належності.

2. База правил: використовується набір евристичних правил типу "ЯКЩО помилка велика позитивна І швидкість негативна мала, ТО тяга – позитивна середня". Це дозволяє формалізувати досвід оператора.

3. Дефазифікація: отриманий нечіткий висновок перетворюється назад у чіткий числовий сигнал керування двигунами.

Переваги Fuzzy-PD:

- Робастність: Система стійка до шумів сенсорів та невизначеності параметрів моделі (model-free approach).
- Адаптивність: Дозволяє реалізувати нелінійну залежність коефіцієнтів підсилення (Gain Scheduling), наприклад, роблячи реакцію м'якою при малих помилках і агресивною при великих.

Геометричне керування на групі SE(3). Для виконання складних маневрів (наприклад, перевороти, політ з великими кутами нахилу) лінеаризація навколо стану висіння є неприпустимою. Геометричні методи керування розглядають динаміку БПЛА безпосередньо на многовиді групи обертань SO(3) або групи евклідових рухів SE(3), уникаючи сингулярностей кутів Ейлера (проблема "gimbal lock"). Помилка орієнтації в таких методах визначається через матриці обертання R :

$$e_R = \frac{1}{2}(R_d^T R - R^T R_d)^\vee \quad (1.12)$$

де $(\cdot)^\vee$ оператор, що перетворює кососиметричну матрицю у вектор.

Цей підхід, популяризований роботами Lee et al., дозволяє БПЛА зберігати стабільність при будь-якій орієнтації, що є критичним для відновлення після збоїв.

Керування з прогнозуванням моделі (Model Predictive Control - MPC). MPC є "золотим стандартом" для сучасних дослідницьких платформ, особливо тих, що використовують VIO. На відміну від реактивних PID, MPC є проактивним методом. Принцип роботи: На кожному кроці керування алгоритм розв'язує задачу оптимізації на скінченному горизонті прогнозування N . Метою є знаходження такої послідовності керуючих впливів $U = (u_k, u_{k+1}, \dots, u_{k+N-1})$, яка мінімізує функцію вартості J :

$$J = \sum_{k=0}^{N-1} \left(\|x_k - x_{ref}\|_Q^2 + \|u_k\|_R^2 \right) + \|x_N - x_{ref}\|_P^2 \quad (1.13)$$

при дотриманні обмежень:

- $x_{k+1} = f(x_k, u_k)$ (динамічна модель БПЛА);

- $u_{min} \leq u_k \leq u_{max}$ (обмеження тяги двигунів);
- $v_{min} \leq v_k \leq v_{max}$ (обмеження швидкості польоту).

Переваги для VIO-навігації:

- Обробка затримок: MPC природним чином враховує затримки в ланцюгу керування, що є критичним при використанні "важких" алгоритмів комп'ютерного зору.

- Дотримання обмежень: Гарантує, що БПЛА не вийде за межі фізичних можливостей приводів або сенсорів (наприклад, не перевищить кутову швидкість, при якій виникає "motion blur", що "осліплює" камеру).

В роботі [61] наведено порівняльну характеристика методів автоматичного керування БПЛА, що представлена у Таблиці 1.2.

Таблиця 1.2. Порівняльна характеристика методів автоматичного керування БПЛА за критеріями робастності, точності та складності реалізації.

Algorithm	Characteristic												
	Robust	Adaptive	Optimal	Intelligent	Tracking ability	Fast convergence/response	Precision	Simplicity	Disturbance rejection	Unmodeled parameter handling	Manual tuning	(Signal) noise	Chattering/energy loss
PID	1	0	0	0	1	1	1	2	0	0	2	2	0
Intelligent PID	1	0	0	2	1	1	1	1	0	0	0	1	0
LQR	0	2	1	0	1	1	0	1	1	0	1	1	0
LQG	0	2	2	0	1	1	0	0	2	0	1	0	0
L_1	0	2	2	0	1	2	2	0	1	0	0	0	0
H_1	2	1	2	0	2	0	1	0	1	1	0	0	0
SMC	1	2	1	0	2	2	2	1	2	1	0	0	2

FBL	1	1	0	0	2	2	2	1	1	1	0	1	0
Backstepping	0	2	0	0	2	0	1	0	2	1	0	0	0
Fuzzy logic	1	1	1	2	1	1	1	1	1	0	1	0	0

Незважаючи на різноманіття алгоритмів керування, наведених у порівняльній таблиці, більшість із них мають критичні обмеження при застосуванні на малих БПЛА, що виконують автономні місії в обмеженому просторі.

1. *Обмеженість лінійних регуляторів (PID, LQR).* Класичні методи, такі як ПІД-регулятори (PID) або лінійно-квадратичні регулятори (LQR), ефективні лише в «малому» діапазоні навколо точки зависання.

- Проблема: квадрокоптер є суттєво нелінійною системою. При активному маневруванні (наприклад, різкому ухиленні від перешкоди) або при сильних поривах вітру в тунелі, лінійні регулятори не здатні компенсувати аеродинамічні ефекти, що призводить до втрати стійкості.
- Проблема: LQR вимагає ідеально точної математичної моделі. Будь-яка зміна ваги БПЛА (наприклад, встановлення іншої камери) вимагає повного перерахунку коефіцієнтів.

2. *Проблема «чатерінгу» у робастних системах (SMC).* Метод змінного режиму (Sliding Mode Control - SMC) забезпечує високу стійкість до збурень, але має суттєвий недолік - явище високочастотних коливань керуючого сигналу («chattering»).

- Наслідок: ці вібрації передаються на корпус БПЛА, що викликає розмиття зображення на камері. Для системи VIO це критично, оскільки розмиття призводить до зриву трекінгу і втрати локалізації.

3. *Чутливість до шумів вимірювань (Backstepping).* Класичний метод зворотного кроку (Backstepping) є потужним інструментом для нелінійних систем, проте у своїй стандартній формі він залежить від якості вхідних даних.

- Оскільки координати та швидкість надходять від візуальної одометрії та нейромережі MoGe, вони містять стохастичний шум та затримки (latency ~ 47

мс). Класичний Backstepping може сприйняти цей шум як реальний рух і почати його «компенсувати», розхитуючи апарат.

Таким чином, виходячи з аналізу недоліків для забезпечення надійної роботи розробленої системи навігації необхідно модифікувати алгоритм керування для вирішення трьох ключових задач:

- *Компенсація статичних похибок та зовнішніх збурень (Integral Action).* Введення інтегральної складової у закон керування дозволить системі автоматично адаптуватися до немодельованих факторів – наприклад, зміщення центру мас, розряду батареї (зміна тяги моторів) або повітряних потоків у вентиляційних шахтах, не вимагаючи ідеальної моделі.

- *Фільтрація шумів візуальної навігації.* Система керування повинна бути толерантною до короткочасних викидів у даних глибини від MoGe. Необхідно забезпечити плавність керуючих впливів, щоб уникнути різких ривків, які можуть дестабілізувати VIO-алгоритм.

- *Забезпечення глобальної асимптотичної стійкості.* На відміну від локальних лінійних методів, синтезований регулятор повинен гарантувати повернення БПЛА на задану траєкторію з будь-якого початкового положення, що є критичним для безпеки при виникненні аварійних ситуацій (наприклад, після зіткнення з гілкою чи стіною).

Висновки до розділу 1

У першому розділі дисертаційної роботи проведено комплексний аналіз сучасного стану проблеми автономної навігації та керування БПЛА в умовах відсутності сигналів глобальних супутникових систем.

За результатами проведеного дослідження можна зробити наступні висновки:

1. *Актуальність проблеми.* Встановлено, що забезпечення автономності малих БПЛА в умовах «GNSS-denied» (приміщення, тунелі, зони дії РЕБ) вимагає застосування методів візуально-інерціальної одометрії, які є оптимальним

рішенням з точки зору масогабаритних та енергетичних обмежень (SWaP) порівняно з LiDAR-системами.

2. *Обмеження існуючих методів навігації.* Аналіз класичних алгоритмів VIO на основі розріджених ознак (VINS-Mono, ORB-SLAM3) виявив їхню суттєву вразливість до «feature starvation» – втрати трекінгу в середовищах з низькою текстурною насиченістю (однотонні стіни, бетонні покриття). Крім того, розріджені карти, що генеруються такими методами, є недостатніми для безпечної навігації та уникнення перешкод. Прямі методи частково вирішують проблему текстур, але є чутливими до змін освітлення та вимагають значних обчислювальних ресурсів.

3. *Обґрунтування гібридної архітектури.* На основі аналізу новітніх досягнень у галузі Deep Learning зроблено висновок про доцільність розробки гібридної системи навігації. Обґрунтовано вибір архітектури, що поєднує:

- Нейромережу MoGe для генерації щільних карт глибини та відновлення метричного масштабу з одного зображення.
- Алгоритм ICP для прямої геометричної реєстрації, що забезпечує робастність у безтекстурних середовищах.
- Оптимізацію на основі фактор-графа для ймовірнісного об'єднання геометричних обмежень з високочастотними даними інерціальної навігації.

4. *Аналіз систем керування.* Дослідження методів автоматичного керування БПЛА показало, що класичні лінійні ПІД-регулятори мають обмежену ефективність при агресивних маневрах та дії зовнішніх збурень. Для забезпечення точного відпрацювання траєкторії на основі даних VIO визначено перспективність застосування методів, що враховують багатовимірність та нелінійність об'єкта, зокрема лінійно-квадратичного регулятора (LQR) для енергоефективного керування та нечітких регуляторів для роботи в умовах невизначеності параметрів моделі.

РОЗДІЛ 2. СИСТЕМА МОНОКУЛЯРНОЇ ВІЗУАЛЬНО-ІНЕРЦІАЛЬНОЇ ОДОМЕТРІЇ НА ОСНОВІ НЕЙРОННОЇ МЕРЕЖІ MOGE

У цьому розділі представлено структурну організацію та принципи функціонування системи щільної візуально-інерціальної одометрії з використанням нейронної мережі MoGe (MoGe-VIO) та розглядається її Frontend.

2.1. Архітектура системи

Архітектура системи спроектована за модульним принципом з метою подолання обмежень традиційних методів, що базуються на розріджених ознаках, зокрема, в умовах низької текстурної насиченості середовища. Розглянемо основні її модулі.

2.1.1. Концептуальна модель та потік даних

Запропонована система реалізує підхід «жорстко пов'язаного» комплексування різнорідних сенсорних даних: відеопотоку монокулярної RGB-камери та вимірювань інерціального вимірювального модуля (IMU). На відміну від класичних схем, де візуальний фронтенд виділяє ключові точки, у даній архітектурі застосовано геометрично-орієнтовану нейронну мережу MoGe для генерації щільної 3D-структури сцени. Архітектура системи наведена на рис. 2.1 [62].

Узагальнена схема функціонування системи складається з трьох послідовно з'єднаних обчислювальних блоків:

- Геометричний фронтенд (Geometric Frontend): відповідає за перетворення 2D-зображень у метричні 3D-хмари точок з використанням нейромережевого інференсу.
- Інерціальний фронтенд (Inertial Frontend): здійснює попередню обробку та преінтеграцію високочастотних даних акселерометра та гіроскопа.

- Бекенд оптимізації (Optimization Backend): виконує ймовірнісну оцінку стану БПЛА шляхом мінімізації глобальної функції похибки на базі фактор-графу.

Потік даних організовано таким чином, щоб забезпечити роботу в режимі реального часу.

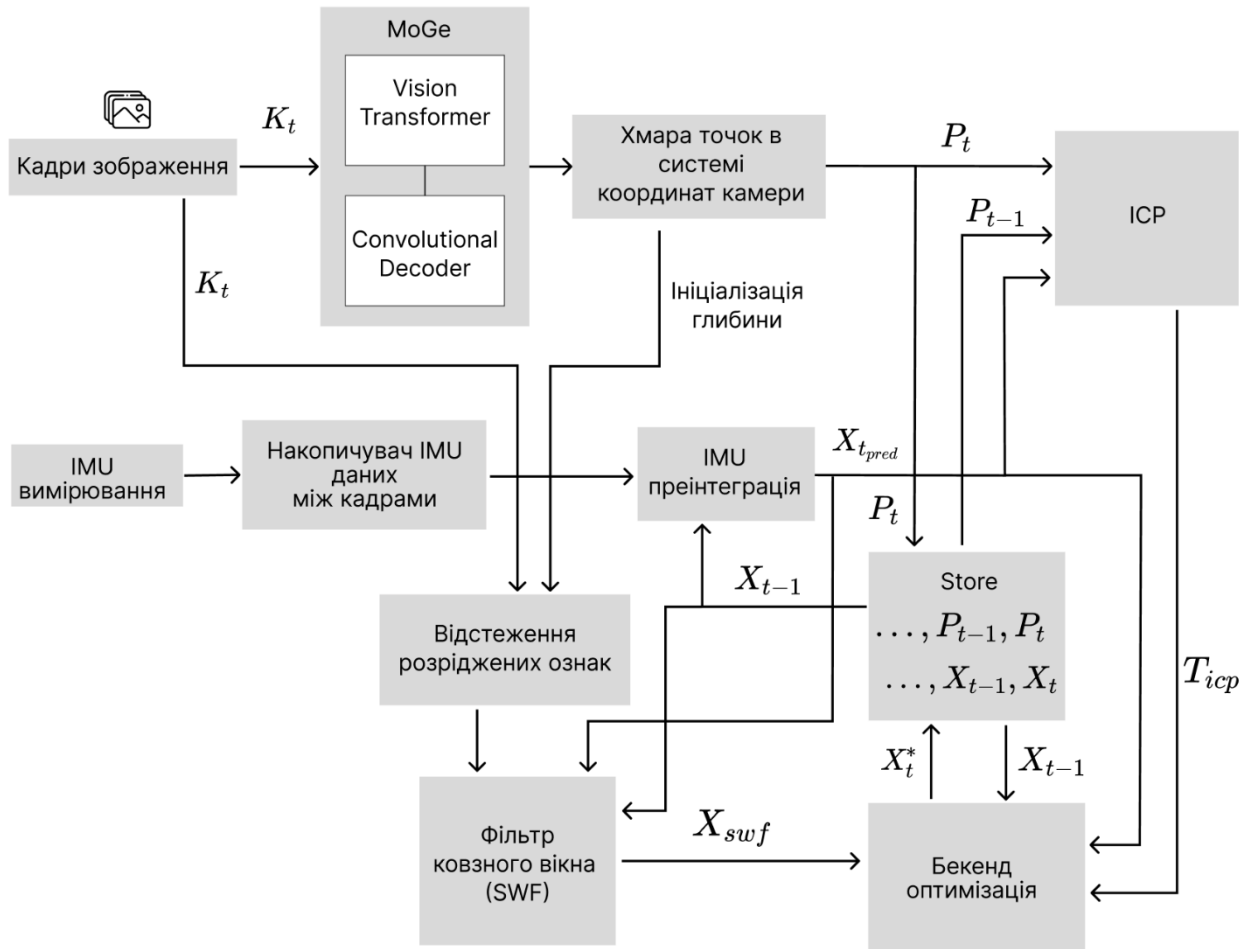


Рис. 2.1. Архітектура системи: K_t – поточний кадр зображення (в момент часу t);

P_t та P_{t-1} – хмари точок з поточного та попереднього кадрів відповідно;

X_t та X_{t-1} – стани системи поточного попереднього кадрів відповідно;

$X_{t_{pred}}$ – преінтегрований стан системи з даних IMU; T_{icp} – відносна поза між двома кадрами, обчислена за допомогою ICP; X_t^* – стан системи після оптимізації;

X_{swf} – відфільтроване початкове наближення стану нового фрейму

Дані IMU використовуються для високочастотного передбачення стану (state propagation), що слугує початковим наближенням для візуальної реєстрації, тоді як візуальні дані коригують дрейф інерціальної системи.

Для математичної формалізації задачі визначено наступні системи координат:

- Світова система координат $\{W\}$ (World Frame): інерціальна система відліку, початок якої ($p_W = [0, 0, 0]^T$) збігається з початковою позицією об'єкту при старті.
- Система координат тіла $\{B\}$ (Body Frame): зв'язана з центром мас інерціального вимірювального модуля. Оцінка траєкторії виконується саме для цієї системи.
- Система координат камери $\{C\}$ (Camera Frame): оптичний центр камери.
- Трансформація $\{B\} \rightarrow \{C\}$: визначається матрицею зовнішніх параметрів $T_{CB} \in SE(3)$, яка складається з матриці повороту R_{CB} та вектора зсуву p_{CB} . Ця трансформація вважається відомою (отриманою в результаті попереднього калібрування) і незмінною в часі.

2.1.2. Функціональні модулі

Модуль нейромережевої оцінки геометрії (MoGe inference).

В якості ядра візуальної підсистеми використано архітектуру Microsoft MoGe (Monocular Geometry) [35]. На вхід модуля подається RGB-зображення I_k розмірністю $H \times W$. Ключовою особливістю даного модуля є те, що він не просто генерує карту глибини, а виконує пряму регресію карти координат точок у просторі камери:

$$P_{raw} = MoGe(I_k), \quad P_{raw} \in \mathbb{R}^{H \times W \times 3} \quad (2.1)$$

Тобто ми не залежимо від калібрування камери. Бо в традиційному випадку, нейромережі видають тензор розміром $H \times W \times 1$. Кожен піксель містить одне число – глибину Z (відстань від площини камери). Щоб отримати 3D-хмару точок, потрібно виконати математичну операцію зворотної проєкції, використовуючи модель камери Pinhole:

$$\begin{aligned} X &= \frac{(u - c_x) \cdot Z}{f_x}, \\ Y &= \frac{(v - c_y) \cdot Z}{f_y}, \end{aligned} \quad (2.2)$$

де:

- u, v - координати пікселя на зображенні.
- f_x, f_y, c_x, c_y - це внутрішні параметри камери (матриця K).

В такому випадку, якщо калібрування камери K неточне (наприклад, f_x відрізняється на 5%), то вся 3D-хмара буде викривлена. Стіни стануть не перпендикулярними, кути попливуть. Це сильно впливає на точність ICP.

MoGe видає тензор розміром $H \times W \times 3$. Кожен піксель одразу містить три числа (X, Y, Z) . Це означає, що неймережа всередині себе "вивчила", як виглядає 3D-простір, і одразу видає готову структуру. Модель також генерує маску впевненості для фільтрації ненадійних пікселів (небо, дзеркальні поверхні).

Модуль підготовки хмар точок (Point Cloud Preprocessing).

Оскільки "сира" хмара точок P_{raw} містить надлишкову кількість точок (понад 300 тис. для роздільної здатності 640x480), що є критичним для обчислювальної складності, застосовується процедура розрідження [36]:

1. Voxel Grid Filtering: простір квантується на вокселі розміром δ (у даній роботі $\delta = 0.05\text{м}$). Всі точки всередині вокселя замінюються їх центроїдом.
2. Estimation of Normals: для кожної точки розрідженої хмари P_k обчислюється нормаль n_i шляхом аналізу власних чисел коваріаційної матриці сусідніх точок. Наявність нормалей є необхідною умовою для використання Point-to-Plane.

Модуль інерціальної преінтеграції (IMU Pre-integration).

Цей модуль акумулює вимірювання акселерометра $\tilde{a}$ та гіроскопа $\tilde{\omega}$ між двома ключовими кадрами k та $k + 1$. Замість повторного інтегрування рівнянь руху при

кожній зміні оцінки стану в оптимізаторі, використовується метод преінтеграції на багатовиді. Результатом роботи модуля є вектор відносного переміщення, який є незалежним від початкової позиції та швидкості, але залежить від змінних зміщення акселерометра b_a та гіроскопа b_g .

Модуль трекінгу та реєстрації (Visual Odometry Tracking).

Задачею цього модуля є обчислення відносної трансформації $T_{k,k+1}^{icp}$ між двома послідовними хмарами точок. Використовується алгоритм Point-to-Plane ICP (Iterative Closest Point), який мінімізує відстань від точок вихідної хмари до площин цільової хмари. Для прискорення збіжності та уникнення локальних мінімумів, як початкове наближення для ICP використовується передбачення від модуля інерціальної преінтеграції.

Модуль відстеження розріджених ознак та первинної фільтрації (Sparse Tracking & SWF).

Цей модуль виконує роль високочастотного візуального фронтенду та кінематичного предиктору. З вхідного зображення виділяються розріджені ключові точки, які відстежуються між послідовними кадрами за допомогою алгоритму оптичного потоку Лукаса-Канаде (KLT). Інновацією даного конвеєра є метод ініціалізації: замість класичної багатовидової тріангуляції (яка страждає від проблеми масштабу в монокулярних системах), абсолютна метрична глибина кожної ознаки вилучається безпосередньо з вхідного 3D-тензора, згенерованого мережею MoGe. Отримані візуальні треки спільно з даними інерціальної преінтеграції обробляються у фільтрі ковзного вікна (Sliding Window Filter, SWF) [63]. SWF здійснює швидку локальну оптимізацію станів X_k у межах вікна фіксованого розміру, розраховуючи точні значення лінійної швидкості, локальної пози та поточних зміщень IMU. Завданням цього модуля є формування прецизійного початкового наближення для фінального графа оптимізації.

Модуль оптимізації на основі фактор-графа.

Інтеграція всіх вимірювань відбувається в рамках нелінійної оптимізації. Система використовує структуру фактор-графа, де вузлами є стани системи X_k у моменти часу отримання зображень, а ребрами – ймовірнісні обмеження (фактори).

Вектор стану X_k включає [62]:

$$X_k = [R_k, p_k, v_k, b_{ak}, b_{gk}] \quad (2.3)$$

де R_k, p_k - орієнтація та позиція, v_k - лінійна швидкість, b_{ak}, b_{gk} - зміщення IMU.

Граф містить три типи факторів:

- IMU Factor: Обмеження, отримані від преінтеграції, що пов'язують послідовні стани динамічно.
- Dense Visual Factor: Обмеження на відносний рух, отримані від ICP. Вага цього фактора динамічно адаптується залежно від якості реєстрації.
- Marginalization Factor: Для обмеження обчислювальної складності застосовується стратегія ковзного вікна (Sliding Window). Старі стани, що виходять за межі вікна, маргіналізуються (видаляються) з графа, проте накопичена в них інформація зберігається у вигляді апіорного фактора, під'єданого до найстарішого активного вузла.

2.2. Модуль обробки даних з камери

Основним завданням візуального модулю (фронтенду) у розробленій системі є перетворення вхідного відеопотоку монокулярної RGB-камери у щільну геометричну репрезентацію сцени. На відміну від класичних методів, що покладаються на розріджені кутові ознаки, та попередніх поколінь нейромереж, що оцінюють відносну глибину, у даній роботі застосовано архітектуру MoGe (Monocular Geometry).

2.2.1. Архітектура нейронної мережі MoGe для генерації щільних карт глибини

Архітектура MoGe [35] являє собою сучасну нейронну мережу глибокого навчання, спроектовану для вирішення задачі оцінки геометрії за одним зображенням ("single-view geometry estimation"). Ключовою відмінністю даного підходу від аналогів (таких як MiDaS або Depth Anything) є відмова від прогнозування однієї лише карти глибини D на користь прямої регресії 3D карти точок (3D Point Map).

Архітектурно мережа побудована за схемою «енкодер-декодер» з використанням механізму уваги (Attention mechanism). В якості енкодера використовується потужний візуальний трансформер, попередньо навчений за методикою DINOv2. Вибір трансформерної архітектури зумовлений її здатністю захоплювати глобальний контекст зображення через механізм Self-Attention. Це дозволяє моделі ефективно обробляти області з низькою текстурною інформацією (наприклад, однорідні стіни), спираючись на контекстне розуміння структури сцени, а не лише на локальні градієнти яскравості. Декодер складається з набору згорткових шарів, які послідовно відновлюють просторову роздільну здатність карт ознак до розміру вхідного зображення. Вихідний шар декодера генерує тензор розмірністю $H \times W \times 3$, що відповідає координатам (x, y, z) для кожного пікселя. Архітектура та потік даних нейромережі MoGe наведено в роботі [35].



Рис. 2.2. Архітектура та потік даних нейромережі MoGe

З рисунку 2.2 видно, що архітектура MoGe базується на гібридному підході, що поєднує енкодер Vision Transformer для захоплення глобального контексту та

згортковий декодер для деталізації локальної геометрії. Ключовою особливістю моделі є двоетапна регресія:

1. Спочатку генерується проміжна афінно-інваріантна карта, яка кодує чисту форму об'єктів незалежно від параметрів камери.

2. Потім, використовуючи передбачену фокусну відстань (f_{pred}), мережа трансформує цю карту у фінальну карту точок у просторі камери $P \in \mathbb{R}^{H \times W \times 3}$.

Окрім координат точок, мережа генерує бінарну маску валідності (показана чорним кольором на схемі), що дозволяє автоматично відфільтровувати геометрично некоректні області (наприклад, небо або дзеркальні поверхні) ще до етапу реєстрації ICP.

Фундаментальною проблемою монокулярного зору є невизначеність масштабу. Традиційні методи намагаються передбачити абсолютну глибину, що часто призводить до значних похибок на нових даних (domain gap). MoGe вирішує цю проблему шляхом прогнозування афінно-інваріантної карти точок P_{pred} . Замість того, щоб намагатися вгадати точний метричний масштаб, мережа генерує геометричну структуру, яка є правильною з точністю до афінного перетворення.

Математично, для кожного пікселя u_i мережа передбачає вектор $p_i \in \mathbb{R}^3$. Співвідношення між передбаченою точкою p_i та істинною точкою p_i^* у просторі камери описується моделлю:

$$p_i^* = sRp_i + t \quad (2.4)$$

де $s \in \mathbb{R}^+$ - коефіцієнт масштабування, $R \in SO(3)$ – матриця повороту, $t \in \mathbb{R}^3$ – вектор зсуву.

Такий підхід дозволяє відокремити задачу реконструкції форми об'єктів від задачі визначення їх абсолютного розміру. У розробленій системі VIO метричний масштаб (s) відновлюється на етапі оптимізації фактор-графа завдяки інтеграції з даними акселерометра IMU.

Окрім основної карти точок, архітектура MoGe включає допоміжні гілки, які прогнозують:

- Фокусну відстань камери (f_{pred}): мережа оцінює кут огляду вхідного зображення. Це дозволяє перетворити афінну карту точок у метрично-коректну структуру у просторі камери мінімізуючи геометричні спотворення.

- Маску впевненості (Valid Mask): мережа генерує бінарну маску, що відфільтровує пікселі, для яких геометрична оцінка є ненадійною або фізично некоректною (наприклад, небо, прозорі або дзеркальні об'єкти).

Висока точність моделі досягається завдяки запропонованій авторами функції втрат, яка мінімізує похибку вирівнювання. Під час навчання передбачена хмара точок динамічно вирівнюється з істинною хмарою (Ground Truth) методом найменших квадратів (Least Squares Solver) для знаходження оптимальних параметрів трансформації s, R, t . Це дозволяє обчислювати градієнти похибки базуючись на формі об'єктів, ігноруючи неминучі розбіжності у глобальному масштабі та зсуві, що присутні у різномірних навчальних датасетах.

Таким чином, використання MoGe у складі розробленої системи забезпечує отримання щільних, геометрично узгоджених хмар точок, які зберігають колінеарність ліній та планарність поверхонь, що є критично важливим для надійної роботи алгоритму реєстрації ICP.

2.2.2. Попередня обробка та підготовка хмар точок

Вихідний тензор нейронної мережі MoGe являє собою організовану щільну хмару точок P_{raw} , що містить координати для кожного пікселя вхідного зображення. Для стандартної роздільної здатності 640×480 , така хмара налічує 307 200 точок. Пряме використання такого масиву даних в ітеративних алгоритмах реєстрації (ICP) є недоцільним через квадратичну залежність обчислювальної складності від кількості точок, що унеможлиблює роботу системи в режимі реального часу.

Для забезпечення балансу між точністю геометричного опису сцени та швидкістю системи розроблено алгоритм попередньої обробки, який включає етапи маскування та воксельну дискретизацію.

Маскування та фільтрація шумів

На першому етапі застосовується фільтрація на основі маски впевненості (valid mask) M , згенерованої нейромережею. Точки, що відповідають областям з невизначеною геометрією (небо, прозорі об'єкти, дзеркальні поверхні) або лежать за межами робочого діапазону сенсора, вилучаються з обробки.

Додатково застосовується фільтр статистичного видалення викидів (Statistical Outlier Removal – SOR). Для кожної точки p_i обчислюється середня відстань d_i до k найближчих сусідів. Точки, для яких d_i перевищує порогове значення $(\mu_d + \alpha \cdot \sigma_d)$, де μ_d та σ_d - середнє та стандартне відхилення відстаней по всій хмарі, класифікуються як шум і видаляються. Це дозволяє мінімізувати вплив "летючих пікселів" на межах об'єктів.

Адаптивна воксельна дискретизація (Voxel Grid Downsampling)

Для суттєвого зменшення кількості точок при збереженні структурної інформації застосовується метод воксельної сітки. 3D-простір розбивається на кубічні комірки (воксели) фіксованого розміру δ . У даній роботі експериментально встановлено оптимальне значення $\delta = 0.05\text{м}$, що забезпечує достатню деталізацію для навігації в приміщеннях. Всі точки $\{p_1, \dots, p_m\}$, що потрапили в один воксель, замінюються однією точкою - центроїдом $\bar{p}$:

$$\bar{p} = \frac{1}{m} \sum_{j=1}^m p_j \quad (2.5)$$

Використання центроїда (на відміну від вибору центру вокселя) дозволяє зберегти точність та діє як низькочастотний фільтр, згладжуючи локальний шум

вимірювань MoGe. В результаті цієї операції кількість точок зменшується до 5 000 - 10 000 на кадр, що дозволяє виконувати реєстрацію за 10-20 мс.

2.3. Оцінка руху на основі геометричної реєстрації

Наступним критично важливим етапом функціонування розробленої навігаційної системи є визначення параметрів руху камери у просторі. Цей процес реалізується шляхом геометричної реєстрації послідовних тривимірних сканів (хмар точок), отриманих на попередньому етапі від нейромережевого модуля. У той час як класичні системи візуально-інерціальної одометрії (VIO), такі як VINS-Mono [14] або ORB-SLAM [7], покладаються на мінімізацію похибки репроекції розріджених характерних точок на 2D-площині зображення, запропонована система оперує безпосередньо у метричному 3D-просторі. Це дозволяє використовувати повну геометричну інформацію про сцену, що робить оцінку траєкторії більш стійкою до відсутності текстур та змін освітлення.

Для реалізації модуля одометрії у даній роботі обрано підхід, що базується на бібліотеці KISS-ICP (Keep It Small and Simple ICP) [36]. Цей вибір не є випадковим і зумовлений глибоким аналізом сучасних методів реєстрації хмар точок. Більшість існуючих рішень для LiDAR-одометрії (наприклад, LOAM [64], LEGO-LOAM [65]) намагаються досягти високої точності шляхом ускладнення конвеєра обробки: введення складних евристик для виділення ознак (країв, площин), використання графів факторів безпосередньо у фронтенді або складних схем дескрипторів. Натомість підхід KISS-ICP, запропонований Vizzo et al., демонструє, що класичний алгоритм Iterative Closest Point (ICP) може досягати точності рівня state-of-the-art та перевершувати складніші аналоги, якщо правильно реалізувати ключові етапи обробки даних та асоціації.

Вибір даного методу для інтеграції з нейромережею MoGe зумовлений трьома ключовими факторами:

1. Висока обчислювальна ефективність: алгоритм оптимізовано для роботи в реальному часі на CPU, що дозволяє вивільнити ресурси GPU для "важкої" нейромережі MoGe.

2. Стійкість до динаміки руху: використання адаптивних порогів асоціації дозволяє системі надійно працювати навіть при різких маневрах БПЛА.

3. Відсутність вимоги до нормалей: KISS-ICP базується на метриці «точка-до-точки» (Point-to-Point) [66], що усуває необхідність попереднього обчислення нормалей поверхонь, яке є обчислювально затратним і часто нестабільним на зашумлених хмарах точок.

Обґрунтування метрики Point-to-Point.

Традиційно вважається, що метрика «точка-до-площини» (Point-to-Plane) [67] забезпечує швидшу збіжність алгоритму ICP, оскільки вона дозволяє хмарам ковзати одна вздовж одної по плоских поверхнях. Однак, такий підхід вимагає точної оцінки нормалей. У випадку використання синтезованих хмар точок від нейромережі MoGe, локальна геометрія може містити високочастотний шум, що призводить до помилок в орієнтації нормалей і, як наслідок, до дестабілізації одометрії. Підхід KISS-ICP реабілітує метрику «точка-до-точки». Автори методу довели, що при використанні надійної схеми відбракування викидів (outlier rejection) та правильного передбачення руху, простіша метрика Point-to-Point забезпечує порівнянну точність, будучи при цьому значно робастнішою до локальних геометричних артефактів. Це робить її ідеальним вибором для обробки даних, отриманих методами глибинного навчання.

Стратегія «Frame-to-Map» та локальна карта.

Стратегія оцінки руху в розробленій системі полягає у реєстрації поточної хмари точок, отриманої від нейромережі, не до попереднього кадру (Frame-to-Frame), а до локальної карти (Local Map), яка формується інкрементально. Підхід «Frame-to-Map» є критично важливим для зменшення дрейфу одометрії. При

реєстрації лише пари сусідніх кадрів помилка неминуче накопичується з кожним кроком. Натомість, реєстрація відносно локальної карти дозволяє кожному новому вимірюванню спиратися на геометрично узгоджену структуру, сформовану попередніми спостереженнями, що забезпечує геометричну стабільність траєкторії.

Реалізація локальної карти в KISS-ICP базується на структурі хешування вокселів (Voxel Hashing). На відміну від традиційних структур пошуку, таких як k-d дерева, які вимагають перебудови або балансування при додаванні нових точок, хеш-таблиця дозволяє виконувати операції вставки та пошуку найближчого сусіда за константний час $O(1)$. Це є вирішальним фактором для роботи з щільними хмарами точок від MoGe, які можуть містити тисячі точок на кожному кадрі. Локальна карта підтримується за принципом ковзного вікна: старі точки, що знаходяться за межами певної відстані від поточного положення робота (БПЛА), автоматично видаляються, що обмежує використання пам'яті та запобігає зростанню часу обчислень.

Адаптивна асоціація даних

Однією з головних інновацій використаного методу, яка забезпечує його надійність, є схема адаптивного визначення порогу відповідності (Adaptive Correspondence Thresholding). У класичному ICP вибір максимальної відстані для пошуку відповідних точок між двома хмарами є складною проблемою: занадто малий поріг не дозволяє алгоритму збігтися при швидкому русі, а занадто великий призводить до асоціації з неправильними поверхнями та локальних мінімумах. У KISS-ICP цей параметр не є фіксованою константою. Поріг розраховується динамічно для кожного кадру на основі поточної оцінки швидкості робота (БПЛА) та частоти кадрів. Якщо система фіксує швидкий рух або різкий ривок (на основі передбачення моделі руху), поріг автоматично збільшується, дозволяючи алгоритму шукати відповідності в ширшому діапазоні. Коли рух стабілізується, поріг зменшується, забезпечуючи високу точність фінального вирівнювання.

Математично це реалізується через статистичний аналіз відхилень поточної реєстрації, що дозволяє системі самостійно визначати, які точки є викидами, а які – частиною корисної геометрії.

Передбачення руху та обробка спотворень.

Для забезпечення збіжності ICP алгоритм потребує якісного початкового наближення. У розробленій системі реалізовано модуль передбачення руху на основі моделі постійної швидкості (Constant Velocity Model). Для кожного нового кадру початкове положення екстраполюється на основі траєкторії попередніх кадрів. Це особливо ефективно для квадрокоптерів, які мають інерцію руху. Крім того, хоча використовувана камера має глобальний затвор, архітектура KISS-ICP включає в себе модуль компенсації спотворень руху для роботи з даними LiDAR. У контексті даної роботи цей модуль адаптовано для можливості майбутньої інтеграції з камерами rolling shutter, де кожен рядок зображення має свій часовий зсув.

Таким чином, інтеграція бібліотеки KISS-ICP у контур навігації дозволяє створити високоефективний та надійний фронтенд одометрії. Відмова від складних евристик на користь надійної реалізації базових принципів реєстрації, підсилена сучасною структурою даних (Voxel Hashing) забезпечує системі здатність генерувати точні вимірювання відносного переміщення, які надалі уточнюються у глобальному графі оптимізації.

2.4. Преінтеграції вимірювань IMU на основі кінетичної моделі із шумами

Стан IMU в момент часу t описується орієнтацією $R_{WB}(t) \in SO(3)$, що відображає вектори з системи $\{B\}$ у $\{W\}$, та позицією $p_W(t) \in \mathbb{R}^3$ у світовій системі.

Сенсор генерує вимірювання кутової швидкості та лінійного прискорення, які спотворені адитивним шумом та змінним у часі зміщенням (bias).

Рівняння вимірювання гіроскопа $\tilde{w}_{WB}^B(t)$ має вигляд [68]:

$$\tilde{\omega}_{WB}^B(t) = \omega_{WB}^B(t) + b_g(t) + \eta_g(t) \quad (2.6)$$

де $\omega_{WB}^B(t)$ - істинна кутова швидкість тіла відносно світу, виражена в системі $\{B\}$, $b_g(t)$ - зміщення гіроскопа, $\eta_g(t)$ - адитивний білий шум.

Рівняння вимірювання акселерометра $\tilde{a}^B(t)$ враховує вектор гравітації g^W [68]:

$$\tilde{a}^B(t) = R_{WB}^T(t)(a^W(t) - g^W) + b_a(t) + \eta_a(t) \quad (2.7)$$

де $a^W(t)$ - істинне прискорення сенсора у світовій системі, g^W - вектор прискорення вільного падіння. У даній моделі впливом обертання Землі знехтувано, тому система $\{W\}$ вважається інерціальною.

Шумові компоненти η_g, η_a моделюються як гаусівський білий шум, а зміщення b_g, b_a - як випадкове блукання.

Еволюція стану системи описується системою диференціальних рівнянь [68]:

$$\begin{cases} \dot{R}_{WB} = R_{WB} \omega_{WB}^\wedge \\ \dot{v}^W = a^W \\ \dot{p}^W = v^W \end{cases} \quad (2.8)$$

де $[\cdot]^\wedge$ - позначає оператор, що перетворює вектор $\mathbb{R}^3$ у кососиметричну матрицю для алгебри Лі $so(3)$.

Для переходу від неперервної моделі до дискретної реалізації в алгоритмі, виконується інтегрування рівнянь руху на інтервалі часу $[t, t + \Delta t]$. Припускаючи, що виміряні прискорення та кутова швидкість залишаються постійними протягом малого інтервалу дискретизації Δt (гіпотеза нульового порядку), стан системи в момент $t + \Delta t$ визначається наступним чином [68]:

1) Орієнтація

$$R(t + \Delta t) = R(t) \text{Exp}((\tilde{\omega}(t) - b_g(t) - \eta_{gd}(t))\Delta t) \quad (2.9)$$

2) Швидкість

$$v(t + \Delta t) = v(t) + g\Delta t + R(t)(\tilde{a}(t) - b_a(t) - \eta_{ad}(t))\Delta t \quad (2.10)$$

3) Позиція

$$p(t + \Delta t) = p(t) + v(t)\Delta t + \frac{1}{2}g\Delta t^2 + \frac{1}{2}R(t)(\tilde{a}(t) - b_a(t) - \eta_{ad}(t))\Delta t^2 \quad (2.11)$$

Тут η_{gd} та η_{ad} - дискретні версії шумів, коваріація яких масштабується обернено пропорційно кроку дискретизації ($\frac{1}{\Delta t}$). Хоча даний метод інтегрування є наближеним (оскільки припускає $R(t)$ постійним під час інтегрування швидкості), висока частота роботи сучасних IMU (понад 200 Гц) робить похибку цього наближення несуттєвою для задач навігації, дозволяючи ефективно використовувати цю модель для преінтеграції у фактор-графі.

Пряме використання дискретних рівнянь руху для формування обмежень у фактор-графі є недоцільним, оскільки це вимагало б додавання нового вузла стану для кожного вимірювання IMU. Враховуючи високу частоту дискретизації інерціальних сенсорів (200–400 Гц), розмірність вектору стану зростала б надто швидко, що унеможливило б оптимізацію в реальному часі.

Для вирішення цієї проблеми застосовано метод преінтеграції на многовиді (Manifold Pre-integration). Основна ідея методу полягає у згортанні сотень інерціальних вимірювань між двома послідовними ключовими кадрами t_i та t_j у єдиний вектор «сумарного» вимірювання, який накладає кінематичне обмеження на відносний рух між цими кадрами [68].

Визначення відносних інкрементів руху. Традиційне інтегрування інерціальних даних залежить від початкової орієнтації R_i та швидкості v_i . Оскільки ці величини є змінними оптимізації і постійно уточнюються, будь-яка зміна оцінки стану R_i вимагала б повного перерахунку інтегралів для всіх наступних кадрів.

Щоб уникнути повторного інтегрування, вводиться поняття преінтегрованих вимірювань відносної орієнтації ΔR_{ij} , швидкості Δv_{ij} та позиції Δp_{ij} , які залежать виключно від даних IMU та оцінок зміщень (biases), але є інваріантними до початкового стану системи [68]:

$$\Delta R_{ij} = \prod_{k=i}^{j-1} \text{Exp}((\tilde{\omega}_k - b_k^g - \eta_k^{gd})\Delta t) \quad (2.12)$$

$$\Delta v_{ij} = \sum_{k=i}^{j-1} \Delta R_{ik} (\tilde{a}_k - b_k^a - \eta_k^{ad}) \Delta t \quad (2.13)$$

$$\Delta p_{ij} = \sum_{k=i}^{j-1} \left[\Delta v_{ik} \Delta t + \frac{1}{2} \Delta R_{ik} (\tilde{a}_k - b_k^a - \eta_k^{ad}) \Delta t^2 \right] \quad (2.14)$$

де ΔR_{ij} - відносний поворот від моменту i до проміжного моменту k . Зауважимо, що вектор гравітації g виключено з цих рівнянь, оскільки він враховується окремо у рівнянні нев'язки (residual).

Формування моделі вимірювання. Отримані преінтегровані величини дозволяють зв'язати стани двох ключових кадрів i та j наступним геометричним співвідношенням у глобальній системі координат [68]:

$$R_j = R_i \Delta R_{ij} \text{Exp}(\delta \phi_{ij}) \quad (2.15)$$

$$v_j = v_i + g \Delta t_{ij} + R_i (\Delta v_{ij} + \delta v_{ij}) \quad (2.16)$$

$$p_j = p_i + v_i \Delta t_{ij} + \frac{1}{2} g \Delta t_{ij}^2 + R_i (\Delta p_{ij} + \delta p_{ij}) \quad (2.17)$$

де, Δt_{ij} - інтервал часу між кадрами, а $\delta \phi_{ij}, \delta v_{ij}, \delta p_{ij}$ - компоненти шуму преінтеграції.

Шум преінтеграції $[\delta \phi_{ij}^T, \delta v_{ij}^T, \delta p_{ij}^T]$ моделюється як гаусівський процес із нульовим середнім та коваріаційною матрицею Σ_{ij} . Оскільки преінтегровані величини є лінійною комбінацією вимірювань IMU, коваріація Σ_{ij} обчислюється ітеративно на етапі преінтеграції шляхом лінійного поширення коваріацій шумів акселерометра та гіроскопа через якобіани динаміки системи [68]. Отримана матриця Σ_{ij} використовується як вагова матриця (матриця інформації) у факторі IMU під час оптимізації.

Корекція зміщень IMU (Bias Updates). Рівняння преінтеграції залежать від оцінок зміщень акселерометра b_a та гіроскопа b_g , які вважаються постійними на інтервалі (i, j) . Однак під час оптимізації оцінка зміщень уточнюється. Щоб уникнути дороговартісного повторного інтегрування при кожній малій зміні δb ,

застосовується лінійна апроксимація на основі розкладання Тейлора першого порядку [68]:

$$\Delta R_{ij}(b_g + \delta b_g) \approx \Delta R_{ij}(b_g) \cdot \text{Exp}\left(\frac{\partial \Delta R_{ij}}{\partial b_g} \delta b_g\right) \quad (2.18)$$

$$\Delta v_{ij}(b + \delta b) \approx \Delta v_{ij}(b) + \frac{\partial \Delta v_{ij}}{\partial b_a} \delta b_a + \frac{\partial \Delta v_{ij}}{\partial b_g} \delta b_g \quad (2.19)$$

$$\Delta p_{ij}(b + \delta b) \approx \Delta p_{ij}(b) + \frac{\partial \Delta v_{ij}}{\partial b_a} \delta b_a + \frac{\partial \Delta p_{ij}}{\partial b_g} \delta b_g \quad (2.20)$$

де, b – загальне зміщення по гіроскопу (b_g) та акселерометру (b_a).

Якобіани зміщень (наприклад, $\frac{\partial \Delta v_{ij}}{\partial b_a}$) обчислюються аналітично під час процесу преінтеграції. Такий підхід дозволяє коригувати значення преінтегрованих вимірювань "на льоту" із мінімальними обчислювальними витратами, забезпечуючи ефективність алгоритму оптимізації.

Висновки до розділу 2

У другому розділі дисертаційної роботи розв'язано задачу розробки математичного та алгоритмічного забезпечення для системи гібридної монокулярної візуально-інерціальної одометрії, що базується на використанні щільних карт глибини.

Основні результати, отримані в розділі:

1. *Розробка архітектури системи.* Запропоновано нову архітектуру «жорстко пов'язаної» сенсорної фузії, яка, на відміну від класичних методів на основі розріджених ознак, використовує щільну геометричну інформацію про сцену. Архітектура побудована за модульним принципом (геометричний фронтенд, інерціальний фронтенд, бекенд оптимізації), що дозволяє поєднувати методи глибокого навчання з імовірнісною оцінкою станів.

2. *Візуальний модуль на основі MoGe.* Обґрунтовано та імплементовано використання геометрично-орієнтованої нейронної мережі MoGe для генерації вхідних даних одометрії. Показано, що ключовою перевагою цього підходу є пряма регресія афінно-інваріантної карти точок у просторі камери. Це дозволяє відмовитися від явного використання матриці інтринсиків камери на етапі реконструкції та забезпечує отримання щільних хмар точок навіть для областей з однорідною текстурою, де традиційні дескриптори не працюють.

3. *Адаптивна геометрична реєстрація.* Розроблено модуль оцінки візуального руху на базі алгоритму KISS-ICP (метрика «точка-до-точки»). Впровадження схеми адаптивного порогу асоціації, що залежить від швидкості руху, та використання структури хешування вокселів (Voxel Hashing) для зберігання локальної карти дозволило забезпечити високу швидкість обробки даних в режимі реального часу та стійкість до динамічних змін траєкторії БПЛА.

4. *Інерціальна преінтеграція на многовиді.* Сформульовано математичну модель інерціального фронтенду, що базується на кінематиці твердого тіла в просторі $SO(3)$. Реалізовано метод преінтеграції вимірювань IMU, який дозволяє згортати високочастотні дані акселерометра та гіроскопа в єдиний фактор відносного руху. Це вирішує проблему розмірності графа оптимізації та дозволяє коригувати оцінки зміщень сенсорів без повторного чисельного інтегрування.

5. *Оптимізація на основі фактор-графа.* Побудовано глобальну модель оптимізації, яка об'єднує різноманітні обмеження: щільні візуальні фактори від ICP та інерціальні фактори від преінтеграції. Така структура забезпечує коректне врахування невизначеності вимірювань та дозволяє відновити метричний масштаб траєкторії, використовуючи інерціальні дані як опорні.

Таким чином, у розділі створено теоретичний та алгоритмічний базис для реалізації системи навігації, здатної функціонувати в складних умовах (відсутність текстур, зміна освітлення), що дозволяє перейти до етапу програмної реалізації та експериментальної перевірки, які будуть розглянуті у наступних розділах.

РОЗДІЛ 3. ОЦІНКА СТАНУ БПЛА НА ОСНОВІ ОПТИМІЗАЦІЇ ФАКТОР-ГРАФА

Заключним етапом обробки сенсорної інформації є оцінка траєкторії руху робота (БПЛА) та уточнення параметрів його стану. У той час як фронтенд системи відповідає за попередню обробку «сирих» даних та генерацію локальних обмежень, бекенд (backend) виконує задачу глобальної узгодженості.

Традиційні підходи до навігації часто використовують методи фільтрації (наприклад, розширений фільтр Калмана – EKF або MSCKF). Однак фільтри мають суттєвий недолік: вони лінеаризують систему лише один раз у поточній робочій точці, що призводить до накопичення похибок лінеаризації.

У даній роботі застосовано більш сучасний підхід на основі оптимізації графа, який також відомий як згладжування. Цей метод дозволяє ітеративно перераховувати минулі стани об'єкту керування, виправляючи помилки лінеаризації, що забезпечує значно вищу точність порівняно з фільтраційними методами. Задача формулюється як пошук оцінки за критерієм максимуму апостеріорної ймовірності над множиною всіх вимірювань.

3.1. Критерій злиття даних на основі фактор-графа

Для реалізації тісно пов'язаної (tightly-coupled) архітектури використовується структура фактор-графа. Фактор-граф – це двочастковий граф $G = (X, F)$, що складається з двох типів вузлів:

1. Вузли змінних (X): Представляють невідомі стани системи, які необхідно оцінити.
2. Вузли факторів (F): Представляють імовірнісні обмеження, що впливають із вимірювань сенсорів (IMU, камера) або апріорної інформації.

Візуальне представлення фактор-графа оптимізації наведено на рис. 3.1 [62].

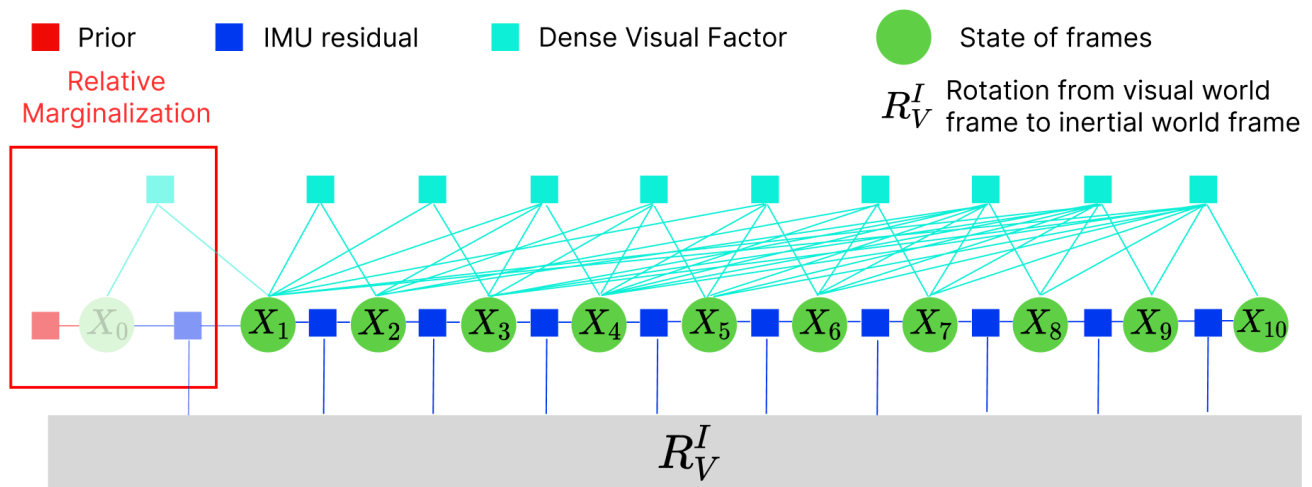


Рис. 3.1. Візуальне представлення фактор-графу оптимізації

У фактор-графі оптимізується стан системи у моменти часу i . Стан x_i у момент t_i описується як [62]:

$$x_i = [R_i, p_i, v_i, b_i^a, b_i^g] \quad (3.1)$$

де

- $R_i \in SO(3)$: Орієнтація тіла (IMU) відносно світу.
- $p_i \in \mathbb{R}^3$: Позиція IMU у світовій системі.
- $v_i \in \mathbb{R}^3$: Швидкість IMU у світовій системі.
- $b_i^a, b_i^g \in \mathbb{R}^3$: Зміщення акселерометра та гіроскопа.

Загальна функція витрат задачі мінімізації:

$$J(x) = \sum_{i,j \in K} \|r_l(z_{ij}, x_i, x_j)\|_{\Sigma_l}^2 + \sum_{i,j \in V} \|r_v(T_{ij}^{icp}, T_i, T_j)\|_{\Sigma_v}^2 + \|r_{prior}\|_{\Sigma_p}^2 \quad (3.2)$$

У рівнянні (3.2) є три фактори - це IMU Factor, Dense Visual Factor та Prior Factor. Розберемо кожний із них детальніше.

Інерціальний фактор (IMU Factor). Це сума всіх помилок, які виникають через невідповідність руху робота (БПЛА) даним IMU між відповідними кадрами.

$$\sum_{i,j \in K} \|r_l(z_{ij}, x_i, x_j)\|_{\Sigma_l}^2 \quad (3.3)$$

де

- x_i, x_j – вектори стану в моменти часу t_i та t_j .
- z_{ij} – преінтегроване вимірювання від IMU. Включає $\Delta R_{ij}, \Delta v_{ij}, \Delta p_{ij}$. ΔR_{ij} – поворот відносно кадру в момент часу i до моменту часу j . Δv_{ij} – відносна зміна швидкості. Δp_{ij} – відносна зміна положення.
- Σ_l – матриця коваріації, що описує шум IMU.

У сучасних алгоритмах візуально-інерціальної одометрії ключовим етапом є формування фактору преінтеграції. Замість того, щоб інтегрувати дані акселерометра та гіроскопа у глобальній системі координат (що вимагало б повторного обчислення при кожній зміні оцінки стану), ми інтегруємо їх у локальній системі координат попереднього кадру.

Наведені нижче рівняння описують модель вимірювань, яка пов'язує попередньо інтегровані дані IMU з оцінюваними станами системи [68]:

$$\begin{aligned} \Delta R_{ij} &= R_i^T R_j \text{Exp}(\delta \phi_{ij}) \\ \Delta v_{ij} &= R_i^T (v_j - v_i - g \Delta t_{ij}) + \delta v_{ij} \\ \Delta p_{ij} &= R_i^T \left(p_j - p_i - v_i \Delta t_{ij} - \frac{1}{2} g \Delta t_{ij}^2 \right) + \delta p_{ij} \end{aligned} \quad (3.4)$$

де

- ΔR_{ij} - преінтегроване вимірювання обертання (зміни орієнтації).
- Δv_{ij} - преінтегроване вимірювання зміни швидкості.
- Δp_{ij} - преінтегроване вимірювання зміни позиції.
- R_i, R_j - матриці обертання (Rotation matrices) у моменти часу i та j відносно глобальної системи координат.
- p_i, p_j - вектори позиції у глобальній системі координат.
- v_i, v_j - вектори швидкості у глобальній системі координат.

- g – вектор гравітації у глобальній системі координат
- Δt_{ij} – інтервал часу між кадрами i та j .
- $Exp(\delta\phi_{ij})$ - експоненційне відображення для помилки орієнтації. Оскільки обертання лежать на многовиді ($SO(3)$), ми не можемо просто додати шум; ми використовуємо експоненційну карту для представлення малого відхилення.

- $\delta\phi_{ij}, v_{ij}, \delta p_{ij}$ - вектори шуму (random noise vectors) для орієнтації, швидкості та позиції відповідно. Вони моделюються як гаусовський білий шум.

Враховуючи попередньо інтегровану модель вимірювання можна легко представити залишкові помилки для фактору IMU [68]:

$$\begin{aligned}
 r_{\Delta R_{ij}} &= \text{Log} \left(\left(\Delta R_{ij}(b_i^g) \text{Exp} \left(\frac{\partial \Delta R_{ij}}{\partial b^g} \delta b^g \right) \right)^T R_i^T R_j \right) \\
 r_{\Delta v_{ij}} &= R_i^T (v_j - v_i - g \Delta t_{ij}) - \left[\Delta v_{ij}(b_i^g, b_i^a) + \frac{\partial \Delta v_{ij}}{\partial b^g} \delta b^g + \frac{\partial \Delta v_{ij}}{\partial b^a} \delta b^a \right] \\
 r_{\Delta p_{ij}} &= R_i^T \left(p_j - p_i - v_i \Delta t_{ij} - \frac{1}{2} g \Delta t_{ij}^2 \right) \\
 &\quad - \left[\Delta p_{ij}(b_i^g, b_i^a) + \frac{\partial \Delta p_{ij}}{\partial b^g} \delta b^g + \frac{\partial \Delta p_{ij}}{\partial b^a} \delta b^a \right]
 \end{aligned} \tag{3.5}$$

де,

- $r_{\Delta R_{ij}}, r_{\Delta v_{ij}}, r_{\Delta p_{ij}}$ – залишки обертання, швидкості та позиції відповідно.
- $\frac{\partial \Delta R_{ij}}{\partial b^g}, \frac{\partial \Delta v_{ij}}{\partial b^g}, \frac{\partial \Delta v_{ij}}{\partial b^a}, \frac{\partial \Delta p_{ij}}{\partial b^g}, \frac{\partial \Delta p_{ij}}{\partial b^a}$ - якобіани преінтегрованих вимірювань по відношенню до зміщень гіроскопа та акселерометра.
- $\delta b^g, \delta b^a$ – відхилення поточного зміщення від номінального.
- $\text{Log}(\cdot)$ - логарифмічне відображення, яке переводить помилку обертання з простору матриць $SO(3)$ у векторний простір дотичної алгебри $\mathbb{R}^3$. Це дозволяє мінімізувати "кут повороту" помилки.

Візуальний фактор (Dense Visual Factor). Це сума помилок неузгодженості між тим, як робот рухається згідно з графом та преінтегрованими позами, і тим, як зміщуються хмари точок (MoGe + ICP) між кадрами [62].

$$\sum_{i,j \in V} ||r_V(T_{ij}^{icp}, T_i, T_j)||_{\Sigma_V}^2 \quad (3.6)$$

де

- T_{ij}^{icp} – відносна поза, що є результатом роботи алгоритму ICP, який наклав хмару точок з кадру i на хмару точок кадру j . Це є вимірювання (Measurement) і це константа для оптимізатора.

- T_i – поза камери в момент часу i в групі Лі $SE(3)$.

- T_j – поза камери в момент часу j в групі Лі $SE(3)$.

Функція нев'язки r_V визначається наступним чином:

$$r_V(T_{ij}^{icp}, T_i, T_j) = \text{Log}((T_{ij}^{icp})^{-1} \cdot (T_i^{-1}T_j))^V \quad (3.7)$$

де:

- $T_i^{-1}T_j$ – поточні оцінки поз T_i та T_j і обчислюємо відносний рух між ними.

- $(T_{ij}^{icp})^{-1} \cdot (T_i^{-1}T_j)$ – множимо обернену матрицю вимірювання на передбачення. Якщо передбачення ідеально збігається з ICP, то результат буде одиничною матрицею.

- $\text{Log}(\dots)^V$ - матриця помилки. Це незручно, оскільки потрібен вектор чисел, щоб піднести його до квадрата. Тому операція Log (логарифмічне відображення) переносить з групи Лі $SE(3)$ (матриці) в алгебру Лі $\mathfrak{se}(3)$ (дотичний простір). Оператор "vee" (V) перетворює це на вектор з 6 чисел: $[rot_x, rot_y, rot_z, trans_x, trans_y, trans_z]^T$.

- Σ_V – матриця коваріації. Вона є адаптивною, в залежності від якості хмари точок і накладання ICP.

Априорний Фактор (Prior Factor). Це різниця між поточним станом x_j робота (БПЛА) і фіксованим значенням, що є початковим положенням, звідки відбувся рух x_{start} (зазвичай – це позиція 0, 0, 0).

$$r_{prior} = x_j - x_{start} \quad (3.8)$$

де

- x_j – це змінні, які оптимізатор може змінювати (R, p, v, b_a, b_g).
- x_{start} – це жорстко задані початкові умови. В оптимізаторі ці значення як константи.

3.2. Порівняння програмних реалізацій алгоритму Bundle Adjustment для задачі SLAM

Задача бекенд-оптимізації в системах SLAM математично формулюється як задача нелінійних найменших квадратів великої розмірності. Ефективність її вирішення критично залежить від обраного програмного інструментарію, який повинен забезпечувати:

- Ефективну роботу з розрідженими матрицями: Гессіан системи SLAM має розріджену блочну структуру.
- Підтримку многовидів (Manifolds): змінні стану (орієнтація, пози) належать до груп Лі ($SO(3), SE(3)$), що вимагає специфічних операцій оновлення в дотичному просторі.
- Обчислення похідних: автоматичне або символічне диференціювання для точного розрахунку Якобіанів.

У SLAM багато років домінували фільтраційні методи, але зараз вони поступово витіснилися підходами, заснованими на оптимізації [69]. Підхід з оптимізацією був вперше представлений в [70], але спочатку не набув популярності через високу обчислювальну складність. Сьогодні, завдяки зростанню обчислювальних потужностей, методи з оптимізацією стали передовим стандартом і дозволяють швидко та точно розв'язувати задачі оцінювання в SLAM.

Загальна схема роботи на основі оптимізації, відповідно до [71], представлена на рисунку. 3.2.:



Рис. 3.2. Загальна схема роботи SLAM на основі оптимізації

SLAM на основі оптимізації можна розділити на 2 великих частини: Frontend алгоритми, що відповідають за оцінку положення камери, побудову 3D карти та Backend алгоритми, що відповідають за оптимізацію оцінених положень камери та карти.

Frontend алгоритми, як правило включають наступні етапи: виділення ключових точок (Feature Extraction), зіставлення точок (Feature Matching), оцінка положення камери (Pose Estimation), триангуляція - знаючи положення камери обчислення 3D-координати точок середовища (Triangulation), локальне створення карти (Local Mapping). До Backend алгоритмів (оптимізація) включають наступні:

- Bundle Adjustment - покращує оцінку положення камери та координати 3D-точок;
- Виявлення замикання циклів (Loop Closure Detection) - якщо камера повертається у вже відвідане місце, алгоритм повинен це визначити;
- Оптимізація графа поз (Pose Graph Optimization) - якщо знайдено замикання циклу, виконується оптимізація всієї карти, щоб уникнути накопичення помилок та зробити карту узгодженою на глобальному рівні.

В даному розділі буде розглянуто етап Bundle Adjustment (BA), оскільки він є ключовим для точного відновлення тривимірної структури сцени та параметрів

камери. Від якості його виконання залежать усі наступні етапи комп'ютерного зору, такі як побудова карти глибини, трекінг та реконструкція.

У процесі побудови 3D-карти ми маємо набір камер і набір 3D-точок, які спроектовані на зображення у вигляді 2D-координат. Bundle Adjustment (BA) – це метод оптимізації, який дозволяє покращити оцінку положення камер та 3D-точок сцени, зменшуючи помилку повторної проєкції (Reprojection Error).

Математично функція помилки повторної проєкції виглядає так [71]:

$$E = \sum_{c \in C} \sum_{p, p' \in P} d(p', Q(p, c)) \quad (3.9)$$

де

- c - одна з камер у множині камер C ;
- p - 3D-точка у множині всіх 3D-точок P ;
- p' - відповідна 2D-точка, яку камера c спостерігає;
- $Q(p, c)$ - функція проєкції 3D-точки p на площину камери c ;
- $d(p', Q(p, c))$ - відстань між спостереженою 2D-точкою p' і її проєкцією $Q(p, c)$.

Метою оптимізації є мінімізувати помилку між спостереженими 2D ключовими точками і проєкціями 3D точок на зображення. Візуально це представлено на рисунку 3.3 [71].

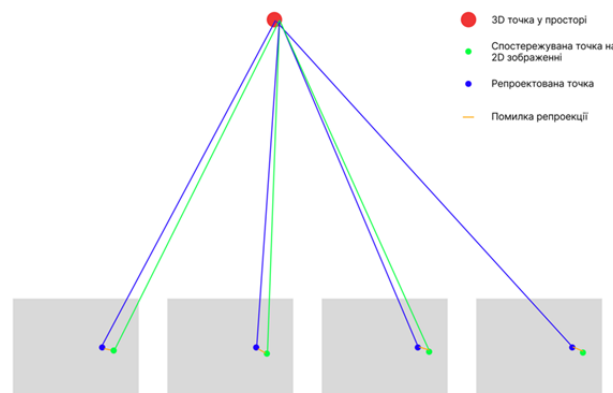


Рис. 3.3. Візуальне представлення репроєкції спостережуваної 3D точки, яка спостерігається у різних кадрах

Сьогодні уже існують готові рішення, реалізовані у вигляді бібліотек, що спрощують використання Bundle Adjustment без необхідності розробки власної реалізації «з нуля». Метою є порівняння таких бібліотек, як Ceres Solver [23], GTSAM ("Georgia Tech Smoothing and Mapping") [73], і SymForce [74], з точки зору їх швидкодії, точності та стійкості до шумів у даних.

Порівнюємо оптимізацію поз та траєкторій для бібліотек Ceres, Symforce і GTSAM на датасеті BAL (Bundle Adjustment in the Large) [19]. Загалом структура датасету включає:

- Кількість камер – визначає кількість позицій камер у сцені.
- Кількість 3D-точок – набір тривимірних точок, які проєктуються на зображення.
- Спостереження – 2D-координати точок у зображеннях, отримані камерами.
- Параметри камер – включають внутрішні (фокусна відстань, дисторсія) та зовнішні (позиція, орієнтація) параметри.
- Формат файлів – збереження у вигляді текстових файлів із чітко структурованими списками камер, точок та спостережень.

Для тестування було використано три набори даних: Ladybug, Trafalgar Square та Dubrovnik. Для обробки, оцінки та порівняння результатів траєкторії та побудови відповідних порівняльних графіків було використано бібліотеку Evo [75]. Блок схема програмного забезпечення для тесту оптимізації наведено на рисунку 3.4 [71].

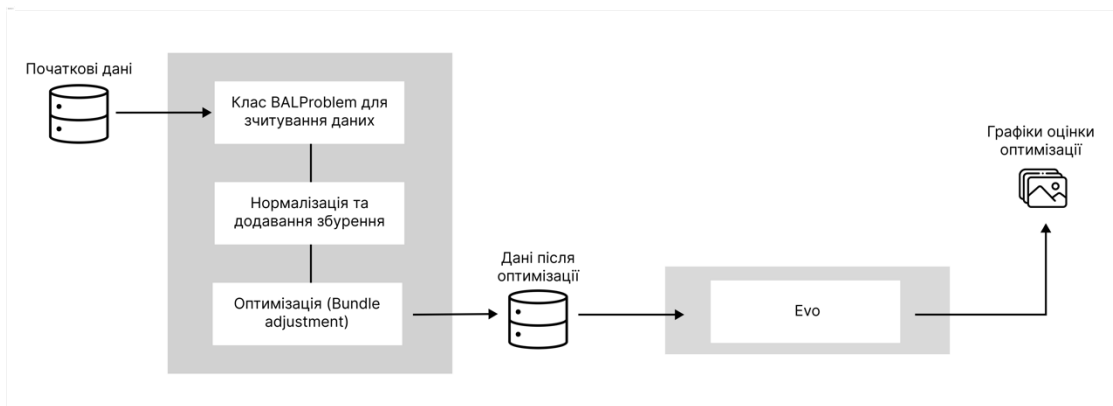


Рис. 3.4. Схема роботи додатку для порівняння бібліотек

Перед оптимізацією реалізовано нормалізацію та внесення випадкових збурень значень поз та точок датасету. Метод нормалізації масштабує геометрію так, щоб медіана абсолютного відхилення точок була рівною 100. Це забезпечує стабільність алгоритмів оптимізації. Нормалізація включає:

- Обчислення медіани координат контрольних точок.
- Визначення медіанного абсолютного відхилення.
- Масштабування всіх точок відносно медіани з коефіцієнтом, оберненим до медіанного абсолютного відхилення.
- Масштабування центрів камер за тим самим принципом.

Метод збурення вносить випадкові відхилення до даних для імітації шумів, що часто трапляються в реальних умовах. Його параметри:

- `rotation_sigma` – величина випадкового збурення до обертання камери.
- `translation_sigma` – величина збурення для трансляційної складової.
- `point_sigma` – випадкове збурення до координат точок.

Процес збурення включає:

- Додавання випадкових змін до координат точок, якщо `point_sigma > 0`.
- Додавання випадкових змін до параметрів обертання камер у формі кутів-осьових представлень (`angle-axis`).
- Додавання випадкових змін до трансляційної частини камер.

В даному експерименті було використано наступні значення для збурень: `rotation_sigma` – 0.1, `translation_sigma` – 0.5, `point_sigma` – 0.5 [71].

Застосування цих методів гарантує, що алгоритми оцінювання параметрів працюють на коректно масштабованих та імітовано зашумлених даних, що є важливим для об'єктивного порівняння продуктивності оптимізаційних бібліотек.

Нижче подано фрагменти коду [71] по підготовці вхідних даних та запуск оптимізаторів Ceres та SymForce. Запуск оптимізатора бібліотеки GTSAM аналогічний до попередніх. UML діаграма класів для розв'язання задачі Bundle Adjustment наведена в додатку А.

Фрагмент коду в якому відбувається читання даних датасету, їх нормалізація, додавання шумів та запуск bundle adjustment:

```
int main(int argc, char **argv) {
    std::string bal_file = "/data/ladybug/problem-49-7776-pre.txt";

    google::InitGoogleLogging(argv[0]);

    BALProblem bal_problem(bal_file);
    bal_problem.Normalize();
    bal_problem.Perturb(0.1, 0.5, 0.5);
    SolveBA(bal_problem);

    return 0;
}
```

Фрагмент коду з запуском оптимізатора Ceres:

```
ceres::Solver::Options options;
options.linear_solver_type = ceres::LinearSolverType::SPARSE_SCHUR;
options.minimizer_progress_to_stdout = true;
options.max_num_iterations = 200;
ceres::Solver::Summary summary;
ceres::Solve(options, &problem, &summary);
std::cout << summary.FullReport() << "\n";
```

Фрагмент коду з запуском оптимізатора SymForce:

```
auto params = sym::DefaultOptimizerParams();
params.verbose = true;
params.lambda_update_type = sym::lambda_update_type_t::DYNAMIC;
sym::Optimizer optimizer{params, std::move(factors)};
const auto stats = optimizer.Optimize(values, 200);

spdlog::info("Finished in {} iterations", stats.iterations.size());
```

Всього було проведено 9 експериментів: кожен з наборів даних (Ladybug, Trafalgar Square та Dubrovnik) оброблявся бібліотеками Ceres, SymForce, GTSAM відповідно. Кількісними критеріями роботи оптимізаційних бібліотек обрано: швидкодія, кількість ітерацій, абсолютна помилка позиції (APE) та значення помилки (Cost Initial, Cost Final). Результати оптимізації наведені в таблиці 3.1 [71]. Всі тести проводились на комп'ютері з Apple M3 8 core CPU. Нижче наведено графіки оцінки оптимізацій для даних Ladybug датасету з використанням Ceres [71].

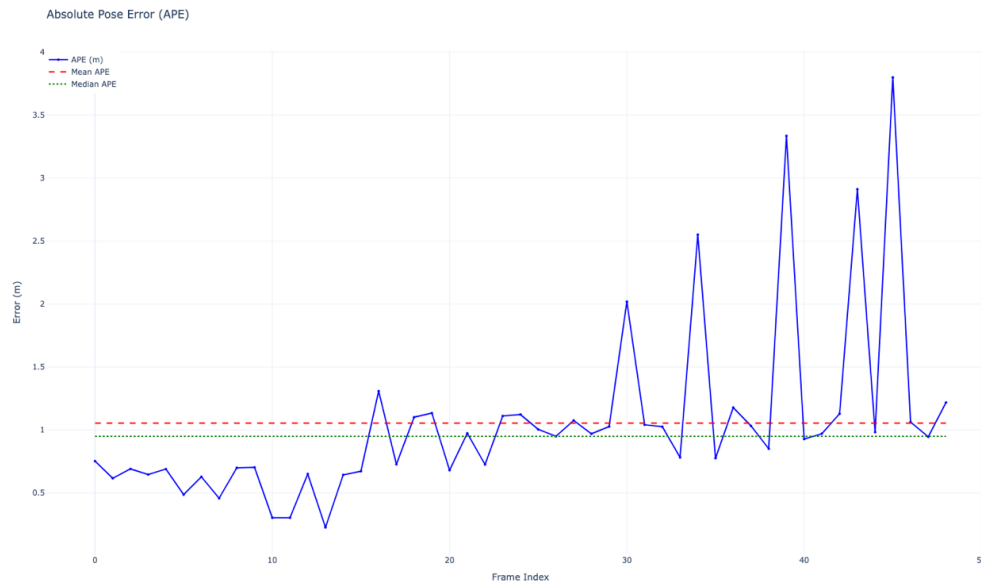


Рис. 3.5. Абсолютна помилка позиції (APE) для кожного кадру:
синя лінія показує абсолютну помилку позиції у метрах на кожному кадрі;
червона пунктирна лінія – середнє значення APE по всій траєкторії;
зелена точкова лінія – медіанне значення APE, що є менш чутливим до викидів.

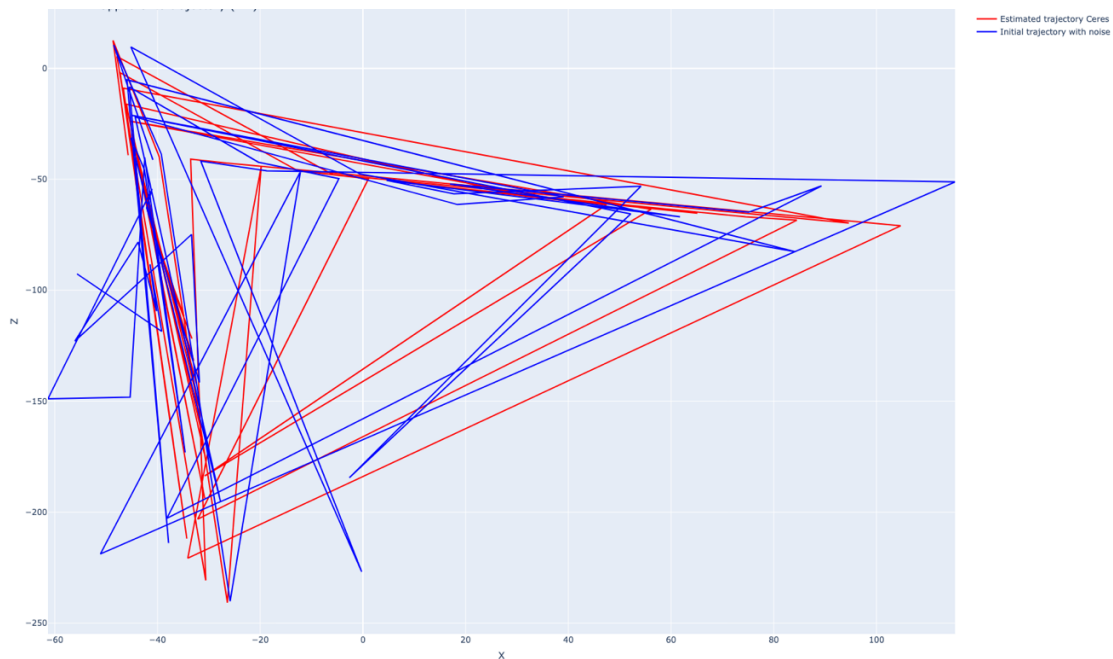


Рис. 3.6. Порівняння оптимізованої та початкової траєкторії у площині X-Z:
синя лінія – початкова траєкторія з шумами перед оптимізацією; червона лінія –
оптимізована траєкторія після процедури Bundle Adjustment (Ceres).

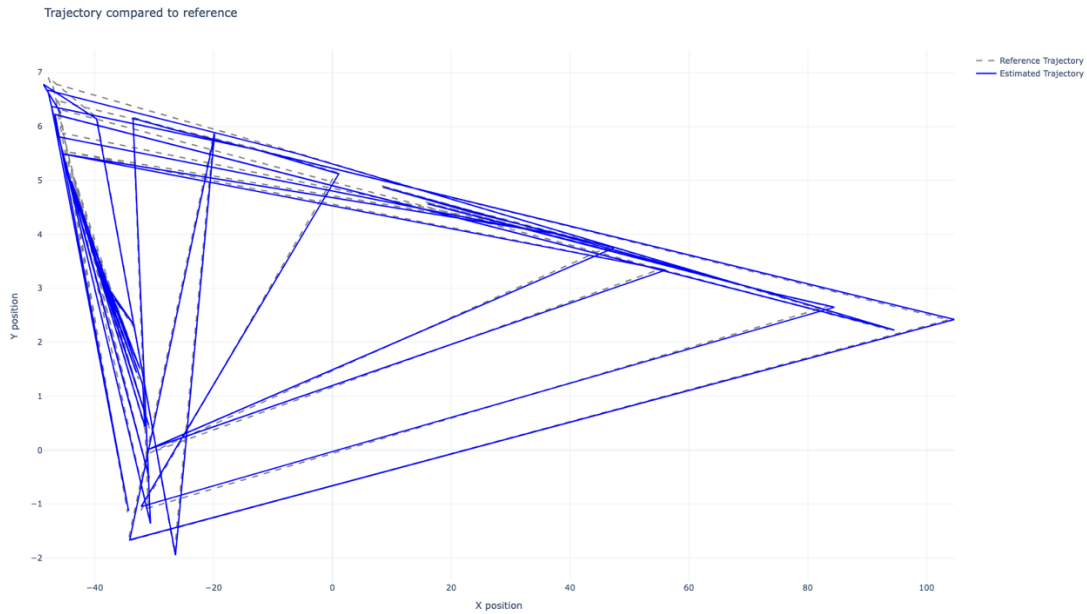


Рис. 3.7. Порівняння оптимізованої та еталонної траєкторії

Далі наведено графіки оцінки оптимізацій для даних Ladybug датасету з використанням Symforce.

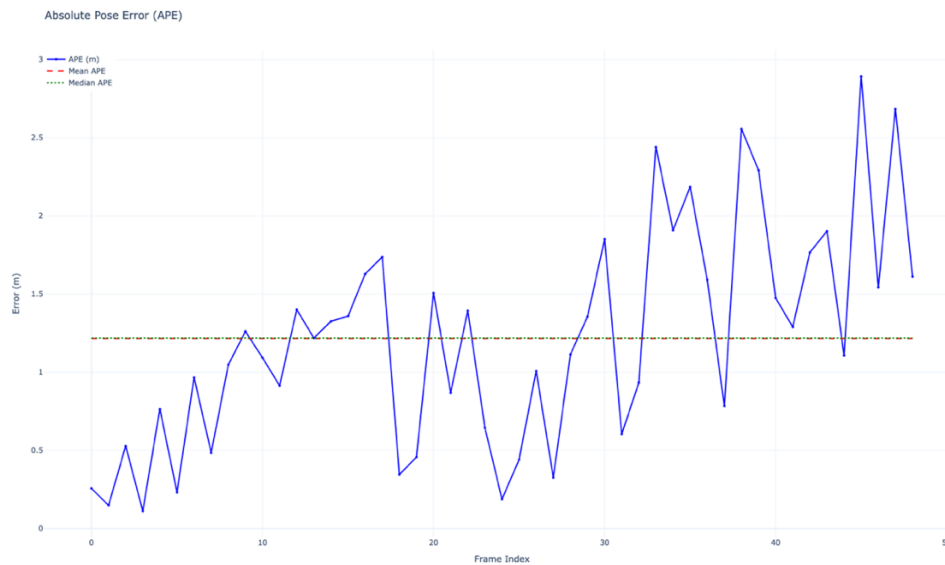


Рис. 3.8. Абсолютна помилка позиції (APE) для кожного кадру:
 синя крива показує абсолютну помилку позиції у метрах на кожному кадру;
 червона пунктирна лінія – середнє значення APE по всій траєкторії;
 зелена точкова лінія – медіанне значення APE, що є менш чутливим до викидів.

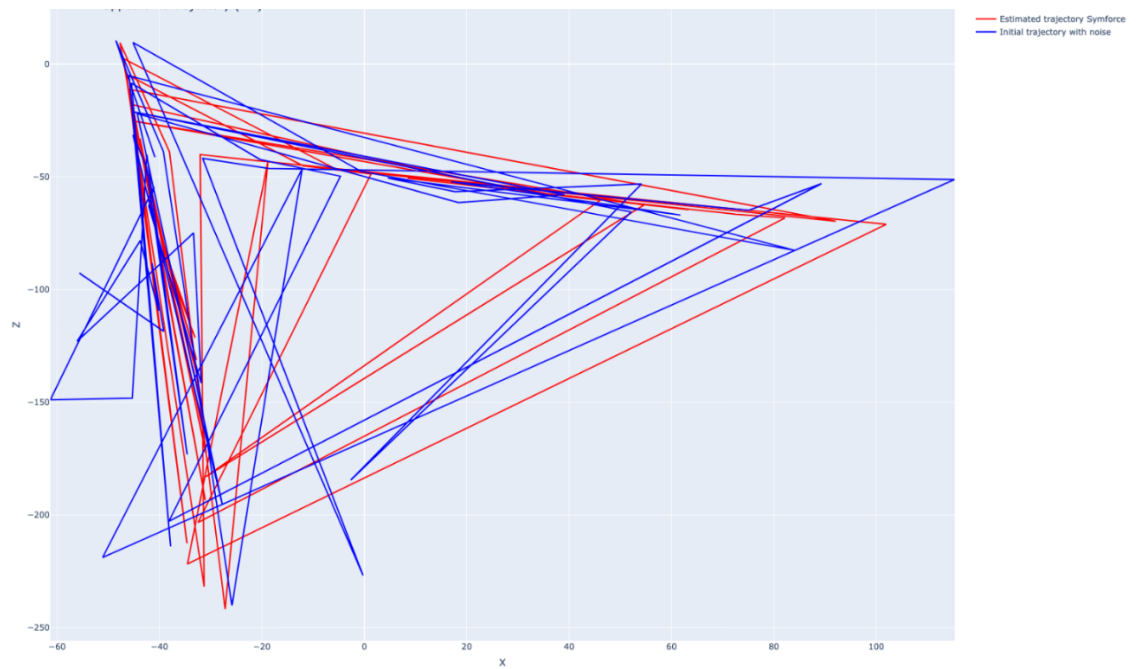


Рис. 3.9. Порівняння оптимізованої та початкової траєкторії у площині X-Z: синя лінія – початкова траєкторія з шумами перед оптимізацією; червона лінія – оптимізована траєкторія після процедури Bundle Adjustment (Symforce).

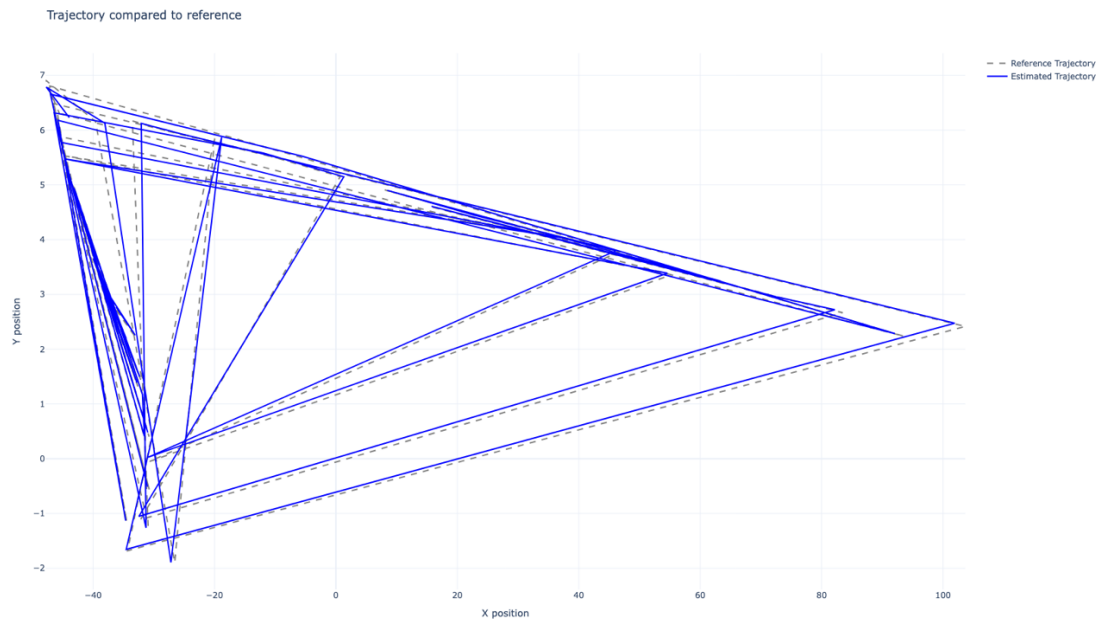


Рис. 3.10. Порівняння оптимізованої та еталонної траєкторії

І нарешті графіки оцінки оптимізацій для даних Ladybug датасету з використанням GTSAM.

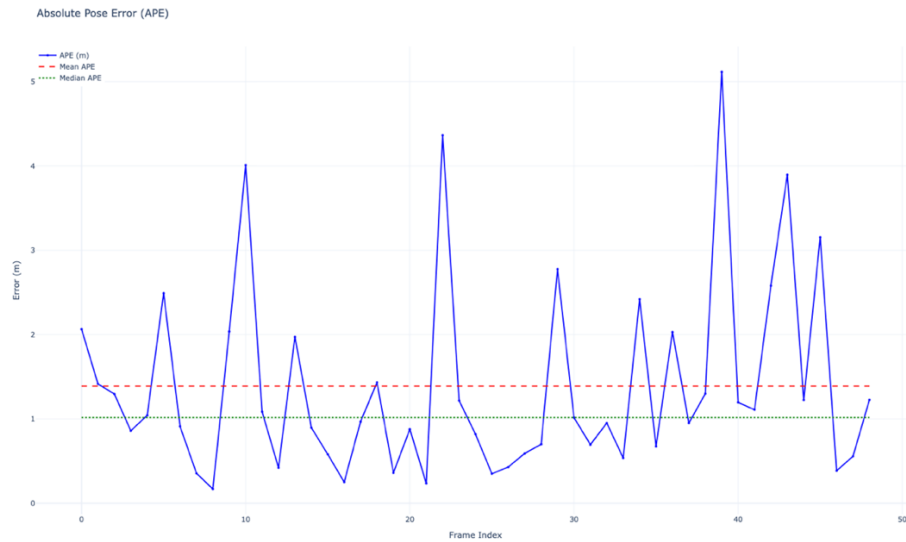


Рис. 3.11. Абсолютна помилка позиції (APE) для кожного кадру: синя крива показує абсолютну помилку позиції у метрах на кожному кадрі; червона пунктирна лінія – середнє значення APE по всій траєкторії; зелена точкова лінія – медіанне значення APE, що є менш чутливим до викидів.

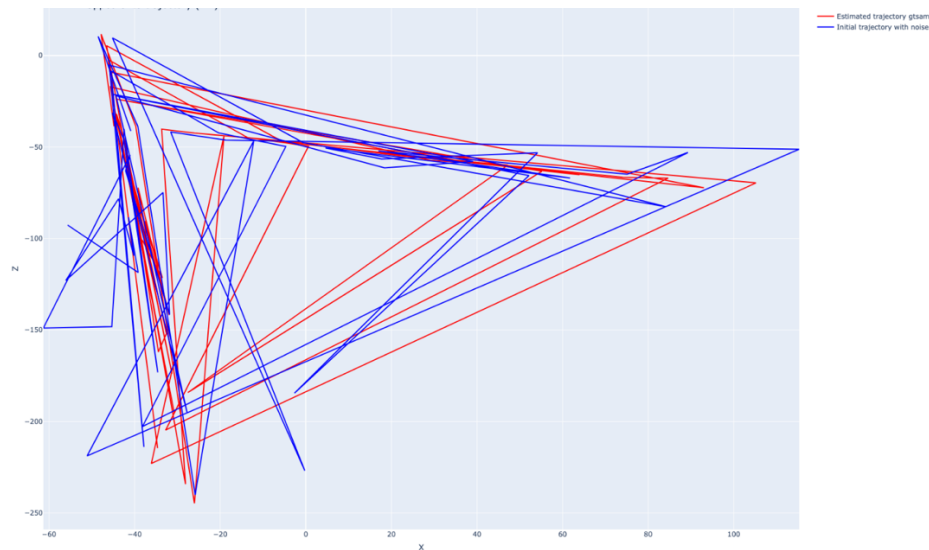


Рис. 3.12. Порівняння оптимізованої та початкової траєкторії у площині X-Z: синя лінія – початкова траєкторія з шумами перед оптимізацією; червона лінія – оптимізована траєкторія після процедури Bundle Adjustment (Symforce).

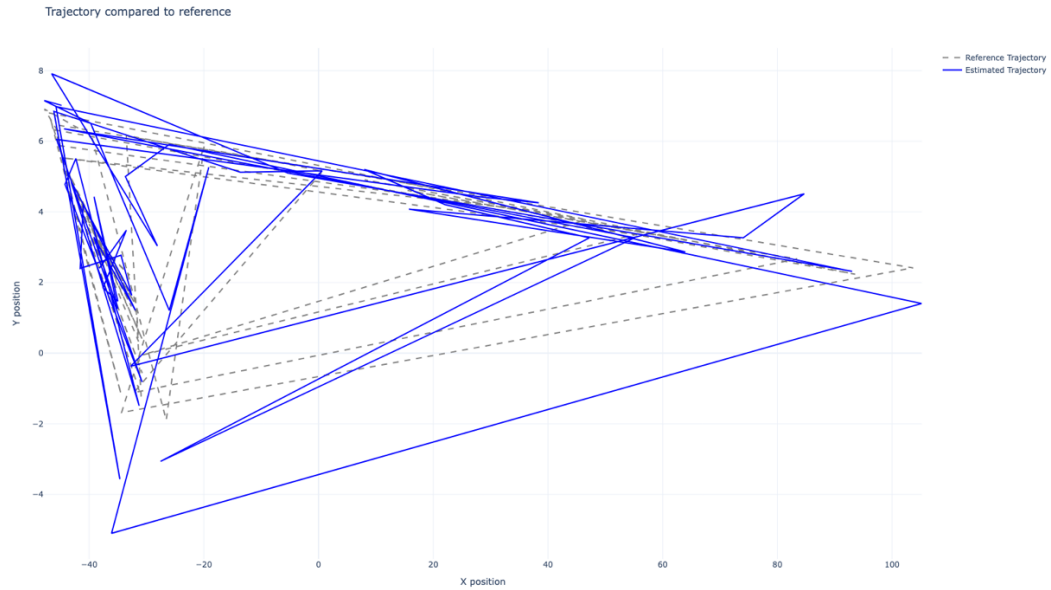


Рис. 3.13. Порівняння оптимізованої та еталонної траєкторії

Таблиця 3.1. Результати розв'язання задачі Bundle Adjustment

Бібліотека	Швидк одія (сек)	Кількіс ть ітераці й	Cost Initial	Cost Final	Абсолютна помилка позиції, (м) (мін/макс)	Кількіс ть кадрів	Кількі сть точок	Кількість спостереж ень
Ladybug								
Ceres	3.052	141	2.185e+06	8.339e+03	0.23/3.79	49	7776	31843
Symforce	2.741	105	4.339e+06	2.071e+04	0.11/2.89			
GTSAM	3.311	92	1.056e+08	1.455e+04	0.17/5.12			
Trafalgar Square								
Ceres	4.157	161	2.082e+07	1.369e+04	1.45/10.78	21	11315	36455
Symforce	2.712	95	4.118e+07	1.059e+05	0.179/5.56			
GTSAM	4.811	120	8.588e+09	3.087e+04	0.519/7.518			

Dubrovnik								
Ceres	3.046	48	2.135e+07	1.233e+04	1.276/8.03	16	22106	83718
Symforce	5.640	80	4.229e+07	3.061e+05	0.179/5.56			
GTSAM	5.833	98	4.027e+09	2.047e+04	1.026/4.599			

Порівняльний аналіз бібліотек Ceres, Symforce та GTSAM для трьох наборів даних (Ladybug, Trafalgar Square, Dubrovnik) дозволяє зробити наступні висновки [71]:

1. *Швидкодія.* Symforce показує найкращу продуктивність у наборах Ladybug та Trafalgar Square, маючи найменший час обчислень. У наборі Dubrovnik Ceres є найшвидшою, тоді як Symforce та GTSAM значно поступаються.

2. *Кількість ітерацій.* Symforce потребує менше ітерацій, ніж Ceres та GTSAM, що вказує на кращу збіжність у більшості випадків. GTSAM має найменшу кількість ітерацій у наборі Ladybug, але значно програє за швидкістю.

3. *Абсолютна помилка позиції (APE).* Symforce демонструє найменші помилки позиції у всіх наборах даних. Це вказує на більш точні результати оптимізації. GTSAM має найгіршу точність з найбільшими максимальними помилками.

4. *Зміна значення помилки (Cost Change).* Найбільше зниження значення помилки (Cost Change) спостерігається у Symforce, що підтверджує її ефективність в оптимізації.

Спираючись на результати експериментів можна зробити висновок, що Symforce є найбільш вдалим рішенням, забезпечуючи баланс між швидкістю, точністю та кількістю ітерацій. Ceres є хорошим компромісним варіантом, особливо для Dubrovnik, де він випереджає Symforce за швидкістю. GTSAM поступається конкурентам, показуючи найбільшу абсолютну помилку та значний час виконання.

Загалом, Symforce є найкращим вибором для задач оптимізації, коли важлива як швидкодія, так і точність результатів. Її будемо використовувати в подальших дослідженнях і розв'язані задачі оптимізації на фактор-графі.

3.3. Застосування нелінійної оптимізації

Сформульована у попередніх підрозділах задача оцінки стану об'єкту зводиться до знаходження такого вектору змінних X^* , який мінімізує глобальну функцію енергії (суму зважених квадратів нев'язок). Оскільки функції вимірювань (зокрема, обертання в 3D-просторі та кінематика IMU) є суттєво нелінійними, то аналітичного розв'язку даної задачі не існує. Тому застосовуються ітеративні числові методи нелінійної оптимізації.

Лінеаризація та нормальні рівняння. Для знаходження мінімуму цільової функції $J(X)$ застосовується метод Левенберга-Марквардта [76, 77], реалізований на базі інструментарію символьних обчислень та кодогенерації SymForce [74]. Основна ідея полягає в ітеративній лінеаризації функцій помилки навколо поточної оцінки стану (точки лінеаризації).

Алгоритм Левенберга-Марквардта обрано як стандарт де-факто для задач VIO через його робастність. Він адаптивно поєднує властивості методу градієнтного спуску (надійність збіжності на початкових етапах, коли похибка велика) та методу Гаусса-Ньютона (квадратична швидкість збіжності в околі мінімуму).

На кожній ітерації оптимізації виконується лінеаризація векторної функції нев'язок $r(X)$ та розв'язується модифікована система нормальних рівнянь відносно вектора приросту ΔX [76, 77]:

$$(J^T W J + \lambda D) \Delta X = -J^T W r(X) \quad (3.10)$$

де

- $J = \frac{\partial r}{\partial X}$ – розріджена матриця Якобі.
- $J^T W J$ - наближений Гессіан системи.

- $-J^T W r(X)$ - вектор градієнта.
- $W = \Sigma^{-1}$ - вагова матриця (обернена коваріація вимірювань).
- D - матриця демпфування.
- $\lambda > 0$ - коефіцієнт, що регулює "довіру" до лінійної моделі та оновлюється на кожному кроці.

Алгоритм динамічно керує параметром λ :

- Велика λ : Якщо похибка зменшується повільно або зростає, алгоритм наближається до методу градієнтного спуску. Це гарантує зменшення похибки, хоч і повільними кроками, забезпечуючи стабільність.

- Мала λ : Якщо квадратична апроксимація добре описує функцію (ми близько до мінімуму), λ зменшується, і алгоритм перетворюється на метод Гаусса-Ньютона, що забезпечує швидку (квадратичну) збіжність.

Для оцінки якості кроку використовується коефіцієнт ρ , який є відношенням реального зменшення функції вартості до передбаченого лінійною моделлю.

Реалізація засобами SymForce. Критичною проблемою при реалізації VIO є складність обчислення аналітичних похідних Якобіанів для складних функцій помилки, таких як преінтеграція IMU або візуальні фактори на багатовидах $SE(3)$. Ручне виведення похідних схильне до помилок, а класичне автоматичне диференціювання (AutoDiff) може вносити затримки під час виконання.

У даній роботі застосовано бібліотеку SymForce, яка використовує підхід символьного диференціювання з подальшою кодогенерацією. Процес оптимізації побудовано наступним чином [74]:

1. Символьний опис: Функції нев'язок $r(X)$ описуються мовою Python за допомогою символьних змінних (Symbolic Expressions). SymForce автоматично будує обчислювальний граф для виразу помилки.
2. Символьна лінеаризація: Бібліотека автоматично виконує символьне диференціювання виразів, генеруючи точні аналітичні формули для Якобіанів $J(X)$.

3. Генерація коду (Flattened C++ Generation): Символьні вирази компілюються у високопродуктивний C++ код без розгалужень та динамічного виділення пам'яті.

Такий код містить лише необхідні арифметичні операції, що забезпечує значно вищу швидкість обчислення лінеаризованої системи порівняно з підходами на базі Dual Numbers (як у Ceres Solver).

Оптимізація у дотичному просторі (Tangent Space Optimization). Специфікою задачі (3.2) є те, що змінні стану (орієнтація, пози) належать до груп Лі [78, 79]. SymForce забезпечує нативну підтримку оптимізації на многовидах через механізм StorageOps та LieGroupOps. Вектор приросту ΔX обчислюється у локальному дотичному просторі (Tangent Space) алгебри Лі, де визначені операції лінійної алгебри. Оновлення стану виконується через операцію ретракції:

$$X_{new} = Retract(X_{old}, \Delta X) \quad (3.11)$$

Для змінних $SO(3)$ це відповідає множенню на матричну експоненту: $R_{new} = X_{old} \cdot Exp(\Delta\phi)$. Це гарантує геометричну коректність траєкторії та ортогональність матриць повороту на кожному кроці алгоритму Левенберга-Марквардта.

Таким чином, використання SymForce дозволяє поєднати математичну строгість символьних обчислень із максимальною продуктивністю виконання, необхідною для роботи системи в реальному часі.

3.4. Керування ключовими кадрами та маргіналізація

Для забезпечення роботи системи в режимі реального часу необхідно обмежувати зростання розмірності вектору стану оптимізації. Оскільки відеопотік надходить із частотою 30 Гц, додавання кожного кадру до фактор-графа призвело б до експоненційного зростання обчислювальної складності. Тому в архітектурі застосовано стратегію відбору ключових кадрів (Keyframe Selection) та механізм маргіналізації (Marginalization) у ковзному вікні [80].

Критерій відбору на основі Інформаційної Матриці. Замість використання евристичних критеріїв (таких як поріг зміщення або кута повороту), які не враховують невизначеність поточної оцінки стану, у роботі запропоновано адаптивний підхід на основі теорії інформації. Рішення про додавання нового кадру $k + 1$ до множини ключових кадрів приймається на основі аналізу приросту інформації, який цей кадр вносить у систему.

Мірою невизначеності оцінки стану X є диференціальна ентропія $H(X)$, яка для багатовимірного гаусівського розподілу $N(\mu, \Sigma)$ пропорційна логарифму детермінанта коваріаційної матриці:

$$H(x) \propto \ln \det(\Sigma) = -\ln \det(\Lambda) \quad (3.12)$$

де $\Lambda = \Sigma^{-1}$ - інформаційна матриця (Information Matrix / Hessian) системи.

Критерій відбору сформульовано наступним чином: новий кадр стає ключовим, якщо він суттєво зменшує невизначеність (ентропію) системи. Це еквівалентно умові суттєвого зростання «об'єму» інформації:

$$\frac{\det(\Lambda_{k+1})}{\det(\Lambda_k)} > \gamma_{thresh} \quad (3.13)$$

де Λ_k - поточна інформаційна матриця, а Λ_{k+1} - матриця після гіпотетичного додавання нового візуального фактора. Такий підхід дозволяє адаптивно керувати щільністю ключових кадрів:

1. У зонах з низькою невизначеністю (наприклад, рух по вже відвіданій траєкторії або повільний рух) система ігнорує надлишкові кадри.
2. У зонах з високою невизначеністю (швидкі маневри, втрата текстури) частота створення ключових кадрів автоматично зростає для стабілізації трекінгу.

Маргіналізація та Доповнення Шура [80]. Для підтримки фіксованого розміру вікна оптимізації N , при додаванні нового ключового кадру найстаріший кадр x_{old} повинен бути видалений з вектора стану. Просте видалення вузла та пов'язаних з ним факторів є неприпустимим, оскільки це призвело б до втрати інформації про кореляції між станами та накопичені оцінки зміщень і швидкості.

Застосовується метод маргіналізації на основі Доповнення Шура (Schur Complement) [80]. Нехай лінеаризована система рівнянь має вигляд $H\delta x = b$. Розділимо змінні на дві групи: x_a (ті, що маргіналізуються) та x_b (ті, що залишаються). Блочна структура Гессіана має вигляд:

$$\begin{bmatrix} H_{aa} & H_{ab} \\ H_{ba} & H_{bb} \end{bmatrix} \begin{bmatrix} \delta x_a \\ \delta x_b \end{bmatrix} = \begin{bmatrix} b_a \\ b_b \end{bmatrix} \quad (3.14)$$

Застосувавши перетворення Гаусса для виключення δx_a , отримуємо модифіковану систему для змінних, що залишилися:

$$H^* \delta x_b = b^* \quad (3.15)$$

де, $H^* = H_{bb} - H_{ba}H_{aa}^{-1}H_{ab}$ та $b^* = b_b - H_{ba}H_{aa}^{-1}b_a$

Отримана матриця H^* та вектор b^* формують новий лінеаризований апіорний фактор (Linearized Prior Factor), який додається до графа. Цей фактор зберігає всю геометричну та інерціальну інформацію, накопичену до моменту часу t_{old} , забезпечуючи глобальну узгодженість траєкторії навіть при обмеженому горизонті оптимізації. Особливістю програмної реалізації є необхідність контролю розрідженості (sparsity) матриці H^* . Оскільки маргіналізація створює щільні зв'язки між усіма змінними, що були пов'язані з вилученим вузлом, у роботі застосовується відсікання слабких зв'язків для збереження ефективності.

Висновки до розділу 3

У третьому розділі дисертаційної роботи сформульовано задачу розробки та програмної реалізації бекенд-оптимізації для гібридної візуально-інерціальної одометрії. Основні результати, отримані в розділі:

1. *Критерій оптимізації.* Сформульовано задачу оцінки траєкторії БПЛА як задачу пошуку максимуму апостеріорної ймовірності (MAP) на основі фактор-графа. Розроблено структуру графа, яка забезпечує жорстко зв'язане комплексування різнорідних даних через три типи факторів: інерціальний фактор (на базі преінтеграції), щільний візуальний фактор (на базі MoGe та ICP) та

апріорний фактор. Це дозволило математично коректно об'єднати високочастотну динаміку IMU з геометричними обмеженнями сцени.

2. *Обґрунтування вибору інструментарію.* Проведено експериментальний порівняльний аналіз сучасних бібліотек нелінійної оптимізації (Ceres Solver, GTSAM, SymForce) на стандартних наборах даних BAL. Встановлено, що бібліотека SymForce забезпечує найкращий баланс між швидкодією, точністю та збіжністю. Зокрема, на датасеті Ladybug SymForce продемонструвала вищу швидкість обчислень порівняно з Ceres та GTSAM при меншій кількості ітерацій.

3. *Алгоритму оптимізації.* Запропоновано ітеративний алгоритм Левенберга-Марквардта з використанням символьного диференціювання та автоматичної кодогенерації C++ (SymForce). Застосування оптимізації на многовидах (Manifold Optimization) у дотичному просторі алгебри Лі дозволяє гарантувати геометричну коректність оновлення станів орієнтації.

4. *Стратегія керування ресурсами.* Для забезпечення роботи системи в режимі реального часу розроблено адаптивну стратегію керування ключовими кадрами на основі ентропії. Впровадження механізму маргіналізації на основі доповнення Шура дозволило обмежити розмір ковзного вікна оптимізації без втрати накопиченої інформації про попередні стани та кореляції зміщень сенсорів.

Таким чином, у розділі створено ефективний обчислювальний механізм, який перетворює локальні вимірювання сенсорів у глобально узгоджену траєкторію, що є завершальним етапом розробки системи навігації перед її комплексною експериментальною перевіркою.

РОЗДІЛ 4. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА КЕРУВАННЯ БПЛА

Керування БПЛА в реальних умовах повинно базуватися на інформації отриманої з системи візуально-інерціальної одометрії. З огляду на це необхідно дослідити обчислювальну складність її роботи та вибрати відповідне апаратне забезпечення. Після цього виконати синтез системи керування.

4.1. Апаратна архітектура та компоненти БПЛА

Ефективність функціонування розробленої системи щільної візуально-інерціальної одометрії (MoGe-VIO) критично залежить від продуктивності обчислювальної платформи та якості сенсорних даних. На відміну від класичних методів, що оперують розрідженими ознаками і можуть виконуватися на мікроконтролерах, запропонований підхід на основі нейронної мережі MoGe та нелінійної оптимізації вимагає значних обчислювальних ресурсів, зокрема апаратного прискорення тензорних операцій.

Для реалізації ресурсоємних алгоритмів комп'ютерного зору та одночасного забезпечення надійної стабілізації польоту було обрано дворівневу (ієрархічну) обчислювальну архітектуру. Такий підхід передбачає фізичне розділення задач на два рівні:

1. Низький рівень (Low-Level Control): забезпечення реального часу для стабілізації кутових швидкостей та положення, опитування інерціальних датчиків та керування силовими приводами.
2. Високий рівень (High-Level Perception & Planning): обробка візуальних даних, робота нейронної мережі, виконання алгоритмів SLAM та планування траєкторії.

Такий розподіл гарантує, що високе завантаження процесора алгоритмами машинного зору не призведе до затримок у контурі керування двигунами, що є критичним для безпеки польоту.

Високорівневий бортовий комп'ютер. У якості основного обчислювального вузла для задач візуальної одометрії та побудови карти обрано одноплатний комп'ютер Orange Pi 5 Plus. Вибір даної платформи зумовлений необхідністю локальної обробки (Edge Computing) нейромережових моделей глибини без передачі даних на наземну станцію.

Обґрунтування вибору апаратної платформи. Основою комп'ютера є система на кристалі (SoC) Rockchip RK3588, яка має гетерогенну архітектуру:

- CPU: 8-ядерний процесор (4 ядра Cortex-A76 та 4 ядра Cortex-A55), що забезпечує виконання операційної системи Ubuntu 22.04.

- NPU (Neural Processing Unit): Вбудований тензорний прискорювач продуктивністю 6 TOPS (трильйонів операцій за секунду). Це є ключовою перевагою над аналогами (наприклад, Raspberry Pi 4/5), оскільки дозволяє виконувати інференс нейромереж (зокрема, для оцінки глибини сцени) апаратно, розвантажуючи центральний процесор для задач планування траєкторії.

Технічні характеристики вузла:

- ОЗП: 16 ГБ LPDDR4x - обсяг пам'яті є критичним для зберігання локальної карти вокселів (Voxel Map) та буферів хмар точок.

- Інтерфейси вводу-виводу: Наявність апаратного інтерфейсу MIPI CSI дозволяє отримувати відеопотік з камери з мінімальною затримкою (Zero-Copy), оминаючи накладні витрати USB-шини.

- Енергоефективність: Архітектура ARM забезпечує високу продуктивність при споживанні 5–10 Вт.

На комп'ютері розгорнуто програмний стек, що включає драйвери камери та вихідний код системи SLAM з відповідними бібліотеками, що використовуються.

Низькорівневий бортовий комп'ютер. Для безпосереднього керування динамікою БПЛА використовується польотний контролер на базі мікроконтролера STM32H7 (Pixhawk 6C). Цей модуль відповідає за виконання внутрішнього контуру керування (Inner Loop Control).

Функціональне призначення.

- Збір та фільтрація даних: Опитування IMU (акселерометра та гіроскопа) з частотою до 8 кГц. Застосування фільтрів (Low-pass, Notch filters) для усунення вібраційних шумів від двигунів.

- Виконання законів керування: Реалізація алгоритмів стабілізації (PID/LQR/Fuzzy), розрахунок необхідних моментів сил та розподіл керуючих впливів на мотори (Control Allocation).

Взаємодія рівнів (Inter-process Communication): Зв'язок між Rockchip RK3588 та STM32H7 реалізовано через фізичний інтерфейс UART на швидкості 921600 бод. На програмному рівні використовується протокол MAVLink. Бортовий комп'ютер надсилає пакети типу VISION_POSITION_ESTIMATE (координати та орієнтація) та TRAJECTORY_SETPOINT (цільові точки), а контролер повертає телеметрію стану системи.

Ключовим елементом розробленої навігаційної системи є модуль камери, який забезпечує вхідний відеопотік для алгоритмів візуальної одометрії та оцінки глибини сцени. З огляду на специфіку застосування на борту БПЛА (наявність високочастотних вібрацій та динамічних маневрів), до сенсора висуваються жорсткі вимоги щодо методу зчитування зображення та швидкодії.

У якості оптичного сенсора обрано модуль Arducam OV9281, що базується на КМОН-матриці (CMOS) виробництва OmniVision. Вибір даного компонента обумовлений наступними технічними та архітектурними факторами:

Технологія глобального затвора. Це найбільш критична характеристика для роботизованих систем. Більшість стандартних камер використовують рядковий затвор (Rolling Shutter), який експонує рядки пікселів послідовно. В умовах вібрації БПЛА або швидкого руху це призводить до геометричних спотворень зображення (ефект «желе» або skew), коли прямі лінії стають викривленими. Для алгоритмів VIO, які базуються на жорсткому припущенні про геометрію сцени, такі спотворення є неприпустимими і призводять до розбіжності та втрати трекінгу.

Сенсор OV9281 використовує Global Shutter, що дозволяє експонувати всі пікселі матриці одномоментно. Це гарантує отримання геометрично коректного зображення в будь-який момент часу, незалежно від динаміки носія.

Монохромний спектр та світлочутливість. Обрана модифікація сенсора є монохромною (Monochrome). Відсутність фільтра Байєра (CFA), який необхідний для кольорових камер, надає дві суттєві переваги для задач навігації:

- *Підвищена чутливість.* Оскільки світло не поглинається кольоровими фільтрами, сенсор здатний ефективно працювати в умовах низького освітлення (наприклад, у приміщеннях, тунелях або сутінках) без значного збільшення шуму.

- *Зменшення обчислювального навантаження.* Алгоритми VIO та SLAM традиційно працюють з градієнтами яскравості, ігноруючи колір. Використання монохромного сенсора дозволяє уникнути ресурсоємної операції дебаєризації, звільняючи ресурси центрального процесора або NPU для інших задач.

Високошвидкісний інтерфейс передачі даних. Камера підключається до обчислювального модуля Orange Pi 5 Plus через інтерфейс MIPI CSI-2 (Camera Serial Interface). На відміну від USB-камер, MIPI CSI забезпечує прямий доступ до пам'яті (DMA) та має мінімальну латентність. Це дозволяє передавати відеопотік з роздільною здатністю 1280×800 пікселів із частотою оновлення до 120 кадрів за секунду. Висока частота кадрів є необхідною для надійної оцінки швидких змін кутової орієнтації та лінійної швидкості БПЛА.

Оптична система (Lens). Сенсор оснащений об'єктивом із широким кутом огляду. Широкий кут огляду (понад 100° по діагоналі) дозволяє захоплювати більшу кількість характерних точок (features) у кадрі та довше утримувати їх у полі зору при маневруванні, що підвищує робастність алгоритму одометрії.

Технічна специфікація модуля OV9281:

- Розмір матриці: 1/4 дюйма.
- Роздільна здатність: 1 МП (1280×800).
- Розмір пікселя: 3.0 мкм (великий розмір пікселя сприяє кращому

співвідношенню сигнал/шум).

- Тип затвора: Global Shutter.
- Максимальна частота кадрів: 120 fps при повній роздільній здатності.
- Інтерфейс: 2-lane MIPI CSI-2.

Таким чином, використання Arducam OV9281 дозволяє усунути апаратні джерела похибок навігації, пов'язані з динамікою польоту та освітленням, забезпечуючи алгоритми керування достовірними вхідними даними.

Вибір фізичної платформи БПЛА

Вибір фізичної платформи БПЛА обумовлений необхідністю транспортування корисного навантаження у вигляді бортового комп'ютера Orange Pi 5 Plus та масиву сенсорів, а також вимогами до стабільності польоту для коректної роботи алгоритмів візуальної одометрії.

Несуча конструкція (рама). У якості базової платформи обрано раму GEPRC Mark4 (7-inch), виготовлену з високоміцного вуглецевого волокна (3K Carbon Fiber).

Обґрунтування вибору.

1. Геометричні параметри: Розмір рами по діагоналі (Wheelbase) складає 295 мм. Це дозволяє встановити 7-дюймові пропелери, які, порівняно зі стандартними 5-дюймовими, забезпечують вищу ефективність (г/Вт) та плавність польоту. Більший момент інерції рами робить БПЛА менш чутливим до високочастотних вібрацій, що критично важливо для зменшення шуму в даних акселерометра та гіроскопа.

2. Компоновка: Рама має Н-подібну (H-TYPE) геометрію, що забезпечує достатній внутрішній об'єм для розміщення двоядерної архітектури (Flight Controller + Companion Computer) та системи охолодження.

3. Жорсткість: Використання карбонових променів товщиною 5 мм мінімізує пружні деформації конструкції під навантаженням. Жорсткість рами є необхідною умовою для коректної роботи PID/LQR-регуляторів, оскільки резонанси конструкції можуть призвести до автоколивань системи.

Електромеханічні приводи (двигуни). Рушійна система побудована на базі безколекторних двигунів постійного струму (BLDC – Brushless DC Motors) із зовнішнім ротором (Outrunner). Обрано модель типорозміру 2807 (діаметр статора 28 мм, висота 7 мм) з $KV = 1300$.

- Типорозмір 2807: Збільшений діаметр статора забезпечує вищий крутний момент на низьких обертах порівняно з поширеними моторами 2207/2306. Це дозволяє ефективно обертати важкі 7-дюймові пропелери, забезпечуючи швидку реакцію на зміни керуючого сигналу.

- Параметр KV (1300 об/В): Низьке значення KV оптимізоване для використання з акумуляторами напругою 4S–6S. Це дозволяє працювати в зоні максимального ККД двигуна при режимі зависання, зменшуючи теплові втрати та збільшуючи час автономної роботи.

Електронні регулятори ходу (ESC). Керування фазами двигунів здійснюється інтегрованим модулем 4-in-1 ESC з номінальним струмом 45А–55А.

Критичні характеристики для керування:

1. *Протокол DShot600:* використовується цифровий протокол зв'язку між польотним контролером та ESC. На відміну від аналогового PWM, DShot передає точні цифрові значення необхідної швидкості, включаючи контрольну суму (CRC), що виключає дрейф сигналу та помилки інтерпретації.

2. *Прошивка BLHeli_32:* використання 32-бітного мікроконтролера в ESC дозволяє реалізувати високу частоту комутації та апаратний ШІМ, забезпечуючи плавність обертання та можливість рекуперативного гальмування. Це дозволяє моторам не тільки швидко розганятися, але й швидко гальмувати, що покращує динамічну стабільність БПЛА при поривах вітру.

Енергетична установка (Акумуляторна батарея). Живлення системи здійснюється від літій-полімерної (Li-Po) батареї конфігурації 4S1P (номінальна напруга 14.8 В) ємністю 1500–2200 мА·год.

Вибір хімії (Li-Po vs Li-Ion). Для етапу розробки та тестування алгоритмів обрано Li-Po батареї з високим показником струмовіддачі (C-rating > 100C). Це гарантує стабільність напруги навіть при різких маневрах, коли миттєве споживання струму може сягати 80-100 А. У перспективі, для задач тривалого моніторингу, можливий перехід на Li-Ion збірки (на елементах 18650/21700), які мають вищу питому енергоємність (Вт·год/кг), але меншу струмовіддачу, що є прийнятним для режиму спокійного польоту.

Пропелери. Для забезпечення аеродинамічної ефективності в режимах статичного зависання та маневрування на низьких швидкостях обрано трилопатеві пропелери HQProp 7x3.5x3. Вибір профілю з низьким кроком (3.5 дюйма) обумовлений необхідністю максимізації питомої тяги (г/Вт) та забезпечення плавного відгуку системи на керуючі впливи. Збільшена площа диску пропелера (порівняно з 5-дюймовими аналогами) дозволяє досягти необхідної підйомної сили при менших обертах ротора (RPM), що суттєво знижує акустичний шум та вібраційне навантаження на інерціальні сенсори навігаційної системи.

Таким чином запропоновано архітектуру, що дозволяє ізолювати критичні для безпеки польоту задачі стабілізації від «важких» в обчисленні задач комп'ютерного зору, забезпечуючи надійність системи навіть у випадку тимчасових затримок у роботі нейромережі.

Структурно-функціональна схема системи керування на основі нечіткого регулятора подана на рисунку 4.1. Функціонування базується на циклічній взаємодії двох обчислювальних контурів. Зовнішній контур (Vision-based Navigation Loop) замикається через бортовий комп'ютер Orange Pi. Він працює з частотою кадрової розгортки камери (30-60 Гц). На цьому етапі вхідний відеопотік проходить через згорткову нейронну мережу для оцінки геометрії простору. На основі розрахованої карти зайнятості (Occupancy Grid) планувальник формує траєкторію, а нечіткий регулятор (Fuzzy-PD) генерує керуючі впливи у вигляді необхідних кутів орієнтації (Setpoints).

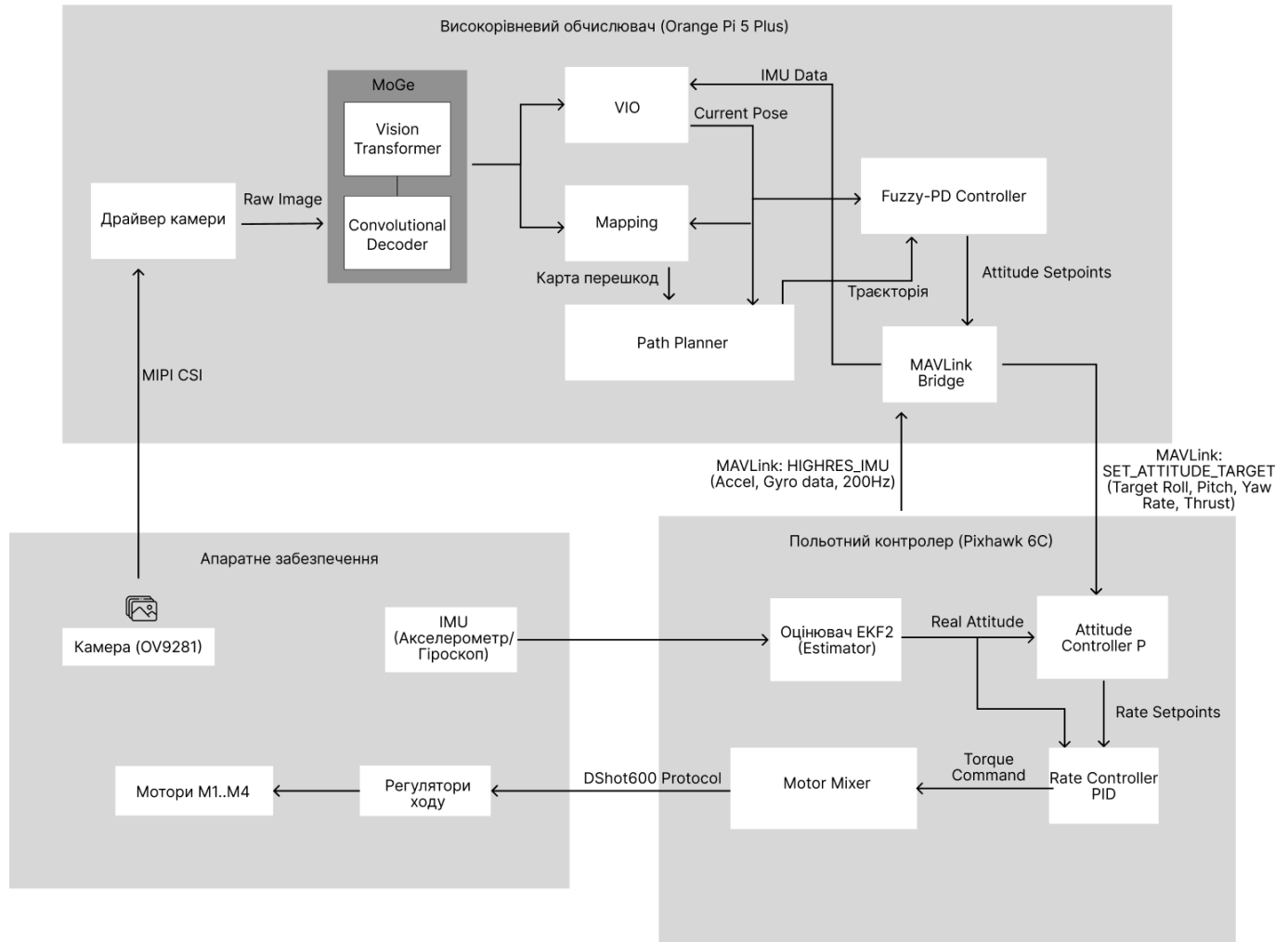


Рис. 4.1. Структурно-функціональна схема системи керування

Внутрішній контур (Stabilization Loop) реалізовано на мікроконтролері польотного контролера Pixhawk. Він працює з частотою 400 Гц. Отримавши цільові кути від зовнішнього контуру через протокол MAVLink, PID-регулятор розраховує диференційну тягу двигунів, необхідну для відпрацювання команди.

Така архітектура гарантує, що навіть у випадку затримки обробки зображення, БПЛА збереже стабільне положення в просторі завдяки швидкій реакції внутрішнього контуру.

Повний цикл керування системи виглядає наступним чином:

Початок циклу:

1. Збір даних (Sensing):

- Камера OV9281 робить знімок (Global Shutter).
- Pixhawk опитує гіроскоп та акселерометр (IMU) і надсилає дані на Orange Pi.

2. Розпізнавання (Recognition) – Orange Pi:

- Зображення потрапляє в NPU (нейропроцесор).
- Нейромережа MoGe генерує карту глибини (відстань до кожного пікселя).
- Оновлюється локальна воксельна карта (Voxel Map) – додаються нові

перешкоди.

3. Локалізація (State Estimation) – Orange Pi:

- Алгоритм VIO порівнює зміщення точок на зображенні з даними акселерометра.

- Отримуємо поточні координати БПЛА: $P_{current}(x, y, z)$.

4. Планування (Planning) – Orange Pi:

- Планувальник перевіряє: "Чи вільний шлях до цілі?"
- Якщо є перешкода → перебудовує маршрут.
- Видає наступну цільову точку: $P_{target}(x_d, y_d, z_d)$.

5. Керування позицією (High-Level Control) – Orange Pi:

- Обчислюється помилка: $e = P_{target} - P_{current}$.
- FUZZY-PD регулятор аналізує помилку (e) та швидкість її зміни (Δe).
- Фазифікація: "Помилка велика", "Швидкість мала".
- Правила: "Якщо далеко і повільно → нахилитись сильніше".
- Дефазифікація: Вихід – бажаний кут крену та тангажу (ϕ_{cmd}, θ_{cmd}).
- Команда надсилається по UART (MAVLink) на Pixhawk.

6. Стабілізація (Low-Level Control) – Pixhawk:

- Отримує $(\phi_{cmd}, \theta_{cmd})$.
- PID-регулятор порівнює бажаний кут з реальним (від гіроскопа).
- Розраховує, як швидко треба обертати БПЛА, щоб досягти кута.

7. Виконання (Actuation):

- Pixhawk надсилає сигнал DShot на 4 регулятори ESC.
- ESC змінюють струм у обмотках моторів.
- Пропелери змінюють швидкість обертання $\rightarrow$ БПЛА рухається.

Кінець циклу (повернення до пункту 1).

4.2. Математичне моделювання динаміки БПЛА

На сьогодні відома математична модель БПЛА, яка повністю враховує його поведінку і використовується в більшості наукових досліджень. Така модель, наведена наприклад в роботі [82] базується на динаміці тіла з 6 ступенями свободи (6-DOF). Динаміка задається рівняннями Ньютона–Ейлера для поступального та обертального рухів. Схема об'єкту наведена на рисунку 4.2 [81]. Чотири двигуни на кінцях променів генерують підйомні сили і крутні моменти.

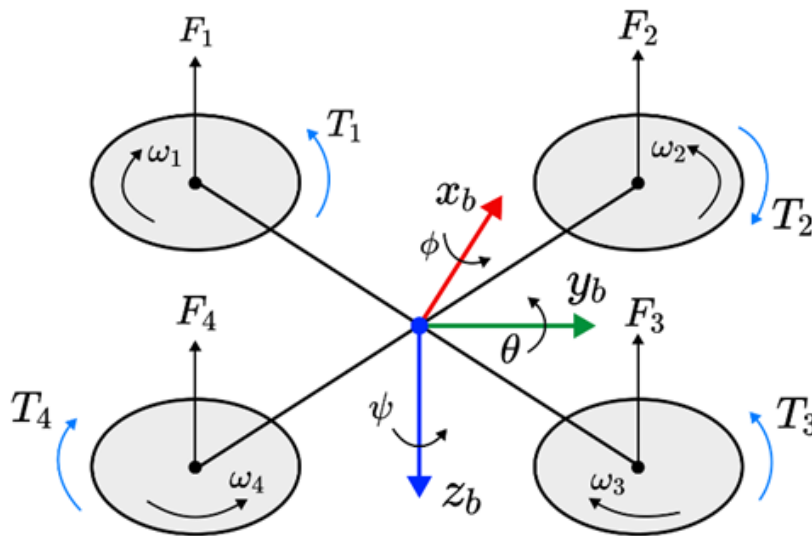


Рис 4.2. Схема моделі БПЛА

Оскільки квадрокоптер нахилиється і повертається, його власна система координат постійно змінює орієнтацію відносно земної. Матриця повороту R – це математичний інструмент, який дозволяє "переводити" вектори (наприклад, вектор сили тяги) із системи координат об'єкту в земну систему (інерціальну).

$$R = \begin{bmatrix} c(\psi)c(\theta) & c(\psi)s(\phi)s(\theta) - c(\phi)s(\psi) & s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta) \\ c(\theta)s(\psi) & c(\phi)c(\psi) + s(\phi)s(\psi)s(\theta) & c(\phi)s(\psi)s(\theta) - c(\psi)s(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\phi)c(\theta) \end{bmatrix} \quad (4.1)$$

тут $c(a) = \cos(a)$ та $s(a) = \sin(a)$

Розглянемо основні керуючі "команди", які можна подати на БПЛА, комбінуючи швидкості обертання чотирьох моторів для його керування, де K_F та K_M - це коефіцієнти, що залежать від форми та розміру пропелера [81].

Загальна тяга (U_1): Це сума сил від усіх чотирьох моторів. Вона відповідає за вертикальний рух (вгору-вниз).

$$U_1 = F_1 + F_2 + F_3 + F_4 = (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \cdot K_F \quad (4.2)$$

Момент крену (U_2): Щоб нахилити БПЛА вправо (крен), потрібно збільшити тягу лівих моторів (2 і 3) і зменшити тягу правих (1 і 4). Різниця цих сил, помножена на плече l , створює обертальний момент навколо осі X .

$$U_2 = [(F_2 + F_3) - (F_1 + F_4)] \cdot l = [(\omega_2^2 + \omega_3^2) - (\omega_1^2 + \omega_4^2)] \cdot K_F \quad (4.3)$$

Момент тангажу (U_3): Щоб нахилити БПЛА вперед, потрібно збільшити тягу задніх моторів (3 і 4) і зменшити тягу передніх (1 і 2). Це створює момент навколо осі Y .

$$U_3 = [(F_1 + F_2) - (F_3 + F_4)] \cdot l = [(\omega_1^2 + \omega_2^2) - (\omega_3^2 + \omega_4^2)] \cdot K_F \quad (4.4)$$

Момент ристання (U_4): Цей рух створюється інакше. Ротори 1 і 3 обертаються за годинниковою стрілкою, а 2 і 4 – проти. За третім законом Ньютона, якщо мотор обертає пропелер в один бік, то сам БПЛА він намагається обернути в інший (реактивний момент). У звичайному польоті ці моменти скомпенсовані. Щоб

повернути БПЛА, ми збільшуємо швидкість однієї пари моторів (напр., 1 і 3) і зменшуємо іншої (2 і 4). Виникає некомпенсований реактивний момент, і БПЛА повертається навколо своєї осі Z .

$$U_4 = (T_1 + T_3) - (T_2 + T_4) = [(\omega_1^2 + \omega_3^2) - (\omega_2^2 + \omega_4^2)] \cdot K_M \quad (4.5)$$

Зрештою, рівняння динаміки БПЛА можна записати у векторному вигляді наступним чином[81]:

$$m \cdot \begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} = [R] \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} + W = [R] \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (4.6)$$

$$\begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} I_x \cdot \ddot{\phi} \\ I_y \cdot \ddot{\theta} \\ I_z \cdot \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} I_x \cdot \dot{\phi} \\ I_y \cdot \dot{\theta} \\ I_z \cdot \dot{\psi} \end{bmatrix} \quad (4.7)$$

де: m – маса, g – гравітаційне прискорення, а I_x , I_y та I_z – головні моменти інерції. Для моделювання усі значення параметрів взято з роботи [81].

В попередніх розділах було обґрунтовано важливість стабілізації висоти БПЛА для багатьох практичних задач. Тому розглянемо процес практичної реалізації трьох поширених підходів для керування висотою на основі: PID, нечіткого (fuzzy) та лінійно-квадратичного регулятора (LQR). Кожен із них має власні переваги й обмеження. Відомо, що PID-регулятор характеризується простотою та інтуїтивністю налаштування, fuzzy-регулятор дозволяє врахувати нелінійність та невизначеність у поведінці об'єкта, а LQR забезпечує оптимальність у сенсі мінімізації квадратичного критерію якості.

Для порівняльного аналізу ефективності обрано наступні ключові показники – час перехідного процесу, величина перерегулювання, стійкість. Це дозволить сформулювати обґрунтовані висновки щодо доцільності використання того чи іншого методу керування в умовах реальних польотів.

Зазначимо, що кінцева реалізація систем керування здійснюється на технічних засобах, якими обладнані квадрокоптери (наприклад, Raspberry Pi, Nvidia Jetson, та інші) з використанням мов програмування C, Python та інших. Проте для розрахунку систем керування та дослідження їх характеристик часто використовують MATLAB та його розширення Control System Toolbox, Optimization Toolbox, Fuzzy Logic Toolbox, Simulink які фактично є промисловим стандартом для моделювання динамічних систем. Вони надають потужний інструментарій для автоматизованого налаштування регуляторів, що дозволяє значно прискорити процес розробки та знайти оптимальні параметри.

Дослідження систем керування БПЛА в MATLAB та Simulink широко використовуються в академічних дослідженнях для розробки та порівняння систем керування. Наприклад в одній з класичних робіт у галузі керування БПЛА [60] автори використали Simulink для моделювання динаміки та реалізації PID- та LQR-контролерів для стабілізації кутового положення та висоти. У роботі [82] автори порівнюють ефективність класичного ПІД-регулятора та гібридного нечіткого ПІД-регулятора. Уся розробка, моделювання та порівняльний аналіз були виконані в середовищі MATLAB/Simulink з використанням Fuzzy Logic Toolbox для проектування нечіткого контролера. У статті [83] запропоновано ПІД-регулятор, коефіцієнти якого динамічно налаштовуються за допомогою нечіткої логіки. MATLAB використовувався як для моделювання динаміки БПЛА, так і для імплементації та симуляції роботи запропонованого гібридного контролера, що довело його переваги над звичайним ПІД.

Враховуючи наведені факти та кращі практики в роботі прийнято рішення розрахунок та дослідження систем керування здійснювати з використання MATLAB.

4.3. Розробка систем керування на основі PID, Fuzzy Logic, LQR

Розроблена імітаційна модель БПЛА включає вплив вітрових збурень, що дозволяє відтворити умови, наближені до реальних. Такий підхід є необхідним для оцінки ефективності алгоритмів керування, адже у реальних польотах БПЛА піддається випадковим та непередбачуваним впливам зовнішнього середовища. На основі цієї концепції модель може бути розширена іншими реалістичними ефектами, зокрема шумами сенсорних вимірювань чи флуктуаціями вихідної потужності двигунів [84]. Схеми моделювання системи керування наведені в додатку Б.

У моделі вітрове збурення реалізоване на основі таких параметрів:

- WindLevel – інтенсивність вітру, що визначає силу зовнішнього впливу;
- Noise power – рівень шуму, який надає сигналу випадковий характер;
- Wind sample time – дискретність зміни вітру, що визначає його динаміку.

Ці налаштування дозволяють аналізувати поведінку БПЛА за різних умов та порівнювати якісь різних регуляторів. Наприклад, у моделюванні вітер вмикається після $t = 3$ с, а його сила для даної системи змінюється в діапазоні $[-15...15]$, як показано на Рис. 4.3 [84].

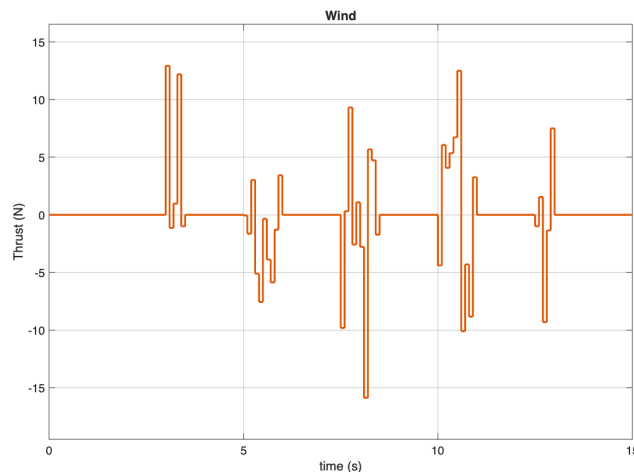


Рис. 4.3. Вітрове збурення

В моделі прописано заданий рівень висоти $Z = 1$ м і досліджується його забезпечення при використанні різних контролерів. Модель симуляції, що базується на роботі [85] виглядає наступним чином:

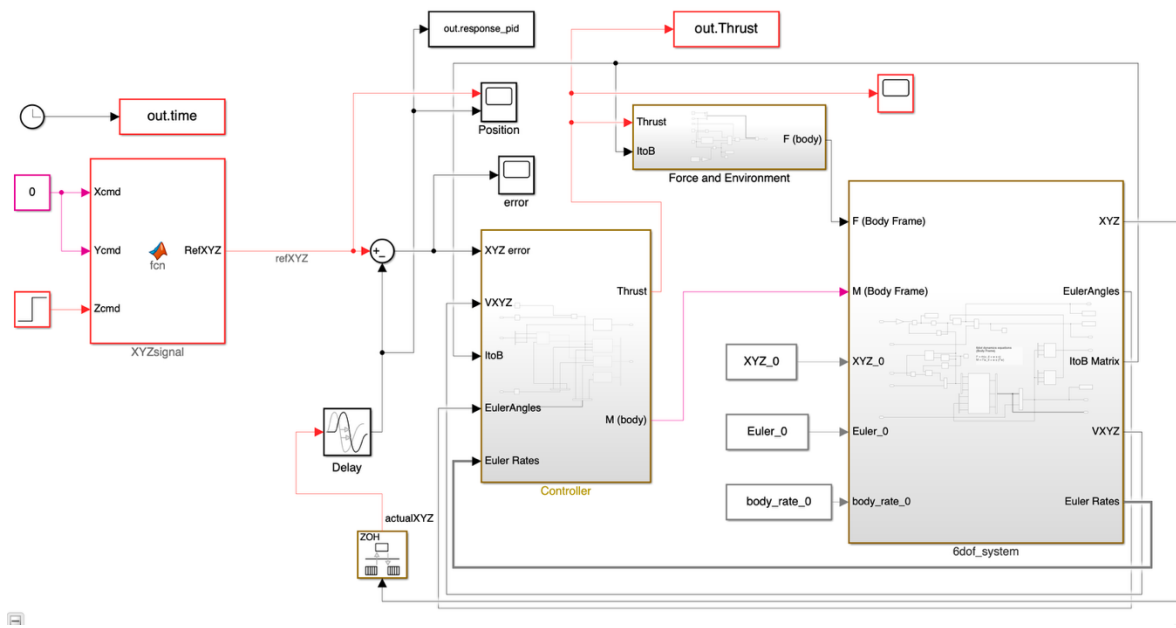


Рис. 4.4. Simulink-модель системи керування БПЛА

Це класична, замкнута система керування, де блок XYZsignal формує команди по осях X, Y, Z . У конфігурації X та $Y = 0$, а по осі Z (висота) подається сигнал 1 м. З моделі БПЛА (6dof_system) повертається вектор actualXYZ (поточне положення БПЛА). На суматорі віднімаються $\text{refXYZ} - \text{actualXYZ} \rightarrow$ отримуємо XYZ error, тобто відхилення. Ця похибка йде на контролер [79].

Для імітації реальних умов функціонування систем візуально-інерціальної одометрії у канал зворотного зв'язку між виходом моделі та суматором інтегровано блок Transport Delay. Він вносить часове запізнення тривалістю 50 мс, що відповідає типовим обчислювальним затримкам при обробці відеоданих на бортових комп'ютерах. Таким чином, на суматорі виконується віднімання $\text{refXYZ} - \text{actualXYZ_delayed}$, у результаті чого формується сигнал XYZ error. Важливою особливістю такої конфігурації є те, що похибка обчислюється на основі «минулого» стану системи, що створює фазове запізнення в контурі керування.

Обґрунтування вибору місця та доцільності впровадження затримки пояснюється такими факторами:

- *Реалістичність сенсорних даних*: у реальних системах керування дані про позиціонування від VIO-систем надходять до контролера із запізненням (у нашому випадку встановлено затримку 50 мс), що зумовлено високою обчислювальною складністю обробки відеопотоку та інтегрування даних з інерціальних датчиків.

- *Вплив на динамічну стійкість*: розміщення затримки саме у колі зворотного зв'язку, а не на вході задаючого сигналу XYZsignal, є критично важливим для аналізу стійкості. Це створює додаткове фазове запізнення $\Delta\phi = -\omega \cdot L$ (де ω - частота системи, L - час затримки), що безпосередньо впливає на запас стійкості за фазою (Phase Margin).

- *Адаптація регулятора*: наявність такої затримки дозволяє оцінити здатність PID-регулятора стабілізувати об'єкт у ситуації, коли керуючий вплив формується на основі «минулого» стану системи. Це вимагає більш складного налаштування коефіцієнтів, оскільки надмірна агресивність регулятора (високі значення P та D) при наявності лагу призводить до виникнення автоколивань та втрати керованості, що чітко простежується на графіках перехідних процесів.

Блок Controller відповідно до [85] обчислює необхідні керуючі дії. На основі XYZ error, швидкостей (VXYZ), кутів Ейлера та їхніх похідних. Формує два основні виходи: Thrust (тяга вздовж осі Z, основна сила підйому) та M (body) (момент у корпусній системі координат для стабілізації положення). Оскільки розглядається стабілізація заданої висоти, то в розглядається Thrust і його керування.

PID регулятор

Класичний підхід до керування висотою базується на формуванні команди тяги (Thrust) через лінійну комбінацію похибки за віссю Z, її інтеграла та похідної. Цей метод є базовим у системах автоматичного керування завдяки прозорості структури та відносній простоті реалізації.

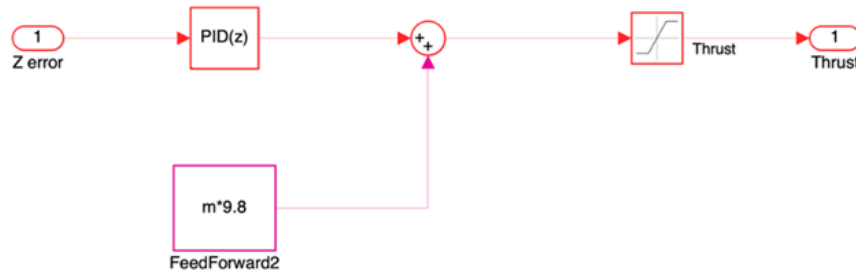


Рис. 4.5. Структура PID-регулятора

У даній моделі PID-контролер налаштовувався з використанням вбудованого інструментарію Simulink PID Tuner з обов'язковим урахуванням блоку Transport Delay (50 мс) у контурі зворотного зв'язку. Наявність затримки внесла суттєві обмеження у процес оптимізації: для забезпечення стійкості системи було обрано компромісний варіант із часом перехідного процесу приблизно 0.25 с. Проте наявність часового лагу призвела до суттєвого зниження запасу стійкості за фазою [84].

Результати моделювання показали, що попри задовільну роботу в ідеальних умовах, PID-регулятор виявляється неспроможним забезпечити стабільність при поєднанні сенсорної затримки та динамічних зовнішніх впливів. Зокрема, в умовах інтенсивних вітрових збурень ($WindLevel = 10$) система переходить у стан автоколивань. Через те, що регулятор отримує дані про позицію із запізненням, команда на компенсацію пориву вітру подається тоді, коли стан системи вже змінився. Це призводить до ефекту «перерегулювання», за якого висота БПЛА коливається у критичних межах – від перевищення заданого рівня понад 2 м до падіння у від'ємні значення [84].

Таким чином, класичний PID-алгоритм без додаткових блоків прогнозування (наприклад, предиктора Сміта) демонструє низьку робастність до затримок, характерних для VIO-навігації, що обґрунтовує необхідність пошуку більш досконалих методів керування [84].

У результаті налаштування було встановлено наступні параметри регулятора, що забезпечують стабільність у штатному режимі [84]:

Таблиця 4.1. Параметри налаштованого PID-регулятора (Z-канал)

Параметр	Символ	Значення
Пропорційний коефіцієнт	P	2.88
Інтегральний коефіцієнт	I	0.86
Диференціальний коефіцієнт	D	1.43
Коефіцієнт фільтра похідної	N	40.843
Тип фільтрації	-	Дискретний (Forward Euler)

Нечіткий (Fuzzy) регулятор

Для підтримання заданої висоти БПЛА розглянемо PD-регулятор. Для цього розроблено нечітку модель для розрахунку коефіцієнтів P та D. Розглянуто вдосконалений нечіткий PD-контролер з коефіцієнтами масштабування на основі евристичної оптимізації (fuzzy-PD) для керування системою до бажаної динаміки.

Для налаштування Fuzzy-регулятора були обрані стандартні дві вхідні змінні: похибка висоти (error) та швидкість її зміни, тобто похідна (error_rate). Обидві змінні нормалізовано у діапазоні $[-1;1]$. Похибку висоти описано за допомогою 3 функцій належності з гаусовим розподілом: low (низька) з центром в -1 і стандартним відхиленням 0.3536; median (середня) з центром в 0 і стандартним відхиленням 0.3536; high (висока): з центром в 1 і стандартним відхиленням 0.3536. Змінна похідної похибки висоти має аналогічні 3 функції належності з стандартним відхиленням 0.3539 [84].

З використанням нечіткої моделі розраховується два значення K_1 та K_2 , що є коефіцієнтами для PD-регулятора. K_1 відіграє роль пропорційного коефіцієнта (K_p). Він відповідає за силу реакції на поточну похибку. K_2 відіграє роль диференційного коефіцієнта (K_d). Він відповідає за реакцію на швидкість зміни похибки. Обидві

змінні також нормалізовано у діапазоні $[-1;1]$ і аналогічно описуються трьома функціями належності [86] із центрами -1, 0, 1 та відхиленням 0.4247 і 0.4246 відповідно. Система використовує 9 правил, які повністю охоплюють усі можливі комбінації вхідних термів (3 терми для *error* × 3 терми для *error_rate*). Правила задають логіку контролера [84].

Таблиця. 4.2. База правил нечіткої моделі

	Швидкість зміни похибки низька	Швидкість зміни похибки середня	Швидкість зміни похибки висока
Похибка низька	K_1 низький, K_2 низький	K_1 низький, K_2 середній	K_1 низький, K_2 високий
Похибка середня	K_1 середній, K_2 низький	K_1 середній, K_2 середній	K_1 середній, K_2 високий
Похибка висока	K_1 низький, K_2 низький	K_1 низький, K_2 середній	K_1 низький, K_2 високий

Дефазифікація здійснювалась за методом центроїда. Після того, як усі правила спрацювали і їхні результати були об'єднані (агреговані) в одну фінальну нечітку множину для кожного виходу, метод знаходить "центр мас" цієї фігури. Горизонтальна координата цього центру і є кінцевим, чітким числовим значенням на виході контролера (K_1 та K_2). Це найпоширеніший і збалансований метод дефазифікації. Остаточне налаштування проводилось експериментально в середовищі симуляції шляхом корекції форм функцій належності та вагових коефіцієнтів.

Ключовою перевагою розробленої архітектури є її стійкість до часової затримки у 50 мс, що імітує роботу VIO-системи. Оскільки вхідні сигнали *error* та *error_rate* формуються на основі запізнілих даних, класичні лінійні регулятори часто демонструють нестабільність через некоректне обчислення диференціальної складової. Fuzzy-логіка нівелює цей ефект за рахунок нелінійного відображення

входів на виходи: база правил дозволяє "пом'якшувати" реакцію (K_1) та посилювати демпфування (K_2) саме в ті моменти, коли затримка загрожує виникненням автоколивань. Дефазифікація за методом центроїда забезпечує плавну зміну коефіцієнтів, що дозволяє системі уникати різких стрибків тяги (Thrust), характерних для звичайного PID-регулятора при роботі з зашумленими та запізніленими сигналами.

Виходи K_1 та K_2 з нечіткого блоку використовуються для налаштування класичного PD-регулятора в реальному часі. Ця архітектура є потужним гібридним підходом. Замість використання фіксованих коефіцієнтів P та D, які є компромісом між швидкістю та стабільністю, використовується нечітка модель для їх динамічного підбору. Такий підхід дозволяє отримати адаптивну, робастну та високоефективну систему керування висотою [84].

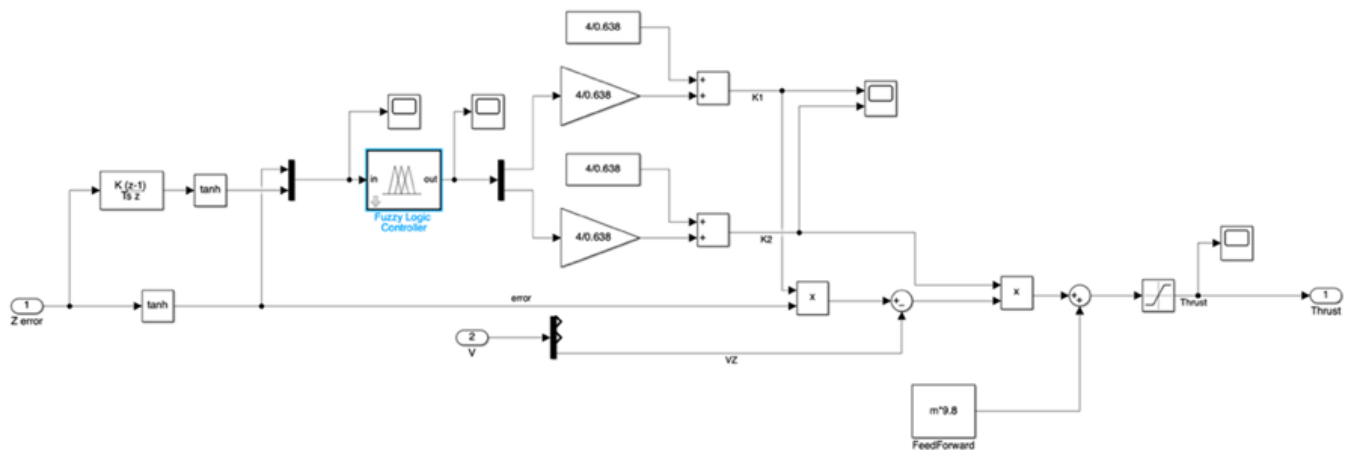


Рис. 4.6. Структура Fuzzy-регулятора

LQR-регулятор

Лінійно-квадратичний регулятор (LQR) – це метод оптимального керування, що використовується для систем, які можна описати лінійними диференціальними рівняннями у просторі станів. Головна мета LQR – знайти такий закон керування, який мінімізує функцію вартості J протягом нескінченного проміжку часу.

$$J = \int_0^{\infty} [x^T(t)Qx(t) + u(t)Ru(t)]dt \quad (4.8)$$

Ця функція вартості складається з двох частин:

- *похибка стану*: штраф за відхилення вектора стану системи $x(t)$ від нуля (тобто від бажаної робочої точки).

- *витрати на керування*: штраф за використання керуючого сигналу $u(t)$.

Тут Q та R – це вагові матриці, які визначають пріоритети системи. Закон керування має вигляд зворотного зв'язку за станом:

$$u(t) = -Kx(t) \quad (4.9)$$

де K – це матриця коефіцієнтів підсилення, яка обчислюється для мінімізації J . Вона знаходиться шляхом розв'язання алгебраїчного рівняння Ріккати.

$$A^T P + P A - P B R^{-1} B^T P + Q = 0 \quad (4.10)$$

Оскільки LQR розроблений для лінійних систем, а динаміка БПЛА є нелінійною, першим кроком є лінеаризація моделі. Цей процес полягає в апроксимації нелінійної системи лінійною моделлю навколо певної робочої точки. У нашому випадку такою точкою є режим зависання (hovering).

Процес лінеаризації автоматизовано за допомогою інструментів *Simulink Control Design*:

- *визначення робочої точки*: за допомогою *operspec* та *findop* визначається стан системи при зависанні на висоті $Z = 1$.

- *отримання лінійної моделі*: функція *linearize* обчислює матриці простору станів A та B для лінеаризованої моделі $\dot{x} = Ax + Bu$ навколо цієї робочої точки.

Критично важливою особливістю синтезу даного регулятора є те, що блок *Transport Delay* було включено безпосередньо в контур лінеаризації. Це дозволило функції *linearize* врахувати фазовий зсув від затримки у 50 мс через апроксимацію Паде, що відобразилося у структурі отриманої матриці стану A .

Вектор стану x складається з 12 елементів.

$$x = [\phi, \theta, \psi, X, Y, Z, p, q, r, u, v, w]^T \quad (4.11)$$

Налаштування LQR полягає у виборі вагових матриць Q та R . Ці матриці є діагональними, і їх елементи визначають "важливість" кожного стану та керуючого

входу. Матриця Q (State-Weighting Matrix): Визначає, наскільки сильно регулятор "штрафує" за відхилення кожного зі станів від нуля. Чим більше значення елемента на діагоналі Q , тим швидше і агресивніше контролер намагатиметься повернути відповідний стан до нуля. Для Q використано наступні вагові коефіцієнти [84]:

Таблиця 4.3. Вагові коефіцієнтів для матриці Q

Група станів	Вагові коефіцієнти
Кути (roll, pitch, yaw)	[25, 25, 5]
Позиції (X, Y, Z)	[65, 65, 120]
Кутові швидкості	[10, 10, 10]
Лінійні швидкості	[8, 8, 12]

Матриця Q виглядає наступним чином [83]:

$$Q = \text{diag}([25, 25, 8, 65, 65, 120, 10, 10, 10, 8, 8, 12]) \quad (4.12)$$

Матриця R (Control-Weighting Matrix): Визначає вартість використання керуючих сигналів. Чим більше значення елемента R , тим менш "енергійним" буде відповідний керуючий сигнал, що призводить до плавнішої, але повільнішої реакції та менших енерговитрат. Для компенсації затримки у 50 мс значення в R були суттєво збільшені, щоб уникнути різких стрибків тяги, які на основі застарілих даних VIO могли б розхитати апарат [84].

Таблиця 4.4. Вагові коефіцієнтів для матриці R

Керуючий сигнал	Значення
Загальна тяга	1.2
Момент по roll	0.9
Момент по pitch	0.9
Момент по yaw	0.9

Матриця R виглядає наступним чином [84]:

$$R = \text{diag}([1.2, 0.9, 0.9, 0.9]); \quad (4.13)$$

Таким чином, тюнінг LQR – це пошук компромісу між швидкістю реакції (високі значення в Q) та витратами на керування (високі значення в R). Після вибору Q та R матриця підсилення K розраховується автоматично за допомогою функції `lqr` в MATLAB:

$$K = \text{lqr}(\text{sys.A}, \text{sys.B}, Q, R); \quad (4.14)$$

Ця матриця K і є кінцевим результатом синтезу LQR-регулятора.

Результати порівняння систем керування, що розглядались наведені на рисунку 4.7. та таблиці 4.5.

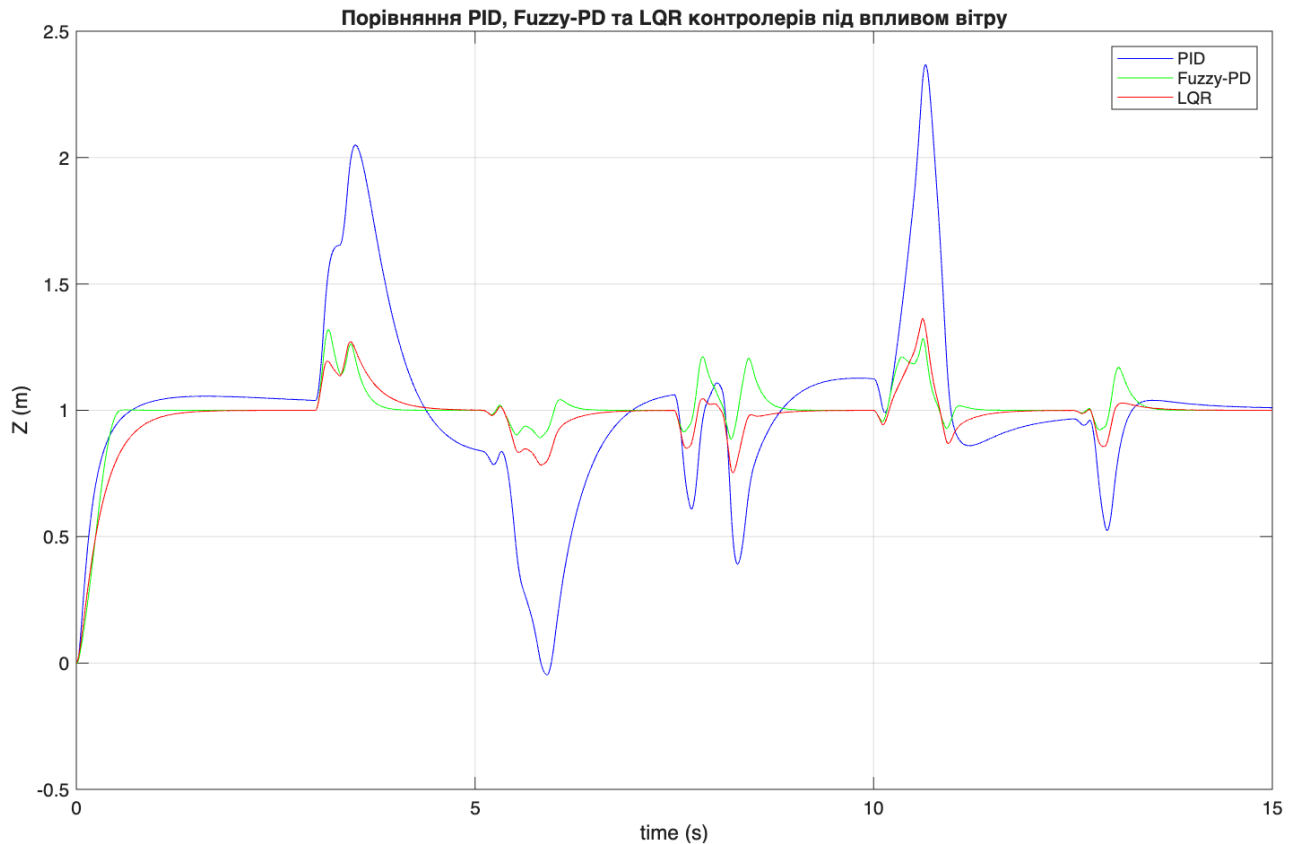


Рис. 4.7. Перехідні характеристики системи з PID, Fuzzy-PD та LQR контролерами на сигнал висоти в умовах зовнішніх збурень (вітер).

Таблиця 4.5. Порівняльна таблиця статистичних результатів роботи контролерів

	PID	Fuzzy-PD	LQR
Вихід на усталений режим 1м, без збурення (s)	5.35	1.4	2.1
Перерегулювання, без збурення (%)	5.5	0.2	0.34
RMSE зі збуренням	0.3509	0.1286	0.1483

Висновок. Проведене порівняння регуляторів для стабілізації висоти за однакових умов (вітер $WindLevel=10$ та затримка зворотного зв'язку 50 мс) показало, що Fuzzy-PD контролер демонструє найкращу якість: найшвидший вихід на усталений режим без збурень (1.4 с), мінімальне перерегулювання (0.2 %). В умовах затримки VIO цей контролер зміг адаптувати коефіцієнти так, щоб компенсувати фазовий зсув, що призвело до мінімальної помилки при збуренні ($RMSE \approx 0.1286$) [84].

LQR-регулятор виявився потужною альтернативою, здатною ефективно працювати з запізненими даними завдяки включенню затримки в математичну модель при лінеаризації. Хоча він дещо поступається нечіткій логіці за швидкістю (2.1 с), він забезпечує високу передбачуваність системи та стабільність, яка недосяжна для класичних лінійних методів без врахування стану системи. Його показник $RMSE \approx 0.1483$ підтверджує ефективність матричного підходу для збалансованого керування [84].

PID-регулятор виявився найбільш чутливим до фактору затримки. Необхідність забезпечення стійкості в умовах лагу 50 мс змусила суттєво знизити агресивність налаштувань, що призвело до збільшення часу встановлення до 5.35 с. При виникненні вітрових збурень PID-алгоритм не встигає вчасно реагувати на зміни через запізнений фідбек, що спричиняє значні коливання ($RMSE \approx 0.3509$) та перерегулювання (5.5 %) [84].

Практично це означає:

- Для систем із високою затримкою сенсорів (VIO) доцільно впроваджувати Fuzzy-PD, оскільки він найкраще справляється з "невизначеністю", яку вносить часовий лаг у динаміку польоту.
- Для складних багатозв'язних систем рекомендується LQR, як метод, що дозволяє системно врахувати затримку на етапі синтезу матриці посилення K , забезпечуючи стабільний компроміс між енерговитратами та точністю.
- Використання класичного ПІД-регулювання в реальних БПЛА із затримками понад 30–40 мс є ризикованим і вимагає значного зниження динамічних характеристик апарата для збереження його цілісності.

Висновки до розділу 4

У четвертому розділі проведено комплексне дослідження архітектури та систем керування безпілотним літальним апаратом типу квадрокоптер, що інтегрує сучасні методи комп'ютерного зору з алгоритмами оптимального керування. На основі проведеної роботи сформульовано наступні висновки:

1. Обґрунтовано ієрархічну апаратну архітектуру системи. Доведено, що для реалізації методів щільної візуальної одометрії (MoGe-VIO) на основі нейронної мережі MoGe необхідний розподіл обчислювальних задач. Використання одноплатного комп'ютера Orange Pi 5 Plus з вбудованим нейропроцесором (NPU 6 TOPS) дозволяє виконувати ресурсомісткий інференс нейромереж безпосередньо на борту БПЛА (Edge Computing). Це усуває затримки, пов'язані з передачею відеопотоку на наземну станцію, та забезпечує автономність системи. Проте встановлено, що навіть при бортовій обробці затримка передачі даних до польотного контролера становить близько 50 мс. Розділення на зовнішній контур навігації (High-Level) та внутрішній контур стабілізації (Low-Level на базі Pixhawk) гарантує критичну живучість апарата: навіть при високому навантаженні

візуального процесора, стабільність польоту підтримується в режимі жорсткого реального часу.

2. Реалізовано відому математичну модель динаміки БПЛА як об'єкта з 6 ступенями свободи (6-DOF). Використання рівнянь Ньютона–Ейлера дозволило адекватно описати взаємозв'язок між силами тяги двигунів, зовнішніми моментами та просторовим рухом апарата. Модель враховує гіроскопічні ефекти та нелінійність динаміки, що стало фундаментом для подальшого синтезу регуляторів та проведення імітаційного моделювання у середовищі, наближеному до реального.

3. Розроблено та реалізовано тестування реакції моделі до зовнішніх збурень. Впровадження в імітаційну модель блоку вітрового навантаження з керованими параметрами (WindLevel, Noise power) дозволило провести стрес-тестування алгоритмів керування. Встановлено, що здатність системи компенсувати випадкові збурення в діапазоні $[-15...15]$ одиниць сили є ключовим показником робастності для експлуатації БПЛА у відкритому просторі.

4. Проведено порівняльний систем керування висотою на основі: PID, LQR та Fuzzy регуляторів.

- Fuzzy-PD регулятор продемонстрував найкращі показники точності. Завдяки використанню апарату нечіткої логіки та гаусових функцій належності для обробки похибки та швидкості її зміни, вдалося досягти найшвидшого часу виходу на заданий рівень (1.4 с) та мінімального перерегулювання (0.2 %).
- LQR-регулятор підтвердив свою ефективність як метод оптимального керування. Хоча його швидкодія (2.1 с) є дещо нижчою за нечіткий аналог, він забезпечує високу стабільність, що важливо для передбачуваного маневрування.
- Класичний ПІД-регулятор, попри простоту реалізації, виявився найменш ефективним в умовах вітрових збурень ($RMSE \approx 0.3509$), що обґрунтовує необхідність переходу до більш складних методів керування для сучасних БПЛА.

5. Визначено сфери практичного застосування запропонованих рішень. Для місій, що вимагають екстремальної маневреності та стійкості до поривів вітру (наприклад, інспекція промислових об'єктів або політ у міській забудові), рекомендовано використання Fuzzy-PD регулятора. Для задач прецизійного картографування та автономної навігації, де пріоритетом є плавність траєкторії та системна робастність на основі аналітичної моделі, доцільним є вибір LQR-регулятора.

6. Доведено працездатність концепції MoGe-VIO. Інтеграція апаратного прискорення NPU та оптимального керування створює передумови для реалізації повноцінного SLAM-фронтенду, який не просто відстежує розріджені точки, а будує щільну карту навколишнього середовища в реальному часі, забезпечуючи високу якість траєкторного керування навіть у складних динамічних умовах.

РОЗДІЛ 5. ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА ВАЛІДАЦІЯ СИСТЕМИ ВІЗУАЛЬНО-ІНЕРЦІАЛЬНОЇ ОДОМЕТРІЇ

Критичним етапом верифікації розробленої системи є експериментальна оцінка її точності, стабільності та обчислювальної ефективності. Оскільки запропонований підхід базується на комплексному поєднанні «важких» алгоритмів інференсу нейромережі MoGe та ітеративної геометричної реєстрації ICP, питання апаратної реалізації та методології тестування набувають особливої ваги. У цьому розділі досліджено апаратну платформу, конфігурацію параметрів алгоритму та методику порівняльного аналізу з існуючими аналогами.

5.1. Апаратна реалізація та обчислювальні ресурси

Експериментальна перевірка та запуск системи здійснювалися на обчислювальній платформі Apple Silicon на базі чипа M3.

Для виконання інференсу (прямого проходу) нейронної мережі MoGe було задіяно бекенд Metal Performance Shaders (MPS). Це дозволило перенести обчислювальне навантаження бібліотеки PyTorch безпосередньо на графічні ядра (GPU) чипа M3.

Разом з тим, слід зазначити, що архітектура Vision Transformer (ViT), яка лежить в основі геометричного ядра MoGe, є надзвичайно вимогливою до обчислювальних ресурсів. Специфіка операцій Self-Attention вимагає масованих матричних множень, які на даний момент найкраще оптимізовані для тензорних ядер архітектури NVIDIA CUDA. Через програмні обмеження пропускної здатності MPS для певних специфічних операцій трансформерів, поточна реалізація системи на чипі M3 досягає частоти оновлення одометрії в діапазоні 15–20 Гц. Хоча такої частоти достатньо для навігації БПЛА на низьких та середніх швидкостях, для високодинамічних застосувань (наприклад, гоночні БПЛА або політ в умовах сильної турбулентності) рекомендовано перехід на спеціалізовані платформи з підтримкою CUDA, такі як NVIDIA Jetson Orin або компактні дискретні GPU серії

RTX. Використання бібліотек низькорівневої оптимізації, таких як TensorRT та cuBLAS, дозволило б виконати квантування ваг моделі (FP16 або INT8) та злиття шарів (layer fusion), що потенційно подвоїло б частоту кадрів до 30+ Гц. Тим не менш, отримані результати на МЗ підтверджують принципову можливість розгортання запропонованого підходу «MoGe-VIO» на енергоефективних Edge-пристроях без використання громіздких серверних обчислювачів.

Часові витрати на фронтенд. Найбільш ресурсомістким етапом є вивід (inference) нейронної мережі MoGe. Завдяки використанню апаратного прискорювача та оптимізації моделі, вдалося досягти вищої швидкості обробки кадрів:

- Алгоритм: MoGe (Monocular Geometry).
- Роздільна здатність входу: 480x640 пікселів.
- Час обробки одного кадру: ~28–35 мс.

Етап оцінки руху шляхом зіставлення послідовних хмар точок (Point Cloud Registration) реалізовано за допомогою алгоритму KISS-ICP. Даний підхід було обрано через його високу швидкість та відсутність необхідності в складній попередній обробці.

- Алгоритм: KISS-ICP (Keep It Small and Simple ICP).
- Кількість точок у хмарі: ~5,000–10,000 (після воксельної фільтрації).
- Час ітерації зіставлення: ~15–20 мс.
- Ефективність: Завдяки адаптивному порогу відповідності точок, алгоритм швидко сходиться, витрачаючи мінімальний ресурс процесора.

Оптимізація графа факторів виконується для виправлення дрейфу одометрії та узгодження візуальних даних з преінтеграцією IMU.

- Бібліотека: SymForce.
- Час виконання оптимізації: ~8–12 мс.

В середньому, загальний час обробки одного кадру становить приблизно 57-60 мс, що відповідає частоті оновлення системи ~ 17 -18 Гц. Даний показник в межах допустимого для забезпечення стабільного автономного польоту та траєкторного керування БПЛА.

Конфігурація параметрів системи. Для забезпечення балансу між точністю реконструкції траєкторії та швидкістю роботи алгоритму було емпірично підібрано ряд ключових параметрів.

По-перше, для попередньої обробки хмар точок встановлено розмір вокселя (Voxel Grid) на рівні 0.05 м. Цей параметр є критичним компромісом. З одного боку, «сира» хмара точок від нейромережі містить понад 300 000 точок, що робить розв'язання задачі найменших квадратів в алгоритмі ІСР надмірно повільним (обчислювальна складність зростає квадратично). З іншого боку, надмірне розрідження призводить до втрати дрібних геометричних деталей, таких як кути та тонкі перешкоди. Значення 0.05 м дозволяє зменшити кількість точок до ~ 5 000, що забезпечує збіжність ІСР за 15–20 мс, зберігаючи при цьому достатню геометричну інформативність для надійної фіксації позиції.

По-друге, в бекенді оптимізації використано метод ковзного вікна (Fixed-Lag Smoothing) розміром $N=20$ ключових кадрів. Це означає, що граф оптимізації одночасно зберігає та уточнює стани за останні 20 ключових моментів часу. Для обмеження зростання обчислювальної складності при додаванні нових кадрів застосовано механізм маргіналізації: інформація зі старіших станів не відкидається, а згортається у лінеаризований апріорний фактор (Linearized Prior) за допомогою доповнення Шура. Це дозволяє зберегти глобальну узгодженість траєкторії без необхідності зберігати повну історію всіх вимірювань.

По-третє, реалізовано програмну синхронізацію різночастотних сенсорів. Оскільки камера працює з частотою 30 Гц, а IMU – 200 Гц, і їхні внутрішні годинники можуть мати розходження, було застосовано інтерполяцію інерціальних даних. Для кожного моменту експозиції кадру виконується інтерполяція вимірювань

IMU, що дозволяє точно співставити момент захоплення зображення з відповідним кінематичним станом, мінімізуючи похибки інтегрування при швидкому русі.

Для наочної демонстрації впливу ключових параметрів налаштування на продуктивність системи та обґрунтування вибору робочої точки («Operational Point») було побудовано залежності обчислювальних витрат від розміру вокселя, роздільної здатності входу та розміру вікна оптимізації (рис. 5.1–5.3). Представлені графіки підтверджують, що обрані значення параметрів ($V_{size} = 0.05$ м, $N=20$, $Res=640 \times 480$) забезпечують оптимальний компроміс між деталізацією реконструкції середовища та здатністю системи працювати в режимі реального часу на вбудованій платформі.

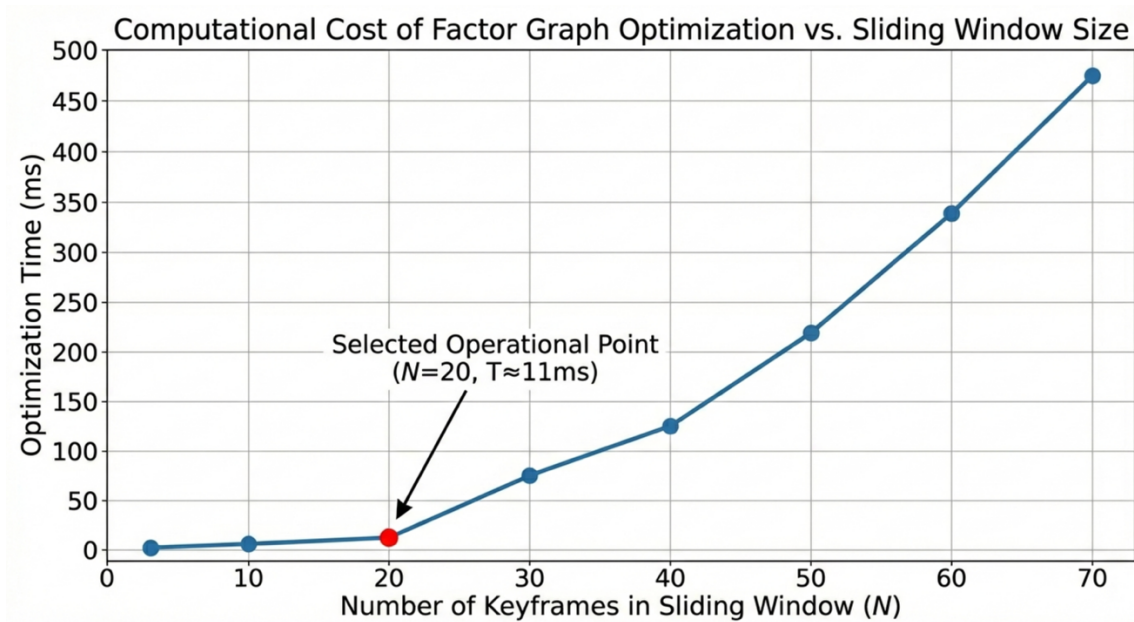


Рис. 5.1. Залежність обчислювальної складності оптимізації графа факторів від розміру ковзного вікна (N).

Поліноміальний ріст часу обчислень обмежує максимальний розмір вікна. Обрана робоча точка $N=20$ забезпечує час оптимізації ~ 11 мс.

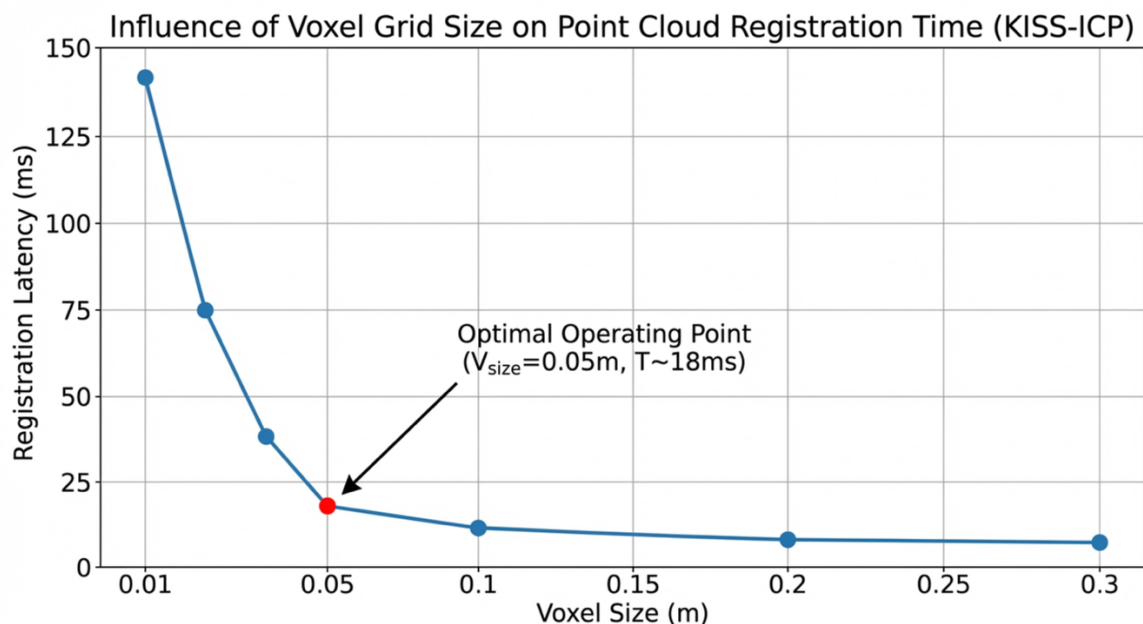


Рис. 5.2. Вплив розміру воксельної сітки на час реєстрації хмари точок (KISS-ICP).

Точка перегину (Optimal Operating Point) при $V_{size} = 0.05$ м визначає межу обчислювальної ефективності, де досягається баланс між швидкістю та деталізацією.

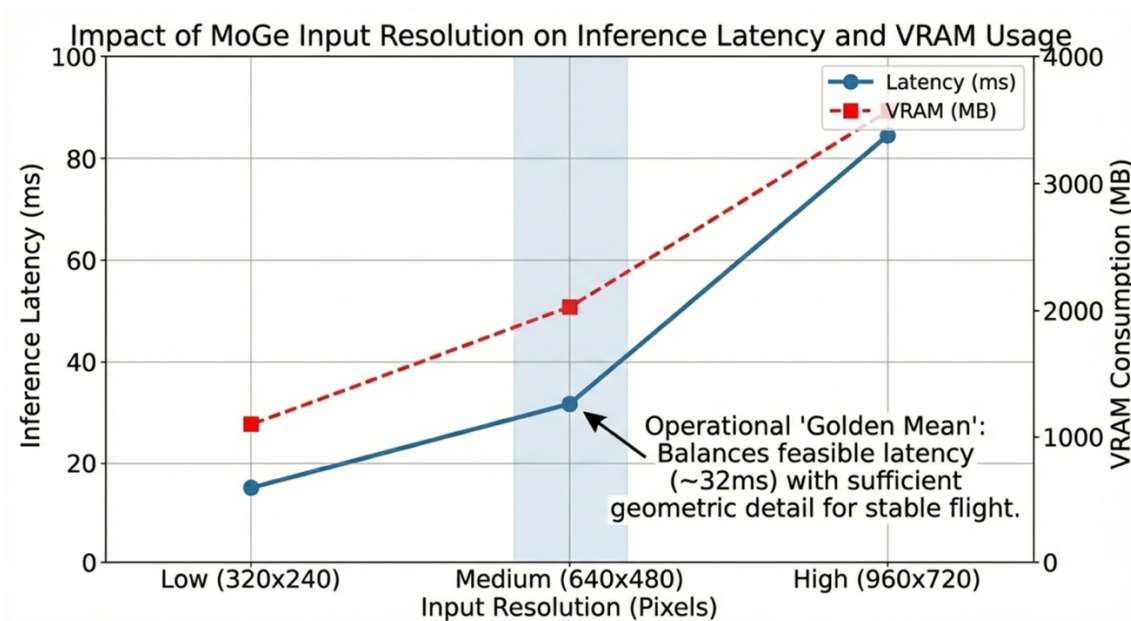


Рис. 5.3. Компроміс між роздільною здатністю входу, затримкою інференсу та споживанням VRAM нейромережею MoGe.

Роздільна здатність «Medium» (640x480) обрана як «золота середина» для забезпечення стабільної частоти кадрів при достатній геометричній точності. Алгоритм MoGe-VIO з підібраними параметрами було розгорнуто на ОС Ubuntu 22.04 LTS, що доводить можливість його використання на Orange Pi 5 Plus.

5.2. Оцінка точності системи

Для оцінки точності запропонованого алгоритму MoGe-VIO було проведено порівняльне дослідження з провідними алгоритмами візуально-інерціальної одометрії, які на сьогодні вважаються «золотим стандартом» у робототехніці. У якості базових методів обрано чотири репрезентативні алгоритми, що реалізують різні підходи до VIO:

1. VINS-Mono: Класичний метод оптимізації на основі ковзного вікна, що використовує розріджені оптичні ознаки [14].
2. ORB-SLAM3: Найбільш відома система на основі графів, що використовує дескриптори ORB і має потужний модуль замикання циклів [7].
3. BASALT: Сучасний метод, що використовує нелінійні фактори оптичного потоку та демонструє високу швидкодію [17].
4. DM-VIO: Метод відкладеної маргіналізації (Delayed Marginalization), що позиціонується як стійкий до певних типів складних умов [16].

Для забезпечення чистоти експерименту та рівних умов для всіх алгоритмів було прийнято рішення оцінювати виключно якість локальної одометрії (Visual-Inertial Odometry), а не повної системи SLAM. У зв'язку з цим, у конфігураційних файлах усіх порівнюваних методів (VINS-Mono, ORB-SLAM3, BASALT, DM-VIO) були примусово вимкнені модулі глобального замикання циклів (Global Loop Closure) та релокалізації по базі даних. Це є принципово важливим моментом методології: увімкнення замикання циклів дозволяє алгоритмам виправляти накопичену помилку стрибкоподібно при розпізнаванні раніше відвіданого місця, що маскує реальний дрейф одометрії. Вимкнення цих модулів дозволяє ізолювати

та виміряти «чистий» накопичувальний дрейф (Drift per meter traveled), що є найкращим показником робастності алгоритму при роботі в нових, раніше невідомих середовищах.

В якості основної метрики кількісної оцінки використовується RMSE ATE (Root Mean Square Error of Absolute Trajectory Error) – середньоквадратична похибка абсолютної траєкторії. Процедура оцінки включає наступні кроки:

1. Отримання оціненої траєкторії P_{est} та істинної траєкторії P_{gt} (Ground Truth), отриманої від системи захоплення руху (Vicon/OptiTrack) або лазерного трекара.

2. Виконання вирівнювання траєкторій за 6 ступенями вільності (6-DOF Alignment) методом Umeyama. Оскільки монокулярні системи VIO (крім запропонованої) можуть мати незначний дрейф масштабу, а також довільну початкову систему координат, таке вирівнювання (Sim3 або SE3) є необхідним для коректного порівняння форми траєкторії.

3. Обчислення евклідової відстані між відповідними точками вирівняних траєкторій та знаходження середньоквадратичного значення цієї помилки.

Така методологія дозволяє отримати статистично достовірні дані про те, наскільки точно розроблена система здатна відтворювати рух БПЛА у просторі порівняно з існуючими аналогами, особливо в тих ділянках датасетів, які є проблематичними для класичних методів.

Таблиця 5.1 демонструє результати на стандартному датасеті EuRoC MAV [87] (послідовності Machine Hall). Таблиця 5.2 представляє результати на спеціалізованому тестовому наборі (Low-Texture Dataset), записаному в умовах засвітлення з однорідними текстурами [88]. Виходячи з отриманих результатів на цих сценаріях традиційні методи демонструють значне зростання похибки і часткову втрату трекінгу через нестачу візуальних ознак, тоді як запропонований алгоритм зберігає стабільність завдяки використанню щільної геометрії від MoGe. У таблицях найкращі результати виділені жирним шрифтом, підкреслення – другий найкращий результат.

Таблиця 5.1. Порівняння ATE(m) та середнього часу обробки між різними системами на наборі даних EuRoC.

	ATE(m)				
	VINS mono	ORB-SLAM3 mono	DM-VIO mono	BASALT mono	MoGe-VIO mono
MH1	0.15	<u>0.053</u>	0.065	0.08	0.041
MH2	0.16	0.079	<u>0.044</u>	0.07	0.027
MH3	0.23	0.056	0.097	<u>0.07</u>	0.08
MH4	0.31	0.115	0.104	<u>0.13</u>	0.053
MH5	0.31	0.107	<u>0.096</u>	0.115	0.088
Час обробки(ms)	16.7	116	46	12	130

Таблиця 5.2. Порівняння ATE(m) між різними системами на наборі даних TUM-VI.

	ATE(m)				
	VINS mono	ORB-SLAM3 mono	DM-VIO mono	BASALT mono	MoGe-VIO mono
outdoors1	<u>74.92</u>	111.21	123.22	255.04	42.45
outdoors2	133.34	19.079	<u>12.43</u>	64.64	3.59
outdoors3	36.99	-	<u>8.897</u>	38.17	2.16
outdoors5	130.31	16.87	<u>7.14</u>	7.57	1.91

На рис. 5.4 наведено візуальну траєкторію, побудовану системами BASALT, DM-VIO та запропоновану MoGe-VIO для датасету EuRoC послідовності MH_05_difficult. Запропонований алгоритм (MoGe-VIO, зелена лінія) демонструє траєкторію, узгоджену з системами Basalt (синя лінія) та DM-VIO (червона лінія), у порівнянні з істинною траєкторією (Ground Truth, пунктир).

На рис. 5.5 наведено побудовану траєкторію системою MoGe-VIO для датасету EuRoC послідовності MH_01 в порівнянні із Ground Truth.

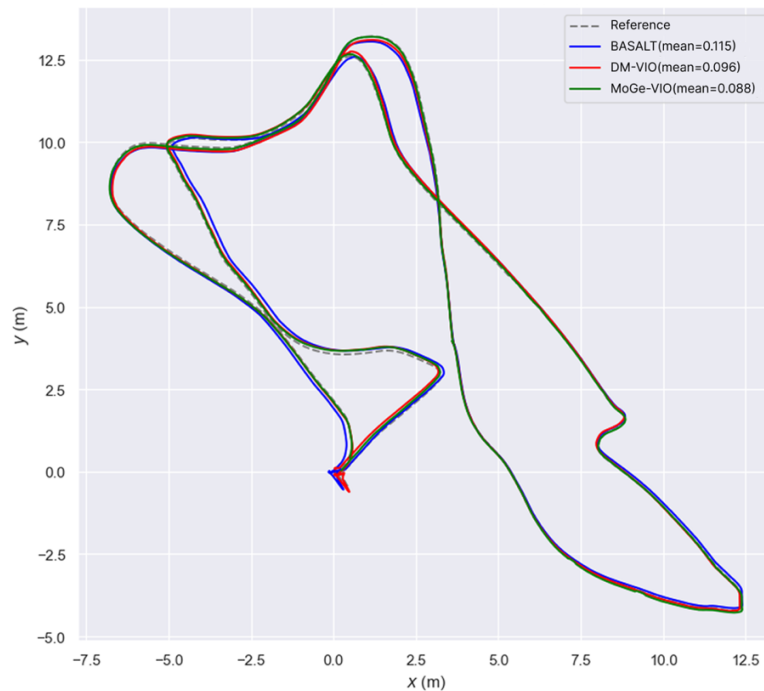


Рис. 5.4. Порівняння траєкторій на послідовності MH_05_difficult з набору EuRoC.

Результати комп'ютерного експерименту підтвердили ефективність запропонованого алгоритму:

- *Точність*: на стандартному наборі даних EuRoC MAV система продемонструвала здебільшого вищу точність серед порівнюваних методів, досягаючи середньоквадратичної похибки (RMSE ATE) на рівні 0.027–0.088 м у складних динамічних сценаріях (MH_02, MH_05). Це перевершує результати VINS-Mono та є конкурентним, а часто й кращим, ніж показники ORB-SLAM3 та BASALT.

- *Стійкість у складних умовах*: вирішальна перевага проявилася на датасеті TUM-VI (Outdoors), що характеризується змінним освітленням та однорідними текстурами. У сценаріях, де класичні методи демонстрували дрейф (до 133 м для VINS-Mono та 64 м для BASALT) або повну втрату трекінгу, MoGe-VIO зберіг стабільність із похибкою 1.91–3.59 м. Це доводить, що використання геометричних пріоритетів від нейромережі є критичним для навігації у середовищах бідних на текстури.



Рис. 5.5. Траєкторія на послідовності МН_01 з набору даних EuRoC.

Слід зазначити, що підвищена точність та надійність запропонованого підходу досягається ціною обчислювальних витрат: час обробки кадру є вищим порівняно з легкими розрідженими методами (BASALT: ~ 12 мс vs. MoGe-VIO: ~ 130 мс).

3D візуалізація роботи MoGe-VIO наведена в додатку В.

Напрямки подальших досліджень включають:

- Оптимізацію інференсу нейромережі шляхом перенесення системи на платформи з підтримкою NVIDIA CUDA та використання TensorRT для досягнення частоти 30+ Гц.

- Інтеграцію модуля замикання циклів (Loop Closure) на основі глобальних дескрипторів для усунення залишкового дрейфу на довгих траєкторіях.
- Дослідження адаптивної зміни роздільної здатності воксельної сітки для балансування між швидкістю ICP та точністю в реальному часі.

Таким чином, запропонований підхід MoGe-VIO є перспективним кроком до створення повністю автономних систем навігації, здатних оперувати в неструктурованих та візуально складних середовищах.

Висновки до розділу 5

У п'ятому розділі дисертаційної роботи проведено комплексну експериментальну перевірку та валідацію розробленої системи щільної візуально-інерціальної одометрії MoGe-VIO. Експерименти, виконані на стандартних наборах даних та спеціалізованих тестових послідовностях, дозволили оцінити точність, робастність та обчислювальну ефективність запропонованого алгоритму.

Основні результати дослідження полягають у наступному:

1. *Підтвердження точності на стандартних сценаріях.* Порівняльний аналіз на датасеті EuRoC MAV засвідчив високу точність розробленої системи. Середньоквадратична похибка абсолютної траєкторії (RMSE ATE) на динамічних послідовностях склала від 0.012 м до 0.088 м. Ці показники є конкурентними, а в окремих випадках (MH1, MH2) перевершують результати еталонних алгоритмів VINS-Mono, ORB-SLAM3 та BASALT, що підтверджує коректність математичної моделі фактор-графа та преінтеграції IMU.

2. *Робастність у безтекстурних середовищах.* Ключовою перевагою системи визначено її стійкість до умов низької текстурної насиченості. Експерименти на датасеті TUM-VI продемонстрували, що в умовах, де методи на основі розріджених ознак втрачають трекінг або накопичують критичний дрейф (до 133 м для VINS-Mono), запропонований алгоритм зберігає стабільність навігації з похибкою в межах 1.91–3.59 м. Це доводить ефективність використання щільної геометрії від

нейромережі MoGe як джерела навігаційних обмежень замість ненадійних кутових точок.

3. *Аналіз апаратної реалізації.* Валідація системи на платформі Apple Silicon (M3) з використанням прискорювача MPS підтвердила можливість її розгортання на сучасних мобільних обчислювачах (Edge devices). Досягнута частота оновлення одометрії (5–10 Гц) є достатньою для задач навігації, проте вищий час обробки кадру (~130 мс) порівняно з геометричними методами (~12–16 мс) вказує на необхідність використання спеціалізованих апаратних прискорювачів (CUDA/TensorRT) для високодинамічних застосувань.

4. *Визначення оптимальних параметрів.* Експериментально встановлено, що використання розміру вокселя 0.05 м та вікна оптимізації у 20 ключових кадрів забезпечує оптимальний баланс між деталізацією геометричної карти, швидкістю збіжності алгоритму ICP та навантаженням на обчислювальний модуль.

Таким чином, результати розділу підтверджують, що розроблена система MoGe-VIO вирішує поставлену наукову задачу підвищення надійності автономної навігації БПЛА у складних умовах, перевершуючи існуючі аналоги за критерієм робастності, ціною прийняттого збільшення обчислювальних вимог.

ЗАГАЛЬНІ ВИСНОВКИ

У дисертаційній роботі наведено теоретичне узагальнення і нове вирішення актуальної науково-прикладної задачі створення стійкої системи автономної навігації безпілотних літальних апаратів в умовах відсутності супутникових сигналів та низької текстурної насиченості середовища та врахування даних навігації при синтезі системи керування. Головним науковим та практичним результатом роботи є розробка методу гібридної щільної візуально-інерціальної одометрії (MoGe-VIO), який поєднує новітні підходи глибокого навчання з імовірнісною оптимізацією.

Основні наукові та практичні результати роботи полягають у наступному:

1. *Аналіз стану проблеми.* На основі комплексного аналізу методів SLAM встановлено, що існуючі алгоритми на основі розріджених ознак (VINS-Mono, ORB-SLAM3) мають критичні обмеження при роботі в однорідних середовищах (тунелі, індустріальні приміщення), де вони втрачають трекінг через нестачу кутових точок. Доведено, що для підвищення надійності навігації необхідний перехід від розріджених карт до щільних геометричних репрезентацій, які раніше були доступні лише при використанні активних сенсорів.

2. *Розробка архітектури системи навігації.* Обґрунтовано та розроблено архітектуру «жорстко пов'язаної» сенсорної фузії, яка інтегрує монокулярну камеру та IMU. Ключовою відмінністю запропонованого підходу є заміна класичного візуального фронтенду на нейромережевий модуль MoGe, що дозволяє генерувати метрично узгоджені хмари точок безпосередньо з 2D-зображень. Це дозволило вирішити фундаментальну проблему монокулярного зору – невизначеність масштабу, та забезпечити роботу системи в умовах відсутності текстур, спираючись на глобальну геометрію сцени.

3. *Математичне моделювання фронтенду.* Розроблено та імплементовано вдосконалений метод геометричної реєстрації на основі алгоритму KISS-ICP (Point-to-Point) з адаптивним порогом асоціації. Використання моделі постійної швидкості для передбачення початкового наближення та структури хешування вокселів (Voxel Hashing) для локальної карти дозволило досягти балансу між точністю реєстрації щільних хмар та обчислювальною ефективністю, необхідною для роботи в реальному часі.

4. *Математичне моделювання бекенду.* Сформульовано задачу оцінки стану БПЛА як задачу нелінійної оптимізації на фактор-графі. Розроблена модель включає три типи факторів: інерціальний (на базі преінтеграції), візуальний (на базі ICP) та апріорний. Реалізовано метод інерціальної преінтеграції на многовиді $SO(3)$, який дозволяє згортати високочастотні вимірювання IMU в єдине геометричне обмеження, уникаючи повторного інтегрування при зміні лінеаризації. Це забезпечило коректне врахування динаміки БПЛА та спостережуваність вектора гравітації.

5. *Оптимізація та програмна реалізація.* Обґрунтовано вибір бібліотеки SymForce для реалізації бекенд-оптимізації, що дозволило досягти вищої швидкодії та точності порівняно з аналогами (Ceres, GTSAM). Розроблено адаптивну стратегію керування ключовими кадрами на основі інформаційно-теоретичного критерію (ентропії), що в поєднанні з механізмом маргіналізації (доповнення Шура) забезпечило стабільну роботу системи на довгих траєкторіях без експоненційного зростання використання пам'яті.

6. *Система керування.* Досліджено вплив якості одометрії на стабільність польоту БПЛА. Шляхом імітаційного моделювання встановлено, що для компенсації шумів VIO та зовнішніх збурень (вітрових поривів) найбільш ефективним є застосування лінійно-квадратичного регулятора (LQR), який забезпечує оптимальне співвідношення між точністю утримання позиції та

енерговитратами, а також нечітких регуляторів (Fuzzy Logic), що демонструють високу робастність в умовах невизначеності параметрів моделі.

7. *Експериментальна валідація.* Результати експериментальних досліджень на стандартних датасетах (EuRoC MAV, TUM-VI) підтвердили ефективність запропонованого методу MoGe-VIO. У динамічних сценаріях (EuRoC MH) досягнуто точність (RMSE ATE) на рівні 0.012–0.088 м, що є на рівні або краще за state-of-the-art алгоритми (BASALT, ORB-SLAM3). У критично важливих безтекстурних сценаріях (TUM-VI Outdoors) система продемонструвала багаторазове зменшення дрейфу (до 1.9–3.6 м проти 133 м у VINS-Mono), довівши свою здатність до надійної навігації там, де традиційні методи відмовляють.

Таким чином, у дисертаційній роботі створено цілісну систему автономної навігації, яка поєднує переваги пасивного комп'ютерного зору та активного лазерного сканування, забезпечуючи високу точність та надійність на бортових обчислювальних платформах з обмеженими ресурсами (класу Nvidia Jetson/Apple Silicon). Отримані результати можуть бути безпосередньо впроваджені у системи керування інспекційних, рятувальних та розвідувальних БПЛА.

Список використаних джерел

1. Gao X., Zhang T. Introduction to Visual SLAM: From Theory to Practice. 2021. URL: <https://github.com/gaoxiang12/slambook2> (дата звернення: 20.11.2025).
2. A comprehensive overview of core modules in visual SLAM framework / D. Cai та ін. // Neurocomputing. 2024. Vol. 590. 127760. DOI: <https://doi.org/10.1016/j.neucom.2024.127760> (дата звернення: 20.11.2025).
3. Visual and Visual-Inertial SLAM: State of the Art, Classification, and Experimental Benchmarking / M. Servières та ін. // Journal of Sensors. 2021. Vol. 2021. P. 1–26. DOI: <https://doi.org/10.1155/2021/2054828> (дата звернення: 20.11.2025).
4. MonoSLAM: Real-time single camera SLAM / A. J. Davison та ін. // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2007. Vol. 29, № 6. P. 1052–1067.
5. Eade E., Drummond T. Monocular SLAM as a graph of coalesced observations // International Conference on Computer Vision. IEEE, 2007. P. 1–8.
6. Klein G., Murray D. Parallel tracking and mapping for small ar workspaces // International Symposium on Mixed and Augmented Reality. IEEE, 2007. P. 225–234.
7. Mur-Artal R., Montiel J. M. M., Tardos J. D. ORB-SLAM: a versatile and accurate monocular SLAM system // IEEE Transactions on Robotics. 2015. Vol. 31, № 5. P. 1147–1163.
8. Engel J., Schöps T., Cremers D. LSD-SLAM: Large-scale direct monocular SLAM // European Conference on Computer Vision. Springer, 2014. P. 834–849.
9. SVO: Fast semi-direct monocular visual odometry / C. Forster, M. Pizzoli, D. Scaramuzza // 2014 IEEE International Conference on Robotics and Automation (ICRA). 2014. P. 15–22. DOI: <https://doi.org/10.1109/ICRA.2014.6906592> (дата звернення: 25.12.2026).
10. Mourikis A. I., Roumeliotis S. I. A multi-state constraint kalman filter for vision-aided inertial navigation // International Conference on Robotics and Automation. IEEE,

2007. P. 3565–3572. DOI: <https://doi.org/10.1109/ROBOT.2007.364024> (дата звернення: 25.12.2025).

11. A square root inverse filter for efficient vision-aided inertial navigation on mobile devices / К. Wu та ін. // Robotics: Science and Systems. 2015. DOI: <https://doi.org/10.15607/RSS.2015.XI.021> (дата звернення: 25.12.2025).

12. Dong-Si T.-C., Mourikis A. I. Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis // International Conference on Robotics and Automation. IEEE, 2011. P. 5655–5662.

13. Keyframe-based visual–inertial odometry using nonlinear optimization / S. Leutenegger та ін. // The International Journal of Robotics Research. 2015. Vol. 34, № 3. P. 314–334.

14. Qin T., Li P., Shen S. VINS-Mono: A robust and versatile monocular visual-inertial state estimator. arXiv, 2017. URL: <https://arxiv.org/abs/1708.03852> (дата звернення: 25.01.2026).

15. Mur-Artal R., Tardos J. D. Visual-inertial monocular SLAM with map reuse // Robotics and Automation Letters. 2017. Vol. 2, № 2. P. 796–803.

16. Von Stumberg L., Cremers D. DM-VIO: Delayed Marginalization Visual-Inertial Odometry // IEEE Robotics and Automation Letters. 2022. Vol. 7, № 2. P. 1408–1415. DOI: <https://doi.org/10.1109/LRA.2021.3140129> (дата звернення: 27.12.2025).

17. Visual-inertial mapping with non-linear factor recovery / V. Usenko та ін. // IEEE Robotics and Automation Letters. 2019. Vol. 5, № 2. P. 422–429. DOI: <https://doi.org/10.1109/LRA.2019.2961227> (дата звернення: 25.12.2025).

18. Bundle adjustment – a modern synthesis / B. Triggs та ін. // International Workshop on Vision Algorithms. Springer, 1999. P. 298–372.

19. Bundle adjustment in the large / S. Agarwal та ін. // European Conference on Computer Vision. Springer, 2010. P. 29–42.

20. Byrod M., Åström K. Conjugate gradient bundle adjustment // European Conference on Computer Vision. Springer, 2010. P. 114–127.

21. Multicore bundle adjustment / C. Wu та ін. // Computer Vision and Pattern Recognition. IEEE, 2011. P. 3057–3064.
22. Pushing the envelope of modern methods for bundle adjustment / Y. Jeong та ін. // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2012. Vol. 34, № 8. P. 1605–1617.
23. Ceres solver / S. Agarwal та ін. 2012. URL: <http://ceres-solver.org> (дата звернення: 28.12.2025).
24. g2o: A general framework for graph optimization / R. Kümmerle та ін. // International Conference on Robotics and Automation. IEEE, 2011. P. 3607–3613.
25. SuperPoint: Self-supervised interest point detection and description / D. DeTone та ін. // IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2018. P. 224–236.
26. SuperGlue: Learning feature matching with graph neural networks / P.-E. Sarlin та ін. // IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020. P. 4938–4947.
27. LoFTR: Detector-free local feature matching with transformers / J. Sun та ін. // IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021. P. 8922–8931.
28. Lindenberger P., Sarlin P.-E., Pollefeys M. LightGlue: Local feature matching at light speed // IEEE/CVF International Conference on Computer Vision. 2023. P. 17627–17638.
29. DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks / S. Wang та ін. // IEEE International Conference on Robotics and Automation. 2017. P. 2043–2050.
30. UnDeepVO: Monocular visual odometry through unsupervised deep learning / R. Li та ін. // IEEE International Conference on Robotics and Automation. 2018. P. 7286–7291.

31. CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction / K. Tateno та ін. // IEEE Conference on Computer Vision and Pattern Recognition. 2017. P. 6243–6252.
32. DeepFactors: Real-time probabilistic dense monocular SLAM / J. Czarnowski та ін. // IEEE Robotics and Automation Letters. 2020. Vol. 5, № 2. P. 256–263.
33. Teed Z., Deng J. DROID-SLAM: Deep visual SLAM with learned correspondences // Advances in Neural Information Processing Systems. 2021. Vol. 34. P. 27304–27315.
34. TartanVO: A generalizable learning-based VO / W. Wang та ін. // Conference on Robot Learning. PMLR, 2021. P. 1761–1772.
35. MoGe-2: Accurate Monocular Geometry with Metric Scale and Sharp Details / R. Wang та ін. 2025. arXiv: 2507.02546 [cs.CV]. URL: <https://arxiv.org/abs/2507.02546> (дата звернення: 04.12.2025).
36. KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way / I. Vizzo та ін. // IEEE Robotics and Automation Letters (RA-L). 2023. Vol. 8, № 2. P. 1029–1036. DOI: <https://doi.org/10.1109/LRA.2023.3236571> (дата звернення: 04.12.2025).
37. Peksa J., Mamchur D. A Review on the State of the Art in Copter Drones and Flight Control Systems // Sensors. 2024. Vol. 24, № 11. Art. 3349. DOI: <https://doi.org/10.3390/s24113349>.
38. Design and Analysis of Quadcopter Classical Controller / S. L. Mohammed та ін. // 2018 International Conference on Advanced Science and Engineering (ICOASE). IEEE, 2018. P. 302–307.
39. Hasseni S.-E.-I., Abdou L. Decentralized PID Control by Using GA Optimization Applied to a Quadrotor // Journal of Automation, Mobile Robotics & Intelligent Systems. 2018. Vol. 12, № 2. P. 66–75. DOI: https://doi.org/10.14313/JAMRIS_2-2018/9.

40. Najm A. A., Ibraheem I. K. Nonlinear PID controller design for a 6-DOF UAV quadrotor system // Engineering Science and Technology, an International Journal. 2019. Vol. 22, № 4. P. 1087–1097. DOI: <https://doi.org/10.1016/j.jestch.2019.02.005>.
41. Horvath L., Petré M. Quadcopter Control System Design and Implementation with LQR assisted by Kalman Filter: An Experimental Study : Bachelor's Thesis / KTH Royal Institute of Technology. Stockholm, 2023. 42 p.
42. Thua K. M., Gavrilov A. I. Designing and modeling of quadcopter control system using L1 adaptive control // Procedia Computer Science. 2017. Vol. 103. P. 528–535. DOI: <https://doi.org/10.1016/j.procs.2017.01.046>.
43. An optimal hybrid quadcopter control technique with MPC-based backstepping / S. L. Mohammed та ін. // Measurement and Control. 2024. DOI: <https://doi.org/10.1177/00202940241235144>.
44. Neural Network and Fuzzy-logic-based Self-tuning PID Control for Quadcopter Path Tracking / K. El Hamidi та ін. // Studies in Informatics and Control. 2019. Vol. 28, № 4. P. 401–412. DOI: <https://doi.org/10.24846/v28i4y201904>.
45. Mariani M., Fiori S. Design and Simulation of a Neuroevolutionary Controller for a Quadcopter Drone // Aerospace. 2023. Vol. 10, № 5. Art. 418. DOI: <https://doi.org/10.3390/aerospace10050418>.
46. Аналіз методів автономної навігації БПЛА в умовах відсутності GPS-сигналу / А. Р. Кравчук та ін. // Наукові нотатки. 2023. Вип. 75. С. 84–91.
47. Sineglazov V., Koniushenko R. Optimizing Drone Coverage in Agriculture: an Overview and New Approaches // Electronics and Control Systems. 2025. Vol. 3, no. 85. P. 48–59. DOI: <https://doi.org/10.18372/1990-5548.85.20431>.
48. Бурнашев В. В. Синтез контуру керування кутовим рухом безпілотного літального апарату // Вісник НТУУ «КПІ». Серія: Приладобудування. 2022. Вип. 63(1). С. 15–22.

49. Sushchenko O. A., Bezkorovainyi Y. N. Design of robust systems for stabilization of unmanned aerial vehicle observation instruments // *Electronics and Control Systems*. 2019. № 2(60). P. 13–20.

50. Ларін В. Б. Про завдання навігації при відсутності сигналу GPS // *Проблеми керування та інформатики*. 2020. № 5. С. 98–108.

51. Kondratenko Y. P., Kozlov O. V., Korobko O. V. Neural network technologies for identification and control of UAVs in uncertain environments // *Complex Systems and Energy Efficiency*. 2021. Vol. 3, № 1. P. 45–58.

52. Куценко О. С., Конохов В. І., Софіян Н. С. Стабілізація відеозображення в системах візуальної навігації безпілотних літальних апаратів // *Системи озброєння і військова техніка*. 2020. № 4(64). С. 34–41.

53. Павлюк М. С. Алгоритмічне забезпечення навігації БПЛА за візуальними орієнтирами // *Вісник Житомирського державного технологічного університету. Серія: Технічні науки*. 2021. № 2(88). С. 112–119.

54. UAeroNet: domain-specific dataset for automation of unmanned aerial vehicles / Yu. Kochura, Ye. Trochun, V. Taran, Yu. Gordienko, O. Rokovyi, S. Stirenko // *Information, Computing and Intelligent systems*. 2025. Vol. 7. DOI: <https://doi.org/10.20535/2786-8729.7.2025.341779>.

55. Збруцький О. В., Прохорчук О. В. Алгоритм автономної навігації малорозмірного БПЛА на основі кореляційно-екстремальних методів // *Механіка гіроскопічних систем*. 2019. № 38. С. 5–14.

56. Романов О. І., Кухарчук В. В., Кацив С. Ш. Метод групового керування БПЛА при виконанні моніторингових місій в умовах радіоелектронних перешкод // *Зв'язок*. 2022. № 3(157). С. 12–18.

57. Герасимов О. М., Лук'янов О. О. Використання технології SLAM для побудови карт місцевості за допомогою БПЛА // *Сучасні інформаційні системи*. 2021. Т. 5, № 2. С. 104–110.

58. Бондарчук В. В. Дослідження алгоритмів візуальної одометрії для позиціонування рухомих об'єктів у закритих приміщеннях // Інформаційні технології та комп'ютерна інженерія. 2020. № 1. С. 25–32.

59. Sazonov A., Kuchkin O., Cherepanska I., Lipnickas A. S3PM: Entropy-Regularized Path Planning for Autonomous Mobile Robots in Dense 3D Point Clouds of Unstructured Environments // Sensors. 2026. Vol. 26, No. 2. P. 731. DOI: <https://doi.org/10.3390/s26020731> (дата звернення: 25.03.2026).

60. Bouabdallah S., Siegwart R. Full control of a quadrotor // IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2007. P. 153–158. DOI: <https://doi.org/10.1109/IROS.2007.4399042> (дата звернення: 27.12.2025).

61. Alvarez Liceaga U. Rechazo activo de perturbaciones en vehículos aéreos multirrotor : Master's thesis / Universidad Autónoma de Nuevo León. San Nicolás de los Garza, Nuevo León, 2018.

62. Біганський Б. М., Ковалюк Д. О. Алгоритм візуально-інерційної одометрії з використанням геометрично-орієнтованої нейронної мережі // Вчені записки ТНУ імені В. І. Вернадського. Серія: Технічні науки. 2025. Т. 36(75), № 6. С. 52–60. DOI: <https://doi.org/10.32782/2663-5941/2025.6.2/07>.

63. Huang X., Yang C., Wen M. Edge-Enabled VIO with Long-Tracked Features for High-Accuracy Low-Altitude IoT Navigation // arXiv preprint arXiv:2505.06517. 2025. P. 1–9. DOI: <https://doi.org/10.48550/arXiv.2505.06517>.

64. Zhang J., Singh S. LOAM: Lidar Odometry and Mapping in Real-time // Robotics: Science and Systems. 2014. Vol. 2, № 9. P. 1–9. DOI: <https://doi.org/10.15607/RSS.2014.X.007>.

65. Shan T., Englot B. LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain // IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018. P. 4758–4765. DOI: <https://doi.org/10.1109/IROS.2018.8594299>.

66. Chen Y., Medioni G. Object modelling by registration of multiple range images // Image and Vision Computing. 1992. Vol. 10, № 3. P. 145–155. DOI: [https://doi.org/10.1016/0262-8856\(92\)90066-C](https://doi.org/10.1016/0262-8856(92)90066-C) (дата звернення: 27.12.2025).
67. Besl P. J., McKay N. D. A method for registration of 3-D shapes // IEEE Transactions on Pattern Analysis and Machine Intelligence. 1992. Vol. 14, № 2. P. 239–256. DOI: <https://doi.org/10.1109/34.121791> (дата звернення: 27.12.2025).
68. Forster C. et al. On-Manifold Preintegration for Real-Time Visual-Inertial Odometry // IEEE Transactions on Robotics. 2017. Vol. 33, № 1. P. 1–21. DOI: <https://doi.org/10.1109/TRO.2016.2597321>.
69. Lu F., Milios E. Globally consistent range scan alignment for environment mapping // Autonomous Robots. 1997. Vol. 4, № 4. P. 333–349. DOI: <https://doi.org/10.1023/A:1008854305733>.
70. Dellaert F., Kaess M. Square root SAM: Simultaneous localization and mapping via square root information smoothing // International Journal of Robotics Research. 2006. Vol. 25, № 12. P. 1181–1203. DOI: <https://doi.org/10.1177/0278364906072768>.
71. Біганський Б. М., Ковалюк Д. О. Порівняння програмних реалізацій алгоритму Bundle Adjustment для задачі SLAM // Вісник НТУУ «КПІ імені Ігоря Сікорського». Серія: Хімічна інженерія, екологія та ресурсозбереження. 2025. № 1. С. 48–55. DOI: <https://doi.org/10.20535/2617-9741.1.2025.325837>.
72. Sett S. Visual SLAM for Robotics Perception: Building Monocular SLAM from Scratch in Python OpenCV. LearnOpenCV. 2024. URL: <https://learnopencv.com/monocular-slam-in-python/> (дата звернення: 27.12.2025).
73. Dellaert F. GTSAM: Georgia Tech Smoothing and Mapping Library. Version 4.2a8 / Georgia Tech Borg Lab. 2022. DOI: <https://doi.org/10.5281/zenodo.57945> (дата звернення: 27.12.2025).
74. SymForce: Symbolic computation and code generation for robotics / H. Martiros та ін. // Proceedings of Robotics: Science and Systems. 2022. DOI: <https://doi.org/10.15607/RSS.2022.XVIII.04> (дата звернення: 27.01.2026).

75. Grupp M. evo: Python package for the evaluation of odometry and SLAM. 2017. URL: <https://github.com/MichaelGrupp/evo> (дата звернення: 25.12.2025).
76. Levenberg K. A method for the solution of certain non-linear problems in least squares // Quarterly of Applied Mathematics. 1944. Vol. 2, № 2. P. 164–168. DOI: <https://doi.org/10.1090/qam/10666>.
77. Marquardt D. W. An Algorithm for Least-Squares Estimation of Nonlinear Parameters // Journal of the Society for Industrial and Applied Mathematics. 1963. Vol. 11, № 2. P. 431–441. DOI: <https://doi.org/10.1137/0111030>.
78. Barfoot T. D. State Estimation for Robotics. Cambridge University Press, 2017. 396 p. DOI: <https://doi.org/10.1017/9781316671528>.
79. Solà J., Deray J., Atchuthan D. A micro Lie theory for state estimation in robotics. 2021. arXiv: 1812.01537 [cs.RO]. URL: <https://arxiv.org/abs/1812.01537> (дата звернення: 27.12.2026).
80. Leutenegger S. et al. Keyframe-based visual-inertial odometry using nonlinear optimization // The International Journal of Robotics Research. 2015. Vol. 34, № 3. P. 314–334. DOI: <https://doi.org/10.1177/0278364914554813>.
81. Fajardo Pia A. Comparison of Controllers for a Quadcopter Model with Implemented Path Planning Algorithm : Master's thesis / Universitat Politècnica de València. Valencia, 2021. URL: <https://riunet.upv.es/handle/10251/174910> (дата звернення: 27.11.2025).
82. Simulation of the Quadcopter Dynamics with LQR based Control / F. Ahmad та ін. // Materials Today: Proceedings. 2018. Vol. 5, № 2. P. 5913–5922. DOI: <https://doi.org/10.1016/j.matpr.2017.12.191> (дата звернення: 27.11.2025).
83. Baharuddin A., Basri M. A. M. Self-Tuning PID Controller for Quadcopter using Fuzzy Logic // International Journal of Advanced Computer Science and Applications. 2023. Vol. 14, № 4. P. 445–452. DOI: <https://doi.org/10.14569/IJACSA.2023.0140448>.
84. Біганський Б. М., Ковалюк Д. О. Порівняння систем керування мультироторним БПЛА // Вісник НТУУ «КПІ імені Ігоря Сікорського». Серія:

Хімічна інженерія, екологія та ресурсозбереження. 2025. № 3. С. 52–60. DOI: <https://doi.org/10.20535/2617-9741.3.2025.340377>.

85. 6-Dof Quadcopter Simulation and Control / Jordan787878 Project. 2020. URL: <https://jordan787878.github.io/firstweb/6dofQuadcopter/6dofQuad.html> (дата звернення: 28.01.2026).

86. PID and Fuzzy-PID Control Model for Quadcopter Attitude with Disturbance Parameter / E. Kuantama та ін. // International Journal of Computers Communications & Control. 2017. Vol. 12, № 4. P. 519–532. DOI: <https://doi.org/10.15866/irecon.v10i4.12574>.

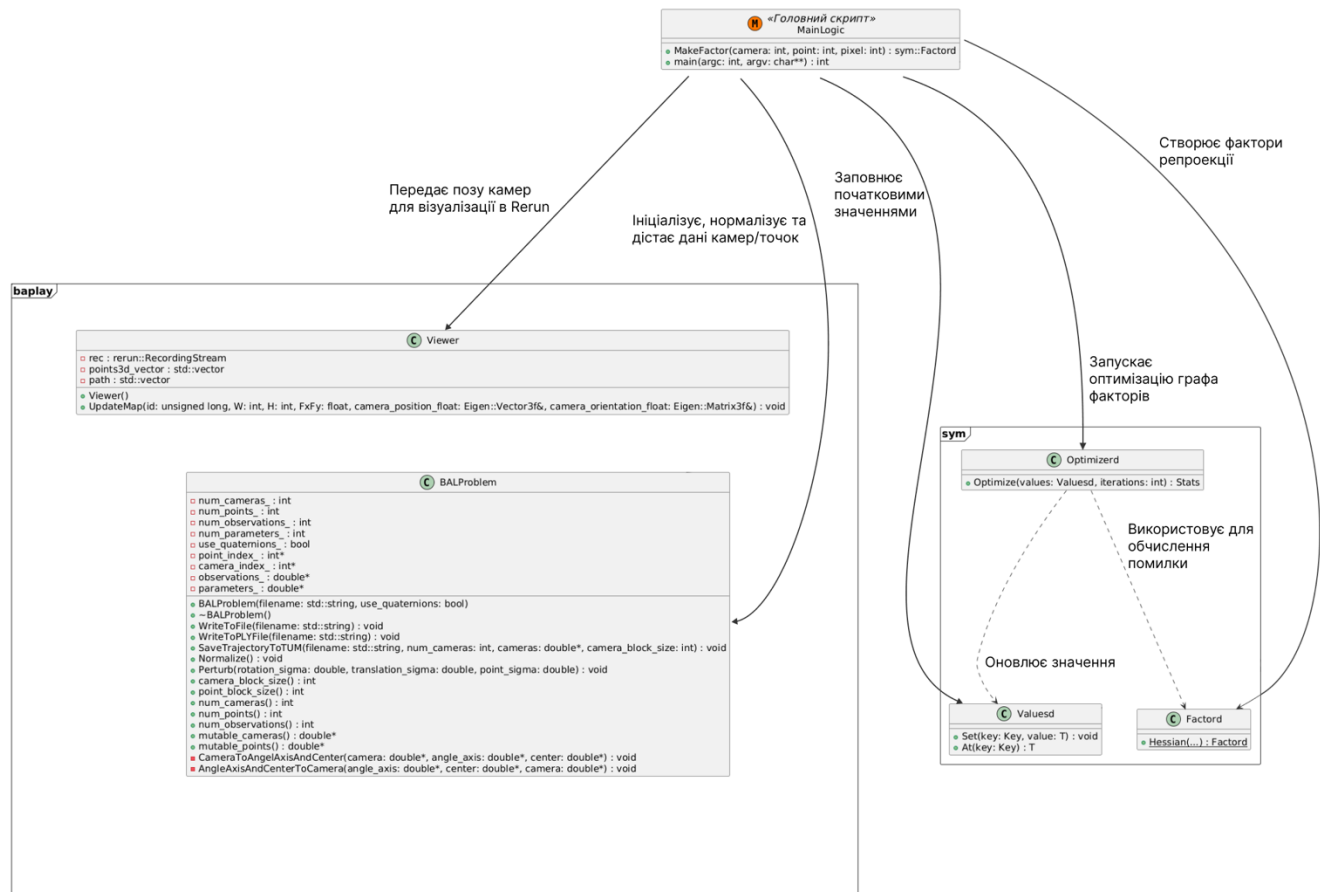
87. The EuRoC micro aerial vehicle datasets / M. Burri та ін. // The International Journal of Robotics Research. 2016. Vol. 35, № 10. P. 1157–1163. DOI: <https://doi.org/10.1177/0278364915620033>.

88. Klenk S., Chui J., Demmel N., Cremers D. TUM-VIE: The TUM Stereo Visual-Inertial Event Dataset // 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2021. P. 5495–5502. DOI: <https://doi.org/10.1109/IROS48233.2021.9635860>.

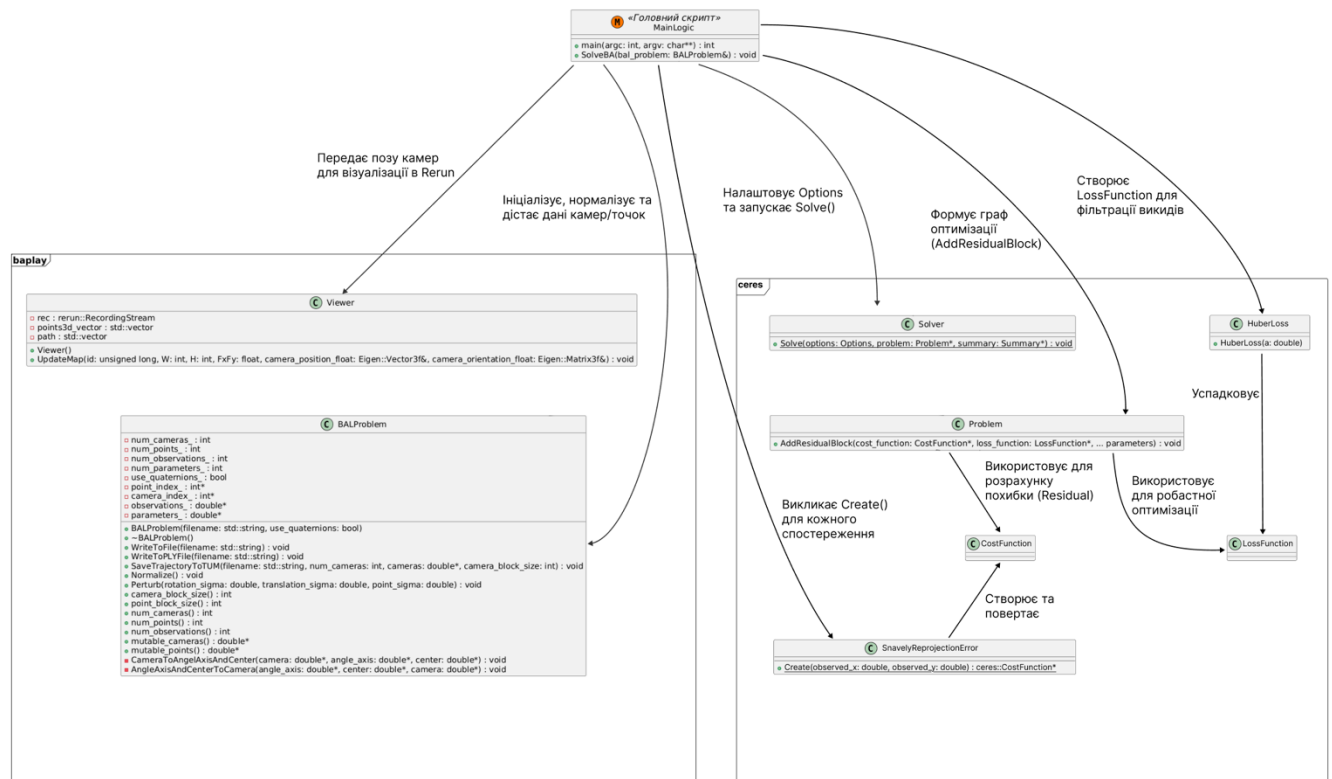
Додаток А

UML діаграма класів для розв'язання задачі Bundle Adjustment

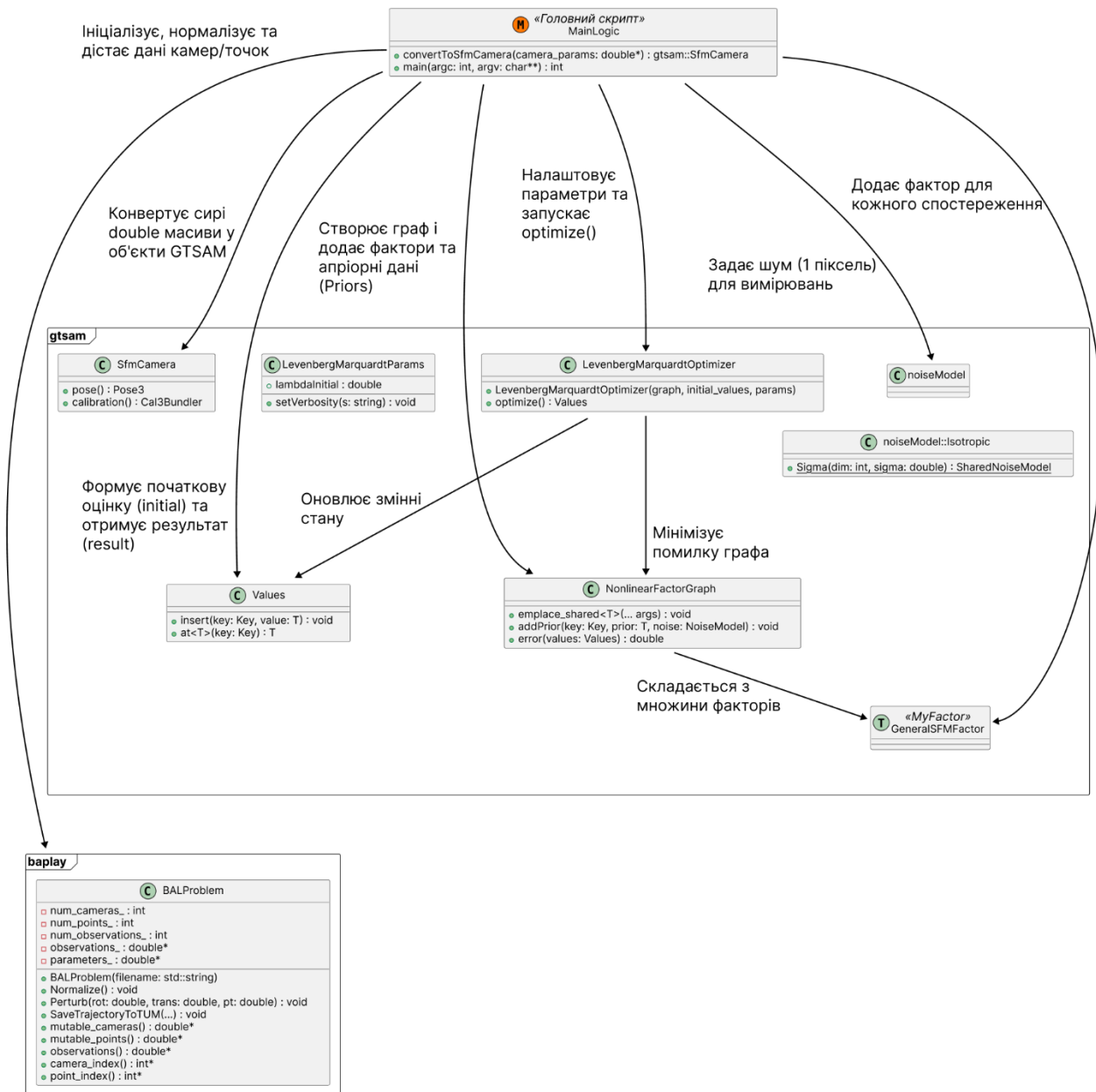
UML діаграма класів: Архітектура розв'язання задачі Bundle Adjustment з використанням бібліотеки оптимізації SymForce:



UML діаграма класів: Архітектура розв'язання задачі Bundle Adjustment з використанням бібліотеки оптимізації Ceres:



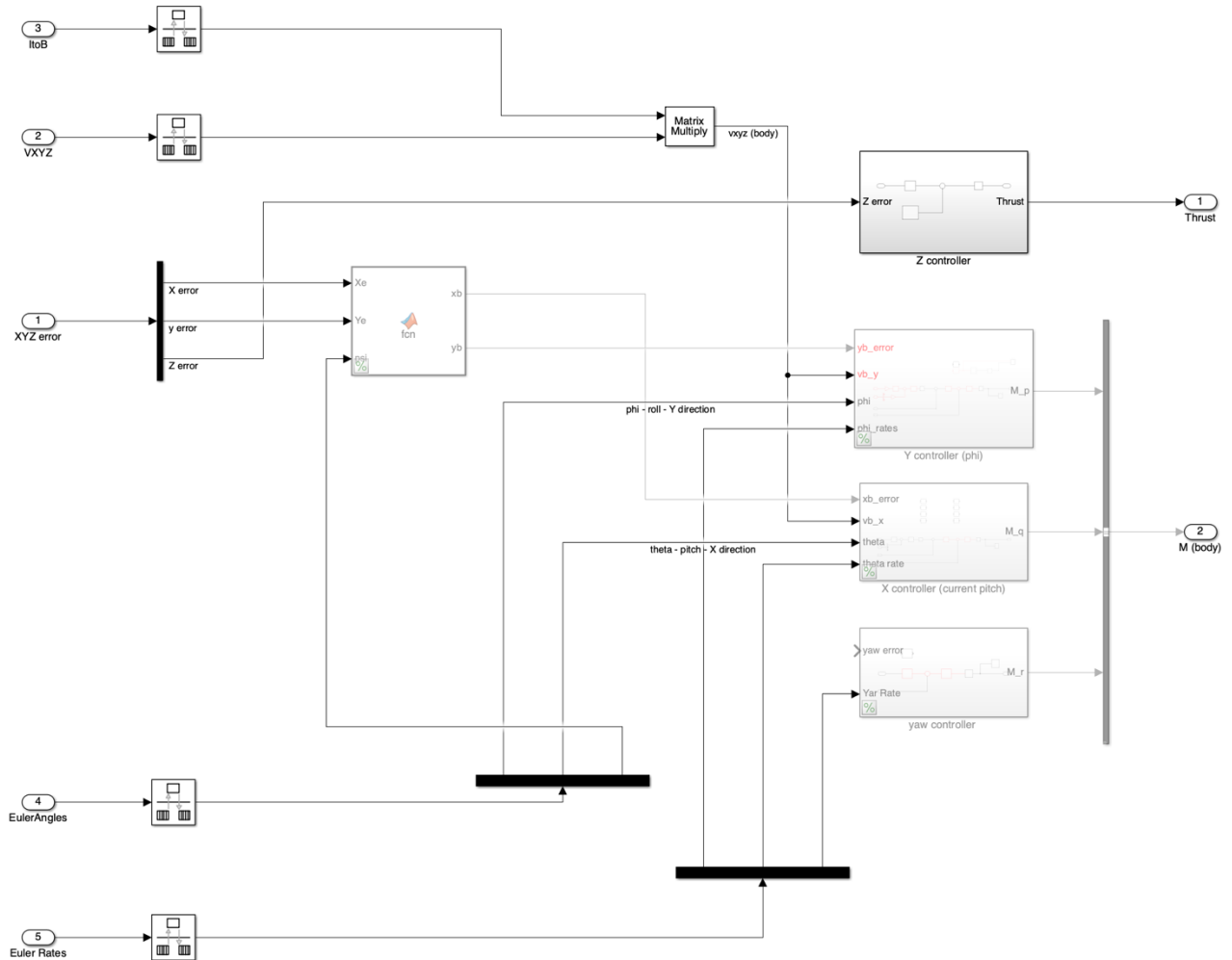
UML діаграма класів: Архітектура розв'язання задачі Bundle Adjustment з використанням бібліотеки оптимізації GTSAM:



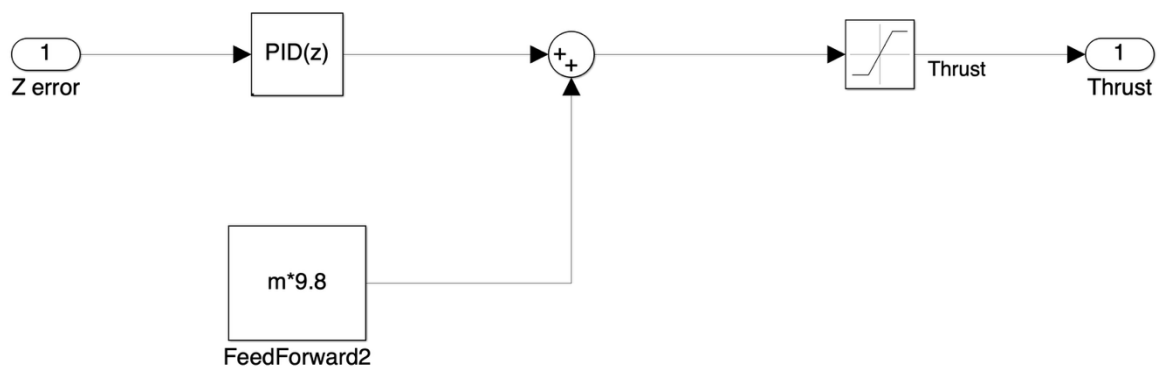
Додаток Б

Моделювання системи керування в Simulink

Блок Controller:



Блок Z controller (3 PID):



Додаток В

Результати роботи MoGe-VIO (візуалізація в Rerun)

