

М. Ю. Войтех¹
О. П. Кравченко¹

СИМУЛЯЦІЯ МОНІТОРИНГОВИХ МУЛЬТИАГЕНТНИХ СИСТЕМ

¹Національний університет «Київський авіаційний інститут»

Розглянуто аспекти симуляції моніторингових мультіагентних систем (МАС), що поєднують мобільні безпілотні літальні апарати (БПЛА) та стаціонарні безпроводові сенсорні мережі (БСМ). Актуальність роботи зумовлена тим, що системи типу БПЛА–БСМ одночасно охоплюють фізичну динаміку руху мобільних агентів, мережеві затримки, втрати пакетів, енергетичні обмеження сенсорних вузлів та задачі міжагентної координації, що ускладнює використання одного універсального симуляційного середовища. Метою роботи є систематизація підходів до симуляції таких систем та формування критеріальної схеми вибору програмного середовища залежно від рівня деталізації моделі, цільових метрик, обчислювальних витрат і можливості перенесення результатів у реальні умови. У роботі виділено низький, середній, високий та мультіагентний рівні моделювання, для яких визначено основні об'єкти симуляції, типові задачі та ключові метрики. Сформовано багатокритеріальну схему оцінювання придатності симуляційної платформи, що враховує точність відтворення цільових процесів, масштабованість, придатність до Sim-to-Real переходу, обчислювальні витрати та складність інтеграції. На основі варіантного порівняння програмних середовищ визначено доцільні напрями їх застосування для різних етапів дослідження: абстрактні або спеціалізовані середовища — для високорівневих і мультіагентних експериментів, робототехнічні 3D-симулятори — для інженерної перевірки вузлів, дискретно-подієві симулятори — для аналізу мережевих процесів. Для систем БПЛА–БСМ сформульовано гібридну схему організації моделювання, що передбачає використання Python або Mesa/NumPy для апробації високорівневих алгоритмів, Gazebo або Webots для інженерної верифікації мобільних вузлів та ns-3 або OMNeT++ для детального аналізу мережевих процесів. Практична значущість роботи полягає у формуванні методологічного каркасу для узгодження різних рівнів симуляції та зменшення розриву між абстрактним моделюванням і подальшою інженерною верифікацією моніторингових систем БПЛА–БСМ.

Ключові слова: екологічний моніторинг, мультіагентні системи, БПЛА, безпроводові сенсорні мережі (БСМ), ієрархічна симуляція.

Вступ

Базові архітектурні положення щодо застосування мультіагентних систем в екологічному моніторингу розглянуто в [1]. Загальні підходи до багаторівневого моделювання мультіагентних систем передбачають розділення середовища, логіки прийняття рішень та мережевої взаємодії агентів [2]. У цій роботі предмет дослідження звужено до задачі ефективної симуляції моніторингових систем типу БПЛА–БСМ, у яких мобільні безпілотні літальні апарати взаємодіють зі стаціонарними вузлами безпроводової сенсорної мережі. Для таких систем характерне одночасне поєднання фізичних, мережевих, ресурсних та координаційних процесів, що ускладнює використання одного універсального симуляційного середовища.

У системах БПЛА–БСМ стаціонарні сенсорні вузли виконують вимірювання параметрів середовища, локальне накопичення даних та їх передавання за наявності доступного каналу зв'язку, тоді як БПЛА можуть виконувати функції мобільних ретрансляторів, засобів збору даних або додаткової верифікації зон інтересу [3], [4]. Ефективність такого моніторингу визначається не лише маршрутом руху БПЛА, а й тривалістю перебування в зоні радіодоступу, затримками передавання, імовірністю втрати пакетів, енергетичним станом вузлів та здатністю агентів координувати дії в умовах неповної інформації.

Операційний цикл систем БПЛА–БСМ може бути узагальнено як послідовність етапів збору



сенсорних даних, їх передавання через безпроводову мережу, агрегування на рівні наземного або розподіленого обчислювального вузла та подальшого корегування маршрутів або параметрів взаємодії агентів [4], [5]. З позиції симуляції важливим є не лише відтворення цих етапів, а й урахування взаємного впливу між фізичним рухом БПЛА, параметрами радіоканалу, енергетичним станом сенсорних вузлів і логікою прийняття рішень.

Комп'ютерне моделювання таких систем є критично важливим інструментом проектування, оскільки дозволяє порівнювати алгоритми планування, комунікаційні протоколи та стратегії координації без значних матеріальних витрат і ризику втрати обладнання [5], [6]. Проте адекватна симуляція МАС потребує врахування реальних динамічних обмежень середовища та каналів зв'язку, які часто ігноруються в абстрактних теоретичних моделях [7]. Для ефективного моніторингу недостатньо лише просторової близькості вузлів; моделювання має враховувати кінематику та динаміку БПЛА, протокольні накладні витрати на встановлення з'єднання, імовірнісні втрати пакетів у безпроводовому каналі та мінімальний час перебування в зоні доступу, необхідний для завершення сеансу передавання даних. Окрім інформаційних обмежень, варто враховувати просторові фактори, зокрема зони заборони польотів, які можуть залишатися проникними для радіосигналів, але фізично обмежувати траєкторії руху БПЛА.

Складність ефективного моделювання МАС полягає в інтеграції декількох ієрархічних рівнів в єдине середовище комп'ютерного моделювання, а саме фізичного рівня (динаміка польоту БПЛА та збір сенсорної інформації в реальному часі), мережевого рівня (затримки передачі даних та втрати пакетів), ресурсного рівня (енергетичний бюджет автономних вузлів) та колективного рівня (логіка взаємодії великої кількості автономних одиниць). Тому актуальним є розгляд багаторівневої архітектури симуляції, де різні рівні можуть бути реалізовані у спеціалізованих платформах [8]. Ключовим завданням розробника є пошук компромісу між масштабованістю, точністю та можливістю подальшого перенесення алгоритмів на реальні апаратні платформи [9], [10] (табл.1).

Таблиця 1

Рівні ієрархічної симуляції моніторингової МАС

Рівень моделювання	Основний об'єкт моделювання	Типові задачі	Ключові метрики
Низький (Фізичний)	динаміка БПЛА, актуатори, контури керування, сенсорні моделі	стабільність польоту, точність трекінгу, стійкість системи керування	середньоквадратична похибка траєкторії, час перехідного процесу, миттєве енергоспоживання
Середній (Мережевий)	канали зв'язку, мережеві затримки, втрати пакетів, енергоресурс вузлів	забезпечення надійності доставки даних, оцінка якості каналів (lqi/rssi)	коефіцієнт доставки пакетів (pdr), середня затримка (latency), флуктуація затримки, залишковий енергозапас
Високий (Операційний)	алгоритми планування, логіка операційного циклу, стратегії покриття території	оптимізація маршрутів збору даних, адаптивне планування польотних завдань	відсоток покриття території, актуальність (свіжість) даних, загальний час виконання місії
Мультиагентний (Координаційний)	міжагентна взаємодія (БПЛА–БПЛА, БСМ–БСМ, БПЛА–БСМ)	розподіл завдань, децентралізоване узгодження рішень, підтримання зв'язності групи	показник впорядкованості (order), безпека дистанцій (safety), зв'язність графа мережі, масштабованість

Реалізація такого підходу вимагає детального аналізу ієрархії симуляції, вибору рівня деталізації моделі та узгодження інструментальних засобів з метою конкретного експерименту. У межах цієї роботи системи типу БПЛА–БСМ розглядаються як приклад моніторингових мультиагентних систем, для яких одночасно важливими є фізична динаміка мобільних агентів, мережеві процеси передавання даних, ресурсні обмеження стаціонарних вузлів та логіка міжагентної координації.

Метою статті є систематизація комплексного підходу до симуляції моніторингових мультиагентних систем типу БПЛА–БСМ та формування критеріальної схеми вибору програмного середовища залежно від рівня деталізації моделі, цільових метрик, обчислювальних витрат і можливості подальшого перенесення результатів у реальні умови.

Наукова новизна роботи полягає у систематизації методів гібридного моделювання систем БПЛА–БСМ. На відміну від попередньої роботи [1], де основну увагу приділено концепції мультиагентної системи екологічного моніторингу, у цій статті розглядається задача організації ефективного симуляційного процесу, вибору інструментальних засобів і узгодження різних рівнів моделювання. Отримані результати формують методологічну базу для узгодження різних рівнів моделювання, що дозволяє мінімізувати розрив між абстрактним алгоритмічним моделюванням та інженерною верифікацією вузлів мережі.

Ієрархічна модель симуляції моніторингової системи

Симуляцію моніторингової мультиагентної системи доцільно розглядати як ієрархію моделей, де складність зростає від моделювання окремого агента до моделювання системи в цілому, а ключовим обмеженням стає компроміс між точністю відтворення реальних процесів та обчислювальними витратами. У межах такого підходу виділяють чотири фундаментальні рівні моделювання: низький [11], [12], середній [6], [13], високий [4] та мультиагентний [5], [14].

1. Низький рівень (Low-level) фокусується на фізичних станах вузлів. Для стаціонарних сегментів це параметри обробки сигналів, для мобільних — динаміка польоту, робота актуаторів та реакція контурів керування на входні команди.

2. Середній рівень (Mid-level) охоплює процеси отримання даних із сенсорів та їх транспортування через безпроводові канали зв'язку з урахуванням мережових обмежень.

3. Високий рівень (High-level) описує виконання польотного завдання як цілісного процесу. Тут оцінюються глобальні метрики продуктивності: час повного покриття території та ефективність агрегації даних.

4. Мультиагентний рівень (Multi-agent level) зосереджений на протоколах колективної координації, алгоритмах децентралізованого уникнення зіткнень (колізій) та підтриманні зв'язності групи мобільних агентів під час виконання операційного циклу.

Симуляційні платформи можуть інтегруватися, утворюючи середовище спільного моделювання (co-simulation) [15], [16]. Наприклад, коли внутрішній симулятор (MATLAB/Simulink) виконує роль обчислювального ядра (логіка планування та керування), а зовнішній (Gazebo, FlightGear) відтворює тривимірне середовище, аеродинамічні збурення (вітер, турбулентність) та візуалізацію сценаріїв [17], [18]. Такий підхід збільшує навантаження на обчислювальну систему, проте є критично важливим для мінімізації невизначеності перед польовим випробуваннями у складних умовах довкілля.

Окрему групу складають підходи, що забезпечують перехід від теоретичних моделей до реальної експлуатації: а) Software-in-the-Loop (SIL) — віртуалізація коду керування в середовищі симуляції та б) Hardware-in-the-Loop (HIL) — підключення реального польотного контролера до симулятора, який імітує сигнали датчиків [9], [10], [19] (табл. 2).

Таблиця 2

Рівні симуляції, типові задачі та інструментальні засоби

Рівень ієрархії	Об'єкт моделювання	Основна задача симуляції	Набір інструментів
Низький (SIL/HIL)	аеродинаміка, актуатори, двигуни	перевірка стабільності польоту та стійкості керування під впливом збурень	MATLAB/Simulink, FlightGear, Gazebo
Середній (BCM)	сенсорні моделі, RSSI, флуктуація затримки	надійність збору даних та верифікація мережових протоколів	ns-3, OMNeT++
Високий (Операційний)	траєкторії, логістика, граф покриття	оптимізація часу моніторингу та актуальності даних	Python (Mesa), NetLogo
Мультиагентний (Координаційний)	алгоритми взаємодії, Swarm AI	уникнення колізій, самоорганізація mesh-мереж	ARGoS, ROS/Gazebo

Проектуючи МАС, вибір глибини деталізації симуляції безпосередньо залежить від балансу між достовірністю результатів та часовими ресурсами [8]. Технічно неможливо в межах однієї моделі з однаковою точністю відтворювати мікросекундні протокольні затримки та багатогодинні польотні завдання для великої групи вузлів одночасно. Для досягнення належної структуризації симуляційного моделювання є доцільним на високому та мультиагентному рівні використовувати абстрактне моделювання, коли потрібно виконати концептуальний дизайн, перевірити алгоритми рою (Swarm AI) та оптимізувати маршрути. Це дозволяє виконувати тисячі ітерацій за короткий проміжок часу та масштабувати систему до значної кількості функціональних одиниць. Проте існує ризик надмірної ідеалізації процесів, що може призвести до формування алгоритмів, нездатних до функціонування в реальних умовах [7]. На середньому рівні проводиться деталізована фізико-мережева симуляція для валідації протоколів передачі даних у BCM та оцінки енергоспоживання. На цьому етапі можна підвищити відповідність моделі реальним умовам передавання даних, проте різко зростають вимоги до обчислювальних потужностей (CPU/RAM) [5], [8]. Наприклад, моделювання 10 хвилин реального часу моніторингової місії може потребувати кількох годин серверних обчислень.

На низькому рівні застосовуються методи емуляції та технології HIL/SIL безпосередньо перед польовими випробуваннями. Це необхідно для перевірки надійності програмного забезпечення та контурів керування, а також для виявлення критичних помилок у режимі реального часу. Цей рівень

потребує найбільших обчислювальних витрат і унеможливорює симуляцію великих груп мобільних агентів через апаратні обмеження фізичного обладнання.

Низький рівень моделювання (фізичний рівень)

Низький рівень симуляції є критичним у випадках, коли достовірність результатів безпосередньо залежить від динаміки руху мобільних агентів, специфіки алгоритмів керування та обмежень, що накладаються функціонуванням виконавчих механізмів (актуаторів). На цьому етапі доцільним є використання модульних стендів, які поєднують нелінійні математичні моделі з імітаційними компонентами, що дозволяє збалансувати швидкість налаштування системи та її відповідність реальному об'єкту [12], [15]. Типовий архітектурний підхід передбачає використання середовища MATLAB/Simulink, де окремими блоками реалізуються генератор траєкторії, контури керування (регулятори), моделі виконавчих механізмів, нелінійна динамічна модель об'єкта [11]. Така структура забезпечує високу відтворюваність експериментів та можливість коректного порівняння результатів. З іншого боку, інтеграція з фізичними 3D-симуляторами (наприклад, Gazebo) додає візуальної та фізичної реалістичності (врахування зіткнень, властивостей поверхні). Проте варто зважати, що ітераційні чисельні методи обчислень у таких середовищах іноді створюють розрахункові артефакти, що може ускладнити прецизійне порівняння алгоритмів. Ключовим завданням цього рівня є верифікація здатності БПЛА витримувати задану траєкторію та аналіз стабільності апарата під впливом зовнішніх збурень. Особлива увага приділяється динаміці автопілота та налаштуванню регуляторів, що забезпечують стійкість системи у складних метеоумовах [15]. Високий ступінь відповідності моделі реальності досягається не лише завдяки 3D-візуалізації, а насамперед через глибоку інтеграцію динаміки польоту з реальною архітектурою програмного забезпечення керування [19], [20].

Базою моделі низького рівня (наприклад, для мультироторного БПЛА) є динамічна система з шістьма ступенями вільності (6-DOF) [12]. Сумарна тяга та обертальні моменти формуються силовими установками (роторними двигунами), а вектор стану літального апарата описується його просторовим положенням, кутовою орієнтацією, лінійною швидкістю та вектором кутових швидкостей. Ключовим технічним аспектом моделювання є розв'язання рівнянь руху для формування керуючих сигналів на двигуни, що забезпечує мінімізацію похибки позиціонування. Архітектура керування на низькому рівні симуляції реалізується за каскадним принципом, де внутрішній контур відповідає за високочастотну стабілізацію кутової орієнтації, зовнішній контур забезпечує керування вектором лінійної швидкості та просторовим положенням вузла, вхідний каскад регулюється генератором траєкторії або трекером. Така ієрархічна структура дозволяє проводити диференційований аналіз похибок стабілізації та навігації, уникаючи їх змішування у загальних системних метриках моніторингу. Робочий ланцюг типу «опорна точка–регулятор–верифікація» демонструє високу масштабованість і дозволяє інтегрувати різні типи алгоритмів керування від класичних ПД-регуляторів до сучасних методів керування у просторі станів.

Окремим аспектом моделювання низького рівня є узгодження часових масштабів математичної моделі, сенсорних систем та контурів керування. Навіть за умови високої адекватності динамічної моделі, результати симуляції можуть бути недостовірними, якщо частота дискретизації контролера та періодичність оновлення даних сенсорів не відповідають фізичному сценарію, або якщо ітераційні чисельні методи вносять розрахункові артефакти. Тому в сучасних інтегрованих модульних стендах акцент зміщується з розв'язання ізольованих диференціальних рівнянь руху на побудову цілісної програмної архітектури та коректну часову детермінованість взаємодії компонентів.

Архітектурно модульний стенд низького рівня поділяється на функціональні блоки динамічної моделі об'єкта (математичне ядро), виконавчого рівня (автопілот та каскадні контури керування), сенсорного рівня (імітація вимірювальних пристроїв з урахуванням шумів та затримок), віртуального середовища (фізичні умови та зовнішні збурення) та інтерфейсів інформаційного обміну (протоколи взаємодії між модулями).

Практична цінність такого розділення полягає у забезпеченні модульності системи через можливість заміни окремих компонентів без модифікації всього програмного комплексу. Наприклад, розробник може інтегрувати нову модель сенсора або альтернативний алгоритм керування, зберігаючи незмінними фізичне середовище та сценарій польотного завдання. У фреймворках на базі ROS (Robot Operating System) та Gazebo ці можливості реалізуються через механізм окремих вузлів (nodes) та плагінів [20], [21]. Це дозволяє нарощувати функціональність та змінювати сценарії моделювання шляхом конфігурування зв'язків між модулями, що забезпечує високу гнучкість і

масштабованість процесу розробки МАС.

Для забезпечення адекватності моделей низького рівня критично важливо розмежовувати верифікацію динаміки об'єкта та валідацію логіки функціонування автопілота. Одним із найдостовірніших методів наближення симуляції до реальних умов є застосування підходу SITL (Software-in-the-Loop) [9], [10]. У цьому сценарії керування здійснюється реальним програмним забезпеченням бортового комп'ютера (наприклад, прошивкою польотного контролера), що функціонує у віртуальному середовищі без залучення фізичного апарата. Обґрунтовано ефективність розширення середовища ROS/Gazebo для моделювання кооперативних груп БПЛА з використанням ПЗ з відкритим кодом (наприклад, ArduPilot) як базового контролера просторової орієнтації (attitude controller). Технічна реалізація масштабування на групу апаратів у такому разі здійснюється через виділення окремих мережових портів обміну даними для кожного віртуального вузла. Саме на етапі такої симуляції виявляються специфічні обмеження автопілота та інтерфейсів інформаційного обміну, які зазвичай ігноруються в абстрактних математичних моделях.

Після завершення валідації моделей низького рівня та контурів керування, їх інтегрують до систем вищого ієрархічного рівня. Ця логіка реалізується в інтегрованих середовищах моделювання, де детальна симуляція динаміки польоту та програмна архітектура керування слугують фундаментом для нарощування модулів міжагентної взаємодії, сценаріїв моніторингу та мережових ефектів. Такий підхід забезпечує керований перехід від аналізу стійкості та точності трекінгу до повномасштабних системних експериментів, зберігаючи при цьому високу відтворюваність результатів.

Середній рівень моделювання (мережовий та сенсорний рівні)

Середній рівень моделювання зосереджений на сенсорній компоненті та інформаційному обміні. Основними об'єктами дослідження тут є критичне енергоспоживання вузлів, затримки зв'язку, втрати пакетів, пропускна здатність та загальна надійність трансляції даних. У цьому контексті симулятори БСМ функціонують у дискретно-подієвій парадигмі (Discrete-Event Simulation, DES), що дозволяє ефективно моделювати протоколи та мережеві події без надлишкового обчислювального розгляду безперервної динаміки польоту [6]. Проте проблема виникає за потреби одночасного врахування кінематики БПЛА та мережових процесів у БСМ. Це зумовлює необхідність синхронізації підсистем, що використовують різні принципи квантування часу, а саме моделювання на основі подій (event-based) проти моделювання на основі неперервного часу (time-based) [22].

Середній рівень формалізує перехід від опису руху до моделювання процесів отримання та доставки інформації. Визначальними стають характеристики інформаційно-комунікаційних процесів, зокрема показники якості зв'язку та рівень завад, часові мітки та синхронізація вузлів, мережеві затримки та їх флуктуація, енергоспоживання сенсорних засобів та стійкість обміну даними в умовах мобільності. З цієї позиції середній рівень доцільно розглядати як частину ітераційного циклу «отримання–відгук–комунікація». Трасекторії або алгоритми, що є коректними з погляду динаміки польоту, можуть виявитися неефективними через обмеження пропускної здатності або специфіку мережових протоколів [13].

Сенсори в моделі середнього рівня описуються через сукупність інформаційних параметрів (шум, зміщення, смуга пропускання, квантування, частота дискретизації), синхронізованих з часом симуляції. У багатокомпонентних системах це реалізується у вигляді автономного сенсорного шару, інтегрованого над фізичними процесами та контурами керування. Така архітектура дозволяє проводити заміну моделей сенсорів без модифікації всього програмного комплексу.

Мережева складова середнього рівня включає моделі каналу зв'язку та поширення сигналу. У випадку БСМ моделювання виконується у дискретно-подієвій парадигмі, оскільки саме таким чином найзручніше описувати процеси передачі пакетів, таймери протоколів та активацію сплячих режимів для агентів з метою економії енергетичних ресурсів [6]. Сумісність різних способів моделювання реалізується на середньому рівні через створення інтерфейсу між відповідними шарами для передачі даних про динаміку руху (позиція, швидкість та орієнтація літального апарату) та події у мережі (передача, прийом, затримка, втрата інформації) [16], [22].

Існує декілька репрезентативних підходів до інтеграції компонентів моделювання середнього рівня, які демонструють різні стратегії поєднання фізичних та інформаційних процесів. Інтегровані архітектури передбачають створення єдиного програмного середовища, що поєднує тривимірну візуалізацію, фізичну модель руху та мережевий стек. Це дозволяє розглядати сенсори та засоби зв'язку як невід'ємні компоненти архітектури системи. Синхронізовані платформи (на прикладі

FlyNetSim) акцентують увагу на часовій синхронізації спеціалізованого мережевого симулятора (наприклад, ns-3) з середовищем SITL (наприклад, ArduPilot) [13]. Такий підхід дозволяє досліджувати перехресний вплив, а саме, яким чином мережеві затримки та втрати пакетів корегують поведінку автопілота, і навпаки, яким чином маневри та зміна просторового положення вузла впливають на параметри радіоканалу.

Концепція цифрового двійника є сучасним рівнем розвитку MAC, що базується на тісній інтеграції Gazebo (фізика та сенсорики) з ns-3 (пакетне моделювання мережі) [16]. Це дозволяє одночасно моделювати фізичну взаємодію вузлів та динаміку каналу зв'язку на рівні окремих пакетів для багатокomпонентних сценаріїв.

Симуляцію середнього рівня доцільно інтерпретувати як рівень, де адекватність моделі визначається в першу чергу коректністю відтворення інформаційно-комунікаційних та причинно-наслідкових залежностей, а не точністю аеродинаміки чи кінематики польоту. Критичним є те, чи відтворює симуляція вплив руху платформи та особливості радіоканалу на формування параметрів зв'язку, які визначають доступність і повноту даних, які, в свою чергу, безпосередньо відображаються на метриках якості моніторингу. Якщо метою симуляції є отримання висновків щодо протокольних механізмів, режимів енергоспоживання та надійності доставки даних, моделювання на пакетному рівні виконується у дискретно-подієвій парадигмі [6], [3].

Високий рівень моделювання (операційний рівень)

Високий рівень симуляції описує операційний цикл з точки зору функціонування системи як одного цілого. На цьому рівні розв'язується задача планування такого циклу та здійснюється оцінка його виконання. Це включає вибір маршрутів і точок збору, розподіл ресурсів у часі, визначаються метрики покриття та ресурсні витрати на виконання операційного циклу. Моделювання на цьому рівні базується на спрощених кінематичних моделях (наприклад, модель матеріальної точки — point-mass), що забезпечує високу масштабованість [20]. Це дозволяє виконувати велику кількість ітерацій для отримання статистично надійних висновків на множині сценаріїв, жертвуючи при цьому деталізацією польотної динаміки. Пріоритетом стає не прецизійна точність окремого маневру, а верифікація логіки прийняття рішень у межах цільового польотного завдання. У центрі симуляції високого рівня знаходяться: 1) формулювання задачі моніторингу (цільові параметри та об'єкти), 2) просторово-часова структура невизначеності середовища та 3) механізми координації групи БПЛА з урахуванням обмежень щодо залишкового енергозапасу, тривалості польоту та пропускної здатності каналів зв'язку [23]. Середовище подається у вигляді цифрової карти з перешкодами та зонами заборони польотів (NFZ), а об'єкти спостереження моделюються як події або динамічні поля пріоритетів. Симуляція реалізує ітеративний цикл «перцепція – оновлення карти – перепланування», а ефективність оцінюється за інтегральними показниками якості операційного циклу (місії). Універсальні середовища (наприклад, V-REP) часто використовуються для верифікації таких високорівневих стратегій завдяки їх гнучкості у налаштуванні сценаріїв [20].

Типовою задачею високого рівня для екологічного або інфраструктурного моніторингу є збір даних із стаціонарної наземної БСМ з використанням БПЛА як мобільного агента, що відповідає за збір та трансляцію відповідної інформації. У місії такого типу процес структурується на етапи Оцінка просторового розміщення стаціонарних вузлів, кластеризації мережі з визначенням головних вузлів, з подальшим плануванням оптимального маршруту збору даних [4]. Принциповою особливістю високого рівня симуляції є те, що траєкторія аналізується не лише за геометричними критеріями (довжина шляху), а й за здатністю забезпечити критичний час перебування в зоні доступу для гарантованого завершення сеансу передачі даних. Додавання обмеження на мінімальний час контакту наближає місійне моделювання до реальних експлуатаційних умов систем БПЛА–БСМ. Інший клас задач високорівневого моделювання охоплює пошук та виявлення об'єктів на великих територіях в умовах просторово-часової невизначеності, наприклад, пріоритетність пошуку може задаватися у вигляді часозалежного просторового розподілу. В цьому випадку колективна стратегія групи БПЛА формулюється як задача наближення розподілу, що індукується траєкторіями вузлів, до еталонного розподілу пріоритетів. Для кількісної оцінки якості покриття використовується метрика Вассерштейна (Wasserstein distance), що дозволяє формалізувати компроміс між рівномірним скануванням та адаптивним пошуком на основі поточних спостережень [4]. Тобто, симуляція високого рівня такого класу задач повинна відтворювати динаміку пріоритетного поля та демонструвати операційний вииграш через ефективніший розподіл просторово-часових ресурсів, а не за рахунок

збільшення інтенсивності польотів. Окремим напрямком розвитку операційного рівня є створення симуляторів для систем керування трафіком (Unmanned Traffic Management — UTM), де моделюється одночасне функціонування великої кількості незалежних агентів у спільному повітряному просторі [18]. Такі системи вимагають інтеграції алгоритмів деконфліктизації та дотримання правил польотів на рівні всієї інфраструктури.

Симуляцію високого рівня доцільно розглядати як дискретний процес прийняття рішень у часі, де стан системи визначається поточною картою покриття (пріоритетів), ресурсним станом вузлів (енергозапас, час до завершення польоту) та мережевим станом доступності каналів зв'язку). Оцінювання ефективності при цьому переходить від одиничних показників до багатовимірного профілю «результативність–витрати», де одночасно враховуються ступінь покриття, актуальність інформації та питомі витрати ресурсів [18], [23].

Мультиагентний рівень симуляції (рівень координації)

Мультиагентний рівень описує колективну динаміку системи як сукупності взаємодіючих мобільних та стаціонарних агентів (вузлів). На цьому рівні якість моніторингу визначається не лише індивідуальними характеристиками БПЛА, а насамперед механізмами координації, протоколами обміну даними та алгоритмами децентралізованого узгодження дій. На цьому рівні симуляції моделюються функціональні ролі між агентами, координація в умовах обмеженого інформаційного обміну, уникнення колізій, підтримання зв'язності та вплив топології мережі на загальну системну ефективність [5], [14]. Взаємодія на цьому рівні описується часово-змінним графом зв'язності, структура якого детермінується просторовою конфігурацією вузлів та характеристиками безпроводових каналів. На мультиагентному рівні доцільно виділити три основні типи міжагентної взаємодії: БПЛА–БПЛА, який забезпечує координацію мобільних сегментів (розподіл зон сканування, узгодження траєкторій, підтримання формації та уникнення конфліктів руху), БСМ–БСМ, який описує внутрішньомережеву кооперацію стаціонарних вузлів та БПЛА–БСМ, тип міжагентної взаємодії, який забезпечує екстракцію даних із сенсорної мережі, адаптацію маршрутів БПЛА до стану вузлів та доставку інформації до наземного шлюзу.

Для відтворюваного порівняння алгоритмів використовують формалізовані метрики колективної поведінки, що переводять якісні характеристики у кількісні показники та дозволяють зіставляти результати експериментів, виконаних у різних сценаріях. До такого стандартизованого набору об'єктивних показників належать: 1) показник впорядкованості (order), який регулює ступінь узгодженості векторів стану вузлів; 2) показник безпеки дистанцій (safety), який фіксує відсутність порушень мінімально допустимих відстаней (колізій); 3) показник цілісності групи (union), який характеризує здатність системи функціонувати як єдиний кластер; 4) показник зв'язності (connectivity), який регулює здатність графа мережі забезпечувати передачу даних між усіма сегментами [8].

У процесі розробки доцільно розрізняти алгоритмічне моделювання, де агенти описуються спрощеними моделями та керуються абстрактними правилами та інженерне моделювання, у якій координація агентів тестується разом з автопілотом, програмною архітектурою системи та комунікаційною підсистемою.

Так реалізовано підхід на базі ROS/Gazebo з використанням ArduPilot як базового контролера, де координаційний алгоритм реалізується як окремий модуль, що спостерігає за іншими агентами та формує команди керування [17], [24]. Це дає змогу перейти від абстрактного опису колективної поведінки до перевірки її працездатності з використанням реальних програмно-апаратних засобів. У мультиагентних системах якість координації тісно пов'язана з характеристиками комунікацій, тому моделі без реалістичної мережевої складової часто дають завищену оцінку ефективності. В іншому випадку FlyNetSim реалізує синхронізовану сумісну симуляцію ArduPilot SITL та ns-3, щоб затримки та втрати пакетів впливали на поведінку автопілота, а зміни руху, у свою чергу, впливали на параметри каналу. Аналогічну ідею на рівні «цифрового двійника» розвиває Gazebo/NS3, який поєднує фізичну симуляцію та пакетний рівень мережі для багатоагентних сценаріїв, що дозволяє одночасно аналізувати зіткнення та порушення зв'язності групи мобільних агентів. Кожен екземпляр автопілота має використовувати унікальні ідентифікатори, порти та канали телеметрії, інакше виникатимуть конфлікти станів та помилки, які на зовнішньому рівні можуть виглядати як порушення координації. У документації ArduPilot SITL та PX4 multi-vehicle simulation ці аспекти описані як необхідні умови коректної роботи багатьох одночасних сценаріїв взаємодії [8], [17]. Додатковими обмеженнями мультиагентної симуляції є масштабованість обчислень та коректна синхро-

нізація часу. Зі зростанням кількості агентів обчислювальні витрати суттєво зростають особливо за необхідності узгодженості часових кроків між компонентами моделі. Тому виправданим є використання двоетапної схеми моделювання, де спочатку виконуються масові запуски на високорівневих абстрактних моделях для відбору перспективних стратегій, після чого обмежену кількість експериментів проводять в інтегрованих середовищах, щоб перевірити стійкість координації до мережних затримок та апаратних обмежень [14].

Методика експериментів, верифікація та відтворюваність

Методика проведення експериментів у моделюванні моніторингових систем має базуватися на формалізації дослідницької гіпотези та визначенні того рівня ієрархії, на якому ця гіпотеза набуває значущості, а не на апіорному виборі програмного інструментарію. Якщо перевіряється стійкість керування та точність трекінгу, визначальним є низькорівневий опис динаміки та контурів керування. Якщо мета полягає в оцінюванні надійності збору та трансляції даних, використовується середній рівень із залученням сенсорних та мережних ефектів, зокрема флуктуацію затримки та втрати пакетів. Для аналізу ефективності покриття та стратегій планування достатнім є високий рівень з використанням абстрактних моделей операційного циклу. Дослідження колективної динаміки та децентралізованого управління вимагає використання мультиагентного рівня для аналізу зв'язності, уникнення колізій та розподілу завдань між сегментами системи. Такий методологічний підхід формує основу для створення відтворюваного експериментального каркасу. У межах цієї структури вибрана деталізація моделі є функціонально необхідною і достатньою для отримання валідних висновків, що відповідають цілям симуляції. Це дозволяє розробнику або проектувальнику МАС уникати надлишкових обчислювальних витрат на ідеалізованих етапах та забезпечувати необхідну точність у критичних сегментах розробки.

Для забезпечення метрологічної коректності експериментів у процесі моделювання необхідно чітко розмежовувати керовані фактори та некеровані збурення, аналізуючи їхній вплив на цільові метрики оцінювання. До керованих факторів належать: 1) параметри алгоритмів, що включають коефіцієнти регуляторів та вагові коефіцієнти функцій вартості; 2) кількість БПЛА, що визначає масштабованість системи; 3) інтенсивність мережевого трафіку; 4) моделі каналів, їх пропускна здатність та вибрані протоколи зв'язку; 5) частота дискретизації сенсорів та періодичність міжагентного обміну даними. До некерованих збурень, що вносять елемент невизначеності відносяться: 1) зовнішні впливи середовища; 2) шуми та зміщення вимірювань, пов'язані з інструментальними похибками; 3) стохастичні варіації початкових умов; 4) випадкове просторове розміщення цільових об'єктів або стаціонарних вузлів БСМ. Вихідні метрики повинні бути функціонально узгоджені з дослідницькою гіпотезою та визначені у такий спосіб, щоб результати експериментів були відтворюваними та придатними до статистичного порівняння. Тільки за умови такого розподілу параметрів можливо коректно оцінити чутливість алгоритмів до зовнішніх факторів та визначити межі працездатності мультиагентної системи моніторингу в умовах, наближених до експлуатаційних.

У межах забезпечення якості моделювання необхідно чітко розмежовувати верифікацію (Verification), де підтверджується правильність програмної реалізації математичної моделі (відповідь на питання «чи правильно побудована система?») та валідацію (Validation), коли доводиться, що модель достатньо точно відображає фізичну реальність для досягнення конкретної мети дослідження (відповідь на питання «чи ту систему ми побудували?») [7]. Систематична схема забезпечення достовірності включає перевірку концептуальної адекватності, верифікацію програмної реалізації, оцінювання операційної валідності та документування обмежень, у яких висновки моделювання залишаються чинними. Для підвищення надійності результатів на низькому рівні ієрархії доцільно застосовувати крос-верифікацію моделей за ідентичних вхідних впливів. Зокрема, порівняння динамічного відгуку математичної моделі в середовищі MATLAB/Simulink з відгуком у робототехнічному середовищі ROS/Gazebo, за умови використання ідентичного контуру керування, що дозволяє відокремити похибки реалізації від властивостей самого алгоритму. Такий порівняльний аналіз дає змогу верифікувати узгодженість динаміки та часових характеристик системи перед переходом до інтерпретації результатів у метриках якості керування та ефективності моніторингу. Це забезпечує надійний фундамент для подальшого масштабування системи до мультиагентного рівня.

Окремою вимогою до методики проведення експериментів є впровадження методів спільного моделювання (co-simulation) та проміжних етапів тестування у контурі SIL (Software-in-the-Loop) та HIL (Hardware-in-the-Loop). Ці підходи виступають ключовим інструментом мінімізації розриву між

абстрактними математичними моделями та реальними польовими випробуваннями. Зокрема, інтеграція середовища MATLAB/Simulink з професійними авіаційними симуляторами (такими як X-Plane або FlightGear) дозволяє верифікувати контури керування та алгоритми навігації у високоточному тривимірному середовищі, враховуючи складні стохастичні збурення та проводити ітераційне налаштування параметрів регуляторів. Для забезпечення наукової достовірності та відтворюваності результатів, симуляційна платформа та експериментальний стенд повинні відповідати визначеним технічним вимогам, що включають наскрізне логування, детермінованість стохастичних процесів та конфігураційну цілісність.

Критеріальна схема вибору симуляційної платформи

Вибір симуляційної платформи для моніторингових МАС доцільно розглядати як багатокритеріальну задачу, у якій платформа оцінюється не ізольовано, а відносно типу експерименту та рівня моделювання. Для платформи p_i та експериментальної задачі e_j узагальнена оцінка придатності може бути подана у вигляді

$$Q(p_i, e_j) = w_A A_i + w_S S_i + w_R R_i - w_C C_i - w_I I_i, \quad (1)$$

де A_i — точність відтворення цільових процесів, S_i — масштабованість, R_i — придатність до перенесення результатів у реальні умови, C_i — обчислювальні витрати, I_i — складність інтеграції з іншими компонентами симуляційного середовища, а w_A, w_S, w_R, w_C, w_I — вагові коефіцієнти, які визначаються метою експерименту.

Для низькорівневої верифікації БПЛА пріоритетними є точність фізичної моделі та придатність до подальшого переходу в режими SITL/HIL, тому зростає значення коефіцієнтів w_A та w_R . Для мережевого рівня визначальними стають коректність моделювання затримок, втрат пакетів, пропускну здатності та енергоспоживання вузлів. Для високорівневих і мультиагентних експериментів основне значення має масштабованість, можливість багаторазового запуску сценаріїв та статистичного порівняння результатів, тому зростає вага w_S , тоді як надмірна фізична деталізація може розглядатися як недоцільна.

Отже, компроміс між точністю, масштабованістю та обчислювальними витратами не є універсальним для всіх етапів моделювання. Він визначається дослідницькою гіпотезою: для аналізу стійкості польоту потрібна деталізована фізична модель, для аналізу мережевої надійності — пакетна симуляція, а для перевірки стратегій координації великої кількості агентів — абстрактна або гібридна модель з контрольованим рівнем деталізації.

Порівняння симуляційних платформ

Застосування наведеної критеріальної схеми дозволяє перейти від загального опису симуляційних платформ до їх варіантного вибору залежно від типу експерименту. У межах моніторингових МАС платформа не може розглядатися як універсально краща або гірша, оскільки її придатність визначається співвідношенням між точністю відтворення цільових процесів, масштабованістю, обчислювальними витратами, складністю інтеграції та можливістю подальшого перенесення результатів у реальні умови. Тому порівняння платформ доцільно виконувати не лише за їх функціональними можливостями, а й за відповідністю конкретному рівню ієрархічної симуляції та дослідницькій задачі.

Webots доцільно розглядати як збалансовану платформу загального призначення для інженерних перевірок у задачах середньої складності, де важливими є наявність готових моделей, сенсорів, фізичного рушія та відносно швидке розгортання відтворюваних сценаріїв [5], [21]. Водночас ARGoS є придатнішим для сценаріїв, у яких головним обмеженням є масштабування до великої кількості агентів та проведення масових ітераційних експериментів [8], [4].

Використання середовища Gazebo є доцільним у випадках, коли необхідна глибока інтеграція з екосистемою ROS та високоточне моделювання тривимірних сценаріїв з урахуванням фізичної взаємодії об'єктів і сенсорних систем [17], [24]. Проте впровадження цієї платформи потребує врахування значних обчислювальних витрат, які стають критичними у разі масштабування системи до великої кількості мобільних агентів та підвищення складності сенсорного шару [5], [8]. У порівняльних дослідженнях Gazebo визначається як ефективний інструмент для прототипування робототехнічних систем, що водночас має суттєві обмеження щодо навантаження на центральний процесор (CPU) у мультиагентних сценаріях [8]. Ефективність застосування Gazebo в інтегрованих середовищах підтверджується публікаціями, де зв'язка ROS/Gazebo використовується як базова платформа для

моделювання кооперативних груп БПЛА [16], [17] та вирішення завдань організації повітряного руху в мульти-БПЛА середовищах [18].

Застосування CoppeliaSim (V-REP) є доцільним у сценаріях, що потребують гнучкого проектування тривимірних сцен, підтримки різноманітних режимів фізичного моделювання та наявності розвинених інтерфейсів для реалізації скриптової автоматизації керування [20]. Водночас, під час масштабування у випадку мультиагентних сценаріїв та проведенні масових ітераційних експериментів, високі вимоги платформи до обчислювальних ресурсів можуть стати визначальним обмежувальним фактором [5], [8].

Використання середовища ARGoS є найобґрунтованішим у сценаріях, де основними вимогами є масштабованість до сотень або тисяч агентів та забезпечення високого ступеня контрольованості експериментів з алгоритмами групового інтелекту [14]. Архітектура цієї платформи оптимізована для високопродуктивних паралельних обчислень, що дозволяє реалізовувати масові симуляційні ітерації. Проте висока обчислювальна ефективність досягається за рахунок спрощення тривимірної геометрії та обмеження функціональних можливостей щодо моделювання прецизійних сенсорних систем [5], [14].

У випадках, коли в польотних завданнях моніторингу пріоритетними є задачі комп'ютерного зору або виникає потреба у генерації фотореалістичних синтетичних даних, доцільно використовувати середовища на базі рушія Unreal Engine, зокрема платформу AirSim [19].

Для забезпечення адекватності цих моделей та їх відповідності реальним умовам польоту критично важливим є використання методів верифікації та валідації [7], а також залучення підходів Software-in-the-Loop (SITL) та Hardware-in-the-Loop (HIL) [9], [10], що дозволяють мінімізувати розрив між симуляцією та реальною експлуатацією [12], [15]. Особливого значення набувають гібридні архітектури, такі як FlyNetSim та GazeboNS3, які поєднують робототехнічні та мережеві аспекти для комплексного аналізу MAC [13], [16], [22], [23], що знаходять своє застосування в актуальних задачах екологічного моніторингу [4] та аналітичних оглядах сучасних засобів моделювання [6], [11].

У табл. 3 подано якісне узагальнювальне порівняння платформ і середовищ, що застосовуються на різних рівнях симуляції. Оцінки мають експертно-аналітичний характер і сформовані на основі узагальнення джерел.

Таблиця 3

Порівняльна характеристика MAC симуляційних платформ

Назва платформи	Рівень симуляції	Обчислювальні витрати (CPU)	Масштабованість	Точність, реалізм	Складність перенесення результатів (Sim-to-Real)	Посилання
MATLAB/Simulink + FlightGear / X-Plane	низький динамічний	високі	низька	висока фізика польоту	середня	[11], [12], [15], [19]
ns-3 / OMNeT++	середній мережевий	середні	висока	висока мережеві процеси	середня	[6], [13], [22]
Python (Mesa), NetLogo	високий операційний	низькі	висока	низька абстракція	висока	[4], [23]
ARGoS	мультиагентний високий	низькі–середні	висока	середня	висока	[8], [14]
Webots	низький мультиагентний	середні	середня	середня–висока	середня	[5], [21]
Gazebo	низький мультиагентний	високі	середня	висока	середня	[5], [8], [17], [18], [24]
CoppeliaSim (V-REP)	низький високий	середні–високі	середня	середня	середня	[8], [20]
AirSim	низький високий	високі	низька–середня	висока візуальні дані	середня–висока	[19]
FlyNetSim	середній мультиагентний	високі	середня	висока інтегрована	середня	[13]
GazeboNS3	середній мультиагентний	високі	середня	висока інтегрована	середня	[13], [16]

Подане порівняння має узагальнювальний характер, оскільки детерміновані значення параметрів суттєво варіюються залежно від архітектури конкретного сценарію, кількості агентів, ступеня

деталізації сенсорного шару та конфігурації обчислювальної інфраструктури. Водночас таблиця демонструє фундаментальну закономірність: існує зворотна кореляція між рівнем симуляції та масштабованістю системи. Збільшення фізичної достовірності неминуче призводить до зростання обчислювальних витрат, тоді як високорівневі мультиагентні середовища орієнтовані на проведення масових ітераційних експериментів шляхом цілеспрямованої абстракції фізичних та сенсорних процесів.

Методологічне правило вибору інструментарію для задач моніторингу довкілля можна сформулювати через релевантність етапу дослідження, а саме:

1) на етапі концептуального проектування найефективнішим є ARGoS, який застосовується для високорівневих досліджень, де визначальними є статистична значущість результатів та розмір групи агентів;

2) на етапі інженерної верифікації використовується Gazebo або Webots, коли виникає необхідність тестування сенсорних систем у тривимірному просторі та інтеграції з екосистемою ROS;

3) на етапі гнучкого прототипування пріоритет надається CoppeliaSim, де використовуються сценарії з обмеженою кількістю агентів, які потребують швидкої адаптації параметрів середовища;

4) на етапі розробки алгоритмів комп'ютерного зору середовище AirSim варто розглядати як спеціалізований інструмент для генерації фотореалістичних даних і не використовувати його у ролі універсальної платформи для моделювання загальної логіки місії.

Отже, для узагальнення запропонованого гібридного підходу на рис. 1 показано структуру комплексного симуляційного середовища для систем БПЛА–БСМ. Схема має концептуальний характер і відображає розподіл функцій між рівнями моделювання: високорівневий алгоритмічний шар використовується для планування місій та аналізу стратегій координації, робототехнічний 3D-симулятор — для інженерної перевірки поведінки мобільних вузлів і сенсорних моделей, а мережевий симулятор — для аналізу затримок, втрат пакетів, пропускну здатності та енергоспоживання вузлів БСМ. Взаємодія між рівнями може бути організована через інтерфейси обміну командами, телеметрією та параметрами стану середовища.

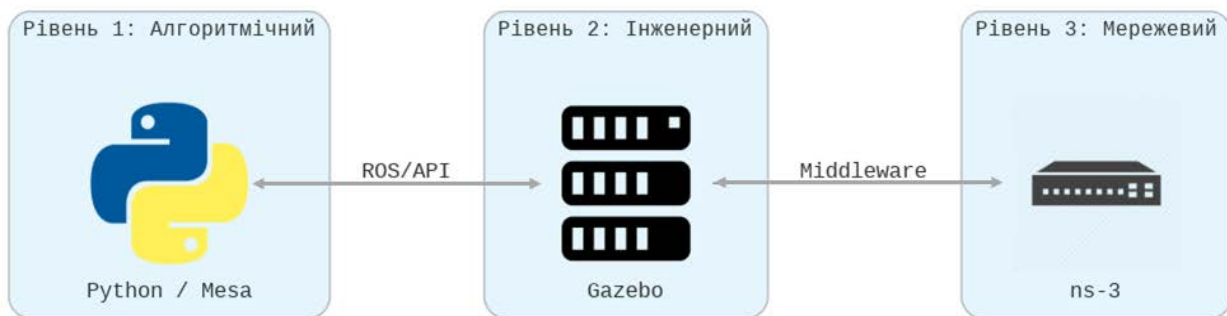


Рис. 1. Структура гібридного симуляційного середовища для БПЛА–БСМ систем

Аналіз доцільності використання такого підходу ґрунтується на порівняльному оцінюванні характеристик окремих інструментальних середовищ (табл. 4).

Таблиця 4

Порівняльний аналіз інструментальних середовищ для гібридного моделювання

Критерій	Python (Mesa)	Gazebo	ns-3	Гібридний підхід
Точність моделювання	низька–середня, залежно від деталізації моделі	висока для фізики, сенсорики та 3d-взаємодії	висока для мережевих процесів	комплексна
Масштабованість	дуже висока	низька–середня	середня	середня-висока
Складність налаштування	низька	висока	висока	середня-висока
Sim-to-Real здатність	низька	середня–висока	середня для мережевої частини	середня-висока

Відповідно до табл. 4, жоден із сучасних симуляторів не забезпечує одночасно високої точності моделювання фізики, мережевих протоколів та масштабованості для МАС. Тому запропонована гібридна архітектура виступає прийнятним технічним рішенням, що дозволяє поєднати переваги спеціалізованих платформ.

Окрім якісних критеріїв, поданих у табл. 4, для обґрунтування ефективності пропонованого

підходу проведено оцінювання витрат ресурсів на етапах розгортання та налаштування симуляційного середовища. Порівняння проводилося за двома параметрами: часом налаштування системи (у годинах) та загальною складністю реалізації (в умовних одиницях від 1 до 10). Результати порівняльного аналізу показано на рис. 2.

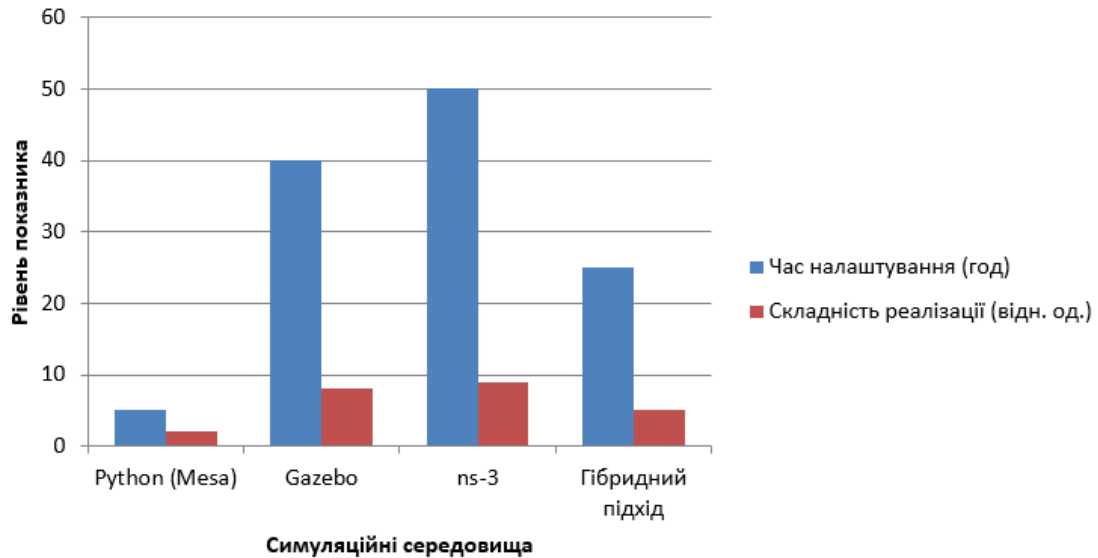


Рис. 2. Порівняння інструментальних середовищ за часом налаштування та складністю реалізації

Гістограма демонструє, що використання гібридного підходу дозволяє суттєво зменшити складність інтеграції (порівняно з комплексними інженерними симуляторами) зі збереженням високої точності моделювання кожного окремого шару системи. Емпіричні значення, використані для побудови діаграми (рис. 2), отримані шляхом експертної оцінки та узагальнення технічної документації відповідних програмних середовищ. Час налаштування включає середню тривалість встановлення середовища, конфігурацію зв'язків між вузлами та написання базових скриптів взаємодії для типового сценарію моніторингу МАС. Складність реалізації оцінюється на основі аналізу обсягу програмного коду, необхідного для розгортання початкового робочого середовища.

Висновки

У результаті дослідження систематизовано комплексну структуру середовища для симуляції моніторингових мультиагентних систем типу БПЛА–БСМ шляхом виділення низького, середнього, високого та мультиагентного рівнів моделювання, причому для кожного рівня визначено основні об'єкти моделювання, типові задачі та цільові метрики, що дозволяє узгоджувати рівень деталізації моделі з дослідницькою гіпотезою.

Запропоновано критеріальну схему вибору симуляційної платформи, яка враховує точність відтворення цільових процесів, масштабованість, придатність до перенесення результатів у реальні умови, обчислювальні витрати та складність інтеграції. Такий підхід дозволяє розглядати вибір програмного середовища як багатокритеріальну задачу, а не як вибір універсально кращого симулятора. На основі варіантного порівняння симуляційних платформ визначено доцільні напрями їх застосування для різних рівнів моделювання. Для високорівневих та мультиагентних експериментів доцільно використовувати абстрактні або спеціалізовані середовища, які орієнтовані на масштабованість і багаторазові запуски сценаріїв; для інженерної верифікації окремих вузлів — тривимірні робототехнічні симулятори з підтримкою сенсорних моделей та інтеграції з автопілотом; для мережевого аналізу — дискретно-подієві симулятори, що дозволяють оцінювати затримки, втрати пакетів, пропускну здатність та енергоспоживання вузлів.

Визначено вимоги до відтворюваності симуляційних експериментів, зокрема фіксацію початкових умов, контроль стохастичних процесів, наскрізне логування параметрів, документування конфігурації моделі та розмежування процедур верифікації й валідації. Практична значущість роботи полягає у формуванні методологічного каркасу, який може бути використаний для зменшення розриву між абстрактною симуляцією та подальшою інженерною перевіркою моніторингових систем БПЛА–БСМ.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] М. Ю. Войтех, і О. П. Кравченко, «Концепція мультиагентної системи в екологічному моніторингу», *Вісник Вінницького політехнічного інституту*, № 4, с. 49-57, 2025. <https://doi.org/10.31649/1997-9266-2025-181-4-49-57>.
- [2] Kh. Aloui, A. Guizani, M. Hammadi, T. Soriano, M. Haddar, "System Level Specification and Multi-agent Simulation of Manufacturing Systems," in Ben Amar, M., Bouguecha, A., Ghorbel, E., El Mahi, A., Chaari, F., Haddar, M. (eds) *Advances in Materials, Mechanics and Manufacturing II*. A3M 2021. *Lecture Notes in Mechanical Engineering*. Springer, Cham. https://doi.org/10.1007/978-3-030-84958-0_4.
- [3] L. Jia, and Y. Pei, "Recent Advances," in *Multi-Agent Reinforcement Learning for Intelligent Automation and Control of Water Environment Systems*. *Machines* 2025, 13, 503. <https://doi.org/10.3390/machines13060503>.
- [4] R. H. Kabir, and K. Lee, "Wildlife monitoring using a multi-UAV system with optimal transport theory," *Applied Sciences*, vol. 11, p. 4070, Apr. 2021. <https://doi.org/10.3390/app11094070>.
- [5] Z. Chen, J. Yan, B. Ma, K. Shi, Q. Yu, and W. Yuan, "A survey on open-source simulation platforms for multi-copter UAV swarms," *Robotics*, vol. 12, p. 53, Apr. 2023. <https://doi.org/10.3390/robotics12020053>.
- [6] G. H. Adday, S. K. Subramaniam, Z. A. Zukarnain, and N. Samian, "Investigating and analyzing simulation tools of wireless sensor networks: A comprehensive survey," *IEEE Access*, vol. 12, pp. 22938-22977, 2024. <https://doi.org/10.1109/ACCESS.2024.3362889>.
- [7] R. G. Sargent, "Verification and validation of simulation models," *Journal of Simulation*, vol. 7, pp. 12-24, Feb. 2013. <https://doi.org/10.1057/jos.2012.20>.
- [8] L. Pitonakova, M. Giuliani, A. Pipe, and A. Winfield, "Feature and performance comparison of the V-REP, Gazebo and ARGoS robot simulators," in *Towards Autonomous Robotic Systems*. Cham, Switzerland: Springer International Publishing, 2018, pp. 357-368. https://doi.org/10.1007/978-3-319-96728-8_30.
- [9] "SITL Simulator (Software in the Loop)," *ArduPilot Dev Documentation*, 2026. [Electronic resource]. Available: <https://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html>.
- [10] "Simulation," PX4 User Guide, 2026. [Electronic resource]. Available: <https://docs.px4.io/main/en/simulation>.
- [11] E. Soria, F. Schiano, and D. Floreano, "SwarmLab: A Matlab drone swarm simulator," in *Proc. 2020 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2020. <https://doi.org/10.1109/IROS45743.2020.9340854>.
- [12] O. Bouaïss, R. Mechgoug, and R. Ajgou, "Modeling, control and simulation of quadrotor UAV," in *Proc. 2020 1st Int. Conf. Commun., Control Syst. Signal Process. (CCSSP)*, 2020. <https://doi.org/10.1109/CCSSP49278.2020.9151687>.
- [13] S. Baidya, Z. Shaikh, and M. Levorato, "FlyNetSim: An open source synchronized UAV network simulator based on ns-3 and ArduPilot," in *Proc. 21st ACM Int. Conf. Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2018. <https://doi.org/10.1145/3242102.3242118>.
- [14] C. Pinciroli, et al. "ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems," *Swarm Intelligence*, vol. 6, pp. 271-295, Nov. 2012. <https://doi.org/10.1007/s11721-012-0072-5>.
- [15] N. Horri, and M. Pietraszko, "A tutorial and review on flight control co-simulation using Matlab/Simulink and flight simulators," *Automation*, vol. 3, pp. 486-510, Sep. 2022. <https://doi.org/10.3390/automation3030025>.
- [16] D. Cauwe, C. Wang, and Q. Han, "GazeboNS3: A digital twin system for UAV swarms," in *Proc. 23rd Annual Int. Conf. Mobile Systems, Applications and Services*, 2025. <https://doi.org/10.1145/3711875.3737659>.
- [17] C. Bernardeschi, A. Fagiolini, M. Palmieri, G. Scrima, and F. Sofia, "ROS/Gazebo based simulation of co-operative UAVs," in *Modelling and Simulation for Autonomous Systems*. Cham, Switzerland: Springer International Publishing, 2019, pp. 321-334. https://doi.org/10.1007/978-3-030-14984-0_24.
- [18] J. A. Millan-Romera, J. J. Acevedo, Á. R. Castaño, H. Perez-Leon, C. Capitán, and A. Ollero, "A UTM simulator based on ROS and Gazebo," in *Proc. 2019 Workshop Research, Education and Development of Unmanned Aerial Systems (RED UAS)*, 2019. <https://doi.org/10.1109/REDUAS47371.2019.8999705>.
- [19] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*. Cham, Switzerland: Springer International Publishing, 2017, pp. 621-635. https://doi.org/10.1007/978-3-319-67361-5_40.
- [20] E. Rohmer, S. P. N. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," in *Proc. 2013 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2013. <https://doi.org/10.1109/IROS.2013.6696520>.
- [21] O. Michel, "WebotsTM: Professional mobile robot simulation," 2004. <https://doi.org/10.5772/5618>.
- [22] M. Calvo-Fullana, D. Mox, A. Pyataev, J. Fink, V. Kumar, and A. Ribeiro, "ROS-NetSim: A framework for the integration of robotic and network simulators," arXiv preprint, 2021. <https://doi.org/10.48550/arXiv.2101.10113>.
- [23] F. D'Urso, C. Santoro, and F. F. Santoro, "An integrated framework for the realistic simulation of multi-UAV applications," *Computers & Electrical Engineering*, vol. 74, pp. 196-209, Mar. 2019. <https://doi.org/10.1016/j.compeleceng.2019.01.016>.
- [24] N. Koenig, and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proc. 2004 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2004. <https://doi.org/10.1109/IROS.2004.1389727>.

Рекомендована кафедрою системного аналізу та інформаційних технологій БНТУ

Дата надходження 21.04.2026

Дата прийняття після доопрацювання 14.06.2026

Дата прийняття до друку після рецензування 11.07.2026

Дата публікації 2.10.2026

Войтех Микола Юрійович — аспірант кафедри інтелектуальних кібернетичних систем, e-mail: kolyavojteh2@gmail.com . <https://orcid.org/0009-0005-9720-6260> ;

Кравченко Ольга Петрівна — канд. техн. наук, доцент, доцент кафедри інтелектуальних кібернетичних систем, e-mail: olgakravchenko111@gmail.com . <https://orcid.org/0000-0001-7262-0899>.

Національний університет «Київський авіаційний інститут», Київ

M. Yu. Voitekh¹
O. P. Kravchenko¹

Simulation of Monitoring Multiagent Systems

¹National University «Kyiv Aviation Institute»

The article considers the aspects of simulation of monitoring multi-agent systems (MAS) that combine mobile unmanned aerial vehicles (UAVs) and stationary wireless sensor networks (WSNs). The relevance of the work is determined by the fact that UAV–WSN systems simultaneously involve the physical dynamics of mobile agents, network delays, packet losses, energy constraints of sensor nodes, and inter-agent coordination tasks, which complicates the use of a single universal simulation environment. The aim of the work is to systematize the approaches to the simulation of such systems and to form a criteria-based scheme for selecting a software environment depending on the level of model detail, target metrics, computational costs, and the possibility of transferring results to real-world conditions. The paper distinguishes low, medium, high, and multi-agent levels of modeling, for which the main simulation objects, typical tasks, and key metrics are defined. A multi-criteria scheme for evaluating the suitability of a simulation platform is formed, taking into account the accuracy of reproducing target processes, scalability, suitability for the Sim-to-Real transition, computational costs, and integration complexity. Based on a variant comparison of software environments, appropriate areas of their application for different stages of research are identified: abstract or specialized environments for high-level and multi-agent experiments, robotic 3D simulators for engineering verification of nodes, and discrete-event simulators for the analysis of network processes. For UAV–WSN systems, a hybrid modeling organization scheme is formulated, which involves the use of Python or Mesa/NumPy for testing high-level algorithms, Gazebo or Webots for engineering verification of mobile nodes, and ns-3 or OMNeT++ for detailed analysis of network processes. The practical significance of the work lies in the formation of a methodological framework for coordinating different levels of simulation and reducing the gap between abstract modeling and further engineering verification of UAV–WSN monitoring systems.

Keywords: environmental monitoring, multi-agent systems, UAVs, wireless sensor networks (WSN), hierarchical simulation.

Voitekh Mykola Yu. — Post-Graduate Student of the Chair of Intellectual Cybernetic Systems, e-mail: kolya-vojteh2@gmail.com . <https://orcid.org/0009-0005-9720-6260>;

Kravchenko Olga P. — Cand. Sc. (Eng.), Associate Professor, Associate Professor of the Chair of Intellectual Cybernetic Systems, e-mail: olgakravchenko111@gmail.com . <https://orcid.org/0000-0001-7262-0899>