

Міністерств освіти і науки України
Чорноморський національний університет імені Петра Могили

Ірина Журавська

**Гетерогенні комп'ютерні мережі
критичного застосування
на основі роїв та зграй БПЛА**

Монографія



Миколаїв – 2019

УДК 004.7:629.7-519
Ж 91

*Рекомендовано до друку вченою радою ЧНУ ім. Петра Могили
(протокол № 10 від 17 травня 2018 р.).*

Рецензенти:

Рудницький В. М., доктор технічних наук, професор, Черкаський державний технологічний університет, проректор з науково-дослідної роботи та міжнародних зв'язків;

Ситніков В. С., доктор технічних наук, професор, Одеський національний політехнічний університет, завідувач кафедри комп'ютерних систем Інституту комп'ютерних систем.

Ж 91

Журавська І. Гетерогенні комп'ютерні мережі критичного застосування на основі роїв та зграй БПЛА : монографія. – Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2019. –192 с.

ISBN 978-966-336-402-5

Монографія написана в актуальній галузі, що швидко розвивається та пов'язана зі створенням і функціонуванням гетерогенних комп'ютерних мереж, побудованих на основі зграй і суб-роїв безпілотних літальних апаратів (БПЛА). Розглянуто питання використання моніторингових і технологічних мереж критичного застосування, обумовленого виходом з ладу або втраченою управління окремими БПЛА. Також здійснено розвиток, створення нових і тестування алгоритмів, здатних зменшити обсяг обчислювальної потужності, засобів зв'язку і втручання людини, необхідних для виконання безпілотними літальними апаратами мікро-завдань, таких як запобігання зіткнень у критичних ситуаціях, у т. ч. у навколишньому середовищі, переважаному перешкодами. Основу монографії становить матеріал, підготовленої до захисту докторської дисертації. У монографії проаналізовано загальні питання підвищення ефективності функціонування та подовження часу життя різноманітних об'єктів зазначених комп'ютерних мереж, вузлів БПЛА з різними конструкціями та схемотехнічними рішеннями. Детально розглянуто особливості забезпечення стабільності зв'язку, поліпшення захищеності даних, що передаються між компонентами кіберфізичної системи (CPS) бездротовими каналами зв'язку та рівномірного навантаження компонентів обчислювальної плати на БПЛА у керованому або автономному режимах функціонування. Проведено аналіз систем та способів енергоживлення БПЛА і сформульовано перспективи вдосконалення таких систем. Книга буде корисна студентам, бакалаврам, магістрам, викладачам і професіоналам, які цікавляться сучасними проблемами використання БПЛА різних конструкцій з урахуванням їх групової поведінки.

УДК 004.7:629.7-519

ISBN 978-966-336-402-5

© Журавська І., 2019
© ЧНУ ім. Петра Могили, 2019

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	5
ПЕРЕДМОВА	7
Розділ 1 АНАЛІЗ МЕТОДІВ, МОДЕЛЕЙ ТА АПАРАТНО-ПРОГРАМНИХ СИСТЕМ ДЛЯ ЗАБЕЗПЕЧЕННЯ ФУНКЦІОНУВАННЯ РУХОМИХ МЕРЕЖ.....	11
1.1 Існуючі підходи вирішення задач керування групою безпілотних апаратів.....	11
1.2 Способи обміну інформацією між об'єктами моніторингових мереж	15
1.3 Моделі поведінки об'єктів рою	17
1.4 Аналіз існуючих апаратно-програмних систем для вирішення задачі керування БПЛА	21
1.5 Вибір апаратно-програмної платформи для розгортання програмного забезпечення у диспетчерському центрі керування РММ	25
1.6 Планування маршрутів РММ	26
1.7 Аналіз існуючих тривимірних мап для РММ	30
Висновки до розділу 1.....	32
Розділ 2 ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ КЕРУВАННЯ ГРУПОЮ БПЛА ІЗ ЗАПОБІГАННЯМ ЗІТКНЕНЬ	34
2.1 Завдання визначення цільової задачі	34
2.2 Задача планування маршруту	36
2.3 Визначення способу моніторингу цілі.....	39
2.4 Моделювання поведінки суб-роїв роботів у просторових коридорах	40
2.5 Синтез маршрутів суб-роїв БПЛА на основі нейронної мережі Хопфілда у просторових коридорах на місцевості.....	47
Висновки до розділу 2.....	57
Розділ 3 МЕТОДИ ТА АЛГОРИТМИ ВЗАЄМОДІЇ МІЖ ОБ'ЄКТАМИ РУХОМОЇ МОНІТОРИНГОВОЇ МЕРЕЖІ	59
3.1 Забезпечення якісного зв'язку в кіберфізичній системі з рухомою моніторинговою мережею	59
3.2 Система контролю за мережею безпілотних апаратів, заснована на використанні мобільних пристроїв та хмарних сервісів	73
Висновки до розділу 3.....	84
Розділ 4 СТВОРЕННЯ ТА РЕАЛІЗАЦІЯ БЕЗПЕЧНИХ МЕХАНІЗМІВ ОБМІНУ ДАНИМИ МІЖ НАЗЕМНИМ ПУНКТОМ КЕРУВАННЯ ТА БЕЗПЛОТНИМ ЛІТАЛЬНИМ АПАРАТОМ.....	87
4.1 Розробка блокового методу шифрування на базі операції XOR для рухомих об'єктів з обмеженими обчислювальними ресурсами	87
4.2 Упровадження методу стеганографічного закриття в телекомунікації БПЛА на основі Open Source протоколу MAVLink	94
Висновки до розділу 4.....	106
Розділ 5 МОДЕЛЮВАННЯ ГРУПОВОЇ ПОВЕДІНКИ ОБ'ЄКТІВ РУХОМОЇ МОНІТОРИНГОВОЇ МЕРЕЖІ	108
5.1 Методи моделювання поведінки БПЛА у зграї	108
5.2 2D-моделювання	109
5.3 3D-моделювання	113
Висновки до розділу 5.....	117

Розділ 6 МЕТОДИ ТА АЛГОРИТМИ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ВИКОРИСТАННЯ ТА ПОДОВЖЕННЯ ЧАСУ ЖИТТЯ РУХОМОЇ МОНІТОРИНГОВОЇ МЕРЕЖІ.....	118
6.1 Покращення ефективності використання безпілотних літальних апаратів шляхом розподілу завдань між ядрами процесора.....	118
6.2 Бездротові енергонезалежні інформаційно-вимірювальні мережі з передачею енергії між КФО та розподілом завдань між ядрами CPU.....	128
Висновки до розділу 6	137
Розділ 7 ПРИКЛАДНІ РІШЕННЯ З ВИКОРИСТАННЯМ РУХОМОЇ МОНІТОРИНГОВОЇ МЕРЕЖІ.....	139
7.1 Відслідковування маршрутів БПА РММ на тривимірних картографічних моделях місцевості	139
7.2 Розробка алгоритмів реєстрації великорозмірних вантажів контейнерного типу за допомогою мобільних первинних реєстраторів	147
7.3 Розробка методу обліку рухомих вантажів у пунктах пропуску із запобіганням подвійної реєстрації	157
Висновки до розділу 7	162
Додаток А Застосунок для визначення координат траєкторії руху БПЛА (на C#)	163
Додаток Б Застосунок для моделювання руху БПЛА за визначеними координатами заданої траєкторії (MATLAB)	164
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	170
Алфавітний покажчик	191

ПЕРЕЛІК СКОРОЧЕНЬ

АС	–	автоматизована система
БПА	–	безпілотний апарат
БПЛА	–	безпілотний літальний апарат
ГРKM	–	гетерогенна рухома комп'ютерна мережа
ДЦ	–	диспетчерський центр
ІВМ	–	інформаційно-вимірювальна мережа
КО	–	кібероператор
КФО	–	кіберфізичний об'єкт
КФС	–	кіберфізична система
КЦ	–	командний центр
МРЧ	–	метод рою часток
НКРЗІ	–	Національна комісія, що здійснює державне регулювання у сфері зв'язку та інформатизації України
НМХ	–	нейронна мережа Хопфілда
РЕБ	–	радіоелектронна боротьба
РМ	–	рухома мережа
РММ	–	рухома моніторингова мережа
СВР	–	стаціонарний вантажний робот
СК	–	система керування
ТД	–	точка доступу
ТР	–	транспортний робот
ЦЗ	–	цільова задача
ЦОД	–	центр обробки даних
AP	–	Access point
API	–	Application development interface
AS	–	Automated system
CO	–	Cyber operator
CPO	–	Cyber physical object
CPS	–	Cyber physical system
CPU	–	Central processing unit
CS	–	Control system
DARPA	–	Defense Advanced Research Projects Agency
DC	–	Dispatch center
DOD	–	Department of defense
EW	–	Electronic warfare
FPV	–	First person view
GCC	–	Ground control center
HMCN	–	Heterogeneous mobile computer network
HNN	–	Hopfield neural network
ICC	–	Intel C++ compiler
IMN	–	Information-measuring network
ISM band	–	Industrial, scientific, and medical radio band
MIMO	–	Multiple-input and multiple-output
MKL	–	Math kernel library
MMN	–	Mobile monitoring network
MN	–	Mobile network
MRAS	–	Mobile robot autonomous software

MS	–	Microsoft
OS	–	Operation system
OSD	–	On screen display
PMBSNU	–	Petro Mohyla Black Sea National University
PSO	–	Particle swarm optimization
RADIUS	–	Remote authentication dial-in user service
RF	–	Radio frequency
RTDB	–	Real-time database
RTH	–	Return-to-home
SDK	–	Software development kit
SSID	–	Service set identifier
TMR	–	Tactical mobile robots
TSP	–	Traveling salesman problem
TT	–	Target task
UAV	–	Unmanned aerial vehicle (or drone)
UGV	–	Unmanned ground vehicle
UI	–	User interface
UMV	–	Unmanned vehicle
UUV	–	Unmanned underwater vehicles
USV	–	Unmanned water (surface) vehicles
VLAN	–	Virtual (logical) local area network
VPS	–	Virtual private server
WDS	–	Wireless distribution system

ПЕРЕДМОВА

Для вирішення наукової, соціально-економічної та безпекової проблематики світового рівня поширюється використання кіберфізичних систем (КФС, англ. Cyber Physical System або CPS) на базі безпілотних літальних апаратів (БПЛА або дронів, англ. Unmanned Aerial Vehicle або UAV), спроможних виконувати свої функції як під керуванням диспетчерського (командного) центру (ДЦ або КЦ), кібероператорів, так і автономно, на багатотисячній відстані від власників та центрів керування.

Необхідно зазначити, що поширення галузей застосування безпілотних апаратів призвело до розвитку їх конструкційного виконання не тільки як літальних апаратів, але й наземних, надводних, підводних та комбінованих пристроїв. Тому в монографії представлені особливості розробки і впровадження методів, алгоритмів та апаратних-програмних рішень для БПЛА та відокремлено – для безпілотних апаратів (БПА, англ. Unmanned Vehicle або UMV), узагальнених за принципами конструктивних ознак, віддаленого керування або автономної роботи. Предметом дослідження монографії є гетерогенні рухомі комп'ютерні мережі (ГРKM) з акцентом на БПЛА у якості клієнтів мережі.

У розділах 1 та 7 наведена концепція розвитку гетерогенних моніторингових і технологічних мереж критичного застосування, побудованих на основі зграй і суб-роїв БПЛА, обґрунтовано місце розроблюваних методів та алгоритмів при вирішенні прикладних задач різних галузей із застосуванням груп БПЛА.

У розділах 2–6 проведено дослідження особливостей керування групою БПЛА із запобіганням зіткнень, запропоновані методи, алгоритми, які забезпечують подовжений час життя зграй та захищену передачу інформації в гетерогенній мережі.

Неможливість віддаленого вирішення проблем, які виникають у функціонуванні та технічному обслуговуванні автоматизованих систем (АС) зазначених КФС, потребує переходу від керованої до автономної поведінки БПЛА у разі впливу засобів радіоелектронної боротьби (РЕБ) на деякі компоненти АС [328]. До того ж, необхідно планувати поведінку зграї у разі неможливості повернення деяких об'єктів ГРKM при критичному застосуванні.

Крім військово-технічних та соціально-економічних питань, пов'язаних як з проведенням розвідки, здійсненням патрулювання території, знешкодженням живої та неживої сили супротивника, так і з перешкоджанням дорожньому руху, забрудненню територій, технологічних збоїв у роботі одержувачів вантажів, у разі виникнення технічних проблем у таких безпілотних АС, необхідно розглядати й науковий аспект проблеми. Для забезпечення моніторингу на відстані декілька тисяч кілометрів необхідно вирішити проблеми збільшення відповідного часу енергоживлення комп'ютерних компонентів об'єктів рухомої мережі та її спроможність використовувати для інформаційного обміну з ДЦ відкриті канали зв'язку (аеропортів, військового та цивільного транспорту тощо) на території дії такої мережі, незалежно від специфіки країни, де вона використовується.

Таким чином, на сьогодні з'явилася можливість використовувати ГРKM для вирішення проблем відновлення функціонування модулів безпілотних АС шляхом деталізованого моніторингу вузлів критичного застосування, радіоелектронного блокування таких вузлів, при певних умовах, за рахунок живлення через використання бездротових каналів, за якими здійснюється й передача даних.

Розробкою керованих та автономних інформаційно-вимірювальних систем та компонентів до них займалася велика кількість провідних вчених та науковців: R. Austin [14], J. Child [32], R. D'Andrea [45], M. Durišić, Z. Z. Tafa, G. Dimić, V. Milutinović [53], M. W. Müller [120], Y. S. Choi, J. S. Park [77], T. Prevot, P. Kopardekar, J. Rios, J. Jung [139], Yi. Tan [171], W. Zhang [198], О. Д. Азаров [283], С. Г. Антошук [212], В. С. Глухов

[78], І. Д. Горбенко [320], І. А. Жуков [233], О. В. Збруцький [285], Ю. П. Кондратенко [191], В. В. Кочан [333], Г. А. Кучук [289], В. П. Квасніков [277], Р. Н. Кветний [161], С. В. Ленков [292], В. Лужецький [294], М. В. Максимов [211], А. О. Мельник [115], В. В. Нікольський [306], В. М. Опанасенко [310], Д. П. Орнатський [278], В. М. Рудницький [329], А. О. Саченко [153], С. Г. Семенов [334], В. С. Ситніков [337], С. О. Субботін [340], О. М. Трунов [344], М. Т. Фісун [348], В. І. Хаханов [72], В. С. Харченко [94], О. К. Юдін [355], В. В. Яцків [196] та багато інших.

Проте, під час розробки КФО виникає чимало важливих науково-технічних питань, які на сьогодні поки що не вирішені: вибір архітектури обчислювального елемента на борту БПЛА та методи оптимального використання його компонентів, оптимальні налаштування системи енергоживлення БПЛА та способи передачі енергії від первинного джерела, розміщення кіберфізичних об'єктів (КФО, англ. Cyber Physical Object або CPO) в ГРKM, вибір параметрів бездротової технології передачі даних, забезпечення завадостійкості, гарантований зв'язок між КФО мережі і т. п.

Нині найбільш розповсюджені рухомі моніторингові мережі (РММ) з множиною інформаційно-вимірювальних функцій [53; 179; 275]. Цієї множини функцій достатньо лише для контролювання прогнозованих процесів. У разі, коли процес функціонування об'єктів моніторингу є швидкодинамічним, виникає складність адаптації інформаційно-вимірювальної мережі (ІВМ) до нових властивостей об'єктів, що підлягають моніторингу. Виникає проблема модифікації топології ГРKM для вирішення задач моніторингу при просторовому переміщенні [32]. Важливим є також вирішення проблеми розподілення завдань між об'єктами РММ [45; 196]. Проблемність виконання завдань може бути обумовлена обмеженістю обчислювальних потужностей чи ресурсів енергозабезпечення компонентів рухомої мережі [32; 97; 197; 239; 272]. Складність полягає у ефективному поєднанні дистанційного керування та автономної поведінки РММ для виконання завдань у динамічному просторі [115; 125; 232].

Відомі методи створення та застосування рухомих мереж мають суттєві недоліки: батарейне живлення їх рухомих компонентів вимагає жорсткого контролю за часом та дальністю функціонування мережі, обмеження трафіку передаваної інформації для економії енерговитрат, жорстку траєкторію моніторингу через відсутність інтелектуальних властивостей компонентів ІВМ [114; 196; 347; 348; 349]. Разом з тим, постійно зростаюча кількість різних об'єктів моніторингу навколишнього середовища, технологічних процесів із застосуванням рухомих мереж, вимагає збільшення довготривалості функціонування та зміни поведінки їх рухомих компонентів адекватно змінам в оточуючому середовищі [213; 219; 225]. Такий підхід потребує створення нових більш ефективних рішень щодо інтелектуальної або керованої взаємодії компонентів рухомих мереж.

Результати аналізу досліджень вітчизняних та закордонних вчених засвідчили, що створення рухомих компонентів ГРKM необхідно будувати на якісно новому принципі взаємодії КФО з можливістю доповнення режимів роботи віддаленим керуванням через відкриті канали зв'язку [77], із забезпеченням завадостійкості [71; 152; 154; 231; 321] та захисту інформації [236; 264; 283; 291; 293; 320].

У основу монографії покладено ідею створення та використання рухомих ГРKM критичного застосування з подовженим часом енергонезалежності на базі автономної та дистанційно керованої взаємодії КФО, які поєднані у суб-рої для виконання мікро-завдань, з подальшим розгляданням таких суб-роїв як частин зграї БПЛА, що виконує єдину макро-задачу. Тобто, рій та суб-рій (swarm & sub-swarm) – це менша спільнота ніж зграя (flock), та виконує часткові завдання, які можуть відрізнитися від завдань інших суб-роїв. Розмір таких завдань визначається обмеженими обчислювальними ресурсами на борту БПЛА. Зграя – більш головне поняття. Наприклад, для РММ на рівні зграї узагальнюється та наноситься на єдину мапу вся інформація, здобута суб-роями. Тому

результати процесу вимірювання (нп., заміри температури, показників забрудненості та/або випромінювання у зоні функційного модуля, фотографії об'єктів на обстежених невеликих ділянках території, тощо) у шифрованому та/або стеганографічному вигляді передаються у складі загальної інформації до головного БПЛА та/або ДЦ компанії-власника ГРКМ засобами досяжних відкритих каналів зв'язку.

У монографії проаналізовано загальні питання підвищення ефективності функціонування та подовження часу життя різноманітних об'єктів ІВМ, вузлів БПЛА з різними конструкціями та схемотехнічними рішеннями. Виконаний розвиток існуючих та створені нові алгоритми, здатні зменшити обсяг обчислювальної потужності, засобів зв'язку і втручання людини, необхідних для виконання безпілотними літальними апаратами мікро-завдань, таких як запобігання зіткнень у критичних ситуаціях, в т. ч. у навколишньому середовищі, переважаному перешкодами. Детально розглянуті особливості забезпечення стабільності зв'язку, поліпшення захищеності даних, що передаються між компонентами КФС бездротовими каналами зв'язку, питання рівномірного навантаження компонентів на БПЛА у керованому або автономному режимах функціонування.

Наведені приклади вирішення задач у деяких галузях із застосуванням ГРКМ з детальним аналізом алгоритмів роботи, процесів та апаратно-програмної реалізації запропонованих рішень.

Авторка щиро дякує Ректорові Чорноморського національного університету (ЧНУ) ім. Петра Могили професору Леоніду Клименку та проректору з науково-дослідної роботи професору Володимиру Беглиці за їхню всебічну підтримку.

Дослідження, описані в монографії, виконані за підтримки Міністерства освіти і науки України в рамках держбюджетних НДР, що проводились у ЧНУ ім. Петра Могили за темами «Створення поліметричних датчиків інформаційно-вимірювальних систем з живленням елементів від енергії вимірювального сигналу» (держ. реєстр № 0115U000316, 2015–2016 рр.) та «Розроблення бездротових енергонезалежних інформаційно-вимірювальних мереж критичного застосування військово-цивільного призначення» (держ. реєстр № 0117U000447, 2017–2018 рр.).

Особлива подяка науковому керівнику вищезазначених НДР, доктору технічних наук, професору Мусієнку М. П., який всі ці роки також був науковим консультантом всіх розробок, висвітлених у монографії. Щиро й терпляче, майже цілодобово Максим Павлович, як кажуть, «тримав руку на пульсі» не тільки наукових трендів зазначеної галузі, а й докторської дисертації Авторки, в якій планується узагальнити напрацювання з теоретичних основ, методів та засобів створення та функціонування швидкодинамічних гетерогенних комп'ютерних мереж критичного застосування.

Дякую моїм колегам, які допомагали в обговоренні ідей та надали багато корисних порад щодо їх втілення: Тетяні Андрущенко, Анжелі Бойко, Світлані Боровльовій, Івану Бурлаченку, Євгену Давиденку, Ользі Дворник, Олексію Денисову, Ігорю Кандибі, Наталії Коваль, Олександрі Корецькій, Ярославу Крайнику, Інессі Кулаковській, Сергію Пузирьову, Ніні Романовій, Наталії Руденко, Володимирі Савінову, Євгену Сіденку, Ларисі Солобуто.

Велика подяка також бакалаврам і магістрам ЧНУ ім. Петра Могили за ретельне опрацювання програмного коду, тестування застосунків та проведення натурних експериментів з БПЛА та іншими рухомими пристроями: Вадиму Ахундову, Максиму Барчинському, Владиславу Борисенку, Михайлу Васильєву, Андрію Гапчуку, Володимирі Єлезарову, Дмитру Жаруку, Олексію Жерегі, Станіславу Димедюку, Катерині Івановій, Михайлу Костирі, Світлані Лавриненко, Роману Лейзеровичу, Костянтину Морозову, Назару Мошулю, Катерині Обуховій, Валерії Олійник, Роману Ончі, Дмитру Петренку, Денису Поліщуку, Дмитру Румянкову, Борису Салтану, Дмитру Собку, Надії Ухань,

Дмитру Шулякову, Валентину Шебетюку та Дмитру Юрину. Окреме спасибі Ігорю Полонському за створення обкладинки цієї книги.

Я глибоко вдячна офіційним рецензентам монографії – доктору технічних наук, професору Володимирі Рудницькому та доктору технічних наук, професору Володимирі Ситнікову, – а також багатьом анонімним (так званим, «сірим» рецензентам!), які переглядали книгу та раніш надруковані матеріали, що ввійшли до неї, і висловили цінні зауваження й поради, які значно покращили самі дослідження й публікації.

Дякую за терпіння моїм рідним і друзям, за те, що з розумінням поставилися до моєї відсутності, поки я була зайнята польотами зграй БПЛА...

Сподіваюсь, що книга буде корисна студентам, бакалаврам, магістрам, спеціалістам і викладачам, які цікавляться сучасними проблемами використання БПА різних конструкцій з урахуванням їх групової поведінки.

Ірина Журавська

Розділ 1

АНАЛІЗ МЕТОДІВ, МОДЕЛЕЙ ТА АПАРАТНО-ПРОГРАМНИХ СИСТЕМ ДЛЯ ЗАБЕЗПЕЧЕННЯ ФУНКЦІОНУВАННЯ РУХОМИХ МЕРЕЖ

1.1 Існуючі підходи вирішення задач керування групою безпілотних апаратів

Виходячи з того, в якому середовищі, детермінованому чи недетермінованому, працює рухома мережа (РМ) з БПА, визначаються два підходи до управління рухомих модулем: управління в умовах детермінованого та недетермінованого середовища.

У випадку детермінованого середовища система керування може повністю вирахувати траєкторію руху за уже відомими алгоритмами. Такий підхід базується на тому, що система керування має всю необхідну інформацію про перешкоди у своїй зоні відповідальності, тобто має електронну мапу місцевості, і в змозі побудувати безпечний та оптимальний маршрут слідування. Таке рішення є більш ефективним за рахунок того, що маршрути можуть бути прораховані заздалегідь та не вимагатимуть від РМ постійного корегування маршруту. Від дрону вимагається лише «сліпе» слідування командам центру керування, що в свою чергу зменшує вимоги до апаратного забезпечення самої РМ [341]. Водночас, додатковими умовами для функціонування такої системи є заздалегідь побудована електронна мапа місцевості.

Прикладом досліджень в області групового керування роботами може служити проект «MARTHA», який виконувався в лабораторії Аналізу системних архітектур Франції [7]. Метою цього проекту була розробка методів організації групової взаємодії роботів (від 10 до 100 шт.), призначених для транспортування вантажів в складських терміналах. В проекті «MARTHA» використовується централізоване управління групою роботів, при якому планування дій кожного робота групи здійснюється одним центральним пристроєм управління. Концепція проекту «MARTHA» полягає в наступному. Система управління групою роботів складається з двох основних частин: центру управління і бортових обчислювачів з комунікаційним модулем. Центр управління встановлюється стаціонарно в командному центрі і вирішує завдання планування дій всієї групи роботів. Крім того, центр управління забезпечує зв'язок людини-оператора з кожним роботом групи у разі непередбачених ситуацій, а також використовується для постановки цільової задачі (ЦЗ). На борту кожного робота встановлюється обчислювальний пристрій, який призначений для вирішення завдання руху до зазначеної мети. Контролер зв'язку служить для забезпечення зв'язку робота з центральною станцією та з іншими роботами групи.

Іншим прикладом використання централізованого управління групою роботів або, що частіше зустрічається в літературі, мультиробототехнічної системи, є проект, який виконувався в Центрі розподілених робототехнічних систем Університету Міннесоти (США) за підтримки управління DARPA (Defense Advanced Research Projects Agency) Міністерства оборони США. Проект був пов'язаний з розробкою програмного забезпечення і апаратних засобів системи управління групою мініатюрних роботів-розвідників («Scout»), призначених для вирішення завдань охорони, розвідки і спостереження за обстановкою в різних приміщеннях. У цьому

проекті акцент робиться на розробку програмного забезпечення для центрального пристрою керування, за допомогою якого оператор може керувати роботами групи в режимі спостереження.

Військове відомство Російської Федерації (РФ) в квітні 2018 р. прийняло рішення сформувати в складі військово-морського флоту нові полки БПЛА з БПЛА «Форпост» [284]. З 2011 р. кількість БПЛА в Збройних силах РФ збільшилася більш ніж у 10 разів. Апарати проводять розвідку, здійснюють наведення на цілі, в режимі онлайн коректують артилерійські авіаудари, допомагають оцінити втрати, нанесені супротивникові.

Апарат здійснює дистанційно керований з землі політ в автономному режимі або згідно з попередньо заданою програмою, а також здійснює навігацію за підтримки наземної станції управління і диференціальної системи глобального позиціонування. Виконує передачу даних про параметри польоту і стан корисного навантаження на наземну станцію управління. БПЛА РФ підтримують постійний зв'язок з наземною станцією управління по дубльованих каналах в дуплексному режимі.

Радіоелектронні системи БПЛА малого класу «Корсар» (РФ) здійснюють передачу на дальні відстані (декілька десятків км) великих масивів інформації [327]. Якщо зазначені вище БПЛА «Форпост» передавали дані лише наземному центру обробки даних (ЦОД, англ. Data Center), то БПЛА «Корсар» використовують для цього інші аналогічні апарати, або гелікоптери, літаки та ін. об'єкти гетерогенних систем. Відпрацьовується також поведінка зазначених БПЛА у критичних ситуаціях, для чого «Корсари» виробляються з композитних матеріалів (що робить їх невидимими для радарів) та укомплектовані парашутною системою.

Якщо «Корсар» – це БПЛА літакового типу, то наступною розробкою РФ є БПЛА вертолітного типу – «Катран» з меншими габаритними розмірами та підвісом електронно-оптичної системи (ЕОС) знизу [227; 314].

Тому переважна увага в роботі приділена проблемам, які мають місце при створенні і функціонуванні саме рухомих моніторингових мереж [159].

Метою проекту «AMADEUS», запропонованого японськими дослідниками, є створення групи роботів, які забезпечують підвезення і вивезення виробів для конвеєрних ліній [150]. Використовуються два типи роботів: транспортний робот (ТР), що виконує безпосереднє перевезення виробів, і стаціонарний вантажний робот (СВР), розташований в безпосередній близькості від конвеєрної лінії, який виконує розвантажувально-навантажувальні роботи з ТР на конвеєр і навпаки. Є безліч ТР і СВР. Транспортні роботи переміщуються від одного СВР до іншого заданим маршрутом. Рух здійснюється за однією направляючою лінією в обох напрямках, однак, у разі виникнення перешкоди (у вигляді людини, іншого ТР або іншого об'єкта), ТР повинен зійти з лінії і обійти перешкоду. Коли виникає необхідність підвезення або вивезення виробів з конвеєрної лінії, відповідний СВР видає запит по каналах зв'язку на всі ТР. Вільні ТР надсилають у відповідь свій ідентифікаційний номер і поточні координати. Вільний ТР визначається найближчим СВР, який і передає цьому ТР своє завдання. СВР діють незалежно один від одного у виборі того чи іншого ТР.

Основною відмінною особливістю проекту «AMADEUS» являється те, що він побудований на основі децентралізованого управління, при якому планування дій роботів групи здійснюється кількома пристроями. Дійсно, на відміну від проекту «MARTHA», в проекті «AMADEUS» дії ТР планують декілька СВР, кожен з яких обирає з безлічі ТР (за певними ознаками) потрібний йому і формує для нього необхідне завдання на виконання деякої дії.

Описані вище групи роботів призначені для використання в стаціонарних, заздалегідь підготовлених середовищах, що дозволяє заздалегідь знайти різні

оптимальні варіанти дій роботів групи, а в процесі функціонування відтворювати відповідні дії в залежності від ситуації, яка склалася. Проблема групового керування роботами істотно ускладнюється у разі функціонування роботів у заздалегідь невідомому і динамічно змінному середовищі, в якому можуть діяти деякі зовнішні сили, які не залежать від роботів групи.

Максимальна централізовано керована зграя («легіон», «бригада», «група») з 1180 дронів була запущена в грудні 2017 р. у Китаї [30]. Зграя, розподілена на 6 суб-роїв, керується лише однією консоллю та одним оператором. Деталізовані команди щодо функцій кожного дрону (позиціонування, колір випромінюваного світла тощо) зашифровані у відповідні коди. Водночас, кожен з дронів здатний одночасно регулювати власне положення та маршрут. Якщо ж він втрачає синхронізацію з іншими БПЛА рою і не може піднятися до запрограмованої висоти, дрон повернеться на землю, не перешкоджаючи іншим.

Рішення задачі у недетермінованому середовищі ускладнюється тим, що модуля керування відомо тільки про початкові і кінцеві координати, які РММ повинна «відвідати». Всі задачі з розрахунку траєкторії маршруту перекладаються на самих дронів, тим самим ускладнюючи апаратну та програмну частину роботи. Через те, що для проходження певного маршруту дрон має сам визначати, чи може він продовжити рух в тому чи іншому напрямку, він повинен сам побудувати локальну мапу перешкод середовища, в якому знаходиться, а вже потім рухатись до цілі. Хоча такий алгоритм підвищує автономність кожного елемента групи, в той же час він є не вигідним для використання під час управління моніторинговими мережами, оскільки призведе до здорожчання кожного окремого елемента мережі, вірогідність втрати якого дуже висока в умовах підвищеного ризику.

Проблемі групового керування роботами, функціонуючих в реальних, заздалегідь невідомих середовищах, присвячено ряд досліджень, проведених в США і Японії.

У майбутньому Intel планує збільшити кількість безпілотних літаків у групі до 1500 штук [83].

В Японії дослідження в області систем групової взаємодії роботів активно ведуться в Нагойському (Nagoya) університеті. Тут розроблена система DARS (Distributed Autonomous Robotic System), за допомогою якої відпрацьовуються алгоритми та методи планування й управління скоординованими діями групи роботів, що функціонують у природному неорганізованому середовищі, з єдиного мобільного командного центру [49]. Ідея підходу полягає в тому, що кожен робот може виконувати, наприклад, одне з двох завдань, і в процесі функціонування може міняти його на інше. Спочатку роботи обирають собі завдання довільно. Періодично, у визначені моменти часу, обмінюючись інформацією з найближчими сусідами, роботи приймають рішення щодо продовження виконання обраного ними завдання або про його зміну. Якщо більшість найближчих сусідів виконують таке ж завдання, що й даний робот, він продовжує його виконання, якщо інше завдання, то робот приймає рішення про зміну свого завдання.

Управлінням DARPA Міністерства оборони США також фінансувалась розробка тактичних мобільних мікророботів для групового застосування в міських умовах, яка виконувалася в рамках програми «Тактичні мобільні робототехнічні системи» [147]. Дослідження були спрямовані на відпрацювання командної взаємодії групи, що складається з людей і роботів. Система управління дозволяє оператору керувати кількома роботами з єдиного командного центру. Оператор задає ціль для кожного робота, а ті вже переміщуються до цілей автономно. Дослідження проводилися кількома науковими центрами в рамках таких програм управління DARPA, як Mobile Autonomous Robot Software (MARS) і Tactical Mobile Robots (TMR).

У даному випадку верхній, найбільш інтелектуальний рівень управління, на якому здійснюється декомпозиція загальної ЦЗ на окремі суб-завдання, реалізується оператором, а дрони тільки вирішують їх, адаптуючись до умов середовища.

В Університеті Карнегі-Меллона розроблена система GRAMMPS (Generalized Robotic Autonomous Mobile Mission Planning System), призначена для рішення задач планування дій групи мобільних роботів у неструктурованих середовищах [122]. Розробка фінансувалася управлінням DARPA.

Іншим способом групового управління, що застосовується для робототехнічних систем, є управління з використанням лідируючого (leader-, master-, mother- й т. п.) пристрою. При використанні такого способу групового управління один з роботів є «ведучим» і управляється автономно або дистанційно оператором, а решта роботів – slaves. Вони керуються в режимі стеження за «ведучим» або командами, які надходять від ведучого робота. Спосіб групового управління «з ведучим» дозволяє оператору, дистанційно керуючи одним роботом (провідним), фактично керувати цілою групою роботів. У США в даному напрямку працює ряд університетів.

Так, в Пенсільванському університеті в рамках проекту MARS (Multiple Autonomous Robots) були розроблені методи керування рухомою групою мобільних роботів на основі локальної сенсорної інформації в режимі «ведучий-ведений». Система включає один ведучий робот і кілька ведених. Методи відпрацьовувалися з використанням двоколісних робото-технічних мобільних платформ Segway RMP і чотириколісних робототехнічних платформ Clodbustera також Clodbuster-II і Clodbuster-III [14]. Зокрема, при виконанні проекту відпрацьовувалися методи координації груп мобільних автономних роботів-агентів Clodbuster з використанням правил найближчого сусідства. У основу цих правил покладена теорія пов'язаних систем. Сутність методу полягає в тому, що роботи визначають свої дії з дій, виконуваних сусідами. Наприклад, якщо в групі є ведучий, який рухається запланованою траєкторією, то і всі інші роботи групи будуть виконувати ті ж рухи, що й ведучий.

Також, в Лабораторії загальної робототехніки, автоматички, сприйняття та сприйняття (GRASP) в Пенсільванському університеті з 2016 р. йде робота щодо розгортання та розвитку безпроводної системи зв'язку та складних алгоритмів пілотування, які визначають конкретні кроки для досягнення такої мети, наприклад, як «Перехід від пункту А до пункту В». Люди не маніпулюють міні-БПЛА в процесі прийняття рішень на низькому рівні. Завдяки просторовій роздільності з точністю до декількох міліметрів, система GRASP успішно добилася неймовірно точного позиціонування групи до 20 квадрантів одночасно у польоті [25].

У Центрі розподілених робототехнічних систем Університету Мінесоти проводилися дослідження з управління групою роботів, призначеної для вирішення завдань охорони і спостереження. Група є неоднорідною і складається з великого мобільного робота, званого «Ranger», і декількох мініатюрних роботів, забезпечених різними сенсорними пристроями так званих «Scout» (розвідник). Робот «Ranger» призначений для транспортування, розгортання і спостереження за роботами-розвідниками [22]. При цьому, якщо робот «Ranger» може функціонувати автономно, то роботи-розвідники не є автономними, а керуються командами, які надходять від робота «Ranger», бортова система управління якого на підставі інформації про поточну ситуацію визначає ту чи іншу стратегію поведінки роботів-розвідників.

Наступний напрямок досліджень складають проекти, направлені на розробку повністю розподілених мобільних мультиробототехнічних систем, керованих децентралізовано.

Під час моделювання децентралізованої поведінки членів суб-рою врахований досвід автономної адаптивної системи на основі рою з 103 безпілотників Perdix (Пентагон, 2016), запущених з трьох «материнських літаків» Boeing F/A-18E/F. Мікро-

дрони Perdix не є попередньо запрограмованими, вони представляють собою колективний організм, розділяючи один розподілений мозок для прийняття рішень і адаптації один до одного [136].

Так діють рої комах у природі. З огляду на розмір зграї (флоту), безпілотниками неможливо управляти з єдиного центру кожним окремо. Оскільки кожен Perdix спілкується і співпрацює з усіма іншими апаратами, рій не має лідера і може плавно адаптуватися до будь-яких змін в команді. Але рій отримує централізовані команди загального призначення від людини-оператора, після чого виконує їх найбільш оптимальним способом.

1.2 Способи обміну інформацією між об'єктами моніторингових мереж

Узагальнена модель групи дронів показана на рис. 2.1, за якою роботи спілкуються один з одним. У деяких випадках, глобальне позиціонування або центральні команди посилаються з командного центру, але зграя має виконати поставлене завдання навіть при блокуванні комунікації з командним центром.

У більшості робіт дослідницького характеру групи БПЛА класифікуються як «рої дронів», враховуючи їх колективну поведінку. Але, з урахуванням сучасних тенденцій розвитку галузей, в яких застосовуються БПЛА, доцільно розділяти зграю на суб-рої, зважаючи, що різні БПЛА виконують різні завдання. Об'єкти, що мають виконувати однакові (споріднені) завдання, – наприклад, вимірювання метеоданих, – є сенс узагальнювати за ієрархією у «суб-рій», всі суб-рої за макро-задачею поєднуються у «зграю».

Тому, щоб не змінювати в аналізі авторську нотацію, розуміємо під авторським терміном «рій» в огляді літератури збільшену, нерозподілену на «суб-рої» групу БПЛА (дронів), яка у наступних (дослідницьких) розділах вже буде іменуватися «зграєю».

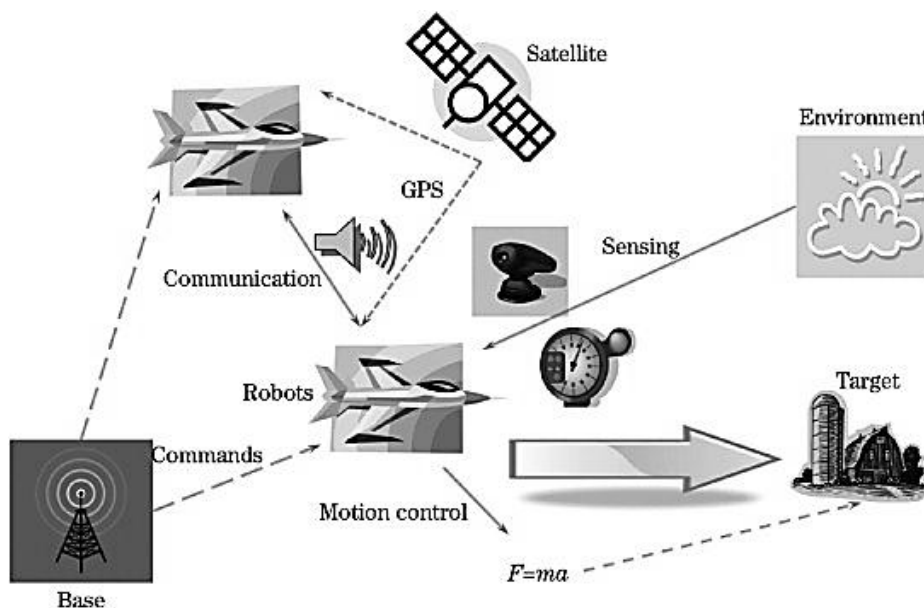


Рис. 1.1. Загальна модель групи дронів [171]

Обмін інформацією необхідний, коли БПЛА співпрацюватиме один з одним, тому модуль обміну сигналами є основним компонентом для управління поведінкою

рою. Основні функції об'єктів, що беруть участь у такій взаємодії, обмежені зв'язком типу «Peer-to-Peer» (однорангова мережа).

Взаємодія об'єктів зграї у гетерогенній мережі є найбільш поширеним видом спілкування, та їх поведінка може бути описана подібно біологічним об'єктам [22]. Такий механізм з позитивним зворотнім зв'язком допоможе оптимізувати поведінку дронів.

Є три способи обміну інформацією в зграї: прямий зв'язок, зв'язок через середовище і сенсорний [18]. Більш ніж один тип взаємодії може використовуватись в одному рої, наприклад, кожен дрон сприймає навколишнє середовище та зв'язується з сусідами. Valch розглядав вплив трьох видів комунікацій в рою [18]. Він поставив три завдання і порівняв продуктивність під час моделювання. Деякі дослідники також обговорювали можливість рою взаємодіяти без комунікації. Однак, зв'язок і сенсорнування може дійсно підвищити ефективність рою у більшості випадків.

Прямий зв'язок аналогічний зв'язку об'єктів (вузлів) бездротової мережі і також складається з двох типів: однорангова мережа і трансляція. Завдяки розвитку мобільних пристроїв можна відразу прийняти кілька існуючих технологій з використанням IEEE802.11b, Bluetooth та ін. бездротових стандартів зв'язку. Проте, бездротові датчики коштують майже половину загальної ціни робота. Інший недолік такої схеми є те, що необхідна пропускна можливість каналу зв'язку буде експоненціально зростати в міру зростання популяції рою. Таким чином, прямий зв'язок у рої або зграї має бути обмежений, наприклад, національним регулятором (FAA США, Державною авіаційною адміністрацією України тощо) або компанією, яка є власником дронів та інтегрує їх у національну повітряну систему (рис. 1.2).

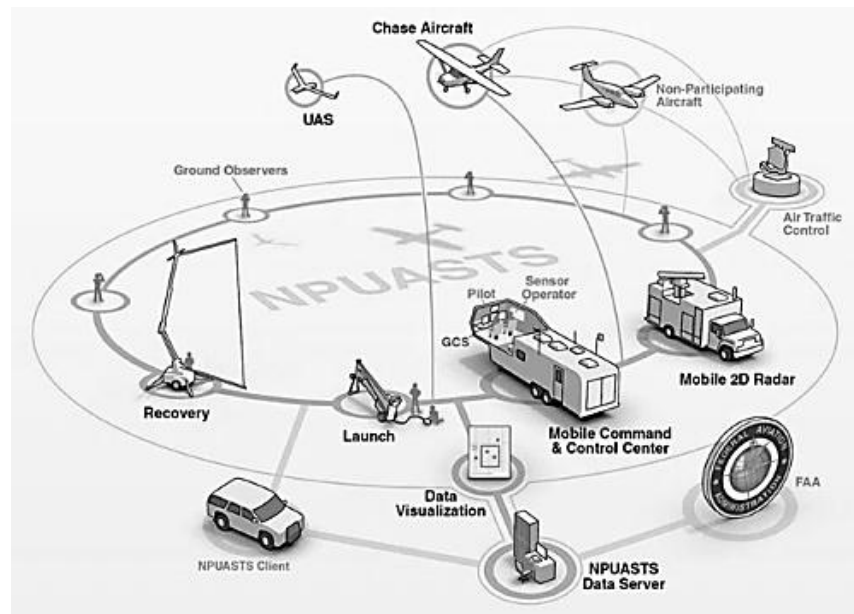


Рис. 1.2. Спільна інфраструктура групи безпілотних апаратів, що впроваджується Північно-Дакотським випробувальним центром безпілотних систем (NP UAS TS), який співпрацює з NASA [129]

Техніка зв'язку, навігації та спостереження має вирішальне значення для забезпечення того, щоб дрони опинилися під постійним контролем оператора зграї [128].

Хоча кілька існуючих бездротових технологій доступні, протоколи і топології, які спеціалізуються на груповій робототехніці залишаються нерозкритими. Існуючі комп'ютерні мережі призначені для обробки даних і обміну інформацією між вузлами [207]. Комунікація у групі дронів має повною мірою використовувати сенсування та сигнали щоб підвищити спільну поведінку дронів і динамічну топологію рою [86; 219].

Навколишнє середовище може виступати в якості посередника для взаємодії дронів. Дрони залишають свої сліди в навколишньому середовищі після певної дії щоб інформувати інших роботів, які можуть зчитати слід без безпосередньої комунікації між іншими об'єктами рою. Таким чином, подальші дії, як правило, підсилюються і спираються один на одного, що призводить до спонтанної появи діяльності на рівні рою. Рій імітує поведінку мурах або бджіл і взаємодіє за допомогою «віртуальних феромонів». Така інтерактивна схема уникає експоненціального зростання кількості зв'язків між роботами, але має деякі обмеження на навколишнє середовище, щоб підтримати «віртуальні феромони» [135].

Дрони можуть відчути інших роботів і середовище поблизу з використанням бортових датчиків, які можуть розрізняти роботів та інші об'єкти навколишнього середовища. Роботи зчитують об'єкти або цілі в навколишньому середовищі і виконують певні дії, такі як ухилення від перешкод, цільовий пошук й т. ін. [97]. Основна проблема цієї схеми в тому, як інтегрувати всі датчики в рій ефективно для співпраці. Cortes та ін. досліджували, як управляти і координувати групу автономних транспортних засобів, що розглядаються в якості агентів з датчиками, в адаптивний, розподілений і асинхронний спосіб [39].

Основна відмінність між комунікацією і зондуванням є те, чи індивіди відправляють повідомлення активно, чи приймають повідомлення пасивно. Хоча більш точне спілкування вимагає більш складних апаратних засобів і синхронізації, пропускна здатність, енергія та час виростає дуже швидко разом з популяцією рою. Кооперативна модель роботизованої групи має спробувати спростити спосіб комунікації і використовувати стільки зондування, наскільки це можливо. Кольори, яскравість і взаємне розташування можуть бути використані для зондування і забезпечення інформації без зв'язку. В деяких завданнях рій може обмінюватися всією інформацією тільки за допомогою датчиків.

1.3 Моделі поведінки об'єктів рою

Сучасні методи створення груп БПЛА вимагають використання комплексних (узагальнених) критеріїв, впроваджених математичних моделей у процес оптимізації параметрів компонентів БПЛА. Такі моделі мають імітувати функціонування БПЛА (суб-рою, зграї) при виконанні свого завдання.

Основу поведінки членів рою включає в себе функції, забезпечені модулем управління (автопілот). Тоді об'єкти рою можуть менше покладатися на прямий зв'язок один з одним. Рій може поліпшити продуктивність з меншим обміном інформацією і високою масштабованістю.

Детальному аналізу підлягають питання безпеки та технічні чинники, що можуть перешкодити дрону-доставщику виконати завдання:

- температура більш ніж 40°C влітку та менше ніж 20°C взимку (при якій непрацездатне комп'ютерне, механічне, та вимірвальне обладнання);
- висока швидкість вітру (та іноді піщані бурі), здатні змінити курс безпілотної;
- високий ступінь електромагнітного випромінювання, при якому БПЛА стає некерованим, тощо.

Крім вирішення питань керування такими поодинокими дронами (флотом), необхідно приділити увагу таким питанням, як використання системою безпеки для

захисту дронів та їх вантажу різних біометричних даних адресатів (відбитків пальців, розпізнавання очей та ін.). Потрібно також прийняти рішення (створити алгоритм поведінки дрону), які дії повинен вчинити БПЛА з вмістом носимого контейнера (модуля пам'яті) у разі захоплення дрону сторонніми особами: запустити механізм знищення контейнера, здійснити ураження загарбника або ін.

Роботи в комплексній системі рою можуть мати додаткові функції, включаючи, задачі декомпозиції, розподіл завдань, адаптивне навчання і т. д. [171]. Роботи за допомогою цих функцій в апаратній частині можуть спростити структуру алгоритму руху для створення більш складної фізичної конструкції дронів. Роботи можуть також виконувати аналогічні функції з ретельно сконструйованими кооперативними алгоритмами. Реалізації таких функцій в апаратному або програмному забезпеченні, залежить від фізичної конструкції роботів, контролерів та датчиків, таким чином, щоб краще використовувати їх комп'ютерні компоненти [105].

Розподіл завдань і навчання в цьому випадку є досить важливим для рою роботів. Задача розкладання і призначення може значно підвищити ефективність особливо складних завдань. Kalra та Martinoli порівняли витрати і користь від різних типів завдань на розподілення [87]. Навчання також корисно, так як параметри механізму управління важко оновлювати. За допомогою самоадаптаційного навчання і методів оптимізації, рій показує кращу пристосованість в різних середовищах. Лі та ін. обговорили проблему використання різних методів навчання в рої та порівняли їх продуктивність на симуляторі [106]. Zhang та ін. застосували еволюційну нейронну мережу для розвитку управління роєм і використовували цей метод у проблемі огляду структури рою [199].

Моделювання використовується в багатьох областях дослідження, щоб краще зрозуміти внутрішню структуру системи. Моделювання допомагає рою, оскільки рій повинен бути масштабованим до декількох сотень та тисяч роботів. Час і гроші обмежені для таких масштабних експериментів, тому експерименти можуть виконуватись лише з поодинокими дронами.

З огляду на характеристики групи дронів, методи моделювання діляться на чотири типи [171; 340]:

- на основі датчиків;
- мікроскопічне моделювання;
- макроскопічне моделювання;
- моделювання на основі інтелекту.

У методі моделювання на основі датчиків, датчики і пристрої роботів змодельовані як основні компоненти системи разом з об'єктами в середовищі. Тоді взаємодії між роботами моделюються як реалістичні і прості, наскільки це можливо. Цей метод моделювання є найпопулярнішим та найстарішим, який використовується для роботизованого експерименту.

Попередні дослідження з використанням методів моделювання на основі датчиків не враховували реальні фізичні обмеження [81], але тепер дослідники ввели реальні фізичні закони в моделі [180].

У мікроскопічному моделюванні, роботи і взаємодії моделюються як кінцеві автомати. Поведінка кожного робота визначається як кілька станів, і умови передачі засновані на вхідних даних від системи телекомунікацій і сенсорів. Оскільки модель заснована на поведінці кожного робота, моделювання має бути запущено кілька разів, щоб отримати середній результат поведінки рою.

У більшості досліджень ймовірна мікроскопічна модель використовується з тих пір, як шум був змодельований як ймовірність в моделі. У ймовірній мікроскопічній моделі ймовірності оцінюються в залежності від експериментів з реальними роботами, і модель повторює цю ймовірність в симуляції, щоб передбачити поведінку рою [17].

Макроскопічне моделювання є метод моделювання, протилежний мікроскопічному. У макроскопічному моделюванні, поведінка системи визначається як різниця рівняння, і стан системи являє собою середню кількість роботів у цьому стані в певний проміжок часу.

Основна відмінність між мікроскопічним і макроскопічним моделюванням є рівень деталізації моделей. Мікроскопічна модель використовується для симуляції на індивідуальному рівні та на рівні групи, тоді як макроскопічна імітує поведінку на рівні рою. Мікроскопічна модель перебирає поведінку рою, а макроскопічна може видавати одразу кінцевий стан рою. Таким чином, макроскопічна модель матиме глобальний погляд на рій, тоді як мікроскопічна модель тільки показуватиме деталі поведінки рою [282].

Макроскопічні ймовірнісні моделі також широко використовуються дослідниками. Martinoli та ін. застосували макроскопічне моделювання, щоб вирішити проблему розширення з базової моделі, яка містить тільки два стани, до моделі з усіма станами [42]. Вони також порівняли мікроскопічну, макроскопічну і модель на основі датчиків та описали недоліки макроскопічної моделі.

Спільні схеми з алгоритмів інтелектуального рою були введені в рій робототехніки в багатьох дослідженнях. Оскільки роботи використовують ті ж або аналогічні схеми з цими алгоритмами, моделі та інші методи можуть бути використані також для роїв та зграй БПА.

Найбільш часто використовуваний алгоритм на основі інтелекту є метод рою часток (МРЧ), який імітує процес скупчення птахів. Метод чисельної оптимізації, для використання якого не потрібно знати точного градієнта оптимізованої функції. МРЧ був доведений Кеннеді, Еберхартом та Ші і спочатку призначався для імітації соціальної поведінки [90; 91]. Алгоритм був спрощений, і було зауважено, що він придатний для виконання оптимізації. Книга Кеннеді й Еберхарта описує багато філософських аспектів МРЧ і так званого роювого інтелекту [91].

Крім МРЧ, дослідники також ввели інший роювий алгоритми пошуку. Багато успішних моделей рою були створені з огляду на поведінку колоній мурашок. Ці підходи забезпечують ефективні евристичні алгоритми для пошуку у динамічному середовищі [158; 188] і маршрутизації [180; 183].

Деякі дослідники вважають, що рух рухомої групи роботів з достатнім ступенем точності описується математичною схемою руху, так званою «прогулянкою Леві» [169]. Аналогічно можна описати рух БПЛА всередині суб-рою. Це обумовлено тим, що система управління рухом БПЛА при відхиленнях від маршруту, обумовлених зовнішніми причинами (вітер, перепони на шляху і т. п.), повертає БПЛА на маршрут слідування, але між сусідніми «короткими» точками маршруту утворюються довгі траєкторії, які можна віднести до «прогулянок Леві».

Термін «політ Леві» («прогулянка Леві») був введений Бенуатом Мандельбротом. Пізніше дослідники розширили термін «політ Леві» для включення випадків, коли випадковий шлях відбувається на дискретній сітці, а не на безперервному просторі. Названий на честь французького математика Поля Леві, вид руху «прогулянка Леві» характеризується багатьма невеликими рухами в поєднанні з декількома довгими траєкторіями (рис. 1.3).

«Політ Леві» є одним з видів випадкового блукання, в якому кожний наступний хід вибирається випадковим чином і без впливу будь-якого попереднього ходу. У прогулянці Леві більшість кроків знаходяться в невеликій області, але іноді проводяться довгі маршрути.

Можна припустити, що моделювання руху по Леві доцільно чередувати з іншим типом руху – броунівським рухом – який, як правило, утримує фрактали в більш щільному розмірі простору без істотного подовження траєкторій.

Наша мета полягала не в тому, щоб перевірити, яка з моделей найкраще підходить для всіх можливих розподілів, а скоріше для перевірки того, чи присутній в поведінці БПЛА всередині рою «політ Леві» або поведінку броунівського руху.

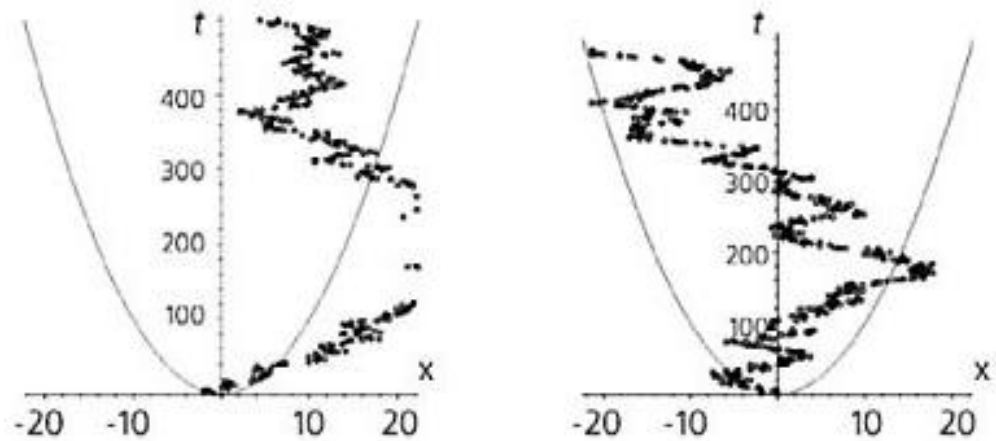


Рис. 1.3. Фрактали з «польотами Леві» [350]

«Левіподібний» або «броуноподібний» зразок маршруту дозволяє спрогнозувати зменшення активного часу використання батареї через зовнішні впливи. Можна припустити, що з метою зменшення додаткового обчислювального навантаження, пов'язаного з розрахунком змін траєкторії / зменшення заряду батареї, можна замінити розрахунок траєкторії броунівського руху в «польоті Леві» на розрахунки за регулярними функціями, що значно заощадить обчислювальну потужність.

Коли в достатку прикладів масовості або тривалості критичної ситуації (сильний вітер, зграя птахів на шляху і т. п.), броунівський підхід є правильним вибором, оскільки відображає мікропереміщення БПЛА з поверненням на маршрут в результаті роботи системи управління, – тобто, БПЛА рухаються траєкторією, близькою до розрахункової.

Однак, коли критична ситуація одиночна (несподівана перешкода на маршруті), в русі БПЛА з'являються зразки довгих траєкторій «польоту Леві». У русі Леві можливі так звані «польоти Леві», що створюють розриви в траєкторії БПЛА за аналогією з рухом частинок.

Таким чином, поведінка дронів фрактала може бути з достатньою точністю описана математично в просторово-часовому масштабі з двома параметрами моделі випадкового блукання. Переміщення суб-роїв і зграй БПЛА в географічних координатах можна оцінювати як супердифузний процес. Проведені дослідження значно поліпшують якість прогнозів витрати батареї обмеженої потужності на борту БПЛА і, відповідно, визначення зменшення часу активного польоту дронів у результаті прогнозованих (сила вітру) та стохастичних (несподівані перешкоди у вигляді птахів, проводів і т. п.) зовнішніх впливів.

Існує ще багато проблем, коли представляють групу дронів з інтелектом рою. Схеми в цих алгоритмах розглядають найбільш глобальні взаємодії і вводять велику кількість випадкових ходів для більш непередбачуваних результатів. Проте, ці операції будуть недоступні для суб-роїв роботів, зважаючи на окрему специфіку їхніх завдань. Тому необхідно розробляти алгоритми поведінки окремих суб-роїв з урахуванням унікальності, масштабованості й гнучкості виконуваних ними завдань у складі зграї.

1.4 Аналіз існуючих апаратно-програмних систем для вирішення задачі керування БПЛА

Апаратною основою будь-якого БПЛА є польотний контролер, який акумулює дані з датчиків, управляє стабілізацією і положенням в просторі. У процесі керування відправляють команди польотному контролеру або з пульта, або з наземної станції (Ground Station). У результаті змінюються обороти двигунів, що дозволяє БПЛА переміщатися в повітрі. Використання бортового міні-комп'ютера з підключенням безпосередньо до польотного контролера поєднує в собі переваги ручного пульта і наземної станції, дозволяє автоматизувати політ і взаємодіяти із зовнішнім світом, використовуючи виміряні параметри навколишнього середовища для здійснення автономного польоту.

Для розробки у подальшому методів та алгоритмів створення та функціонування суб-роїв у складі зграї БПЛА необхідно сформулювати вимоги до наявності електронних компонентів та програмних функцій, які реалізовані компаніями-виробниками на борту БПЛА. Основні критерії до автопілоту можна згрупувати за трьома типами вимог.

Обов'язкові:

1. Функція Return-to-Home (RTH), що зекономить обчислювальні потужності для повернення до точки старту при визначеному рівні розрядки батарей.
2. Функція «Круїз контроль» для утримання висоти польоту або висоти і курсу польоту.
3. Функція On Screen Display (OSD), яка надає можливість відображати поточні координати та інші польотні параметри, дрона поверх зображення з камери.
4. Показання напруги з усіх батарей, що дозволить коригувати час виконання завдання дроном в залежності від залишкової напруги.
5. Показання абсолютної швидкості, що дозволить планувати розмір площі для обстеження.
6. Показання висоти (наявність альтиметра на борту), що надасть можливість врахувати особливості рельєфу під час прокладання маршруту.
7. Показання відстані до точки зльоту.
8. Компас для забезпечення траєкторії польоту відповідно до прокладеного теоретичного маршруту.
9. GPS-трекер для точного визначення місцезнаходження БПЛА.

Додаткові:

1. Показання температури за бортом, що дозволить діагностувати критичне застосування БПЛА відповідно до параметрів працездатності електронного обладнання дрону (в межах мінус 20°C + 40°C).
2. Показання відносної швидкості, що дозволить розробляти алгоритми функціонування зграї при відсутності GPS-координат.
3. Функції контролю периметра, що зекономить обчислювальні потужності для повернення до точки старту під час перетину кордону.
4. Стабілізація під час відносно сильних вібрацій (при віражах та на швидкості більше 90 км/год), що зекономить обчислювальні потужності для компенсації «ефекту желе» на зображеннях.

Необхідні для здійснення керованого польоту:

1. Передача координат у реальному часі (з відображенням положення дрону на Google або Apple mapі).
2. Можливість підключення антени для збільшення дальності польоту за межі прямої видимості оператором (більше ніж 5 км).

Порівняння функцій автопілотів різних виробників наведено у табл. 1.1.

Таблиця 1.1. Порівняння функцій автопілотів різних виробників [110]

	APM	DJI (Naza w/ GPS)	DJI (Wookong-M Waypoint)	ZenUAV (YS-X4 low end)	ZenUAV (YS-X8 High end)	Mikrokopter	HobbyKing w/ GPS	iThine (Ruby)
Contents	Autopilot, GPS, Power Monitor, Long-Range Telemetry	Autopilot, GPS, Power Monitor	Autopilot, GPS, Power Monitor, Telemetry, 50 waypoint package	Autopilot, GPS, Power Monitor, Telemetry, 8 waypoint package	Autopilot, GPS, Power Monitor, Long-Range Telemetry, 16 waypoint package	Autopilot, GPS, Power Monitor, Long-Range Telemetry, 100 waypoint package	Autopilot, GPS only	Autopilot, GPS
List Price	\$249	\$399	~\$3000	\$639	~\$2700	~\$2500	\$900	\$345
Full autonomy	Yes	RTL only	Yes	Yes	Yes	Yes	Yes	Yes
Waypoints (# supported)	199	1	50	8 w/in 500m (Navigator)	16 w/in 500m (Standard)	100 w/in 250m	1	1
In-flight Route Editing	Yes	No	Yes	Guided Mode Only	Guided Mode Only	Yes	No	No
Wireless Configuration	Yes (Radio Telemetry)	No	Yes (Bluetooth)	No	No	No	No	No
Geo-Fence Support	Yes	No	No	No	No	No	No	No
Robust Simulation (HIL/SIL)	Both	No	HIL Only	No	No	SIL Only	No	No
Open Source Tools and Code	Yes	No	No	No	No	No	No	No
Open Source Communication Protocol	Yes (MAVLink)	No	No	No	No	No	No	No
Tricopter	Yes	No	No	No	No	No	No	No
Quadcopter (+ and X)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Hexacopter	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Decoopter	Yes	No	Yes	Yes	Yes	Yes	Yes	No
Y6	Yes	Yes	Yes	Yes	Yes	No	Yes	No
X6	Yes	Yes	Yes	Yes	Yes	No	Yes	No
Traditional helicopters	Yes	No	Yes (Ace One)	No	No	No	No	No
Fixed-wing Aircraft	Yes	No	No	No	No	No	No	Yes
Robots	Yes	No	No	No	No	No	No	No
Boats	Yes	No	No	No	No	No	No	No

Таким чином, при виборі компонентів БПА особливу увагу потрібно приділити вибору автопілоту, котрий має бути оснащений комплектом необхідних сенсорів, матиме можливість встановлення польотних точок та редагування маршруту в польоті, а також відкритий вихідний код програмного забезпечення (ПЗ) та відкритий протокол обміну даними, підтримувати встановлення на різні типи БПА.

Перевагу у виборі автопілотів БПА мають критерії, які забезпечують здатність функціонувати у складі зграї у складних недетермінованих умовах, а саме: використання сервісів Google, управління декількома дронами одночасно, вибір точок маршруту на мапі, комунікацію під час польоту.

SmartAP – це наступне покоління системи керування польотом для мультироторних безпілотних літальних апаратів, що здатні здійснювати повністю автономний політ [4].

Така система має потужний мікроконтролер STM32F4 від виробника ST Microelectronics, 9-осьовий інерціальний вимірювальний блок та сучасний GPS-приймач UBlox, інтегрований в системну плату. SmartAP підтримує будь-який тип мультироторних БПА. SmartAP розроблений з використанням новітніх електронних компонентів, які доступні в даний момент, що дозволило досягти високих результатів.

Загальні характеристики автопілоту наведені нижче [56]:

- мікроконтролер 32 біт, 168 МГц STM32F4 на ядрі Cortex M4;
 - вага 60 г;
 - GPS-приймач (UBlox NEO8, GPS/ГЛОНАСС, 10 Гц);
 - інтегрований модуль телеметрії (100 мВт), зовнішній модуль також підтримується;
 - USB-інтерфейс для конфігурації;
 - інтерфейси UART/модуля USART, інтерфейсу RS-232, інтерфейсів I2C та SPI;
- Сенсори, якими обладнаний апаратний модуль:
- магнітометр (Honeywell HMC5883L);
 - барометр (MS5611);
 - гіростабілізатор (InvenSense MPU-9150);

Цей автопілот дозволяє керувати пристроями в трьох режимах: під керуванням оператора, в напівавтоматичному режимі (підтримка поточного стану), та в повністю автоматичному режимі залежно від задачі. Система позиціонування дозволяє керувати

висотою з точністю до 10 см та положенням в просторі до 40 см. Також імплементовано систему автоповернення до місця старту.

Зовнішній вигляд системи автопілотування SmartAP зображено на рис. 1.4.



Рис. 1.4. Апаратний модуль системи автопілотування SmartAP [56]

Програмне забезпечення для цього апаратного модуля розповсюджується безкоштовно та має добре документовані зовнішні інтерфейси, що дозволяє модифікувати зазначену систему для вирішення широкого кола задач.

Vector – це повністю інтегрований автопілот від виробника Eagle Tree (США) для використання в сучасних повітряних БПЛА і платформ, де важлива автономність та висока кінцева продуктивність польоту [185]. Дане рішення сумісне з усіма типами платформ, в тому числі з фіксованим крилом та мультироторних БПЛА; він також підходить в якості бази для управління польотом системи опціонально-пілотованого корабля (ОПК) або перетворення пілотованих літаків у БПЛА. На рис. 1.5 наведено структурну модель автопілоту Vector.

Vector – це повноцінний та самодостатній автопілот, він оснащений всіма необхідними сенсорами обчислювальними пристроями, працює як єдина система керування польотом, яка має наступні характеристики:

- рівень захисту IP67;
- двоядерний процесор для керування польотом;
- відкритий рівень конфігурації, що дає можливість широкої модифікації автопілоту;
- інтегровану систему POLAR, яка забезпечує Vector набором необхідних датчиків (GPS, альтиметр, гіроскоп) для забезпечення ефективного управління польотом.

Головні особливості автопілоту наведені нижче:

- ОС реального часу, що дозволяє забезпечити безвідмовну роботу системи за рахунок пріоритетизації завдань, за якого основні ресурси системи використовуються для підтримки польоту;
- повний набір засобів сповіщення;
- сучасні алгоритми позиціонування;
- механізми забезпечення постійного зв'язку;
- автоматизація систем управління БПЛА (автоматичний зліт/посадка та інше).

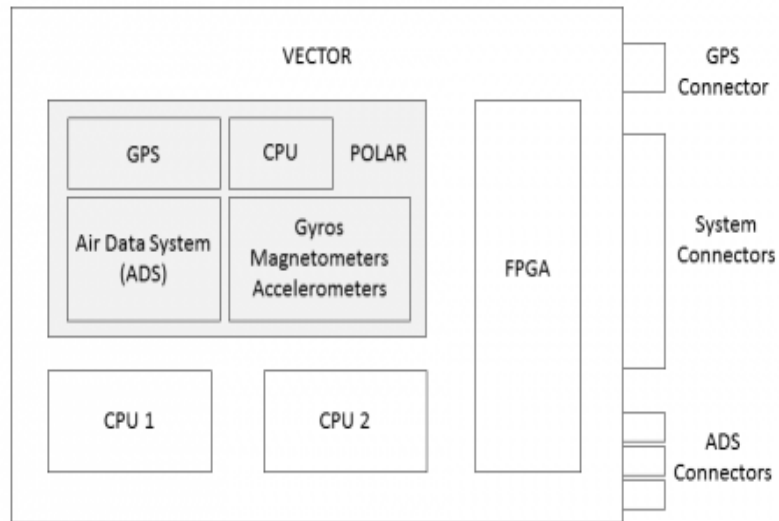


Рис. 1.5. Структурна схема побудови системи автопілотування Vector

Польотний контролер ArduPilot Mega є повноцінним рішенням БПЛА, який дозволяє крім радіокерованого дистанційного пілотування – автоматичне керування за створеним маршрутом, тобто політ по точкам (до 166), і також має можливість двосторонньої передачі телеметричних даних з борту на наземну станцію (телефон, планшет, ноутбук, тощо) і ведення журналу у вбудовану пам'ять.

Контролер ArduPilot Mega заснований на автопілоті APM 2.x, які розробляються спільноту DIY Drones і базується на Open-Source проєкті, що дозволяє перетворити будь-який апарат в автономний [10].

Pixhawk – це нова, розвинута система автопілоту, побудована на базі проєкту з відкритими вихідними кодами PX4 і виготовлена компанією 3D Robotics (3DR). Серед основних достоїнств можна відзначити потужний 32-бітний процесор і сенсор від компанії ST Microelectronics® та оперативну систему реального часу NuttX, що забезпечує високу продуктивність, гнучкість і надійність під час керування будь-яким автономним пристроєм. Система вміщується на microSD-карті обсягом 8 Гбайт, з яких 7,2 Гбайт займає розділ під польотні логи [143].

При включенні контролер створює WiFi-мережу. Доступ до контролеру здійснюється за IP-адресою засобами протоколу SSH.

Система автопілота 3DR Pixhawk може досягти необхідної швидкості польоту 20 м/с, несучи на борту камери та інші датчики, такі як LIDAR, сонар і модулі інерціального виміру.

Останні версії апаратно-програмного забезпечення функції автопілотування квадрокоптерів компанії DJI з інтегрованими програмними рішеннями від компанії Auto Flight Logic LLC набули суттєвих вдосконалень [15].

У порівнянні з іншими, зазначений автопілот тепер обладнаний замість 4 камер для зондування простору 6 камерами-датчиками на задній панелі. Це забезпечує огляд за 5-ма напрямками та дозволяє ще краще уникати перешкод. Зазначені камери у комплекті з парою сенсорів на передній панелі, а також набору інфрачервоних систем зондування з лівого і правого боків, забезпечують коптеру повний огляд простору і можливість визначити перешкоди на відстані 30 м від себе. Виявивши перешкоди (стіни, гілки, дах), апарат самостійно обчислює найбільш зручний спосіб повернути від них, і якщо не знаходить нового маршруту, то буде чекати команд оператора.

Крім того, останні моделі Phantom (Phantom 4 та вищі) літають доволі довго – 30 хвилин без підзарядки за рахунок більш потужного акумулятора LiPo (5870 мА•год).

Всі польотні режими зберігаються, як у попередників.

У режимі Sport («швидка гонка») дрон розвиває швидкість до 72 км/год.

Новий 1-дюймовий 20-мегапіксельний сенсор CMOS і має розширені можливості для розпізнавання образів у порівнянні з попередніми моделями завдяки управлінню різкістю, візуальними ефектами і яскравістю кадрів.

Відстань передачі сигналу онлайн збільшена до 5–7 км, що дозволяє поглибити дослідження у напрямі розробки нових топологій РММ зі зменшеною кількістю трансферних вузлів.

Оптимальне налагодження режимів роботи апаратних компонентів квадрокоптерів компанії DJI забезпечується лінійкою відповідних програмних продуктів:

1. DJI Lightbridge.
2. DJI Ultimate Flight.
3. DJI Ronin Assistant.
4. DJI Go.

Крім того, описаний комплект сенсорів спільно з набором із засобів розробки, утиліт і документації SDK (Software Development Kit) надають більше можливостей для розробки алгоритмів функціонування БПЛА у складі суб-роїв та зграй в автономному режимі польоту. Необхідно зауважити, що SDK підтримує як IOS, так і Android.

1.5 Вибір апаратно-програмної платформи для розгортання програмного забезпечення у диспетчерському центрі керування РММ

Не виключено, що під час виконання завдання зграя БПЛА виходить за периметр, в межах якого така зграя має зв'язок з ДЦ. В такому випадку для запобігання втрати зв'язку доцільно використати відкриті Wi-Fi точки доступу (ТД, англ. Access Point або AP). Але необхідно вживати додаткових заходів щодо збереження конфіденційності та цілісності передаваної інформації між ДЦ та зграєю БПЛА.

Рішення OpenVPN Access Server, обране у якості засобу реалізації серверної частини, потребує для своєї роботи процесор з тактовою частотою не менше ніж 1 ГГц та мінімум 512 Мбайт оперативної пам'яті.

За умови постійного росту потужностей серверних машин, можна зробити висновок про те, що вищенаведені системні вимоги не є високими.

На серверному рівні таке апаратне забезпечення можна вирішити шляхом оренди хмарного (Cloud) сервера, VPS або виділеного сервера (Dedicated).

Віртуальний виділений сервер (англ. Virtual Private Server або VPS) – послуга, в рамках якої користувачеві надають віртуальний сервер. Це повноцінна альтернатива фізичного виділеного сервера з великою кількістю переваг, високою стабільністю, простотою в управлінні і налаштуванні, стійкістю до відмов і набагато меншими фінансовими витратами [157].

Хмарний сервер (Cloud Server) – послуга, в рамках якої користувачеві надається віртуальний сервер з можливістю гнучкого налаштування кількості ресурсів [71].

Dedicated – послуга, у рамках якої користувачеві надається виділений фізичний сервер. Кількість ресурсів обмежується лише його апаратною базою.

У табл. 1.2 відображені результати порівняльного аналізу серверних платформ, які можуть бути використані на ДЦ ГРКМ.

З точки зору вартості оренди та масштабованості серверів найбільш вигідним та зручним варіантом є розгортання серверної частини на Cloud-сервері.

Таблиця 1.2. Порівняльний аналіз серверних платформ

Характеристики	Тип сервера		
	VPS	Cloud	Dedicated
Масштабуємість	Низька	Висока	Висока
Безпосередня робота з апаратним забезпеченням	–	–	+
Ізоляція ядра ОС	–	+	+
Технології віртуалізації	Xen, OpenVZ, Virtuozzo	VMWare	
Firewall	На рівні ОС	На рівні ОС / вбудований	На рівні ОС / Вбудований / зовнішній
Процесор	Один з іншими інстансами VPS, є обмеження за максимальною частотою	Є обмеження за максимальною частотою	Обмежено лише апаратно
Система зберігання даних	Загальна з іншими інстансами VPS	Загальна з іншими інстансами Cloud-серверів	Незалежна
Вартість оренди	Низька	Низька	Висока

У якості серверної операційної системи, яка обіймає значну долю цього ринку, широко використовується Linux. Дистрибутиви Linux та його клони стають дедалі популярнішими у якості ОС не тільки для мейнфреймів та суперкомп'ютерів, але й для БПЛА [62].

Об'єднуючись з великою кількістю системних та прикладних застосунків, ядро Linux формує повнофункціональну ОС.

Найбільш популярні у якості серверних ОС наступні дистрибутиви:

– Debian: один з найстаріших дистрибутивів, який підтримується найбільшою спільнотою розробників. Слугує основою для створення великої кількості інших дистрибутивів;

– Ubuntu Server: дистрибутив, який базується на Debian, та розробкою якого займається компанія Canonical Ltd;

– CentOS: дистрибутив, який базується на вільних вихідних кодах комерційного дистрибутива RHEL (Red Hat Linux Enterprise) компанії Red Hat. Строк підтримки кожної версії CentOS складає 7 років, при цьому нові версії виходять кожні два роки. Це робить її дуже надійною, безпечною та підтримуваною серверною ОС на базі Linux.

У зв'язку з високим рівнем підтримки мережевих технологій, високою надійністю та масштабованістю рішень, безкоштовністю та низькими вимогами до апаратного забезпечення, використання ОС на базі Linux в якості середовища для розгортання OpenVPN-сервера є обґрунтованим.

1.6 Планування маршрутів РММ

Велика кількість факторів, які впливають на вибір шляху пересування БПЛА та велика кількість вимог, пов'язаних з державним регулюванням можливих шляхів пересування БПЛА, роблять цю проблему дуже складною [142; 164; 319]. Отже однією з головних задач є необхідність знайти оптимальне рішення, яке буде дешевим, але

також буде виконувати велику кількість визначених умов. Недоліком є те, що складові критеріїв прийняття рішення можуть часто суперечити одна одній, тому виникає необхідність в їхньому балансуванні.

У цих умовах потрібно знайти маршрут, який буде задовольняти принцип оптимізації Парето, таким чином ми можемо отримати точне рішення. Фактори, які можуть вплинути на остаточне рішення щодо прокладання маршруту, дуже різноманітні і можуть варіюватися в конкретних випадках (структура рельєфу, метеоумови, наявність та зайнятість інших повітряних транспортних коридорів, економічні та соціальні аспекти і т. д.).

Через вплив великої кількості факторів, умов і вимог, процес прийняття рішень доцільно виконувати за допомогою нейронної мережі. Для формування просторових коридорів пересування БПЛА існує позитивний досвід використання нейронної мережі Хопфілда (НМХ) [79; 98; 258; 300].

Основна проблема під час маршрутизації полягає в тому щоб знайти найкоротший шлях між двома точками (А і В). Математично кажучи, це евклідова відстань, яка є прямою лінією траєкторії (рис. 1.6а). У реалістичних умовах зайняті транспортні коридори (або заборонені державним регулюванням для пересування БПЛА) вводять обмеження, і найкоротша відстань визначається мінімальною сумою індивідуальної довжини відрізків, що з'єднують точку А з точкою В (рис. 1.6б). Розв'язання цієї проблеми визначається алгоритмом Дейкстри [195].

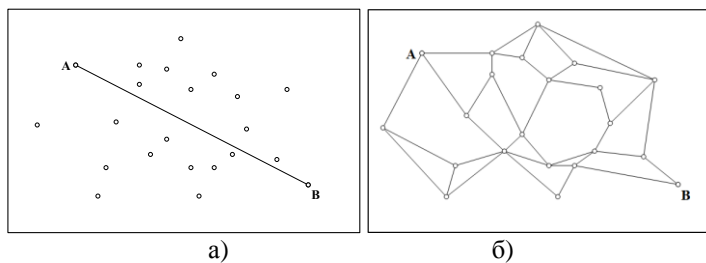


Рис. 1.6. Найкоротша відстань між точками А і В:
а – в ідеалі, б – у реалістичних умовах інфраструктурної мережі

У разі, якщо знадобиться розробити новий шлях серед існуючої інфраструктури (тобто повітряних транспортних коридорів, що вже використовуються), необхідно проаналізувати всі фактори, які можуть на це вплинути. Навіть, якщо існує тільки один фактор, який є реалістичним і просторово орієнтованим, відстань між двома точками має бути параметрично обґрунтованою. Це особливо важливо у випадку, коли спостерігаються чинники, які можуть приймати різні значення в різних просторових зонах (рис. 1.7).

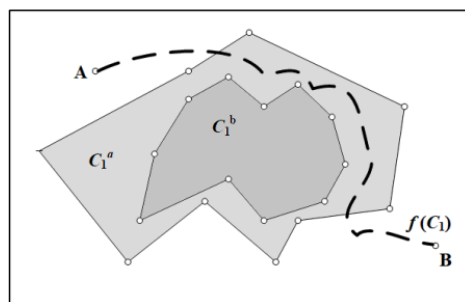


Рис. 1.7. Найкоротша можлива відстань між точками А і В,
коли фактор має два можливих значення, C_1^a й C_1^b

Наприклад, якщо просторова зона C_1 розташована між точками А і В, з двома різними кількісними областями (C_1^a та C_1^b), то відстань між точками залежить від одиниці вартості побудови маршруту через області C_1^a та C_1^b . Проблема вибору в такому випадку залежить від співвідношення ціни та довжини. У разі, коли загальна вартість побудови маршруту через C_1^b значно більша, ніж загальна вартість поза цією областю, оптимальним рішенням був би обхід навколо цієї області, що дозволило б зробити маршрут довшим, але безпечнішим або дешевшим. Проблема вибору маршруту масштабується під час аналізу великої кількості можливих чинників, особливо, коли кожен фактор може мати свої власні складові (рис. 1.8).

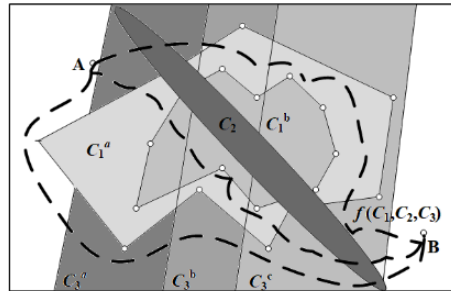


Рис. 1.8. Найкоротша можлива відстань між точками А і В

Найкоротша можлива відстань між точками А і В (рис. 1.8) у просторових зонах C_1 , C_2 та C_3 . C_1 має два можливих значення (C_1^a та C_1^b), C_2 – одне можливе значення, а C_3 має три можливих значення (C_3^a , C_3^b та C_3^c).

Запропонований Коїс та Релїн алгоритм прокладання оптимального шляху складається з трьох етапів [98]: моделювання рельєфу місцевості, управління (закладені умови, вимоги і параметри нейронної мережі) і обробка даних (блок-схема представлена на рис. 1.9).

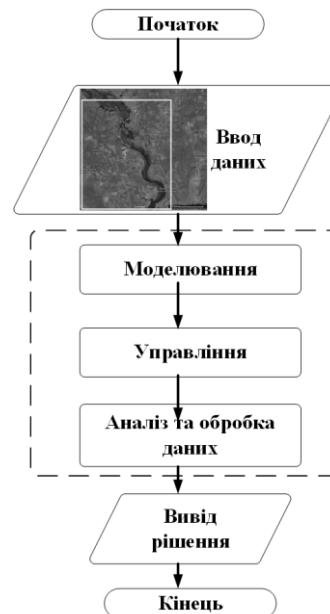


Рис. 1.9. Блок-схема алгоритму, запропонованого Коїс та Релїн

Методика складання мапи місцевості включає в себе розподілення віртуальної мережі на географічній мапі регіону. Мапа може бути розроблена з довільними розмірами квадрата форми $r \times r$. При виборі великих значень параметра r , результуючі регіони мають менші розміри і рішення виходить більш точним. Така область є найменшою формою, із якою може працювати алгоритм, і розроблене рішення може бути використано для ряду регіонів та географічних ділянок, які пропонуються для прокладання шляху.

Управління включає в себе визначення всіх умов і вимог, обмежень та параметрів пропонованої нейронної мережі, які можуть вплинути на кінцевий результат, його точність, швидкість запропонованого алгоритму і т. д. Цей етап також охоплює визначення параметрів, необхідних для роботи алгоритму і умов, передбачених в реальному середовищі. Наприклад, дані обмеження можуть бути пов'язаними із детермінацією чіткої умови: чи може маршрут проходити через певне місто (район), або більш ефективним рішенням буде обхід даної географічної зони.

На етапі аналізу і обробки всі дані з попередніх двох фаз використовуються для створення штучної нейронної мережі. Заради потреб такого алгоритму, було розроблено чітко визначену, особливу логіку, засновану на основі НМХ. Завдання НМХ – обробка всіх даних і пропонування Парето-оптимального рішення, яке враховує всі фактори, які подаються на вхід. Мета НМХ полягає в тому, щоб отримати математичне представлення шуканого вирішення (маршруту), що оптимально підходить його графічній візуалізації. Рішення має надати оптимальний баланс всіх факторів, їх ціни, умови, вимоги і т. д. Подальшою задачею алгоритму є необхідність знайти оптимальне рішення для прокладання маршруту відносно географічної ділянки, яка розглядається. Як результат, фаза дає на виході матрицю V розмірністю $r \times r$, заповнену значеннями «0» або «1». Значення «1» вказує на те, що цю ділянку можна розглядати як точку маршруту, тоді як «0» вказує на протилежне. Залежно від моделі рельєфу і точності, з якою розробляється класифікація, отримане рішення є більш-менш детальним.

Для моделювання Коїс та Рел'їн використали НМХ, структуру якої зображено на рис. 1.10.

Ряд робіт показує, що НМХ дають хороші результати в задачах знаходження найкоротшого або оптимального шляху. З іншого боку, основним недоліком мережі цього виду є її можлива нестабільність і той факт, що НМХ не завжди дає оптимальне рішення [26; 69; 134; 146]. Проте, як показав Хопфілд, у випадку, якщо отримане рішення не є оптимальним, його можна буде віднести до категорії рішень, які дуже близькі до оптимальних, тобто, є суб-оптимальними [79].

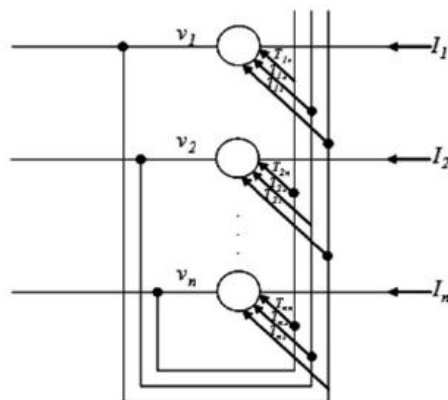


Рис. 1.10. Модель нейронної мережі Хопфілда [98]

Використання НМХ для аналізу пересування БПЛА на теперішній час не є класичним застосуванням та деякими авторами визнається як неперспективне. Тому накопичення досліджень у цьому напрямі, а також вдосконалення критеріїв, умов, та методів обробки результатів як моделювання, так і апаратних реалізацій, у подальшому покажуть доцільність або негатив такого підходу.

Моделювання поведінки БПЛА на основі нейронних мереж є альтернативою іншим видам моделювання на основі математичних методів, тому що їхні результати схожі. Проте, нейронні мережі швидше працюють і не вимагають великих обчислювальних потужностей або складних обчислень. На перший погляд, головний недолік полягає в потребі мати окремі нейронні мережі для кожного етапу польоту, як зазначалось раніше. Але математичні моделі також є специфічними для кожного з етапів, та одне з можливих рішень цієї проблеми може бути використане для іншої нейронної мережі, яка активує певну мережу у відповідний момент. У цілому, моделі ідентифікації використовують орієнтацію у просторі XYZ, щоб моделювати систему БПЛА.

1.7 Аналіз існуючих тривимірних мап для РММ

Для забезпечення функціонування розглянутих вище систем автопілотів БПЛА зазвичай застосовуються два види мап: це мапи від Apple, які входять у стандартний пакет Cocoa Touch [89] та Google Maps.

MapKit Framework надає інтерфейс для вбудовування мапи безпосередньо у вікна застосунку [108]. Цей фреймворк також забезпечує підтримку маркерів на мапі, додавання оверлеїв, а також виконання зворотнього геокодування вибірок для визначення інформації для обраних координат. У цьому фреймворку вже оптимізовано алгоритм перевикористання маркерів для економії оперативної пам'яті та процесорного часу. Маркери та оверлеї можуть бути кастомізовані під будь-які цілі.

Google Maps API має ті самі можливості що й MapKit, але використовує власні мапи, які на цей момент більш деталізовані [65]. Але щодо оптимізації пам'яті, то розробник має власноруч контролювати кількість маркерів, видаляти зайві та додавати нові не залишаючи зайві в оперативній пам'яті.

Але з розглянутих рішень та застосунків для РММ не було знайдено тривимірних мап, які задовольняють всі вимоги. Хоча обидві бібліотеки надають зручні фреймворки, які вже включають в себе тривимірні мапи, додавання міток, побудову маршруту та ін., однак, в результаті дослідження виявилось, що тривимірні мапи і від Google, і від Apple покривають далеко не всю територію та є досить приблизними. Так, наприклад, розглядаючи місто Миколаїв можна знайти недоліки обох бібліотек. Мапи від Google абсолютно не мають рельєфу, навіть на яскраво виражених підйомах, балках та береговій лінії, наприклад: <https://www.google.com/maps/@46.9036834,32.0311711,1707m/data=!3m1!1e3>.

А мапи від Apple взагалі відображають пагорби в місцях, де їх немає (рис. 1.11а).



а)



б)

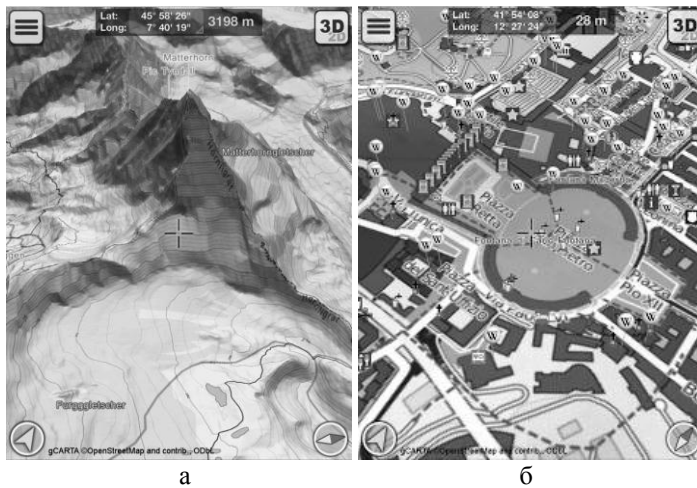
Рис. 1.11. Неіснуючий пагорб на місцевості з координатами Apple Maps (а) та Google Maps (б)

Оскільки жоден застосунок для РММ не має тривимірних мап, розглянемо можливі варіанти в інших сферах застосування на платформі iOS.

FATMAP Ski – застосунок для альпіністів, що має тривимірні мапи найвідоміших лижних курортів [55]. Застосунок підтримує GPS-трекінг та має досить деталізовані мапи (рис. 1.12).



Рис. 1.12. Приклад 3D-мапи в застосунку FATMAP Ski [55]



а

б

Рис. 1.13. Приклад 3D-мапи в застосунку GeoFlyer 3D Map Europe: за межами населеного пункту (а) та у населеному пункті (б) [61]

GeoFlyer 3D Map Europe – iOS-застосунок для GPS-трекінга, прокладання маршруту на тривимірних мапах [61]. Застосунок має автономний режим роботи та призначений для пішоходів та велосипедистів. Мапи використовують OpenStreetMap API для рендерингу поверхні. Головне вікно має табло з інформацією про місцезнаходження та висоту над рівнем моря у будь-якій точці світу, потрібно лише навести спеціальний маркер посередині вікна на потрібну область. Та, незважаючи на детальну інформацію, рельєф відображається лише у гірській місцевості (рис. 1.12а). А наприклад, місто залишається на пласкій місцевості, зазначена лише висота найвищої будівлі (рис. 1.12б). Тому така мапа не є корисною для організації функціонування РММ у населених пунктах.

Компанія Drone.UA розробила повний пакет послуг щодо налаштування моніторингової мережі, від надання БПЛА до ПЗ на iOS та Android.

Drone.UA має власне ПЗ для абонентів БПЛА Pix4Dmapper Capture [67]. Застосунок Pix4Dmapper DJI дозволяє за допомогою квадрокоптера DJI Phantom 2 створювати тривимірну картографічну поверхню над областю, яку виділить користувач. БПЛА виконує знімки повністю автоматично і не потребує ручного управління. Після завершення зйомки отримані фото передаються на десктопну версію застосунку, де вже виконується створення тривимірної поверхні.

Цей факт і є головним недоліком мобільного застосунку, вся обробка інформації виконується не на мобільній платформі, що робить дане ПЗ незручним для користувача. Приклад роботи застосунку під час сканування території наведений на рис. 1.14.

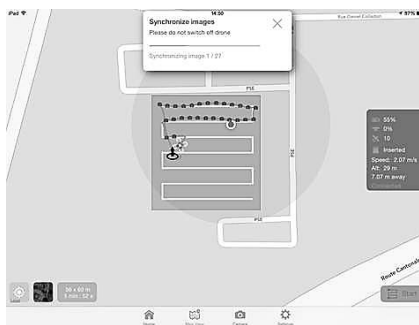


Рис. 1.14. Робота застосунку Pix4Dmapper Capture [138]

Таким чином, можна дійти висновку, що існуючі програмні застосунки побудування тривимірних мап місцевості не є корисними для організації функціонування РММ у населених пунктах.

Висновки до розділу 1

У ході дослідження та аналізу сучасних методів управління групою БПЛА, можна зробити висновок, що підходи до вирішення подібних задач відрізняються один від одного в залежності від типу умов в яких працює система (детермінованому чи недетермінованому), характеристик моніторингових мереж (гетерогенних чи гомогенних, однорангових чи багаторангових), типу цільових задач. У деяких випадках доцільно використовувати системи з «лідуючим» пристроєм, який буде задавати напрям руху іншим пристроям в групі, інколи краще використовувати централізоване управління всією групою пристроєм.

Для обміну інформацією між БПА та з КЦ доцільно використовувати всі наявні за ситуацією канали зв'язку, більшістю бездротові. Необхідно розглядати також здатність рою взаємодіяти без комунікації, в умовах роботи засобів радіоелектронної

боротьби або поганих метеоумов. Не варто занижувати ефективність використання маркерів у середовищі. Однак, зв'язок і сенсорування може дійсно підвищити ефективність рою.

Поєднання можливостей мікро- та макроскопічного моделювання надає можливість передбачити поведінку рою у складних ситуаціях, які не можна відтворити засобами моделювання на базі сенсорів. Таким чином, тільки комбінацією декількох існуючих методів моделювання можна забезпечити життєздатність зграї під час критичного застосування.

Обраним підходом є використання готових рішень систем управління польотом одиночних пристроїв, як складової частини системи управління моніторинговими мережами. Дослідженні існуючі рішення, на прикладі SmartAP та Vector, доводять реалістичність такого підходу. Практика його використання доводить, що прийняте рішення дозволяє зекономити ресурс обмежених потужностей обчислювача на борту БПЛА. У подальшому це дозволить розробити алгоритми, які подовжують час життя груп БПЛА.

Дослідивши існуючі програмні системи для управління поодинокими БПЛА нами було зроблено висновок, що розвиток технологій в цій сфері можна використати під час побудови рухомих моніторингових мереж.

Перевагу у виборі автопілотів БПА мають критерії, які забезпечують здатність функціонувати у складі зграї у складних недетермінованих умовах, а саме: використання сервісів Google, управління декількома дронами одночасно, вибір точок маршруту на мапі, комунікацію під час польоту. Підсумовуючи вищенаведене, можна надати перевагу апаратно-програмним компонентам фірми DJI. Вагомим підґрунтям для цього є те, що DJI надала доступ до пакету засобів розробки (SDK) для своїх квадрокоптерів серії Phantom. Використовуючи SDK, розробники ПЗ, включаючи PixiePath, Pix4D, Drone Deploy, Bright Sky Labs, Field of View і Skyward.io, вже розпочали створювати програмні застосунки, які дозволяють пілотам DJI Phantom редагувати і обмінюватися відео на мобільних пристроях, створювати 3D-мапи, забезпечуючи польоти з дотриманням правил і керуючи кількома польотами одночасно. Варто зазначити, що разом з випуском нових моделей дронів версії SDK мають оновлюватись.

Зважаючи на постійне доповнення функціоналу дронів, існуючі розробки з відкритим кодом доцільно розширяти шляхом створення власного ПЗ для реалізації додаткових сервісів у функціонування зграї БПЛА.

Мобільні системи моніторингу мають широке коло застосувань у різних сферах людської діяльності. Наукові розробки вже були проведені провідними світовими університетами в галузі робототехнічних бджолиних роїв та інших систем моніторингу. В Україні є дві провідні компанії Drone.UA та Parrot AR.Drone з моніторингу в аграрній, топографічній та інших сферах [67; 132]. Кожна із вищезазначених компаній має свої власні програми на платформі мобільних пристроїв iOS, які мають багато функцій та переваг. Але всі вони використовують 2D-карти, що є недоліком для планування оптимально безпечно польоту літака. Тривимірні карти можуть бути використані для усунення загального недоліку. Однак, вони не відображають особливостей рельєфу в степовій зоні та в місті.

Аналізуючи існуючі програми на платформі iOS для MMN, можна зробити висновок, що створення тривимірних карт для абонентів MMN є досить нерозкритим питанням і вимагає подальших досліджень.

Використання хмарних технологій, тривимірних мап місцевості, розподілення Леві у комбінації з моделлю броунівського руху, а також методології нейронних мереж Хопфілда дозволять планувати безаварійні маршрути як для всієї зграї, так і кожного об'єкта кожного суб-рою.

Розділ 2

ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ КЕРУВАННЯ ГРУПОЮ БПЛА ІЗ ЗАПОБІГАННЯМ ЗІТКНЕНЬ

Підходи до управління групою БПЛА своїми складовими схожі на ті, що використовуються для керування одиночними дронами, за тією відмінністю, що при використанні групи пристроїв необхідно виконати ті ж самі операції управління, але вже для більшої кількості об'єктів системи та враховувати їх взаєморозташування. Так як ГРКМ найчастіше використовуються з метою дослідження великих територій, тому важливо правильно оцінити задачу, знайти оптимальну кількість та характеристики дронів, побудувати маршрути для кожного об'єкта зграї та забезпечити систему делегування завдань при критичній ситуації (наприклад, в разі втрати одного або декількох дронів, тощо).

2.1 Завдання визначення цільової задачі

Основною задачею більшості рухомих мереж є вирішення задач моніторингу. Тому, окрім безпосереднього управління фізичними пристроями, система керування такою мережею має також виконувати функції інтеграційного шару між людиною та АС.

Система керування (СК) має надавати відповідний інтерфейс для сторонніх програмних рішень, через який надходять задачі для груп пристроїв.

Для цього потрібно розробити механізми класифікації вхідних та вихідних даних моніторингу, методи та алгоритми їхнього аналізу та обробки. Цільова задача для РММ має бути представлена у вигляді, який буде зрозумілий для СК.

Основним підходом для вирішення проблем позиціонування пристрою в просторі є використання БПЛА навігаційних систем, наприклад, таких як GPS. Ця система може забезпечити доволі точне позиціонування пристрою в просторі. Ще однією перевагою даної технології є її широке використання, безкоштовність та можливість керування точністю позиціонування за допомогою налаштування технічних характеристик модулів [237; 297]. Наразі навігаційні системи використовуються в багатьох безпілотних проектах, таких як SmartAP, ArduPilot, 3DR SOLO Drone та інші.

Виходячи з цього, доцільним буде використання координат GPS як вхідних даних при постановці завдання для моніторингової мережі.

Водночас, просто набір координат не може служити повноцінним та однозначним описом задачі, тому необхідно розробити підхід формалізації задач моніторингу.

Постановка задачі визначення оптимального способу обходу заданої цілі наведена нижче. Маємо:

- характеристики БПЛА;
- мета спостереження, задана географічними координатами кутових точок;
- вид мети (точкова, лінійна або замкнений контур).

Вимагається:

- побудувати оптимальну траєкторію покриття мапи.

Траєкторія визначається координатами точок, над якими має пролетіти БПЛА.

Існує три принципово різних типи цілей спостереження:

- задача слідування лінійним маршрутом (рис. 2.1);
- задача моніторингу замкнутого контура (площинний, рис. 2.2);
- задача відвідування контрольних точок (рис. 2.3).

Тип цілі визначається в залежності від її форми і розмірів, а так само від розмірів видимої для сенсорів БПЛА області, залежить від поточного режиму переміщення.

Побудова траєкторії обходу точкових і лінійних об'єктів значно простіша, ніж площинних. Обхід точкових об'єктів зводиться до пересування ними, а до лінійних – уздовж них.



Рис. 2.1. Приклад задачі слідування за лінійним маршрутом



Рис. 2.2. Приклад задачі з замкненим контуром



Рис. 2.3. Приклад задачі з контрольними точками

Виходячи з досвіду використання БПЛА, візьмо до уваги:

- траєкторія проходу повинна бути простою (для полегшення обробки одержуваних даних з БПЛА і орієнтації дрону);
- траєкторія має бути циклічною (для можливості поширювати цикли певної одиничної площі на площі довільних форм).

Уявімо площу покриття у вигляді смуги, середня лінія якої є траєкторією проекції оптичної осі датчика БПЛА, що знімає показання навколишнього середовища. Накопичений досвід використання БПЛА передбачає форму траєкторії у вигляді «змійки» (рис. 2.4), що відповідає прийнятим вище міркуванням.



Рис. 2.4. Найпростіший вид траєкторії обльоту замкнутого контура

Оптимальна траєкторія обльоту площинної цілі характеризується мінімальною довжиною серед інших траєкторій, а також мінімальним числом розворотів над ціллю і відповідно великою кількістю прямих ділянок.

Для площинних цілей, представлених опуклими багатокутниками, мінімальна кількість розворотів, а також мінімальна довжина траєкторії буде досягтися тоді, коли прямі ділянки секторів спостереження будуть паралельні прямій між двома найбільш віддаленими вершинами багатокутника.

Кожна з таких задач вимагає окремого підходу для вирішення задачі планування маршруту кожного елемента мережі.

2.2 Задача планування маршруту

Задача планування маршруту обходу групою БПЛА множини цілей ставиться наступним чином [258; 300].

Маємо:

- кількість БПЛА та їхні значущі характеристики (координати початкового місце-знаходження, діапазон швидкостей, дальність польоту, кут огляду, мінімальний радіус розвороту);
- кількість цілей та їхні значущі характеристики (мета спостереження, задана географічними координатами кутових точок і видом мети (лінійна, площинна або точкова).

Вимагається:

- розробити критерій якості планування обльоту групою БПЛА множини цілей.

Завдання планування обходу може бути сформульоване в термінах теорії прийняття рішень, теорії комбінаторної оптимізації, зокрема, теорії розкладів та інших. При використанні будь-якого формалізму завдання передбачає велику розмірність простору рішень. Розподіл цілей між БПЛА, послідовність проходження кожним БПЛА підмножини цілей, спосіб переміщення до кожної мети, спосіб її моніторингу – це все параметри, які визначають величезну кількість варіантів рішень задачі планування навіть при урахуванні всіх

обмежень. У якості критерію ефективності пропонується використовувати нормований час об'їзду всіх цілей. Функція оптимізації ефективності дискретна і немонотонна у своїй області визначення і має велику кількість локальних екстремумів. Іншими словами, стоїть завдання оптимізації дискретної функції в багатовимірному дискретному просторі векторів рішень з урахуванням численних обмежень.

Вихідна задача планування декомпонується на три суб-задачі з метою значного зменшення розмірності області визначення:

а) завдання верхнього рівня – завдання призначення. У ньому потрібно розподілити безліч цілей у групі БПЛА (отримання значення цільової функції передбачає рішення задачі середнього рівня);

б) завдання середнього рівня вимагає для кожного БПЛА визначити послідовність проходження призначеної йому на верхньому рівні групи цілей (ця задача зводиться до задачі комівояжера);

в) виконання завдання нижнього рівня вимагає знайти оптимальний спосіб проходження цілі одним БПЛА.

Цю задачу можна представити, як модифіковану задачу транспортної маршрутизації з навантаженням (Capacitated Vehicle Routing Problem). Нехай визначено неорієнтований граф $G = (V, E)$, де множина вершин $V = \{0, 1, \dots, n\}$ – це множина цілей; E – множина ребер. Кожній вершині назначено відповідну вагу:

$$d_k, k = \overline{0, n}. \quad (2.1)$$

Вершина з індексом «0» є базою, з якої стартують всі БПЛА представлені множиною $Q = \{0, 1, \dots, n\}$, де кожний БПЛА має обмеження щодо ваги маршруту:

$$R_l, l = \overline{0, m}. \quad (2.2)$$

Кожне ребро $E_{i,j}$ має невід'ємну вагу $c_{i,j} = c_{j,i}$.

Тоді необхідно визначити безліч маршрутів БПЛА з мінімальною загальною вагою:

$$W = \sum_0^m w_l, \quad (2.3)$$

таких щоб кожен маршрут починався і закінчувався на «базі», а кожна ціль відвідувалася лише один раз:

$$w_l = \sum_0^{r+1} c_i + \sum_0^r d_i, \quad (2.4)$$

де w_l – сумарна вага маршруту кожного БПЛА;

$\sum_0^{r+1} c_i$ – вага всіх $r + 1$ ребер у маршруті;

$\sum_0^r d_i$ – вага всіх r вершин. Причому, $w_l \leq R_l$.

Таким чином, ваги вершин $d_k, k = \overline{0, n}$ відображають витрати на обхід конкретної мети і є шляхом рішення задачі нижнього рівня.

Задача верхнього рівня вирішується шляхом розрахунку необхідної кількості елементів групи виходячи з їх характеристик. Нехай відомі такі параметри: загальна площа ЦЗ (S_{all}), відстань від бази до цілі (l), характеристики БПЛА (швидкість v , запас енергії акумулятора тощо); продуктивність (P).

Для того, щоб оптимізувати виконання задачі, необхідно скоротити час її обробки. Для цього необхідно вирахувати мінімальний час виконання задачі для кожного БПЛА, а потім укомплектувати ними групу. Отже, оптимальним рішенням цієї задачі є:

$$D_{coef} = t_{mt} + t_{mb} + t_{op}, \quad (2.5)$$

де t_{mt} – час пересування до цілі; t_{mb} – час пересування від цілі; t_{op} – час виконання задачі.

Таким чином, моніторингова група буде формуватися за алгоритмом, наведеним на рис. 2.5. Загальний алгоритм вирішення задачі керування групою БПЛА зображено на рис. 2.6.

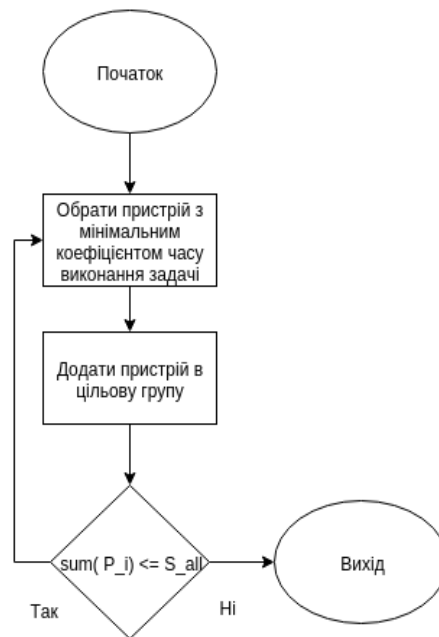


Рис. 2.5. Алгоритм розподілу групи пристроїв



Рис. 2.6. Алгоритм вирішення задачі керування групою БПА

Після дослідження специфіки задач моніторингу було зроблено висновок, що всі задачі з замкнутим контуром характеризуються наявністю граничних меж площі моніторингу, що може бути використане для побудови моделі ЦЗ. ЦЗ може мати будь-яку форму – як правильну, так і у формі багатогранника. Для зменшення енергозатрат системи моніторингу доцільно розробити методи спрощення ЦЗ. ЦЗ, що надходить до системи управління, має наступну структуру (рис. 2.7):

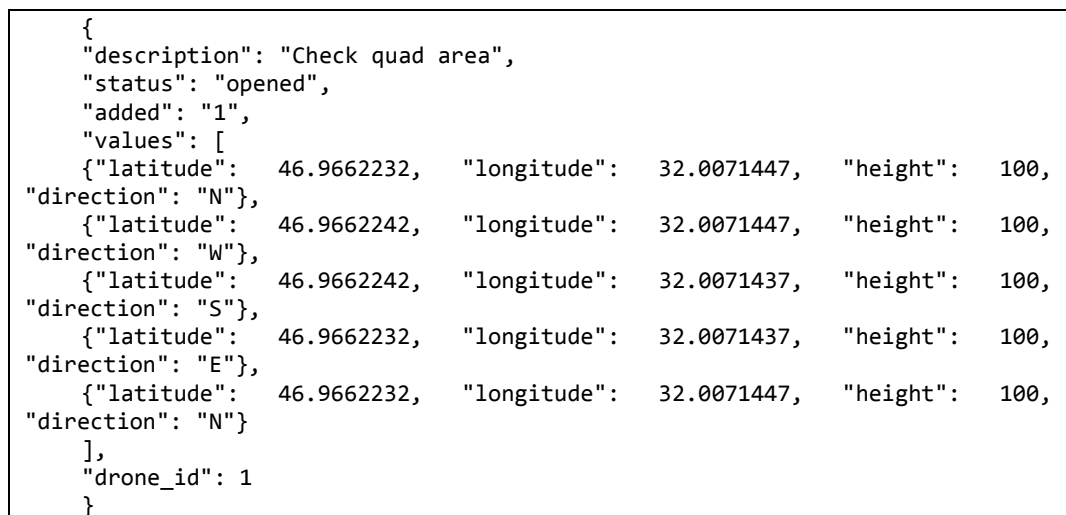


Рис. 2.7. Структура ЦЗ, що надходить до системи управління

Тобто, для того щоб отримати спрощену модель задачі, необхідно знайти координати крайньої північної, південної, східної та західної точок. Приклад спрощення ЦЗ показано на рис. 2.8.



Рис. 2.8. Приклад спрощення ЦЗ:
 1 – крайня північна точка;
 2 – крайня західна точка;
 3 – крайня південна точка;
 4 – крайня східна точка

Таким чином отримана спрощена ЦЗ, що дозволить моніторинговим пристроям зменшити витрати на переміщення, а системі керування – витрати часу на обробку ЦЗ.

2.3 Визначення способу моніторингу цілі

Наступною задачею є визначення області інтересу для кожного з суб-роїв зграї. Проблему було вирішено шляхом розбиття загальної площі області інтересу на сектори.

Для ефективного виконання задач моніторингу СК необхідно оперувати як інформацією про задачу, так і про пристрої, які знаходяться під її управлінням. Використовуючи наявну інформацію можна визначити яку площу може моніторити той чи інший пристрій, і виходячи вже з цього оптимально розподілити ЦЗ на сектори та сформуванати моніторингову групу, розділивши зграю на суб-рої. Приклад розподілу ЦЗ на сектори між чотирма суб-роями моніторингової мережі показано на рис. 2.9.

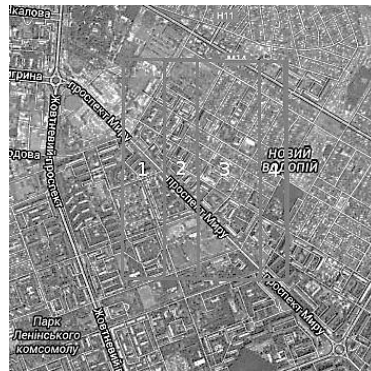


Рис. 2.9. Приклад розподілу ЦЗ на сектори 1, 2, 3, 4
 (області функціонування кожного суб-рою зграї)

Розподіл ЦЗ на сектори залежить від фізичних можливостей апаратних платформ, таких як час автономного польоту, площа захоплення моніторингових даних і т. п.

2.4 Моделювання поведінки суб-роїв роботів у просторових коридорах

Розглянемо методи побудови та функціонування РМ, яка складається з суб-роїв по 2 БПА (робота, дрона тощо) в кожному; кожен суб-рій функціонує у власному просторовому коридорі. На рис. 2.10а наведена ЦЗ моделювання поведінки спарених роботів на фармацевтичному складі. На рис. 2.10б ЦЗ суб-рою з двох БПЛА – побудування канатного моста.

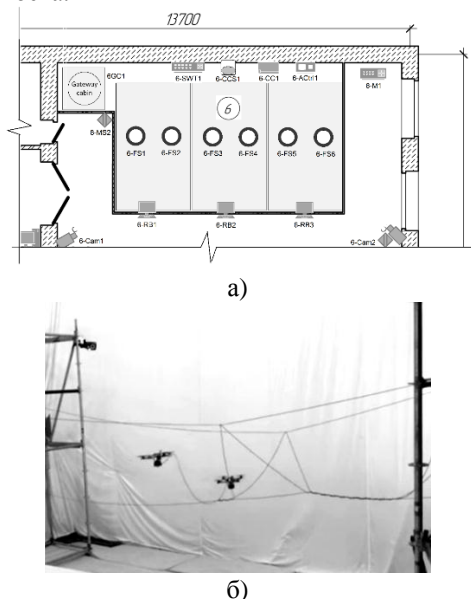


Рис. 2.10. Суб-рої роботів, що складаються з 2 БПЛА (FSi) кожен, розташовані в 3 наземних складських коридорах (а) та в одному коридорі повітряного простору (б) [6]

ГРKM базуються на роях (суб-роях) та зграях таких роботів, як безпілотні наземні апарати (англ. UGV), безпілотні підводні апарати (англ. UUV), безпілотні надводні (поверхневі) апарати (англ. USV) або безпілотні літальні апарати (БПЛА або «дрони»; англ. UAV).

Технологічні завдання використання рою роботів будь-якого виду передбачають алгоритмізацію переміщення роботів всередині обмеженого простору. Досить великим є клас задач, що передбачає потрапляння одного з роботів в зумовлену точку з чітко означеними координатами. Причому, кожна точка повинна бути переглянута хоча б одним з роботів.

Математично така задача зводиться до класичної задачі комівояжера [68; 288]. Але актуальними також є завдання, у яких поведінка робота-«комівояжера» описується не тільки в географічних координатах (2D-завдання), але також враховується, на якій висоті переміщується такий робот. При такому підході має вирішуватися 3D-задача комівояжера [74].

Крім того, весь використовуваний рій роботів часто розбивається на групи (так звані суб-рої), кожна з яких виконує своє завдання. Хоча такі завдання і є аналогічними тим, що виконуються іншою групою, але кожна з таких груп діє у своєму просторовому коридорі.

Проблема полягає в тому, що існуючі алгоритми вирішення так званої «задачі комівояжера» (англ. Traveling Salesman Problem або TSP) на пошук оптимального маршруту витрачають значну кількість обчислювальних і часових ресурсів – до 10 с для 20 точок маршруту [317]. Однак, в швидкодинамічних системах, пов'язаних зі швидкістю

переміщення БПЛА від 5 м/с до 20 м/с [15], просторовий коридор може бути пройдений швидше, ніж буде розрахований оптимальний маршрут.

Тому необхідно розглянути можливість приведення 3D-завдання TSP до більш простої і швидко розв'язуваної двовимірної задачі, або до розбиття 3D-завдання на кілька двовимірних, які паралельно вирішуються на різних обчислювальних ресурсах, з подальшою конкатенацією знайдених рішень.

Крім того, в класичних рішеннях 3D-завдання salesman-«комівояжер» представляється просторовою точкою, яка не має фізичного об'єму [216]. У реальних умовах такий підхід може призвести до краху роботів у результаті зіткнень один з одним, або вимагатиме додаткового обсягу обчислень, пов'язаних зі спробою уникнути перешкод.

Усе вищевикладене логічно призводить до необхідності попереднього проведення математичного моделювання з метою уникнення критичних ситуацій при використанні рою роботів у обмеженому просторі.

Моделювання групової поведінки рою роботів передбачає розбиття останніх на групи (суб-рої), які виконують аналогічні завдання в обмежених просторових коридорах (рис. 2.10).

Наприклад, при вирішенні завдання відбору необхідних за замовленням лікарських засобів на фармацевтичному складі, необхідно оптимізувати рух групи роботів, які працюють спільно в одному з просторових коридорів: 6-FS1 & 6-FS2, 6-FS3 & 6-FS4, 6-FS5 & 6-FS6 (на рис. 2.10а, у приміщенні № 6).

Необхідно відзначити, що така постановка задачі руху роботів у просторовому коридорі не є класичною задачею TSP. Наприклад, для створення міцної конструкції, що витримає людину (рис. 2.10б), підвісний канатний міст має бути побудований з різних елементів, таких як вузли, опорні елементи і т. п. [12]. Для виконання таких елементів UAV-«комівояжер» повинен пройти через певні вузли маршруту декілька раз. До того ж, у процесі такого «будівництва» UAV повинен кілька разів повертатися на базу (стартову чи іншу певну точку) для підзарядки акумуляторних батарей або для поповнення котушки будівельним матеріалом (мотузкою).

Зважаючи на те, що в обох випадках на рис. 2.10 розглядається коридор з боковими ячейками, немає необхідності під час переміщення з однієї частини коридора на іншу враховувати точки посеред коридора. Завдяки цьому таку задачу можна вважати не 3D-задачею, а композицією двох 2D-задач TSP: на лівій та правій вертикальній площині в кожному коридорі окремо.

У такому разі розглядається задача забезпечення узгодженого руху роботів тільки в межах кожного суб-рою окремо задля виконання спільного завдання в кожному коридорі (підбору ліків або ін.). Тоді у випадку, якщо єдиний керуючий центр відсутній, правила руху кожного робота визначаються його діями та діями його сусідів. Для більш ефективного вирішення завдання узгодженого руху необхідно виділити процедуру планування, де кожен член рою визначає для себе певні початкові дії. Такі дії обумовлені обмеженими можливостями членів рою, але потім в залежності від дій інших членів рою відбувається їхнє коригування.

Алгоритми вирішення TSP ґрунтуються на пошуку на графі найкоротшої відстані від вихідної точки до якоїсь необхідної точки, яка відповідає кінцю шляху.

З іншого боку, відомі алгоритми не враховують появу перешкод, які є членами колективу роботів і можуть змінювати своє положення. Це призводить до того, що виникатимуть непередбачені перешкоди, і час на досягнення цілі збільшиться.

Для початку процесу рішення задачі необхідно мати:

- список найменувань ліків/вузлів та їх кількість;
- координати розташування цих ліків/вузлів на площині боків просторового коридору (в ячейках);
- власні координати кожного робота.

У процесі ці дані мають змінюватися:

- список необхідних ліків/вузлів має скорочуватися;
- власні координати робота змінюються;
- координати всіх членів рою роботів змінюються.

Потрібно зауважити, що у разі використання роботів доставки (як на розглянутому фармаскладі), необхідне їх повернення на початкову позицію, для розкладу ліків за замовленням чи завершеним списком, або в результаті переповнення рухомого контейнера, або за іншими обмежувальними умовами.

Оскільки обмін інформацією між членами рою обмежений, використання одного з алгоритмів пошуку шляху, таких як алгоритм Дейкстри, хвильовий алгоритм та ін., в процесі руху до цілі буде неефективним [126].

Найбільш відповідним варіантом буде спланувати узгоджені дії членів колективу, а потім слідувати цьому плану. Для цього всім роботам команди необхідно зробити початковий вибір, щоб потім мати можливість оцінити його ефективність у порівнянні з іншим сценарієм. Кожен з роботів визначає найближчу ціль.

Вибір цілі (ліків, технологічних вузлів тощо) кожним роботом здійснюється за допомогою спеціальної матриці вибору:

$$T = \left\| t_{ij} \right\|, \quad (i = 0, 1, \dots, n; j = 1, 2, \dots, m). \quad (2.6)$$

Наприклад, для фармаскладу матриця вибору являє собою матрицю розміром $n \times m$, де n – загальна кількість роботів, m – кількість найменувань ліків, t_{ij} – відстань, пройдена i -м роботом для досягнення ячейки з j -м лікарським засобом. Кожен i -й робот знаходить мінімальне значення $T_{mn, \min}$ та вважає його найоптимальнішим. Потім перевіряються відстані всіх роботів, і завдання отримує той робот, яким має найменшу відстань і не має інших завдань; модель переобчислюється, і вибір цілей виконується заново. У разі, якщо цілі не помінялися, обраний план затверджується.

Таким чином є можливість оцінити справжній час досягнення роботом ячейки без урахування всіх можливих перешкод, які можуть виникнути на його шляху, що дозволяє вибрати дійсно оптимальний план.

Для реалізації можливості обліку змін зовнішнього середовища (можливості відслідковувати переміщення інших роботів), необхідно через кожен визначений момент часу отримувати нові дані, враховувати можливий перетин траєкторій, і встановити ієрархію зупинок роботів, тобто перераховувати X_{mn} і здійснювати пошук мінімуму в матриці вибору.

Необхідно також створити прототип динамічної інтелектуальної системи, який включає компонент імітаційного моделювання переміщення і виконання технологічних дій (збір ліків, зав'язування вузлів тощо) автоматичними пристроями в обмеженому просторі з рухомими перешкодами. Імітаційна модель системи задається наступною теоретико-множинною моделлю:

$$M_{im} = \langle MOU, MK, MCB, VU, VE, VY, VZ, S, F1, F2 \rangle \quad (2.7),$$

де MOU – модель об'єкта управління, описує стан і характеристики робота;

MK – модель контролера, описує принципи роботи робота;

MBB – модель внутрішніх випадків, описує виникнення випадкових подій, вагу ліків, їх кількість (обмежену);

VU – множина контрольованих керованих параметрів, технічних характеристик робота (швидкість пересування, швидкість збирання, максимальний заряд акумулятора, вага ліків тощо);

VE – множина випадкових збурень під час раптового виникнення динамічних перешкод (наприклад, інших роботів, які рухаються за власними траєкторіями);

VY – множина вихідних параметрів MOU, що використовуються в контролері (координати робота, його поточний стан і оточення тощо);

VZ – множина вихідних параметрів МОУ (поведінка робота при заданих умовах, час повного збору списку тощо);

S – множина станів системи (всі допустимі комбінації елементів множин VU, VE, VY, VZ);

F1 – функція генерації керуючого вектора (правила і процедури, які відповідають за встановлення нових значень керованих параметрів);

F2 – функція відображення входу ОУ в його вихід (правила і процедури, за якими розраховуються значення вихідних параметрів).

Розглянемо детальніше управління рухом. У завданнях управління моделлю VY відповідає за повороти роботів, переміщення апаратів й ін. Виникає необхідність розрахунку положення та орієнтації твердих тіл в дискретні моменти часу. Передані команди управління містять код кожного з заздалегідь визначених маневрів, параметри зазначеного маневру, дані про момент початку маневру та його закінчення. Об'єкти здійснюють складний просторовий рух, який розглядається як комбінація поступального та обертального рухів деякої точки або сукупності точок.

Розглянемо підхід до реалізації алгоритмів повороту, пов'язаний з використанням цілочисельної арифметики. Серед алгоритмів, здатних вирішити цю задачу, може бути обраним алгоритм Брассіні [27], який активно використовується зарубіжними дослідниками та інженерами для апаратної реалізації поворотів.

Розглянемо алгоритм, який реалізує прямолінійний рух з використанням тільки цілочисельних операндів. Згідно з алгоритмом Брассіні для генерації точок на прямій лінії, точка з цілими координатами (X, Y) належить прямій лінії, яка виходить з початку координат з коефіцієнтом нахилу $k = (\Delta Y / \Delta X)$, якщо виконується умова:

$$T = \left\| t_{ij} \right\|, \quad (i = 0, 1, \dots, n; j = 1, 2, \dots, m), \quad (2.8)$$

де $|\epsilon| \leq 0,5$.

Використовуючи тільки цілочисельні змінні $X, Y, \Delta X, \Delta Y$, можливо сформулювати таку умову приналежності точки прямій лінії:

$$- \Delta X \leq 2\Delta Y X - 2\Delta X Y \leq \Delta X.$$

Тоді для зростаючої послідовності значень X , які відрізняються на одиницю, відповідні значення Y можна знайти на основі перевірки виконання наведеного співвідношення.

Стосовно до задачі управління роботами, алгоритм може бути використаний для розрахунку траєкторії руху об'єкта під будь-яким заданим кутом в реальному часі.

Для вирішення задачі плоского повороту на основі алгоритму Брассіні [27] вихідний об'єкт (тверде тіло) представляється в загальному випадку у вигляді матриці елементів – системи фізичних точок, причому в простому випадку маємо єдину точку (X, Y) . Потрібно виконати поворот системи точок на заданий кут.

У запропонованому алгоритмі повороту, заснованому на використанні моделі Брассіні [126], нові координати (X^*, Y^*) точки (X, Y) визначаються наступним чином:

$$\begin{aligned} |2LX^* - 2MX - 2NY| &\leq L, \\ |2LY^* + 2NX - 2MY| &\leq L. \end{aligned} \quad (2.9)$$

При виконанні перетворення повороту, в загальному випадку, кут повороту α задається трійкою (M, N, L) цілих чисел, причому $L^2 = M^2 + N^2$, $M/L = \cos \alpha$, $N/L = \sin \alpha$. Таблиця синусів і косинусів у вигляді частки цілих чисел обчислюється заздалегідь і зберігається в пам'яті системи управління поворотом.

Для послідовності значень координат X і Y , відповідні значення X^* і Y^* можна знайти на основі перевірки виконання співвідношення:

$$\begin{aligned} X^* &= (M/L)X + (N/L)Y, \\ Y^* &= (M/L)X + (N/L)Y, \end{aligned} \quad (2.10)$$

Час виконання алгоритму повороту оцінюється, як $\theta(k)$, де k – число точок об'єкта.

Алгоритм використовує для своєї реалізації тільки цілі числа і операції над ними, не спирається на пряме обчислення тригонометричних функцій, що забезпечує доцільність його застосування в системах управління об'єктами в рухомих системах при обмежених обчислювальних ресурсах. Передбачається, що цілочисельні алгоритми ляжуть в основу повнофункціональних засобів управління траєкторним рухом апаратів з обмеженими обчислювальними можливостями для виконання поставлених завдань. Крім того, вбудоване в бортовий обчислювач програмне забезпечення забезпечить визначення кутів орієнтації і параметрів руху; навігацію і управління під час руху заданою траєкторією, вибір кутів та орієнтацію в просторовому коридорі.

Наведена нижче модель даних була розроблена з урахуванням всіх зазначених параметрів. Тут кожна виробнича одиниця має власні параметри розміщення, такі як кількість стелажів, кількість полиць, спеціальний код позиції та ін. Модель даних була побудована в програмному середовищі платформи ARIS (рис. 2.11).

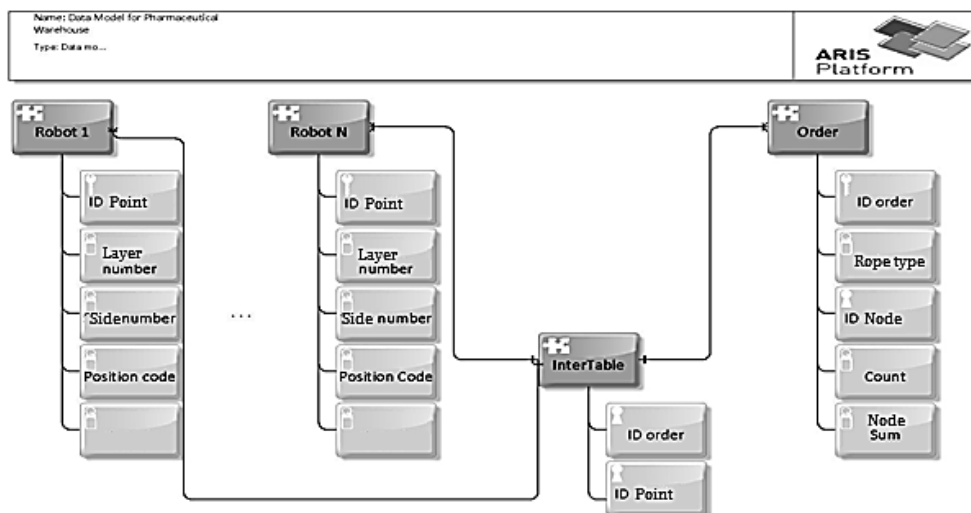


Рис. 2.11. Модель даних для N просторових коридорів

У задачах, які можна віднести до задач TSP, що вирішуються в розділених просторових коридорах, не можна обчислювати стійкі стани мережі роботів ітераційними методами (методами послідовних наближень) з обчисленням помилок. Необхідно всі коефіцієнти матриці (2.6) обчислювати за один цикл, після чого спільне завдання для зграї (групи) роботів розділяється на суб-завдання (кожне для свого суб-рою). На цьому етапі мережа переходить у стійкий стан, і в ньому залишається. Тобто, після розподілу суб-завдань між суб-роями роботів, ці завдання більше перерозподілятися не будуть до закінчення їх виконання.

Таке формулювання алгоритму найбільш близьке до алгоритму навчання мережі Хопфілда [69; 80]. Під час роботи НМХ ознакою знаходження рішення є такий стійкий стан, який збігається з одним із запам'ятованих при навчанні. З огляду на те, що місце розташування ліків на полицях не змінюється, колишні знайдені ліки без помилок будуть знайдені в цьому ж місці.

Є сенс розглядати синхронний режим роботи НМХ. Якщо при виконанні суб-роєм з двох роботів завдання пошуку і вибору ліків, роботами в своєму просторовому коридорі переглянуті всі полиці з однієї сторони коридора, тоді стани роботів одночасно (тобто синхронно) змінюються на нові, і вони можуть помінятися один з одним сторонами коридора. В іншому випадку можливе зіткнення їх у просторі і колапс мережі. Таким

чином, необхідно моделювати паралельну роботу роботів суб-рою, але послідовним алгоритмом.

Було змодельовано поведінку роїв роботів (по 2 роботи у суб-рою), кожен рій рухається у власному просторовому коридорі (рис. 2.12).

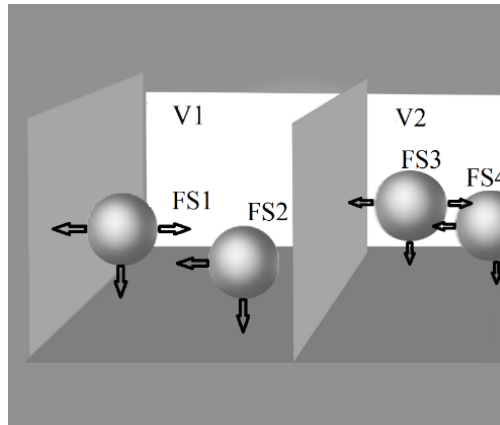


Рис. 2.12. Парний рух роботів

Факт виявлення зіткнень та їх уникнення під час руху кожного робота суб-рою необхідно моделювати окремо. Всі суб-рої мають різні напрямки руху. Продуктивність виконуваної роботи зростає, якщо розділити завдання між двома такими роботами одного суб-рою. Якщо територія виконання завдання всіма роботами (так званою зграєю або групою) має декілька розділених проходів (рядів) для переміщення, у кожному такому проході має працювати окремий суб-рій. Тоді швидкість виконання завдання всією зграєю збільшується.

Один робот виконує завдання не тільки на сусідніх полицях, а й на інших, завдяки своїй здатності обертання до 180 градусів.

Перераховані переваги підвищують продуктивність зграї та швидкість обробки запитів. Кінцевою операцією є видача зібраних ліків у вікно 6-RB, [265].

На рис. 2.13 показано моделювання процесу пошуку рішення за допомогою НМХ [118]. У деяких ситуаціях кілька механізмів, які рухаються один за одним у коридорі, можуть бути конструктивно поєднані в одного робота.

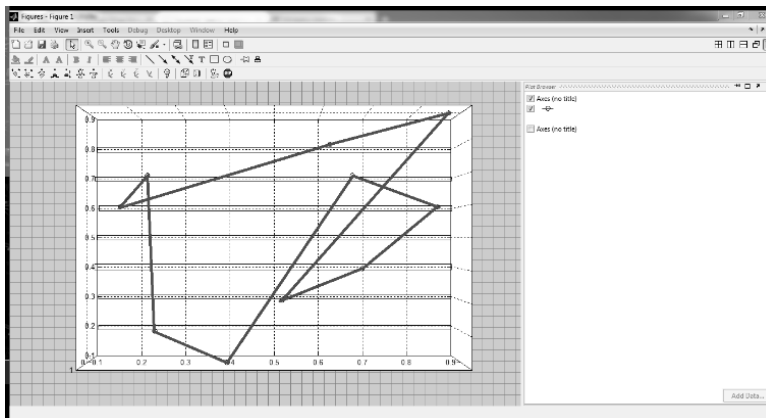


Рис. 2.13. Приклад рішення задачі TSP одним роботом з використанням мережі Хопфілда

На рис. 2.14 показано, що до значення 0,5 по осі абсцис Робот-1 (6-FS1) виконував 2D TSP з однієї сторони просторового коридора 6-V1 (в цей час Робот-2 (6-FS2) виконував аналогічну задачу з іншої сторони цього ж коридора 6-V1. Після значення 0,5 Робот-1 змінив сторону коридора і продовжив виконувати своє суб-завдання 2D TSP на вертикальній площині другої сторони просторового коридора 6-V1.

З огляду на те, що в зазначеному прикладі використовувані роботи не є самостійними, а представлені парним механізмом однієї конструкції, то для побудови подібної траєкторії (див. рис. 2.13) для Робота-2 суб-рою необхідно мережу Хопфілда реалізувати з паралельними обчисленнями. У цьому випадку, щоб розрахувати положення Робота-2 (6-FS2), потрібно знати оновлене положення Робота-1 (6-FS1). Зрозуміло, що фізично неможливо розрахувати положення Робота-1 і положення Робота-2 в один і той же момент часу, оскільки стан Робота-2 залежить від положення Робота-1.

Тобто, при збільшенні кількості роботів у суб-рої для розрахунку положення кожного робота потрібно знати оновлене і старе положення сусідніх роботів. Кількість тактів для передачі положень роботів збільшується при збільшенні кількості сусідніх роботів, стан яких має враховуватися.

Усе вищевикладене дозволяє говорити про використання мережі Хопфілда з асинхронним режимом.

За допомогою двох роботів (або спареної конструкції робота) можна ефективно, швидко і без зіткнень обробляти два стелажі (рис. 2.14а). Для моделювання такої ситуації доцільно кожен з роботів представити у вигляді сфери із заданим діаметром (рис. 2.14б, в).

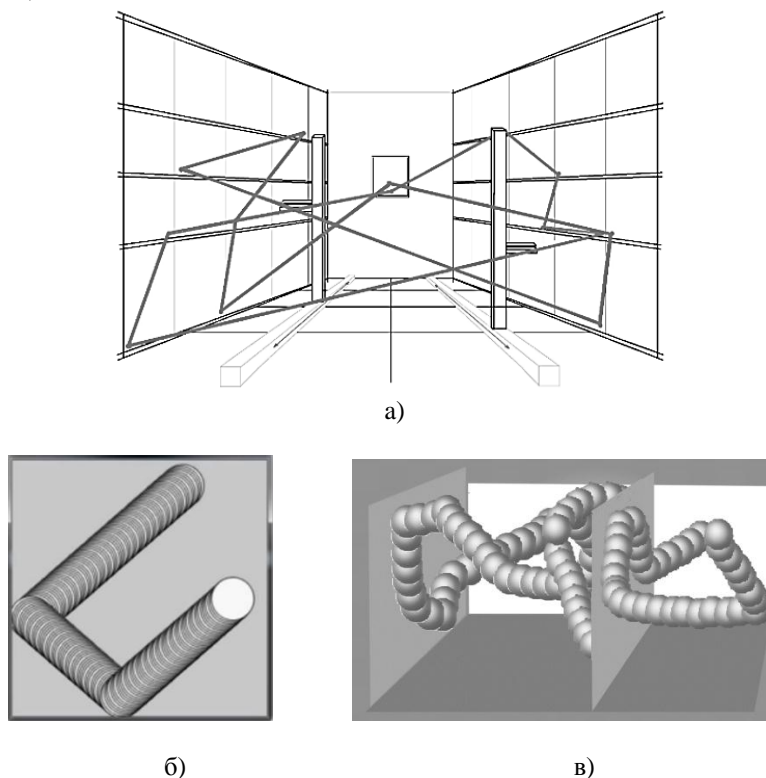


Рис. 2.14. Моделювання руху роботів у просторовому коридорі:
 (а) «комівояжер» у вигляді просторової точки, (б) «комівояжер» як коло,
 (в) маршрути суб-роїв сферичних роботів у сусідніх просторових коридорах

У такому випадку TSP пропонується вирішувати окремо для сфери, що лінійно переміщується (НЕ котиться!) спочатку однією бічною поверхнею просторового коридора, а потім – для цієї ж сфери, яка переміщується другою бічною поверхнею цього ж коридора (рис. 2.14в). При переході сфери з однієї бічної поверхні на іншу внутрішні точки коридора в рішенні TSP не розглядаються. Таким чином, замість вирішення тривимірної 3D-задачі TSP, можна використовувати конкатенацію двох 2D-задач TSP.

Під час моделювання руху роботів розглядаються три просторові координати:

- 1-ша координата: зліва або справа;
- 2-га координата: позиція цільової точки;
- 3-тя координата: номер сфери (технологічного механізму).

Моделювання поведінки підгрупи (суб-рою) може використовуватися на етапі підготовки для виконання завдання (моніторингу або технології) з оптимізацією руху об'єктів суб-рою в обмеженому просторі між сторонами та рівнями (наприклад, полицями відсіку фармаскладу) просторового коридору.

Таким чином, моделювання є дуже важливим при вивченні поведінки суб-роїв роботів; оскільки це дає можливість представити реальну систему, використовуючи запрограмовані математичні моделі, і дозволяє відображати динаміку системи, а також досліджувати поведінку сегментованої групи роботів.

Необхідно відзначити, що за таким підходом пройдений суб-роєм шлях не буде мінімальним, і описане рішення є тільки суб-оптимальним для TSP. Але використання НМХ дозволяє знаходити задовільні рішення досить швидко, без додаткового аналізу. Це робить можливим використовувати розроблений метод формування траєкторії роботів (дронів) суб-рою (з конкатенацією декількох 2D-рішень в 3D-просторі) в швидкодинамічних мережах, наприклад, таких як рій БПЛА.

Використання класичних методів, які дозволяють знаходити оптимальне рішення з використанням множинних ітерацій, може зажадати таких часових і обчислювальних ресурсів, які можуть привести до краху рою БПЛА при несвоєчасному отриманні корекції маршруту.

2.5 Синтез маршрутів суб-роїв БПЛА на основі нейронної мережі Хопфілда у просторових коридорах на місцевості

Розглянемо особливості організації групової взаємодії БПЛА на відкритій місцевості, за межами приміщення. Методика складання мапи місцевості включає в себе розподілення віртуальної мережі на географічній мапі регіону. Мапа може бути розроблена з довільними розмірами квадрата форми $r \times r$. При виборі великих значень параметра r , результуючі регіони мають менші розміри і рішення виходить більш точним. Така область є найменшою формою, з якою може працювати алгоритм, і розроблене рішення може бути використано для ряду регіонів та географічних ділянок, які пропонуються для прокладання шляху.

Управління включає в себе визначення всіх умов і вимог, обмежень та параметрів запропонованої нейронної мережі, що можуть вплинути на кінцевий результат, його точність, швидкість запропонованого алгоритму і т. д. Цей етап також включає в себе визначення параметрів, необхідних для роботи алгоритму і умов, передбачених в реальному середовищі. Наприклад, дані обмеження можуть бути пов'язаними із детермінацією чіткої умови: чи може маршрут проходити через певне місто чи район, або більш ефективним рішенням буде обхід цієї географічної зони.

На етапі аналізу і обробки всі дані з попередніх двох фаз використовуються для створення штучної нейронної мережі. Заради потреб цього алгоритму було розроблено чітко визначену логіку, засновану на основі НМХ. Завдання НМХ – обробка всіх даних і пропонування Парето-оптимального рішення, що враховує всі фактори, які подаються на вхід. Його мета полягає в тому, щоб отримати математичне представлення шуканого вирішення (маршруту), що оптимально підходить його графічній візуалізації. Рішення має надати оптимальний баланс всіх факторів, їх цінності, умови, вимоги і т. д. Подальшою задачею алгоритму є необхідність знайти оптимальне рішення для прокладання маршруту відносно географічної ділянки, яка розглядається. Як результат, ця фаза дає на виході матрицю V розмірністю $r \times r$, заповнену значеннями «0» або «1». Значення «1» вказує на те, що цю ділянку можна розглядати як точку маршруту, в той час як «0» вказує на протилежне. Залежно від моделі рельєфу і точності, з якою розробляється класифікація, отримане рішення є більш-менш детальним.

Основними етапами роботи НМХ, що використовується для планування шляху БПЛА, є:

1. Знайти оптимальне рішення Парето на основі певних чинників.
2. Збалансувати всі входи і визначені умови найкращим чином.
3. З'єднати початкову і кінцеву точки, в унікальному шляху, без переривання.
4. Переконатись, що кожній під'єднаній області присвоєне числове значення для кожного з факторів, на основі яких виконується оптимізація.
5. Забезпечити перевагу певних входів згідно з побажаннями розробника маршруту.
6. Забезпечити безумовне проходження через пов'язані між собою області, або їх перевизначення, якщо є така необхідність.
7. Підтримати роботу з довільним числом взаємопов'язаних областей.
8. Включити кожний з факторів, які варто класифікувати в довільне число поділок.
9. Забезпечити можливість вибору точності і швидкості реалізації шляхом зміни параметрів НМХ.
10. Переконатись, що, в разі будь-якої несправності, є можливість перервати пошук оптимального шляху, та інформувати користувача.

Основними перевагами нейронної мережі є низька обчислювальна потужність і здатність зберігати модель, як тільки мережа була навчена. До того ж, це робить непотрібними попередні знання, тому що мережа може навчатися. Перевагою НМХ є також можливість апаратної реалізації.

Процес ідентифікації має вирішальне значення для контролю, який забезпечує автономію БПЛА. Моделі управління мають бути протестовані з використанням імітаторів, і у випадку, якщо ідентифікація була точною, вони матимуть більш високу продуктивність в реальних умовах.

Проведене дослідження зосереджене на контрольованих нейронних мережах, які потребують моделі навчання для регулювання їх параметрів таким чином, щоб виходи мереж стали якомога ближчими до навчальних шаблонів. Після того, як мережа була скоригована відповідно до навчальної моделі, вона готова до використання і її параметри більше не будуть модифікуватись.

Таким чином, дуже важливо ретельно вибирати навчальні зразки, використовуючи найбільш підходящі шаблони для різних станів системи.

Найбільш ефективною для вирішення поставлених задач можна вважати гібридну мережеву архітектуру, яка об'єднує в собі якості рекурентної та нерекурентної мереж (рис. 2.16).

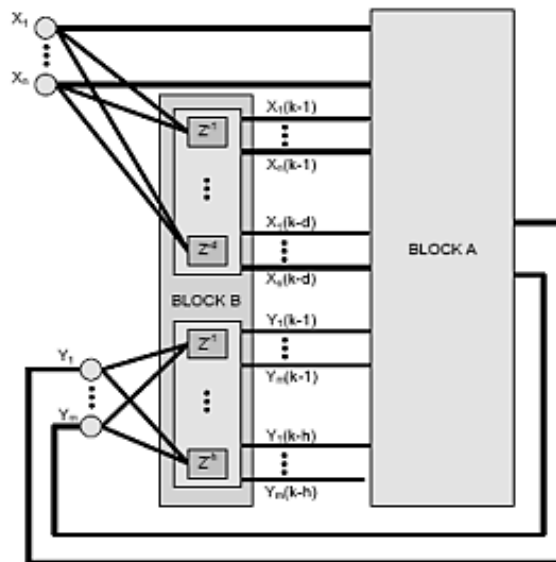


Рис. 2.16. Гібридна мережі Хопфілда

Розглянемо проблему прокладання оптимального шляху у контексті пересування БПЛА на місцевості.

Для ідентифікації такої системи, як БПЛА, потрібно виконати деякі коригування, котрі мають бути проаналізовані на гібридній мережі.

Політ БПЛА заснований на орієнтації у просторі та швидкості. По-перше визначаються причини, які призводять до зміни положення БПЛА. Таким чином, існують два етапи моделювання: перший складається з моделювання положення на основі команд, що посилаються до БПЛА, а другий складається з імітації позиції, заснований на положенні і попередніх станах.

Можна використати два різних методи для навчання контрольованої мережі. У першому з них, обидві системи з'єднуються каскадно, і їхній вихід використовується для навчання системи позиціонування. У другому, здійснюється паралельне навчання обох систем, але ця архітектура не бере до уваги помилку, коли дані йдуть з виходу першої мережі до входу наступної. Саме з цієї причини прийнято використовувати перший метод.

Модель структури гібридної НМХ запропоновано модифікувати за рахунок введення додаткового модуля аналізу географічних координат (рис. 2.17).

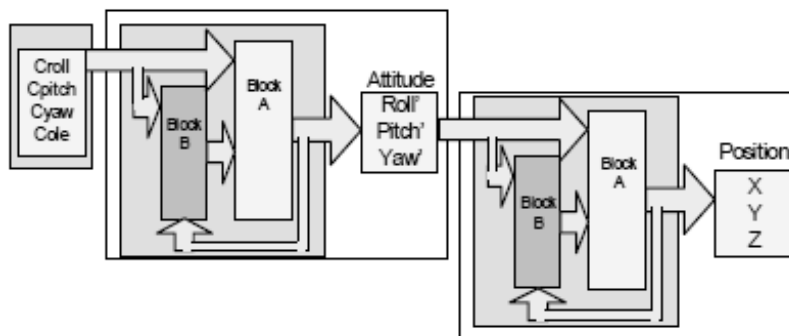


Рис. 2.17. Каскадна архітектура навчання

На рис. 2.17 наведена каскадна архітектура навчання системи польоту БПЛА. Для навчання використовуються дані, отримані з AHRS (курсовертикаль), від GPS, від радіопередавача і шаблони використані під час навчання.

Шаблоном для навчання є вектор T (2.11), який показує стан, отриманий від GPS.

$$T = [x(t), y(t), z(t)]^T. \quad (2.11)$$

Після того як система навчилася, моделювання працює послідовно, визначаючи положення, виходячи з відношення, тому ефективно працювати з каскадною архітектурою. Потреби у використанні вихідного сигналу системи в якості вхідних даних для наступної, передбачають можливість поширення помилки від відношень у мережі до її позицій. Цей метод навчання приймає помилку до уваги і виправляє її у відношеннях системи.

При порівнянні T (2.11) і T (2.12) для корекції повторюваних мережевих параметрів, помилки, які утворюються на виході, розглядаються як вхідний параметр x для рівняння (2.11):

$$T = [x'(t), y'(t), z'(t)]^T. \quad (2.12)$$

Таким чином, за результатами досліджень з існуючого математичного апарату був обраний ефективний метод динамічної ідентифікації таких систем, як БПЛА, та досліджені умови його застосування для цього класу задач. Таким методом було запропоновано різновид НМХ, а саме – контрольовані нейронні мережі Хопфілда.

Процес планування маршрутів БПЛА повинен виконуватись якщо не оптимальним чином, то таким, що забезпечує виконання низки задач всією зграєю з найменшою витратою обчислювальних та енергоспожитих ресурсів. Ефективним в такому випадку може бути метод синтезу маршрутів суб-роїв, з яких складається зграя боїдів, на основі НМХ.

Метою є створення такого методу синтезу маршрутів суб-роїв зграї, який забезпечить вирішення задачі комівояжера з виключенням дубляжу виконання задач у вузлах, якими промаркована досліджувана територія, та з суттєвим прискоренням синтезу маршрутів БПЛА у порівнянні з існуючими методами. Необхідність досягнення зазначеної мети обумовлена обмеженістю енергоресурсів боїдів, що не дозволяє виконувати великий обсяг обчислень, якого потребує застосування класичних методів вирішення задачі TSP. Актуальність досліджень підвищується, якщо завдання обстеження територій пов'язані з надзвичайними ситуаціями та потребують негайних рішень в надзвичайно обмежені строки.

Наочними задачами, для вирішення яких доцільно застосовувати зграї (рої) невеликих боїдів з обмеженими обчислювальними та енергоресурсами, наприклад, є вирішення задач агросектору, огляд технічних споруд, моніторинг метеопоказників, поширення тріщин під час землетрусів, води – при повенях тощо. У подібних задачах завжди актуальними є питання визначення маршруту з моніторингом певних характеристик у визначених вузлах на чітко обмеженій території. Такий маршрут повинен бути, якщо й не оптимальним, то мінімальним за часом та довжиною траєкторії для прийняття своєчасних рішень.

Якщо ці умови не будуть виконані, можливі два варіанти критичного закінчення маршруту зграєю боїдів:

- невиконання задачі обстеження всіх заданих точок маршруту;
- неповернення чисельних боїдів до місця вильоту через зіткнення або закінчення ресурсу елементів електроживлення на борту боїдів.

Враховуючи вищенаведене, задача моніторингу характеристик території з чітко визначеними розмірами може бути приведена до класичної задачі комівояжера з нетрадиційним шляхом вирішення останньої.

Розглянемо моделювання групової поведінки зграї боїдів з урахуванням розбиття їх на суб-рої, які виконують аналогічні задачі в обмежених просторових коридорах.

Для економії обчислювальних ресурсів і продовження життя зграї поведінку боїда-«комівояжера» пропонується описувати тільки в географічних координатах (lat_{ij} , lon_{ij}), тобто, вирішувати 2D-задачу TSP. Висота h , на якій переміщатиметься кожен окремий суб-рій, фіксується залежно від рельєфу місцевості, над якою працює суб-рій. При такому підході повинна вирішуватися 2D-задача комівояжера в кожному просторовому коридорі, в якому працює окремий суб-рій [126]. На відміну від [126] пропонується в розробленому методі формувати просторові коридори (блоки польотної мапи) не прямокутними, а достатньо довільної форми з суміжними межами, без дублювання членства вузлів у суміжних блоках.

Попадання будь-якого m -го боїда n -го суб-рою в точку з координатами (lat_{ij} , lon_{ij}) визначається або за GPS-координатами, або (за відсутністю зв'язку) порівнянням знімків фотокамери боїда з об'єктами з точною локалізацією, виявленими за Google або Apple мапами.

Отже, вхідними даними є масив опорних точок, через які має пролетіти боїд (або суб-рій). Для кожного n -го суб-рою вказана точка вильоту PF_n , яка також є точкою фінішу.

Розглянемо план місцевості, який покривається декількома координатними сітками з K рівномірно нанесеними в лінію контрольними вузлами на чітко визначеній відстані L (рис. 2.21а).

Кожна координатна сітка – це квадрат 10×10 вузлів, який є планом одного польоту зграї боїдів. Кількість суб-роїв у зграї, точки вильоту кожного суб-рою, висота польоту і відстань між вузлами, в яких повинні бути виконані технологічні або моніторингові завдання, визначається оператором цієї КФС перед кожним вильотом.

Якщо задача обстеження території виконується окремим БПЛА, з вильотом у точці PF_0 та закінченням завдання у точці PE_0 (рис. 2.21б), то загальна вартість такого маршруту становить приблизно 2620, якщо вважати, що ваговий коефіцієнт переміщення боїда між вузлами P_{ij} координатної сітки по прямій становить $w_{ij} = 1 k$, а по діагоналі – $w_{ij} = 1,4 k$, де k – коефіцієнт, що динамічно змінюється в залежності від швидкості переміщення боїда.

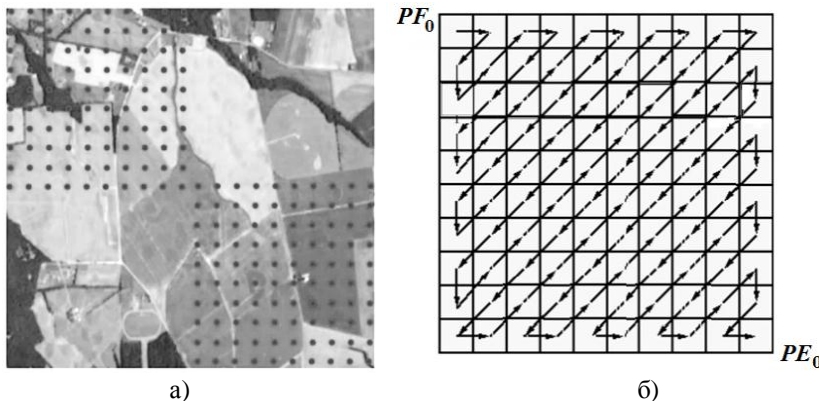


Рис. 2.21. Координатні сітки на мапі (а) та маршрут обстеження її одним БПЛА (б)

Для прискорення обстеження мапи, вкритої декількома координатними сітками (див. рис. 2.21а), доцільно використовувати зграю боїдів, поділену на декілька суб-роїв N , з кількістю M_n боїдів у n -му суб-рої. Для кожного суб-рою складається окрема польотна мапа в межах однієї координатної сітки, що відповідає одному вильоту зграї. Пропонується для роботи кожного n -го суб-рою формувати просторовий коридор SC_n довільної форми.

Алгоритм має розуміти варіанти різного положення точок вильоту безпілотних літальних апаратів: виліт всіх суб-роїв з однієї спільної точки PF_0 (рис. 2.22а), або з різних точок вильоту $PF_1 \dots PF_N$ (рис. 2.22б).

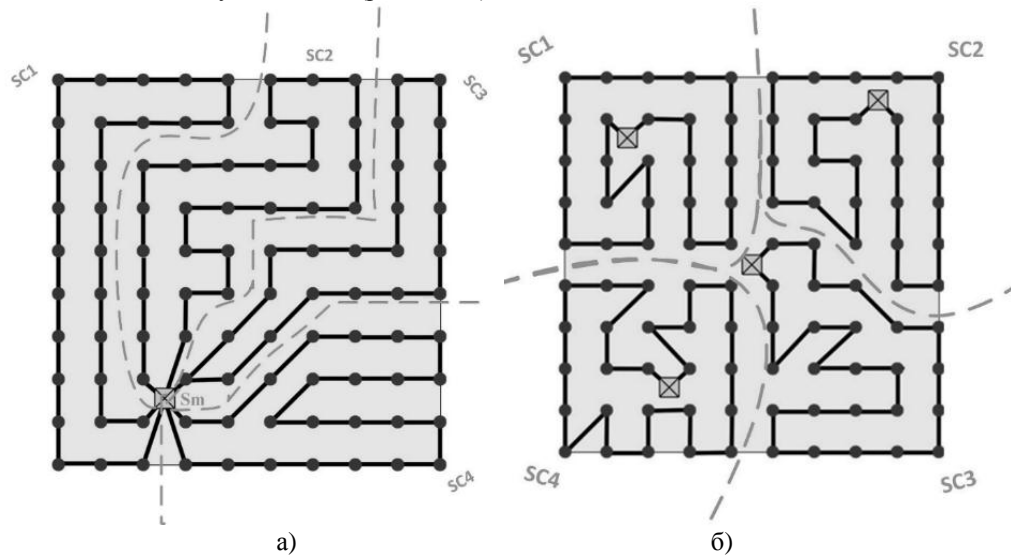


Рис. 2.22. Розподіл координатної сітки польотів на просторові коридори SC_n при $N = 4$: а – при спільній точці PF_0 вильоту боїдів (B_{ij}); б – при рознесених точках вильоту PF_n

У залежності від кількості та положення точок вильоту кожного суб-рою стає кожна координатна сітка розділяється на просторові коридори SC_n (див. рис. 2.22). У наведеному прикладі кількість таких коридорів становить $N = 4$. В кожному коридорі працює один суб-рій з кількістю боїдів один або більше (максимальна кількість боїдів у зграї M). Кожен вузол координатної сітки входить тільки в один просторовий коридор.

Тоді площа території, яка буде обстежена за один виліт зграї боїдів, залежить від відстані L між кожними двома сусідніми вузлами та кількості вузлів K :

$$S = (K - 1)^2 \cdot L^2. \quad (2.13)$$

Ідея запропонованого методу полягає в тому, щоб використовувати польотну мапу також у якості нейронної карти як динамічне представлення заданого простору, інформація про який надходить із зовнішніх джерел. Енергетичні взаємодії нейронів в мережі подібно поширенню хвилі збурень призводять до формування так званого ландшафту активації, який використовується в подальшому як навігаційна мапа для планування траєкторії. Такий підхід забезпечить виключення дубляжу дій боїда у вже обстеженому вузлу. Навіть, якщо боїд B_{mm} потрапить не в свій просторовий коридор SC_n (наприклад, буде знесений вітром), він не буде виконувати задачу у вузлах стороннього просторового коридора.

«Нейрон» у цьому випадку бачимо як обчислювальний процес, якому привласнений певний стан таким чином, що мережа може працювати паралельно.

Дискретна повнозв'язна НМХ складається з одного шару бінарних нейронів, кожен з яких пов'язаний з рештою нейронів і має мережеві вхід і вихід (рис. 2.23а). Координати цілі (вузли координатної сітки), а також інформація, чи є ці вузли вже обстеженими, надходять на вхід НМХ (рис. 2.23б), яка являє собою шар адаптивних суматорів із зворотними зв'язками, вихідні сигнали яких надходять з часовою затримкою Δt_{ij} на входи нейронів, у результаті чого вихідний сигнал НМХ формується лише після того, як мережа досягне динамічної рівноваги [300].

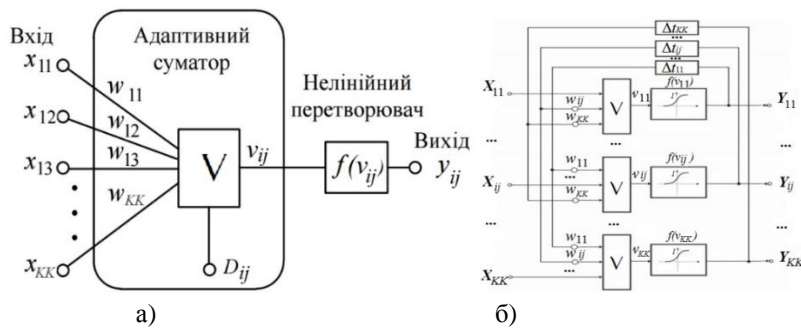


Рис. 3.23. Графічне зображення нейрону (а) та НМХ (б)

Математично подібний нейрон (рис. 2.23а) описується за допомогою системи рівнянь (2.14):

$$\left\{ \begin{array}{l} y_{ij} = f(v_{ij}); v_{ij} = \sum_{i=1}^{k,k} x_{ij} \cdot w_{ij} + D_{ij}; \\ D_{ij} = 1, P_{ij} \notin SC_n; D_{ij} = 0, P_{ij} \in SC_n; \\ w_{ij} = 1, \text{ якщо довжина маршруту між вузлами } L; \\ w_{ij} = 1,4, \text{ якщо довжина маршруту між вузлами } \sqrt{L^2}. \end{array} \right. \quad (2.14)$$

Початковий бінарний рівень нейрону D_{ij} задається оператором КФС та дорівнює «1», якщо нейрон (вузол польотної мапи) не входить до мапи суб-рою; дорівнює «0», якщо вузол підлягає обстеженню боїдом даного суб-рою.

Штучна НМХ «навчається» рішенням задачі диз'юнкції матриць стану вузлів польотної мапи (нейронів нейронної карти), що зводиться до приведення стану всіх нейронів до «1», коли всі вузли польотної мапи будуть обстежені. Тобто, у разі здійснення боїдом у вузлі P_{ij} вимірювання будь-якої характеристики, фотографування місцевості та ін., вихідна функція $f(v_{ij})$ обчислювального процесу на відповідному ij -му нейроні, встановлюється в «1» і передається за правилами НМХ на входи всіх нейронів суб-рою. Якщо інший боїд попадає в точку з такими координатами, обчислювальний процес в цьому боїді не активується, тому що характеристики цієї точки вже виміряні та це відображено у нейронній карті.

Добуток всіх зважених входів $x_{ij} \cdot w_{ij}$ наприкінці задачі має бути не нульовим, враховуючи, що в кожній цільовій точці подія відбулася.

Таким чином, виключається дублювання події в кожній точці обстежуваної території, економляться обмежені обчислювальні ресурси комп'ютерних компонентів боїдів, економиться електроживлення і відповідно збільшується час життя зграї.

Під час планування маршруту суб-рою на основі розподіленої НМХ кожен з боїдів суб-рою формує власну нейронну карту і виконує її динамічне коригування за початковими даними і за інформацією від датчиків на боїді. Сформовану нейронну карту боїд по каналу зв'язку передає всім доступним боїдам зграї. У процесі руху боїд отримує нові «знання» про проведені дослідження у вузлах координатної сітки і відповідно до них коригує нейронну карту.

Кількість таких координатних сіток на цільовій мапі відповідає кількості вильотів зграї боїдів для виконання поставленої задачі з обстеження дослідної поверхні.

На початку роботи алгоритму на основі польотної мапи створюється квадратна матриця подій K -го порядку, де K – кількість опорних точок в одній стороні побудованої квадратної координатної сітки, що покриває частину цільової території за топологією

рис. 2.22а, яка можлива для моніторингу протягом одного вильоту зграї боїдів. У перетину i -го рядка та j -го стовпця даного масиву записуються «0», якщо точка входить n -го з N просторових коридорів SC_n , та «1», якщо точка не входить до цього коридора (тобто, для цього суб-рою точка вважається вже обстеженою). Ця ж матриця A є нейронною картою НМХ (2.16).

$$A = \| \| a_{ij}^{(n)} \|_1^K = \left\| \begin{array}{cccc} a_{11}^{(1)} & \dots & a_{1j}^{(n)} & \dots & a_{1K}^{(n)} \\ \dots & \dots & \dots & \dots & \dots \\ a_{i1}^{(n)} & \dots & a_{ij}^{(n)} & \dots & a_{iK}^{(n)} \\ \dots & \dots & \dots & \dots & \dots \\ a_{K1}^{(n)} & \dots & a_{Kj}^{(n)} & \dots & a_{KK}^{(n)} \end{array} \right\|, \quad (2.16)$$

На другому етапі згідно з запропонованим методом для кожного n -го суб-рою складається окрема нейронна карта НМХ A_n за правилами (2.17).

$$\left\{ a_{ij}^{(n)} \mid n = \overline{1, N}; a_{ij}^{(n)} \mid i = \overline{1, K}; a_{ij}^{(n)} \mid j = \overline{1, K} \right\}; \quad (2.17)$$

$$a_{ij}^{(n)} = 0, \text{ якщо } a_{ij} \in SC_n; a_{ij}^{(n)} = 1, \text{ якщо } a_{ij} \notin SC_n.$$

Тобто, якщо $a_{ij}^{(n)} = 1$, то такий вузол для n -го суб-рою вважається вже обстеженим.

На третьому етапі виконується синтез маршрутів боїдів суб-рою в межах кожного n -го просторового коридора.

Розглянемо обмеження технічної реалізації такої топології в зграї БПЛА, що складається з суб-роїв. Після вимірювань та/або відеомоніторингу та/або виконання інших технологічних процесів стан даного вузла в матриці (тобто нейрона матриці Хопфілда) змінюється з «0» на «1» (ставиться ознака, що вузол обстежений). Такий стан нейронної мережі з виходу обстеженого вузла (фактично – з RF-передавача БПЛА) передається на входи всіх інших вузлів, тобто всім БПЛА, в яких би вони вузлах не знаходились на цей час. Таким чином, вхідні сигнали потрапляють на всі нейрони, а вихідними можуть бути сигнали як з усіх нейронів, так і з частини нейронів після декількох циклів функціонування. Тобто, нейронна карта НМХ поновлюється з деякими часовими затримками Δt_{ij} .

Робота алгоритму завершується, коли вже виконаний обліт всіх опорних точок кожного просторового коридору, а також відома довжина шляху кожного маршруту. Маршрут являє собою масив послідовності опорних точок, причому найперша і остання точка – це завжди буде точка вильоту боїда (суб-рою) зграї.

У нашому випадку, коли в зграї використовується 6 БПЛА, що працюють у складі 4 суб-роїв у 4 просторових коридорах $SC_1 - SC_4$, можна нарахувати 15 зв'язків у повнозв'язному площинному (2D) графі (рис. 2.24).

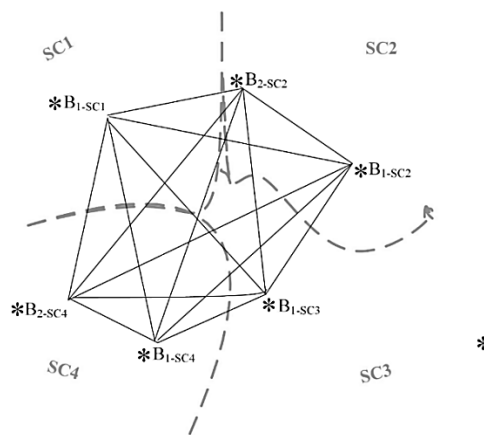


Рис. 2.24. Повнозв'язний граф для $M=6, N=4$

У загальному випадку кількість зв'язків у наведеній топології можна розрахувати за формулою $N(N-1)/2$, де N – загальна кількість боїдів у зграї. Це означає, якщо для обстеження частини території з 100 координатних вузлів випустити зграю з 100 боїдів, то при цьому потрібно забезпечити 4950 бездротових зв'язків. Навіть якщо врахувати необхідність використання трансферних вузлів для передачі даних на вхід всіх нейронів НМХ (окрім вузла-передавача) при відсутності прямої електромагнітної видимості між боїдами або при перевищенні можливої дальності сигналу [8], наявних обчислювальних та телекомунікаційних ресурсів на борту боїдів розглянутої мережі буде недостатньо.

Тому доцільно вважати, що боїди обмінюються картою НМХ з деякою часовою затримкою, коли відстань між ними не перевищує можливу для телекомунікаційного обміну.

Наприклад, для координатної сітки розміром 10×10 точок, яка поділена на 4 просторові коридори за правилами, наведеними на рис. 2.22а, матриці подій A_n для просторових коридорів SC_1 та SC_2 виглядатимуть як наведено у (2.18).

$$A_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}; A_2 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}. \quad (2.18)$$

Для інших коридорів SC_n матриця A_n будується аналогічно.

Матриця A повної координатної сітки має розмір $K \times K$ й отримується шляхом поелементної диз'юнкції матриць $A_1 \dots A_4$ (2.19):

$$A = A_1 \vee A_2 \vee A_3 \vee A_4. \quad (2.19)$$

Наприклад, після виконання завдань у просторових коридорах SC_1 та SC_2 відповідними суб-роями матриця A , обчислена за формулою (2.19), набуває вигляду (2.20).

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (2.20)$$

З матриці у (2.20) та рис. 2.24 видно, що просторові коридори SC_3 та SC_4 ще не обстежені. Якщо у результаті критичного застосування боїди в зазначених коридорах втрачені, тоді оператором КФС мають бути перевизначені початкові змінні D_{ij} нейронам карт A_1 і A_2 НМХ, і боїди, які звільнились від виконаних завдань у просторових коридорах SC_1 та SC_2 , повинні бути направлені у просторові коридори SC_3 та SC_4 .

Алгоритм закінчується, коли всі значення матриці A повністю обстеженої координатної сітки частини досліджуваної території стають одиничними.

Таким чином, поведінка кожного суб-рою присвячена вирішенню єдиної для такого суб-рою задачі – виконанню будь-якої технологічної функції та/або моніторингу характеристик території у призначеному цьому суб-рою просторовому коридорі чітко визначених розмірів та конфігурації, з обов'язковим вимірюванням характеристик (виконанням технологічних функцій) в кожному з наявних на координатній сітці вузлів.

Кожен з цих боїдів буде абсолютно автономною одиницею, яка зможе приймати свої власні рішення і виконувати дії. Кожен боїд зможе «спілкуватися» з рештою боїдів суб-рою, наслідуючи, таким чином, поведінки рою бджіл і виконуючи спільне завдання.

НМХ може бути корисною в цьому випадку для того, щоб жоден вузол не був обстежений двічі. Для цього треба скористатися властивістю мережі Хопфілда, що з виходу кожного з нейронів подається сигнал на входи всіх інших нейронів. Тоді факт виконання дії (вимірювання, фотографування тощо) у вузлі координатної сітки кожним боїдом стає відомим решті боїдам суб-рою, й такий вузол викреслюється з числа опорних точок, відповідний нейрон переходить в нульовий стан, і боїд не виконує повторної дії в такому вузлі.

Таким чином, глобальні рішення щодо обстеження вузлів координатної сітки приймаються «зграсю» на основі всіх наявних колективних даних. Завдяки цьому суб-рій є універсальним утворенням, що добре пристосовується до групової поведінки членів суб-рою (БПЛА) та є малочутливим до втрати певних його членів. Зграя, яка поділена на суб-рої, може втратити деяку кількість боїдів, але все ж залишиться здатною до виконання завдання з обстеження частини мапи, обмеженої координатною сіткою з поділом на просторові коридори.

Проведені експерименти підтвердили працездатність і практичну застосованість запропонованого методу. Результати планування маршрутів боїдів суб-роїв наведені у табл. 2.1 й табл. 2.2. У табл. 2.1 наведені результати застосування запропонованого методу з топологією мережі, де для кожного суб-рою власна окрема точка вильоту, за умови застосування боїдів зі швидкістю польоту 5 м/с та відстанню між вузлами координатної сітки $L = 100$ м.

Таблиця 2.1. Результати методу з різних точок вильоту БПЛА

№ суб-рою (просторового коридору)	№ точки вильоту БПЛА	К-сть БПЛА	К-сть вузлів маршруту (разом/розподіл між БПЛА)	Середній час виконання задачі суб-роєм, хв	Загальна вартість маршруту
SC ₁	PF ₁	1	25/25	8,9	268
SC ₂	PF ₂	2	25/13 і 12	4,0	268
SC ₃	PF ₃	1	25/25	8,5	256
SC ₄	PF ₄	2	25/13 і 12	3,9	252
РАЗОМ:		6	100	23,3	1044

У табл. 2.2 наведені аналогічні дані для БПЛА, які мають спільну точку вильоту.

Таблиця 2.2. Результати методу з однієї точки вильоту БПЛА

№ суб-рою (просторового коридору)	№ точки вильоту БПЛА	К-сть БПЛА	К-сть вузлів маршруту (разом/розподіл між БПЛА)	Середній час виконання задачі суб-роєм, хв	Загальна вартість маршруту
SC ₁	PF ₀	1	28/28	9,3	278
SC ₂		2	27/14 і 13	4,5	264
SC ₃		1	22/22	7,9	236
SC ₄		2	23/12 і 11	4,3	240
РАЗОМ:		6	100	26,0	1018

З аналізу табл. 2.1 – табл. 2.2 видно, що запуск суб-роїв з різних точок вильоту скорочує час обстеження території на 10,4 %. Але при цьому загальна вартість маршруту зростає на 2,6 %. Це означає, що в такому разі треба приділяти більше уваги питанням оптичної навігації, більш точному позиціонуванню БПЛА під час здійснення складного

маневрування між вузлами координатної сітки. Необхідно передбачити режим повернення БПЛА на початкову позицію.

Необхідно відзначити, що при такому підході пройдений суб-роєм шлях не буде мінімальним, і описане рішення є тільки суб-оптимальним для TSP. Але використання мережі Хопфілда дозволяє знаходити задовільні рішення досить швидко, без додаткового аналізу. Це робить можливим використовувати розроблений метод синтезу маршруту відокремленої групи безпілотних апаратів (з конкатенацією декількох 2D-рішень у 3D-просторі) у швидкодинамічних мережах, наприклад, таких, як зграя боїдів. Використання класичних методів, що дозволяють знаходити оптимальне рішення з використанням множинних ітерацій, може зажадати таких тимчасових і обчислювальних ресурсів, які приведуть до краху зграї боїдів при несвоєчасному отриманні корекції маршруту.

Як видно з табл. 2.1 – табл. 2.2, запропонований метод синтезу маршрутів зграї БПЛА з розбиття на суб-рої, що діють в двовимірних просторових коридорах, дозволяє істотно прискорити виконання завдань обстеження ділянок територій. Описаний підхід забезпечує рішення задачі комівояжера з відвідуванням усіх вузлів координатної сітки, що покриває мапу частини досліджуваної поверхні, а також високий рівень узагальнення даних цією моделлю.

Це дозволяє рекомендувати запропонований метод для використання на практиці.

Сформовані матриці подій дозволяють забезпечувати активність моніторингових або функціональних пристроїв БПЛА (відеокамери, метео-датчики, пристрої розпорошення речовин та ін.) тільки у власному просторовому 2D-коридорі, а також і пасивність цих же пристроїв при попаданні деяких БПЛА суб-рою в чужий для нього просторовий коридор. Такий підхід підвищує час життєздатності зграї шляхом виключення дублювання задач у вузлах та економії обмежених енерго-, обчислювальних і технологічних ресурсів БПЛА.

Збільшення кількості БПЛА в кожному суб-рої або ущільнення вузлів координатної сітки прискорює виконання завдання цим суб-роєм, проте веде до значного витрачання загальних ресурсів зграї і до підвищення небезпеки зіткнень БПЛА всередині зграї. Таким чином, вибір характеристик зграї і планування завдань повинні бути спрямовані на пошук оптимального співвідношення K^2/N , що забезпечує прийнятний рівень енергоспоживання, матеріальних і часових витрат на виконання завдання на території.

Висновки до розділу 2

Основним висновком є те, що не існує чітко визначеної, сталої технології для реалізації процесу управління групою БПЛА.

Сучасні апаратні платформи оснащені достатньою кількістю модулів для вирішення задач обходу перешкод та підтримки польоту. Саме тому доцільно використати ці переваги для здешевлення процесу розробки рухомих моніторингових мереж.

Виходячи з цього, гостро постає питання розробки альтернативного підходу вирішення задач керування моніторинговою мережею.

Вирішено задачі планування маршруту для групи пристроїв, розподілу БПЛА на групи відповідно до ЦЗ, зокрема запропоновано алгоритм такого розподілу.

Досліджено та доведено, що нейронні мережі є дійовим інструментом для ідентифікації системи з декількох БПЛА, а також для прогнозування поведінки такої системи у часі та на географічному просторі.

Вирішується проблема синтезу маршрутів суб-роїв на основі нейронної мережі Хопфілда. Запропоновано метод побудови моделей часткових роїв («суб-роїв»), що складаються з БПЛА. Це дає змогу уникнути дублювання виконання завдань у будь-яких точках сітки, яка охоплює досліджувану територію. Крім того, запропоновано використовувати «самозагоєння» в межах суб-роїв у зграях птахоподібних об'єктів («боїдів») та перевизначення завдань підгруп як КФС у разі втрати кількох боїдів під час критичного застосування.

Такий підхід збільшує економію обмежених обчислювальних та енергетичних ресурсів, може збільшити термін служби боїдів у зграї. Проведено експерименти із запропонованими методами в задачах з урахуванням розміру зони обстеження, який вимагає певної кількості накладених на карту місцевості «сіток». У результаті цих експериментів виявлено залежність між розміщенням стартових точок суб-роїв та загальною вартістю маршрутів. Використання зазначених результатів дає змогу визначити оптимальне співвідношення між наявними обчислювальними та енергетичними ресурсами та часом виконання завдань в залежності від кількості суб-роїв зграї.

У цьому розділі також представлений процес моделювання руху групи роботів, коли всі роботи діляться на декілька просторових коридорів. Кожен суб-рій працює у власному просторовому коридорі, але всі роботи виконують аналогічні завдання.

Класична TSP вимагає знайти рішення в 2D-просторі, але робот може рухатися в трьох напрямках. Щоб правильно визначити TSP як 3D-завдання, запропоновано сприймати роботу як сферу замість точки.

Беручи до уваги той факт, що і роботи доставки, і роботи-будівельники виконують завдання тільки у вертикальних площинах коридора, пропонується окремо вирішувати задачу TSP для лівих та правих вертикальних площин з наступною конкатенацією двох 2D-рішень TSP у загальне рішення.

Запропонована імітаційна модель поведінки рою роботів описує його як динамічну інтелектуальну систему. У розділі 2 представлений процес моделювання поведінки двох роботів, які є загальним механізмом (тобто, рухаються синхронно, але виконуються різні завдання).

Необхідно зауважити, що описані алгоритми не можуть гарантувати пошук оптимального маршруту. Їх метою є скорочення часу пошуку прийнятного рішення цієї проблеми. Тобто, ці алгоритми дозволяють знайти суб-оптимальне рішення, яке можливо реалізувати в межах часу функціонування зграї.

Використання хмарних технологій, тривимірних мап місцевості та методології нейронних мереж Хопфілда дозволяють планувати безаварійні маршрути як для всієї зграї, так і кожного об'єкта кожного суб-рою.

Розділ 3

МЕТОДИ ТА АЛГОРИТМИ ВЗАЄМОДІЇ МІЖ ОБ'ЄКТАМИ РУХОМОЇ МОНІТОРИНГОВОЇ МЕРЕЖІ

3.1 Забезпечення якісного зв'язку у кіберфізичній системі з рухомою моніторинговою мережею

Питання мережевої взаємодії кіберфізичних об'єктів (КФО) та сумісності кіберфізичних систем (КФС) з глобальним Інтернетом є актуальним напрямком дослідження [104; 115]. У таких системах керування кількома фізичними (автономними або контрольованими) рухомими пристроями, здійснюється різними людьми – кібероператорами (КО, англ. Cyber Operator або СО) через різні обчислювальні пристрої. При цьому топологія такої гетерогенної мережі постійно змінюється. Сьогодні найбільш недостатньо досліджені швидкодинамічні КФС, у яких КФО взаємодіють через протоколи бездротового зв'язку.

Mesh-топологія КФС, яка складається з рухомих об'єктів, що з'єднані бездротовими каналами зв'язку, передбачає неперіодичні зміни з'єднань між різними парами кіберфізичних компонентів. Такими парами у зазначеній КФС можуть бути:

- перший фізичний компонент (повітряний, наземний, підводний або надводний БПА, оснащений датчиками фізичних процесів) – другий фізичний компонент;
- фізичний компонент – КО;
- КО – КО;
- перший КО – бездротова ТД – другий КО;
- КО – бездротова ТД – телекомунікаційне обладнання – наземний ЦОД.

На рис. 3.1 показана ситуація, коли КФО – вузли топології мережі – переміщуються за траєкторією, ініціюють передачу даних та виконують роль трансферних вузлів для передачі інформації, отриманої від інших вузлів, які за низкою причин не мають можливості її передати з власних датчиків до КО або ЦОД.

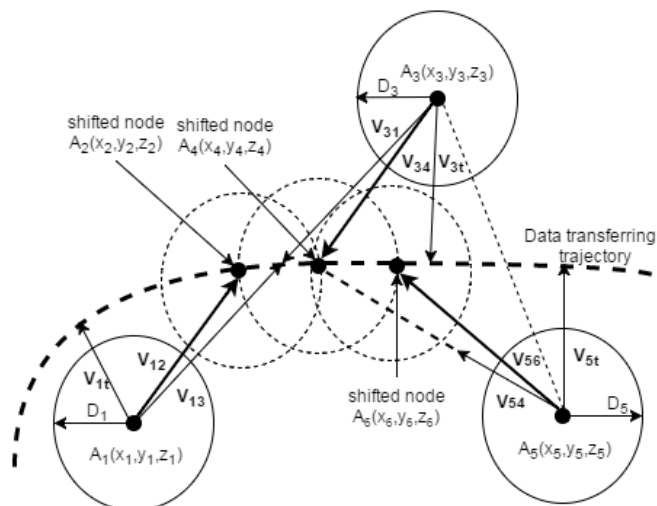


Рис. 3.1. Траєкторія передачі та трансферу даних з рухомими КФО [125]

Таким чином, наприклад, вузол $A_2(x_2, x_{y2}, z_2)$ рухається та передає інформацію.

Найбільш багатоваріантними й завадовразливими, але такими, що дозволяють попереднє налагодження перед сеансом зв'язку, є напівдуплексні бездротові з'єднання [44].

Найчастіше у багаточисельних робочих (неголовних) БПА використовують плати, в яких немає на борту модуля Wi-Fi. Тому, щоб оптимізувати взаємодію між об'єктами КФС, необхідно детально розглянути питання вибору обчислювальної плати. Додаткові модулі мають бути встановлені таким чином, щоб створити канал зв'язку між неголовним та головним БПА та / або КО.

Водночас, радіомодуль мобільного пристрою, з яким працює КО, є вбудованим та не підлягає заміні в польових умовах. Таким пристроєм може бути смартфон, нетбук, ноутбук, пульт дистанційного керування й т. п. Тому потрібно обирати вже не саму плату, а сумісні моделі пристроїв.

Необхідно розглянути взаємодію між такими парами бездротових пристроїв, які забезпечують телекомунікацію між БПА/БПА або БПА/КО, який керує БПА за допомогою будь-якого мобільного пристрою:

1) WiFi-адаптери БПА та мобільного пристрою, які працюють у режимі Ad-hoc («точка-точка»).

2) WiFi-адаптер БПА (або WiFi-адаптер мобільного пристрою) та ТД, яка працює у режимі Infrastructure («Клієнт-Сервер»).

Необхідно також акцентувати увагу на такому очевидному нюансі, що під час організації взаємодії в будь-якій бездротовій КФС, принцип «третьої зайвий» набуває суттєвого значення. Таким «третьим зайвим» є не тільки бездротові мережі через WiFi-канали, що перетинаються, але також звичні радіо-пристрої: USB-адаптер для радіомишки, бездротова клавіатура, Bluetooth-гарнітура та ін. Всі вони можуть створити значні перешкоди взаємодії об'єктів КФС.

Розглянемо взаємодію WiFi-адаптерів: БПА та мобільного пристрою.

Найбільша кількість КФС на основі Wi-Fi використовують модифікації стандарту 802.11b/g/n. Природним бажанням власника КФС є використовувати найбільш новітній стандарт серед всіх доступних,

Це бажання впливає з того, що нові стандарти підтримують найвищі швидкості і повинні бути зворотньо сумісними з попередніми версіями. Можна досить легко змінити бездротовий адаптер на комп'ютері або ноутбуці, але модернізація інших пристроїв Wi-Fi (наприклад, смартфонів, планшетів і БПА) не така проста.

Для забезпечення стабільної швидкості передачі даних є більш доцільним дотримуватися однієї версії стандарту і для ТД, і для мобільних об'єктів, які взаємодіють один з одним через Wi-Fi.

Безумовно, актуальними на сьогодні є версії стандарту 802.11n та 802.11ac. Тим не менш, з аналізу використовуваних пристроїв встановлено, що навіть в однакових моделях мобільних пристроїв з однаковими апаратними модулями, дуже часто сумісність досягається лише на відпрацьованому стандарті 802.11b/g.

У стандарті 802.11b/g не передбачений повнодуплексний режим роботи [193]. Тому, у разі організації напівдуплексної взаємодії між об'єктами КФС, необхідно приділити увагу як вибору моделей бездротових адаптерів, так і налагоджуванню режимів роботи цих адаптерів на всіх об'єктах КФС.

На сьогодні актуальною відстанню передачі сигналу у КФС є відстань в 50, 100, 250 метрів. Для того, щоб забезпечити такий вибір, достатньо стандартного рівня сигналу до 20 дБм. Такий рівень сигналу відповідає рівню потужності 100 мВт і є типовим для більшості комерційних WiFi-адаптерів [34; 100].

Для забезпечення більшої дальності (до 2 км за відсутністю перешкод), необхідно встановити додаткові потужні WiFi-антени різних конфігурацій [59] і використовувати багато потокову технологію (MIMO). Встановлення додаткових антен не збільшить

потужність передавача WiFi-модуля. Тим не менш, така модернізація може значно поліпшити діаграму направленості модуля. Таким чином, відстань взаємодії кіберфізичних компонентів збільшиться.

Коли ми говоримо про взаємодію БПА і мобільних пристроїв, роль мережевої ТД виконує WiFi-адаптер БПА. Керуючий пристрій (смартфон, ноутбук, пульт дистанційного керування й т. п.) підключається до такої ТД [176; 177]. Іноді необхідно передавати ширококомовний трафік з БПА кільком кіберфізичним компонентам. У цьому випадку, кібероператор має підключитися до ТД, в якості якої виступає БПА, в режимі моста. Тоді стає можливим подальше поширення мережі, яка створена з БПА, до інших кіберфізичних компонентів [177].

Розглянемо взаємодію пари WiFi-адаптер БПА (або мобільного пристрою) – ТД.

Режим автовибору найбільш вільного каналу (Clear Channel Select) встановлюється за замовченням у заводських налаштуваннях більшості моделей WiFi-обладнання. Однак, автоматична зміна каналу може привести до обриву комунікації та до втрати відео у режимі реального часу від БПА в умовах швидко динамічної ситуації у зоні дії КФС. Тому доцільно у налаштуваннях бездротового обладнання виконати деактивацію режиму автоматичного вибору каналу та вручну встановити номер WiFi-каналу, визначений в результаті випробувань.

Зміна каналу бездротового зв'язку може допомогти збільшити діапазон і ефективність взаємодії об'єктів КФС, особливо якщо є сусідні мережі, які встановлені на тому ж каналі, або присутня інтерференція від інших пристроїв. Найкраще було б перевірити канали і рекомендувати конкретний за допомогою таких програм-аналізаторів, як Vistumbler або InSSIDer [178].

Доречно зберегти конфігурацію телекомунікаційних пристроїв (ТД, маршрутизатора тощо) у файл для багатоагентної КФС з однорідними об'єктами. Тоді можна передати налаштування з однієї ТД до інших у межах КФС. Це значно спрощує і прискорює налаштування бездротового з'єднання у випадку, якщо КФС буде розширюватися. Крім того, можна легко інтегрувати новий кіберфізичний компонент в існуючу mesh-топологію і врахувати зміну поведінки КФС у матмоделі.

Якщо забезпечити управління і налаштування кіберфізичних компонентів за допомогою пульта дистанційного управління або з використанням безпосереднього доступу до вбудованого веб-інтерфейсу, тоді їх зможе конфігурувати будь-який кібероператор, який має права адміністратора, з будь-яких мобільних об'єктів КФС, які працюють під керуванням iOS, Android або Windows Phone.

Доцільно вибрати бездротове обладнання, яке підтримує 802.1Q. Це дозволяє спільне використання фізичної мережі багатьма логічними (віртуальними) підмережами (VLAN) засобами RADIUS. Стандарт RADIUS (Remote Authentication Dial-In User Service) забезпечує стиковку паролів від різних систем автентифікації, дозволяє рухомих об'єктам, що залишаються членами тієї ж VLAN (підмережа безпілотних літальних апаратів – UAV, підмережа безпілотних роверів, підмережа інших БПА, підмережа кібероператорів та інші підмережі КФС), переміщуватись у межах зони дії КФС і переключатись від однієї ТД до іншої.

Для швидкого розгортання КФС довільної, а не тільки mesh-топології, у новій зоні дії та з більшою кількістю КФО (наприклад, моніторингової мережі UAV), корисним буде підтримка бездротовим телекомунікаційним обладнанням технології бездротових магістралей (WDS). Використання режиму WDS дозволить об'єднати бездротовими магістралями декілька ТД без втрат швидкості передачі даних. При цьому можливо або дозволити паралельне обслуговування бездротових клієнтів на проміжних ТД (у режимі «repeater»), або навпаки, заборонити доступ бездротових клієнтів (у режимі «bridge»).

У задачах передачі поодиноких дискретних сигналів від різних сенсорів з борту КФО, задля досягнення оптимального співвідношення ціна/якість, достатньо використання WiFi-модуля за стандартом 802.11g замість хоча й більш швидкісного, але також й більш енергоспоживаного стандарту 802.11n.

За необхідністю організації більш швидкісної лінії зв'язку матиме сенс перехід на стандарт 802.11ac (підтриманий багатьма виробниками бездротових чипсетів – Quantenna, Broadcom, Buffalo, Cisco тощо), який, крім високої швидкості передачі даних, характеризується також й зменшенням у 6 разів енергоспоживанням порівняно з 802.11n.

Для забезпечення стабільного трафіку в КФС необхідно:

1. Зрівняти між собою швидкості потоків «up» та «down» між КФО.
2. Оцінити зміни швидкості передачі даних від кількості ретрансляцій.

Значним кроком вперед в організації зв'язку з безпілотними КФО – БПА була розробка моделей, які дозволили користувачеві підключатися до відеокамери та сенсорів БПА засобами WiFi-мережі, використовуючи смартфон, планшет або аналогічний сумісний пристрій [104; 125]. Такий підхід представляє дві значні проблеми [58; 323]. Однією з них є вузький діапазон Wi-Fi та, як наслідок, необхідність точного налаштування на канал передачі даних у сеансі зв'язку. Другою проблемою є близькість інших частот радіоперешкод до частот управління БПА та в результаті інтерференція між декількома частотними піддіапазонами.

Експериментально доведено, що сусідні WiFi-мережі з рівнем сигналу 80 дБм спричиняють інтерференцію, яка призводить до втрати біля 25 % потужності сигналу вже на 25 м [107].

Під час цього виникають збої, обриви зв'язку, які потребують досліджень щодо стабілізації останнього та забезпечення його певної якості.

Тобто, після обриву сеансу повинна здійснюватися повторна ініціалізація блоку синхронізації приймача (*Timing Recovery*) та компонента, що відповідає за відокремлення блоку даних від преамбули пакету (*Preamble Detector*) – рис. 3.2. Компоненти у приймачі (*Receiver*), вразливі до дії перешкод, наведені курсивом [70; 242].

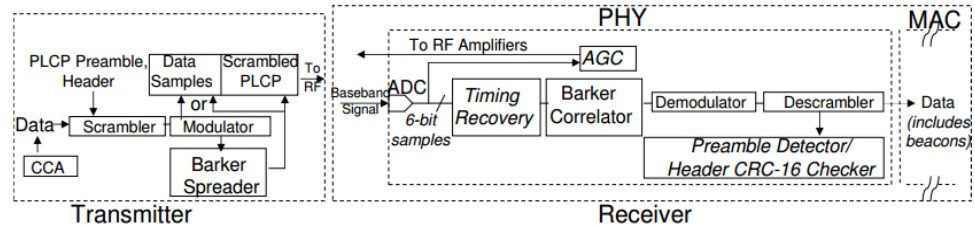


Рис. 3.2. Склад компонентів WiFi-модуля на борту об'єкта КФС

Одним з можливих рішень для виправлення помилок у залежності від рівня завад може бути використання адаптивної системи, в якій дані представляються у вигляді залишків від ділення на розширену систему модулів та передаються за методом стрибкоподібної зміни частоти [318]. Такий метод здатний забезпечити якісний WiFi-зв'язок, але потребує достатньо великих обчислювальних ресурсів, тому не завжди може бути застосованим у рухомих КФС з мобільними об'єктами.

Спробуємо запропонувати інший метод уникнення зниження швидкості через інтерференцію від нових ТД, які під'єднуються до радіоефіру, але не входять до складу КФС. У такому разі стабільність зв'язку досягається шляхом перевизначення радіоситуації в

зоні дії КФС і зміною характеристик кіберфізичних компонентів (комп'ютерних компонентів на борту КФО). У такий же спосіб можливо уникнути падіння швидкості передачі даних, коли рухомі об'єкти КФС входять у зони з високим рівнем електромагнітного випромінювання або оптична видимість між КФО втрачається.

Для ефективної життєдіяльності КФС необхідно досягнути максимальної симетрії каналу зв'язку між Клієнтом (Станція – STA) та Точкою доступу (AP). Це вимагає зрівняти між собою швидкості потоків «upload» та «download». У результаті різниця DF між співвідношенням сигнал-шум (SNR) AP та Клієнта має прагнути до нуля, як показано у формулі (3.1):

$$DF = (SNR_{STA} - SNR_{AP}) \rightarrow 0. \quad (3.1)$$

Режим передачі даних з $DF < 0$ (асиметрія каналів зв'язку між кіберфізичними компонентами) є критичним для КФС. У такому випадку швидкість передачі даних від Клієнта до AP («upload») буде повільніше ніж у зворотньому напрямку («download»), або зв'язок буде зовсім втраченим. У результаті, кібероператор не спроможний буде керувати БПА з свого мобільного пристрою, оскільки дані телеметрії не опрацювуватимуться за адекватний час, і БПА може потерпіти крах.

Для досягнення симетрії у каналі зв'язку між кіберфізичними компонентами необхідно регулювати потужність AP таким чином, щоб $DF = 0$.

Розглянемо залежність між потужністю передатчика Rx та чутливістю приймача $RxSens$ відповідно для Клієнта (STA) та AP (3.2):

$$\begin{aligned} SNR_{AP} &= Rx_{STA} - RxSens_{AP}; \\ SNR_{STA} &= Rx_{AP} - RxSens_{STA}. \end{aligned} \quad (3.2)$$

Вважається, що поріг фоновому шуму нижче за порогову чутливість приймачів Клієнта і AP, тому що якщо рівень фоновому шуму для AP і Клієнта однаковий – це ніяк не впливає на симетрію каналу [145].

Тоді потужність Rx передавачів AP і Клієнта (STA) відповідно:

$$\begin{aligned} Rx_{AP} &= Tx_{AP} + TxGain_{AP} - L_{FS} + RxGain_{STA}; \\ Rx_{STA} &= Tx_{STA} + TxGain_{STA} - L_{FS} + RxGain_{AP}; \\ TxGain_{AP} &= RxGain_{AP} = Gain_{AP}; \\ TxGain_{STA} &= RxGain_{STA} = Gain_{STA}, \end{aligned} \quad (3.3)$$

де Tx_{AP} та Tx_{STA} – потужність передавача відповідно AP та Клієнта на порту антени; $TxGain_{AP}$, $RxGain_{AP}$ та $TxGain_{STA}$, $RxGain_{STA}$ – коефіцієнт підсилення антени під час передачі $TxGain$ та під час прийому $RxGain$ відповідно для AP та Клієнта; L_{FS} – втрати в каналі зв'язку AP – Клієнт (однакові в обох напрямках).

Враховуючи, що клієнтські пристрої, як правило, оснащені незмінюваними антенами, потужність передавача й чутливість приймача Клієнта вказується відразу з урахуванням антени. Тоді асиметрія каналу становить:

$$DF = (Tx_{AP} - Tx_{STA}) - (RxSens_{STA} - RxSens_{AP}). \quad (3.4)$$

З рівнянь (3.1) – (3.3) видно, що асиметрія каналу не залежить від типів антен на AP та Клієнті (якщо не використовується багатопотокова технологія MIMO), а залежить від різниці між потужностями передавачів й чутливостями приймачів КФО [97; 235; 237; 311; 313].

Залежно від рівня падіння потужності через перешкоди підвищується частка ретрансляції даних, які передаються (рис. 3.3). При цьому час затримки у передачі даних зростає аж до повної втрати сигналу.

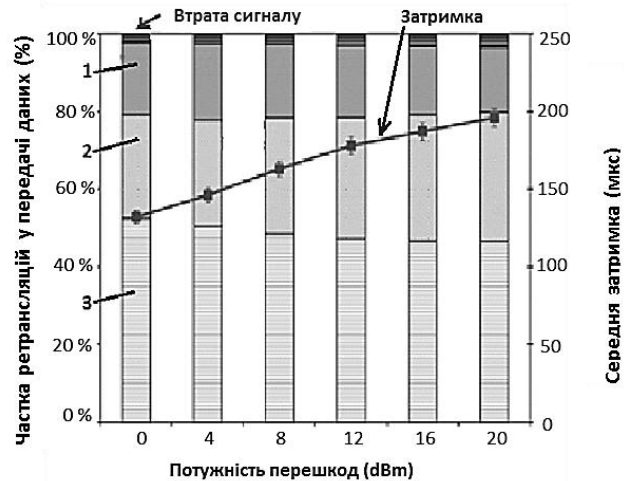


Рис. 3.3. Зростання затримки у часі передачі даних від кількості ретрансляцій:

- 1 – багаторазова ретрансляція;
- 2 – одноразова ретрансляція;
- 3 – без ретрансляцій

За проведеним експериментом у міській місцевості швидкість сигналу, який передається, зменшується зі збільшенням перешкод (дерева, рухомі перешкоди – живі істоти, авто- та ін.) та відстанню між Адресатом та Відправником інформації.

Для розширення дальності дії WiFi-мережі до 366 м можна при конструюванні БПА використати модуль Espressif ESP8266.

У реальних міських умовах зашумленого радіоефіру швидкість передачі даних майже на 30 % менша за теоретичну (лінія 1 на рис. 3.4). Крім того, наявність електромагнітної перешкоди (лінія 2 на рис. 3.4) у вигляді трамвайної лінії електропередачі (600 В постійного струму) негативним чином впливає на пропускну здатність WiFi-каналу.

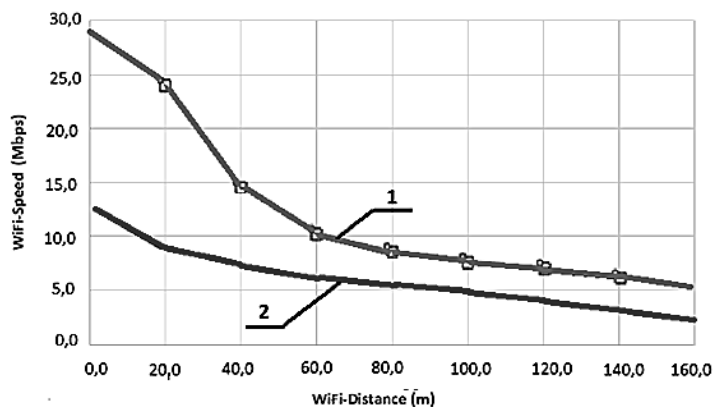


Рис. 3.4. Падіння швидкості передачі даних (WiFi-Speed) від збільшення перешкод за відстанню (WiFi-Distance):

- 1 – без додаткових електромагнітних перешкод;
- 2 – з додатковими електромагнітними перешкодами

Оскільки йдеться про малогабаритні швидкодинамічні сенсорні мережі, то можливість встановлення додаткових антен для збільшення дальності передачі даних не розглядається.

Доцільно розглянути налаштування режимів роботи комп'ютерних компонентів КФО, сукупність технічних характеристик котрих забезпечить життєздатність такої топології КФС, що склалась під час руху об'єктів сенсорної мережі. В першу чергу, це безпілотні КФО (БПА), характеристики яких не можуть бути налаштовані вручну.

Для БПА сигнал потужності передавача стандартних WiFi-пристроїв (не більше 20 дБм/100 мВт) забезпечує відстань активності обладнання до 200 м без додаткового підсилення. Але часто рівень сигналу від WiFi-модуля відправника недостатньо сильний (знижується за причини перешкод), тому на лінії зв'язку між КФО можуть виникати обриви зв'язку.

У рухомих КФО на борту знаходяться компоненти, які забезпечують WiFi-зв'язок у визначеному діапазоні частот на певну відстань.

Одними з найбільш вживаними у сучасних мобільних пристроях (а також планшетах, ноутбуках тощо) є комп'ютерні компоненти від фірми Broadcom Corp. Для забезпечення WiFi-зв'язку використовується модуль BCM4330 (2,4/5 ГГц). Підтримка стандарту 802.11n в однопотоковому режимі на каналі 20 МГц забезпечує швидкість передачі PHY шару до MCS7 (72 Мбіт/с). Такий модуль має хоча й обмежені, але достатні для виконання автономного та керованого завдання обчислювальні ресурси. Це вбудований процесор ARM® Cortex-M3, який здатний розрахувати стандартні функції WLAN [10].

BCM4330 включає в себе інтегрований дводіапазонний бездротовий приймач ВЧ, який був оптимізований для використання у діапазонах 2,4 і 5 ГГц бездротової локальної мережі (але не обидва одночасно).

На різних КФО також можуть бути встановлені конкурентні WiFi-модулі RL-UM12BS-8188EUS від вендора NATER, Ralink RT3290 від MediaTek, Atheros AR9331 від Qualcomm тощо.

Характеристики зазначених модулів наведені у табл. 3.1 [34; 54; 145].

Усі вищезазначені модулі дозволяють двом і більш WiFi-пристроїв спілкуватися один з одним без маршрутизаторів і хот-спотів у режимі Wi-Fi Direct (раніше відомий як Wi-Fi Peer-to-Peer).

З аналізу табл. 3.1 видно, що не всі з модулів повноцінно підтримують стандарт 802.11n (ESP8266, BCM4330), а лише забезпечують сумісність з іншим обладнанням цього стандарту.

До того ж, у каналі радіозв'язку між БПА та стаціонарним наземним пунктом керування мають місце явища перевіддзеркалювання, розсіювання і дифракції, які разом утворюють загальну картину замирань у каналі [356].

Для організації процесу ретрансляції необхідно по два бездротових мережевих інтерфейси на кожному КФО. Тоді у кожному випадку необхідної ретрансляції один бездротовий інтерфейс працює у режимі Отримувача даних від Клієнта, а другий – у якості Відправника даних Отримувачу (у разі наявності одного ретранслятора у mesh-топології) або іншому ретранслятору, якщо їх декілька на шляху між Відправником та Отримувачем.

Таблиця 3.1. Характеристики WiFi-модулів різних виробників

Назва мікросхеми з вбудованим WiFi-модулем	Наявність T/R	Потужність передавача (Т), дБм (мВт)	Чутливість приймача (R), дБі	Стандарт передачі даних (частота)	Максимальна швидкість передачі даних, Мбіт/с
Broadcom BCM4330	1T1R	18/	-93/-76/-62	b/g/n	72
Nater RL-UM12BS-8188EUS (Realtek RTL8188EUS)	1T1R	18/14/13	-86/-73/-66	b/g/n 13 каналів	11/54/150
Realtek RTL8192CU	2T2R Multiple BSSID	Не визначена	Не визначена	b/g/n	144,4/300 (полоса 20/40 МГц)
Ralink RT5350F, RT3050F	1T/1R	17/14/11 (50/25/12,5)	-83/-70/-64	b/g/n (2,412 – 2,462 ГГц) 11 каналів	11/54/130
Ralink RT3052	2T2R	Не визначена	Не визначена	b/g/n	300
Qualcomm Atheros AR9331	1T1R	Не визначена	Не визначена	b/g/n	72/150 (20/40 МГц)
Espressif ESP8266	1T1R	25/...	Не визначена	b/g/n	10

Варто зазначити, що в разі використання в якості WiFi-ретранслятора смартфона з ОС Android, остання має бути не нижче версії Android 4+.

Крім того, необхідно враховувати, що на підтримку 2 лінків потрібні подвійні витрати електроенергії – тобто, час життєвого циклу БПА теж пропорційно скоротиться. Таким чином, ретранслятор погіршує енергетику лінків, для яких є пряма видимість, і призначений виключно для вирішення питання прямої видимості.

Якщо у топології сенсорної мережі роль КФО виконує кібероператор з ноутбуком, на якому, як правило, вбудовано один WiFi-модуль, то для того, щоб додати до топології ретранслятор, необхідно додатково встановити WiFi-адаптер на USB-порт.

На борту БПА в такому разі доцільно встановлювати компоненти, до яких вбудовано відразу по 2 передавача та приймача (2T2R), нп., на базі мікросхеми Realtek RTL8192CU або Ralink (MediaTek) RT3052 (табл. 3.1). Використання зазначених компонентів у обладнанні наземного КФО дозволить останньому виступати в якості ТД відразу для декількох SSID – від 8 до 16 разом на 256 Клієнтів [34; 145].

Проаналізуємо інтерференцію від сусідніх WiFi-мереж і вибір номера каналу зв'язку.

Було проведено дослідження динаміки взаємодії двох найбільш поширених WiFi-адаптерів Ralink RT3290 802.11b/g/n [145], вбудованих в ноутбуки моделі ASUS X501U під управлінням ОС Windows 8.1 SL. У цьому режимі один з адаптерів працює в якості ТД.

Проблема полягає в тому, що у випадку, коли на ТД встановлений режим роботи «тільки 802.11n», інший мережевий WiFi-адаптер Ralink RT3290 «не побачить» цю ТД, яка була створена навіть на аналогічному адаптері та діє у межах зони покриття.

Було відзначено, що рішення обмежити швидкість передачі даних у бездротовій КФС на рівні нижче, ніж максимум у режимі «тільки 802.11n», нераціонально тільки на перший погляд. Тим не менш, це виправдано після розгляду наступних аргументів.

Отже, найвищий досяжний результат використаних в експерименті WiFi-адаптерів Ralink RT3290 знаходиться на рівні 80–90 Мбіт/с, незважаючи на те, що максимальна швидкість з'єднання може досягати до 150 Мбіт/с. У той же час, в режимі 802.11n, ТД

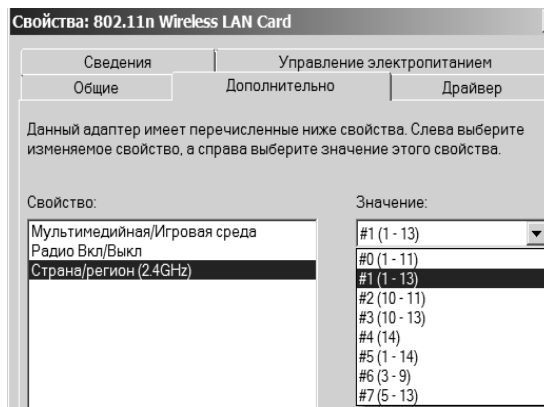
використовує дві частотних смуги, це означає, що споживана потужність буде більшою. Ураховуючи, що ТД створюється на БПА, це призводить до невиправданої перевитрати життя батареї.

Щоб зменшити споживання енергії, доцільно обмежити швидкість взаємодії об'єктів КФС єдиним рівнем стандарту, а саме 802.11g. Якщо ж встановити режим експлуатації «802.11b/g/n змішаний» – зв'язок буде встановлений все одно, як у режимі «тільки 802.11g», з притаманною цьому протоколу швидкістю до 54 Мбіт/с (насправді 25 Мбіт/с). Але, такого рівня швидкості передачі даних достатньо, щоб забезпечити взаємодію КФО, навіть для реалізації режиму FPV (First Person View) на UAV.

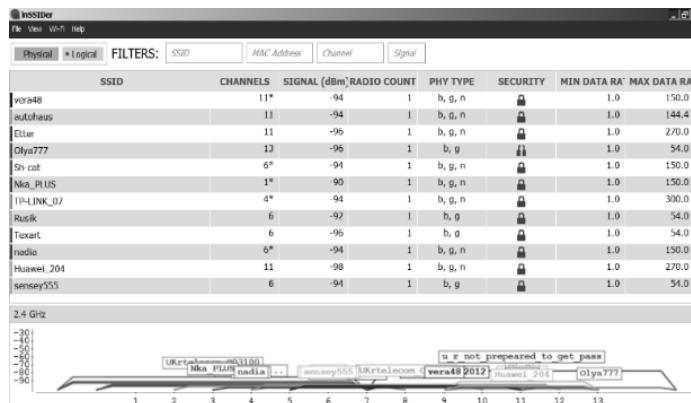
Оновлення драйверів [145] адаптера не вирішує цю проблему.

Робота на частоті 5 ГГц не розглядалась, оскільки цей діапазон є більш вимогливим до наявності оптичної прямої видимості. Якщо поставити завдання скорочення часу життя мережевих пакетів (параметру TTL передачі даних) у КФС або використання менш енергоємних (отже, легших) батарей, необхідно буде розглянути для такого завдання використання обладнання, яке працює за стандартом IEEE 802.11ac. У цьому випадку кіберфізичні об'єкти будуть взаємодіяти з більш високою швидкістю передачі даних (з досяжним експериментальним результатом у 290 Мбіт/с) на частоті 5 ГГц [1].

Таким чином, в налаштуваннях WiFi-адаптера Ralink RT3290 802.11b/g/n попередньо були обрані канали # 1–13, які відповідають смузі частот 2,4 ГГц (рис. 3.5а).



а)



б)

Рис. 3.5. Вибір каналів (а) та шумове забруднення радіосистеми на каналах # 1–13 (б) у смузі 2,4 ГГц

За допомогою програми InSSIDer 4 [178] було проаналізовано радіоситуацію на обраних каналах (див. рис. 3.5), виконано тестування якості зв'язку в межах цільової мережі Wi-Fi, яка має SSID «*u_r_not_prepared*» (рис. 3.6).

```
C:\>ping 192.168.0.22

Pinging 192.168.0.22 with 32 bytes of data:
Reply from 192.168.0.22: bytes=32 time=1849ms TTL=128
Reply from 192.168.0.22: bytes=32 time=1ms TTL=128
Reply from 192.168.0.22: bytes=32 time=1849ms TTL=128
Reply from 192.168.0.22: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.0.22:
    Packets: Sent = 4, Received = 4, Lost = 0 <0% loss>,
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1849ms, Average = 925ms

C:\>
```

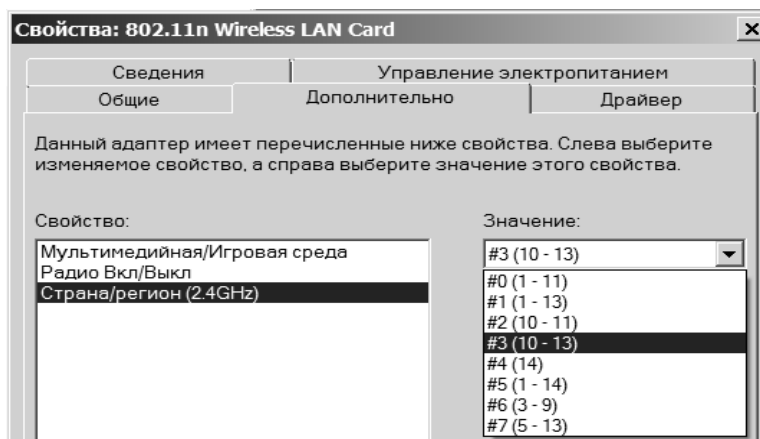
Рис. 3.6. Відповідь на луна-пакет на каналі # 1–13 у смузі 2,4 ГГц

Результат показав незадовільну характеристику каналу зв'язку, коли відгук луна-пакету становив близько 2 секунд. Така характеристика не дозволяє переглядати і обробляти переданий потік відеоданих в реальному часі.

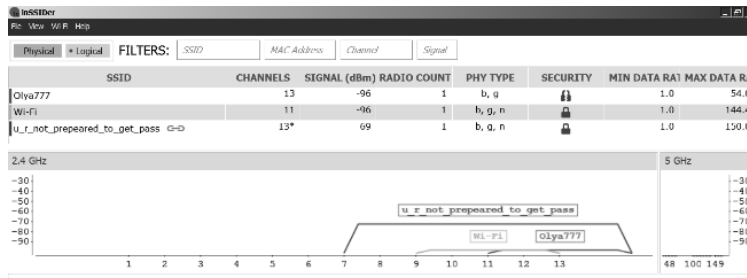
Для поліпшення якості зв'язку було прийнято рішення звузити смугу пропускання каналу зв'язку. Як правило, це роблять шляхом використання у діапазоні 2,4 ГГц каналів # 1, 6, 11, які не перетинаються.

Але, з каналів діапазону 2,4 ГГц, які не перетинаються, найбільш вживаним є канал # 6. Цей канал встановлюється за замовченням у більшості комутаційного WiFi-обладнання [23; 322]. На рис. 3.5 також показано, що наявні 5 мереж, в яких потужність сигналу наближена до рівня сигналу цільової мережі з SSID «*u_r_not_prepared*»; ці 5 мереж відображуються в частково наведеній моніторинговій області програми InSSIDer. Таким чином, конфігурувати об'єкти КФС на цьому каналі # 6 немає сенсу.

Зважаючи на те, що у більшості UAV модулі Wi-Fi зконфігуровані на граничні канали # 11–13 діапазону 2,4 ГГц [177], наступний експеримент з аналізу радіоситуації був виконаний з використанням програми WiFi-аналізатора InSSIDer 4 для ширини смуги пропускання, яка обмежується каналами # 10–13 (рис. 3.7).



а)



б)

Рис. 3.7. Вибір каналів # 10–13 (а) та робота цільової мережі «u_r_not_prepared» на переповнених каналах #10–13 (б) у смузі пропускання 2,4 ГГц

Було встановлено, що цільова мережа з SSID «u_r_not_prepared» (див. рис. 3.8) має найбільшу потужність (мінус 69 дБм) в обраному каналі # 13 у порівнянні з наявними всього двома менш потужними сусідніми мережами (мінус 96 дБм), які працюють на цьому каналі.

Однак, така позитивна радіоситуація і відсутність шуму в каналі зв'язку між об'єктами КФС незначно поліпшили якість зв'язку (рис. 3.8). Замість 2 секунд в попередньому експерименті, з новими налаштуваннями конфігурації кіберфізичних компонентів, час відгуку луна-паketу впав всього у 2 рази і склав близько 1 с. Такої швидкості зв'язку недостатньо для швидкодинамічних процесів і обробки відеопотоків у режимі реального часу.

```
C:\>ping 192.168.0.22

Pinging 192.168.0.22 with 32 bytes of data:
Reply from 192.168.0.22: bytes=32 time=984ms TTL=128
Reply from 192.168.0.22: bytes=32 time=1ms TTL=128
Reply from 192.168.0.22: bytes=32 time=987ms TTL=128
Reply from 192.168.0.22: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.0.22:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 987ms, Average = 493ms
```

Рис. 3.8. Відгук луна-паketу на каналах # 10–13 в смузі пропускання 2,4 ГГц

У такій ситуації цілком виправдано зробити висновок, що вибір стандарту зв'язку і ненавантаженого каналу не призведе до подальшого підвищення якості каналу зв'язку між бездротовими об'єктами КФС.

Для поліпшення якості каналу зв'язку дослідимо симетрію каналу зв'язку між парою ТД – Клієнт. Перевіримо попередні налаштування потужності сигналу передавача вуличної (англ. outdoor) ТД моделі D-Link DAP-3310 (рис. 3.9). Видно, що ТД має в експерименті максимальну потужність переданого сигналу (Transmit Power). Згідно з Data Sheet, 100 % вихідна потужність передавача DAP-3310 становить 20 дБм на швидкостях зв'язку 6–36, 48, 54 Мбіт/с. Чутливість приймача дорівнює мінус 74 ± 2 дБм на швидкості зв'язку 54 Мбіт/с [322]. WiFi-адаптер Ralink RT3290 802.11b/g/n [145], вбудований у ноутбук моделі ASUS X501U, має характеристики:

- вихідна потужність 802.11g: 14 дБм +/- 1,5 дБм;
- чутливість приймача 802.11g: менше ніж мінус 70 дБм (54 Мбіт/с).

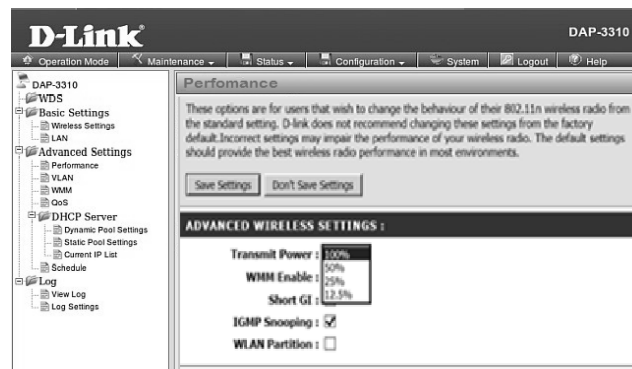


Рис. 3.9. Веб-інтерфейс налаштувань потужності сигналу на ТД моделі D-Link DAP-3310

Для встановлених потужностей сигналу ТД (100 мВт = 20 дБм) та Клієнта (18–35 мВт \approx 12,5–15,5 дБм) різниця потужностей становить мінус 4,5–7,5 дБ.

Розрахуємо максимальну асиметрію DF каналу зв'язку для ноутбука ASUS та ТД моделі D-Link DAP-3310 на 802.11g@54Мбіт/с:

$$DF = (12,5 - 20) - (70 - 72) = -7,5 + 2 = -5,5 \text{ дБ.} \quad (3.5)$$

Як показано у (3.5), при взаємодії об'єктів КФС можуть виникати проблеми, причому, з вини ТД.

Таким чином, для отримання стабільного зв'язку потужність сигналу ТД необхідно зменшити. Мінімальна асиметрія каналу зв'язку, якої можна досягнути, розраховується, як показано у (3.6):

$$DF = (12,5 - 50 \% \times 20) - (70 - 72) = -2,5 + 2 = -0,5 \text{ дБ.} \quad (3.6)$$

Перевіримо відгук («пінг») між досліджуваними кіберфізичними компонентами, коли потужність передавача виставлена у веб-інтерфейсі ТД моделі D-Link DAP-3310 на рівень 50 % (рис. 3.10). Можна побачити, що час відгуку луна-паketу знизився майже на порядок.

```
C:\>ping 192.168.0.22

Pinging 192.168.0.22 with 32 bytes of data:
Reply from 192.168.0.22: bytes=32 time=103ms TTL=128
Reply from 192.168.0.22: bytes=32 time=25ms TTL=128
Reply from 192.168.0.22: bytes=32 time=47ms TTL=128
Reply from 192.168.0.22: bytes=32 time=70ms TTL=128

Ping statistics for 192.168.0.22:
    Packets: Sent = 4, Received = 4, Lost = 0 <0% loss>,
    Approximate round trip times in milli-seconds:
        Minimum = 25ms, Maximum = 103ms, Average = 61ms

C:\>
```

Рис. 3.10. Відгук луна-паketу на потужності сигналу 50 % від ТД моделі DAP-3310

У результаті останнього експерименту якість зв'язку в бездротовому каналі значно поліпшується.

Визначимо залежність швидкості Wi-Fi від відстані між КФО.

Було досліджено декілька моделей бездротового обладнання SOHO- і SMB-сегменту для зовнішнього застосування, експлуатованих в режимі ТД (виробництва компаній TP-Link, Huawei-3Com і D-Link).

Теоретична швидкість, яка заявлена на рівні 54 Мбіт/с, в реальних умовах падає до 12 Мбіт/с на відстані 50 м (рис. 3.11, рядок 1). Так відбувається через досить складну

радіоситуацію в частоті 2,4 ГГц, тому що цей діапазон належить до неліцензійного ISM-діапазону. Також причиною зниження швидкості є інтерференція сигналів від інших мереж, які працюють на тому ж стандарті. Швидкість передачі даних може впасти до 1–2 Мбіт/с на тій же відстані для рухомих об'єктів [193]. Відповідно, швидкість 54 Мбіт/с може бути забезпечена при чутливості приймача мінус 74 дБм \pm 2. Якщо чутливість приймача становить мінус 90 дБм, швидкість падає до 2–5 Мбіт/с [242].

За інформацією компанії Cisco, передача даних за стандартом 802.11a/b/g може бути забезпечена з більш високим рівнем швидкості (рис. 3.11, рядок 3) на обладнанні Cisco Aironet Series 1130 AG [1]. Однак, ступенева зміна швидкості передачі даних у різних зонах КФС призведе до нерівномірного навантаження процесора обчислювальної плати і, як наслідок, до перегріву і можливого «зависання» пристрою кібероператора з подальшою втратою даних від фізичних компонентів (датчики, відеокамери й т. д.).

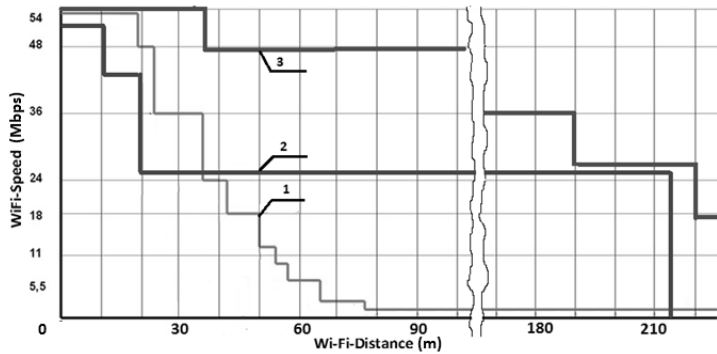


Рис. 3.11. Залежність швидкості від дистанції між КФО для передачі даних за стандартом 802.11g:

- 1 – падіння швидкості в реальних умовах значно нижче теоретичного рівня;
- 2 – залежність у вищенаведених умовах дослідження;
- 3 – залежність для обладнання фірми Cisco

Для UAV сигнал потужності передавача стандартних WiFi-пристроїв (не більше 20 дБм/100 мВт [59; 250]) забезпечує відстань активності обладнання до 200 м без додаткового підсилення. Для розширення зони взаємодії КФО до 400 м потрібен сигнал більшої потужності (відношення 25 дБм/316 мВт і більше). Однак, в Україні це вимагає дозволу від Національної комісії, що здійснює державне регулювання у сфері зв'язку та інформатизації (НКРЗІ) України.

Збільшення потужності передавача сигналу вище ніж обчислена раніше необхідно, щоб передати відеопотік через Wi-Fi в реальному часі в HD-якості на відстань 700 м. Такі можливості є, наприклад, у квадрокоптера DJI Phantom 2 Vision, котрий може бути використаний в якості головного дрону «зграї» UAV. Звичайна дистанція дії обладнання UAV не перевищує 50 м (наприклад, для квадрокоптера Parrot AR.Drone 2.0 Elite Edition Sand). Його приймач підтримує стандарт Wi-Fi 802.11b/g/n та передбачає управління з боку кібероператора за допомогою мобільного пристрою на основі iOS або Android з підтримкою Wi-Fi 802.11b/g/n.

Швидкість передачі даних в залежності від відстані у метрах показана на рис. 3.11, рядок 2. Ця залежність була отримана в результаті експерименту, описаного вище. Швидкість передачі даних стабілізувалася на значенні 24 Мбіт/с на всій площі КФС (до 200 м) для поточної топології бездротової мережі.

Область експерименту була обмежена за причини значних перешкод від лінії електропередачі (ЛЕП) високої напруги. Ці шуми викликали різке падіння швидкості

передачі даних і повне переривання каналу зв'язку між КФО, коли відстань між ними перевищила більш ніж 200 м. Таким чином, падіння лінії розриву 3 до нуля на рис. 3.11 не вказує на граничні можливості стандарту 802.11g, а зумовлена завадами зовнішніх електромагнітних полів.

Щоб відновити канал зв'язку, необхідно зробити нове дослідження навколишньої радіоситуації за допомогою програмного забезпечення (наприклад, InSSIDer) або апаратного бездротового аналізатора, і перевизначити параметри зв'язку між об'єктами КФС. Аналогічні нові дослідження будуть необхідні також, якщо нові ТД почнуть працювати на тому ж каналі зв'язку з порівняним рівнем потужності сигналу в безпосередній близькості до зони дії КФС.

Отже, перш ніж запускати до функціонування КФС на базі протоколів бездротового зв'язку, необхідно:

1) Провести вивчення радіозв'язку в цілому по області дії КФС за допомогою одного з аналізаторів бездротових мереж (програмного чи апаратного).

2) Встановити взаємодію об'єктів КФО на найбільш вільному від інших підмереж каналу (з найменшою кількістю SSID).

3) Налаштувати обмеження потужності передатчика WiFi-модуля відповідно до розрахованих параметрів асиметрії каналу на основі SNR кожної пари КФО.

4) Визначити характеристики відеопотоку та обрати тип матриці відеокамери, яка має бути встановлена на борту БПА.

У результаті проведеного дослідження взаємозалежностей характеристик та режимів роботи WiFi-модулів для забезпечення стабільного трафіку в КФС був запропонований адаптивний метод уникнення зниження швидкості через інтерференцію від точок доступу (WiFi-мереж), які не є вузлами mesh-топології КФС, але знаходяться в радіусі дії останньої. Застосування розробленого алгоритму перевизначення енергетичних та частотних характеристик кіберфізичних компонентів у залежності від ефірної ситуації дозволило суттєво підвищити стабільність зв'язку між КФО. Виконана експериментальна верифікація розроблених методу та алгоритму, а також проведений порівняльний аналіз їхніх результатів з результатами ненормованого формування топології та бездротових каналів зв'язку КФС, доведені переваги запропонованого підходу.

Особливістю запропонованого адаптивного методу забезпечення стабільного бездротового зв'язку між рухомими об'єктами є те, що результати залежать від початкового ефірного стану у зоні дії КФС, заданої системи наявних перешкод, незалежних від характеристик КФО, а також від кількості запусків алгоритму адаптивного методу для одних і тих же вхідних даних (характеристик вузлів у топології КФС) з метою покращення зв'язку, що ускладнює повну автоматизацію процесу стабілізації швидкості.

Подальші дослідження доцільно проводити у напрямку збільшення граничної відстані між вузлами бездротової КФС та розширення зони дії останньої за допомогою додаткових спеціальних конфігурацій антен з різними коефіцієнтами підсилення, а також за допомогою управління потужністю сигналу на кіберфізичних компонентах всіх рівнів КФС.

Складнощі у впровадженні отриманих результатів дослідження пов'язані з двома основними чинниками. Перший з них – технологічна політика компаній-виробників комп'ютерних компонентів, яка обмежує характеристики вироблених модулів не теоретичними можливостями стандартів зв'язку, а цінним фактором виробів. Другий фактор – форс-мажорні обставини під час використання КФС (погодні умови, застосування засобів радіоелектронної боротьби з рухомими КФО, електромагнітне випромінювання промислових об'єктів тощо). Тому, при довгостроковому прогнозуванні вкладених коштів, необхідно враховувати тренди ринку новітніх матеріалів для конструювання БПА, які здатні посприяти покращенню розглянутих показників та суттєво змінити напрямок досліджень означених проблем.

3.2 Система контролю за мережею безпілотних апаратів, заснована на використанні мобільних пристроїв та хмарних сервісів

Система контролю за мережею БПЛА у загальному сенсі складається із множини підсистем, таких як підсистеми керування, навігації, енергоживлення та даних (рис. 3.12). Вказані підсистеми відповідним чином при подальшій деталізації розбиваються на інші складові.

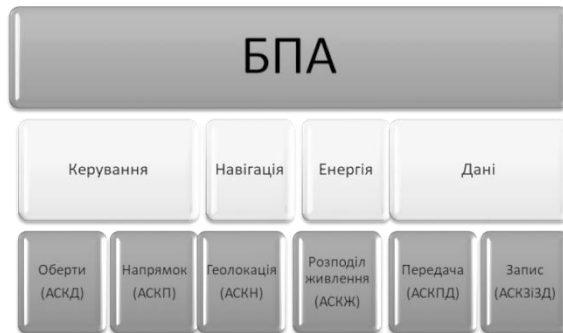


Рис. 3.12. Склад підсистем системи контролю за мережею БПА

У разі перевищення параметрів спеціальних режимів (надалі – спецрежими) система контролю забезпечує здійснення реплікації таких параметрів до наземного КЦ. Такими спецрежимами можуть бути, наприклад:

- температура більше ніж плюс 40 оС влітку та нижче мінус 20 оС взимку (при якій непрацездатне комп’ютерне, механічне, та вимірювальне обладнання);
- зависока швидкість вітру (та іноді піщані бурі), здатні змінити курс безпілота;
- високий ступінь електромагнітного випромінювання, при якому БПА стає некерованим, тощо.

Сервери наземного КЦ повинні прийняти параметри спецрежимів, розрахувати поправки до траєкторії руху БПА та засобами зворотнього зв’язку передати їх напряму на рухомий головний БПА («матку рою») або через комп’ютерний пристрій (планшет, телефон тощо) рухомого Абоненту, найближчого до «матки». Після отримання поправок до траєкторії БПА система керування «матки» коригує поведінку БПА, у т. ч. до самознищення або повернення на базу.

Після перебільшення параметрів спеціальних режимів та/або втрати працездатності GPS-модуля (в умовах великого електромагнітного опромінення у результаті застосування засобів радіоелектронної боротьби – створення «пухирів РЕБ») зграя БПА повинна перейти від режиму керування до автономного режиму [228; 232; 240; 298].

Зазвичай для виконання задач функціонування зграї БПА використовується рішення на основі до трьох одиниць БПА в одному суб-рої, що керується з одного пульту, а також кваліфікованого оператора, що може здійснювати управління апаратом. Такий підхід є ефективним для вирішення задач з високим рівнем складності, але він вимагає багато ресурсів та постійної присутності людини за пультом управління, що є малоефективним при виконанні великої кількості відокремлених задач.

Більш ефективними є гетерогенні зграї, що об’єднують декілька різнофункціональних суб-роїв, які складаються з наземних, над-/підводних та літальних БПА різних типів, здатних виконувати спільними зусиллями єдине завдання, поставлене перед зграєю.

Програмне забезпечення мобільної системи контролю за гетерогенною рухомою мережею оперує трьома типами дронів (рис. 3.13).

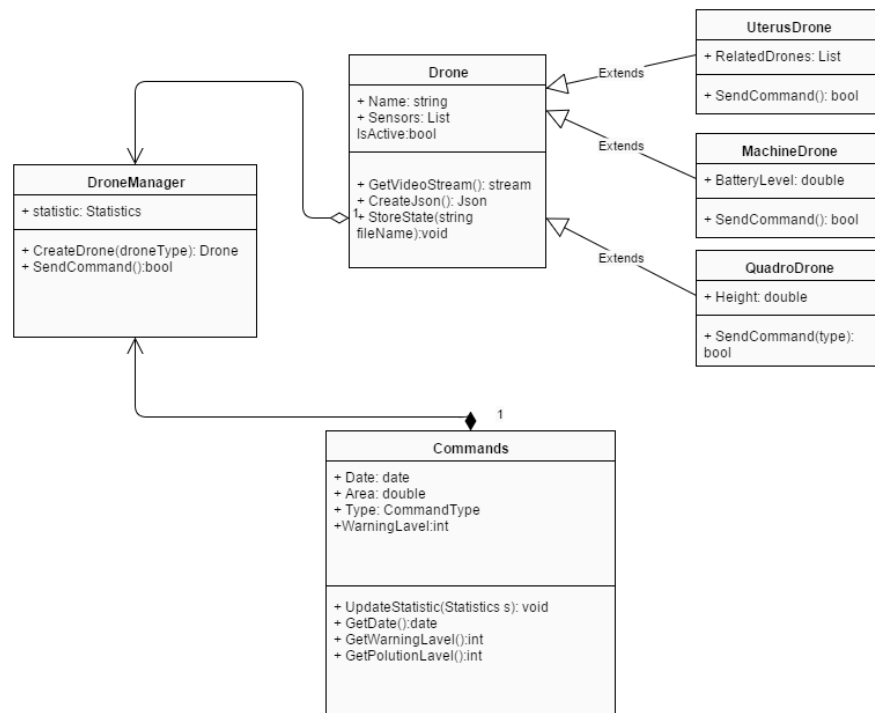


Рис. 3.13. UML-діаграма типів БПА (дронів)

Основні типи дронів на рис. 3.13 такі:

- а) Quadcopter – повітряний дрон;
- б) Machine – наземний дрон;
- в) Uterus – над-/підводний дрон.

Зважаючи на обмеженість ваги вантажу, який може піднімати БПА (в межах 30 % власної ваги), є сенс розділяти зграю БПА на суб-рої не за конструктивним типом, а за функціями (рис. 3.14):

1) суб-рій моніторингового призначення, БПЛА якого навантажені цифровими камерами покращених характеристик (до 12 Мпкс.), а на модуль пам'яті завантажені мапи моніторингової місцевості, фотознімки цільових об'єктів;

2) суб-рій транспортного призначення (суб-рій № 3), на борту БПА якого розміщений відповідний вантаж.

Необхідно зазначити, що у складі зграї можуть бути один (нп., суб-рій № 3 – транспортний, для ураження цілей чи доставки вантажів) або декілька суб-роїв із зазначеними функціями (нп., суб-рій № 1 – моніторинговий для взаємодії з повітряними та наземними транспортними БПА; суб-рій № 2 – моніторинговий для взаємодії з над-/підводними БПА).

При такому розподілі функцій між суб-роями, навіть під час відсутності зв'язку з командним центром та можливістю визначити GPS-координати у «РЕБ-пухирях», БПЛА суб-роїв № 1 та № 2 завантажують з модуля пам'яті в обчислювач частину мапи з останніми визначеними GPS-координатами. Потім, за допомогою бортових систем стереозору, на відокремленому сегменті завантаженої мапи здійснюється пошук цільового об'єкта за даними, отриманими з сенсорів [25; 28; 70; 220]. Програмне

забезпечення дрона створює тривимірну карту оточення, тому дрон може літати навіть за відсутності зв'язку з супутниками та визначати перешкоди на відстані до 30 м.

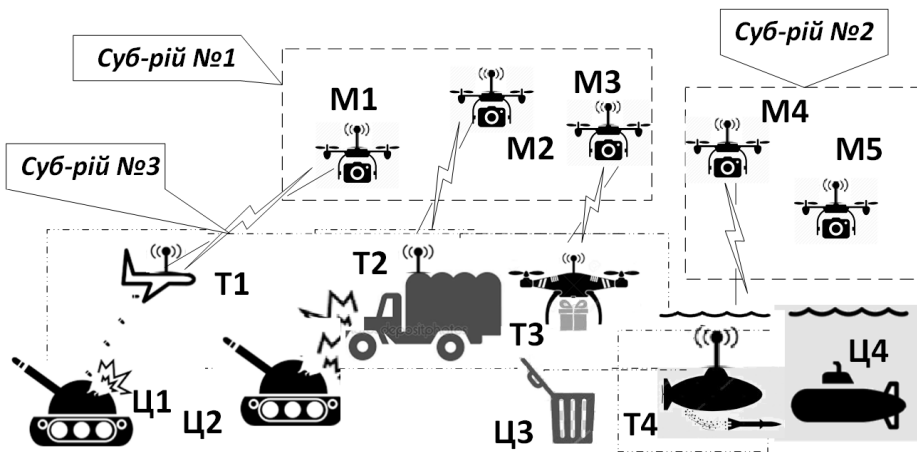


Рис. 3.14. Розподіл зграї БПА на суб-рої за функціями

Передача інформації між БПА зграї може здійснюватися через радіоканал з переключенням між частотами (нп., 2,4 і 5,8 ГГц), якщо на одній з них виникають перешкоди. Якщо радіозв'язок зовсім неможливий у «РЕБ-пухирях», тоді квадрокоптер моніторингового суб-рою «зависає» над знайденою ціллю, і дрон транспортного суб-рою може знайти його за допомогою власної системи стереозору.

Особливості контролю за суб-роями зграї БПЛА з урахуванням показників бортових датчиків та можливостей вбудованої системи комп'ютерного зору повинні розглядатись більш детально. Так, наприклад, при використанні у складі зграї БПЛА типу DJI Phantom 4 Pro необхідно враховувати дані з таких датчиків фронтального, заднього та нижнього стереозору [50]:

- цифрова камера з дюймовою CMOS-матрицею, що здатна знімати фото до 60 кадрів на секунду з 20-мегапіксельною якістю, у т. ч. в умовах низького (біля 15 люкс) та високого рівня яскравості (сонячне небо і темна земля під ним) на швидкості до 50 км/год;

- спарені датчики відео-системи на фронтальній поверхні корпусу;
- спарені датчики відео-системи на задній поверхні корпусу;
- спарені датчики відео-системи на нижній частині корпусу.

Розмір частини мапи має відповідати обмеженим обчислювальним можливостям одноплатного бортового комп'ютера БПЛА. Тому при відсутності цільового об'єкта на завантаженій частині мапи, в обчислювач БПЛА завантажувється наступний фрагмент сусідньої частини мапи.

БПА у складі суб-рою транспортного призначення за допомогою бортових відеосистем отримують інформацію від БПЛА моніторингового суб-рою щодо взаємного місцезнаходження (відстані) з цільовими об'єктами (Ц1–Ц3 на рис. 3.14).

Якщо метою транспортного БПА є не ураження цілі, а доставка вантажу, то на такому БПА повинен бути реалізований алгоритм автоматичного скидання посилки, або за ідентифікацією адресата. При помилковій ідентифікації БПА транспортного призначення може бути запрограмованим на знищення вантажу або ураження хибного адресата.

Під час розробки системи контролю враховані такі характеристики для кожної окремої моделі БПЛА [274]:

- виробник;
- максимальна злітна маса, кг (від 10 до 85);
- маса корисного навантаження, кг (від 2 до 25);
- максимальна швидкість, км/год (від 70 до 200);
- льотна межа («стеля»), км (від 0,5 до 4,0);
- дальність дії, км (від 3 до 500);
- тривалість польоту, годин (від 0,3 до 9,0).

Приклад сценарію використання системи контролю для збору поодиноких поточкових даних наведено на рис. 3.15.

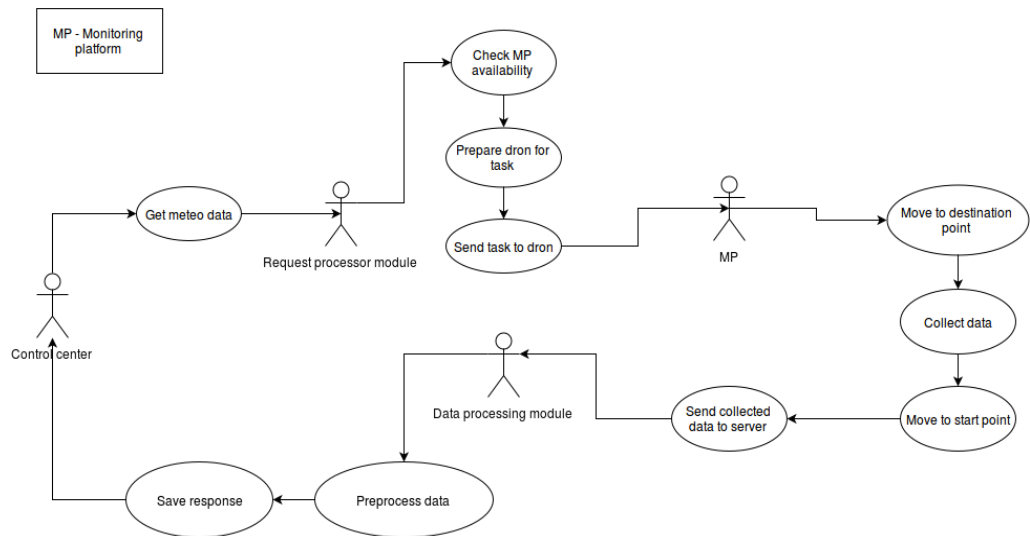


Рис. 3.15. Приклад сценарію роботи системи контролю

У складі ПЗ системи контролю за мережею БПА, заснованою на використанні мобільних пристроїв, передбачається функціонування таких модулів:

- програмний модуль зв'язку;
- програмний модуль керування;
- модуль обробки даних;
- клієнтський API;
- модуль зберігання та резервування даних.

При формуванні гетерогенної зграї з різних за конструкцією та функціями БПА, клієнтські застосунки (Client Applications) рухомих абонентів мережі можуть бути виконані на будь-якій сучасній мобільній платформі (Windows Phone/Mobile, iOS, Android тощо) [130; 230; 233].

Створення програмно-апаратної системи контролю БПЛА можливо на основі мобільного аналізу даних з використанням кросплатформного хмарного сервісу Google Firebase.

На сьогодні накопичений значний зарубіжний та вітчизняний досвід контролю за поодинокими БПЛА та зграями БПЛА з використанням інтегрованого підходу, за яким контроль здійснюється з використанням наземних та повітряних трансферних вузлів [125; 174; 354]. Сучасні системи контролю БПЛА використовують для цього координати БПЛА, отримані від вбудованих в останні GPS-модулів. Але за 1 хв польоту похибка маршруту БПЛА, скоординованого за даними GPS-модуля, може становити до 50 м

[354]. Проблемні питання точного досягнення кінцевої точки маршруту БПЛА під час використання сучасних супутникових технологій визначення координат повністю не вирішені [332].

Варто зауважити, що причинами виникнення неточних вимірів GPS-координат можуть бути:

1. Через штучне обмеження точності визначення координат цивільними GPS-навігаторами у межах 2–20 метрів [82; 303].

2. Хибна геометрія супутників. Виникає тоді, коли всі прийняті супутники згруповані близько один до другого, чи вибудовані в лінію щодо положення приймача, який у свою чергу не забезпечує необхідну розбивку сигналу на так званий трикутник, від чого точність обчислення координат стає менш надійною. Таке непередбачуване зміщення положення супутників на орбіті відбувається через різні збурювальні фактори та приводять до похибок 0,6–10 м [315].

3. Сигнали перед тим, як досягти GPS-приймача, можуть бути відбиті від високих будинків чи інших перешкод, тим самим збільшується відстань сигналу, або сигнал збивається, зменшуючи точність.

Зважаючи на те, що сучасні міні-БПЛА живляться від акумуляторів на борту, ємність яких забезпечує тривалість польоту від 7 до 28 хв [14]. Отже, можна зробити висновок, що для забезпечення якості виконання завдання необхідно здійснювати додаткове керування польотом БПЛА з коригуванням маршруту останнього до точки цілі. Поширеним підходом до вирішення питань такого керування є використання бездротових сигналів від мобільних пристроїв.

Авторами пропонується спосіб контролю БПЛА за допомогою мобільних пристроїв, які здійснюють мобільний аналіз даних з БПЛА та оточуючого середовища, що базується на використанні сервісу Google Firebase як бази даних реального часу (БДРЧ), засобу авторизації Клієнтів та Об'єктів системи контролю БПЛА, засобу синхронізації даних [336]. Зазначений безкоштовний хмарний сервіс є універсальною платформою для побудування Android- та iOS- мобільних застосунків. Перевагою саме цього сервісу є стабільна робота навіть при збоях Інтернет-зв'язку. Варто зауважити, що при використанні Клієнтів з різних платформ необхідно зареєструвати два облікових записи в Google (окремо для Android та для iOS).

На рис. 3.16 наведені можливі взаємозв'язки між Клієнтами та Об'єктами розроблюваної системи з використанням мобільних пристроїв для контролю БПЛА.

У Firebase SDK сервер має вбудований метод для створення користувацьких ідентифікаторів. Firebase створює відповідний унікальний ідентифікатор (рис. 3.17), який ідентифікує пристрій у мережі і надає певний доступ до ресурсів, а саме до бази даних у реальному часі Firebase Storage. Тобто, маємо ідентифікатор Об'єкта, за допомогою якого він ініціалізується Клієнтом [229; 301].

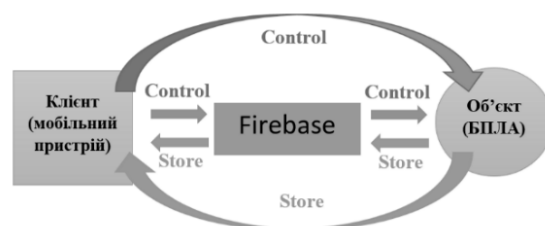


Рис. 3.16. Взаємодія мобільного пристрою з БПЛА через сервіс Google Firebase

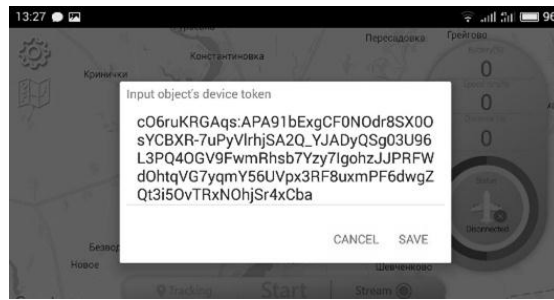


Рис. 3.17. Ідентифікатор Об'єкта в налаштуваннях Клієнта

На доступній хмарі Google Firebase реалізована структура збору даних щодо подій у розрізі користувачів (Клієнтів та Об'єктів), які пройшли авторизацію в названій хмарі.

Сегменти користувачів (так звані «аудиторії») розподіляються як за попередньо налагодженими параметрами (зона знаходження, технічні дані платформ БПЛА/мобільний пристрій тощо), так і за даними про події.

Наприклад, на рис. 3.18а наведена «аудиторія», створена на Google Firebase, яка складається з мобільного пристрою (Клієнт, зображений окремою великою крапкою) та двох керованих БПЛА (об'єктів, переміщення яких зображене двома низками менших крапок). Зазначена «аудиторія» сформована за географічним принципом знаходження в одному квадраті місцевості.

На рис. 3.18б наведена «аудиторія», сформована за ознаками платформи (Android).

Таким чином, унікальний рядок символів може бути переданим іншим членам «аудиторії» або через SMS-сервіс, або іншими соціальними мережами (коли відсутня можливість користування сервером). Після цього клієнт встановлює з'єднання з об'єктом і отримує доступ до ресурсів Firebase, де надається можливість зберігати дані об'єкта у реальному часі.

Після авторизації в Google Firebase, Об'єкт (БПЛА) кожні 5 секунд відправляє свої дані місцезнаходження Клієнтові своєї «аудиторії». При цьому не має значення наявність або відсутність безпосереднього зв'язку між Клієнтом та Об'єктом (Об'єктами), кого контролює цей Клієнт – засоби мобільного аналізу даних Google Firebase забезпечать передачу коригуючих команд саме необхідному Об'єкту.

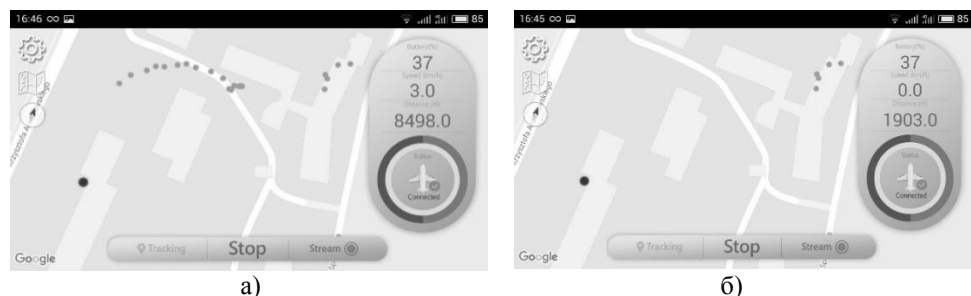


Рис. 3.18. «Аудиторія» Google Firebase, сформована за географічним принципом (а) та за принципом платформи (б)

З правого боку екрана Клієнта розташований індикатор поточного заряду акумулятора Об'єкта (у відсотках), швидкість Об'єкта (км/год) та дистанція між Клієнтом та Об'єктом (в метрах).

У запропонованій системі контролю відтворений розроблений протокол взаємодії між Клієнтом та кожним з Об'єктів з урахуванням показників цілого набору пристроїв на

борту БПЛА: акселерометра, барометра, гіроскопів, магнітографа, GPS-модуля, GSM-модуля тощо.

За результатами мобільного аналізу сукупності наведених показників Клієнтом можуть бути змінені, наприклад, координати фінішу Об'єкта – широта і довгота (Latitude and Longitude). Діаграма класів для програмної реалізації зазначеного завдання наведена на рис. 3.19.

Приклад оформлення протоколу взаємодії GPS-модуля Об'єкту з Клієнтом наведений на рис. 3.20, деталізація параметрів протоколу представлена у табл. 3.2.

>\$GPRMC,170840.00,A,5445.33016,N,1704.38950,E,1.6198,,310317,,,A*7D

Рис. 3.20. Нотація протоколу взаємодії Клієнта з Об'єктом при зміні координат фінішу БПЛА

Таблиця 3.2. Деталізація параметрів зміни GPS-координат фінішу під час керування БПЛА

170840.00	Час UTC: 17 год 08 хв 40,00 с
A	Статус достовірності: A – достовірні дані, V – недостовірні дані
5445.33016	Широта: 54 град. 45,33016 хв
N	N – північ, S – південь
1704.38950	Довгота: 17 град. 04,38950 хв https://data.aad.gov.au/aadc/calc/decimal_dms.cfm
E	E – схід, W – захід
1.6198	Горизонтальна швидкість, вузлів/год (відповідає 3 км/год на рис. 3.18а)
–	Напрямок курсу відносно півночі (градуси)
310317	Дата: 31 березня 2017 р.
–	Магнітне відхилення (градуси)
–	Напрямок відхилення: E – східний, W – західний
A	Режим: A – автономний, D – диференціальний, E – апроксимація, N – недостовірні дані
*7D	Контрольна сума

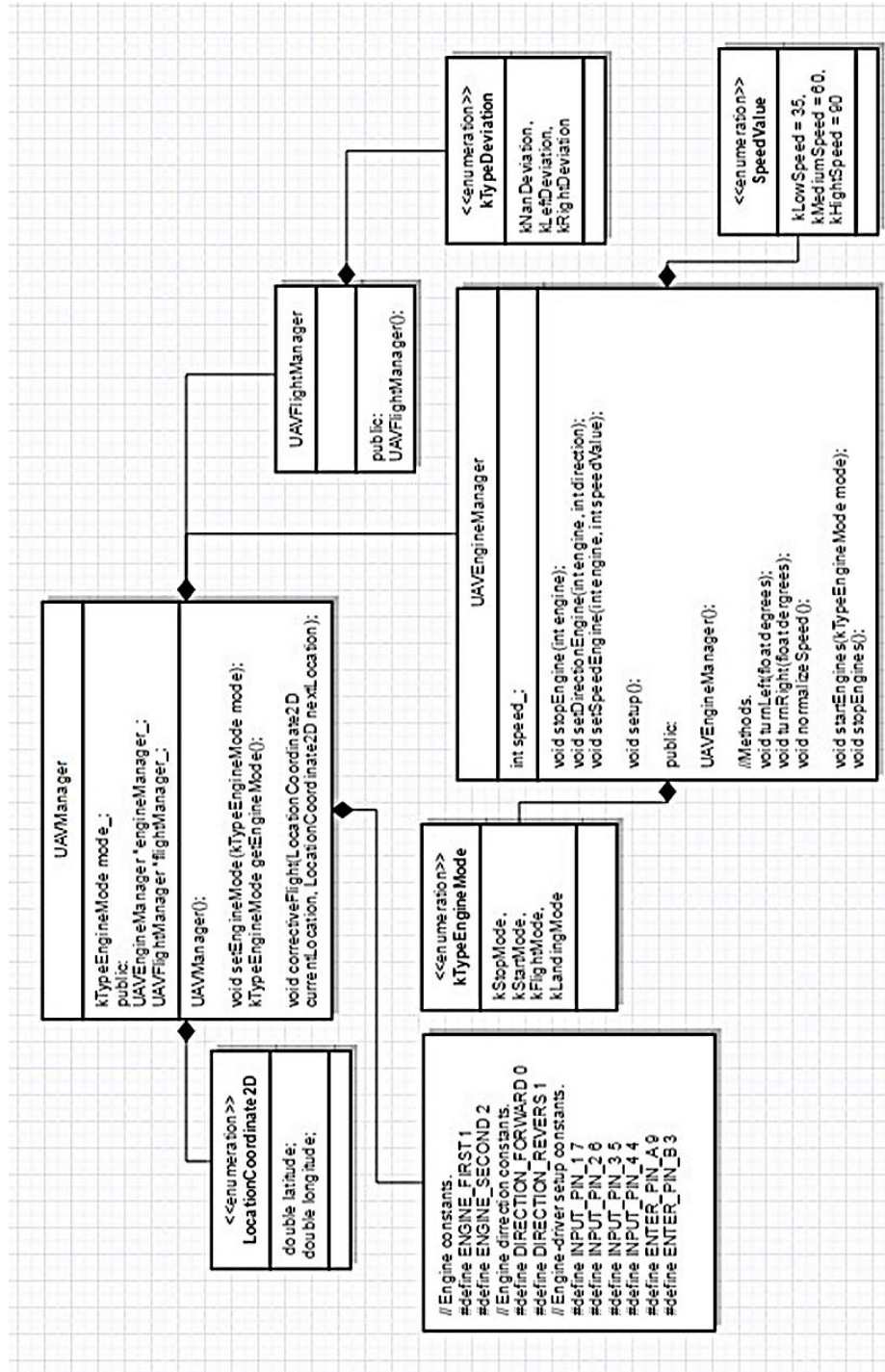


Рис. 3.19. UML-діаграма класів зміни координат фінішу під час керування БПЛА

Коли Об'єкт прибув до цілі, на екрані Клієнта з'являється відповідне повідомлення (рис. 3.21).

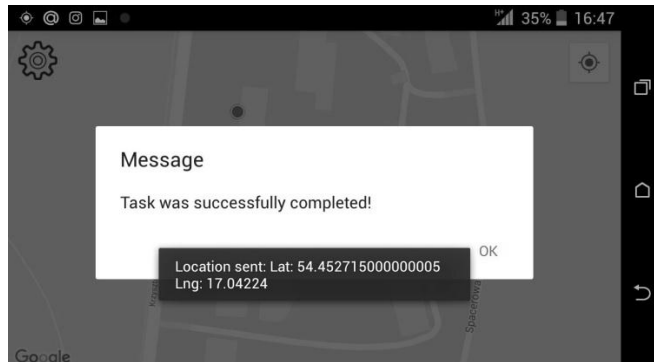


Рис. 3.21. Повідомлення на екрані Клієнта після виконання завдання керованим БПЛА

Зіставлення координат широти та довготи рис. 3.20, табл. 4.2 та рис. 3.21 свідчить, про відхилення між координатами заданими системою керування та координатами фінішу після виконання завдання. Таке відхилення є наслідком неідеальної роботи GPS-модуля БПЛА, в результаті чого відстань рахується з похибками.

Найчастіше програмні застосунки за клієнт-серверною структурою розробляються таким чином, що серверна частина розташована на наземному командному центрі або на планшеті (ноутбуку й т. п.) кібероператора, а мобільний застосунок, виконуючий реєстрацію заданих подій/параметрів, працює на пересувних/рухомих об'єктах (мобільних пристроях, WiFi-камерах, дронах, тощо) з обмеженими обчислювальними можливостями. Необхідно зауважити, що використання даних у хмарних сховищах вимагає спеціальних заходів для захисту такої інформації [71; 72; 203; 249].

Якщо КФС має охоплювати декілька пунктів з різними GPS-координатами, окрему увагу необхідно приділяти вирішенню проблем у налаштуванні серверів. Наразі перше, за рахунок чого намагаються зменшити навантаження на сервер, це горизонтальне масштабування (шардінг) бази даних (рис. 3.22), розподіленої між різними фізичними серверами [119]. Вказана система виконує такі функції, як зменшення навантаження за рахунок використання двох або більше баз даних, зменшення кількості запитів до однієї бази, кешування, збереження даних, обробки запитів, балансування навантаження та ін.

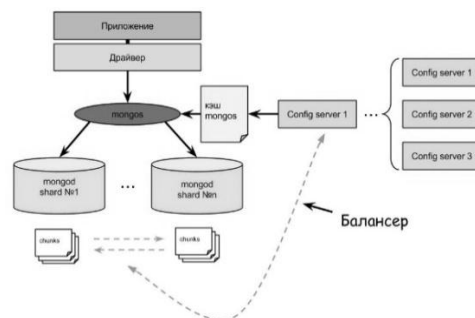


Рис. 3.22. Шардована схема без репліка-сетів [119]

Шардування надає можливість зменшити навантаження і забезпечити необхідну швидкість системи. Але це підхід, який потребує налаштування, куди підуть дані після їх прибуття в базу. Це зумовлено тим, що розподіл ведеться по Shard Key, і з кожною новою базою перенос та розподілення даних буде важчий за минулий.

Аналогом є схема з репліка-сетами (рис. 3.23).

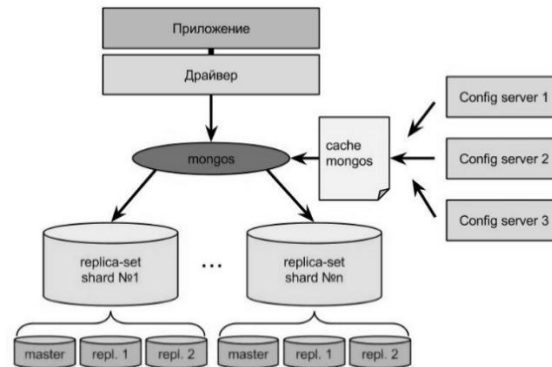


Рис. 3.23. Шардована схема з репліка-сетами [119]

Ця система має такі переваги:

- збереження даних;
- репліки беруть на себе запити зчитування, чим зменшують навантаження на систему;
- балансування навантаження;
- репліки можуть знаходитись в іншому місці ніж майстер.

Структурна схема РММ, яка має клієнт-серверну структуру, наведена на рис. 3.24.

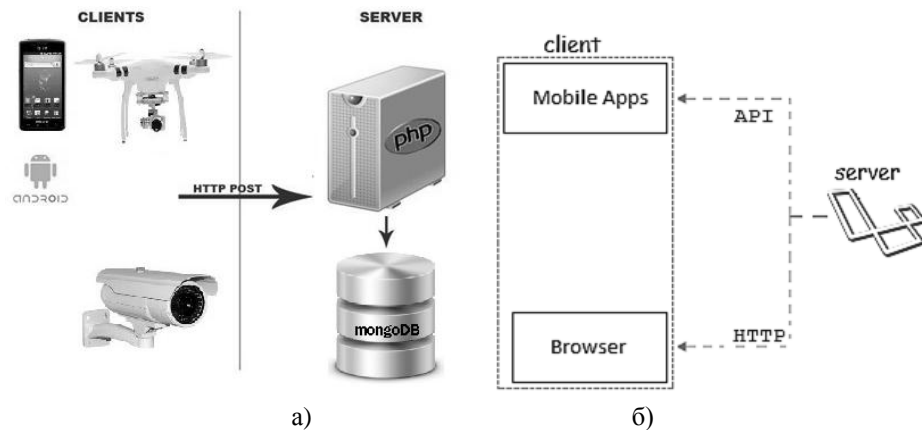


Рис. 3.24. Структура РММ (а) та клієнт-серверного застосунку (б)

Така КФС працюватиме також в автономних умовах (у т. ч. при неможливості визначити GPS-координати рухомих КФО) та спроможна обробляти інформацію від гетерогенної мережі великої кількості реєстраторів з декількох віддалених один від одного пунктів обліку. Використання Open Source рішення на базі NoSQL СУБД

mongoDB забезпечить більш економне рішення в поєднанні з досить простою масштабованістю при збільшенні пунктів обліку.

Розроблюване рішення працездатне при обмеженому доступі до Інтернету, необхідному тільки під час синхронізації даних з декількох пунктів обліку для формування необхідної статистики в реальному часі.

Система також має бути укомплектована модулями безпеки, які дозволять регулювати права доступу користувачів до певного функціоналу [35; 51; 102; 226].

Має сенс під час проектування інформаційної підсистеми КФС розробити структуру сервера обробки запитів, в якій реалізоване поєднання технологій Gearman та Varnish (рис. 3.25).

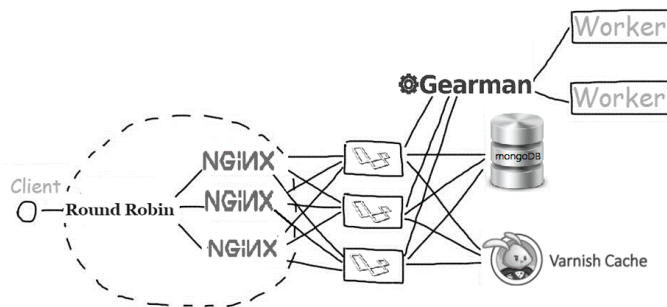


Рис. 3.25. Структура сервера обробки запитів

Застосування технології Gearman забезпечує організацію системи черг для потоку клієнтів (в т. ч. і для PHP). Одна з основних частин Gearman – Client – ставить завдання на виконання, звертаючись до сервера завдань. Навантаження КФС розподіляється за алгоритмом Round Robin (від англ. Round-robin – циклічний), який реалізує розподіл завдань між серверами NGINX методом перебору і впорядкування елементів круговим циклом. Компонент Worker опитує сервер на наявність завдань, які він може обробити. Переваги такої архітектури у тому, що всі три компоненти не пов'язані між собою, і кожен з них може бути запущений в будь-якій кількості на будь-яких машинах, спілкуючись за допомогою чіткого інтерфейсу створення застосунків (API).

Упровадження описаного рішення дозволяє масштабувати завдання на кілька серверів, а також має можливість пріоритизації завдань. Завдяки впровадженню сервера черг, кожен запит з різних пристроїв буде виконаний у своєму контексті (за наявності декількох систем зчитування) та не буде втрачено жодного запиту навіть за великої їх кількості.

Доповнення розробленого рішення кеш-сервером Varnish дозволяє підвищити продуктивність на рівні веб-сервера. При незмінному складі компонента (облікової інформації на пункті реєстрації) буде віддаватися його кешована версія без зайвих запитів всередині сервера. Якщо дані змінилися, то компонент буде перекомпільованим передаватись у Varnish, потім з нього у веб-сервер, а вже потім клієнту.

Запропонований метод побудування КФС забезпечує реалізацію переваг багаторівневої базової платформи КФС [114] з погляду як проектування, модифікації, інтеграції нових рішень в КФС, так і з погляду використання такої системи.

Створена інформаційна система представлена у вигляді веб-сервера з інтерфейсом для користувача (рис. 3.26 – рис. 3.27). Сервер має такі функції:

- оновлення даних з сервера;
- обнулення лічильників подій;

- перегляд актуальних даних за кількістю зареєстрованих подій на кожному пункті реєстрації;
- запити до клієнтських застосунків для оновлення конкретної інформації.

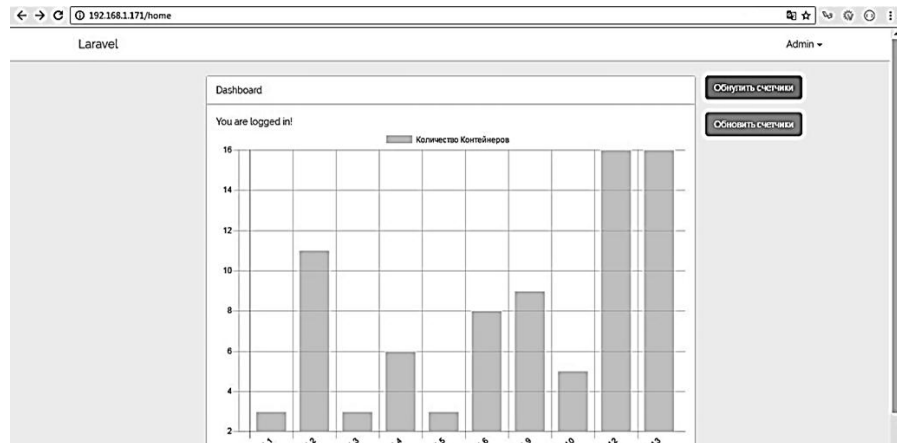


Рис. 3.26. Сторінка адміністрування актуальності даних

```

Load URL http://192.168.1.171/device
Split URL
Execute
Enable Post data [checked] Enable Referrer [unchecked]
Post data device_token=192.168.1.1&counter=2
JSON | Необработанные данные | Заголовки
Сохранить | Копировать
id: 5
device_token: "192.168.1.1"
counter: 13
updated_at: "2017-04-19 20:17:19"
created_at: "2017-04-19 00:00:00"

```

Рис. 3.27. Сторінка запитів сервера

Таким чином, інформаційна підсистема КФС реєстрації подій/параметрів надає власникові інформацію щодо їхньої кількості у розрізі за кожним встановленим реєстратором, інтенсивність потоків подій/вимірювань у визначений час, градації параметрів за значенням та ін. За допомогою інтерфейсу можливо авторизуватись в системі під звичайним користувачем та переглянути дані, але змінити їх може тільки особа, яка є у затверженому списку доступу.

Висновки до розділу 3

Запропоновано адаптивний метод покращення стабільності бездротового зв'язку між КФО шляхом перевизначення інтегрованих характеристик частоти та потужності

WiFi-модулів комп'ютерних компонентів КФО в залежності від радіоситуації. Показано, що при певній сукупності вихідних параметрів, зниження максимальної потужності WiFi-модуля на 50 % покращує якість зв'язку майже у 8 разів. Проведені дослідження дозволили досягнути максимальної симетрії каналу зв'язку між об'єктами КФС та зрівняти між собою швидкості потоків «up» та «down» між КФО.

Виконане оцінювання зміни швидкості передачі даних від кількості ретрансляцій. Шляхом перескоку лінії зв'язку в інший частотний діапазон з меншим рівнем перешкод та впровадження функції ретрансляції сигналу засобами наявних КФО, що є вузлами mesh-топології КФС, покращені показники стабільності швидкості передачі даних на рівні 90 % замість 47 %, у зазначених літературних джерелах [242].

Розроблено програмну платформу системи контролю за об'єктами гетерогенної рухомої мережі, в яку введено додаткові зв'язки між БПА та рухомими абонентами (користувачами, оснащеними комп'ютерними або мобільними пристроями), які знаходяться у зоні дії каналів зв'язку БПА. Ураховано часові затримки між сеансами синхронізації даних, зумовлені тимчасовою втратою зв'язку БПА з мобільним та/або наземним серверами зазначеної гетерогенної мережі.

Показано, що клієнтські застосунки рухомих абонентів мережі можуть бути виконані окремо на будь-якій сучасній мобільній платформі (Windows Phone/Mobile, iOS, Mac OS, Android тощо) та забезпечувати взаємодію між різнофункціональними суб-роями зграї.

Розроблено багатомодульну архітектуру мобільної системи контролю за рухомою мережею моніторингу та доставки, яка базується на конвергентному механізмі централізованого керування та автономній роботі кожного пристрою (нп., при оминанні механічних перешкод та/або відсутності радіосигналу), дозволяє суттєво збільшити ефективність гетерогенної мережі БПА.

Удосконалення розглянутої системи контролю за зграєю БПА виконано за рахунок використання хмарного сервісу Google Firebase (можливо використання інших хмарних сервісів). Запровадження такого підходу дозволить здійснювати тимчасове зберігання часткових даних на хмарному сервісі з наступною їх синхронізацією між декількома об'єктами гетерогенної рухомої мережі БПА.

Проаналізовані та узагальнені причини похибки маршруту БПЛА у разі прокладання маршруту на підставі даних з GSM-модуля БПЛА. Доведено необхідність коригування маршруту БПЛА.

Запропоновано спосіб контролю БПЛА з використанням мобільних пристроїв та застосуванням хмарного сервісу Google Firebase, який надає інтегровані послуги авторизації мобільних пристроїв та БПЛА.

Використання властивостей Google Firebase у розробленій програмно-апаратній системі дозволяє позбавитись від зберігання отриманої моніторингової інформації як на самому БПЛА, так і на пристрої контролю БПЛА, залишаючи натомість інформацію про залежність цих даних у зазначених пристроях.

Такий підхід суттєво підвищує безпеку поточної інформації щодо функціонування БПЛА у разі перехоплення Об'єкта сторонніми особами.

Крім того, надійна платформа та технології Google забезпечують доступність, інтеграцію в інші Google-сервіси (Google Maps ті ін.), а також масштабованість Об'єктів у зграї БПЛА.

Напрямом подальшого дослідження є розробка способів коригування похибок визначення відстані від мобільного пристрою до керованого БПЛА, спричиненої неточністю результатів функціонування GPS-модулів БПЛА.

Розглянуто архітектуру клієнт-серверних рішень взаємодії КФО з шардуванням на стороні сервера та використанням мобільних пристроїв з обмеженими обчислювальними можливостями на клієнтській стороні. Запропонований підхід може бути використаним у системах обліку великорозмірних вантажів (морських та сухопутних контейнерів, автомобілів, які проходять через пункт обліку тощо) та дозволяє отримувати у режимі реального часу інформацію щодо кількості вантажів, які перетнули зазначений пункт. Розроблена архітектура клієнт-серверного застосунку з шардуванням дозволяє використовувати в системі реєстратори з обмеженими обчислювальними можливостями.

Запропоновано рішення розраховане на роботу у складі локальної мережі, що суттєво знижує вартість такої системи у порівнянні з існуючими аналогами, які потребують наявності зв'язку з супутниками або з глобальною мережею. Також відсутні додаткові витрати на попереднє електронне маркування вантажів.

Віддалений доступ до мережі необхідний в описаній системі не під час поточної роботи, а тільки при потребі складання узагальнених звітів (за день, за місяць та ін.) шляхом синхронізації з серверами обробки даних. Зважаючи на обмежені обчислювальні можливості застосованих реєстраторів, в архітектуру КФС імплементовані технології Geotman та Varnish з репліка-сетами, що дозволяє зробити шардування й винести функціонал обробки даних на будь-яку необхідну кількість фізичних серверів, а сховище даних – у віртуальне середовище хмарних технологій.

Запропонований підхід суттєво зменшує навантаження на систему.

Недоліками представленої системи можна вважати відсутність кросплатформеного рішення на боці первинних реєстраторів.

Розділ 4

СТВОРЕННЯ ТА РЕАЛІЗАЦІЯ БЕЗПЕЧНИХ МЕХАНІЗМІВ ОБМІНУ ДАНИМИ МІЖ НАЗЕМНИМ ПУНКТОМ КЕРУВАННЯ ТА БЕЗПЛОТНИМ ЛІТАЛЬНИМ АПАРАТОМ

4.1 Розробка блокового методу шифрування на базі операції XOR для рухомих об'єктів з обмеженими обчислювальними ресурсами

Усі сучасні криптосистеми спираються на принцип Керкгоффа, відповідно якому секретність закодованих даних визначаються секретністю лише ключа кодування інформації користувача [209]. Тобто, всі алгоритми шифрування даних загальновідомі.

Алгоритми шифрування поділяють на симетричні і асиметричні. Кожен з них має свої переваги та недоліки.

У ході симетричного шифрування використовується лише один ключ, заздалегідь відомий всім учасникам протоколу. Перевагою цього виду шифрування є те, що швидкість створення зашифрованого документа та його відкриття займає небагато часу. Недолік симетричного шифрування в тому, що воно передбачає використання захищеного каналу зв'язку для передачі ключа між користувачами.

Під час асиметричного шифрування використовуються два ключа – відкритий і секретний. Відкритий ключ використовується для зашифровки повідомлення, а секретний – для розшифровки. Це створює достатньо надійний захист інформації. Але недоліком цього виду шифрування є низька швидкодія через складну реалізацію та велику кількість обчислень.

Тому проблема забезпечення високої швидкодії алгоритмів шифрування пояснює більшу поширеність симетричних алгоритмів. Фактично, асиметричні алгоритми використовуються лише для передачі ключів шифрування, які потім застосовують у симетричному шифруванні [36].

Шифри, які забезпечують високий рівень криптостійкості, є надлишково складними в реалізації. Це робить їх непридатними для використання на СРО з обмеженими обчислювальними ресурсами (БПЛА, планшети, мобільні апарати й т. п.). Щоб можна було ними скористатися для об'єктів РММ, доцільно «полегшити» такі алгоритми шляхом зменшення складності криптографічних перетворень.

Дослідження сучасних вчених-криптографів свідчать, що при збереженні потрібного рівня криптостійкості можливо використовувати спрощені алгоритми, що забезпечить підвищення швидкості шифрування [279; 291; 293]. Тому у багатьох сучасних методах шифрування застосовуються прості логічні операції (наприклад, XOR). Потреби в більш витончених алгоритмах не виникає, оскільки XOR вже забезпечує абсолютну стійкість. Зрозуміло, що це можливо тільки у випадку, якщо виконуються три необхідні й достатні умови стійкого ключа, сформульовані Клодом Шенноном [353].

Останнім часом стало популярним для захисту трафіка рухомих об'єктів, у т. ч. БПЛА, використання блокових методів шифрування [260; 294].

Метою роботи є розробка блокового методу шифрування зі знизеними вимогами до обчислювальних ресурсів за рахунок використання простих логічних операцій: додавання, перемішування, здвиг бітів та бінарна операція XOR [255; 266].

Одним з найбільш легких та найпростіших відносно обчислювальних ресурсів є алгоритм з використанням операції XOR. Виходячи з цього, була здійснено спробу розробити блоковий метод, в основу якого було покладено таку операцію.

Варто зазначити, що шифрування відеосигналів, які передаються поточно з БПЛА, є не цілком пріоритетною задачею [214]. Головною метою шифрування даних є підвищити прихованість наступних дій БПЛА в умовах зорового контакту з противником. Оскільки кодування передаваної відеоінформації з рухомого пристрою вимагає немалих ресурсів, то, відповідно, вбудовувати в керуючий модуль алгоритм шифрування відеоданих є недоцільним. Це обумовлено тим, що така складна архітектура модуля потребує великих обчислювальних ресурсів. У такому разі система на базі одного обчислювача має виконувати багато задач за одиницю часу – а саме: шифрування та передачу даних; геопозиціонування рухомого об'єкта; аналіз перешкод руху або відпрацювання команд кібероператора тощо. Описане рішення за сумою факторів обумовлює критичне застосування такої кіберфізичної системи та може призвести до втрати самого літального апарату [115; 121; 302].

Підвищити час виявлення маршруту БПЛА можливо за рахунок кодування наступних координат геопозиціонування, які вказують на наступну точку положення БПЛА в просторі. Тому така мета є першочерговою для забезпечення успіху в моніторингових операціях.

Навіть якщо прийняте рішення щодо кодування відеопотоку, наприклад, за допомогою ВРС-алгоритму, то для кодування геопозиційних координат такий або подібний метод не є доцільним через його «прожерливість» до обчислювальних ресурсів та архітектурну складність. Конкатенація даних з номером сеансу для розпізнавання «Свій-Чужий» є небезпечною. Умовний противник, підбравши номер сеансу зв'язку, може заволодіти конфіденційними даними системи.

Загальна модель управління процесом шифрування здійснюється за допомогою періодичної зміни ключа шифрування, який забезпечує кожного разу оригінальне представлення інформації під час використання одного й того ж алгоритму або пристрою [209; 279].

Процес шифрування передбачає узгодження ключа між користувачами і використання його в процесі обміну та шифрування даних [291; 320; 353]. Технічна особливість шифрування така, що злоумисник, не маючи унікального ключа, який використовувався користувачами, не зможе отримати миттєвий доступ до інформації.

Найпростішим та достатньо економним з точки зору використання обчислювача КФО є алгоритм шифрування з використанням простої логічної функції XOR (рис. 4.1) [268; 273]. Однак, він занадто вразливий до зламу.

Для подальшого ускладнення вищезазначеного алгоритму спробуємо ввести до нього модифікації, що дозволять суттєво збільшити час, необхідний для його зламу. Тобто, час криптоаналізу повинен перевищувати час актуальності переданої інформації.

Звичайно, замість XOR можна використовувати який-небудь інший алгоритм. Але так як XOR є одним з найшвидших (обчислювально-ефективних алгоритмів), це дає можливість без затримок швидко отримувати й розшифровувати інформацію, яка надходить до отримувача. Це важливо при шифруванні даних для БПЛА, для якого необхідно забезпечити високу швидкість отримання вказівок щодо траєкторії переміщення апарату.

Таким чином, розробка модифікованого алгоритму на базі операції XOR є актуальним та обґрунтованим рішенням.

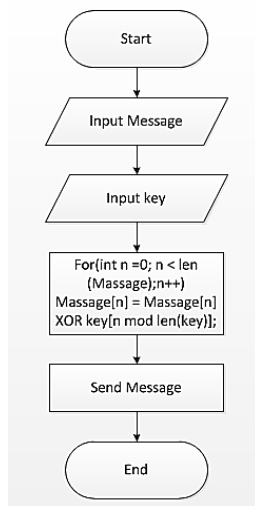


Рис. 4.1. Класичний алгоритм шифрування з операцією XOR

Оскільки операція XOR використовує ключ (x), який можна визначити за допомогою підрахунку індексів збігу, то початковим завданням є необхідність забезпечити непрацездатність використання цього алгоритму для визначення ключа.

У результаті проведених тестувань з математичними операціями, було прийнято рішення про введення для шифрування всього тексту повідомлення єдиного числового ключа, який утворюється за рахунок символів (рис. 4.2).

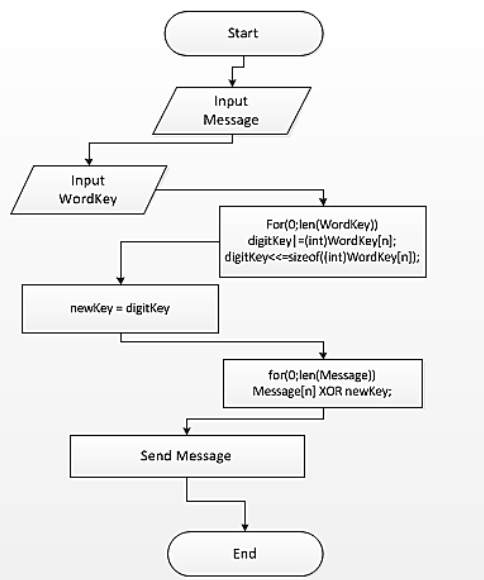


Рис. 4.2. Модифікований алгоритм шифрування з операцією XOR із застосуванням числового ключа

Модифікований алгоритм виглядає наступним чином:

Крок 1. Вводимо значення ключа.

Крок 2. Складаємо числовий ключ, кожного разу зміщуючи на один біт після числового значення символу символів ключа.

Крок 3. Виконуємо операцію « \wedge » (XOR) до кожного символу повідомлення.

Крок 4. Отримуємо зашифроване повідомлення.

Тим самим було створено алгоритм функції створення унікального ключа на основі тієї ж операції XOR. Таким чином вдалося забезпечити захист від описаного попередньо алгоритму зламу.

Але позбутися того, що в алгоритмі присутня частота повтору відповідних символів, не вдалося, оскільки це властивість операції XOR. Тому з вищезазначеного алгоритму зламу спрацьовує лише обрахування частоти повтору символів.

Актуальною задачею є підвищення криптостійкості XOR-шифрування шляхом перемішування блоків у раунді.

Для цього необхідно створити схему послідовності етапу шифрування (так званий «раунд» в криптографії), за якою можна було б отримати унікальне значення кожного біту задля того, щоб вирішити проблему зламу XOR-шифрування.

У початковому наборі зробити кожний біт унікальним неможливо, оскільки їхні значення знаходяться у виборці, яка обмежена значенням 255. Для вирішення питання унікальності необхідно скористатися розбиттям інформації на блоки по 32/64/128/256 бітів. Таким чином можна досягнути унікальності набору в блоці.

Але цього недостатньо, так як сама інформація хоча і кодується, але можливість дістати її першопочатковий образ залишається доступною. Для забезпечення криптостійкості необхідно застосувати перемішування цих блоків між собою у певному порядку [353]. Причому, порядок перемішування генерується геш-функцією, яка може розбити блоки на 10/12/14 в залежності від довжини ключа [324]. Таким чином досягається унікальність набору, і тим самим підвищується криптостійкість системи.

Після етапу перемішування блоків необхідно використати повторне розбиття блоку на більш малі частини, які б займали по 32 біти. Але перед цим необхідно виконати перевірку на умову повноти. Сутність повноти блоку полягає в тому, що вхідна інформація має ділитися націло на 32 біти. Якщо ж це не виконується, то вхідна інформація доповнюватиметься дописуванням спеціального символу.

Отже, розбивши вже закодовану інформацію на менші блоки, можна перейти до основного кодування інформації – це кодування наступного блоку попереднім блоком, щоб зав'язати відповідну послідовність розташування малих блоків (рис. 4.3).

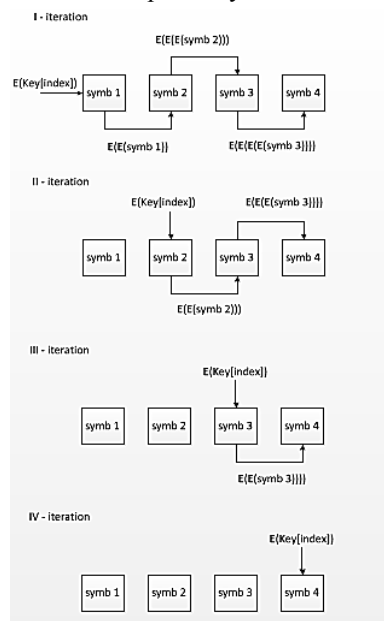


Рис. 4.3. Раунди блокового шифрування інформації

Таким чином, головна ідея методу полягає в тому, що інформація кодує сама себе, але за участю ключа користувача, який коригує схему кодування. Цей метод шифрування відповідає вимогам сучасності та надає повну свободу вибору ключа користувачем. Такий ключ може складатися з будь-яких символів з таблиці ASCII (окрім NULL).

Для реалізації шифрування повідомлень з забезпеченням захисту від підрахунку збігу індексів доцільно використовувати геш-функцію. На даному етапі розробки реалізуємо геш-функцію, беручи за основу операцію XOR-шифрування (рис. 4.4). Алгоритм, схожий на AES, реалізується досить просто. Крім того, одна й та ж функція виконує як шифрування, так і дешифрування. Це позитивно впливає на швидкість обміну та використання алгоритмів шифрування повідомлень.

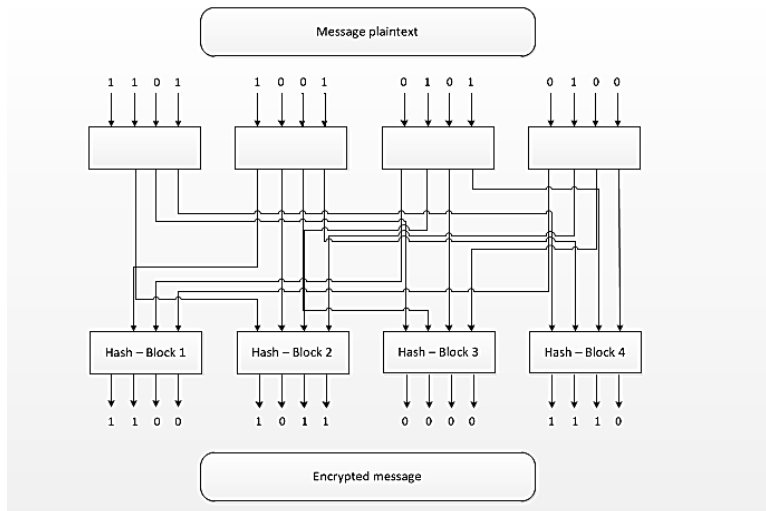


Рис. 4.4. Реалізація геш-функції на основі блокового алгоритму

Результатом створення даної функції стало те, що її застосування приходиться на частину числового ключа. Тобто, ми генеруємо числовий ключ, як і раніше, але після ініціалізуємо геш-функцію з 32-бітними константами. Після ініціалізації виконуємо комбінування значень кожного символу ключа з цими константами, яких чотири. Наступним кроком є те що, ми організуємо шифрування, використовуючи операцію XOR, як і раніше, але шифруємо циклічно кожен літеру повідомлення відповідною константою геш-функції (рис. 4.5).

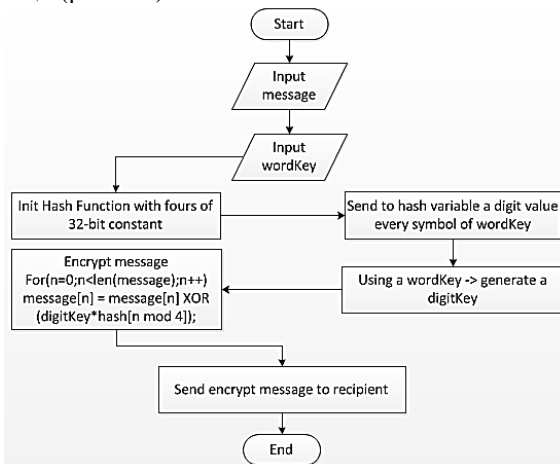


Рис. 4.5. Модифікований алгоритм шифрування з використанням геш-функції

У результаті внесення до алгоритму шифрування деякої геш-функції метод зламу, який успішно працював раніше, вже не спрацьовує або кожного разу помиляється з визначенням довжини ключа.

Актуальним є оцінка часу шифрування та складності криптоаналізу розробленого алгоритму на базі XOR.

Алгоритм описаного вище методу шифрування було реалізовано на мові програмування C++, в середовищі QtCreator. QtCreator було обрано, тому що він надає можливість кросплатформності. Отже, в подальшому можна встановлювати програму на інші операційні системи.

Дослідження створеного методу показало, що його застосування не потребує великих ресурсних витрат на генерацію раундових ключів. Це надає змогу виконувати операції кодування та декодування інформації в короткі проміжки часу, Таким чином забезпечується висока криптостійкість даної системи.

Застосовані операції виконуються за один такт процесора та можуть мати як програмну, так і апаратну імплементацію, у т. ч. на пристроях з обмеженими можливостями (наприклад, мікроконтролерах).

Оскільки XOR-шифрування – це симетричний алгоритм, то відкритий текст піддається операції «виключне АБО» (Logical Exclusive OR, XOR) разом з ключовим текстом для отримання шифротексту. Повторне застосування операції XOR відновлює оригінал. Для шифрування і дешифрування використовується одна й та ж функція.

На графіку рис. 4.6 зображено залежність часу шифрування та дешифрування даних залежно від обсягу інформації. Як видно з рис. 4.6, застосування такого методу кодування даних в БПЛА (клієнт) забезпечить швидку детермінацію напряду, який надсилає командний пункт (сервер).

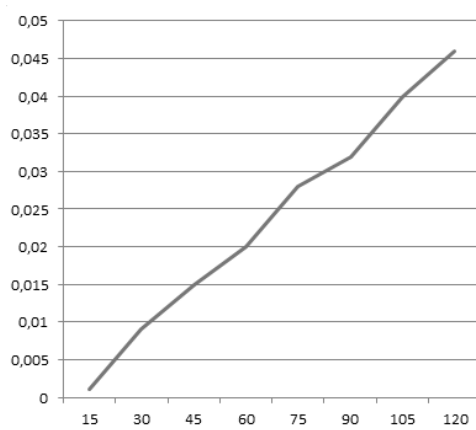


Рис. 4.6. Залежність часу шифрування (с) від обсягу інформації (байт)

Якщо в запропонованому методі на базі операції XOR використовуються прості логічні операції, то кожна операція виконується в середньому 5 мкс. Тому нескладно обчислити теоретичний час P шифрування даних:

$$P = P_{сер}m(k1 + k2 + k3), \quad (4.1)$$

де P – загальний теоретичний час;

$P_{сер}$ – опосередкований час виконання кожної операції;

m – вага (обсяг) вхідної інформації;

$k1$ – кількість операцій зсуву;

$k2$ – кількість операцій логічного виключення;

$k3$ – кількість пов'язаних блоків.

Треба зауважити, що в формулі (4.1) величини k_1 , k_2 , k_3 динамічно змінюються в залежності від довжини ключа та інформації, що шифрується.

З аналізу рис. 4.6 можна дійти висновку, що достатній обсяг поодиноких даних для передачі з БПЛА (показання датчиків окремих величин, координати знайдених об'єктів тощо) може бути зашифрованим за дуже невеликий час, який не перевищує значення, співвідносні з часом в 1 с.

Такий же час необхідний сучасному обладнанню для зламу систем БПЛА з відомими характеристиками [221; 234]. Цей тип шифрування легко зламуються. Оцінимо кроки та час, необхідний для криптоаналізу алгоритму на базі XOR.

Зазвичай, відкритий текст повідомлень між об'єктами РММ використовує англійську мову (латиницю з цифрами та іншими ASCII-символами). Зламати цей шифр можна у такій послідовності:

1. Визначаємо довжину ключа за допомогою процедури, відомої як «підрахунок збігів» [353]. Застосуємо операцію XOR до шифротекст, використовуючи як ключ сам шифротекст з різними зсувами, і підрахуємо співпадаючі байти. Якщо величина зсуву кратна довжині ключа, то співпаде понад 6 % байтів. Якщо ні, то будуть збігатися менше ніж 0,4 % (вважаючи, що звичайний ASCII текст кодується випадковим ключем). Це називається показником збігів. Мінімальне зміщення від одного значення, кратного довжині ключа, до іншого і є довжиною ключа.

2. Зміщуємо шифротекст на цю довжину і проведемо операцію XOR для зміщеного і оригінального шифротексту. Результатом операції буде видалення ключа та отримання відкритого тексту, підданого операції XOR з самим собою, зміщеним на довжину ключа.

Оскільки в англійській мові на один байт доводиться 1,3 біта дійсної інформації, існуюча значна надмірність дозволяє визначити спосіб шифрування. Незважаючи на це, велика кількість постачальників ПЗ все ще нав'язують цей алгоритм в якості майже такого ж безпечного як DES.

Тепер розглянемо алгоритм підрахунку «індексу збігу» [320].

Повернемося до алгоритму злому XOR-шифрування. Після того, як ми визначили довжину ключа, необхідно знайти сам ключ. Для цього виконуємо наступні кроки [353]:

Крок 1. Розбиваємо текст повідомлення на групи символів.

Крок 2. Кількість груп дорівнює довжині ключа. Наприклад, якщо у нас шифротекст «*longitude*» і ключ довжиною 3 символи, то в першу групу потраплять букви *l, o, n*; в другу: *g, i, t*; а в третю – *u, d, e*).

Крок 3. У кожну групу входять символи, які кодуються *i*-м символом ключа.

Крок 4. У кожній з груп виконуємо бітову операцію XOR із символами з найбільш поширеним в алфавіті символом (для англійської мови це пробіл).

Крок 5. Вважаємо частоти кожного отриманого символу серед групи.

Крок 6. Вибираємо в якості *i*-ї букви ключа елемент з найбільшою частотою входження.

Слідуючи такому простому алгоритму дій, можна провести злам шифротексту, який був зашифрований за допомогою функції XOR, й дізнатися зміст повідомлення.

Тому необхідно оцінити час, за який комп'ютерна система рухомого об'єкта має встигнути до зламу не тільки зашифрувати, а й передати та потім знищити з власних носіїв дані, отримані з датчиків та/або відеокамер.

Крім того, необхідно вживати додаткових заходів для забезпечення конфіденційності інформації, що циркулює між об'єктами РММ.

4.2 Упровадження методу стеганографічного закриття в телекомунікації БПЛА на основі Open Source протоколу MAVLink

У швидкодинамічних гетерогенних мережах відправником даних може виступати БПЛА, який передає дані наземним або літальним об'єктам (серверам або клієнтам) РММ. Така мережа може знаходитися в різних станах – як відкритої передачі/прийому даних, так і закритої передачі/прийому інформації з обмеженим доступом.

Для передачі даних між вузлами РММ найчастіше використовується телекомунікаційний протокол MAVLink (Micro Air Vehicle Communication Protocol), доступний за ліцензією LGPL [113]. MAVLink – це відкритий комунікаційний протокол, який підтримуються більшістю відкритих польотних контролерів: APM/Pixhawk, MultiWii, Navio та ін. За допомогою MAVLink можна відправляти керуючі команди, задавати місії і отримувати телеметрію. З часу своєї появи його почали використовуватися не тільки для мікроавіації (copter, plane тощо), але і для управління мікромашинами, катерами та навіть підводними човнами. Залежно від типу транспортного засобу та використовуваного контролера, функціонал дещо відрізняється.

Найчастіше у ролі корисної інформації виступає відеосигнал і GPS-координати БПЛА, які передаються у РММ засобами протоколу MAVLink. Однак, зазначений протокол має недостатній рівень захисту [24; 44], тому інформація, що передається з його використанням, може бути перехоплена сторонніми особами, а сам дрон – знищений.

Метою є розробка стеганографічного методу вбудовування GPS-координат БПЛА у відкриті дані, які передаються за допомогою протоколу MAVLink.

У якості вхідних даних обрані кліматичні дані (температура, тиск, вологість, швидкість вітру, освітленість тощо), а також зашумленість ефіру, напруженість електромагнітного поля та інші, передача яких здійснюється пакетами розміром 256 бітів. На борту БПЛА знаходиться не менше 8 датчиків для вимірювання параметрів навколишнього середовища,

За допомогою методу стеганографічного закриття у вхідні дані впроваджуються GPS-координати, розмір яких становить 128 бітів.

Координати записуються у вигляді числа з плаваючою комою за стандартом IEEE 754-2008 (рис. 4.7).

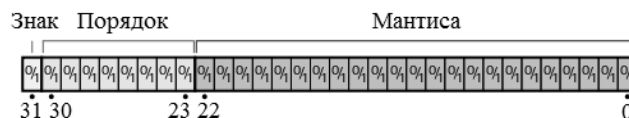


Рис. 4.7. Структура дійсного числа за стандартом IEEE 754-2008:
«знак» – додатне або від'ємне число; «порядок» – ступінь числа;
«мантиса» – саме число

Для реалізації методу стеганографічного закриття GPS-координат у кожного значення вхідних даних забирається 8 молодших бітів під стежоконтейнер (рис. 4.8). При цьому похибка значення становить $3,04 \times 10^{-5} \%$, що практично не впливає на коректність вхідних даних.

КЛІМАТИЧНІ ДАНІ



Рис. 4.8. Структура стежоконтейнера в кліматичних даних

В процесі розробки зазначеного методу були виявлені дві проблеми:

- 1) недостатній розмір стежоконтейнера (який дорівнює 64 біта);
- 2) недостатня захищеність даних при визначенні наявності стежоконтейнера.

Перша проблема вирішується шляхом поділу GPS-координат на 2 пакети по 64 біта і додаванням лічильника парності, за яким можна визначити, що GPS-координати були отримані повністю. Спочатку передаються широта та довгота, потім швидкість пересування. Щоб зрозуміти, коли прийшли широта та довгота, а коли – швидкості, був доданий лічильник парності (рис. 4.9). Якщо він парний, то прийшов перший пакет, інакше – другий.

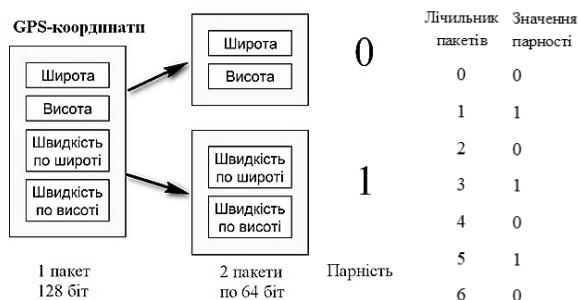


Рис. 4.9. Впровадження лічильника парності в пакет передачі GPS-координат

Другу проблему запропоновано вирішувати за допомогою класичного методу стеганографії (рис. 4.10а), модифікованого шляхом додаткового кодування даних послідовністю випадкових чисел (рис. 4.10б). При цьому створюється послідовність випадкових натуральних чисел, довжина якої дорівнює кількості бітів даних, після чого біти шикуються по даній послідовності та вбудовуються у вихідні дані методом стеганографії [330].

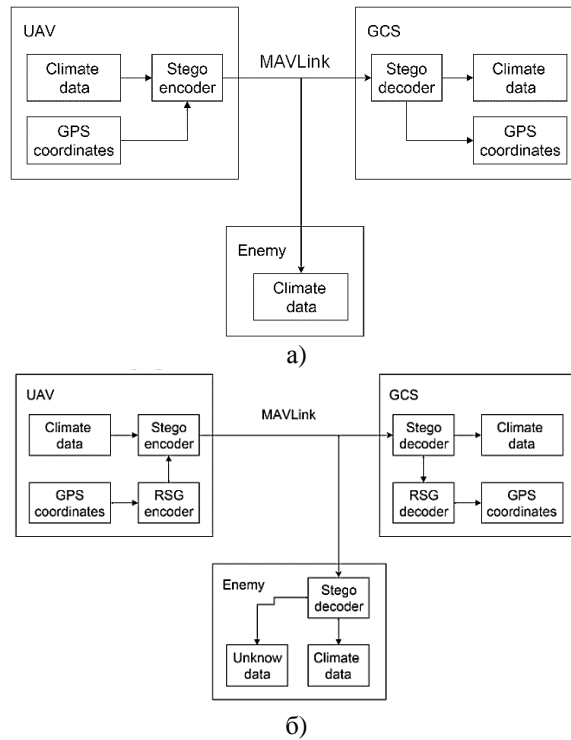


Рис. 4.10. Метод стеганографії класичний (а) та з додатковим кодуванням даних (б)

Зазвичай кліматичні дані передаються з БПЛА відкритим способом. У той же час, дані вимірювання радіаційного, акустичного та електромагнітного фону, а також GPS-координати (тобто, місцезнаходження самого БПЛА), доцільно відправляти з БПЛА на обробку в стегоконтейнері (рис. 4.11).

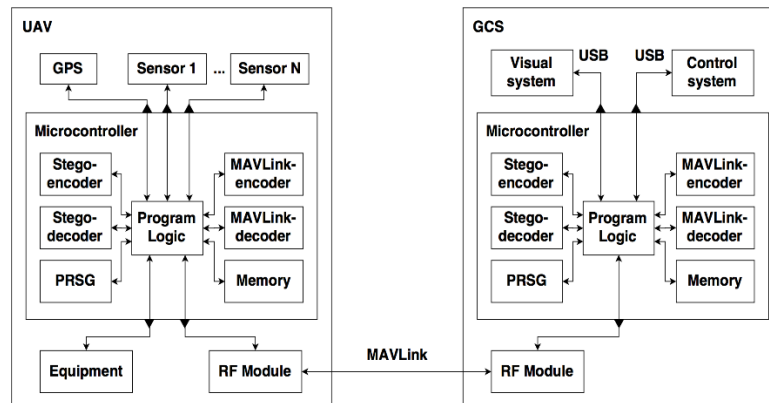


Рис. 4.11. Взаємодія БПЛА з КЦ за протоколом MAVLink

Послідовність генерується і записується в програмний код на етапі прошивки БПЛА (див. рис. 4.11), тим самим обмежуючи можливість отримання інформації про неї сторонніми особами (Enemy). Розпізнати послідовність в адекватні терміни практично неможливо у зв'язку з її великим розміром, який дорівнює 128 натуральним числам.

Оскільки це рішення має малий ступінь складності алгоритмізації, система буде працювати стабільно і практично без затримок, що дозволить власнику (GCS) управляти БПЛА навіть в екстремальних ситуаціях і при великих швидкостях руху РММ.

Упровадження стежоконтейнера у відкриті дані відбувається за наступним алгоритмом:

Крок 1. 8 значень з датчиків створюють послідовність дробових чисел, яка дублюється для надійності даних.

Крок 2. У отриманій послідовності з 16 значень виділяються останні 8 біт даних кожного значення, що в сумі становить 128 біт для стежоконтейнера.

Крок 3. За допомогою випадкової послідовності з 128 цілих чисел від 0 до 128 визначається особливий порядок запису бітів прихованої інформації в стежоконтейнер.

Крок 4. Після цього формується повідомлення.

Крок 5. За допомогою протоколу MAVLink повідомлення пересилається на наземний командний центр або наземному оператору. Там ці дані розкриваються за такою ж послідовністю чисел. Частота передачі даних становить один раз на секунду.

Існує ймовірність, що при виявленні РММ на території, сторонні особи здійнять перехоплення прихованих даних і захочуть розкрити їх. Вищенаведений алгоритм стеганографії дуже простий. Тому достатньо буде менше півгодини для здійснення стегоаналізу прихованих даних, якщо будуть застосовані спеціальні інструменти форензика [36].

У зв'язку з цим запропоновано впровадити у вищеописаний алгоритм стеганографії додатковий алгоритм випадкової послідовності. За цим алгоритмом послідовність бітів повинна бути перетасована випадковим чином.

Алгоритм стеганографії з впровадженням випадкової послідовності буде доповнений такими кроками:

Крок 3а. Обрати ключ, за яким будується випадкова послідовність (*seed*).

Крок 3б. Створити послідовність від 1 до n (*seq*).

```
while(s<6){
  i = 0;
  t = seed+(6-s)*s;
  while(i<size){
    if(t&27){
      tmp = seq[i];
      i2 = i*s+t;
      i2 %= size;
      seq[i] = seq[i2];
      seq[i2] = tmp;
      t = roll_bits(t);
    }
    i++;
  }
  s++;
}
```

Усього бітів даних 128, і за формулою перестановок при спробі стегоаналізу перехоплювачу знадобиться перебрати $128!$ варіантів послідовності чисел, тобто більше $3,85462 \times 10^{215}$ перестановок. Таким чином, на криптоаналіз буде витрачено часу набагато більше ніж час актуальності даних, що передаються.

На додачу до використання стеганографії, захищеність каналів зв'язку між БПЛА підвищується також завдяки передачі інформації на основі протоколу MAVLink. Цей протокол може підтримувати зв'язок у мережі з кількістю БПЛА у зграї до 255 одиниць [111].

Таким чином, застосування запропонованого рішення дозволяє приховати місцезнаходження БПЛА, й тим самим збільшити термін його життя у зв'язку з неможливістю визначити його місцезнаходження сторонніми особами у разі перехвату трафіку UAV – GCS.

Натурне моделювання стеганографічного вирішення задачі можливо провести на основі двох різних апаратних рішень:

- мікроконтролера ATmega32 (рис. 4.12а) з характеристиками: пам'ять 32 кбайт, тактова частота 8 МГц, розрядність процесора 8 бітів, а також системна плата та програматор до нього,

- плати STM32F429 Discovery (рис. 4.13) з характеристиками: пам'ять 2 Мбайт, тактова частота 16 МГц, розрядність процесора 32 бітів.

Плата STM32F429 Discovery має вже вбудований програматор ST-LINK, тому програматор має бути обраний тільки для ATmega32. Був обраний найпоширеніший (для цього виду мікроконтролерів) програматор USBasp v2.0 (рис. 4.12б).

У якості пристроїв комунікації у контролерів ATmega32 та STM32F429 Discovery використовуються вбудовані модулі USART (Universal Synchronous and Asynchronous Serial Receiver and Transmitter) для передачі даних. Вони працюють незалежно від коду програми, таким чином мають велику стійкість від помилок.

Кінцевий вигляд макета для натурного моделювання наведений на рис. 4.14.

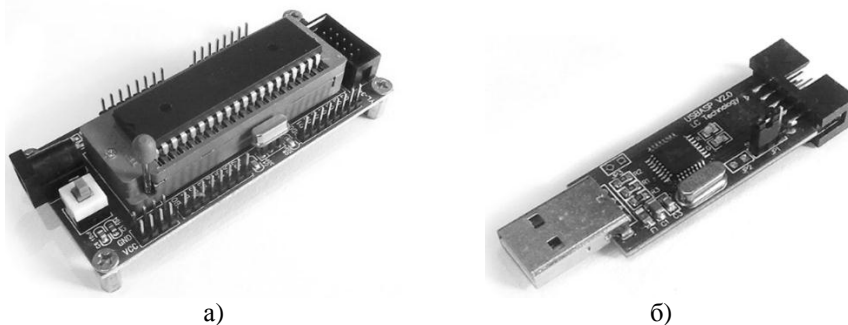


Рис. 4.12. Апаратні засоби для натурного моделювання стеганографічного рішення на основі контролера ATmega32 (а) з програматором USBasp v2.0 для ATmega32 (б)



Рис. 4.13. Апаратні засоби для натурного моделювання стеганографічного рішення на основі плати STM32F429 Discovery

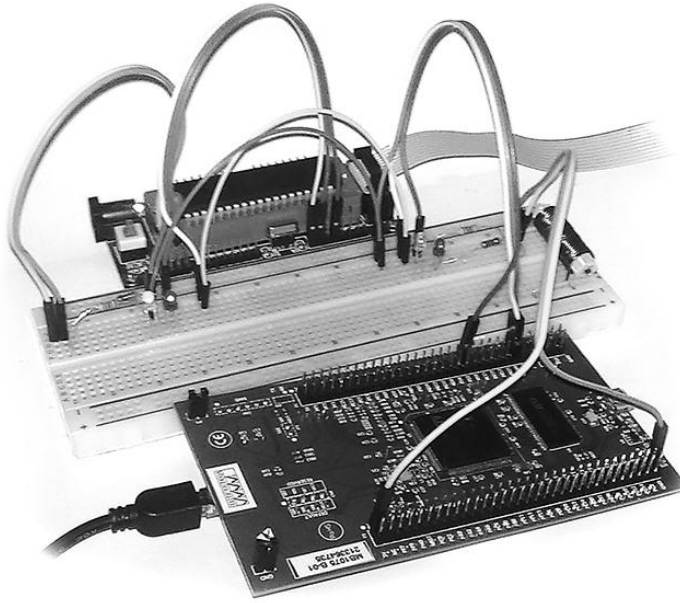


Рис. 4.14. Кінцевий вигляд макета для натурного моделювання

Для роботи системи потрібно апаратні модулі з'єднати між собою. Для цього можна використати кабельні або бездротові з'єднання – RF (Radio Frequency) передавач/приймач (рис. 4.15).

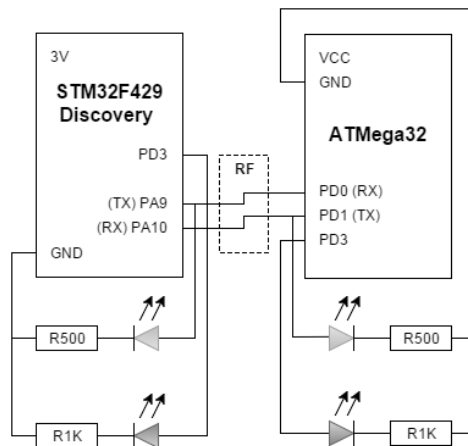


Рис. 4.15. Схема підключення модулів апаратної частини

Зелені індикатори (зверху) показують передачу даних, а червоні (знизу) – що дані були прийняті. Червоний індикатор UAV блимає з частотою приблизно 40 Гц, тому око не може це побачити. Індикатор GCS мигає у два рази рідше – тобто 20 Гц, – і його блимання можна побачити.

Після остаточного вибору апаратних засобів, встановлюються драйвери для цих програматорів на персональний комп'ютер. Для програматора ST-LINK драйвер завантажується з офіційного сайту STMicroelectronics [166], а для програматора USBasp v2.0 драйвер можна завантажити за посиланням [184].

Для запуску процесу натурального моделювання стеганографічного рішення передачі даних в PMM необхідно налаштувати деякі параметри мікроконтролера та плати:

1. Додати заголовки до проекту:

а) для ATmega32:

```
#include <avr/io.h>
#include <avr/interrupt.h>
```

б) для STM32F429 Discovery:

```
#include "stm32f4xx.h"
#include "stm32f4xx_gpio.h"
#include "stm32f4xx_rcc.h"
#include "stm32f4xx_usart.h"
```

2. Налаштувати порти USART (*baud_b* – швидкість обміну даними):

а) для ATmega32:

```
UBRRH = baud_b>>8;
UBRRL = baud_b&0xFF;
UCSRB = (1<<TXEN) | (1<<RXEN) | (1 << RXCIE );
UCSRC = (1<<URSEL) | (1<<UCSZ1) | (1<<UCSZ0);
```

б) для STM32F429 Discovery:

```
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA,
ENABLE);
GPIO_InitTypeDef gpio;
gpio.GPIO_Mode = GPIO_Mode_AF;
gpio.GPIO_Pin = GPIO_Pin_9 | GPIO_Pin_10;
gpio.GPIO_Speed = GPIO_Speed_2MHz;
gpio.GPIO_OType = GPIO_OType_PP;
gpio.GPIO_PuPd = GPIO_PuPd_UP;
GPIO_Init(GPIOA, &gpio);
GPIO_PinAFConfig(GPIOA, GPIO_PinSource9,
GPIO_AF_USART1);
GPIO_PinAFConfig(GPIOA, GPIO_PinSource10,
GPIO_AF_USART1);

RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1,
ENABLE);
USART_InitTypeDef usart;
usart.USART_StopBits = USART_StopBits_1;
usart.USART_WordLength = USART_WordLength_8b;
usart.USART_Mode =
USART_Mode_Rx|USART_Mode_Tx;
usart.USART_BaudRate = speed;
usart.USART_Parity = USART_Parity_No;
usart.USART_HardwareFlowControl =
USART_HardwareFlowControl_None;
USART_Init(USART1, &usart);
USART1->BRR = baud_b*15;
USART_Cmd(USART1, ENABLE);
```

3. Активізувати системні переривання:

а) для ATmega32:

```
sei();
```

б) для STM32F429 Discovery:

```
__enable_irq ();
USART_ITConfig(USART1, USART_IT_RXNE, ENABLE);
NVIC_EnableIRQ(USART1_IRQn);
```

Тепер можна передавати дані. При надходженні даних викликається системне переривання:

а) для ATmega32:

```
ISR(USART_RXC_vect){
    if(uart_reciever)
        uart_reciever(UDR);
}
```

б) для STM32F429 Discovery:

```
void USART1_IRQHandler(){
    if ((USART1->SR&USART_FLAG_RXNE) != RESET){
        if(uart_reciever)
            uart_reciever(USART1->DR);
    }
    USART_ClearITPendingBit(USART1,
    USART_IT_RXNE);
}
```

Функція *mavlink_parse_char* отримує прийнятий байт з іншої одиниці системи та записує її у власний буфер. Якщо всі байти повідомлення прийшли коректні, функція повертає додатне значення, що свідчить про успішність отримання даних.

Функції *heartShow* та *gpsShow* посилають на 3-й порт сигнал, який повідомляє про те, що дані були успішно отримані.

GCS кожні 10 мс посилає пустий пакет *heartbeat* до UAV. В свою чергу, UAV, після отримання цього пакету, надсилає GCS-дані про свої GPS-координати.

Дані передаються на частоті 433 МГц (входить в діапазон LPD433) або на іншій частоті ISM-діапазону.

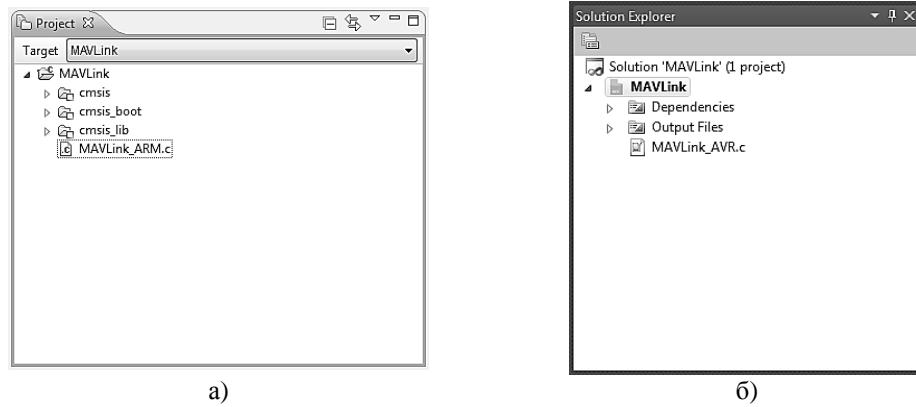
Для комп'ютерного моделювання стеганографічного рішення існують різноманітні середовища розробки, але не всі мають змогу працювати з тим чи іншим типом мікроконтролерів та плат. Тому для обраних апаратних засобів найкращими середовищами розробки є CoIDE компанії CoCoX для плати STM32F429 Discovery, та AVR Studio 5.0 компанії Atmel для мікроконтролеру ATmega32 (рис. 4.16).

Завантажити їх також можна за посиланнями:

а) CoIDE [38];

б) AVR Studio 5.0 [16].

Після завантаження і встановлення середовищ розробки, в кожному з них створюється проект. Вони зберігаються в одній папці для спрощення доступу до спільних файлів (рис. 4.17).



а) б)
Рис. 4.16. Проекти в CoIDE (а) та AVR Studio 5.0 (б)

Имя	Тип	Размер
cmsis	Папка с файлами	
cmsis_boot	Папка с файлами	
cmsis_lib	Папка с файлами	
syscalls	Папка с файлами	
componentfile.history	Файл "HISTORY"	1 КБ
MAVLink.avrgccproj	AVR GCC Project File	5 КБ
MAVLink.avrsln	AVR Studio 5 Solution File	1 КБ
MAVLink.avrsuo	Файл "AVRSUO"	26 КБ
MAVLink.cogui	Файл "COGUI"	26 КБ
MAVLink.comarker	Файл "COMARKER"	1 КБ
MAVLink.coproj	CoIDE Project	6 КБ
MAVLink_ARM.c	Файл "C"	2 КБ
MAVLink_AVR.c	Файл "C"	2 КБ

Рис. 4.17. Структура папки проекту

Наступним етапом є підключення протоколу MAVLink та створення повідомлення.

MAVLink є одночасно протоколом для комунікації між GCS та UAV, або UAV – UAV, та header-only C бібліотекою (містить тільки заголовки).

Спочатку скачується бібліотека MAVLink для мови програмування C за посиланням [112]. Зберігається ця бібліотека також в папці проектів для швидкого доступу (рис. 4.18). Підключається вона у налаштуваннях проекту в розділі «Compiler – Include Paths». Необхідно вказати директорію, в якій знаходиться завантажена бібліотека.

Имя	Тип	Размер
ardupilotmega	Папка с файлами	
ASLUAV	Папка с файлами	
autoquad	Папка с файлами	
common	Папка с файлами	
matrixpilot	Папка с файлами	
message_definitions	Папка с файлами	
minimal	Папка с файлами	
slugs	Папка с файлами	
test	Папка с файлами	
ualberta	Папка с файлами	
checksum.h	Файл "H"	3 КБ
mavlink_conversions.h	Файл "H"	7 КБ
mavlink_helpers.h	Файл "H"	22 КБ
mavlink_types.h	Файл "H"	9 КБ
protocol.h	Файл "H"	13 КБ

Рис. 4.18. Бібліотека MAVLink для мови програмування C

Для створення повідомлення завантажується генератор MAVLink-повідомлень [112]. Він написаний на Python, тому також встановлюється інтерпретатор Python, який можна скачати з офіційного сайту, та запускається файл mavgenerate.py (рис. 4.19).

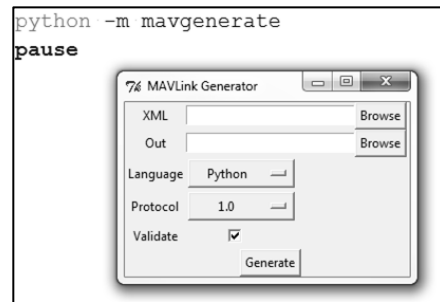


Рис. 4.19. Генератор MAVLink повідомлень

Перший етап формування повідомлення – це створення XML-файлу, в якому записана структура повідомлення. Для даних про клімат була створена наступна структура:

```
<?xml version="1.0"?>
<mavlink>
<include>common.xml</include>
  <enums>
  </enums>
  <messages>
    <message id="200" name="SENSORS_DATA">
      <field type="uint32_t"
name="count"></field>
      <field type="float"
name="pressure"></field>
      <field type="float"
name="temperature"></field>
      <field type="float"
name="humidity"></field>
      <field type="float"
name="illumination"></field>
      <field type="float"
name="noisiness"></field>
      <field type="float"
name="wind_speed_x"></field>
      <field type="float"
name="wind_speed_y"></field>
      <field type="float"
name="em_intensity"></field>
    </message>
  </messages>
</mavlink>
```

Наведена структура містить в собі дані про тиск, температуру, вологість, освітленість, рівень шуму, швидкість вітру, напруженість електромагнітного поля, а також лічильник парності, який надалі знадобиться для стеганографії.

Підключення файлу common.xml необхідно для того, щоб у згенерованій бібліотеці були присутні всі стандартні види повідомлень MAVLink.

ID-повідомлення є найважливішою частиною структури, адже за цим значенням протокол MAVLink визначає, яке повідомлення було отримано. Назва створеного повідомлення SENSORS. Всі параметри, окрім лічильника – дійсні числа, тому що

показники клімату мають дійсну природу. Також це знадобиться для реалізації стеганографічного рішення [329; 330].

Другим етапом створення повідомлення є компіляція XML-файлу. У вікні генератора MAVLink обирається XML-файл, а в якості папки для виводу обирається папка, де знаходиться C бібліотека MAVLink (рис. 4.20). Обирається мова програмування C та протокол MAVLink версії 1.0, після чого натискається кнопка «генерувати». Нова бібліотека з новим типом повідомлення створена.

Имя	Тип	Размер
ardupilotmega	Папка с файлами	
ASLUAV	Папка с файлами	
autoquad	Папка с файлами	
common	Папка с файлами	
matrixpilot	Папка с файлами	
message_definitions	Папка с файлами	
minimal	Папка с файлами	
slugs	Папка с файлами	
test	Папка с файлами	
ualberta	Папка с файлами	
checksum.h	Файл "H"	3 КБ
mavlink_conversions.h	Файл "H"	7 КБ
mavlink_helpers.h	Файл "H"	22 КБ
mavlink_types.h	Файл "H"	9 КБ
protocol.h	Файл "H"	13 КБ

а)

б)

Рис. 4.20. Структура бібліотеки до компіляції (а) та після компіляції (б)

Тепер до проекту треба додати ініціалізацію MAVLink:

```
#include <sensors_data/mavlink.h>

static mavlink_system_t mavlink_system;
static mavlink_message_t msg;
static mavlink_sensors_data_t data;
static gps_data gps;
static uint8_t buf[MAVLINK_MAX_PACKET_LEN];
static uint16_t buflen;

int main(){
    mavlink_system.sysid = 200;
    mavlink_system.compid = MAV_COMP_ID_IMU;

    /* налаштування даних для gps та data */

    mavlink_msg_sensors_data_encode(    mavlink_system.sysid,
                                       mavlink_system.compid,
                                       &msg, &data);
    buflen = mavlink_msg_to_send_buffer(buf, &msg);

    /* main loop */
}
```

Функція *mavlink_msg_sensors_data_encode* кодує дані типу *sensors_data* у повідомлення, а потім це повідомлення кодується у потік байтів функцією *mavlink_msg_to_send_buffer*.

Структура *gps_data*:

```
typedef struct __gps_data{
float latitude;
float longitude;
float vlatitude;
float vlongitude;
} gps_data;
```

Таким чином, були створені повідомлення та дані про клімат та GPS-координати. Після цього вони запаковуються та відправляються у буфер, який буде використовуватися для відправлення даних до інших компонентів системи.

На наступному етапі створення зазначеної системи необхідно провести аналіз швидкості роботи та передачі даних в системі.

На швидкість роботи розробленої системи мають вплив:

- швидкість процесора;
- швидкість USART;
- обсяг даних, що передається;
- кількість операцій для кодування та декодування даних.

Як вже було зазначено, тактова частота процесора ATmega32 8 МГц, а STM32F429 Discovery – 16 МГц. Модулі системи можуть взаємодіяти тільки за найнижчим показником, тому для обрахунків береться 8 МГц.

Швидкість USART також не може перевищувати швидкість процесора, тому, після здійснення багатьох спроб налаштування швидкості, була визначена максимальна швидкість, при якій зв'язок залишається стабільним – 55 кбіт/с.

Один пакет повідомлення MAVLink містить $8 + n$ байтів даних, де n – довжина самого повідомлення. Передаються 9 значень по 4 байти кожне, тому один пакет має розмір $8 + 9 \times 4 = 44$ байти = 352 біта. Для порожнього пакета це буде 16 байтів = 128 бітів.

За результатами натурального моделювання визначено, що для створення пакета з кліматичними даними та стежоконтейнером у вигляді GPS-координат необхідно приблизно 30 тис. операцій, а для зчитування цих даних – 40 тис. операцій. Для створення порожнього пакета необхідно 25 тис. операцій, а для зчитування – 35 тис. операцій.

Ураховуючи вищенаведене, можливо обрахувати швидкість роботи системи (табл. 4.1):

Таблиця 4.1. Обрахування часу створення та передачі повідомлення (поетапно)

Назва етапу	Кількість операцій на етапі	Швидкість виконання етапу, кбіт/с	Час виконання етапу, с
Створення порожнього пакета	25000	8000	3,1
Відправлення порожнього пакета	128	55	2,3
Зчитування порожнього пакета	35000	16000	2,2
Створення кліматичного пакета	30000	16000	1,9
Відправлення кліматичного пакета	352	55	6,4
Зчитування кліматичного пакета	40000	8000	5,0
Сумарний час			20,9

Таким чином, на весь цикл обробки одного повідомлення витрачається час 20,9 мс. З урахуванням можливих затримок або помилок, округлимо це значення до 25 мс. Тоді, за 1 с буде передано 40 повідомлень. Але, з урахуванням того, що GPS-координати

передаються за 2 пакети, швидкість отримання GPS-координат дорівнює 20 разів за секунду.

Система була протестована на стабільність зв'язку. Якщо перервати зв'язок на певний час, а потім встановити його знову – система продовжуватиме працювати, тобто вона є достатньо стабільною.

Висновки до розділу 4

Під час виконання дослідження було проаналізовано блокові алгоритми кодування інформації й віднайдені основні риси, які характерні для кожного з них. Сформовано фундаментальні правила для методу, який спроможний забезпечити належний рівень захисту конфіденційної інформації, що передається в умовах неможливості забезпечення охорони периметра рухомої мережі (КФС).

Проаналізовано методи, які створюються шляхом комбінування операцій, простих для обчислювального процесу як ЕОМ, так і мікроконтролерних пристроїв. До таких операцій можна віднести операції зсуву, додавання, перемішування, логічного виключення.

У результаті роботи було запропоновано новий модифікований блоковий метод шифрування з використанням операції XOR.

Розроблений метод було покладено в основу програмного застосунку з шифруванням повідомлень, який забезпечує захищений обмін повідомленнями між територіально розосередженими об'єктами однієї СРС. Програма є кросплатформною, протестована для ОС з розрядністю x64, та x32.

Встановлено, що запропонований підхід дозволяє підвищити криптостійкість шифрованої інформації при зменшеному часі шифрування, у т.ч. на процесорах з низькою обчислювальною потужністю, які використовуються у БПЛА.

Використання такого «легкого» методу шифрування в мікроконтролерах малих БПЛА надає змогу розробити рухомий об'єкт невеликого розміру. Передусім, суттєво збільшує час виявлення БПЛА в умовах розвідувальних дій та підвищує шанс, що літальний апарат вийде неушкодженим з території противника [3]. Крім того, невеликі розміри об'єкта роблять можливим здешевити його виготовлення.

Організація сеансу зв'язку була виконана у виродженій підмережі «точка-точка» в межах локальної мережі однієї території, розміри якої не перевищують дальність каналів безпосереднього зв'язку між КФО, а також між СРО, розташованими у різних підмережах, поєднаних через Інтернет.

У останньому випадку КФО можуть бути членами однієї КФС, яка територіально розташована не тільки у різних містах, а навіть у різних країнах.

Для обміну повідомленнями через мережу Інтернет в цьому випадку має бути виконане налаштування відповідних параметрів маршрутизаторів, що використовуються для підключення до мережі Інтернет.

Створений застосунок не залишає даних про сеанс зв'язку в мережі Інтернет за рахунок встановлення прямого зв'язку між двома об'єктами КФС без проміжних серверів, які могли б зберігати дані та історію сеансів зв'язку.

Проведено аналіз на стійкість до злому криптосистеми, яка базується на алгоритмі швидкого шифрування XOR. Отримано співвідношення часу шифрування розробленим методом блоку даних для передачі іншому об'єкту мережі з часом, необхідним для зламу бортової системи БПЛА сучасним обладнанням

Для підвищення криптостійкості повідомлень у РММ розроблено стеганографічний метод вбудовування GPS-координат БПЛА у відкриті дані, які передаються за допомогою протоколу MAVLink.

Проведено натурне та комп'ютерне моделювання розробленої системи зі стеганографічним закриттям інформації. Визначений час повного циклу створення та обробки повідомлення у каналі зв'язку РММ в залежності від обраних комп'ютерних компонентів.

Запропоновані методи мають достатньо високу криптостійкість. Використання цих методів сприятиме подовженню життєвого циклу РММ.

Розділ 5

МОДЕЛЮВАННЯ ГРУПОВОЇ ПОВЕДІНКИ ОБ'ЄКТІВ РУХОМОЇ МОНІТОРИНГОВОЇ МЕРЕЖІ

5.1 Методи моделювання поведінки БПЛА у зграї

Під час моделювання функціонування РММ доцільно вивчати поведінку не окремих об'єктів, а одно-/різнотипних БПЛА у складі суб-рою, а суб-роїв – у складі зграї. Прийнято розподіляти дрони між суб-роями за територіальною ознакою або технічним призначенням таких дронів.

При формуванні суб-роїв за територіальною ознакою кожен суб-рій виконує завдання на спільній території. Зграя з таких суб-роїв обслуговує певну територію, поділену на сектори, власні для кожного суб-рою.

По-перше, задачею моделювання є попередження критичних ситуацій у випадку зіткненні дронів суб-рою один з одним (рис. 5.1).



Рис. 5.1. Креш БПЛА при зіткненні

По-друге, доцільно змоделювати послідовне переміщення БПЛА за маршрутом, що математично приводиться до методів вирішення задачі комівояжера (рис. 5.2). Ураховуючи обмежені можливості дрона знаходитись у польоті (обмежену ємність батареї), необхідно на кожному з маршрутів визначати довжину та час шляху кожного дрона.

Під час спільної роботи дронів у суб-рої кожен має знати відсоток виконання спільного завдання, тобто ту частину роботи, яка вже виконана іншими дронами. Як описано у попередніх розділах, для самонавчання такої мережі можливо використання НМХ. Але тоді необхідно визначати та враховувати при реалізації задачі комівояжера достатньо велику похибку у розрахунках оптимального шляху [126; 351].

Такий підхід особливо ефективний, коли кожен дрон достатньо примітивний та не виділяється серед інших. У такому випадку індивідуальні траєкторії й характеристики окремих об'єктів стають несуттєвими. На перший план виходять колективні типи рухів, що забезпечують безаварійне переміщення дронів, та колективні параметри, які визначають відсоток виконання всього завдання зграєю або кожним суб-роєм окремо.

Для моделювання зв'язків між ними доцільно використовувати математичний апарат статистичної фізики.

5.2 2D-моделювання

Для управління кількома дронами на великій території (згідно з мапою) вся територія може бути розбита на сегменти, у кожному з яких працює окремий суб-рій зграї (рис. 5.2). Вважається, що дрони з різних суб-роїв не перетинають межу власного сегмента території.

У такому випадку наземний оператор задає точки, які має відвідати безпілотник у межах свого сегмента. Оператор це може зробити, задавши безпосередньо координати точок або відмітивши їх на мапі.

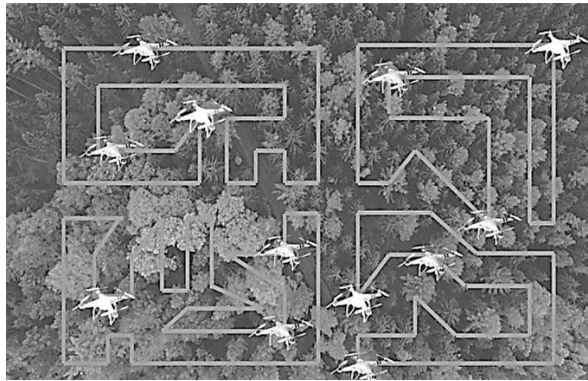


Рис. 5.2. Розподіл території на чотири сектори для кожного суб-рою з трьох БПЛА

Кожну точку дрон має відвідати один раз. Кожну точку може відвідати лише один дрон або кожен дрон суб-рою.

Перевагою використаного методу є, коли дрони суб-рою у кожний момент часу знають про місцезнаходження одне одного.

Нехай задано місцевість для обстеження її з допомогою БПЛА. На мапу місцевості наноситься координатна сітка і задається точка відліку (рис. 5.3).



Рис. 5.3. Карта місцевості з нанесеною системою координат

Якщо оператор хоче, щоб БПЛА зробив знімки/вимірювання у трьох вузлах місцевості $(0;0)$, $(0,3;1)$, $(1;0,5)$, достатньо одного БПЛА (рис. 5.4). Тоді існує лише один маршрут, який є найкоротшим. Проте, якщо додати ще один вузол – наприклад $(1;1,5)$, –

то маршрутів стає значно більше, а точніше $e \frac{(n-1)!}{2}$ маршрутів, де n – кількість вузлів у маршруті.

Якщо площа обстежуваної території і кількість обстежуваних вузлів зростатиме, на перший план виходить обмежений час польоту БПЛА. Такі обмеження накладає не тільки ємність батареї БПЛА, а також метеумови (швидкість вітру, температура середовища тощо). У такому разі кількість обстежуваних сегментів території визначає кількість використовуваних суб-роїв у зграї. Кількість БПЛА у суб-рою визначається площею сегмента.

Як показано у попередніх розділах, у всіх БПЛА одного суб-рою точка старту може бути спільною. Інакше для кожного БПЛА визначається власна точка (рис. 5.4), до якої він повертається після виконання завдання або на підзарядку акумуляторної батареї.

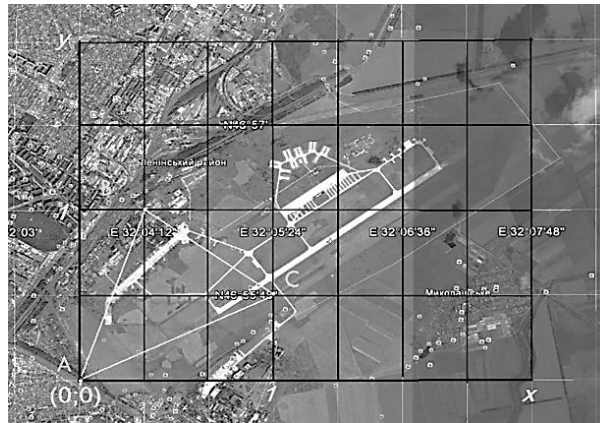


Рис. 5.4. Карта місцевості з нанесеними точками старту трьох БПЛА одного суб-рою

Отже, описане завдання є не чим іншим, як вирішенням задачі комівояжера [288; 290; 299].

Як математична проблема, задача комівояжера відома з 19-го століття. Історично вона полягає у тому, який маршрут має обрати комівояжер (торговець-постачальник), щоб об'їхати всі точки призначення, обравши найкоротший шлях і жодного разу не опинитися в одному місці двічі [290].

Задача комівояжера – одне з найвідоміших завдань комбінаторної оптимізації, що полягає у пошуку найвигіднішого маршруту, який проходить через вказані точки хоч би по одному разу з подальшим поверненням у початкову [299]. В умовах завдання вказується критерій вигідності маршруту (найкоротший, найдешевший, сукупний критерій і т. п.) і відповідні матриці відстаней, вартості і т. ін. Як вже зазначалось, вважатимемо найкращим маршрут найкоротший, а кожен вузол має бути відвіданим лише один раз.

Нейронна мережа Хопфілда може використовуватися для вирішення завдання комівояжера. Для цього НМХ має відповідати певним вимогам:

Мережа повинна складатися з $N = n \times n$ нейронів, які ми розглядатимемо як квадрат з n рядків і n стовпців [351].

1. Відповідь мережі міститиме тільки один активний нейрон в кожному рядку і кожному стовпці.
2. Активний нейрон у першому стовпці задає першу точку маршруту, в другому – другу точку, і так далі.

Усі ці умови задовольняє формула (5.1) обчислення ваги між нейроном, який відповідає точці x на позиції в i -му маршруті, і нейроном, що відповідає точці y на позиції j [126].

$$W_{xi,yi} = -A\delta_{x,y}(1 - \delta_{i,j}) - B\delta_{i,j}(1 - \delta_{x,y}) - Cd(x,y) \times (\delta_{i,j+1} + \delta_{i,j-1}) + D, \quad (5.1)$$

де A, B, C, D – деякі константи;

$d(x, y)$ – відстань між точками x і y ;

$\delta_{x,y}$ – символ Кронекера, що набуває значення «1», якщо $x = y$ і значення «0» інакше.

Як можна побачити, перший член дорівнює ($-A$) для всіх зв'язків в тому ж рядку ($x = y$), окрім зв'язку нейрона з самим собою (при $i = j$). Другий дорівнює ($-B$) для всіх зв'язків у тому ж стовпці ($i = j$), окрім зв'язку з самим собою ($x = y$). Третій пропорційний відстані між точками x і y , якщо ці місця сусідні в маршруті ($i = j - 1$ чи $i = j + 1$).

Якщо таку мережу привести у випадковий початковий стан, то можна очікувати, що результуючий стабільний стан дасть нам суб-оптимальний шлях, довжина якого не занадто перевершує оптимальну (сам шлях може значно відрізнятись від оптимального). Відповідно, для практичного застосування мережу варто запустити кілька разів.

За результатами розрахунків можна обрати найкращий шлях [282].

Але завдання може ставитись інакше. Наприклад, для розрахунку середнього значення необхідно зробити декілька однакових вимірювань у заданих точках. Таке завдання доцільно виконувати за допомогою суб-рою БПЛА, для яких згенеровані різні маршрути через точки з такими ж координатами місцевості. Тоді кожен БПЛА у суб-рою рухається за власним згенерованим маршрутом.

Наприклад, кожен БПЛА має облетіти територію, відвідавши 12 вузлів (рис. 5.3): (1;0), (0,4;0,44), (0,3;0,14), (1,2;0,23), (0,22;1), (0,51;0,93), (0,87;0,65), (0,68;0,55), (0,85;0,35), (0,67;0,25); (0,61;0,1), (0,91;0,95).

Для вирішення описаної задачі було створено програмний застосунок DRAG'N'DRONE, який генерує маршрути на основі введених оператором точок для обстеження. Порівняння функціонала зазначеного застосунку з промисловими аналогами наведено у табл. 5.1.

Таблиця 5.1. Порівняння функціонала застосунку DRAG'N'DRONE з промисловими аналогами

Назва проекту	Використання сервісів Google	Управління декількома дронами одночасно	Вибір точок на арті безпосередньо	Взаємозв'язок дронів під час польоту
DRAG'N'DRONE	✓	✓	✓	✓
PixiePath	✓	✓	✗	✗
ArduPilot	✗	✓	✗	✗
Alphabet (платформа UTM)	✓	✓	✓	✓

Програма має запускатися стільки разів, скільки унікальних маршрутів необхідно згенерувати відповідно до кількості БПЛА у суб-рою. Результат генерації програмою (в результаті її послідовних викликів на основі одних й тих самих даних) субоптимальних маршрутів для кожного з трьох БПЛА у складі суб-рою наведений на рис. 5.5а – рис. 5.5в.

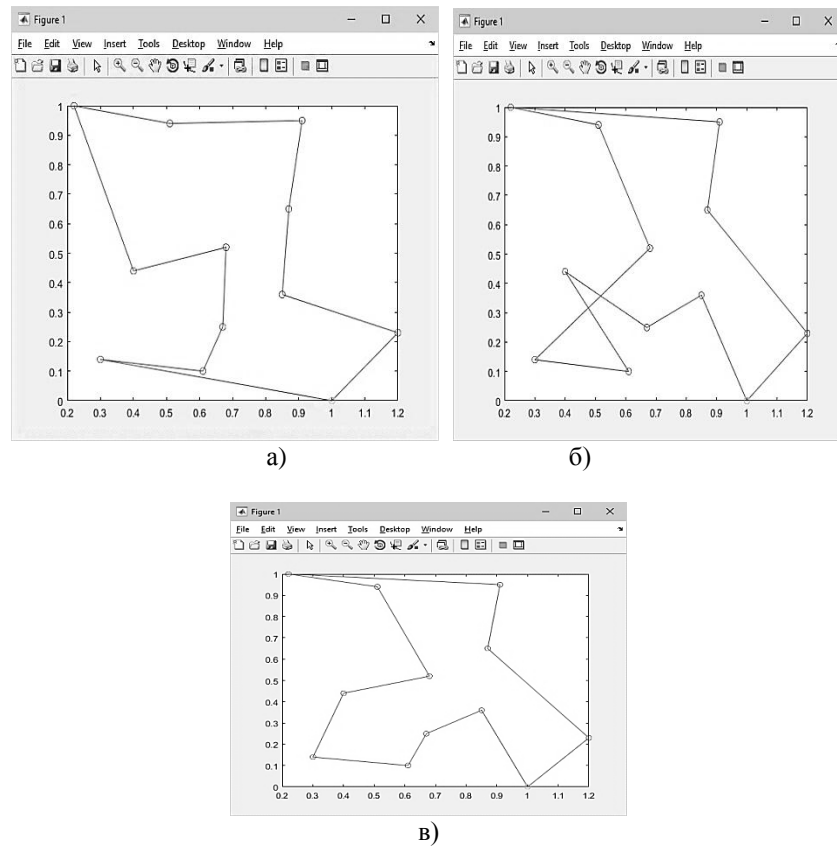


Рис. 5.6. Програмно згенеровані субоптимальні маршрути для кожного з трьох (а, б, в) БПЛА суб-рою

При моделюванні вважається, що:

- оптимальним маршрутом є найкоротший;
- облетівши всі вузли сегмента території, кожен з БПЛА суб-рою має повернутись у власну точку вильоту;
- БПЛА має відвідати кожен вузол лише один раз.

Для попередження подвійного відвідання одного вузлу було обрано НМХ.

Таблиця 5.2. Аналіз згенерованих НМХ маршрутів для трьох БПЛА

№ маршруту	Довжина маршруту L_{ep} , км	Відхилення довжини маршруту	
		абсолютне $ \Delta D $, км	відносне, $\varepsilon = \frac{\Delta D}{D}$, %
1	4,3536	0,0092	2,96
2	4,5278	0,1834	
3	4,1519	0,1925	

З табл. 5.2 видно, що маршрути, згенеровані в результаті виконання програми, мають приблизно однакову довжину, проте абсолютно різний вигляд. Як вже зазначалось, річ у тім, що НМХ може вирішити задачу комівояжера лише приблизно (результат зазвичай трохи довший за ідеальне значення). Тому маршрути, які генеруються з використанням НМХ, завжди приблизно однакові за довжиною, але різні за порядком проходження вузлів.

Ризики, з якими можна зіткнутись:

- декілька БПЛА можуть опинитись в одній точці одночасно;
- висота руху БПЛА може бути занадто малою для маневрування, etc.

5.3 3D-моделювання

Використання двовимірної мапи території для переміщення зграї БПЛА у просторі дозволяє за допомогою системи навігації БПЛА відпрацьовувати його маршрут, що веде до мети. Певну увагу при цьому необхідно приділити оминанню перешкод, які зустрічаються на шляху БПЛА, а також попередити зіткнення БПЛА один з одним.

Задля приближення результатів моделювання до реального переміщення БПЛА необхідно враховувати, що він змінює не тільки власні GPS-координати, а й висоту. Зазначені зміни виконуються та відстежуються навігаційною системою, що використовує GPS/ГЛОНАСС, доповненої засобами системи орієнтації: технічного зору, далекоміром, інфрачервоними датчиками і т. д.

До того ж, сучасні БПА здатні не тільки виконувати команди, які надходять з центру управління та регламентують параметри руху (азимут, швидкість, прискорення), але й самостійно планувати траєкторію з огляду на необхідність вирішувати завдання обходу перешкод та запобігання зіткненням.

Сукупність цих засобів забезпечує рухомий об'єкт (БПА) досить об'єктивною інформацією про поточне місцезнаходження. Досягнення цільового положення здійснюється за певною траєкторією, для попереднього планування якої необхідна інформація про зовнішнє середовище – простір, в якому буде проводитися рух.

Відмінною особливістю стратегічного планування є використання менш детальної карти місцевості ніж при тактичному плануванні. На цьому етапі в якості перешкод можуть розглядатися досить великі об'єкти, наприклад, будівля, болото, крутий гірський схил і т. д. У таких умовах розміри стратегічної карти можуть стати прийнятними для здійснення планування траєкторії безпосередньо бортовим обчислювачем, обчислювальні можливості якого, зрозуміло, не надто великі.

Тому перехід до тактичного планування, з опрацюванням більш детальних мап, виконується лише у разі настання загрози критичного застосування БПА. Тоді використовується більш детальна карта місцевості, яка може бути отримана з різних джерел. На етапі детального планування побудована траєкторія доповнюється і згладжується, враховуючи особливості реалізації транспортного механізму БПА, до яких відносяться, наприклад, найменший радіус повороту, мінімальні і максимальні значення прискорення і швидкості руху, максимально допустима крутизна траєкторії руху й т. ін.

Тривимірний рух БПЛА може бути змодельованим, наприклад, у середовищі MATLAB від компанії The MathWorks [109]. У цьому випадку доцільно використовувати, як вбудовані у зазначений пакет прикладні програми для числового аналізу, так і мову програмування, яка використовується в цьому пакеті. За допомогою MATLAB можуть бути не тільки розраховані траєкторії руху БПЛА, а й відтворена візуалізація взаємного переміщення БПЛА суб-рою у 3D-просторі.

Варто зазначити, що для такого 3D-моделювання може бути використана базова комерційна версія MATLAB без інструментів (за ціною близько 2000 USD) та обмежена версія для навчальних закладів з мінімальним набором інструментів (лише за 100 USD).

Для зазначених потреб у середовищі MATLAB можуть бути використані функції для аналізу даних, зокрема:

- а) матриці та лінійна алгебра – алгебра матриць, лінійні рівняння, власні значення і вектори, сингулярності, факторизація матриць та інше;
- б) многочлени та інтерполяція – корені многочленів, операції над многочленами та їх диференціювання, інтерполяція та екстраполяція кривих;
- в) обробка даних – набір спеціальних функцій, включаючи побудову графіків, оптимізацію, пошук нулів, чисельне інтегрування та інше;
- г) диференційні рівняння – вирішення диференційних і диференційно-алгебраїчних рівнянь, диференційних рівнянь із запізненням, рівнянь з обмеженнями, рівнянь в часткових похідних та інше.

Необхідно зауважити, що в реальних умовах може знадобитись як досить довготривале детальне обстеження території з відеозйомкою, так і використання датчика вимірювання якогось одного параметра у певній координаті місцевості. Тому для моделювання групової поведінки дронів, які виконують різні за складністю завдання у певному сегменті мапи, було обрано дві моделі БПЛА різного рівня складності та потужності (рис. 5.7): DJI Phantom 4 (більш потужний і складний за функціоналом) та UDIRC LarkFPV (простіша модель).



Рис. 5.7. Дрони DJI Phantom 4 (а) та UDIRC LarkFPV (б)

Для моделювання групового руху БПЛА у суб-рою з реалізацією механізму запобігання зіткненням та своєчасного повернення БПЛА на точку старту для підзарядки акумуляторної батареї використані основні технічні характеристики зазначених моделей (табл. 5.3).

У середовищі MATLAB виконано моделювання групової поведінки суб-рою з 5 різних дронів: двох дронів моделі DJI Phantom 4 та трьох дронів моделі UDIRC LarkFPV.

Для того, щоб суб-рій з зазначених дронів облетів потрібну територію, необхідно зчитати з мапи (нп., у п. 5.2) координати точок, які мають бути відвіданими. Для полегшення цього завдання доцільно використовувати застосунок, який при натисканні мишкою на точку на мапі записує двовимірні координати точки у текстовий файл. Приклад такого застосунку, написаного у середовищі Microsoft Visual Studio на мові програмування C#, наведений у додатку А. Вигляд кінцевого файлу з координатами траєкторії руху БПЛА (рис. Б.1) та код застосунку для моделювання у середовищі MATLAB руху БПЛА за визначеними координатами наведені у додатку Б.

Аналогічні дії необхідні для зчитування послідовності точок зі складного рисунку, що задає траєкторії групової поведінки об'єктів рухомої мережі. Складні траєкторії руху БПЛА необхідні при використанні зграй дронів у районах бойових дій. Такі маршрути

унеможливають прогнозування руху БПЛА та суттєво ускладнюють їхню ліквідацію засобами протиповітряної оборони (ППО) у порівнянні з маршрутами, за якими відбувається прямолінійний рух дрона між точками маршруту. Приклад повної (summary) непрямої складної траєкторії одного та всіх дронів суб-рою наведений на рис. 5.8.

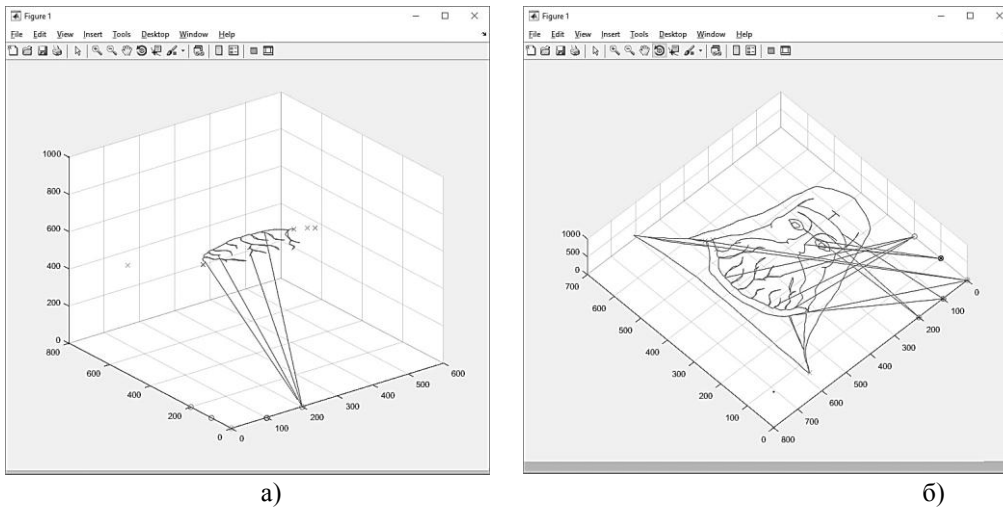


Рис. 5.8. Повна траєкторія одного дрону (а) та всіх дронів (б) з мінімізованими ділянками прямолінійного руху (початок та кінець маршруту дрону марковані однаковим кольором)

Після того, як отримані всі координати траєкторії польоту кожного з дронів, у середовищі MATLAB можна відтворити візуалізацію групової взаємодії дронів суб-рою. Приклад програми, яка має графічне представлення польоту суб-рою БПЛА, наведений у додатку Б.

У програмі для кожного дрона задається максимальний час польоту відповідно до технічних характеристик кожної з моделей БПЛА (табл. 5.3). Коли час закінчується, дрон летить на свою базу (точку старту саме цього дрону), де йому змінюють або дозаряджають акумулятор. Після цього дрон повертається на свою попередню позицію та продовжує рух власною траєкторією.

Таблиця 5.3. Технічні характеристики дронів суб-рою, обраних для моделювання групової взаємодії [50; 103]

Технічні характеристики дронів	DJI Phantom 4	UDIRC LarkFPV
Розмір по діагоналі, мм	590	580
Горизонтальний розмір без гвинтів, мм	350	370
Діаметр гвинта, мм	232	190
Вертикальний розмір, мм	210	120
Кількість двигунів: безколекторні електричні	4	4
Ємність акумулятора, мА×год	5350	1000
Максимальний час польоту, хв	25–28	8–10
Максимальна дальність польоту, м	5000–7000	80–200
Вага, г	1380	250

Результати моделювання групової взаємодії зазначених моделей 5 дронів у єдиному суб-рою наведені на рис. 5.9 – рис. 5.11.

На рис. 5.9 відображено початок руху дронів. Внизу на осях координат різнокольоровими кружечками позначені бази для кожного дрона (стартова позиція та/або місце підзарядки акумуляторної батареї), хрестиками відповідного кольору (у кожного дрона свій колір) позначені точки, звідки БПЛА починає рух.

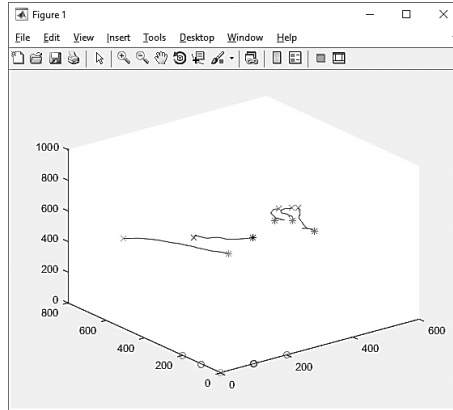


Рис. 5.9. Початкове положення дронів

На рис. 5.10 та рис. 5.11 видно, що у дрона, позначеного чорним кольором, в точці траєкторії, відміченою чорною зірочкою, закінчується заряд акумулятора (менше 17 %), та дрон прямує до своєї бази.

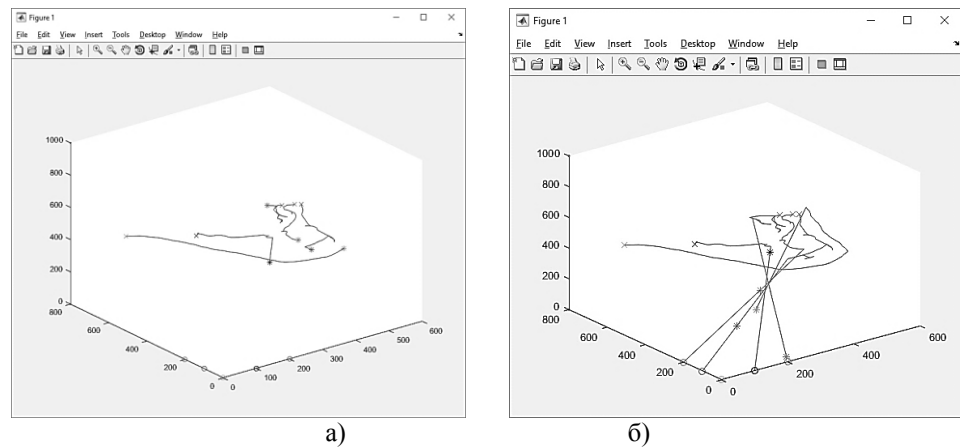


Рис. 5.10. Дрони у процесі підзарядки: помічений чорним кольором рухається на власну базу для підзарядки (а), помічені червоним та зеленим кольором повертаються на маршрут з власних баз (точок старту) після підзарядки (б)

Після того, як дрон облетів свій маршрут, він повертається на базу та знаходиться там до кінця роботи програми (рис. 5.11).

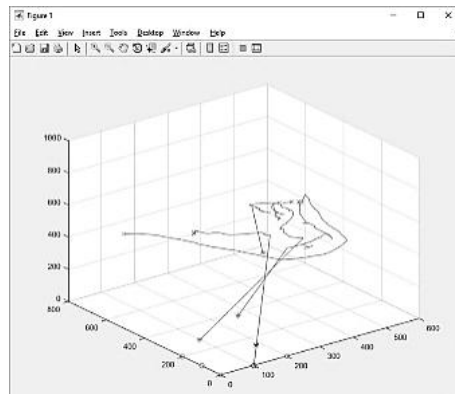


Рис. 5.11. Дрони, помічені зеленим та червоним кольорами, прямують на власні бази після закінчення виконання завдань

Висновки до розділу 5

У розділі було розглянуто моделювання групової поведінки об'єктів рухомої моніторингової мережі. Математично задача приводиться до методів вирішення задачі комівояжера.

Для генерації суб-оптимального маршруту для кожного БПЛА суб-рою розроблений програмний застосунок, в якому використана НМХ. Було погоджено, що під час 2D-моделювання на політ дрона нічого не впливає (метеоумови не враховуються), БПЛА має облетіти всі задані оператором точки, причому в жодній не має бути більше одного разу. В кінці польоту апарат має повернутись у точку вильоту. Оптимізація шляху проводиться за критерієм найкоротшої довжини.

Вважаючи, що НМХ є доволі приблизним методом вирішення проблеми, вона використана для генерування маршрутів за кількістю БПЛА у суб-рою. Суб-оптимальні шляхи отримані в результаті декількох послідовних запусків (за кількістю БПЛА у суб-рою) програми з тими самими вхідними даними. Такі маршрути досить схожі на оптимальний за довжиною (незначно його перевищують), але суттєво відрізняються від нього. У результаті такі маршрути можуть використовуватися по одному для кожного БПЛА у суб-рою.

Передбачається автономна або напівкерована поведінка БПЛА. Тобто, шлях може бути згенерованим автоматично або обраним оператором, включно з обранням послідовності опрацьованих точок місцевості.

Для відпрацювання алгоритмів запобігання зіткненням БПЛА виконане 3D-моделювання групової поведінки дронів у складі суб-рою.

Наведений приклад програми для автоматизованого формування файлу з координатами точок маршруту, в яких здійснюватиметься моніторинг. Створена візуалізація польоту гетерогенного суб-рою БПЛА, який складається з різних моделей дронів. Під час моделювання враховано максимальні дальність та час польоту кожної з моделей дронів; реалізовано алгоритми повернення дронів на власні бази (точки старту) для підзарядки акумуляторних батарей з наступним продовженням руху своєю траєкторією.

Подальші дослідження групової поведінки БПА у складі зграй, поділених на суб-рої, стимулюються численними завданнями в самих різних областях людської діяльності (автоматизація управління рухом транспортних засобів, боротьба з тероризмом і розмінування підозрілих предметів, робота в умовах сильної задимленості під час пожежогасіння, інспекція територій, забруднених хімічними речовинами, самостійне патрулювання призначених територій та ін.).

Розділ 6

МЕТОДИ ТА АЛГОРИТМИ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ВИКОРИСТАННЯ ТА ПОДОВЖЕННЯ ЧАСУ ЖИТТЯ РУХОМОЇ МОНІТОРИНГОВОЇ МЕРЕЖІ

6.1 Покращення ефективності використання безпілотних літальних апаратів шляхом розподілу завдань між ядрами процесора

Обчислювальні системи більшості безпілотних літальних апаратів (БПЛА) базуються на однокристальних 4-ядерних 32-бітних процесорах ARM Cortex, Qualcomm Snapdragon та ін. [40; 140; 175]. Кількість обчислювальних потоків в таких процесорах дорівнює кількості фізичних ядер, між котрими можуть бути розподілені завдання, що виконуються (орієнтація у просторі, керування рухом, отримання та обробка показників з датчиків, тощо). Водночас, перше ядро може миттєво стати надзвичайно перевантаженим у разі виникнення термінового завдання від системи управління БПЛА.

Для комп'ютерної системи (КС) з обмеженими ресурсами (обчислювальними або енергоживлення) виникає наступна проблема – робота автоматичного планувальника (scheduler) потоків операційної системи (ОС) призводить до невиправданого навантаження процесора. Але для мобільних комп'ютерів, на яких базується конструкція БПЛА, частіше є важливим економічність в енергоспоживанні (на практиці – час роботи від акумуляторної батареї), ніж залучення в обчислювальному процесі всіх наявних ядер в однокристальному обчислювальному процесорі (англ. Central Processing Unit або CPU).

Тому актуальною задачею є підвищення ефективності використання БПЛА за рахунок розробки кращих, ніж автоматичні, алгоритмів роботи багатоядерного процесора, на базі якого побудована обчислювальна система БПЛА.

Зважаючи на те, що безпосереднє натурне моделювання на БПЛА може призвести до критичного застосування останнього, було вирішено провести імітаційне моделювання диспетчеризації завдань на стаціонарній КС з 4-ядерним одноплатним процесором.

Тест проводився на КС з CPU Intel Core i7-4790 3.60GHz при вимкненій функції підтримки процесором технології Hyper-Threading [48]. Для експерименту було створено тестову програму, яка на кожній ітерації сортує масив з 10 000 елементів.

У межах роботи було прийнято рішення під час написання програмного забезпечення на C# самостійно реалізувати навантаження ядер заданими потоками. Також планувалось здійснити процес еталонного зіставлення з автоматичною диспетчеризацією в ОС Windows. Інакше цей процес можна визначити як бенчмаркінг ефективності використання ядер багатоядерного процесора.

Досліджувались такі критерії, як таймінг завдань, енергоспоживання процесора, температура та відсоткове навантаження кожного ядра та всього процесора протягом виконання кожного з тестових завдань чотирьох типів. Таймінг оцінювався за результатами роботи власноруч написаного програмного забезпечення. Температура (або тепловиділення – параметр TDP, Thermal Design Power), енергоспоживання та відсоткове використання процесора можуть бути оцінені за допомогою попередньо запущеної моніторингової утиліти [202; 254].

Одним з найбільш проблемних місць є перевантаження та перегрів процесора обчислювальної системи БПЛА. Це може привести до зависання КС, втрати керування та крешу дрону. Крім того, у критичній ситуації перевантажений процесор не забезпечить терміново необхідні обчислення зміни курсу БПЛА або перехід на автономний режим польоту.

Підвищити ефективність використання БПЛА з подовженням часу польоту останніх можливо за рахунок зниження перенавантаження та енергоспоживання обчислювальної системи.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

1. Проаналізувати можливість диспетчеризації завдань процесора БПЛА з багатопотоковою організацією розподілу завдань між ядрами CPU та розробити відповідні алгоритми.

2. Виконати імітаційне моделювання роботи розроблених алгоритмів для 4-ядерного CPU.

3. Дослідити технічні показники стану CPU КС БПЛА (температуру в розрізі по ядрах, енергоспоживання та навантаженість процесора, кількість обертів кулера для забезпечення необхідного тепловідводу тощо) при реалізації різних алгоритмів диспетчеризації.

4. Дослідити можливість підвищення ефективності використання БПЛА за рахунок джерел додаткового енергоспоживання.

5. Визначити з досліджених оптимальний алгоритм розподілу завдань між ядрами багатоядерного однокристального процесора.

Проблема оптимізації енергоспоживання, навантаження та запобігання перегріву комп'ютерних компонентів обчислювальних систем БПЛА широко розглянута в ресурсах світової наукової періодики. Основним направленням її вирішення є багатопотокова диспетчеризація завдань між ядрами багатоядерного процесора [205; 217; 270]. Також існують різні технічні рішення, пов'язані з проектуванням процесорних кулерів для забезпечення необхідного теплового розсіювання [23; 312].

Однак, існуючі алгоритми планування розподілу завдань на рівні ОС [173; 325] неспроможні вирішувати проблему запобігання максимальному навантаженню процесора та кожного з його ядер майже до 100 % [182]. Крім того, можлива ситуація, коли завдання стоятимуть у черзі до процесора [172], що неприпустимо для завдань оптичної навігації та керування БПЛА.

Для зменшення енергоспоживання у системах, що живляться від батареї, іноді використовується механізм відключення модулів зв'язку [22] під час швидкісного декодування даних або компресії отриманого відео [28; 99]. Але, у такому випадку може мати місце так званий ефект Дала, коли частина ядер процесора простоє, незважаючи на те, що своєчасність виконання певних завдань не забезпечується [127]. Таке планування завдань є неприпустимим для БПЛА, тому що він, наприклад, не зможе прийняти керуючі сигнали від головного БПЛА або з наземного КЦ.

Втрати енергії на постійне підключення до батареї живлення модулів зв'язку можуть бути частково компенсовані за допомогою механізмів, описаних у [201]. Передача надлишкової енергії між об'єктами КФС є вельми енергетично привабливою для вирішення поставленої задачі побудови ефективних інформаційно-вимірювальних мереж (ІВМ), до яких можна віднести і зграї БПЛА [179].

Оцінку ефективності вжитих заходів щодо підвищення ефективності використання компонентів КС можна провести за допомогою низки програмних продуктів для моніторингу та зміни режимів навантаження процесорів [41]. Однак, варто зауважити, що для КС БПЛА категорично не можна використовувати так звані «прожарочні тести» (AIDA64, LinX, IntelBurnTest, Prime95, OCCT Perestroika тощо). Принципом їхньої дії є стрес-тестування, коли процесор виконує спеціалізовані інтенсивні обчислення, що імітують сценарій роботи в найбільш важких умовах та призводять до максимального

нагрівання ядер [29]. Застосування таких тестів може спричинити невіправні наслідки з вигоранням окремих компонентів КС. При введенні граничних значень в тестові режими таких утиліт можливо вивести досліджувану КС з ладу.

Тому для бенчмаркінгу доцільно застосувати програмне забезпечення (ПЗ), яке серед інших показників стану апаратного забезпечення КС виводить на монітор показання вбудованих у таку КС датчиків (Core Temp, CPU-Z, HWMonitor, SpeedFan й т. п.).

З перелічених для бенчмаркінгу була обрана утиліта HWMonitor v.1.33.0.

Таким чином, результати аналізу дозволяють зробити висновок про те, що питання підвищення ефективності використання БПЛА потребують додаткових досліджень.

По-перше, доцільно детальніше розглянути проблему оптимального навантаження CPU у розрізі ядер процесора.

Прив'язка процесу до заданих ядер не викликає ніяких труднощів, потрібно лише скористатися властивістю *ProcessorAffinity* класу *Process*, прописаним у просторі імен *System.Diagnostics* [141]. Залежно від бітової маски, яка присвоюється цій властивості, можна отримати всі комбінації ядер на обраному пристрої. При наявності 4 ядер, планувальник потоків ОС автоматично буде використовувати їх всі, намагаючись отримати максимальну вигоду від багатоядерності процесора (надалі – CPU). У такому випадку *ProcessorAffinity* дорівнює 15 та визначається за формулою:

$$ProcessorAffinity = 2^n - 1, \quad (6.1)$$

де n – кількість ядер багатоядерного процесора, у тесті $n = 4$.

Варто зазначити, що під час розробки програмних додатків рекомендується уникати використання для обчислень ядра, на якому операційна система виконує передачу даних, очищення пам'яті та інші системні процеси. Зазвичай всі такі процеси обмежені першим ядром процесора. Таким чином, виключення першого ядра з обчислювальної задачі може привести до підвищення продуктивності додатків. Тому для бенчмаркінга доцільно обрати ядра з 2-го по 4-те та присвоїти нову бітову маску, отримавши значення 14 (рис. 6.1).

```

C:\WINDOWS\system32\cmd.exe
Total # of processors: 4
Current processors affinity: 15
*****
Insert selected processors, separated by comma (first CPU index is 1):
2,3,4
*****
Processor #2 was selected for affinity.
Processor #3 was selected for affinity.
Processor #4 was selected for affinity.
*****
Current processor affinity is 14
Для продовження натисніть будь-яку клавішу...

```

Рис. 6.1. Значення властивості *ProcessorAffinity* при використанні 4 ядер та ядер з 2-го по 4-те

Відповідно, для кожного з чотирьох ядер 4-ядерного процесора значення *ProcessorAffinity* будуть такими:

- для 1-го ядра *ProcessorAffinity* = 1;
- для 2-го ядра *ProcessorAffinity* = 2;
- для 3-го ядра *ProcessorAffinity* = 4;
- для 4-го ядра *ProcessorAffinity* = 8.

Зауважимо, що більш корисним буде прив'язуватись не до ядер процесорів, а використовувати прив'язку до потоків, але цей шлях містить багато труднощів. Справа в

тому, що Intel намагається полегшити роботу програміста, взявши прив'язку потоків до ядер на себе [48]. Очевидно, це має сенс, бо більшість великих компаній розробляють продукцію для масового користувача, для якого розуміння і використання власної паралелізації не є необхідним.

Для високої математичної продуктивності можна використовувати бібліотеку готових компонентів для програмування додатків на мовах C/C++ та Fortran – Intel Math Kernel Library (MKL) [84]. Для швидкої роботи циклу доцільно включити автопаралелізацію в компіляторі від фірми Інтел – Intel C++ Compiler (ICC) [48]. Для максимального використання кешу включають підтримку технології Hyper-Threading. Хоч наявність такої технології є хорошим маркетинговим рішенням, але для тих, кому потрібно визначати паралелізацію вручну, це ставить на шляху певні перешкоди. Програміст може встановити власну прив'язку потоків до ядер, однак бібліотека Intel зведе їх нанівець.

У досліджуваному випадку додаткові труднощі виникають через те, що потік .NET, яким керує загальномовне виконуюче середовище Common Language Runtime (CLR), не відповідає потоку ОС, а прив'язувати до ядер можна виключно потоки операційної системи [66; 149]. Щоб вирішити цю проблему, можна скористатися наведеними методами класу *Thread*:

```
Thread.BeginThreadAffinity();
...
Thread.EndThreadAffinity().
```

У такому разі код між наведеними викликами виконується на єдиному потоці ОС, що суттєво послаблює менеджмент потоків CLR.

Тепер можна перейти до самої прив'язки. Отримати ОС-потоки додатку .NET можна, використовуючи *Process.GetCurrentProcess().Threads* – колекцію поточкових об'єктів. Щоб отримати ОС-потік, який виконується на цей момент, потрібно використати наступний код:

```
[DllImport("kernel32.dll")]
public static extern int GetCurrentThreadId().
```

Застосовуючи повернутий ID, можна знайти потік завдання, що виконується, і використати властивість *ProcessThread.ProcessorAffinity*, робота якої дуже схожа на *Process.ProcessorAffinity*, яка описана вище.

Далі оголошено клас *DistributedThread*, використання якого дозволяє запускати потоки на обраних ядрах. Сутність полягає в інкапсуляції звичайного *Thread*-об'єкта, обмежуючи його діючим потоком ОС і установкою бажаної прив'язки до ядра:

```
using System;
using System.Diagnostics;
using System.Linq;
using System.Runtime.InteropServices;
using System.Threading;
namespace DistributedWorkManager{
    public class DistributedThread{
        [DllImport("kernel32.dll")]
        public static extern int GetCurrentThreadId();
        [DllImport("kernel32.dll")]
        public static extern int GetCurrentProcessorNumber();
        private ThreadStart threadStart;
        private Thread thread;
        public int ProcessorAffinity { get; set; }
        public Thread ManagedThread{
```


3. Проекти запускаються по чергово при однаковій кількості фонових задач.

Зміст проектів та результати їхнього виконання наведені на рис. 6.4 – рис. 6.7 та зведені у табл. 6.1.

У Проекті № 1 виконане імітаційне моделювання роботи автопланувальника ОС: створено 4 паралельних потоки по 150 ітерацій на ядрах з 1-го по 4-те (рис. 6.4а), зважаючи на те, що автопланувальник намагатиметься рівномірно навантажити кожне з чотирьох ядер (рис. 6.4б).

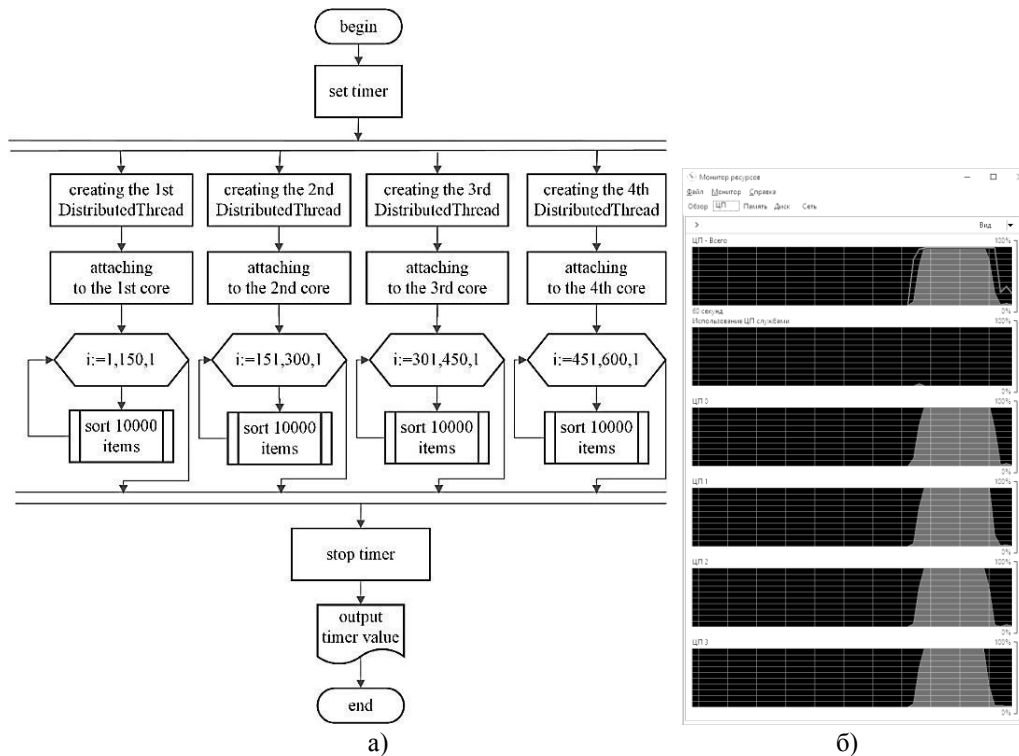


Рис. 6.4. Використання CPU за Проектом № 1:
а – блок-схема; б – осцилограма

За Проектом № 2 створюються два паралельних потоки: 300 ітерацій на 1-му ядрі та 300 ітерацій, розподілених між 2–4-им ядрами (рис. 6.5а). Другий потік розподіляється автопланувальником ОС Windows на три цикли по X_i ітерацій у кожному так, що:

$$X_2 + X_3 + X_4 = 300.$$

Точні значення X_i не визначені, оскільки невідомо, як саме автопланувальник ОС поділить потік між трьома ядрами. Але, з моніторингових даних на рис. 6.4б видно, що автопланувальник ОС намагається рівномірно навантажити ядра з 2-го по 4-те. Отже, підтверджується алгоритм, реалізований у Проекті № 1 для імітації дій автопланувальника ОС з розподілу завдань між ядрами процесора.

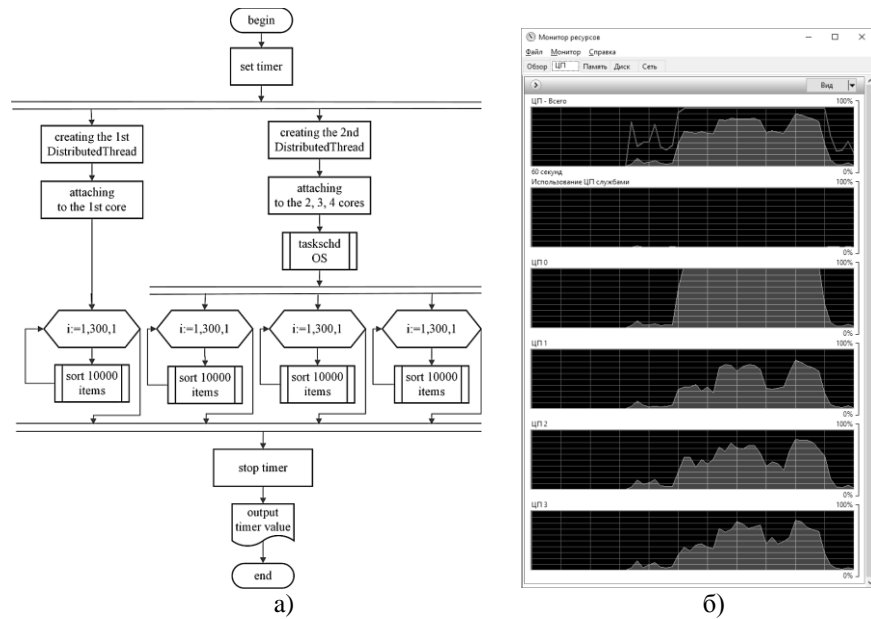


Рис. 6.5. Використання CPU за Проектом № 2: а – блок-схема; б – осцилограма

У Проекті № 3 реалізований алгоритм, за яким створений один послідовний потік з 600 ітерацій на 1-му ядрі (рис. 6.6). Аналіз результатів цього алгоритму є важливим, тому що в 4-ядерних однокристальних процесорах, на базі яких побудовані обчислювальні системи БПЛА (найчастіше це різні моделі процесорів ARM Cortex), немає підтримки технології віртуальної багатопотоковості Hyper-Threading, як у процесорів компанії Intel. Тому, якщо не розділяти завдання на потоки, як запропоновано в цій роботі, то, наприклад, для процесора ARM Cortex A9 можливо у режимі Asymmetric Multiprocessing (AMP) отримати все навантаження на 1-ше ядро (рис. 6.6а) [2]. Тоді таймінг виконання завдання за Проектом № 3 зросте майже втричі у порівнянні з бенчмарком Проекту № 1 (табл. 6.1).

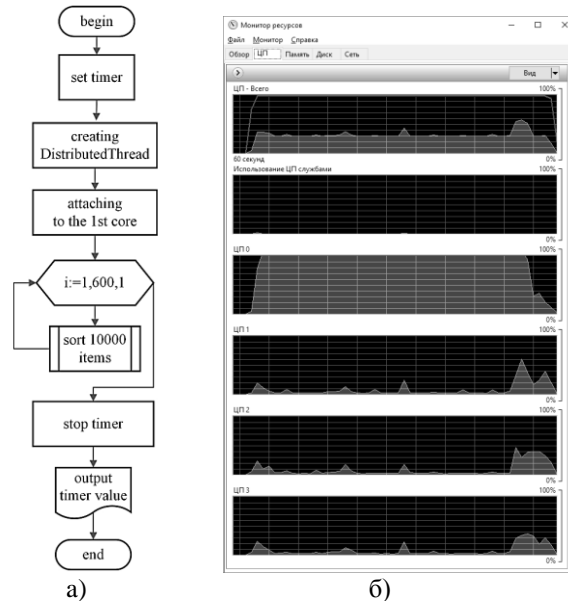


Рис. 6.6. Використання CPU за Проектом № 3: а – блок-схема; б – осцилограма

У Проекті № 4 реалізовано алгоритм, згідно з яким 1-ше ядро не задіяне, а до 2–4-го ядер зроблена прив'язка трьох паралельних потоків по 200 ітерацій кожен (рис. 6.7а). Але, за моніторингом ресурсів на 1-му ядрі все ж зареєстрована активність, що можна пояснити обробкою фонових процесів ОС (рис. 6.7б). Саме тому, запропонований алгоритм ручної прив'язки тестового завдання на 2–4-те ядра можна вважати найоптимальнішим. У такому разі, розвантажене 1-ше ядро найшвидшим чином зможе відпрацювати завдання, пов'язані з критичним застосуванням БПЛА. Наприклад, коригування маршруту та час появи механічної перешкоди на шляху БПЛА, перехід на оптичну навігацію при неможливості отримання GPS-координат тощо. У якості тестового завдання Проекту № 4 у такому разі можна розглядати компресію отриманого відеопотоку, підготовку даних для передачі за допомогою модуля зв'язку з БПЛА до наземного командного центру або до головного дрону.

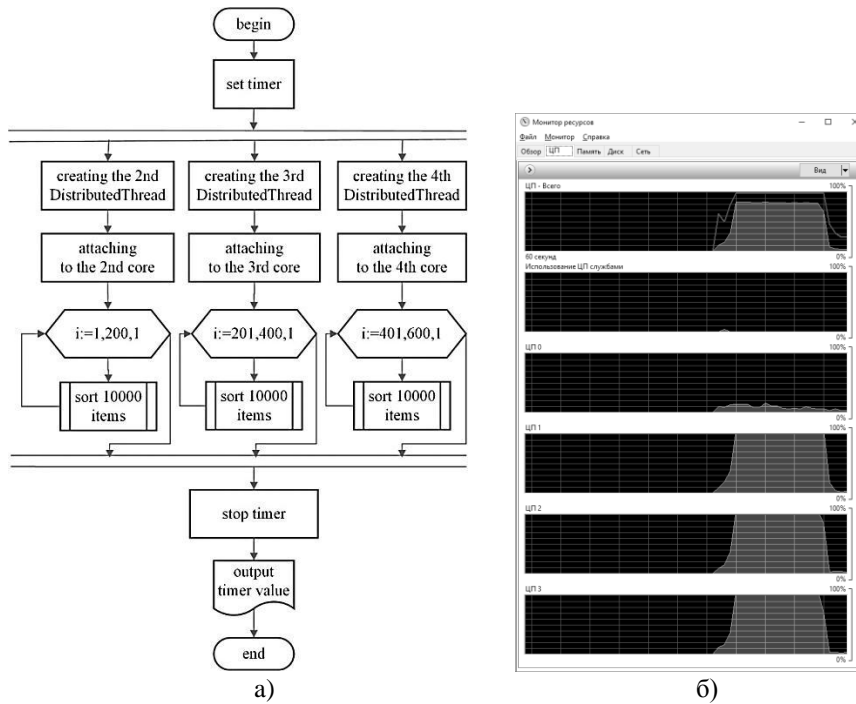


Рис. 6.7. Використання CPU за Проектом № 4:
а – блок-схема, б – осцилограма

Таблиця 6.1. Результати бенчмаркінгу за чотирма проектами

Показник	Проект № 1	Проект № 2	Проект № 3	Проект № 4	Використане ПЗ
Гаймінг (AverageTime), с	12,7	24,6	47,2	16,5	Власна розробка
Енергоспоживання (Average Power), Вт	58,5	38,9	27,3	49,3	HWMonitor v.1.33.0
Температура (Temperature), °C	75	62	62	80	HWMonitor v.1.33.0
Використання CPU (Utilization), %	100	56	31	77	HWMonitor v.1.33.0
Кількість обертів кулера на процесорі (Fan's number of rotations), RPM	1900	1560	1505	1599	HWMonitor v.1.33.0

При бенчмаркінгу можлива розбіжність показників, яка обумовлена тим, що програми моніторингу зчитують показання датчиків раз на 2, 3 або 5 с. Це може не збігатися з таймінгом проектів, що тестуються.

За даними моніторингу ресурсів (табл. 6.1), отриманими тестувальником HWMonitor, вдалося суттєво знизити температуру на 1-му ядрі шляхом ручного перерозподілу завдання між ядрами процесора за Проектом № 4 (рис. 6.8). Майже на 13 % у порівнянні зі всіма іншими проектами. Завдяки цьому при запуску нового фоновому процесу ОС, який може бути пов'язаним з критичним застосуванням БПЛА (втраті GPS-координат, включення задачі оптичної навігації, миттєвий перерахунок курсу за причини пориву вітру та ін.), значно підвищується ефективність використання КС.

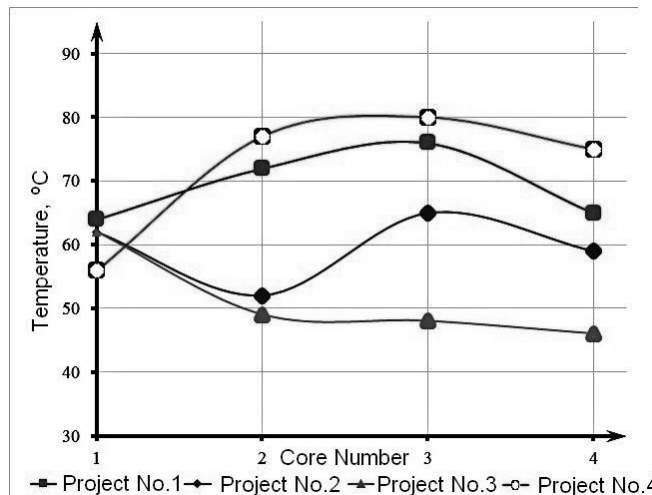


Рис. 6.8. Температура однокристалного процесора з деталізацією по ядрах

Зважаючи на те, що за бенчмаркінгом алгоритм саме Проекту № 4 виявився оптимальним, програмний код запропонованого алгоритму потребує детальніших пояснень.

По-перше, необхідно підключити простір імен, у якому оголошено розроблений клас *DistributedThread* [205]:

```
using System;
using System.Threading;
using DistributedWorkManager;
```

```
namespace ConsoleApplication1{
class Program{
```

Наступним кроком встановлюються таймери і розмір масиву:

```
static Random rnd = new Random();
static readonly int N = 10000;
static void Main(string[] args){
DateTime t= DateTime.Now;
```

Клас *CountdownEvent*, який використовується в наступному коді, є необхідним для синхронізації та розблокування потоків, які очікують у черзі. Отже, основний код не буде виконуватись до тих пір, поки не отримає певну кількість сигналів (у розглянутому випадку – три).

Потім створюється об'єкт класу *DistributedThread*, прив'язується до 2-го ядра та запускається. Для дотримання логіки дій змінну буде названо *thread2*, а не *thread1*). Така нотація надає наочності, до якого саме ядра виконується прив'язка:

```

using (CountdownEvent e = new CountdownEvent(3)){
    DistributedThread thread2 = new DistributedThread(delegate
    {
        for (int k = 0; k < 200; k++){
            int[] intArray = new int[N];
            for (int i = 0; i < N; i++){
                intArray[i] = rnd.Next(100 + k, 1000 + k);
            }
            int temp, j;
            for (int i = 1; i < intArray.Length; i++){
                temp = intArray[i];
                j = i - 1;
                while (j >= 0 && intArray[j] > temp){
                    intArray[j + 1] = intArray[j];
                    j--;
                }
                intArray[j + 1] = temp;
            }
        }
        e.Signal();
    });
    thread2.ProcessorAffinity = 2;
    thread2.Start();
}

```

Створення та робота потоків *thread3* та *thread4* аналогічні потоку *thread2*, тому код не наведений. Після завершення роботи всіх трьох потоків на консоль виводиться значення таймера.

```

        e.Wait();
    }
    Console.WriteLine($"{DateTime.Now - t}");
}
}
}

```

Нижче представлена діаграма, що наочно демонструє підвищення споживаної енергії залежно від кількості задіяних ядер (рис. 6.9).

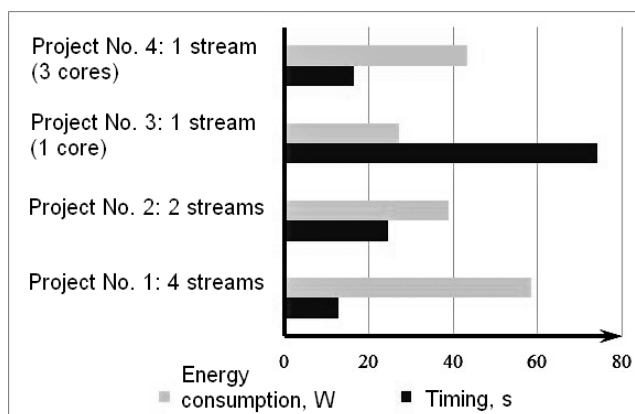


Рис. 6.9. Енергоспоживання в однокристальному обчислювальному процесорі залежно від кількості навантажених потоків

Підхід з використанням 4 потоків є найбільш ефективним за таймінгом, однак, він також є найменш енергозберігаючим. Тому найбільш ефективне використання обчислювальних ресурсів БПЛА забезпечить розподіл згідно з алгоритмом Проекту № 4 завдань, не пов'язаних з навігацією та з керуванням дроном.

У результаті аналізу механізму роботи автоматичних планувальників завдань ОС з'ясовано, що у зазначених умовах здійснюється розподіл процесів між усіма чотирма ядрами CPU. Але, в такому разі 1-ше ядро не має резервів у порівнянні з іншими ядрами за навантаженістю (навантажене на 100 %) та температурою (є на рівні з іншими задіяними ядрами). Кількість обертів кулера та енергоспоживання процесора в режимі автопланування є найвищим. Тому, при виникненні критичної ситуації, коли необхідним може бути суттєве збільшення обчислювань на 1-му ядрі для задач системи управління БПЛА, такі задачі можуть стояти у черзі за попередньо запущеними завданнями. З точки зору життєзабезпечення БПЛА це є неприпустимим.

Дослідження показали, що розпаралелювання завдань, які можуть бути виконані як багатопоточні (компресія відеопотоку, передача даних різними каналами зв'язку та ін.), підвищує ефективність використання БПЛА. Прив'язка процесів до трьох ядер замість чотирьох на 23 % збільшує загальний час виконання завдання у порівнянні з часом виконання аналогічного завдання на одному ядрі.

6.2 Бездротові енергонезалежні інформаційно-вимірювальні мережі з передачею енергії між КФО та розподілом завдань між ядрами CPU

Розробкою бездротових енергонезалежних інформаційно-вимірювальних мереж (БЕІВМ) критичного застосування військово-цивільного призначення та компонентів до них займалася велика кількість провідних вчених та науковців. Проте, під час розробки таких систем виникає чимало важливих науково-технічних питань, які на сьогодні не вирішені: методи оптимального використання живлячих компонентів з обмеженою енергоємністю, розміщення КФО в мережі (оптимальна топологія БЕІВМ), вибір типу бездротової технології передачі даних, забезпечення завадостійкості, гарантований зв'язок між об'єктами КФС, довготривала автономна робота компонентів КФО тощо.

Нині моніторингові мережі є множиною компонентів з вимірювальними функціями [53]. Цієї множини функцій достатньо лише для контролювання прогнозованих процесів. У разі, коли процес функціонування об'єктів моніторингу є динамічним, виникає складність адаптації інформаційно-вимірювальної системи до нових властивостей об'єктів, що підлягають моніторингу. Виникає проблема модифікації архітектури та топології БЕІВМ для вирішення задач моніторингу при просторовому переміщенні [2; 126]. Важливим є також вирішення проблеми розподілення завдань для актуального аналізу компонентів рухомої моніторингової мережі об'єктів [316]. Проблемність виконання завдань може бути обумовлена обмеженістю обчислювальної потужності або ресурсів енергозабезпечення компонентів моніторингової мережі [14]. Складність полягає у ефективному поєднанні дистанційного керування автоматизованою мережею компонентів, що є автономними і самодостатніми, для виконання завдань контролю динамічних просторових процесів [281].

Відомі методи створення та застосування моніторингових мереж мають суттєві недоліки: батарейне живлення рухомих компонентів таких мереж вимагає жорсткого контролю за часом та дальністю функціонування мережі, обмеження трафіку передаваної інформації для економії енерговитрат, жорстку траєкторію моніторингу через відсутність інтелектуальних властивостей компонентів мереж [281]. Разом з тим, постійно зростаюча кількість різних об'єктів моніторингу навколишнього середовища, технологічних процесів, а також моніторинг рухомих безпілотних автономних систем вимагає збільшення довготривалості моніторингу та зміни поведінки рухомих компонентів такої мережі [139]. Такий підхід потребує створення нових більш ефективних рішень щодо інтелектуальної або керованої взаємодії компонентів БЕІВМ.

Таким чином, необхідно сформулювати вимоги та основні підходи розроблення рухомих компонентів БЕІВМ критичного застосування з подовженим часом енергонезалежності

та принципово новим типом їх дистанційно керованої взаємодії при військово-цивільного призначенні, що матиме суттєву економічну та національно-безпекову значимість через свою конкурентоспроможність на вітчизняному та закордонних ринках.

У основу підходу покладено ідею створення та використання рухомих компонентів БЕІВМ критичного застосування з подовженим часом енергонезалежності. Результати процесу вимірювання (наприклад, розмір винайдених дефектів, швидкість пересування каравану, температура, показники забрудненості та/або випромінювання у зоні функційного модуля тощо) у шифрованому та/або стеганографічному вигляді передаються у складі загальної інформації (нп., метеодані під час руху АС, відносні координати функційного модуля АС у складі каравану, тощо) до військового центру керування та/або диспетчерського центру компанії-власника АС засобами досяжних відкритих каналів зв'язку [77].

Тенденції розвитку сучасної техніки направлені на створення легких, автономних мобільних пристроїв. Вони мають бути максимально компактними та не втрачати функціональної ефективності. Однією з ключових характеристик пристроїв є час їхньої автономної роботи. Цей час залежить від багатьох факторів. Основні з них:

- ємність батареї;
- енергоспоживання компонентів приладу;
- оптимізація програмного забезпечення.

Є багато способів вирішення проблем, пов'язаних зі збільшенням часу автономної роботи пристрою. Це і збільшення об'єму батареї, оптимізація програмних продуктів, а також покращення режимів роботи основних модулів мобільного пристрою, які потребують великої кількості енергії. Останній підхід є одним з найперспективніших. Одним з найбільш енерговитратних модулів системи є процесор, тому оптимізація роботи цього модуля на алгоритмічному рівні може значно покращити ефективність використання мобільного пристрою.

Тому перспективною є організація ефективного розподілення навантаження процесу мобільних об'єктів БЕІВМ між ядрами процесора кожного об'єкта для зменшення рівня енергоспоживання мобільних об'єктів на базі ОС Android, iOS, Windows Phone та ін. ОС низькоресурсних обчислювачів. Для цього необхідно розробити нові методи покращення енергоживлення, розподілення енергоресурсів, що направлені на зменшення енергоживлення кожного ядра у багатоядерній енергообмеженій системі.

Зважаючи на те, що багатоядерні процесори деяких об'єктів БЕІВМ не є технологічно однаковими, вирішення проблеми економії енергоспоживання процесора може полягати у розподілі задач між кластерами, до яких належить те чи інше ядро. При сучасній трикластерній технології енергоекономні ядра нижніх кластерів процесорів (нп., Cortex-A53 з частотою 1,0–1,8 ГГц, Snapdragon 6xx та 8xx з частотою 1,6 ГГц) доцільно задіяти на максимум потужності для обробки поодиноких сигналів від вимірювальних датчиків фізичних величин. Окремий сопроцесор і6 в технологіях Apple також призначений для обробки даних з гіроскопа, барометра, малопотужного GPS, акселерометра та сенсорів БПЛА. Крім того, під час виконання таких завдань енергоекономними ядрами, нагрів чіпсета практично відсутній.

Більш потужні ядра верхніх кластерів (нп., Cortex-A72 з частотою до 2,5 ГГц, Snapdragon 8xx з частотою 2,15–2,4 ГГц) у цьому випадку мають бути задіяними на 60–70 % під час виконання подібних завдань. На повне навантаження вони використовуються лише при виконанні задач оптичного розпізнавання, обробки аудіо- та ін. мультимедійних завданнях.

Такий перерозподіл завдань між ядрами різних кластерів процесора дозволяє платі-обчислювачу БПЛА споживати майже вдвічі менше енергії, що суттєво подовжує час життя апарата.

Необхідно зазначити, що виробниками процесорів для мобільних пристроїв від Apple, Huawei, Mediatek докладено значних зусиль щодо створення базових алгоритмів автоматичного розподілу завдань між кластерами багатоядерних процесорів. Але, такий розподіл в залежності від типу виконуваного завдання веде до того, що при обробці сигналів від сенсорів завантажені лише ядра нижнього кластера, а середній та верхній кластери під час цього простоюють. Навпаки, при обробці зображень та відео з камер БПЛА відключеними виявляються ядра нижнього кластера. Отже, на БЕІВМ чекають зайві витрати енергії на повторні вклучення ядер після режиму сну. Доцільно модифікувати базові алгоритми розподілу завдань таким чином, щоб усі ядра були завантажені постійно, на різному рівні енергоефективності, без витрат енергії на вклучення після сну та переключення частоти функціонування.

Для натурального експерименту щодо вирівнювання таймлайну поточного завантаження кожного з ядер процесора (коефіцієнт завантаження процесора) був обраний чипсет Qualcomm Snapdragon 400 (з 4 ядрами Krait, тактова частота яких 1,4–1,7 ГГц), вбудований у смартфон Microsoft Lumia 640 під керуванням ОС Windows Phone 8.1. Цей вибір був заснований на тому, що ядра Krait мають можливість асинхронної роботи, тобто, кожне ядро може бути керованим незалежно від інших, і його ресурси будуть використовуватися тільки у разі потреби. У цьому і полягає відмінність Krait від рішень інших виробників, яким, з метою зниження енергоспоживання, необхідно використовувати п'яте енергозберігаюче ядро (нп., і6 в чипсетах Apple) і об'єднувати групи ядер. Крім того, процесори Qualcomm Snapdragon можуть працювати під керуванням інших ОС – Android, Linux та ін. [140].

Розглянемо процес розробки алгоритму передачі енергії з використанням WiFi-каналів, що не перетинаються.

Конфігурація об'єктів БЕІВМ може бути такою, коли деякі КФО, не пов'язані з виконанням пріоритетних завдань БЕІВМ, мають більший запас енергії або накопичують її під час вимірювань (датчики тиску, акселерометри на базі п'єзокерамічних датчиків фізичних величин та ін.). У такому випадку накопичена енергія може бути переданою між об'єктами БЕІВМ засобами бездротових технологій, за допомогою яких утворюються канали зв'язку [201].

Треба дослідити можливі канали передачі накопиченої енергії від поліметричних датчиків для подальшого її використання на інших об'єктах БЕІВМ. Якщо мета датчиків, крім вимірювання фізичних величин, полягає в зборі радіочастотної потужності і перетворення її в енергію постійного струму, то завданням бездротових модулів (Wi-Fi, Bluetooth, ZigBee тощо), через які здійснюється передача даних, може стати також передача накопиченої датчиками енергії іншим об'єктам БЕІВМ.

Більшість Wi-Fi модулів зв'язку, які використовуються в моніторингових БЕІВМ, працюють у діапазоні частот 2,412–2,484 ГГц, що належить до неліцензованого ISM-діапазону [240]. На сьогодні мабуть, найбільш популярними пристроями для бездротового зв'язку через Wi-Fi і організації Інтернету речей (IoT) є плати ESP8266 від компанії Espressif Systems, RTL8710 від компанії Realtek. Для об'ємних за кодом програм можна задіяти модуль Digistump Oak [92]. Важливо, щоб обраний модуль міг працювати у режимі Sleep and Deep Sleep modes for low power usage.

Основи сукупної технології (технологія PoWiFi – Power-over-WiFi), яка дозволяє засобами одних і тих же модулів бездротовим каналом пересилати інформацію і енергію, успішно випробувані для живлення сенсорів та інших пристроїв, від роутера, підключеного до електромережі змінного струму 220 В [170]. Доведено, що енергії, перенесеної сигналами Wi-Fi, достатньо для забезпечення роботи малопотужних пристроїв – наприклад, датчиків температури, монохромної камери низької роздільності (176 x 144 пікс., потужністю 60 мВт, напругою 2,4 В) – кожні 35 хв на відстані до 8,5 м. Авторами [170] зазначено, що за технологією PoWiFi може бути перенесена і більша

енергія, але в США Федеральна комісія зі зв'язку (англ. Federal Communications Commission або FCC) потужність бездротових пристроїв обмежує до 1 Вт. В Україні відповідною установою є НКРЗІ, що встановила обмеження на потужність активного бездротового обладнання, на яке не потрібні дозволи, на рівні 200 мВт [321].

Варто зазначити, що використання PoWiFi може привести до істотного зростання внутрішньої інтерференції і загального зниження пропускної здатності мережі Wi-Fi. Тому доцільно для відправки енергетичних фреймів з метою акумулювання енергії на інших об'єктах БЕІВМ використовувати інші частотні канали, а не той, на якому здійснюється передача даних. Також необхідно передавати додаткові «енергетичні пакети» в інші проміжки часу, коли інформація не передається.

Отже, задля уникнення інтерференції сигналів, які передаються в єдиному бездротовому просторі (радіоефірі), в одному сеансі зв'язку (нп., при виконанні i -го з m завдань) доцільно використовувати частотні канали, які не перетинаються.

Більшість пристроїв на основі вищезазначених модулів підтримує 11 частотних каналів технології Wi-Fi. На таке обладнання встановлено вбудоване програмне забезпечення (якщо мова йде про активне вендорське Wi-Fi обладнання – роутери, точки доступу та ін.) або драйвери (якщо мова про клієнтські адаптери) для домену або набору частотних каналів FCC. У зазначеному домені всього три непересічних канали (Ch. 1, 6 і 11 на рис. 6.10).

У Європі Wi-Fi обладнання підтримує діапазон або домен ETSI (англ. European Telecommunications Standards Institute – Європейський інститут зі стандартизації у галузі телекомунікацій), у якому використовується додатково 12-й і 13-й канал. У цьому випадку додатково утворюється ще 2 комплекти непересічних каналів: Ch. 2, 7, 12 та Ch. 3, 8, 13 (рис. 6.10).

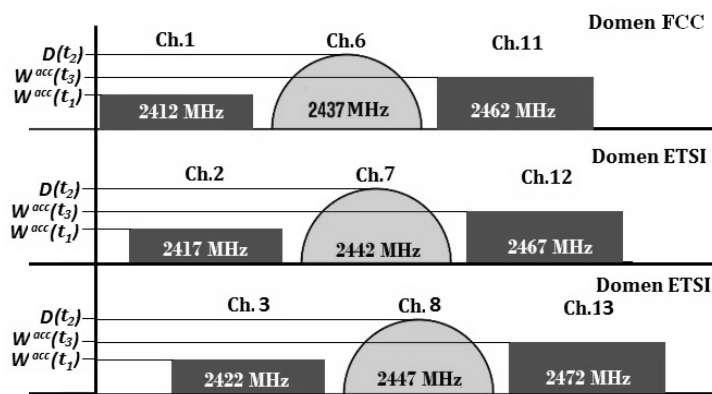


Рис. 6.10. Розподіл функцій передачі даних та енергетичних фреймів між каналами, що не пересікаються у діапазоні 2,4 ГГц

У японському обладнанні не перекривається з сусідніми також 14-й канал з центральною частотою 2484 МГц.

З метою запобігання інтерференції каналів, які передають дані та енергетичні фрейми має сенс розробити алгоритм пересилання енергії між WiFi-компонентами об'єктів БЕІВМ, за яким дані та енергія передаватиметися на каналах, що не пересікаються, та у різні проміжки часу (рис. 6.11).

Для таких наборів частот в кожному з них пропонується передавати дані одним каналом, який є центральним у кожному наборі – Ch. 6, 7 або 8 (див. рис. 6.10). Як показано на рис. 6.10, енергетичні фрейми у такому випадку можна передавати двома іншими непересічними каналами відповідного набору (відповідно, 1-му та 11-му, 2-му та

12-му, 3-му та 13-му). Причому, враховуючи, що передача накопиченої потужності вимірювань $W_{ij}^{rec}(t_1)$ згідно з TimeLine передусе моменту передачі даних $D_{ij}(t_2)$, а передача потужності $W_{ij}^{rec}(t_3)$ здійснюється після передачі даних, рівень акумульованої потужності $W^{acc}(t_3)$ буде вищим ніж $W^{acc}(t_1)$. До моменту t_3 можна приурочити вимірювання іншими датчиками (не п'єзоелектричними), фотозйомку або інші технологічні режими, для виконання яких достатній результуючий рівень накопиченої потужності W^{mod} .

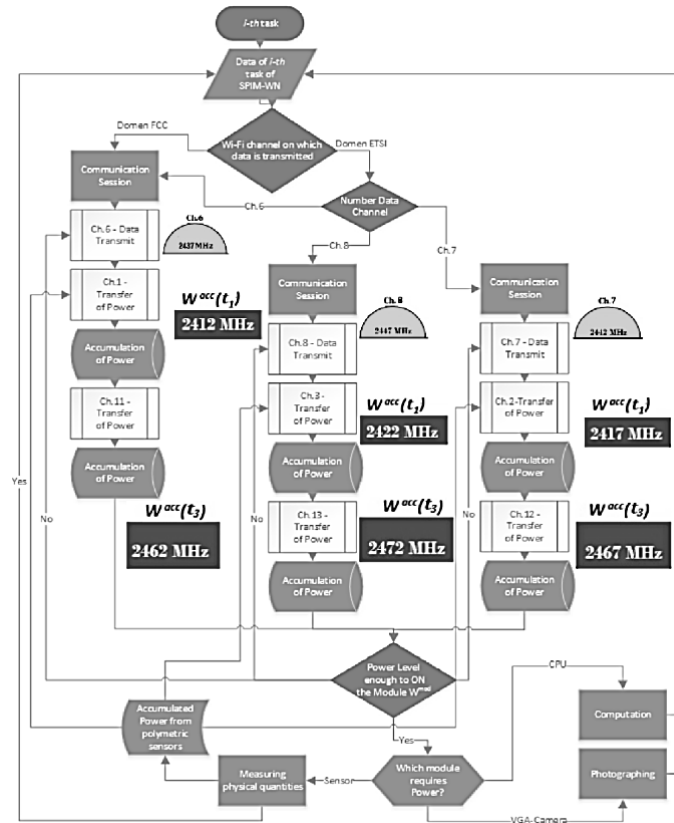


Рис. 6.11. Блок-схема алгоритму розподілу за каналами та часом передачі даних та енергетичних фреймів через Wi-Fi

З алгоритму рис. 6.11 можна побачити, що надлишкова енергія W_{ij}^{acc} , наприклад, накопичена вимірювачами на інших КФО, передана через канали зв'язку під час виконання i -го завдання на j -му ядрі. Якщо ця енергія є достатньою для включення модуля та виконання ним власних функцій, то процес накопичення енергії під час сеансу зв'язку закінчується, і названий функційний модуль (сенсор, камера, обчислювач тощо) починає виконання своїх специфічних функцій (вимірювання фізичних величин, зйомку, обробку даних, маневрування тощо):

$$W_{ij}^{acc}(t_i) = \sum_{i=1}^m W_i^{rec} \rightarrow \max, W_{ij}^{mod} \quad (6.2)$$

У запропонованому алгоритмі також передбачено, що модулі БЕІВМ, які живляться від надлишкової енергії інших КФО, включатимуться лише після накопичення

енергії достатнього рівня. У інших випадках при нестачі струму, наприклад, для живлення модулів зв'язку, не буде відгуку на будь-яких швидкостях, навіть при зниженні рівня швидкості до мінімуму (9,6 кбіт/с).

При використанні вищеописаного алгоритму деградації даних, переданих через Wi-Fi, та деградації продуктивності мережі не відбувається.

Розглянемо розвиток математичної моделі диспетчеризації завдань між обчислювачами об'єктів БЕІВМ з обмеженими енергоресурсами.

Необхідно враховувати, що інноваційні технології, які з'явилися в останні роки та пов'язані з отриманням, накопиченням, передачею енергії бездротовим способом, мають на увазі досить обмежений рівень потужності. Хоча такої потужності і достатньо для живлення обчислювальних модулів БПЛА, але все ж питання енергоефективності обчислень мобільних пристроїв залишається актуальним, тому що кількість функцій БПЛА зростає за рахунок різноманітності вимірюваних величин, ускладнення задач фотозйомки у різних умовах. Це підвищує навантаження обчислювача і витрати на електроживлення додаткових модулів. Зменшення споживання енергії є найважливішим завданням при розробці та використанні сучасних процесорів. Оптимізація роботи обчислювачів пов'язана з вирішенням цілого ряду завдань, які дозволяють підвищити їхню економічну ефективність на основі методів покращення енергоживлення. Це питання є доволі актуальним саме для обчислювачів БПЛА, коли перехід в енергозберігаючий режим здатний значно підвищити час польоту.

Один з механізмів економії електроенергії полягає в тому, що коли система знаходиться в режимі очікування, вона може бути в певний момент переведена в малопотужний стан споживання енергії (стан сну) [97]. Але при цьому проблемою є те, що зворотній перехід зі стану сну в активний режим вимагає додаткових витрат енергії.

У загальних принципах та сучасних апаратних рішеннях для підвищення енергоефективності комп'ютерів передбачено планування розподілу завдань по вузлах і керування чергою може бути джерелом значної додаткової економії електроенергії [222].

Тому доцільно сформулювати і реалізувати новий алгоритм енергоефективної роботи модулів БЕІВМ з диспетчеризацією завдань та з використанням стану сну модулів.

Розглянутий підхід змушує дещо змінити існуючу математичну модель процесу диспетчеризації завдань з переведенням системи у малопотужний стан споживання електроенергії, яка була запропонована у роботі [155] та мала вигляд:

$$\begin{cases} \sum_{i=1}^m \sum_{j=1}^n x_{ij} \cdot t_{ij} \rightarrow \min \quad \vee \quad \sum_{i=1}^m \sum_{j=1}^n x_{ij} \cdot t_{ij} \leq t_{\text{lim}}, \\ V_i^{RAM} \leq x_{ij} \cdot V_{ij}^{RAM}, \\ \left(x_{ij} \cdot W_{ij}^{\text{нак}} \right) \rightarrow \max, W_{ij}^{\text{нак}} > 0, \\ W_{ij}^{\text{мод}} \leq W_j^{\text{нак}} + W_{ij}^{\text{поступ}}. \end{cases} \quad (6.3)$$

де t_{ij} – час завершення i -го завдання на j -му модулі; V_{ij}^{RAM} – обсяг оперативної пам'яті j -го модуля для виконання i -го завдання; $W^{\text{мод}}$ – енергоспоживання модуля (загальна енергія, споживана віддаленим модулем об'єкта БЕІВМ); W^{acc} – накопичена енергія (accumulated); W^{rec} – енергія, що надійшла (received); x_{ij} – елемент матриці оптимального розподілу завдань, який може приймати такі значення:

$$\begin{cases} x_{ij} = 0, i - \text{th task isn't appoined to } j - \text{th module,} \\ x_{ij} = 1, i - \text{th task is appoined to } j - \text{th module.} \end{cases}$$

Зазначена модель потребує розвитку, має бути розширеною наведеними вище залежностями (6.2) та не передбачати режиму сну окремих ядер під час виконання завдання. За модифікованим підходом j означає номер CPU-ядра на однокристальному q -му процесорі з n ядрами. Енергія, акумульована під час виконання i -го завдання, яка надійшла до q -го процесора від PD –поліметричних датчиків або від іншого КФО:

$$W_{iq}^{acc}(t_i) = \sum_{pd=1}^{PD} W_{pd}^{rec} \rightarrow \max, W_{iq}^{mod}, \quad (6.4)$$

де $q = 1, \dots, z$ (z – кількість КФО у БЕІВМ).

Тоді модернізована математична модель диспетчеризації завдань між ядрами однокристального процесора набуває такого вигляду:

$$\left\{ \begin{array}{l} W_{iq}^{acc}(t_i) = \sum_{pd=1}^{PD} W_{pd}^{rec} \rightarrow \max, W_{iq}^{mod} \\ (x_{ij}, W_{iq}^{acc}) \rightarrow \max, W_{iq}^{acc} \geq W_{iq}^{mod} \\ x_{ij} = 0, i - \text{th task isn't appoined to } j - \text{th CPU core,} \\ x_{ij} = 1, i - \text{th task is appoined to } j - \text{th CPU core.} \end{array} \right. \quad (6.5)$$

Однією з відомих технологій для зниження споживання (або покращення енергоживлення) енергії є динамічне масштабування швидкості (напруги і частоти, DVFS Scheduling Algorithms) процесора, при використанні якої процесор може варіювати швидкість динамічно. Використовується виробниками процесорів як для десктопних систем (Intel, AMD і IBM) так і процесорів для мобільних пристроїв.

Традиційні рішення, використані вендорами однокристальних обчислювачів для мобільних систем, передбачають виконання різних завдань на різних, спеціалізованих за типами завдань, типах процесорів та ядер. З метою балансування навантаження багатоядерного процесора авторами зроблена спроба розробити математичну модель та алгоритми її реалізації з використанням одних і тих же ядер для виконання різних завдань.

Високі швидкості виконання завдань призводять до високої продуктивності, але при цьому має місце і високе споживання енергії. Саме тому в багатоядерних системах доцільно здійснювати динамічне масштабування швидкості при зменшенні середнього коефіцієнта завантаження $u_{k,i}$ процесора. При цьому обчислення не виконуються за найшвидші терміни при максимальному завантаженні процесора, а його швидкість S регулюється на підставі поточного завантаження і обмежень продуктивності. Можна показати, що оптимізація енергоспоживання має місце у випадку, якщо всі завдання, подані на рішення на процесор за певний проміжок часу $e_{k,i}$, виконуються в директивні терміни $d_{k,i}$, тобто без запізнювання.

Серед мікропроцесорів, що працюють зі змінною швидкістю, можна назвати процесори Qualcomm Snapdragon.

Математичні моделі та алгоритм розподілу завдань для багатоядерних систем однокристальних процесорів можуть бути побудовані шляхом модифікації методів розподілу завдань між процесорами однорідних і неоднорідних обчислювальних кластерів [296]. Базова математична модель виглядає таким чином:

$$\begin{aligned} P_k \sim S_k^s T_k &= \{t_{k,j}(e_{k,i}, d_{k,i}), i = 1, \dots, n_k\} \\ d_{k,i} &\leq d_{k,i+1} \\ u_{k,i} &= \frac{\sum_{j=1}^i \square e_{k,j}}{d_{k,i}} \\ S_k &= \max_{i=1}^{n_k} \{u_{k,i}\} \\ v_k &= \min_{i=1}^m \{S_i \geq S_k\} \\ u_{k,i} &= \frac{\sum_{j=1}^i \square e_{k,j}}{d_{k,i} - r_{k,i}}, \end{aligned} \quad (6.6)$$

де P_k – потужність, споживана процесором; S_k – швидкість процесора; T_k – безліч незалежно виконуваних завдань; v_k – поточна напруга процесора; V_i – режим роботи процесора; $u_{k,i}$ – поточний коефіцієнт використання (сума тривалостей завдань, загальне завантаження, віднесені до директивного терміну завершення останнього завдання).

Для порівняльного аналізу базового та модифікованого алгоритмів розглянемо набір завдань T_k , де їхні характеристики представлені у табл. 6.2, осцилограми – на рис. 6.12а. У базовому методі [269] співставлені режими роботи процесора (4 рівня) і відповідні відносні швидкості (частоти) процесора.

Але для однокристальних систем програмним шляхом неможливо обрати режим роботи процесора мобільного пристрою на відміну від десктопної версії, для якої були створені вищеописані алгоритми. Тому перспективним є використання іншого підходу до масштабування швидкості, а саме – ефективне навантаження ядер процесора для зменшення енергозатрат об'єкта. Тому з базової моделі [269] виключене поняття режиму роботи процесора.

Частково використовуючи в межах аналогії, яка може бути використана між багатопроцесорними та багатоядерними комп'ютерними системами, алгоритми диспетчеризації завдань багатопроцесорних систем, наведені в [155; 296], опишемо етапи модифікованого алгоритму розподілу завдань між ядрами однокристального процесора, який може бути використаний у мобільних об'єктах БЕІВМ.

На першому етапі всі завдання об'єкта (розрахунок рівня накопиченої виміральної енергії поліметричних датчиків, обробка вимірних фізичних величин, обробка аудіо- та відео- матеріалів тощо) сортується в порядку спадання (не зростання) їх директивних термінів.

На другому етапі відсортовані завдання розподіляються таким чином, що одне завдання розподіляється тільки на одне ядро.

На третьому етапі всі ядра упорядковуються за рівнем їхнього поточного навантаження.

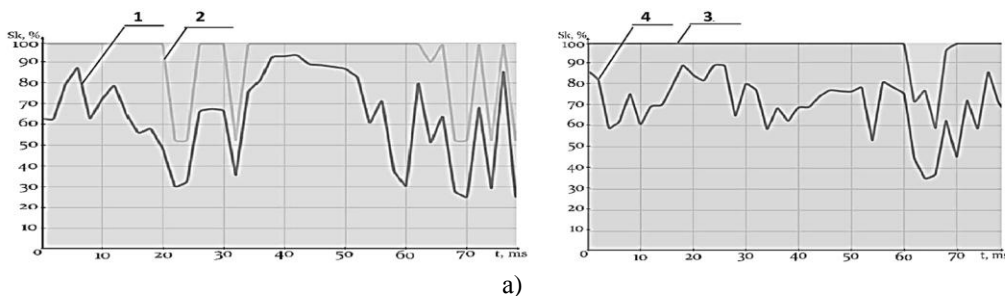
На четвертому етапі перевіряється достатність рівня енергії для початку обчислень.

На п'ятому етапі перше з нерозподілених завдань розподіляється на ядро, що має найменший рівень навантаження на момент розподілу, другий з решти – на ядро, наступне за рівнем навантаження доки не будуть рівномірно завантажені всі ядра.

Процедура повторюється допоки всі завдання вихідної множини не будуть розподілені по ядрах процесора, рівень акумульованої енергії не досягне необхідного до початку функціонування модуля.

Запропонований алгоритм розподілу завдань по ядрах не передбачає відключення та простою будь-якого з ядер CPU, що забезпечить економію енергії, яка б витрачалась на повторне включення ядра. Такий підхід дозволяє забезпечити оптимальне завантаження процесора, виконати всі завдання без черги та запізнювання.

Алгоритм було протестовано на процесорі Qualcomm Snapdragon 400, 4 ядра Krait (1,7 ГГц), на якому базується Microsoft Lumia 640 під ОС Windows Phone 8.1 (рис. 6.12б).



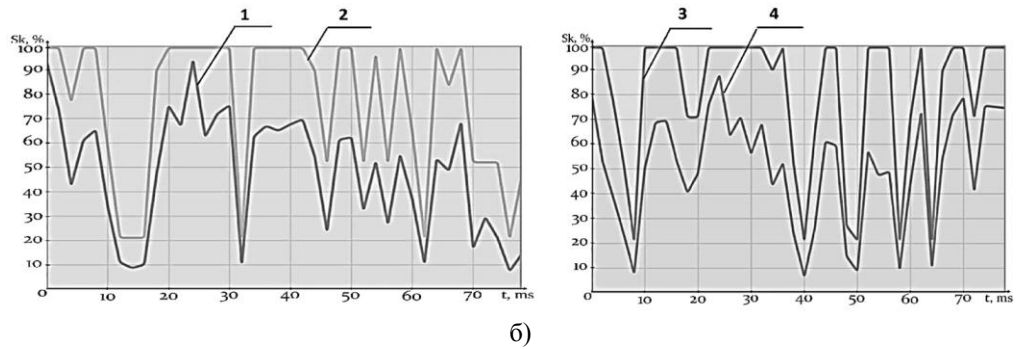


Рис. 6.12. Timeline навантаження 1–4 ядер CPU перед (а) та після (б) застосування алгоритму диспетчеризації завдань

У табл. 6.2 – табл. 6.3 наведені узагальнені експериментальні дані щодо часових показників (Timeline) навантаження процесора в розрізі ядер. Розраховані та графічно відображені середні показники навантаження кожного з 4 ядер. Обчислений коефіцієнт завантаження k_i процесора при використанні алгоритму диспетчеризації завдань та без нього.

У результаті проведених досліджень на математичній моделі із застосуванням алгоритму розподілу завдань між 4 ядрами Cortex-A7 (1,2 ГГц) процесора Qualcomm MSM8926 Snapdragon 400 було доведено, що використання запропонованого алгоритму дозволяє зменшити коефіцієнт навантаженості CPU на 27%. У такому разі загальне енергоспоживання об'єктами БЕІВМ, де використовуються такі процесори, теж суттєво зменшиться.

Ще більшої актуальності рішення набуває при зростанні кількості ядер у процесорах сучасних мобільних пристроїв (8 ядер 64-bit в Oukitel K6000 PRO Gray, MTK 6592 від MediaTek, Snapdragon 616 та 810 від Qualcomm та ін.) й можливістю розгону ядер до частоти 2,2–2,4 ГГц.

Таблиця 6.2. Навантаження CPU в розрізі ядер без планування завдань

№ ядра CPU	Timeline навантаження CPU, %							Середній коеф-т навантаженості
	70	50	65	95	85	30	50	
Ядро 1	70	50	65	95	85	30	50	64
Ядро 2	100	90	85	100	100	100	25	86
Ядро 3	100	100	100	100	100	100	100	100
Ядро 4	60	85	80	70	75	75	45	70

$$u_{k,i} = 0,78$$

Таблиця 6.3. Навантаження CPU в розрізі ядер з використанням планувальника завдань

№ ядра CPU	Timeline навантаження CPU, %							Середній коеф-т навантаженості
	65	70	10	65	60	30	20	
Ядро 1	65	70	10	65	60	30	20	46
Ядро 2	70	100	100	100	90	50	50	80
Ядро 3	25	70	100	20	20	20	100	51
Ядро 4	10	40	60	10	10	10	60	29

$$u_{k,i} = 0,51$$

Подальший розвиток принципів подовження часу енергонезалежності модулів БЕІВМ може збільшити ефективність віддаленого керування безпілотними АС з ушкодженими функційними модулями. Такий підхід дозволить вирішувати певні технічні проблеми безпілотних АС без зупинення каравану (або стаї, тобто сукупності всіх елементів БЕІВМ) та без направлення у зону поточного місцезнаходження комплексних ремонтних бригад завдяки деталізації проблем та визначення фахової спрямованості необхідних спеціалістів, що в умовах військових дій збереже життя військових та цивільних фахівців.

Досягнення оптимального балансу між високою продуктивністю і тривалим часом автономної роботи – важливе завдання при розробці принципів роботи БЕІВМ. Подальшого успіху в цьому напрямку можна досягнути із застосуванням технології гетерогенних обчислень. Ця теорія заснована на розподіленні навантаження за різними функціональними ядрами різних процесорів однокристальних плат – CPU, GPU, DSP. Обчислювально-інтенсивні операції з CPU мають передаватись на виконання в DSP (нп., мультимедійні операції над звуком та зображеннями), який справляється з ними куди ефективніше. За попередніми обчисленнями створення алгоритмів одночасного використання CPU і GPU в однокристальних системах Snapdragon дозволить знизити рівень енергоспоживання до 40 %.

Таким чином, алгоритмічна підтримка гетерогенних обчислень (коли використовуються можливості і багатоядерних CPU, і спеціалізованих DSP або GPU одночасно) здатна забезпечити необхідну гнучкість і підвищити продуктивність мобільної однокристальної системи при зниженні енергоспоживання і тепловиділення.

Висновки до розділу 6

Під час використання БПЛА всі стикаються з необхідністю мати стабільно працюючу систему, особливо при можливості критичного застосування БПЛА. Вкрай важливим є миттєве відпрацювання обчислювальною системою БПЛА необхідних змін курсу у випадку появи перешкод, втрати зв'язку з оператором, різкому падінні заряду батареї та ін. У такому разі алгоритми розвантаження 1-го ядра, на якому зазвичай виконуються всі системні процеси, від обрахування моніторингових та функціональних завдань можуть принести безперечну користь.

Отримані результати досліджень розроблених алгоритмів диспетчеризації завдань на 4-ядерному процесорі дозволили визначити як оптимальний алгоритм резервування ресурсів 1-го ядра багатоядерного однокристального CPU для обчислень першочергової важливості. Застосування такого алгоритму обчислень сприяє підвищенню ефективності використання БПЛА. Так, наприклад, для DJI Phantom 4 розподіл завдань лише на 2–4-те ядро зменшує енергоспоживання на 10,3 % та забезпечує подовження часу польоту на 3,1 хв. Навіть при немаксимальній швидкості польоту цієї моделі БПЛА у 50 км/год (Режим Р) [137] це збільшить дальність професійної аерофотозйомки на 1,3 км з урахуванням повернення на базу.

У подальшому необхідно провести аналіз енергозатрат під час виконання різних за якістю завдань з використанням запропонованого алгоритму диспетчеризації у порівнянні з автопланувальником деяких моделей CPU. Повна автоматизація поядерної диспетчеризації ускладнюється тим, що кожен алгоритм запускається автономно, тобто немає можливості вибору між режимами найкращого таймінгу або найменшого енергоспоживання.

Також у перспективі доцільно провести дослідження впливу розподілення обчислень між ядрами CPU для БПЛА на базі різних моделей однокристальних процесорів.

У цьому розділі було сформульовано проблему збереження енергії для мобільних об'єктів БЕІВМ, розглянуто шляхи вирішення зазначеної проблеми засобами оптимізації

навантаження процесора. Були визначені підходи та алгоритми, які дозволили зменшити коефіцієнт завантаженості CPU на 27 % та, відповідно, енергоспоживання об'єкта БЕІВМ на 1,5 % шляхом ефективного розподілення завдань між ядрами процесора, а потім підбору необхідної тактової частоти роботи для виконання цих завдань (процес масштабування швидкості процесора). Доцільність та ефективність використання запропонованих алгоритмів доведена експериментальними дослідженнями розподілу завдань між ядрами чотириядерного процесора Qualcomm Snapdragon 400. Розглянуті алгоритми легко реалізуються програмним шляхом.

У разі отримання від власників БЕІВМ або центру управління вказівок щодо блокування ушкодженого функційного модуля засобами електровипромінювання, ефект накопичення енергії за рахунок використання поліметричних датчиків на об'єктах БЕІВМ дозволяє зекономити до 3 % енергії батареї БПЛА та використати енергію від вимірювальних сигналів у технологічних цілях, у т. ч. шляхом бездротової передачі енергії іншим працездатним об'єктам БЕІВМ. Запропонований алгоритм забезпечує чергування інтервалів передачі даних та енергії за технологією PoWiFi з використанням різних груп Wi-Fi каналів, що не перетинаються, згідно з правилами доменів FCC та ETSI.

Подальший розвиток досліджень в розглянутому напрямку передбачає модернізацію математичної моделі диспетчеризації для організації гетерогенних обчислень, коли завдання будуть рівномірно розподілятися між CPU-ядрами і спеціалізованим DSP одночасно. Це дозволить підвищити продуктивність обчислювачів об'єктів БЕІВМ на мобільних однокристальних системах у разі зниження енергоспоживання і тепловиділення, що збільшить термін життя розглянутої комп'ютерної системи.

Розділ 7

ПРИКЛАДНІ РІШЕННЯ

З ВИКОРИСТАННЯМ РУХОМОЇ

МОНІТОРИНГОВОЇ МЕРЕЖІ

7.1 Відслідковування маршрутів БПА РММ на тривимірних картографічних моделях місцевості

Увесь час життя тривимірних картографічних моделей можна поділити на чотири частини:

1. Створення. Полягає у виготовленні моделі «з чистого аркуша» або перетворення (напівавтоматичну конвертацію) закінченої цифрової мапи в тривимірну модель.

2. Редагування. Редакторський перегляд і виправлення помилок виконавця і похибок конвертації.

3. Підготовка до публікації та публікація. Перенесення оригінальної моделі в загальнодоступний формат, що підтримується платформою і програмним забезпеченням пристрою виведення.

4. Використання. Споживання корисних властивостей моделі.

Створення програми, що за своїми характеристиками конкуруватиме з відомими сьогодні комплексами і основною функцією якої буде побудова картографічних моделей – справа економічно неприйнятна. Тому виходом із ситуації, що склалася, є пристосування наявного програмного забезпечення для потреб тривимірної картографії.

Можна виділити 5 груп програм, що певною мірою можуть задовольнити потреби картографії у відображенні на рухомих пристроях подій на місцевості. Кожна така група має назву, що закріпилася за нею під час виникнення програм-конкурентів, які мали подібні властивості до програм-флагманів (перших у своєму класі). Всередині групи вони мають схожий інтерфейс (подібно взаємодіють з користувачем) і майже однаковий принцип роботи. Але набір функцій суттєво відрізняється.

У всіх зазначених програмах використовується один або поєднання декількох видів триангуляції. Найбільш прийнятною для задач картографії вважається триангуляція Делоне [338].

Найпоширеніші алгоритми побудування триангуляції Делоне були порівняні за трудомісткістю та часом роботи на 10 тис. точок (табл. 7.1). За авторською оцінкою складності реалізації було обрано ітеративний алгоритм з динамічним кешуванням пошуку.

Таблиця 7.1. Тестування продуктивності алгоритмів побудування триангуляції Делоне [338]

Назва алгоритму	Час роботи на 10 000 точок	Оцінка простоти реалізації («5» макс.)
Простий ітеративний алгоритм	5,80	5
Ітераційний алгоритм «Видаляй і будуй»	8,42	2
Ітеративні алгоритми з індексуванням пошуку трикутників	9,23	3
Алгоритм «Розділяй і володай»	2,79	4
Ітеративний алгоритм з індексуванням центрів трикутників 2D-деревом	7,14	3
Ітеративний алгоритм з індексуванням центрів трикутників квадродерева	1,68	5

Закінчення таблиці 7.1.

Ітеративні алгоритми з кешуванням пошуку трикутників	1,49	5
Ітеративний алгоритм зі статичним кешуванням пошуку	3,60	5
<u>Ітеративний алгоритм з динамічним кешуванням пошуку</u>	<u>2,61</u>	<u>5</u>
Ітеративні алгоритми зі зміненим порядком додавання точок	1,93	4
Ітеративний смуговий алгоритм	5,01	4
Ітеративний квадратний алгоритм	5,31	5
Ітеративний алгоритм з пошаровим згущенням	5,80	4

Google Static Maps API дозволяє отримувати зображення з Google Maps через HTTP запити, та не потребує встановлення фреймворків для тої чи іншої платформи. Google Static Maps API має певні ліміти при безкоштовному використанні:

- 25 тис. запитів на статичне зображення мапи протягом 24 годин 90 днів поспіль;
- максимальна роздільна здатність для зображення 640 x 640.

Google Static Maps API повертає зображення (GIF, PNG або JPEG) у відповідь на HTTP-запит через URL. Для кожного запиту можна вказати місце розташування на мапі, розмір зображення, масштаб, тип мапи, і розміщення додаткових маркерів у місцях розташування. Можливо також додаткове маркування з використанням букв і цифр.

Для отримання зображення сегмента мапи за допомогою Google Static Maps API необхідно створити URL за такою формою [65]:

<https://maps.googleapis.com/maps/api/staticmap?parameters>.

Наступний приклад генерує статичне зображення для міста Миколаїв (рис. 7.1):



Рис. 7.1. Зображення сегмента території, отримане за допомогою Google Static Maps API [65]

Окрім 2D-картографії для рухомих пристроїв також можливо 3D-відображення переміщення об'єкта. З розглянутих у розділі 1 алгоритмічних рішень та програмних застосунків для РММ не було знайдено тривимірних мап, які задовольняють всім вимогам для застосування на пристроях з обмеженими обчислювальними можливостями. Хоча бібліотеки і Google Maps API, і Map Kit надають зручні фреймворки, що вже включають в себе тривимірні мапи, додавання міток, побудову маршруту та ін., однак, в результаті дослідження виявилось, що тривимірні мапи і від Google, і від Apple покривають далеко не всю територію та є досить приблизними.

Розглянемо докладніше процес створення власних тривимірних мап для забезпечення функціонування гетерогенних комп'ютерних мереж на базі безпілотних апаратів.

Наприклад, для операційної системи iOS існує три базові програмні засоби роботи з 3D-графікою: OpenGL, Metal та SceneKit.

OpenGL ES – підмножина графічного інтерфейсу OpenGL, розроблена спеціально для вбудованих систем: мобільних телефонів, кишенькових комп'ютерів, гральних консолей. OpenGL ES визначається і просувається консорціумом Khronos Group, до якого входять виробники програмного і апаратного забезпечення, зацікавлені у відкритому API для графіки і мультимедіа.

Metal – графічний API від компанії Apple, який переносить код якомога ближче до GPU, наскільки це можливо, щоб максимізувати продуктивність. Metal призначений для потужних застосунків, яким необхідно більш потужні обчислювальні ресурси. Він у 10 разів продуктивніший від OpenGL ES.

SceneKit – це графічний інтерфейс прикладного програмування (API) для Apple Inc. платформ, написаний на Objective-C. Він являє собою компроміс між продуктивністю та зручністю розробки ПЗ. SceneKit містить графічне ядро для побудови видимих об'єктів на сцені та фізичне ядро для моделювання фізичної взаємодії між об'єктами. SceneKit також підтримує імпорт і експорт 3D-сцен, які використовують формат COLLADA. SceneKit був вперше випущений для Mac OS у 2012 р., та для iOS у 2014 р.

Для цієї роботи було обрано SceneKit – компромісне значення між продуктивністю та складністю реалізації.

Для створення тривимірної поверхні було реалізовано алгоритм триангуляції: ітеративний алгоритм з динамічним кешуванням (рис. 7.2) та включені як обмеження структурні елементи триангуляції.

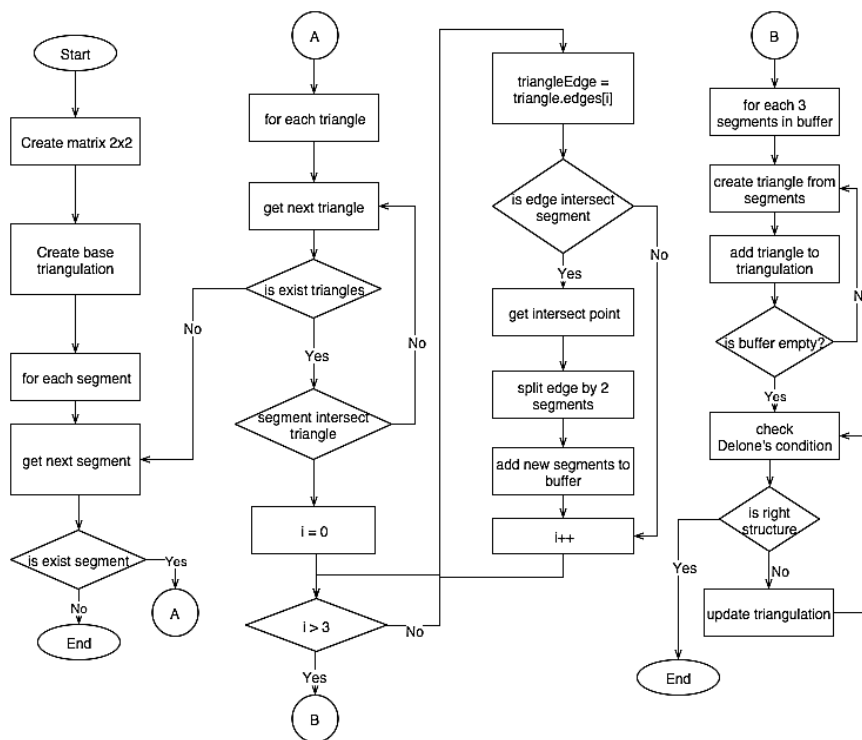


Рис. 7.2. Ітеративний алгоритм з динамічним кешуванням

Отже, розроблений ітеративний алгоритм з динамічним кешуванням пошуку для побудовання тривимірної мапи складається з таких кроків:

а) поверхня землі ділиться на 4 квадрати, згідно з структурою Quadtree. Крайні координати заносяться у структуру як кореневі;

б) при збільшенні масштабу до 1 км на дюйм починається завантаження мапи висот:

1) беруться координати які відповідають крайнім точкам на екрані, перевіряється, чи є інформація про них в кеші. При відсутності такої інформації посилається запит на сервер Google Static Map API, який повертає мапу висот та радіус, в межах якого зазначена висота є однаковою. Наприклад, на річках та озерах будуть великі радіуси, а в гірській місцевості – маленькі;

2) після першого запиту будується базова триангуляція на основі отриманої мапи висот;

3) після чого рекурсивно та асинхронно відправляються запити на отримання більш детальної мапи висот, та будується нова триангуляція на основі вищезгаданого алгоритму. При надходженні нових результатів, структура Quadtree розширюється, будуються нові гілки у вигляді менших квадратів, а дані зберігаються у кеш;

4) так продовжується до 5-го рівня деталізації, а саме до моменту, коли в структурі Quadtree від кореневого квадрату буде відходити 5 дочірніх гілок.

5) У разі збільшення масштабу алгоритм повторюється. При цьому з пам'яті видаляються крайні координати;

б) у разі зменшення масштабу координати на маленькій ділянці групуються в одну точку. Пам'ять при цьому звільнюється.

У цій реалізації алгоритму за рахунок динамічного перерозподілу координат під час масштабування, досягається використання 50–80 Мбайт оперативної пам'яті, що є прийнятним для iOS застосунків.

За результатами виконання зазначеного алгоритму сегмент території, наведений на рис. 7.1, виглядатиме таким чином (рис. 7.3):

Створений та реалізований алгоритм для побудови тривимірної картографічної поверхні на основі мапи висот і алгоритмів триангуляції забезпечує асинхронне завантаження мапи висот з Google API сервера. Така модифікація дозволяє поліпшити швидкість рендерингу картографічної поверхні до максимальних 55–60 кадрів на секунду.

Оптимізований алгоритм триангуляції картографічної поверхні для використання на мобільній платформі iOS дозволяє використовувати тривимірні мапи на БПА з обмеженими технічними даними. Оскільки Apple виділяє лише 70 MB оперативної пам'яті на неігрові застосунки, то описаний алгоритм дозволяє не перевищувати цей показник.

ПЗ автоматизованої побудови тривимірних мап для РММ складається з 4 основних діалогів, які надають функціональні можливості для моніторингу РММ, дозволяють переглядати необхідну інформацію, а також відправляти необхідні команди на сервер. Зовнішній вигляд розробленого ПЗ має стандартний дизайн для платформи iOS, всі графічні компоненти були створені за допомогою Interface Builder без кастомізації окремих компонентів.

Для повноцінного функціонування застосунку користувач смартфона має надати ПЗ дозвіл на використання GPS-даних, а також мати стабільний доступ до Інтернету. В іншому випадку програмний застосунок може працювати некоректно.

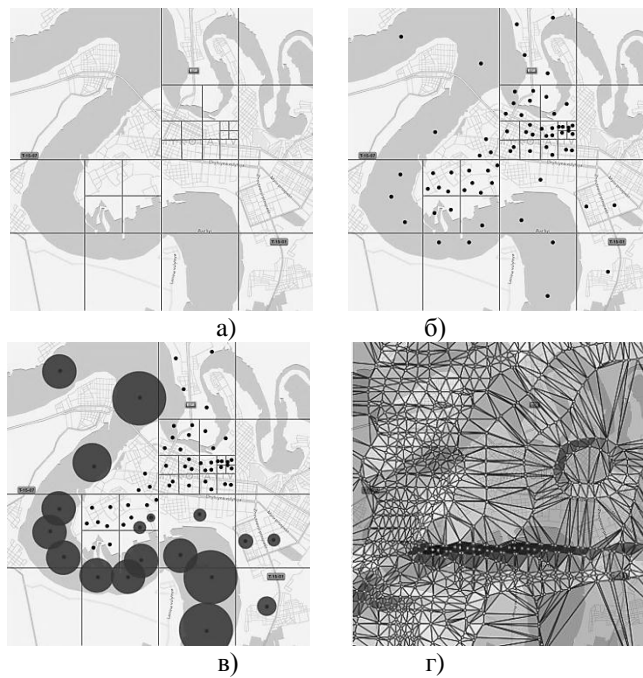


Рис. 7.3. Приклад покрокової роботи (а – г) ітеративного алгоритму триангуляції Делоне з динамічним кешуванням пошуку для сегмента мапи

Головне вікно програми розділене на три секції (Tabs). Перейти до кожної з секцій можна, натиснувши відповідну кнопку в нижній секції.

Основною секцією головного вікна є секція «Мар». Саме тут розміщений елемент мапи, на якому зображені всі БПА в РММ у вигляді маркерів. Графічний інтерфейс Map Controller зображено на рис. 7.4.

Маркери на мапі бувають трьох типів. Маркування кожного типу БПА відрізняються лише внутрішньою піктограмою.

Програмний модуль кожні 5 секунд оновлює інформацію про кожен БПА, і тим самим моделює їхнє переміщення на мапі. При додаванні нового БПА на мапу створюється окремий потік, у якому і перевіряється інформація з сервера.



Рис. 7.4. Секція «Мар» з маркерами БПА

У налаштуваннях секції «Мар» можна змінити масштаб мапи, ввімкнути або вимкнути опцію відображення маршруту переміщення БПА, а також вибрати елемент до якого можна виконувати переміщення операції «до центру мапи». За замовчуванням, центральним елементом є «user», тобто користувач смартфона. Центральним елементом можна визначити будь-який доступний БПА.

Наступною секцією головного діалогу є секція відеотрансляцій «Streams» (рис. 7.5). Якщо БПА неактивний (повзунок ліворуч, рис. 7.5а), користувач зможе переглядати архівні відео. Для цього, необхідне відео передається на сторону клієнта, і переглядається при необхідності (рис. 7.5б). Алгоритми оптимізації при цьому не застосовуються.

Також забезпечується можливість дивитись відео у режимі реального часу, якщо БПА активний і виконує зараз поставлене завдання (повзунок праворуч). При натисканні на ID БПА відкривається стандартний програвач потокового відео, якщо БПА оснащений камерою (рис. 7.6).

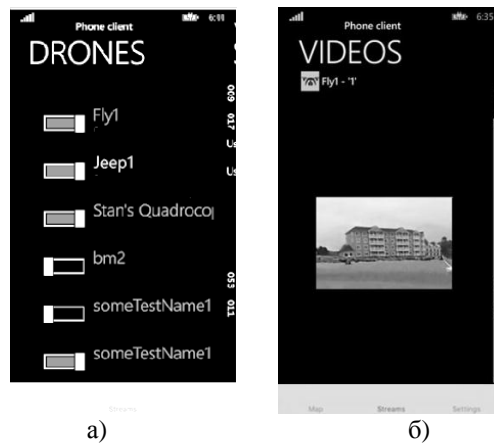


Рис. 7.5. Секція відео: перелік БПА суб-рою (а) та перегляд відео з БПА (б)



Рис. 7.6. Програвач потокового відео

Для БПЛА (дронів) також можна переглянути відео з дрону в режимі з реалізованою власною тривимірною мапою. Оскільки такі мапи не повністю протестовані та налаштовані під різні території, то було вирішено розділити стандартну мапу з бібліотеки MapKit та власну реалізацію тривимірних мап. Для цього було додано кнопку «3D-View» до головної секції Map Controller. При натисканні на кнопку відкривається вікно Map3DController (рис. 7.7) з урахуванням мапи висот та тривимірності місцевості. Цей контролер працює лише в режимі read-only, що означає мінімальний набір інтерактивних функцій. Для додавання або модифікації маршруту дронів необхідно повернутися до попереднього вікна та натиснути на піктограму дрона. Відкриється список всіх наявних дронів з можливістю оновити список та відкрити детальну інформацію щодо кожного.

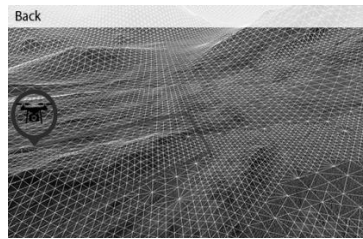


Рис. 7.7. Секція Map3DController

Останньою секцією головного вікна є секція налаштувань «Settings».

Тут необхідно вказати адресу URL-сервера, логін та пароль користувача, а також виконати команду з'єднання з сервером (рис. 7.8).

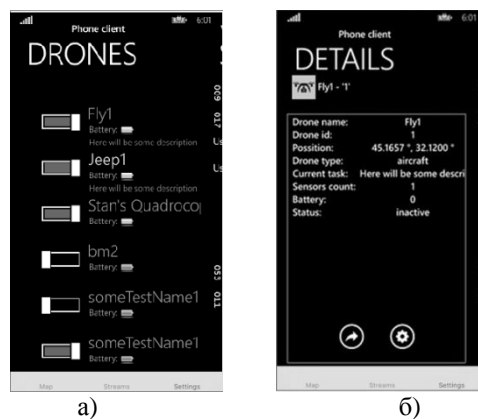


Рис. 7.8. Авторизація на секції налаштувань

Кнопка «З'єднання з сервером» буде недоступна до тих пір, поки користувач не введе валідні дані. Вони зберігаються в налаштуваннях ПЗ і використовуються при наступному запуску програми. Окрім цього, зберігається також і остання активна секція. З цієї секції розпочне роботу програма при наступному запуску.

У секції «Settings» також зберігається перелік в РММ усіх можливих БПА різних конструкцій (рис. 7.9а). Задається назва БПА (його ID) та помічається як заголовок відповідним до типу БПА кольором. Трохи нижче зазначено рівень батареї БПА. Наступним є опис поточної місії БПА.

Про кожен БПА можна отримати більш детальну інформацію (рис. 7.9б).



а)

б)

Рис. 7.9. Перелік БПА на підсекції DRONES: (а) та деталізована інформація про дрон з ID «Fly-1» (б) у складі суб-рою на підсекції DETAILS

Угорі підсекції рис. 7.9б представлена піктограма для дрону з ID «Fly-1», його ім'я та унікальний ідентифікатор у мережі. Нижче надана таблиця з наступною інформацією:

- ім'я;
- унікальний ідентифікатор;
- координата поточного положення;
- тип дрону;
- нинішнє завдання;
- статус батареї;
- статус дрону;
- кількість сенсорів.

Окрім інформаційного навантаження ця підсекція інших функцій не виконує.

Обраному БПА можуть бути відправлені команди оновлення певної інформації про цей БПА. На рис. 7.10 показано призначення нового завдання (нових координат положення) для дрону з ID «Fly-1».

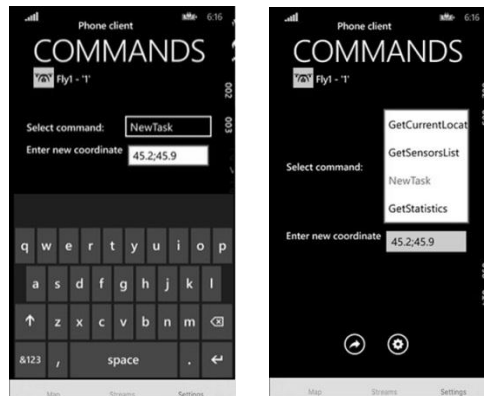
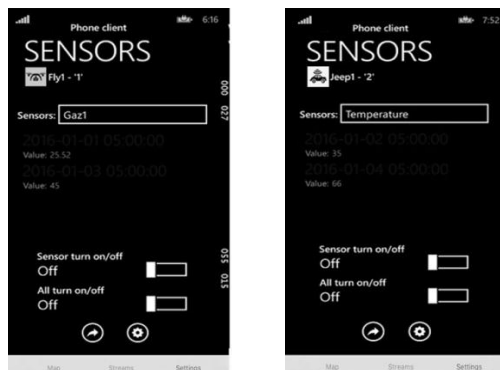


Рис. 7.10. Підсекція задання нових команд для дрону

Серед основних команд можна назвати:

- оновлення інформації про БПА;
- оновлення інформації про сенсори;
- оновлення статистики з сервера;
- відправка нової координати БПА.

У підсекції датчиків користувач може побачити інформацію, яку зібрав БПА за певні проміжки часу (рис. 7.11).



а)

б)

Рис. 7.11. Секція сенсорів: концентрації CO₂ (а) та температури (б)

Користувач може ввімкнути або вимкнути відповідний датчик на БПА (без видалення інформації, що вже була зібрана). Таким чином, можливо заборонити або дозволити БПА збирати необхідну інформацію. Підсекція дозволяє переглядати інформацію лише про один сенсор, тому потрібно обрати необхідний елемент в combobox перед тим, як оцінити зібрані значення.

Кожен БПА може бути виключений зі складу РММ. Ця операція відправить команду на сервер, з вимогою зупинити функціонування БПА і припинити виконувати завдання. Після цього БПА буде недоступний на мапі. Поновити його функціонування можна в будь-який час.

При натисканні пальцем на піктограму будь-якого БПА відобразиться новий діалог з детальною інформацією про БПА.

Таким чином, досліджено процеси побудови тривимірної картографічної поверхні на основі мапи висот і алгоритмів триангуляції. Вдосконалено алгоритми триангуляції, оптимізації рендерингу. Оптимізовано структури даних Quadtree для деталізації окремих ділянок тривимірної мапи та створення динамічних тривимірних мап для мобільних пристроїв.

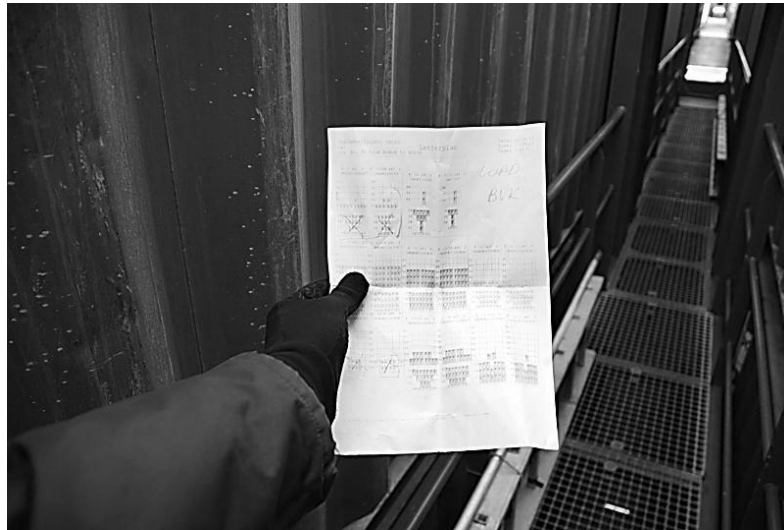
Описаний процес відслідковування маршрутів БПА РММ на мапі місцевості з автоматизованою побудовою тривимірних мап для РММ. Протестовано роботу ПЗ для визначеного сегмента місцевості.

7.2 Розробка алгоритмів ресстрації великорозмірних вантажів контейнерного типу за допомогою мобільних первинних ресстраторів

Незважаючи на зростаючу комп'ютеризацію водних транспортних перевезень, наразі процес обліку завантаження/розвантаження безпосередньо на судні відбувається вручну працівником вантажного комплексу – тальманом, який є робітником стивідорної компанії і надає інформацію про стан вантажу (рис. 7.12а) капітану судна тільки у кінці зміни у вигляді звичайного звіту, який називається тальманською розпискою (рис. 7.12б) [346].



а)



б)

Рис. 7.12. Ручний облік контейнеровозів на судні (а) з формуванням тальманської розписки (б) [308]

Компанії, які займаються перевезенням вантажу у контейнерах, несуть відповідальність за його збереження. Тому, екіпаж має забезпечити необхідну остійність судна, контролювати його крен та диферент. Зважаючи на це, капітан має потребу в отриманні детальної інформації щодо завантаження/розвантаження судна у реальному часі, під час стоянки у порту [37].

Таким чином, актуальним є питання автоматизації обліку процесу розвантаження і завантаження контейнеровозів для економії часу і ресурсів екіпажу, виключення похибок обрахунків, зумовлених людським фактором. Для досягнення зазначеної мети необхідно вирішити задачу надання капітанові судна у режимі реального часу на ноутбук/планшет або інший пристрій інформації про кількість вантажу у кожній секції контейнерів на судні. Визначити позицію кожного контейнера на судні можливо за допомогою трьох координат: *Bay* – секція, *Row* – ряд і *Tier* – ярус (рис. 7.13).



Рис. 7.13. Нотація обліку розташування контейнерів на палубі [192]

Необхідно дослідити архітектуру та процес побудови локальної WiFi-мережі, через яку вищезазначений пристрій, який знаходиться у капітана, має приймати дані від великої кількості пристроїв-реєстраторів, розміщених на судні та поєднаних у локальну мережу (рис. 7.14). Навантаженість каналів передачі даних у цій мережі та режими роботи таких пристроїв-реєстраторів є важливим предметом дослідження.

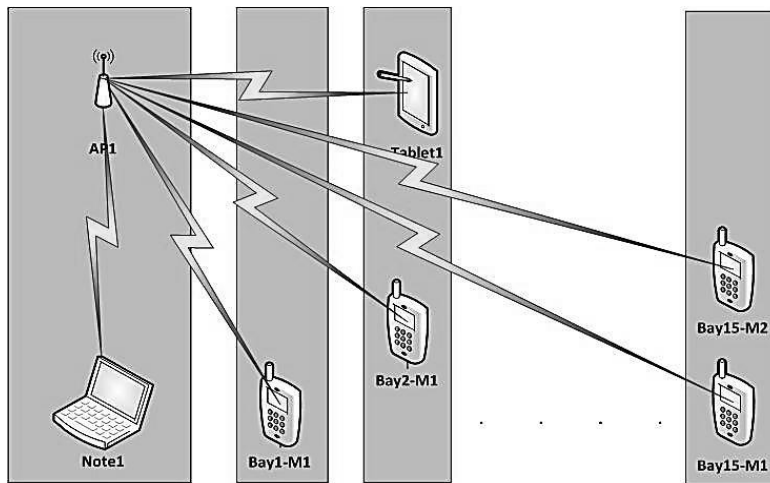


Рис. 7.14. Структурна схема локальної WiFi-мережі судна-контейнеровоза

Існує обчислювальний метод виявлення об'єктів, що рухаються, на серії CCD-кадрів, але він передбачає тільки незначні рухи біля положення об'єкта [95]. Існують також системи, які передбачають для розгортання подібної системи обліку створення бездротового маршруту передачі інформації «судно – вантажний комплекс» [162; 304]. Але, на жаль, підключення судна у WiFi-мережу порту не завжди можливо за низкою технічних обставин (нп., металеві конструкції виробничих терміналів). У такому випадку доцільно запропонувати рішення, за яким облікова система контейнеровоза працюватиме у власній локальній мережі, яке б не потребувало ані підключення до портів серверів, ані підключення до мережі Інтернет [253; 259; 305].

Для впровадження висловленої ідеї на даху контейнерів у кожній секції ряду на палубі, найближчого до причалу, запропоновано розмістити мобільний пристрій на ОС Android, який реєструє контейнер, коли стріла портокрану переносить його над фотокамерою девайсу (рис. 7.15).

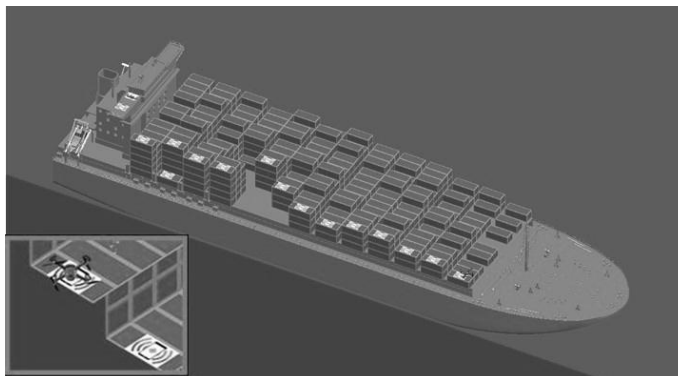


Рис. 7.15. Загальний вигляд судна-контейнеровоза з WiFi-реєстраторами

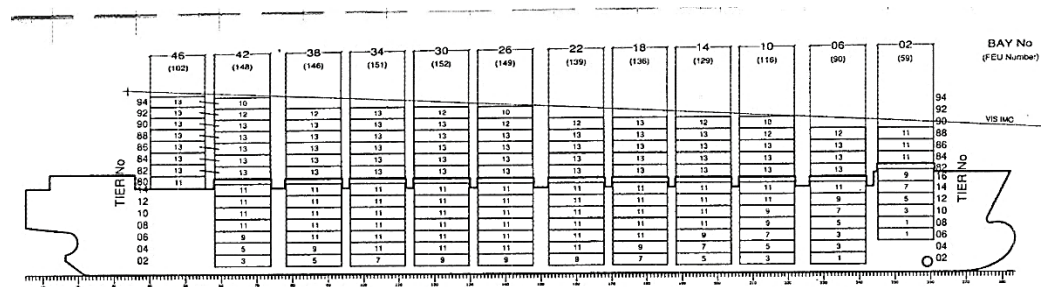
У якості засобу доставки мобільного реєстратора на верхній Tier крайнього Row можливе використання дрона з захватом (Grabber) [133]. На сьогодні вже існують декілька конструкцій спеціальних захватів для дронів, розроблених французькими, японськими та ін. фірмами (рис. 7.16).



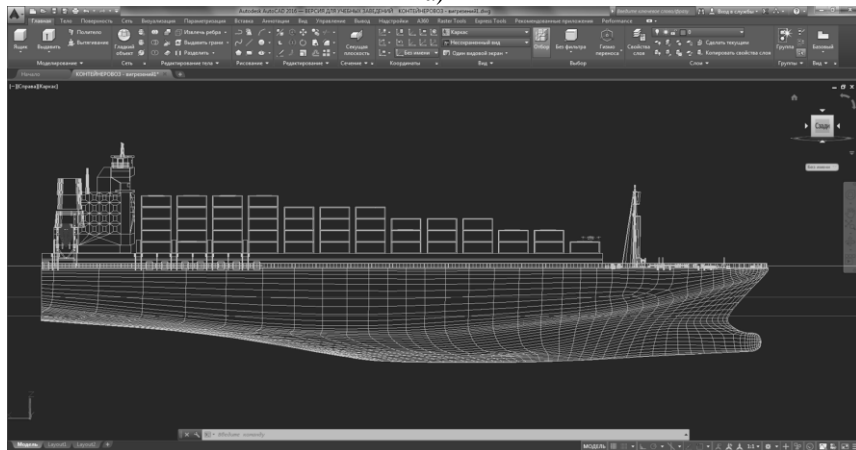
Рис. 7.16. Спеціалізовані захвати для дронів PRODRONE (а) [57], Parrot (б) [133], Mantis Drone Claw (в) [280]

При торканні реєстратором контейнера при спуску або мотор відкриває Grabber (керування здійснюється через USB-порт), або захват відкривається автоматично від удару при падінні.

Для підвищення ефективності експлуатації контейнеровозів кількість ярусів контейнерів на палубі може бути до 8–9 шт. Для розробки ПЗ автоматизованого обліку необхідно спочатку створити 3D-візуалізацію бей-плану розміщення контейнерів на судні (рис. 7.17).



а)



б)

Рис. 7.17. Бей-план контейнеровоза (а) та його 3D візуалізація (б)

У такому разі, при нерівномірному розвантаженні контейнерів у рядах, у кожній секції необхідно розмістити мобільні пристрої на дахах самого високого ярусу та на декількох ярусах проміжної висоти до мінімального (рис. 7.18).

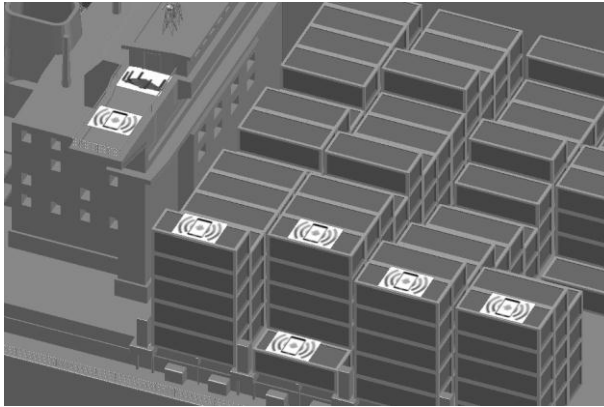


Рис. 7.18. Розміщення реєстраторів на декількох ярусах різної висоти

Методами дослідження є сукупність методів встановлення параметрів фотокамер мобільних пристроїв таким чином, щоб їхні лічильники спрацьовували в момент проходження стріли портокрану з контейнером, що за-/вивантажується, та змінені показання лічильників передавались на пристрій капітана через бездротову локальну мережу судна-контейнеровоза.

Значення лічильника кадрів одразу після зміни стану пристрою, тобто фіксації руху, передається через локальну WiFi-мережу судна на пристрій капітана, де відображається зменшення кількості контейнерів у відповідній секції (рис. 7.19).

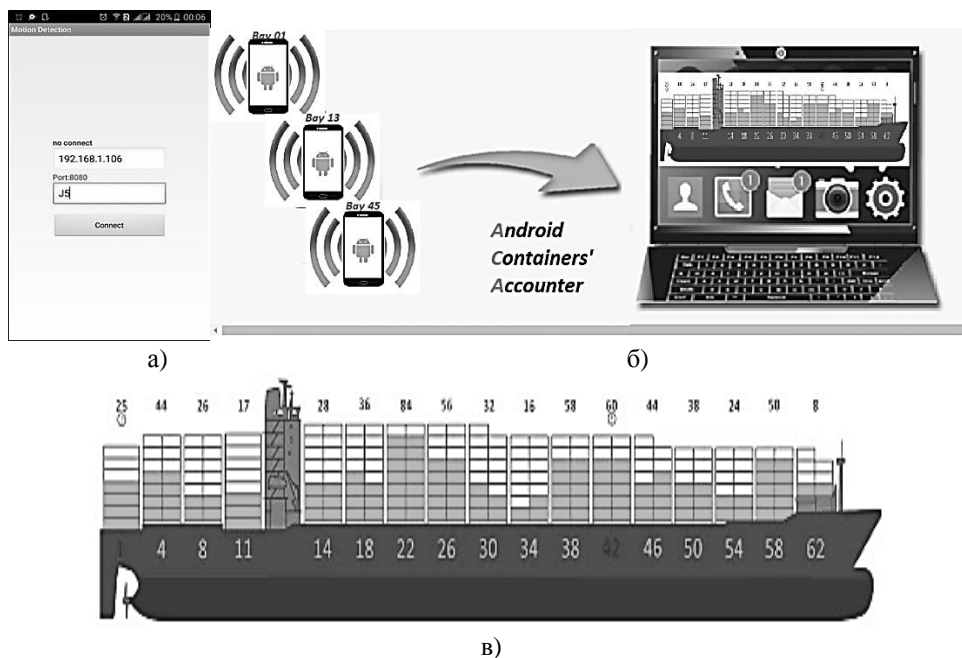


Рис. 9.19. Загальний вигляд графічного інтерфейсу на мобільному реєстраторі (а) та на пристрої капітана (б, в)

Коли мобільний пристрій разом з контейнером, на якому він розміщений, забирає портокран, квадрокоптер приносить у секцію новий пристрій на місце попереднього контейнера, найближчого до причалу (див. рис. 7.15). Дрон переміщується виключно над судном, не заходячи на територію порту, що узгоджується зі стивідорною компанією, яка обслуговує судно-контейнеровоз у порту.

Для реалізації описаного методу було розроблене ПЗ під керуванням ОС Android [130; 352]. Варто підкреслити, для задач реєстрації руху та передачі даних в описаній бездротовій моніторинговій локальній мережі достатньо мінімальних технічних характеристик і ресурсу акумулятора мобільних пристроїв. Так, досліджувані мобільні пристрої дали позитивні результати у процесі реєстрації великого вантажу над пристроєм вже на рівнях API Level 14-17, які відповідають версіям платформи Android 4.0-4.2 [218].

На даному етапі програма пристрою-реєстратора контейнерів працює за наступним алгоритмом.

Екран пристрою при працюючому мобільному застосунку стає максимально блідим, більше ніж це можливо зробити за допомогою кареток, що є елементами меню пристрою, а також виставлений у режим FLAG_KEEP_SCREEN_ON, що не дає екрану гаснути навіть після тривалої взаємодії з тач-скріном.

Оскільки оперативна пам'ять (RAM – Random Access Memory) у продуктивній версії системи пристроїв припускається якомога низька (до 512 Мбайт), повинна бути забезпечена висока продуктивність обробки зображень. Цього було досягнуто шляхом використання найменшої можливої роздільної здатності, яка виставляється для кожного пристрою програмним шляхом під час першого запуску пристрою. Це неухильно означає втрату якості зображення, проте вона є не першорядною для цієї системи.

У подальших версіях можливе виникнення потреби у більшій роздільній здатності зображень (наприклад, для кешування), тоді модифікація здійснюється не в сторону погіршення якості зображення, а розпаралелювання процесу обробки даних між потоками виконання.

Щоб фіксація руху предметів перед камерою була адекватною та об'єктивною, потрібно виключити вплив руху самого пристрою. Оскільки навіть вживані Android-пристрої мають акселерометр, бо мають можливість перевертання екрана при повороті пристрою, він є єдиним точним та чутливим сенсором, потрібним для встановлення режиму «У русі» для пристрою.

Підписавши слухача у вигляді SensorActivity на зміну сенсорів пристрою, у методі onSensorChange(SensorEvent) відбувається обробка даних, отриманих із event-об'єкта, що містить ці дані, зміна котрих і спричинила виклик методу.

Акселерометр повертає масив, що містить три значення у m/c^2 (всі значення у системі одиниць SI):

- values[0]: Прискорення мінус G_x по осі X;
- values[1]: Прискорення мінус G_y на осі Y;
- values[2]: Прискорення мінус G_z на осі Z.

Датчик цього типу вимірює прискорення, що діє на пристрій Ad . Він робить це шляхом вимірювання сил, що діють на сам датчик F_s , використовуючи співвідношення (7.1):

$$Ad = - \sum F_s / mass. \quad (7.1)$$

Зокрема, сила тяжіння завжди впливає на виміряне прискорення:

$$Ad = - g - \sum F / mass. \quad (7.2)$$

З цієї причини, коли пристрій лежить нерухомо та імовірно не прискорюється, акселерометр зчитує величину $G = 9,81 \text{ m/c}^2$.

Коли пристрій знаходиться у стані вільного падіння, а отже змінює прискорення у напрямку до землі на $9,81 \text{ m/c}^2$, його акселерометр зчитує величину 0 m/c^2 .

Шляхом проведення експериментів було встановлено, що приблизне значення, при якому значення прискорення змінюється хоча б на одній з осей, є «1» (одиниця). Тобто, якщо різниця минулого зафіксованого будь-якого із осьових значень є більшими за «1», то фіксація руху на зображенні камери не відбудеться.

Аналогічні процедури проводяться для пристроїв, що мають сенсор, сприйнятливий до змін значення магнітного поля. У цьому випадку, граничним значенням, який спиняє процес роботи фіксації камерою, є також «1» (одиниця), що отримано шляхом апроксимації експериментально отриманих даних. Фіксація руху великорозмірного вантажу відбувається у разі виявлення переміщення контейнера над мобільним пристроєм (рис. 7.20).



Рис. 7.20. Виявлення переміщення контейнера над мобільним пристроєм

Для отримання достовірних даних, потрібно було виключити випадок дублювання фіксації одного й того ж контейнера, що проходить за приблизно однаковий проміжок часу. Використаємо метод *callback*, котрий відповідає за отримання байтів із зображення, зробленого камерою. Тоді в *onPreviewFrame(byte[], Camera)* першим параметром є той самий масив байтів, який підлягає обробці, а другим – об'єкт камери, що фіксує зображення.

Частота виклику цього методу операційною системою безпосередньо залежить від FPS (frames-per-second), встановленому на пристрої програмно. Це є значення частоти отримання кадрів за секунду. Можливі значення було отримано програмним шляхом із списку доступних FPS у пристрої – з методу *Camera.getSupportedFps()*.

UML-діаграма класів клієнтського застосунку пристрою-реєстратора наведена на рис. 7.21.

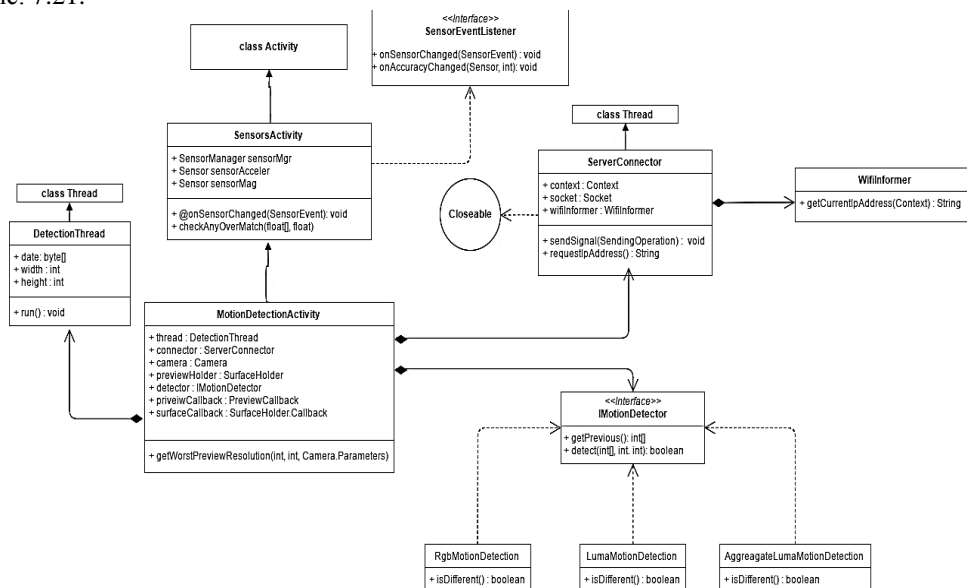


Рис. 7.21. Діаграма класів клієнтського застосунку програми Motion Detector

Експериментально було встановлено, що частоти у хоча б 1 кадр/с досягти неможливо програмним шляхом. Тому було вирішено налаштувати не частоту отримання кадрів, а лише проводити обробку зображення. Цей процес вміщує у собі конвертацію пікселів із формату YUV420SP у ARGB, перевірку даних акселерометра на нерухомість пристрою і перевірку схожості зображень попіксельно, щосекундно або за інші проміжки часу, які можна встановити програмно. «Старіння» даних при виконанні розрахунків для фіксації руху є неможливим, оскільки системний метод *onPreviewFrame* викликається із великою частотою FPS, що набагато більша за бажану частоту замірів, передаючи на обробку «свіжі» дані.

Оригінальне візуальне оформлення програми, що відображає зміни на сервері для особи, відповідальної за завантаження/розвантаження контейнерів, включає в себе анімаційні ефекти, які безпосередньо залежать від даних, які надходять від пристроїв-реєстраторів. UX побудований досить ергономічно і компактно, одночасно і функціонально, оскільки дає можливість користувачеві своєчасно отримати інформацію не тільки про завантаженість беїв, а й про доступність пристроїв-реєстраторів, якості їхнього зв'язку з сервером.

Працездатність та завантаженість моніторингової облікової WiFi-мережі контейнероза була оцінена шляхом математичного моделювання зазначеної мережі за допомогою зв'язного графу [101; 206; 252].

Мережі Ad-Hoc, або як їх ще називають бездротові самоорганізовані мережі, є децентралізованими бездротовими мережами, де всі вузли є рівноправними з точки зору управління мережею [76]. У таких мережах відсутні виділені сервери, де кожен вузол використовується або як клієнт, або для виконання функцій сервера, що дозволяє зберігати працездатність мережі при будь-якій кількості доступних вузлів. При функціонуванні мережі з різних причин можуть губитися зв'язки між окремими вузлами, тобто конфігурація мережі непостійна і, найчастіше, непередбачувана.

Для мережі такої змінної топології доцільно використовувати математичну модель невизначеності структури мережі як випадковий граф [295]. У такому разі оцінюються теоретико-ймовірнісні характеристики потоків у випадкових графах [88].

Представимо мережу у вигляді зв'язного графа $G = (V, U)$, де V – множина вершин або вузлів, U – множина ребер, та змінної множини W повідомлень (пакетів), які генеруються в ході роботи мережі. Розглянемо мережу з фіксованою кількістю вузлів n . Інші складові мережі вважатимемо випадковими.

Множина графа $G = (V, U)$ формується наступним чином: генерується випадкове число $|U|$ – потужність множини вершин, виходячи з дискретного розподілу ймовірностей PU . Далі з множини $V \times V \setminus \text{diag}(V \times V)$ рівноймовірно обирається $|U|$ різних вершин, які утворюють випадкову множину U' . Утворений граф $G = (V, U')$ – зв'язний (від дерева з однією вершиною до дерева з n вершинами), тоді $U' \in U$.

У мережі кожний вузол $v_i, i = 1 \dots n$ є джерелом потоку повідомлень w_i до певного кореневого вузла $v_j, j = 1 \dots n, j \neq i$, в якому повідомлення цього потоку мають оброблятися. Потік w_i – це випадковий дискретний процес з кінцевою кількістю подій передачі повідомлень. Кількість подій в потоці – випадкова величина з дискретним розподіленням ймовірностей PS . Її значення не перевищує числа P . Інтервали часу τ_i між послідовними подіями в потоці описуються функціями розподілу ймовірностей $A_i(x)$ – скільки вузлів буде задіяно для передачі повідомлення.

Первинні процеси w_i (проходження контейнера над вузлом-реєстратором) породжують потоки дублюючих повідомлень, кожне передане повідомлення в мережі йде через цільовий вузол обробки. При надходженні повідомлення до цільового вузла,

воно обробляється протягом часу σ_i з функцією розподілу ймовірностей $R_i(x)$ (через перевантаження центрального вузла і створення черги FIFO).

Сумарно первинні процеси утворюють багатовимірний (розподілений між вузлами мережі) випадковий процес W з кінцевим числом подій. Цей процес не визначається однозначно тільки множиною первинних процесів $\{w_i\}$. Він залежить також від алгоритму R маршрутизації. Використовується оптимальна маршрутизація, заснована на прогнозі часів очікування та обробки повідомлення від вузла-джерела до цільового вузла. Основними завданнями аналізу є вивчення таких характеристик розподіленого випадкового процесу W :

1) час α відповіді на повідомлення від моменту генерації вузлом-джерелом до моменту одержання його цільовим вузлом);

2) час β обробки потоку дублюючих повідомлень (від генерації першого повідомлення до отримання останнього повідомлення з однієї секції);

3) відношення числа повідомлень у процесі W до загальної можливості надходження подій в первинних процесах $\{w_i\}$.

Обчислювалися функції розподілу ймовірностей, при наведених умовах (7.1):

$$RM(x) = P\{\alpha \leq x\} \text{ та } RS(x) = P\{\beta \leq x\}, \quad (7.3)$$

для кількості вузлів у мережі $n = 1 \dots 20$.

Була поставлена задача маршрутизації повідомлень у системі складної структури, заснованої на ефективному виконанні всіх процесів, які протікають в цій системі. Стан системи визначається величезною кількістю повідомлень, які обробляються та очікують у черзі на обробку повідомлень. Потрібно визначити оптимальний час обробки повідомлень в кореновому вузлі, витративши при цьому мінімальний час (виокремлення дублюючих).

При представленні моделі в часі кожний вузол та повідомлення мають структуру, показану на рис. 7.22.

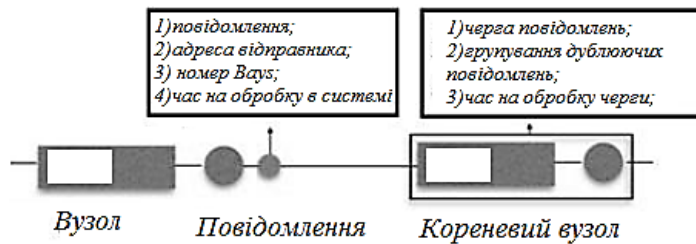


Рис. 7.22. Елементи моделі бездротової мережі судна-контейнеровоза

Аналіз ефективності маршрутизації проводився на подійно-орієнтованій моделі, яка задовольняє наступним умовам:

– зміна часу роботи моделі залежить від кількості повідомлень, які змінюють її стан та навантаженість;

– логіка настання подій визначає кількість згенерованих вузлів та надісланих повідомлень з однієї секції, які пов'язані з їх настанням.

Під час виконання випадково згенерованої системою моделі в конкретний момент часу t утворюється список вузлів (вершин) в кожній секції та список надісланих від них повідомлень, кожне з яких має свою ймовірність появи. Після чого обчислюється загальний час на обробку згенерованої моделі (рис. 7.23).

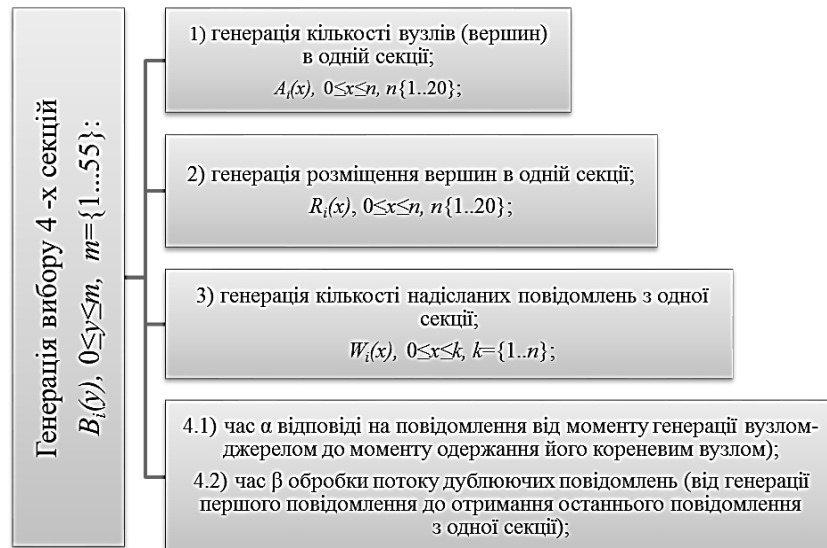


Рис. 7.23. Порядок генерації функції моделювання ймовірностей надходження повідомлень на кореневий вузол

Враховуючи, що в портах зазвичай не задіюють більше чотирьох кранів для завантаження/розвантаження одного судна, необхідно ввести функцію вибору ймовірного розташування кранів та вибору секції, з якою працюють портокрани.

При імітації систем великих масштабів, для оцінки стійкості даних необхідно коректне обчислення параметрів, при яких виконання таких систем буде надійним. Розглянемо ці параметри і залежність кінцевих даних від цих змінних:

- а) кількість вершин графа $|V|$;
- б) кількість згенерованих розміщень вершин графа. Для отримання коректних даних необхідні їх розгляд і обчислення за всіма можливими генераціями графа. Тобто, маючи $|V|$ кількість вершин, потрібно розглянути всі можливі варіанти генерацій цих графів, кожен раз генеруючи новий набір розміщень вузлів з множини V ;
- в) кількість надісланих повідомлень з кожного графа. Після генерації графа необхідно проходження певної кількості тестів для виявлення надійних даних (порядок повідомлень повинен йти в порядку зростання номера вершин);
- г) розмірність потоку (знаходження дублюючих). Варто враховувати, що при кожному проході відбувається повторне породження потоків повідомлень $w_i \in W$ однієї і тієї ж розмірності.

Таким чином, розглянуті питання, пов'язані із функціонуванням сучасних бездротових комп'ютерних мереж, які розгортаються в межах відокремленого технологічного об'єкта (судна-контейнеровоза). Побудована математична модель для проведення оцінки вартості потоків на графах сегментів бездротової комп'ютерної мережі при розвантаженні контейнеровозів та розраховано навантаження трафіку сегмента топології мережі для одного з бей-планів (вантажного плану на 4 секції).

Для організації кросплатформності ПЗ розробленої автоматизованої системи немає необхідності повністю переписувати код застосунку. Алгоритм визначення руху об'єктів, який використано, є універсальним. Проте, під час розробки програмних застосунків для інших мобільних операційних систем (Windows Mobile, iOS тощо) необхідно буде використовувати відповідні нативні бібліотеки для маніпулювання робочими режимами камер реєстраторів.

Для подальшого розвитку методу реєстрації великих об'єктів доцільно дослідити можливість датчика освітленості мобільних пристроїв у разі тривалого перекриття такого датчика контейнером, що рухається.

Такий комплексний підхід надасть можливість швидшого масового поширення продукту.

7.3 Розробка методу обліку рухомих вантажів у пунктах пропуску із запобіганням подвійної реєстрації

Нині у світі є потреба визначити кількість великих вантажів під час їх руху/завантаження/розвантаження на тимчасових пунктах пропуску, для розрахунку вартості перевезення через міста та мостові споруди, тощо.

У світі використовуються декілька систем на базі GPS-трекерів та відеокамер з сервером розпізнавання образів вантажу номера, та внесення інформації за номером контейнера в базу даних. Найбільш відомими з таких систем для вантажообліку є Tetis (Трігон), Helios, Контейнер – КОНТРОЛЬ та ін. [208; 286], які або зчитують інформацію з бортового контролера на вантажі, або є стаціонарно встановленими у транспортних вузлах. Такі системи є надмірно складними технологічно і надають інформацію лише власникові вантажу або компанії-перевізнику. Але іноді зазначена інформація необхідна саме на місці проходження вантажу. Крім того, всі зазначені системи використовують для реєстрації GPS-координати вантажів і тому втрачають працездатність при відсутності зв'язку з супутниками.

Для обробки зображень рухомих об'єктів використовуються різноманітні методи, що відмінні за параметрами і вихідними результатами. Це лінійні та нелінійні фільтри, фільтр Гауса, медіанні, ранжируючі, адаптовані, гібридні та ін., здатні коригувати розмитість зображення, зумовлену саме рухом досліджуваного об'єкта [223].

Виявлення на кадрі певного об'єкта можна поділити на наступні етапи:

- знаходження градієнта зображення, тобто контурів зображення, нп., за допомогою алгоритму Собеля [186];
- побудова гістограми зображення;
- бінаризація;
- знаходження пікселів, які входять в певний діапазон значень;
- знаходження координат об'єкта.

Правильно побудований алгоритм дасть можливість опустити деякі етапи, такі, як бінаризація, а побудову гістограми і знаходження контурів об'єднати.

Збільшити швидкість прийняття рішення щодо наявності великорозмірного вантажу в кадрі та забезпечити високоякісну фільтрацію без додаткових спотворень дозволяють застосування подвійного медіанного фільтра [124].

Зважаючи на те, що рухатись може не тільки сам вантаж, а й засіб його реєстрації (нп., дрон з відеокамерою), треба докласти певних зусиль для запобігання реєстрації замість великого вантажу – стороннього об'єкта, що попадає у кадр.

Задача виявлення рухомих великих об'єктів на відео-потоці при практичному підході розбивається на кілька менших підзадач: виявлення на кадрі потрібного об'єкта (автомашини, контейнера, тощо), співставлення пари кадрів і порівняння положень об'єкта на них.

Розрахуємо градієнт зображення. Градієнт у загальному випадку – векторна величина, яка показує напрям максимального збільшення деякої величини. У нашому випадку «деяка величина» – двомірна функція яскравості зображення I . Яскравість зображення в пікселі P_{ij} це значення пікселя в градаціях сірого (Grayscale). Розрахунок значення в градаціях сірого можна провести за допомогою формули (7.4):

$$I_{ij} = 0,2125r + 0,7154g + 0,00721b, \quad (7.4)$$

де r – значення червоного каналу (red channel);
 g – значення зеленого каналу (green channel);
 b – значення синього каналу (blue channel).

Вектор градієнта зображення для функції яскравості I матиме наступний вигляд:

$$\vec{g} = grad(I) = (\partial I / \partial x; \partial I / \partial y),$$

Для знаходження градієнта зображення були використані фільтри Собеля. Фільтри Собеля – двомірні вектори-маски розмірністю 3×3 , якими початкове зображення обробляється в наступному порядку:

$$\begin{aligned} \partial x &= (p7 + 2p8 + p9) - (p1 + 2p2 + p3); \\ \partial y &= (p3 + 2p6 + p9) - (p1 + 2p4 + p7), \end{aligned}$$

Таким чином, маємо значення похідних за двома напрямками у зображенні: по осі x та осі y . Безпосередньо значення градієнта в точці можна розрахувати за допомогою наступної формули:

$$G = \sqrt{dx^2 + dy^2}.$$

Розглянемо побудовання гістограми – графіку частотного розподілення елементів цифрового зображення з різною яскравістю, в якому по горизонталі представлена яскравість (значення функції I , яка була розглянута раніше), а по вертикалі – значення пікселів з даним значенням яскравості (рис. 7.24).

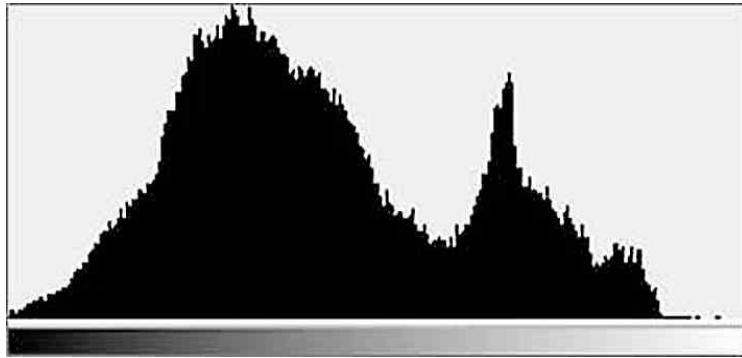


Рис. 7.24. Типова гістограма яскравості

Алгоритм розрахування гістограми:

1. Подаємо зображення A ($W \times H$) в градаціях сірого, використавши для кожного пікселя формулу (7.4);
2. Обчислимо значення A_{min} та A_{max} ;
3. Проаналізуємо зображення по осям x та y для гістограми розміром N , де $N = A_{min} - A_{max} + 1$, та для кожного елемента зі значенням A_{ij} ($i = 1 \dots W, j = 1 \dots H$) додаємо «1».

Процес подання гістограми в розробленому алгоритмі описаний нижче.

Процес бінаризації – це процес перетворення зображення з кольорового чи в градаціях сірого в чорно-біле (бінарне). Як правило, коли йдеться про бінаризацію, мають на увазі порогову бінаризацію. Принцип функціонування порогової бінаризації наступний: зображення подається в градаціях сірого, після чого кожне значення пікселя порівнюється з пороговим значенням (воно може бути як константою, так і значенням, вирахованим за допомогою певного методу, наприклад, методу Отсу). Якщо значення в градаціях сірого пікселя менше порогового значення, на місце пікселя записується «1»

($r: 255, g: 255, b: 255$). Інакше – «0» ($r: 0, g: 0, b: 0$). Після чого маємо зображення, яке складається лише з білого та чорного кольорів. У розробленому алгоритмі цей етап не розглядається.

Потім знаходимо пікселі, що входять в певний діапазон. Після знаходження градієнта зображення отримаємо повноцінну картину контурів всіх об'єктів на зображенні. Для пошуку контурів потрібного об'єкта використовуємо градієнт зображення, найбільші значення котрого знаходяться на межах об'єкта, який найбільше контрастує з оточенням. У нашому випадку великорозмірний вантаж є саме таким об'єктом.

Для того, щоб віднайти на мапі градієнта контури потрібного об'єкта, знайдемо на гістограмі пікселі, які входять у діапазон, рівний у розглянутому випадку $\delta = 3\%$. Значення у 3% не є константою, а визначається у кожному пункті обліку залежно від розміщення мобільного реєстратора щодо відношення до рухомого вантажу. Значення гіперпараметра δ добирається таким чином, щоб на факт реєстрації не впливали сторонні шуми, нп., великорозмірні об'єкти поза межами зони огляду при завищених значеннях δ (рис. 7.25а). При занижених значеннях δ , навпаки, може бути незархований сам великорозмірний об'єкт (рис. 7.25б).

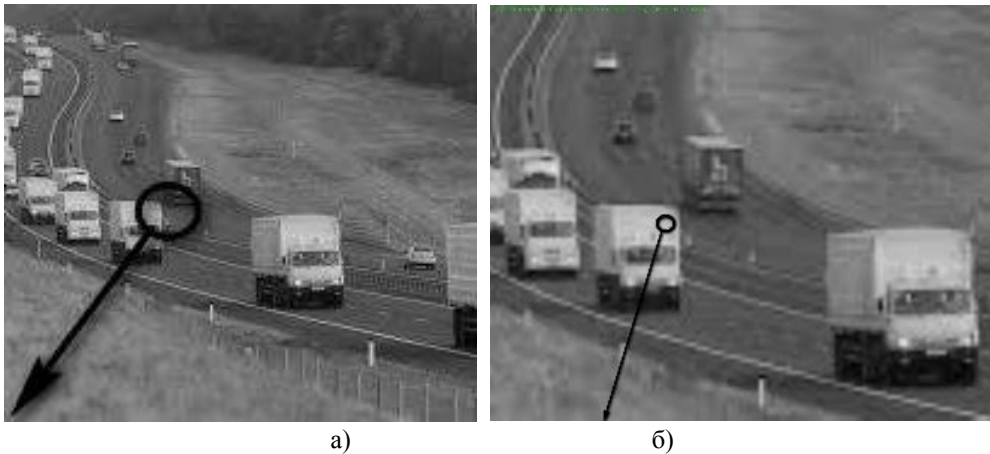


Рис. 7.25. Невдалий вибір значення δ : зavelike (а) та замале (б)

Таким чином, всі пікселі, концентрація яких в гістограмі менша за δ від абсолютної кількості пікселів (тобто, від розміру гістограми), вважатимемо пікселями, що знаходяться на контурі об'єкта, який має бути зареєстрованим.

Координати об'єкта знаходять наступним чином: зміні підлягають чотири значення – координати верхнього лівого кута прямокутника, в який вписаний об'єкт, та координати нижнього правого кута. Ітеративно знаходяться потрібні пікселі, а їхні координати записуються в змінні. Після проходження зображенням матимемо координати прямокутника, в межах якого знаходиться об'єкт.

Реалізація розробленого алгоритму складається з трьох кроків: завантаження зображення в градаціях сірого (використовується бібліотека OpenCV); складання гістограми зображення (з відмінністю в тому, що на вертикальній осі розташовуються значення градієнта в точках, а не яскравості); знаходження координат пікселів, значення градієнта яких знаходиться в межах дібраного значення δ . Блок-схема алгоритму реєстрації великорозмірного вантажу наведена на рис. 7.26.

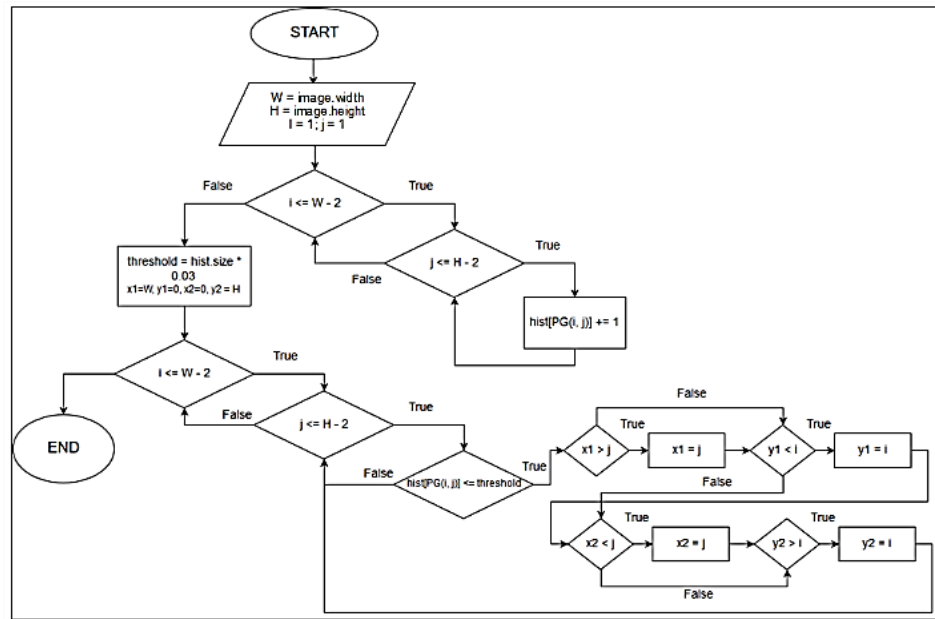


Рис. 7.26. Блок-схема алгоритму запобігання подвійної реєстрації

Спочатку підключаються стандартні бібліотеки та основні файли бібліотеки OpenCV.

Після отримання зображення розміром $W \times H$ допоміжна функція повертає значення градієнта пікселя PG_{ij} методом Собеля.

Далі головна функція, яка реалізує алгоритм, описаний вище, повертає тип *std::tuple*, елементами якого є координати прямокутника, в який вписано об'єкт. Якщо рухомий об'єкт не відноситься до великорозмірних та його зображення-прямокутник не задовольняє умовам алгоритму (його площа займає менше 15% площі зображення), повертається кортеж з чотирьох значень «мінус 1».

Потім відбувається формування гістограм. Вона являє собою об'єкт *std::map* (асоціативний масив), в якому ключем є градієнт, а значенням – кількість пікселів з таким градієнтом.

Спочатку значення кількості пікселів з даним значенням градієнта в гістограмі дорівнює «0». Ітеративно, $i = 1 \dots W - 2$, $j = 1 \dots H - 2$, $i \in 1 \dots W - 2$, $j \in 1 \dots H - 2$, елементи горизонтальної осі гістограм ініціюються, і на виході буде отримана гістограма, яка представлена асоціативним масивом.

У циклі для кожного пікселя зображення A розраховується значення градієнта, яке слугує ключем для гістограм – для отримання «насиченості» пікселів і порівняння його з порогом, який був зазначений раніше. Піксель, насиченість якого менша за величину δN (де N – кількість унікальних значень градієнта), вважаємо за піксель, що знаходиться на контурі об'єкта. У такому випадку значення координат прямокутника об'єкта перераховується.

На останньому кроці здійснюється перевірка отриманих даних і повернення результату. Перевірка відбувається за наступним принципом: якщо відношення площі прямокутника, в який вписано об'єкт, до площі зображення менше за 0,15 (15%), то вважається, що об'єкта на зображенні немає (або об'єкт, що має такі розміри, що не дозволяють його класифікувати як великорозмірний вантаж – це можуть бути, наприклад, літаки, які пролітають над реєстратором).

Після порівняння координат об'єкта на двох сусідніх кадрах приймається рішення, чи є об'єкт рухомим для запобігання подвійної реєстрації нерухомих об'єктів.

Проведено тестування програми на запобігання реєстрації об'єктів, що не рухаються (будинки, авто на стоянці тощо). Для перевірки обрані два зображення розміром 240 x 200 (рис. 7.27).

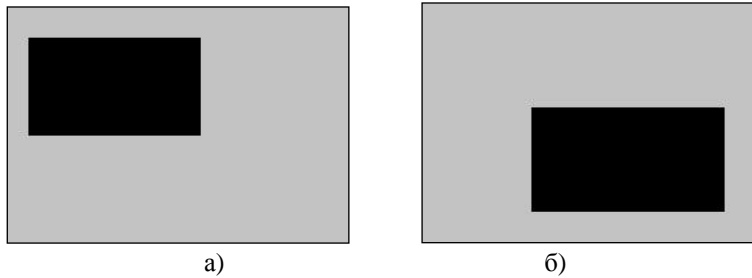


Рис. 7.27. Положення вантажу: *frame1* (а) та *frame2* (б)

При тестуванні АС на коректність при зміні положення вантажу, коли два кадри є різними з точки зору руху об'єкта (*frame1* та *frame2*), системою приймається рішення «*Object moves*».

Якщо замінити *frame1* та *frame2* одним зображенням *frame1*, АС прийме рішення про нерухомість або відсутність об'єкта: «*Object doesn't move*». Таким чином запобігається подвійна реєстрація об'єктів, що не рухаються.

Також проводилось тестування швидкодії роботи програми для вантажів різного розміру, що надають на світлочутливій матриці реєстратора зображення різної розмірності у пікселях (табл. 7.2).

Таблиця 7.2. Тестування швидкодії клієнтської частини

Розмірність $W \times H$, пікс.	Швидкість виконання, мс
240 x 200	53
400 x 400	156
560 x 560	710
800 x 800	2099
1000 x 1000	5202
1250 x 1250	7953

Запропонований метод обліку великорозмірних вантажів із запобіганням подвійної реєстрації об'єктів розрахований на роботу в локальній мережі, що суттєво знижує вартість системи у порівнянні з існуючими аналогами, які потребують наявності зв'язку з супутниками, або з глобальною мережею. Також відсутні додаткові витрати на попереднє електронне маркування вантажів.

Віддалений доступ до мережі необхідний в описаній АС не під час поточної роботи, а тільки при потребі складання узагальнених (за день, за місяць та ін.) звітів шляхом синхронізації з серверами обробки даних. Зважаючи на обмежені обчислювальні можливості застосованих реєстраторів, у архітектуру АС імплементовані технології Gearman та Varnish з репліка-сетями, що дозволяє зробити шардування й винести функціонал обробки даних на будь-яку необхідну кількість фізичних серверів, а сховище даних – у віртуальне середовище хмарних технологій.

Запропонований підхід суттєво зменшує навантаження на систему.

Недоліками представленої системи можна вважати відсутність кросплатформного рішення на боці первинних реєстраторів.

Висновки до розділу 7

Представлені прикладні рішення з використанням гетерогенних рухомих моніторингових мереж потребують використання сучасних підходів до організації розподілених систем. Розроблені програмні рішення надають програмний інтерфейс до ключових елементів системи: створення задач, отримання інформації про пристрої, модифікація параметрів системи, конфігурація, та інше. Такі вимоги реалізовані за допомогою технології REST та веб-сервера, що розташований на модулі системи керування.

Водночас, необхідно підтримувати зв'язок з кожним елементом мережі. Цю задачу вирішено за допомогою програмного інтерфейсу SOCKET та протоколу TCP/IP. Усі дані, якими обмінюються система керування та безпілотний апарат, передаються спеціальним протоколом у форматі JSON.

У цьому розділі було розглянуто функції та інтерфейси розробленого програмного модуля для відслідковування маршрутів БПА РММ на мапі місцевості.

Для відображення отриманих тривимірних об'єктів було розглянуто три фреймворки для роботи з 3D-графікою, виділено технологію SceneKit для роботи з тривимірною графікою. Під час розробки мобільного застосунку було адаптовано для моніторингу абонентів РММ та реалізовані ітеративний алгоритм з динамічним кешуванням пошуку та ітеративний алгоритм побудови триангуляції Делоне з обмеженнями.

Крім того було досліджено Google Static Maps API для отримання достатньо точних мап висот та статичних зображень картографічної поверхні для подальшого використання в якості 2D-текстури.

Представлено опис основних діалогів застосунку з детальним оглядом всіх можливостей програми. Продемонстровано приклад побудованої тривимірної картографічної поверхні з відповідними маршрутами дронів у тривимірному просторі.

Представлений UI є інтуїтивно зрозумілим, простим, не вимагає багато часу для опанування.

Розроблена інформаційна система обліку великорозмірних вантажів дозволяє отримувати у режимі реального часу інформацію щодо кількості вантажів, які перетнули зазначений пункт.

Розроблений алгоритм реєстрації великорозмірного вантажу за допомогою знаходження градієнта зображення цільового об'єкта в градаціях сірого дозволяє використовувати в АС реєстратори з обмеженими обчислювальними можливостями, забезпечує запобігання подвійної реєстрації вантажу та реєстрації руху сторонніх об'єктів.

На судах-контейнеровозах впровадження системи автоматизації обліку розвантаження дозволить капітану в режимі реального часу отримувати інформацію, скільки контейнерів розвантажено з кожного ролу. Це, в свою чергу, дозволить капітану зберегти метацентричну висоту в допустимому діапазоні, тобто запобігти утворенню крену і диференту судна, зменшити паперовий документообіг, підвищити продуктивність праці, скоротити час на обробку інформації.

Розроблено рішення щодо реєстрації великорозмірних вантажів працездатні для гетерогенної мережі з великою кількістю реєстраторів при обмеженому доступі до Інтернету, необхідному тільки під час синхронізації даних з декількох пунктів обліку задля формування необхідної статистики в реальному часі.

Як наслідок, утворюються додаткові тимчасові ресурси для розробки і реалізації нових економічних та інвестиційних проектів. Таким чином вирішується проблема відшкодування витрат і підвищення рентабельності впровадження моніторингової мережі на базі мобільних пристроїв, що розміщаються за місцем моніторингу за допомогою БПА.

Додаток А

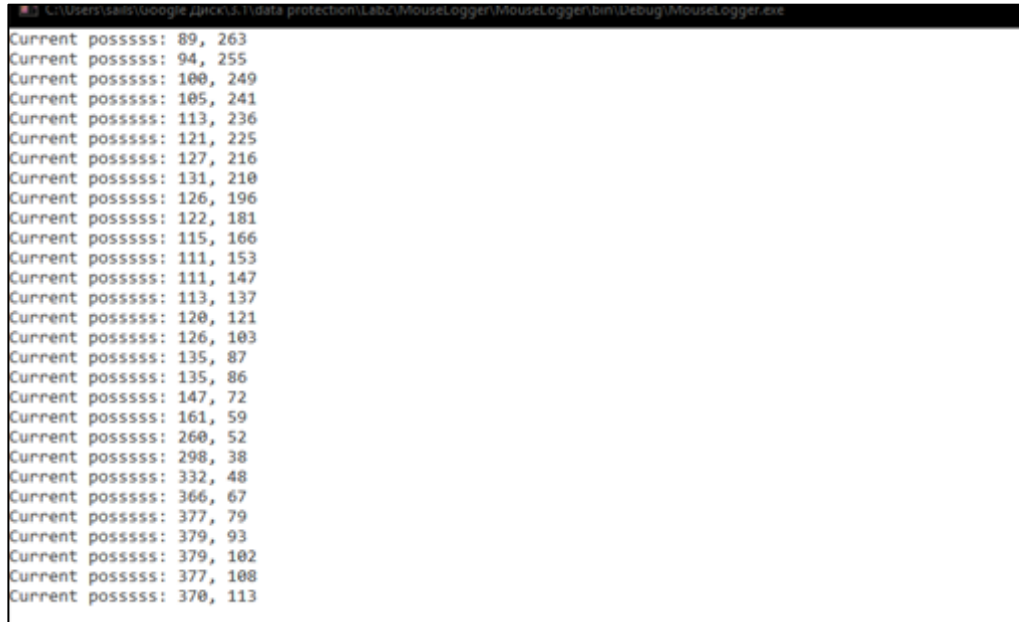
Застосунок для визначення координат траєкторії руху БПЛА (на С#)

```
private static IntPtr HookCallback(int nCode, IntPtr wParam, IntPtr lParam)
{
    int xPos = 0;
    int yPos = 0;
    if (nCode >= 0 && MouseMessages.WM_LBUTTONDOWN ==
(MouseMessages)wParam)
    {
        xPos = Cursor.Position.X;
        yPos = Cursor.Position.Y;
        MSHLLHOOKSTRUCT hookStruct =
(MSHLLHOOKSTRUCT)Marshal.PtrToStructure(lParam, typeof(MSHLLHOOKSTRUCT));
        using (StreamWriter cX = new StreamWriter("x.txt", true,
System.Text.Encoding.UTF8))
        {
            cX.WriteLine(" " + hookStruct.pt.x);
        }
        using (StreamWriter cY = new StreamWriter("y.txt", true,
System.Text.Encoding.UTF8))
        {
            cY.WriteLine(" " + hookStruct.pt.y);
        }
        Console.WriteLine("Current posssss: " + hookStruct.pt.x + ", " +
hookStruct.pt.y);
    }
    return CallNextHookEx(_hookID, nCode, wParam, lParam);
}
```

Додаток Б

Застосунок для моделювання руху БПЛА за визначеними координатами заданої траєкторії (MATLAB)

Файл з координатами траєкторії руху БПЛА наведений на рис. Б.1.



```
Current posssss: 89, 263
Current posssss: 94, 255
Current posssss: 100, 249
Current posssss: 105, 241
Current posssss: 113, 236
Current posssss: 121, 225
Current posssss: 127, 216
Current posssss: 131, 210
Current posssss: 126, 196
Current posssss: 122, 181
Current posssss: 115, 166
Current posssss: 111, 153
Current posssss: 111, 147
Current posssss: 113, 137
Current posssss: 120, 121
Current posssss: 126, 103
Current posssss: 135, 87
Current posssss: 135, 86
Current posssss: 147, 72
Current posssss: 161, 59
Current posssss: 260, 52
Current posssss: 298, 38
Current posssss: 332, 48
Current posssss: 366, 67
Current posssss: 377, 79
Current posssss: 379, 93
Current posssss: 379, 102
Current posssss: 377, 108
Current posssss: 370, 113
```

Рис. Б.1. Приклад роботи програми зчитування точок маршруту з рисунку (мапи)

Програмний код застосунку для моделювання у середовищі MATLAB групової поведінки 5 дронів (за визначеними координатами):

```
clear all;

scale=1;

drone1X=scale*[23 38 54 68 82 92 104 114 ...
drone1Y=scale*[626 617 604 592 580 565 ...
drone1RespX = scale*0;
drone1RespY = scale*0;
drone1RespZ = scale*0;
drone1StartCharge = 60;

drone2X=scale*[ 525 517 513 506 500 491 480 ...
drone2Y=scale*[ 506 495 482 468 453 436 421 ...
drone2RespX = scale*0;
drone2RespY = scale*100;
drone2RespZ = scale*0;
```

```
drone2StartCharge = 45;

drone3X=scale*[ 514 510 500 490 485 481 470 ...
drone3Y=scale*[ 486 490 495 499 504 504 499 ...
drone3RespX = scale*0;
drone3RespY = scale*200;
drone3RespZ = scale*0;
drone3StartCharge = 35;

drone4X=scale*[ 177 184 191 198 205 204 207...
drone4Y=scale*[ 467 469 472 479 489 481 472 464 ...
drone4RespX = scale*100;
drone4RespY = scale*0;
drone4RespZ =scale* 0;
drone4StartCharge = 25;

drone5X=scale*[ 485 477 473 465 459 450 439 434...
drone5Y=scale*[ 503 508 514 520 520 511 499 477 ...
drone5RespX = scale*200;
drone5RespY = scale*0;
drone5RespZ = scale*0;
drone5StartCharge = 50;

lineStyleToResp='-b';
lineStyleUsual='-b';

drone1Color='y';
drone2Color='r';
drone3Color='g';
drone4Color='k';
drone5Color='m';
pointStyle1=strcat('*', drone1Color);
pointStyle2=strcat('*', drone2Color);
pointStyle3=strcat('*', drone3Color);
pointStyle4=strcat('*', drone4Color);
pointStyle5=strcat('*', drone5Color);

pointStyleWhite='ok';
sleepTime=0;
j = 0;

normalZ = 500;
startCharge = 100;
xResp1 = 0;
yResp1 = 0;
zResp1 = 0;

[toDrawX1,toDrawY1,toDrawZ1] =
calculateCoords(drone1X,drone1Y,drone1RespX,drone1RespY,drone1RespZ,drone1StartCharge,normalZ,scale);
[toDrawX2,toDrawY2,toDrawZ2] =
calculateCoords(drone2X,drone2Y,drone2RespX,drone2RespY,drone2RespZ,drone2StartCharge,normalZ,scale);
```

```

[toDrawX3,toDrawY3,toDrawZ3] =
calculateCoords(drone3X,drone3Y,drone3RespX,drone3RespY,drone3RespZ,drone3StartCharge,normalZ,scale);
[toDrawX4,toDrawY4,toDrawZ4] =
calculateCoords(drone4X,drone4Y,drone4RespX,drone4RespY,drone4RespZ,drone4StartCharge,normalZ,scale);
[toDrawX5,toDrawY5,toDrawZ5] =
calculateCoords(drone5X,drone5Y,drone5RespX,drone5RespY,drone5RespZ,drone5StartCharge,normalZ,scale);
maxLength =
getMax(length(toDrawX1),length(toDrawX2),length(toDrawX3),length(toDrawX4),length(toDrawX5));

for i=1:maxLength-1
    plot3(0, 800, 1000, 'b');
    hPlotData = plot3(0, 800, 0, 'b');
    %отрисовка респануов
    plot3(drone1RespX, drone1RespY, drone1RespZ, strcat('o', drone1Color));
    plot3(drone2RespX, drone2RespY, drone2RespZ, strcat('o', drone2Color));
    plot3(drone3RespX, drone3RespY, drone3RespZ, strcat('o', drone3Color));
    plot3(drone4RespX, drone4RespY, drone4RespZ, strcat('o', drone4Color));
    plot3(drone5RespX, drone5RespY, drone5RespZ, strcat('o', drone5Color));
    %отрисовка начальных точек
    plot3(toDrawX1(1), toDrawY1(1), toDrawZ1(1), strcat('x', drone1Color));
    plot3(toDrawX2(1), toDrawY2(1), toDrawZ2(1), strcat('x', drone2Color));
    plot3(toDrawX3(1), toDrawY3(1), toDrawZ3(1), strcat('x', drone3Color));
    plot3(toDrawX4(1), toDrawY4(1), toDrawZ4(1), strcat('x', drone4Color));
    plot3(toDrawX5(1), toDrawY5(1), toDrawZ5(1), strcat('x', drone5Color));
    %отрисовка траекторий
    if(i<=length(toDrawX1))
        if(toDrawZ1(i) < normalZ)
            plot3(toDrawX1(1:i), toDrawY1(1:i), toDrawZ1(1:i),
lineStyleToResp);
        else
            plot3(toDrawX1(1:i), toDrawY1(1:i), toDrawZ1(1:i),
lineStyleUsual);
        end
        hPlotData1 = plot3(toDrawX1(i), toDrawY1(i), toDrawZ1(i),
pointStyle1);
    else
        plot3(drone1RespX, drone1RespY, drone1RespZ, pointStyle1);
    end

    if(i<=length(toDrawX2))
        if(toDrawZ2(i) < normalZ)
            plot3(toDrawX2(1:i), toDrawY2(1:i), toDrawZ2(1:i),
lineStyleToResp);
        else
            plot3(toDrawX2(1:i), toDrawY2(1:i), toDrawZ2(1:i),
lineStyleUsual);
        end
        hPlotData2 = plot3(toDrawX2(i), toDrawY2(i), toDrawZ2(i),
pointStyle2);
    else
        plot3(drone2RespX, drone2RespY, drone2RespZ, pointStyle2);
    end
end

```

```
        if(i<=length(toDrawX3))
            if(toDrawZ3(i) < normalZ)
                plot3(toDrawX3(1:i), toDrawY3(1:i), toDrawZ3(1:i),
lineStyleToResp);
            else
                plot3(toDrawX3(1:i), toDrawY3(1:i), toDrawZ3(1:i),
lineStyleUsual);
            end
            hPlotData3 = plot3(toDrawX3(i), toDrawY3(i), toDrawZ3(i),
pointStyle3);
        else
            plot3(drone3RespX, drone3RespY, drone3RespZ, pointStyle3);
        end

        if(i<=length(toDrawX4))
            if(toDrawZ4(i) < normalZ)
                plot3(toDrawX4(1:i), toDrawY4(1:i), toDrawZ4(1:i),
lineStyleToResp);
            else
                plot3(toDrawX4(1:i), toDrawY4(1:i), toDrawZ4(1:i),
lineStyleUsual);
            end
            hPlotData4 = plot3(toDrawX4(i), toDrawY4(i), toDrawZ4(i),
pointStyle4);
        else
            plot3(drone4RespX, drone4RespY, drone4RespZ, pointStyle4);
        end

        if(i<=length(toDrawX5))
            if(toDrawZ5(i) < normalZ)
                plot3(toDrawX5(1:i), toDrawY5(1:i), toDrawZ5(1:i),
lineStyleToResp);
            else
                plot3(toDrawX5(1:i), toDrawY5(1:i), toDrawZ5(1:i),
lineStyleUsual);
            end
            hPlotData5 = plot3(toDrawX5(i), toDrawY5(i), toDrawZ5(i),
pointStyle5);
        end

        hold on;
        grid;
        F(i) = getframe;
        pause(sleepTime);
        delete(hPlotData1);
        delete(hPlotData2);
        delete(hPlotData3);
        delete(hPlotData4);
        delete(hPlotData5);
    end
    grid;

% функция вычисления координат
function [drawnX,drawnY,drawnZ] =
calculateCoords(coordsX,coordsY,xResp,yResp,zResp,startCharge,normalZ,scale)
```

```

charge = startCharge;
drawnX = [];
drawnY = [];
drawnZ = [];
for i = 3:length(coordsX)-1
    x1=coordsX(i);
    y1=coordsY(i);
    x2=coordsX(i+1);
    y2=coordsY(i+1);
    z1=normalZ;
    z2=normalZ;

    if(charge <= 0)
        charge = startCharge;
        [corX, corY, corZ] =
goToRespawnАндBack(x1,y1,normalZ,xResp,yResp,zResp);
        for j=1:length(corX)
            drawnX(length(drawnX)+1) = corX(j);
            drawnY(length(drawnY)+1) = corY(j);
            drawnZ(length(drawnZ)+1) = corZ(j);
        end
    end

    [corX, corY, corZ] = getCoordsBetween(x1,y1,x2,y1,normalZ,scale);
    charge = charge - length(corX);
    for j=1:length(corX)
        drawnX(length(drawnX)+1) = corX(j);
        drawnY(length(drawnY)+1) = corY(j);
        drawnZ(length(drawnZ)+1) = corZ(j);
    end
end
[corX, corY, corZ] =
goToRespawn(coordsX(length(coordsX)),coordsY(length(coordsX)),normalZ,xResp,y
Resp,zResp);
for j=1:length(corX)
    drawnX(length(drawnX)+1) = corX(j);
    drawnY(length(drawnY)+1) = corY(j);
    drawnZ(length(drawnZ)+1) = corZ(j);
end
return
%функция для вычисления координат между респауном и точкой
function [coordsX,coordsY,coordsZ] = goToRespawnАндBack(x1,y1,z1,x2,y2,z2)
    len = 0;
    len0=20;
    Si = sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1)+(z2-z1)*(z2-z1));
    coordsX=[x1 x1];
    coordsY=[y1 y1];
    coordsZ=[z1 z1];
    while len < Si
        xi = x1+len*(x2-x1)/Si;
        yi = y1+len*(y2-y1)/Si;
        zi = z1+len*(z2-z1)/Si;
        len = len+len0;

        coordsX(length(coordsX)+1) = xi;
        coordsY(length(coordsY)+1) = yi;

```



```
        coordsZ(length(coordsZ)+1) = zi;
    end
    len=0;
    while len < Si
        xi = x2+len*(x1-x2)/Si;
        yi = y2+len*(y1-y2)/Si;
        zi = z2+len*(z1-z2)/Si;
        len = len+len0;
        coordsX(length(coordsX)+1) = xi;
        coordsY(length(coordsY)+1) = yi;
        coordsZ(length(coordsZ)+1) = zi;
    end
    return
```

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. 802.11ac : The Fifth Generation of Wi-Fi. *Technical White Paper. Cisco Systems Incorp.* San Jose, CA, USA : 2014. URL : https://www.cisco.com/c/en/us/products/collateral/wireless/aironet-3600-series/white_paper_c11-713103.pdf.
2. Ab Aziz N. A., Ibrahim Z. Asynchronous particle swarm optimization for swarm robotics. *Procedia Engineering*. 2012. No. 41. P. 951–957. DOI : 10.1016/j.proeng.2012.07.268.
3. Ademoye T. A., Davari A., Cao W. Three-dimensional obstacle avoidance maneuver planning using mixed integer linear programming. *Robotics and Applications (RA '2006)* : Proceedings of the International Conference / Ed. M. H. Hamza, Honolulu, 14–16 Aug., 2006. ACTA Press, 2006. 210 p.
4. Advanced UAVs & Flight Control Systems. URL : <http://sky-drones.com>.
5. Advances in Unmanned Aerial Vehicles : State of the art and the road to autonomy / Ed. K. P. Valavanis. Springer Science & Business Media, 2008. 544 p.
6. Aerial Construction. *Research D'Andrea* / Institute for Dynamic Systems and Control. URL : <http://www.idsc.ethz.ch/research-dandrea/research-projects/aerial-construction.html>.
7. Alami R., Fleury S., Herrb M., et al. Multi Robot Cooperation in the Martha Project : Report / Laboratory for Analysis and Architecture of Systems (LAAS-CNRS), Toulouse, France, 1998. 21 p. URL : <https://homepages.laas.fr/felix/publis-pdf/ieee-ram98.pdf>.
8. Amin I. N., Boskovic J. D., Mehra R. K. A fast and efficient approach to path planning for unmanned vehicles. *AIAA Guidance, Navigation, and Control Conf. and Exhibit* : Proceedings of the International Conference, Colorado, 21–24 Aug., 2006.
9. Antoshchuk S. G., Maksymov O. M., Wendl M. Autopilot model for returning an unmanned aerial vehicle to its starting point in case of electromagnetic noise. *Proceedings of Odessa Polytechnic University*. 2017. Is. 3 (53). P. 94–101. URL : <http://www.pratsi.opu.ua/app/webroot/articles/1517479093.pdf>.
10. APM 2.6 APM flight controller board for multicopter ARDUPILOT MEGA 2.6 Version. URL : http://www.nyplatform.com/index.php?route=product/product&product_id=1365.
11. Apptest maps 3d watch and download. URL : <http://hdvidzpro.mobi/list/apptest-maps-3d/page=CBQQAQ>.
12. Augugliaro F., Zarfati E., Mirjan A., D'Andrea R. Knot-tying with flying machines for aerial construction. *Intelligent Robots and Systems (IROS)* : Proceedings of IEEE/RSJ International Conference, Hamburg, Germany, 28 Sept. – 2 Oct., 2015. P. 5917–5922. DOI : 10.1109/IROS.2015.7354218.
13. Auke J. I., Alcherio M., Aude B., et al. Collaboration through the exploitation of local interactions in autonomous collective robotics. The stick pulling experiment. *Auton Robot.* 2001. No. 11 (2). P. 149–171.
14. Austin R. Unmanned aircraft systems: UAVS design, development and deployment. Chichester, West Sussex, UK : Wiley, 2010. 365 p.
15. Autopilot for DJI : Overview. URL : <https://developer.dji.com/showcase/c394119cb61f9d8b2d49f2b6af92e4c8/>.
16. AVR Studio 5.0. URL : <http://microsin.net/programming/avr/avrstudio-all-versions.html>.
17. Bahgec I. E., Sahin E. Evolving aggregation behaviors for swarm robotic systems: a systematic case study. *Proceedings of IEEE Swarm intelligence symposium*. 2005. P. 333–340. DOI : 10.1.1.187.7579.
18. Balch T. Communication, diversity and learning: cornerstones of swarm behavior. *Swarm Robotics : Lecture Notes in Computer Science*. 2005. Vol. 3342. P. 21–30.

19. Bandurian A. Wi-Fi, the unobvious and overlooked: Power. *Technical Blog*. Publ. 2014, Feb. 16. URL : <https://arsenb.wordpress.com/2014/02/16/wi-fi-the-unobviousand-overlooked-details-part-1/>.
20. Bayindir L., Sahin E. A review of studies in swarm robotics. *Turkish journal of electrical engineering & Computer sciences*. 2007. Vol. 15, No. 2. P. 115–147.
21. BCM4330 – Single Chip IEEE 802.11™ a/b/g/n MAC/Baseband/Radio with Integrated Bluetooth® 4.0 + HS and FM Transceiver : Preliminary Data Sheet. URL : <http://linux-sunxi.org/images/0/05/4330-DS206-R.pdf>.
22. Bekey G. A. Autonomous robots : from biological inspiration to implementation and control. MIT Press, 2005. 577 p.
23. Bello-Ochende T., Liebenberg L., Meyer J. P. Constructal cooling channels for micro-channel heat sinks. *International Journal of Heat and Mass Transfer*. 2007. Vol. 50, Is. 21–22. P. 4141–4150. DOI : 10.1016/j.ijheatmasstransfer.2007.02.019.
24. Benchoff B. Hijacking drones with a MAVLink exploit. Publ. Oct. 15, 2015. URL : <https://diydrones.com/profiles/blogs/hijacking-quadcopters-with-a-mavlink-exploit>
25. Bhojani F. For good or bad, intelligent, swarming nanobots are the next frontier of drones. Publ. May 21, 2014. URL : https://motherboard.vice.com/en_us/article/qkvkqd/why-the-us-military-is-funding-tiny-autonomous-flying-robots.
26. Bortoff S. A. Path planning for UAVs. *American Control Conference* : Proceedings of the conference, Chicago, Illinois, June 2000. Vol. 1(6). P. 364–368.
27. Braccini C., Cocurullo F., Lavagetto F. A fast algorithm for high quality vector quantization codebook design, *Image Analysis and Processing (ICIAP '95)* : Proceedings of the 8th International Conference, London, UK, 1995. P. 643–648.
28. Burlachenko I., Zhuravska I., Musiyenko M. Devising a method for the active coordination of video cameras in optical navigation based on multi-agent approach. *Eastern-European Journal of Enterprise Technologies*. 2017. Vol. 1, No. 9 (85). P. 17–25. DOI : 10.15587/1729-4061.2017.90863.
29. Chakos B. Here's how. *PCWorld*. 2013. P. 89.
30. Chen F. China shows off drone brigade at Guangzhou Fortune Forum gala. *Asia Times*. Publ. Dec. 8, 2017. URL : <http://www.atimes.com/article/china-shows-off-drone-brigade-guangzhou-fortune-forum-gala/>.
31. Cheung W. UAS data processing : software and applications : Presentation / Palomar college. Publ. Sept. 20, 2017. URL : https://static1.squarespace.com/static/57d1797d2994cac3020bbc3a/t/59d123394c326dd2b357bd08/1506878268269/IEGIS_UAS+Data+Processing+and+Applications.pdf.
32. Child J. FPGA Boards and systems boost UAV payload compute density. *The Journal of Military Electronics & Computing*. 2009, Feb. URL : <http://www.cotsjournalonline.com/articles/view/100910>.
33. Colomina I., Molina P. Unmanned aerial systems for photo-grammetry and remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*. 2014. Vol. 92. P. 79–97. DOI : 10.1015/j.isprsjprs.2014.02.013.
34. Communications Network ICs > Wireless LAN ICs > Realtek RTL8188EUS... RTL8192CU / Realtek Semiconductor Corp. URL : <http://www.realtek.com.tw/products/productsView.aspx?Conn=4&Langid=1&Level=5&PFid=48&ProdID=277>.
35. Composer : Documentation, *Dependency Manager for PHP* : web-site. URL : <https://getcomposer.org/doc/>.
36. Computer technologies in information security : Monograph / V. Zadiraka, Ya. Nykolaichuk (eds.). Ternopil : Kart-blansh, 2015. P. 387.
37. Container Stowage Planning and how it works. URL : <http://panbras.com.br/CONTAINER%20STOWAGE%20PLANNING.pdf>.
38. CoCoX CoIDE. URL : <http://www.cocox.org/software/coide.php>.
39. Cortes J, Martinez S, Karatas T, Bullo F. Coverage control for mobile sensing networks. *IEEE Trans Robot Autom*. 2004. No. 20 (2). P. 243–255.

40. Cortex-A9. Revision r4p1 : technical reference manual. 2012. URL : http://infocenter.arm.com/help/topic/com.arm.doc.ddi0388i/DDI0388I_cortex_a9_r4p1_trm.pdf.
41. CPU Stability Test. *BenchmarkHQ. Test utilities [BenchmarkHQ. Testovye utility]*. URL : <http://www.benchmarkhq.ru/russian.html?b.html>.
42. Craig R. Flocks, herds and schools: A distributed behavioral model. *Computer graphics and interactive techniques (Association for Computing Machinery) SIGGRAPH'87 : Proceedings of the 14th Annual conference*. 1987. P. 25–34. DOI :10.1145/37401.37406.
43. Crowther B. Flocking of autonomous unmanned air vehicles. *The aeronautical journal*. February 2003. Vol. 107, Is. 1068. P. 99–109. DOI : 10.1017/S0001924000018388.
44. CUAV hack link 2.4G digital link HDMI PPM & Mavlink for Pixhawk. Publ. Mar. 01, 2017. URL : <https://www.rcgroups.com/forums/showthread.php?2843604-CUAV-HACK-LINK-2-4G-Digital-Link-HDMI-PPM-Mavlink-for-PIXHAWK>.
45. D'Andrea R. Guest editorial can drones deliver? *IEEE Transactions on Automation Science and Engineering*. 2014, July. Vol. 11, Is. 3, P. 647–648. DOI : 10.1109/TASE.2014.2326952.
46. Darintsev O. V., Migranov A. B., Yudin B. S. Neural network algorithm of planning trajectories for a group of mobile robots. *Artificial Intelligence / Ufa State Aviation Technical University*. 2011. No. 1. P. 154–160.
47. De 6 pitch finalisten van Web Summit 2015. *Technologium : website*. Publ. Nov. 5, 2015. URL : <http://www.technologium.be/web-summit-2015-de-pitch-finalisten>.
48. Development of multi-threaded applications using optimization method for platforms. *Intel software developer zone : documentation*. Publ. 2011, February 3. URL : <https://software.intel.com/ru-ru/articles/61695>.
49. Distributed Autonomous Robotic Systems 7 / M. Gini, R. Voyles (eds.). Springer Science & Business Media, 2007. 248 p.
50. DJI Phantom 4 WM330A – Drone : Product teardown report. Publ. June 20, 2016. URL : <https://techinsights.com/reports-and-subscriptions/open-market-reports/Report-Profile/?ReportKey=11239>.
51. Documentation Gearman, *Gearman : web-site*. URL : <http://gearman.org/documentation/>.
52. Drones in agriculture. *Digital Transformation Monitor / European Commission*. 2018, Jan. 6 p. URL : <https://agropro.club/articles/bezpilotniki-suchasnijinstrument-dlya-agrariya/>.
53. Durišić M., Tafa Z. Z., Dimić G., et al. A Survey of military applications of wireless sensor networks : presentation. *Embedded computing (MECO-2012) : Proceedings of mediterranean conference, Bar, Montenegro, Serbia, 19 – 21 June, 2012*. P. 196–199. URL : http://home.etf.rs/~vm/ppt/1%20A_Survey_Military_Apps_WSNs_MPDj.ZT.GD.VM.pdf.
54. Espressif ESP8266, Qualcomm Atheros AR9331 : Datasheet Search. URL : http://www.datasheet4u.com/share_search.php?sWord=ESP8266.
55. FATMAP : Hi-Res 3D Ski Mapping App Launches in the Alps / A. Traynor. Publ. Feb. 10, 2017. URL : <https://snowbrains.com/fatmap-hi-res-3d-ski-mapping-app-launches-alps/>.
56. Flight Control Systems : User's Guide. URL : <http://docs.sky-drones.com/SmartAP-User-Guide.pdf>.
57. Frank Announces Drones with Enhanced Capabilities to Address Industrial Use. Publ. : Sept. 8, 2016. URL : <https://dronelife.com/2016/09/08/interdrone-update-prodrone-announces-drones-with-enhanced-capabilities-to-address-industrial-use/>.
58. Friese L., Jenzen-Jones N.R., Smallwood M. The Chinese QLZ87 automatic grenade launcher emerging unmanned threats : The use of commercially-available UAVs by armed non-state actors (Special report) / Australia by Armament Research Services (ARES). 2016, Feb. 2016. No. 2. 66 p.
59. Geier E. 7 Tips to Increase Wi-Fi Performance. *Wi-Fi planet homepage*. Publ. 2012, March 8. URL <http://www.wi-fiplanet.com/tutorials/7-tips-to-increase-wi-fiperformance.html>.
60. Geier E. How to extend your Wi-Fi network. *PC World*. Publ. Aug. 28, 2012. URL : http://www.pcworld.com/article/261495/how_to_extend_your_wi-fi_network.html.

61. GeoFlyer Europa 3D – Mappa Offline GPS Percorsi. URL : <https://play.google.com/store/apps/details?id=com.geomind.europe&hl=it>.
62. Gibbs M. Linux in the Air : Drone systems go open-source. Ten drone systems that use Linux : Presentation. *Network World from IDG* : website. Publ. Apr. 21, 2015. URL : <https://www.networkworld.com/article/2912381/wireless/linux-in-the-air-drone-systems-go-open-source.html> (Last accessed : June 7, 2018).
63. Goldman J. Drones hit new heights at CES 2016. CNET, Gadgets, January 10, 2016. URL : <http://www.cnet.com/news/drones-ces-2016/>.
64. Google Maps Static API. Publ. June 7, 2018. URL : <https://developers.google.com/maps/documentation/maps-static/intro?hl=en>.
65. Google Maps Static. URL : <https://maps.googleapis.com/maps/api/staticmap?center=46.9682166,32.0129256&zoom=12&size=400x400>.
66. Grama A., Karypis G., Kumar V., et al. Introduction to parallel computing. 2nd ed. Addison-Wesley, 2003. 656 p. URL : <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.726.694&rep=rep1&type=>
67. Green zones : Drone.UA together with ecologists are exploring Kiev. Publ. Oct. 29, 2016. URL : <http://drone.ua/green-zones-drone-ua-together-with-ecologists-are-exploring-kiev/>.
68. Grossman P. Discrete mathematics for computing. 2nd ed. N. Y. : Palgrave Macmillan, 2002. 316 p.
69. Guang Y., Vikram K. Optimal path planning for unmanned air vehicles with kinematic and tactical constraints. *Decision and control* : Proceedings of the 41th IEEE Conference, Las Vegas, Nevada, USA, 10–13 Dec., 2002. Vol. 2. P. 1301–1306.
70. Gummadi R., Wetherall D., Greenstein B., Seshan S. Understanding and mitigating the impact of RF interference on 802.11 networks (ACM SIGCOMM'07). New York, NY, USA : ACM, 2007. P. 385–396.
71. Hahanov V. I., Gus O. A., Ziarmand A., et al. Cloud traffic control system. *East-West Design & Test Symposium (EWDTS'2013)* : Proceedings of IEEE Symposium, Rostov-on-Don, Russia, 26–29 Sept., 2013. P. 1–6. DOI : 10.1109/EWDTS.2013.6673092.
72. Hahanov V., Gharibi W., Abramova L. S., et al. Cyber physical system-smart cloud traffic control. *East-West Design & Test Symposium (EWDTS'2014)* : Proceedings of IEEE Symposium, Kiev, Ukraine, 26–29 Sept., 2014. P. 1–18. DOI : 10.1109/EWDTS.2014.7027107.
73. Hamann H. Swarm Robotics: A Formal Approach. Springer, 2018. 210 p.
74. Haxhimusa Y., Carpenter E., Catrambone J., et al. 2D and 3D Traveling Salesman Problem. *The Journal of Problem Solving*. 2011. Vol. 3, Iss. 2, Art. 8. P. 167–193. DOI : 10.7771/1932-6246.1096.
75. Hekmat R. Ad-hoc networks : Fundamental properties and network topologies. Zoetermeer, The Netherland : Springer, 2006. 154 p.
76. Hiromoto R. E., Sachenko A., Kochan V., et al. Mobile Ad Hoc wireless network for pre- and post-emergency situations in nuclear power plant. *Intelligent Data Acquisition and Advanced Computing Systems* : Proceedings of IEEE 2nd International Symposium on Wireless Systems, Offenburg, Germany, 11–12 September, 2014. P. 92–96. DOI : 10.1109/IDAACS-SWS.2014.6954630.
77. Hla K. H. S., Choi Y. S., Park J. S. The multi agent system solutions for wireless sensor network. *Agent and Multi-Agent Systems : Technologies and Applications*. Vol. 4953. Springer, Incheon, Korea, 2008. P. 454–463. (Series : Lecture Notes in Computer Science).
78. Hlukhov V., Lukenyuk A., Shenderuk S. Satellite scientific data collection and accumulation system as a basis for cyber-physical systems construction. *Advances in Cyber-Physical Systems*. 2016. Vol. 1, No. 2. P. 1–6.
79. Hopfield J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*. 1982. Vol. 79. P. 2554–2558.

80. Hopfield J. J., Brody C. D. What is a Moment? Transient synchrony as a collective mechanism for spatiotemporal integration, *Proceedings of the NAS of the USA*. 2001. Vol. 98, No. 3. P. 1282–1287. DOI : 10.1073/pnas.98.3.1282
81. Howard A., Mataric M. J., Sukhatme G. S. An incremental self-deployment algorithm for mobile sensor networks. *Autonomous Robots, Special Issue on Intelligent Embedded Systems*. 2002. No. 13 (2). P. 113–126.
82. Increased accuracy of GPS coordinates Збільшення точності визначення GPS координат. *Intelli : система контролю та моніторингу транспорту в Україні* : веб-сайт. URL : <http://intelli.com.ua/ua/statti/zbilshennia-tochnosti-vyznachennia-gps-koordynat.html>.
83. Intel plans to break drone light show record with over 1,500 drones flown at 50th Anniversary Celebrations. *Intel Newsroom* : [site]. Publ. April 30, 2018. URL : <https://newsroom.intel.com/news/intel-plans-break-drone-light-show-record-over-1500-drones-flown-50th-anniversary-celebrations/>.
84. Intel® Math Kernel Library – documentation. Publ. Sept. 1, 2017. URL : <https://software.intel.com/en-us/articles/intel-math-kernel-library-documentation>.
85. Jun S., Yatskiv V., Sachenko A., Yatskiv N. Data transmission optimal routing in WSN using ant colony algorithm. *Modern Problems of Radio Engineering Telecommunications and Computer Science (TCSET'2012)* : Proceedings of IEEE International Conference, Lviv-Slavske, Ukraine, 21–24 Feb., 2012.
86. Junior L. S., Nedjah N. Efficient strategy for collective navigation control in swarm robotics. *Procedia Computer Science*. 2016. Vol. 80. P. 814–823. DOI : 10.1016/j.procs.2016.05.371.
87. Kalra N., Martinoli A. Comparative study of market-based and threshold-based task allocation. *Distributed autonomous robotic systems 7*. Japan : Springer, 2006. P. 91–101.
88. Karnon J. Modeling using discrete event simulation. *Medical decision making*. 2012. P. 705–706.
89. Kelley J. Learn Cocoa Touch for iOS. Detroit : Apress, 2012. 396 p.
90. Kennedy J. The particle swarm : social adaptation of knowledge. *Evolutionary Computation* : Proceedings of IEEE International Conference. 1997. P. 303–308.
91. Kennedy J., Eberhart R. C., Shi Y. *Swarm Intelligence*, San Francisco : Morgan Kaufmann Publishers, 2001. 512 p.
92. Kettenburg E. Oak by Digistump : Wi-Fi for all things! (Arduino Compatible) : Project. 2016. URL : <https://www.kickstarter.com/projects/digistump/oak-by-digistump-wi-fi-for-all-things-arduino-comp>.
93. Kharchenko V., Kondratyuk V., Ilnytska S., Kutsenko O., Larin V. Urgent problems of UAV navigation system development and practical implementation. *Actual Problems of Unmanned Air Vehicles Developments (APUAVD'2013)* : Proceedings of IEEE 2nd International Conference, Kiev, Ukraine, 15–17 Oct., 2013. P. 157–160. DOI : 10.1109/APUAVD.2013.6705313.
94. Kharchenko V., Sklyar V., Siora A. Dependability of safety-critical computer systems through component-based evolution. *Dependability of Computer Systems* : Proceedings of IEEE 4th International Conference, Brunow, Poland, 30 June – 2 July, 2009. DOI : 10.1109/DepCoS-RELCOMEX.2009.22.
95. Khlamov S. V., Savanevich V. E., Briukhovetskyi A. B., Oryshych S. S. Development of a computational method for detecting near zero visible motion of objects on a series of CCD-frames. *Eastern-European Journal of enterprise technologies*. 2016. No. 2/9 (80). P. 41–48 [in Russian]. DOI : 10.15587/1729-4061.2016.65999.
96. Kingston D., Holt R., Beardy R., et al. Decentralized perimeter surveillance using a team of UAVs. *AIAA Guidance, Navigation, and Control Conference and Exhibit*. San Francisco, California, 15–18 Aug., 2005. 7 p. DOI : 10.2514/6.2005-5831.
97. Kochan V., Sachenko A., Yatskiv V., Kocha O. Energy-efficient method for controlling the transmitters power of wireless sensor network. *Electrical and Computer Engineering (UKRCON)* : Proceedings of IEEE 1st Ukraine Conference, Kiev, Ukraine, 29 May – 2 June, 2017. P. 1117–1120. DOI : 10.1109/UKRCON.2017.8100423.

98. Kojic N., Reljin I., Reljin B. Route selection problem based on Hopfield neural network. *Radioengineering*. 2013, Dec. Vol. 22, No. 4. P. 1182–1193.
99. Krainyk Y., Perov V., Musiyenko M., Davydenko Y. Hardware-oriented turbo-product codes decoder architecture. *Intelligent data acquisition and advanced computing systems: technology and applications (IDAACS – 2017)*: Proceedings of the 2017 IEEE 9th International conference, Bucharest, Romania, 21–23 Sept., 2017. Vol. 1. P. 151–154. DOI : 10.1109/IDAACS.2017.8095067.
100. Kubasov I. A., Serdyukov P. N., et al. Estimation of communications capacity and range when employing Wi-Fi equipment in real urban environment. Special equipment. 2013, Feb. URL : <http://www.bnti.ru/showart.asp?aid=1018&lvl=04.01.01.03.&p=1>.
101. Kulakovska I. V., Zhuravska I. M. Mathematical modeling container network via a connected graph. *11th International Algebraic Conference in Ukraine dedicated to the 75th anniversary of V. V. Kirichenko*: Abstracts, Kyiv, Ukraine, 3–7 July, 2017 / Taras Shevchenko National University of Kyiv. P. 68. Also available from URL : <https://www.imath.kiev.ua/~algebra/iacu2017/abstracts>.
102. Laravel по-русски: Документация 5.x ; опубл. 30.06.2017. URL : <https://laravel.ru/docs/v5/>.
103. LARK FPV U842-1 Guide – UDIRC. URL : www.udirc.com/6920.
104. Lee E. A. Cyber-physical systems – are computing foundations adequate? *Cyber-physical systems : Research motivation, techniques and roadmap*: Position paper for NSF workshop, Austin, TX, USA, Oct. 16–17, 2006. URL : http://ptolemy.eecs.berkeley.edu/publications/Material/06/CPSPositionPaper/Lee_CPS_PositionPaper.pdf.
105. Li C., Zhang S., Wu H. Flying robot localization method based on multi-sensor fusion. *International Journal of Advanced Robotic Systems*. 2014. No. 11.178. 7 p. DOI : 10.5772/58927. URL : <http://journals.sagepub.com/doi/pdf/10.5772/58927>.
106. Li L., Alcherio M., Abu-Mostafa Ya. S. Learning and measuring specialization in collaborative swarm systems. *Adaptive Behavior*. 2004. No. 12 (3–4). P. 199–212.
107. Mahanti A., Carlsson N., Williamson C., Arlitt M. Ambient interference effects in Wi-Fi Networks. *IFIP Networking. Chennai [India]*. 2010. URL : <https://www.cs.usask.ca/grads/nic169/Material/networking10b.pdf>.
108. MapKit. Framework : documentation. URL : https://developer.apple.com/documentation/mapkit?changes=_8.
109. MathWorks. Free MATLAB Trial. URL : <https://se.mathworks.com/campaigns/products/trials.html>.
110. Maurice P. Pixhawk le nouveau contrôleur de vol 32bits. Publ. Oct. 15, 2013. URL : <https://www.mondrone.net/pixhawk-nouveau-controleur-de-vol-32bits/>.
111. MAVLINK Common Message Set: Official MAVLink Documentation (#24). URL : <http://mavlink.org/messages/common>.
112. MAVLink library for the C programming language. URL : https://github.com/mavlink/c_library.
113. MAVLink Micro Air Vehicle Communication Protocol. URL : <http://qgroundcontrol.org/mavlink/start>.
114. Melnyk A. Cyber-physical systems multilayer platform and research framework. *Advances in Cyber-Physical Systems*. 2016. Vol. 1, No. 2. P. 77–85.
115. Melnyk A. Cyber-physical systems: the problems of creation and directions of development. *Proceedings of of the National University «Lviv Polytechnic» [Visnyk]*. 2014. No. 806. P. 154–161 (Computer systems and networks).
116. Mesas-Carrascosa F.-J., Torres-Sánchez J., Clavero-Rumbao I., et al. Assessing optimal flight parameters for generating accurate multispectral orthomosaics by UAV to support site-specific crop management. *Remote Sensing*. 2015. No. 7. P. 12793–12814. DOI : 10.3390/rs71012793.

117. Min H. K., Wang Z. D. Group escape behavior of multiple mobile robot system by mimicking fish schools. *Robotics and biomimetics* : Proceedings of IEEE International conference. 2010. P. 320–326.
118. Modeling and Simulation-Based Systems Engineering Handbook / D. Gianni, A. D’Ambrogio, A. Tolk (eds.). CRC Press, 2014. 513 p.
119. MongoDB от теории к практике. Руководство по установке кластера mongoDB ; опубл. 28.03.2014. URL : <https://habrahabr.ru/post/217393/>.
120. Mueller M. W. Increased autonomy for quadcopter systems : trajectory generation, fail-safe strategies and state estimation : Doctoral Thesis ETH No. 23137. Zürich, 2016. DOI : 10.3929/ethz-a-010655275.
121. Mueller M. W., D’Andrea R. Critical subsystem failure mitigation in an indoor UAV testbed. *Intelligent Robots and Systems (IROS), 2012* : Proceedings of IEEE/RSJ International Conference, Vilamoura, Portugal, 7–12 Oct., 2012. P. 780–785. DOI : 10.1109/IROS.2012.6385910. Multiple Heterogeneous Unmanned Aerial Vehicles / A. Ollero, I. Maza (eds.). Springer, 2007. 238 p.
122. Multiple Heterogeneous Unmanned Aerial Vehicles / A. Ollero, I. Maza (eds.). Springer, 2007. 238 p.
123. Multi-Robot Systems : From Swarms to Intelligent Automata : Proceedings from the 2002 NRL Workshop on Multi-Robot Systems / A. C. Schultz, L. E. Parker (eds.). Springer Science & Business Media, 2013. 235 p.
124. Musiyenko M. P., Burlachenko I. S., Denysov O. O., Zhuravska I. M. Development of double median filter for optical navigation problems. *Data Stream Mining & Processing (DSMP–2016)* / Lviv Polytechnic National University : Proceedings of the IEEE 1st International Scientific and Technical Conference, Lviv (vil. Stare Selo), 23–27 Aug., 2016. P. 177–181. DOI : 10.1109/DSMP.2016.7583535. Also available from URL : <http://ieeexplore.ieee.org/document/7583535/>.
125. Musiyenko M. P., Zhuravska I. M., Burlachenko I. S., Denysov O. O. The principles of the cyber-physical components’ organization based on the methods of the multi-agent interaction of the moving objects. *Advances in Cyber-Physical Systems* / Lviv Polytechnic National University. 2016. Vol. 1, No. 1. P. 48–57. Also available from URL : <http://science2016.lp.edu.ua/acps/acps-vol-1-no-1-2016/1-1>.
126. Musiyenko M. P., Zhuravska I. M., Kulakovska I. V., Kulakovska A. V. Simulation the behavior of robot sub-swarm in spatial corridors. *Electronics and Nanotechnology (ELNANO – 2016)* : Proceedings of IEEE 36th International Conference, Kyiv, Ukraine, 19–21 Apr., 2016. P. 382–387. DOI : 10.1109/ELNANO.2016.7493090. Scopus EID : 2-s2.0-84979517480.
127. Nikiforov V. V. Feasibility of real-time applications on multicore processors. *SPIIRAS Proceedings*. 2009. No. 8. P. 255–284.
128. North Dakota UAS Test Site Continues Research Collaboration with NASA / North Dakota Department of Commerce. Publ. Apr. 24, 2018. URL : <https://www.commerce.nd.gov/news/NorthDakotaUASTestSiteContinuesResearchCollaborationwithNASA/>.
129. NP UAS TS Infrastructure. *Northern Plains Unmanned Aircraft Systems Test Site : Research*. URL : <http://www.npuasts.com/research.html>.
130. Oncha R. Project development of client side monitoring moving objects system for Android platform. *Innovations in Science and Technology* : Proceedings of the XV All-Ukrainian Students R&D Internet Conference, Kyiv, 01–18 Dec., 2015. URL : http://konfist.fl.kpi.ua/sites/default/files/oncha_thesis.pdf.
131. Parrot Apps : Android, iOS, Windows. *Parrot SA* : web-site. URL : <http://global.parrot.com/usa/apps/>.
132. Parrot Bebop + Skycontroller – quadcopter with navigation and camera. *Medgadgets : First blog-shop smart electronics*. Publ. Oct., 2015, URL : <http://medgadgets.ru/shop/parrot-bebop-skycontroller.html>.
133. Parrot Mambo Grabber. URL : <https://www.parrot.com/us/spareparts/drones/parrot-mambo-grabber>.

134. Pashkevich A., Kazheunikau M. Neural network approach to trajectory synthesis for robotic manipulators. *Journal of Intelligent Manufacturing*, 2005. Vol. 16. P. 173–187.
135. Payton D, Estkowski R, Howard M. Pheromone robotics and the logic of virtual pheromones. *Swarm robotics : Lecture notes in computer science*. 2005. Vol. 3342. P. 45–57.
136. Perdix fact sheet : Release of the strategic capabilities office DoD. URL : <https://www.defense.gov/Portals/1/Documents/pubs/Perdix%20Fact%20Sheet.pdf>.
137. Phantom 4 Pro : specifications. 2017. URL : <http://www.dji.com/ru/phantom-4-pro/info>.
138. Pix4Dcapture turns your consumer drone into a professional mapping tool. URL : <https://pix4d.com/product/pix4dcapture/>.
139. Prevot T., Kopardekar P., Rios J. et al. UAS traffic management (UTM) concept of operations to safely enable low altitude flight operations. *Aviation Technology Integration and Operations* : Paper of 16th AIAA conference, Washington Hilton, Washington D.C., USA , 13–17 June, 2016. DOI : 10.2514/6.2016-3292.
140. Products. Catalog. Processors. *Qualcomm Technologies Inc.* 2016. URL : <https://developer.qualcomm.com/hardware/which-processor>.
141. Property Process. ProcessorAffinity. Publ. 2016, Oct. URL : [https://msdn.microsoft.com/ru-ru/library/system.diagnostics.process.processoraffinity\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/system.diagnostics.process.processoraffinity(v=vs.110).aspx).
142. Proposal to create common rules for operating drones in Europe / European Aviation Safety Agency (EASA). 2015, Sept. URL : https://www.easa.europa.eu/system/files/dfu/205933-01-EASA_Summary%20of%20the%20ANPA.pdf.
143. PX4 autopilot. URL : <https://pixhawk.org/>.
144. Rahman T., Hariadi M., Sumpeno S. NCP striking pattern in combat situation using boids behaviour. *Intelligent technology and its application (ISITIA)* : Proceedings of IEEE International seminar, Surabaya, Indonesia, 22–24 May, 2014. Access mode: <http://isitia.its.ac.id/base/index.php/SITIA/2014/paper/viewFile/343/143>.
145. Ralink RT3290...RT5350 : Datasheet PDF. URL : <http://datasheet-pdf.com/PDF/>.
146. Rana A. S., Zalzal A. M. S. A neural networks based collision detection engine for multi-arm robotic systems. *Artificial neural networks* : Proceedings of the 5th International conference. 1997. P. 140–145.
147. Real-time Cooperative Behavior for Tactical Mobile Robot Teams Skills. Impact Study for Tactical Mobile Robot Operational Units : Report / Georgia Institute of Technology, Atlanta, Georgia, November 2000. 92 p. URL : <https://www.cc.gatech.edu/ai/robot-lab/tmr/skillsassessment.pdf>.
148. Reynolds C. Flocks, herds, and schools : a distributed behavioural model. *Computer Graphics*. July 1987. No. 21 (4). P. 25–34.
149. Richter J. CLR via C#. Full coverage of multicore programming Microsoft .NET Framework 4.5. 4th ed. Redmond, Washington : Microsoft Press, 2012. 813 p. URL : http://www.johnchukwuma.com/training/clr_via_c_4th_edition.pdf.
150. Robinson G., Davies J. B. C. The Amadeus project : an overview. *Industrial Robot* : an International Journal. 1997. Vol. 24, Is. 4. P. 290–296. DOI : 10.1108/01439919710176381.
151. Rosen K. Discrete mathematics and its applications. 6th ed. McGraw-Hill Education, 2006. 1006 p.
152. Rumiankov D., Zhuravska I., Solobuto L., Musiyenko M. Reduce noise like solar interference in computer networks based on Power Line Communication. *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2017)* : Proceedings of the 8th IEEE International Conference, Bucharest, Romania, 21–23 Sept., 2017. Vol. 1. P. 215–221. DOI : 10.1109/IDAACS.2017.8095079. URL : <http://ieeexplore.ieee.org/document/8095079/>.
153. Sachenko A., Kulakov Yu., Kochan V., et al. Computer networks. Ternopil, 2012. 476 p.
154. Sachenko A., Yatskiv V., Krepych R. Modified method of noise-immune data transmission in wireless sensors networks. *Networks Security, Wireless Communications and*

Trusted Computing : Proceedings of IEEE International Conference, Wuhan, Hubei, China, 25–26 Apr., 2009, Vol. 2. P. 847–850. DOI : 10.1109/NSWCTC.2009.391.

155. Savinov V. Development of energy efficient distributed computer systems with self-contained remote modules. *Electronics and Nanotechnology (ELNANO)* : Proceedings of 2014 IEEE 35th International conference, Kyiv, Ukraine, 15–18 Apr., 2014. P. 456–458. DOI : 10.1109/ELNANO.2014.6873427.

156. Schouwenaars T., How J., Feron E. Decentralized cooperative trajectory planning of multiple aircraft with hard safety guarantees. *AIAA Guidance, Navigation, and Control Conference and Exhibit* : AIAA Meeting Material, Providence, Rhode Island, 16–19 Aug., 2004. No. 5141. 14 p. DOI : 10.2514/6.2004-5141. URL : <http://acl.mit.edu/Material/AIAA-2004-5141.pdf>.

157. Shared vs VPS vs Dedicated vs Cloud Hosting. Publ. December 12, 2017. URL : <https://wp-rocket.me/blog/shared-vs-vps-vs-dedicated-vs-cloud-hosting/> (Last accessed : June 7, 2018).

158. Shi Z. G., Tu J., Zhang Q. et al. survey of swarm robotics system. *Advances in Swarm Intelligence* : Lecture Notes in Computer Science. 2012. Vol. 7331. P. 564–572.

159. Singh S. P. N., Thayer S. M. ARMS : Autonomous robots for military systems / The Robotics Institute Carnegie Mellon University Pittsburgh, Pennsylvania, 2001. 72 p. URL : https://www.ri.cmu.edu/pub_files/pub3/singh_surya_2001_2/singh_surya_2001_2.pdf.

160. Slusar V. Data transmission on board the UAV: NATO standards. *Electronics : Science, Technology, Business*. 2010. No. 3. P. 80–86.

161. Smailova S., Kvyetnyy R., Sofina O., et al. Method of image texture segmentation using Laws' energy measures. *Proceedings of SPIE – International Society for Optics and Photonics* [Poland]. 2017. Vol. 10445 ; Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments. DOI:10.1117/12.2280891.

162. Solesvik M., Kondratenko Yu., Kondratenko G., et al. Fuzzy decision support systems in marine practice. *Fuzzy Systems (FUZZ-IEEE)* : Proceedings of IEEE International Conference, Naples, Italy, 9–12 July, 2017. P. 1–6. DOI : 10.1109/FUZZ-IEEE.2017.8015471.

163. Space-time continuous models of swarm robotic systems : Supporting global-to-local programming / Ed. H. Hamann. Springer Science & Business Media, 2010. 160 p. DOI : 10.1007/978-3-642-13377-0.

164. State and Local Regulation of Unmanned Aircraft Systems (UAS) Fact Sheet / Federal Aviation Administration Office of the Chief Counsel. 2015, December 17. URL : https://www.faa.gov/uas/regulations_policies/media/UAS_Fact_Sheet_Final.pdf.

165. Stewart J. M., Chapple M., Gibson D. CISSP (ISC)2 Certified Information Systems Security Professional Official Study Guide. 7th ed. Indianapolis, Indiana : John Wiley & Sons. Inc., 2015. 1080 p.

166. STM32F429AG : Download Datasheet. URL : http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32-high-performance-mcus/stm32f4-series/stm32f429-439/stm32f429ag.html.

167. Subbotin S. A. Building a fully defined neuro-fuzzy network with a regular partition of a feature space based on large sample. *Radio electronics, computer science, control*. 2016. No. 3. P. 47–53.

168. Sullivan B. The UK wants swarms of drones for defence missions. Publ. Sept. 6, 2016. URL : https://motherboard.vice.com/en_us/article/yp35kv/the-uk-wants-swarms-of-drones-for-defence-missions.

169. Sutantyo D. K., Kernbach S., Nepomnyashchikh V. A., Levi P. Multi-robot searching algorithm using Levy Flight and artificial potential field. *Safety, Security, and Rescue Robotics (SSRR-2010)* : Proceedings of the 8th IEEE International Workshop, Bremen, Germany, 26–30 July, 2010. 6 p. URL : <https://arxiv.org/pdf/1108.5624.pdf>.

170. Talla V., Kellogg B., Ransford B., Naderiparizi S. et al. Powering the next billion devices with Wi-Fi. *Emerging Networking Experiments and Technologies (CoNEXT'2015)* :

Paper of the 11th International Conference, Heidelberg, Germany, 01–04 Dec., 2015. URL : <https://homes.cs.washington.edu/~gshyam/Material/powifi.pdf>. DOI : 10.1145/1235.

171. Tan Yi. Handbook of Research on Design, Control, and Modeling of Swarm Robotics. IGI Global, 2015. 854 p.

172. Tanenbaum A. S., Bos H. *Modern operating systems*. 4th ed. Amsterdam, The Netherlands : Pearson Prentice-Hall, 2015. 1072 p.

173. Task scheduler how to... *Microsoft TechNet*. 2017. URL : [https://technet.microsoft.com/en-gb/library/cc766428\(v=ws.11\).aspx](https://technet.microsoft.com/en-gb/library/cc766428(v=ws.11).aspx).

174. Teh S., Mejias L., Corke P., et al. Experiments in Integrating Autonomous Uninhabited Aerial Vehicles (UAVs) and Wireless Sensor Networks. *Robotics and Automation* : Proceedings of the Australasian Conference, Canberra, Australia, 3–5 Dec., 2008. URL : <http://www.araa.asn.au/acra/acra2008/Material/pap118s1.pdf>.

175. Tencent and Zerotech unveil commercial drone based on Qualcomm Snapdragon flight platform (Press release). *Qualcomm Technologies, Inc. and/or its affiliated companies* : website. Publ. 2016, May 6. URL : <https://www.qualcomm.com/news/releases/2016/01/05/tencent-and-zerotech-unveil-commercial-drone-based-qualcomm-snapdragon>.

176. The U. S. military wants to distribute Wi-Fi with drones. Publ. Apr. 18, 2014. URL : http://m.kaskus.co.id/thread/5350c9f2f9ca174c398b473c/the-us-military-wants-to-distribute-wi-fi-with-drones/?ref=postlist-54&med=recommended_or_you.

177. The WiFi-amplifier to AR.Drone 2 and other small items. *The art of system administration*. Publ. Apr. 12, 2015. URL : <http://macrodmin.ru/2015/04/wifi-ar-drone-2>.

178. To diagnose WiFi problems in complex networks. *MetaGeek.Com*. URL : <http://www.inssider.com/>.

179. Trasviña-Moreno C. A., Blasco R., Marco Á., et al. Unmanned aerial vehicle based wireless sensor network for marine-coastal environment monitoring, *Sensors [Basel, Switzerland]*, 2017, Mar. No. 17 (3). Vol. 460. 22 p. URL : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5375746/>. DOI : 10.3390/s17030460.

180. Trends in Intelligent Robotics, Automation, and Manufacturing / S. G. Poonambalam et al. (eds.). Springer, 2012. 528 p.

181. Trianni V., Groß R., Labella T. H. et al. Evolving aggregation behaviors in a swarm of robots. *Advances in artificial life. Lecture notes in computer science*. 2003. Vol. 2801. P. 865–874.

182. Troubleshooting task scheduler. *Microsoft TechNet*. 2017. URL : [https://technet.microsoft.com/en-gb/library/cc721846\(v=ws.11\).aspx](https://technet.microsoft.com/en-gb/library/cc721846(v=ws.11).aspx).

183. Tsourdos A., White B., Shanmugavel M. Cooperative path planning of unmanned aerial vehicles. Wiley, 2011. 190 p.

184. USBasp driver for Windows 7 and Windows Vista X64 : Announcements. Publ. May 24, 2011. URL : www.protostack.com/blog/2011/05/usbasp-driver-for-windows-7-and-windows-vista-x64.

185. VECTOR Technical Specification. URL : <https://www.uavnavigation.com/support/kb/autopilots/vector/vector-technical-specification>.

186. Vincent O. R., Folorunso O. A. Descriptive algorithm for Sobel image edge detection. *Proceedings of Informing Science & IT Education Conference (InSITE'2009)*, Macon, United States, 12–15 June, 2009. P. 97–107. DOI: 10.1.1.330.3734.

187. Wang N., Wang L., Go X., et al. Hopfield neural network guided evolutionary algorithm for aircraft penetration path planning. *Advances in neural network research and applications* : Lecture notes in electrical engineering. Vol. 67. Berlin ; Heidelberg : Springer-Verlag, 2010. P. 235–243. DOI: 10.1007/978-3-642-12990-2_27.

188. Wang X., Yadav V., Balakrishnan S. N. Cooperative UAV formation flying with obstacle/collision avoidance. *IEEE transactions on control systems technology*. 2007. Vol. 15, Is. 4. P. 672–679. DOI: 10.1109/TCST.2007.899191.

189. Watanabe T., Johnson E. N. Automatic landing on a moving platform using deep neural network estimation. *VTOL Unmanned Aircraft Systems and Autonomy* / American

Helicopter Society International : Proceedings of the 7th AHS Technical Meeting, Mesa, United States, 24–26 Jan., 2017. URL : <https://www.scopus.com/record/display.uri?eid=2-s2.0-85032922077&origin=inward&txGid=12fd37a855e5ff746a8ba3a228a30890> (Last accessed : June 11, 2018).

190. Werners B., Kondratenko Y. P. Alternative fuzzy approaches for efficiently solving the capacitated vehicle routing problem in conditions of uncertain demands. *Complex Systems : Solutions and Challenges in Economics, Management and Engineering* Part of the Part of the Studies in Systems, Decision and Control (SSDC) : book series. Vol. 125. Springer, Oct 31, 2017. P. 521–543.

191. Werners B., Kondratenko Y. P. Tanker routing problem with fuzzy demands of served ships. *Системні дослідження та інформаційні технології*. 2009. № 1. P. 47–64.

192. What Is Container Stowage Planning? Publ. Jan. 21, 2017. URL : <https://posey-intl.com/what-is-container-stowage-planning/> (Last accessed : June 11, 2018).

193. Wi-Fi – wireless standard. *TAdviser Portal (Technologies + Adviser)*. Publ. Mar. 13, 2018. URL : [http://www.tadviser.ru/index.php/Статья:Wi-Fi_\(Wireless_Fidelity\)_-_стандарт_беспроводной_связи_802.11](http://www.tadviser.ru/index.php/Статья:Wi-Fi_(Wireless_Fidelity)_-_стандарт_беспроводной_связи_802.11).

194. Wulfsohn D., Lagos I. Z. The use of a multicopter and high-resolution imaging for precision horticulture in Chile : an industry perspective. *Precision Agriculture : Proceedings of the 12th International Conference, Sacramento, California, USA, July, 2014*. Vol. 1688. 15 p.

195. Yan M. Dijkstra's algorithm : Presentation / Massachusetts institute of technology, department of mathematics. 2016. URL : <http://math.mit.edu/~rothvoss/18.304.3PM/Presentations/1-Melissa.pdf>.

196. Yatskiv V., Yatskiv N., Sachenko A., et al. Concept of designing the wireless sensor networks based on ant intelligence. *Intelligent Data Acquisition and Advanced Computing Systems : Technology and Applications (IDAACS'2015)* : Proceedings of IEEE 8th International Conference, Warsaw, Poland, 24–26 Sept., 2015. Vol. 2. P. 863–866. DOI : 10.1109/IDAACS.2015.7341426.

197. Yu P., Tate S. R. Online/Offline signature schemes for devices with limited computing capabilities. *The Cryptographers' Track at the RSA Conference'2008* : Proceedings of the conference, San Francisco, CA, USA, 8–11 Apr., 2008. DOI : 10.1007/978-3-540-79263-5_19.

198. Zhang W., Mueller M. W., D'Andrea R. A controllable flying vehicle with a single moving part. *Robotics and Automation (ICRA'2016)* : Proceedings of, IEEE International Conference, Stockholm, Sweden, 16–21 May, 2016. P. 3275–3281. DOI : 10.1109/ICRA.2016.7487499.

199. Zhang Yi. Z., Antonsson E. K., Martinoli A. Evolving neural controllers for collective robotic inspection. *Applied soft computing technologies, the challenge of complexity, advances in soft computing*. 2006. Vol. 34. P. 717–729.

200. Zhong C., Luo C., Chu Z., et al. A continuous Hopfield neural network based on dynamic step for the traveling salesman problem. *Neural Networks (IJCNN'2017)* : Proceedings of IEEE International Joint Conference, Anchorage, AK, USA, 14–19 May, 2017. P. 3318–3323. DOI : 10.1109/IJCNN.2017.7966272.

201. Zhuravska I. M., Koretska O. O., Musiyenko M. P., Surtel W., et al. Self-powered information measuring wireless networks using the distribution of tasks within multicore processors. *Proceedings of SPIE – International Society for Optics and Photonics [Poland]*. 2017. Vol. 10445 : Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments. No. 1044527. P. 1–13. DOI: 10.1117/12.2280965.

202. Zhuravska I. M., Lavrynenko S. V. Implementation the monitoring CPU loading application for mobile devices. *Information technology and interactions (IT&T-2017) / Taras Shevchenko National University of Kyiv* : Proceedings of the 4th International Scientific Conference. Kyiv, Ukraine, 8–10 Nov., 2017. P. 112–113.

203. Zhuravska I. M., Lernatovych D. O. Fuzzy modeling system of human behavior and biometric identification using cloud services. *Applied Sciences in Europe : Tendencies of*

Contemporary Development : Materials of the 5th International Scientific Conference, Stuttgart, Germany, 24 March, 2014. P. 30–32.

204. Zhuravska I. M., Popel M. I. Automation of pharmaceutical warehouse using groups robots with remote climate control and video surveillance. *Science and Innovation in the XXI century* : Collection of Conference Materials of International Scientific-Practical Conference, London, the United Kingdom, 11–12 Dec., 2014. P. 30–32. DOI : 10.5281/zenodo.14090.

205. Zhuravska I., Borovlova S., Kostyria M., Koretska O. Efficiency improvement of using unmanned aerial vehicles by distribution of tasks between the cores of the computing processor. *Technology audit and production reserves*. 2017. Vol. 6, No. 6 (38). P. 4–13. DOI : 10.15587/2312-8372.2017.117889.

206. Zhuravska I., Kulakovska I., Musiyenko M. Development of a method for determining the area of operation of unmanned vehicles formation by using the graph theory. *Eastern-European Journal of Enterprise Technologies*. 2018. Vol. 2, No. 3 (92). P. 4–12. DOI : 10.15587/1729-4061.2018.128745.

207. Абрамов В. О., Клименко С. Ю. Базові технології комп'ютерних мереж : навч. посіб. 2-ге вид. Київ : А.С.К., 2013. 248 с.

208. Автоматизированные информационные системы для удалённого отслеживания / Starcom Systems Ltd – GPS Tracking Systems URL : <https://www.starcomsystems.com/ru/products>.

209. Альбов А. Квантовая криптография. СПб. : ООО «Страта», 2015. 248 с.

210. Антошук С. Г., Годовиченко Н. А. Модели представления событий при анализе видеопотока. *Электротехнические и компьютерные системы*. 2013. № 11 (87). С. 142–149.

211. Антошук С. Г., Крылов В. Н., Максимов М. В. Помехоустойчивая сегментация бинаризованных изображений символов. *Труды Одес. политехн. ун-та*. 1998. № 2. 3 р. URL : [http://storage.library.opu.ua/online/periodic/opu_1998_2\(6\)/2_14.pdf](http://storage.library.opu.ua/online/periodic/opu_1998_2(6)/2_14.pdf).

212. Антошук С. Г., Сербина Н. А. Система распознавания текстурных изображений при экологическом мониторинге. *Искусственный интеллект*. 2002. № 4. С. 406–413. URL : http://www.iai.dn.ua/public/JournalAI_2002_4/Razdel4/01_AntoshchukSerbina.pdf.

213. Барчинський М. В., Димедюк С. В., Журавська І. М. Програмна платформа мобільної системи керування рухомою моніторинговою мережею. *Інформаційні технології та взаємодії (IT&I – 2015)* / Київ. нац. ун-т ім. Тараса Шевченка та ін. : тези доп. II Міжнар. наук.-практ. конф., Київ, 3–5 листоп. 2015 р. С. 85–87.

214. Белецкий А. Я., Максименко А. В., Навроцкий Д. А. и др. Программно-моделирующий комплекс ВРС алгоритма поточного шифрования и помехоустойчивого кодирования видеосигналов, передаваемых с борта БПЛА. *Захист інформації*. 2014. Т. 16, № 3. С. 184–191. DOI : 10.18372/2410-7840.16.7534.

215. Беспилотные решения для работы с площадными и протяженными объектами. *Drone.UA* : веб-сайт. URL : http://drone.ua/?gclid=CjwKEAju7O6vBRDpi7O-8OWSkwESJACNFsgxZxOf5GI-hpr3StSFdlCtnxjOzPKAjCfcugf0RY4DXhoC607w_wcB.

216. Биард Р. У., МакЛэйн Т. У. Малые беспилотные летательные аппараты : теория и практика. М. : Техносфера, 2015. 312 с.

217. Боровльова С. Ю., Журавська І. М., Костирия М. А. Бенчмаркінг ефективності організації ітераційних процесів у багатоядерному процесорі. *Могілянські читання – 2017* / Чорном. нац. ун-т ім. Петра Могили : тези доп. XX Всеукр. наук.-метод. конф., Миколаїв, 13–17 листоп. 2017 р. С. 57–60 (Секція : АСУ, CASE-засоби та програмна інженерія).

218. Бражников Р. Уровни API, версии Android, –target, файл AndroidManifest.xml ; опубл. 20.03.2015. URL : <http://romanbrazhnikov.ru/blog/2015/03/20/уровни-апи-версии-android-target-файландroidmanifest-xml/>.

219. Бурлаченко І. С., Денисов О. О., Журавська І. М. Вибір апаратної платформи для проектування та реконфігурації рухомих об'єктів моніторингових мереж. *Могілянські читання – 2015* / Чорном. держ. ун-т ім. Петра Могили : тези доп. XVIII Всеукр. наук.-метод. конф. Миколаїв, 12–20 листоп. 2015 р. Т. 1. С. 113–115.

220. Бурнашев В. В., Збруцкий А. В. Управление автоматическим приземлением беспилотного самолета : навч. посіб. Киев : НТУУ «КПІ», 2015. 125 с.
221. Взлом беспилотника займет у комплекса РЭБ «Шиповник-АЭРО» секунду. *Интерфакс. Новости ВПК* ; опубл. 14.09.2016. URL : http://vpk.name/news/163641_vzлом_bespilotnika_zaimet_u_kompleksa_reb_shipovnikaero_sekundu.html.
222. Головинський А. Л., Маленко А. Л., Сергієнко І. В., Тульчинський В. Г. Енергоефективний суперкомп'ютер СКІТ-4. *Вісник Національної академії наук України*. 2013. №. 2. С. 50–59. URL : <http://dspace.nbuv.gov.ua/handle/123456789/43042>.
223. Гонсалес Р., Вудс Р. Цифровая обработка изображений. 3-е изд. М. : Техносфера, 2012. 1082 с.
224. Давыдов В. О., Максимов М. В. Сравнительный анализ моделей нейронных сетей для решения задач классификации образов. *Труды Одес. политехн. ун-та*. 2002. № 1 (17). С. 1–3. URL : [http://storage.library.opu.ua/online/periodic/opu_2002_2\(18\)/4/4-5.pdf](http://storage.library.opu.ua/online/periodic/opu_2002_2(18)/4/4-5.pdf).
225. Довгенко О. Є., Журавська І. М. Моделі та методи розробки інформаційно-вимірвальних систем моніторингу температурного режиму побутових споруд. *Могілянські читання – 2017* / Чорном. нац. ун-т ім. Петра Могили : тези доп. XX Всеукр. наук.-метод. конф., Миколаїв, 13–17 листоп. 2017 р. С. 31–32 (Секція «Комп'ютерна інженерія»).
226. Документация Varnish ; опубл. 21.10.2014. URL : <http://php-include.ru/stati/dokumentaciya-varnish>.
227. Дрон «Катран»: что о нём известно ; опубл. 11.05.2018. URL : <http://rusnext.ru/news/1526016168>.
228. Єлезаров В. О., Журавська І. М. Технологія Інтернету речей для забезпечення почергового підключення побутової техніки до електромережі. *Могілянські читання – 2017* / Чорном. нац. ун-т ім. Петра Могили : тези доп. XX Всеукр. наук.-метод. конф., Миколаїв, 13–17 листоп. 2017 р. С. 32–34 (Секція : Комп'ютерна інженерія).
229. Жарук Д. О., Журавська І. М., Мусієнко М. П. Програмне забезпечення на базі хмарного сервісу Google Firebase для керування безпілотними апаратами. *Інформаційне, програмне та технічне забезпечення систем управління організаційно-технологічними комплексами* / Луцьк. нац. техн. ун-т. : зб. тез Міжнар. наук.-практ. конф., Луцьк, 28–29 квіт. 2017 р. Луцьк : Вежа-Друк, 2017. С. 16.
230. Жерегі О. М. Дослідження та розробка програмного забезпечення абонентів рухомих моніторингових мереж на мобільній платформі Windows Phone. *Інтелектуальні інформаційні системи – 2016* : тези доп. Всеукр. наук.-практ. конф., Миколаїв, 16–17 лютого 2016 р. Миколаїв : Вид-во Чорномор. держ. ун-ту ім. Петра Могили, 2016. Т. 1. С. 150–152.
231. Жуков І. А., Ластовченко М. М., Іскренко Ю. Ю. Анализ процессов адаптивной коррекции радиоспектра при передаче мультимедийного трафика шумоподобными сигналами. *Проблеми інформатизації та управління*. 2008. Т. 2, Вип. 24. С. 66–73.
232. Жуков І. А., Печурин Н. К., Кондратова Л. П. и др. Распределение ресурсов в вычислительном кластере для БПЛА. *Проблеми інформатизації та управління*. 2016. Т. 3, Вип. 55. С. 34–38.
233. Жуков І. А., Дрововозов В. І., Масловський Б. Г. Експлуатація комп'ютерних систем та мереж : навч. посіб. Київ : НАУ, 2007. 368 с.
234. Журавская И. Н. Система восстановления пароля доступа к сторонней WiFi-сети в чрезвычайных ситуациях. *Надзвичайні ситуації : безпека та захист (НСБЗ'2016)* / Держ. служба України з надзв. ситуацій, Черкас. ін-т пожеж. безпеки ім. Героїв Чорнобиля Нац. ун-ту цивільн. захисту України та ін. : матеріали VI Міжнар. наук.-практ. конф., Черкаси, 21–22 жовт. 2016 р. С. 138–140.
235. Журавская И. Н. Современное состояние резонансных высоковольтных высокочастотных источников питания. *Теория, эксперимент, практика электроразрядных технологий* : сб. науч. тр. / Отв. ред. В. В. Шамко. Киев : Наук. думка, 1995. Вып. 2. С. 110–113.

236. Журавская И. Н., Божаткин С. М. Защита компьютерной сети применительно к облачным средам. *Сучасні проблеми інформаційної безпеки на транспорті* / Нац. ун-т кораблебудування : тези доп. III Всеукр. наук.-техн. конф. з міжнар. участю, Миколаїв, 3–6 верес. 2013 р. С. 61–64 (Секція 2 : Захист інформації в каналах зв'язку та глобальних мережах передачі даних). Відомості доступні також в Інтернеті URL : <http://conference.nuos.edu.ua/catalog/lectureDetail;jsessionid=00e8013f98b0bcb01232beacdce6?lectureId=23848&conferenceId=19978&isProjectorView=false>.
237. Журавская И. Н., Назарова Н. С. Определение эквивалентной диэлектрической проницаемости изоляции двухобмоточного трансформатора. *Теория, эксперимент, практика электроразрядных технологий* : сб. науч. тр. / Отв. ред. В. С. Крутиков ; ИИПТ НАН Украины. Николаев : Атолл, 2002. Вып. 4. С. 115–122.
238. Журавская И. Н., Щербюк В. О. Создание стегоконтейнеров в телекоммуникации с БПЛА на основе Open Source протокола MAVLink. *Free and Open Source Software (FOSS-2015)* / Харк. нац. ун-т будівництва та архітектури : тези доп. VII Всеукр. наук.-практ. конф., Харків, 24–27 листоп. 2015 р. С. 36. Відомості доступні також в Інтернеті URL : http://ekit.org.ua/sites/default/files/file_attach/FOSS/FOSS2015/FOSS-2015_Sbornik_tez.pdf.
239. Журавська І. М. IoT-мережа на базі Bluetooth-модулів для автоматизованого керування споживанням енергоресурсів. *Комп'ютерно-інтегровані технології : освіта, наука, виробництво* / Луцьк. нац. техн. ун-т. 2018. Вип. 30–31. С. 37–44. Відомості доступні також в Інтернеті URL : <http://ki.lutsk-ntu.com.ua/node/138/section/11>.
240. Журавська І. М. Аналіз та вибір WiFi-мережі в умовах нестабільної ефірної ситуації. *Технологічний аудит та резерви виробництва* : наук. журн. Харків : Технологічний центр, 2013. Т. 5. № 5 (13). С. 8–10. DOI: 10.15587/2312-8372.2013.18371.
241. Журавська І. М. Використання трансферних вузлів рухомих мереж для атаки на комп'ютерні системи наземних абонентів мережі. *Методи та засоби кодування, захисту й ущільнення інформації* / Вінниц. нац. техн. ун-т: тези доп. VI Міжнар. наук.-практ. конф., Вінниця, 24–25 жовт. 2017 р. С. 58–60.
242. Журавська І. М. Забезпечення стабільного бездротового зв'язку у кіберфізичній системі з рухомими об'єктами. *Технологічний аудит та резерви виробництва* : наук. журн. Харків : Технологічний центр, 2016. Т. 5, № 2 (31). С. 58–64. DOI: 10.15587/2312-8372.2016.80784.
243. Журавська І. М. Забезпечення функціонування суб-роїв безпілотних літальних апаратів за допомогою бортових датчиків. *Датчики, прилади та системи (ДПС – 2017)* / Черкас. держ. технол. ун-т : тези доп. VI Міжнар. наук.-техн. конф., Херсон–Лазурне, 18–22 верес. 2017 р. С. 27–32.
244. Журавська І. М. Метод організації захищеного бездротового каналу з інтегрованим стегоконтейнером для моніторингової мережі відеокамер. *Інформаційна безпека та комп'ютерні технології (IS&CT 2017)* / Центральноукр. нац. техн. ун-т : зб. тез доп. II Міжнар. наук.-практ. конф., Кропивницький, 20–22 квіт. 2017 р. С. 46–47.
245. Журавська І. М. Проектування комп'ютерних мереж. Ч. I : Локальні мережі : навч.-метод. посіб. Миколаїв : Вид-во ЧДУ ім. Петра Могили, 2011. 124 с. (Методична серія ; вип. 171).
246. Журавська І. М. Проектування та монтаж локальних комп'ютерних мереж : навч. посіб. Миколаїв : Видавництво ЧДУ ім. Петра Могили, 2016. 396 с.
247. Журавська І. М. Реєстрація великорозмірних вантажів за допомогою автоматизованої системи на базі пристроїв з обмеженими обчислювальними можливостями. *Електро-технічні і комп'ютерні системи* / Одес. нац. політехн. ун-т. 2017. № 26 (102). С. 60–67. DOI: 10.15276/eltecs.26.102.2017.7.
248. Журавська І. М., Богаченко Є. В., Вдовиченко А. В. Організація відеотрансляції прес-конференції з локальної комп'ютерної мережі на центральний канал телебачення з використанням технології VLAN. *Могілянські читання – 2014* / Чорном. держ. ун-т ім. Петра Могили : тези доп. XVII Всеукр. наук.-метод. конф., Миколаїв, 17–21 листоп. 2014 р. С. 57–58.

249. Журавська І. М., Божаткін С. М. Забезпечення мережевої безпеки хмарних середовищ. *Інформаційні управляючі системи та технології (ІУСТ-ОДЕСА-2013)*. / Одес. нац. морський ун-т : матеріали Міжнар. наук.-практ. конф. / відп. ред. В. В. Вичужанін, Одеса, 8–10 жовт. 2013 р. С. 207–210 (Секція 4 : Засоби і методи захисту інформаційних систем).

250. Журавська І. М., Божаткін С. М., Мосензов Є. О. Програмне забезпечення підбору параметрів та розробка абонентського обладнання доступу до WiFi-мережі. *Комп'ютерні системи та мережеві технології (CSNT'2010)* / Нац. авіац. ун-т : тези доп. III Міжнар. наук.-техн. конф., Київ, 15–17 черв. 2010 р. С. 18.

251. Журавська І. М., Залобовський М. В., Мільошин А. С. Автоматизована система обліку великих вантажів з реєстраторами на базі мобільних пристроїв. *Могиланські читання – 2017*. / Чорном. нац. ун-т ім. Петра Могили : тези доп. XX Всеукр. наук.-метод. конф., Миколаїв, 13–17 листоп. 2017 р. С. 35–37 (Секція : Комп'ютерна інженерія).

252. Журавська І. М., Кулаковська І. В., Кулаковська А. В. Математичне моделювання мережі контейнеровозу за допомогою зв'язного графа. *Ольвійський форум – 2017 : Стратегії країн Причорноморського регіону в геополітичному просторі* / Чорном. нац. ун-т ім. Петра Могили : тези доп. XI Міжнар. наук.-практ. конф., Миколаїв – Коблево, 8–11 черв. 2017 р. С. 51–54.

253. Журавська І. М., Кулаковська І. В., Мусієнко М. П. Автоматизація обліку розвантаження суден-контейнеровозів. *Сучасні інформаційні та інноваційні технології на транспорті (MINTT – 2017)* / Херсон. держ. мор. академія : тези доп. 9 Міжнар. наук.-практ. конф., Херсон, 23–25 трав. 2017 р. С. 203–206.

254. Журавська І. М., Лавриненко С. В., Обухова К. О. Моніторинг навантаження обчислювальних систем БПЛА. *Free and Open Source Software* / Харків. нац. ун-т буд-ва та архітектури : тези доп. IX Міжнар. наук.-практ. конф., Харків, 21–23 листоп. 2017 р. С. 72.

255. Журавська І. М., Мусієнко М. П., Румянков Д. І. Підвищення ефективності шифрування керуючого трафіку БПЛА засобами модифікованого блокового методу. *Методи та засоби кодування, захисту й ущільнення інформації* / Вінниц. нац. техн. ун-т : тези доп. V Міжнар. наук.-практ. конф., Вінниця, 19–21 квіт. 2016 р. С. 39–41. Відомості доступні також в Інтернеті URL : <http://ir.lib.vntu.edu.ua/handle/123456789/13214>.

256. Журавська І. М., Мусієнко М. П. Мобільна система керування гетерогенною рухомою мережею безпілотних апаратів. *Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка*. Київ : ВІКНУ, 2017. Вип. 57. С. 30–40. Відомості доступні також в Інтернеті URL : http://mil.univ.kiev.ua/files/240_1382369481.pdf.

257. Журавська І. М., Мусієнко М. П. Мобільний реєстратор завантаження / розвантаження великорозмірних об'єктів військово-цивільного призначення. *Військова освіта і наука : сьогодення та майбутнє* / Військ. ін-т Київ. нац. ун-ту ім. Тараса Шевченка : тези доп. XIII Міжнар. наук.-практ. конф. Київ, 24 листоп. 2017 р. С. 37–38.

258. Журавська І. М., Мусієнко М. П. Синтез маршрутів суб-роїв безпілотних апаратів з використанням нейронної мережі Хопфілда для обстеження територій. *Radio Electronics, Computer Science, Control* / Zaporizhzhya National Technical University, 2017. No. 3. P. 86–94. DOI: 10.15588/1607-3274-2017-2-10. Also available from URL : <http://ric.zntu.edu.ua/article/view/115479>.

259. Журавська І. М., Мусієнко М. П., Кулаковська І. В. Автоматизована система обліку розвантаження суден контейнеровозів з використанням мобільних пристроїв. *Наук. вісник Херсон. держ. мор. академії* : наук. журн. 2017. Вип. 1 (16). С. 37–46.

260. Журавська І. М., Мусієнко М. П., Румянков Д. І. Блоковий метод шифрування для рухомих об'єктів з обмеженими обчислювальними ресурсами. *Інформаційні технології та комп'ютерна інженерія* / Вінниц. нац. політехн. ун-т. 2016. Т. 3, № 37. С. 28–32. Відомості доступні також в Інтернеті URL : <http://itce.vntu.edu.ua/index.php/itce/article/view/523/394>.

261. Журавська І. М., Мусієнко М. П., Щебетюк В. О. Захист інформації щодо місцезнаходження БПЛА за допомогою стеганографічних рішень. *Інформація, комунікація, суспільство – 2016 (ICS – 2016)* / Нац. ун-т «Львів. політехніка»: тези доп. V Міжнар. наук. конф., Львів – Славське, 19–21 трав. 2016 р. С. 24–25. Відомості доступні також в Інтернеті URL : <http://ena.lp.edu.ua:8080/bitstream/ntb/33285/1/04-24-25.pdf>.
262. Журавська І. М., Мусієнко М. П., Юрін Д. В. Аналіз безпеки механізмів обміну даними між наземним пунктом керування та безпілотним літальним апаратом. *Інформаційні технології та взаємодії (IT&I – 2015)* / Київ. нац. ун-т ім. Тараса Шевченка та [ін.] : тези доп. II Міжнар. наук.-практ. конф., Київ, 3–5 листоп. 2015 р. С. 47–48.
263. Журавська І. М., Олійник В. В. Підсистема катапультування та визначення GPS-координат блоків БПЛА при критичному застосуванні. *Могілянські читання – 2016* / Чорном. нац. ун-т ім. Петра Могили : тези доп. XIX Всеукр. наук.-метод. конф., Миколаїв, 12–20 листоп. 2016 р. Т. 5. С. 83–84.
264. Журавська І. М., Поліщук Д. В. Аналіз загроз у публічних бездротових мережах та захисту користувацьких даних. *Могілянські читання – 2015* / Чорном. держ. ун-т ім. Петра Могили : тези доп. XVIII Всеукр. наук.-метод. конф., Миколаїв, 12–20 листоп. 2015 р. Т. 1. С. 78–79.
265. Журавська І. М., Попель М. І. Автоматизація фармацевтичного складу з використанням спеціалізованих роботів, дистанційного клімат-контролю та відеоспостереження. *Могілянські читання – 2014* / Чорном. держ. ун-т ім. Петра Могили : тези доп. Всеукр. наук.-метод. конф., Миколаїв, 17–21 листоп. 2014 р. С. 63.
266. Журавська І. М., Румянков Д. І. Корпоративний месенджер з шифруванням трафіку модифікованим блочним методом на базі операції XOR. *Комп'ютерні інтелектуальні системи та мережі (KICM – 2016)* / Криворіз. нац. ун-т : тези доп. IX Всеукр. наук.-практ. WEB конф. студентів, аспірантів та молодих вчених, Кривий Ріг, 22–24 берез. 2016 р. С. 135–137.
267. Журавська І. М., Румянков Д. І. Система автоматичного керування на основі Arduino з мікроконтролером Atmega для прийняття рішень щодо керування польотом БПЛА. *Могілянські читання – 2016* / Чорном. нац. ун-т ім. Петра Могили : тези доп. XIX Всеукр. наук.-метод. конф., Миколаїв, 12–20 листоп. 2016 р. Т. 5. С. 24–26.
268. Журавська І. М., Румянков Д. І. Створення модифікованого блокового методу шифрування на базі операції XOR для корпоративного месенджера. *Наукові праці / Чорном. держ. ун-т ім. Петра Могили : наук. журн. 2015. Т. 266, Вип. 254. С. 97–104 (Серія : Комп'ютерні технології)*. Відомості доступні також в Інтернеті URL : http://nbuv.gov.ua/UJRN/Npchduct_2015_266_254_17.
269. Журавська І. М., Савінов В. Ю., Корецька О. О., Буренко В. О. Розподілення навантаження між багатоядерними обчислювачами для задач енергонезалежних інформаційно-вимірювальних мереж. *Наукові праці / Чорном. нац. ун-т ім. Петра Могили : наук. журн. 2017. Т. 307, Вип. 295. С. 42–46 (Серія : Комп'ютерні технології)*.
270. Журавська І. М., Савінов В. Ю., Лавриненко С. В., Обухова К. О. Імплементация програмного забезпечення моніторингу та диспетчеризації навантаження ядер процесора рухомого пристрою. *Могілянські читання – 2017.* / Чорном. нац. ун-т ім. Петра Могили : тези доп. XX Всеукр. наук.-метод. конф. Миколаїв, 13–17 листоп. 2017 р. С. 112–115 (Секція : Методи, моделі та інформаційні технології).
271. Журавська І. М., Салтан Б. А., Собко Д. А. Аналіз захищеності Wi-Fi та механізми розбудови безпечної мережі. *Могілянські читання – 2015* / Чорном. держ. ун-т ім. Петра Могили : тези доп. XVIII Всеукр. наук.-метод. конф., Миколаїв, 12–20 листоп. 2015 р. Т. 1. С. 79–81.
272. Журавська І., Корецька О., Мусієнко М. Розроблення бездротових енергонезалежних інформаційно-вимірювальних мереж критичного застосування військово-цивільного призначення. *Інтернет – Освіта – Наука – 2016* / Вінниц. нац. політехн. ун-т та [ін.] : матеріали X Міжн. наук.-практ. конф., Вінниця, 11–14 жовт. 2016 р. С. 79–81.

273. Журавська, І. М., Мусієнко М. П., Румянков Д. І. Підвищення ефективності шифрування керуючого трафіку БПЛА засобами модифікованого блокового методу. *Методи та засоби кодування, захисту й ущільнення інформації* / Вінниц. нац. техн. ун-т : тези доп. V Міжнар. наук.-практ. конф., Вінниця, 19–21 квіт. 2016 р. С. 75–77.
274. Згурец С. Беспилотники: призыв на войну. Київ : Defense Express, 2015. 96 с. (Серия : Оружие Украины).
275. Карташов В. М., Бабкін С. І., Кушнір М. В. Удосконалення технологій радіоакустичного зондування атмосфери. *Радіотехніка*. 2014. Вип. 178. С. 5–12.
276. Квасников В. П., Дзюбаненко А. В. Улучшение визуального качества цифрового изображения путём поэлементного преобразования. *Авиационно-космическая техника и технология*. 2009. Вип. 8. С. 200–204.
277. Квасников В. П., Бабич Л. О. Управление интеллектуальным мобильным роботом на основе гибридной нейро-нечеткой системы. *Авиационно-космическая техника и технология*. 2009. № 7. С. 140–144.
278. Квасніков В. П., Орнатський Д. П., Гаврилов І. В. та ін. Способи побудови аналогових інтерфейсів інформаційно-вимірювальних систем механічних величин. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2013. № 1. С. 164–169.
279. Кветний Р. Н., Титарчук Є. О. Алгоритм частково гомоморфного шифрування на основі еліптичних кривих. *Контроль і управління в складних системах (КУСС – 2016)* : матеріали XIII Міжнар. конф., Вінниця, 3–6 жовт. 2016 р. URL : <https://ir.lib.vntu.edu.ua/handle/123456789/13077>.
280. «Клешня» для дрона ; опубл. 10.12.2015. URL : <https://naked-science.ru/article/concept/kleshnya-dlya-drona>.
281. Козленко В. Ethernet сети в системах управления транспортом. 2006. URL : http://www.moxa.com.ua/news_events/news/2006/mm_03_2006.htm.
282. Комашинский В. И., Смирнов Д. А. Нейронные сети и их применение в системах управления и связи. М. : Горячая линия-Телеком, 2003. 94 с.
283. Комп'ютерні мережі : навч. посіб. / О. Д. Азаров, С. М. Захарченко, О. В. Кадук та ін. Вінниця : ВНТУ, 2013. 371 с.
284. Комплекс с беспилотным летательным аппаратом «Форпост» / А. В. Карпенко. *Оружие Отечества. Оружие и военная техника* : выпуски сборника ; опубл. 14.01.2018. URL : <http://bastion-opk.ru/forpost/>.
285. Комплексне використання групи безпілотної літальних апаратів для задач промислово-екологічної розвідки / Нац. техн. ун-т України «Київ. політехн. ін-т ім. Ігоря Сікорського» : наук.-дослід. робота (№ держ. реєстрації 0115U002524) / наук. керівник О. В. Збруцький. URL : http://report.kpi.ua/files/2016_2860.pdf.
286. Контейнер – КОНТРОЛЬ, система взвешивания контейнеров и распознавания их номеров. URL : <http://www.azpw.ru/node/24>.
287. Крылов В. Н., Максимов М. В., Давыдов В. О. Автоматизированная система распознавания. *Искусств. интеллект*. 2002. № 4. С. 462–464.
288. Кутковецкий В. Я. Дослідження операцій : навч. посіб. Київ : Професіонал, 2004. 350 с.
289. Кучук Г. А. Інформаційні технології управління інтегральними потоками даних в інформаційно-телекомунікаційних мережах систем критичного призначення ; Харків. нац. ун-т Повітряних Сил ім. І. Кожедуба. Харків : ХУПС, 2013. 254 с.
290. Левитин А. В. Алгоритмы : введение в разработку и анализ. М. : Вильямс, 2006. 576 с.
291. Ленков С. В., Перегудов Д. А., Хорошко В. А. Методы и средства защиты информации : в 2-х т. Киев : Арий, 2008. Т. 1 : Несанкционированное получение информации. Т. 2 : Информационная безопасность. 344 с.
292. Ленков С. В., Шворов С. А., Болбот І. М. та ін. Дискретна векторна оптимізація траєкторій руху мобільних роботів. *Сучасна спеціальна техніка*. 2012. Вип. 1(28). С. 13–19.

293. Лужецький В. А., Баришев Ю. В. Криптографічні примітиви для реалізації керованого хешування. *Вісник Вінницького політехнічного інституту* : наук. журн. 2011. № 1. С. 108–111.
294. Лужецький В. А., Остапенко А. В. Блоковий шифр на основі псевдовипадкової послідовності криптопримітивів. *Системи обробки інформації*. 2010. Вип. 3(84). С. 136.
295. Миков А. И. Стохастические характеристики разрезв в графах AD НОС сетей. *Информатизация и связь*. 2014. № 1. С. 76–79.
296. Минухин С. В. Энергоэффективные алгоритмы масштабирования скорости процессоров вычислительного кластера. *Cluster Computing (CC'2013)* : Proceedings of Second International Conference, Lviv, Ukraine, 3–5 June, 2013. P. 131–140.
297. Мирошниченко Л. Н., Пусев А. Н., Журавская И. Н. Эффективное управление многоячеечными зарядными устройствами на базе однотактных транзисторных инверторов. *Теория, эксперимент, практика электроразрядных технологий* : сб. науч. тр. / Отв. ред. В. В. Шамко. Киев : Наук. думка, 1995. Вып. 2. С. 107–110
298. Моисеев В. С. Основы теории эффективного применения беспилотных летательных аппаратов : монография. Казань : Школа, 2015. 444 с.
299. Мудров В. И. Задача коммивояжера. М. : Знание, 1969. 62 с.
300. Мусієнко М. П., Журавська І. М. Алгоритми прокладання маршруту безпілотних літальних апаратів на основі застосування нейронних мереж Хопфілда. *Вісник Черкаського держ. технол. ун-ту* : зб. наук. праць. 2016. № 1. С. 20–27 (Серія : Технічні науки). Відомості доступні також в Інтернеті URL : http://visnyk.chdtu.edu.ua/images/tech/1_2016/c20-27_Мусієнко_Журавська.pdf.
301. Мусієнко М. П., Журавська І. М. Система контролю безпілотних апаратів, заснована на використанні мобільних пристроїв. *Комп'ютерно-інтегровані технології : освіта, наука, виробництво* / Луцьк. нац. техн. ун-т. 2017. Вип. 26. С. 199–203. Відомості доступні також в Інтернеті URL : <http://ki.lutsk-ntu.com.ua/node/134/section/33>.
302. Мусієнко М. П., Журавська І. М., Бурлаченко І. С. та ін. Рухомі моніторингові мережі критичного застосування : проблеми створення та напрями розвитку. *Наукові праці / Чорном. держ. ун-т ім. Петра Могили* : наук. журн. 2015. Т. 266, Вип. 254. С. 112–118 (Серія : Комп'ютерні технології).
303. Найман В. С., Самойлов А. Е., Ильин Н. Р. и др. Всё о GPS-навигаторах. М. : НТ Пресс, 2005. 392 с.
304. Накул Ю. А., Стовманенко В. О. Модель системи завантаження крупнотонажних суден. *Наукові праці / Чорном. нац. ун-т ім. Петра Могили* : наук. журн. 2016. Т. 275, Вип. 287. С. 27–33 (Серія : Комп'ютерні технології).
305. Никольский В. В., Накул Ю. А., Стовманенко В. О. Система контролю завантаження контейнеровозів. *Судовождение*. 2017. Вип. 27. С. 127–136.
306. Никольский В. В., Хнюнин С. Г., Накул Ю. А. Выбор датчиков определения координат контейнера для компьютерной системы учета загрузки судна. *Автоматизация судовых технических средств* : науч.-техн. сб. 2017. Вып. 23. С. 57–64.
307. Нікольський В. В., Нікольський М. В., Накул Ю. А. Система підтримки прийняття рішення по навантаженню великотонажного контейнеровоза. *Наукові праці / Чорном. нац. ун-т ім. Петра Могили* : наук. журн. 2016. Т. 271, Вип. 283. С. 70–73 (Серія : Комп'ютерні технології).
308. Один день на контейнеровозе. *Образовательный портал для судоводителей*. URL : <https://deckofficer.ru/news/item/odin-den-na-kontejnerovoze>.
309. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы. 4-е изд. СПб. : Питер, 2012. 943 с.
310. Опанасенко В. М., Лисовой О. М. Бортові проблемно-орієнтовані процесори для апаратної реалізації алгоритмів управління космічними апаратами. *Проблеми інформатизації та управління*. 2014. Т. 47, № 3. С. 71–75.
311. Пат. на винахід 21356 Україна, МПК (2006.01) H03K 17/61. Спосіб передачі уніполярних прямокутних імпульсів через гальванічно розв'язане коло та пристрій для

його здійснення / Л. М. Мірошниченко, І. М. Журавська, О. М. Пусев. № 93007773; заявл. 08.12.93; держ. реєстр. 02.12.97; опубл. 30.04.98, Бюл. № 2. 11 с. Відомості доступні також в Інтернеті URL : <http://base.uipv.org/searchINV/search.php?action=viewdetails&IdClaim=47479>.

312. Пат. на винахід 3031 Україна, МПК⁵ H01L 23/34. Пристрій для охолодження напівпровідникових приладів / І. М. Журавська, Л. М. Мірошниченко, О. М. Пусев, О. С. Євстаф'єв, М. В. Турти. № 93300896; заявл. 16.03.93; держ. реєстр. 29.04.94; опубл. 26.12.94, Бюл. № 5. 10 с. Відомості доступні також в Інтернеті URL : <http://base.uipv.org/searchINV/search.php?action=viewdetails&IdClaim=39424&chapter=biblio>.

313. Пат. на винахід 3033 Україна, МПК (2006.01) H02M 3/335. Багатоканальний перетворювач напруги / О. М. Пусев, І. М. Журавська, Л. М. Мірошниченко, В. М. Рябенський. № 93300894; заявл. 16.03.93; держ. реєстр. 29.04.94; опубл. 26.12.94, Бюл. № 5–І. 8 с. Відомості доступні також в Інтернеті URL : <http://base.uipv.org/searchINV/search.php?action=viewdetails&IdClaim=39426>.

314. Патент на изобретение 2 155 702 РФ, МПК B64C 27/10. Система двух соосных несущих винтов летательного аппарата / Б. С. Малышев, А. И. Соловьев, Г. В. Зевиг и др.; заявл. 15.04.1999; гос. регистр. 15.04.1999; опубл. 10.09.2000, Бюл. № 25. URL : <http://www.freepatent.ru/patents/2155702>.

315. Петрівний О. І. Дослідження точності супутникових навігаційних систем. *Наукові читання – 2013* / Житомир. нац. агроєкологіч. ун-т : наук.-теорет. зб. 2013. Т. 1. С. 367–373.

316. Полетайкин А. Н., Гузь О. А., Каганюк А. К. Интеллектуальные средства и технологии инфраструктуры городского дорожного движения. *Комп'ютерно-інтегровані технології : освіта, наука, виробництво*. 2014. Вып. 14. С. 155–163. URL : <http://ki.lutsk-ntu.com.ua/node/122/section/24>.

317. Поликарпова Н., Герасименко А. Методы решения труднорешаемых задач. С.-Петербург. нац. исслед. ун-т информ. технологий, механики и оптики (ИТМО), 2007. URL : <http://rain.ifmo.ru/cat/view.php/theory/unsorted/approx-2004> (дата обращения: 11.06.2018).

318. Польшинкин А. В., Ле Х. Т. Исследование характеристик радиоканала связи с беспилотными летательными аппаратами. *Известия Тул. гос. ун-та. Техн. науки*. 2013. Т. 7–2. С. 98–107.

319. Правила повітряної експлуатації безпілотних повітряних суден в Україні (концепція) / Р. Байдуж : презентація; опубл. 6.10.2017. URL : <https://www.slideshare.net/SAAU2017/ss-80646310>.

320. Прикладна криптологія. Теорія. Практика. Застосування : монографія / І. Д. Горбенко, Ю. І. Горбенко; Харк. нац. ун-т радіоелектроніки, Приват. акціонер. т-во «Ін-т інформ. технологій». Вид. 2-ге. Харків : Форт, 2012. 878 с.

321. Про визначення можливості застосування радіоелектронних засобів на території України : рішення НКРЗІ від 19.12.2017 № 648. URL : <http://nkrzi.gov.ua/index.php?r=site/index&prg=342&id=7396&language=uk>.

322. Продукты и решения – Беспроводное оборудование – DAP-3310. *D-Link (Europe) Ltd* : веб-сайт. URL : http://www.dlink.ru/ru/products/2/1790_b.html.

323. Пролетарский А. В., Баскаков И. В., Федотов Р. А. и др. Беспроводные сети Wi-Fi. 2-е изд. М. : BINOM, 2016. 284 р.

324. Пролетарский А. В. Технологии защиты информации в компьютерных сетях / Моск. гос. техн. ун-т им. Баумана. М. : D-Link, 2013. 278 с.

325. Простая модель планировщика ОС. *Хабрахабр : пост*; опубл. 12.10.2012. URL : <https://habrahabr.ru/post/154609/>.

326. Проценко М. М. Аналіз структури та варіантів побудови безпілотних авіаційних комплексів. *Вісник Житомир. держ. техн. ун-ту*. 2012. № 2 (61). С. 113–118.

327. Разведывательный беспилотный аппарат малого класса «Корсар» / А. В. Карпенко. *Военно-техн. сб. «Бастиян»* : журн. оборонно-промышл. комплекса; опубл. 04.09.2015. URL : <http://bastion-karpenko.ru/korsar-bla/>.

328. Российские войска установили в Донецке станцию радиоэлектронной борьбы «Красуха-4» и глушат беспилотники ОБСЕ. *Цензор.НЕТ* : веб-сайт. URL : https://censor.net.ua/photo_news/310972/rossiyiskie_voyiska_ustanovili_v_donetske_stantsiyu_radioelektron_noyi_borby_krasuha4_i_glushat_bespilotniki.
329. Рудницький В. М., Бабенко В. Г. Синтез математичних моделей пристроїв декодування інформації для криптографічних систем. *Системи обробки інформації*. 2009. Вип. 2 (76). С. 124–128.
330. Рудницький В. М., Костирка О. В. Стійке стеганоперетворення в просторовій області зображення-контейнера. *Інформатика та математичні методи в моделюванні*. 2013. Вип. 3, № 4. С. 353–360.
331. Румянков Д. І., Журавська І. М. Проектування системи керування безпілотним літальним пристроєм на основі мікроконтролерів Atmega на базі Arduino. *Інтелектуальні інформаційні системи – 2017* / Чорном. нац. ун-т ім. Петра Могили : тези доп. Всеукр. наук.-практ. конф., Миколаїв, 15–17 лют. 2017 р. С. 126–128.
332. Савчук С. Г. Проблемні питання під час використання сучасних супутникових технологій визначення координат. *Геодезія, картографія і аерофотознімання* / Нац. ун-т «Львівська політехніка». 2007. Вип. 69. С. 20–33.
333. Саченко А. О., Кочан В. В., Харченко В. С., Яцків В. В. та ін. Концепція побудови мобільних систем пост-аварійного моніторингу АЕС з використанням флоту квадрокоптерів. *Радіоелектронні і комп'ютерні системи*. 2016. № 5 (79). С. 207–214. URL : http://nbuv.gov.ua/UJRN/recs_2016_5_34.
334. Семенов С. Г., Гавриленко С. Ю., Давыдов В. В. Защита данных в компьютеризированных управляющих системах. Саарбрюккен, Германия : LAP Lambert Academic Publishing, 2014. 236 с.
335. Сидоренко Б. Микроконтроллеры Atmel SAM D – Cortex-M0+: оптимальное соотношение производительности и энергоэффективности. *Электроника : наука | технология | бизнес*. 2014. № 3 (00134). С. 78–85.
336. Синицкая И. Firebase Analytics : настраиваем аналитику мобильных приложений. *Netpeak Blog* : веб-аналитика. URL : <https://netpeak.net/ru/blog/nastraivaem-analitiku-mobilnyh-prilozheniy-s-pomoshchyu-firebase-analytics/>.
337. Ситников В. С., Троян А. П. Динамическое планирование действий автономной мобильной робототехнической системы. *Науковий вісник Чернівецького університету*. 2011. Т. 2, № 2. С. 30–33 (Серія : Комп'ютерні системи та компоненти).
338. Скворцов А. В., Мирза Н. С. Алгоритмы построения и анализа триангуляции. Томск : Изд-во Томского ун-та, 2006. 168 с.
339. Современные информационные технологии в задачах навигации и наведения беспилотных маневренных летательных аппаратов / под ред. М. Д. Красильщикова, Г. Г. Серебрякова. М. : Физматлит, 2009. 556 с.
340. Субботин С. А. Формирование выборки и анализ качества моделей на основе нейронных и нейро-нечетких сетей в задачах диагностики и распознавания образов : монография. Саарбрюккен, Германия : LAP Lambert academic publishing, 2012. 232 с.
341. Терентьев В. М. Задача управления полетом ДПЛА в групповых порядках и способы и средства ее решения. *Авиакосмическое приборостроение*. 2009. №. 1. С. 10–25.
342. Тимочко О. І., Голубничий Д. Ю., Третяк В. Ф. та ін. Класифікація безпілотних літальних апаратів. *Системи озброєння і військова техніка*. 2007. Вип. 1 (9). С. 61–66. URL : http://www.hups.mil.gov.ua/periodic-app/article/1045/soivt_2007_1_19.pdf.
343. Тимчасовий порядок використання повітряного простору України (ВППУ) : затв. Держ. авіац. службою України 31 трав. 2018 р. 169 с. URL : https://avia.gov.ua/wp-content/uploads/2018/05/ТУМЧАСОВУЙ-poryadok-VPPU_31.05.18-2.doc.
344. Трунов О. М. Розвиток методів та засобів створення аск глибоководними технологічними комплексами : автореф. дис. ... д-ра техн. наук : 05.13.07 – автоматизація процесів керування / Чорном нац. ун-т ім. Петра Могили. Херсон, 2017. 44 с.

345. Уоссермен Ф. Нейрокомпьютерная техника : Теория и практика. М. : Мир, 1992. 240 с.
346. Ушаков Д. В. Организация контейнерных перевозок грузов. М. : Транслит, 2015. 208 с.
347. Фісун М. Т., Журавська І. М., Горбань Г. В. Використання OLAP-технології для аналізу мережевого трафіку засобами об'єктної СКБД. *Комп'ютерні системи та мережеві технології (CSNT'2012)* / Нац. авіац. ун-т : тези доп. V Міжнар. наук.-техн. конф. Київ, 13–15 черв. 2012 р. С. 129.
348. Фісун М. Т., Журавська І. М., Горбань Г. В. Інтеграція даних мережевого трафіку мультисервісної корпоративної мережі з класами постріляційної СКБД Caché. *Наукові праці : наук. журн. / Чорном. держ. ун-т ім. Петра Могили*. 2011. Т. 173, Вип. 161. С. 105–110 (Серія : Комп'ютерні технології). Відомості доступні також в Інтернеті URL : <http://lib.chdu.edu.ua/pdf/naukpraci/computer/2011/173-161-16.pdf> (дата звернення : 25.08.2016).
349. Фісун М. Т., Журавська І. М., Горбань Г. В. Інтеграція даних щодо мережевого трафіку мультисервісної корпоративної мережі з класами постріляційної СКБД Caché. *Комп'ютерні системи та мережеві технології (CSNT'2011)* / Нац. авіац. ун-т : тези доп. IV Міжнар. наук.-техн. конф., Київ, 14–16 черв. 2011 р. С. 63.
350. Фракталы увеличили эффективность превращения тепла в электричество ; опубл. 02.10.2015. URL : <https://nplus1.ru/news/2015/10/02/levy-fractals>.
351. Хайкин С. Нейронные сети : полный курс. 2-е изд. М. : Вильямс, 2008. 1103 с.
352. Харди Б., Филлипс Б., Стюарт К. и др. Android. Программирование для профессионалов. 2-е изд. СПб. : Питер, 2016. 640 с.
353. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. М. : Триумф, 2002. 816 с.
354. Шульц Р. В., Крельштейн П. Д., Маліна І. А. Дослідження систематичних похибок інерціальної навігаційної системи при аерофотозніманні з безпілотних літальних апаратів. *ScienceRise : scientific journal*. 2015. № 9/2 (14). С. 6–18. DOI: 10.15587/2313-8416.2015.49183.
355. Юдін О. К. Кодування в інформаційно-комунікаційних мережах : монографія. Київ : НАУ, 2007. 308 с.
356. Яцків В. В. Теоретичні основи створення і структурна організація компонентів безпроводних сенсорних мереж підвищеної ефективності : автореф. дис. ... д-ра техн. наук : 05.13.05 – комп'ютерні системи та компоненти. Львів : НУ «Львівська політехніка», 2016. 40 с.

АЛФАВІТНИЙ ПОКАЖЧИК

- автопілот, 17, 22, 23, 24
адаптер, 60, 61, 66, 69
АС, 7, 34, 129, 137, 161, 162
БПА, 10, 11, 19, 22, 32, 33, 38, 40, 59,
60, 61, 62, 63, 64, 65, 66, 67, 72, 73,
74, 75, 76, 85, 113, 117, 139, 142,
143, 144, 145, 146, 147, 162
БПЛА, 7, 8, 9, 10, 12, 13, 14, 15, 17,
18, 19, 20, 21, 22, 23, 24, 25, 26, 27,
30, 32, 33, 34, 35, 36, 37, 40, 41, 47,
48, 49, 50, 51, 54, 56, 57, 73, 74, 75,
76, 77, 78, 79, 81, 85, 87, 88, 92, 93,
94, 96, 97, 98, 106, 108, 109, 110,
111, 112, 113, 114, 115, 116, 117,
118, 119, 120, 124, 125, 126, 127,
128, 129, 130, 133, 137, 138, 144,
163, 164, 181, 182, 183, 184, 185,
186
Брассіні, 43
великорозмірний вантаж, 159, 160
великорозмірний об'єкт, 159
ГРКМ, 7, 8, 9, 25, 34, 40
Дейкстри, 42
Дрони, 17, 114, 116, 117
ДЦ, 7, 25
ЕОС, 12
задача комівояжера, 51, 110
інтерференція, 61, 62, 71
канал зв'язку, 60, 72
КО, 59, 60
коригування маршруту, 85, 125
КФО, 8, 59, 61, 62, 63, 65, 66, 67, 70,
71, 72, 82, 84, 85, 86, 88, 106, 128,
130, 132, 134
КФС, 9, 51, 53, 55, 57, 59, 60, 61, 62,
63, 65, 66, 67, 68, 69, 70, 71, 72, 81,
82, 83, 84, 85, 86, 106, 119, 128
МРЧ, 19
нейронна карта, 54
НМХ, 27, 29, 30, 44, 48, 50, 52, 53,
54, 55, 56, 108, 110, 112, 113, 117
операція XOR, 88, 89
перешкоди, 12, 20, 24, 41, 60, 63, 64,
75, 121, 125
політ Леві, 19, 20
потужність передавача, 61, 63, 69, 70
просторовий коридор, 41, 51, 52, 57
протокол MAVLink, 94, 103, 104
Рій, 17
РМ, 11, 40
РММ, 8, 13, 25, 26, 30, 31, 32, 34, 82,
87, 93, 94, 97, 100, 106, 107, 108,
139, 140, 142, 143, 145, 147, 162
СВР, 12
синтез маршрутів, 54
СК, 34, 39
стегоконтейнер, 94, 97
ТД, 25, 59, 60, 61, 62, 66, 69, 70, 72
ТР, 12
триангуляція Делоне, 139
тривимірні мапи, 30, 31, 140, 142
ЦЗ, 11, 14, 37, 38, 39, 40, 57
ЦОД, 12, 59
чутливість приймача, 63, 69, 71
шардінг, 81
Швидкість передачі даних, 71
шифрування, 87, 88, 89, 90, 91, 92,
93, 106, 184, 185, 186

Наукове видання

*Ірина
ЖУРАВСЬКА*

**Гетерогенні комп'ютерні мережі
критичного застосування
на основі роїв та зграй БПЛА**

Монографія

Редактор *А. Грубкіна*. Технічний редактор, комп'ютерна верстка *Д. Кардаш*.
Друк *С. Волинець*. Фальцювальні-палітурні роботи *О. Кутова*.

Підп. до друку 05.12.2018.
Формат 70×100¹/₁₆. Папір офсет.
Гарнітура «Times New Roman». Друк ризограф.
Ум. друк. арк. 15,6. Обл.-вид. арк. 11,21.
Тираж 300 пр. Зам. № 5533.

Видавець і виготовлювач: ЧНУ ім. Петра Могили.
54003, м. Миколаїв, вул. 68 Десантників, 10.
Тел. : 8 (0512) 50–03–32, 8 (0512) 76–55–81, e-mail: rector@chmnu.edu.ua.
Свідоцтво суб'єкта видавничої справи ДК № 6124 від 05.04.2018