

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інформаційної безпеки та електронних комунікацій

(повне найменування інституту, факультету)

Інформаційних технологій та телекомунікацій

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

Магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ
НАВЕДЕННЯ БПЛА КВАДРОКОПТЕРНОГО ТИПУ НА РУХОМІ ОБ'ЄКТИ ПІД ЧАС
ПОЛЬОТУ

Виконав: студент(ка) 6 курсу, групи БК-612м

Спеціальності 151 Автоматизація та комп'ютерно-
інтегровані технології

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Автоматизація, мехатроніка та робототехніка

Керівник Олександр МАЛИЙ

(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Факультет інформаційної безпеки та електронних комунікацій
 Кафедра Інформаційних технологій електронних засобів
 Ступінь вищої освіти Магістр
 Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
(код і найменування)
 Освітня програма (спеціалізація): Автоматизація, мехатроніка та робототехніка
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ
 Завідувач кафедри ІТЕЗ _____
 «_____» _____ 20__ року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Моїсєєв Володимир Євгенович
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Дослідження та програмна реалізація методів наведення БПЛА квадрокоптерного типу на рухомі об'єкти під час польоту

керівник проекту (роботи) Малий Олександр Юрійович, к.т.н, доцент, завідувач каф.ІТЕЗ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від «23» квітня 2024 року № 162

2. Строк подання студентом проекту (роботи) 05 червня 2024

3. Вихідні дані до проекту (роботи) структура системи керування БПЛА

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Огляд області розробки та постановка задач, Математичні та алгоритмічні засади наведення на рухомий об'єкт, Вибір середовища симуляції, Структура системи, Результати моделювання алгоритмів, Тривалість переслідування, Поведінка при переслідуванні, Алгоритмічні передумови для вибору елементів, Порядок роботи з апаратним забезпеченням, Вибір апаратного забезпечення, Відпрацювання роботи апаратної частини

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
28 слайдів

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1	Олександр МАЛИЙ	03.04.24	
2	Олександр МАЛИЙ	17.04.24	
3	Олександр МАЛИЙ	07.05.24	
4	Олександр МАЛИЙ	21.05.24	
нормоконтроль	Ірина ПОСПЕЄВА	01.06.24	

7. Дата видачі завдання «19» лютого 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Математичні та алгоритмічні засади наведення на рухомий об'єкт	02.03.24	
2	Вибір середовища симуляції	11.03.24	
3	Структура системи	22.03.24	
4	Результати моделювання алгоритмів	02.04.24	
5	Порядок роботи з апаратним забезпеченням	15.04.24	
6	Вибір апаратного забезпечення	05.05.24	
7	Відпрацювання роботи апаратної частини	22.05.24	
8	Оформлення ПЗ та захист дипломного проєкту	01.06.24	

Студент(ка)

(підпис) Володимир МОЇСЄВ
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Олександр МАЛИЙ
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до магістерської роботи: 81 сторінок, 74 рисунки, 25 джерел.

БЕЗПЛОТНИЙ ЛІТАЛЬНИЙ АПАРАТ, ЦІЛЬ, НАВЕДЕННЯ, НАВІГАЦІЯ, ПОЛЬОТНИЙ КОНТРОЛЕР, ПРОТОКОЛ, ПРОГРАМА

Мета роботи – дослідження методів та розробка програмно-апаратних алгоритмів наведення БПЛА на рухомі повітряні об'єкти для захисту територій при несанкціонованому проникненні.

Об'єкт дослідження: технологія комп'ютерного зору для обробки зображень з камер БПЛА з метою виявлення рухомих об'єктів та програмно-апаратна реалізація систем відстеження рухомих об'єктів.

Предмет дослідження: методи наведення БПЛА квадрокоптерного типу на рухомі об'єкти під час польоту.

У цьому дипломному проекті розроблено методи, алгоритми та систему на їх основі яка може успішно виявляти та відстежувати рухомий об'єкт у сценаріях реального часу. Ця система реалізується за допомогою чотирьох різних алгоритмів.

При розрізненні кадрів порівнюються два послідовні кадри, а рухомий об'єкт виявляється та відстежується. Після виявлення та відстеження рухомого об'єкту апаратна частина подає сигнал контролеру БПЛА для керування рухом для слідування на виявлений об'єкт. Система реалізована на двох платформах. Перша — це платформа моделювання, а друга — апаратна платформа. Для моделювання використовувалось ПЗ Gozebo, Matlab та Proteus. Повна апаратна модель розроблена в Proteus і змодельована.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 81 pages, 74 drawings, 1 appendices, 25 sources.

UNMANNED AIRCRAFT, TARGET, GUIDANCE, NAVIGATION, FIELD CONTROLLER, PROTOCOL, PROGRAM

The purpose of the work is to research methods and develop software and hardware algorithms for targeting UAVs at moving aerial objects to protect the territory in the event of unauthorized intrusion.

Research object: computer image technology for image processing from a UAV camera for the purpose of detecting moving objects and software and hardware implementation of a moving object tracking system.

The subject of research: methods of guiding quadrocopter-type UAVs on moving objects during flight.

This thesis project develops methods, algorithms and a system based on them that can successfully detect and track a moving object in real-time scenarios. This system is implemented using four different algorithms.

When slicing, frame resolutions are compared to the next two frames, and a moving object is detected and tracked. After detecting and tracking a moving object, the hardware sends a signal to the UAV controller to control the movement to follow the given object. The system is implemented on two platforms. The first is a simulation platform and the second is a hardware platform. Gozebo, Matlab and Proteus software were used for simulation. The complete hardware model was developed in Proteus and simulated.

ЗМІСТ

Реферат.....	4
Вступ	8
1 Огляд області розробки та постановка задач.....	10
1.1 Протидія насанкціонованим польотам БПЛА.....	10
1.2 Проблеми реалізації наведення квадрокоптерів на рухомі повітряні цілі	12
1.3 Постановка задач	13
2 Математичні та алгоритмічні засади наведення на рухомий об'єкт.....	15
2.1 Класичне візуальне захоплення цілі.....	15
2.2 Математичне забезпечення наведення ракет.....	16
2.3 Генерація та відстеження траєкторії	18
2.4 Наведення по прямій видимості	20
2.4.1 Виведення закону істинно пропорційного навігаційного управління...	20
2.4.2 Істинно пропорційна навігація.....	22
2.4.3 Пропорційна навігація з контролем курсу.....	24
2.4.4 Гібридне керування курсом TPN	25
2.5 Проходження по траєкторії	27
2.5.1 Дотримання траєкторії прискорення LOS	27
2.5.2 Прогнозування руху цілі.....	28
3 Реалізація системи	30
3.1 Вибір середовища симуляції	30
3.2 Структура системи	33
3.3 Результати моделювання алгоритмів	41
3.4 Тривалість переслідування.....	56

3.5 Поведінка при переслідуванні	60
4 Розробка апаратного забезпечення	64
4.1 Алгоритмічні передумови для вибору елементів	64
4.2 Порядок роботи з апаратним забезпеченням	66
4.3 Вибір апаратного забезпечення	69
4.4 Відпрацювання роботи апаратної частини	77
Висновки.....	81
Перелік джерел посилання	82
Додаток А – Презентація	85

ВСТУП

Актуальною проблемою сьогодення є системи безпеки. Комп'ютерний зір відіграє важливу роль у системі безпеки. У комп'ютерному зорі аналіз відео в реальному часі за допомогою камери має особливе значення. Виявлення, відстеження та блокування рухомих цілей є важливим застосуванням системи обробки та контролю відео. Він має велике значення у військових, спортивних, транспортних і багатьох інших сферах застосування. Виявлення рухомих цілей і супровід вручну за допомогою платформи камери вимагає постійної уваги людей, робить завдання стеження складним, трудомістким і помилковим.

У міру бурхливого розвитку ринку комерційно доступних безпілотних літальних апаратів (БПЛА) у захищеному чи вразливому повітряному просторі з'являється все більше невеликих, телекерованих чи автономних літальних апаратів. Існуючі рішення щодо видалення цих літаків або є військовими і занадто руйнівними для внутрішнього використання, або включають громіздкі телекеровані засоби боротьби з БПЛА, які довели свою неефективність у гучних внутрішніх справах.

У цій роботі пропонується дослідити використання квадрокоптера для автономного наведення на напівстаціонарні та рухомі повітряні об'єкти з невеликими попередніми знаннями характеристик польоту цілі або взагалі без них. Команди наведення та управління генеруються на основі інформації, що надходить лише від бортової монокулярної камери. Джерелами дослідження є відкрита література з наведення ракет з розробкою оптимального методу наведення, реалізованого на квадрокоптері, але непридатного для використання ракетами.

У роботі описано повний інтерфейс моделювання та апаратну платформу, яка спочатку виявляє рухомий об'єкт, відстежує його та блокує. Для виявлення та відстеження використовуються два алгоритми, тобто алгоритм розрізнення кадрів і алгоритм середнього зсуву, а блокування досягається за допомогою послідовного зв'язку між matlab і proteus. У апаратній реалізації відео в реальному часі

записується за допомогою камери, рухомі об'єкти виявляються та відстежуються за допомогою комп'ютерного процесора. Повна система тестується на записаних відео та відео в реальному часі. Результати показують, що система має хорошу ефективність для об'єктів, які рухаються.

Наведено результати для успіху влучення з першого проходу та тривалості переслідування різними методами.

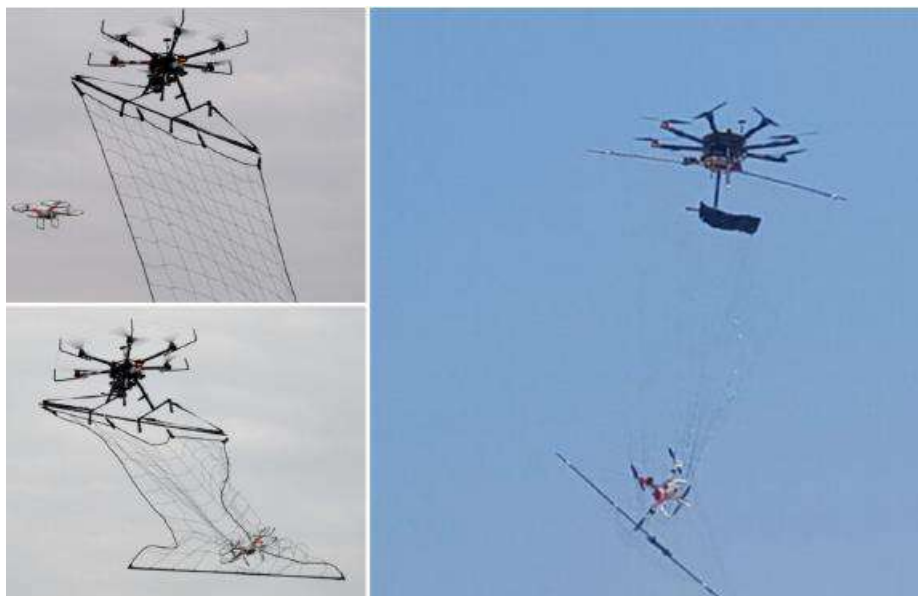
1 ОГЛЯД ОБЛАСТІ РОЗРОБКИ ТА ПОСТАНОВКА ЗАДАЧ

1.1 Протидія насанкціонованим польотам БПЛА

Мікроповітряні літальні апарати (MAV), також звані дронами, безпілотні літальні апарати (БПЛА) та невеликі безпілотні літальні системи (sUAS) отримали величезне зростання в різних секторах ринку по всьому світу [5]. У корпоративному та державному секторах зазвичай діють суворі правила експлуатації MAV, проте експлуатація цих літаків у споживчому секторі слабо регулюється та існує сильний опір співтовариства проти будь-якого такого законодавства. Суворіший нагляд за використанням приватних дронів мотивується численними інцидентами за участю невеликих дронів, зазвичай керованих дистанційно, в громадських місцях [6]. Занепокоєння щодо можливості використання БПЛА над натовпом на зимових Олімпійських іграх 2018 року спонукало південнокорейську владу тренуватися щодо відключення дронів, включаючи розробку телекерованої платформи, яка виводить з ладу інші за допомогою скинутої сітки (рис. 1.1а) [7]. Крім цих прикладів в Інтернеті є безліч відеороликів про те, як користувачі рекреаційних дронів втрачають контроль над своїм БПЛА через погодні умови, втрати сигналу розташування від GPS або низького заряду батареї.

Хоча ці проблеми можна пом'якшити шляхом введення суворих правил та нагляду на ринку споживчих дронів, це також може різко обмежити незалежність аматорів та дослідників. Потенційною альтернативою може стати захоплення чи виведення з ладу несанкціонованих БПЛА без порушення роботи. Сучасні технології боротьби з БПЛА існують здебільшого у військовому секторі у вигляді глушників (що використовуються для порушення сигналу телеуправління від найближчого радіоконтролера) або ракет типу «Стінгер» (призначених для виведення з ладу БПЛА шляхом удару). Жоден із цих варіантів не підходить для цивільного використання, де проблемою є як шум, так і сміття не кажучи вже

провартість цих рішень. Тому потрібна розробка рішення, яке зведе до мінімуму руйнування і буде досить гнучким, щоб захопити широкий спектр БПЛА в цивільному повітряному просторі.



(а) Південнокорейська поліцейська протидія БПЛА (система, оснащена сітками) [7]



(б) Застосуванням «тарана» для виведення цілі з ладу [8]

Рисунок 1.1 – Поточні можливі системи протидії БПЛА для цивільного використання

Квадрокоптери мають високу маневреність з 4 ступенями свободи (DOF – degrees of freedom) і можуть розганятися до високих швидкостей до цілі з невідомим

планом польоту чи характеристиками. Крім того, нещодавні дослідження в галузі повітряної робототехніки показали, що комплекс засобів запобігання, виявлення, планування та управління перешкодами може повністю працювати на борту автономної квадрокоптерної платформи. Загальні недоліки квадрокоптерів включають низький час автономної роботи, але при місії перехоплення польоти короткі, але з високими прискореннями (і, отже, з більш високою енергоємністю).

1.2 Проблеми реалізації наведення квадрокоптерів на рухомі повітряні цілі

Проблеми автономного наведення на невідомий невеликий літальний апарат за допомогою квадрокоптера численні, включаючи швидкий політ через потенційно загрозоване середовище, і навіть розробку механічно надійного методу захоплення цілі без ушкодження будь-якого агента. Однак основне завдання, що розглядається в дипломній роботі, пов'язане з наведенням на ціль та управлінням квадрокоптерним БПЛА при русі до цілі.

Виявлення рухомих об'єктів є складним завданням, а виявлення в реальному часі в сценаріях реального часу ускладнює це завдання. Для виявлення рухомих об'єктів використовуються різні методи. Як згадувалося вище, метод оптичного потоку, розрізнення кадрів, віднімання фону і стільникові нейронні мережі та багато інших алгоритмів використовуються для виявлення рухомого об'єкта у відео в реальному часі.

Після успішного виявлення рухомого об'єкта відстеження рухомого об'єкта досягається шляхом знаходження центральних координат виявленого об'єкта. Блокування виявленого і відстежуваного об'єкта - складне завдання. Для блокування виявленого об'єкта використовуються різні методи. Центральні координати, знайдені при стеженні, використовуються для знаходження кутів відхилення

рухомого об'єкта від контрольної точки. Ці кути передаються на плату мікроконтролера через послідовний зв'язок, який, у свою чергу, рухає двигуни.

1.3 Постановка задач

Враховуючи розглянуту проблематику у поточній роботі пропонується розглянути та модифікувати численні методи управління і наведення, запозичені з різних галузей літератури, для реалізації в сценарії ураження БПЛА.

Ми розділили наш проект на дві частини:

- розробка програмного забезпечення;
- розробка базового апаратного забезпечення.

Для спрощення математичного розгляду задачі чинники довіклля виключаються, щоб зосередитись на оцінці кількох методів наведення. Крім того пропонується оцінити керування польотом квадрокоптера-перехоплювача в умовах прямої видимості цілі (LOS – line-of-sight) в умовах польових змагань БПЛА щодо виявлення і наведення на об'єкт заздалегідь відомої форми та кольору.

Для вирішення такої задачі її слід розбити на більш прості підзадачі:

- проведення аналізу математичного апарату відстеження рухомих повітряних цілей;
- розробка методів та алгоритмів, що можуть застосовуватись для виявлення та наведення БПЛА на рухомі повітряні об'єкти;
- планування регульованого та надійного шляху навколо фіксованої території (арени) на основі координат отриманих від GPS;
- контроль над напівстаціонарними цілями, розташованими по всій фіксованій території (арені);
- виявлення невеликої жовтої кульки, що рухається з відомою швидкістю 8 м/с на захаращеному фоні;
- симуляція роботи розроблених алгоритмів;

- оцінка ефективності роботи запропонованих алгоритмів;
- вибір апаратних компонентів для реалізації системи.

Залежно від розміру та форми цілі (наприклад, нерухоме крило, мультиротор, вертоліт, дирижабль), а також відстані до неї не можна припускати, що 3D-датчик, такий як лідар або стереоскопічний зір, може використовуватися для точної локалізації цілі у просторі. Наприклад, через їх рідкісні, а іноді й сітчасті рами, мультикоптери особливо складно локалізувати за допомогою дешевих скануючих лідарів або стереокамер на великій відстані. Таким чином, основна увага в роботі приділяється використанню монокулярного зору та адаптації методів наведення для використання лише приблизних оцінок глибини, коли це необхідно.

Основні положення цієї роботи полягають у наступному:

- оцінка різних методів наведення та управління та того, як їх можна адаптувати для використання на квадрокоптері в моделюваному середовищі;
- розробка програмного комплексу з використанням LOS-наведення для пошуку та наведення напівстаціонарних та рухомих цілей у межах фіксованої території (арени);
- розробка апаратного забезпечення на основі відеокамери для реалізації запропонованих алгоритмів.

Першою задачею є проведення огляду робіт у різних галузях, включаючи класичний візуальний супровід, наведення ракет, а також формування та відстеження траєкторій.

2 МАТЕМАТИЧНІ ТА АЛГОРИТМІЧНІ ЗАСАДИ НАВЕДЕННЯ НА РУХОМИЙ ОБ'ЄКТ

2.1 Класичне візуальне захоплення цілі

Візуальне сортирування об'єктів охоплює широкий спектр досліджень, спрямованих на керування ланками робота щодо вхідних даних від візуальних датчиків. Найбільш поширене застосування візуального сервоприводу - операції із захоплення та розміщення, що виконуються за допомогою роботизованих маніпуляторів, оснащених камерами. Ці роботи зазвичай мають установку «око в руці» (управління зі зворотним зв'язком) або «око в руці» (управління із розімкненим контуром) [9]. Двома найбільш поширеними підходами у цій галузі є візуальне відстеження на основі зображень (IBVS) та візуальне відстеження на основі пози (PBVS), причому різниця між ними полягає в оцінці відносної пози мети по відношенню до роботи [10].

IBVS, як описано Хатченсоном, корисний тільки в невеликій області простору завдань, якщо якобіан зображення не обчислюється онлайн зі знанням відстані до мети, що ще більше ускладнюється системою, заснованої на монокулярному зорі. Якщо швидкість цілі не є постійною, помилки або запізнення в площині зображення цілі, що рухається, призводять до помилок в управлінні будь-яким методом, які, у свою чергу, повинні бути відрегульовані за допомогою більш складної системи управління. Чауміт та Сантос [11] відстежували рухому ціль з допомогою IBVS, але передбачали постійне прискорення. Коли ціль різко маневрувала, провіснику руху цілі на основі фільтра Калмана потрібно кілька циклів зворотного зв'язку, щоб перекалібруватися під новий рух. У [12] основним недоліком PBVS є необхідність тривимірної інформації про мету, зокрема про глибину, яка може бути недоступна.

2.2 Математичне забезпечення наведення ракет

Для розуміння математичних моделей та алгоритмів наведення на рухомі цілі було розглянуто наведення ракет як класичний засіб ураження повітряних рухомих цілей.

Самонавідні авіаційні ракети орієнтовані виключно на забезпечення ураження повітряної цілі. Принаймні з 1956 пропорційна навігація в тій чи іншій формі є стандартом у методах наведення ракет [1]. Адлер описує кілька таких методів, включаючи навігацію з постійним пеленгом та пропорційну навігацію.

Навігація з постійним пеленгом, при якій для підтримки постійного пеленгу на ціль застосовується прискорення, вимагає миттєвої корекції відхилень у напрямі прямої видимості (LOS). Це робить його несумісним з динамікою ракет, які можуть безпосередньо виконувати команди бічного прискорення (аналогічно літакам), тому Адлер пропонує використовувати 3D пропорційну навігацію (ПН), яка застосовує швидкість повороту, пропорційну до зміни напрямку прямої видимості. У пізніших роботах термін «пропорційна навігація» використовується як взаємозамінний між цими двома схемами, і навіть поширюється інші подібні методи.

У дипломній роботі ПН пропонується використовувати як загальний термін для позначення будь-якого закону управління, який використовує швидкість обертання LOS для створення команди прискорення. Як зазначено в [4], ПН, якщо припустити відсутність затримки автопілота, є оптимальним законом керування, який дуже мало передбачає характеристики прискорення цілі. Однак були розроблені варіанти класичної ПН, які адаптуються до різних профілів польоту, включаючи постійне прискорення і постійний ривок [13]. ПН зазвичай поділяють на дві категорії: «істинний» варіант та «чистий» варіант [2]. Хоча назва багато в чому довільна, основна відмінність полягає в системі відліку, в якій до ракети, що переслідує, застосовується бічне прискорення. Істинна ПН застосовує це прискорення, ортогональне поточній швидкості ракети. Чиста ПН застосовує ортогональне прискорення.

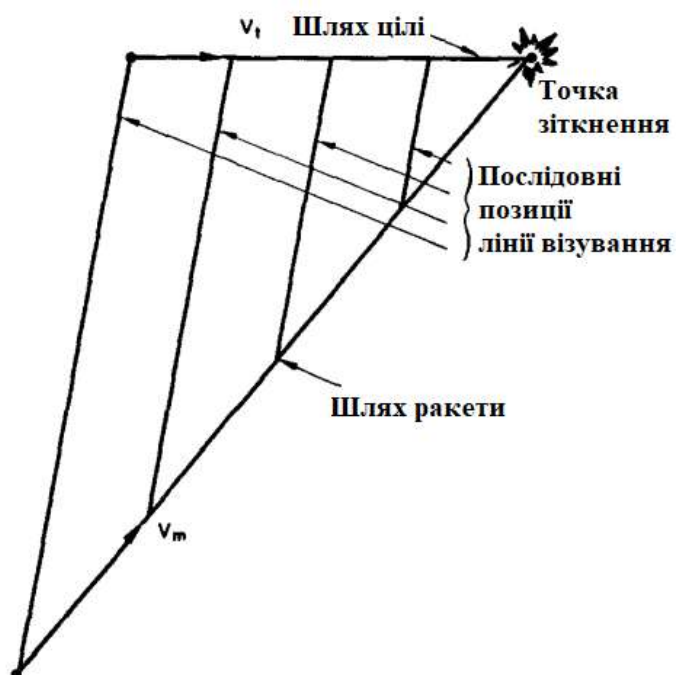
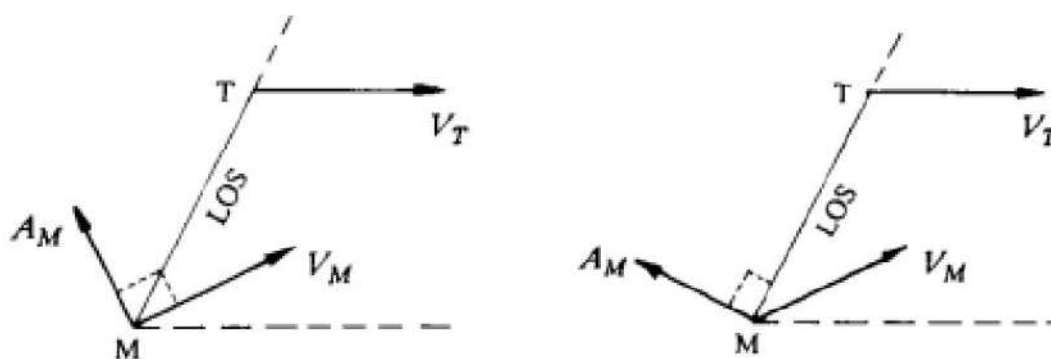
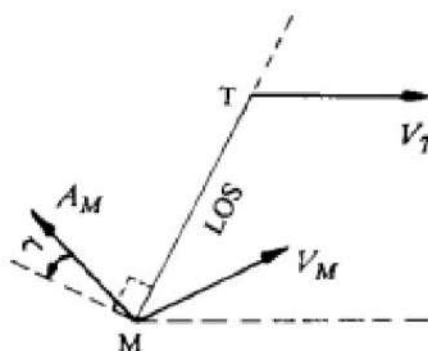


Рисунок 2.1 – Діаграма курсу зіткнення з постійним пеленгом та миттєвим бічним прискоренням, доданим до ракети [1]



Чиста пропорційна навігація Справжня пропорційна навігація



Узагальнена справжня пропорційна навігація

Рисунок 2.2 – Порівняння чистої та справжньої пропорційної навігації. A_M і V_M відносяться до бажаного прискорення та поточної швидкості ракети відповідно [2]

2.3 Генерація та відстеження траєкторії

Траєкторії надають роботам гладкі чи кінодинамічно можливі шляхи проходження через простір станів. Це суперечить відправленню необроблених диференціальних команд, які можуть вийти за межі обмежень робота та призвести до нестабільності контролера або навіть до фізичного виходу з приводу робота. Таким чином, була проведена велика робота зі створення та відстеження траєкторій для використання з різними типами роботів та додатків, в першу чергу з роботами-маніпуляторами для захоплення та безпілотних транспортних засобів [14-17]. Це було поширене на MAV, щоб забезпечити плавний та ефективний політ. Ріхтер [18] показав, що поліноміальні траєкторії усувають необхідність у широкому пошуку на основі вибірки у просторі станів транспортного засобу. Такий підхід, хоч і не забезпечував асимптотичної збіжності до глобального оптимального шляху, забезпечував плавний та швидкий політ квадрокоптера. [3] було показано, що при безперервних траєкторіях з мінімальним ривком і швидкому переплануванні вони досягають більш високої плавності траєкторії в порівнянні з іншими, більш реактивними планувальниками. На рис. 2.3 показано плавну траєкторію, створену шляхом відстеження шляхів, згенерованих примітивами руху. Гао [19] спочатку знаходить індексований за часом мінімальний шлях прибуття, який може виявитися неможливим для польоту квадрокоптера, а потім формує коридор польоту у вільному просторі, в якому вони генерують допустиму траєкторію. Вони використовують поліноміальний базис Бернштейна і представляють траєкторію у вигляді шматкової кривої Безьє.

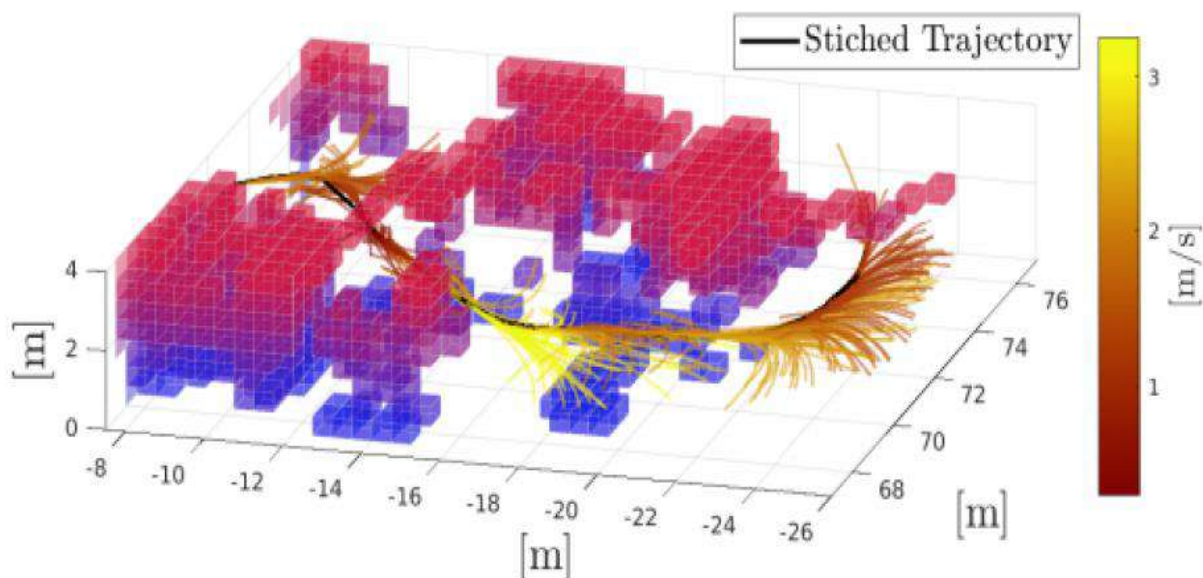


Рисунок 2.3 – Приклад складених траєкторій у захищеному середовищі. Кольорові лінії становлять окремі примітиви руху; Квадрокоптер відстежує початкову частину кожної згенерованої траєкторії, утворюючи повну зшити траєкторію, представлену чорною лінією [3]

Щоб йти по траєкторіях, контролери вибирають бажану траєкторію, що зазвичай складається з колійних точок, кожна з яких має відповідну швидкість, і видають команди приводу роботу. У разі MAV програмне забезпечення автопілота може приймати команди орієнтації або швидкості орієнтації, які від такого контролера. Хоффман [20] продемонстрував контролер, який приймав на вході неможливу траєкторію та видавав можливі команди орієнтації для квадрокоптера, який точно дотримувався вихідного шляху. Це було продемонстровано на відкритому повітрі з точністю до 50 см. Аналогічний підхід був використаний (але розширений до 3D) для контролера прямування шляхом, реалізованого в [21]. Тут бажане положення та бажана швидкість з найближчої точки траєкторії використовуються для визначення помилок положення та швидкості робота. Вони використовуються для розрахунку бажаного прискорення робота за допомогою зворотного зв'язку PD та фактора прямого зв'язку.

2.4 Наведення по прямій видимості

2.4.1 Виведення закону істинно пропорційного навігаційного управління

Методи наведення прямої видимості (LOS) використовуються для подачі переслідувачу команд прискорення, які мінімізують зміну вектора прямої видимості в напрямку цілі. У цьому розділі представлена базова геометрія LOS, яка використовується для наведення пропорційної навігації (ПН). У наступних підрозділах показано, як це використовується при виявленні цілей для розрахунку величин, що використовуються для застосування наведення ПН.

Як видно на рис. 2.4, фіксована світова система координат визначається одиничними векторами $\bar{1}_x, \bar{1}_y, \bar{1}_z$. Система координат LOS, прикріплена до ракети, що рухається, задається одиничними векторами $\bar{1}_r, \bar{1}_n, \bar{1}_\omega$; $\bar{1}_r$ точок уздовж LOS $\bar{r}$; $\bar{1}_n$ - зміна напрямку (тобто обертання) вектора LOS; $\bar{1}_\omega$ — векторне твір перших двох у вказаному порядку (що утворює праву систему координат).

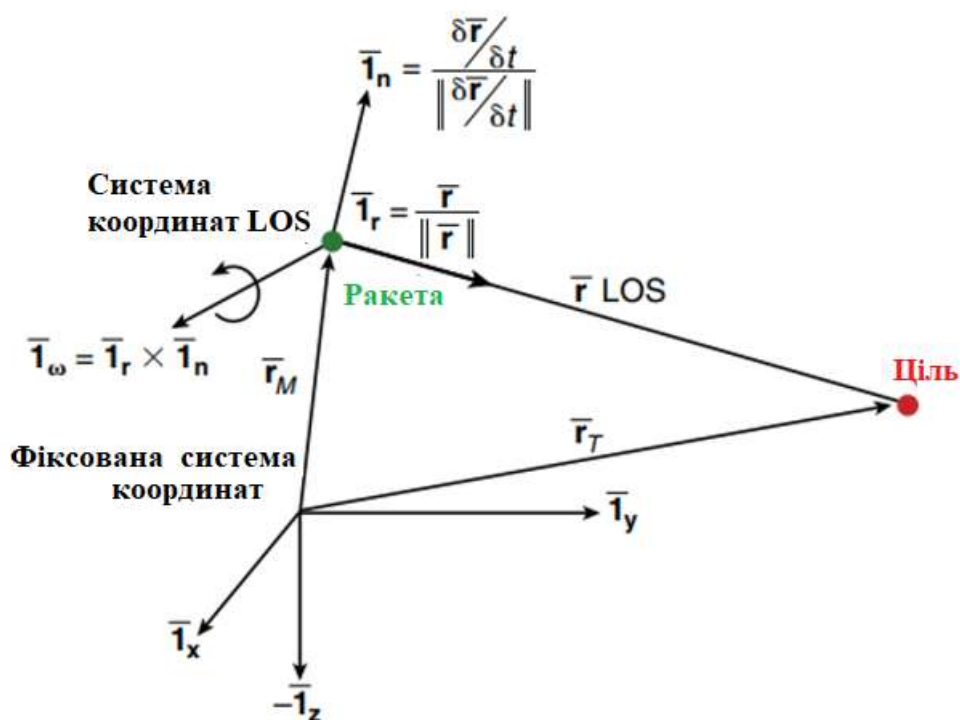


Рисунок 2.4 – Система координат LOS [4]

Загалом, кутова швидкість системи координат LOS визначається виразом:

$$\bar{\dot{\phi}} = \dot{\phi}_r \bar{1}_r + \dot{\phi}_n \bar{1}_n + \dot{\phi}_\omega \bar{1}_\omega \quad (2.1)$$

де $\dot{\phi}_r, \dot{\phi}_n, \dot{\phi}_\omega$ – величини складових кутової швидкості, визначені як:

$$\dot{\phi}_r = \bar{\dot{\phi}} \cdot \bar{1}_r \quad (2.2)$$

$$\dot{\phi}_n = \bar{\dot{\phi}} \cdot \bar{1}_n \quad (2.3)$$

$$\dot{\phi}_\omega = \bar{\dot{\phi}} \cdot \bar{1}_\omega \quad (2.4)$$

Складовими відносного прискорення між ракетою і метою, отриманими [4], але не відтвореними тут, є:

$$(\bar{a}_T - \bar{a}_M) \cdot \bar{1}_r = \ddot{R} - R \dot{\phi}_\omega^2 \quad (2.5)$$

$$(\bar{a}_T - \bar{a}_M) \cdot \bar{1}_n = 2\dot{R}\dot{\phi}_\omega + R\ddot{\phi}_\omega \quad (2.6)$$

$$(\bar{a}_T - \bar{a}_M) \cdot \bar{1}_\omega = R\dot{\phi}_\omega \dot{\phi}_r \quad (2.7)$$

де $\bar{a}_T$ та $\bar{a}_M$ — прискорення мети та ракети відповідно.

З цього результату, безпосередньо використовуючи умову в рівнянні 2.5, ми можемо перерахувати достатні умови для задоволення рівняння та забезпечення перехоплення: (i) перехоплювач здатний досягти прискорення вздовж прямої видимості, більшої, ніж прискорення цілі $((\bar{a}_T - \bar{a}_M) \cdot \bar{1}_r < 0)$, (ii) початкова швидкість зміни в діапазоні R негативна $(\dot{R} < 0)$, що потім забезпечує $\ddot{R} < 0$ за першої умови, та (iii) швидкість зміни LOS дорівнює 0 $(\dot{\phi}_\omega = 0)$. Умова (i) залежить від характеру перехоплювача та мети; умова (ii) передбачає, що переслідування ПН має бути ініціалізоване з позитивною швидкістю зближення; умова (iii) передбачає, що перехоплювач повинен виконувати команди прискорення те щоб вектор LOS залишався постійним. Палумбо та ін. (2010) знаходить наступний істинний закон ПН (TPN), який забезпечує стабільність системи:

$$\bar{a}_M \cdot \bar{1}_n - NV_c \dot{\phi}_\omega, N > 2 \quad (2.8)$$

де N – пропорційний коефіцієнт посилення;

V_c – швидкість закриття.

Іншими словами, прискорення перехоплювача a_M повинно мати складову, ортогональну LOS, пропорційну швидкості обертання LOS, як зазначено у рівнянні 2.11.

2.4.2 Істинно пропорційна навігація

Щоб згенерувати бажаний вектор прискорення з величиною, заданою рівнянням 2.8, і напрямом, ортогональним LOS, ми спочатку обчислюємо як Φ_w (безпосередньо представлене у рівнянні 3.8), так і $1n$ (напрямок прискорення). Центроїд мети в координатах кадру зображення в моменти часу $t - 1$ і t представлений як (u_{t-1}, v_{t-1}) і (u_t, v_t) відповідно, як показано на рис. 2.5. Основна точка камери, зазначена у внутрішній матриці каліброваної камери (зазвичай позначається K), позначається (c_x, c_y) .

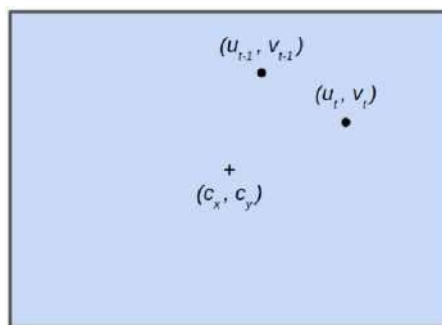


Рисунок 2.5 – Приклад кадру зображення з виявленим центроїдом об'єкта в моменти часу $t - 1$ та t

Вектор LOS rt в системі відліку камери на момент часу t визначається наступним чином.

$$r_t = \begin{bmatrix} \frac{U_t - C_x}{f_x} \\ \frac{U_t - C_y}{f_y} \\ 1 \end{bmatrix} \quad (2.9)$$

В окремому випадку першої ітерації алгоритму, при $t = 0$, вектор LOS розраховується відповідно до рівняння 2.9, а потім зберігається для використання як r_{t-1} на наступній ітерації. Для циклу обчислень $t = 0$ керуючий вихід встановлюється 0.

Кут, охоплений двома векторами r_t і r_{t-1} , виглядає так:

$$\phi_\omega = \arccos \frac{\langle r_t, r_{t-1} \rangle}{\|r_t\| \|r_{t-1}\|} \quad (2.10)$$

Отже, якщо різниця в часі одного циклу представлена Δt , то величина швидкості обертання вектора LOS дорівнює:

$$\dot{\phi}_\omega = \frac{\phi_\omega}{\Delta t} \quad (2.11)$$

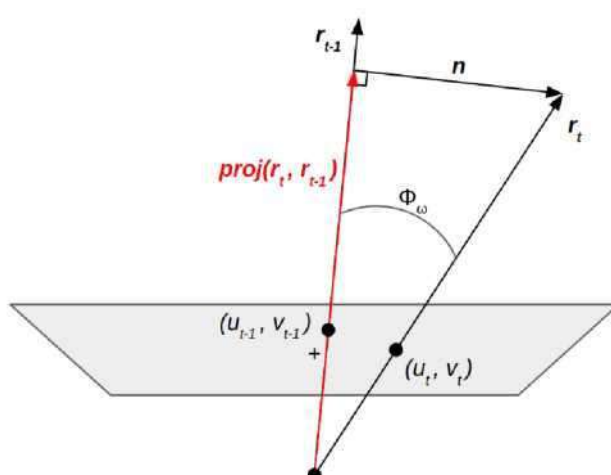


Рисунок 2.6 – Діаграма згори донизу, що показує проміжні величини при розрахунку бажаної команди прискорення

Напрямок прискорення (напрямок обертання LOS) 1_n показано на рис. 2.6 і розраховано нижче.

$$n-r_t = \text{proj}(r_t, r_{t-1}) \quad (2.12)$$

$$= r_t - \langle r_t, \frac{r_{t-1}}{\|r_{t-1}\|} \rangle \frac{r_{t-1}}{\|r_{t-1}\|} \quad (2.13)$$

$$1_n = \frac{n}{\|n\|} \quad (2.14)$$

де векторна проекція r_t на r_{t-1} представлена червоним вектором на рис.2.6.

Використовуючи скаляр Φ_ω та вектор 1_n , ми можемо обчислити бажане прискорення наступним чином:

$$A_{LOS} = NV_c \dot{\Phi}_\omega 1_n \quad (2.15)$$

Стосовно квадрокоптера в цій роботі вектор прискорення подається в контролер швидкості шляхом інтеграції команди, яка передає команди крену, тангажу, відхилення від курсу та тяги внутрішнім контролерам автопілота. Таким чином, замість того, щоб коригувати курс задоволення бічних прискорень, застосування TPN в цій роботі засноване на контролі кута крену. Цей прямий метод досягнення бічного прискорення (що вимагає швидкості руху вперед) неможливий на ракетах чи літаках з нерухомим крилом.

2.4.3 Пропорційна навігація з контролем курсу

Алгоритм TPN, наведений вище, зберігає цілісність алгоритму, зазвичай представленого в літературі з наведення ракет, але застосовує команду управління більш безпосередньо, керуючи кутом нахилу БПЛА. Під час польоту ракети

транспортний засіб виконує задані команди прискорення, літаючи дугою, поступово змінюючи свій курс, покладаючись на рух уперед і використовуючи вектор тяги або поверхні управління. Однак квадрокоптер має прямий контроль над своїм курсом, застосовуючи керування швидкістю відхилення від курсу. У цьому розділі ми описуємо алгоритм, який використовує прискорення ПН у всіх осях, крім бічної осі, і натомість керує курсом досягнення поперечного прискорення. Оскільки він не використовує ПН на бічній осі, ми не присвоюємо мітку "істинно".

Ми визначаємо інерційну систему відліку в центрі БПЛА, де x вказує вперед, y вказує вліво, а z вказує вгору. Прискорення по осях x та z просто береться з рівняння 2.15 як відповідні компоненти:

$$a_x = a_{LOS} \cdot 1_x \quad (2.16)$$

$$a_z = a_{LOS} \cdot 1_z \quad (2.17)$$

Управління курсом складається із заданої швидкості нишпорення, яка включає в себе обчислення курсу:

$$y\dot{a}w = K_{p,yaw} \text{heading} \quad (2.18)$$

$$= K_{p,yaw} \arctan \frac{r_t \cdot 1_y}{r_t \cdot 1_x} \quad (2.19)$$

де $K_{p,yaw}$ – налаштований пропорційний коефіцієнт посилення, а r_t – поточний вектор LOS.

2.4.4 Гібридне керування курсом TPN

Обидва методи, подані в розділах 2.4.2 та 2.4.3, мають потенційні переваги. TPN, спеціально застосований до квадрокоптерів за допомогою управління кутом крену, може забезпечити більш швидкий час реакції на об'єкт, що рухається. ПН,

зберігаючи ціль у центрі кадру, гарантує, що ціль не загубиться з кадру; інакше у повноцінній системі переслідуючий БПЛА повинен буде повернутися до схеми пошуку. Мета гібридного алгоритму – скористатися перевагами обох методів.

Цей метод перемикається між двома режимами: ПН та Heading. Перехід між ними залежить від налаштованого порога k_{heading} , що визначає напрямок до мети.

Якщо $|\text{heading}| < k_{\text{heading}}$, вводимо ПН стан:

$$a_x = a_{\text{LOS}} \cdot 1_x \quad (2.20)$$

$$a_y = a_{\text{LOS}} \cdot 1_y \quad (2.21)$$

$$a_z = a_{\text{LOS}} \cdot 1_z \quad (2.22)$$

$$y\dot{a}w = 0.2K_{p,yaw}\text{heading} \quad (2.23)$$

Якщо $|\text{heading}| \geq k_{\text{heading}}$, входимо у стан «Контроль курсу»:

$$a_x = a_{\text{LOS}} \cdot 1_x \quad (2.24)$$

$$a_y = 0.2 a_{\text{LOS}} \cdot 1_y \quad (2.25)$$

$$a_z = 0 \quad (2.26)$$

$$y\dot{a}w = K_{p,yaw}\text{heading} \quad (2.27)$$

При цьому курс k може бути налаштований залежно від певних факторів системи БПЛА, включаючи поле зору камери (FOV), або максимальну швидкість відхилення від курсу. У будь-який час, незалежно від заголовка, і a_u , у $u\dot{w}$ не дорівнюють нулю і замість цього пригнічуються з коефіцієнтом менше 1. Емпірично було виявлено, що коефіцієнт 0,2 працює добре і дає відповідний вплив на обидва прискорення та швидкість нищпорення. Використовуючи цей гібридний метод, БПЛА потенційно зможе реагувати на зміни руху цілі, зберігаючи при цьому її в полі зору.

2.5 Проходження по траєкторії

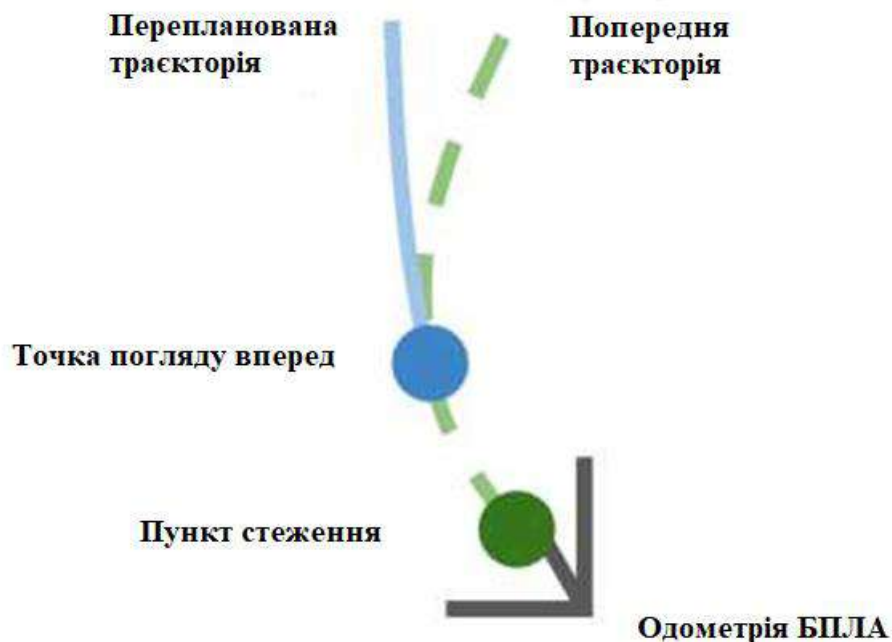


Рисунок 2.7 – Схема перепланування траєкторії шляхом зшивання траєкторії із прогнозованою точкою. БПЛА уважно слідує за точкою стеження

Усі методи відстеження траєкторії було реалізовано з деякою частотою перепланування, з якою публікуються оновлені траєкторії. Перепланування постійно виконується з погляду прогнозу, яка підтримується контролером траєкторії.

2.5.1 Дотримання траєкторії прискорення LOS

Ці траєкторії формуються шляхом прийняття бажаної команди прискорення, розрахованої за рівнянням 2.15, та розрахунку дорожніх точок положення та швидкостей з урахуванням початкового положення, встановленого для точки

перегляду вперед. Розрахунки положень та швидкостей виконуються за допомогою наступних кінематичних рівнянь, заданих в інерційній системі відліку БПЛА.

$$p_t = v_0 t + \frac{1}{2} a_{LOS} t^2 \quad (2.28)$$

$$v_t = v_0 + a_{LOS} t, t = 0, 0.1, \dots, T \quad (2.29)$$

де T – тривалість кожної траєкторії. Коли T наближається до 0 цей метод стає еквівалентним безпосередньому завданню бажаного прискорення $LOS1$. Дискретизація тимчасових кроків також є параметром, що настраюється.

2.5.2 Прогнозування руху цілі

У своїй найпростішій формі прогнозування руху цілі включає оцінку швидкості цілі в 3D, розрахунок майбутнього розташування мети по прямолінійному шляху і створення траєкторії курсу зіткнення до цієї точки простору зі швидкістю, яка завершує траєкторію. у вказаний час. Цей метод робить три важливі припущення: (i) ціль має нульове прискорення, (ii) ми можемо приблизно оцінити час до зіткнення, розрахувавши час вздовж поточного вектора прямої видимості, і (iii) прогнозоване положення цілі змінюватиметься повільно, тому створення прямих траєкторій достатньо, щоб отримати остаточну, гладку зшити траєкторію. Спочатку обчислюються одиничні вектори LOS вдвічі. Вони використовуються разом з оцінкою глибини для визначення швидкості цілі:

$$V_{\text{target}} = \frac{d_1(1_{LOS1}) - d_0(1_{LOS0})}{t_1 - t_0} \quad (2.30)$$

де d_1 і d_0 , 1_{LOS1} і 1_{LOS0} , а також t_1 і t_0 – це передбачувана глибина, розрахований одиничний вектор LOS і час на двох тимчасових кроках.

Отримавши швидкість, ми знаходимо приблизний час до зіткнення вздовж вектора прямої видимості, використовуючи компонент швидкості БПЛА вздовж прямої лінії видимості:

$$t_{\text{collision}} = \frac{d_1}{\langle V_{UAV}, 1_{LOS1} \rangle} \quad (2.31)$$

де V_{UAV} – поточний вектор швидкості БПЛА.

Отже, приблизна точка у просторі для планування траєкторії така: $P_{\text{collision}}$ знаходиться у системі відліку БПЛА.

$$P_{\text{collision}} = V_{\text{target}}^t + d_1(1_{LOS1}) \quad (2.32)$$

3 РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Вибір середовища симуляції

Враховуючи неможливість польотного випробування системи наведення на повітряні рухомі об'єкти в рамках виконання магістерської дипломної роботи – було прийнято рішення проводити випробування розроблених методів з використанням системи симуляції.

У якості системи для симуляції польотів було обрано програмне забезпечення Gazebo. Gazebo – програмне забезпечення для дослідження сенсорних систем та робототехніки, що має відкритий код.

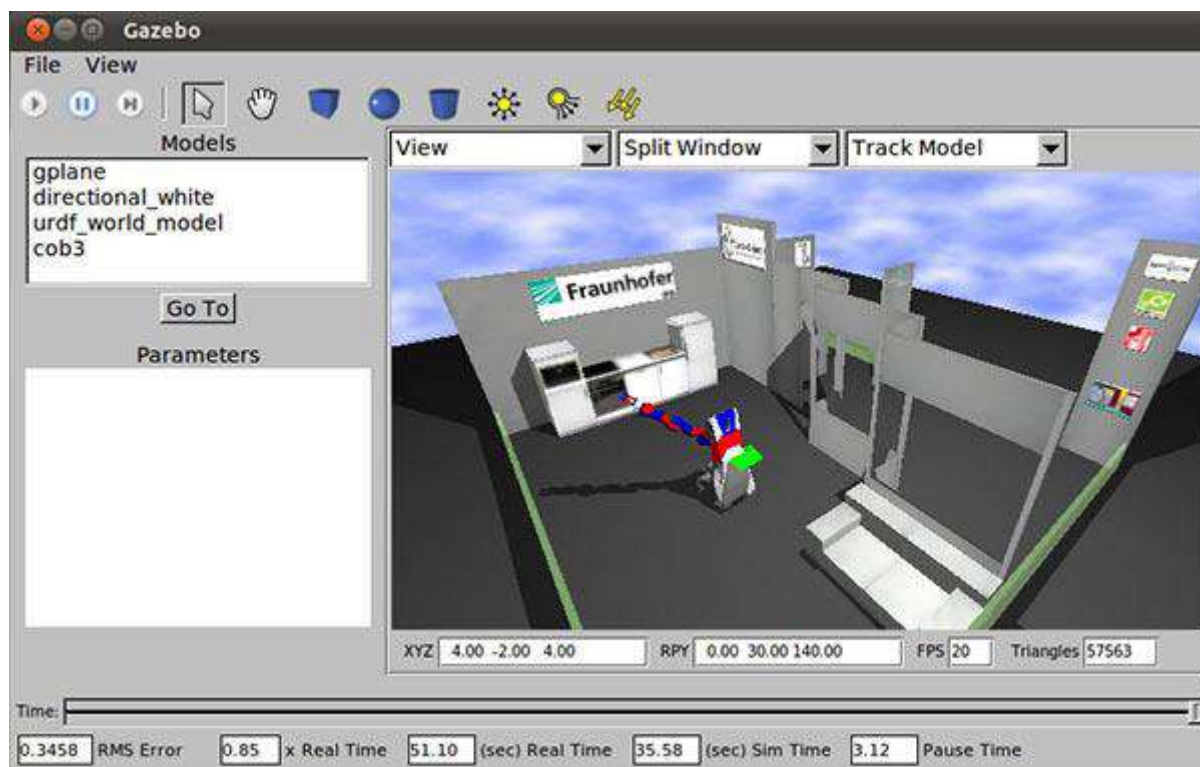


Рисунок 3.1 – Симуляція робототехнічних систем в ПЗ Gazebo

Програмний комплекс Gazebo є середовищем для симулювання роботи віртуальних роботів з різними сенсорами в оточенні всіляких об'єктів. Додаток складається з графічної частини та частини з імітування взаємодії твердих об'єктів, дозволяючи моделювати динаміку та кінематику механізмів роботів (включаючи моменти взаємодії з тілами зовнішнього середовища) та формувати фізично

правдоподібні показання віртуальних датчиків. Програма Gazebo має гнучкий дизайн та зручний інтерфейс, що підтримує одночасну роботу з кількома пристроями.

Симулятор Gazebo має власний редактор, що дозволяє без програмування створювати тривимірні сцени і включає величезну бібліотеку моделей. Програма також надає такі можливості:

- використання найпопулярніших загальновідомих моделей роботів;
- є можливість самостійного проектування необхідних пристроїв (сенсорів та роботів), завантаження їх та подальшої симуляції;
- підтримка та симулювання роботи безлічі різних сенсорів, у тому числі сонара, лазерного далекоміра, датчиків сімейства IMU, моно- та стереокамер, кінект-сенсорів, приладу для читання RFID-міток та інших.

Для створення якісної графіки до ПЗ Gazebo можна підключити об'єктно-орієнтований графічний двигун OGRE з відкритим вихідним кодом. Також симулятор здатний читати файли у форматі COLLADA, що дозволяє додавати у додаток об'єкти, створені у сторонніх редакторах 3D-моделей.

Поряд із програмою Stage (двовимірним симулятором роботів), Gazebo сумісний із додатком Player – крос-платформним програмним забезпеченням для дослідження робототехнічних систем. Це середовище забезпечує мережевий інтерфейс між керуючими програмами і доданими розробниками модулями/драйверами реального обладнання. Таким чином, написана в Gazebo і там же налагоджена програма управління може без змін бути перенесена на справжнє обладнання або на інший симулятор. Також середовище симуляції Gazebo сумісне з ROS – операційною системою для програмування роботів.

ПЗ Gazebo було створено незалежною групою розробників, які фінансуються організацією Open Source Robotics Foundation (США, штат Каліфорнія, місто Маунтін-В'ю), створеною у свою чергу з метою підтримки розвитку програмного забезпечення в галузі робототехніки. В даний час проект Gazebo активно розвивається – одна за одною виходять нові версії симулятора, а велика спільнота користувачів надає власні бібліотеки та інструменти, сенсори та плагіни. Ця

програма використовується в конкурсах DARPA Robotics Challenge – щорічному змаганні роботизованих механізмів, що проводиться агентством передових досліджень при Міністерстві Оборони США та покликаному допомагати рятувальникам у ході ліквідації наслідків техногенних аварій та стихійних лих.

Програмне забезпечення Gazebo є відкритим та абсолютно безкоштовно для використання. На сайті є безліч підручників (розділених за рівнем підготовки користувачів), бібліотеки моделей, що завантажуються, а також блог. Сама програма містить безліч навчальних прикладів і хорошу документацію.

Програмний комплекс Gazebo підтримується Unix-подібними операційними системами. Оскільки програма дозволяє моделювати в 3D, вона вимоглива до ресурсів комп'ютера та моделі відеокарти.

Основною функціональністю Gazebo Sim є зв'язок клієнт-сервер і різні плагіни, які він завантажує для полегшення моделювання. Плагіни — це бібліотеки, які завантажуються під час виконання. Вони дозволяють Gazebo Sim використовувати інші бібліотеки Gazebo. У структурі Gazebo є багато типів плагінів. Наприклад, є плагіни, які вводять нові фізичні механізми або механізми візуалізації. Однак у цьому документі будь-яка згадка плагінів стосується лише плагінів сервера Gazebo Sim і GUI.

Оскільки вони завантажуються під час виконання, Gazebo не потрібно перекомпілювати, щоб додати або видалити плагіни. Gazebo Sim поставляється з багатьма плагінами за замовчуванням (плагіни сервера, плагіни Gazebo GUI, плагіни Gazebo Sim GUI тощо), усі вони є необов'язковими та можуть бути видалені користувачем. Користувачі також можуть додавати більше плагінів і навіть писати власні плагіни, які будуть скомпільовані у файли бібліотеки.

Бібліотеки для Gazebo модульні. Модулі дозволяють Gazebo Sim використовувати інші бібліотеки. Наприклад, Gazebo Physics і Gazebo Sim незалежні один від одного, тому Gazebo Sim має фізичний плагін, який використовує Gazebo Physics як бібліотеку та включає в себе особливості Gazebo, що дозволяє Physics бути системою, що працює в циклі моделювання.

Процес моделювання зображено на рис.3.2.

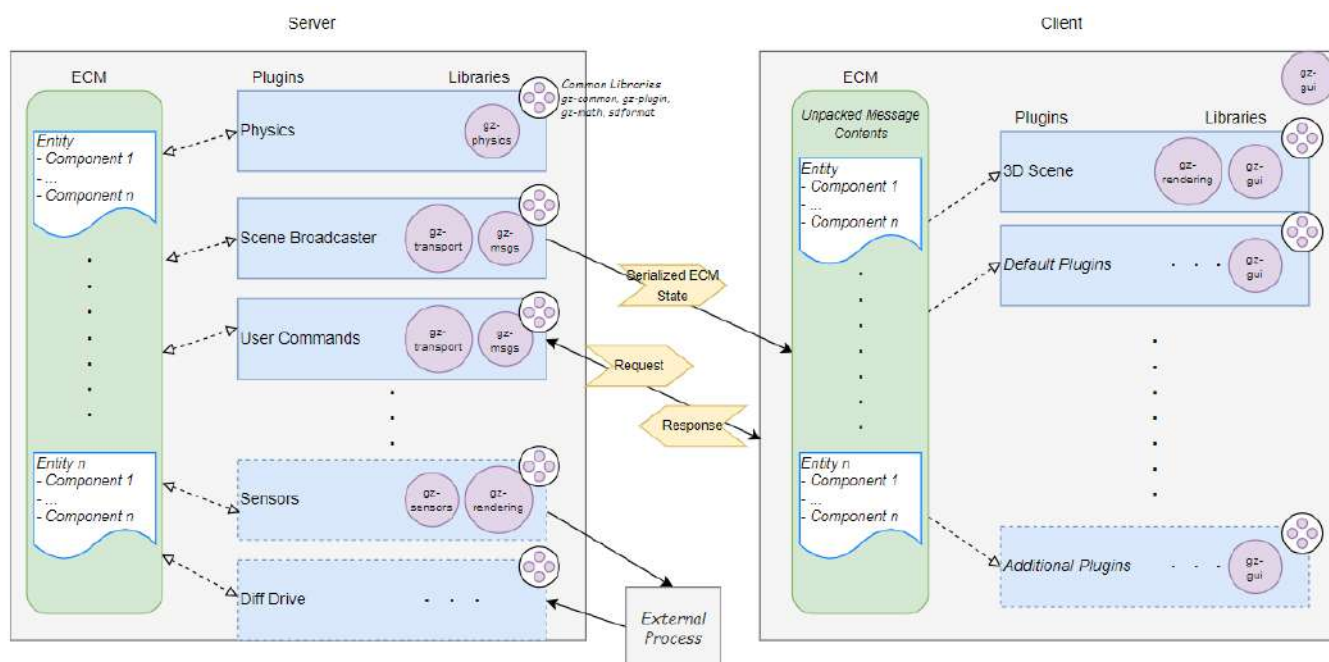


Рисунок 3.2 – Структура процесу модулювання в ПЗ Gazebo

3.2 Структура системи

На рис.3.3 показана схема найважливіших частин системи, включаючи виявлення, управління та програмне забезпечення для моделювання. Кожен блок зазвичай складається з одного або двох вузлів у ROS (операційна система робота) – структурі, що використовується у цій роботі.

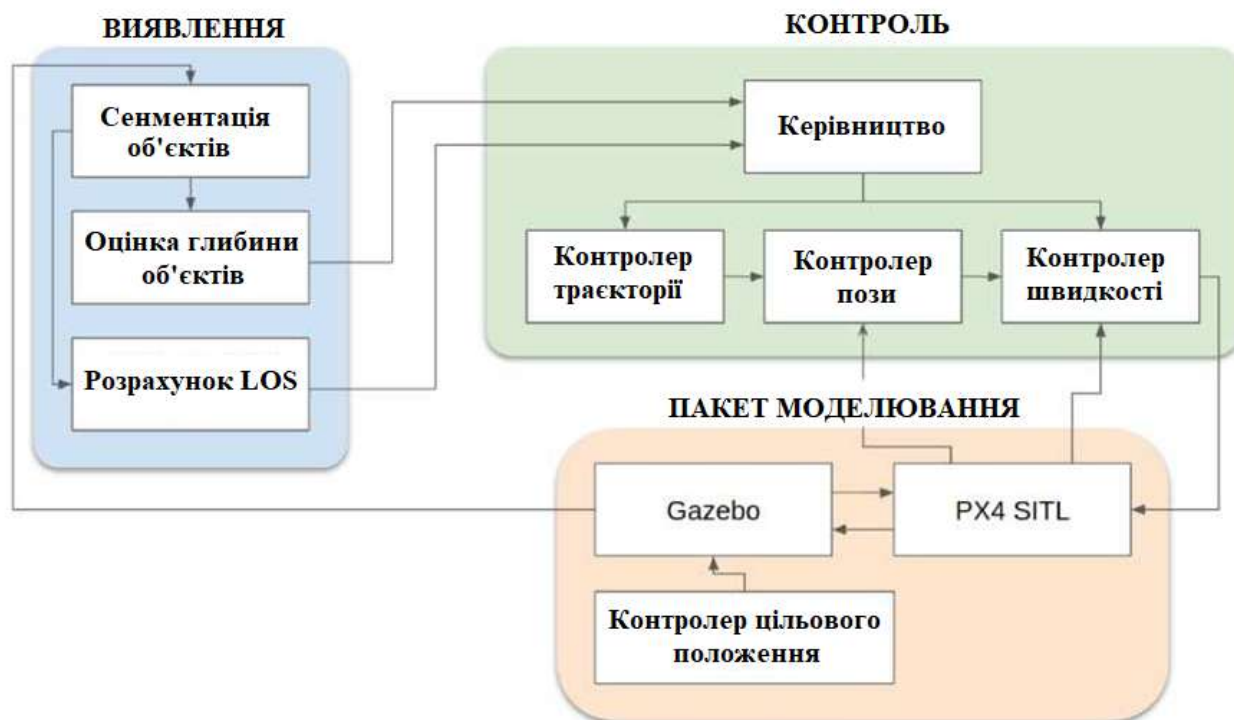
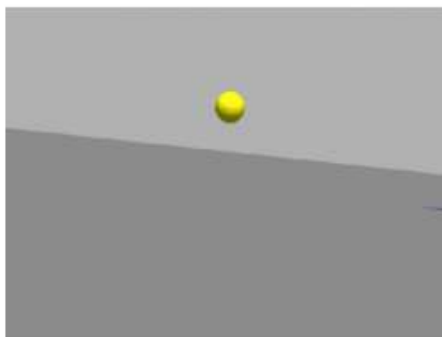
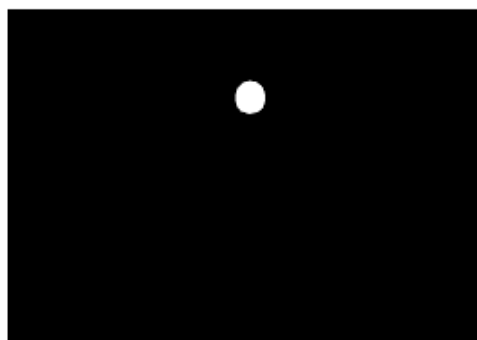


Рисунок 3.3 – Схема системи

Сегментація об'єктів



(а) Кадр RGB-зображення з ціллю під час переслідування



(б) Двійкове сегментоване зображення з виділеною ціллю

Рисунок 3.4 – Вхідні та вихідні дані вузла сегментації об'єктів

Оскільки в цьому розділі основна увага приділялася розгляду методів наведення, завдання сприйняття спрощено. Симуляція не включає насичені фонові візуальні ефекти, в кадрі немає випадень, а об'єкт був зменшений з потенційної форми БПЛА до жовтої сфери. Вузол сегментації складається з колірного порогу, який створює зображення двійкової сегментації кожного кадру камери в реальному часі (30 Гц для NTSC), як показано на рис.3.3. Крім того, цей вузол публікує центроїд (C_x , C_y) виявленого об'єкта, використовуючи моменти зображення.

$$\begin{bmatrix} C_x \\ C_y \end{bmatrix} = \begin{bmatrix} \frac{M_{10}}{M_{00}} \\ \frac{M_{01}}{M_{00}} \end{bmatrix} \quad (3.1)$$

Оцінка глибини об'єкта

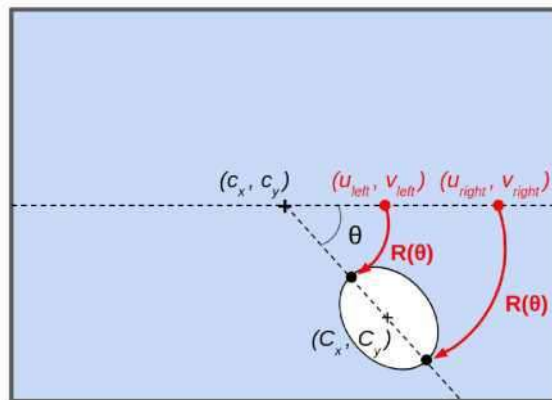


Рисунок 3.5 – Приклад кадру зображення з центроїдом виявленого об'єкта та частиною обчислень оцінки глибини. (U_{left}, U_{left}) і (U_{right}, U_{right}) точки вздовж горизонталі зображення, що відповідають цільовим краям, показані до (червоним) і після (чорним) повороту назад до початкових сегментованих координат. Через широке поле зору зображення навколо країв рамки зображення можуть викривлятися, навіть якщо в цій моделі камери немає спотворень.

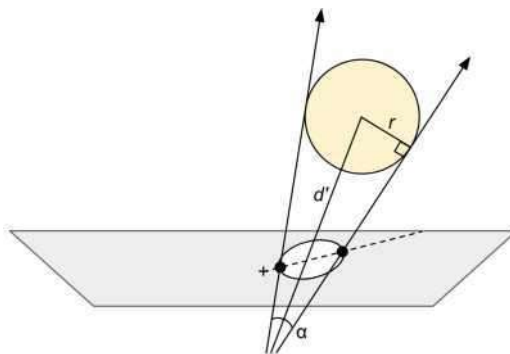


Рисунок 3.6 – Діаграма зверху вниз, що показує розрахунок розрахункової відстані до центру об'єкта d' за допомогою 3D-проекції променів до цільових країв.

Оцінка глибини за допомогою зображень із монокулярної камери ґрунтується на попередньому знанні розміру цілі, а також на припущенні, що ціль є сферичною (що є правильним у цьому моделюванні, але його можна поширити на реальні цілі з приблизним знанням типу цілі чи літака). Оцінка враховує внутрішні параметри камери та проектує 2D-інформацію (пікселі) у 3D (промені, що починаються в центрі камери), щоб отримати глибину.

Потрібно визначити координати пікселів у двох точках зображення, які утворюють двовимірну лінію на зображенні, яка проектується на тривимірну лінію вздовж діаметра об'єкта. Найпростішою такою лінією є лінія, що проходить через головну точку зображення, яка проходить через центроїд, оскільки ціль є сферою. Ця пунктирна лінія проходить через ціль на рис. 3.5. Спочатку ми повинні знайти координати пікселів уздовж цієї лінії, які відповідають краям об'єкта на зображенні.

Спочатку ми знаходимо кут повороту, утворений координатами центроїда відносно горизонталі зображення, позначений Θ на рис. 3.5.

$$\theta = \arctan \frac{c_y - c_y}{c_x - c_x} \quad (3.2)$$

З цим кутом ми можемо використовувати матрицю обертання, щоб повернути сегментований об'єкт на лінію $y = c_y$. Зверніть увагу, що ми використовуємо Θ для досягнення правильного повороту, і що ми спочатку обчислюємо нові піксельні координати (u_i, v_i) відносно центру зображення, від вихідних координат (x_i, y_i) .

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} x_i - c_x \\ y_i - c_y \end{bmatrix}, i=1, \dots, N \quad (3.3)$$

$$\begin{bmatrix} u_{i,R} \\ v_{i,R} \end{bmatrix} = \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix} \begin{bmatrix} u_i \\ v_i \end{bmatrix}, i=1, \dots, N \quad (3.4)$$

Де N — загальна кількість сегментованих пікселів у бінарному сегментованому зображенні. Після обертання об'єкта на зображення горизонтально, прості операції $\min$ і $\max$ дають крайні піксельні координати ліворуч і праворуч від цілі (або праворуч і ліворуч, якщо $(C_x < c_x)$).

$$\begin{bmatrix} u_{left} \\ v_{left} \end{bmatrix} = \begin{bmatrix} \min_i(u_i) \\ c_y \end{bmatrix}, i=1, \dots, N \quad (3.5)$$

$$\begin{bmatrix} u_{right} \\ v_{right} \end{bmatrix} = \begin{bmatrix} \max_i(u_i) \\ c_y \end{bmatrix}, i=1, \dots, N \quad (3.5)$$

Ці дві точки показані червоним кольором на рис. 3.5. Ми застосовуємо зворотне обертання та трансляцію, щоб отримати вихідні координати цих двох ідентифікованих піксельних координат.

$$\begin{bmatrix} u_{left,R^{-1}} \\ v_{left,R^{-1}} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} u_{left} \\ v_{left} \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} \quad (3.6)$$

$$\begin{bmatrix} u_{right,R^{-1}} \\ v_{right,R^{-1}} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} u_{right} \\ v_{right} \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} \quad (3.7)$$

Ці два пікселі проєктуються на 3D-промені, подібно до обчислення променів LOS у рівнянні 2.9. Їх також можна побачити в подібному вигляді зверху вниз на рис. 3.6. Коли два промені знайдено, глибина (до найближчої точки на об'єкті) буде наступною:

$$d = \frac{\omega/2}{\sin(\alpha/2)} - \frac{\omega}{2} \quad (3.8)$$

LOS і обчислення LOS

Обчислення r_t , вектора LOS, можна знайти в рівнянні 2.9. Обчислення F_{sh} і $1n$, масштабування та спрямованості компонентів LOS', можна знайти в рівняннях 2.11 і 2.14. Перед введенням у вузол Guidance величини проходять згладжування за допомогою плоского ковзного середнього фільтра. Відфільтровані значення використовуються для навігацій на основі траєкторії, тоді як навігації на основі LOS використовують необроблені значення.

Вузол «Наведення» виконує наведення на основі LOS або на основі траєкторії. Настанова на основі LOS, як описано в розділі 2.4, використовує LOS і обчислення LOS з блоку виявлення. У цьому випадку вузол видає команди прискорення, які задовольняються контролером швидкості. Наведення на основі траєкторії використовує як інформацію LOS, так і оцінку глибини для створення тривимірних траєкторій до поточного стану або прогнозованого руху цілі. Тут вузол виводить траєкторію, що складається з маршрутних точок і відповідних швидкостей, яка приймається контролером траєкторії. Кожен метод ініціалізується за допомогою двох секунд простого наведення LOS, коли БПЛА приймає команди швидкості безпосередньо вздовж поточного вектора LOS.

Контролери траєкторії, пози та швидкості

Контролер траєкторії приймає траєкторію у формі списку маршрутних точок, кожна з яких має відповідне (x, y, z) положення, поворот і швидкість (скаляр). Він виводить точку відстеження, яку контролер пози приймає як вхідні дані для відстеження траєкторії, і точку огляду вперед, яка використовується як точка перепланування. Настроюваний час прогнозування встановлює, наскільки далеко вперед знаходиться точка прогнозу від точки відстеження, і приблизно дорівнює $1/f + \Delta$, де f — частота повторного планування, а Δ_{buffer} — буфер на випадок, якщо в системі є певна затримка. Контролер позиції складається з ПД-регуляторів із точкою відстеження з контролера траєкторії як еталонною та одометрією з інтерфейсу PX4 SITL як фактичною. Це виводить еталонну швидкість (і термін прямого зв'язку швидкості з контролера траєкторії) до контролера швидкості, який також приймає одометрію з інтерфейсу PX4 SITL як фактичну. Контролер

швидкості також використовує ПД-регулятори та виводить команди крену, тангажу, повороту, тяги до інтерфейсу PX4 SITL.

Моделювання використовувалося для розробки, розгортання та швидкої оцінки багатьох алгоритмів наведення БПЛА. У цьому середовищі поведінка БПЛА та цілі легко змінювалася, і її можна було змінити без ризику для безпеки, пов'язаного з тестуванням у реальному світі. Gazebo [22], симулятор вибору, зазвичай використовується для тестування робототехніки та моделює динаміку робота за допомогою фізичного механізму ODE. Це було поєднано з автопілотом PX4 [23] програмним забезпеченням у циклі (SITL) для керування квадрокоптером.

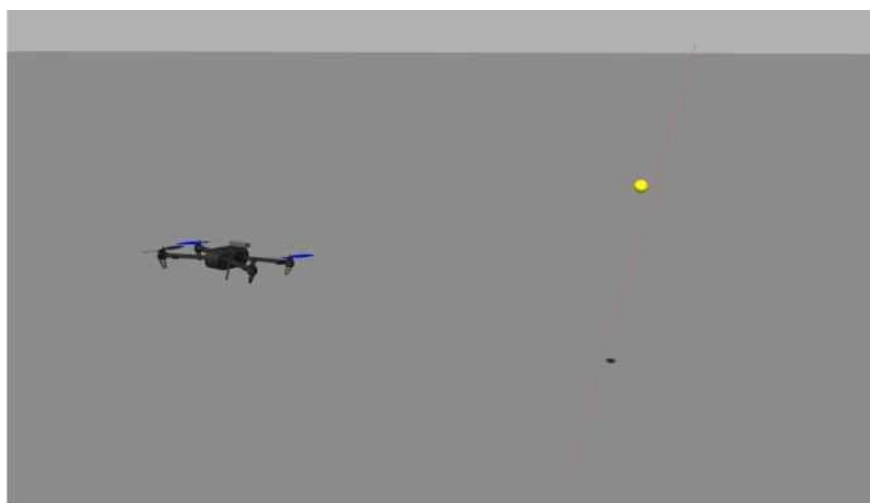
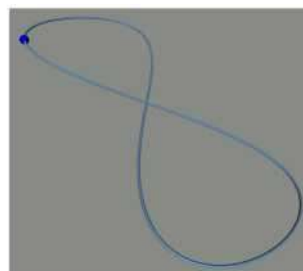
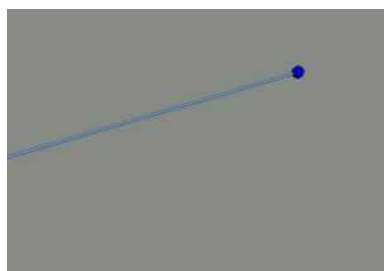


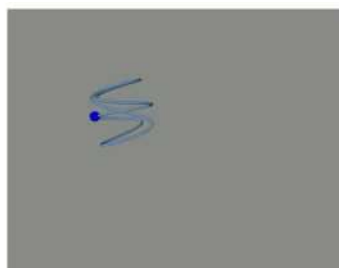
Рисунок 3.7 – Симуляція в Gazebo з моделлю БПЛА та ціллю

Приклад змодельованого світу можна побачити на рис. 3.7. Перешкоди, візуально насичений фон та інші фактори були усунені в установці моделювання, щоб зменшити вплив зовнішніх факторів на оцінку алгоритмів наведення. При розгортанні на реальних UAS алгоритми тут можна об'єднати в більшу архітектуру автономності, включаючи надійне виявлення та відстеження, а також уникнення перешкод тощо. Квадротор оснащений камерою RGB, яка моделює реалістичний датчик зору

Контролер цільової позиції



(а) Прямий цільовий шлях. (б) Маршрут цілі у формі вісімки.



(в) Цільовий шлях вузла.

Рисунок 3.8 – Бібліотека цільового шляху, взята з подібних точок зору. Темно-синя сфера – поточне положення цілі. Світло-блакитна послідовність — це трасований шлях. (а) прямий шлях з випадковим нахилом; (б) доріжка у формі вісімки з довільним нахилом, розміром 10 м x 6 м; (в) шлях вузла з випадковою центральною точкою розміром 2 м x 2 м x 2 м.

Бібліотека цільових траєкторій була використана для навантаження на здатність БПЛА перехопити ціль. Перший, найпростіший рух цілі — це пряма траєкторія з постійною швидкістю, яка може імітувати автономний літальний апарат на пошуковому шаблоні або літак з нерухомим крилом. Початкова позиція розміщує ціль по обидва боки від поля зору БПЛА, а шлях перетинає перед БПЛА з деякою випадковою зміною нахилу. Друга траєкторія — це цифра-вісімка з випадковим тривимірним нахилом, схожим на маневр ухилення, який може зробити малий літак. Ці траєкторії можна побачити на рис. 3.8. Третя траєкторія складається з форми вузла, що заповнює простір розміром 2 м x 2 м x 2 м у випадковому місці в межах зони 10 м x 20 м x 10 м перед БПЛА. Цей більш швидкий рух вперед-назад схожий на зависання мультиротора в мінливому полі вітру.

3.3 Результати моделювання алгоритмів

Було оцінено п'ять алгоритмів навігації, описаних у попередніх розділах. Конфігурація експерименту, що складається з вибору одного параметра з кожної з наступних категорій. Кожна конфігурація пройшла 50 випробувань.

Наведення БПЛА:

- справжня пропорційна навігація (TPN);
- пропорційна навігація з керуванням курсом (PN-напрямок);
- гібридна справжня пропорційна навігація з керуванням курсом (гібридна TPN-навігація);
- траєкторія LOS;
- прогнозована траєкторія.

Швидкість БПЛА [м/с]: 2.0, 3.0, 4.0, 5.0.

Шлях мішені: пряма, цифра-8, вузол.

Швидкість цілі [% від швидкості БПЛА]: 25%, 50%, 75%, 100%.

У випадку синусоїдальних цільових траєкторій (цифра 8 і вузол) цільову швидкість встановлювали шляхом визначення довжини траєкторії та ділення на бажану швидкість для обчислення періоду синусоїд.

Основним показником для порівняння різних методів є частота влучень першого проходу, представлена в розділі 2.6. Усі наведені нижче умови повинні бути виконані, щоб спроба вважалася успішною першою передачею в мішень:

- БПЛА знаходиться в межах 0,5 м від цілі (вимірюється від найближчої точки цілі до центру БПЛА);
- тривалість переслідування менше 20 с;
- БПЛА знаходиться в зоні 35 x 100 x 40 метрів навколо цілі;
- ціль не знаходиться за межами поля зору камери БПЛА більше 3 секунд.

Ці умови були визначені після розгляду сцени моделювання, моделі БПЛА та моделі цілі. Тестова ціль — жовта сфера діаметром 1 м. Камера RGB має горизонтальний кут зору 105° з розміром 680x480. Замість курсової камери під час моделювання модель БПЛА мала жорстко закріплену камеру під кутом, яка

регулювалася для кожної швидкості, щоб компенсувати стійкий нахил вниз під час польоту вперед на високих швидкостях.

Наступний показник — це середня тривалість переслідування. Вони були обчислені за допомогою середнього вимірювання часу до попадання для кожного з (а) швидкості цілі та (б) швидкості БПЛА відповідно. Це було зроблено з використанням лише успішних ударів під час кожного експерименту.

Точки даних на наступних теплових картах, розташовані всередині чорного квадрата, представляють експерименти, які не змогли завершити понад 95% випробувань без збоїв через нестабільність БПЛА з конкретною конфігурацією експерименту.

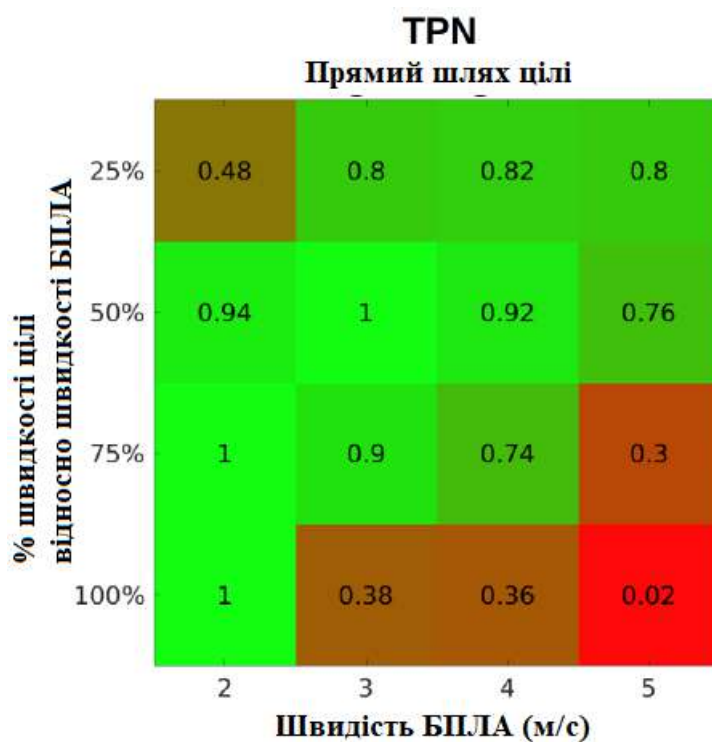


Рисунок 3.9 – Частота влучень True Proportional Navigation (Пряма траєкторія цілі)

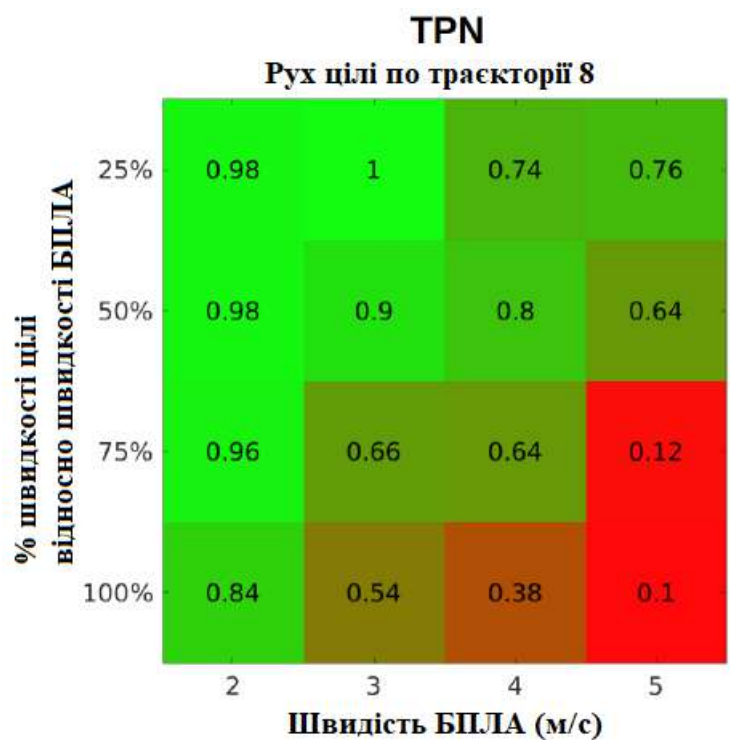


Рисунок 3.10 – Частота влучень True Proportional Navigation (Ціль з траєкторією у вигляді вісімки)

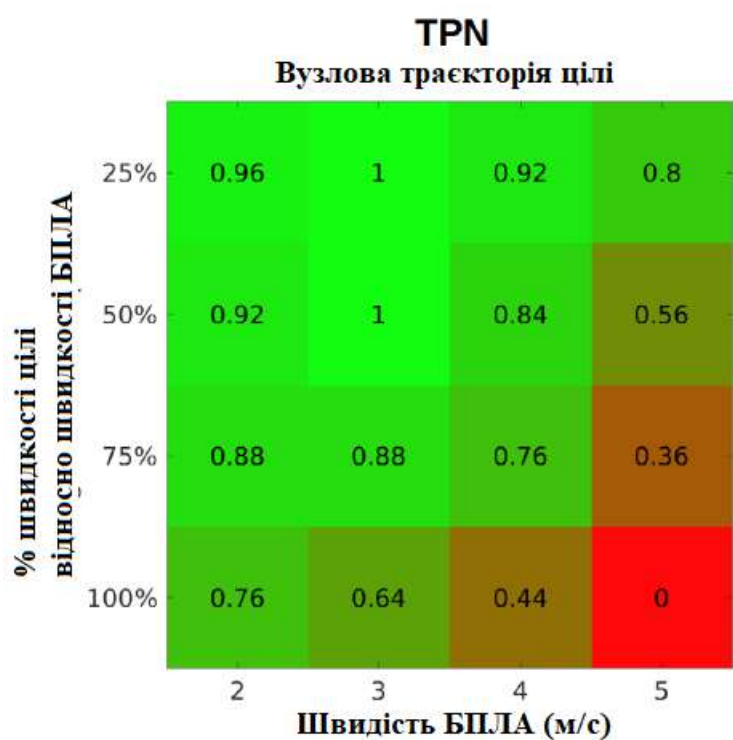


Рисунок 3.11 – Частота влучень True Proportional Navigation (Вузлова траєкторія цілі)

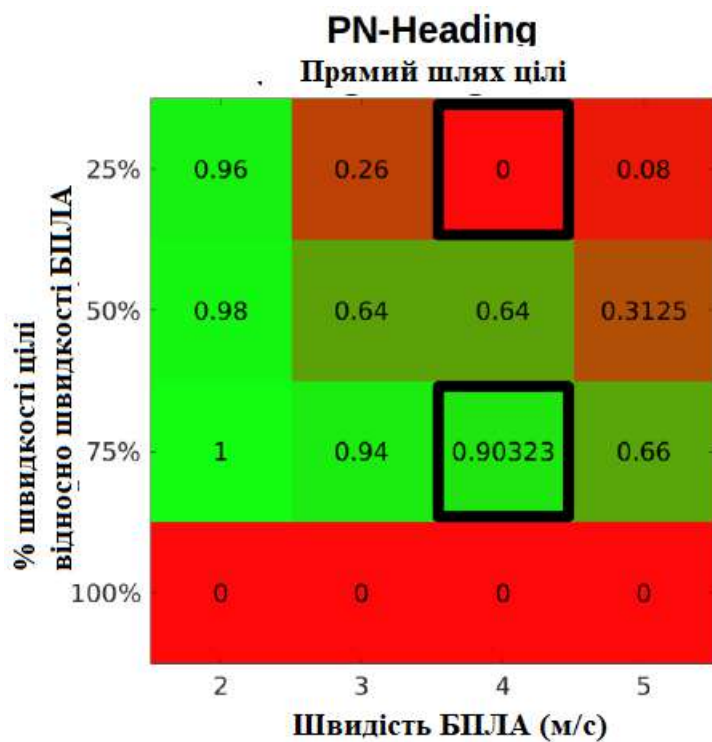


Рисунок 3.12 – Пропорційна навігація з частотою влучень Heading Control
(Пряма траєкторія цілі)

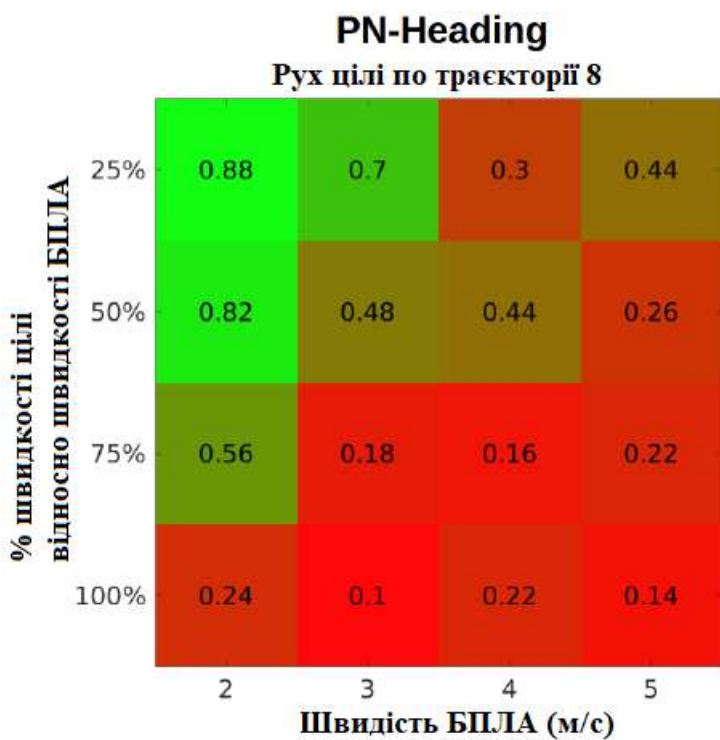


Рисунок 3.13 – Пропорційна навігація з частотою влучень Heading Control
(Ціль з траєкторією у вигляді вісімки)

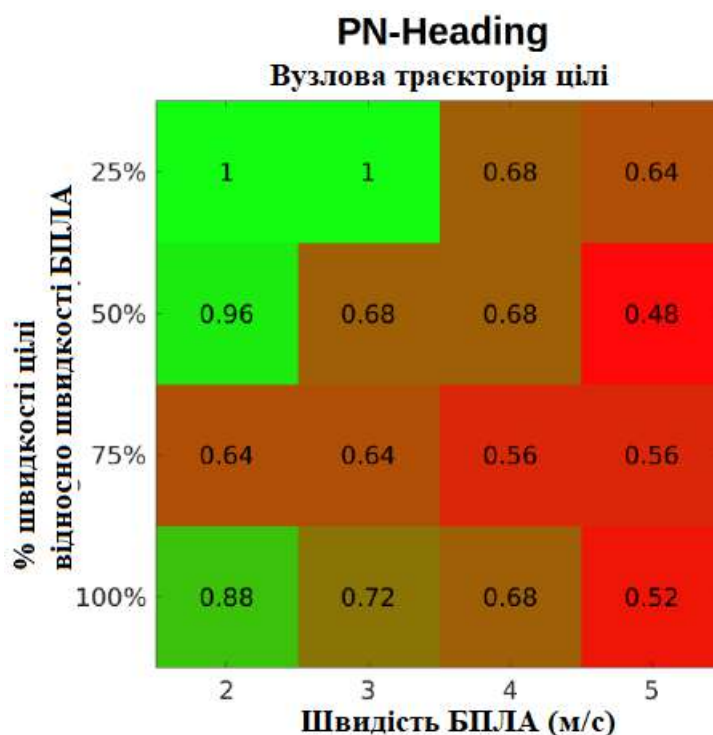


Рисунок 3.14 – Пропорційна навігація з частотою влучень Heading Control
(Вузлова траєкторія цілі)

ТРN досягає найвищої частоти влучень у майже всіх конфігураціях порівняно з іншими методами в обох класах (LOS Guidance та Trajectory Following). У всіх експериментах спостерігається тенденція до зниження швидкості БПЛА та цілі, що призводить до вищих показників попадання, іноді навіть 1,0 (тобто 50 із 50 випробувань призводять до попадання). Переміщення вниз і вправо в межах кожної підфігури показує результати на вищих цілях і швидкостях БПЛА. Це можна розглядати як збільшення швидкості зближення, що зменшує час, який БПЛА має відреагувати на зміни в русі цілі. Як видно на малюнках, це призводить до нижчих показників попадання, що може бути пов'язано з відставанням у здатності контролерів БПЛА виконувати бажане прискорення.

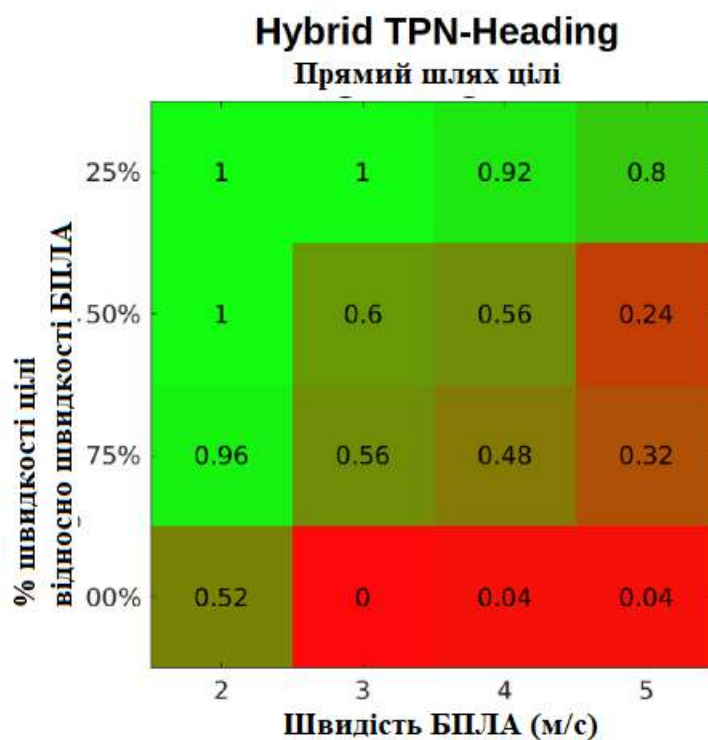


Рисунок 3.15 – Змішана істинно пропорційна навігація-керування частота попадань (Пряма траєкторія цілі)

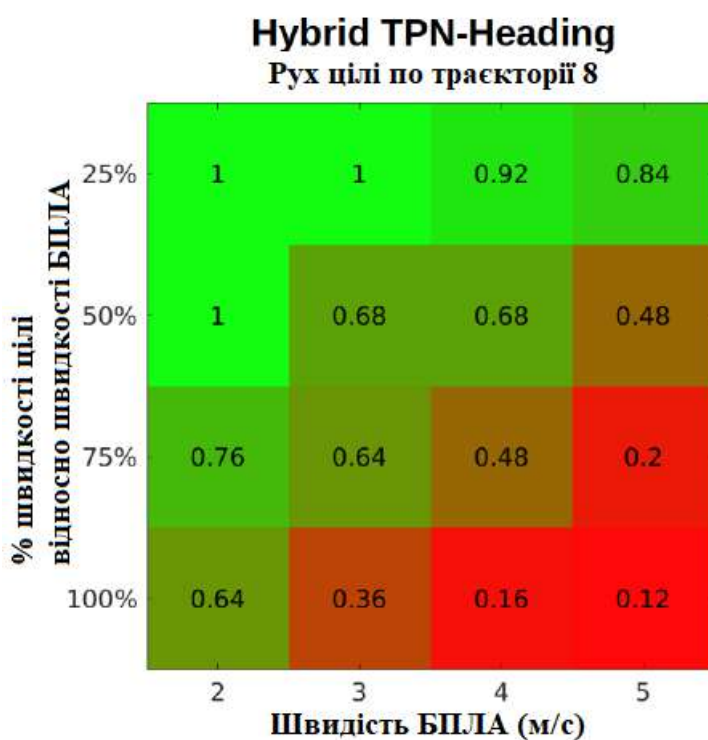


Рисунок 3.16 – Змішана істинно пропорційна навігація-керування частота попадань (Ціль з траєкторією у вигляді вісімки)

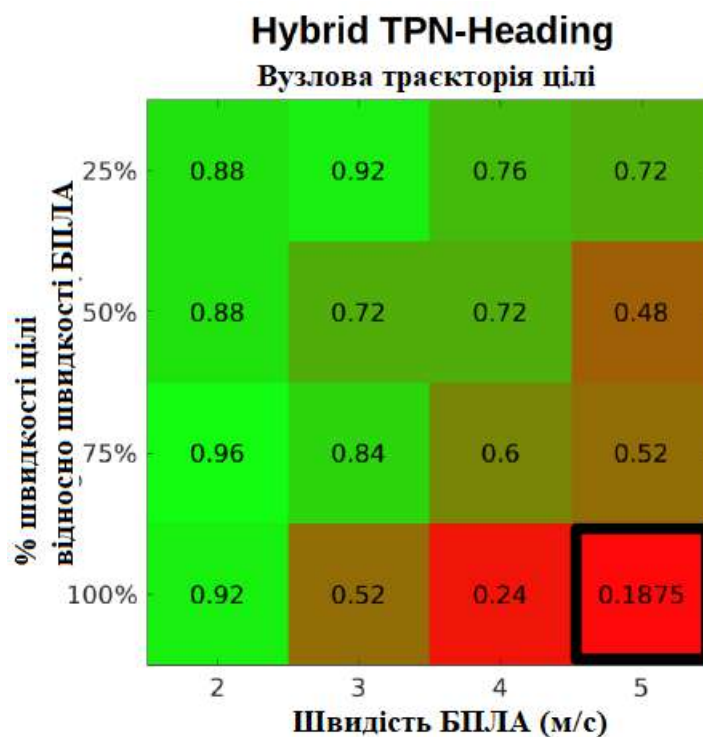


Рисунок 3.17 – Змішана істинно пропорційна навігація-керування частота попадань (Вузлова траєкторія цілі)

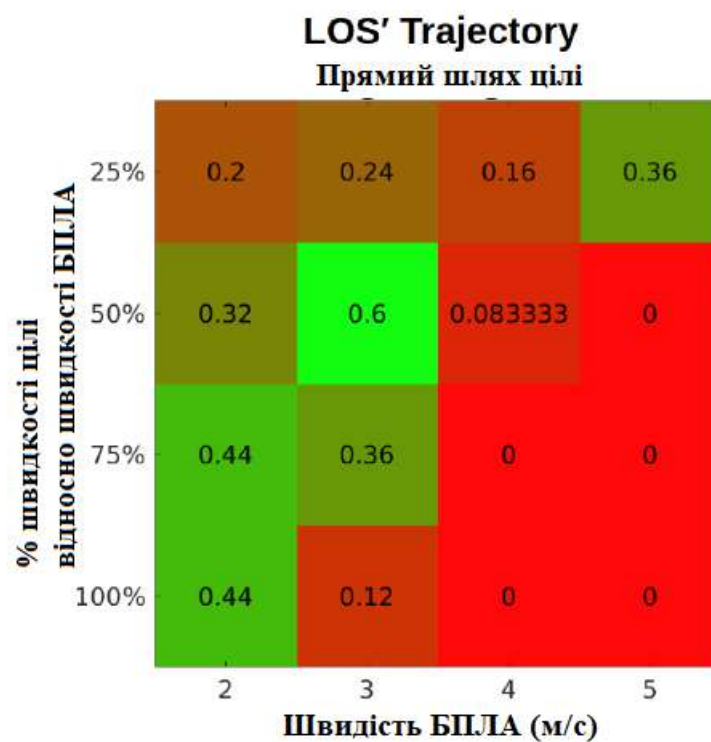


Рисунок 3.18 – Коефіцієнт попадання траєкторії LOS (Пряма траєкторія цілі)

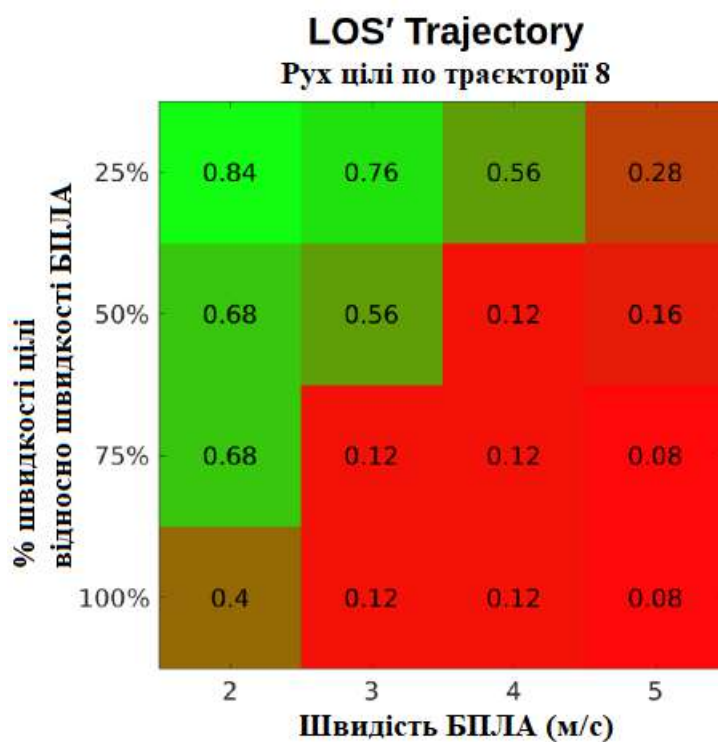


Рисунок 3.19 – Коефіцієнт попадання траєкторії LOS (Ціль з траєкторією у вигляді вісімки)

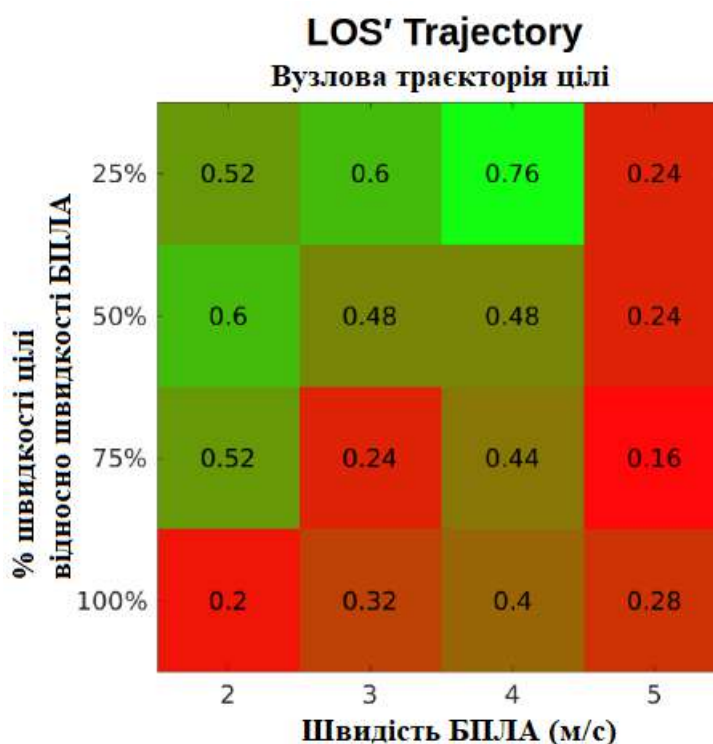


Рисунок 3.20 – Коефіцієнт попадання траєкторії LOS (Вузлова траєкторія цілі)

Квадрокоптер досягає бокового прискорення більш безпосередньо, ніж корабель із нерухомим крилом, створюючи кут крену і таким чином зміщуючи

компонент тяги до цієї осі. Однак і момент літака, і час відгуку диспетчера сприяють відставанню в досягненні необхідного кута крену. Таким чином, оскільки час, відведений для цього контролю, зменшується зі збільшенням швидкості зближення, менш імовірно, що буде досягнуто необхідне поперечне прискорення. Подібна логіка застосовується до контролера тяги для досягнення бажаного прискорення по осі z . Цілком можливо, що точніший контролер може збільшити показники влучень при високих швидкостях закриття.

Метод гібридного заголовка TPN загалом мав вищу частоту попадань, ніж метод PN-заголовка, але нижчий, ніж TPN. Проте Hybrid TPN-Heading незмінно показував кращі показники, ніж TPN, на низькій цільовій швидкості (25%) на прямій траєкторії та цільовій траєкторії у формі восьмерки, з найвищим збільшенням показника влучень у цих конфігураціях – 0,28. Це означає, що подібно до поведінки PN-Heading на низьких цільових швидкостях (додатково описано в розділі нижче), Hybrid TPN-Heading здатний переслідувати ціль на цих

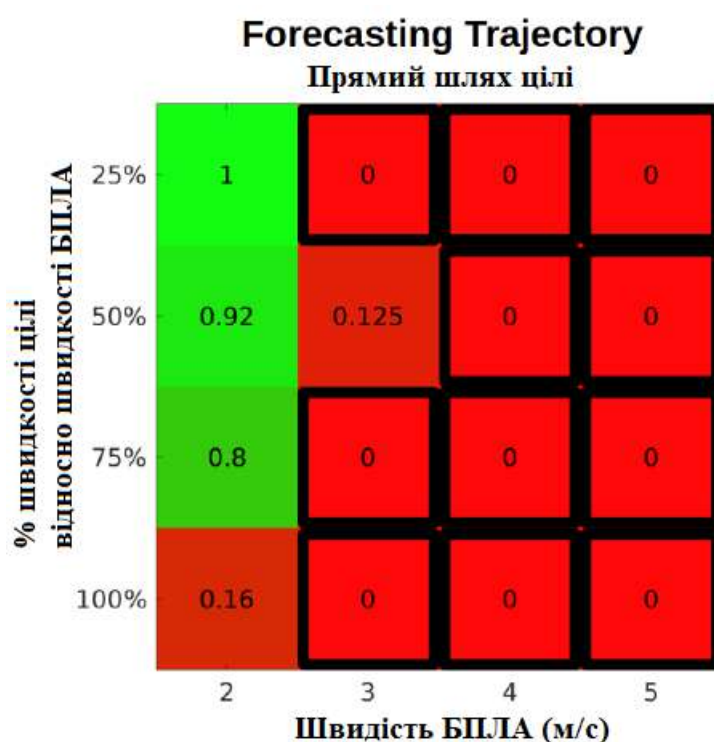


Рисунок 3.21 – Прогноз траєкторії влучень (Пряма траєкторія цілі)

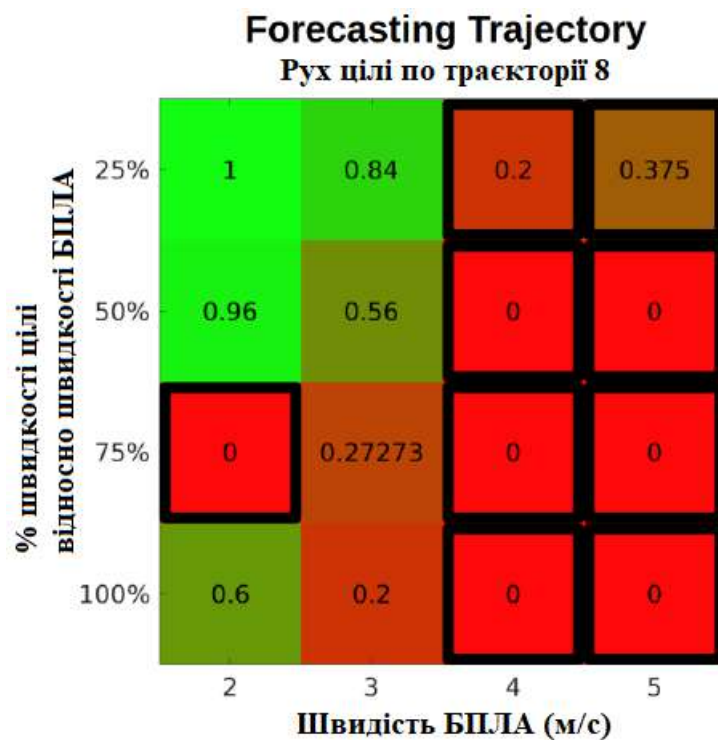


Рисунок 3.22 – Прогноз траєкторії влучень (Ціль з траєкторією у вигляді вісімки)

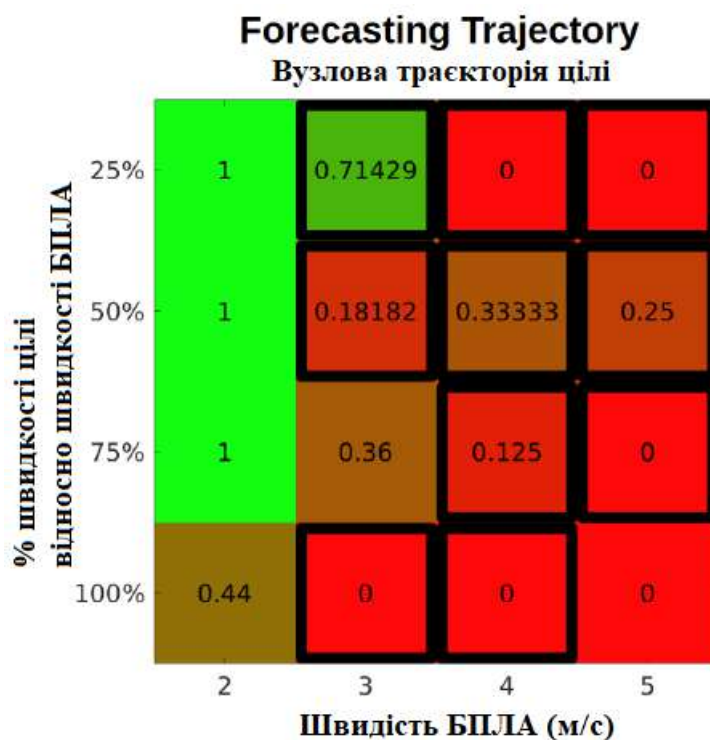


Рисунок 3.23 – Прогноз траєкторії влучень (Вузлова траєкторія цілі)

Клас методів LOS Guidance зазвичай має вищі показники влучень, ніж методи слідування траєкторії, реалізовані тут. При розробці цих алгоритмів та їх реалізації

було виявлено, що значні налаштування та фільтрація параметрів необхідні для покращення результатів методів слідування траєкторії. Наприклад, хоча метод траєкторії LOS створює плавні траєкторії та, отже, більш плавний політ БПЛА, він повинен використовувати відфільтрований (згладжений фільтром плоского ковзного середнього) LOS, щоб створити послідовні траєкторії. Ця фільтрація вводить затримку, яка швидко стає нерозв'язною, коли швидкість цілі або БПЛА збільшується за межі, які використовувалися для налаштування системи. Це посилюється у випадку траєкторії прогнозування. Хоча рівняння 2.32 геометрично обґрунтоване, недосконалість змодельованої системи в оцінці глибини, LOS-обчисленні та відповідних трансформаціях кадру (особливо під час високих швидкостей крену та тангажу) спричиняють неточності, які потребують фільтрації та налаштування, щоб зробити тривимірний прогноз цілі можливим. Ці обмеження здебільшого є побічним продуктом використання інформації лише про монокулярну камеру для оцінки 3D-позиціонування. Ефект можна побачити в нестабільності БПЛА в більшості експериментів, показаних на рис. 3.21-3.23.

Хоча TPN загалом досягає найвищого відсотка влучень порівняно з усіма іншими методами, представленими тут, помітним винятком є найповільніша конфігурація швидкості БПЛА 2 м/с і швидкості цілі 25% з прямою траєкторією до цілі, яка має коефіцієнт влучності 0,48. На рис. 3.24-3.30 показано, що ціль поступово віддаляється від поля зору, коли БПЛА пролітає повз. Пряма цільова траєкторія починається з одного боку попередньо визначеного простору арени та закінчується з іншого боку з випадковими варіаціями нахилу шляху. Оскільки ціль рухається дуже повільно в цій конфігурації, вона залишається збоку від зображення БПЛА. Компонент у команді прискорення кадру тіла, створений TPN, коли ціль рухається до краю зображення (через швидкість БПЛА вперед), недостатньо, щоб утримувати ціль у полі зору, і вона повільно вислизає з поля зору. Однак ця проблема стає менш імовірною на вищих швидкостях БПЛА, оскільки більший рух БПЛА створює більше руху цілі на зображенні, таким чином створюючи більшу команду бокового прискорення, яка тримає ціль у полі зору. Ця проблема очевидна лише у випадку прямолінійної траєкторії, оскільки лише за цієї траєкторії початкове

цільове розташування мало бути на краях зображення, щоб компенсувати більший рух на вищих цільових швидкостях. Ця проблема не виникає в методах PN-Heading або Hybrid TPN-Heading, оскільки вони компенсують за допомогою команди yaw-rate для центрування об'єкта на зображенні.

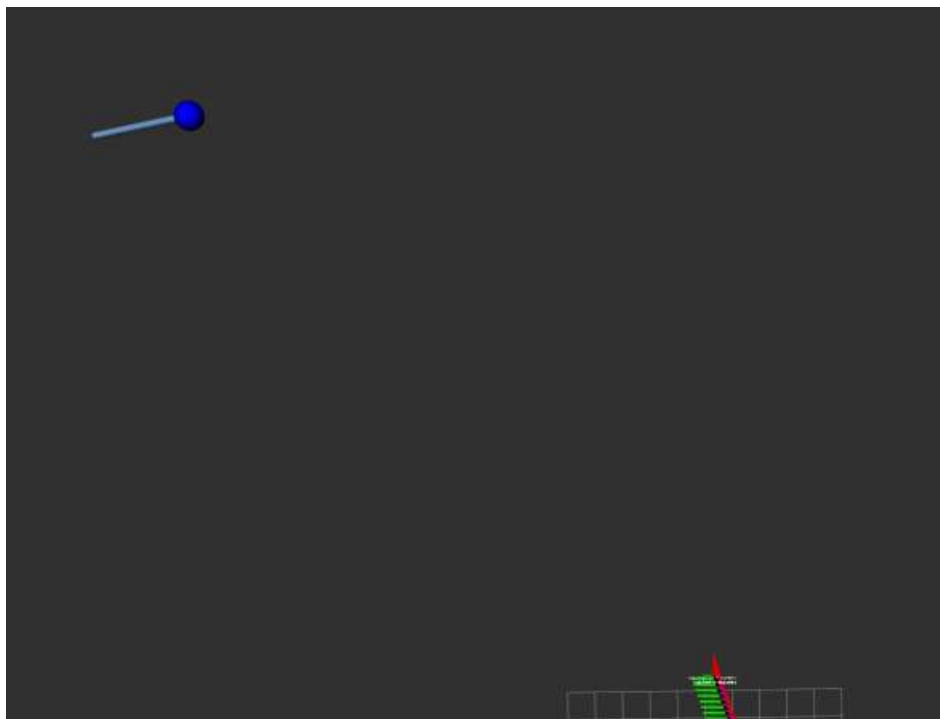


Рисунок 3.24 – Демонстрація TPN, пряма траєкторія цілі, швидкість БПЛА 2 м/с і швидкість цілі 25% (0,5 м/с). Вид зверху вниз БПЛА (осі RGB), що наближається до цілі (синій). Відображені шляхи тривають приблизно 5 секунд, траєкторія БПЛА продовжується за межами екрана.



Рисунок 3.25 – Демонстрація TPN, пряма траєкторія цілі, швидкість БПЛА 2 м/с і швидкість цілі 25% (0,5 м/с). Захоплення кадру зображення з камери БПЛА в точці, показаній на рис.3.24

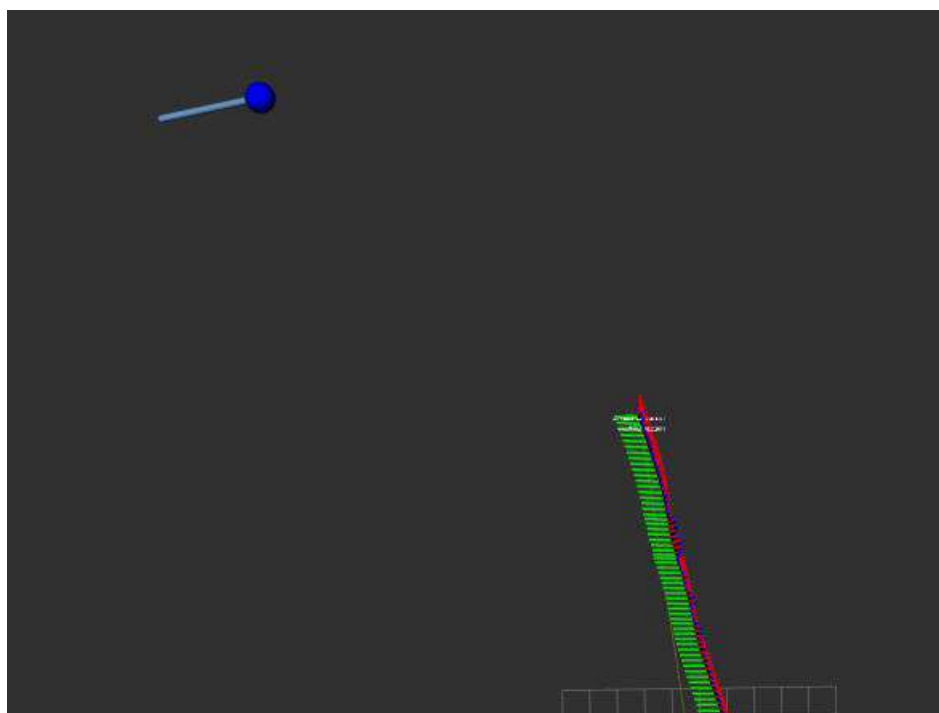


Рисунок 3.26 – Демонстрація TPN, пряма траєкторія цілі, швидкість БПЛА 2 м/с і швидкість цілі 25% (0,5 м/с). Вид зверху вниз приблизно через 2 с після 3.24



Рисунок 3.27 – Демонстрація TPN, пряма траєкторія цілі, швидкість БПЛА 2 м/с і швидкість цілі 25% (0,5 м/с). Захоплення кадру зображення з камери БПЛА в точці, показаній на рис. 3.26

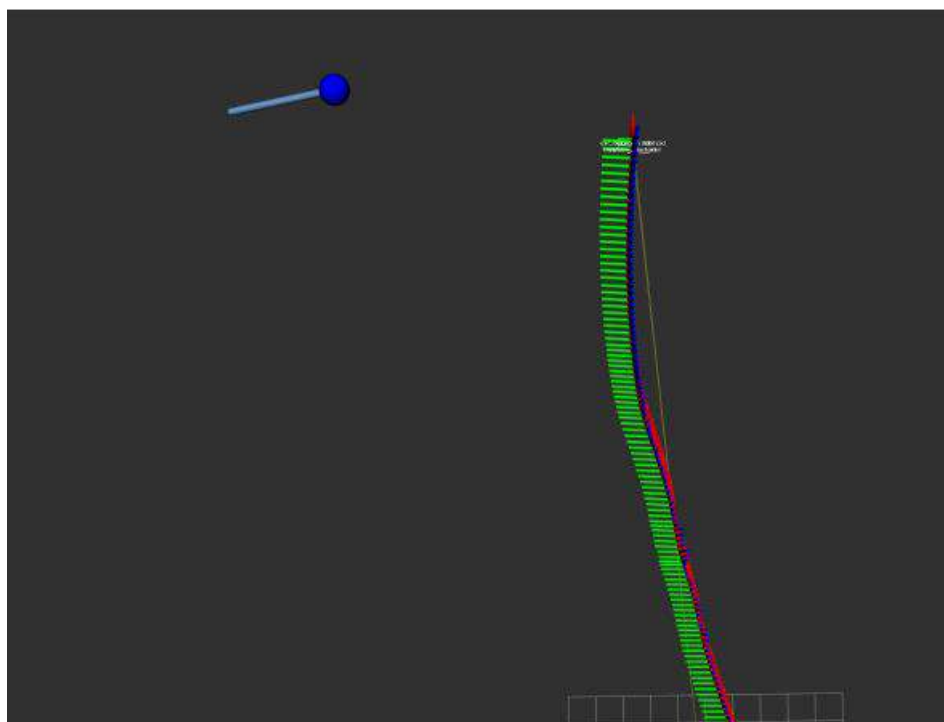


Рисунок 3.28 – Демонстрація TPN, пряма траєкторія цілі, швидкість БПЛА 2 м/с і швидкість цілі 25% (0,5 м/с). Вид зверху вниз приблизно через 2 с після 3.28



Рисунок 3.29 – Демонстрація TPN, пряма траєкторія цілі, швидкість БПЛА 2 м/с і швидкість цілі 25% (0,5 м/с). Захоплення кадру зображення з камери БПЛА в точці, показаній на рис. 3.28

PN-заголовок на високих швидкостях

PN- заголовок має нижчу продуктивність на високих БПЛА або цільових швидкостях порівняно з TPN або Hybrid TPN-Heading. Часто спостерігалось, що це пов'язано з часом, необхідним для ефективної зміни напрямку швидкості БПЛА через швидкість повороту, і ця часова затримка ще більше збільшується зі збільшенням швидкості БПЛА. Помітний артефакт у результатах можна побачити на рис. 3.12, де результати за 100% цільової швидкості дорівнюють 0 для кожної швидкості БПЛА. Це пов'язано з тим, як БПЛА «переслідує» ціль у сценарії прямолінійної траєкторії. На рис. 3.37 показано, що спочатку БПЛА може пройти повз ціль, але повертається до неї за допомогою команди yawrate і знаходить її знову (протягом 3 с тайм-ауту виявлення). Потім він може переслідувати ціль за допомогою команд прискорення PN у напрямках x і z . Однак протягом цього періоду погоні швидкість БПЛА знаходиться в тому ж напрямку, що й ціль. Тому швидкість цілі повинна бути меншою, ніж швидкість БПЛА, інакше неможливо буде підтримувати ненульову швидкість зближення. Це особливо стосується

нижнього рядка рисунка 3.12. Це не відбувається з іншими рухами цілі, оскільки ціль має ненульове прискорення.

3.4 Тривалість переслідування

Якщо в усіх експериментах для певної конфігурації не було збігів, то ці точки даних відсутні на малюнках, як показано на рис. 3.30 та 3.31.

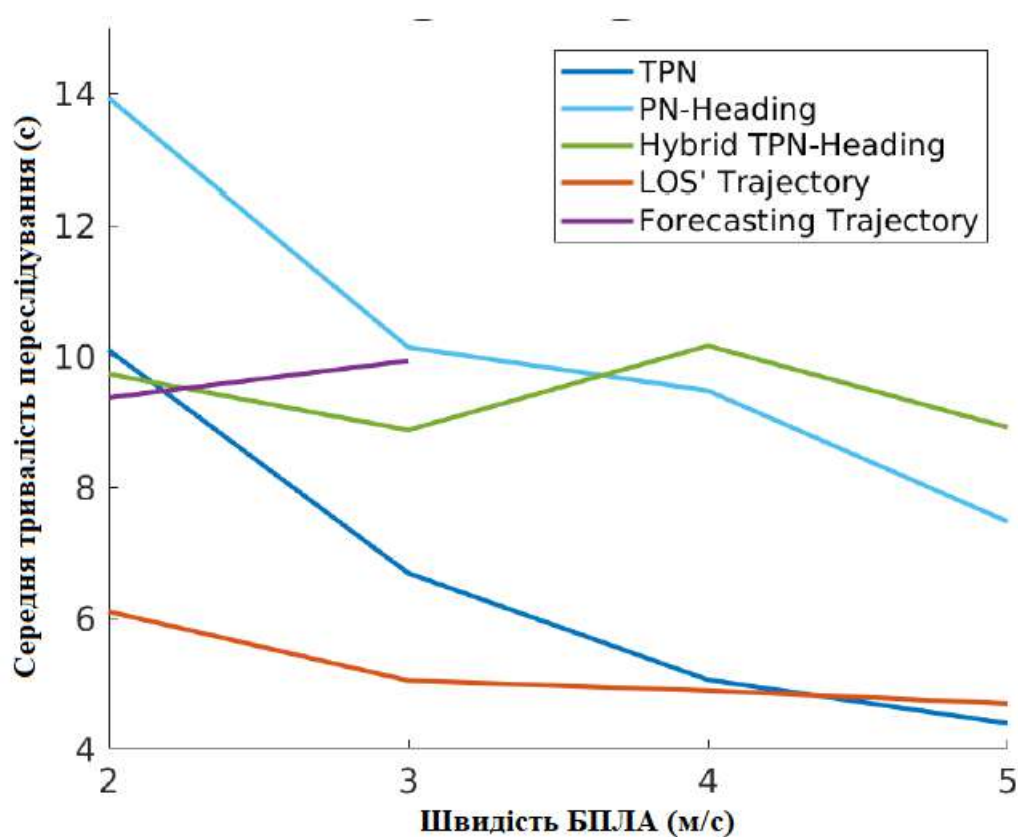


Рисунок 3.30 – Залежність тривалості переслідування від швидкості БПЛА для прямої траєкторії цілі

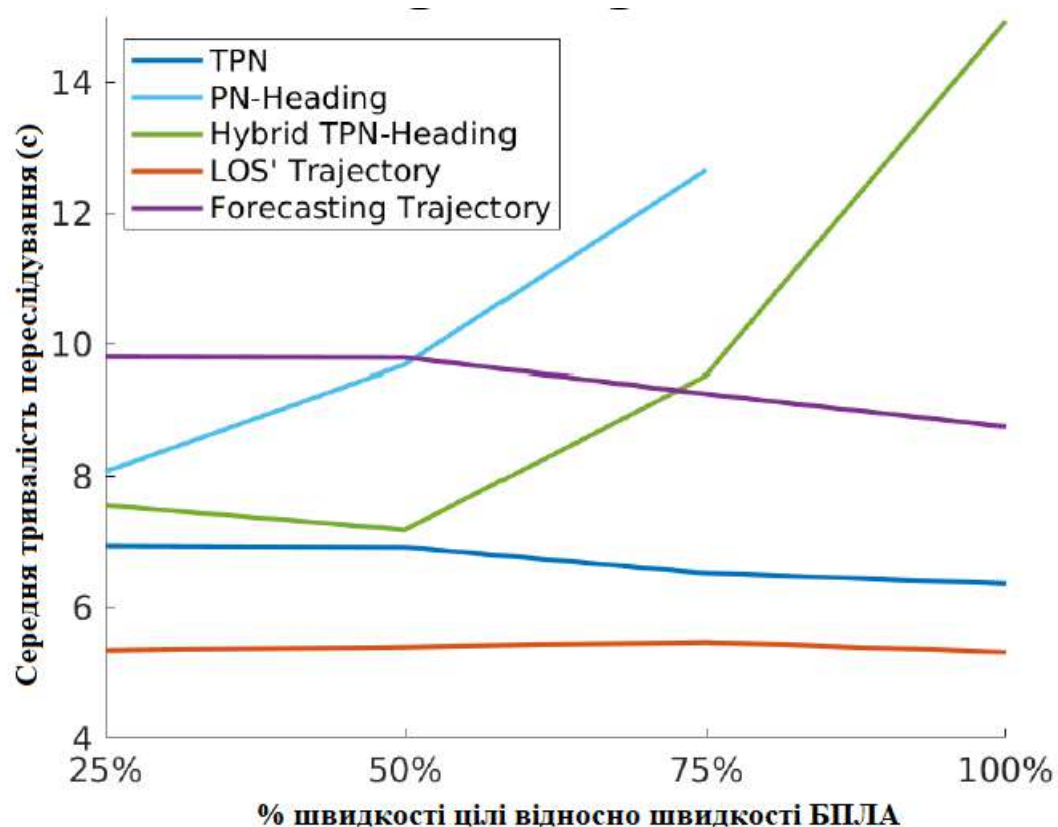


Рисунок 3.31 – Залежність тривалості переслідування від швидкості цілі для прямої траєкторії цілі

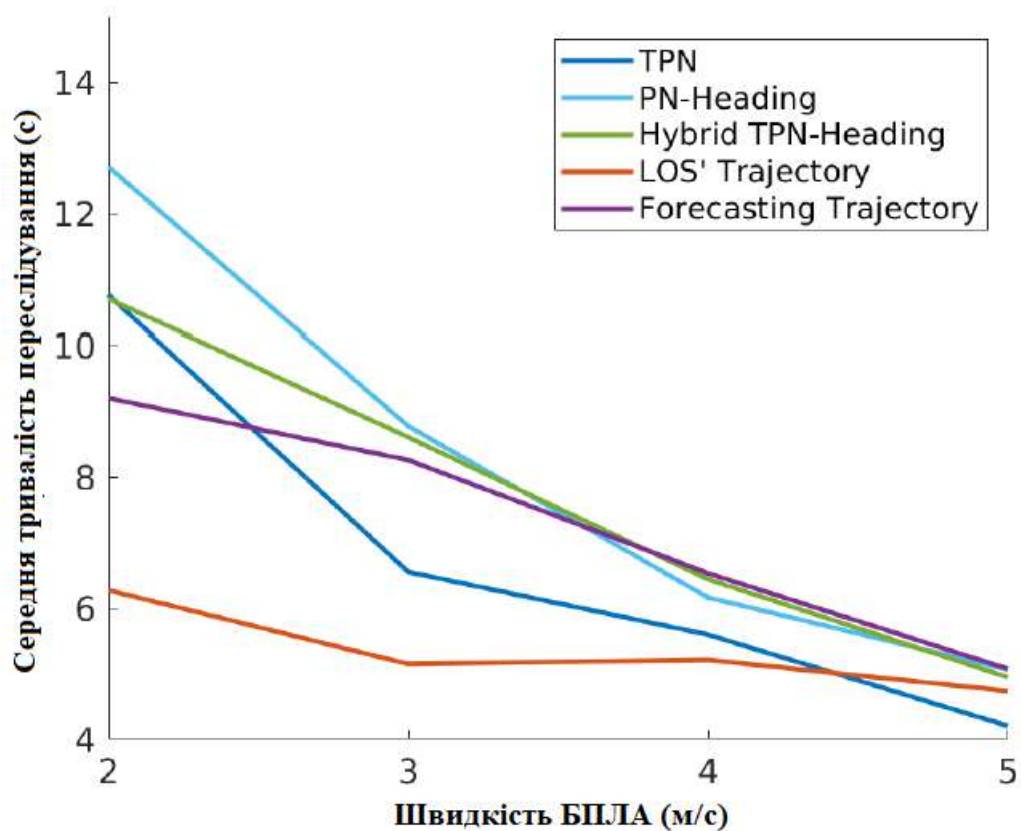


Рисунок 3.32 – Залежність тривалості переслідування від швидкості БПЛА для траєкторії цілі у формі вісімки

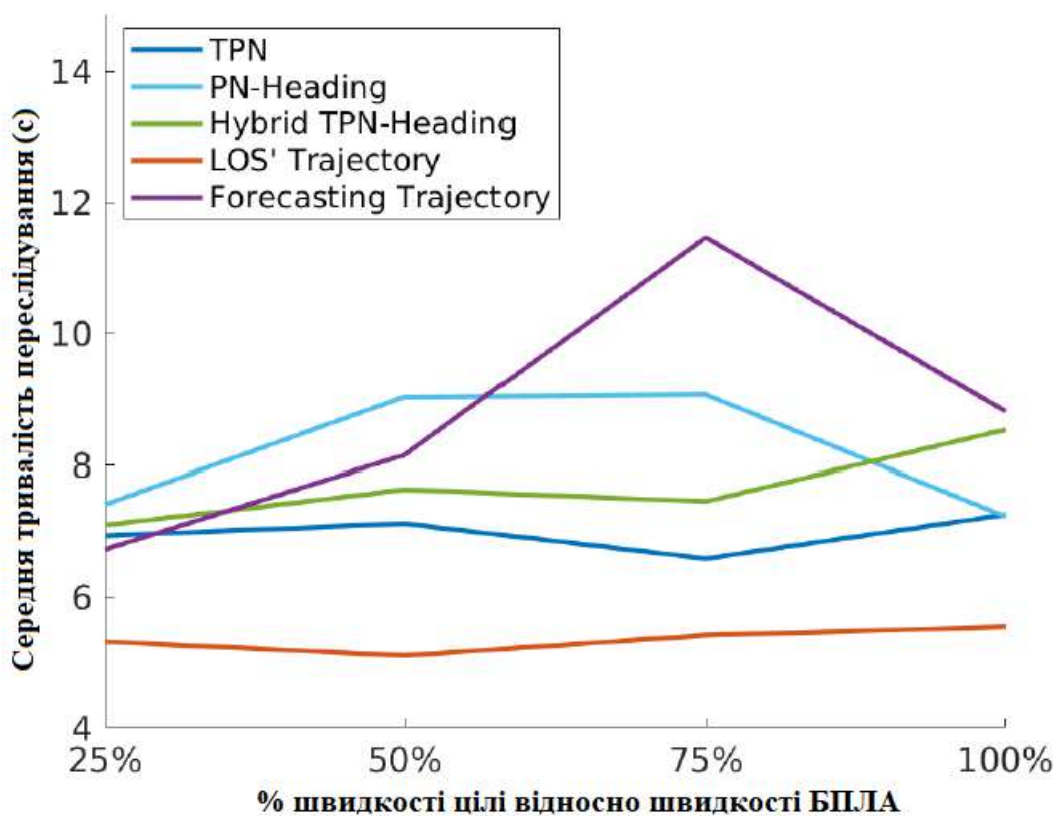


Рисунок 3.33 – Залежність тривалості переслідування від швидкості цілі для траєкторії цілі у формі вісімки

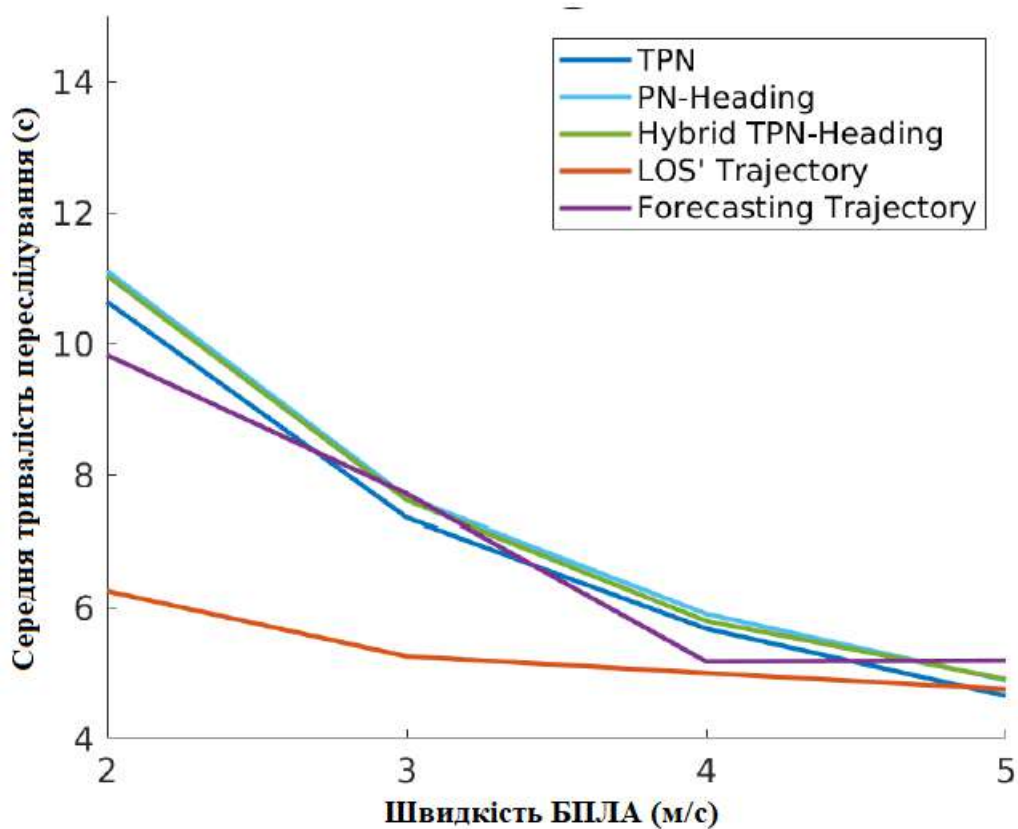


Рисунок 3.34 – Залежність тривалості переслідування від швидкості БПЛА для вузлової траєкторії цілі

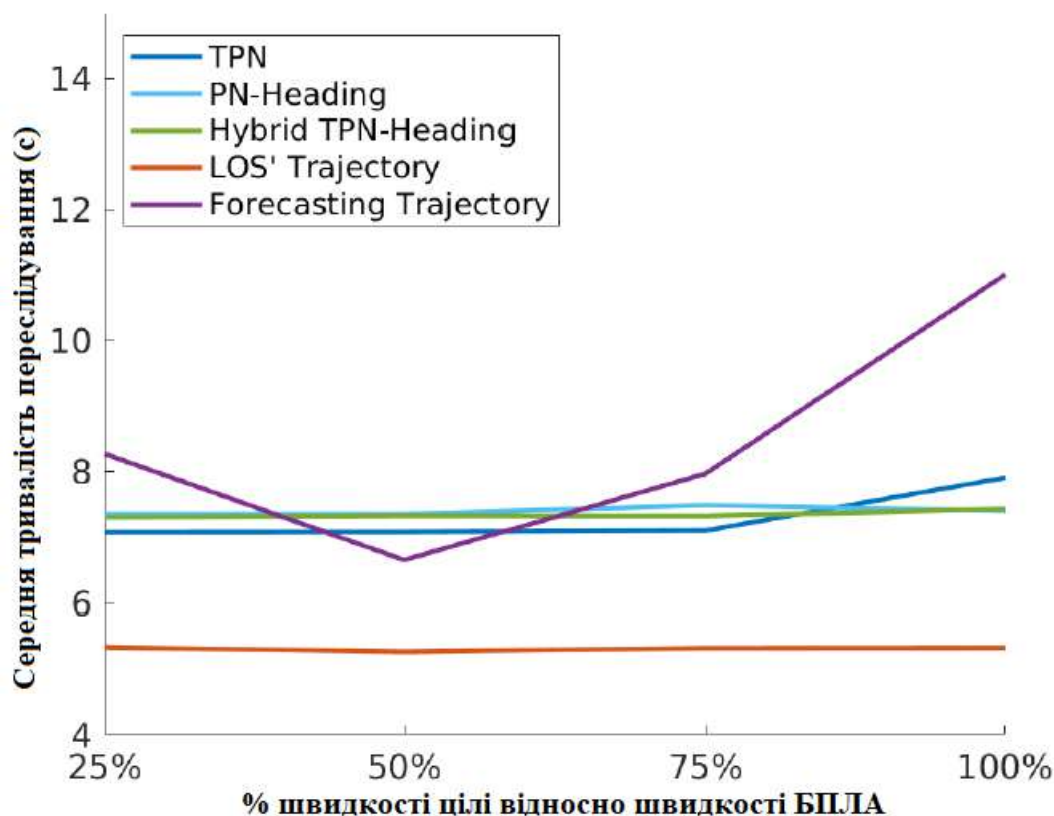


Рисунок 3.35 – Залежність тривалості переслідування від швидкості цілі для вузлової траєкторії цілі

Час переслідування загалом зменшився зі збільшенням швидкості БПЛА, хоча ця тенденція не є такою очевидною в деяких методах наведення на рисунку 3.30. У сценарії прямої траєкторії цілі, оскільки ціль має постійну швидкість, вона врешті-решт вийде з поля зору БПЛА, обмежуючи часове вікно, у якому можливе перехоплення. У TPN і PN-Heading все ще спостерігається сильна тенденція до зниження.

Вплив швидкості цілі на тривалість переслідування найбільше спостерігається у випадку прямої траєкторії цілі. Тут збільшення цільової швидкості викликало збільшення часу для PN-Heading і Hybrid TPN-Heading. Обидва ці методи використовують керування швидкістю повороту, щоб утримувати ціль у центрі зображення камери БПЛА, і, як показано на рис. 3.37, це може призвести до того, що БПЛА повертатиметься та «переслідуватиме» ціль під час проходження. Зі збільшенням швидкості цілі ймовірність успішного попадання на першому заході зменшується, що збільшує ймовірність такого маневру переслідування.

Траєкторія LOS, хоча й використовує згладжену версію LOS, що використовується для TPN, не демонструє такого сильного зменшення часу переслідування зі збільшенням швидкості БПЛА чи швидкості цілі. Ці результати насправді показують, що з усіх методів, траєкторія LOS має найменшу тривалість переслідування за кількома винятками. Однак у поєднанні з результатами на рисунку 3.18-3.20 видається більш імовірним, що ці часи є побічним результатом найімовірнішого влучення в ціль, яка ініціалізована ближче до початкової точки БПЛА.

3.5 Поведінка при переслідуванні

Далі було проемульовано траєкторію БПЛА для кожного методу відносно руху цілі в конфігураціях, обраних вручну для якісної оцінки.

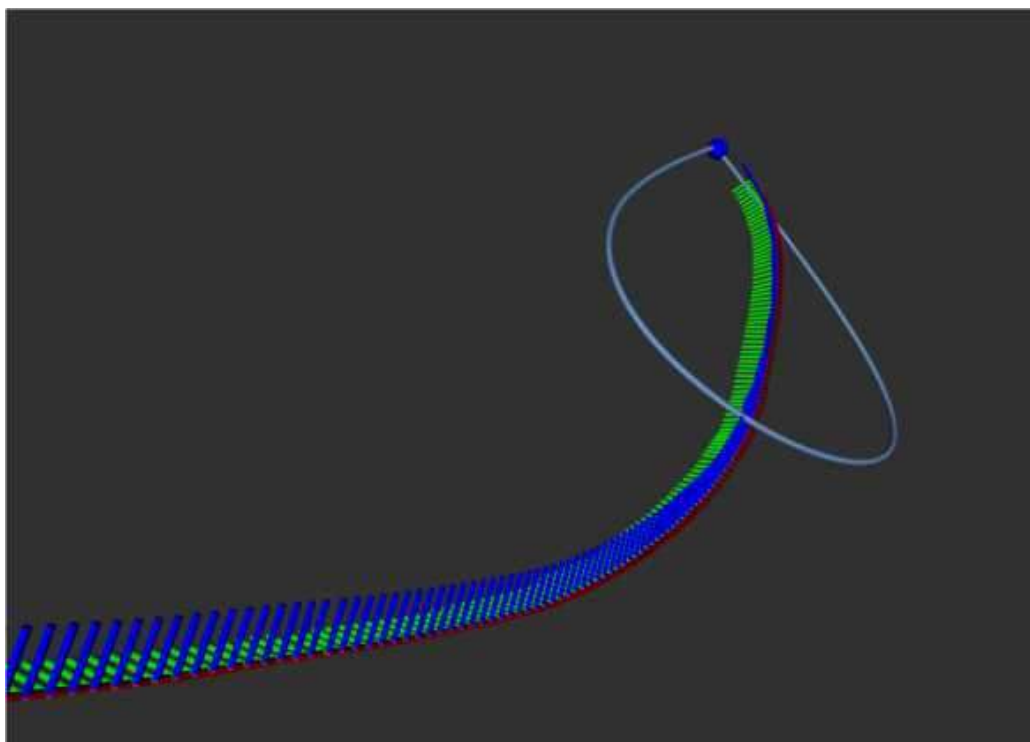


Рисунок 3.36 – Поведінка при методі TPN.

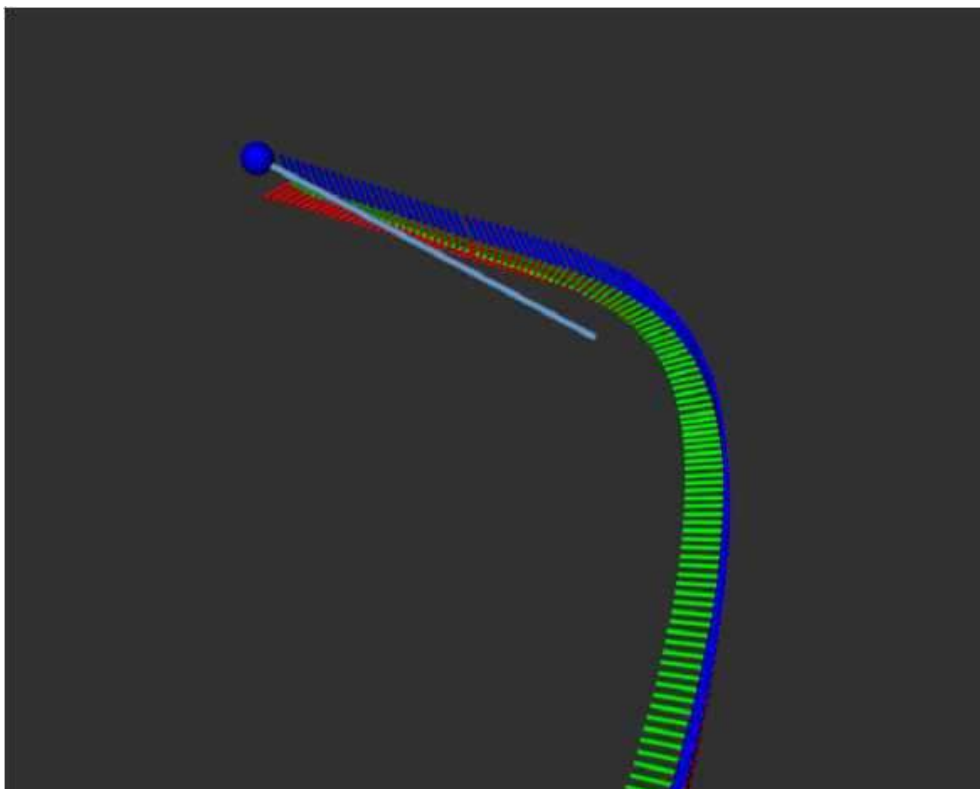


Рисунок 3.37 – Поведінка при методі PN.

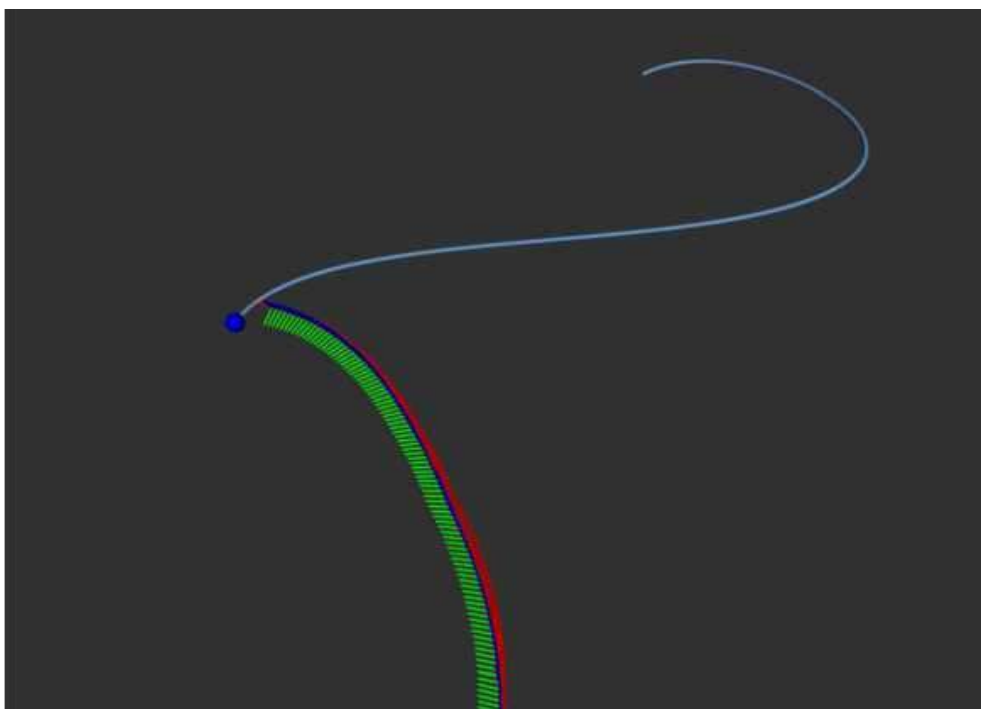


Рисунок 3.38 – Поведінка при гібридному курсі TPN.

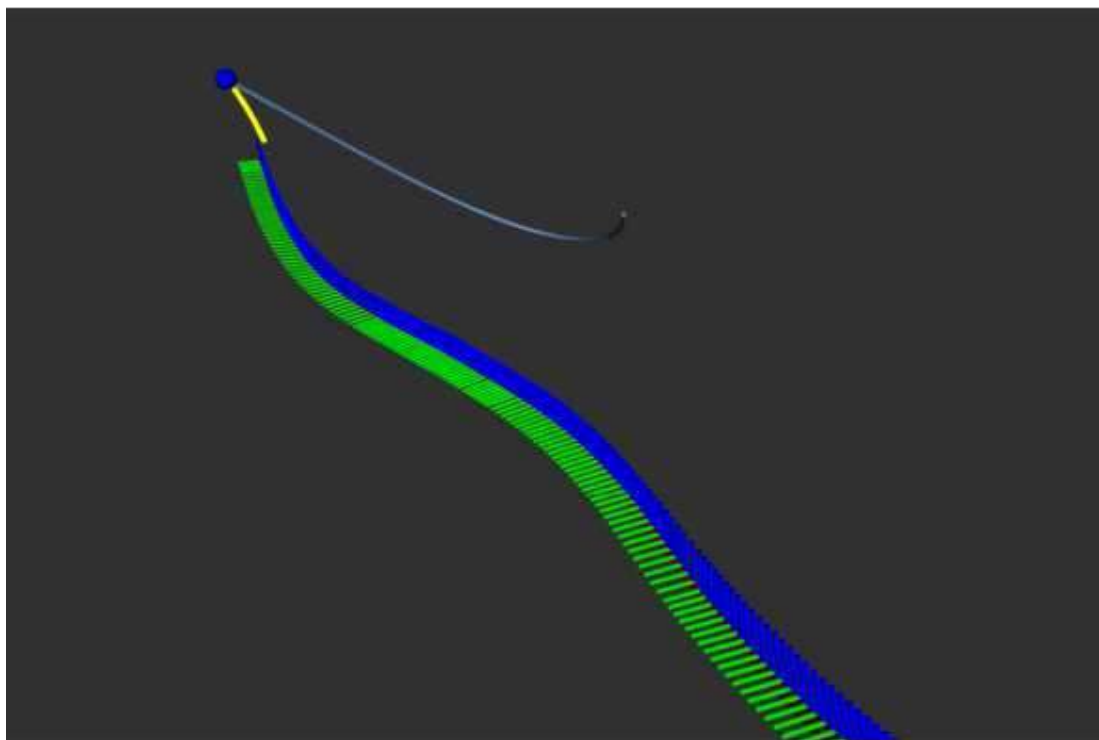


Рисунок 3.39 – Поведінка траєкторії LOS

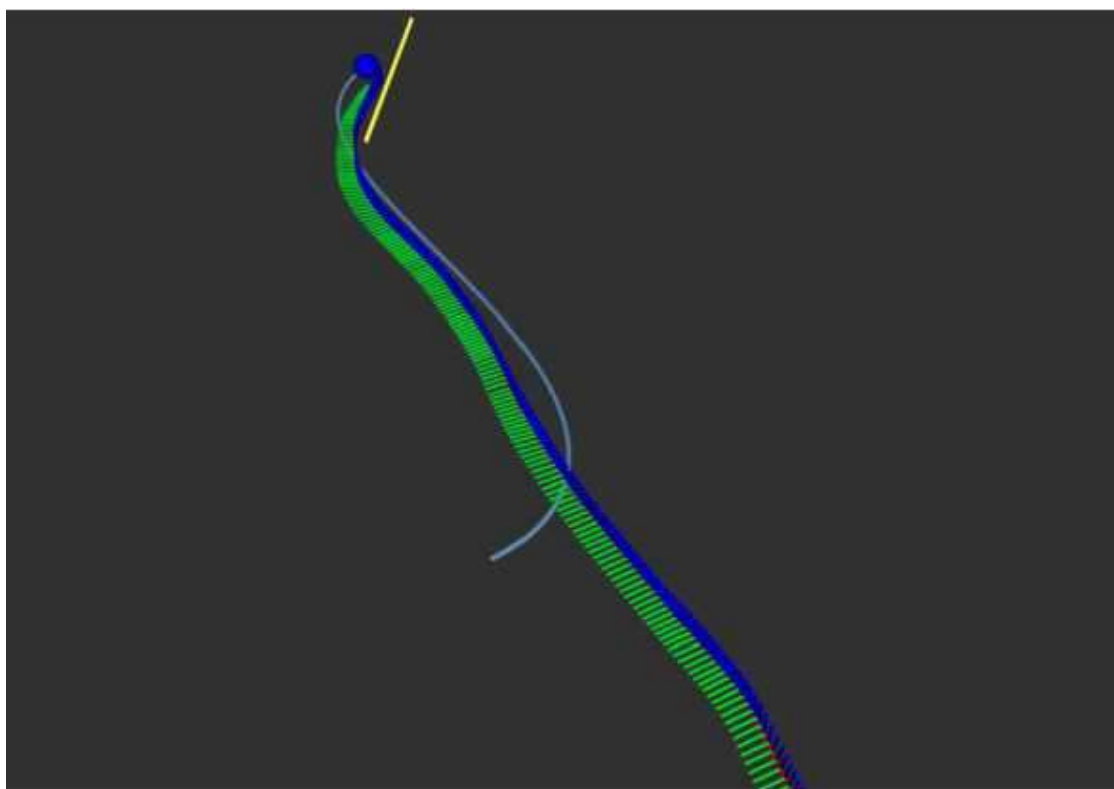


Рисунок 3.40 – Прогноз поведінки траєкторії.

На рис. 3.36 показано, як БПЛА відстежує рух цілі, коли вона опускається, а потім знову піднімається, коли рухається половиною траєкторії цифри-8. На рис. 3.37 показано, що БПЛА наближається до прямої траєкторії цілі, але робить різкий

поворот ліворуч за допомогою команди швидкості курсування, коли він проходить повз. Потім він реалізує PN, щоб зловити ціль. Рисунок 3.38 починається з руху, подібного до TPN, потім використовується керування курсом, щоб відхилитися до цілі, коли вона наближається, і курс стає більшим. На рисунку 3.39 показано більш плавний шлях, ніж TPN, в результаті згладжування команд PN і використання траєкторії. Траєкторія, згенерована в мітку часу, показану на зображенні, показана жовтим кольором. На малюнку 3.40 показана траєкторія, що імітує рух цілі, але зміщена в просторі за рахунок прогнозування руху цілі. Поточна прогнозована траєкторія показана жовтим кольором.

4 РОЗРОБКА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Алгоритмічні передумови для вибору елементів

Для того, щоб вибір апаратного забезпечення був адекватний задачам, які будуть стояти перед ним – було проведено огляд програмних алгоритмів на низькому рівні які мають виконуватися апаратним забезпеченням.

Виявлення рухомої цілі є дуже важливим завданням аналізу рухомих об'єктів. Для ефективного і точного виявлення рухомих об'єктів більш популярні три основні методи:

- алгоритм розрізнення кадрів;
- метод віднімання фону;
- техніка оптичного потоку.

Алгоритм розрізнення кадрів (FD) широко використовується для аналізу руху. Він має дуже ефективні результати при виявленні рухомих об'єктів. Алгоритм FD залежить від опорного кадру та поточного кадру, змін порогу та освітлення. Він дає центроїд (координати центру) рухомого об'єкта та площу рухомого об'єкта як вихідні дані. Використовуючи інформацію про центроїд, ми обчислюємо траєкторію бажаного відстежуваного рухомого об'єкта. Для фіксації відстежуваного об'єкта розраховуються кути руху двигунів відносно центрального положення рами.

Алгоритм FD використовує попередній кадр як опорний кадр і кадр у момент часу (t) як поточний кадр. Ці кадри вводяться в алгоритм FD. Береться різниця між опорним кадром і кадром-кандидатом або поточним кадром. Рухомі об'єкти виявляються повторно обробляються та передаються до функції відстеження для відстеження рухомих об'єктів. Після виявлення рухомого об'єкта отримане зображення повторно обробляється для уточнення виявленого об'єкта. Повторна обробка містить фільтрацію, порогове очищення, морфологічні операції, такі як корозія. Потім визначаються координати об'єкта і передаються в функцію стеження.

Алгоритм віднімання фону (BS) широко використовується для виявлення рухомих об'єктів і є дуже ефективним методом в умовах статичного фону та

статичної камери. Алгоритм BS має власне значення та застосування в аналізі руху, взаємодії людини з комп'ютером (HCI) та інтелектуальних системах керування. В аналізі руху найважливішою частиною є виявлення рухомої цілі. Мета полягає в тому, щоб відокремити потрібний рухомий об'єкт від фону у відео та використовувати його для деяких операцій, таких як сегментація, розпізнавання та відстеження. Алгоритм BS заснований на моделюванні динамічного фону та динамічному пороговому значенні. Техніка BS дуже чутлива до змін зовнішнього середовища, таких як вплив світла та інші фонові перешкоди.

Алгоритм BS використовує техніку FD для виявлення рухомих об'єктів. Попередній кадр із деяким динамічним моделюванням виступає фоном для поточного кадру. Розбіжність двох кадрів показує рухомі об'єкти. Потім застосовуємо деякі операції, тобто морфологічні операції, щоб уточнити зображення та чітко показати виявлену ціль. Він дуже чутливий до змін фону, тому динамічне моделювання фону використовується, щоб певною мірою подолати цей ефект. Ідеальна умова — коли фон статичний, це означає, що камера працює в статичних умовах. Подальший метод вибору динамічного порогового значення на основі віднімання фону поєднується з моделюванням динамічного фону на основі точного виявлення об'єкта для покращення результатів і посилення ефекту виявлення рухомого об'єкта.

Метод оптичного потоку описує процес руху пікселів, у якому оптичний потік використовується для отримання інформації про рух між кожними двома сусідніми кадрами. Відповідно до теорії оптичного потоку: оптичний потік базується на трьох припущеннях, і ці три припущення:

- припущення про значення сірого;
- градієнтне припущення;
- припущення гладкості.

Припущення гладкості використовується на місці, де перші два припущення не виконуються.

Відстеження рухомого об'єкта є важливим завданням під час аналізу рухомих об'єктів. Для ефективного відстеження рухомих тіл визначено багато алгоритмів:

- алгоритм відстеження зіставлення шаблонів;
- середній зсув;
- масштабне інваріантне перетворення ознак [24], [25];
- трекер Канаде-Лукаса-Томазі;
- фільтр Калмана.

Кожен із згаданих алгоритмів має свою важливість, властивості та обмеження.

4.2 Порядок роботи з апаратним забезпеченням

Для розуміння взаємодії апаратних компонентів системи було розроблено порядок обробки інформації на етапах проектування, тестування та роботи.

На платформі симуляції записується відео або використовується живе відео з камери для подальшої обробки. Потім це відео (записано/живе) обробляється в MATLAB для виявлення рухомого об'єкта та визначення його місцезнаходження. Положення об'єкта визначається у вигляді координат x та y , які потім перетворюються на горизонтальні та вертикальні кути, через які ціль переміщена відносно контрольної точки. Розраховані кути потім передаються в Proteus з MATLAB через віртуальний послідовний порт. Схема мікроконтролера в Proteus обробляє ці кути та генерує сигнал ШІМ відповідно до отриманих кутів. ШІМ застосовується до електронних контролерів швидкості для руху безколекторних двигунів або для сервоприводу для переміщення серводвигуна під кутом, отриманим від MATLAB. Цей процес показано на рис. 4.1.

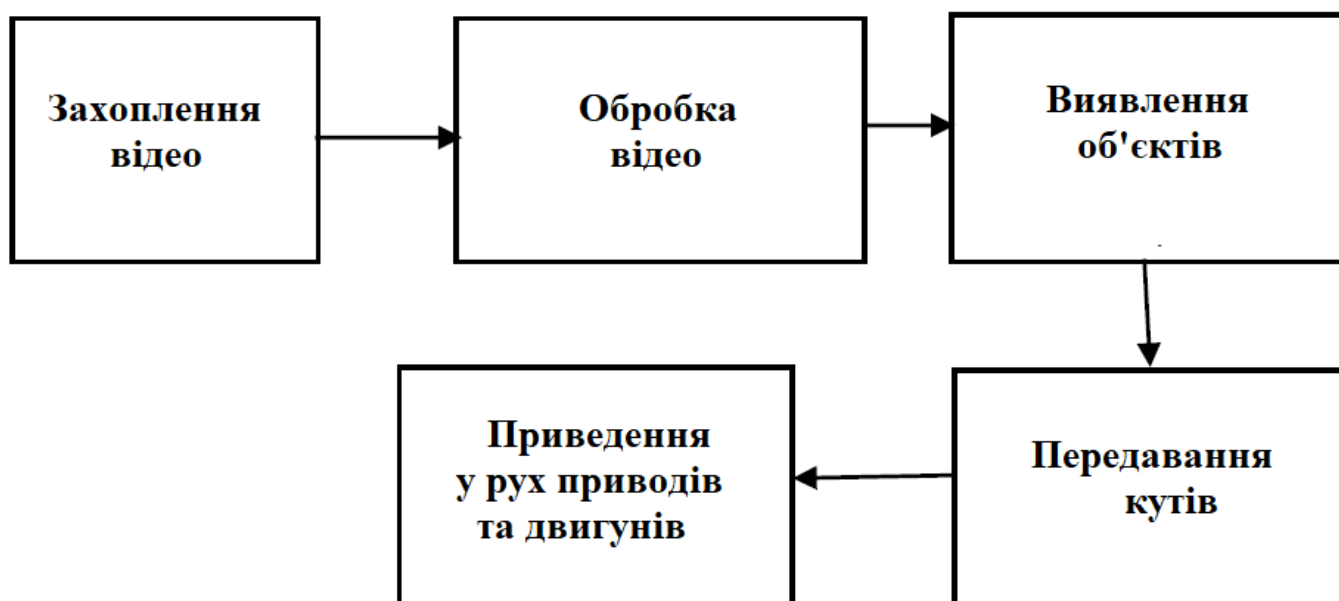


Рисунок 4.1 – Порядок роботи апаратної частини з відео

Перед впровадженням системи на апаратному забезпеченні було виконано моделювання всієї системи.

При моделюванні всієї апаратної частини системи використовувалися такі інструменти:

- Matlab;
- Proteus;
- драйвер віртуального послідовного порту Eltima;
- AVR studio 5
- HyperTerminal.

Matlab є потужним інструментом для обробки відео, а також для виконання обчислень. Для моделювання вся робота з обробки відео проводилася в MATLAB. За допомогою обробки відео об'єкт виявляється у відео в реальному часі, і його положення обчислюється в системі координат x та y відносно контрольної точки. Координати x , y перетворюються на горизонтальні та вертикальні кути через переміщений об'єкт.

Proteus — потужний інструмент для моделювання схем. При моделюванні системи мікроконтролерна схема зроблена в Proteus. Ця схема включає мікроконтролер (тобто Atmega16), два серводвигуни (первинна симуляція

проводилась для БПЛА дітакового типу), компілятор і осцилограф. Мікроконтролер генерує ШІМ-сигнал для керування кутовим положенням серводвигуна відповідно до кутів, отриманих з matlab. Повна схема показана на рис. 4.2.

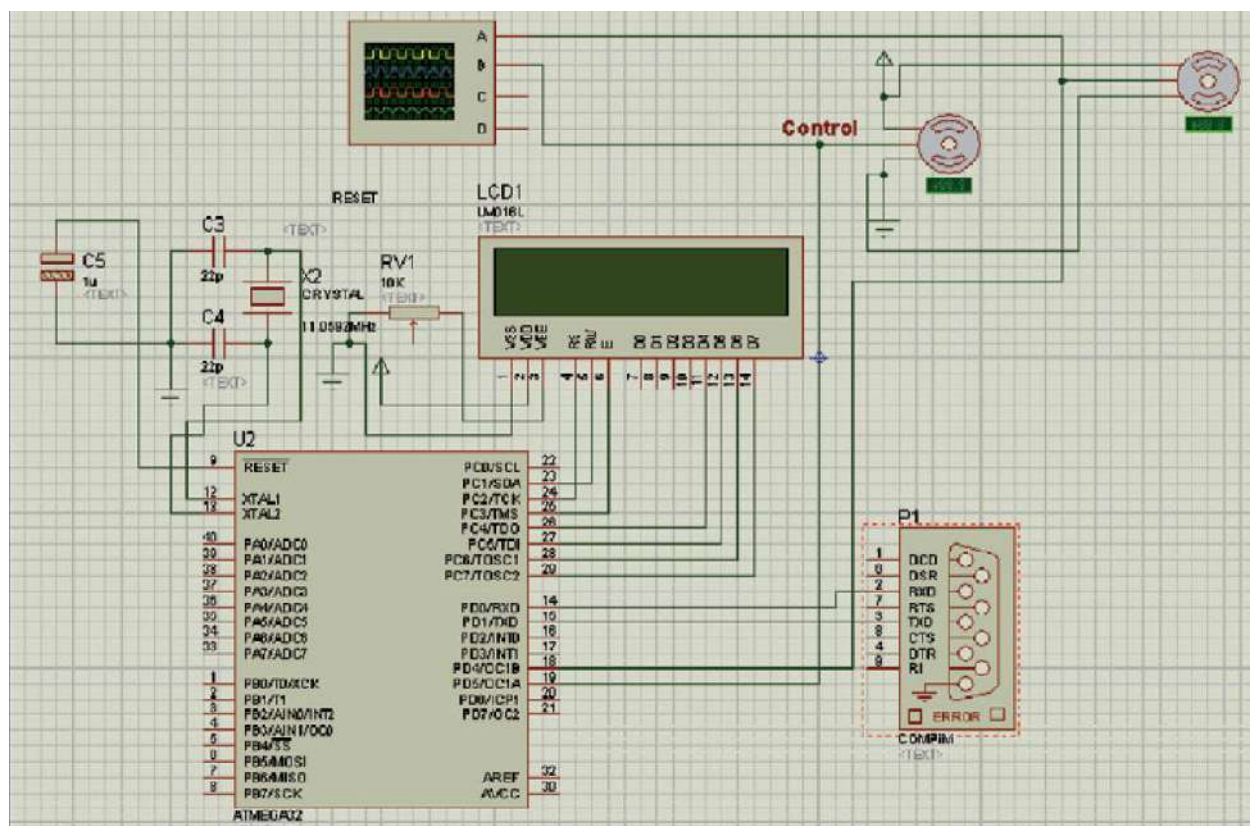


Рисунок 4.2 – Схема для моделювання в Proleus

На рис.4.2 видно, що компілятор підключений до мікроконтролера. Це використовується для встановлення зв'язку між терміналом комп'ютера або MATLAB і Proteus.

Драйвер віртуального послідовного порту — це програмне забезпечення, яке використовується для створення віртуальних портів на комп'ютері. Він створює порти парами, як показано на рис. 4.3. Він також показує реальні послідовні порти комп'ютера.

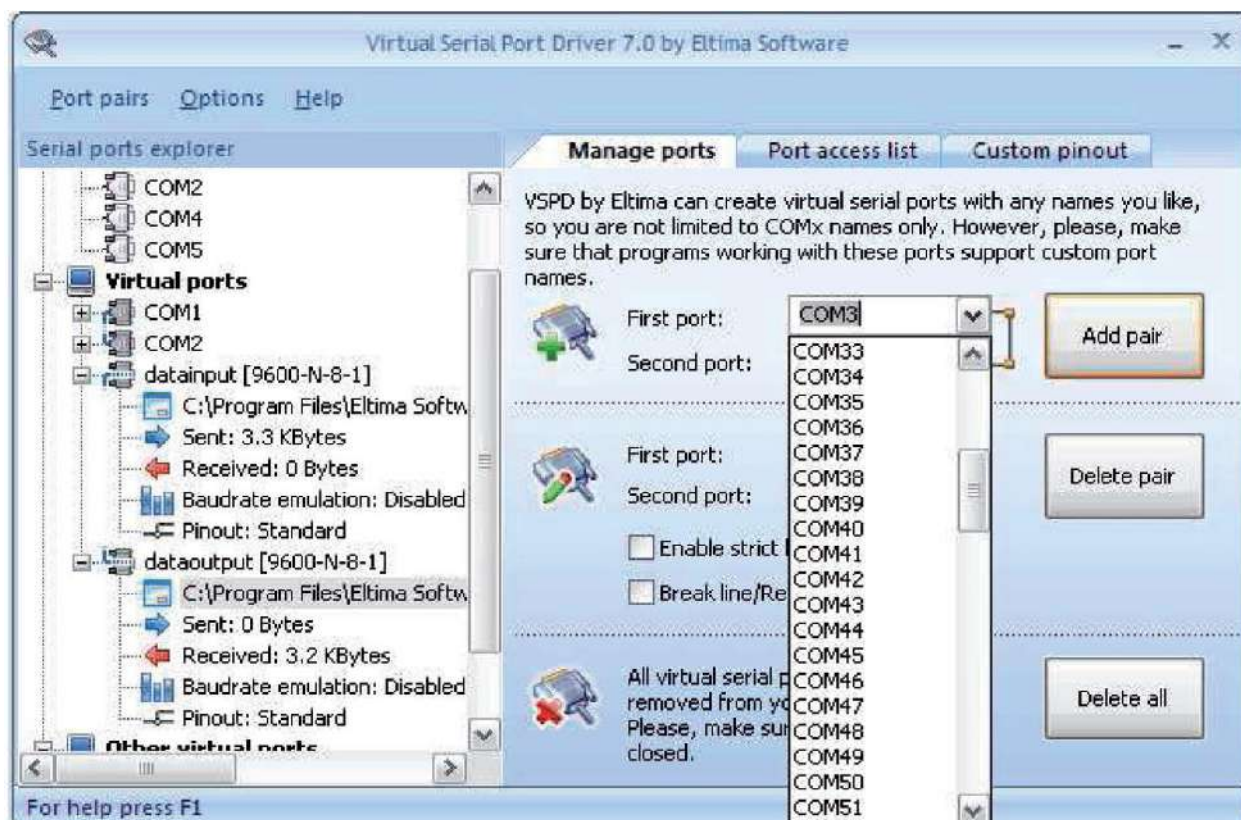


Рисунок 4.3 – Драйвер віртуального послідовного порту

AVR studio - це інструмент для програмування мікроконтролерів AVR. Його 5-та версія досить проста і зручна. В студії AVR розписаний код мікроконтролера для прийому даних з matlab і генерації ШІМ-сигналу для управління серводвигунами, підключеними до мікроконтролера.

Для встановлення послідовного зв'язку між мікроконтролером і комп'ютером використовується HyperTerminal.

4.3 Вибір апаратного забезпечення

Основними компонентами апаратної реалізації системи виявлення та супроводу рухомих цілей є:

- система повороту-нахилу;
- плата мікроконтролера;
- плата блоку живлення.

Система повороту-нахилу — це система, що складається з двох двигунів, один — двигун повороту, а інший — двигун нахилу, а також деяких інших компонентів для фіксації цих двигунів таким чином, щоб вони могли рухатися в площині $x-y$. Розглядається такий підхід, який притаманний БПЛА літокового типу, бо в рамках виконання диплому завадило часу адаптувати алгоритми керування рухом для БПЛА квадрокоптерного типу. Двигуни, які використовуються для створення системи Pan-Tilt, — це серводвигуни Hitec Dc. Серводвигуни дуже точні з точки зору положення та вбудованого механізму зворотного зв'язку.

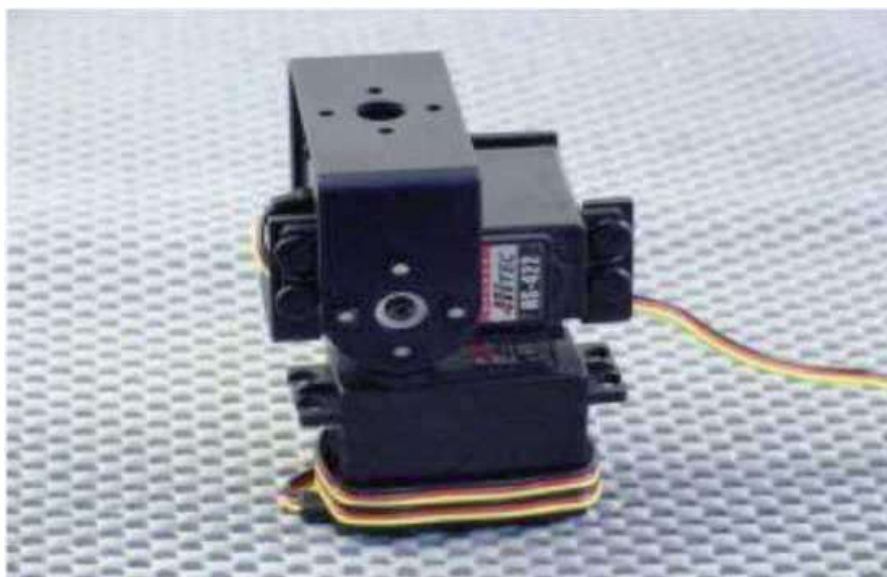


Рисунок 4.5 – Панорамно-нахильна система

Серводвигуни — це тип двигунів постійного струму з сервоприводом, який забезпечує точне кутове положення. Серводвигуни бувають різних видів, але розглядалися лише трипровідні серводвигуни.

Серводвигун - це збірка наступних чотирьох елементів:

- звичайний двигун постійного струму;
- блок редуктора;
- пристрій визначення положення (потенціометр);
- схема керування.

Функція сервоприводу полягає в отриманні керуючого сигналу від мікроконтролера або від іншого керуючого пристрою, наприклад таймера IC555, який представляє бажане вихідне положення вала серводвигуна. Коли серводвигун

отримує керуючий сигнал від керуючого пристрою (наприклад, Atmega16), сервомеханізм подає живлення на свій двигун постійного струму, поки його вал не повернеться в потрібне положення. Серводвигун має пристрій визначення положення (потенціометр), який діє як зворотний зв'язок із підсилювачем виявлення помилок. Він визначає положення обертання вала двигуна, перетворює його в напругу та подає на підсилювач виявлення помилок для знаходження різниці між фактичним і бажаним положенням вала. Різниця напруг визначає, у який бік має обертатися двигун. Діапазон руху сервомотора становить 0-180 градусів.

Серводвигун має трипровідну систему підключення (рис.4.5).

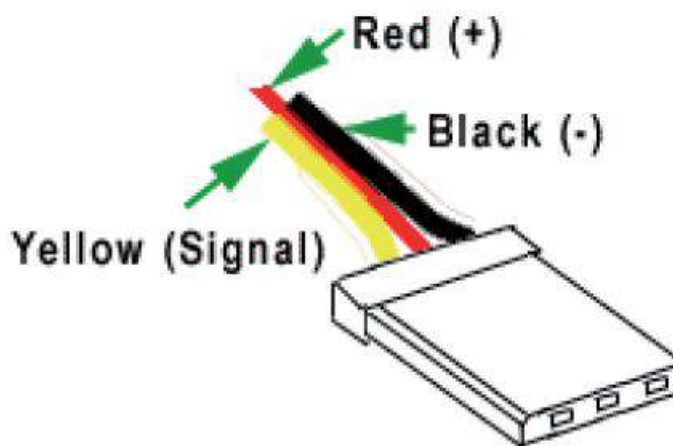


Рисунок 4.5 – Конфігурація проводів серводвигунів

Червоний провід використовується для забезпечення позитивного живлення двигуна. Чорний дріт використовується для забезпечення негативного живлення (маси) двигуна. Провід третього кольору використовується для подачі керуючого сигналу на серводвигун для контролю положення його валу.

Сигнал керування в основному є сигналом ШІМ. Період часу цього керуючого сигналу повинен становити 20 мс, тобто частота сигналу ШІМ повинна бути 50 Гц.

Ширина імпульсу сигналу визначить положення валу (необхідне положення) серводвигуна. Діапазон ширини імпульсу ШІМ-сигналу різний для різних серводвигунів, напр. для серводвигуна Futaba це від 1 мс до 2,5 мс. Для Hitec hs422. Розрахована ширина імпульсу, необхідна для повного діапазону (тобто від 0 до 180 градусів), становить від 0,3 мс до 2,6 мс для серводвигуна, який використовується в

проекті. При ширині імпульсу 0,3 мс вал двигуна знаходиться в нульовому положенні (надзвичайний підйом). Збільшуючи ширину імпульсу від 0,3 мс до 2,6 мс, серводвигун починає рух за годинниковою стрілкою, а якщо вона зменшується з 2,6 мс до 0,3 мс, серводвигун рухатиметься проти годинникової стрілки. Тепер для переміщення вала двигуна в певне положення була розроблена загальна формула, яка реалізована в мікроконтролері. Значення OCR обчислюються за допомогою формули та кутів, отриманих із matlab. Мікроконтролер використовує ці значення OCR (вихідний регістр захоплення) і генерує сигнал ШІМ із шириною імпульсу відповідно до необхідного кутового положення.

Плата мікроконтролера AVR складається з наступних компонентів:

- Atmega16 (мікроконтролер);
- два DB9 (9 контактний роз'єм);
- MAX232 (перетворювач рівнів);
- 16x2 LCD (дисплей монохромний рідиннокристалічний).

Atmega16 — це 40-контактний мікроконтролер, що належить до сімейства мікроконтролерів Atmega. Як ми вже згадували раніше, нам потрібен керуючий сигнал для керування положенням вала серводвигуна. Для цієї мети ми використали atmega16 для отримання керуючого ШІМ-сигналу відповідно до кута, на який ми хочемо перемістити вал двигуна.

Робоча частота мікроконтролера відіграє дуже важливу роль у генерації сигналу ШІМ для керування серводвигуном. Мікроконтролер надає два варіанти вибору робочої частоти: один — це його внутрішній генератор, а інший — зовнішній кристалічний генератор, який можна приєднати до мікроконтролера, коли це необхідно. Внутрішній генератор мікроконтролера має чотири варіанти, тобто 1 МГц, 2 МГц, 4 МГц і 8 МГц, будь-який з них може бути використаний залежно від вимоги. Якщо потрібна частота вище 8 МГц, до мікроконтролера слід приєднати зовнішній генератор. Зовнішній генератор у діапазоні від 8 МГц до 16 МГц можна використовувати для atmega16. Atmega16 не може підтримувати вище 16 МГц.

Вибір внутрішнього або зовнішнього генератора залежить від вимоги. Внутрішній генератор не настільки стабільний, як зовнішній кристалічний

генератор, тому для отримання високої точності використовується зовнішній генератор.

Для мінімально можливого керування сервомотором мікроконтролер повинен працювати на високій частоті. На 8 МГц сервопривод можна контролювати мінімум до 0,7 градуса. У той час як на 1 МГц мінімальний градус становитиме 5 градусів.

Рідкокристалічний дисплей (РК) має два типи регістрів, тобто регістри даних і команд. Регістри даних використовуються для зберігання даних, які відображаються на РК-дисплеї, тоді як регістри команд використовуються для зберігання команд на РК-дисплеї. Команда — це вказівка, надана РК-дисплею для виконання таких дій, як очищення екрана, встановлення положення курсору, керування дисплеєм тощо. РК-дисплей, який використовується, показаний на малюнку 5.7.

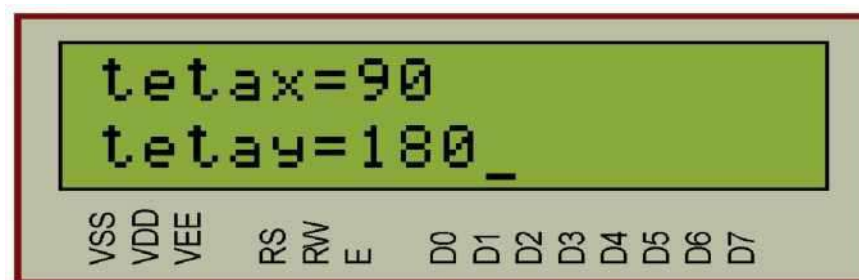


Рисунок 4.6 – LCD 16x2

MAX232 — це інтегральна схема, вперше створена Maxim Integrated Products, яка перетворює сигнали з рівня RS-232 на TTL-сумісні цифрові логічні схеми. MAX232 є подвійним драйвером/приймачем і зазвичай перетворює сигнали RX, TX, CTS і RTS. MAX232 використовується в схемі, де послідовний зв'язок має здійснюватися з мікроконтролером. MAX232 показано на малюнку 5.8.

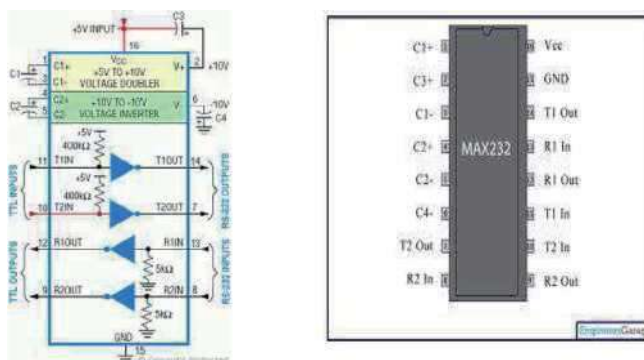


Рисунок 4.7 – Внутрішня схема Max232 і доступний пакет на ринку

DB9 - це дев'ятиконтактний роз'єм, який використовується для підключення кабелю послідовного порту від комп'ютера до мікроконтролера з метою послідовної передачі даних від персонального комп'ютера до мікроконтролера. Він має 9 дев'ять різних контактів, імена, призначення та конфігурація яких показані вище. DB9 показано на малюнку 5.9.

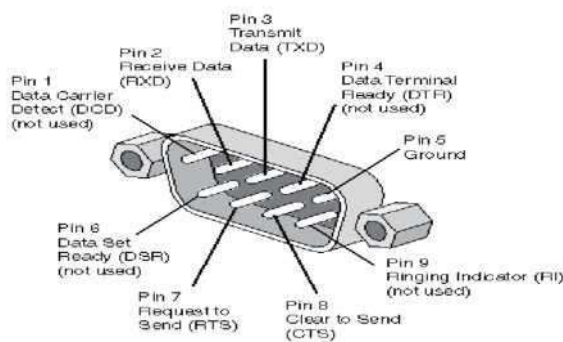


Рисунок 4.8 – 9-контактний роз'єм DB9

Для виявлення та відстеження рухомих цілей у режимі реального часу камера використовується як датчик зору. З камери знімається відео в реальному часі регіону, де знаходиться ціль, для обробки. Для зйомки живого відео використовується Logitech HD Pro Webcam C920. Ця камера має USB-інтерфейс для підключення до комп'ютера і проста у використанні. Це якісна веб-камера порівняно з іншими веб-камерами цього типу. Відео знімається цією камерою, а потім обробляється в MATLAB за допомогою комп'ютерного процесора. Використана камера показана на рис.4.9.



Рисунок 4.9 – Logitech HD Pro Webcam C920.

Серводвигуни працюють від джерела живлення 5 В постійного струму. Для забезпечення живлення 5 В серводвигуна була виготовлена плата живлення. Він складається з регулятора напруги 5 В (7805), кнопки ввімкнення/вимкнення та з'єднувальних контактів. З цієї плати подається живлення на обидва мотори. Є три контакти для кожного двигуна, один для +ve, 2-й - заземлення, а третій - ШІМ-штирь від мікроконтролера.

Принципова схема силової плати розроблена в програмі Proteus наведена на рис.4.10.

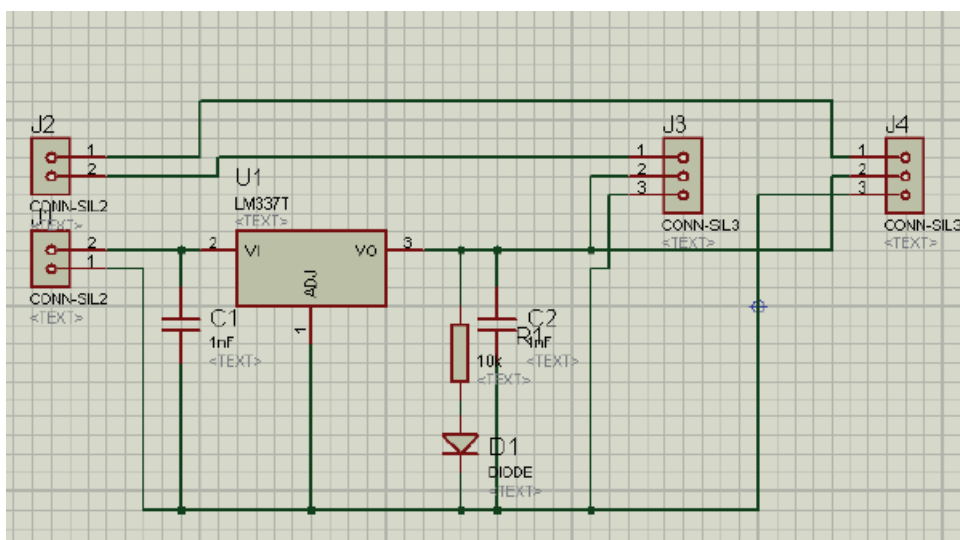


Рисунок 4.10 – Принципова схема плати живлення

Компонування силової плати в програмі Proteus наведено на рис.4.11.

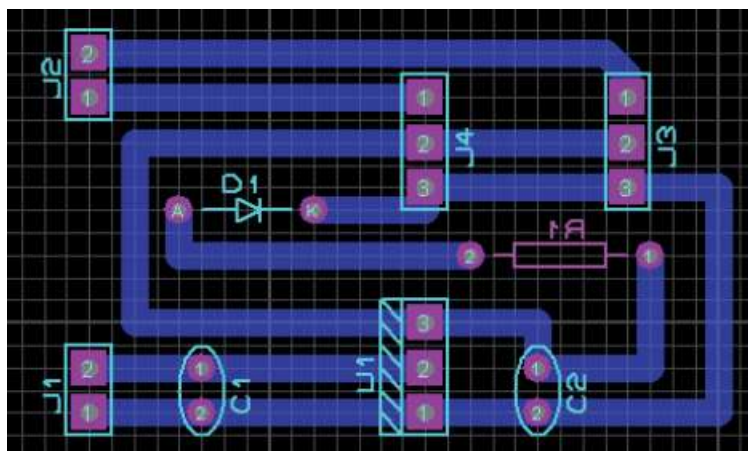


Рисунок 4.11 – Конструкція плати живлення

Компонування плати керування на базі мікроконтролера яку розроблено в програмі Proteus наведено на рис.4.12.

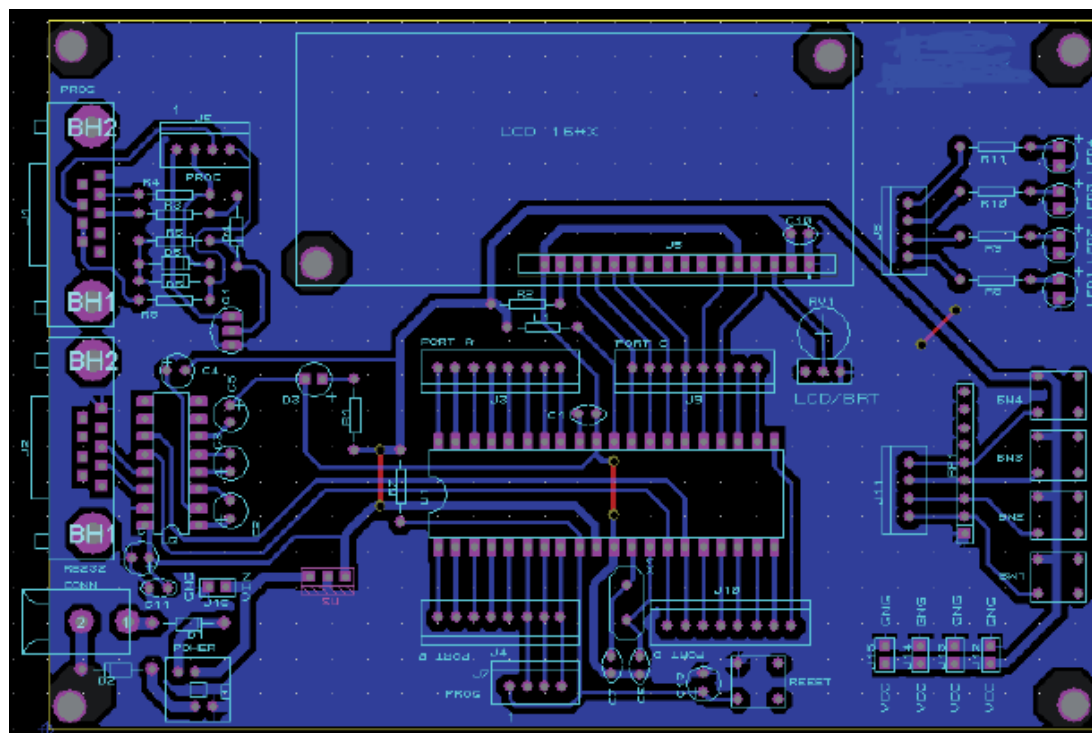


Рисунок 4.12 – Компонування плати керування в ПЗ Proteus

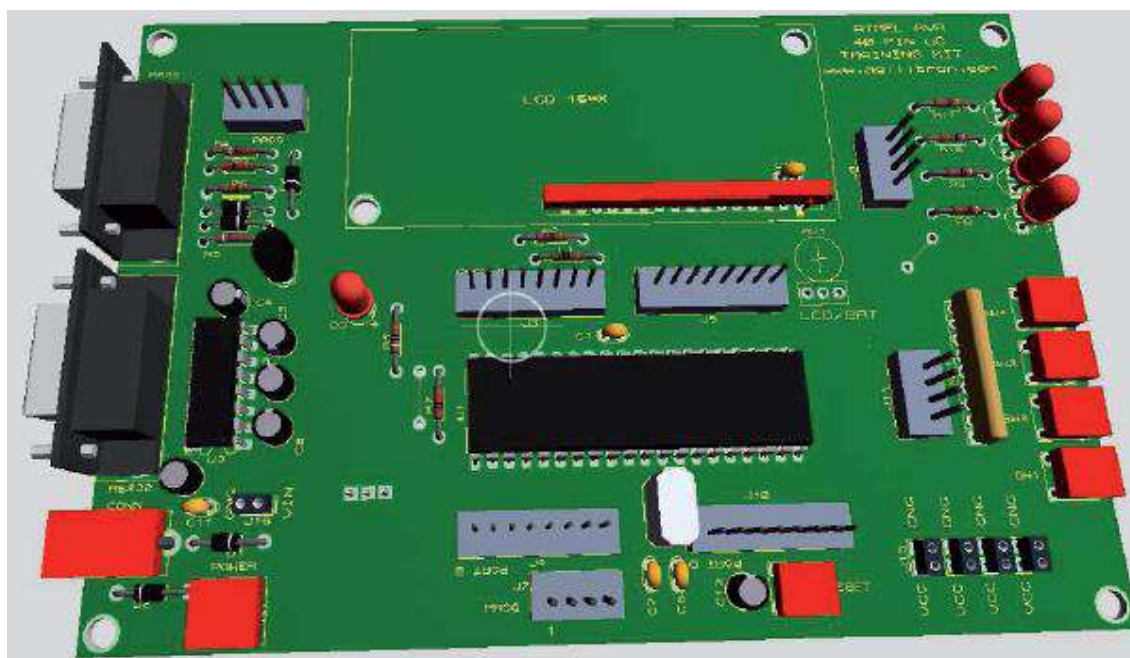


Рисунок 4.13 – Тривимірна схема плати керування

4.4 Відпрацювання роботи апаратної частини

Відстежуваний об'єкт блокується за допомогою інформації системи керування та функції стеження. З інформації `centroid` знаходимо кути від `reference` до `center` кадру (x_0 , y_0). Розраховані кути потім передаються на плату мікроконтролера, яка відповідно керує рухом двигунів.

Нехай обчислені наразі кути дорівнюють $T1$ і $T2$, де $T1$ — горизонтальний кут, а $T2$ — вертикальний кут. T_x і T_y — попередні горизонтальний і вертикальний кути відповідно. Потім кути руху обчислюються за допомогою наведених нижче рівнянь.

$$T_{mx}=T1-T_x \quad (4.1)$$

$$T_{my}=T2-T_y \quad (4.2)$$

де T_{mx} . і T_{my} – горизонтальний кут двигуна та вертикальний кут двигуна відповідно.

Ці кути передаються на плату мікроконтролера через послідовний зв'язок.



Рисунок 4.14 – Схема операцій

Алгоритм відстеження видає положення потрібного рухомого об'єкта як вихідні дані. Отримане положення порівнюється з еталонним положенням (центром кадру) і обчислюються кути. Тут обчислюються два кути, тобто горизонтальний кут і вертикальний кут. Горизонтальний кут показує рух двигуна панорамування, а вертикальний кут показує рух двигуна нахилу. Ціль замикається на основі руху двигунів. Координати (x_0, y_0) – центр кадру, (x_1, y_1) – центральне положення відстежуваного об'єкта, D – відстань між центром кадру та центром камери, dx і dy – відстані від базової позиції до центру об'єкта вздовж осі x та осі y відповідно, $T1$ – це горизонтальний або поворотний кут двигуна, а $T2$ – вертикальний або нахилний кут двигуна. Кути обчислюються за допомогою рівнянь:

$$T1 = \arctan\left(\frac{dx}{D}\right) \quad (4.3)$$

$$T2 = \arctan\left(\frac{dy}{D}\right) \quad (4.4)$$

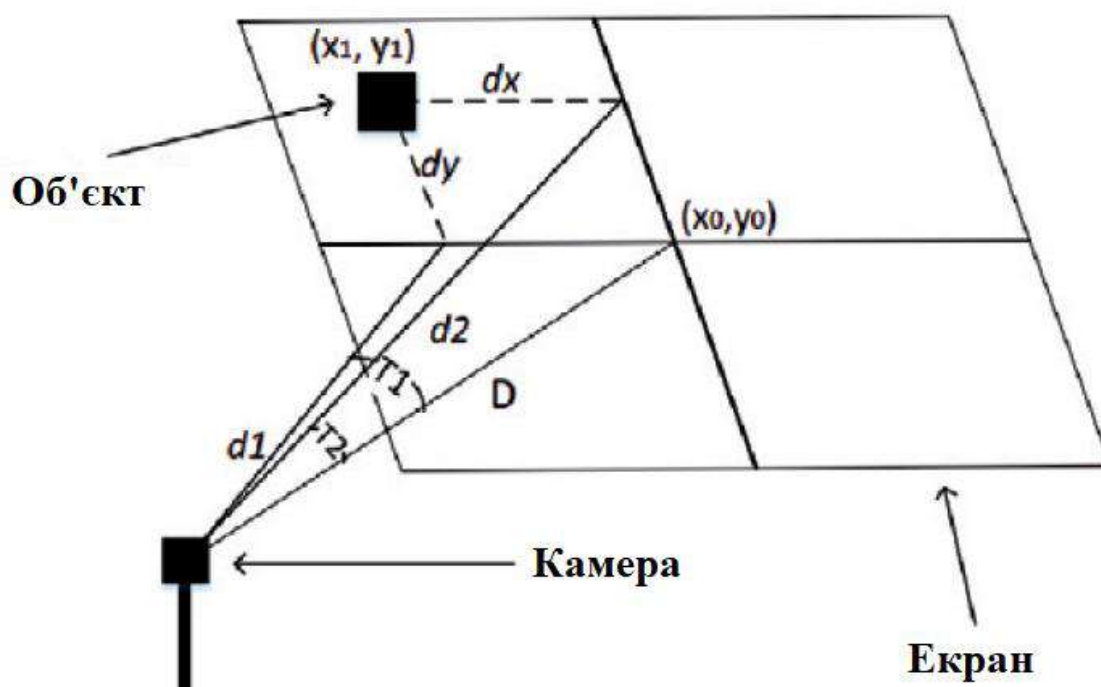


Рисунок 4.15 – Обчислення кутів

Система управління складається з трьох етапів: послідовний зв'язок, контрольний сигнал і рух двигунів.

Кути, розраховані за допомогою рівнянь 4.3 і 4.4, послідовно передаються на плату мікроконтролера. В інтерфейсі симуляції використовуються віртуальні послідовні порти та proteus. Створюються віртуальні послідовні порти, дані (кути) передаються послідовно через ці порти та отримуються на платі мікроконтролера, створеної в proteus. На апаратній платформі кути передаються через послідовний порт до плати мікроконтролера.

Плата мікроконтролера генерує сигнали широтно-імпульсної модуляції (ШІМ) для руху серводвигунів. ШІМ-сигнали генеруються за допомогою значень вихідного регістру порівняння (OCR – output compare register). Значення OCR отримані на основі отриманих кутів.

$$Y = 2.434 * X + 95.978 \quad (4.5)$$

Y — це значення OCR мікроконтролера,

X — кут, отриманий послідовним зв'язком.

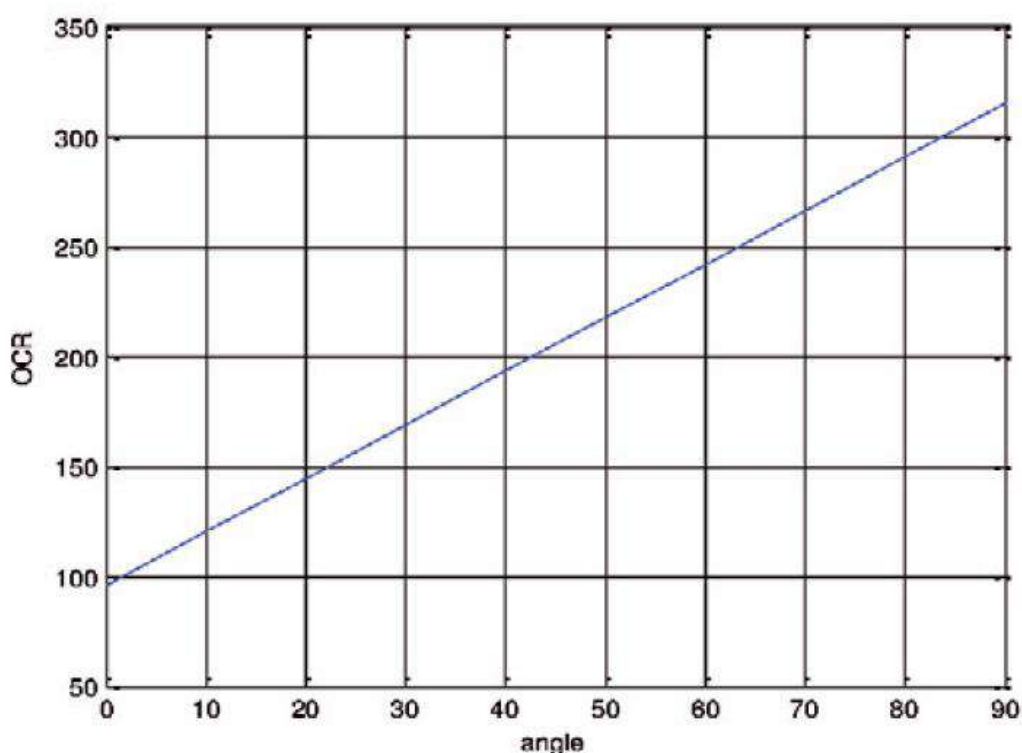


Рисунок 4.16 – Розрахунок OCR

Згенеровані ШІМ-сигнали подаються як керуючий вхід до серводвигунів (двигунів повороту та нахилу). Серводвигуни переміщуються в певні позиції відповідно до керуючого сигналу.

ВИСНОВКИ

У кваліфікаційній роботі магістра було розроблено та реалізовано різноманітні алгоритми наведення в змодельованому та реальному середовищах. Показники, які використовуються для оцінки ефективності кожного методу, включають частоту влучень першого проходу та тривалість переслідування. Методи наведення LOS були отримані з алгоритму справжньої пропорційної навігації, який взято з літератури з наведення ракет. Реалізація на квадрокоптері забезпечує більший контроль і маневреність за рахунок додаткових ступенів свободи.

Усі ці методи спиралися лише на інформацію з монокулярної RGB-камери на борту квадаторного БПЛА. Камера була як жорстко закріплена на рамі БПЛА, так і була встановлена на підвісі та встановлена так, щоб вона була направлена вперед відносно курсу БПЛА. У змодельованій системі зір було змодельовано реалістично, так що маленька ціль на великій відстані страждала від шумової оцінки LOS через дискретизацію пікселів. Тому методи траєкторії покладалися на згладжування LOS, щоб уникнути генерації траєкторій із великою варіацією.

Ця робота відкриває питання щодо того, які алгоритми наведення можна розробити для квадрокоптерів, використовуючи лише зображення монокулярної камери RGB. Подальший розвиток гібридного методу, що включає як пропорційну навігацію, так і контроль курсу, може призвести до кращої продуктивності. Це може призвести до підвищення частоти влучень, але також до меншої тривалості переслідування, оскільки потреба «переслідувати» ціль може зникнути, оскільки БПЛА утримує ціль у полі зору під час початкового заходу на посадку. Крім того, певні випадки збою алгоритмів, представлених в роботі, свідчать про те, що більш точні контролери можуть покращити продуктивність.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Fred P Adler. Missile guidance by three-dimensional proportional navigation. *Journal of Applied Physics*, 27(5):500-507, 1956.
2. Uday S Shukla and Pravas R Mahapatra. The proportional navigation dilemma-pure or true? *IEEE Transactions on Aerospace and Electronic Systems*, 26(2):382-392, 1990.
3. Markus Ryll, John Ware, John Carter, and Nick Roy. Efficient trajectory planning for high speed flight in unknown environments. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 732-738. IEEE, 2019.
4. Neil F Palumbo, Ross A Blauwkamp, and Justin M Lloyd. Basic principles of homing guidance. *Johns Hopkins APL Technical Digest*, 29(1):25-41, 2010.
5. Business Insider Intelligence. Commercial unmanned aerial vehicle (uav) market analysis - industry trends, forecasts and companies, Feb 2020.
6. Sheena McKenzie and Gianluca Mezzofiore. Police hunt drone pilots in unprecedented gatwick airport disruption, Dec 2018.
7. Harry Pettit For Mailable. Drone-catching drones armed with nets hit winter olympics, Feb 2018.
8. Russell Brandom. Watch anduril's anvil take down an off-the-shelf drone in midflight, Oct 2019.
9. F. Chaumette and S. Hutchinson. Visual servo control, part ii: Advanced approaches. *IEEE Robotics and Automation Magazine*, 14(1):109-118, March 2007.
10. Seth Hutchinson, Gregory D Hager, and Peter I Corke. A tutorial on visual servo control. *IEEE transactions on robotics and automation*, 12(5):651-670, 1996.
11. F. Chaumette and A. Santos. Tracking a moving object by visual servoing. *IFAC Proceedings Volumes*, 26(2):643-648, 1993.
12. Peter I Corke and Seth A Hutchinson. Real-time vision, tracking and control. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on*

Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), volume 1, pages 622-629. IEEE, 2000.

13. Neil F Palumbo, Ross A Blauwkamp, and Justin M Lloyd. Modern homing missile guidance theory and techniques. Johns Hopkins APL technical digest, 29(1):42-59, 2010.

14. Martin Stolle and Christopher G Atkeson. Policies based on trajectory libraries. In Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., pages 3344-3349. IEEE, 2006.

15. Emilio Frazzoli, Munther A Dahleh, and Eric Feron. Robust hybrid control for autonomous vehicle motion planning. In Proceedings of the 39th IEEE Conference on Decision and Control, volume 1, pages 821-826. IEEE, 2000.

16. Debadeepta Dey, Tian Yu Liu, Boris Sofman, and James Andrew Bagnell. Efficient optimization of control libraries. In Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012.

17. Dmitry Berenson, Rosen Diankov, Koichi Nishiwaki, Satoshi Kagami, and James Kuffner. Grasp planning in complex scenes. In 2007 7th IEEE-RAS International Conference on Humanoid Robots, pages 42-48. IEEE, 2007.

18. Charles Richter, Adam Bry, and Nicholas Roy. Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In Robotics Research, pages 649-666. Springer, 2016.

19. Fei Gao, William Wu, Yi Lin, and Shaojie Shen. Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial. In 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 344-351. IEEE, 2018.

20. Gabriel Hoffmann, Steven Waslander, and Claire Tomlin. Quadrotor helicopter trajectory tracking control. In AIAA guidance, navigation and control conference and exhibit, page 7410, 2008.

21. Daniel Mellinger, Nathan Michael, and Vijay Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. The International Journal of Robotics Research, 31(5):664-674, 2012.

22. Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In 2004IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), volume 3, pages 2149-2154. IEEE.

23. Lorenz Meier, Daniel Agar, Beat Kiing, Julian Oes, Thomas Gubler, Anton Babushkin, px4dev, Roman Bapst, Mark Charlebois, Matthias Grob, Andreas Daniel Antener, James Goppert, Andrew Tridgell, Paul Riseborough, Dennis Mannhart, Nuno Marques, Mathieu Bresciani, Mark Whitehorn, Kabir Mohammed, Simon Wilks, Sander Smeets, Pavel Kirienko, Martina Rivizzigno, Mark Sauder, David Sidrane, JohanJansen, Julien Lecoeur, PX4 Build Bot, Don Gagne, and Daniele Pettenuzzo. Px4/firmware: v1.11.0 beta 2, May 2020.

24. Seok-Wun Ha, Yong-Ho Moon, "Multiple Object Tracking Using SIFT Features and Location Matching", International Journal of Smart Home Vol.5, No. 4, October, 2011.

25. David. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision 60(2), 91-110, 2004.

ДОДАТОК А – ПРЕЗЕНТАЦІЯ

Національний університет “Запорізька Політехніка”
Кафедра інформаційних технологій електронних засобів

Магістерська робота на тему:

Дослідження та програмна реалізація методів наведення БПЛА квадрокоптерного типу на рухомі об'єкти під час польоту



Студент групи РТ-612м

Моїсєєв В.Є.

Науковий керівник к.т.н., доц.

Малий О.Ю..

1

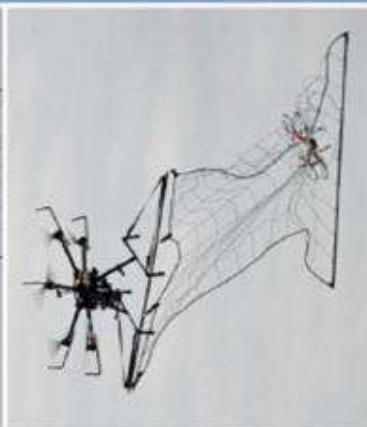
Мета, об'єкт, предмет

Мета роботи – дослідження методів та розробка програмно-апаратних алгоритмів наведення БПЛА на рухомі повітряні об'єкти для захисту територій при несанкціонованому проникненні.

Об'єкт дослідження: технологія комп'ютерного зору для обробки зображень з камер БПЛА з метою виявлення рухомих об'єктів та програмно-апаратна реалізація систем відстеження рухомих об'єктів.

Предмет дослідження: методи наведення БПЛА квадрокоптерного типу на рухомі об'єкти під час польоту.

Протидія насанкціонованим польотам БПЛА



Постановка задач

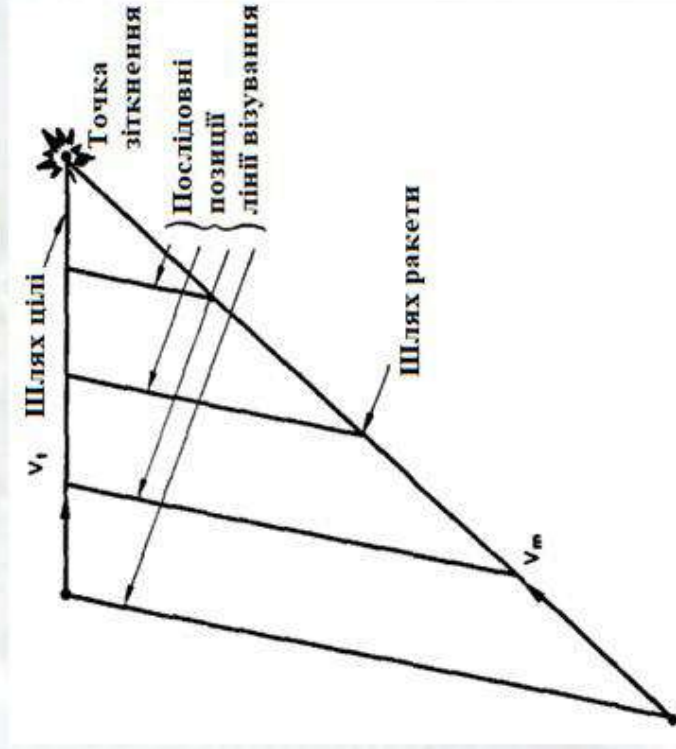
Для вирішення мети роботи було сформовано наступні задачі:

- проведення аналізу математичного апарату відстеження рухомих повітряних цілей;
- розробка методів та алгоритмів, що можуть застосовуватись для виявлення та наведення БПЛА на рухомі повітряні об'єкти;
- симуляція роботи розроблених алгоритмів;
- оцінка ефективності роботи запропонованих алгоритмів;
- вибір апаратних компонентів для реалізації системи.

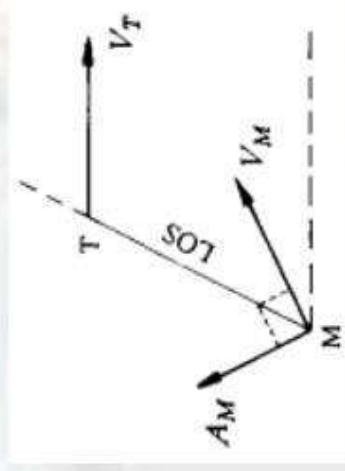
Основні положення цієї роботи полягають у наступному:

- оцінка різних методів наведення та управління та того, як їх можна адаптувати для використання на квадрокоптері в моделюваному середовищі;
- розробка програмного комплексу з використанням LOS-наведення для пошуку та наведення напівстаціонарних та рухомих цілей у межах фіксованої території (арени);
- розробка апаратного забезпечення на основі відеокамери для реалізації запропонованих алгоритмів.

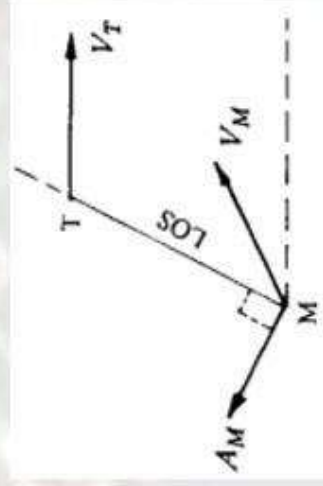
Математичне забезпечення наведення ракет



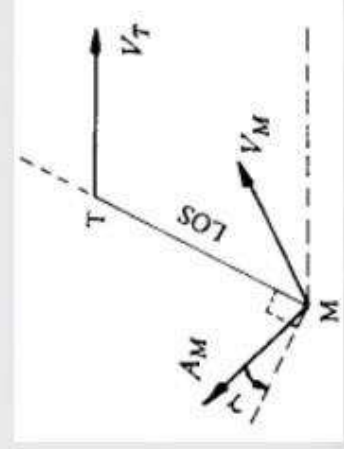
Діаграма курсу зіткнення з постійним пеленгом та миттєвим бічним прискоренням



Чиста пропорційна навігація

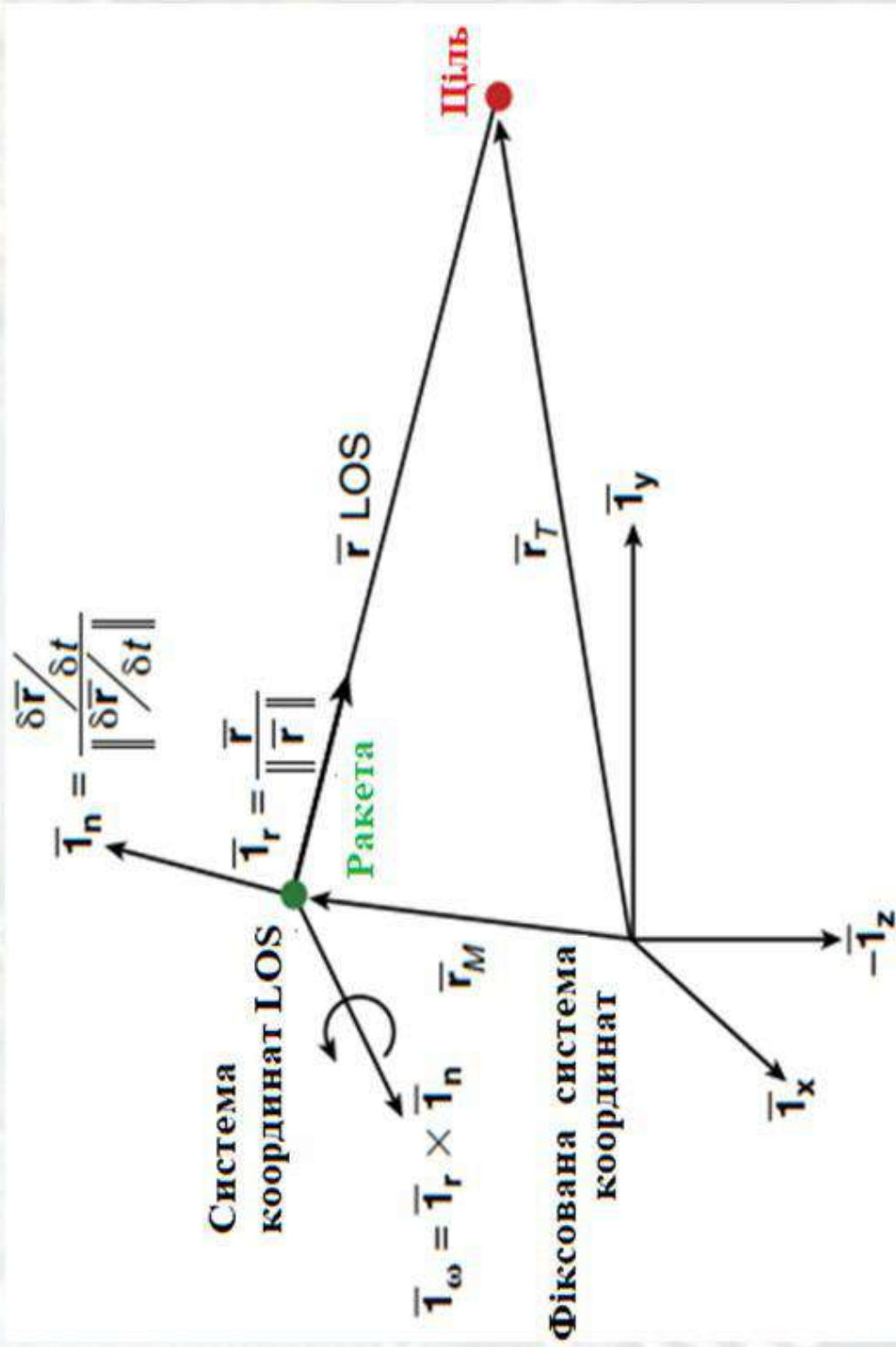


Справжня пропорційна навігація



Узагальнена справжня пропорційна навігація

Система координат LOS

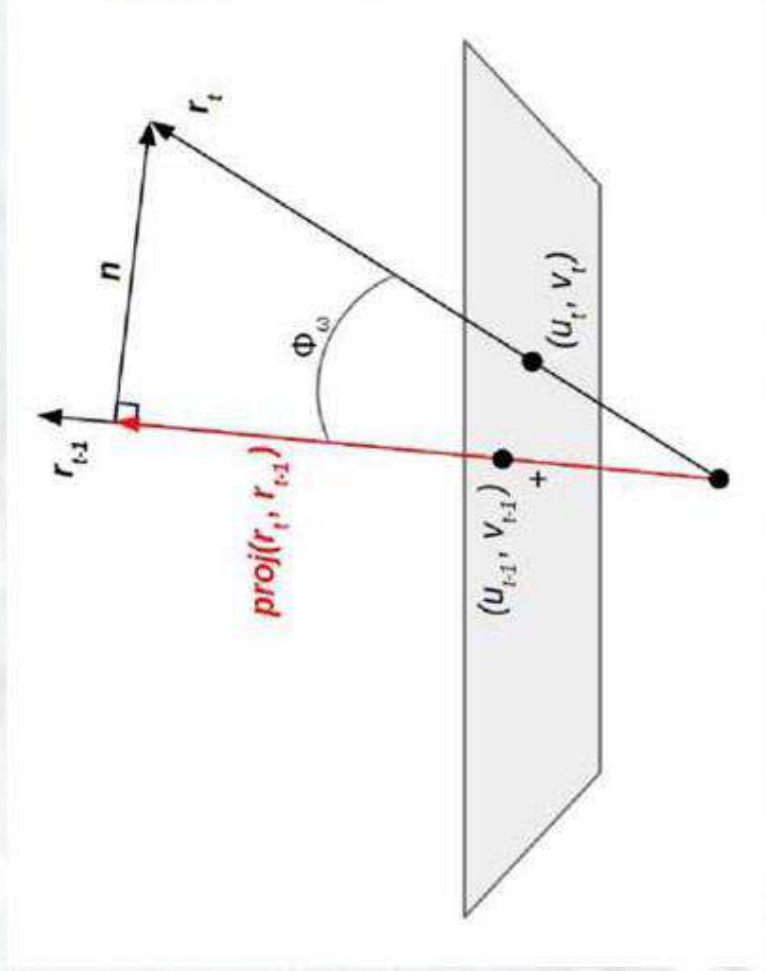


Істинно пропорційна навігація (ТРН)

Кут, охоплений двома векторами r_t і r_{t-1} , виглядає так:

$$\Phi_{\omega} = \arccos \frac{\langle r_t, r_{t-1} \rangle}{\|r_t\| \|r_{t-1}\|}$$

Напрямок прискорення (напрямок обертання LOS) l_n показано на рис.

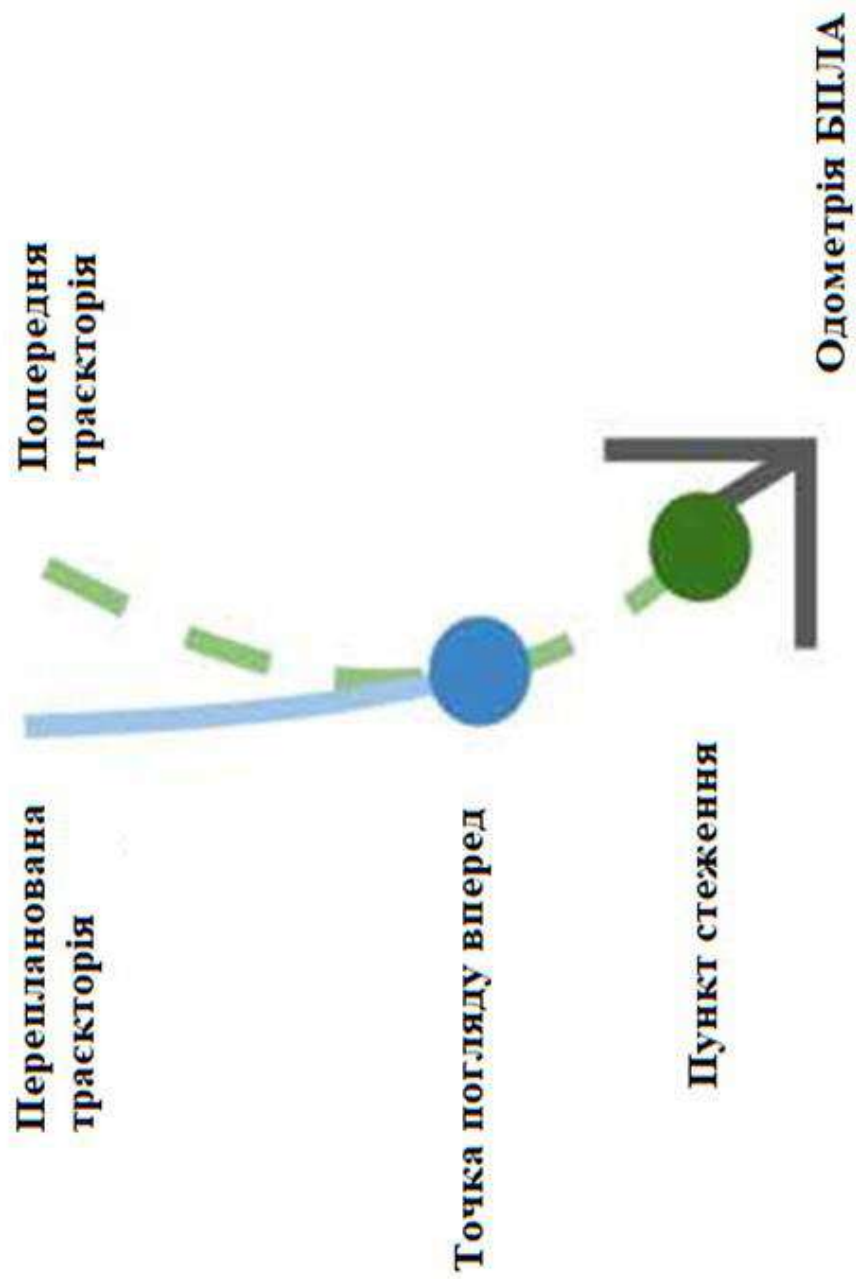


$$n - r_t = \text{proj}(r_t, r_{t-1})$$

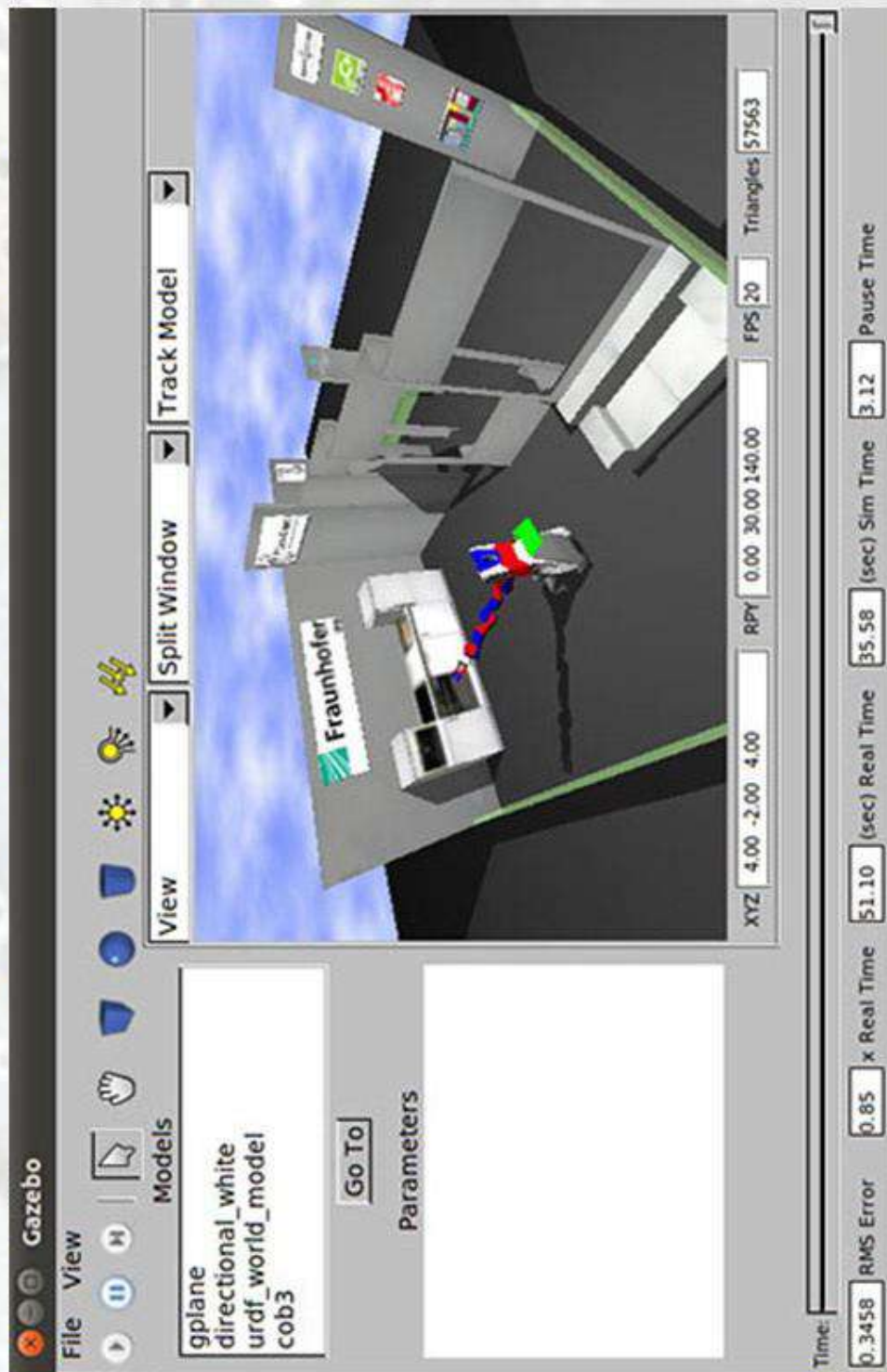
$$\|n - r_t\| < \|r_t\| \frac{r_{t-1}}{\|r_{t-1}\|} > \frac{r_{t-1}}{\|r_{t-1}\|}$$

$$l_n = \frac{n}{\|n\|}$$

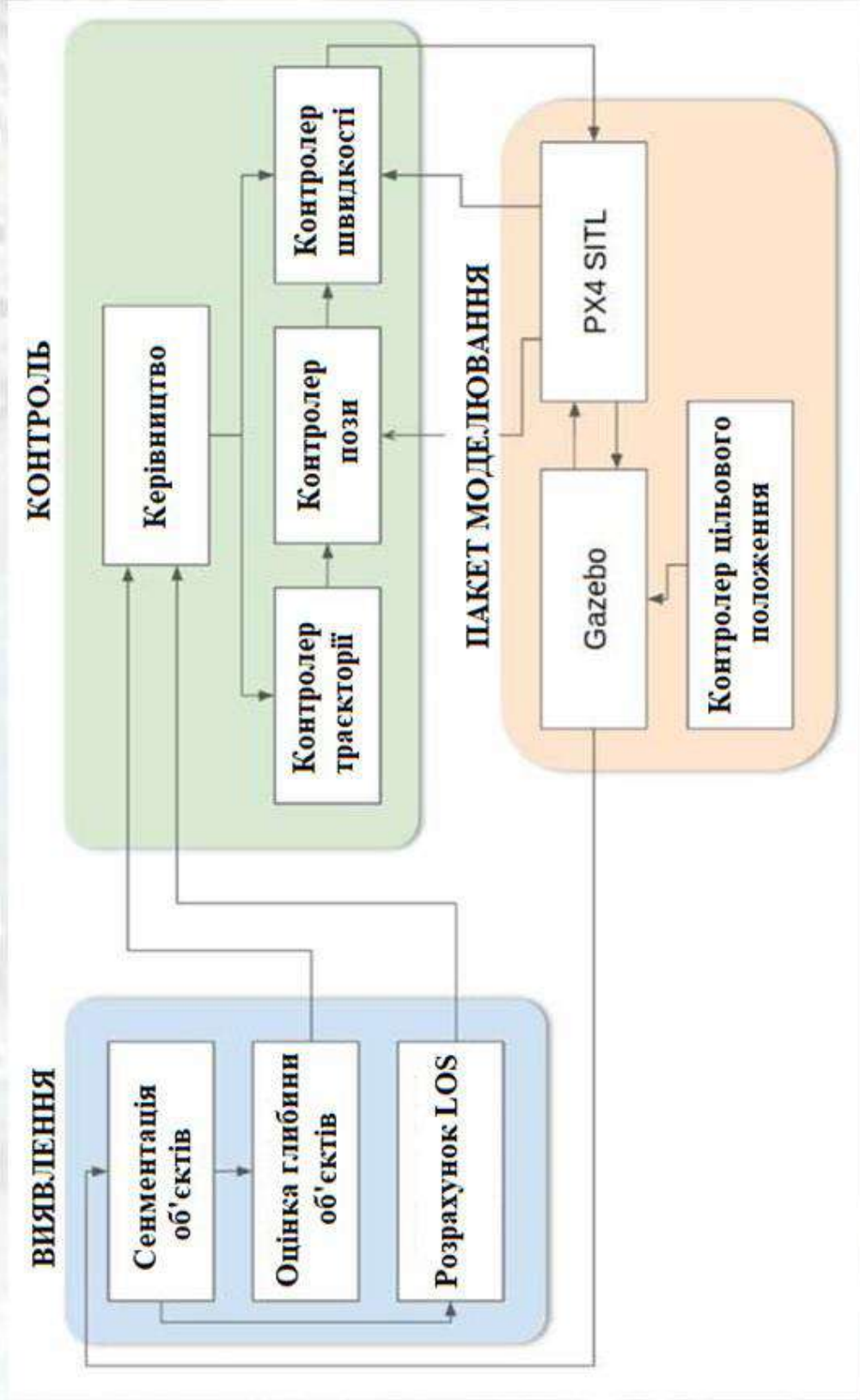
Проходження по траєкторії



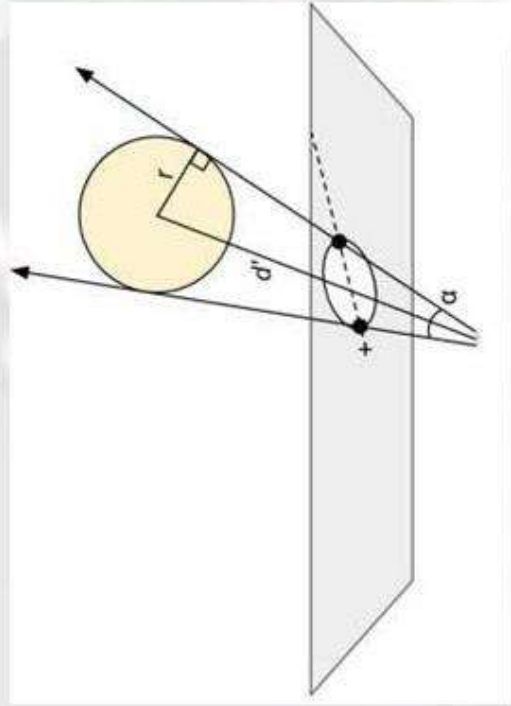
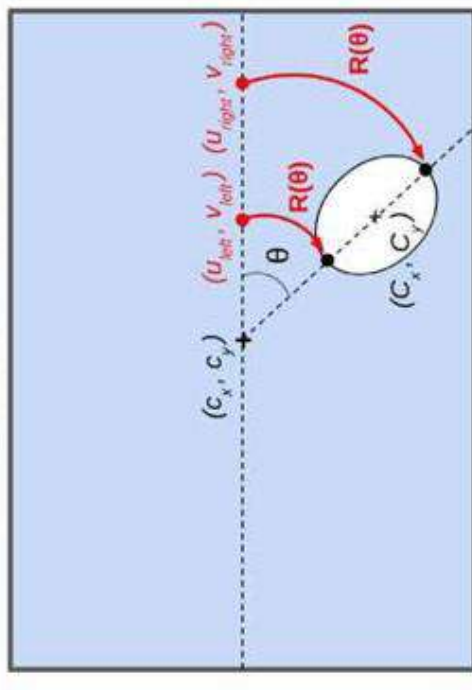
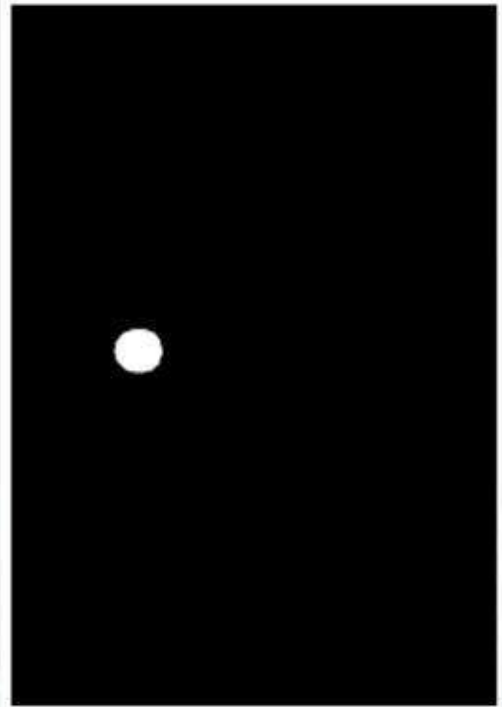
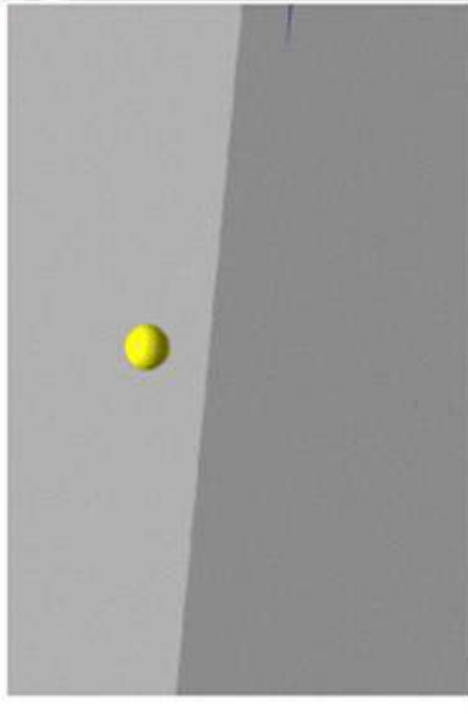
Вибір середовища симуляції



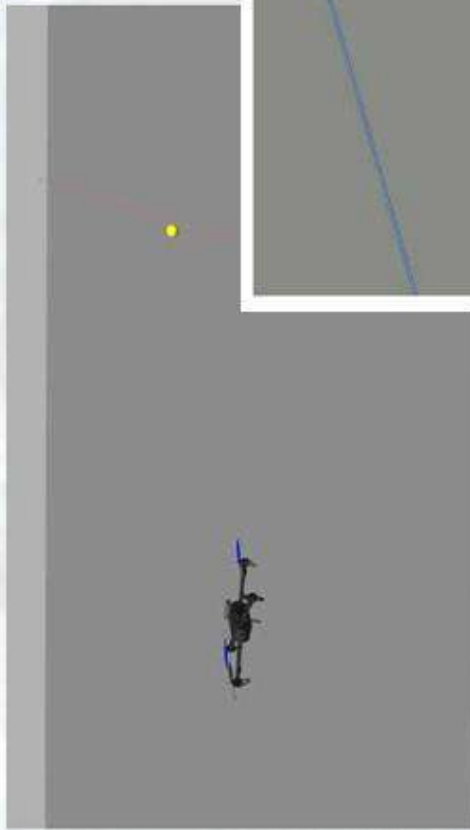
Структура системи



Сегментація об'єктів



Симуляція в Gazebo з моделлю БПЛА та цілю



(а) Прямий цільовий шлях.



(б) Маршрут цілі у формі вісімки.



(в) Цільовий шлях вузла.

Параметри моделювання алгоритмів

Було оцінено п'ять алгоритмів навігації. Конфігурація експерименту, що складається з вибору одного параметра з кожної з наступних категорій. Кожна конфігурація пройшла 50 випробувань.

Наведення БПЛА:

- справжня пропорційна навігація (TPN);
- пропорційна навігація з керуванням курсом (PN-напрямок);
- гібридна справжня пропорційна навігація з керуванням курсом (гібридна TPN-навігація);
- траєкторія LOS;
- прогнозована траєкторія.

Швидкість БПЛА [м/с]: 2.0, 3.0, 4.0, 5.0.

Шлях мішені: пряма, цифра-8, вузол.

Швидкість цілі [% від швидкості БПЛА]: 25%, 50%, 75%, 100%.

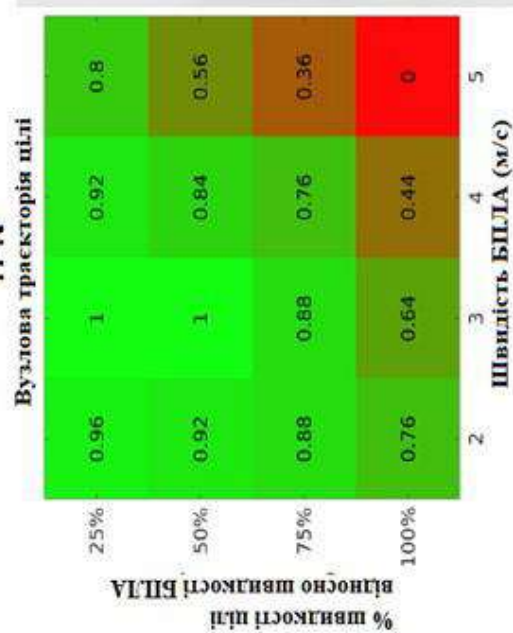
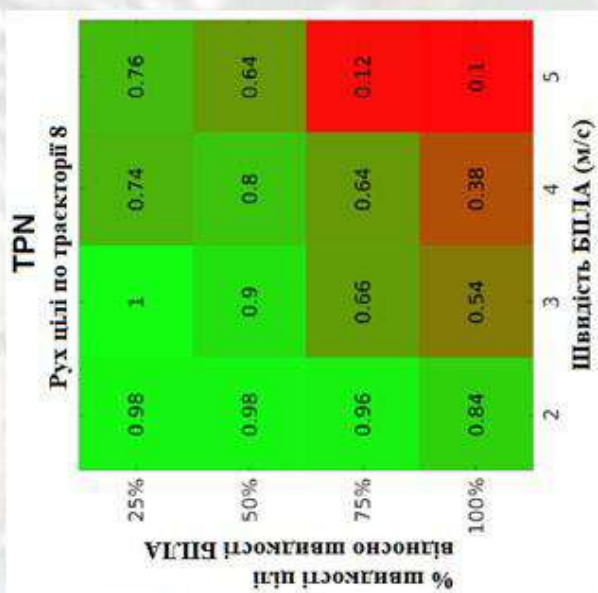
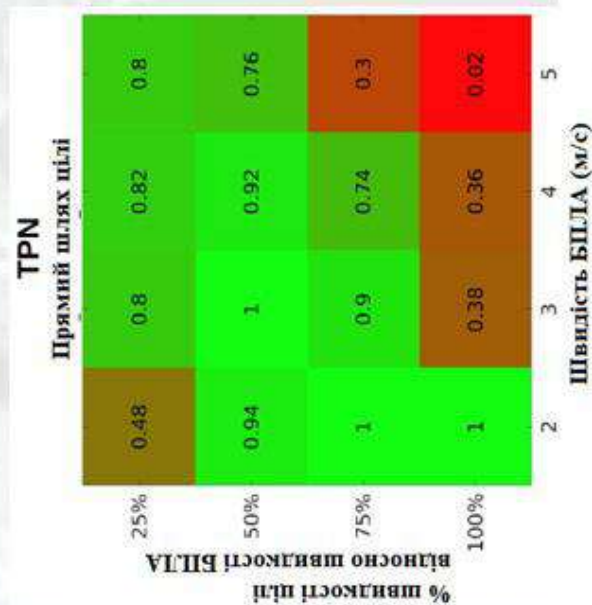
Умови моделювання алгоритмів

Усі наведені нижче умови повинні бути виконані, щоб спроба вважалася успішною першою передачею в мішень:

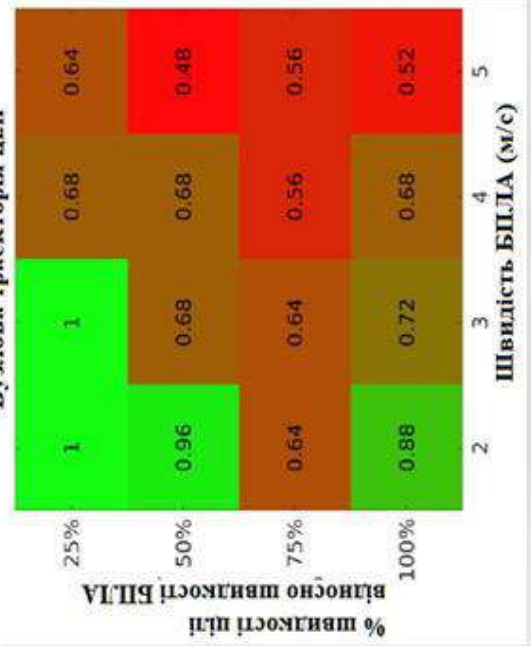
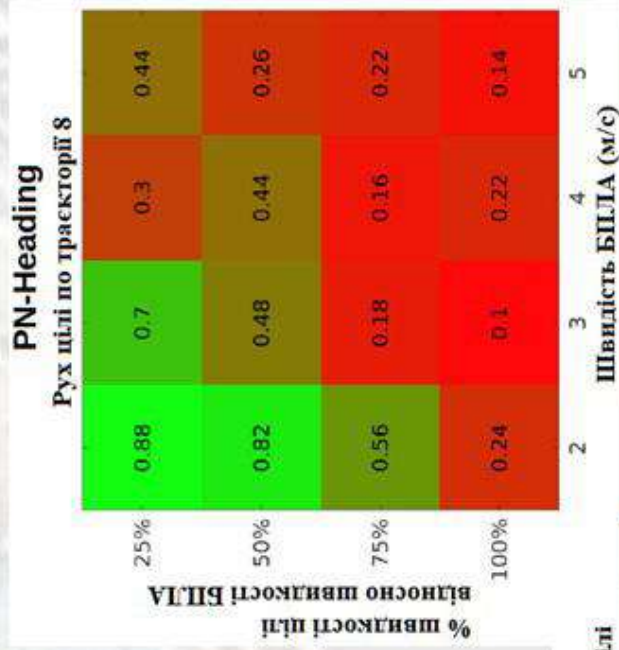
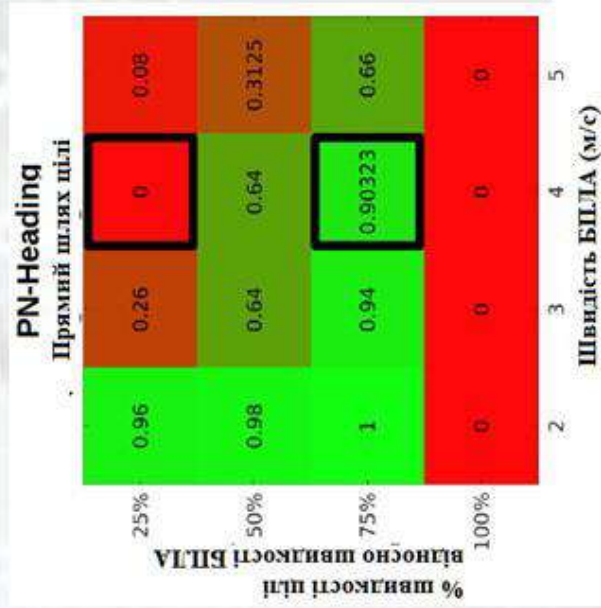
- БПЛА знаходиться в межах 0,5 м від цілі (вимірюється від найближчої точки цілі до центру БПЛА);
- тривалість переслідування менше 20 с;
- БПЛА знаходиться в зоні 35 x 100 x 40 метрів навколо цілі;
- ціль не знаходиться за межами поля зору камери БПЛА більше 3 секунд.

Ці умови були визначені після розгляду сцени моделювання, моделі БПЛА та моделі цілі. Тестова ціль — жовта сфера діаметром 1 м. Камера RGB має горизонтальний кут зору 105° з розміром 680x480. Замість курсової камери під час моделювання модель БПЛА мала жорстко закріплену камеру під кутом, яка регулювалася для кожної швидкості, щоб компенсувати стійкий нахил вниз під час польоту вперед на високих швидкостях.

Частота влучень True Proportional Navigation

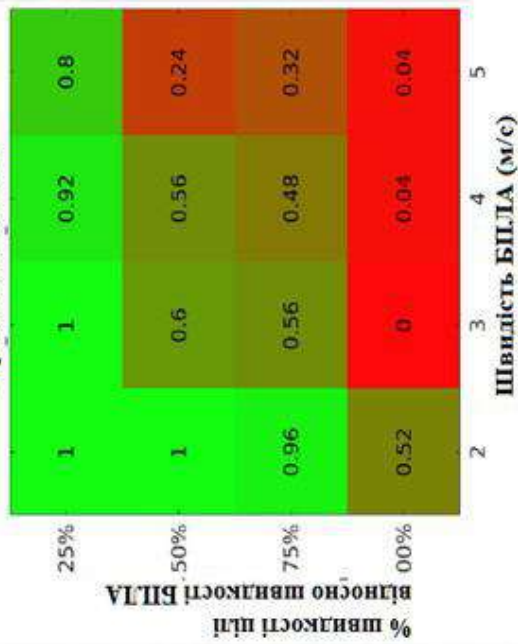


Пропорційна навігація з частотою влучень Heading Control

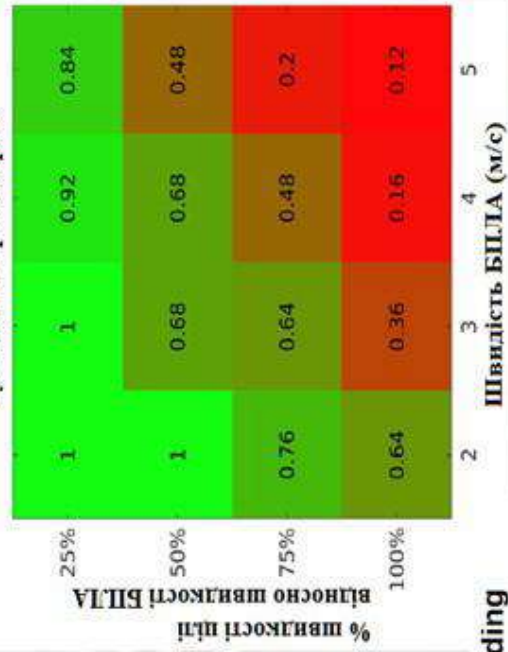


Змішана істинно пропорційна навігація-керування

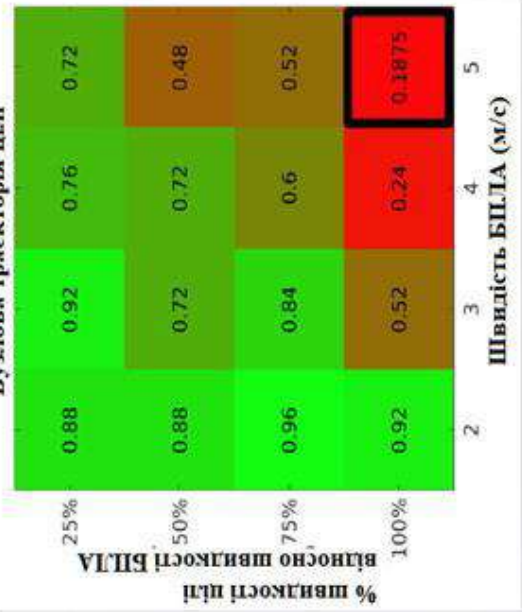
Hybrid TPN-Heading
Прямий шлях цілі



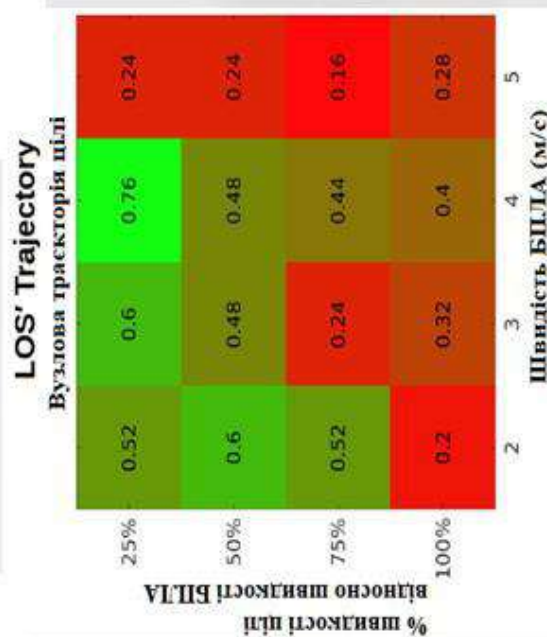
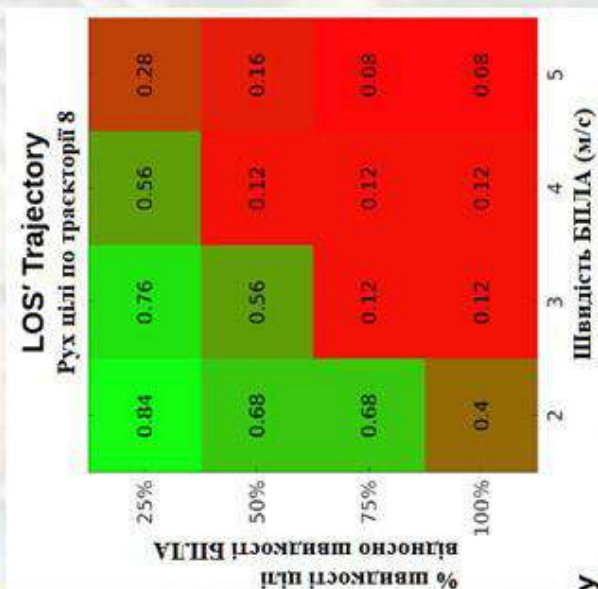
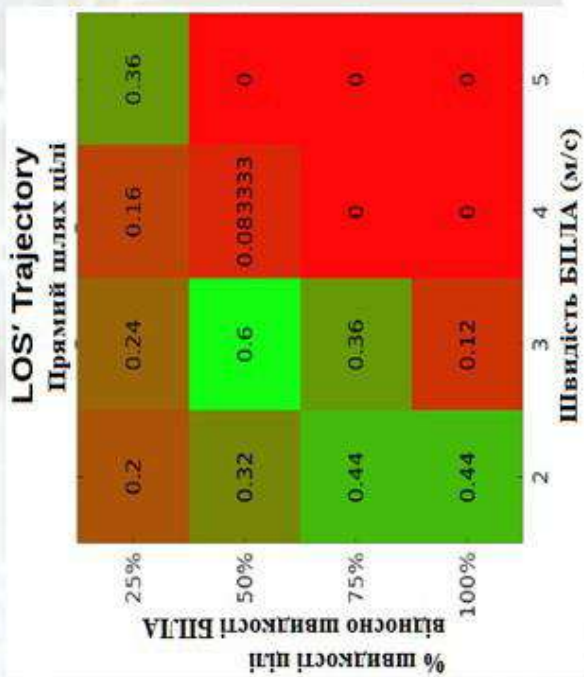
Hybrid TPN-Heading
Рух цілі по траєкторії 8



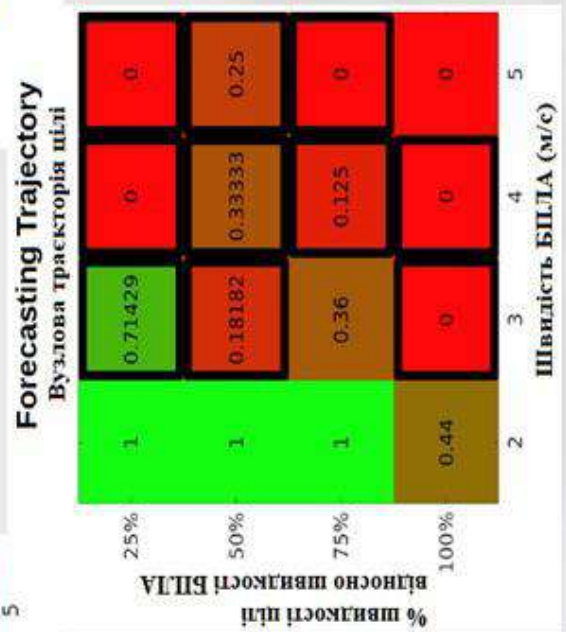
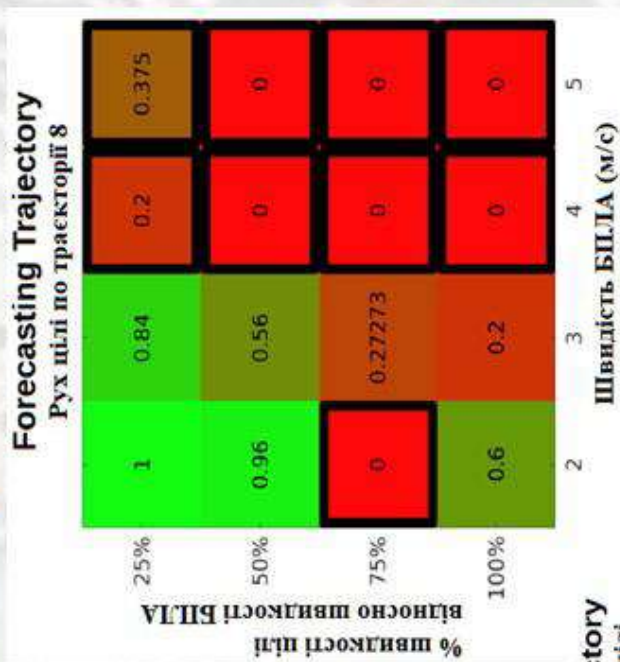
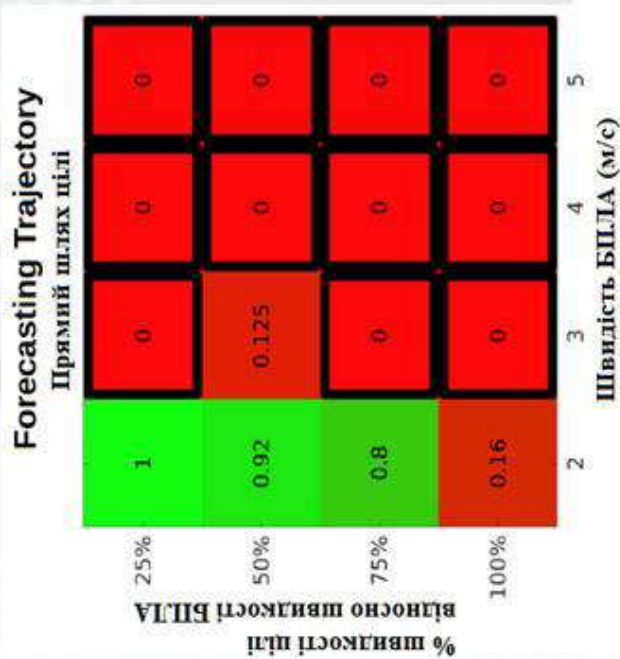
Hybrid TPN-Heading
Вузлова траєкторія цілі



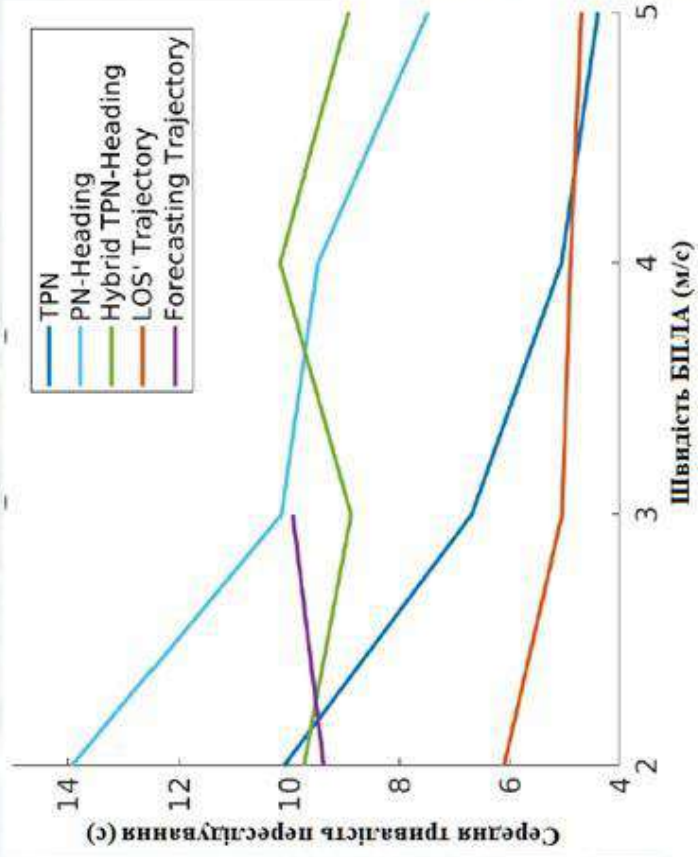
Коефіцієнт попадання траєкторії LOS



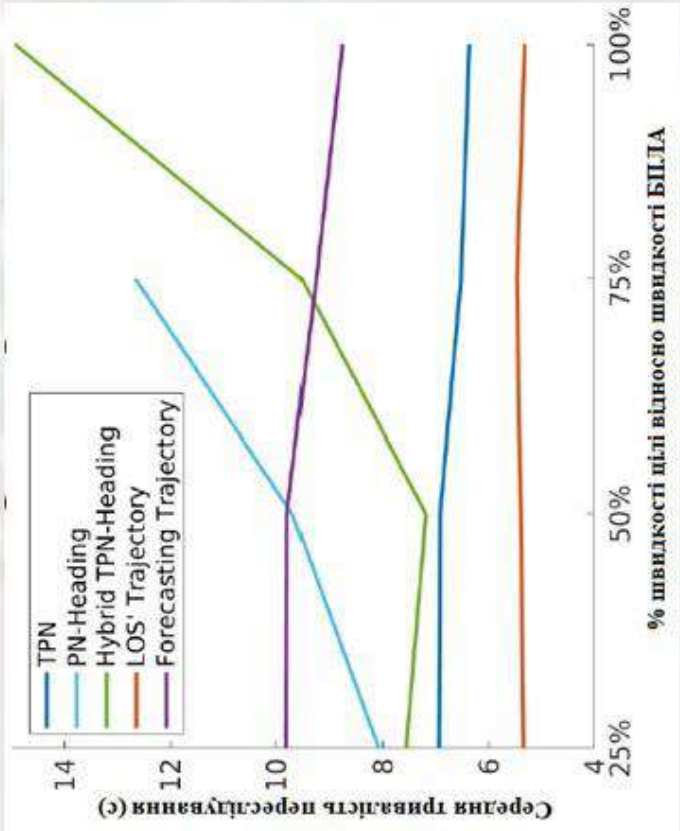
Прогноз траєкторії влучень



Залежність тривалості переслідування для прямої траєкторії цілі

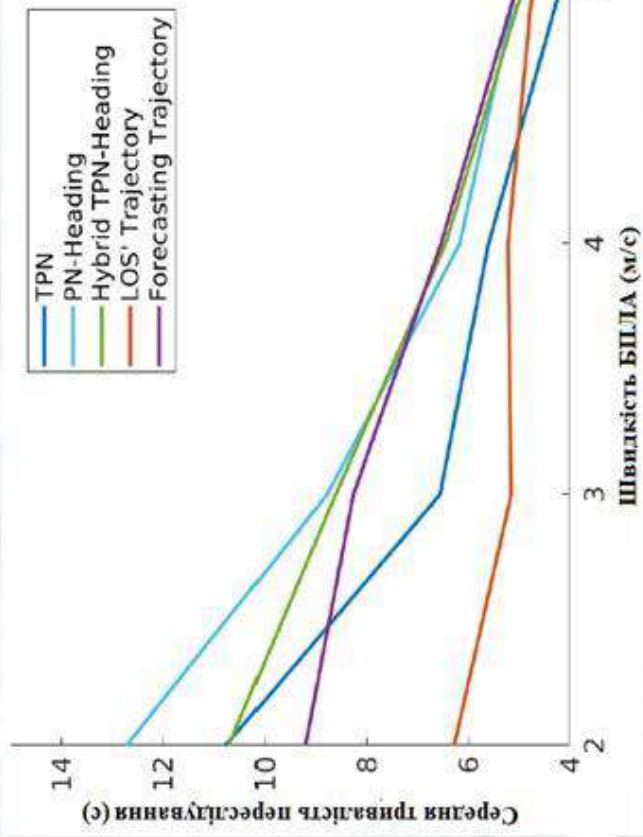


Від швидкості БПЛА

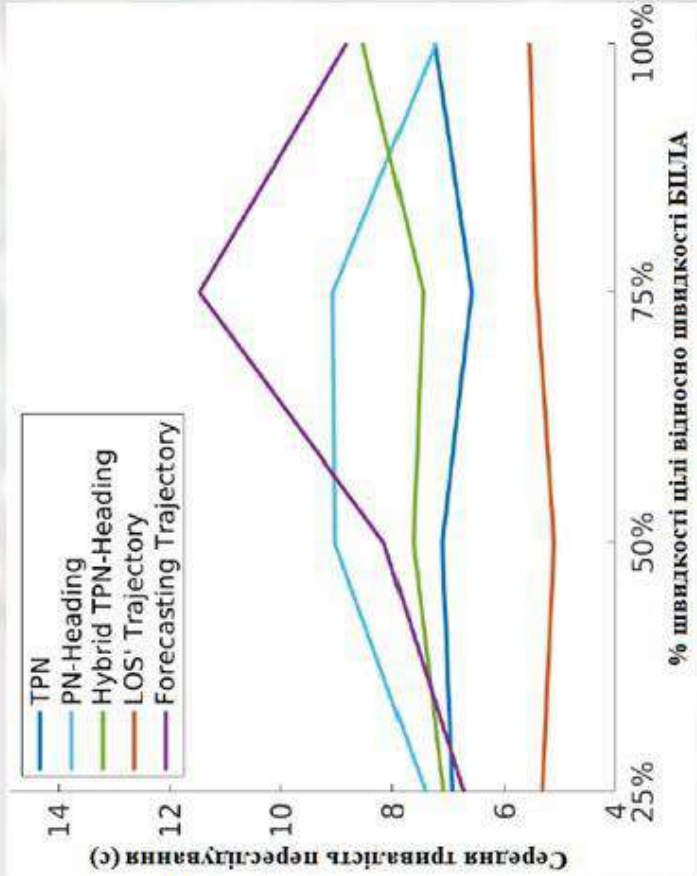


Від швидкості цілі

Залежність тривалості переслідування для траєкторії цілі у формі вісімки

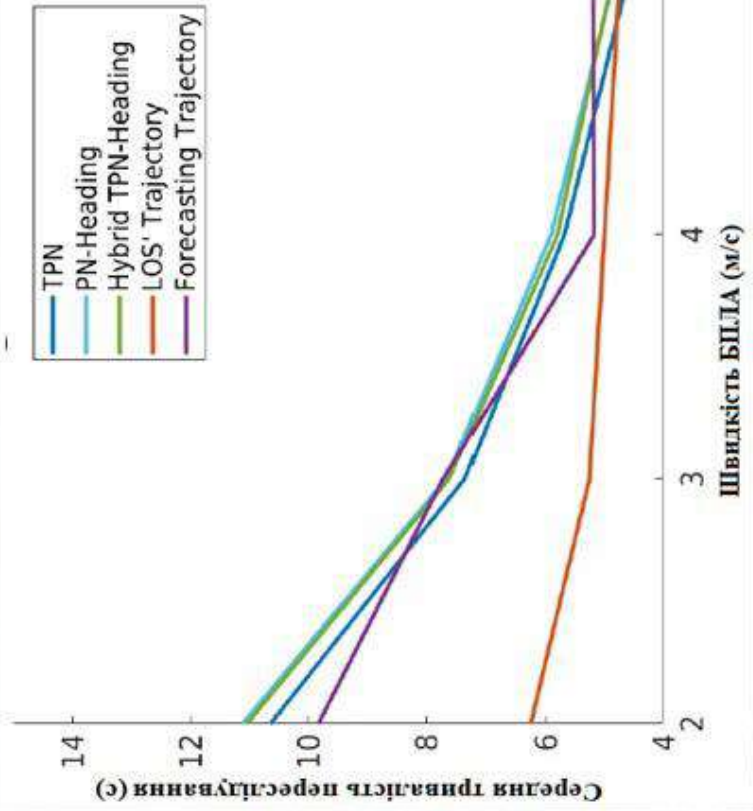


Від швидкості БПЛА

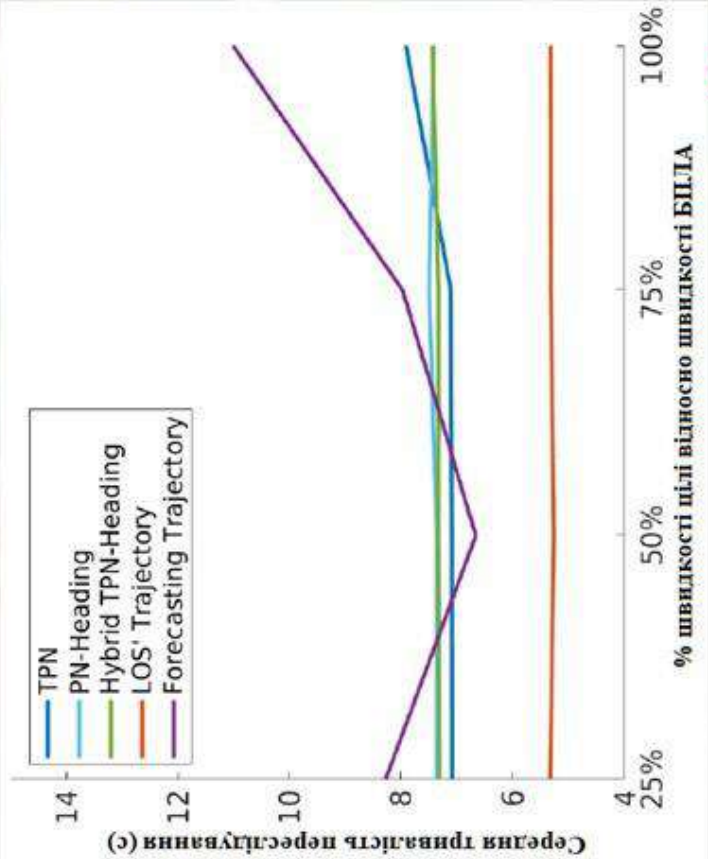


Від швидкості цілі

Залежність тривалості переслідування для вузлової траєкторії цілі

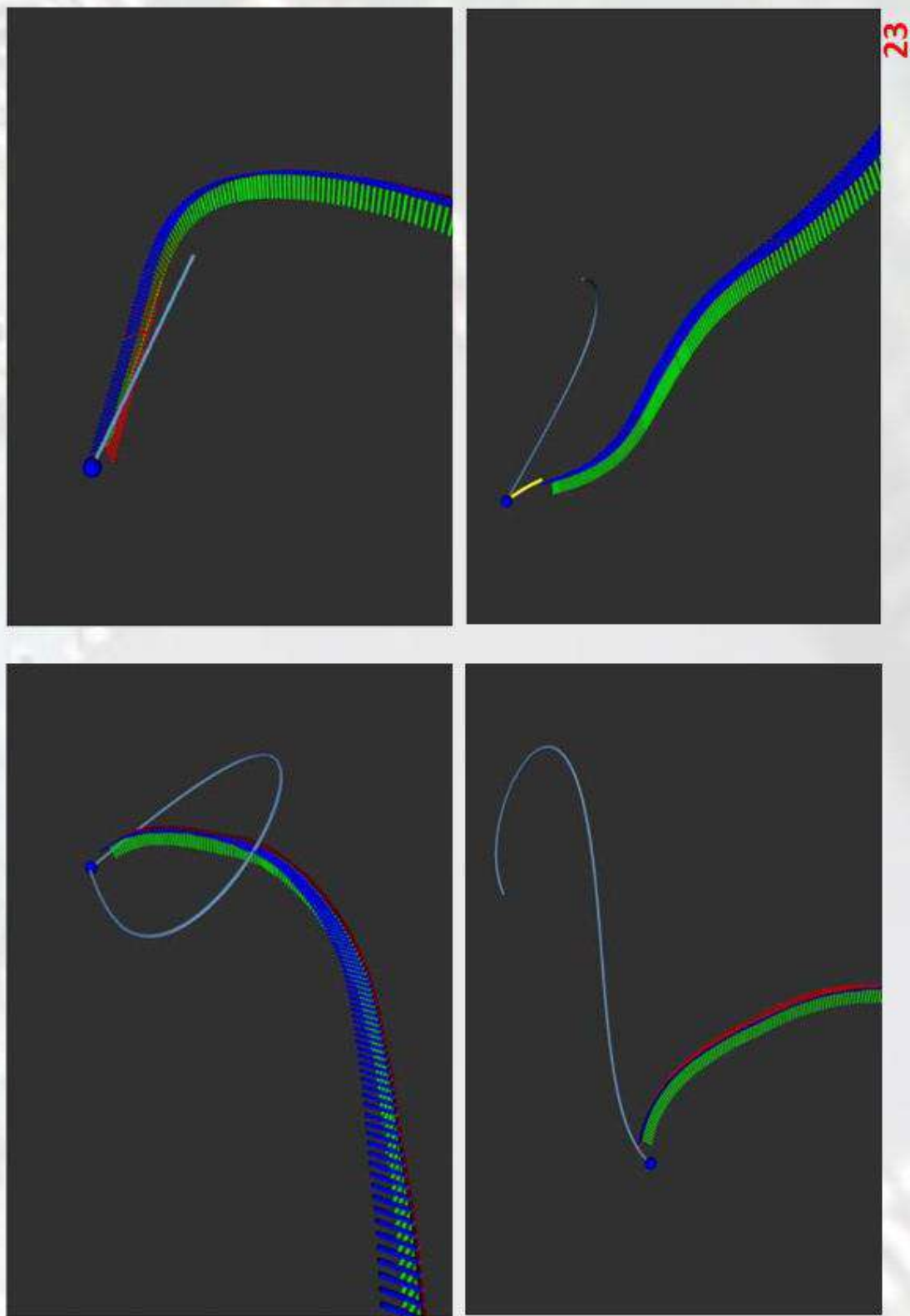


Від швидкості БПЛА

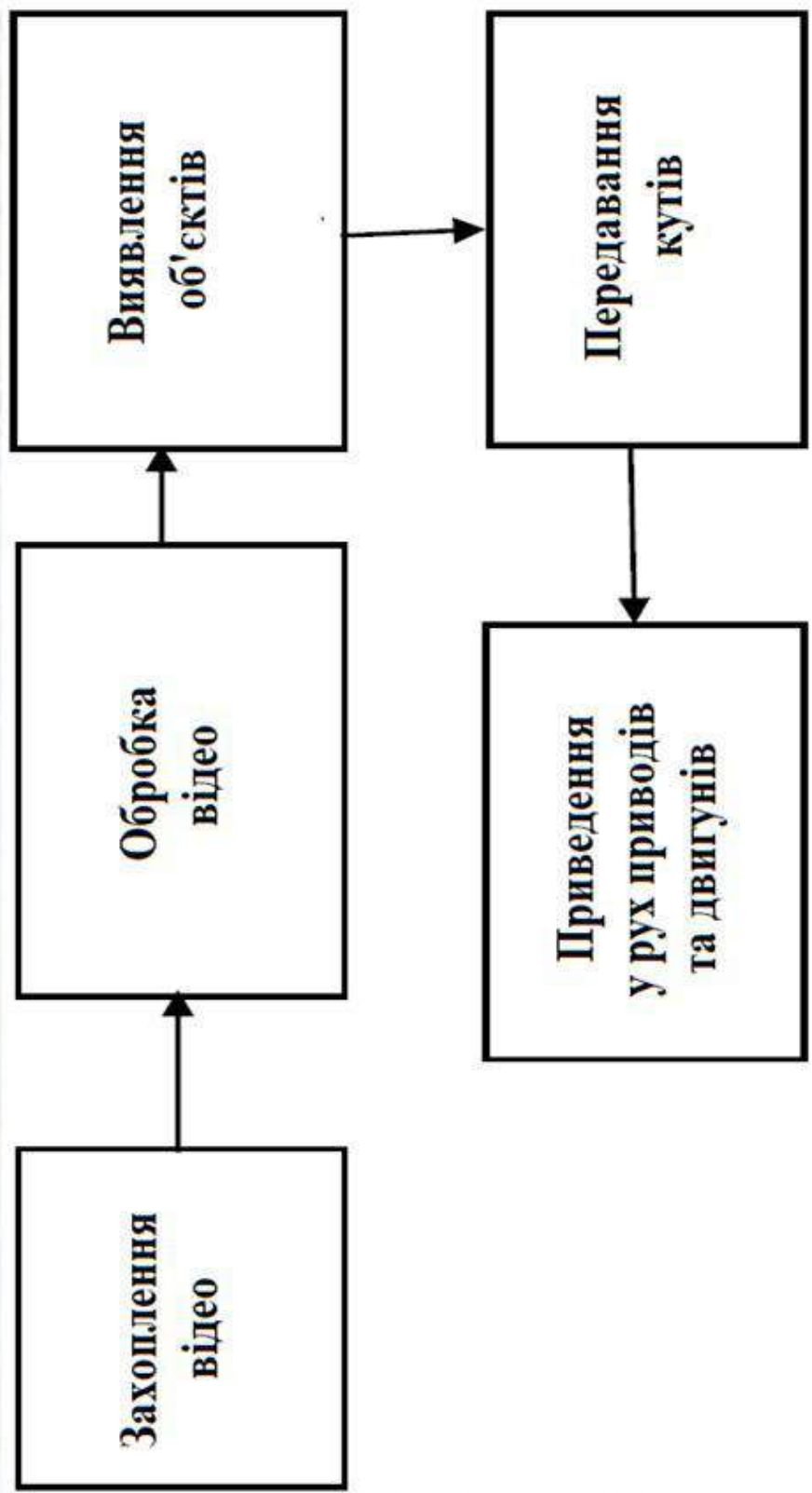


Від швидкості цілі

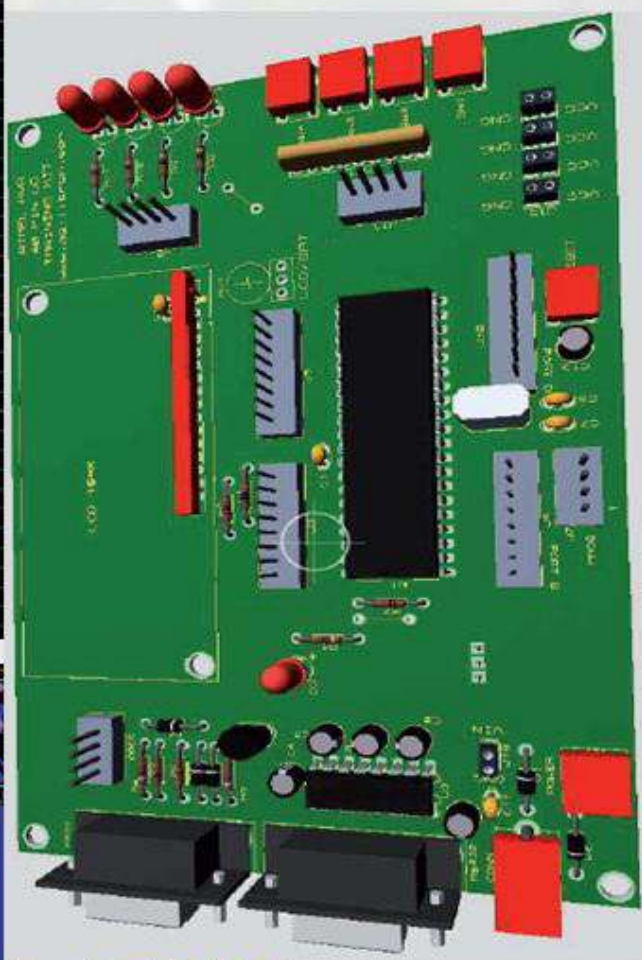
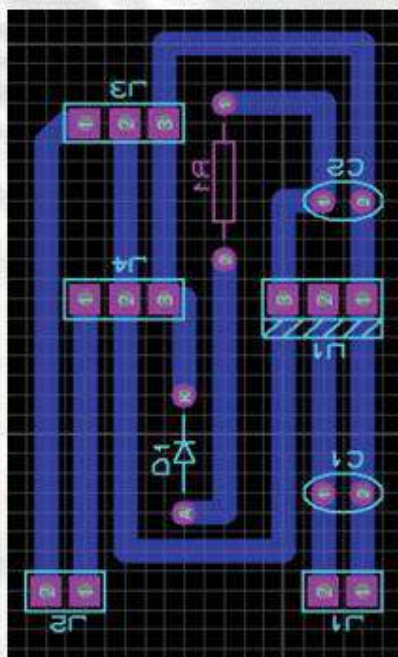
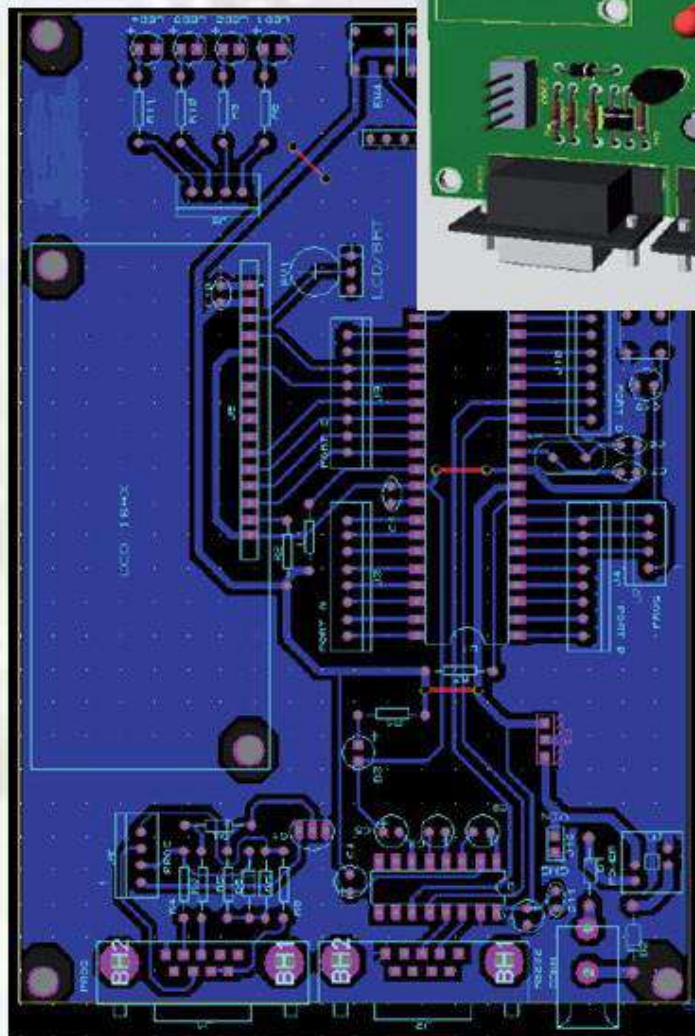
Поведінка при переслідуванні



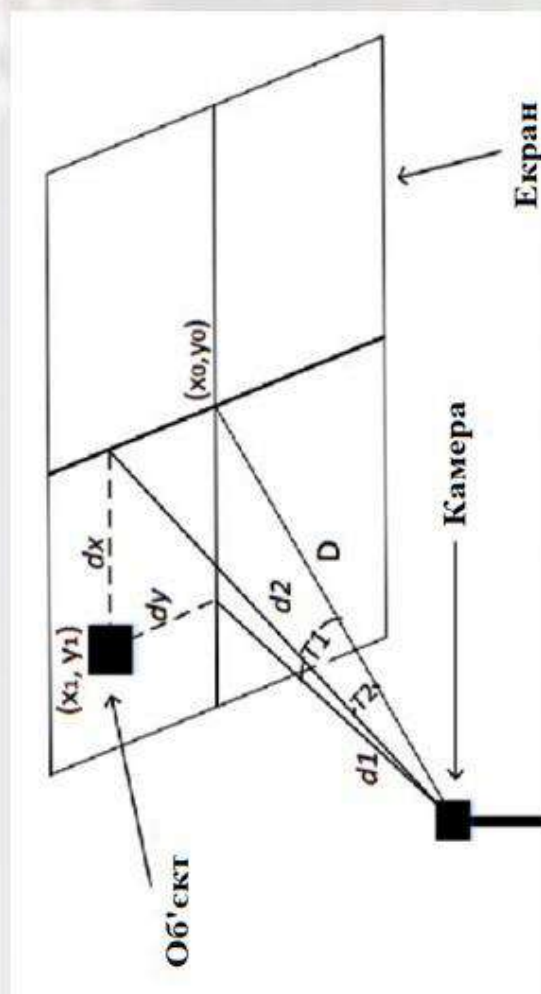
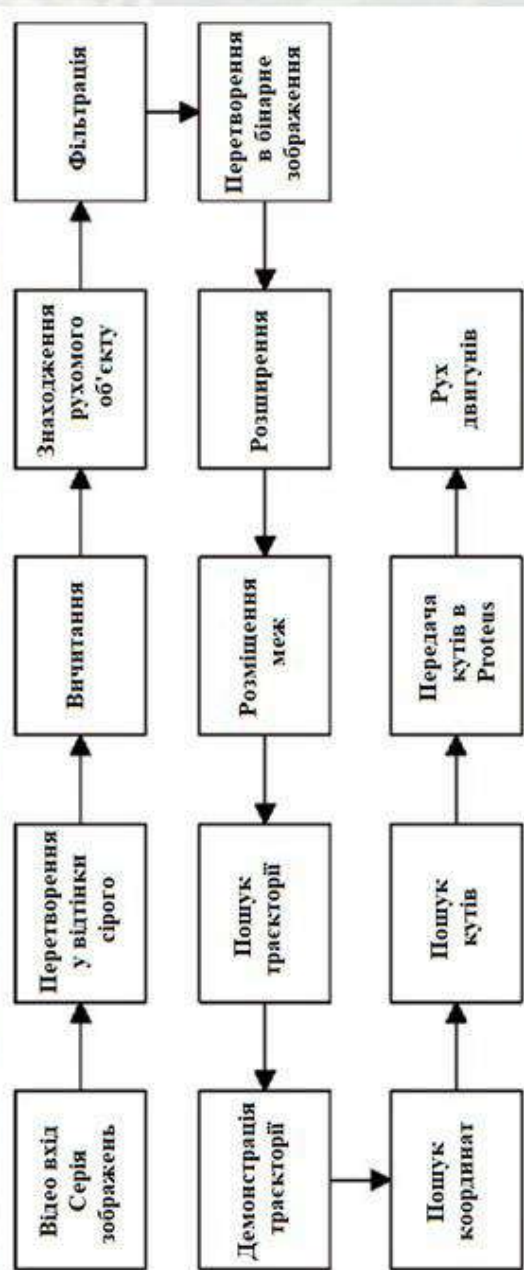
Порядок роботи апаратної частини з відео



Конструкція апаратної частини



Відпрацювання роботи апаратної частини



Висновки

У кваліфікаційній роботі магістра було розроблено та реалізовано алгоритми наведення в змодельованому та реальному середовищах. Методи наведення LOS були отримані з алгоритму справжньої пропорційної навігації, який взято з літератури з наведення ракет. Реалізація на квадрокоптері забезпечує більший контроль і маневреність за рахунок додаткових ступенів свободи.

Усі ці методи спиралися лише на інформацію з монокулярної RGB-камери на борту квадаторного БПЛА. Камера була як жорстко закріплена на рамі БПЛА, так і була встановлена на підвісі та встановлена так, щоб вона була направлена вперед відносно курсу БПЛА.

Малий О.Ю., Моїсєєв В.Є. Наведення БПЛА квадрокоптерного типу на рухомі об'єкти під час польоту // Тиждень науки-2024. Факультет інформаційної безпеки та електронних комунікацій. Тези доповідей науково-практичної конференції, Запоріжжя, 15–19 квітня 2024 р. [Електронний ресурс] / Редкол. : Вадим ШАЛОМЄЄВ (відпов. ред.) Електрон. дані. – Запоріжжя : НУ «Запорізька політехніка», 2024. – с.46-47