

Хмельницький національний університет
Факультет програмування та
комп'ютерних і телекомунікаційних систем
Кафедра комп'ютерної інженерії та системного програмування

ДИПЛОМНА РОБОТА МАГІСТРА

Галузь знань 12 – Інформаційні технології

Спеціальність 123 – Комп'ютерна інженерія

на тему «Інтелектуалізована система керування безпілотними літальними апаратами»


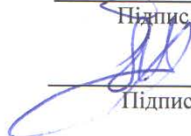
ДРКІСПр. 015100. 16.13 ПЗ

Виконав: студент 2 курсу, група КІ2м-19-1

Керівник доктор техн. наук, доцент
Науковий ступінь, вчене звання

До захисту допускаю:
Зав. кафедри КІСП, д.т.н., проф.
Т.О. Говорущенко

05 05 2021 р.


Підпис

Підпис

Румянцев С.В.
Ініціали, прізвище

Лисенко С.М.
Ініціали, прізвище

Хмельницький, 2021

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ПРОГРАМУВАННЯ ТА КОМП'ЮТЕРНИХ І ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА СИСТЕМОГО ПРОГРАМУВАННЯ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЬО-НАУКОВА ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко



“ 03 ” 09 2019 р.

ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)

Румянцеву Сергію Володимировичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Інтелектуалізована система керування безпілотними літальними апаратами

Керівник проекту (роботи) Лисенко С.М., к.т.н., доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.09.2020 р. № 118

2. Строк подання студентом проекту (роботи) на кафедру 05.05.2021 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Дослідження методів планування траєкторії польоту за допомогою безпілотних літальних апаратів





Побудова математичної моделі системи керування безпілотними літальними апаратами

Інтелектуалізована система керування безпілотними літальними апаратами

Реалізація інтелектуалізована система керування безпілотними літальними апаратами

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КІСП		
Антиплагіат	Нічепорук А.О., доцент кафедри КІСП		

7. Дата видачі завдання « 03 » вересня 2020 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики ДРМ з керівником	05.09.2019	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	05.10.2019	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	05.11.2019	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	-	виконано
5	Робота над науковою статтею	05.03.2020	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.06.2020	виконано
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	05.08.2020	виконано
8	Оформлення пояснювальної записки згідно вимог	15.04.2021	виконано
9	Попередній захист ДРМ	1.03.2021	виконано
10	Захист ДРМ на засіданні ЕК	До 06.05.2021	

Студент


Підпис

Румянцев С.В.
Ініціали, прізвище

Керівник проекту (роботи)


Підпис

Лисенко С.М.
Ініціали, прізвище

РЕФЕРАТ

Тема дипломної роботи: «Інтелектуалізована система керування безпілотними літальними апаратами».

Автор роботи: Румянцев Сергій Володимирович

Керівник роботи: д.т.н., доцент Лисенко С.М.,

Пояснювальна записка: 87 с., 52 рис., 12 табл., 3 дод., 60 джерел.

ДРОН, БЕЗПЛОТНИЙ ЛІТАЛЬНИЙ АПАРАТ, НЕЧІТКА ЛОГІКА, ТРАЄКТОРІЯ, PID-КОНТРОЛЕР, СИСТЕМА КЕРУВАННЯ.

Об'єктом дослідження є процес керування безпілотними літальними апаратами типу дрон.

Предметом дослідження є моделі, методи та інтелектуалізована система керування безпілотними літальними апаратами.

Метою дипломної магістра роботи є підвищення точності траєкторії руху безпілотного літального апарату типу дрон.

Для розв'язання поставлених задач використовувалися методи основні положення системного аналізу, методів аналізу даних, теорії дискретної математики, апарату нечіткої логіки, теорії комп'ютерних мереж та систем.

Наукова новизна отриманих результатів:

- удосконалено метод керування безпілотними літальними апаратами, який на відміну від відомих застосовує компоненти штучного інтелекту, і який забезпечує підвищення точності траєкторії руху безпілотного літального апарату типу дрон.

- набула подальшого розвитку інтелектуалізована система керування безпілотними літальними апаратами, яка забезпечує підвищення точності траєкторії руху безпілотного літального апарату типу дрон.

Практична значимість отриманих результатів полягає у розробленні інтелектуалізованої системи керування безпілотними літальними апаратами, яка забезпечуватиме задання траєкторії руху безпілотного літального апарату типу дрон з високою точністю.

Дослідження, представлені у кваліфікаційній роботі, проводились в рамках держбюджетної НДР Хмельницького національного університету № 1Б-2019 «Агентно-орієнтована система підвищення безпеки та якості програмного забезпечення комп'ютерних систем» (номер державної реєстрації 0119U100662).

За темою дипломної роботи опубліковано статтю на тему «Інтелектуалізована система керування безпілотними літальними апаратами» в журналі «Комп'ютерні системи та інформаційні технології» №1 за 2020 рік.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	4
ВСТУП.....	5
1 ДОСЛІДЖЕННЯ МЕТОДІВ ПЛАНУВАННЯ ТРАЄКТОРІЇ ПОЛЬОТУ ЗА ДОПОМОГОЮ БЕЗПЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ	8
1.1 Класифікація БПЛА	8
1.2 Огляд відомих методів побудови траєкторії польоту БПЛА	9
1.3 Планування траєкторії польоту	11
1.4 Клітинне розкладання.....	12
1.5 Показники ефективності польоту.....	15
1.6 Методи задання траєкторії без розкладання	17
1.7 Методи задання траєкторії точного клітинного розкладання.....	21
1.8 Методи задання траєкторії єдиної стратегії.....	21
1.9 Висновки та постановка задачі.....	25
2 ПОБУДОВА МАТЕМАТИЧНОЇ МОДЕЛІ СИСТЕМИ КЕРУВАННЯ БЕЗПЛОТНИМИ ЛІТАЛЬНИМИ АПАРАТАМИ	27
2.1 Математична модель системи керування безпілотними літальними апаратами	27
2.2 Методи управління квадродотором	33
2.3 PID-контроль	33
2.4 Самоналаштуваний нечіткий PID-контролер.....	35
2.5 Динаміка та самоналаштування нечіткого PID-контролю квадродотора AR.Drone	39
2.6 Висновки	46

3 ІНТЕЛЕКТУАЛІЗОВАНА СИСТЕМА КЕРУВАННЯ БЕЗПІЛОТНИМИ ЛІТАЛЬНИМИ АПАРАТАМИ.....	47
3.1 Побудова інтелектуалізованої системи керування безпілотними літальними апаратами	47
3.2 Система керування БПЛА	50
3.3 Експерименти	54
3.4 Висновки	67
4 РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛІЗОВАНОЇ СИСТЕМИ КЕРУВАННЯ БЕЗПІЛОТНИМИ ЛІТАЛЬНИМИ АПАРАТАМИ	68
4.1 Використання системи MATLAB для реалізація інтелектуалізованої системи керування безпілотними літальними апаратами.....	68
4.2 Програмування контролера інтелектуалізованої системи керування безпілотними літальними апаратами	77
4.3 Автономне відстеження траєкторії квадродротора	83
4.4 Платформа квадродротора AR.Drone та передача даних.....	83
4.5 Висновки	85
ВИСНОВКИ.....	86
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	88
Додаток А Результати використання системи MATLAB для реалізація інтелектуалізованої системи керування безпілотними літальними апаратами	95
Додаток Б Програмна реалізація нечіткого PID-контролера	104
Додаток В Презентація доповіді.....	118
Додаток Г Опублікована стаття на тему «Інтелектуалізована система керування безпілотними літальними апаратами» в журналі «Комп’ютерні системи та інформаційні технології» №1 за 2020 рік	151

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БПЛА - Безпілотні літальні апарати

ПТП - планування траєкторії польоту

ПЗ - програмне забезпечення

ЕРШ - електронними регуляторами швидкості

ВСТУП

Безпілотні літальні апарати (БПЛА) все частіше використовуються в різних додатках, таких як спостереження [1], розумне сільське господарство [2], фотограмметрія [3], управління катастрофами, цивільна безпека [4], відстеження лісових пожеж [5], моніторинг хмар [6], нагляд за конструкцією [7] та інспекція ліній електропередач [8]. БЛА складається з безпіотної повітряної платформи на транспортному засобі. Такі платформи контролюють людей віддалено та вручну, але вони також виконують автоматичні запрограмовані польоти. Інтелектуальна система з вбудованими бортовими датчиками може бути використана для автоматичного польоту.

Проблема планування траєкторії польоту (ПТП) класифікується як план руху підтеми у робототехніці, і в цьому випадку необхідно побудувати шлях для робота, щоб вивчати кожну позицію в певній сцені [9]. Незважаючи на те, що на цьому типі автономних польотних платформ було досягнуто технологічних досягнень, слід підкреслити, що з міркувань безпеки фази, пов'язані зі зльотом, місією та посадкою, зазвичай допомагають двом людям на кожен безпілотник. Пілот контролює завдання і може вручну змінити режим польоту, коли під час польоту виникає аварія або надзвичайна ситуація, тоді як основний оператор контролює навігаційні дані під час польоту, такі як зміна висоти та розряд акумулятора [10].

Метою дипломної магістра роботи є підвищення точності траєкторії руху безпілотного літального апарату типу дрон.

Поставлена мета досягається розв'язанням таких основних задач:

1. дослідити особливості задання траєкторії руху безпілотного літального апарату типу дрон;
2. проаналізувати сучасні задання траєкторії руху безпілотного літального апарату типу дрон;
3. розробити модель задання траєкторії руху безпілотного літального апарату типу дрон;

4. розробити метод керування безпілотними літальними апаратами, який на відміну від відомих застосовує компоненти штучного інтелекту, і який забезпечує підвищення точності траєкторії руху безпілотного літального апарату типу дрон;

5. розробити інтелектуалізовану систему керування безпілотними літальними апаратами, яка забезпечує підвищення точності траєкторії руху безпілотного літального апарату типу дрон.

Об'єкт дослідження – процес керування безпілотними літальними апаратами типу дрон.

Предмет дослідження – моделі, методи та інтелектуалізована система керування безпілотними літальними апаратами.

Методи дослідження. Для розв'язання поставлених задач використовуються основні положення системного аналізу, методів аналізу даних, теорії дискретної математики, апарату нечіткої логіки, теорії комп'ютерних мереж та систем.

Наукова новизна отриманих результатів:

1. Удосконалено метод керування безпілотними літальними апаратами, який на відміну від відомих застосовує компоненти штучного інтелекту, і який забезпечує підвищення точності траєкторії руху безпілотного літального апарату типу дрон.

2. Набула подальшого розвитку інтелектуалізована система керування безпілотними літальними апаратами, яка забезпечує підвищення точності траєкторії руху безпілотного літального апарату типу дрон.

Практична цінність. В результаті виконаного наукового дослідження було розроблено інтелектуалізовану систему керування безпілотними літальними апаратами, яка забезпечуватиме задання траєкторії руху безпілотного літального апарату типу дрон з високою точністю.

Зв'язок роботи з науковими програмами, планами, темами. Дослідження, представлені у кваліфікаційній роботі, проводились в рамках держбюджетної НДР Хмельницького національного університету № 1Б-2019 «Агентно-

орієнтована система підвищення безпеки та якості програмного забезпечення комп'ютерних систем» (номер державної реєстрації 0119U100662).

За темою дипломної роботи опубліковано статтю на тему «Інтелектуалізована система керування безпілотними літальними апаратами» в журналі «Комп'ютерні системи та інформаційні технології» №1 за 2020 рік [48].

1 ДОСЛІДЖЕННЯ МЕТОДІВ ПЛАНУВАННЯ ТРАЄКТОРІЇ ПОЛЬОТУ ЗА ДОПОМОГОЮ БЕЗПЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ

1.1 Класифікація БПЛА

БПЛА можна розділити на дві основні конфігурації верхнього рівня, а саме нерухоме крило та поворотне крило. Обидва типи представляють конкретні переваги та надані системи управління та наведення [11].

БПЛА з нерухомими крилами - це жорсткі крила з крилами, які дозволяють літати на базі підйомника, створюваного вдосконаленим повітрям. Навігаційний контроль здійснюється за допомогою контрольних поверхонь на крилах (ліфти, ліфти, рульові колеса). Технологія аеродинаміки підтримує більшу витривалість та польовий переліт, а також дозволяє швидко рухатися. Крім того, ці транспортні засоби витримують більші навантаження в порівнянні з обертовими крилами транспортних засобів. Однак ці платформи вимагають зльоту та посадки на злітно-посадкову смугу і не можуть виконувати місії на висоті, оскільки їм потрібно постійно рухатися під час місій [12].

Обертові крила забезпечують переваги маневреності обертових лопатей. Ці платформи здатні здійснювати вертикальний зліт і посадку (VTOL), польоти на низькій висоті та місії наведення. Застосування обертових лопатей створює аеродинамічну тягу і не вимагає відносної швидкості [11]. Цю повітряну платформу також можна розділити на однороторні (вертоліт) та багатороторні (чотирироторні та шестироторні).

Один ротор має два ротори, основний ротор використовується для навігації, а хвіст - для управління курсом. Ці транспортні засоби, як правило, здатні злітати і приземлятися вертикально і не потребують руху повітря на лопатках для руху вперед. Навпаки, лезо саме генерує необхідний потік повітря. Порівняно з мультироторним, газовий двигун дозволяє літати більш довговічно. Цей тип транспортного засобу може нести великі навантаження, такі як датчики та маніпулятори при виконанні завдань, що зависають, і при виконанні міжміських польотів у повітрі, але ці платформи представляють механічну

складність та збільшують витрати [12]. Механічна та електронна складність дуже низька, оскільки ці компоненти є абстрактними в контролерах польоту та двигуна [12]. Існує також гібридний безпілотник, який є спеціальною повітряною платформою, що включає переваги нерухомого крила та поворотного крила, з можливістю вертикального зльоту та посадки, високою швидкістю польоту та збільшеним часом польоту. Ці транспортні засоби можна розділити на транспортні та ззаду. Перший складається з гібридної платформи, яка виконує основні маневри, утримуючи базову лінію БПЛА в горизонтальному напрямку. Останній є платформою, яка може злітати вертикально і приземлятися на хвіст, і може нахилитися вперед для досягнення горизонтального польоту [13]. Нарешті, враховуючи вимоги місії (такі як зріст та витривалість), в літературі можна знайти інші типи класифікацій, що стосуються безпілотників. У цих випадках повітряні платформи можна класифікувати за низькою, середньою та великою висотою, а також за короткостроковою та довгостроковою довговічністю [14].

1.2 Огляд відомих методів побудови траєкторії польоту БПЛА

У літературі було проведено велике дослідження щодо досліджень, що стосуються контролю, сприйняття та управління БПЛА, таких як системний метод ідентифікації недорогих БПЛА [15], планування траєкторії з диференціальними обмеженнями та відсутність перешкод [16]. , Автономне наведення БЛА в умовах невизначеності [17], технологія навігації та управління вертольотами [11], а також сприйняття та оцінка стану БЛА [12]. Враховуючи конкретне застосування, Каністра та ін. [18] запропонував дослідження дронів для моніторингу та контролю дорожнього руху, тоді як в [19] розглянуто технологію безпілотників для точного землеробства.

У [9] пропонується рішення щодо ПТП для мобільних роботів, і автор класифікує метод як евристичний або повний. В евристичних методах робот дотримується набору простих правил, що визначають його поведінку, але такі методи не можуть гарантувати точність траєкторії польоту. З іншого боку,

повний підхід може надати ці гарантії шляхом розпаду клітин навколишнього середовища, що включає вибірку простору в клітини для спрощення шляху польоту в кожному субрегіоні. Ще одне важливе питання, про яке згадує автор, - це час польоту, який можна мінімізувати за допомогою декількох роботів і зменшити кількість маневрів. Нарешті, автор висвітлює наявну інформацію про навколишнє середовище. Існує кілька способів розпізнати попередні знання робота про область пошуку (в автономному режимі), тоді як сенсорні методи можуть отримувати таку інформацію в режимі реального часу під час польоту (в режимі онлайн).

В останньому рішенні про дорожньо-транспортні пригоди пропонується декілька методів і прийомів, які в основному використовують наземні транспортні засоби для виконання завдань [21]. Беручи до уваги невизначеність у дослідженні, [22] в даний час вивчає стратегію відображення для невідомих середовищ. Робот повинен самостійно досліджувати навколишнє середовище для збору даних та створення навігаційної карти. За відсутності інформації про глобальне позиціонування, необхідно використовувати технологію одночасного позиціонування та картографування для постійного коригування оцінки позиціонування та орієнтації робота. Для зменшення часу виявлення або покращення якості карти можна використовувати кілька роботів, але для цього потрібна стратегічна координація. У цій стратегії роботи можуть ділитися ідеями та складати спільну карту робочого простору.

Існуючі рішення, пов'язані з безпілотниками, можуть вирішити такі важливі питання, як контроль, сприйняття та лідерство. Деякі розслідування вирішували проблему дорожньо-транспортних пригод, але лише розглядали наземні транспортні засоби та коротко згадували безпілотники як продовження цих транспортних засобів. Незважаючи на те, що переглянуте в попередньому дослідженні метод наземної розвідки може бути розширений і на безпілотники, є кілька інших аспектів, які необхідно враховувати при роботі з літаками, такі як фізичні характеристики транспортного засобу, довговічність, обмеження маневреності, обмежена корисна навантаження та зовнішні стан навколишнього

середовища. . Бортові камери та датчики збільшать вагу автомобіля та зменшать довговічність, яка є досить обмеженою, особливо у мультироторів. У цьому транспортному засобі час автономної роботи становить приблизно 20-25 хвилин, навіть у більш складних моделях [23]. Крім того, поворотні маневри [24,25] та вітрові електростанції [26] збільшать енергоспоживання зовнішніх завдань.

Класичний метод класифікації, визначений у [9], використовується для класифікації існуючих методів відповідно до методу розкладання клітин. Розглянемо метод нерозкладання та метод використання точного та наближеного розкладання елементів. Останній також називають методом на основі сітки, який поділяється на дві частини: повну інформацію та часткову інформацію.

1.3 Планування траєкторії польоту

Враховуючи задачу планування польоту, яка складається із вільного простору робота та його меж, проблема ПТП полягає в плануванні шляху, який охоплює все цільове середовище, враховуючи обмеження руху та характеристики датчиків транспортних засобів, уникаючи при цьому перешкод. У повітрі перешкодами в робочій зоні можуть бути зони поза польотом (NFZ), які безпілотник не повинен враховувати на етапі планування, наприклад, райони біля аеропортів або покинуті будівлі.

За допомогою методів розкладання цільове середовище, як правило, поділяється на розділені ділянки, які називаються клітинами. Розмір та роздільна здатність комірок може змінюватися залежно від типу розкладання, і для забезпечення точності траєкторії польоту повинні бути прийняті конкретні стратегії. Більшу одиницю потрібно перемістити лише один раз, щоб повністю охопити одиницю, тоді як меншу одиницю потрібно перемістити лише один раз. Ці одиниці зазвичай мають однаковий розмір робота (опорний шлях шляху) або пропорційний діапазону датчика (опорний шлях повітряного шляху), який є лише однією точкою передбачуваного шляху (рис.1.1).

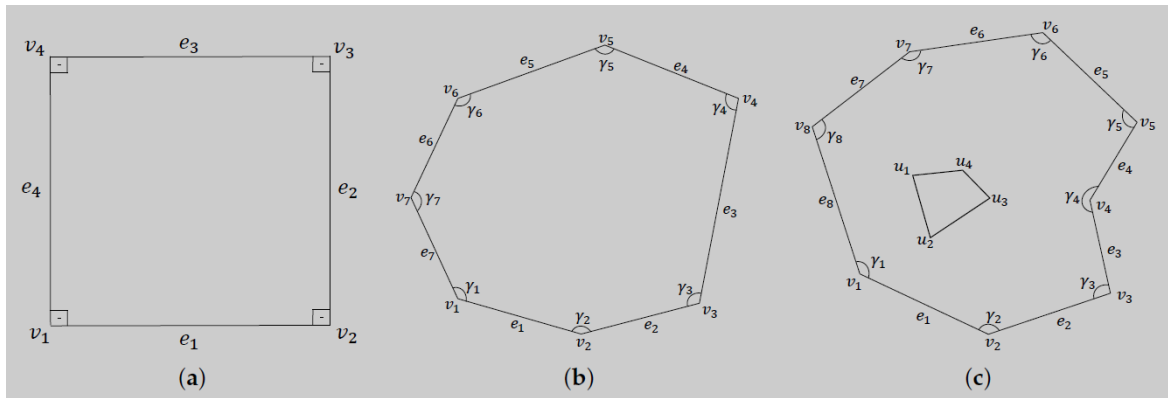


Рисунок 1.1 - Форми польоту: (а) Прямокутні; (б) Опуклий багатокутник; (с) багатокутник із зонами

Форма польоту є відповідним фактором, який слід враховувати при плануванні шляху польоту. Деякі методи досліджують лише прямокутні ділянки або спрощують форму поперечного перерізу до прямокутника, тоді як інші методи підтримують більш складні форми, такі як клини та опуклі багатокутники, які представляють неправильні ділянки. Деякі методи можуть навіть обробляти цікаві області, що містять NFZ, яких слід уникати під час польоту. Ці зони, що не виконують польоти, можуть представляти райони, де взагалі не потрібно встановлювати траєкторію польоту, або райони, де безпілотникам заборонено літати. Зазвичай використовуються різні методи розкладання, щоб зменшити розрив складної зони або розділити область на менші одиниці, щоб полегшити місію траєкторії польоту.

1.4 Клітинне розкладання

Однією з головних проблем при встановленні маршруту польоту є забезпечення повного охоплення місця події. Зазвичай це досягається за допомогою розкладання клітини в цікавій зоні для розподілу цільового вільного простору на клітини для спрощення завдання траєкторії польоту [9]. У літературі існує безліч методів розпаду клітин, і найпоширенішою проблемою ПТП за участю дронів є точний та приблизний розпад клітин.

Точне розкладання одиниць включає розподіл робочої зони на підплощі (їх також називають одиницями), підплощі яких точно займають цільову область. Ці

клітини зазвичай вивчаються за допомогою минулого та простих рухів минулого. Отже, проблема ПТП може бути зведена до планування руху з однієї комірки в іншу [9]. Ці рухи виконуються між сусідніми клітинами, що мають спільну межу. З огляду на представлення графіка прибуття, вузли можуть представляти комірки, а ребра можуть ідентифікувати сусідні комірки, як показано на рисунку 1.2. Тому створіть розкладену одиницю, проводячи лінію з боку в бік в зоні польоту.

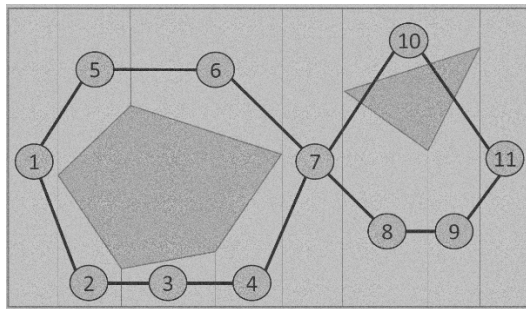


Рисунок 1.2 - Робоча область польоту, розділена на клітинки.

Варто згадати, що існує два методи розкладання клітин: трапецієподібне розкладання та розкладання зубця Т, як показано на рисунку 1.3а та б відповідно. Спочатку область, що цікавить, поділяється на опуклі трапецієподібні клітини, що рухаються вперед і назад, і за допомогою детальної ходьби визначають порядок дослідження клітин для виконання завдання траєкторії польоту. Останній розглядає лише вершини перешкод для створення неопуклих більших пікселів.

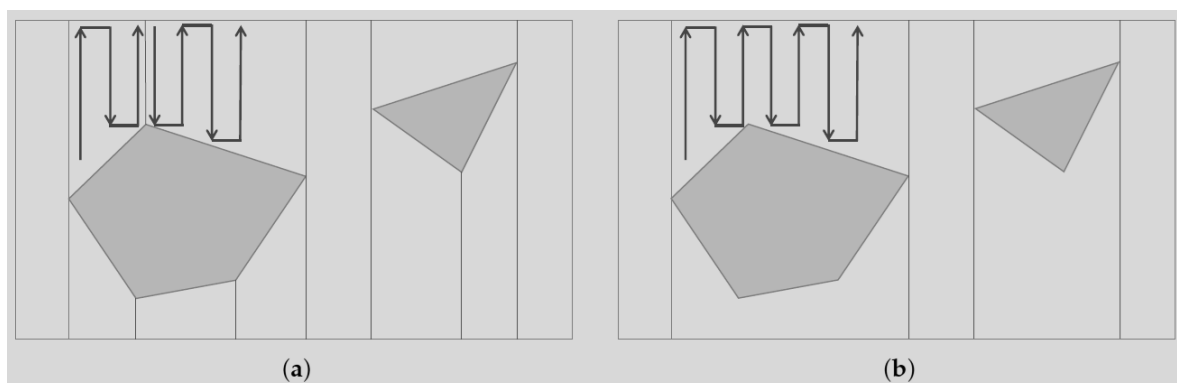


Рисунок 1.3 - Два типи точного клітинного розкладання: (а) трапецієподібне розкладання; (б) розкладання бустрофедону.

Приблизна методика розпаду клітин дискретизує ділянку в групу нормальних клітин [9]. Ці нормальні одиниці зазвичай квадратні, але вони також можуть бути представлені трикутниками або шестикутниками. Метод, заснований на сітці, може бути застосований до приблизної області для створення контуру установки траєкторії польоту [21]. Розглядаючи завдання траєкторії польоту наземного робота, розмір блоку зазвичай відповідає розміру робота. Однак у повітрі дрон летить на певній висоті над землею і використовує камеру як датчик для виконання завдань. У цьому випадку розмір пікселя пропорційний траєкторії руху камери в БПЛА, як показано на рисунку 1.4а, а роздільна здатність сітки залежить від вимог до зображення, таких як роздільна здатність і швидкість перекриття та характеристики датчик зображення.

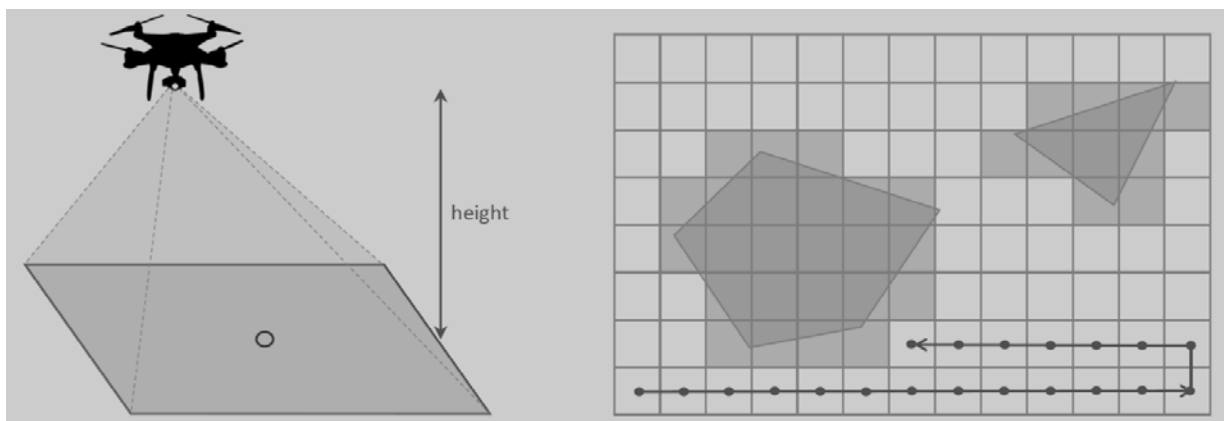


Рисунок 1.4 - Приблизне розкладання клітин: (а) проєктована область; (б) звичайна сітка з точками шляху

Траєкторія польоту БПЛА складається з набору пунктів точки $k\{w_1, \dots, w_k\}$. Кожна маршрутна точка w_i представляє навігаційну команду транспортного засобу, таку як зліт, зміна швидкості або переміщення до певного місця, і містить інформацію про широту, довготу та висоту. Оскільки маршрутна точка має всю необхідну інформацію про положення, необхідну для керування транспортним засобом, а піксель пропорційний розміру камери, можна спростити цю проблему, припустивши, що центр кожного пікселя вказує на маршрутну точку, як показано на рисунку 4б.

1.5 Показники ефективності польоту

Алгоритм траєкторії польоту повинен враховувати декілька питань для забезпечення успіху місії шляху польоту, таких як складність зони, що цікавить, чи є зона польоту, та можливість використовувати методи розкладання клітин. Крім того, алгоритм позначення шляху польоту повинен генерувати шлях позначення шляху польоту відповідно до вимог програми. Наприклад, основною метою фотограмметричної програми є створення позитивної мозаїки, складеної з набору аерофотознімків, що перекриваються. У цьому випадку вимога програми полягає у забезпеченні необхідної кількості перекриття спереду та збоку на фотографії. Іншою необхідною вимогою для цього типу додатків є роздільна здатність, яка може бути розрахована як відстань до землі (GSD) [27]. GSD - це довжина сторони, що відповідає пікселю на зображенні на землі, або відстань між центрами пікселів, виміряних на землі. Чим менша висота польоту безпілота, чим менший GSD, тим краща якість зображення.

Отже, показники ефективності, що використовуються для оцінки рішення кандидата щодо маршруту польоту, повинні відповідати вимогам плану. Крім того, вам слід розглянути прості або постійні завдання для шляху польоту. При простому покритті область інтересу охоплюється лише один раз, тоді як при постійному покритті сценарій сканується кілька разів. В обох випадках встановлення траєкторії польоту може виконуватися одним або кількома транспортними засобами. Конкретні показники безперервного охоплення включають кількість виявлених об'єктів / подій, інтервал і частоту відвідувань у кожній одиниці середовища, середнє квадратичне значення інтервалу (QMI) [28] і стандартне відхилення частоти (SDF) [28].

У простому районі покриття найпоширенішими показниками ефективності, які зустрічаються в літературі, є: загальна відстань проїзду або довжина шляху [29,30], час місії [31,32], максимальна та максимальна зона встановлення шляху польоту [33] та кількість польоти Кермова дія [34,35]. Мінімізація тривалості траєкторії польоту є компромісом для максимізації

регіональної траєкторії польоту. У робочому просторі, відокремленому розкладанням одиниці, довжина шляху повинна бути не тільки складена всередині кожного блоку, але і в середньому шляху між сусідніми одиницями, тобто шлях, що з'єднує кінець одного блоку і початок наступного блоку.

Використання декількох роботів може зменшити час польоту, встановивши траєкторію польоту. Однак для робота можна розглянути площу, охоплену кожною одиницею шляху. Мінімізація цієї кількості може скоротити час завершення роботи однієї машини та робота [9]. Використання кількох роботів часто вимагає скоординованого процесу, який передбачає розподіл зони інтересу та розподіл отриманих підзон між дронами. Беручи до уваги відносні можливості транспортних засобів, робоче місце можна розділити та розподілити на два різні етапи [34], або одночасно розподілити за допомогою розподіленого методу [10], використовуючи протокол узгодження. Однак повне рішення, що включає оптимальне розкладання домену, розподіл та ефективне встановлення траєкторії польоту, є спільною проблемою управління, яка зазвичай класифікується як NP-важка. Крім того, після того, як транспортний засіб залишає завдання або місце події змінюється, повинен бути проведений процес реконфігурації для розподілу та розподілу деталей до інших транспортних засобів. Тому, враховуючи, що транспортні засоби літають на різній висоті, щоб уникнути зіткнень, багато досліджень спростили цю задачу.

Автори часто пов'язують такі показники, як довжина шляху, час місії та кількість поворотів із енергоспоживанням, намагаючись мінімізувати їх для економії енергії. Однак, щоб ефективно економити енергію на безпілотнику, слід вивчити більше функцій, таких як рух транспортного засобу та обмеження, кути повороту та оптимальні швидкості. Відповідно до [25], різні відстані можуть мати різні оптимальні швидкості, а споживання енергії може бути мінімізовано відповідно до їх довжини. Тому, як основне технічне обмеження для використання БПЛА, споживання енергії привернуло увагу дослідників [24,31-33,36] і стало основним критерієм оптимізації через обмежену довговічність БПЛА при плануванні режимів польоту.

1.6 Методи задання траєкторії без розкладання

Місії з охоплення, що виконуються над звичайними і районами, що представляють інтерес з одним БПЛА, зазвичай не вимагають будь-якого типу розкладання. Для вивчення таких областей достатньо простих геометричних візерунків. Найбільш поширеними візерунками є зворотний і вісь (BF) і спіральний (SP). Перший прийнятний Планувальником місії, найпопулярнішим програмним забезпеченням для управління польотом [37], щоб забезпечити задання траєкторії польоту місцевості за стандартною схемою. У цій схемі рухи складаються з прямих ліній, схрещених в обох напрямках з маневрами закритого кута в кінці кожного раунду. Останній зазвичай виконує рухи, проходячи повз зовнішні вершини площі і зменшуючи радіус до центральної точки.

Згідно з [38], деякі методи досліджують лише прямокутні перерізи, тут автор порівнює різні типи режимів польоту. У цій роботі, як показано на рисунку 1.5а, b, і її краще класифікувати на паралельні лінії та лінії повзання у зворотному порядку, коли область пошуку велика і немає інформації про можливе перетин цілі.

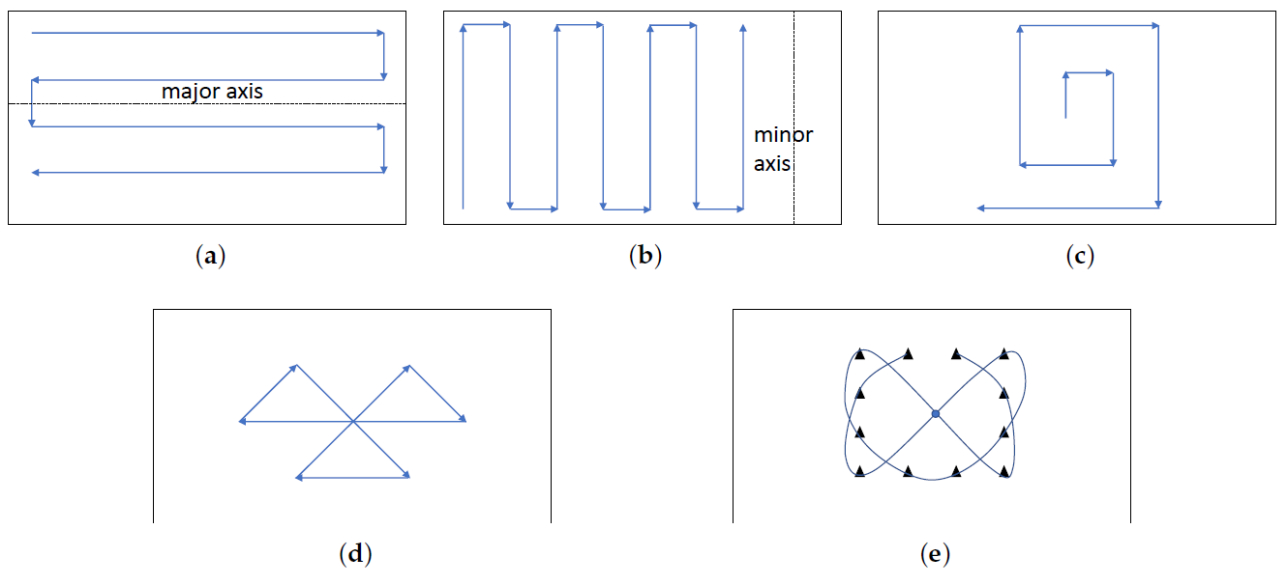


Рисунок 1.5 - Шаблони польоту в областях без розкладання: (а) паралельні; (b) повзуча лінія; (c) квадрат; (d) пошук у секторі; (e) Barrier Patrol.

Схема пошуку секторів, показана на рис. 5г, складається з прямої лінії, яка обертається на 120 градусів праворуч, коли транспортний засіб досягає межі області. Після трьох секторів контур повернеться до вихідної точки центру області. Потім повторить ту ж схему зі зміщенням на 30° . Патрулювання перешкод включає визначення 12 пунктів, які швидко розподіляються в зоні пошуку, як показано на рисунку 1.5e. Транспортний засіб починає свою траєкторію руху у вихідній точці і рухається круговими рухами до наступної точки. З цього моменту він вже не продовжує кругову траєкторію, а доходить до точки ближче до правого кута і досягає центральної точки.

В [39] запропонували аналіз впливу впливу вітру на метод встановлення маршруту польоту доменної печі під час реалізації безпілотного літального апарата з нерухомими крилами. Використовуючи кругову область інтересу, охоплену рухом BF, автор вивчив різні напрямки розгортки від 0 до 360 градусів із кроком 10 градусів із шістьма різними швидкостями заздалегідь заданих змін напрямку вітру. Відповідно до імітаційного експерименту, напрямок шляху польоту повинен бути перпендикулярним напрямку вітру, щоб мінімізувати час польоту. Однак на операцію повороту безпосередньо впливає вибір вертикального напрямку (за годинниковою стрілкою або проти годинникової стрілки). Автор вважає, що в більш складних випадках розкладання на клітини відстань між цими клітинами має більший вплив на час польоту, ніж напрямок вітру.

В [25] запропонували метод вимірювання енергії ПТП (E-BF) для вимірювання енергії, енергія та роздільна здатність якого обмежені завданням. Таким чином, алгоритм визначає найкращу конфігурацію зворотного руху на максимальній висоті, виходячи з обмеження роздільної здатності, тим самим мінімізуючи кількість поворотів. Автор стверджує, що енергію, необхідну для польоту з оптимальною швидкістю, можна мінімізувати. Оптимальна швидкість залежить від пройденої відстані. Алгоритм знаходить першу вершину найдовшого ребра і обчислює паралельний їй напрямок сканування. Потім він обчислює послідовні точки шляху та ставки, що перекриваються. Нарешті, пряма

лінія вказує на початкову вершину. Як показано на рисунку 1.6а, також пропонується вдосконалений алгоритм, щоб уникнути попередньої області. Повертаючи кількість смуг рівномірно і швидко, шлях зворотного шляху також може бути використаний як шлях сканування, на що вказують запропоновані помилки в автономному режимі та в Інтернеті. Перший - перевірити, чи достатньо заряду акумулятора для виконання завдання. Останній є польотом і постійно аналізує, чи можна використати решту енергії як вихідну точку.

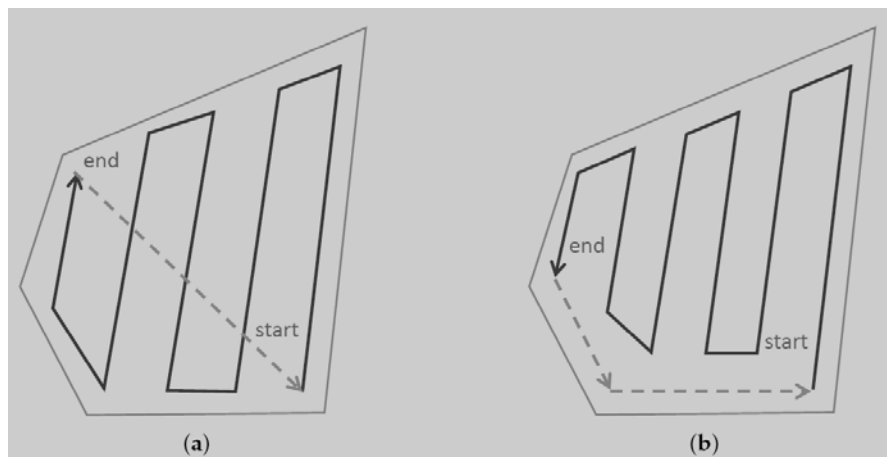


Рисунок 1.6 - Алгоритм планування траєкторії польоту з урахуванням енергії:
(а) Непарна кількість смуг; (б) Парна кількість смуг

В [40] запропоновано енергозберігаючий спіральний алгоритм ПТП (E-Spiral). Алгоритм передбачає побудову траєкторії польоту через кожну вершину регіону. Після завершення першого шару встановлення траєкторії польоту алгоритм повинен зменшити радіус переміщення літака до центральної точки, як показано на рисунку 1.7. Кут дії, що виконується алгоритмом, більший, і немає необхідності зменшувати швидкість до нуля на кожному кроці, що зменшує час прискорення та уповільнення. Така поведінка може підтримувати оптимальну швидкість, що використовується на прямій ділянці шляху, протягом більш тривалого періоду часу, а отже, має більш високий енергозберігаючий ефект, ніж [25]. В [25] та в [40] прийняли енергетичну модель, запропоновану в [24], яка була отримана з фактичних вимірювань. Ці методи порівнювали в моделюванні, проведеному в 3750 різних опуклих областях багатокутника, з різними ознаками

(такими як кутові точки, нерівності та розміри). Автор також проводив реальні експерименти з використанням квадаторів, використовуючи прямокутні та багатокутні схеми для аналізу енергії, споживаної під час місії. E-Spiral долає E-BF, незалежно від того, чи це імітований політ, або фактичний політ, а враховуючи енергію, спожиту під час місії, її можна вважати найбільш ефективним методом ПТП для опуклих багатокутних областей.

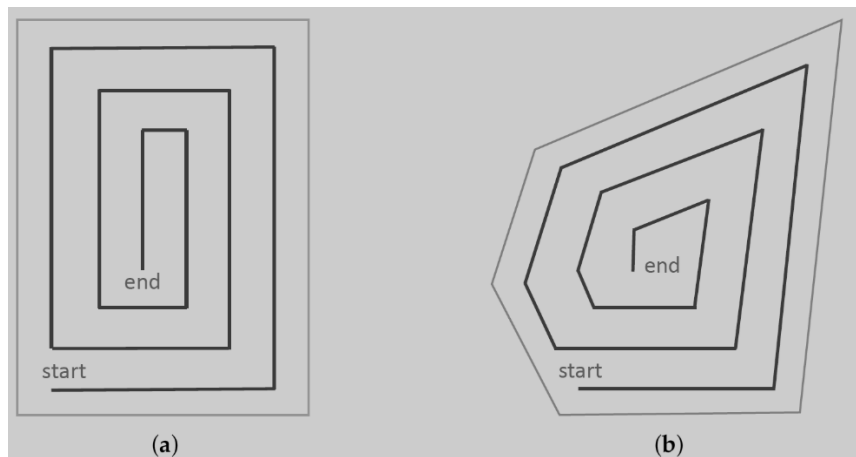


Рисунок 1.7 - Спіральний алгоритм: (а) Прямокутна область; (б) Багатокутна область

У роботі [25] запропоновано триступеневий алгоритм ПТП для БПЛА. Автор вивчав такі важливі функції, як корисне навантаження та зміни потужності. Першим кроком є використання контрольно-пропускних пунктів для побудови тривимірної моделі місцевості для отримання моделі аналізу. Далі розраховується стабільне споживання енергії з урахуванням зльоту, швидкості польоту та тертя повітря. Автор вважає, що транспортний засіб рухається з постійною швидкістю у стабільному стані, і вивчив оптимальну швидкість для мінімізації енергії.

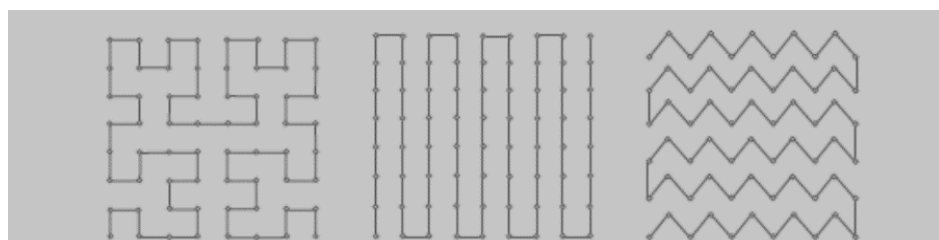


Рисунок 1.8 - Звичайні шляхи ПТП: (а) Криві Гільберта; (b) SCAN; (c) LMAT.

Автор [41] використовує змішане цілочисельне лінійне програмування (MILP), щоб вказати траєкторію польоту місії в прямокутній області за участю безпілота. Бортова камера безпілота розгляне шлях розподілу в певній зоні, щоб отримати повне посилення на траєкторію польоту. Автор спрощує проблему, враховуючи лише прямокутні перешкоди та розміщуючи літаки в певних районах району, не використовуючи ніякого типу розкладання доменів і не вирішуючи належним чином проблему уникнення зіткнень між БПЛА.

Враховуючи різні обмежені ситуації, такі як порядок відвідування доріг та зменшення діапазону камер, були проведені імітаційні експерименти для оцінки запропонованого рішення.

1.7 Методи задання траєкторії точного клітинного розкладання

Залежно від розміру та складності робочого простору, ви можете застосувати точну розбивку одиниць в зоні, що нас цікавить. За допомогою цієї техніки ділянку неправильної форми поділяють на кілька підділянок, щоб зменшити нахил та спростити регулювання траєкторії польоту. Ці підзони можуть бути покриті одним або кількома безпілотами. У першому випадку метод ПТП повинен бути пов'язаний із траєкторією польоту в кожному субрегіоні та проміжним шляхом, що з'єднує ці субрегіони. В останньому, метод ПТП повинен враховувати відносні можливості кожного транспортного засобу для обчислення розміру кожного субрегіону. Крім того, слід враховувати запас міцності для запобігання зіткненню транспортних засобів.

1.8 Методи задання траєкторії єдиної стратегії

В [43] подано метод розгляду клітинного розпаду цих полігональних областей. Спочатку робоче середовище було розкладено на непророкаційні субрегіони за допомогою методу мінімальної ширини суми, і вивчався жадібний

рекурсивний метод, запропонований раніше в [45]. Потім здійснюється зворотний рух перпендикулярно напрямку розгортки (тобто мінімальна відстань між краєм і верхом) для мінімізації рульового руху [46]. Як показано на рисунку 9b, дві підрегіони повністю суміжні і мають однаковий напрямок розгортки, отриманий в результаті опуклого розкладання (рисунок 1.9a), і об'єднані в підкатегорію P4, щоб уникнути непотрібного розвороту та входу. Ви також можете змінити напрямок руху від однієї підгрупи до іншої, щоб покращити налаштування траєкторії польоту, як показано на рисунку 1.9c. Нарешті, визначте найкращий порядок підгрупи для поєднання кінцевої траєкторії, як показано на рисунку 1.9d.

В [35] запропоновано метод встановлення траєкторії польоту, який вивчає точне розкладання опуклих і відбивних поверхневих клітин. Метою автора є фотографування за допомогою літаків для досягнення 3D-реконструкції. Опуклі багатокутники можна переміщати рухом BF відповідно до найкращого напрямку. Однак у більш складних областях (наприклад, на можливих багатокутниках) необхідно перевірити, чи можна виконати завдання таким же чином, і між смугами немає зазорів, тобто жодна з смуг не проходить через частини поза багатокутник. Ця конкретна ситуація показана на рисунку 1.10a. Коли контур переривається, як показано на рисунку 1.10b, використовується точне розкладання багатокутника для спрощення області.

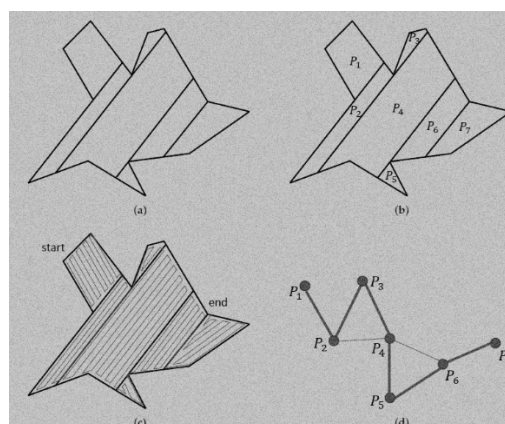


Рисунок 1.9 - Розкладання та охоплення увігнутих багатокутників: (a) опукле розкладання; (b) комбінація; (c) покриття; (d) Неорыэнтований граф

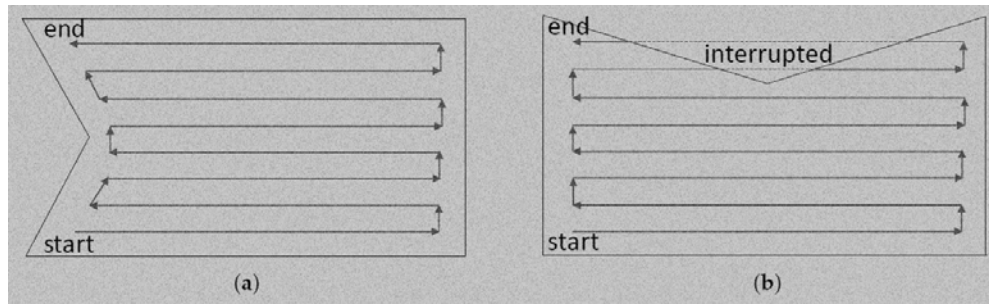


Рисунок 1.10 -- Задання траєкторії польоту з використанням візерунка в багатокутниках: (a) контур; (b) перерваний шлях

Як тільки буде визначено найкращий напрямок сорту для кожної підгрупи, будуть розглянуті два критерії, пов'язані з напрямком та напрямком, та вивчено чотири різні альтернативи. Перше завдання для перевірки траєкторії польоту буде слідувати найкращому напрямку руху, і навпаки. Останній враховує напрямок першої дії повороту, за годинниковою стрілкою (поверніть ліворуч) або проти годинникової стрілки (поверніть праворуч). Альтернатива впливає на відстань переходу, тобто відстань між останньою точкою даної підкатегорії А і першою точкою даної підкатегорії В. Після завершення експлуатації маршрутного рейсу цей метод безпосередньо з'єднує кінцеву точку з першою кінцевою точкою прямою лінією.

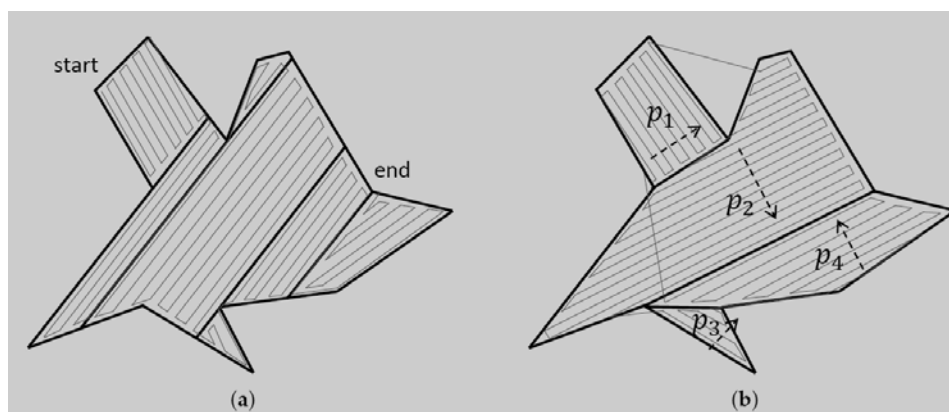


Рисунок 1.11 - Порівняння підходів розкладання в усюди: (a) опуклий шлях [44]; (b) спад і опуклості розкладання [35]

В [47] запропоновано метод планування траєкторії руху БПЛА з нерухомими крилами, що використовуються для вивчення вітру, щоб скоротити

час польоту. Автори додали вітру до моделі, щоб розрахувати метод задання траєкторії польоту [39], і продовжили свою роботу, запропонувавши метод розкладання, щоб розділити складну область на опуклі багатокутники. Область, що нас цікавить, розкладається трапецієподібним розкладом, і досліджуються кола багатокутника. Використовуйте динамічне програмування для реорганізації комірок для об'єднання комірок у опуклі багатокутники. Важливо зазначити, що цей метод розкладання також враховує необов'язкові одиниці за межами області, так що можна знайти різні розклади з меншим часом польоту.

Запропонований метод (FTIW) порівнюється з попереднім методом, який спрямований на мінімізацію кількості витків (NT) [44] та загальної висоти (MSA) [46]. Використовуючи поле поля та кілька випадкових полігонів, створених за допомогою моделювання Монте-Карло, автори вважають, що їх метод FTIW долає попередні два методи, враховуючи час польоту, необхідний для виконання місії шляху польоту.

Алгоритми, запропоновані для вирішення проблем задання траєкторії польоту БПЛА, зазвичай пов'язані зі стадією планування отримання траєкторій польоту на основі показників ефективності. При використанні БПЛА з фіксованим крилом підхід також повинен враховувати обмеження руху таких транспортних засобів з метою планування можливих траєкторій. Однак ці методи не враховують важливих завдань управління, таких як відстеження шляху або траєкторії, і покладаються лише на внутрішній контролер БПЛА для виконання запланованого маршруту в реальному польоті.

1.9 Висновки та постановка задачі

У розділі досліджено методи планування траєкторії польоту безпілотних літальних апаратів, рішення для простих геометричних моделей польоту (наприклад, вперед і назад і спіраль), а також більш складні рішення на основі сіток, які враховують повну та часткову інформацію, пов'язану з БПЛА. Обмежена точність безпілотників є головною проблемою, що вимагає більш складних завдань покриття. Виявлено, що певні методи використовують кілька літальних апаратів у цій місії для покращення ефективності встановлення траєкторії польоту, тим самим відокремлюючи енергетичні потреби довгострокового та довгострокового польоту в більш здійсненному польоті. Однак ця технологія зазвичай вимагає обчислювальної складності для вирішення проблем координації та спілкування. У цьому контексті було зроблено висновок про необхідність розроблення нових рішень, які б вирішили виявлені недоліки відомих методів.

Одним з способів є залучення апарату штучного інтелекту, який би дозволив створити інтелектуалізовану систему управління БПЛА.

Таким чином, необхідним є удосконалення методу керування безпілотними літальними апаратами, який на відміну від відомих застосовує компоненти штучного інтелекту, і який забезпечує підвищення точності траєкторії руху безпілотного літального апарату типу дрон, а також надати подальший розвиток інтелектуалізованої системи керування безпілотними літальними апаратами, яка забезпечує підвищення точності траєкторії руху безпілотного літального апарату типу дрон.

Для вирішення завдання необхідним є

1. розробити модель задання траєкторії руху безпілотного літального апарату типу дрон;
2. розробити метод керування безпілотними літальними апаратами, який на відміну від відомих застосовує компоненти штучного інтелекту, і який

забезпечує підвищення точності траєкторії руху безпілотного літального апарату типу дрон;

3. розробити інтелектуалізовану систему керування безпілотними літальними апаратами, яка забезпечує підвищення точності траєкторії руху безпілотного літального апарату типу дрон.

2 ПОБУДОВА МАТЕМАТИЧНОЇ МОДЕЛІ СИСТЕМИ КЕРУВАННЯ БЕЗПІЛОТНИМИ ЛІТАЛЬНИМИ АПАРАТАМИ

2.1 Математична модель системи керування безпілотними літальними апаратами

Розробка малогабаритних легких електронних компонентів, що підвищують швидкість та надійність комунікаційних технологій та вдосконалюють алгоритми управління командами та системи передачі даних, зробила квадаторні безпілотні літальні апарати популярними серед дослідників, академіків та військових користувачів.

Цей тип літальних апаратів використовувався дослідником в університетах для цілей тестування та оцінки в багатьох областях, таких як теорія управління польотом, навігація, системи реального часу та робототехніка.

БПЛА також використовуються силами безпеки для військових цілей, включаючи відстеження, розвідку та пошуково-рятувальні роботи в містах.

Квадрокоптер - це вертоліт із чотирма роторами, встановленими симетрично. Таким чином, рух квадрокоптера схожий на звичайний вертоліт.

Різниця полягає в тому, що рух досягається шляхом варіювання кожної з чотирьох швидкостей двигуна, щоб отримати бажаний ефект, який викликає рух БРЛА.

БРЛА може злітати і приземлятися вертикально і парити в певному положенні в повітрі, а також літати на низьких швидкостях.

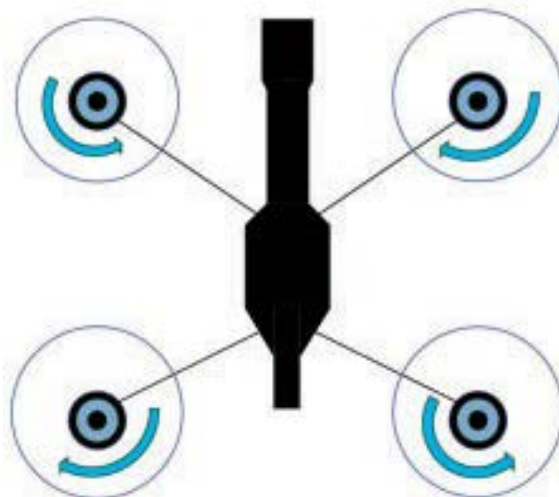
Інші переваги цього літального апарату включають високу маневреність, можливість проектування безпілотного варіанту менших розмірів, знижену механічну складність через відсутність рухомих частин, крім гвинтів, легку перевірку платформи в лабораторії та низьку вартість.

Вертикальний рух квадаторотора є результатом одночасного збільшення або зменшення швидкості обертання всіх роторів.

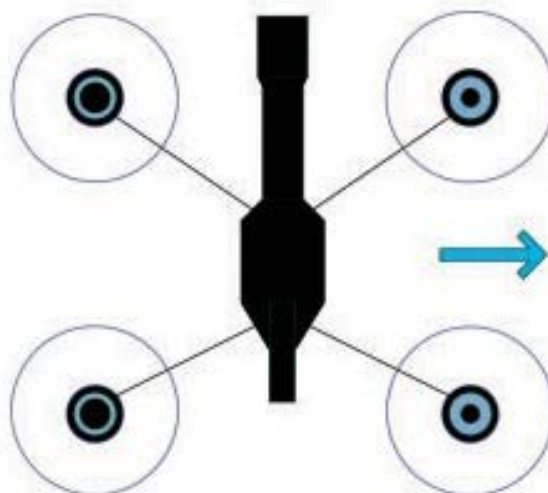
Рух вздовж будь-якого напрямку на бічній осі досягається зменшенням швидкості обертання роторів уздовж бажаного напрямку руху та збільшенням

швидкості обертання роторів, протилежних бажаному напрямку руху траєкторії БПЛА.

Поздовжній рух та рух похилу досягаються моментом роторів, які обертаються з різною швидкістю так само, як і поперечний рух.



a)



б)

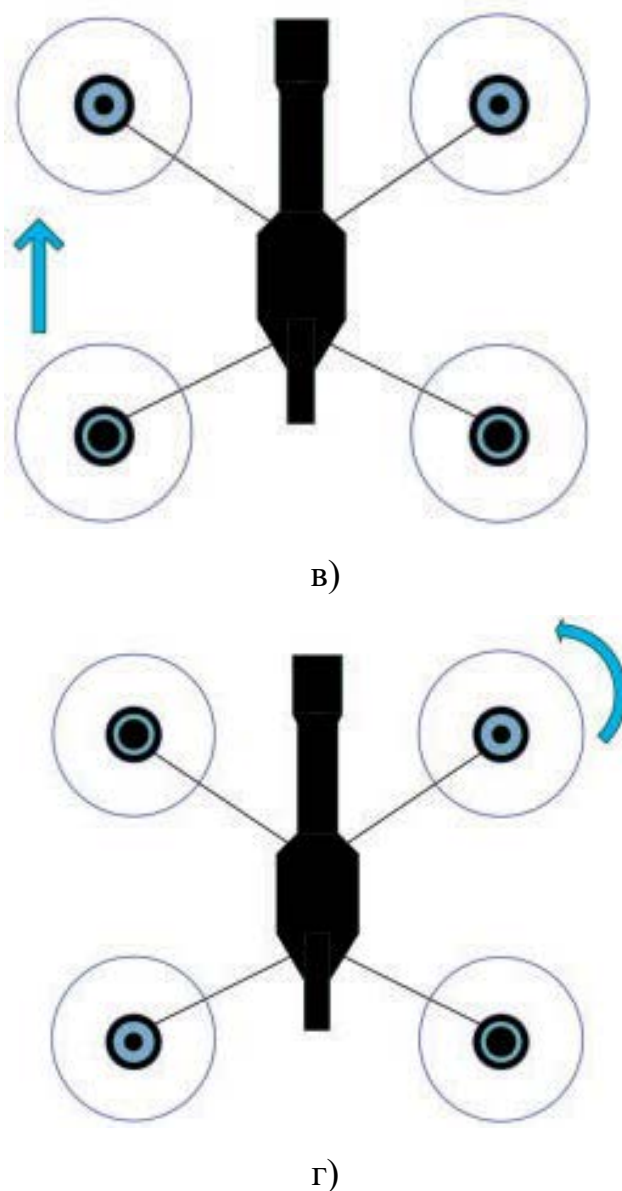


Рисунок 2.1 - Основний рух квадродотора. CCW: проти годинникової стрілки

Вісь руху квадродотора з шістьма ступенями свободи (6-DOF) показана на рисунку 2.2.

Система відліку та абстрактні входи управління (f_i) пов'язані з моделлю літального апарату.

Інерційна і основна осі тіла представлені x, y, z та i, j, k відповідно.

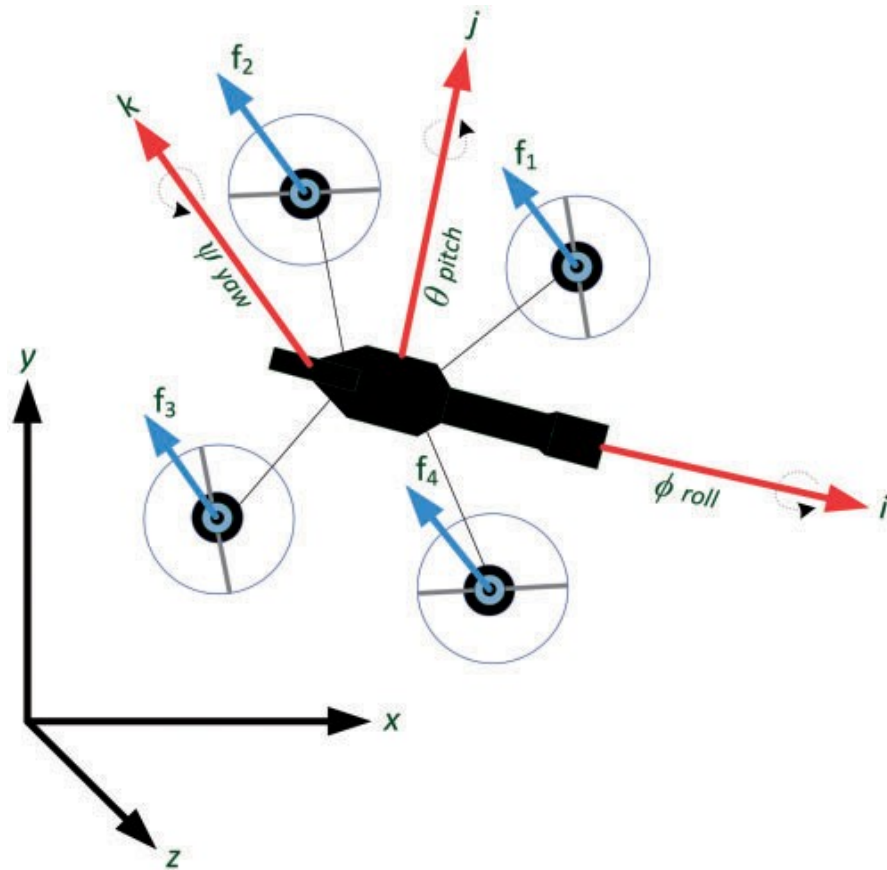


Рисунок 2.2 - 6-DOF модель квадродрона з системою відліку.

Висотний рух (z) створюється в напрямку осі x в результаті обертового руху навколо осі y .

Рух польоту здійснюється в напрямку осі y в результаті обертового руху навколо осі x .

Поворотний рух - це обертовий рух навколо осі z . Таким чином, квадродрон обертається у напрямку вертикальної осі.

Квадродрон піднімається або опускається на вісь z за допомогою дроселя. Таким чином, квадродрону дано змогу літати до будь-якої заданої точки простору або послідовністю різних рухів, або комбінуючи кілька рухів БПЛА разом.

Встановлено зв'язок між віссю опорного корпусу та інерційною опорною віссю літального апарату через усі вектори положення x , y , z та кути Ейлера θ , ϕ , ψ .

БРЛА вважається жорстким і симетричним.

Однак центр мас, центр ваги та точка початку осі тіла вважаються збіжними.

В цих умовах рівняння (2.1) - (2.3) забезпечують математичний вираз матриці $T_{z,y,x}$ для опису перетворення від осі тіла до інерційної осі.

Тут і в подальшому використанні математичні позначення c та s представляють косинусну функцію та синусоїду відповідно.

$$T_{z,y,x} = T_z \cdot T_y \cdot T_x \quad (2.1)$$

$$T_{z,y,x} = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \quad (2.2)$$

$$T_{z,y,x} = \begin{bmatrix} c\theta c\psi & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ c\theta s\psi & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (2.3)$$

Тяга і моменти в протилежному напрямку обертання виникають на вертикальній осі літака в результаті обертального руху кожного ротора квадроротора.

Ротори відповідають взаємністю, щоб збалансувати загальний момент.

Ротори 1 і 3 виробляють момент проти годинникової стрілки, а ротори 2 і 4 - за годинниковою стрілкою.

Експериментально спостерігається, що ці моменти піддаються діям лінійно діючих сил на низьких швидкостях.

Є чотири вхідних сили та шість виходів ($x, y, z, \theta, \phi, \psi$).

Рівняння (2.4) забезпечує математичне вираження вхідних сил квадроротора:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -b & 0 & b \\ -b & 0 & b & 0 \\ -d & d & -d & d \end{bmatrix} \cdot \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \quad (2.4)$$

У рівнянні (2.4) U_1 являє собою силу тяги, що діє на корпус квадротора, U_2 являє собою силу, що спричиняє рух кочення, U_3 являє собою силу, що спричиняє крутний момент та U_4 представляє силу, що спричиняє момент кручення. позначає кутову швидкість роторів ($i = 1, 2, 3, 4$), b вказує коефіцієнт тяги, а d вказує коефіцієнт дрейфу.

Повна математична модель системи Ейлера – Лагранжа з припущеннями, що структура квадротора повністю жорстка і симетрична, а центр мас та центр тіла знаходяться на одній осі, подана у рівнянні (2.5):

$$\left\{ \begin{array}{l} \ddot{x} = (c\psi s\theta s\phi + s\psi c\phi) \cdot \frac{U_1}{m} \\ \ddot{y} = (s\psi s\theta c\phi + s\psi c\phi) \cdot \frac{U_1}{m} \\ \ddot{z} = -g + (c\theta c\phi) \cdot \frac{U_1}{m} \\ \ddot{\phi} = \psi \dot{\theta} \left(\frac{I_y - I_z}{I_x} \right) - \frac{J}{I_x} \dot{\theta} (-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4) + \frac{l}{I_x} U_2 \\ \ddot{\theta} = \psi \dot{\phi} \left(\frac{I_x - I_z}{I_y} \right) + \frac{J}{I_y} \dot{\phi} (-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4) + \frac{l}{I_x} U_3 \\ \ddot{\psi} = \dot{\theta} \dot{\phi} \left(\frac{I_x - I_y}{I_z} \right) + \frac{1}{I_y} U_4 \end{array} \right. \quad (2.5)$$

Рівняння (2.5) обчислює прискорення, отримане по осях x , y , z та кутові прискорення.

Тут J представляє інерційність одного ротора; $I_{x,y,z}$ - інерція тіла квадротора відповідно по осях x , y , z ; l - відстань ротора до центру мас; g - гравітаційне прискорення, m - маса літального апарату.

2.2 Методи управління квадранотом

Літальним апаратом керують різні керуючі структури, які включені до розробленої математичної моделі квадранотора.

Для проектування таких контролерів використовуються U_1, U_2, U_3 і U_4 , які є входами системи, що рухає БПЛА.

Ці входи контролюють вертикальний рух (h) по осі z , рух по осі y (кут нахилу— ϕ), рух по осі x (кут нахилу— θ) та кут повороту (ψ) навколо осі z , відповідно.

Числові значення, що присвоюються цим входам, обчислюються за алгоритмами контролера, розробленого тими самими входами.

Як передбачено у рівняннях, вхідні дані U_i отримуються з кутової швидкості гвинтів.

Ця кутова швидкість досягається обертанням двигуна БПЛА.

Завдяки модельній системі, ϕ кут нахилу, θ кут нахилу, ψ швидкість повороту та h вертикальна швидкість, які представляють входи U_i , використовуються для управління літальним апаратом замість кута повороту двигунів при створенні алгоритмів контролера.

2.3 PID-контроль

PID-контролер - це контролер зворотного зв'язку, який широко використовується в різних типах систем, робототехніці та галузі автоматизації у всьому світі завдяки своїй простій і надійній структурі та характеристикам, що часто дає задовільні результати.

Вихідне рівняння PID-регулятора, який є різновидом лінійного регулятора, подано у рівнянні (2.6):

$$U_{PID} = k_p e(t) + k_i \int e(t) dt + k_d \frac{de(t)}{dt} \quad (2.6)$$

де k_p , k_i та k_d - пропорційні, інтегральні та похідні коефіцієнти підсилення, відповідно;

$e(t)$ - похибка, спричинена різницею між еталоном та відгуком розроблюваної системи.

Пропорційний коефіцієнт посилення використовується для контролю часу наростання реакції системи.

Інтегральний коефіцієнт підсилення використовується для усунення помилок стаціонарного стану.

Коефіцієнт посилення похідної дозволяє зменшити кількість переповнення та розвинути перехідну реакцію.

У таблиці 2.1 показано вплив таких підвищених параметрів на керовану систему.

Таблиця 2.1. Ефекти незалежного налаштування P, I та D.

Відповідь замкнутого циклу	Час підйому	Перевитрата	Час поселення	Помилка стаціонарного стану	Стабільність
Збільшення k_p	Зменшення	Збільшити	Невелике збільшення	Зменшення	Деградувати
Збільшення k_i	Невелике зменшення	Збільшити	Збільшити	Велике зменшення	Деградувати
Збільшення k_d	Невелике зменшення	Зменшення	Зменшення	Незначні зміни	Поліпшити

Успіх PID-регулятора залежить від правильного вибору параметрів посилення. PID-контролер у літературі можна розділити головним чином на дві категорії.

У першій категорії параметри контролера не змінюються під час управління після того, як їх було налаштовано або вибрано певними оптимальними способами.

На даний момент найбільш відомим методом є налаштування коефіцієнтів PID за формулою Циглера – Ніколса.

Класичні PID-контролери цієї категорії мають просту структуру, але не завжди можуть ефективно керувати системами відповідно до мінливих умов і можуть потребувати частого оновлення в режимі онлайн.

Крім того, оскільки класичний PID-регулятор має фіксований коефіцієнт підсилення, його продуктивність обмежена в межах, що перевищують робочі точки.

Контролери другої категорії мають структуру класичного PID-регулятора, але їх параметри можуть бути адаптовані в режимі онлайн.

Тут потрібні певні знання, такі як структура моделі рослин. Контролери в цій категорії можна визначити як динамічні або адаптивні.

2.4 Самоналаштовуваний нечіткий PID-контролер

Оскільки ПД-регулятор (PID-контролер) базується на лінійній моделі, нелінійність системи приносить невизначеність та погіршує продуктивність.

Це виключає здатність до навчання та адаптованість, необхідні для подолання нелінійності та невизначеності.

Таким чином, перетворення пропорційних, інтегральних та похідних коефіцієнтів підсилення ПД-регуляторів у структуру самоналаштування, з метою адаптації в Інтернеті, збільшує рівень успішності управління нелінійними платформами.

Це момент, коли можна включити нечітку логіку, метод управління, що базується на знаннях, в алгоритмі управління, щоб визначити параметри PID відповідно до мінливих умов.

Вхідний сигнал проходить етапи нечіткості, умовиводу та дефазифікації, щоб знайти нечіткий вихід управління, відповідний входу в контролері нечіткої логіки.

Етапи нечіткості, умовиводу та дефазифікації використовуються для чіткого збігу між базою правил та базою знань.

Нечіткі значення можуть коливатися від 0 до 1.

Це є суттєвою перевагою нечіткої логіки.

Крім того, нечітка логіка не вимагає математичної моделі системи, до якої вона буде застосована.

Для проектування нечіткого контролера має бути достатньо лише експертних знань.

Отже, нечітка логіка - це метод управління, який легше зрозуміти порівняно із звичайними методами управління.

Використання всесвіту нечітких міркувань та дискурсу для регулювання посилення PID на основі класичного PID-регулятора дозволяє включити особливості надійності та пристосованості нечіткої системи до створеного алгоритму управління.

На рисунку 2.3 показана схема нечіткого PID-контролю, що застосовується у дослідженні.

Термін самоналаштування вказує на характеристики контролера автоматичної настройки його параметрів управління в режимі онлайн, щоб мати найбільш підходящі значення тих отриманих коефіцієнтів підсилення, що призводить до оптимізації виходу процесу.

Розробка правил управління для самоналаштування нечіткого PID-регулятора базується на теоретичному та експериментальному аналізі системи.

Коефіцієнти підсилення k_p , k_i та k_d можна налаштувати, регулюючи інші параметри управління та коефіцієнти в режимі онлайн, що підвищує точність управління, що забезпечує кращу продуктивність, ніж класичний PID-контролер.

Основними перевагами цього контролера є уникнення переповнення та скорочення часу осідання.

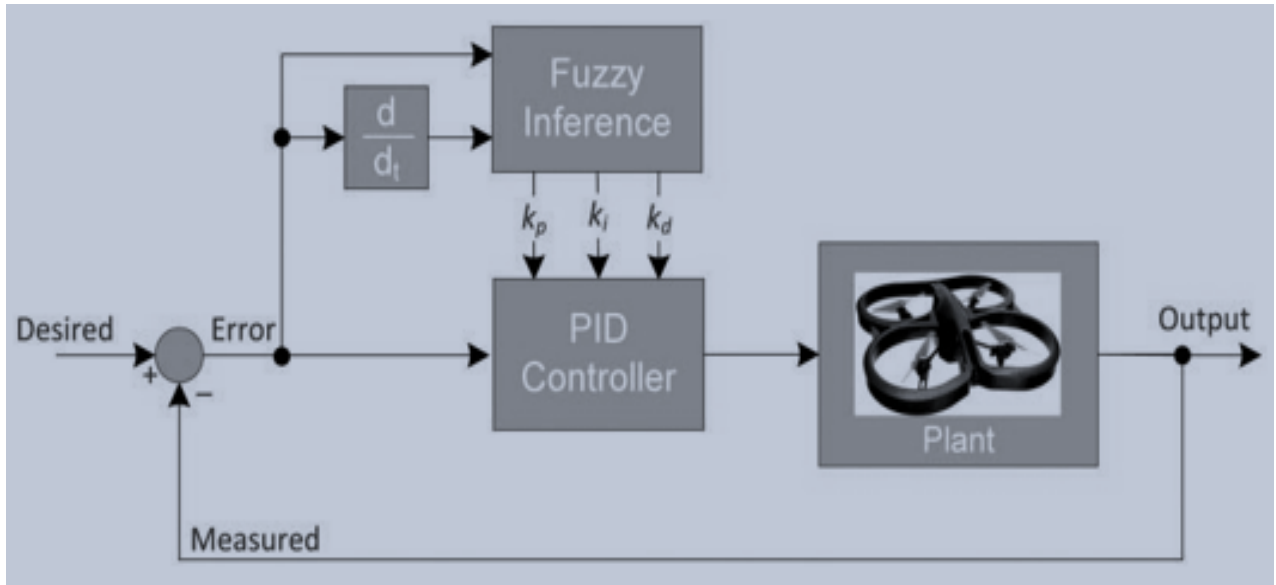


Рисунок 2.3 - Основна структура самоналаштовуваного нечіткого алгоритму управління PID. PID: пропорційна інтегральна похідна

Самоналаштовуваний нечіткий PID-контролер описує, як знайти нечітке співвідношення між коефіцієнтами посилення k_p , k_i і k_d та $e(t)$ та $de(t)$, та модифікувати ці три коефіцієнти посилення, щоб відповідати різним вимогам згідно з принципами нечіткого управління.

Вхід саморегульованого нечіткого PID-регулятора - помилка $e(t)$, зміна помилки $de(t)$, а на виході - сигнал управління, що передається в систему.

Підсилення PID регулюється контролером нечіткої логіки, де такі параметри вважаються змінними в діапазоні $[k_p \text{ min}, k_p \text{ max}]$, $[k_i \text{ min}, k_i \text{ max}]$ та $[k_d \text{ min}, k_d \text{ max}]$ відповідно під час регулювання вирає k_p , k_i та k_d .

Кожен діапазон параметрів визначається тестами, проведеними на PID-контролері для досягнення застосовних баз правил з високою ефективністю висновку.

Параметри k_p , k_i та k_d для зручності нормуються між нулем та одиницею за допомогою формул, наведених у рівнянні (2.7).

$$\begin{cases} k'_p = (k_p - k_{p \min}) / (k_{p \max} - k_{p \min}) \\ k'_i = (k_i - k_{i \min}) / (k_{i \max} - k_{i \min}) \\ k'_d = (k_d - k_{d \min}) / (k_{d \max} - k_{d \min}) \end{cases} \quad (2.7)$$

Для того, щоб регулювати коефіцієнти підсилення, нечіткі правила створюються таким чином:

(якщо) помилка $e(t)$ - це A_i , а зміна помилки $de(t)$ - B_i ,

(тоді) k'_p посилення C_i , k'_i посилення D_i та k'_d коефіцієнт підсилення - E_i ($i = 1, 2, \dots, m$, m - загальна кількість правил).

A_i, B_i, C_i, D_i і E_i представляють нечіткі множини.

Ці нечіткі набори характеризуються лінгвістичними термінами через створені функції належності.

Після нечіткості з використанням функцій членства та основ правил, дефузіфікація проводиться відповідно до обраного методу.

Нарешті, отримані параметри k'_p , k'_i і k'_d вставляються в рівняння (8), де це доцільно, щоб знайти коефіцієнти посилення контролера.

$$\begin{cases} k_p = (k_{p \max} - k_{p \min})k'_p + k_{p \min} \\ k_i = (k_{i \max} - k_{i \min})k'_i + k_{i \min} \\ k_d = (k_{d \max} - k_{d \min})k'_d + k_{d \min} \end{cases} \quad (2.8)$$

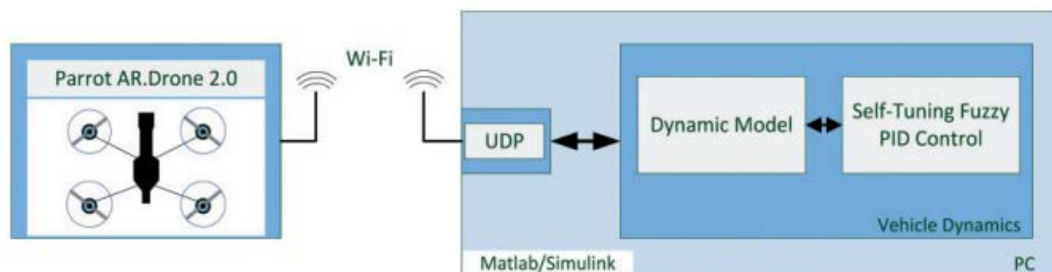


Рисунок 2.4 - Блокове представлення розробленої системи. UDP: Протокол датаграм користувачів

2.5 Динаміка та самоналаштування нечіткого PID-контролю квадродротора AR.Drone

Динамічну модель Parrot AR.Drone 2.0 можна отримати, використовуючи експериментальні дані вводу-виводу, структуру рівнянь для руху літаків та методи ідентифікації системи.

Динаміка AR.Drone описується за допомогою передавальних функцій, представлених у рівнянні (9). Передавальні функції G_ϕ , G_θ , G_ψ та G_h представляють лінійну динаміку, отриману з використанням експериментальних вхідних та вихідних даних від рухів літаків.

$$\begin{cases} G_\phi = \frac{\phi}{\phi_r} \\ G_\theta = \frac{\theta}{\theta_r} \\ G_\psi = \frac{\psi}{\psi_r} \\ G_h = \frac{h}{h_r} \end{cases} \quad (2.9)$$

У рівнянні (9), ϕ , θ і ψ представляють рухи нахилу, нахилу та похилу БРЛА відповідно, де h являє собою висоту БПЛА.

Опорними командами ϕ_r , θ_r , ψ_r та h_r є кут нахилу, кут нахилу, швидкість похилу та вертикальна швидкість відповідно.

Бажаний рух досягається введенням опорних значень у внутрішній контролер автомобіля як вхідних даних, що є значеннями з плаваючою точкою, масштабованими в діапазоні $[-1, 1]$.

Вихідні значення ϕ_m , θ_m , ψ_m , h_m , u_m та v_m виміряні інерційними датчиками та алгоритмами зору БПЛА, вказують кути нахилу, кроку та похилу в радіані, висоту в метрах і швидкості в м/с на поздовжній та поперечній осях відповідно.

AR.Drone використовує інерційну інформацію з свого ІМУ для оцінки стану БПЛА. Він поєднує дані ІМУ з інформацією з алгоритмів зору та аеродинамічної моделі для оцінки швидкості БПЛА.

Модуль наведення вбудований у внутрішній контролер для того, щоб ПС слідував за пунктами маршруту.

Математична модель динаміки літальних апаратів використовується для проектування внутрішніх контролерів C_ϕ , C_θ , C_ψ та C_h з модулем задання траєкторії.

Контролери G_u , G_v , G_ψ та G_h , показані на рисунку 9 (а) та (б), використовуються для відстеження поздовжньої швидкості u , поперечної швидкості v , кута похилу та висоти h відповідно.

Модуль наведення БПЛА, показаний на рисунку 9 (в), спрямовує літак з поточного інерційного положення $[X, Y]^T$ у бажане положення $[X_{ref}, Y_{ref}]^T$ за допомогою команд опорної швидкості $[u_{ref}, v_{ref}]^T$.

На рисунку 10 показано схему розташування, орієнтації та векторів швидкості для літального апарату в площинах $x - y$ місцевої системи координат.

Відстань P між БПЛА та ціллю визначається рівнянням (10) з використанням осей тіла i, j та k .

$$P_\tau = \begin{bmatrix} e_i \\ e_j \end{bmatrix} = T \cdot \begin{bmatrix} X_{ref} - X \\ Y_{ref} - Y \end{bmatrix} \quad (2.10)$$

де T - перетворення координат, яке переносить вектор з інерційного кадру в кадр рухомого тіла, а його формула наведена у рівнянні (2.11):

$$T = \begin{bmatrix} \cos(\psi) & \sin(\psi) \\ -\sin(\psi) & \cos(\psi) \end{bmatrix} \quad (2.11)$$

Результати вираження P_τ множаться на коефіцієнти PID k , визначені або класичним PID-регулятором, або контролером нечіткої логіки, щоб отримати поздовжні та поперечні швидкості, наведені у рівнянні (2.12):

$$[u_{ref}, v_{ref}]^T = k \cdot P_\tau \quad (2.12)$$

Для того, щоб забезпечити квадротору AR.Drone слідування бажаної траєкторії та досягнення контрольних точок з мінімальними коливаннями та похибками, в дослідженні пропонується самоналаштування нечіткого алгоритму управління ПД, що є вмістом контролерів, представлених C_ϕ , C_θ , C_ψ та C_h блоки.

Основна структура контролера нечіткої логіки, який використовується для налаштування коефіцієнтів підсилення ПД подана на рис. 2.11.

Вхідними даними нечіткого логічного контролера, показаними на рисунку 2.11, є помилка $e(t)$ та зміна помилки $de(t)$, а виходами є коефіцієнти посилення k_p , k_i , та k_d . Для мовного вираження входів $e(t)$ і $de(t)$ контролера вибрано сім нечітких значень: NB, NM, NS, Z, PS, PM і PB.

Негативні велике представлене як NB, негативне середовище як NM, негативне маленьке як NS, нуль як Z, позитивне маленьке як PS, позитивне середовище як PM та позитивне велике як PB.

Для мовного вираження виходів k_p , k_i , та k_d , контролера вибрано сім нечітких значень: VVS, VS, S, M, B, VB і VVB.

Дуже дуже малий представлений як VVS, дуже маленький як VS, малий як S, середній як M, великий як B, дуже великий як VB і дуже дуже великий як VVB.

Вхідні та вихідні функції належності, що використовуються для регулювання коефіцієнтів підсилення PID-контролерів, мають однакову форму, тоді як діапазони значень нечітких наборів різні.

Ці діапазони визначаються з використанням підходу методом проб і помилок. На рисунку 2.12 показані визначені функції належності.

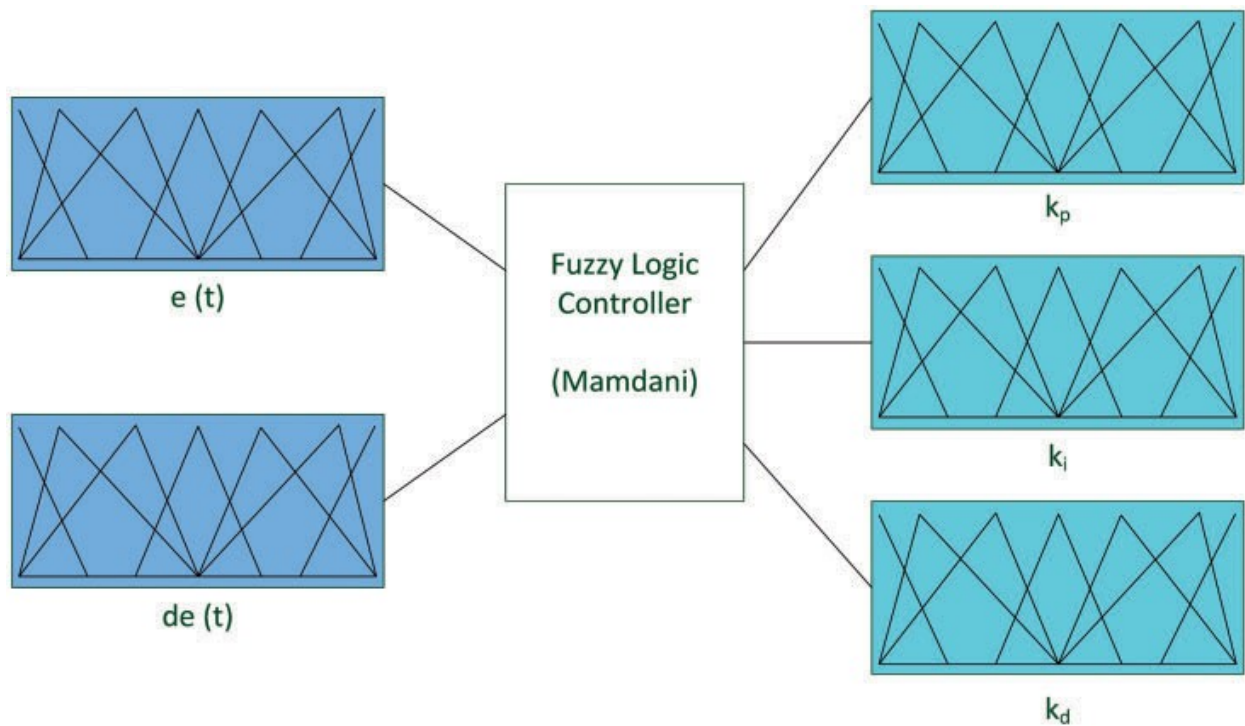


Рисунок 2.11 – Схема системи нечіткого логічного висновку

Функції належності, наведені на рисунках 2.12 (а) та (b), складаються з трикутних функцій належності.

Ширину нечітких множин вибирають із входів:

[1, 1] для похибки $e(t)$;

[10, 10] для зміни похибки $de(t)$.

Якщо значення вхідних даних системи виходить за межі цих діапазонів значень, для збереження значень у цьому діапазоні використовується точка насичення.

Діапазон значень для виходів вибирається як [0,2, 0,7] для пропорційного коефіцієнта підсилення k_p , як [0,001, 0,01] для інтегрального коефіцієнта підсилення k_i і як [0,1, 0,15] для похідного коефіцієнта підсилення k_d .

Набір лінгвістичних правил є основною частиною нечіткого контролера. У більшості випадків перетворити думки експертів у ці правила легко, і можна створити будь-яку кількість правил для визначення рухів контролера.

У дослідженні виходи k_p , k_i та k_d мають по одній базі правил. Використовуючи засіб для встановлення правил, наданих редактором системи

нечітких умовиводів у наборі інструментів нечіткої логіки MATLAB, створено 49 правил для трьох контролерів нечіткої логіки з трьох баз правил.

Ці правила базуються на висновках експертів та досвіді.

Правила, наведені в таблицях 2.2-2.4, можна інтерпретувати так: наприклад, відповідно до таблиць правил, якщо помилка $e(t)$ дорівнює NL, а зміна похибки $de(t)$ дорівнює PS, коефіцієнт посилення k_p буде VL, коефіцієнт посилення k_i буде VL, а коефіцієнт посилення k_d буде VS.

Вихід нечіткої логічної системи нечіткий і не може бути безпосередньо застосований до динамічної системи.

Необхідний процес дефузіфікації, щоб перетворити ці нечіткі виходи в числа, що представляють ті самі нечіткі виходи.

Сигнал управління повинен бути безперервним, і будь-яка зміна на вході не повинна призводити до значних змін на виході.

Алгоритм дефузіфікації повинен бути чітким, а процес визначення вихідного сигналу повинен бути чітко визначений.

Тому в цьому дослідженні обрано метод центру ваг.

Таблиця 2.2 - Нечіткі правила для k_p .

Зміна помилки (de)	Помилка(e)						
	NB	NM	NS	Z	PS	PM	PB
NB	M	S	VS	VVS	VS	S	M
NM	B	M	S	VS	S	M	B
NS	VB	B	M	S	M	B	VB
Z	VVB	VB	B	M	B	VB	VVB
PS	VB	B	M	S	M	B	VB
PM	B	M	S	VS	S	M	B
PB	M	S	VS	VVS	VS	S	M

NB: негативний великий; NM: негативне середовище; NS: негативний малий;

Z: нуль; PS: позитивний малий; PM: позитивне середовище; PB: позитивний

великий; VVS: дуже дуже малий; VS: дуже маленький; S: маленький; M: середній; B: великий; VB: дуже великий; VVB: дуже дуже великий.

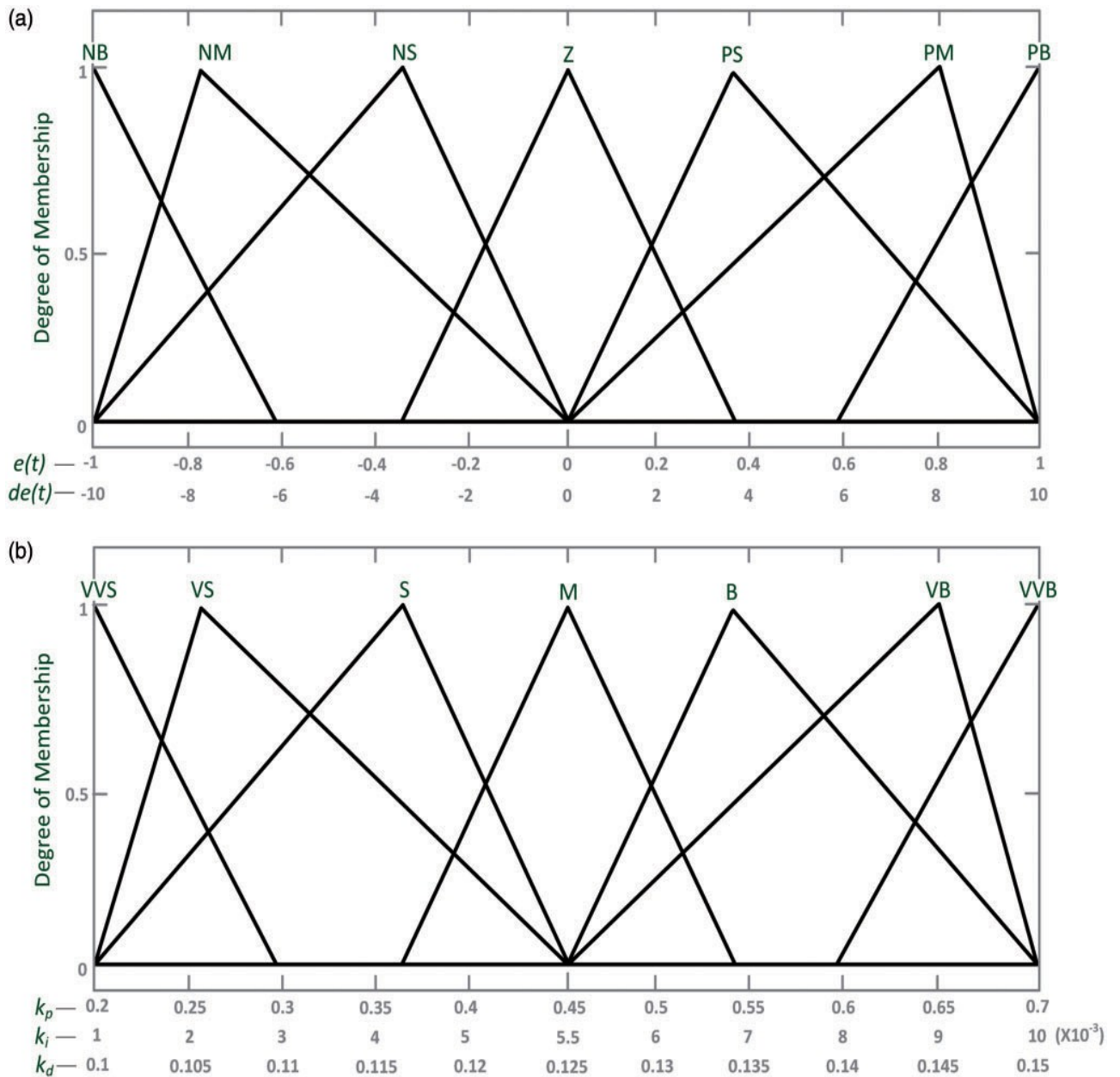


Рисунок 2.12. Функції належності для всіх входів та виходів:

(a) $e(t)$ та $de(t)$ входи;

(b) k_p , k_i , та k_d виходи

Таблиця 2.3 - Нечіткі правила для k_i .

Зміна помилки (de)	Помилка(e)						
	NB	NM	NS	Z	PS	PM	PB
NB	M	S	VS	VVS	VS	S	M
NM	B	M	S	VS	S	M	B
NS	VB	B	M	S	M	B	VB
Z	VVB	VB	B	M	B	VB	VVB
PS	VB	B	M	S	M	B	VB
PM	B	M	S	VS	S	M	B
PB	M	S	VS	VVS	VS	S	M

NB: негативний великий; NM: негативне середовище; NS: негативний малий; Z: нуль; PS: позитивний малий; PM: позитивне середовище; PB: позитивний великий; VVS: дуже дуже малий; VS: дуже маленький; S: маленький; M: середній; B: великий; VB: дуже великий; VVB: дуже дуже великий.

Таблиця 2.4 - Нечіткі правила для k_d .

Зміна помилки (de)	Помилка(e)						
	NB	NM	NS	Z	PS	PM	PB
NB	M	B	VB	VVB	VB	B	M
NM	S	M	B	VB	B	M	S
NS	VS	S	M	B	M	S	VS
Z	VVS	VS	S	M	S	VS	VVS
PS	VS	S	M	B	M	S	VS
PM	S	M	B	VB	B	M	S
PB	M	B	VB	VVB	VB	B	M

NB: негативний великий; NM: негативне середовище; NS: негативний малий; Z: нуль; PS: позитивний малий; PM: позитивне середовище; PB: позитивний великий; VVS: дуже дуже малий; VS: дуже маленький; S: маленький; M: середній; B: великий; VB: дуже великий; VVB: дуже дуже великий

2.6 Висновки

В розділі представлено математичну модель системи керування безпілотними літальними апаратами, яка описує рух БПЛА у всіх напрямках польоту.

Також в розділі описано методи управління квадротором, наведено опис функціонування PID-контролера, а також описано модель самоналаштовуваного нечіткого PID-контролера.

В розділі представлено модель руху із залученням нечіткого PID-контролера на основі квадротора AR.Drone.

3 ІНТЕЛЕКТУАЛІЗОВАНА СИСТЕМА КЕРУВАННЯ БЕЗПЛОТНИМИ ЛІТАЛЬНИМИ АПАРАТАМИ

3.1 Побудова інтелектуалізованої системи керування безпілотними літальними апаратами

Для побудови інтелектуалізованої системи керування безпілотними літальними апаратами розглянемо їх структуру. Зокрема, за основу системи використаємо квадрокоптер - безпілотний літальний апарат, що включає в себе чотири гвинти, розташовані симетрично навколо центрального корпусу [47].

Особливістю дрону є те, що його суміжні гвинти повинні обертатися протилежного один від одного.

Ця вимога пояснюється необхідністю запобігання обертання системи навколо власної центральної осі.

В залежності від необхідної траєкторії руху, важливим аспектом є здатність задання різних значень потужності двигунів дрону.

Попри те, що БПЛА характеризуються простотою своєї структури, вони в процесі експлуатації здатні імплементувати велику множину моделей руху разом з демонстрацією високої маневреності.

Це досягається наявністю шести ступенів вільності, які складаються з трьох поступальних та трьох обертових компонентів задання траєкторії руху БПЛА.

Узагальнене схематичне представлення інтелектуалізованої системи керування безпілотними літальними апаратами подано на рисунку 3.1.

Схематичне представлення БПЛА подано на рисунку 3.2.



Рисунок 3.1 - Схематичне представлення інтелектуалізованої системи керування безпілотними літальними апаратами

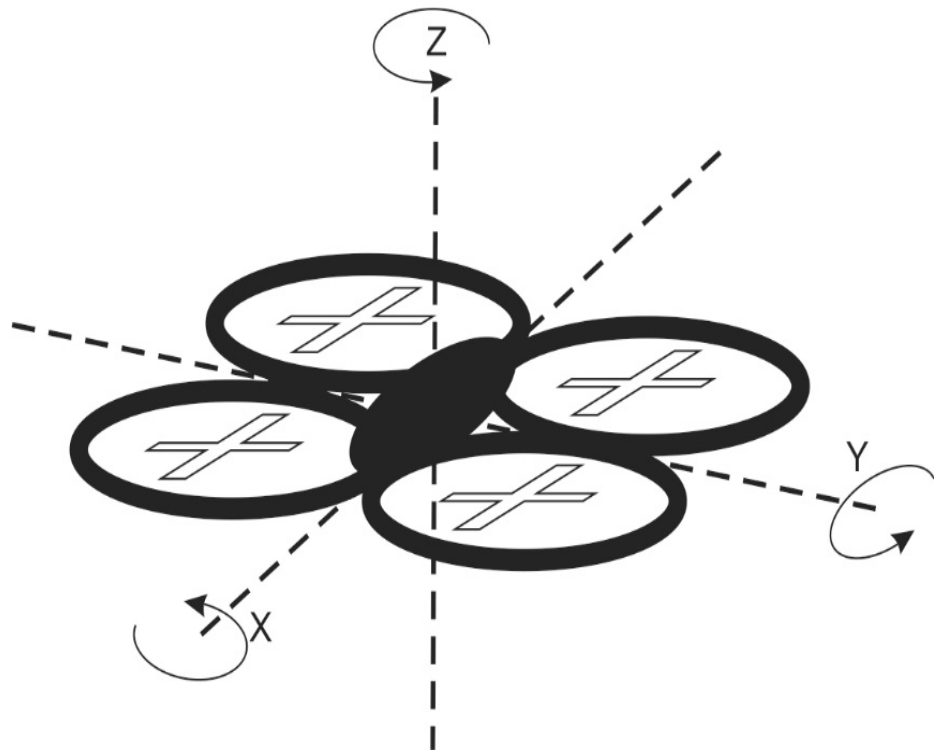


Рисунок 3.2 - Схема БПЛА

Представимо систему керування безпілотними літальними апаратами моделлю у вигляді кортежу параметрів:

$$F_{UAV} = \langle z, v_x, v_y, \theta, \varphi, \psi \rangle, \quad (3.1)$$

де z - висота від землі до БПЛА в повітрі (м);

v_x - лінійна швидкість БПЛА в напрямку x , (м/с);

v_y - лінійна швидкість безпілота у напрямку y , (м/с);

θ - кут нахилу БПЛА по осі y , (рад);

φ - кут нахилу (крену) БПЛА по осі x , (рад);

ψ - кут повороту БПЛА навколо вертикальної осі z , (рад).

Розглянемо множину команд, які сприймає внутрішній контролер польотів безпілотного літального апарату.

З цією метою подамо набір команди управління траєкторією руху БПЛА кортежем:

$$C_{UAV} = \langle \xi_z, \xi_\psi, \xi_\theta, \xi_\varphi \rangle, \quad (3.2)$$

де ξ_z - команда задання лінійної швидкості безпілотного літального апарату в напрямку по осі z ;

ξ_ψ - команда задання кута повороту руху БПЛА навколо вертикальної осі z ;

ξ_θ - команда задання висоти польоту БПЛА по осі y ;

ξ_φ - команда задання кута нахилу (крену) БПЛА по осі x .

Очевидно, що задання команд руху ξ_θ, ξ_φ реалізують поступальний рухи вздовж x - і y -напрямоків відповідно.

Тому для регулювання швидкостей дрона в необхідних напрямках застосуємо команд ξ_θ, ξ_φ , а для забезпечення орієнтації БПЛА використовуватиметься команда ξ_ψ .

Вхідні та вихідні параметри системи керування БПЛА представлено на рисунку 3.3.

3.2 Система керування БПЛА

Керування БПЛА є складною задачею, оскільки продуктивність контролера залежить від точності математичної моделі проєктованої системи БПЛА.

Відомі математичні моделі часто нехтують зовнішніми впливами, які в реальних умовах зазнає дрон в польоті:

1. сила тяжіння;
2. тертя повітря;
3. сила вітру;
4. інші сили.

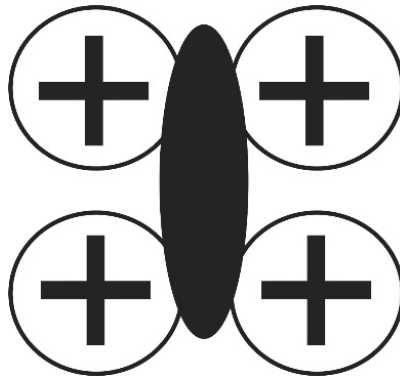
Команди задання руху
траєкторії БПЛА

$\xi_z \Rightarrow$

$\xi_\psi \Rightarrow$

$\xi_\theta \Rightarrow$

$\xi_\varphi \Rightarrow$



Множина параметрів
БПЛА

$\Rightarrow z$

$\Rightarrow u_x$

$\Rightarrow u_y$

$\Rightarrow \theta$

$\Rightarrow \varphi$

$\Rightarrow \psi$

Рисунок 3.3 - Вхідні та вихідні параметри системи керування траєкторією руху БПЛА

Іншим аспектом, що суттєво впливає на зниження точності виконання заданої траєкторії, є шум при одержанні інформації, зібраної з усіх датчиків БПЛА.

Це призводить до суттєвої невизначеності при практичному застосуванні дрону і необхідності адаптивної корекції траєкторії руху безпілотної літальної апаратури.

З метою вирішення даної проблеми можливим її вирішення є застосування апаратури нечіткої логіки як основи математичної моделі системи безпілотної літальної апаратури.

Це дозволить забезпечити нечітке логічне керування БПЛА, і, в свою чергу інтелектуалізувати поведінку дрона в повітрі в умовах зовнішніх впливів на зміну заданої траєкторії його руху.

В основі інтелектуалізованої системи керування безпілотними літальними апаратами лежить побудова двох нечітких контролерів для продукції значень сигналів керування для:

ξ_θ - команда задання висоти польоту БПЛА по осі y ;

ξ_φ - команда задання кута нахилу (крену) БПЛА по осі x .

Функції належності для команд управління БПЛА ξ_θ та ξ_φ подано на рисунку 3.4

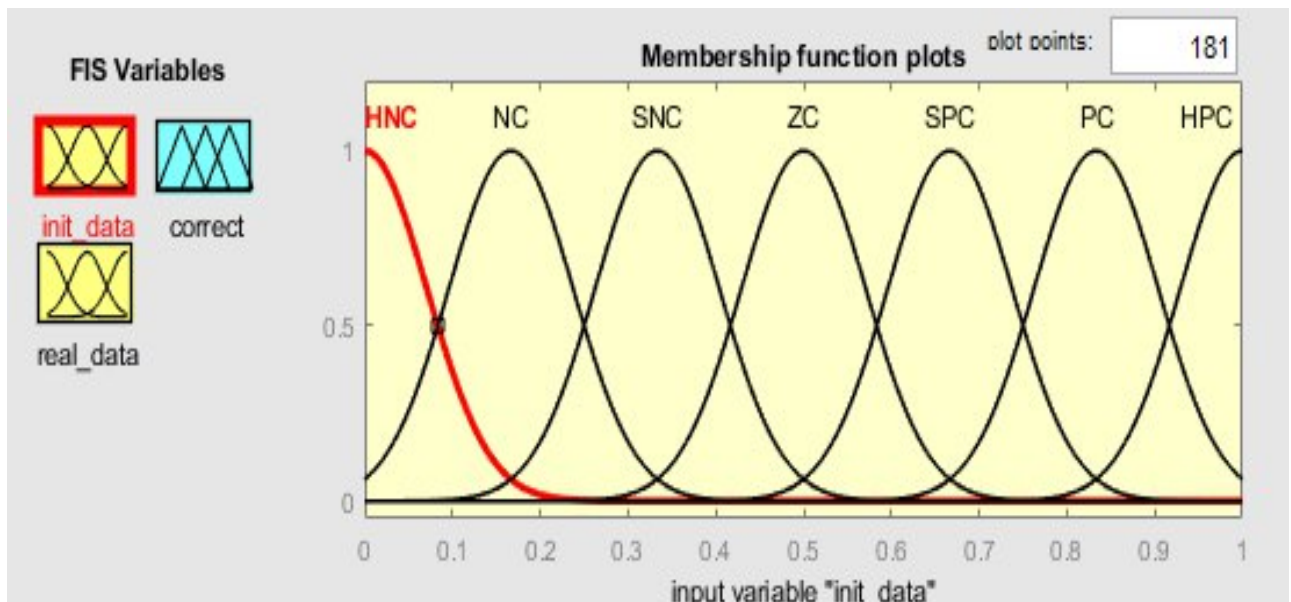


Рисунок 3.4 - Функції належності для команд управління ξ_θ та ξ_φ

У якості вхідної та вихідної лінгвістичних змінних для нечіткого керування БПЛА X приймемо ім'я:

«Ступінь корегування траєкторії дрону».

Значення змінної буде визначатися на етапі його експлуатації в умовах зовнішніх впливів.

Для прийнятих лінгвістичних змінних визначимо терм-множини і функції належності.

Терм-множини значень керуючого сигналу БПЛА приймемо наступним чином:

1. «High negative correction (HNC)»;
2. «Negative correction (NC)»;
3. «Slight negative correction (SNC)»;
4. «Zero correction (ZC)»;
5. «Slight positive correction (SPC)»;
6. «Positive correction (PC)»;
7. «High positive correction (HPC)», з областю визначення $[0,1]$.

Для генерації команд управління ξ_θ та ξ_φ відповідний нечіткий контролер має два входи, які, в свою чергу, оперують значеннями для формування траєкторії польоту в x - або y -напрямку:

- 1) початкові значення (σ_x, σ_y) ;
- 2) та скореговані під впливом зовнішніх чинників (σ'_x, σ'_y) .

Інтелектуалізована система керування безпілотними літальними апаратами керується множиною правила нечітких продукцій (ПНП).

Кожне із правил має деяку вагу $F=[0,1]$, що визначає значимість правила або впевненість у ступені істинності висновку, який отримується за окремим нечітким правилом.

В дослідженні було залучено двадцять п'ять правил нечітких продукцій, приклади яких подано нижче:

П.1. If (σ_x, σ_y) is HNC and (σ'_x, σ'_y) is HNC then HPC (1);

П.2. If (σ_x, σ_y) is HNC and $((\sigma'_x, \sigma'_y)$ is NC then PC (1);

.....

П.25. If (σ_x, σ_y) is HPC and $((\sigma'_x, \sigma'_y)$ is HPC then HNC (1).

Дефазифікація, тобто процедура одержання чисельного значення для прийняття остаточного рішення щодо корегування траєкторії руху дрона було виконано за допомогою застосування методу центра тяжіння.

3.3 Експерименти

Розроблена система управління в реальному часі виконуються на квадаторі AR.Drone 2.0 за допомогою динамічної моделі та контролерів, створених у середовищі MATLAB / Simulink.

Основна мета цього дослідження полягає в тому, щоб дати можливість БПЛАу слідувати визначеній траєкторії з алгоритмом вдосконалений контролер.

Коли БПЛА злітає, початкове положення та орієнтація БПЛА встановлюються на нуль.

Потім повітряне БПЛА повідомляється про контрольні пункти. Інформація про шляхові точки, необхідна для відстеження траєкторії, включає п'ять значень даних.

Вказаними даними є:

1. X_{ref} ,
2. Y_{ref} ,
3. h_{ref} ,

кут похилу та час очікування в цій точці.

Відносне положення між транспортним засобом і ціллю обчислюється в MATLAB / Simulink.

А контролери забезпечують відстеження бажаних параметрів.

Положення та швидкість БПЛАа вимірюються його інерційними датчиками та системою зору.

Ефективність класичного PID-регулятора для дотримання визначеної траєкторії порівнюється з продуктивністю самоналаштування нечіткого PID-регулятора.

На додаток до цих двох алгоритмів управління, контролер нечіткої логіки також тестувався окремо для управління БПЛА, щоб слідувати визначеній траєкторії.

Однак результати роботи менш успішні, ніж перші два методи контролю БПЛА.

Перший етап експериментів передбачав випробування слідування траєкторії БПЛА, керованого класичним PID-контролером.

Переваги класичного PID-регулятора визначаються за допомогою методу Циглера – Ніколса відповідно до аналізів реакції системи, проведених на моделі БПЛА в середовищі MATLAB / Simulink.

Отже, коефіцієнти підсилення PID вибираються як

1. $k_p=0,25$;
2. $k_i=0,014$;
3. $k_d=0,12$.

На наступному етапі саморегулюючий нечіткий PID-контролер використовується для того, щоб БПЛА виконував визначену траєкторію

Нечітка логічна система використовується для налаштування підсилення PID.

На рисунках 3.5-3.10 показано графічне представлення даних, отриманих під час відстеження траєкторії в режимі реального часу класичним PID-управлінням та самоналаштуванням нечітких методів PID-контролю.

Ці траєкторії утворені послідовними точками, що мають певну відстань між собою і створюють різні візерунки.

У таблицях 3.1-3.3 відображаються помилки, допущені під час відстеження траєкторії з використанням вищезазначених методів.

Для обох алгоритмів управління експерименти проводяться в приміщенні на постійній висоті 1 м та за відсутності ефекту вітру.

Для рівносторонньої траєкторії продуктивність класичного PID та продуктивність самоналаштовуваних нечітких PID-контролерів порівнюються у

випадку наступних опорних точок на однаковій відстані один від одного, виконуючи перехресні рухи.

БПЛА, який утримується загалом 24 с у визначених опорних точках, здійснив політ за 40 с за допомогою класичного PID контролера і за 39 с за допомогою самоналаштовуваного нечіткого ПІД-регулятора.

У таблиці 3.1 наведено порівняння між методами контролю за величиною похибки для дотримання рівносторонньої траєкторії.

БПЛА з PID-контролером виконав чотири опорні точки траєкторії з максимальною та сумарною похибками 0,32 та 8,02 м відповідно.

За цією ж траєкторією рухається нечіткий PID-регулятор, а максимальна та загальна похибки були визначені як 0,15 та 4,35 м відповідно.

Для прямокутної траєкторії на рисунку 3.5, продуктивність класичного PID та ефективність самоналаштовування нечіткого PID-регулятора порівнюються у випадку наступних визначених опорних точок, виконуючи поздовжні та поперечні рухи.

БПЛА, який утримувався загалом 48 с у визначених опорних точках, завершив польотну доріжку за 68 с класичним PID-контролером та за 66 з самоналаштовуваним нечітким PID-контролером.

У таблиці 3.2 наведено порівняння між методами контролю за величиною похибки для дотримання квадратної траєкторії.

БПЛА завершив вісім опорних точок траєкторії за допомогою PID-регулятора, а максимальна та загальна похибки визначаються відповідно 0,39 та 12,17 м.

За тією самою траєкторією слідує самоналаштовування нечіткого PID-регулятора, а максимальна та загальна похибки визначаються як 0,18 та 6,94 м відповідно.

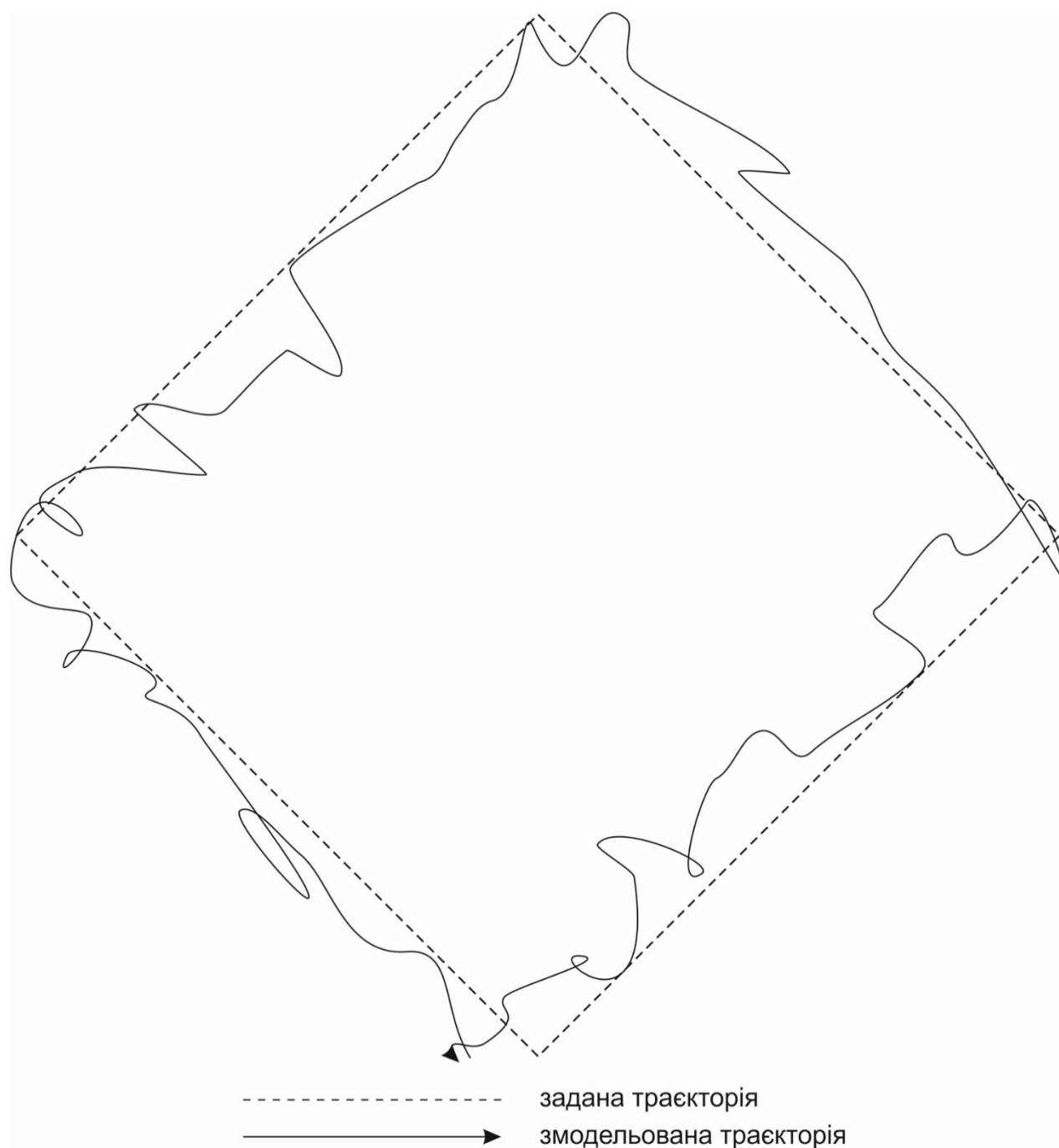


Рисунок 3.5 - Рівностороннє відстеження траєкторії за допомогою PID-контролю та самоналаштування нечіткого PID-контролю.

PID: пропорційна інтегральна похідна

Для кругової траєкторії на рисунку 3.6 продуктивність класичного PID та продуктивність самоналаштування нечіткого PID регулятора порівнюються з точки зору слідувачі визначеним опорним точкам, виконуючи рух на короткій відстані. БПЛА, який утримувався загалом 64 с у визначених опорних точках,

здійснив політ за 102 с за допомогою класичного PID-регулятора та за 95 с за допомогою самоналаштовуваного нечіткого PID-контролера.

У таблиці 3.3 наведено порівняння між методами контролю за величиною похибки для проходження кругової траєкторії.

БПЛА здійснив 16 опорних точок траєкторії за допомогою PID-регулятора, а максимальна та загальна похибки траєкторії визначаються як 0,45 та 13,24 м відповідно.

За тією самою траєкторією слідує самоналаштовування нечіткого PID-регулятора, а максимальна та загальна похибки визначаються як 0,16 та 5,64 м відповідно.

Цифри та таблиці похибок наочно показують, що політні характеристики покращуються, а успішні результати досягаються, коли для регулювання коефіцієнтів підсилення PID використовується нечітка система.

Очікується, що БПЛА виконуватиме траєкторію якомога ближче до лінійної траси, яка представляє найкоротший шлях між опорними точками траєкторії польоту.

Однак інерційні ефекти, що виникають внаслідок будови БПЛА та датчиків, спричиняють ряд відхилень в експерименті в реальному часі.

Крім того, в кінці експериментів вимірюється рівень заряду акумулятора, і спостерігається, що польоти виконуються з використанням приблизно 25% енергії для класичного алгоритму управління PID та 20% енергії для самоналаштовування нечіткого алгоритму управління PID.

На основі MATLAB / Simulink динамічна модель програм керування відстеженням траєкторій квадрантора та реального часу була успішно виконана за допомогою класичного PID-регулятора та самоналаштовуваного нечіткого PID-регулятора і представлена в цій роботі.

Незважаючи на те, що література включає багато досліджень щодо самоналаштовування нечіткого PID-управління динамічними системами, цей метод лише нещодавно застосовується для управління БПЛА.

Таким чином, стає можливим здійснення висновку, що в дослідженнях використовувався цей метод управління, який виконувався в середовищах моделювання та додатках у реальному часі.

У представленому дослідженні новинка використовує цей метод для відстеження траєкторії в реальному часі.

По-перше, класичний PID-регулятор використовується для автономного слідування траєкторії цього БПЛА. Спочатку значення коефіцієнта посилення PID-регулятора визначаються методом Циглера – Ніколса.

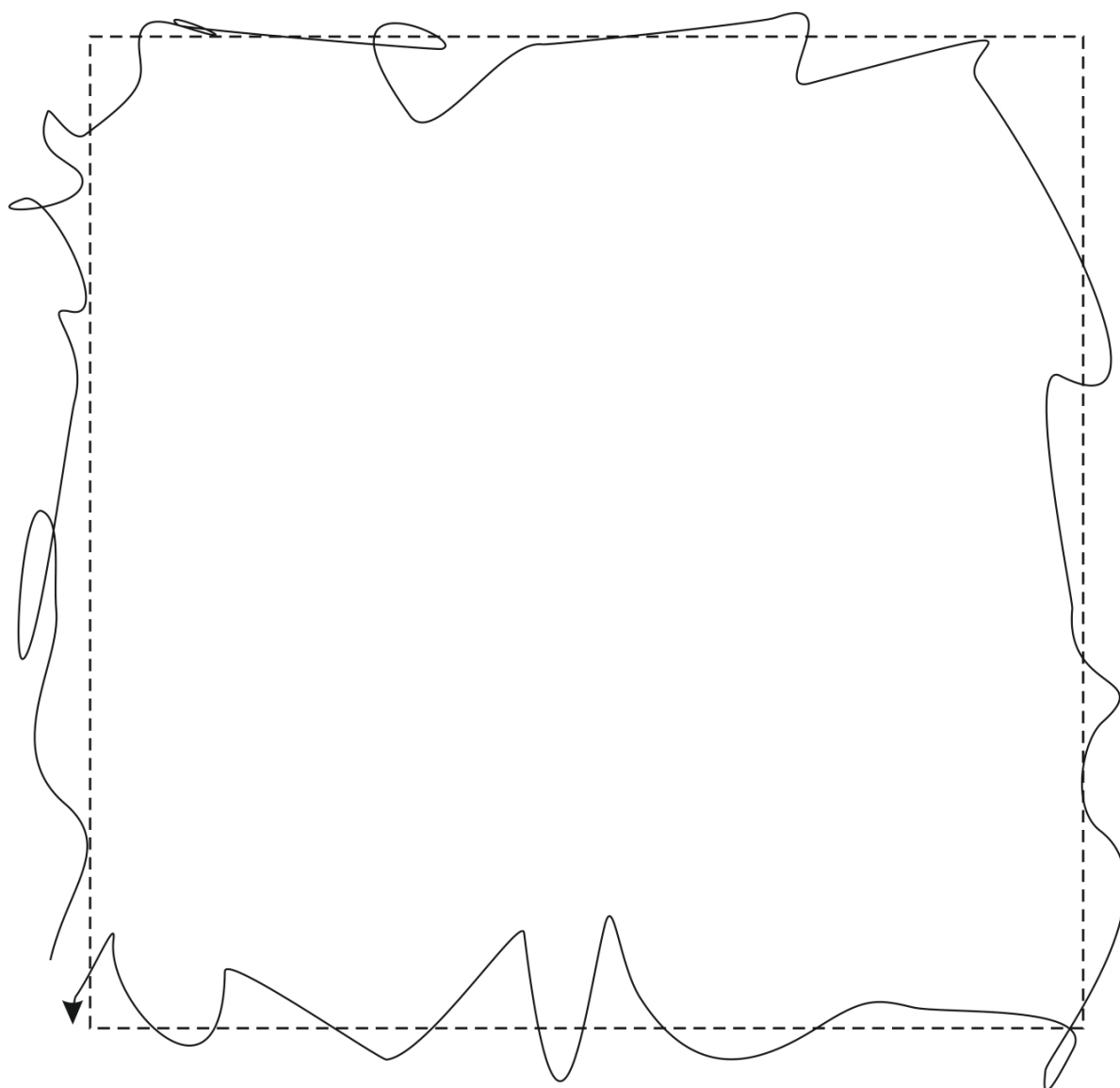
На наступному етапі нечіткий контролер використовується для регулювання коефіцієнтів підсилення PID-регулятора.

За допомогою цих контролерів ефективність БПЛА для відстеження визначеної траєкторії з мінімальними коливаннями спостерігається в режимі реального часу та порівнюється.

Залежність нечіткого контролера від моделі значно менша порівняно із звичайними контролерами.

Нечіткий контролер не є лінійним. Тому він більше підходить для нелінійних систем, таких як квадатор.

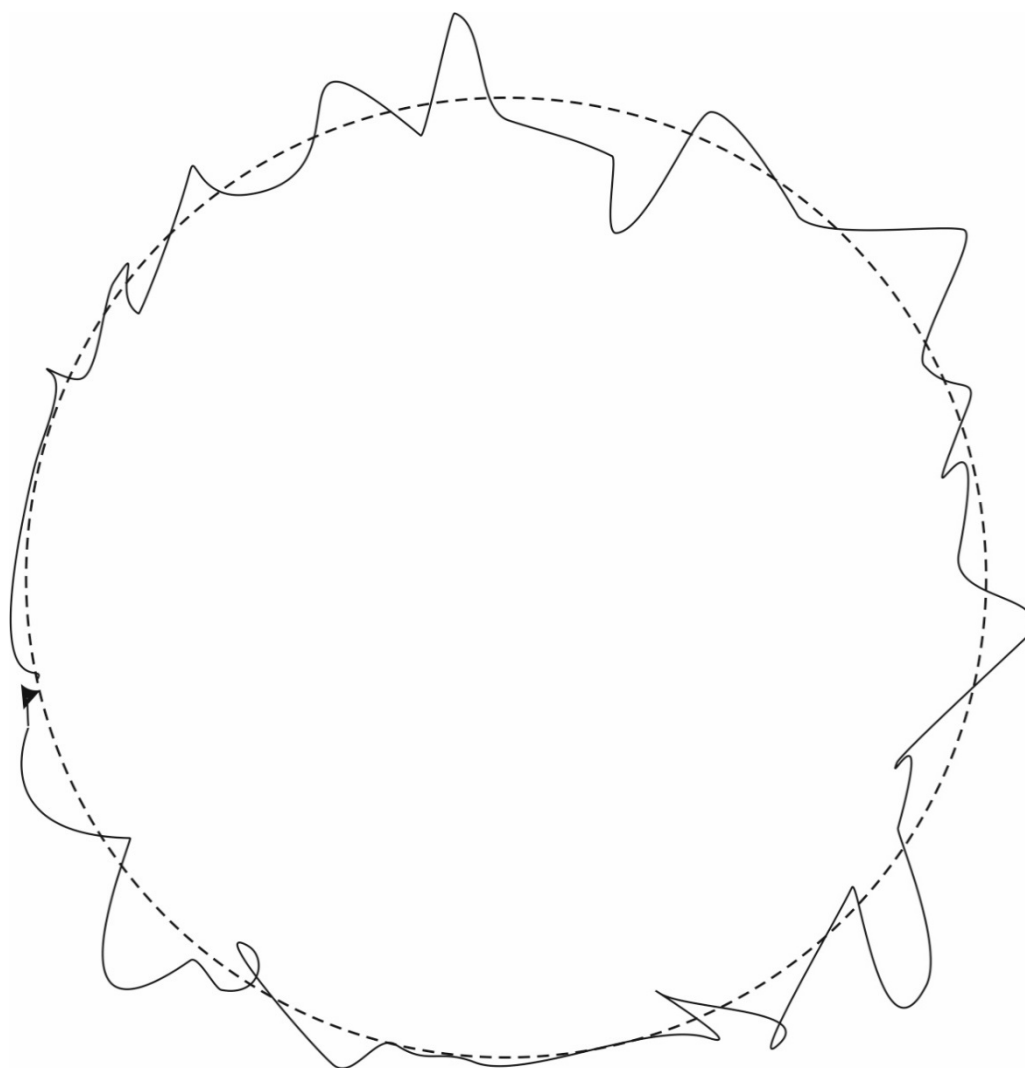
Експериментальні результати вказують на те, що самоналаштовуваний нечіткий PID-контролер має високу продуктивність у порівнянні зі звичайними контролерами за величиною помилок.



----- задана траєкторія
—————> змодельована траєкторія

Рисунок 3.6 - Відстеження квадратних траєкторій за допомогою PID-контролю та самоналаштування нечіткого PID-контролю.

PID: пропорційна інтегральна похідна



----- задана траєкторія
—————> змодельована траєкторія

Рисунок 3.7 - Кругове відстеження траєкторії з PID-керуванням та самоналаштуванням нечіткого PID-контролю.

PID: пропорційна інтегральна похідна

Таблиця 3.1 - Порівняння результатів: рівнобічна траєкторія.

Алгоритм	Максимальна похибка (м)	Загальна похибка (м)
PID	0.32	8.02
Нечіткий-PID	0.15	4.35

PID: пропорційна інтегральна похідна

Таблиця 3.2 - Порівняння показників: квадратна траєкторія.

Алгоритм	Максимальна похибка (м)	Загальна похибка (м)
PID	0.39	12.17
Нечіткий-PID	0.18	6.94

PID: пропорційна інтегральна похідна.

Таблиця 3.3 - Порівняння показників: кругова траєкторія.

Алгоритм	Максимальна похибка (м)	Загальна похибка (м)
PID	0.45	13.24
Нечіткий-PID	0.16	5.64

PID: пропорційна інтегральна похідна.

Для оцінки ефективності застосування апарату нечіткої логіки для побудови інтелектуалізованої системи керування безпілотними літальними апаратами було проведено ряд експериментальних досліджень.

З цією метою було проаналізовано поведінку дрона без та із зовнішнім впливом (вітром) як без залучення нечіткого керування, так і з ним.

Як приклад траєкторії руху було обрано прямокутник. Для здійснення експериментальних досліджень поведінки дрона без зовнішніх впливів було два PID-контролери використовувались для генерації команд управління БПЛА ξ_θ та ξ_φ .

Для стабілізації траєкторії сигнал контролера визначається трьома компонентами:

K_P ,

K_I ,

K_D

які є коефіцієнтами підсилення пропорційної, інтегральної та диференційних складових PID контролера.

Значення параметрів для експериментів подано в таблиці 3.4, а траєкторія польоту дрона подана рис. 3.8.

Таблиця 3.4 - Значення параметрів PID контролера

Параметр PID контролера	Значення параметра (розрахункові)	Значення параметра (з уточненням)
K_P ,	0,41	0,56
K_I	0,06	0,02
K_D	0,07	0,12

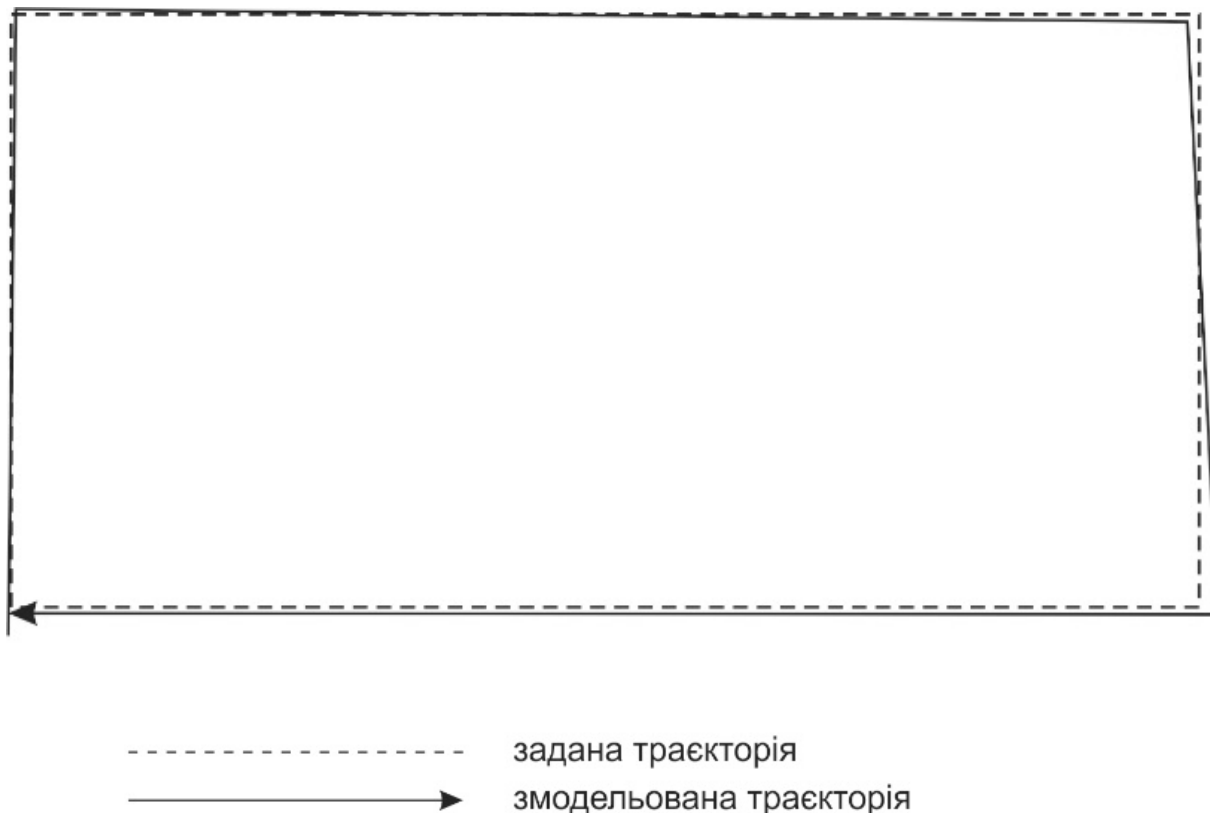


Рисунок 3.8 - Траєкторія польоту дрона без зовнішніх впливів

Модель керування дроном в середовищі Simulink Matlab [14] подано на рисунку 3.9.

Наступний експеримент полягав у дослідженні поведінки дрон в польоті із зовнішнім впливом вітру.

Результати моделювання, отримані на основі застосування нечіткого логічного контролера, представлені на рисунку 3.10.

Результати експериментальних досліджень продемонстрували, що залучення нечітких контролерів дуже наближені до необхідної еталонної траєкторії руху дрона, і демонструють вищу точність у порівнянні зі стоковим PID контролером.

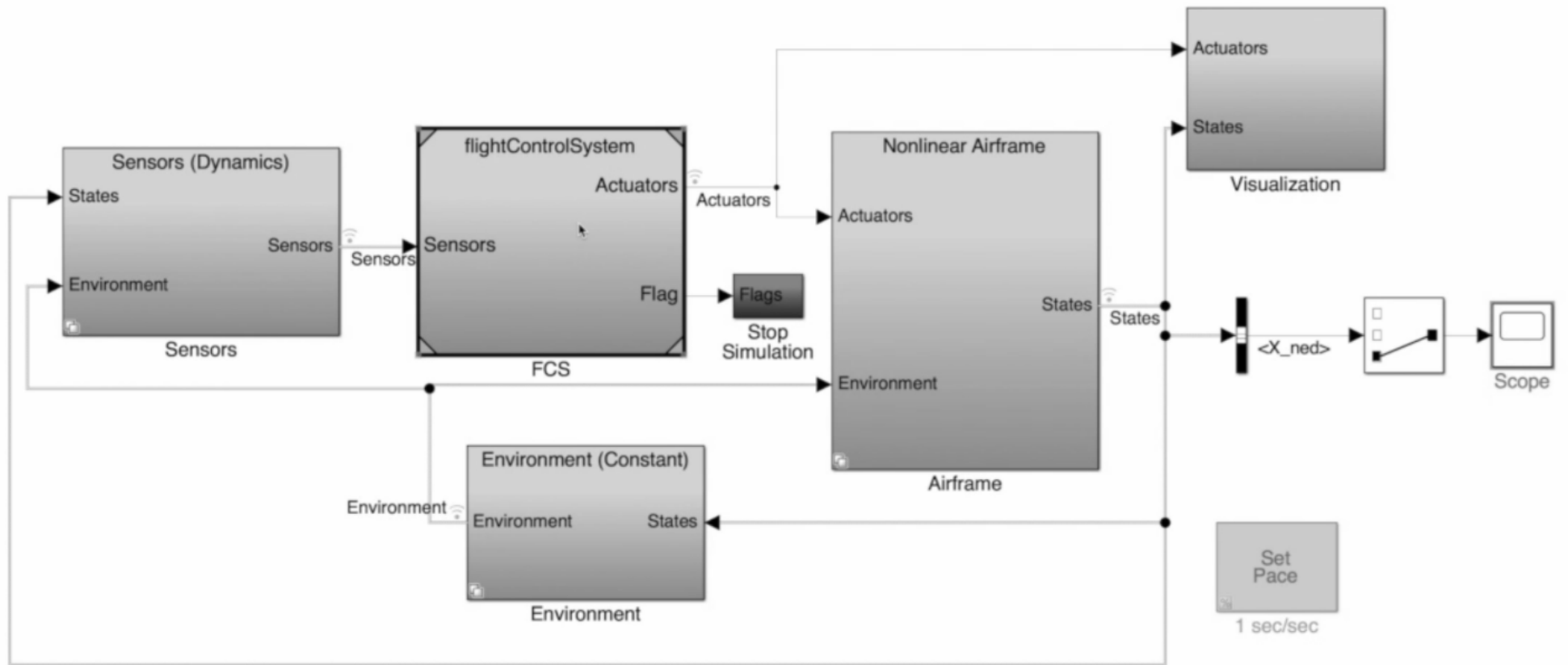
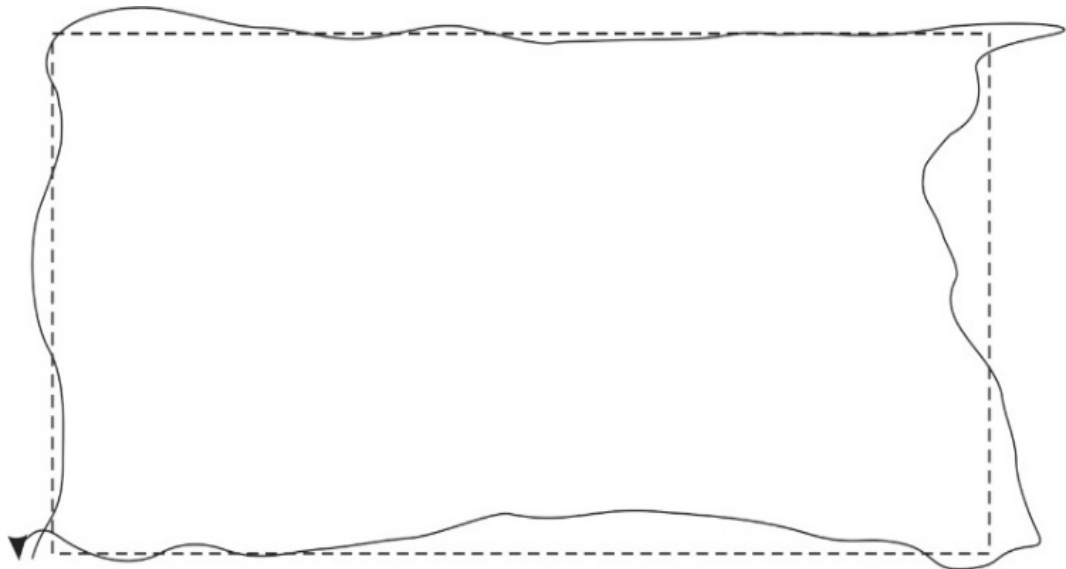
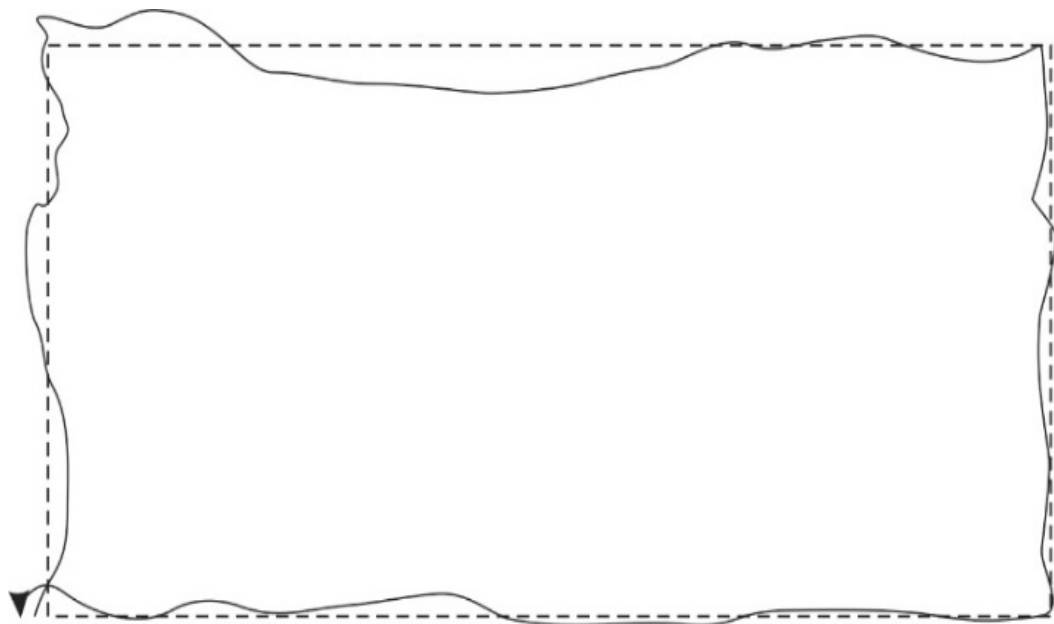


Рисунок 3.9 - Модель керування дроном в середовищі Simulink Matlab



----- задана траєкторія
 —————> змодельована траєкторія

а)



----- задана траєкторія
 —————> змодельована траєкторія

б)

Рисунок 3.10 -Траєкторія польоту:

а) без використання нечіткого контролера;

б) з використанням нечіткого контролера: без з урахуванням впливу вітру

3.4 Висновки

В розділі представлена інтелектуалізована система керування безпілотними літальними апаратами. В її основі лежить застосування апарату нечіткої логіки, імплементація якої в контролер БПЛА дозволила відстежувати та керувати траєкторію його руху.

Проведені експериментальні дослідження доводять ефективність застосування апарату нечіткої логіки для керуванням дрону в умовах здійснення зовнішніх впливів.

4 РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛІЗОВАНОЇ СИСТЕМИ КЕРУВАННЯ БЕЗПІЛОТНИМИ ЛІТАЛЬНИМИ АПАРАТАМИ

4.1 Використання системи MATLAB для реалізація інтелектуалізованої системи керування безпілотними літальними апаратами

З метою реалізації інтелектуалізованої системи керування безпілотними літальними апаратами було побудовано просту модель об'єкта дослідження в системі Simulink [49-60].

Для моделювання було створено підсистему, в якій в якості вихідних даних було використано x , y , h або z і Ψ , а в якості вхідних даних - $M1$, $M2$, $M3$ і $M4$ (рисунок 4.1).

Підсистема моделі включає кілька підсистем для представлення різних параметрів залежно від їх рівнянь та застосування.

Мета реалізації полягає в тому, щоб створити модель, що включає усі динамічні рівняння БПЛА, а потім побудувати систему управління для нього.

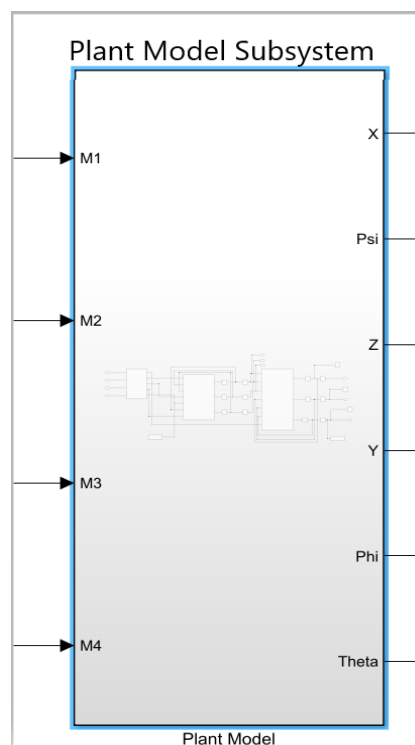


Рисунок 4.1 - Підсистема моделі БПЛА

Розглянемо відкриту підсистему моделі БПЛА, яка представлена на рисунку 4.2.

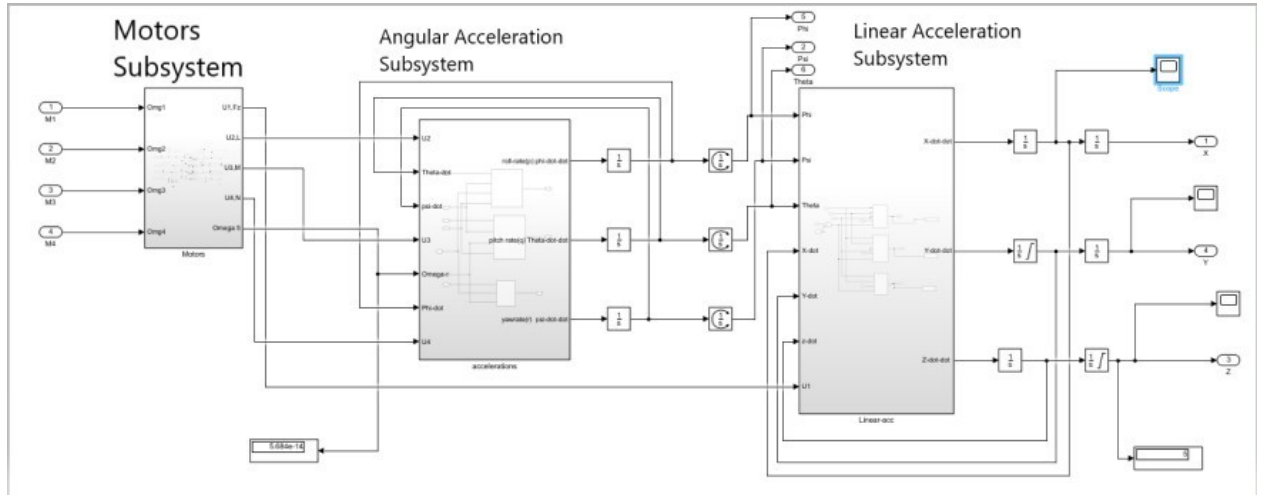


Рисунок 4.2 - Відкрита підсистема моделі БПЛА

Наступним кроком реалізації системи керування була підсистема задання орієнтації та керування двигунами (рисунку 4.3).

Для конкретної орієнтації пристрою, квадрокоптер має наступну схему, подану на рисунку 4.4.

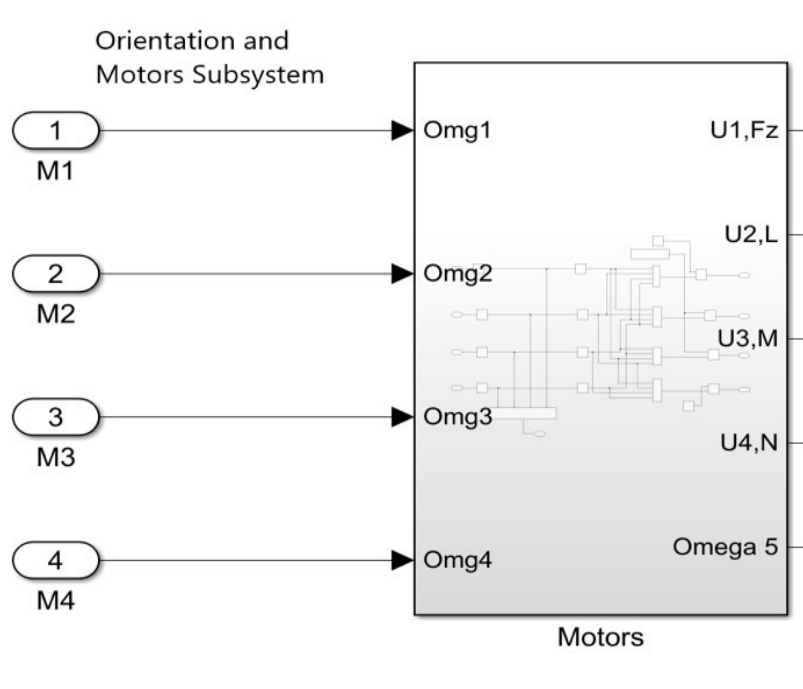


Рисунок 4.4 - Підсистема орієнтації та двигунів БПЛА

Підсистема задання орієнтації і керування двигунами виглядає наступним чином при відкритті (рисунок 4.5).

Дана підсистема приймає команди для двигунів і перетворює їх в моменти для управління орієнтацією і висотою польоту БПЛА.

Загальна кутова швидкість квадрокоптера задається гвинтами, які обертаються в протилежному напрямку для балансування загальної кутової швидкості.

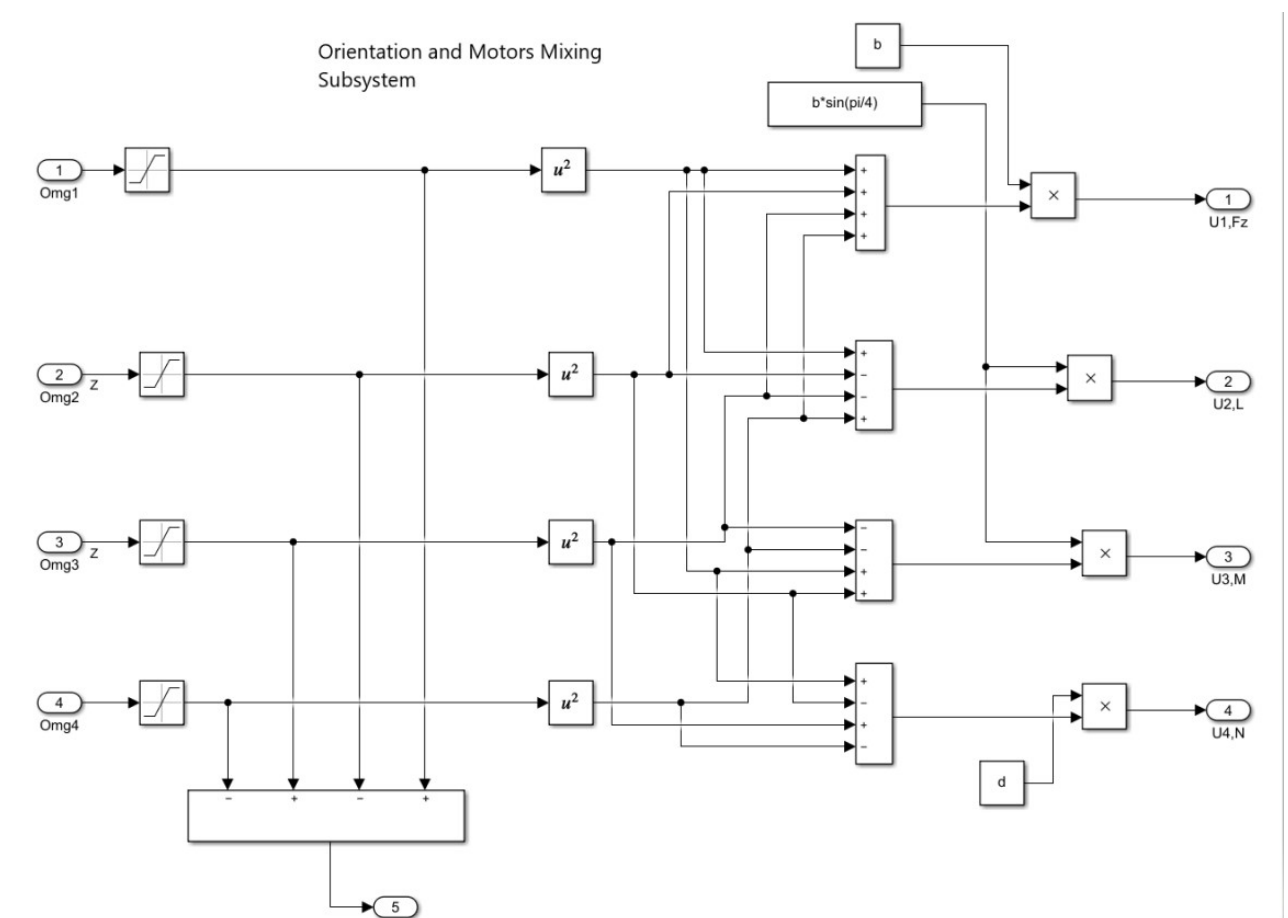


Рисунок 4.5 - Підсистема керування орієнтацією та керуванням двигунами в розгорнутому вигляді

Побудуємо підсистему кутового прискорення для розроблюваного БПЛА (рисунок 4.6).

Для цього приймемо параметри U_2 , U_3 , U_4 і Ω від підсистеми двигунів і виходи p , q і r .

У той час як U_1 подається прямо до наступної підсистеми, так як U_1 є повною тягою всіх двигунів в напрямку Z .

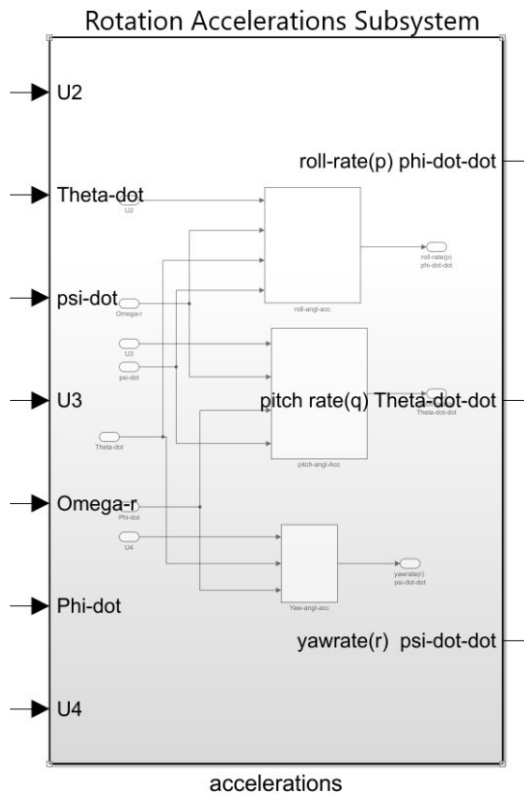


Рисунок 4.6 - Підсистема кутового прискорення

Підсистема кутового прискорення має три додаткові підсистеми для управління прискоренням обертання, нахилу і зміни курсу польоту БПЛА окремо.

Підсистеми можуть бути відкриті, щоб побачити рівняння і їх розташування (рисунок 4.7-4.10).

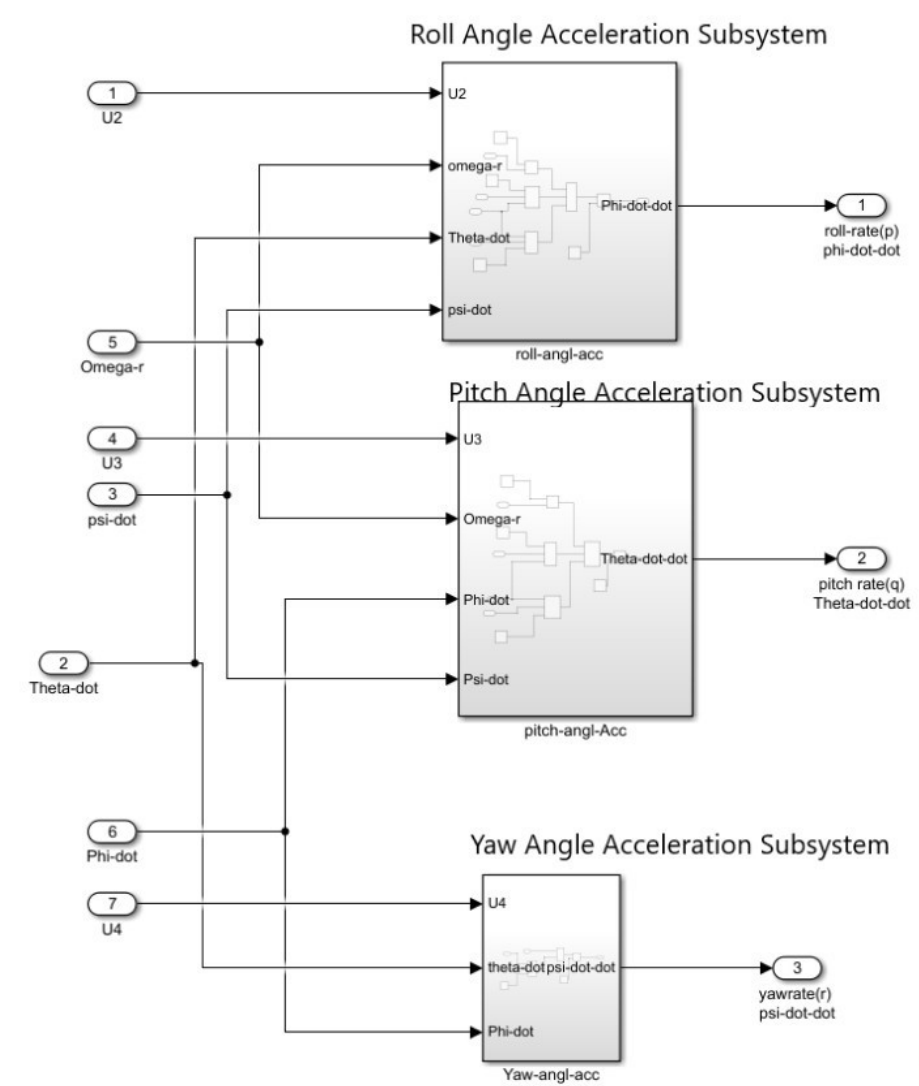


Рисунок 4.7 - Підсистема кутового прискорення

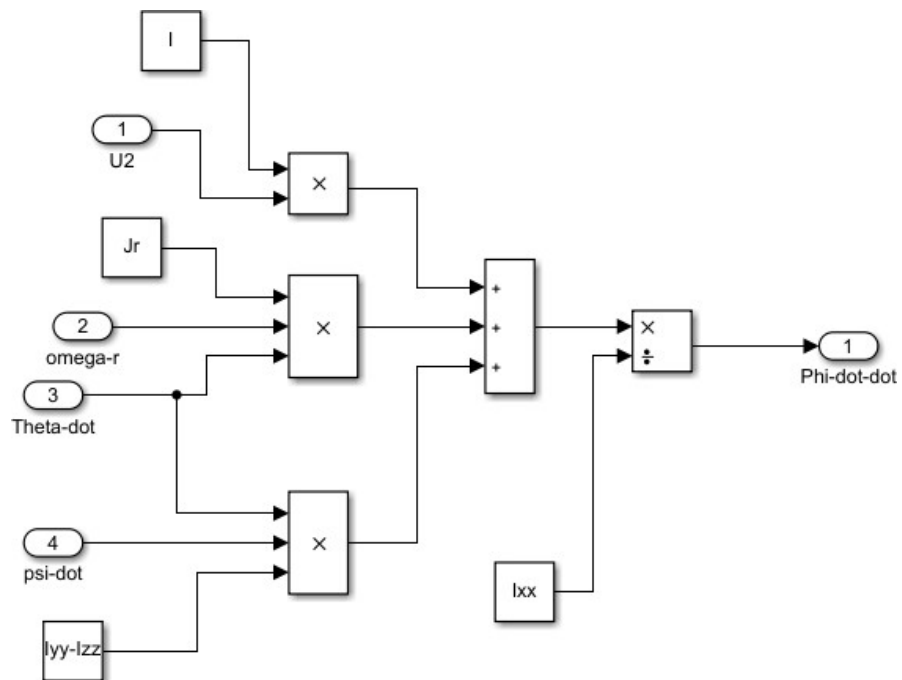


Рисунок 4.8 - Підсистема прискорення кута обертання

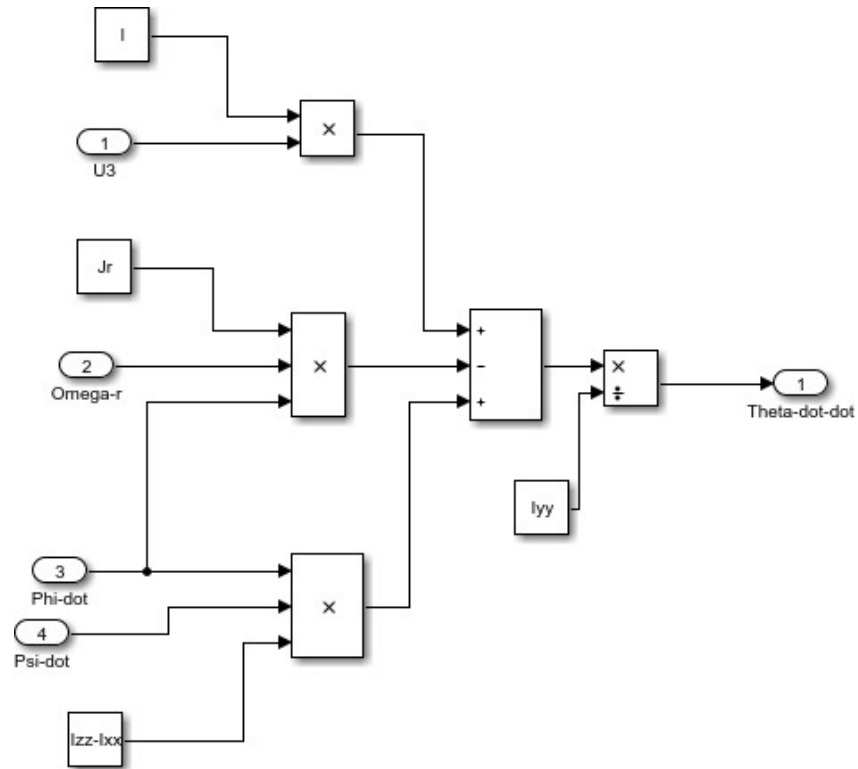


Рисунок 4.9 - Підсистема прискорення кута нахилу

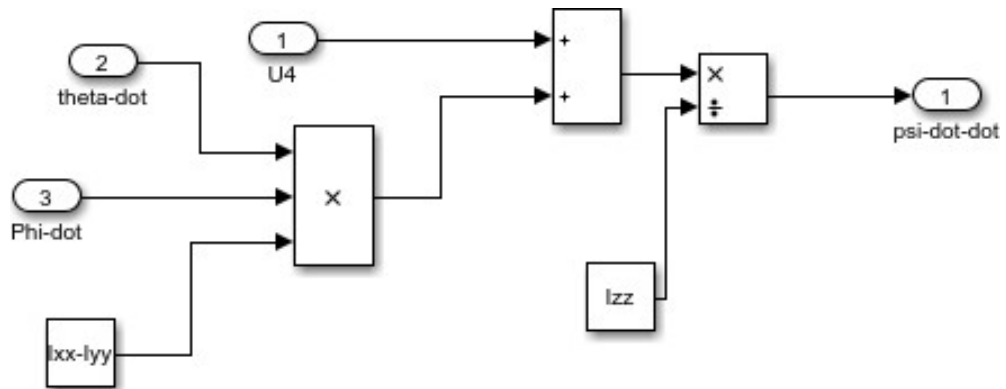


Рисунок 4.10 - Підсистема визначення кута рискання(ковзання)

Кутові швидкості попередньої підсистеми перетворюються в кути за допомогою інтеграторів.

Тепер підсистема лінійного прискорення (рисунок 4.11) перетворює кути в лінійне прискорення в інерційному фреймі в залежності від рівнянь координат навігації для задання траєкторії навігації БПЛА в інерційному координатному фреймі.

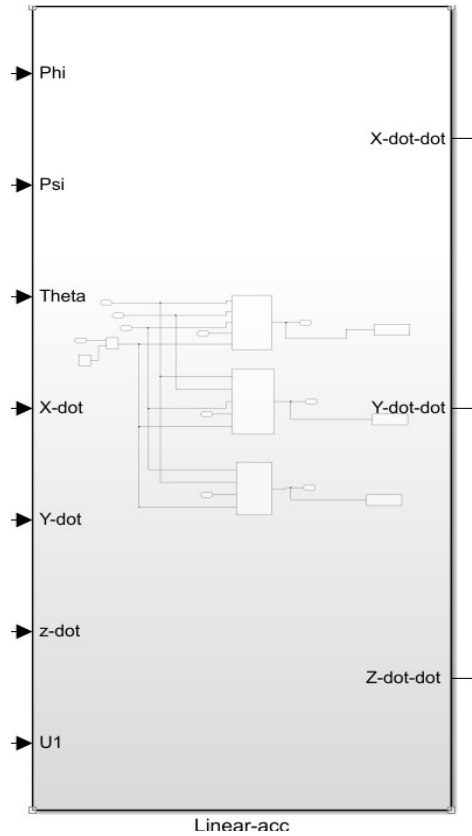


Рисунок 4.11 - Підсистема лінійного прискорення

Підсистема лінійного прискорення також має три підсистеми для прискорення x , y і z окремо, що містять рівняння для кожної з них (рис. 4.12-4.15).

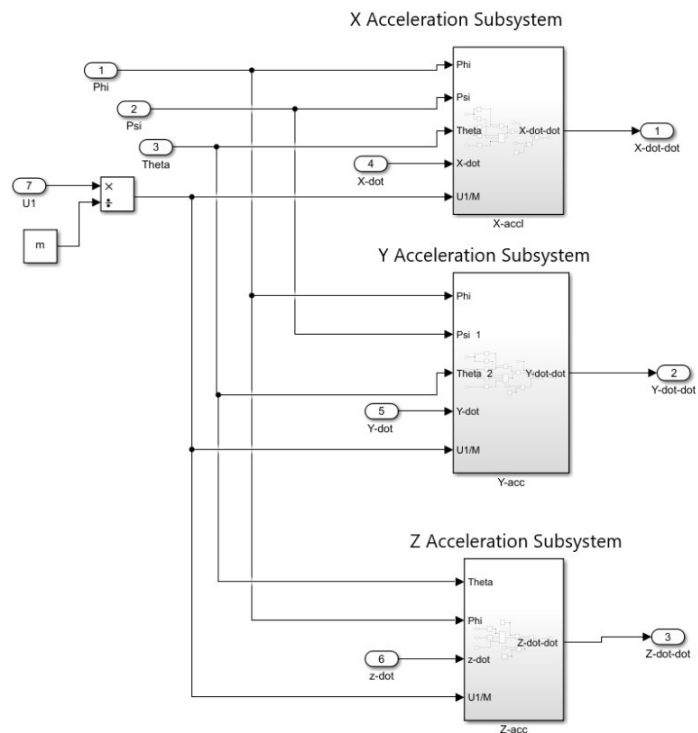


Рисунок 4.12 - Підсистема лінійного прискорення від зсередини

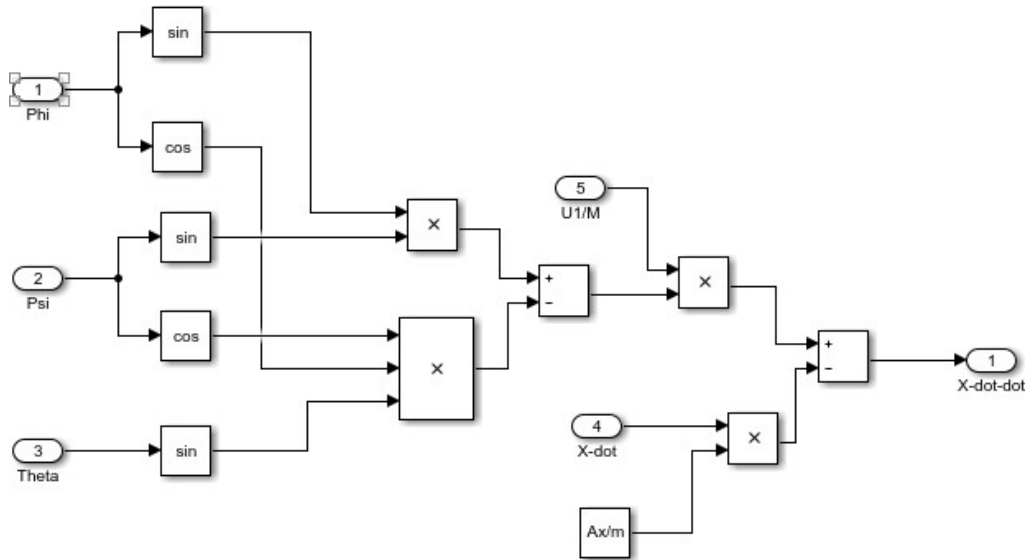


Рисунок 4.13 - Підсистема X-прискорення

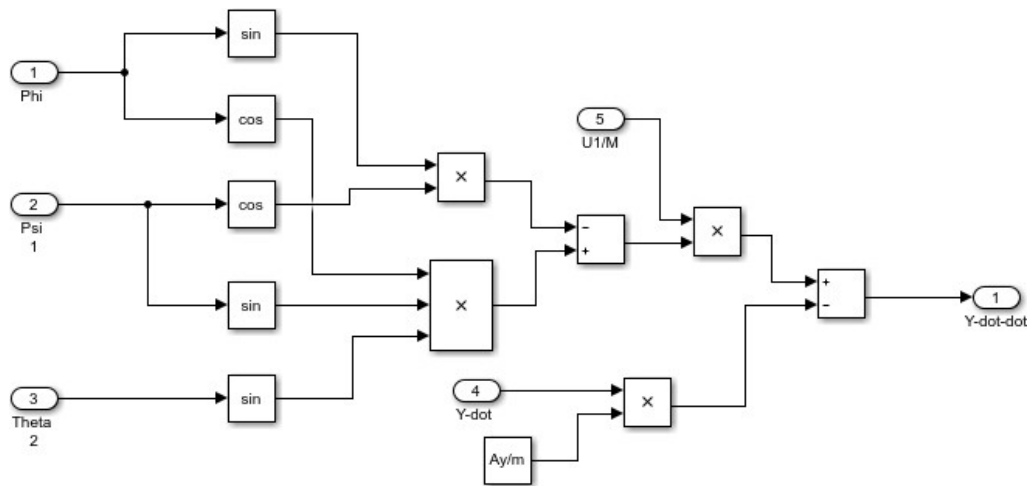


Рисунок 4.14 - Підсистема Y-прискорення

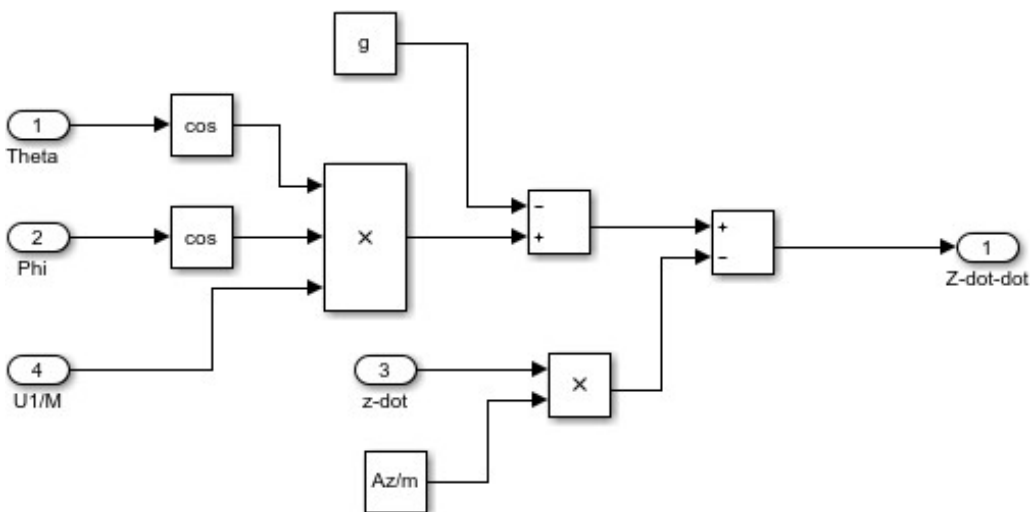


Рисунок 4.15 - Підсистема Z-прискорення

З метою перевірки реалізації моделі системи керування безпілотними літальними апаратами було проведено ряд експериментальних досліджень поведінки БПЛА в Simulink.

З цією метою розглянемо параметри ϕ , θ і ψ , зображені в одній області для порівняння і для спостереження за зміною поведінки.

Перша тяга забезпечується за рахунок обертання всіх двигунів з однаковою швидкістю. Реакція БПЛА показана на рис. 4.16.

Видно, що висота змінюється, в той час як кути обертання, нахилу і зміни траєкторії залишаються незмінними.

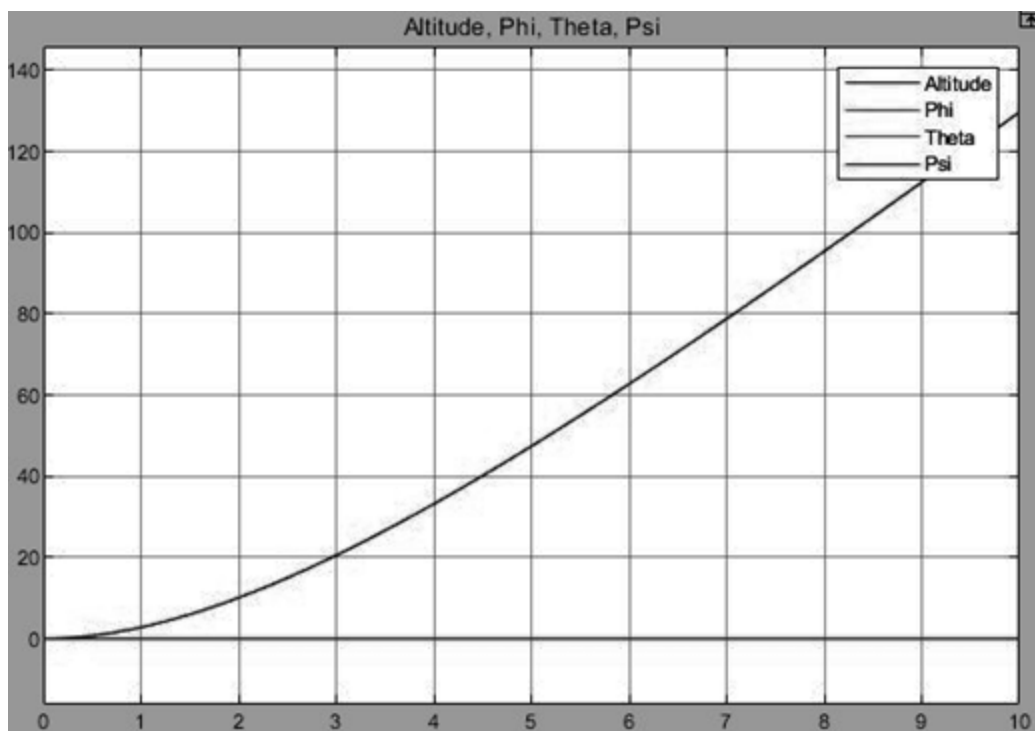


Рисунок 4.16 - Висота польоту БПЛА

Зміна швидкостей M_2 і M_4 або M_1 і M_3 одночасно призведе до зміни кута нахилу польоту БПЛА.

Аналогічним чином, зміна швидкостей M_1 і M_2 або M_3 і M_4 в один і той же час призведе до зміни кута обертання.

Для перевірки M_2 і M_4 збільшується, що призводить до зміни кута нахилу θ (рисунок 4.17).

Відповідно зміна швидкості всіх 4 двигунів впливає на кут ковзання ψ . Це підтверджує, що побудована модель БПЛА правильна і працює належним чином.

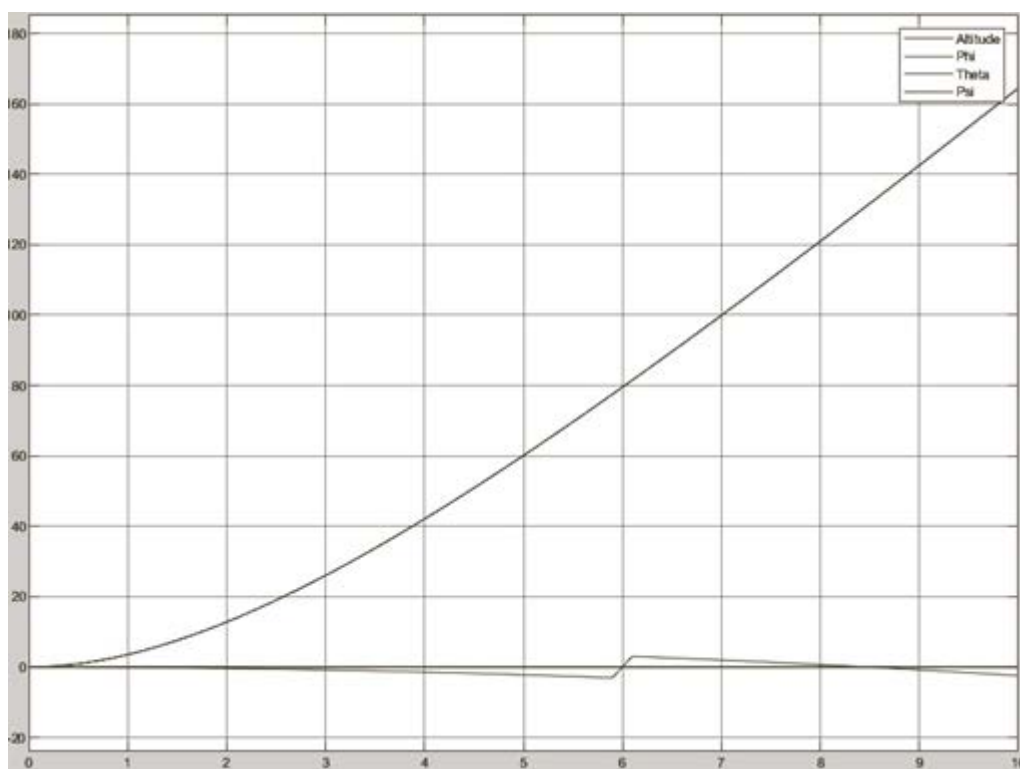


Рисунок 4.17 - Перевірка моделі Theta

4.2 Програмування контролера інтелектуалізованої системи керування безпілотними літальними апаратами

Основою керування траєкторією польоту є контролер.

Для програмування PID контролера з метою задання траєкторії було використано контролер висоти, оскільки необхідно регулювати висоту для настройки контролерів (обертання, нахилу і зміни напрямку).

Вбудовані блоки PID використовуються дозволили реалізувати в управлінні фактичними значеннями, отриманими з моделі, відповідно до потрібних вхідних значень.

Таким чином, було залучено чотири PID для висоти, обертання, нахилу і відхилення.

Наступним кроком є задання сигналів для двигунів. Тому результати роботи контролерів PID подаються на двигуни для задання орієнтації розробленого БПЛА. Результат виконання контролера виглядає як подано на рисунках 4.18 та 4.19.

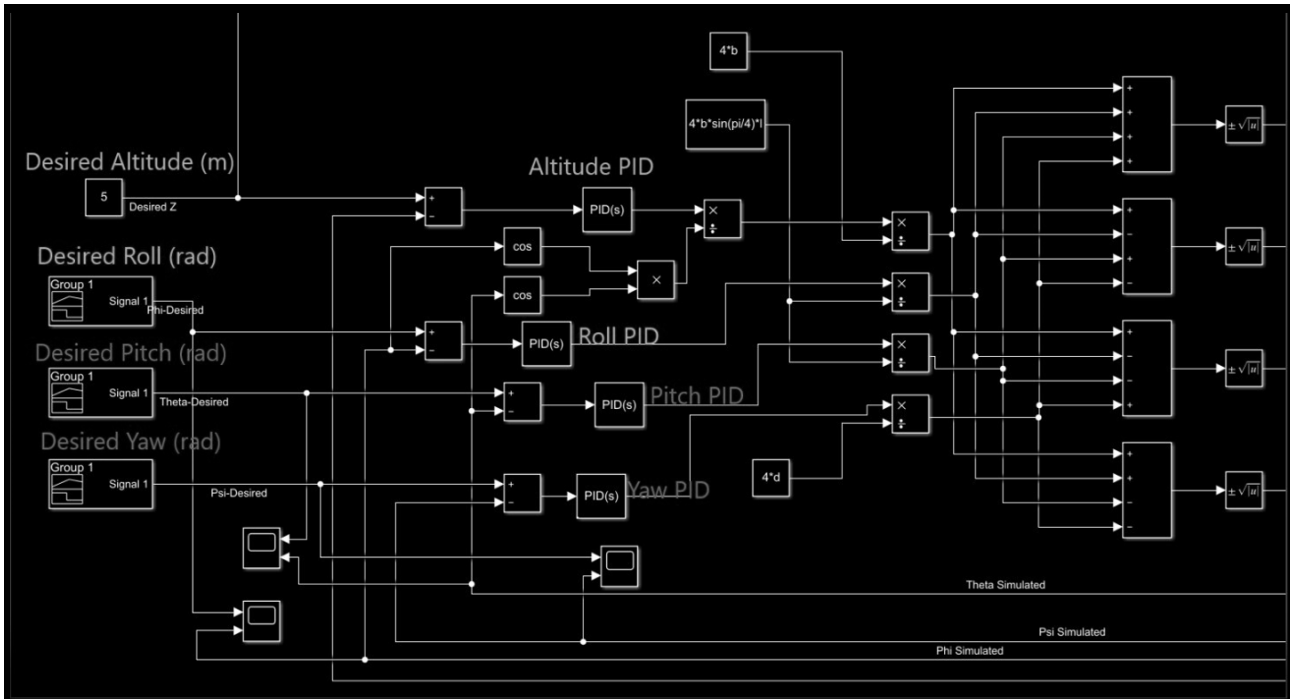


Рисунок 4.18 - Вхідні команди та контролер

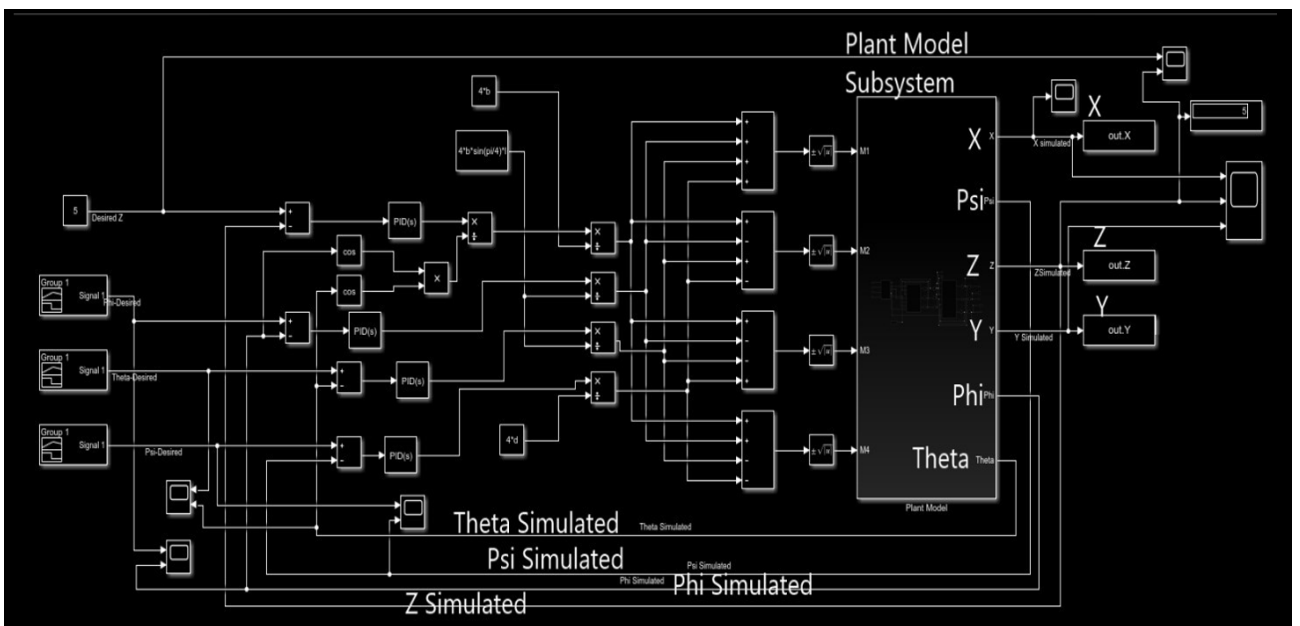


Рисунок 4.19 - Контролер із заводською моделлю

Таким чином, було налаштовано параметр P , в той час як значення I і D дорівнюють нулю, і продовжувати збільшувати значення, поки фактичне значення не перевищить бажане значення (див. Додаток А.1).

Значення P вказує діапазон значення, яке може бути скориговано при налаштуванні I і D .

Після обробки параметра P було опрацьовано параметр D , який прогнозує і видаляє майбутні помилки.

Експериментально було збільшено значення і спостерігався вплив на результати. Це дозволило усунути майбутні помилки, які можуть призвести до зменшення коливань в процесі моделювання, що дозволить згладити відповідь (рисунок А.2, Додаток А).

Тепер, фактична реакція значення дуже гладка, і перевищення мізерне, але є безперервна помилка, і фактичне значення відгуку не те ж саме, що бажане значення. Ця безперервна помилка може бути видалена інтегратором, який накопичує значення помилки в часі і намагається досягти бажаного значення для коректної траєкторії польоту.

На рисунок А.3, Додатку А показано, як було отримано зменшення помилки стійкого стану.

Після виконання деяких коригувань та фіксацій, повністю відрегульований PID (рисунок А.4, Додаток А) дав наступні результати. Результати цілком задовільні, квадрокоптер стабілізується на висоті 5 м менш ніж за 10 секунд. Ті ж цілі можуть бути досягнуті за менший проміжок часу, але його поведінка буде дуже агресивна.

Рисунок А.5 дозволяє побачити, що поведінку БПЛА є реальною, а саме: з початку запуску на БПЛА була передана команда набирання висоти 5м, і можна побачити, що на початку приблизно протягом трьох секунд БПЛА не злітає, бо сигнал недостатній.

Через три секунди він дуже плавно злітає і стабілізується відповідно до постійно зростаючого сигналу управління висотою БПЛА протягом перших 10 секунд польоту.

У таблиці 4.1 вказані коефіцієнти PID налаштованого PID-контролера висоти.

Контролери PID для обертання, нахилу і ковзання також налаштовуються аналогічно.

Таблиця 4.1. Параметр задання висоти PID

Параметр	Значення
P	4.3
I	4.5
D	2

Параметри Roll і Pitch мають ті ж значення посилення і ті ж відповіді, тому що вони працюють за тим же принципом.

На рисунку А.6 показана Roll і Pitch реакція транспортного засобу, коли бажаний кут складає 12 градусів і транспортний засіб намагається досягти висоти в 1 м одночасно.

У таблиці 4.2 показані PID значення налаштованого контролера PID.

Таблиця 4.2 - Приріст PID в результаті кроку і нахилу

Параметр	Значення
P	0.006
I	0.000009
D	0.013

Значення PID ковзання відрізняються від значення обертання і нахилу, яке необхідно повернути окремо. Важко налаштувати для Yaw PID, так як це залежить від крутного моменту, створюваного всіма чотирма роторами розробленого БПЛА.

У таблиці 4.3 наводяться значення приросту налаштованого контролера Yaw PID.

Таблиця 4.3 - Приріст Yaw PID

Параметр	Значення
P	0.01
I	0.0002
D	0.03

З метою задання траєкторії необхідним є використання генератора сигналів (рис. А 3.1).

В середовищі Simulink було здійснено задання сигналів для *roll*, *pitch* and *yaw* контролюються в системі координат.

Результати застосування генератора сигналів представлено на рис А.8.

Можна побачити, що розроблений пристрій реагує на команди (рис. А.9), що доводить, що PID налаштовані правильно.

На рисунку А.10 продемонстровано траєкторію польоту БПЛА.

Модель реалізації БПЛА виглядає наступним чином (рис. А.10-А12).

Наступним етапом реалізації системи керування безпілотними літальними апаратами була реалізація обробки сигнали для керування пристроєм. Модель приймача використовувалася для необхідного керування БПЛА.

Модель завантажується з підтримуючими файлами, а компілятор також завантажується для конфігурації блоку.

Блок дозволяє зчитувати чотири канали передавача, і використовується для управління сигналом обертання, нахилу, відхилення.

Блок Simulink BNO055 Arduino зчитує дані з BNO055IMU, і працює в режимі злиття, який можна вибрати з вікна параметрів блоку. У режимі злиття блок виводить відкалібровані значення усіх параметрів траєкторії польоту БПЛА.

В результаті реалізації було отримано калібровані значення в режимі синтезу, без необхідності фільтрації даних.

Кутові швидкості використовуються для передачі сигналу орієнтації до контролера.

В роботі було залучено двигуни постійного струму без щіток (BLDC). Вони керуються електронними регуляторами швидкості (ЕРШ), оскільки двигуни BLDC є трифазними.

ЕРШ має два дроти на одному кінці для напруги і заземлення, три дроти на іншому кінці для керування трифазним двигуном. Ще один комплект проводів в ЕРШ відповідають за підключення до сигнального проводу електронної плати і заземлення. ЕРШ використовують той же сигнал, який використовується для управління сервоприводами, 50HZ PWM сигнал з шириною імпульсу варіюється від 1 до 2 мілісекунд.

Таким чином, замість використання вихідних сигналів PWM (в діапазоні 0-255) з різною частотою виходу, було використано стандартні сервоприводи з діапазоном 0-180 з вихідною частотою в 50HZ для управління двигунами BLDC через ЕРШ.

PWM сигнал 50HZ отримує від блоку приймача в Simulink варіюється від 1150 до 2200 мікросекунд, де 1000 і 2000 представляють 0% і 100% ширини імпульсу відповідно. Коли цей сигнал проходить через контролер і алгоритм запуску двигуна, він множиться на значення PID. Таким чином, він змінює діапазон від значення за замовчуванням, яке не є прийнятним для ЕРШ.

Також ЕРШ може використовувати діапазон частоти імпульсів від 11150 до 2200 для управління двигуном, коли він безпосередньо підключений до реального приймача в апаратному забезпеченні. В роботі було 1150 до 2200 було підключено до приймача через мікроконтролер (ATmega 2560) з використанням стандартних двигунів з прийнятним значенням сигналу від 0 до 180. Таким чином, було вказано вихідні значення двигунів з 0 до 180.

4.3 Автономне відстеження траєкторії квадродротора

У цьому дослідженні використовується AR.Drone 2.0. Інтерфейс MATLAB / Simulink розроблений для передачі даних на AR.Drone та зчитування даних з AR.Drone. Розроблений інтерфейс Wi-Fi дозволяє швидко виконувати тести ідентифікації, контролю та наведення системи. Керування квадродротором для автономного досягнення контрольних точок досягається за допомогою вищеописаних контролерів. На першому етапі дослідження використовується класичний алгоритм управління PID, а на другому етапі в цих блоках використовується самоналаштовуваний нечіткий алгоритм управління PID для відстеження траєкторії.

4.4 Платформа квадродротора AR.Drone та передача даних

AR.Drone 2.0 має конструкцію квадродротора з поперечним типом. AR.Drone 2.0 живиться від Li-Po акумулятора потужністю 11,1 В, 1500 мАг зі швидкістю розряду 10 С. Зарядний стан постійно контролюється вбудованою системою. Літак активується чотирма трифазними безщітковими двигунами постійного струму потужністю 14,5 Вт і 28 500 об / хв. Драйвери двигуна оснащені датчиками для отримання інформації про стан зупинки та повороту двигуна. Для кожного драйвера доступний 8-MIPS AVR-процесор. Цей дрон підходить для академічних досліджень завдяки своїм електронним та механічним характеристикам.

Основна плата AR.Drone 2.0 містить 1 ГГц 32-бітний процесор ARM Cortex A8 і процесор цифрових сигналів TMS320DMC64X 800 МГц. Система також має пам'ять оперативної пам'яті DDR2 на 200 МГц і 1 ГБ. Завдяки таким функціям він здатний запускати 32-розрядну операційну систему. Материнська плата містить модулі для спілкування із зовнішнім світом. Ці модулі, зокрема, включають протокол бездротового зв'язку IEEE 802.11, що дозволяє здійснювати зв'язок TCP / IP. Цей протокол використовує ту саму частоту 2,4 ГГц, що і бездротова локальна мережа, яка широко використовується сьогодні. Зв'язок між БПЛА і контролером

(телеоператором або головним комп'ютером) досягається за допомогою використання трьох різних каналів зв'язку UDP (User Datagram Protocol) через з'єднання Wi-Fi на 100 м покриття. Початкові команди, що передаються на літак по командному каналу, - це земля, зліт, регулювання меж, калібрування датчиків, зміна камер, висота тону, нахил, похиска та регулювання вертикальної швидкості. Дані про стан БПЛА передаються каналом NavData для отримання зворотного зв'язку. Ці дані включають поточний стан БПЛА (стан польоту, посадки, зльоту та калібрування БПЛА) та дані датчиків (поточний кут нахилу, кут нахилу, швидкість гойдалки, висоту, рівень заряду акумулятора та швидкість по всіх осях). Нарешті, візуальні дані з камер передаються по потоковому каналу. Відстань зв'язку можна легко збільшити за допомогою реле маршрутизатора та розширювача. Іншим модулем зв'язку є порт USB 2.0, що входить до БПЛА. За потреби будь-які USB-пристрої, напр. GPS і флеш-пам'ять можна вставити в дрон.

Квадратором AR.Drone можна керувати за допомогою комп'ютера з бездротовим з'єднанням IEEE 802.11. Команди AT передаються через порт № 5556 в пакетах UDP. Цей порт прослуховується вбудованою системою AR.Drone. Вбудована система обробляє передані команди кожні 30 мс, щоб забезпечити більш м'який рух БПЛА. Система припускає, що бездротове з'єднання відключається, коли між двома послідовними командними рядками є зазор більше 2 с, а потім переходить у режим аварійної посадки.

У цьому дослідженні MATLAB / Simulink використовується для зв'язку з БПЛА. Спочатку підключення до Інтернету БПЛА вибирається зі списку доступних бездротових з'єднань. Зв'язок встановлюється за допомогою протоколу UDP з використанням блоку UDP в Simulink. Після встановлення зв'язку контрольні сигнали передаються на літак. Одночасно літак починає передавати дані на комп'ютер. Цей обмін даними між БПЛА та моделлю MATLAB / Simulink в комп'ютері виконується на частоті 16 Гц. Дані, отримані MATLAB, зчитуються та записуються. Таким чином, БПЛА безпосередньо керують бездротовим способом за допомогою Simulink. Бездротове підключення, яке встановлено таким чином, забезпечує швидке наведення та управління.

4.5 Висновки

В розділі представлено реалізацію інтелектуалізованої системи керування безпілотними літальними апаратами. З цією метою було використано систему MATLAB.

В розділі подано реалізації:

- 1) підсистеми керування орієнтацією та керуванням двигунами в розгорнутому вигляді;
- 2) підсистеми кутового прискорення;
- 3) підсистеми прискорення кута обертання;
- 4) підсистема лінійного прискорення.

В розділі наведено схеми підсистем X-прискорення, Y-прискорення, Z-прискорення.

В розділі представлено розроблений контролер інтелектуалізованої системи керування безпілотними літальними апаратами. Подано реалізацію автономного відстеження траєкторії дрона та описано використану для досліджень платформу дрона AR.Drone.

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено інтелектуалізовану систему керування безпілотними літальними апаратами.

У першому розділі досліджено методи планування траєкторії польоту безпілотних літальних апаратів, зроблено висновки та постановку задачі

У другому розділі представлено розроблену математичну модель системи керування безпілотними літальними апаратами, яка описує рух БПЛА у всіх напрямках польоту разом з описом методу управління дроном. Також в розділі подано опис функціонування PID-контролера, а також описано модель самоналаштуваного нечіткого PID-контролера.

У третьому розділі представлена інтелектуалізована система керування безпілотними літальними апаратами, в основі якої покладено застосування апарату нечіткої логіки, імплементація якої в контролер БПЛА дозволила відстежувати та керувати траєкторію його руху в високою точністю. В розділі представлені також експериментальні дослідження, які довели ефективність застосування апарату нечіткої логіки для керуванням дроном.

У четвертому розділі представлено реалізацію інтелектуалізованої системи керування безпілотними літальними апаратами. В розділі представлено розроблений контролер інтелектуалізованої системи керування безпілотними літальними апаратами. Подано реалізацію автономного відстеження траєкторії дрона та описано використану для досліджень платформу дрона AR.Drone.

За темою дипломної роботи опубліковано статтю на тему «Інтелектуалізована система керування безпілотними літальними апаратами» в журналі «Комп'ютерні системи та інформаційні технології» №1 за 2020 рік.

Наукова новизна отриманих результатів:

1. Удосконалено метод керування безпілотними літальними апаратами, який на відміну від відомих застосовує компоненти штучного інтелекту, і який

забезпечує підвищення точності траєкторії руху безпілотного літального апарату типу дрон.

2. Набула подальшого розвитку інтелектуалізована система керування безпілотними літальними апаратами, яка забезпечує підвищення точності траєкторії руху безпілотного літального апарату типу дрон.

Практична цінність. В результаті виконаного наукового дослідження буде розроблено інтелектуалізовану систему керування безпілотними літальними апаратами, яка забезпечуватиме задання траєкторії руху безпілотного літального апарату типу дрон з високою точністю.

Зв'язок роботи з науковими програмами, планами, темами. Дослідження, представлені у кваліфікаційній роботі, проводились в рамках держбюджетної НДР Хмельницького національного університету № 1Б-2019 «Агентно-орієнтована система підвищення безпеки та якості програмного забезпечення комп'ютерних систем» (номер державної реєстрації 0119U100662).

За темою дипломної роботи опубліковано статтю на тему «Інтелектуалізована система керування безпілотними літальними апаратами» в журналі «Комп'ютерні системи та інформаційні технології» №1 за 2020 рік [48].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Basilico, N.; Carpin, S. Deploying Teams of Heterogeneous UAVs in Cooperative Two-level Surveillance Missions. *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, 28 September–2 October 2015; pp. 610–615.
2. Lottes, P.; Khanna, R.; Pfeifer, J.; Siegwart, R.; Stachniss, C. UAV-Based Crop and Weed Classification for Smart Farming. *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 29 May–3 June 2017; pp. 3024–3031.
3. Cesetti, A.; Frontoni, E.; Mancini, A.; Ascani, A.; Zingaretti, P.; Longhi, S. A visual global positioning system for unmanned aerial vehicles used in photogrammetric applications. *J. Intell. Robot. Syst.* 2011, 61, 157–168.
4. Maza, I.; Caballero, F.; Capitán, J.; Martínez-de Dios, J.R.; Ollero, A. Experimental results in multi-UAV coordination for disaster management and civil security applications. *J. Intell. Robot. Syst.* 2011, 61, 563–585.
5. Pham, H.X.; La, H.M.; Feil-Seifer, D.; Deans, M. A Distributed Control Framework for A Team of Unmanned Aerial. In *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, 24–28 September 2017; pp. 6648–6653.
6. Renzaglia, A.; Reymann, C.; Lacroix, S. Monitoring the Evolution of Clouds with UAVs. In *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, 16–21 May 2016; pp. 278–283.
7. Guerrero, J.A.; Bestaoui, Y. UAV path planning for structure inspection in windy environments. *J. Intell. Robot. Syst.* 2013, 69, 297–311.
8. Chang, W.; Yang, G.; Yu, J.; Liang, Z.; Cheng, L.; Zhou, C. Development of A Power Line Inspection Robot with Hybrid Operation modes. In *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, 24–28 September; pp. 973–978.

9. Choset, H. Coverage for robotics—A survey of recent results. *Ann. Math. Artif. Intell.* 2001; pp. 113–126.
10. Barrientos, A.; Colorado, J.; del Cerro, J.; Martinez, A.; Rossi, C.; Sanz, D.; Valente, J. Aerial remote sensing in agriculture: A practical approach to area. *J. Field Robot.* 2011, 28, 667–689.
11. Alvarenga, J.; Vitzilaios, N.I.; Valavanis, K.P.; Rutherford, M.J. Survey of unmanned helicopter model-based navigation and control techniques. *J. Intell. Robot. Syst.* 2015, 80, 87–138.
12. Kanellakis, C.; Nikolakopoulos, G. Survey on computer vision for UAVs: Current developments and trends. *J. Intell. Robot. Syst.* 2017, 87, 141–168.
13. Saeed, A.S.; Younes, A.B.; Islam, S.; Dias, J.; Seneviratne, L.; Cai, G. A Review on the Platform Design, Dynamic Modeling and Control of Hybrid UAVs. *In Proceedings of the 2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, Denver, CO, USA, 9–12 June 2015; pp. 806–815.
14. Watts, A.C.; Ambrosia, V.G.; Hinkley, E.A. Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use. *Remote Sens.* 2012, 4, 1671–1692.
15. Hoffer, N.V.; Coopmans, C.; Jensen, A.M.; Chen, Y. A survey and categorization of small low-cost unmanned aerial vehicle system identification. *J. Intell. Robot. Syst.* 2014, 74, 129–145.
16. Goerzen, C.; Kong, Z.; Mettler, B. A survey of motion planning algorithms from the perspective of autonomous UAV guidance. *J. Intell. Robot. Syst.* 2010, 57, 65.
17. Dadkhah, N.; Mettler, B. Survey of motion planning literature in the presence of uncertainty: Considerations for UAV guidance. *J. Intell. Robot. Syst.* 2012, 65, 233–246.
18. Kanistras, K.; Martins, G.; Rutherford, M.J.; Valavanis, K.P. A Survey of Unmanned Aerial Vehicles (UAVs) for Traffic Monitoring. *In Proceedings of the 2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, Atlanta, GA, USA, 28–31 May 2013; pp. 221–234.

- 19.Colomina, I.; Molina, P. Unmanned aerial systems for photogrammetry and remote sensing: A review.
- 20.ISPRS J. Photogramm. *Remote Sens.* 2014, 92, 79–97.
- 21.Gautam, A.; Sujit, P.; Saripalli, S. A Survey of Autonomous Landing Techniques for UAVs. In Proceedings of the 2014 *International Conference on Unmanned Aircraft Systems (ICUAS)*, Orlando, FL, USA, 27–30 May 2014; pp. 1210–1218.
- 22.Galceran, E.; Carreras, M. A survey on coverage path planning for robotics. *Robot. Auton. Syst.* 2013,61, 1258–1276.
- 23.Juliá, M.; Gil, A.; Reinoso, O. A comparison of path planning strategies for autonomous exploration and mapping of unknown environments. *Autonomous Robots* 2012, 33, 427–444.
- 24.McFerran, D.; Swider, M. The Best Drone 2018: DJI, Parrot and More for Beginners and Pros. Available online. URL: <https://www.techradar.com/news/best-drones> (дата доступу вересень 2020).
- 25.Di Franco, C.; Buttazzo, G. Energy-Aware Coverage Path Planning of UAVs. In *Proceedings of the IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, Vila Real, Portugal, 8–10 April 2015; pp. 111–117.
- 26.Di Franco, C.; Buttazzo, G. Coverage path planning for UAVs photogrammetry with energy and resolution constraints. *J. Intell. Robot. Systems* 2016, 1–18.
- 27.Ware, J.; Roy, N. An Analysis of Wind Field Estimation and Exploitation for Quadrotor Flight in the Urban Canopy Layer. In *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, 16–21 May 2016; pp. 1507–1514.
- 28.Kakaes, K.; Greenwood, F.; Lippincott, M.; Dosemagen, S.; Meier, P.; Wich, S. Drones and Aerial Observation: New Technologies for Property Rights, Human Rights, and Global Development: A Primer; New America: Washington, DC, USA, 2015; pp. 6–103.
- 29.Sampaio, P.A.; da Silva Sousa, R.; Rocha, A.N. Reducing the range of perception in multi-agent patrolling strategies. *J. Intell. Robot. Syst.* 2018, 91, 219–231.

30. Öst, G. Search Path Generation with UAV Applications Using Approximate Convex Decomposition. Master's Thesis, Linköpings Universitet, Linköpings, Sweden, 2012.
31. Xu, A.; Viriyasuthee, C.; Rekleitis, I. Optimal Complete Terrain Coverage Using An Unmanned Aerial Vehicle. *In Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, 9–13 May 2011; pp. 2513–2519.
32. Artemenko, O.; Dominic, O.J.; Andryeyev, O.; Mitschele-Thiel, A. Energy-Aware Trajectory Planning for the Localization of Mobile Devices Using an Unmanned Aerial Vehicle. *In Proceedings of the 2016 25th International Conference on Computer Communication and Networks (ICCCN)*, Waikoloa, HI, USA, 1–4 August 2016; pp. 1–9.
33. Valente, J.; Sanz, D.; Del Cerro, J.; Barrientos, A.; de Frutos, M.Á. Near-optimal coverage trajectories for image mosaicing using a mini quad-rotor over irregular-shaped fields. *Prec. Agric.* 2013, 14, 115–132.
34. Li, D.; Wang, X.; Sun, T. Energy-optimal coverage path planning on topographic map for environment survey with unmanned aerial vehicles. *Electron. Lett.* 2016, 52, 699–701.
35. Maza, I.; Ollero, A. Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms. *In Distributed Autonomous Robotic Systems 6; Springer: Berlin, Germany*, 2007; pp. 221–230.
36. Torres, M.; Pelta, D.A.; Verdegay, J.L.; Torres, J.C. Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction. *Expert Syst. Appl.* 2016, 55, 441–451.
37. Nattero, C.; Recchiuto, C.T.; Sgorbissa, A.; Wanderlingh, F. Coverage Algorithms for Search and Rescue with UAV Drones. *In Proceedings of the Workshop of the XIII AI*IA Symposium on Artificial Intelligence*, Pisa, Italy, 10–12 December 2014.
38. Osborne, M. Mission Planner-Ground Station. Available online: <http://planner.ardupilot.com> (accessed on 15 October 2015).

39. Andersen, H.L. Path Planning for Search and Rescue Mission Using Multicopters. Master's Thesis, Institutt for Teknisk Kybernetikk, Trondheim, Norway, 2014.
40. Coombes, M.; Chen, W.H.; Liu, C. Boustrophedon Coverage Path Planning for UAV Aerial Surveys in Wind. *In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, Miami, FL, USA, 13–16 June 2017; pp. 1563–1571.
41. Cabreira, T.M.; Di Franco, C.; Ferreira, P.R., Jr.; Buttazzo, G.C. Energy-Aware Spiral Coverage Path Planning for UAV Photogrammetric Applications. *IEEE Robot. Autom. Lett.* 2018, 3, 3662–3668.
42. Forsmo, E.J.; Grøtli, E.I.; Fossen, T.I.; Johansen, T.A. Optimal Search Mission with Unmanned Aerial Vehicles Using Mixed Integer Linear Programming. *In Proceedings of the 2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, Atlanta, GA, USA, 28–31 May 2013; pp. 253–259.
43. Ahmadzadeh, A.; Keller, J.; Pappas, G.; Jadbabaie, A.; Kumar, V. An Optimization-Based Approach to Time-Critical Cooperative Surveillance and Coverage with UAVs. *In Experimental Robotics*; Springer: Berlin, Germany, 2008; pp. 491–500.
44. Jiao, Y.S.; Wang, X.M.; Chen, H.; Li, Y. Research on the Coverage Path Planning of UAVs for Polygon Areas. *In Proceedings of the 2010 5th IEEE Conference on Industrial Electronics and Applications*, Taichung, Taiwan, 15–17 June 2010; pp. 1467–1472.
45. Li, Y.; Chen, H.; Er, M.J.; Wang, X. Coverage path planning for UAVs based on enhanced exact cellular decomposition method. *Mechatron. Spec. Issue Dev. Auton. Unmanned Aer. Veh.* 2011, 21, 876–885.
46. Levcopoulos, C.; Krznaric, D. Quasi-Greedy Triangulations Approximating the Minimum Weight Triangulation. *Journal of Algorithms* 1998, 27, 303–338. doi:10.1006/jagm.1997.0918
47. Huang, W.H. Optimal Line-sweep-based Decompositions for Coverage Algorithms. *In Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, 21–26 May 2001; Volume 1, pp. 27–32.

48. Лисенко С., Румянцев С. Інтелектуалізована система керування безпілотними літальними апаратами. Комп'ютерні системи та інформаційні технології. 2020. № 1. С. 21-26.
49. R. S. Carapu, A. V. Rodrigues, M. M. Marques, V. Lobo and F. Coito, "Interoperability of unmanned systems in military maritime operations: Developing a controller for unmanned aerial systems operating in maritime environments," OCEANS 2017 - Aberdeen, Aberdeen, 2017, pp. 1-7, doi: 10.1109/OCEANSE.2017.8084862.
50. R. P. Padhy, S. K. Choudhury, P. K. Sa and S. Bakshi, "Obstacle Avoidance for Unmanned Aerial Vehicles: Using Visual Features in Unknown Environments," in IEEE Consumer Electronics Magazine, vol. 8, no. 3, pp. 74-80, May 2019, doi: 10.1109/MCE.2019.2892280.
51. N. Kumar, D. Puthal, T. Theocharides and S. P. Mohanty, "Unmanned Aerial Vehicles in Consumer Applications: New Applications in Current and Future Smart Environments," in IEEE Consumer Electronics Magazine, vol. 8, no. 3, pp. 66-67, May 2019, doi: 10.1109/MCE.2019.2892278.
52. M. Alharbi, E. Alghamdi, S. Alzaid and S. M. Qaisar, "Unmanned Aerial Vehicle for Human Surveillance by Using Face and Upperbody Classifiers," 2020 2nd International Conference on Computer and Information Sciences (ICCIS), Sakaka, Saudi Arabia, 2020, pp. 1-4, doi: 10.1109/ICCIS49240.2020.9257659.
53. H. Lee, H. Liu, J. Lee, J. Lee and D. Jung, "Hybrid Multi-DOF Motor for Multi-Copter Unmanned Aerial Vehicle," in IEEE Transactions on Magnetics, vol. 55, no. 6, pp. 1-5, June 2019, Art no. 8201205, doi: 10.1109/TMAG.2019.2895135.
54. P. Hügler, M. Geiger and C. Waldschmidt, "77 GHz radar-based altimeter for unmanned aerial vehicles," 2018 IEEE Radio and Wireless Symposium (RWS), Anaheim, CA, 2018, pp. 129-132, doi: 10.1109/RWS.2018.8304965.
55. A. E. Cohen, "Automated HDL signal processing deployment performance from high level MATLAB specification for an unmanned aerial vehicle (UAV)," 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, 2018, pp. 900-905, doi: 10.1109/CCWC.2018.8301664.

- 56.Z. Chen and B. Xu, "Control simulation and anti-jamming verification of quadrotor UAV Based on Matlab," 2020 5th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Okinawa, Japan, 2020, pp. 70-75, doi: 10.1109/ICIIBMS50712.2020.9336427.
- 57.Z. Ligu, W. Lu, J. Shude, Z. Wangwang and W. Feng, "UAV Flight Reappearance and Failure Analysis," 2018 Chinese Automation Congress (CAC), Xi'an, China, 2018, pp. 1784-1787, doi: 10.1109/CAC.2018.8623367.
- 58.W. Jun, Y. Xiong-Dong and T. Yu-Yang, "Fault-Tolerant Control Design of Quadrotor UAV Based on CPSO," 2018 IEEE 4th International Conference on Control Science and Systems Engineering (ICCSSE), Wuhan, China, 2018, pp. 279-283, doi: 10.1109/CCSSE.2018.8724803.
- 59.E. ERSOY and M. K. YALÇIN, "Matlab/Simulink ve X-Plane Uçuş Benzetim Programı Arası UDP Haberleşme ile Veri Transferi," 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 2019, pp. 1-4, doi: 10.1109/ISMSIT.2019.8932950.
- 60.A. Kaviyarasu, A. Saravanakumar and M. L. Venkatesh, "Hardware in Loop Simulation of a Way Point Navigation Using Matlab/Simulink and X-Plane Simulator," 2019 International Conference on Intelligent Sustainable Systems (ICISS), Palladam, Tamilnadu, India, 2019, pp. 332-335, doi: 10.1109/ISS1.2019.8908103.

Додаток А
(обов'язковий)

Результати використання системи MATLAB для реалізація інтелектуалізованої системи керування безпілотними літальними апаратами

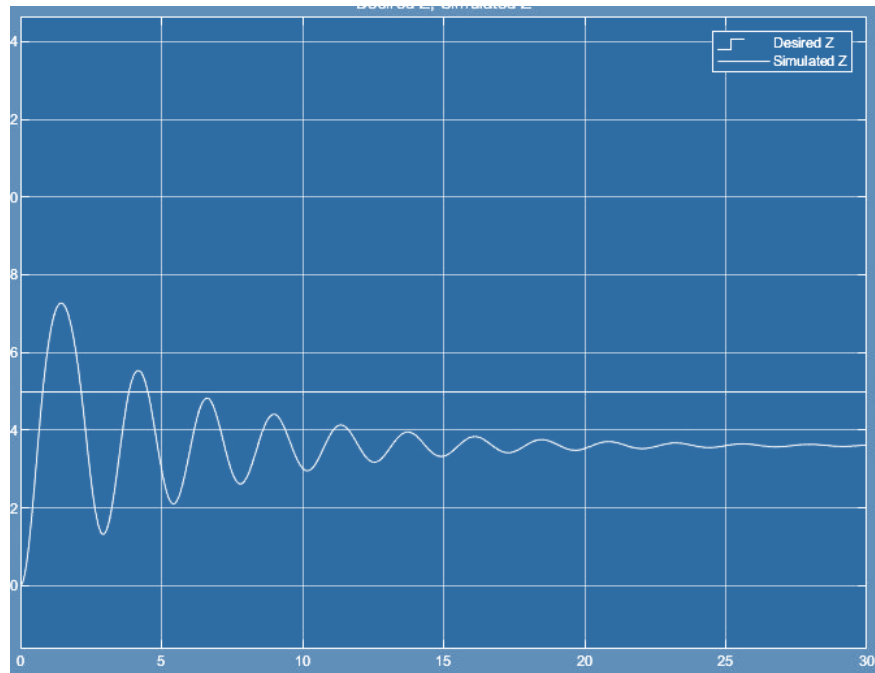


Рисунок А.1 - Налаштування параметру висоти Р

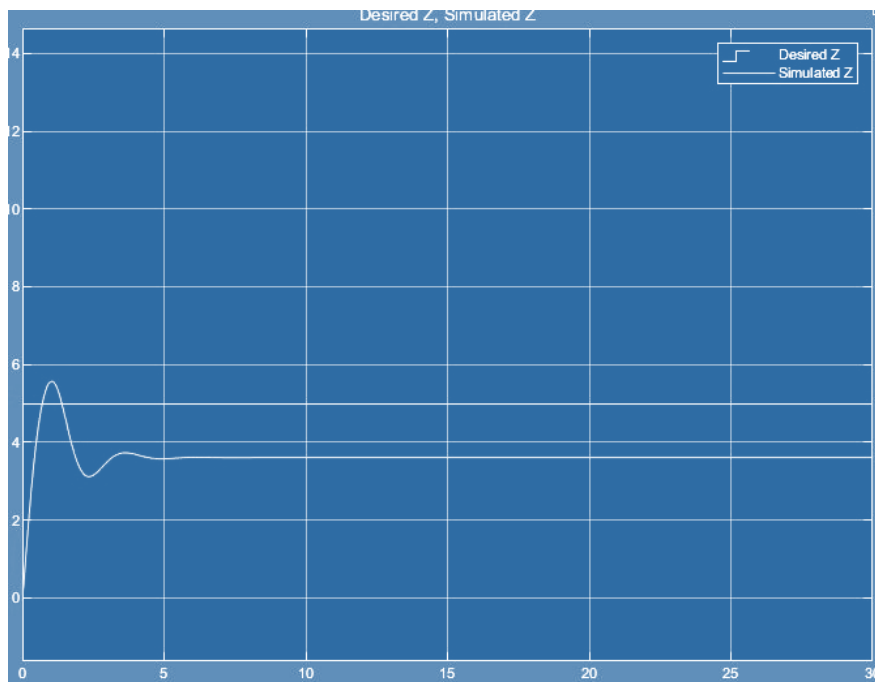


Рисунок А.2 - Налаштування значення збільшення висоти D

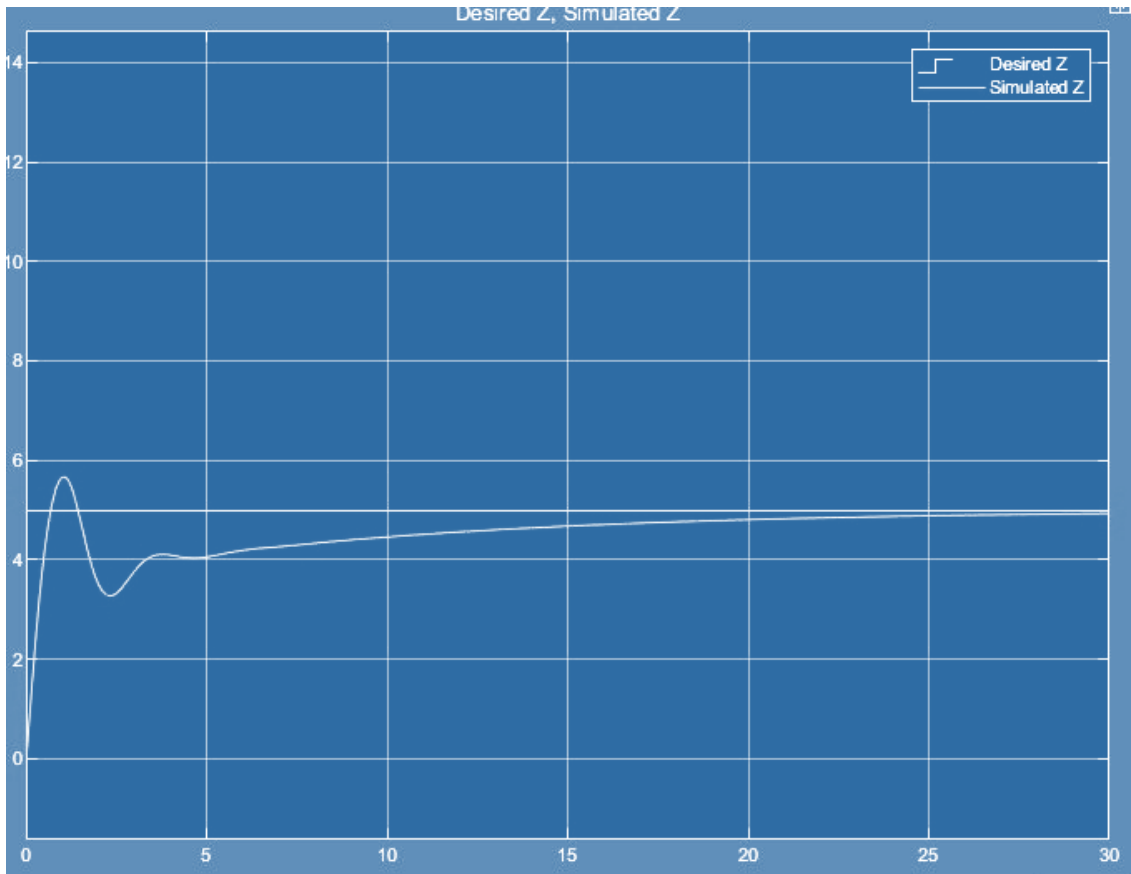


Рисунок А.3 - Налаштування значення збільшення висоти I

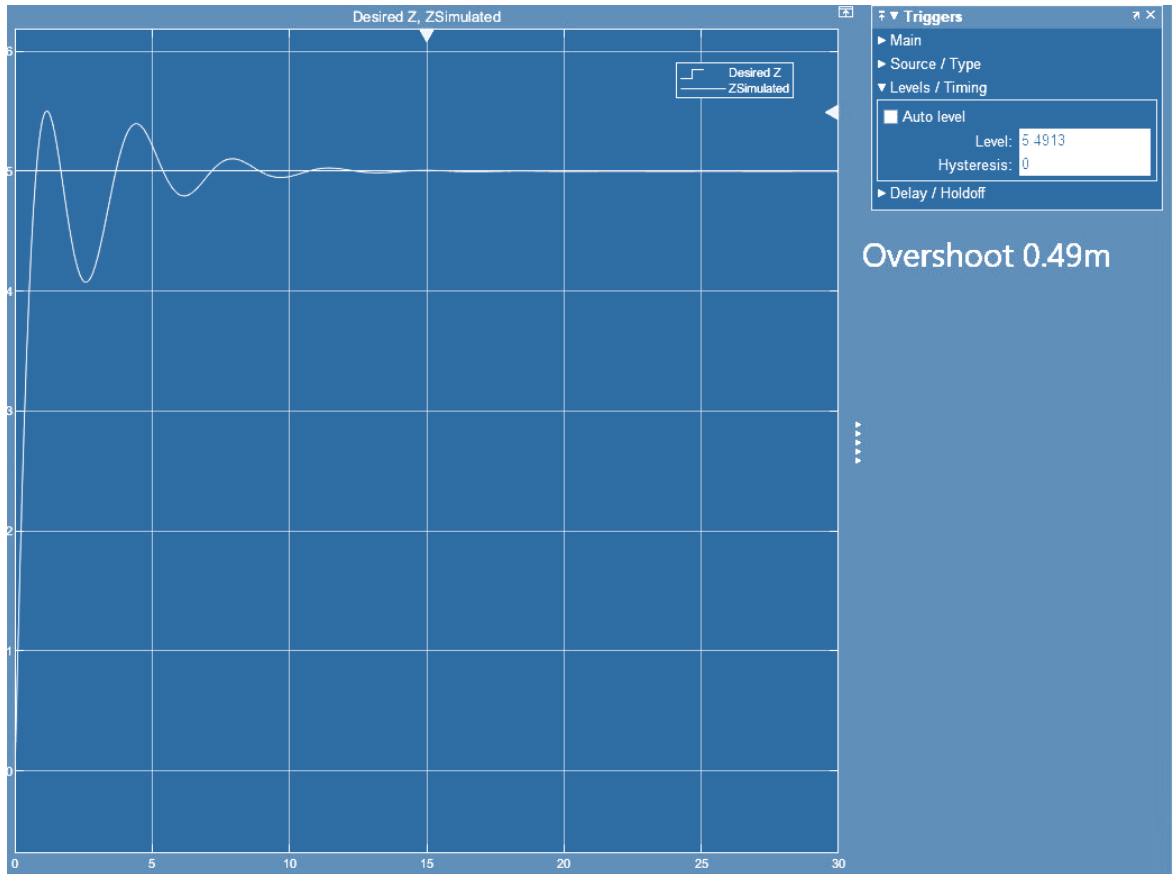


Рисунок А4 - Відповідь ПІД-контролера з налаштованою висотою

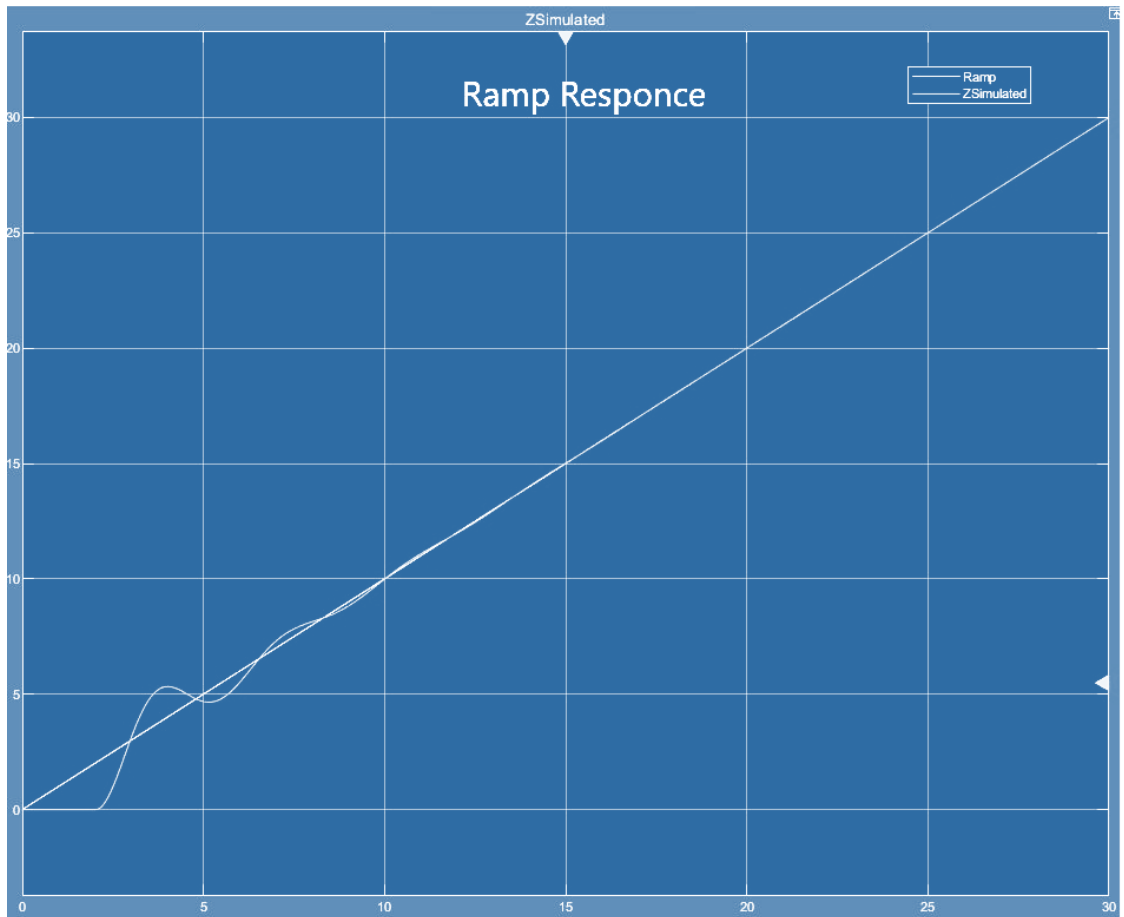


Рисунок А.5 - Поведінка БПЛА при одержанні команди набору висоти

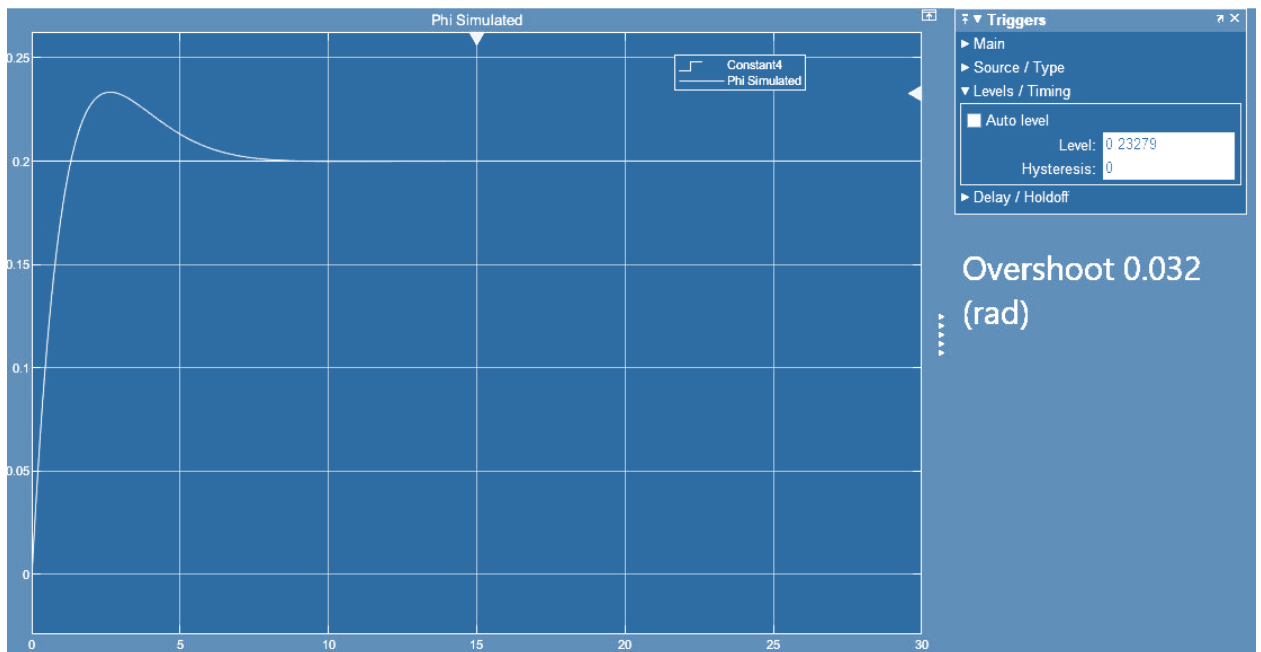


Рисунок А.6 -.Налаштована відповідь ПІД-контролерів обертання та нахилу

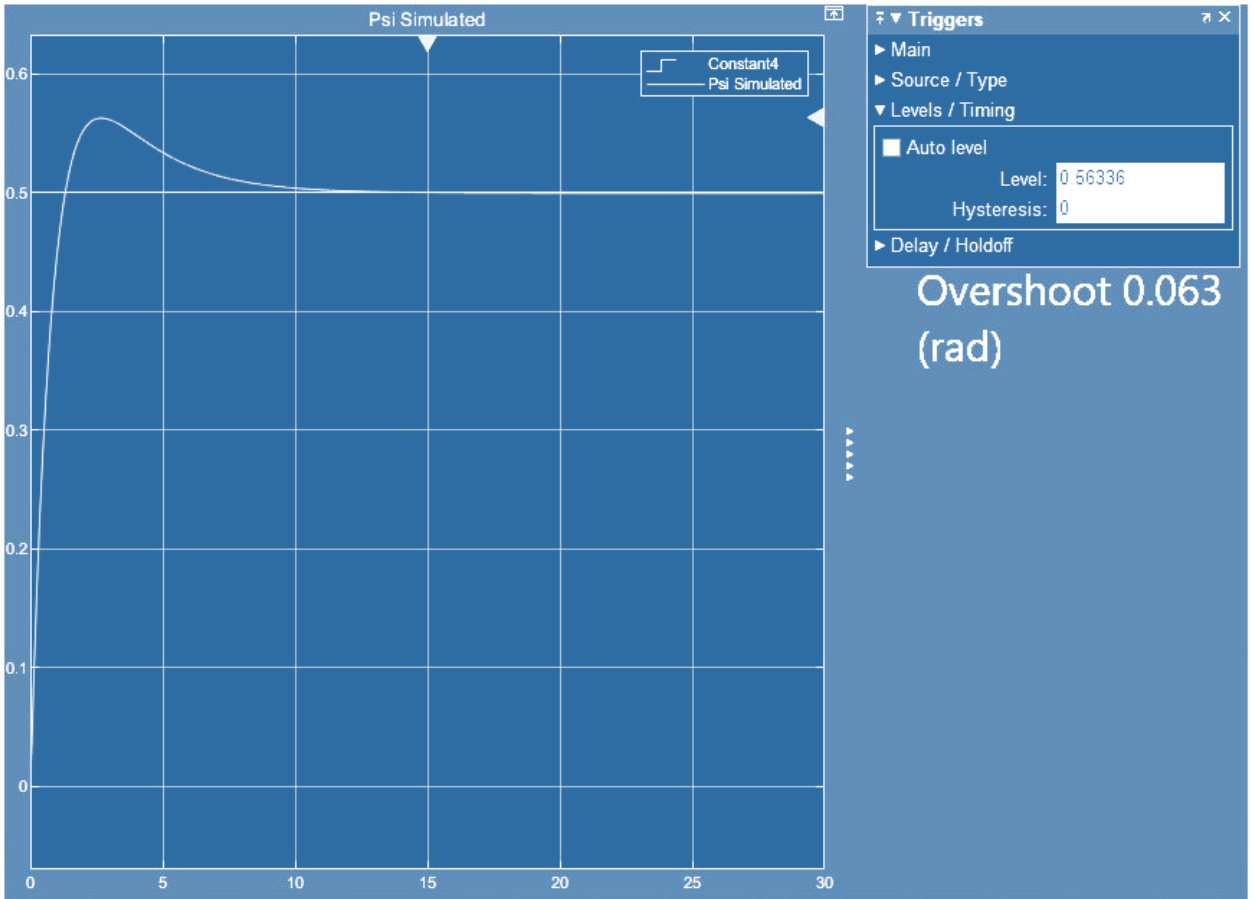
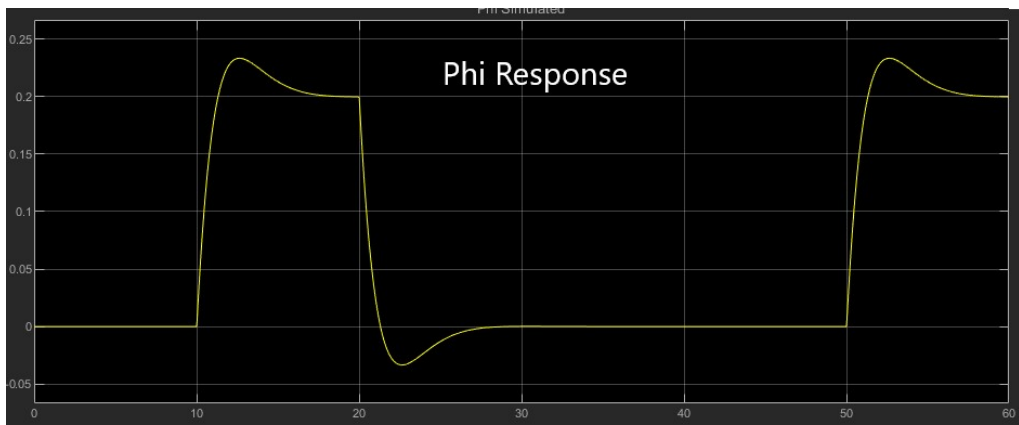
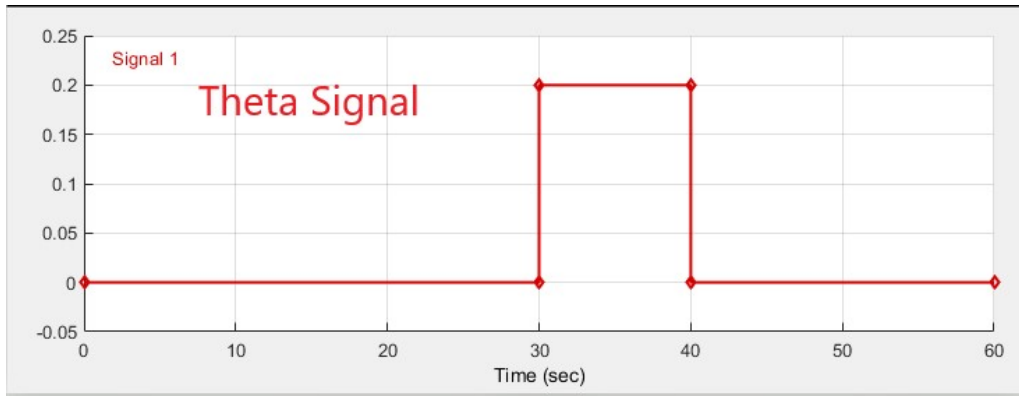


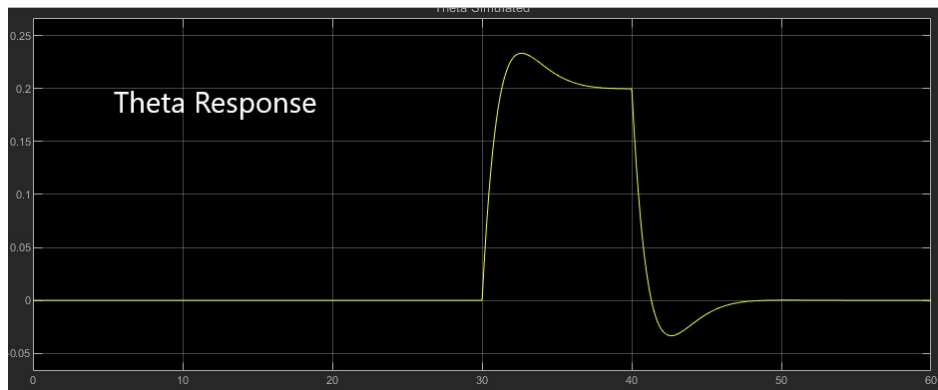
Рисунок А.7 - Налаштована відповідь контролера Yaw



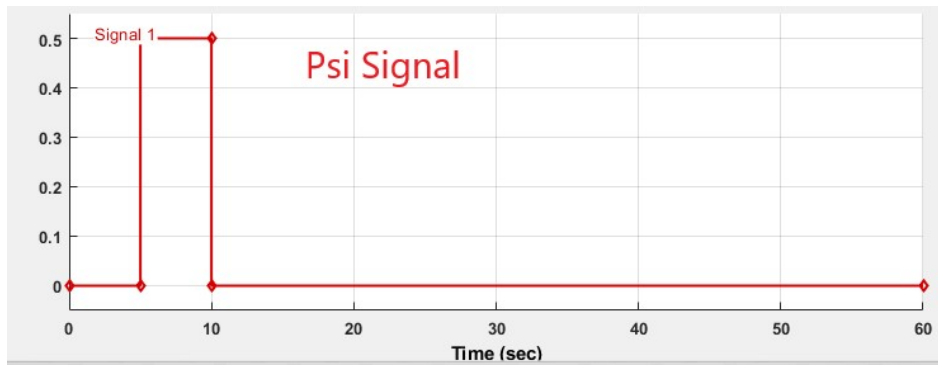
a)



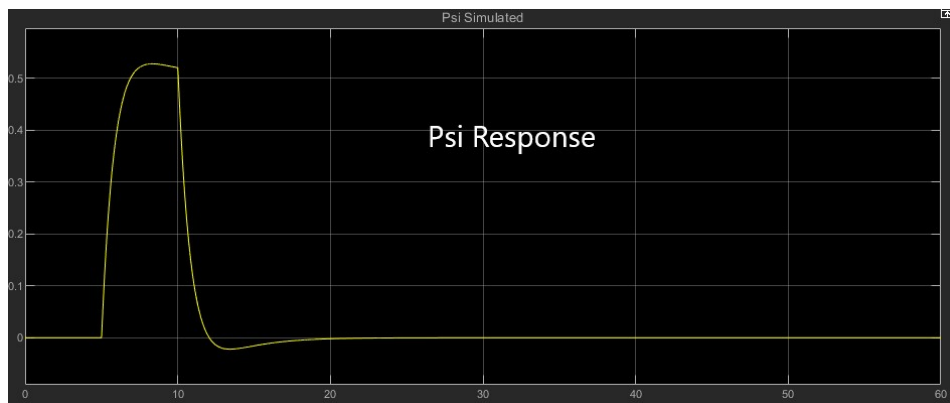
б)



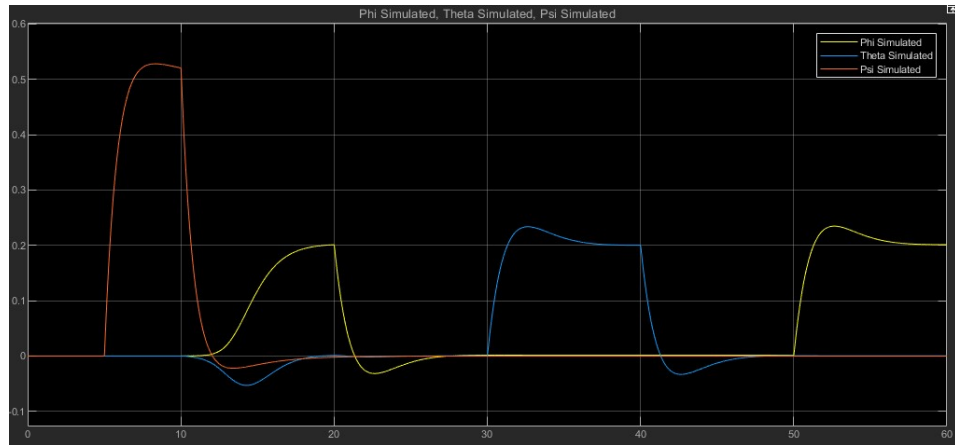
в)



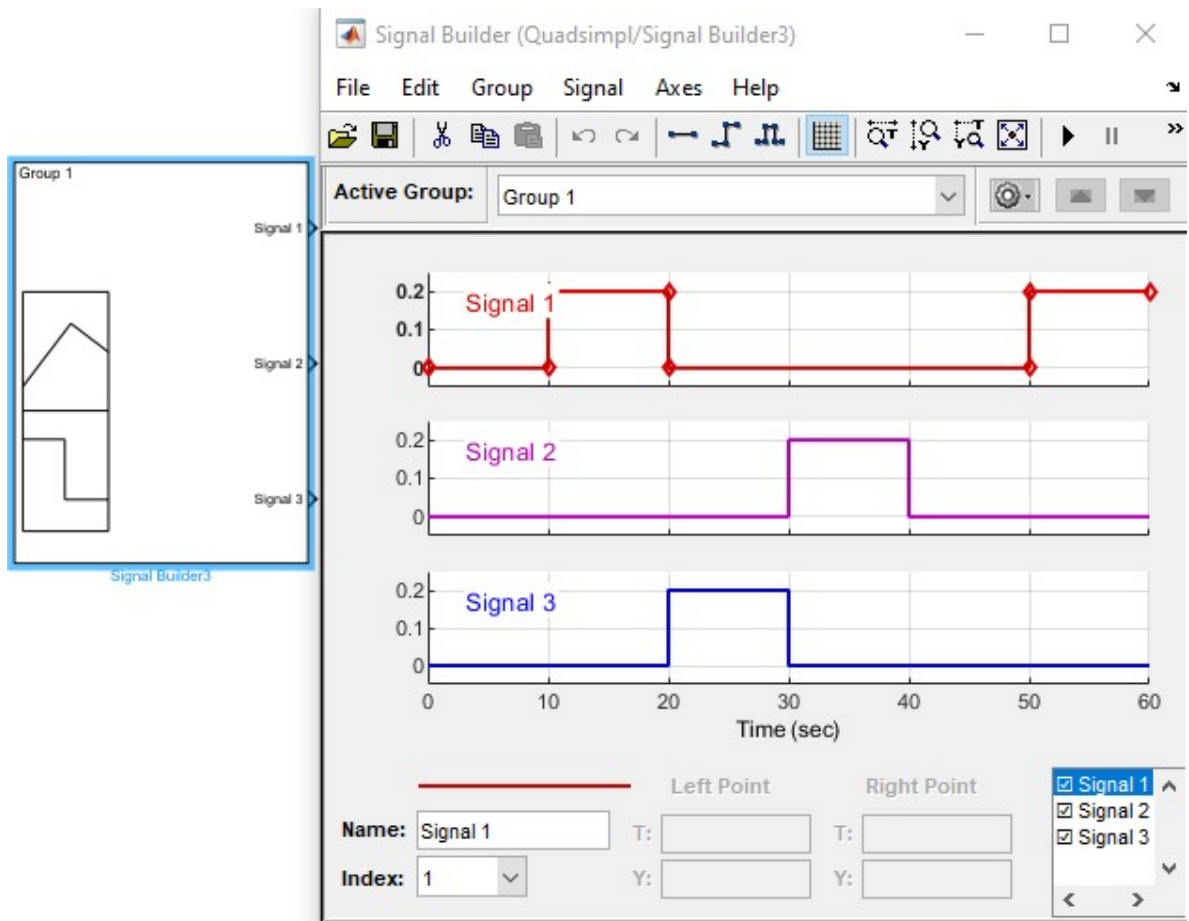
г)



д)



e)



ж)

Рисунок А.8 – Застосування генератора сигналів

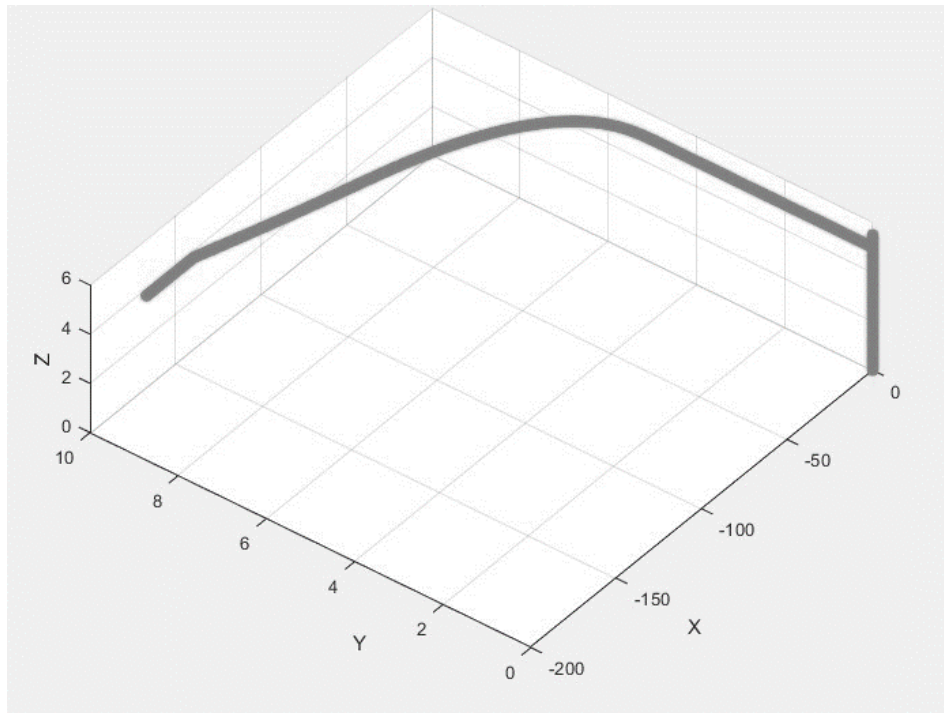


Рисунок А.9 - Тривимірна траєкторія польоту БПЛА

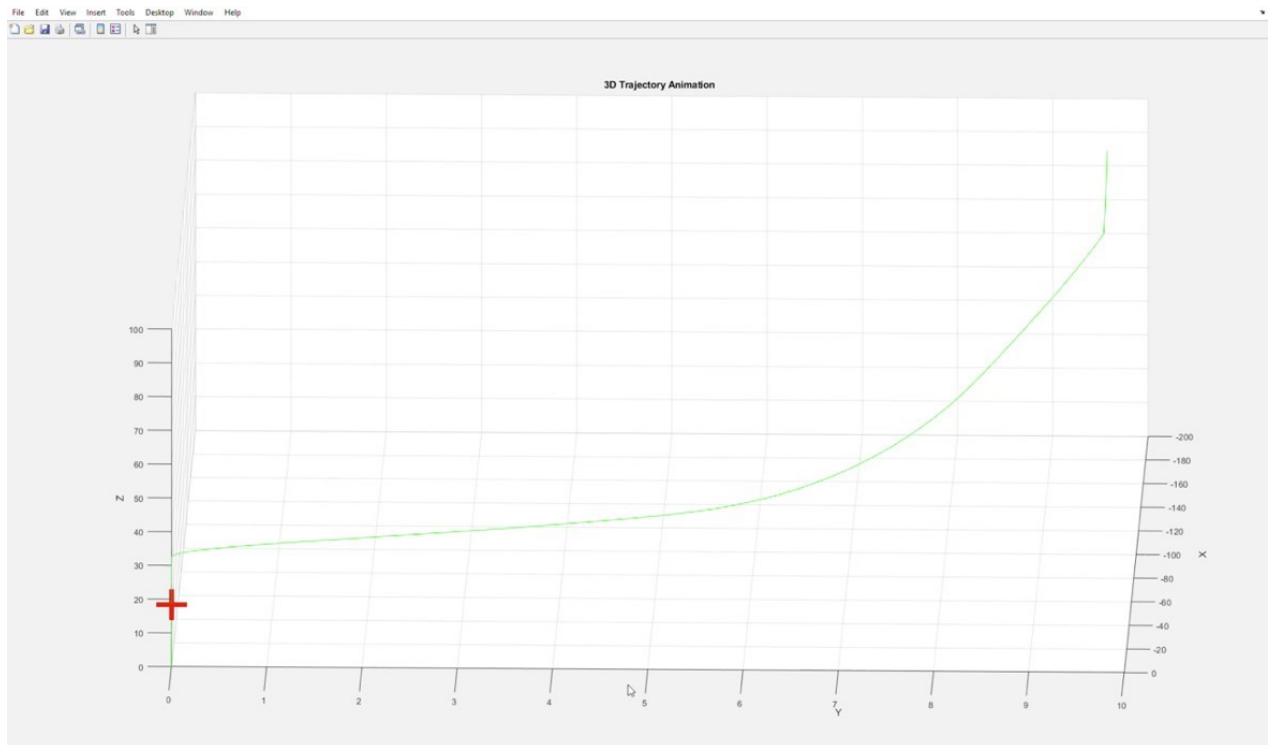


Рисунок А.10 - Анімаційне відео траєкторії

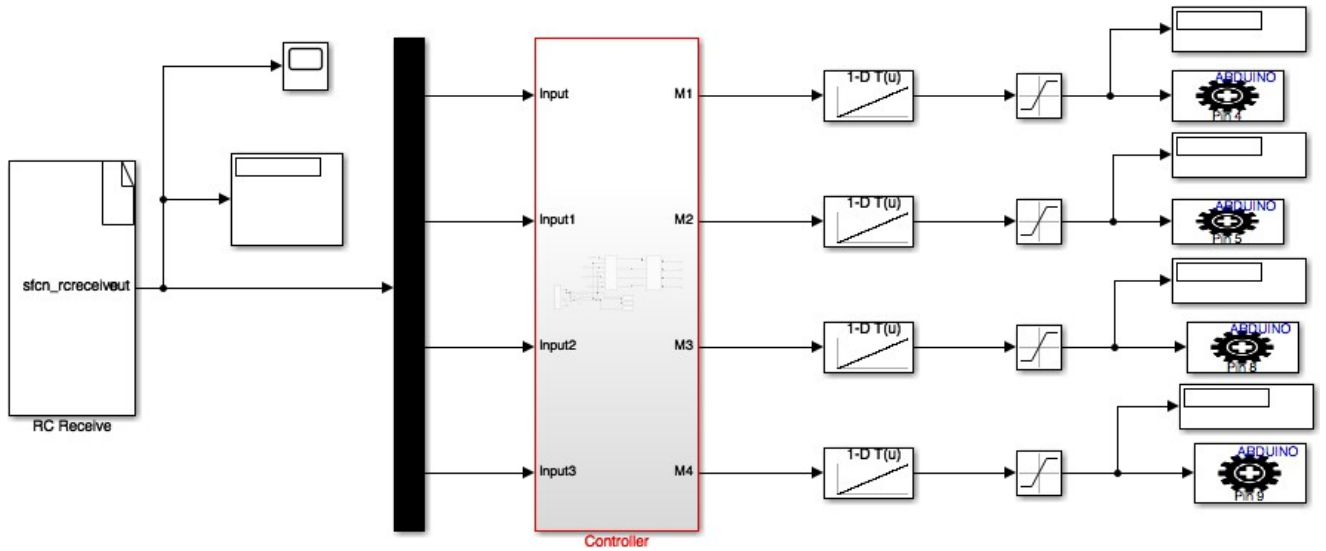


Рисунок А.11 - Модель реалізації

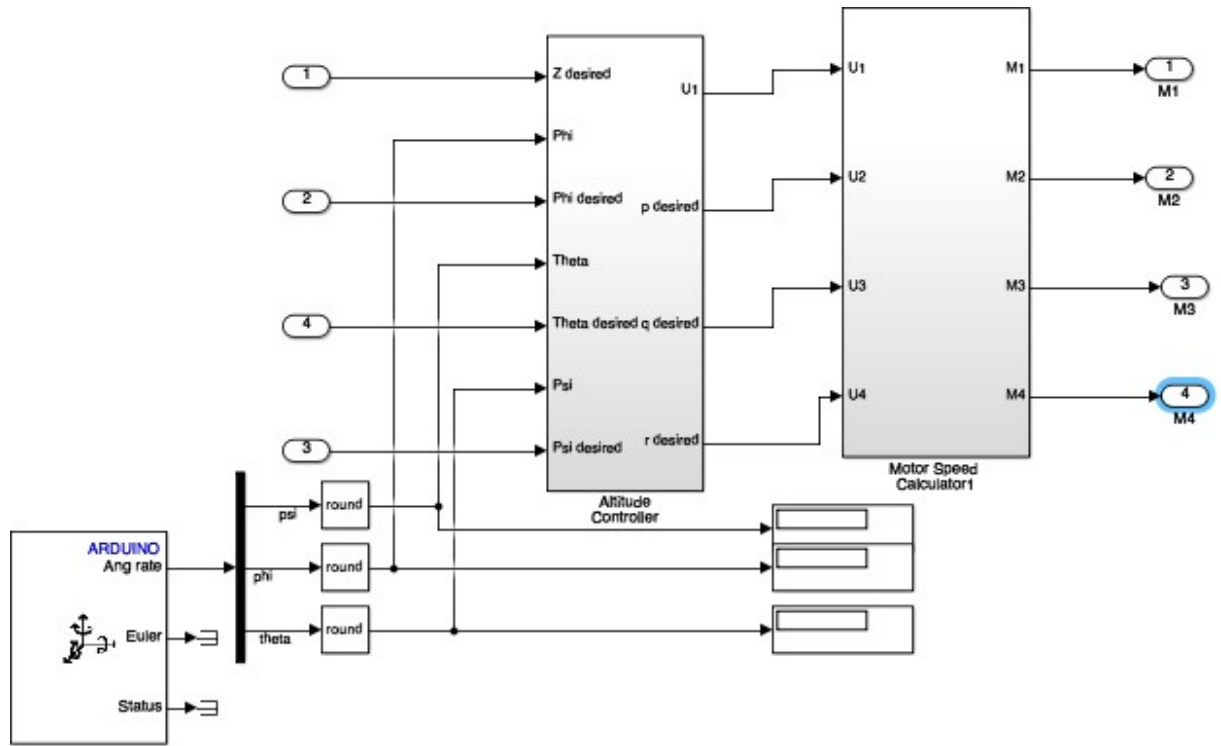


Рисунок А.12 - Внутрішній вигляд підсистеми контролера моделі реалізації

Додаток Б (обов'язковий)

Програмна реалізація нечіткого PID-контролера

```

#include "ardrone_velocity/controlnode.hpp"
#include <random>

ControlNode::ControlNode() {
    ros::NodeHandle params("~");
    // Topic Parameters
    std::string s;

    params.param<std::string>("cmd_vel_ref_topic", s, "/cmd_vel_ref");
    m_cmd_vel_sub = nh.subscribe(s, 1, &ControlNode::cmd_velCallback, this);

    params.param<std::string>("odometry_topic", s, "/ardrone/odometry");
    m_quad_vel_sub = nh.subscribe(s, 1, &ControlNode::quad_odom_callback, this,
        ros::TransportHints().tcpNoDelay());

    params.param<std::string>("cmd_vel_out_topic", s, "/cmd_vel");
    m_cmd_vel_pub = nh.advertise<geometry_msgs::Twist>(s, 1);

    // Dynamic parameter reconfigure
    dynamic_reconfigure::Server<
        ardrone_velocity::dynamic_param_configConfig>::CallbackType f;
    f = boost::bind(&ControlNode::dynamic_reconfigure_callback, this, _1, _2);
    m_server.setCallback(f);

    m_debug_pub = nh.advertise<std_msgs::Float64>("/ardrone_velocity/debug", 1);
    m_i_term_x = 0.0;
    m_i_term_y = 0.0;
}

void ControlNode::cmd_velCallback(const geometry_msgs::Twist& cmd_vel_in) {
    m_current_command = cmd_vel_in; // Units in m/s
    m_cmd_valid = true;
}

void ControlNode::quad_odom_callback(const nav_msgs::Odometry& odo_msg) {
    std_msgs::Float64 debug_msg;
    m_odo_msg = odo_msg;
    m_filtered_vel_x = m_filter_vel_x.filter(m_odo_msg.twist.twist.linear.x);
    m_filtered_vel_y = m_filter_vel_y.filter(m_odo_msg.twist.twist.linear.y);
    debug_msg.data = m_filtered_vel_x;
}

```

```

    m_debug_pub.publish(debug_msg);

    velocity_control();
}

void ControlNode::velocity_control(void) {
    double p_term_x, d_term_x;
    double p_term_y, d_term_y;

    double error_x;
    double error_y;

    geometry_msgs::Twist cmd_vel_out;

    // We limit the maximum reference speed of the quadcopter
    //! TODO: Change this into ROS parameters
    double max_vel = 0.6;
    m_current_command.linear.x = std::min(max_vel, m_current_command.linear.x);
    m_current_command.linear.y = std::min(max_vel, m_current_command.linear.y);

    m_current_command.linear.x = std::max(-max_vel, m_current_command.linear.x);
    m_current_command.linear.y = std::max(-max_vel, m_current_command.linear.y);

    // We are only going to change linear.x and linear.y of this command
    // The rest of the values are the same
    cmd_vel_out = m_current_command;

    // In case that we receive a special command to hover
    if (cmd_vel_out.angular.x == 0 && cmd_vel_out.angular.y == 0 &&
        cmd_vel_out.angular.z == 0 && cmd_vel_out.linear.x == 0 &&
        cmd_vel_out.linear.y == 0 && cmd_vel_out.linear.z == 0) {
        set_hover();
        // reset iterm
        m_i_term_x = 0.0;
        m_i_term_y = 0.0;
        return;
    }

    // otherwise we dont want to hover.
    cmd_vel_out.angular.x = 1;
    cmd_vel_out.angular.y = 1;

```

```

// Control starts here
//!TODO: Separate filter for input of the derivative term.
//!TODO: Consider measurement and controlled variables delays.

// We calculate the velocity error
error_x = m_current_command.linear.x - m_odo_msg.twist.twist.linear.x;
error_y = m_current_command.linear.y - m_odo_msg.twist.twist.linear.y;

// The proportional term is directly the error
p_term_x = error_x;
p_term_y = error_y;
// p_term_x = 0;
// p_term_y = 0;

// For derivative and integral we need the current time and timestep (dt)
t = ros::Time::now();
ros::Duration dt = t - old_t;
old_t = t;

// Derivative term (based on velocity change instead of error change)
// Note that we put the negative part here
// d_term_x = -(m_odo_msg.twist.twist.linear.x - m_last_vel_x)/dt.toSec();
// d_term_y = -(m_odo_msg.twist.twist.linear.y - m_last_vel_y)/dt.toSec();
d_term_x = -(m_filtered_vel_x - m_last_vel_x) / dt.toSec();
d_term_y = -(m_filtered_vel_y - m_last_vel_y) / dt.toSec();

//std_msgs::Float64 debug_msg;
// m_debug_pub.publish(debug_msg);

m_last_vel_x = m_filtered_vel_x;
m_last_vel_y = m_filtered_vel_y;

//! You can use this version as well but it will have some discontinuities
//! when the reference changes
// d_term_x = (error_x - m_last_error_x)/dt.toSec();
// d_term_y = (error_y - m_last_error_y)/dt.toSec();
// m_last_error_x = error_x;
// m_last_error_y = error_y;

//! Taken from tum_autonomy package.
//! This calculates and limits the integral term
// m_i_term is a member of the class
i_term_increase(m_i_term_x, error_x * dt.toSec(), 1.2);
i_term_increase(m_i_term_y, error_y * dt.toSec(), 1.2);

// m_i_term_x = m_i_term_x + error_x*dt.toSec();

```

```

// m_i_term_y = m_i_term_y + error_y; /*dt.toSec());

// Control command (PID)
cmd_vel_out.linear.x =
    m_Kp_xy * (p_term_x + m_Ki_xy * m_i_term_x + m_Kd_xy * d_term_x);
cmd_vel_out.linear.y =
    m_Kp_xy * (p_term_y + m_Ki_xy * m_i_term_y + m_Kd_xy * d_term_y);

// Limit control command to min max values of ardrone SDK (-1.0, 1.0)
cmd_vel_out.linear.x = std::min(cmd_vel_out.linear.x, 1.0);
cmd_vel_out.linear.y = std::min(cmd_vel_out.linear.y, 1.0);

cmd_vel_out.linear.x = std::max(cmd_vel_out.linear.x, -1.0);
cmd_vel_out.linear.y = std::max(cmd_vel_out.linear.y, -1.0);

// Debugging information
// ROS_INFO("d_Time : %f", dt.toSec());
// ROS_INFO("VelRef: %f", m_current_command.linear.x);
// ROS_INFO("Vel : %f", m_odo_msg.twist.twist.linear.x);
// ROS_INFO("Error : %f", error_x);
// ROS_INFO("Cmd : %f", cmd_vel_out.angular.z);
// ROS_INFO("pterm | iterm | dterm : %f | %f | %f", m_Kp_x*p_term_x,
// m_Kp_x*m_Ki_x*m_i_term_x, m_Kp_x*m_Kd_x*d_term_x);
// ROS_INFO("-----");

// We publish the command
m_cmd_vel_pub.publish(cmd_vel_out);
}

void ControlNode::set_hover(void) {
    geometry_msgs::Twist cmd_vel_out;
    ROS_INFO("Sending Hover command");
    cmd_vel_out.linear.x = 0;
    cmd_vel_out.linear.y = 0;
    cmd_vel_out.linear.z = 0;
    cmd_vel_out.angular.x = 0;
    cmd_vel_out.angular.y = 0;
    cmd_vel_out.angular.z = 0;

    m_cmd_vel_pub.publish(cmd_vel_out);
}

void ControlNode::i_term_increase(double& i_term, double new_err, double cap) {
    if (new_err < 0 && i_term > 0)
        i_term = std::max(0.0, i_term + 2.5 * new_err);
    else if (new_err > 0 && i_term < 0)

```

```

        i_term = std::min(0.0, i_term + 2.5 * new_err);
    else
        i_term += new_err;

    if (i_term > cap) i_term = cap;
    if (i_term < -cap) i_term = -cap;
}

void ControlNode::dynamic_reconfigure_callback(
    ardrone_velocity::dynamic_param_configConfig& config, uint32_t level) {
    ROS_INFO("Reconfigure Request: %f %f %f", config.Kp_xy, config.Ki_xy, config.Kd_xy);
    // Coefficients for the PID controller
    m_Kp_xy = config.Kp_xy;
    m_Ki_xy = config.Ki_xy;
    m_Kd_xy = config.Kd_xy;
}

#include "ardrone_velocity/filtervelocity.hpp"

FilterVelocity::FilterVelocity()
    : m_size_filter(30.0),
    m_coeffs({ 0.00498902, 0.00567655, 0.00768429, 0.01092849, 0.01526534,
              0.02049766, 0.02638413, 0.03265082, 0.03900430, 0.04514569,
              0.05078516, 0.05565588, 0.05952694, 0.06221459, 0.06359114,
              0.06359114, 0.06221459, 0.05952694, 0.05565588, 0.05078516,
              0.04514569, 0.03900430, 0.03265082, 0.02638413, 0.02049766,
              0.01526534, 0.01092849, 0.00768429, 0.00567655, 0.00498902 }) {
    m_input_buffer = std::vector<double>(m_size_filter, 0.0);
}

double FilterVelocity::filter(double new_value) {
    double result = 0.0;
    int i;
    m_input_buffer[0] = new_value;

    for (i = 0; i < m_size_filter; i++) {
        result += m_input_buffer[i] * m_coeffs[i];
    }

    for (i = m_size_filter - 1; i > 0; i--) {
        m_input_buffer[i] = m_input_buffer[i - 1];
    }
    return result;
}

```

```

#include "ardrone_velocity/controlnode.hpp"
#include <sstream>
#include <stdio.h>
#include <stdlib.h>
#include <ros/ros.h>
#include "std_msgs/Float32.h"
#include "geometry_msgs/Twist.h"
#include "sensor_msgs/Joy.h"
#include "std_msgs/Empty.h"

int main(int argc, char* argv[]) {
    ros::init(argc, argv, "pid_control"); // Name of the node
    ControlNode Node;

    int32_t looprate = 2000; //hz
    ros::Rate loop_rate(looprate);

    // ros::spin();
    while (Node.nh.ok()) {
        ros::spinOnce();
        loop_rate.sleep();
    }
} #include <sstream>
#include <stdio.h>
#include <stdlib.h>
#include <ros/ros.h>
#include "std_msgs/Float32.h"
#include "geometry_msgs/Twist.h"
#include "sensor_msgs/Joy.h"
#include "std_msgs/Empty.h"
#include "ardrone_autonomy/Navdata.h"

double height = 2;
double velx = 0;
double vely = 0;
double angz = 0;
bool buttonTakeoff = false;
bool buttonLand = false;
bool ardroneFlying = false;
bool buttonEnable = false;
bool last_buttonEnable = false;
std_msgs::Empty empty;
ros::Subscriber cmd_vel_sub, navdata_sub;
ros::Publisher twist_pub, takeoff_pub, land_pub;
ros::Timer timer1;

```

```

ros::Timer timer2;
ros::NodeHandle* n_p;
ardrone_autonomy::Navdata last_navdata;

enum class State {
    ground_init,
    takingoff,
    hovering_init,
    flying,
    hovering_end,
    landing,
    ground_end
};

enum class NavdataState : std::int32_t {
    unknown,
    init,
    landed,
    flying,
    hovering,
    test,
    takingoff,
    gotofixpoint,
    landing,
    looping
};

State quad_state = State::ground_init;
NavdataState navdata_state = NavdataState::unknown;

void set_hover(void) {
    geometry_msgs::Twist cmd_vel_out;
    cmd_vel_out.linear.x = 0;
    cmd_vel_out.linear.y = 0;
    cmd_vel_out.linear.z = 0;
    cmd_vel_out.angular.x = 0;
    cmd_vel_out.angular.y = 0;
    cmd_vel_out.angular.z = 0;
    twist_pub.publish(cmd_vel_out);
}

void land_or_takeoff(void) {
    if (ardroneFlying) {
        ardroneFlying = false;
    }
    else {

```

```

        ardroneFlying = true;
    }
}

void takeoff(void) { takeoff_pub.publish(empty); }

void land(void) { land_pub.publish(empty); }

void joyCallback(const sensor_msgs::Joy& in) {
    geometry_msgs::Twist out_twist;
    velx = double(in.axes[1]);
    vely = double(in.axes[0]);
    height = double(in.axes[2]) * 3;
    angz = double(in.axes[3]);
    buttonTakeoff = bool(in.buttons[2]);
    buttonLand = bool(in.buttons[1]);
    buttonEnable = bool(in.buttons[0]);
    if (buttonTakeoff) {
        takeoff();
    }
    if (buttonLand) {
        land();
    }

    out_twist.linear.x = velx;
    out_twist.linear.y = vely;
    out_twist.linear.z = height;
    out_twist.angular.z = angz;

    if (buttonEnable) {
        // To Disable auto hover
        out_twist.angular.x = 1.0;
        out_twist.angular.y = 1.0;
    }
    else {
        out_twist.angular.x = 0.0;
        out_twist.angular.y = 0.0;
    }

    twist_pub.publish(out_twist);
}

void stopFlying(const ros::TimerEvent&) {
    ROS_INFO("Flying end, sending command to hover");
    quad_state = State::hovering_end;
    set_hover();
}

```

```

}

void startFlying(const ros::TimerEvent& {
    ROS_INFO("Flying Begin (10 seconds)");
    quad_state = State::flying;
    timer2 = n_p->createTimer(ros::Duration(10), stopFlying, true);
}

void navdataCallback(const ardrone_autonomy::Navdata navdata) {
    last_navdata = navdata;
}

int main(int argc, char* argv[]) {
    ros::init(argc, argv, "test_controller"); // Name of the node
    ros::NodeHandle n;

    n_p = &n;
    cmd_vel_sub = n.subscribe("/joy", 1, joyCallback);
    navdata_sub = n.subscribe("/ardrone/navdata", 1, navdataCallback);
    twist_pub = n.advertise<geometry_msgs::Twist>("/cmd_vel_pid", 1);
    takeoff_pub = n.advertise<std_msgs::Empty>("/ardrone/takeoff", 1);
    land_pub = n.advertise<std_msgs::Empty>("/ardrone/land", 1);

    ros::Rate rate(5); // 10 hz
    geometry_msgs::Twist out_twist;
    out_twist.linear.x = 0.0;
    out_twist.linear.y = 0.0;
    out_twist.linear.z = 0.0;
    out_twist.angular.z = 0.0;
    out_twist.angular.x = 0.0;
    out_twist.angular.y = 0.0;

    while (n.ok()) {
        ros::spinOnce();
        if (quad_state == State::ground_init) {
            ROS_INFO("Init Test.");
            ROS_INFO("Sending command to takeoff.");

            quad_state = State::takingoff;
        }
        else if (quad_state == State::takingoff) {
            // checkHover();
            if (last_navdata.state == (int32_t)NavdataState::landed) {
                ROS_INFO("FORCING command to takeoff.");
                ROS_INFO("last_navdata.state %d", last_navdata.state);
                takeoff();
            }
        }
    }
}

```

```

    }
    else if (last_navdata.state == (int32_t)NavdataState::init) {
        ROS_INFO("FORCING command to takeoff.");
        ROS_INFO("last_navdata.state %d", last_navdata.state);
        takeoff();
    }
    else if (last_navdata.state == (int32_t)NavdataState::hovering) {
        quad_state = State::hovering_init;
        ROS_INFO("Take off complete, waiting 0.1 seconds.");
        timer1 = n.createTimer(ros::Duration(0.1), startFlying, true);
    }
}
else if (quad_state == State::hovering_init) {
}
else if (quad_state == State::flying) {
    // send_command();
    //
    //
    twist_pub.publish(out_twist);
    out_twist.linear.x = 0.2;
    out_twist.linear.y = 0.0;
    out_twist.linear.z = 0.0;
    out_twist.angular.z = 0.0;
    out_twist.angular.x = 1.0;
    out_twist.angular.y = 1.0;
}
else if (quad_state == State::hovering_end) {
    ROS_INFO("last_navdata.state %d", last_navdata.state);
    if (last_navdata.state == (int32_t)NavdataState::hovering) {
        ROS_INFO("Hovering, now sending command to land");
        land();
        quad_state = State::landing;
    }
}
else if (quad_state == State::landing) {
    // Check for landing
    if (last_navdata.state == (int32_t)NavdataState::gotofixpoint) {
        // OK do nothing
    }
    else if (last_navdata.state == (int32_t)NavdataState::landed) {
        ROS_INFO("Quadcopter on the ground");
        quad_state = State::ground_end;
    }
    else {
        land();
        ROS_INFO("FORCING LAND");
    }
}

```

```

    }
    else if (quad_state == State::ground_end) {
        // Check for landing
        ROS_INFO("Test Succesfull");
        ros::shutdown();
    }

    rate.sleep(); // exit
}
}
#endif CONTROLNODE_HPP
#define CONTROLNODE_HPP

#include "ros/ros.h"
#include "geometry_msgs/Twist.h"
#include "nav_msgs/Odometry.h"
#include "std_msgs/Float64.h"
#include "ardrone_velocity/filtervelocity.hpp"

#include <dynamic_reconfigure/server.h>
#include <ardrone_velocity/dynamic_param_configConfig.h>

class ControlNode {
public:
    ControlNode();

    ros::NodeHandle nh;

    // ROS message callbacks
    void cmd_velCallback(const geometry_msgs::Twist& cmd_vel_in);
    void quad_odom_callback(const nav_msgs::Odometry& odo_msg);
    void dynamic_reconfigure_callback(
        ardrone_velocity::dynamic_param_configConfig& config, uint32_t level);

    // PID Controller
    void velocity_control(void);

    // Taken from tum_ardrone
    void i_term_increase(double& i_term, double new_err, double cap);

    void set_hover(void);

    bool m_cmd_valid = false;

    // Filter for quadcopter velocities
    FilterVelocity m_filter_vel_x;
    FilterVelocity m_filter_vel_y;
    // double filter_velocity(double new_value_x,double new_value_y);

private:
    ros::Subscriber m_cmd_vel_sub;
    ros::Subscriber m_quad_vel_sub;
    ros::Publisher m_cmd_vel_pub;
    ros::Publisher m_debug_pub; //! For debugging variables in rqt_plot
    // dynamic reconfigure server
    dynamic_reconfigure::Server<ardrone_velocity::dynamic_param_configConfig>
        m_server;

```

```

geometry_msgs::Twist m_current_command;
nav_msgs::Odometry m_odo_msg;
ros::Time t;
ros::Time old_t;
double m_filtered_vel_x;
double m_filtered_vel_y;
double m_last_error_x;
double m_last_error_y;
double m_last_vel_x;
double m_last_vel_y;
double m_i_term_x;
double m_i_term_y;
// PID Coefficients
double m_Kp_xy, m_Ki_xy, m_Kd_xy;
};

#endif // CONTROLNODE_HPP

#ifndef FILTERVELOCITY_HPP
#define FILTERVELOCITY_HPP
#include <vector>
class FilterVelocity {
public:
    FilterVelocity();
    double filter(double new_value);

private:
    double m_size_filter;
    std::vector<double> m_coeffs;
    std::vector<double> m_input_buffer;
};

#endif // FILTERVELOCITY_HPP
#include<iostream>
#include"fuzzy_PID.h"

#define NB -3
#define NM -2
#define NS -1
#define ZO 0
#define PS 1
#define PM 2
#define PB 3

int main()
{
    float target=600;
    float actual=0;
    float u=0;
    int deltaKpMatrix[7][7]={{PB,PB,PM,PM,PS,ZO,ZO},
                             {PB,PB,PM,PS,PS,ZO,NS},
                             {PM,PM,PM,PS,ZO,NS,NS},
                             {PM,PM,PS,ZO,NS,NM,NM},
                             {PS,PS,ZO,NS,NS,NM,NM},
                             {PS,ZO,NS,NM,NM,NM,NB},
                             {ZO,ZO,NM,NM,NM,NB,NB}};
    int deltaKiMatrix[7][7]={{NB,NB,NM,NM,NS,ZO,ZO},

```

```

        {NB,NB,NM,NS,NS,ZO,ZO},
        {NB,NM,NS,NS,ZO,PS,PS},
        {NM,NM,NS,ZO,PS,PM,PM},
        {NM,NS,ZO,PS,PS,PM,PB},
        {ZO,ZO,PS,PS,PM,PB,PB},
        {ZO,ZO,PS,PM,PM,PB,PB}};
int deltaKdMatrix[7][7]={{PS,NS,NB,NB,NB,NM,PS},
        {PS,NS,NB,NM,NM,NS,ZO},
        {ZO,NS,NM,NM,NS,NS,ZO},
        {ZO,NS,NS,NS,NS,NS,ZO},
        {ZO,ZO,ZO,ZO,ZO,ZO,ZO},
        {PB,NS,PS,PS,PS,PS,PB},
        {PB,PM,PM,PM,PS,PS,PB}};
float e_mf_paras[]={-3,-3,-2,-3,-2,-1,-2,-1,0,-1,0,1,0,1,2,1,2,3,2,3,3};
float de_mf_paras[]={-3,-3,-2,-3,-2,-1,-2,-1,0,-1,0,1,0,1,2,1,2,3,2,3,3};
float Kp_mf_paras[]={-3,-3,-2,-3,-2,-1,-2,-1,0,-1,0,1,0,1,2,1,2,3,2,3,3};
float Ki_mf_paras[]={-3,-3,-2,-3,-2,-1,-2,-1,0,-1,0,1,0,1,2,1,2,3,2,3,3};
float Kd_mf_paras[]={-3,-3,-2,-3,-2,-1,-2,-1,0,-1,0,1,0,1,2,1,2,3,2,3,3};
FuzzyPID fuzzypid(1500,650,0.3,0.4,0.2,0.02,0.65,0.005);
fuzzypid.setMf("trimf",e_mf_paras,"trimf",de_mf_paras,"trimf",Kp_mf_paras,"trimf",Ki_mf_paras,"trimf",Kd_mf_paras);
fuzzypid.setRuleMatrix(deltaKpMatrix,deltaKiMatrix,deltaKdMatrix);
cout<<"num target  actual"<<endl;
/*fuzzy.showInfo();*/
for(int i=0;i<200;i++)
{
    u=fuzzypid.realize(target,actual);
    actual+=u;
    cout<<i<<" "<<target<<" "<<actual<<endl;
}
fuzzypid.showInfo();
system("pause");
return 0;
}

#ifndef FUZZY_PID_H_
#define FUZZY_PID_H_
#include<iostream>
#include<string>
using std::string;
using std::cout;
using std::cin;
using std::endl;
class FuzzyPID
{
public:
    const static int N=7;
private:
    float target;
    float actual;
    float e;
    float e_pre_1;
    float e_pre_2;
    float de;
    float emax;
    float demax;
    float delta_Kp_max;

```

```

float delta_Ki_max;
float delta_Kd_max;
float Ke;    //[-3,-2,-1,0,1,2,3]
float Kde;  //[-3,-2,-1,0,1,2,3]
float Ku_p; //[-3,-2,-1,0,1,2,3]
float Ku_i; //[-3,-2,-1,0,1,2,3]
float Ku_d; //[-3,-2,-1,0,1,2,3]
int Kp_rule_matrix[N][N];
int Ki_rule_matrix[N][N];
int Kd_rule_matrix[N][N];
string mf_t_e;
string mf_t_de;
string mf_t_Kp;
string mf_t_Ki;
string mf_t_Kd;
float *e_mf_paras;
float *de_mf_paras;
float *Kp_mf_paras;
float *Ki_mf_paras;
float *Kd_mf_paras;
float Kp;
float Ki;
float Kd;
float A;
float B;
float C;
void showMf(const string & type,float *mf_paras);
void setMf_sub(const string & type,float *paras,int n);
public:
    FuzzyPID(float e_max,float de_max,float kp_max,float ki_max,float kd_max,float Kp0,float
    Ki0,float Kd0);
    FuzzyPID(float *fuzzyLimit,float *pidInitVal);
    ~FuzzyPID();
    float trimf(float x,float a,float b,float c);
    float gaussmf(float x,float ave,float sigma);
    float trapmf(float x,float a,float b,float c,float d);
    void setMf(const string & mf_type_e,float *e_mf,
              const string & mf_type_de,float *de_mf,
              const string & mf_type_Kp,float *Kp_mf,
              const string & mf_type_Ki,float *Ki_mf,
              const string & mf_type_Kd,float *Kd_mf);
    void setRuleMatrix(int kp_m[N][N],int ki_m[N][N],int kd_m[N][N]);
    float realize(float t,float a);
    void showInfo();
};

#endif

```

Додаток В
(обов'язковий)

Презентація доповіді

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Кафедра комп'ютерної інженерії та системного програмування

Румянцев Сергій Володимирович

Інтелектуалізована система керування безпілотними літальними апаратами

Науковий керівник – д.т.н. доц. Лисенко С.М.

Хмельницький - 2021

1

Мета і задачі дослідження

Метою роботи є підвищення точності траєкторії руху безпілотного літального апарату типу дрон.

Об'єкт дослідження – процес керування безпілотними літальними апаратами типу дрон.

Предмет дослідження – дослідження є моделі, методи та інтелектуалізована система керування безпілотними літальними апаратами.

Мета і задачі дослідження

Поставлена мета досягається розв'язанням таких основних задач:

1. дослідити особливості задання траєкторії руху безпілотного літального апарату типу дрон;
2. проаналізувати сучасні задання траєкторії руху безпілотного літального апарату типу дрон;
3. розробити модель задання траєкторії руху безпілотного літального апарату типу дрон;
4. розробити метод керування безпілотними літальними апаратами, який на відміну від відомих застосовує компоненти штучного інтелекту, і який забезпечує підвищення точності траєкторії руху безпілотного літального апарату типу дрон;
5. розробити інтелектуалізовану систему керування безпілотними літальними апаратами, яка забезпечує підвищення точності траєкторії руху безпілотного літального апарату типу дрон.

Наукова новизна отриманих результатів

Наукова новизна отриманих результатів:

1. Удосконалено метод керування безпілотними літальними апаратами, який на відміну від відомих застосовує компоненти штучного інтелекту, і який забезпечує підвищення точності траєкторії руху безпілотного літального апарату типу дрон.

2. Набула подальшого розвитку інтелектуалізована система керування безпілотними літальними апаратами, яка забезпечує підвищення точності траєкторії руху безпілотного літального апарату типу дрон.

Практичне значення отриманих результатів

В результаті виконаного наукового дослідження буде розроблено інтелектуалізовану систему керування безпілотними літальними апаратами, яка забезпечуватиме задання траєкторії руху безпілотного літального апарату типу дрон з високою точністю

Актуальність дослідження

Застосування дронів:

- контроль переміщення людей, техніки, ідентифікація тварин (тегованих);
- логістика: переміщення вантажів, постачання (у тому числі медичне) до важкодоступних місцевостей;



Актуальність дослідження

Застосування дронів:

- сільське господарство: обробка культур пестицидами;
- дика природа: відстеження браконьєрів;
- розвідка: оптична, лазерна, телевізійна, радіаційна.



Недоліки відомих методів задання траєкторії дронів

В результаті дослідження було виявлено, що відомі методи планування траєкторії польоту безпілотних літальних апаратів:

- обмежена точність
- висока обчислювальна складність

Тому було зроблено висновок про необхідність розроблення нових рішень, які б вирішили виявлені недоліки відомих методів.

Одним з способів є залучення апарату штучного інтелекту, який би дозволив створити інтелектуалізовану систему управління БПЛА.

Інтелектуалізована система керування безпілотними літальними апаратами

Схематичне
представлення
інтелектуалізованої
системи керування
безпілотними
літальними апаратами



Інтелектуалізована система керування безпілотними літальними апаратами

Модель системи керування БПЛА

Представимо систему керування безпілотними літальними апаратами моделлю у вигляді кортежу параметрів:

$$F_{UAV} = \langle z, v_x, v_y, \theta, \varphi, \psi \rangle, \quad (3.1)$$

де z - висота від землі до БПЛА в повітрі (м);

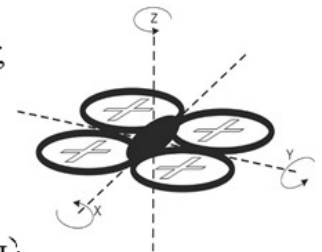
v_x - лінійна швидкість БПЛА в напрямку x , (м/с);

v_y - лінійна швидкість безпілота у напрямку y , (м/с);

θ - кут нахилу БПЛА по осі y , (рад);

φ - кут нахилу (крену) БПЛА по осі x , (рад);

ψ - кут повороту БПЛА навколо вертикальної осі z , (рад).



Інтелектуалізована система керування безпілотними літальними апаратами

Модель системи керування БПЛА

З цією метою подамо набір команди управління траєкторією руху БПЛА кортежем:

$$C_{UAV} = \langle \xi_z, \xi_\psi, \xi_\theta, \xi_\varphi \rangle, \quad (3.2)$$

де ξ_z - команда задання лінійної швидкості безпілотного літального апарату в напрямку по осі z;

ξ_ψ - команда задання кута повороту руху БПЛА навколо вертикальної осі z;

ξ_θ - команда задання висоти польоту БПЛА по осі y;

ξ_φ - команда задання кута нахилу (крену) БПЛА по осі x.

Інтелектуалізована система керування безпілотними літальними апаратами

Вхідні та вихідні параметри системи керування

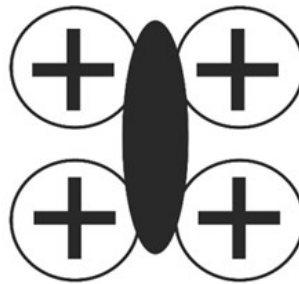
Команди задання руху
траєкторії БПЛА

$\xi_z \Rightarrow$

$\xi_\psi \Rightarrow$

$\xi_\theta \Rightarrow$

$\xi_\varphi \Rightarrow$



Множина параметрів
БПЛА

$\Rightarrow z$

$\Rightarrow v_x$

$\Rightarrow v_y$

$\Rightarrow \theta$

$\Rightarrow \varphi$

$\Rightarrow \psi$

Інтелектуалізована система керування безпілотними літальними апаратами

Нечітке логічне керування БПЛА

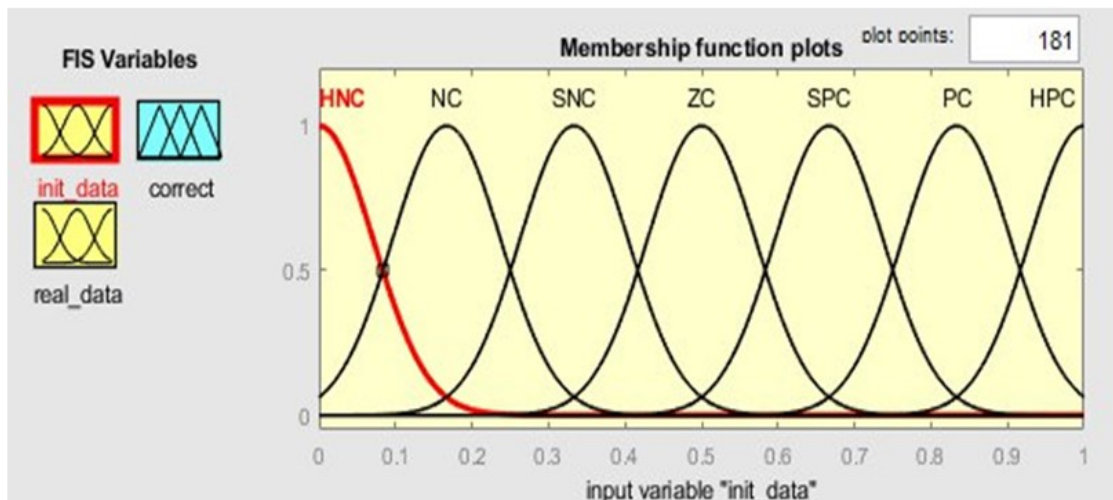
В основі інтелектуалізованої системи керування безпілотними літальними апаратами лежить побудова двох нечітких контролерів для продукції значень сигналів керування для:

ξ_{θ} - команда задання висоти польоту БПЛА по осі y ;

ξ_{φ} - команда задання кута нахилу (крену) БПЛА по осі x .

Інтелектуалізована система керування безпілотними літальними апаратами

Нечітке логічне керування БПЛА



Функції належності для команд управління ξ_θ та ξ_ϕ

Інтелектуалізована система керування безпілотними літальними апаратами

Нечітке логічне керування БПЛА

У якості вхідної та вихідної лінгвістичних змінних для нечіткого керування БПЛА X приймемо ім'я:

«Ступінь корегування траєкторії дрону».

Значення змінної буде визначатися на етапі його експлуатації в умовах зовнішніх впливів.

Для прийнятих лінгвістичних змінних визначимо терм-множини і функції належності.

Терм-множини значень керуючого сигналу БПЛА приймемо наступним чином:

1. «High negative correction (HNC)»;
2. «Negative correction (NC)»;
3. «Slight negative correction (SNC)»;
4. «Zero correction (ZC)»;
5. «Slight positive correction (SPC)»;
6. «Positive correction (PC)»;
7. «High positive correction (HPC)», з областю визначення $[0,1]$.

Інтелектуалізована система керування безпілотними літальними апаратами

Нечітке логічне керування БПЛА

Для генерації команд управління ξ_θ та ξ_φ відповідний нечіткий контролер має два входи, які, в свою чергу, оперують значеннями для формування траєкторії польоту в х- або у-напрямку:

- 1) початкові значення (σ_x, σ_y) ;
- 2) та скореговані під впливом зовнішніх чинників (σ'_x, σ'_y) .

Інтелектуалізована система керування безпілотними літальними апаратами керується множиною правила нечітких продукцій (ПНП).

Кожне із правил має деяку вагу $F=[0,1]$, що визначає значимість правила або впевненість у ступені істинності висновку, який отримується за окремим нечітким правилом.

Інтелектуалізована система керування безпілотними літальними апаратами

Нечітке логічне керування БПЛА

В дослідженні було залучено двадцять п'ять правил нечітких продукцій, приклади яких подано нижче:

П.1. If (σ_x, σ_y) is HNC and (σ'_x, σ'_y) is HNC then HPC (1);

П.2. If (σ_x, σ_y) is HNC and $((\sigma'_x, \sigma'_y)$ is NC then PC (1);

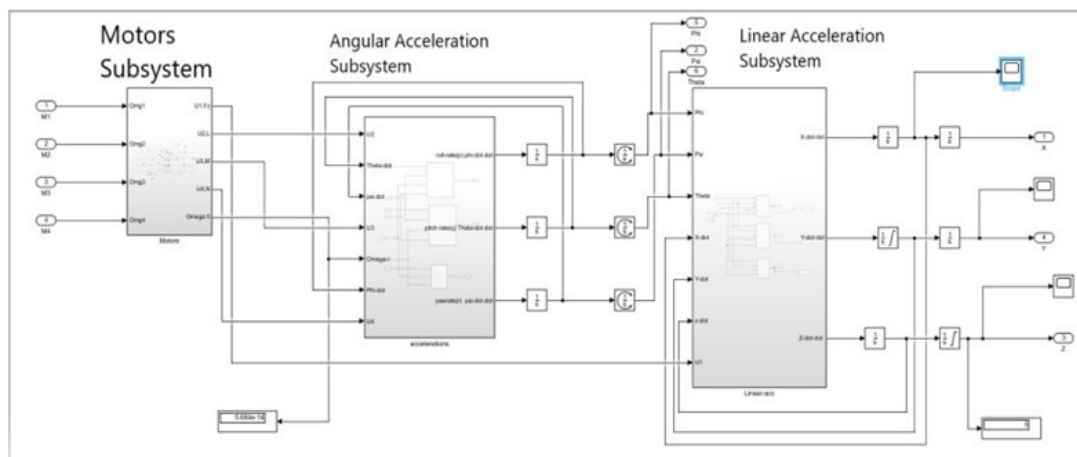
.....

П.25. If (σ_x, σ_y) is HPC and $((\sigma'_x, \sigma'_y)$ is HPC then HNC (1).

Реалізація інтелектуалізованої системи керування безпілотними літальними апаратами

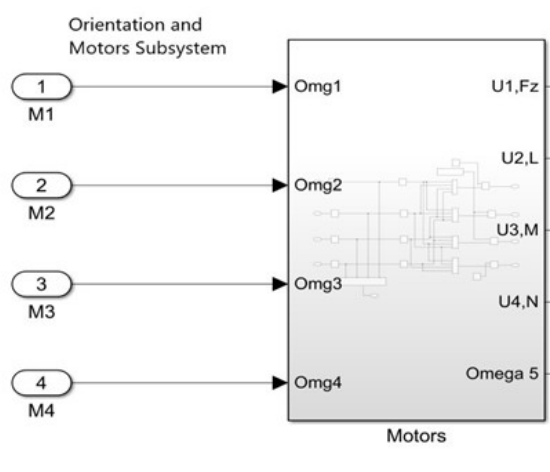
Застосування MATLAB / Simulink

Схема системи управління в реальному часі виконувалася на дроні AR.Drone 2.0 за допомогою динамічної моделі та контролерів, створених у середовищі MATLAB / Simulink



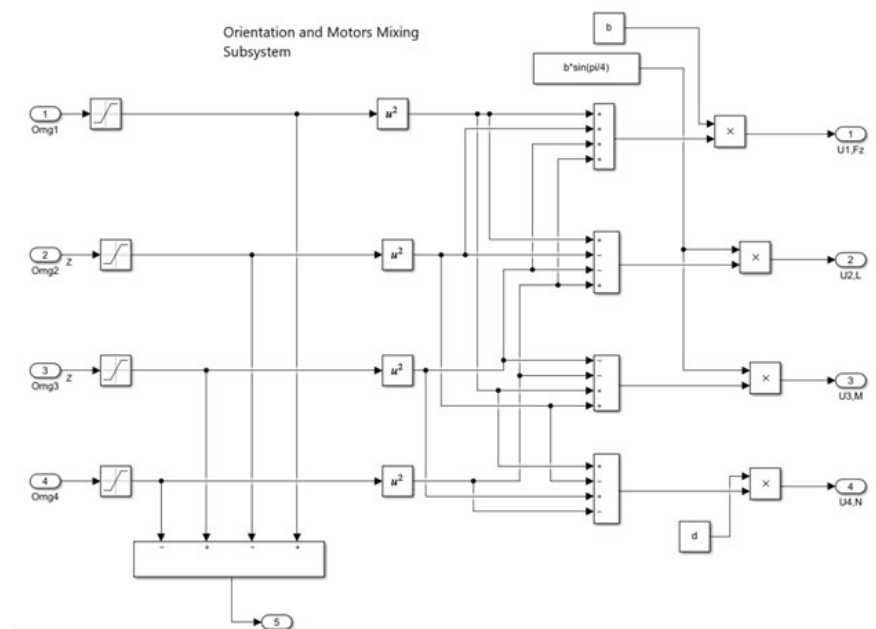
Реалізація інтелектуалізованої системи керування безпілотними літальними апаратами

Підсистема орієнтації та двигунів БПЛА



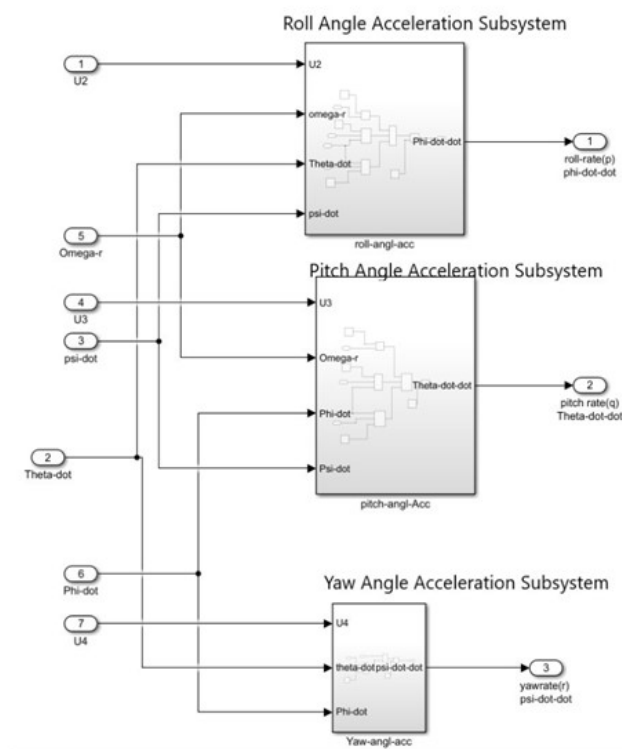
Реалізація інтелектуалізованої системи керування безпілотними літальними апаратами

Підсистема керування орієнтацією та керуванням двигунами в розгорнутому вигляді



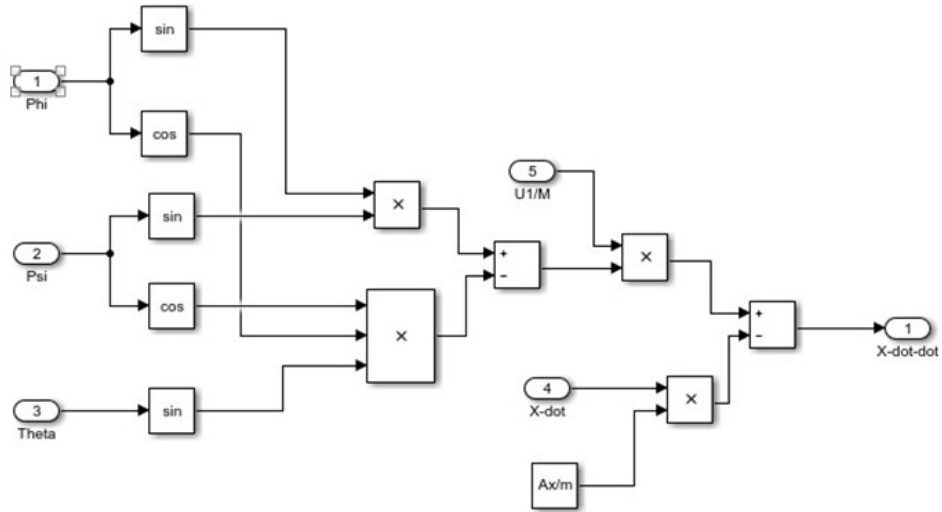
Реалізація інтелектуалізованої системи керування безпілотними літальними апаратами

Підсистема кутового прискорення



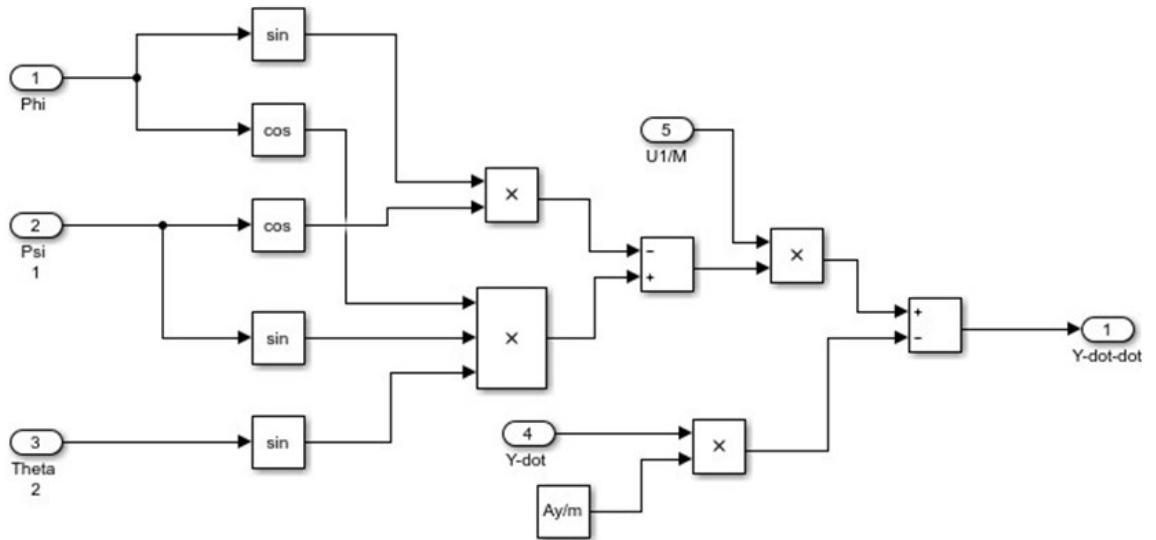
Реалізація інтелектуалізованої системи керування безпілотними літальними апаратами

Підсистема X-прискорення



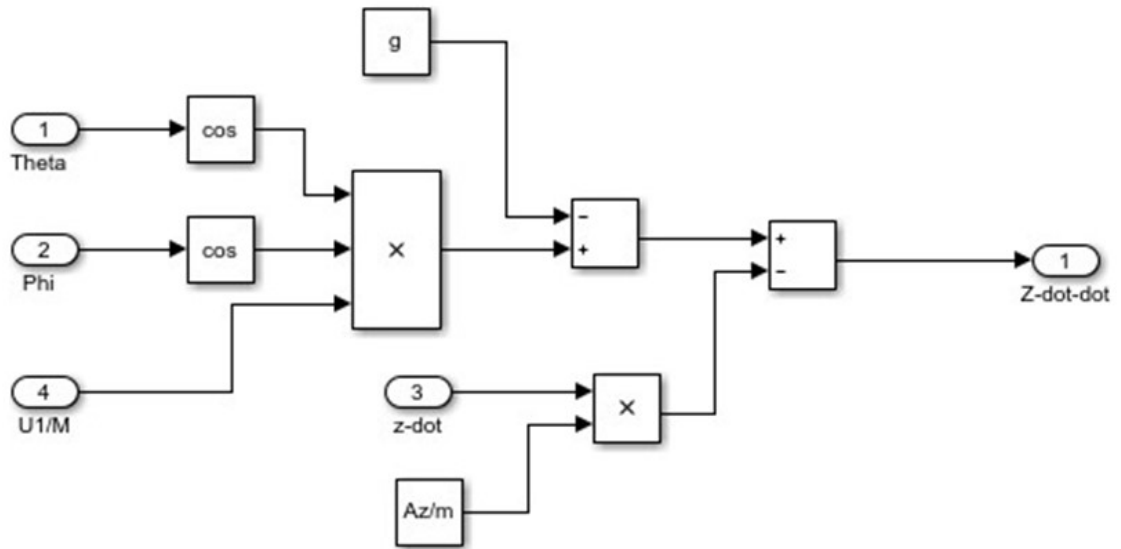
Реалізація інтелектуалізованої системи керування безпілотними літальними апаратами

Підсистема Y-прискорення



Реалізація інтелектуалізованої системи керування безпілотними літальними апаратами

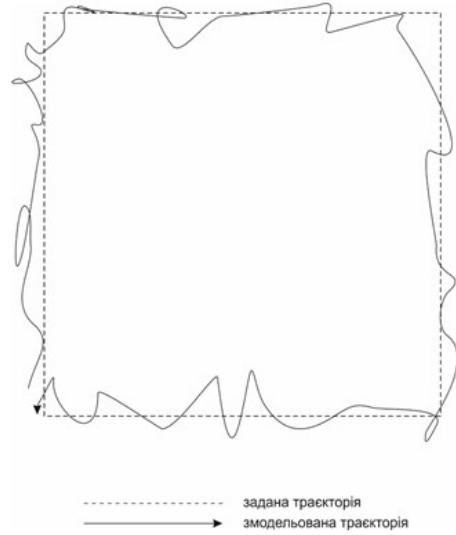
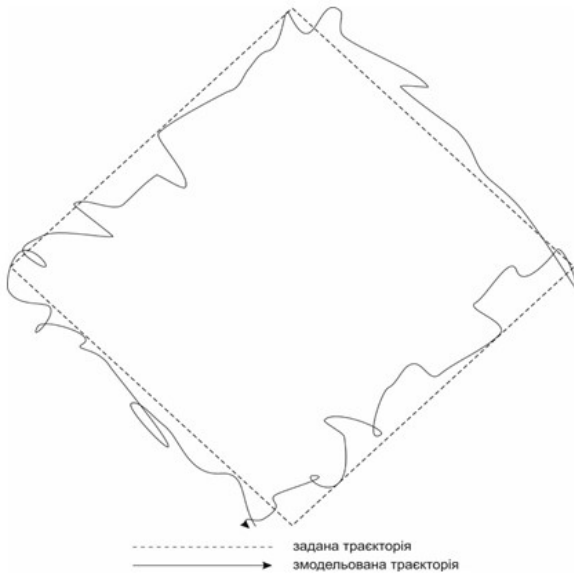
Підсистема Z-прискорення



Реалізація інтелектуалізованої системи керування безпілотними літальними апаратами

Експериментальні дослідження

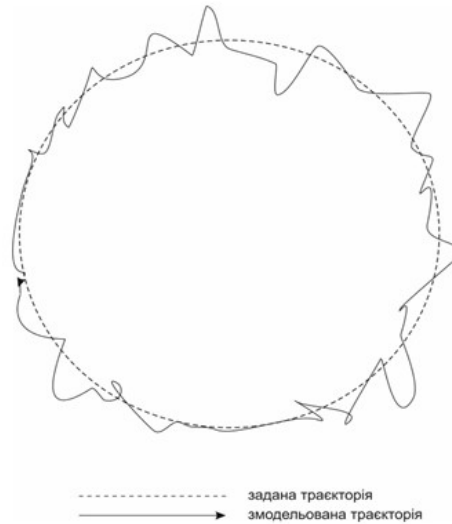
Рівностороннє відстеження траєкторій за допомогою PID-контролю та самоналаштування нечіткого PID-контролю



Реалізація інтелектуалізованої системи керування безпілотними літальними апаратами

Експериментальні дослідження

Рівностороннє відстеження траєкторій за допомогою PID-контролю та самоналаштування нечіткого PID-контролю



Реалізація інтелектуалізованої системи керування безпілотними літальними апаратами

Експериментальні дослідження

Порівняння результатів: рівнобічна траєкторія.

Алгоритм	Максимальна похибка (м)	Загальна похибка (м)
PID	0.32	8.02
Нечіткий-PID	0.15	4.35

Порівняння показників: квадратна траєкторія.

Алгоритм	Максимальна похибка (м)	Загальна похибка (м)
PID	0.39	12.17
Нечіткий-PID	0.18	6.94

Порівняння показників: кругова траєкторія.

Алгоритм	Максимальна похибка (м)	Загальна похибка (м)
PID	0.45	13.24
Нечіткий-PID	0.16	5.64

Публікації за матеріалами дипломної роботи та зв'язок роботи з науковими програмами, планами, темами

- Дослідження, представлені у кваліфікаційній роботі, проводились в рамках держбюджетної НДР Хмельницького національного університету № 1Б-2019 «Агентно-орієнтована система підвищення безпеки та якості програмного забезпечення комп'ютерних систем» (номер державної реєстрації 0119U100662).
- За темою дипломної роботи опубліковано статтю на тему «Інтелектуалізована система керування безпілотними літальними апаратами» в журналі «Комп'ютерні системи та інформаційні технології» №1 за 2020 рік [48].

Висновки

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено інтелектуалізовану систему керування безпілотними літальними апаратами.

У першому розділі досліджено методи планування траєкторії польоту безпілотних літальних апаратів, зроблено висновки та постановку задачі

У другому розділі представлено розроблену математичну модель системи керування безпілотними літальними апаратами, яка описує рух БПЛА у всіх напрямках польоту разом з описом методу управління дроном. Також в розділі подано опис функціонування PID-контролера, а також описано модель самоналаштовуваного нечіткого PID-контролера.

У третьому розділі представлена інтелектуалізована система керування безпілотними літальними апаратами, в основі якої покладено застосування апарату нечіткої логіки, імплементація якої в контролер БПЛА дозволила відстежувати та керувати траєкторію його руху в високою точністю. В розділі представлені також експериментальні дослідження, які довели ефективність застосування апарату нечіткої логіки для керуванням дроном.

Висновки

У четвертому розділі представлено реалізацію інтелектуалізованої системи керування безпілотними літальними апаратами. В розділі представлено розроблений контролер інтелектуалізованої системи керування безпілотними літальними апаратами. Подано реалізацію автономного відстеження траєкторії дрона та описано використану для досліджень платформу дрона AR.Drone.

За темою дипломної роботи опубліковано статтю на тему «Інтелектуалізована система керування безпілотними літальними апаратами» в журналі «Комп'ютерні системи та інформаційні технології» №1 за 2020 рік.

Наукова новизна отриманих результатів:

1. Удосконалено метод керування безпілотними літальними апаратами, який на відміну від відомих застосовує компоненти штучного інтелекту, і який забезпечує підвищення точності траєкторії руху безпілотного літального апарату типу дрон.

2. Набула подальшого розвитку інтелектуалізована система керування безпілотними літальними апаратами, яка забезпечує підвищення точності траєкторії руху безпілотного літального апарату типу дрон.

Висновки

Практична цінність. В результаті виконаного наукового дослідження буде розроблено інтелектуалізовану систему керування безпілотними літальними апаратами, яка забезпечуватиме задання траєкторії руху безпілотного літального апарату типу дрон з високою точністю.

Зв'язок роботи з науковими програмами, планами, темами. Дослідження, представлені у кваліфікаційній роботі, проводились в рамках держбюджетної НДР Хмельницького національного університету № 1Б-2019 «Агентно-орієнтована система підвищення безпеки та якості програмного забезпечення комп'ютерних систем» (номер державної реєстрації 0119U100662).

За темою дипломної роботи опубліковано статтю на тему «Інтелектуалізована система керування безпілотними літальними апаратами» в журналі «Комп'ютерні системи та інформаційні технології» №1 за 2020 рік [48].

Висновки

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено інтелектуалізовану систему керування безпілотними літальними апаратами.

У першому розділі досліджено методи планування траєкторії польоту безпілотних літальних апаратів, зроблено висновки та постановку задачі

У другому розділі представлено розроблену математичну модель системи керування безпілотними літальними апаратами, яка описує рух БПЛА у всіх напрямках польоту разом з описом методу управління дроном. Також в розділі подано опис функціонування PID-контролера, а також описано модель самоналаштуваного нечіткого PID-контролера.

У третьому розділі представлена інтелектуалізована система керування безпілотними літальними апаратами, в основі якої покладено застосування апарату нечіткої логіки, імплементація якої в контролер БПЛА дозволила відстежувати та керувати траєкторію його руху в високою точністю. В розділі представлені також експериментальні дослідження, які довели ефективність застосування апарату нечіткої логіки для керуванням дроном.

Висновки

Дослідження, представлені у кваліфікаційній роботі, проводились в рамках держбюджетної НДР Хмельницького національного університету № 1Б-2019 «Агентно-орієнтована система підвищення безпеки та якості програмного забезпечення комп'ютерних систем» (номер державної реєстрації 0119U100662).

За темою дипломної роботи опубліковано статтю на тему «Інтелектуалізована система керування безпілотними літальними апаратами» в журналі «Комп'ютерні системи та інформаційні технології» №1 за 2020 рік [48].

Додаток Г
(обов'язковий)

Опублікована стаття на тему «Інтелектуалізована система керування
безпілотними літальними апаратами» в журналі «Комп'ютерні системи та
інформаційні технології» №1 за 2020 рік [48]

УДК 528.1:528.4

С.М. ЛИСЕНКО, С.В. РУМЯНЦЕВ
Хмельницький національний університет**ІНТЕЛЕКТУАЛІЗОВАНА СИСТЕМА КЕРУВАННЯ БЕЗПІЛОТНИМИ
ЛІТАЛЬНИМИ АПАРАТАМИ**

В роботі представлено інтелектуалізовану систему керування безпілотними літальними апаратами. В її основі лежить застосування апарату нечіткої логіки, імплементація якої в контролер безпілотного літального апарату дозволила відстежувати та керувати траєкторію його руху. Проведені експериментальні дослідження доводять ефективність застосування апарату нечіткої логіки для керуванням дрону в умовах здійснення зовнішніх впливів.

Ключові слова: безпілотний літальний апарат, дрон, PID-контролер, система керування.

S. LYSENKO, S. RUMIANTSEV

Khmelnyskiy National University, Khmelnytskyi, Ukraine

INTELLECTUALIZED CONTROL SYSTEM FOR UNMANNED AERIAL VEHICLE

Abstract – The paper presents an intellectualized control system for unmanned aerial vehicles. It is based on the use of the fuzzy logic, implementation of which in the Controller of the unmanned aerial device allowed to track and control the trajectory of its movement. The experimental researches prove the efficiency of the device application of fuzzy logic for control of the drone in conditions of external influences implementation. To construct an intellectualized control system of unmanned aerial vehicles, their structure was considered. The base of the system is used for quad-unmanned aerial vehicle, which includes four screws, located symmetrically around the central building. The peculiarity of the drone is that its adjacent screws must spin the opposite one from each other. This requirement is explained by the need to prevent system rotation around its own central axis. Depending on the required trajectory, an important aspect was the ability to set different values of the power of the drone engines. Despite the fact that the simplicity of its structure are characterized, they are able to implement a large set of motion models together with a demonstration of high maneuverability. It is reached the presence of six degrees of freedom, which consist of three progressive and three rotating components to set the trajectory of a movement. In order to solve this problem it was possible to solve the apparatus of fuzzy logic as the basis of mathematical model of the system. This allowed to ensure a vague logical control of the fog, and, in turn, intellectualize the behavior of drone in the air in the conditions of external influences on the change of a predetermined trajectory of its movement. At the heart of the Intellectualized unmanned Aerial vehicle control system, two fuzzy controllers were involved in the production of control signals for the command of a UAV flight height and an angle of inclination.

Keywords: Unmanned aerial vehicle, drone, PID-controller, control system.

Вступ

Безпілотні літальні апарати (БЛА) - літальні апарати, який літають без фізичної присутності пілота на його борту – за останні десять років широко використовуються в багатьох сферах: від побутових задач до військових цілей [1]. Історично БЛА вперше був застосований для місій у військових завданнях, таких як оцінка бою та розвідка авіабази. Сьогодні використання безпілотних літальних апаратів (дронів) розширено до комерційних, наукових, сільськогосподарських та інших сфер. Дрони знайшли своє застосування в системах стеження, доставки, моніторингу сільськогосподарських угідь, оцінки збитку після стихійних лих [2], виявлення жертв [3], і аерофотографії, оскільки вони добре підходять для оснащення різних давачів на борту, є швидкими і характеризуються високою маневреністю платформи [4].

Широкий спектр застосування БЛА полягає в їх простоті побудови та маневреності.

Виходячи з того, що інтерес до БЛА характеризується постійним зростанням, сьогодні вони мають ряд суттєвих недоліків, таких як недостатньо велика вантажопідйомність і малий час польотів. Іншим аспектом, який необхідно ґрунтовно досліджувати є проблема стійкості та оптимального керування, оскільки вони нестійкі до зовнішніх впливів.

Тому актуальною задачею є розроблення інтелектуалізованої система керування безпілотними літальними апаратами, здатна адаптивно реагувати на зовнішні впливи, що здатні змінювати траєкторії руху дронів.

Пов'язані роботи

У сучасній науковій літературі присутня велика кількість наукових праць, які в тій чи іншій формі пропонують методи та засоби до вирішення вищевказаних недоліків.

Так, у [4] розглянуто алгоритм на основі клітинних автоматів для створення траєкторії руху БЛА з урахуванням ширини сліду на землі. В [5] запропонована система навігації БЛА на основі стандарту IEEE 802.15.4. В [6] запропоновано ієрархічну модель взаємодії мобільних давачів БЛА.

В [7] запропоновано математичну модель керування безпілотним літальним апаратом, що враховує дії оператора під час управління БЛА. Запропонована модель дає можливість більш глибокого, якісного та швидкого підбору належних значень параметрів руху, які необхідні для оптимальної та ефективної траєкторії руху БЛА.

У [8] запропоновано схему побудови комплексного навігаційного комплексу безпілотного літного засобу, основою якого є дві підсистеми- інерційні і повітряні. Для кожної підсистеми передбачена реалізація алгоритмів комплексної обробки навігаційної інформації від відповідних навігаційних датчиків та бортового обладнання системи глобального позиціонування на основі схеми неінваріантної компенсації. Логіка використання навігаційних вимірювань з обох підсистем відображає адаптацію навігаційного комплексу до умов польоту повітряного судна. Адаптивний навігаційний комплекс безпілотний літальний апарат підвищує точність визначення навігаційних параметрів польоту, надійності та завадостійкості БЛА під час польоту.

В [9] розроблено систему керування траєкторією польотів дронів, яка включає в себе перевірку давачів перед їх використанням. Експериментальні результати досліджень множини давачів показали, що розроблена випробувальна апаратура дозволяє здійснювати калібрування мікромеханічних датчиків без необхідності виконання тестових рейсів, що знижує ризик пошкодження безпілотних літальних апаратів у разі значних похибок вимірювань.

Авторами роботи [10] запропоновано модель системи керування БЛА яка дає можливість надати інформацію для визначення об'єму, складу та виконання інтервалів використання БЛА, а також дозволяє врахувати зміну складових умов, що є в інтегрованому порядку, викликані експлуатаційним часом і терміном експлуатації (особливості експлуатації та зберігання).

Розроблені методи і даний алгоритм [11] дозволяють побудувати оптимальну траєкторії польоту БЛА над певними піками для того, щоб покрити найбільшу площу. Це мінімізує можливі ризики, пов'язані з невільно здійсненим польотом. Крім того, застосування методів мінімізує енергію і часові витрати.

У статті [12] розглянуто ортогональне проектування для вивчення чинників, що впливають на стабільність польоту БЛА. У статті [13] представлена система безпілотних літальних апаратів, та її інтеграція в IoT.

Розглянуті системи в переважній більшості вирішують завдання підвищення точності побудови траєкторії руху ПЛА, однак не мають можливостей до адаптивної корекції траєкторії, тому актуальною є задача розроблення інтелектуалізованої система керування безпілотними літальними апаратами.

Інтелектуалізована система керування безпілотними літальними апаратами

Для побудови інтелектуалізованої системи керування безпілотними літальними апаратами розглянемо їх структуру. Зокрема, за основу системи використаємо квадрокоптер - безпілотний літальний апарат, що включає в себе чотири гвинти, розташовані симетрично навколо центрального корпусу.

Особливістю дрону є те, що його суміжні гвинти повинні обертатися протилежного один від одного. Ця вимога пояснюється необхідністю запобігання обертання системи навколо власної центральної осі. В залежності від необхідної траєкторії руху, важливим аспектом є здатність задання різних значень потужності двигунів дрону. Попри те, що ДЛА характеризуються простотою своєї структури, вони в процесі експлуатації здатні імплементувати велику множину моделей руху разом з демонстрацією високої маневреності. Це досягається наявністю шести ступенів вільності, які складаються з трьох поступальних та трьох обертових компонентів задання траєкторії руху ДЛА.

Схематичне представлення БЛА подано на рисунку 1.

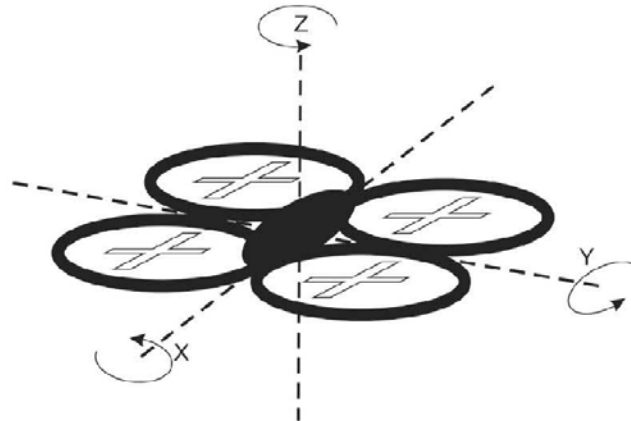


Рис. 1. Схема БЛА

Представимо систему керування безпілотними літальними апаратами моделлю у вигляді кортежу параметрів:

$$F_{UAV} = (z, v_x, v_y, \theta, \varphi, \psi), \quad (1)$$

де z - висота від землі до БЛА в повітрі (м);

v_x - лінійна швидкість БЛА в напрямку x , (м/с);

v_y - лінійна швидкість безпілотної апарату у напрямку y , (м/с);

θ - кут нахилу БЛА по осі y , (рад);

φ - кут нахилу (крену) БЛА по осі x , (рад);

ψ - кут повороту БЛА навколо вертикальної осі z , (рад).

Розглянемо множину команд, які сприймає внутрішній контролер польотів безпілотного літального апарату. З цією метою подамо набір команди управління траєкторією руху БЛА кортежем:

$$C_{UAV} = \{\xi_z, \xi_\psi, \xi_\theta, \xi_\varphi\}, \quad (2)$$

де ξ_z - команда задання лінійної швидкості в напрямку по осі z ;

ξ_ψ - команда задання кута повороту руху БЛА навколо вертикальної осі z ;

ξ_θ - команда задання висоти польоту БЛА по осі y ;

ξ_φ - команда задання кута нахилу (крену) БЛА по осі x .

Очевидно, що задання команд руху ξ_θ, ξ_φ реалізують поступальний рух вздовж x - і y -напрямків відповідно. Тому для регулювання швидкостей дрону в необхідних напрямках застосуємо команд ξ_θ, ξ_φ , а для забезпечення орієнтації БЛА використовуватиметься команда ξ_ψ .

Вхідні та вихідні параметри системи керування БЛА представлено на рисунку 2.

Система керування ДЛА

Керування ДЛА є складною задачею, оскільки продуктивність контролера залежить від точності математичної моделі проектованої системи. Відомі математичні моделі часто нехтують зовнішніми впливами, які в реальних умовах зазнає дрон в польоті (сила тяжіння, тертя повітря, сила вітру тощо).

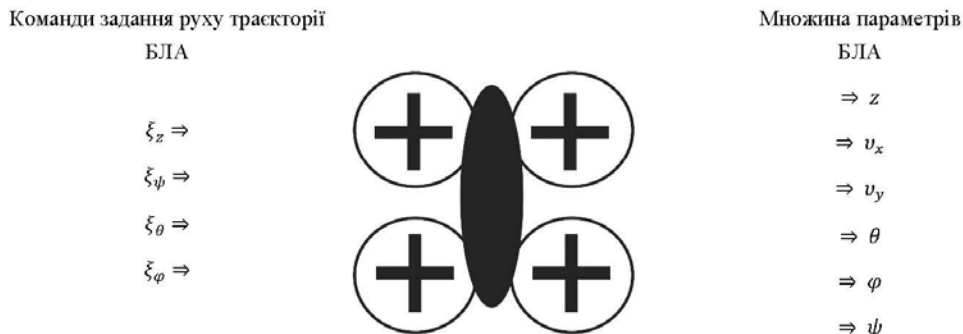


Рис. 2. Вхідні та вихідні параметри системи керування БЛА

Іншим аспектом, що суттєво впливає на зниження точності виконання заданої траєкторії, є шум при одержанні інформації з давачів ДЛА. Це призводить до суттєвої невизначеності при практичному застосуванні дрону і необхідності адаптивної корекції траєкторії його руху.

З метою вирішення даної проблеми можливим її вирішення є застосування апарату нечіткої логіки як основи математичної моделі системи. Це дозволить забезпечити нечітке логічне керування ДЛА, і, в свою чергу інтелектуалізувати поведінку дрона в повітрі в умовах зовнішніх впливів на зміну заданої траєкторії його руху.

В основі інтелектуалізованої системи керування безпілотними літальними апаратами лежить побудова двох нечітких контролерів для продукції значень сигналів керування для:

ξ_θ - команда задання висоти польоту БЛА по осі y;

ξ_φ - команда задання кута нахилу (крену) БЛА по осі x.

Функції належності для команд управління ξ_θ та ξ_φ подано на рисунку 3.

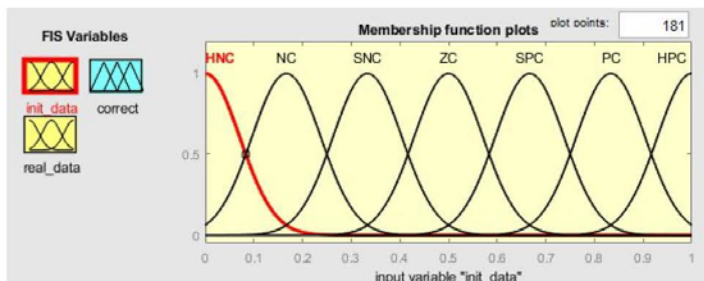


Рис. 3 Функції належності для команд управління ξ_θ та ξ_φ

У якості вхідної та вихідної лінгвістичних змінних для нечіткого керування БЛА X приймемо ім'я: «Ступінь корегування траєкторії» дрону, значення якої буде визначатися на етапі його експлуатації в умовах зовнішніх впливів. Для прийнятих лінгвістичних змінних визначимо терм-множини і функції належності. Терм-множини значень керуючого сигналу приймемо наступним чином: «High negative correction (HNC)», «Negative correction (NC)», «Slight negative correction (SNC)», «Zero correction (ZC)», «Slight positive correction (SPC)», «Positive correction (PC)», «High positive correction (HPC)», з областю визначення [0,1].

Для генерації команд управління ξ_θ та ξ_φ відповідний нечіткий контролер має два входи, які, в свою чергу, оперують значеннями для формування траєкторії польоту в x- або y-напрямку:

- 1) початкові значення (σ_x, σ_y) ;
- 2) та скореговані під впливом зовнішніх чинників (σ'_x, σ'_y) .

Інтелектуалізована система керування безпілотними літальними апаратами керується множиною правил нечітких продукцій (ПНП). Кожне із правил має деяку вагу $F_i = [0, 1]$, що визначає значимість правила або впевненість у ступені істинності висновку, який отримується за окремим нечітким правилом. В дослідженні було залучено двадцять п'ять правил нечітких продукцій, приклади яких подано нижче:

П.1. If (σ_x, σ_y) is HNC and (σ'_x, σ'_y) is HNC then HPC (1);

П.2. If (σ_x, σ_y) is HNC and (σ'_x, σ'_y) is NC then PC (1);

.....

П.25. If (σ_x, σ_y) is HPC and (σ'_x, σ'_y) is HPC then HNC (1).

Дефазифікація, тобто процедура одержання чисельного значення для прийняття остаточного рішення щодо корегування траєкторії руху дрона було виконано за допомогою застосування методу центра тяжіння.

Експерименти

Для оцінки ефективності застосування апарату нечіткої логіки для побудови інтелектуалізованої системи керування безпілотними літальними апаратами було проведено ряд експериментальних досліджень. З цією метою було проаналізовано поведінку дрона без та із зовнішнім впливом (вітром) як без залучення нечіткого керування, так і з ним. Як приклад траєкторії руху було обрано прямокутник. Для здійснення експериментальних досліджень поведінки дрона без зовнішніх впливів було два PID-контролери використовувались для генерації команд управління ξ_θ та ξ_φ . Для стабілізації траєкторії сигнал контролера визначається трьома компонентами K_p , K_i , K_d – коефіцієнти підсилення пропорційної, інтегральної та диференційних складових PID контролера.

Значення параметрів для експериментів подано в таблиці 1, а траєкторія польоту дрона подана рис. 4.

Таблиця 1.

Значення параметрів PID контролера

Параметр PID контролера	Значення параметра (розрахункові)	Значення параметра (з уточненням)
K_p	0,41	0,56
K_i	0,06	0,02
K_d	0,07	0,12

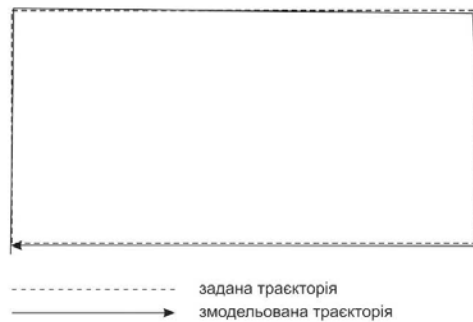


Рис. 4. Траєкторія польоту дрона без зовнішніх впливів

Модель керування дроном в середовищі Simulink Matlab [14] подано на рисунку 5.

Наступний експеримент полягав у дослідженні поведінки дрон в польоті із зовнішнім впливом вітру. Результати моделювання, отримані на основі застосування нечіткого логічного контролера, представлені на рисунку 6.

Результати експериментальних досліджень продемонстрували, що залучення нечітких контролерів дуже наближені до необхідної еталонної траєкторії руху дрона, і демонструють вищу точність у порівнянні зі стоковим PID контролером.

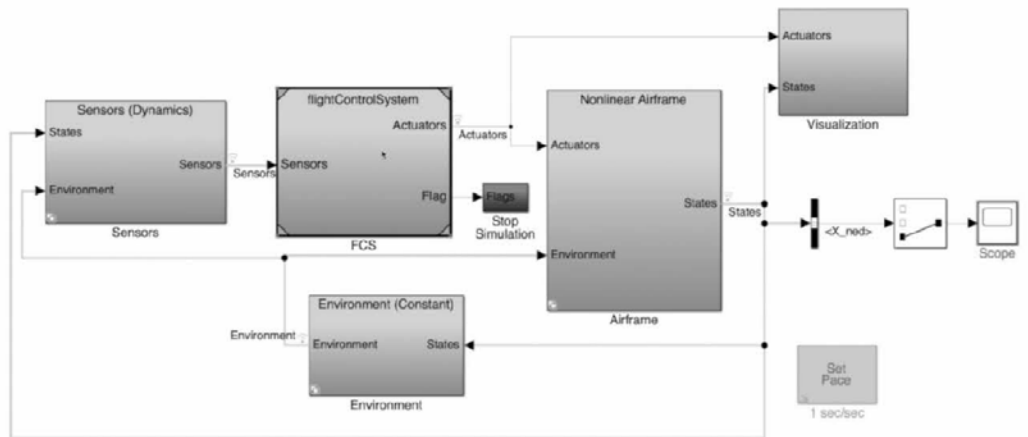


Рис. 5. Модель керування дроном в середовищі Simulink Matlab

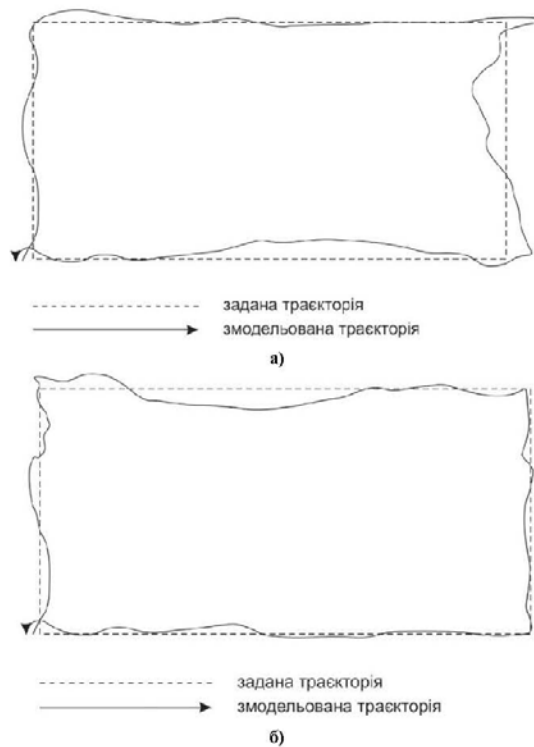


Рис. 6. Траєкторія польоту: а) без використання нечіткого контролера; б) з використанням нечіткого контролера: без з урахуванням впливу вітру

Висновки

В роботі представлена інтелектуалізована система керування безпілотними літальними апаратами. В її основі лежить застосування апарату нечіткої логіки, імплементація якої в контролер БЛА дозволила відстежувати та керувати траєкторію його руху.

Проведені експериментальні дослідження доводять ефективність застосування апарату нечіткої логіки для керуванням дроном в умовах здійснення зовнішніх впливів.

Як розвиток запропонованої інтелектуалізованої системи керування безпілотними літальними апаратами подальші наукові дослідження може бути залучення штучних нейронних мереж для підвищення точності траєкторії дронів.

Література

1. A. Prayitno, V. Indrawati, and G. Utomo, "Trajectory tracking of ar.drone quadrotor using fuzzy logic controller," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 12, no. 4, pp. 819–828, 2014.
2. J. A. Paredes, J. González, C. Saito and A. Flores, "Multispectral imaging system with UAV integration capabilities for crop analysis," 2017 First IEEE International Symposium of Geoscience and Remote Sensing (GRSS-CHILE), Valdivia, 2017, pp. 1-4, doi: 10.1109/GRSS-CHILE.2017.7996009.222
3. A. V. Skripets, A. G. Sytnyk, A. A. Polozhevets and M. O. Onyshchuk, "Optimal aerospace monitoring by means of UAV intelligent control," 2013 IEEE 2nd International Conference Actual Problems of Unmanned Air Vehicles Developments Proceedings (APUAVD), Kiev, 2013, pp. 181-183, doi: 10.1109/APUAVD.2013.6705319.
4. Z. Song, H. Zhang, F. Liu, S. Chen and F. Zhang, "Unmanned Aerial Vehicle Coverage Path Planning Algorithm Based on Cellular Automata*," 2018 *International Conference on Information Systems and Computer Aided Education (ICISCAE)*, Changchun, China, 2018, pp. 371-374, doi: 10.1109/ICISCAE.2018.8666895.
5. V. M. Sineglazov and E. V. Daskal, "Unmanned aerial vehicle navigation system based on IEEE 802.15.4 standard radiounits," 2017 *IEEE 4th International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*, Kiev, 2017, pp. 241-244, doi: 10.1109/APUAVD.2017.8308820.
6. Romanyuk, V. A., & Sova, O. Y. (2015). *The hierarchical control system model of wireless sensor networks using unmanned aerial vehicles. 2015 IEEE International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*. doi:10.1109/apuavd.2015.7346609
7. Kharchenko, V., Matiychyk, D., & Babenko, A. (2017). *Mathematical model of unmanned aerial vehicle control in manual or semiautomatic modes. 2017 IEEE 4th International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*. doi:10.1109/apuavd.2017.8308771
8. Zakharin, F., & Ponomarenko, S. (2017). *Unmanned aerial vehicle integrated navigation complex with adaptive tuning. 2017 IEEE 4th International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*. doi:10.1109/apuavd.2017.8308768
9. Vasylenko, M. P. (2017). *Testing system for unmanned aerial vehicles microelectromechanical sensors. 2017 IEEE 4th International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*. doi:10.1109/apuavd.2017.8308804
10. Chepizhenko, V. I., & Tryzna, O. O. (2015). *Maintenace planing optimisation of Unmanned Aerial systems. 2015 IEEE International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*. doi:10.1109/apuavd.2015.7346577
11. Barabash, O., Dakhno, N., Shevchenko, H., & Sobchuk, V. (2019). *Unmanned Aerial Vehicles Flight Trajectory Optimisation on the Basis of Variational Inequality Algorithm and Projection Method. 2019 IEEE 5th International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*. doi:10.1109/apuavd47061.2019.8943869
12. Hu, X., & Huang, X. (2017). *Orthogonal design and optimization of flight stability test for the quadrotor unmanned aerial vehicle. 2017 IEEE International Conference on Unmanned Systems (ICUS)*. doi:10.1109/icus.2017.8278366
13. E. Vlasceanu, D. Popescu and L. Ichim, "Ground Control Station for an Unmanned Aerial Vehicle Integrated in IoT," 2018 10th *International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, Iasi, Romania, 2018, pp. 1-6, doi: 10.1109/ECAI.2018.8679050
14. MATLAB Simulink. URL: <https://uk.mathworks.com/help/simulink/simulink-environment.html>. (дата звернення: 31.07.2020).

Ім'я користувача:
Кафедра КІ

ID перевірки:
1007647473

Дата перевірки:
01.05.2021 19:27:47 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
01.05.2021 19:28:29 EEST

ID користувача:
100005591

Назва документа: Інтелектуалізована система керування безпілотними літальними апаратами

Кількість сторінок: 84 Кількість слів: 12208 Кількість символів: 89493 Розмір файлу: 7.23 MB ID файлу: 1007757983

7.27% Схожість

Найбільша схожість: 6.19% з Інтернет-джерелом (<http://elar.khnu.km.ua/jspui/bitstream/123456789/9591/1/CSIT-2020-N..>)

6.47% Джерела з Інтернету

13

Сторінка 86

0.8% Джерела з Бібліотеки

11

Сторінка 86

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

104

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 0.0%

Словари проверки: en_US, ru_RU, ua_UA. Ошибок в документах: 7%

ID: 89700 Название: Інтелектуалізована система керування безпілотними літальними апаратами Добавлено в БД: 2021-05-01 Авторы: Румянцев С.В. Руководители: Лисенко С.М. Консультанты: Опоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	81546	656	158 (0%)	3 (0%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

РЕЦЕНЗІЯ НА ДИПЛОМНИЙ ПРОЕКТ

Дипломник студент групи КІ2м-19-1 Румянцев С.В.

Тема Інтелектуалізована система керування безпілотними літальними апаратами

Спеціальність 123 Комп'ютерна інженерія

Обсяг дипломного проекту:

кількість листів креслень 0; кількість сторінок записки 87

1. Короткий зміст ДП та прийнятих рішень Представлена робота присвячена розробленню інтелектуалізованої системи керування безпілотними літальними апаратами, яка забезпечує задання траєкторії руху безпілотного літального апарату типу дрон з високою точністю ...

2. Висновок про відповідність ДП дипломному завданню Кваліфікаційна робота у повній мірі відповідає поставленому завданню із дотриманням усіх вимог

3. Характеристика виконання кожного розділу проекту, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі досліджено методи планування траєкторії польоту безпілотних літальних апаратів, зроблено висновки та постановку задачі

У другому розділі представлено розроблену математичну модель системи керування безпілотними літальними апаратами, яка описує рух БПЛА у всіх напрямках польоту разом з описом методу управління дроном. Також в розділі подано опис функціонування PID-контролера, а також описано модель самоналаштуваного нечіткого PID-контролера.

У третьому розділі представлена інтелектуалізована система керування безпілотними літальними апаратами, в основі якої покладено застосування апарату нечіткої логіки, імплементація якої в контролер БПЛА дозволила відстежувати та керувати траєкторію його руху в високою точністю. В розділі представлені також експериментальні дослідження, які довели ефективність застосування апарату нечіткої логіки для керуванням дроном.

У четвертому розділі представлено реалізацію інтелектуалізованої системи керування безпілотними літальними апаратами. В розділі представлено розроблений контролер інтелектуалізованої системи керування безпілотними літальними апаратами. Подано реалізацію автономного відстеження траєкторії дрона та описано використану для досліджень платформу дрона AR.Drone.

4. Позитивні сторони роботи В результаті виконаного наукового дослідження буде розроблено інтелектуалізовану систему керування безпілотними літальними апаратами, яка забезпечуватиме задання траєкторії руху безпілотного літального апарату типу дрон з високою точністю.

5. Негативні сторони проекту Не в повній мірі здійснено аналіз інших інтелектуалізованих систем керування безпілотними літальними апаратами.

6. Оцінка графічного оформлення та пояснювальної записки проекту Матеріали кваліфікаційної роботи є структурованими у чіткій та логічній формі та відображають послідовність виконання поставлених задач

7. Відгук про проект в цілому Зміст представленої роботи в повній мірі розкриває обрану тему. Дослідження, проведені є аргументованими в повній мірі.

8. Інші зауваження _____

9. Оцінка дипломного проекту Робота заслуговує оцінки «добре», а її автор присвоєння кваліфікації «магістр з комп'ютерної інженерії»

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи) Джулій В.М. , к.т.н., доцент, кафедри кафедри кібербезпеки та комп'ютерних систем і мереж Хмельницького національного університету

« 28 » _____ квітня _____ 2021 р.



_____ (підпис)

Завідувачу кафедри КІСП
д-р.техн.наук, проф. Говорущенко Т. О.

Румянцева С.В.

ПІБ здобувача вищої освіти

ФПКТС, 2 курсу, групи КІ2М-19-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіатоповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

5.05.2021

дата


підпис

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА СИСТЕМНОГО ПРОГРАМУВАННЯ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Інтелектуалізована система керування безпілотними літальними апаратами

Автор: Румянцев Сергій Володимирович

Спеціальність: 123 – Комп'ютерна інженерія та програмування

Освітня програма: освітньо-наукова

Науковий керівник: Лисенко С.М., д.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в РОЗДІЛІ 3, що розміщені в статті студента, які результатами дослідження (з 7.27% виявлених стаття складає 6.19%);
- 2) у розділі ВСТУП є матеріали, які є описом методів дослідження, які було використано студентом для вирішення задач, а також опис зв'язку з роботі з науковими програмами, планами, темами;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами в підписах рисунків;
- 4) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 7.27%, з яких 6.19% є збігом з матеріалами публікації студента, і адресується до 24 першоджерел, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____

С.М. Лисенко

Гарант ОП _____

О. С. Савенко

Завідувач кафедри КІСП _____

Т. О. Говорущенко