

Міністерство освіти і науки України
Національний аерокосмічний університет ім. М.Є. Жуковського
«Харківський авіаційний інститут»

Факультет систем управління літальними апаратами

Кафедра систем управління літальних апаратів

Пояснювальна записка

до дипломної роботи

магістра

(освітньо-кваліфікаційний рівень)

на тему «Розробка автоматизованої системи визначення координат БПЛА за
візуальними орієнтирами»

ХАІ.301.360.200.173.00063020 ПЗ

Виконав: студент 6 курсу, групи 360
спеціальності 173 «Авіоніка»

освітня програма «Системи керування
літальними апаратами та комплексами»

Чароватий Г. Г.

(прізвище та ініціали студента)

Керівник Дергачов К.Ю.

(прізвище та ініціали)

Рецензент Кузнецов Ю.О.

(прізвище та ініціали)

Міністерство освіти і науки України
Національний аерокосмічний університет ім. М.Є. Жуковського
«Харківський авіаційний інститут»

Факультет систем управління літальними апаратами
Кафедра систем управління літальних апаратів
Освітньо-кваліфікаційний рівень магістр
Спеціальність 173 «Авіоніка»
Освітньо-професійна програма «Комп'ютеризовані системи управління та автоматика»

«ЗАТВЕРДЖУЮ»
Завідуючий кафедрою
к.т.н., с.н.с., доцент
_____ К.Ю. Дергачов
«__» _____ 2020 р.

**З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ**

Чароватого Геннадія Геннадійовича
(прізвище, ім'я, по батькові)

1. Тема проекту «Розробка автоматизованої системи визначення координат БПЛА за візуальними орієнтирами».
керівник проекту к.т.н, с.н.с., доцент каф. 301 Дергачов К.Ю., затверджені наказом вищого навчального закладу від _____ року № _____
2. Строк подання студентом проекту: 17 грудня 2020 року.
3. Вихідні дані до проекту технічне завдання на дипломне проектування.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналітичний огляд літератури та існуючих методів рішення задачі, дослідження проблеми та синтез математичних моделей, проектування системи, експериментально-практична частина, розрахунок собівартості й ціни програмного продукту, побудова точки беззбитковості.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) аналітичний огляд існуючих рішень, функціональна блок-схема програмного продукту, алгоритми рішень поставленої задачі, прототипи графічного інтерфейсу, зображення розробленого інтерфейсу, результати експерименту, результати розрахунку собівартості програмного продукту.

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Стан проблеми	к.т.н, с.н.с., доцент каф. 301 Дергачов К.Ю.		
Аналітичний огляд літератури та існуючих методів рішення задачі	к.т.н, с.н.с., доцент каф. 301 Дергачов К.Ю.		
Дослідження проблеми і синтез математичних моделей	к.т.н, с.н.с., доцент каф. 301 Дергачов К.Ю.		
Проектування системи	к.т.н, с.н.с., доцент каф. 301 Дергачов К.Ю.		
Реалізація системи	к.т.н, с.н.с., доцент каф. 301 Дергачов К.Ю.		
Експериментально-практична частина	к.т.н, с.н.с., доцент каф. 301 Дергачов К.Ю.		
Економічне обґрунтування	професор каф.601 Хлівна І.В.		

7. Дата видачі завдання 01.09.2020

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту	Строк виконання етапів проекту	Примітка
1.	Початок переддипломної практики	01.09.2020	
2.	Формулювання теми проекту. Розробка технічного завдання	18.09.2020	
3.	Математичний опис системи управління. Аналіз і синтез системи управління. Проведення експериментальних досліджень	23.10.2020	Залік з переддипломної практики
4.	Конструкторська частина роботи. Технологічна частина роботи. Експериментально-практична частина. Економічне обґрунтування розробки. Розробка питань охорони праці і безпеки в надзвичайних ситуаціях	16.11.2020	
5.	Оформлення розрахунково-пояснювальної записки і графічного матеріалу	27.11.2020	
6.	Попередній захист проекту	1.12.2020	
7.	Рецензування проекту. Захист проекту в ДЕК	17.12.2020	

Студент _____ Чароватий Г.Г.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Дергачов К.Ю.
(підпис) (прізвище та ініціали)

Міністерство освіти і науки України
Національний аерокосмічний університет ім. М.Є. Жуковського
«Харківський авіаційний інститут»

Кафедра 301

«ЗАТВЕРДЖУЮ»

Завідуючий кафедрою

к.т.н., с.н.с., доцент

_____ К.Ю. Дергачов

«__» _____ 2020 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на дипломне проектування
Чароватого Геннадія Геннадійовича

1 Тема роботи: «Розробка автоматизованої системи визначення координат БПЛА за візуальними орієнтирами».

затверджена наказом по університету від «__» _____ 2020 р. № _____.

2 Строк здачі студентом закінченої роботи « 17 » _____ грудня _____ 2020 р.

3 Область застосування розробки: промислові підприємства, військова промисловість.

4 Початкові дані для розроблювальної системи

4.1 Призначення і мета створення системи: розробка системи визначення координат.

4.2 Загальні відомості: динамічне планування маршруту в складному середовищі, що містить як заздалегідь відомі, так і невідомі об'єкти, які можуть використовуватися як орієнтири.

5 Технічні вимоги до каналів системи управління

5.1 Питання, що підлягають розробці: огляд існуючих рішень; аналіз шляхів визначення проблеми; розробка алгоритмів для рішення поставленого завдання класифікації зображень; створення програмного продукту з впровадженими алгоритмами; тестування; експеримент.

5.2 Вимоги до структури й функціонування системи: наявність квадрокоптера з головним контролером PixHawk, GNSS-модулем U-blox Neo m8n 8N, Xiaomi Yi Sport Black.

5.3 Вимоги до показників якості системи: допустима відносна помилка ($\epsilon_{\text{доп}}$) – 2%, час перехідного процесу ($t_{\text{пп}}$) – 6с, перерегулювання (σ) – 20%, запас стійкості за амплітудою (L_3) – 20дБ, запас стійкості за фазою (ϕ_3) – 40°.

5.4 Вимоги до конфігурації обчислювальної техніки: швидкість процесора не нижче 1,2ГГц, обсяг оперативної пам'яті – не нижче 1Гб.

6 Умови експлуатації системи

6.1 Вимоги до програмної та інформаційної сумісності: оперативна система – Windows 10, програмування на мові Python або C#.

6.2 Вимоги до захисту інформації від несанкціонованого доступу: плата немає роз'ємів для зчитування інформації.

6.3 Вимоги до зовнішніх збурень: не передбачені.

Характер роботи системи (безперервної, циклічний, одноразового дії): безперервної дії.

7 Додаткові функції, реалізовані системою (сигналізація про несправності, реєстрація необхідної інформації, самоконтроль самої системи і т.ін.): реєстрація польотної інформації.

8 Обсяг виконуваних розроблювачем робіт

8.1 Етапи проведення роботи: аналіз стану проблеми, вивчення готових рішень, опис математичної моделі, проектування системи, реалізація системи, експериментальна частина, обґрунтування економічної ефективності проекту.

8.2 Обсяг розробки по кожному етапу: титульний лист, технічне завдання, анотація, список умовних позначень та скорочень, зміст, вступ, стан проблеми або досліджуваного питання і постановка задачі проектування, дослідження проблеми, проектування системи, реалізація системи, експериментальна частина,

обґрунтування економічної ефективності проекту, висновок, список використаної літератури.

8.3 Вимоги до чисельності й кваліфікації персоналу: оператор, який виступає в ролі контролера результату.

9 Вимоги до захисту інформації й надійності: при аварійному завершенні роботи БПЛА, інформація та параметри повинні бути збережені.

10 Порядок контролю й приймання системи: випробування і тестування продукту; приймання результатів роботи здійснюється відповідно до плану приймання, затвердженим на випускаючої кафедрі і узгодженим з керівником: здача змісту розрахунково-пояснювальної записки; здача програми; захист дипломної роботи.

11 Дослідницька частина: розробка інструкції програміста.

12 Експериментально-практична частина: розробка програмного продукту, тестування програмного засобу.

13 Економічна частина

13.1 Розробити (розрахувати, одержати): розрахувати собівартість і одержати ціну програмного продукту.

13.2 Умови і вимоги: оцінка економічної ефективності розвитку.

13.3 Очікуваний результат: ринкова вартість програмного продукту.

14 Перелік графічних матеріалів із зазначенням форматів:

10 плакатів формату А1 – функціональна схема системи, зовнішній вигляд програмних засобів, алгоритми у вигляді блок-схем.

Керівник проектування

Дергачов К.Ю.

(П.І.Б.)

Прийняв до виконання

Чароватий Г.Г.

(П.І.Б. студента)

« » _____ 2020 р.

» _____ 2020 р.

«

Погоджено з питань:

проектування

Дергачов К.Ю.

(П.І.Б.)

дослідницької частини

Дергачов К.Ю.

(П.І.Б.)

« » _____ 2020 р.

« » _____ 2020 р.

економіки

Хлівна І.В.

(П.І.Б.)

« » _____ 2020 р.

РЕФЕРАТ

Сторінок 144, рисунків 82, таблиць 14, джерел 31, додатків 2.

В даному дипломному проєкті розглянуто завдання розробки і дослідження автоматизованої системи визначення координат безпілотного летального апарату.

Метою роботи є розробка системи визначення координат для БПЛА на основі візуальних орієнтирів, що дозволяє забезпечити автономне визначення навігаційних параметрів.

Для досягнення поставленої мети вирішуються такі основні завдання:

- системний аналіз методів визначення координат в системах навігації БПЛА;
- розробка напрямів використання цифрової картографічної інформації для уточнення координат БПЛА;
- розробка алгоритмів обробки інформації, що дозволяє підвищити надійність визначення навігаційних параметрів;
- проектування програмної системи визначення координат БПЛА, здатної функціонувати на бортових обчислювальних системах.

В якості методу визначення місцезнаходження був обраний метод SLAM (Simultaneous Localization and Mapping) - метод одночасної локалізації і побудови карти.

Результатом даної роботи є розроблена автоматизована система визначення координат БПЛА.

БЕЗПЛОТНИЙ ЛІТАЛЬНИЙ АПАРАТ, СИСТЕМА
АВТОМАТИЧНОГО УПРАВЛІННЯ, СИСТЕМА ПОЗИЦІЮВАННЯ,
ВІЗУАЛЬНА НАВІГАЦІЯ, ПРОГРАМНИЙ ПРОДУКТ.

СПИСОК СКОРОЧЕНЬ

ТАУ - теорія автоматичного управління;
БПЛА - безпілотний літальний апарат;
ЛА - літальний апарат;
САУ - система автоматичного управління;
GPS - система глобального позиціонування;
ПЗ - програмне забезпечення;
ІНС - інерціальна навігаційна система;
ІІКК - інтегрований інформаційно-керуючий комплекс;
БІНС - безплатформенні інерціальні навігаційні системи;
ЗПСК - земна пов'язана система координат;
SLAM - одночасна локалізація та картографування;
SFM - структура від руху;
НПВО - неявно пряма візуальна одометрія;
РТАМ - паралельне відстеження та картографування;
ІСР - найближча точка ітерації;
GPU - блок обробки графіки;
КУ - коригувальний пристрій;
НСУ - наземна станція управління.

ЗМІСТ

Вступ.....	10
1 Аналітичний огляд літератури та існуючих методів рішення задачі	12
1.1 Аналіз технічного завдання	12
1.2 Аналіз методик вирішення завдань.....	13
1.3 Аналіз існуючих програмних засобів	25
1.4 Постановка задач проектування	47
2 Дослідження проблеми і синтез математичних моделей.....	48
2.1 Цілі, завдання, структура та вибір шляху вирішення проблеми.....	48
2.2 Математичні моделі	50
2.3 Оцінка результатів моделювання	68
3 Проектування системи.....	81
3.1 Алгоритми реалізації моделей	81
3.2 Вибір засобів реалізації	91
4 Реалізація системи.....	108
4.1 Опис програми	108
4.2 Керівництво користувача	114
4.3 Технологія тестування програмного забезпечення	115
5 Експериментально-практична частина.....	117
5.1 Результати експериментального тестування	117
6 Економічне обґрунтування розробки.....	120
6.1 Опис продукту	120
6.2 Сегментування ринку	120
6.3 Аналіз конкурентоспроможності	121
6.4 Розрахунок собівартості і ціни	123
6.5 Висновки	130
Висновки	131
Список використаних джерел.....	132
Додаток А.....	135
Додаток Б.....	141

ВСТУП

В даний час безпілотні літальні апарати (БПЛА) використовуються досить широко і різноманітно. Однак використання БПЛА є досить обмежене, наприклад, режимами керування з пульта оператора і проходження точок маршруту в простому (вільному від перешкод) середовищі.

Актуальною є задача розробки системи управління, що дозволяє здійснювати автономний політ по заданому маршруту. Один з ключових питань у вирішенні цього завдання - динамічне планування маршруту в складному середовищі, в просторі, що містить як заздалегідь відомі, так і невідомі об'єкти, які можуть використовуватися як орієнтири, але одночасно є перешкоди на своєму шляху.

В даний час актуальна задача візуальної навігації БПЛА в умовах відсутності зовнішніх навігаційних сигналів GPS або ГЛОНАСС, так як більшість апаратів оснащується недостатньо точними інерційними датчиками.

Зазвичай для візуальної навігації використовуються одна або кілька відеокамер, тому в останні кілька десятиліть завдання оцінки руху камери в просторі тільки по потоку зображень є об'єктом активних досліджень. На сьогоднішній день добре розвинені два підходи до вирішення цього завдання: V-SLAM і візуальна одометрія. Сучасні версії згаданих алгоритмів показують хороші результати і можуть працювати в реальному часі.

Підхід V-SLAM заснований на ідеї побудови глобальної карти просторових положень видимих на кадрах відео потоку ознак, узгодженої з положеннями камери протягом усього часу руху. Він добре зарекомендував себе при навігації в закритих приміщеннях і по замкнених траєкторіях. Метод візуальної одометрії здійснює поступове відновлення траєкторії (від точки до точки), дбаючи лише про її локальну узгодженість (про розташування поточної точки траєкторії щодо попередніх її точок в локальній системі координат при невідомих глобальних координатах), для чого будується лише локальна карта просторових положень ознак.

Підхід V-SLAM набагато складніше. Він більш точний, тому що накладає набагато більше обмежень на невідомі положення камери, але не обов'язково надійніший. Наприклад, помилки при виявленні точок перетинів траєкторії можуть серйозно вплинути на узгодженість всієї карти. У разі довгої непересічної траєкторії БПЛА метод V-SLAM не дає переваги в порівнянні з

візуальною одометрією. Очевидним недоліком підходу візуальної одометрії є зростання помилок з плином часу. Найбільша проблема обох алгоритмів полягає в неможливості оцінити масштаб сцени в разі використання лише однієї камери без додаткових датчиків.

Останнім часом були розроблені кілька оригінальних методів візуально інерційної одометра. В цих методах дані з інерційних датчиків використовуються як для уточнення відносних переміщень камери, так і для визначення масштабу сцени. Такі методи складніше в реалізації, ніж чисто візуальні, але набагато точніше і надійніше.

Разом з тим, незважаючи на високу точність, з плином часу вони неминуче накопичують помилки при польоті БПЛА по довгих траекторіях.

Для визначення координат БПЛА за допомогою методу візуальної навігації, без використання зовнішніх навігаційних сигналів GPS і ГЛОНАСС будується глобальна карта просторових положень ознак, видимих на кадрах відео потоку, отриманого в той час, коли присутні зовнішні навігаційні сигнали. Після втрати сигналу GPS і ГЛОНАСС здійснюється лише локалізація БПЛА щодо побудованої карти і повернення його в точку старту. В результаті не виробляються складні обчислення, пов'язані зі взаємним погодженням положень всіх ознак до положень камер. Положення кожної ознаки обчислюється незалежно на підставі відомих положень камер, в яких він спостерігався.

Оскільки глобальна карта побудована щодо відомих положень камер, подальша візуальна навігація щодо неї здійснюється без накопичення помилок і у відомому масштабі.

1 АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРИ ТА ІСНУЮЧИХ МЕТОДІВ РІШЕННЯ ЗАДАЧІ

1.1 А аналіз технічного завдання

Згідно з поставленим технічним завданням, необхідно спроектувати систему, яка буде дозволяти визначати координати БПЛА за візуальними орієнтирами.

В якості інструментів і засобів реалізації буде використовуватися мова програмування C#. Вибір мови C# пов'язаний з тим, що він є одним з найбільш зручних мов програмування для розробки Web-додатків, модульна платформа.

Розглядається алгоритм автономної візуальної навігації, призначений для знаходження координат безпілотного літального апарату (БПЛА), обладнаного однією бортовою відеокамерою і бортовим обчислювачем, без використання навігаційних сигналів GPS і ГЛОНАСС. Пропонований алгоритм схожий з широко відомими алгоритмами візуальної навігації, такими як одночасна локалізація і картографування (V-SLAM) і візуальна одометрія, однак відрізняється від них роздільним виконанням процесів картографування і локалізації. Він обчислює географічні координати ознак, знайдених на кадрах, знятих бортовою відеокамерою при польоті від точки старту. Після втрати сигналу запускається місія повернення і обчислюється лише положення БПЛА щодо побудованої на основі знайдених раніше ознак карти, яка використовується для повернення в точку старту. Запропонований підхід не вимагає таких складних обчислень, як V-SLAM, і не накопичує згодом помилки на відміну від візуальної одометра і традиційних методів інерціальної навігації. Алгоритм був реалізований і протестований за допомогою квадрокоптера DJI Phantom 3 Pro.

Всі вимоги, які наведені в технічному завданні, є достатніми для розробки програмного продукту.

1.2 А наліз методик вирішення завдань

Навігація літальних апаратів (аеронавігація), наука про методи та засоби водіння літальних апаратів з однієї точки простору в іншу, яка тісно пов'язана з такими дисциплінами, як теорія управління рухом ЛА, теорія фігури Землі і її гравітаційного і магнітного полів, геодезія. На використанні досягнень аеронавігації будується практич. літаководіння.

Основні завдання, які вирішує аеронавігація: попереднє визначення траєкторії руху об'єкта з урахуванням вимог пов. простору, можливостей наявного обладнання, передбачуваних зовнішніх умов і часу руху; визначення методів навігаційних вимірювань, вихідної і початкової навігаційної інформації і засобів корекції помилок вимірювань, які можуть бути найбільш ефективними в планованих умовах руху об'єкта; проведення вимірювань та обробка отриманої навігаційної інформації в процесі руху об'єкта; виявлення просторово-часових відхилень від заданої програми руху і вироблення команд для управління об'єктом. Перераховані завдання вирішуються з використанням різних методів навігації в бортовому обчислювальному комплексі.

Засоби навігації ЛА за принципом дії діляться на 4 групи: геотехнічні, астрономічні, радіотехнічні і світлотехнічні. Геотехнічні засновані на вимірі параметрів магнітного поля (магнітні компаси), поля земної атмосфери (барометричні висотоміри, вимірювачі повтряної швидкості), топографічні поля (навігаційні карти), поля оптичного контрасту (оптичні візери), гравітаційні поля (гравіметри). Особливо слід виділити групу гіроінерційних засобів навігації ЛА, заснованих на використанні гіроскопічного ефекту і вимірі сил інерції прискореного руху в сукупності з силою тяжіння (гіровертикалі, інерціальні системи навігації та ін.). Засоби астрономічної навігації (астрокомпаси, секстанти, астрономічні зоряно-сонячні орієнтатори) засновані на пеленгації (гл. Обр. Оптичної) небесних світил, дозволяють визначати географіч. координати і істинний курс ЛА. Радіотехнічні засоби представлені бортовими системами і наземними засобами (радіосистеми ближньої і дальньої навігації, радіотехніч. Системи посадки); до них відносяться і стрімко розвиваються глобальні супутникові навігаційні системи (напр., ГЛОНАСС). Світлотехнічні засоби Н. засновані на використанні бортових або наземних

джерел світла, гл. задача яких - полегшення орієнтування в складних метеорологічних. умовах і вночі (перш за все при посадці).

Існують 3 групи методів визначення місця розташування ЛА. До 1-ї групи належить метод числення координат, заснований на вимірюванні проєкцій швидкостей або прискорень і подальшому їх інтегруванні в тій чи іншій системі відліку для отримання координат розташування, а також інерційних методів навігації курсоповітряний і курсодоплеровский методи.

2-я група методів - позиційні методи навігації, засновані на вимірі фізичних величин, що дають лінію або поверхню в просторі, називаються лініями або поверхнями положення. Точка перетину ліній або поверхонь положення визначає місцезнаходження ЛА. Для цих цілей використовуються кошти астрономичної навігації та радіонавігаційні системи.

3-тя група методів - оглядово-порівняльні методи, засновані на порівнянні, реально спостерігається навколишньої місцевості (або окремих орієнтирів) і її зображення (або зображення орієнтирів) на карті, закладеної в пам'яті бортового комплексу.

Вибір методів визначається великим числом умов застосування: станом атмосфери, діапазоном зміни навігаційних параметрів, потрібної точністю їх вимірювання, вимогами перешкодозахищеності, автономності та надійності навігаційних вимірювань і ін. Залежно від довжини польотної траєкторії розрізняють ближню і дальню навігацію. Ближня навігація забезпечує переміщення ЛА в районі аеродрому, а також його зліт і посадку. Дальня навігація застосовується на маршрутах протяжністю сотні і тисячі кілометрів.

Всі перераховані вище методи навігації ЛА вирішуються з використанням сукупності великого числа параметрів, які в процесі польоту безперервно змінюються внаслідок переміщення ЛА, зміни умов навколишнього середовища і діючих зовнішніх збурень. Параметри навігації поділяються на групи: що характеризують положення центра мас ЛА щодо зовнішніх систем відліку (координати, лінійні швидкості і прискорення); що визначають рух ЛА щодо його центру мас (кутові прискорення, швидкості і кутові координати ЛА щодо горизонтного базису - тангаж, крен, курс; щодо вектора істинної повітряної швидкості - кут атаки, ковзання і знесення); що визначають стан навколишнього середовища (атмосфери - температура, тиск, щільність, швидкість вітру, швидкість звуку; геофізичних полів - напруженість магнітного поля, гравітаційний потенціал та ін.); параметри, що характеризують

відноситься рух і фізичні властивості орієнтирів, радіомаяків та ін. повітряних об'єктів.

Напрямки розвитку комплексів автономних БЛА рухається в бік підвищення автономності виконання цільових завдань. У 2010 р. ВВС США опублікували програму, що вимагає підвищення ступеня автономності в сучасних комплексах БПЛА (Рисунок 1.1). Високу роль в підвищенні автономності БПЛА грає рішення навігаційної завдання об'єктів. Міністерство оборони США приносить графіки, в яких підвищення автономності БПЛА пов'язано з розвитком навігаційної системи.

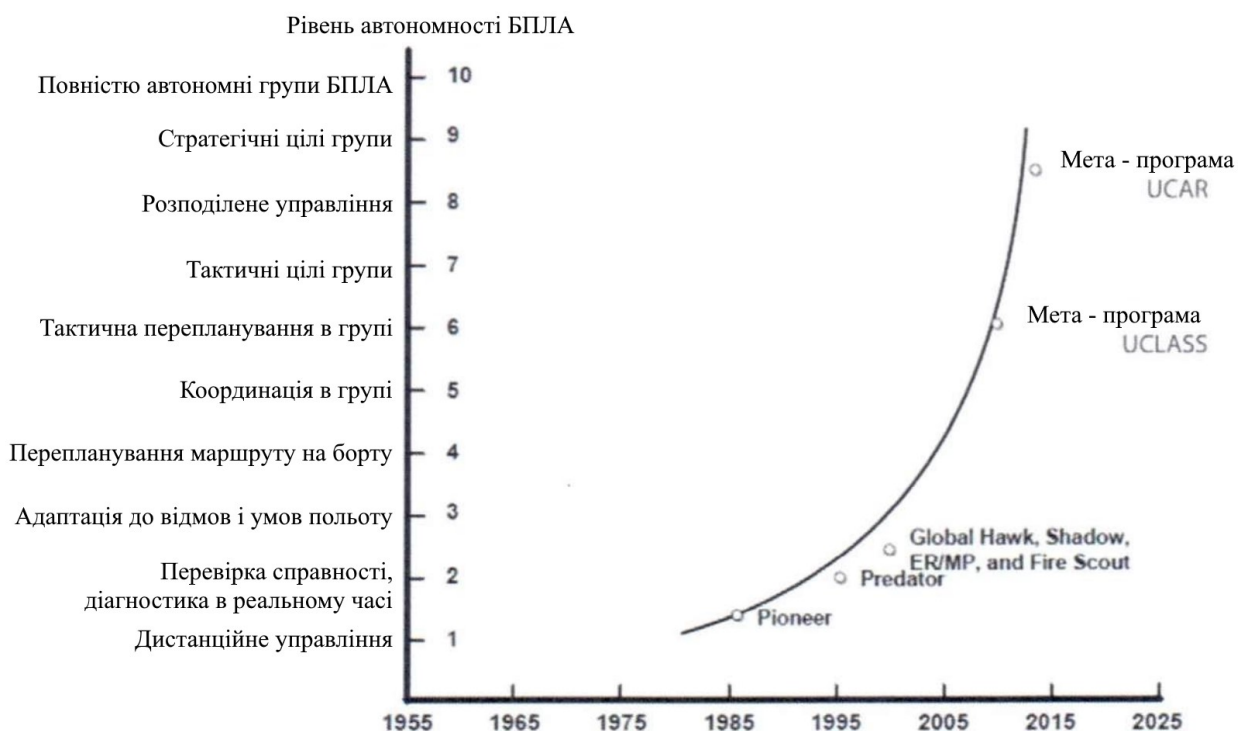


Рисунок 1.1 – Рівні автономності БПЛА

Навігаційні комплекси сучасних БПЛА при розрахунку вектора стану використовують сигнали СНС, а в якості сигналів управління - сигнали з пульта дистанційного керування. Автономні завдання, які вирішуються комплексами, дозволяють коптеру злетіти на висоту і стабілізувати власне кутове положення, виконати політ за маршрутом і здійснити зниження до торкання поверхні. При втраті сигналів СНС більшість сучасних БПЛА припиняє виконувати завдання і переходить в режим висіння (стабілізації кутового положення - крен, тангажу), таким чином виконання завдань стає неможливим зовсім.

Сучасні системи, вирішальні завдання навігації, об'єднуються з алгоритмами обробки в інтегрований інформаційно-керуючий комплекс.

Сучасні ПКК включає в себе в залежності від конфігурації такі системи:

- безплатформенні інерціальні навігаційні системи;
- приймач супутникової навігації GPS / ГЛОНАСС;
- лазерний локатор (далекомір), встановлений в зв'язаній системі координат з носієм;
- системи технічного зору (що складаються з гіростабілізованого фото (відео) камер і обчислювачів обробки).

Основними факторами, що визначають структуру навігаційного комплексу, є необхідна точність і надійність визначення навігаційних параметрів при заданих обмеженнях на масу, габарити і обчислювальні ресурси.

Комплексна система навігації.

Навігаційна завдання сучасного БПЛА складається в безпечному маневруванні на кінцеву точку маршруту з обходом виникаючих перешкод. Завдання маневрування ділиться на 2 рівня - високий (аналіз заданого маршруту і зіставлення з поточним місцем розташування) і низький (алгоритм керування силовою установкою). Пропонована схема рішення навігаційної завдання представлена на рисунку 1.2.

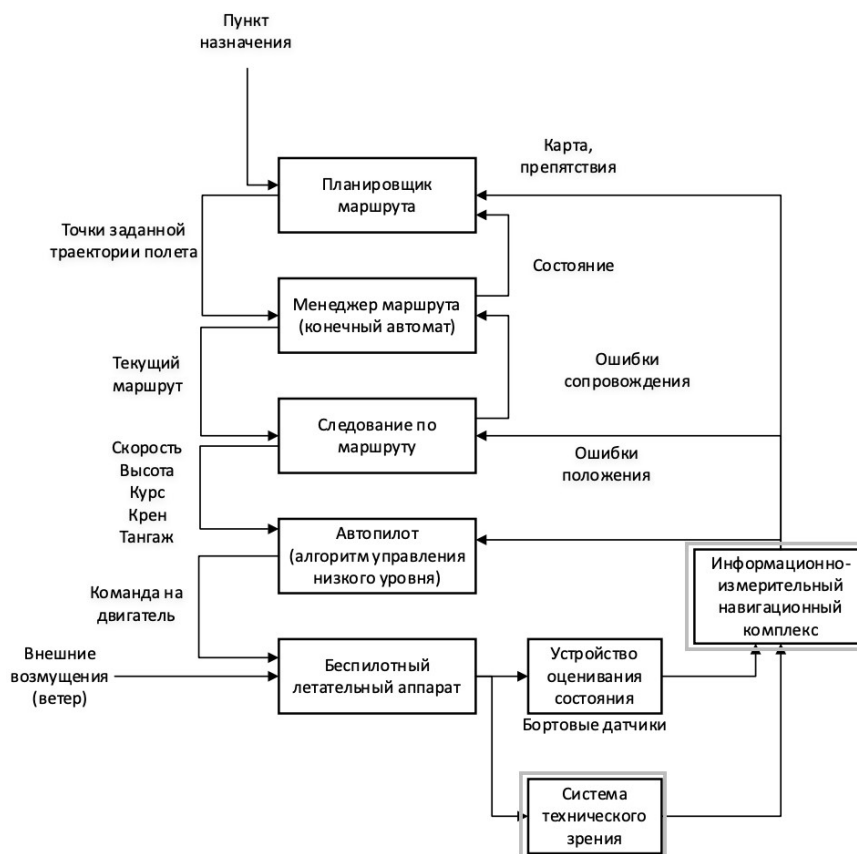


Рисунок 1.2 – Схема рішення навігаційної завдання

Інформаційно-вимірвальний навігаційний комплекс здійснює обчислення вектора стану безпілотного літального апарату ($X_c, Y_c, Z_c, \psi, \nu, \gamma$). Основу комплексної системи навігації виконує безплатформенна інерціальна навігаційна система (БІНС) в сукупності з 3-х осьовим магнітометром, так як є єдиним засобом отримання кутового положення БПЛА. Фільтр Махоні використовується для розрахунку кутів орієнтації БПЛА в просторі (кутів Ейлера-Крилова) за даними гіроскопів і фільтрації прискорень БПЛА, що надходять з акселерометрів в земній пов'язаній системі координат. Фільтр знімає проблеми точності і налаштування параметрів спостереження, апріорної і апостеріорної інформації фільтра Калмана. Обчислювальна складність алгоритму оптимізована за рахунок використання математичних розрахунків на основі кватерніонів, що дозволяє уникати труднощів, пов'язаних з описом просторового положення за допомогою кутів Ейлера [2].

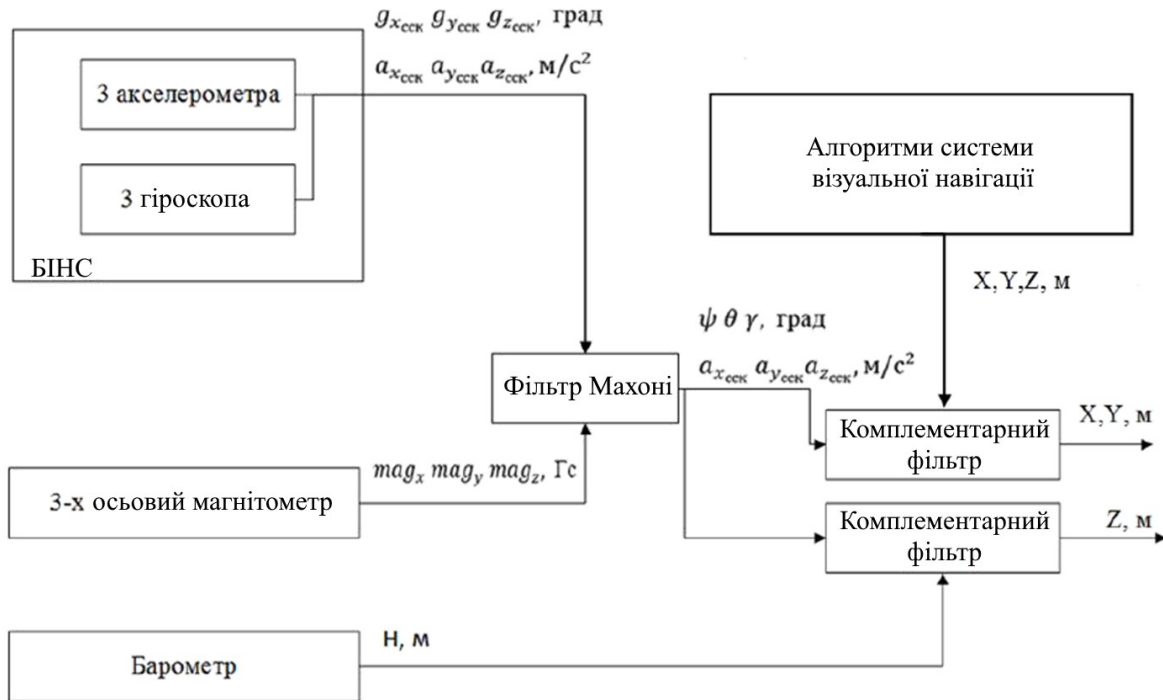


Рисунок 1.3 – Схема роботи інформаційно-вимірювального навігаційного комплексу

Використання засобів технічного зору продиктовано порівняно низькою вагою системи в порівнянні з активними системами володіють істотними енерговитратами і отже масо-габаритними параметрами. Система технічного зору призначена для визначення координат БПЛА у стартовій системі координат.

Система технічного зору складається з:

- відеокамери, розташованої на 2-х осьовому гіростабілізатори;
- бортового обчислювача, що виконує визначення місця розташування візуального орієнтиру щодо БПЛА.

Застосування комплексної системи навігації дозволяє виконати завдання автономного навігаційного числення без сигналів супутникової навігації, завдяки чому можливе використання БПЛА у вирішенні практичних завдань в автоматизованому режимі.

Відеотехнічних технології по використовуваних методів визначення орієнтації і положення БПЛА можна поділити на такі класи:

1. Наземний технічний зір (on-ground vision (OGV)).
2. Візуальна одометрія (visual odometry (VO)).

3. Навігація відносно мети (Target relative navigation (TGRN)).
4. Навігація відносно ландшафту / орієнтирів (Terrain/landmark relative navigation (TRN)).
5. Паралельна оцінка руху та структури: методи одночасної локалізації і побудови карти (Simultaneous localization and mapping (SLAM)) і отримання структури сцени при русі (Structure from motion (SFM)).
6. Навігація з використанням оптичного потоку на основі біонічних підходів (Bio-inspired optic flow navigation (BIOFN)).

1.2.1 Наземний технічний зір

У подібних системах камери і обчислювальні засоби, що здійснюють визначення орієнтації і положення апарату в локальній системі координат, розташовуються на землі. Переваги подібних систем - це відсутність обмежень на корисне навантаження і обчислювальні ресурси, можливість застосування складних алгоритмів видеонаблюдения. З іншого боку, мультікамерного системи (більше 2 камер) в основному використовуються для побудови стендів і дослідження динамічних властивостей БПЛА. Наземні системи мають досить високою точністю. Використання подібних систем для посадки на АСС представляє великий інтерес, особливо для БПЛА, не оснащених високоточними навігаційними системами. Розміщення системи технічного зору на землі звільняє від необхідності розміщення подібної системи на борту БПЛА, що збільшує його корисне навантаження. Крім того, наземна станція, оснащена такою системою, може приймати різні типи БПЛА [1].

1.2.2 Візуальна одометрія

Традиційна одометром використовує дані про переміщення виконавчих механізмів для оцінки змін стану та / або орієнтації в залежності від часу. Візуальна одометрія - це метод оцінки стану і орієнтації об'єкта за допомогою аналізу послідовних зображень, отриманих за допомогою встановленої на об'єкті камери. У загальному випадку координати об'єкта (БПЛА) щодо вихідного чи відомого розташування обчислюються шляхом підсумовування пройдених відстаней, отриманих при відстеженні ключових точок простору за допомогою бортової системи технічного зору. Алгоритм візуальної одометра включає наступні основні кроки:

- 1) детектування ключових точок зображення;

2) виявлення відповідності або трекінг ключових точок між двома послідовними зображеннями;

3) оцінка параметрів руху на основі зсуву ключових точок.

Основним недоліком візуальної одометра є дрейф показань в часі, тому, як правило, показання візуального одометра вимагають подання з іншими системами, що не мають дрейфу [1].

1.2.3 Навігація відносно мети

Навігація відносно мети передбачає детектування певної мети і оцінка її розташування щодо БПЛА. Ця відносна позиція може виражатися в метрах або пікселях. Потенційне застосування візуальних систем навігації щодо мети - це точне приземлення на задану ціль і відстеження рухомих повітряних / наземних цілей, таких як машини, люди або інші БПЛА. На відміну від методів візуальної одометра, в методах навігації щодо мети вводиться поняття «мета», яка повинна залишатися в полі зору камери. Ця мета може бути статичною або рухається, заздалегідь відомої або обраній під час польоту оператором або автоматичною системою візуального сприйняття. Системи навігації щодо мети, в свою чергу, можуть бути розбиті на чотири категорії:

- 1) системи візуальної посадки на відомому мета,
- 2) системи візуального відстеження статичної мети,
- 3) системи візуального відстеження мобільного цілі,
- 4) системи візуального приземлення на незнайому мета.

Системи візуальної посадки на відомому мета (платформу).

Платформа для посадки або мета спочатку розробляються таким чином, щоб спростити відеообробку і здійснювати точне обчислення параметрів руху з різних положень.

Візуальне відстеження нерухомих цілей.

У цих системах головне - це не посадка на мету, а політ у напрямку до мети або зависання над нею. У цьому випадку мета може бути не обов'язково на землі. Наприклад, вона може розташовуватися на стіні.

Системи візуального відстеження мобільного мети.

При навігації на основі технічного зору щодо рухомій цілі передбачається рух наземної або повітряної цілі. Проблема візуального трекінгу рухомій цілі полягає в складності обчислення швидкості цілі і швидкості літального апарату, тому що можливо визначення тільки відносної швидкості. При цьому алгоритм

обробки зображень повинен бути досить надійним для відстеження динамічної цілі на послідовності зображень [1].

1.2.4 Навігація щодо ландшафту/орієнтирів.

Даний вид навігації відноситься до методів оцінки координат і / або швидкості апарату шляхом порівняння вимірювань параметрів місцевості із заздалегідь відомою картою місцевості. Реалізація подібних методів може бути здійснена або пасивно за допомогою обробки зображень, або активно за допомогою засобів вимірювання відстані. В якості еталонної карти може використовуватися карта висот, супутникові знімки, топографічна карта з орієнтирами і т.д. Основне завдання навігації щодо ландшафту - це створення апаратури, у якій відсутній дрейф, що здійснюється шляхом оцінки координат (глобальних або локальних) або орієнтації щодо поверхневих орієнтирів. Як правило, для навігації щодо ландшафту / орієнтирів використовують пасивні системи з обробкою відеоінформації. Оскільки зміни висоти досить малі, то важко отримати гарну відповідність з профілем місцевості, отриманим за допомогою активних далекомірів.

1.2.5 Паралельна оцінка руху та структури: методи SLAM і SFM

Інша перспективна парадигма для визначення місця розташування гвинтокрилих апаратів і картування сцен - це методи SLAM і SFM. Між цими алгоритмами є велика схожість, оскільки обидва вони оцінюють тривимірні координати апарату, а також структуру або карту оточення. Ці методи мають більш високий рівень абстракції, оскільки крім визначення орієнтації і координат апарату здійснюють зорове сприйняття, хоча в деяких випадках вони використовують тільки систему візуальної одометра. Для методів SLAM і SFM характерні загальні невирішені проблеми: недостатня надійність відстеження в реальному масштабі часу, неоднозначність поворотів і переміщень, невизначений масштабний коефіцієнт, дрейф в часі, високі вимоги до обчислювальних ресурсів.

SLAM (Simultaneous localization and mapping) - це завдання одночасної локалізації і побудови карти. Принцип роботи SLAM - це повторювані дії. Спочатку відбувається збір даних з датчиків з метою отримання інформації про об'єкти, розташовані в навколишньої місцевості. Потім визначається власне відносне розташування на основі отриманої інформації і карти перешкод, яка містить попередні вимірювання. Визначивши позицію керованого об'єкта, система переходить до оновлення карти. Таким чином, на кожному кроці алгоритму система має припущення про структуру навколишньої місцевості (карти перешкод) і місцезнаходження керованого об'єкта щодо цієї карти.

SFM (Structure from motion) - це задача отримання структури сцени при русі. Дане завдання відноситься до визначення власного руху камери і реконструкції тривимірної структури сцени з її проекції на рухому двовимірну поверхню. SFM багато десятиліть залишався центральним методом комп'ютерного зору, і в літературі існує велика кількість порівнянь різних схем. Хоча SFM-методи взагалі досить широко використовуються, існує дуже мало робіт по їх застосуванню в реальному масштабі часу для БПЛА. Це пов'язано з тим, що дані методи вимагають великої обчислювальної потужності. Крім того, такі системи мають невисоку надійність в природних умовах [1].

1.2.6 Навігація з використанням оптичного потоку на основі біонічних підходів

У більшості підходів, що використовуються в системах технічного зору, координати апарату і / або його швидкість обчислювалися в явній формі і використовувалися польотним контролером. Деякі літаючі комахи використовують для навігаційних цілей якусь подібну оптичного потоку без безпосереднього визначення параметрів руху та структури оточення. Дійсно, літаючі комахи, такі як бджоли і мухи, використовують альтернативні, прості та ефективні стратегії використання 3D-зору для виконання навігаційних завдань, таких як орієнтація і стабілізація курсу, візуальна одометрія, контроль швидкості польоту, посадка, контроль висоти польоту і огибання рельєфу місцевості, обліт перешкод. Очевидно, що перспективні технічні системи, що використовують подібні стратегії на основі біонічних технологій, будуть виконувати ті ж функції, що і описані вище: визначення просторового положення апарату, зорове сприйняття навколишнього оточення, виявлення перешкод. Системи на основі біонічних технологій потенційно можуть забезпечити БПЛА вищим рівнем автономності [1].

Класифікація відеотехнічних систем посадки за критерієм розміщення відеокамер:

1. Наземні відеотехнічних системи посадки.
2. Бортові відеотехнічних системи посадки.

На додаток до них можна додати комбіновані відеотехнічних системи посадки, в яких камери розташовуються як на землі, так і на БПЛА.

Наземні відеотехнічних системи посадки.

При використанні наземної системи технічного зору камера (камери) розташовується на землі і використовується для відстеження апарату, при цьому здійснюється визначення його орієнтації і / або розташування в локальній системі координат. Перевага подібних систем по відношенню до бортовим відеотехнічних системам полягає у відсутності жорстких обмежень на апаратно-програмні засоби: є можливість побудови систем з застосуванням камер високої роздільної здатності та застосування складних алгоритмів відеонаблюдення, відсутні обмеження по масі, габаритам і енергоспоживання.

1.2.7 Бортові відеотехнічних системи посадки

На відміну від наземних систем, інший вид систем автоматичної посадки використовує бортове відеотехнічних обладнання. Якщо розглядати приземлення VTOL-апаратів (с вертикальним злетом / посадкою), то, як правило, використовуються спеціальні вертолітні майданчики.

1.2.8 Комбіновані відеотехнічних системи посадки.

В роботі [8] розроблена система стабілізації для моделі вертольота на основі технічного зору з використанням двох камер. Пропонується оригінальний метод визначення координат вертольота, при якому одна з камер розташована на борту вертольота, а інша знаходиться на землі і має похило-поворотний пристрій зі збільшувальним об'єктивом. Розроблено метод, що дозволяє обчислювати координати БПЛА за допомогою спрямованих один на одного камер. Цей метод можна використовувати для автоматичного зльоту і посадки, особливо у випадках, коли дуже важливі точні відносні координати, наприклад, при посадці на корабель в умовах сильної качки. В якості маркерів використовуються кольорові кульки радіусом 2,5 см, що приєднуються до нижньої частини квадрокоптера і до наземної камері. Система використовує алгоритм відстеження маркерів для визначення координат. Для прийнятної точності відстеження досить двох маркерів на квадрокоптера і одного на наземної станції [1].

У висновку можна сказати, що найбільш точними системами посадки БПЛА є наземні системи посадки. Переваги подібних систем - це відсутність обмежень на корисне навантаження і обчислювальні ресурси, можливість застосування складних алгоритмів відеонагляду.

Опис роботи системи візуальної одометра.

На рисунку 1.4 представлена блок-схема алгоритму неявно прямої візуальної одометрії (НПВО). Він використовує два паралельні потоки: один для оцінки пересування камери, інший для відображення карти і середовища моніторингу в режимі реального часу. Поділ на два потоки дозволяє розширювати карту простору незалежно від відстеження пересування, що займає певний час.

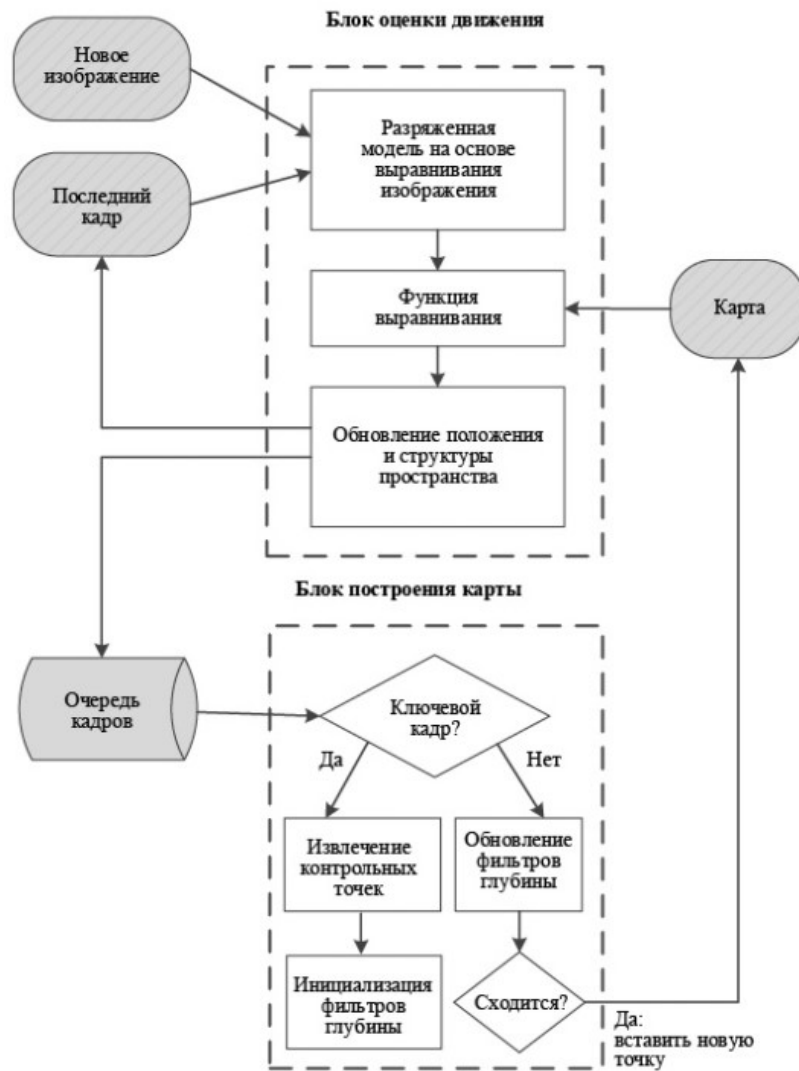


Рисунок 1.4 – Блок-схема работы алгоритма визуальной одометрии

Потік оцінки руху реалізує запропонований метод оцінки відносного зсуву камери. На першому кроці відбувається ініціалізація позиції камери за допомогою розрідженої моделі зображення, далі йде вирівнювання позиції камери щодо попереднього кадру, це відбувається через зведення фотометричної помилки до мінімуму, що відповідає прогнозованому тривимірному розташування одних точок на різних кадрах. Двомірні координати уточнюються на наступному етапі через вирівнювання характеристик наборів точок. Таким чином, оцінка руху спочатку знаходиться шляхом уточнення позиції і структури простору через зведення до мінімуму помилки перепроєціювання на попередньому кроці вирівнювання по контрольним точкам.

У другому потоці (відображення карти) імовірнісний фільтр глибини ініціалізується для кожної контрольної точки на площині з метою отримання її третьої координати. Фільтр глибини ініціалізується заново, коли обраний новий ключовий кадр і знайдено кілька відповідностей тривимірних координат попередніх контрольних точок з двомірними на новому ключовому кадрі. На кожному стежить кадрі оновлюються фільтри глибини. Згодом неточність фільтра глибини стає зовсім малою, тоді нова контрольна точка з трьома координатами відразу проставляється на карту і використовується для оцінки руху.

1.3 Аналіз існуючих програмних засобів

Порівняння роботи алгоритмів візуальної одометра

Більшість алгоритмів монокулярної візуальної одометра для малих БПЛА спираються на технологію паралельного відстеження та побудови карти (PTAM - Parallel Tracking and Mapping) [7]. PTAM, в свою чергу, базується на методі одночасної локалізації і побудови карти SLAM, який забезпечує надійність за рахунок відстеження та відображення сотень контрольних точок. Він працює в режимі реального часу і одночасно виконує завдання відображення пунктів призначення і оцінки пересування, спираючись на ефективну коригування, засновану на обробці зображень з різних кутів огляду. Проте, PTAM був розроблений для додатків доповненої реальності в невеликих просторах, але в декількох модифікаціях, наприклад, при обмеженні числа кадрів, що забезпечує його повноцінну роботу в умовах просторів із забудовою міського типу [2].

Пропонований алгоритм променя візуальної одометра використовує функцію знаходження контрольних точок. При цьому знаходження контрольних точок є неявним результатом прямої оцінки руху, ніж явного знаходження контрольних точок і зіставлення їх положення в кожному кадрі. Таким чином, витяг контрольних точок потрібно тільки тоді, коли ключовий кадр обраний для ініціалізації нових фільтрів глибини (тривимірних точок) (див. рис. 1). Швидкість роботи алгоритму істотно збільшується, коли контрольні точки витягуються не у всіх кадрах, а тільки в ключових. Збільшується точність і при використанні субпиксельної функції вилучення контрольних точок. На відміну від прямих методів використовується багато невеликих ділянок, а не кілька великих плоских ділянок, що позитивно

позначається на швидкості роботи алгоритму і його надійності. Пропонована розріджена модель зображення для оцінки і вирівнювання руху пов'язана з щільною моделлю зображення, проте далі буде показано, що розрідженій інформації про глибини досить, щоб отримати грубу оцінку руху і функції пошуку контрольних точок. Як тільки контрольні точки і початкове положення камери будуть знайдені, алгоритм стане використовувати лише контрольні точки, тим самим пояснюється його назва «променя». Цей прийом дозволяє швидко встановлювати рамки для обробки в новому зображенні.

Байєсівський фільтр, який в явному вигляді прибирає помилкові вимірювання, теж оцінює глибину і розташування контрольних точок. Тривимірна точка наноситься на карту тільки тоді, коли супутній фільтр глибини дав збіжність, що вимагає багаторазових вимірювань.

Результатом роботи є карта з контрольними тривимірними точками, чия надійність була перевірена. Наукова новизна даної роботи у винаході Напівпряма алгоритму візуальної одометра, який працює швидше і надійніше (точніше позиціонує БПЛА і не завдає на карту помилкові точки) встановлених на сучасних малих БПЛА, а також в інтеграції імовірного відображення точок на карті. Алгоритм стійкий до помилкових вимірів.

При завантаженні алгоритм оцінює початкову позицію по першим двом ключовим кадрам і початкової карті - локальному способу позиціонування БПЛА в просторі. Спочатку передбачається локально плоска схема простору і оцінюється початкове проєктивне перетворення. Вихідна карта при цьому визначається з перших двох кутів огляду методом триангуляції.

Щоб впоратися з великими пересуваннями, застосовується вирівнювання розрідженій моделі зображення за схемою від великого до малого за принципом піраміди. Зображення розбивається на два зразка для створення п'ятиуровневої піраміди зображення. Потім невязки інтенсивності оптимізуються на першому рівні піраміди зображення до збіжності. Згодом оптимізація відбувається і на наступних рівнях. Для економії часу обробки зупинка оптимізації відбувається після збіжності на третьому рівні: на цьому етапі оцінка достатня для ініціалізації функції вирівнювання.

Виходячи з міркувань ефективності алгоритм зберігає фіксоване число ключових кадрів на мапі, які потім використовуються в якості еталонних для функції вирівнювання по контрольним точкам і подрібнення структури. Ключовий кадр вибирається, якщо евклідова відстань нового кадру по відношенню до всіх ключових кадрів перевищує 12% від середньої глибини

сцени простору; коли новий ключовий кадр поміщається на карту, видаляється найбільш дальній від поточної позиції ключовий кадр.

У блоці побудови карти зображення ділиться на клітини, наприклад 30×30 . Новий фільтр глибини ініціалізується в кутку, який має максимальне значення рейтингу Ши-Томасі [8], якщо там ще немає відповідності двомірної і тривимірної карт. Це призводить до рівномірного розподілу контрольних точок по зображенню. Ця ж сітка використовується для перепроєціювання карти перед вирівнюванням. Варто звернути увагу, що кути з максимальним значенням рейтингу Ши-Томасі витягуються на кожному рівні піраміди зображення для знаходження найкращих кутів незалежно від масштабу.

Для порівняння результатів роботи використовувався модифікований алгоритм РТАМ [9]. Це пов'язано з тим, що оригінальний алгоритм недостатньо надійний в високочастотних повторюваних структурах, модифікована версія справляється з цим завданням краще. Алгоритм НПВО показав високу точність і ефективність, частково це пояснюється застосуванням фільтром глибини, який відсіває велику частину помилкових вимірювань і положень тривимірних точок. Також даний фільтр забезпечує високу надійність і точність при монотонній, але високочастотній текстурі (трава, асфальт) в силу високої частоти оновлення кадрів. На малюнку 1.5 показана робота алгоритму як в звичайних умовах, так і в умовах високочастотної монотонної структури.



Рисунок 1.5 – Приклади роботи алгоритму в різних умовах

На рисунку 1.6 наведено порівняння роботи алгоритмів НПВО і РТАМ. РТАМ дає дуже багато помилкових вимірів, що може привести до невиконання місії. Варто зауважити, що саме цей алгоритм зараз застосовується на більшій частині малих БПЛА, таким чином, перед алгоритмом НПВО відкриваються широкі перспективи застосування.

Експерименти проводилися на наборах даних, одержуваних з камери, прикріпленої до малого БПЛА і дивиться вниз. Відео з камери оброблялося на Одноплатний комп'ютері Odroid-U2 [10], який був встановлений безпосередньо на БПЛА, а також транслювалося на зовнішній комп'ютер для обробки. Варто зауважити, що в результаті експерименту алгоритм працював з двома різними режимами налаштувань. На Одноплатний комп'ютері були виставлені настройки, що дозволяють зменшити швидкість роботи, на зовнішньому комп'ютері настройки були орієнтовані на високу точність. Параметри настройки дано в таблиці 1. Для подальшого викладу один режим назвемо швидким, інший - точним.

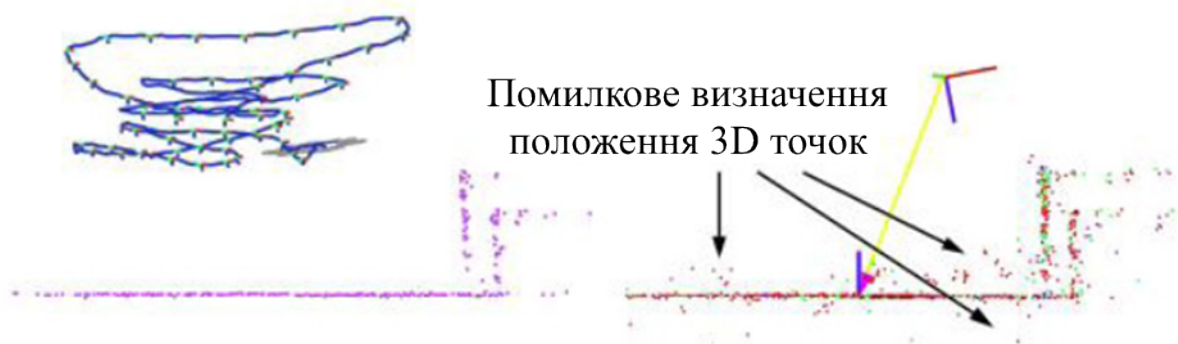


Рисунок 1.6 – Порівняння роботи НПВО і ротами

Класична обчислювальна схема візуальної одометра ґрунтується на зіставленні особливостей на знімках, побудові по ним 3D-хмар з подальшою роботою ICP (Iterative Closest Point) алгоритму, безпосередньо обчислює зміну положення і орієнтації камери.

На сьогодні існує досить велика кількість алгоритмічних рішень в цьому напрямку (з деяким оглядом можна ознайомитися в [1, 2], раніше отримані авторами результати описані в [3, 4]) і програми, доступні для загального користування (бібліотеки OpenCV, PCL і ін.). Разом з тим зберігається актуальність вдосконалення таких алгоритмів, оскільки вузьким місцем залишається проблема накопичення з часом помилки навігації, не завжди

прийнятна швидкість обчислень, неточність калібрування, зашумленість вихідних реальних даних і можлива специфіка практичних застосувань.

Пропонується нова реалізація візуальної одометра по стереоснимку. Її відмінною рисою є оригінальна обчислювальна схема на етапі зіставлення особливостей на знімках і побудови 3D-хмар, спрямована на підвищення точності відновлення траєкторії і оптимізації обчислювальних витрат.

Постановка задачі. У традиційній обчислювальній схемою при обчисленні параметрів позиції траєкторії (6DOF) по стереоснимку враховується тільки локальна інформація, що відноситься до поточної і попередньої позиції камери. У даній роботі ставиться завдання підвищення точності обчислення параметрів траєкторії за рахунок більш повного врахування відеоінформації на ділянці траєкторії, що передує оброблюваній позиції.

Пропонується алгоритм обчислення локального геометричного перетворення в позиціях траєкторії AP / камери на основі тривалого простеження трекером KLT (Kanade-Lucas-Tomasifeaturetracker) особливостей на знімках, що, імовірно, має підвищити точність локалізації в порівнянні з традиційним алгоритмом.

Розробці алгоритму обчислення локального геометричного перетворення передував етап дослідження з оцінки якості (точності) простежуються на знімках особливостей в залежності від часу їх «життя». В експериментах вимірювалася помилка локалізації в кінці першого кроку для особливостей з різним часом «життя», а також результуюча точність локалізації робота в поточній позиції траєкторії і час рахунку. Як показали експерименти, точність простеження особливостей з великим часом «життя» вище, ніж у тих, які простежуються тільки в межах 1-2-х початкових кроків.

З урахуванням цього результату були розроблені і проаналізовані кілька алгоритмів по обчисленню локального геометричного перетворення на основі тривалого простеження трекером KLT.

Алгоритм 1. Введено поняття фрагмента траєкторії довжиною m кроків.

Крок дорівнює q кадрів. Параметри q і m є налаштовувальними. Генерація вихідного безлічі особливостей виконується на початку кожного фрагмента (на 1-му кроці). Відстеження особливостей здійснюється покроково - таким чином, що для простеження на черговому кроці фрагмента використовується безліч особливостей, прослеженное на попередньому кроці (рис. 1.7).

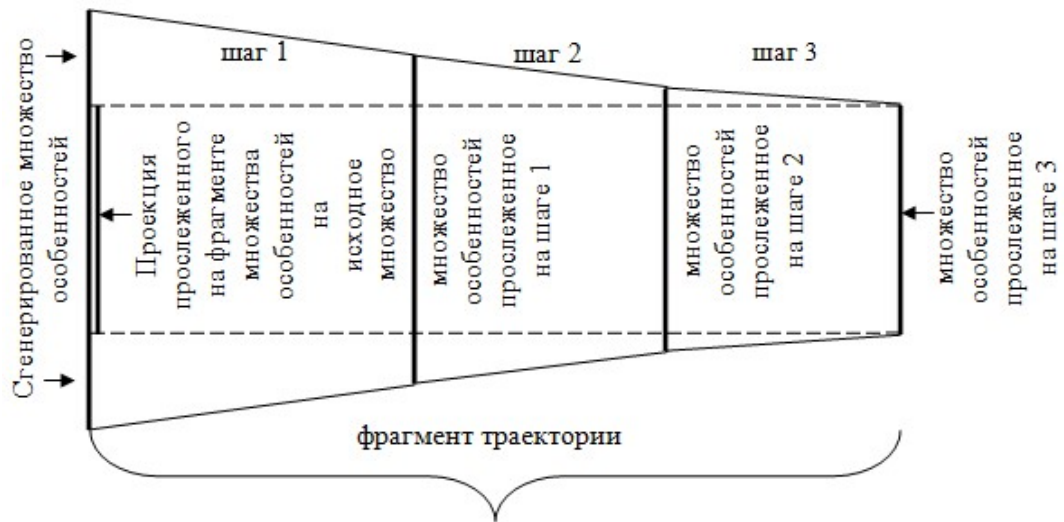


Рисунок 1.7 – Відстеження особливостей на фрагменті траєкторії

Зіставлення на знімках стереопари виконується детектором SURF, а простеження на знімках сусідніх позицій - з допомогою трекара KLT (рис. 1.8).

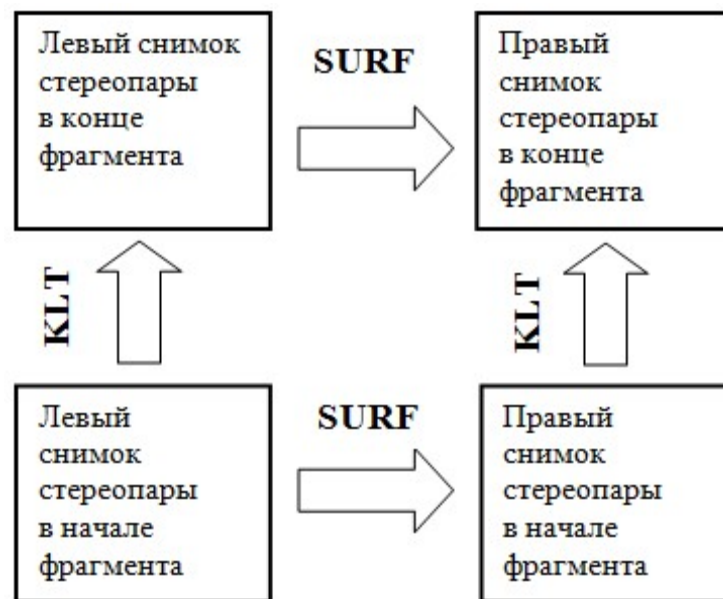


Рисунок 1.8 – Схема обробки знімків двох стереопар

У порівнянні з традиційною обчислювальною схемою це, по-перше, як було зазначено вище, знижує помилку локалізації і, по-друге, скорочує час обчислень в цілому за рахунок збільшення довжини оброблюваних ділянок (фрагменти) траєкторії.

Алгоритм 2. В даному алгоритмі на відміну від попереднього матриця геометричного перетворення для фрагмента виконується об'єднанням локальних матриць, що обчислюються для окремих кроків фрагмента (рис. 1.9).

На 1-му кроці генерується вихідне безліч особливостей, яке простежується трекером до кінця фрагмента. Прослеженное в результаті безліч проектується на вихідне і на прослеженное на 1-му кроці безліч.

За відповідними двом 3D хмар обчислюється матриця H_1 для 1-го кроку. На 2-му і 3-му кроках генеруються свої вихідні безлічі особливостей.

Аналогічним чином обчислюються матриці H_2 і H_3 . Результуюча матриця перетворення виходить множенням локальних матриць: $H_{\text{фрагмента}} = H_1 \cdot H_2 \cdot H_3$.

Далі виконується обробка наступного фрагмента. Схеми обробки в режимах online та offline розрізняються, оскільки в режимі online обробка синхронізується з надходженням знімків у міру руху камери.

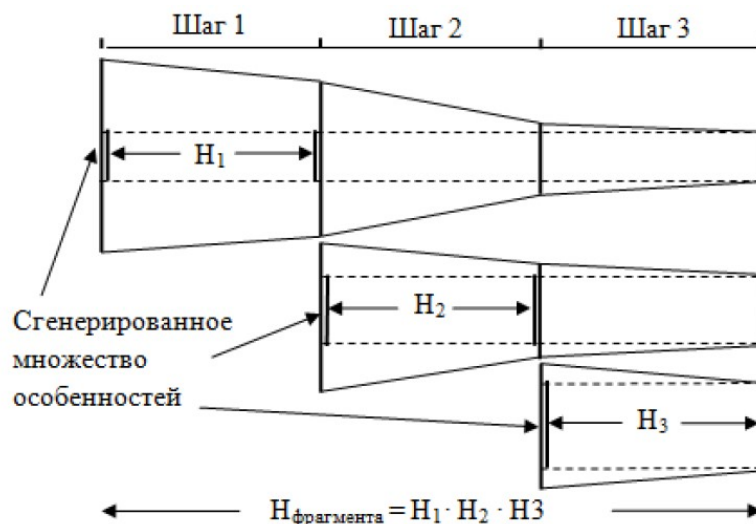


Рисунок 1.9 – Обчислення матриці перетворення для фрагмента, що складається з трьох кроків

Порівняння за результатами обчислювальних експериментів алгоритмів 1 і 2 показали помітну перевагу алгоритму 1. Однак слід зауважити, що для обох розглянутих варіантів характерна обробка траєкторії довгими ділянками, що не завжди прийнятно, коли параметри траєкторії робота необхідно обчислювати з високою частотою. Тому був розроблений алгоритм 3, що усуває цей недолік і, по суті, є розширенням алгоритму 1.

Алгоритм 3. Даний алгоритм є розвитком алгоритму 1, спрямованим на обчислення параметрів траєкторії з частотою, що відповідає частоті зйомки. Розрахунок траєкторії також ґрунтується на обробці фрагментів з означеної вище схемою (рис. 1), але при цьому використовуються кілька ланцюжків фрагментів (рис. 4), кожна з яких зміщена по відношенню до попередньої на один крок.



Рисунок 1.10 – Обробка траєкторії з трьома ланцюжками фрагментів

Абсолютна матриця виходить, як завжди, перемноженням локальних матриць фрагментів. Відмінністю тут є те, що для першої ланцюжка початковий фрагмент прив'язується до МСК, а для початкових фрагментів інших ланцюжків потрібно додатково обчислювати матриці прив'язки до МСК – $H_{0,k-1}$, де k – номер ланцюжка.

Організовується m таблиць по числу ланцюжків, або, що ж саме, по числу кроків у фрагменті. У кожену таблицю заносяться результати обробки поточного фрагмента відповідної ланцюжка. Обробка полягає в прослеживанні особливостей протягом фрагмента з подальшою фільтрацією. Після завершення обробки чергового фрагмента обчислюється матриця локального перетворення, яка потім множиться на накопичену абсолютну матрицю для цього ланцюжка. В результаті отримуємо абсолютну матрицю для поточної позиції траєкторії і обчислюємо координати позиції. Після обчислення абсолютної матриці починається обробка наступного фрагмента в ланцюжку. Таким чином, кожен черговий крок завершує фрагмент однієї з ланцюжків і починає новий фрагмент в цьому ланцюжку. Номер ланцюжка можна визначити як цілий залишок від ділення $k = i / m$, де i - номер кроку; m - кількість кроків у фрагменті.

Оскільки обробка фрагментів в даному алгоритмі виконується таким же чином, як в алгоритмі 1, то і точність навігації забезпечується на такому ж рівні. Однак обчислювальні витрати будуть вище приблизно в m раз.

Для оцінки ефективності запропонованого алгоритму були проведені обчислювальні експерименти як на модельних сценах, так і з реальними даними. Оцінці підлягали точність визначається локалізації робота і обчислювальна продуктивність алгоритму. Алгоритм порівнювався з двома реалізаціями класичної обчислювальної схеми візуальної одометра. У першій реалізації використовувався детектор SURF, в другій - трекер KLT (в наведених нижче таблицях вони позначені відповідно як "Базовий SURF» і «Базовий KLT»).

Запропонований алгоритм візуальної одометра по стереозображення, заснований на прослеживанні особливостей на декількох фрагментах траєкторії із застосуванням фільтрації «коротко живуть» особливостей, дозволяє істотно підвищити точність обчислення траєкторії автономного робота і скоротити час розрахунку в порівнянні з традиційною обчислювальною схемою. У розвиток цього підходу планується додатковий облік оптичного потоку при зіставленні особливостей, що також може перешкодити накопиченню помилки.

Аналіз алгоритмів SLAM-навігації.

У літературі існують два типи характеристичних методів: засновані на фільтрах і методи засновані на ВА. У цьому розділі порівнюються обидва методи. Незважаючи на те, що деякі з методів були запропоновані до 2010 року, їх можна розглядати як фундаментальні структури для інших методів.

1.3.1 MonoSLAM

Перший монокулярний vSLAM був розроблений в 2003 році Девісон і ін. [26, 27]. Вони назвали його MonoSLAM. MonoSLAM розглядається як типовий метод в алгоритмах vSLAM на основі фільтра. У MonoSLAM руху камери і тривимірної структури невідомої середовища оцінюється одночасно з використанням розширеного фільтра Калмана (EKF). Шість ступенів свободи камери (DOF) і тривимірні положення особливих точок представлені у вигляді вектора стану в EKF. Рівномірний рух приймається в моделі прогнозування, а результат відстеження точки об'єкта використовується в якості спостереження. Залежно від руху камери до вектору стану додаються нові особливі точки. Зверніть увагу, що початкова карта створюється шляхом спостереження за відомим об'єктом, де визначена глобальна система координат. Таким чином, MonoSLAM складається з наступних компонентів:

- 1) ініціалізація карти виконується з використанням відомого об'єкта;
- 2) рух камери і тривимірні позиції точок об'єкта оцінюються з використанням ЕКФ.

Проблема цього методу - обчислювальна вартість, яка збільшується пропорційно розміру середовища. У великих середовищах розмір вектора стану стає великим, оскільки кількість особливих точок велике. В цьому випадку важко робити обчислення в реальному часі.

1.3.2 PTAM

Щоб вирішити проблему обчислювальної вартості в MonoSLAM, PTAM [15] розділив відстеження та картографування на різні потоки CPU. Ці два потоки виконуються паралельно, так що обчислювальна вартість картографування не впливає на відстеження. В результаті в картографуванні може використовуватися ВА, що вимагає обчислювальних витрат при оптимізації. Це означає, що відстеження оцінює рух камери в реальному часі, а картографування оцінює точні тривимірні позиції точок з обчислювальною вартістю. PTAM - це перший метод, який включає ВА в алгоритми vSLAM реального часу. Після публікації PTAM більшість алгоритмів vSLAM використовують цей тип багатопоточних підходів.

У PTAM вихідна карта відновлюється з використанням п'ятиточкового алгоритму [28]. При відстеженні відображаються точки проектується на зображення, для створення 2D-3D відповідності, використовуючи зіставлення текстур. З відповідностей можуть бути обчислені положення камери. При зіставленні тривимірні позиції нових точок функції обчислюються з використанням тріангуляції в певних кадрах, які називаються ключовими кадрами. Одним із значних вкладів PTAM є впровадження картографування на основі ключового кадру в VSLAM. Вхідний кадр вибирається як ключовий кадр, при вимірюванні великого невідповідності між вхідним кадром і одним з ключових кадрів. Для точної тріангуляції потрібна велика диспропорція. На відміну від MonoSLAM, 3D-позиції особливих точок оптимізуються з використанням Local ВА з деякими ключовими кадрами, і глобальним ВА з усіма ключовими кадрами на карту. Крім того, в процесі відстеження більш нове бачення PTAM використовує алгоритм релокалізації. Він використовує рандомізований деревовидний класифікатор властивостей для пошуку найближчого ключового кадру вхідного зображення. Таким чином, PTAM складається з наступних чотирьох компонентів:

- 1) ініціалізація карти виконується за допомогою п'ятиточкового алгоритму;
- 2) положення камери оцінюються за співпадаючими особливим точкам між точками карти і вхідним зображенням;
- 3) тривимірні позиції точок об'єкта оцінюються шляхом триангуляції, а оцінені 3D-позиції оптимізуються ВА;
- 4) процес відстеження відновлюється рандомізованих деревовидним пошуком.

У порівнянні з MonoSLAM, в PTAM, система може обробляти тисячі особливих точок, розділяючи відстеження та картографування на різні потоки в процесорі.

Було запропоновано безліч розширених алгоритмів PTAM. Команда розробників Castle розробила багатопланову версію PTAM. Klein розробила версію PTAM для мобільного телефону. Для запуску PTAM на мобільних телефонах, було зменшено дозвіл вхідного зображення, кількість точок карти і ключових кадрів. Крім того, вони враховують спотворення поворотного затвора в ВА для отримання точного результату оцінки, тому що поворотний затвор, через його дешевою вартості, зазвичай встановлюється на більшості мобільних камер. Оскільки PTAM може відновлювати тільки розріджену тривимірну структуру середовища, третій потік можна використовувати для відновлення щільною тривимірної структури середовища.

Різниця між картографування на основі EKF в MonoSLAM та картографування на основі ВА з ключовими кадрами в PTAM обговорювалося в роботі. Відповідно до джерела, для підвищення точності vSLAM важливо збільшити кількість точок на карті. З цієї точки зору підхід на основі ВА краще, ніж підхід на основі EKF, оскільки він дозволяє обробляти велику кількість точок.

1.3.3 Методи оптимізації глобальної карти

Геометрична послідовність всієї карти підтримується з використанням ВА для ключових кадрів, як описано вище. Однак, в загальному, ВА схильний проблеми локального мінімуму через великої кількості параметрів, включаючи положення камери ключові кадрів і точки на карті. Оптимізація графа положення - це рішення, що дозволяє уникнути цієї проблеми при замиканні петлі, як описано в розділі 2. При замиканні петлі спочатку оптимізуються положення камери, використовуючи обмеження циклу. Після оптимізації

положень камери виконується ВА для оптимізації як тривимірних позицій точок об'єкта, так і положення камери. Для замикання петлі використовується підхід, заснований на візуальній інформації. Вони використовували метод пошуку зображень на основі «bag-of-words», щоб виявити один з ключових кадрів, зображення якого аналогічно поточному кадру.

В системі VSLAM в якості візуального датчика вибирається стереокамера. В цьому випадку шкала системи координат є фіксованою і відомою. Однак в монокулярних випадках VSLAM існує невизначеність масштабу, і масштаб може змінюватися під час руху камери, якщо глобальна ВА не виконується. В цьому випадку виникає проблема зміщення масштабу, і масштаб системи координат в кожному кадрі може не збігатися. Щоб виправити зміщення масштабу, положення камери слід оптимізувати з 7 ступенями свободи. Розробники з Strasdat Group запропонували метод оптимізації позицій камери з 7 ступенями свободи на основі перетворення подібності.

Як розширення PTAM, ORB-SLAM включає в себе ВА, візуальне виявлення замкнутих петель і оптимізацію графа положення з 7 ступенями свободи. Таким чином, ORB-SLAM є найповнішою функціональною монокулярною системою VSLAM. ORB-SLAM поширюється на стерео VSLAM і RGB-D VSLAM [39].

Прямі методи. На відміну від функціональних методів прямі методи безпосередньо використовують вхідне зображення без будь-якої абстракції з використанням рукописних функціональних детекторів і дескрипторів. Загалом, фотометрична послідовність використовується як похибка вимірювання в прямих методах, тоді як геометрична послідовність, така як положення особливих точок на зображенні, використовується в характеристичних методах. У цьому розділі ми представимо деякі провідні прямі методи.

1.3.4 DTAM

Newcombe запропонував повністю прямий метод, званий DTAM. У DTAM відстеження виконується шляхом порівняння вхідного зображення з синтетичними зображеннями кадрів, згенерованих з відновленої карти. Це просто еквівалентно реєстрації між зображенням і 3D-моделлю карти і ефективно реалізується на графічному процесорі в DTAM. Картографування виконується з використанням багатоосової стерео, а потім карта оптимізується з

урахуванням безперервності простору, таким чином, щоб можна було обчислити тривимірні координати всіх пікселів. Первісна карта глибини створюється з використанням стереометричного вимірювання, такого як PTAM. Таким чином, DTAM складається з наступних трьох компонентів:

- 1) ініціалізація картки здійснюється стереометричним виміром;
- 2) рух камери оцінюється за допомогою синтетичної генерації кадрів з відновленої карти;
- 3) інформація про глибину оцінюється для кожного пікселя з використанням багатоосові стерео, а потім оптимізується з урахуванням безперервності простору.

Алгоритм DTAM оптимізований для забезпечення обробки в реальному часі на мобільних телефонах. В принципі, ці методи призначені для швидкого і онлайн-3D-моделювання.

Слід зазначити, що група програмістів Stühmer раніше запропонували варіаційний підхід для оцінки інформації про глибину для кожного пікселя. Вони використовують аналогічну вагову функцію для відображення як і в DTAM. Однак, в цьому методі PTAM [15] використовувався для відстеження.

1.3.5 LSD-SLAM

LSD-SLAM є іншим провідним методом серед прямих методів. Основна ідея LSD-SLAM випливає з ідеї полуплотной VO [20]. У цьому методі восстанавливаемое об'єкти обмежені областями з градієнтом яскравості в порівнянні з DTAM, який відновлює повні області. Це означає, що він ігнорує області без текстур, оскільки важко оцінити точну інформацію про глибину по зображенню. При зіставленні, початкові значення глибини для кожного пікселя спочатку встановлюються як випадкові значення, а потім ці значення оптимізуються на основі фотометричної послідовності. Оскільки цей метод не враховує геометричну послідовність всієї карти, цей метод називають візуальною одометр.

У 2014 році полуплотная VO була поширена в LSD-SLAM [21]. У LSD-SLAM виявлення замикання петлі і оптимізацій графа положення з 7 ступенями свободи, як описано в попередніх розділах, додається до алгоритму полуплотной візуальної одометра [20]. Таким чином, LSD-SLAM складається з наступних чотирьох компонентів:

- 1) випадкові значення встановлюються як початкові значення глибини для кожного пікселя;

- 2) рух камери оцінюється за допомогою синтетичної генерації зображень з відновленої карти;
- 3) реконструйовані області обмежені областями градієнта високої яскравості;
- 4) для створення геометричних послідовної карти використовується оптимізація геометрії 7 DOF.

Ці полуплотние підходи можуть забезпечити обробку в реальному часі на процесорі. Крім того, вони оптимізували алгоритм LSD-SLAM для мобільних телефонів, з огляду на їх архітектуру ЦП. В роботі вони також оцінювали точність алгоритму LSD-SLAM для вхідних зображень з низьким дозволом. LSD-SLAM поширюється на стереокамери і односпрямовані камери.

1.3.6 SVO і DSO

Команда розробників Forster Group запропонували полупрямую VO (SVO) [51]. Хоча відстеження виконується за допомогою зіставлення точок, зіставлення виконується прямим методом. У функціональних методах для визначення відповідностей використовуються дескриптори функцій і трекер Lucas-Kanade [52]. На відміну від функціональних методів, рух камери оцінюється шляхом мінімізації фотометричних помилок, пов'язаних з об'єктами. Цей метод можна розглядати як рідкісну версію DTAM і LSD-SLAM.

Розробники з Engel Group запропонували пряму розріджену одометр (DSO) [53]. На відміну від SVO, DSO є повністю прямим методом. Щоб придушити накопичувану помилку, DSO максимально розділяє фактори помилок з геометричною і фотометричною точок зору. У DSO вхідне зображення ділиться на кілька блоків, а потім в якості кандидатів на реконструкцію вибираються точки високої яскравості. Використовуючи цю стратегію, точки поширюються по всьому зображенню. Крім того, для досягнення високоточної оцінки DSO використовує як геометричні, так і фотометричні результати калібрування камери. Слід зазначити, що DSO розглядає тільки локальну геометричну послідовність. Тому DSO класифікується як VO, а не vSLAM.

1.3.7 RGB-D vSLAM

Останнім часом RGB-D камери, що використовують структурований світ, такі як Microsoft Kinect, стають дешевшими і маленькими. Оскільки такі камери дають тривимірну інформацію в режимі реального часу, ці камери також використовуються в алгоритмах vSLAM.

Використовуючи камери RGB-D, можна отримати безпосередньо тривимірну структуру середовища з її текстурної інформацією. Крім того, на відміну від монокулярних алгоритмів VSLAM, масштаб системи координат відомий, оскільки тривимірну структуру можна отримати в метричному просторі.

Базова структура VSLAM на основі глибини (D) полягає в наступному. Для оцінки руху камери широко використовувався ітеративний алгоритм найближчих точок (ICP). Потім тривимірна структура середовища відновлюється шляхом об'єднання кількох карт глибини. Щоб включити RGB в VSLAM на основі глибини, було запропоновано багато підходів.

Слід зазначити, що більшість призначених для користувача камер для вимірювання глибини розроблені для використання в приміщеннях. Вони проєктують ІК-шаблони на навколишнє середовище для вимірювання глибини. Важко знайти випускаються ІК-зображення поза приміщеннями. Крім того, існує обмеження в діапазоні вимірювання глибини, оскільки RGB-D датчики можуть охоплювати навколишнє середовище в діапазоні від 1 до 4 м.

1.3.8 KinectFusion

Розробники з Newcombe Group запропонували KinectFusion в 2011 році. У KinectFusion використовується простір вокселів для подання тривимірної структури середовища. Тривимірна структура середовища реконструюється шляхом об'єднання отриманих карт глибини в воксельного просторі, а рух камери оцінюється за алгоритмом ICP з використанням вже оціненої тривимірної структури і вхідної карти глибини, яка є VSLAM заснованим на використанні даних глибини. KinectFusion реалізований на GPU для досягнення обробки в реальному часі.

Розробники з Kahler Group реалізували обробку KinectFusion в реальному часі на мобільних пристроях. Щоб зменшити обчислювальну вартість, в процесі картографування вони використовують хешування воксельного блоку. RGB-D vSLAM обробляє величезний обсяг даних, що гальмує швидкість роботи

алгоритму. Зменшення кількості даних відбувається шляхом об'єднання копланарних точок.

1.3.9 SLAM ++

Команда Salas-Moreno запропонував алгоритм VSLAM RGB-D на рівні об'єктів. У цьому методі в базу даних, заносяться кілька 3D-об'єктів, і ці об'єкти розпізнаються в режимі онлайн. Оціночна карта уточнюється, шляхом розпізнавання 3D-об'єктів, де 3D-точки замінюються 3D-об'єктами, щоб зменшити обсяг даних.

Як аналогічного алгоритму команда розробників Tateno запропонувала для RGB-D SLAM метод сегментації в реальному часі. Сегментовані об'єкти позначаються, а потім дані об'єкти можуть використовуватися в якості розпізнаються цілей.

1.3.10 Методи RGB-DVO і глобальної оптимізації карт

Для відстеження, RGB зображення також використовуються в алгоритмах RGB-D vSLAM. Відносний рух камери оцінюється шляхом відстеження особливих точок між послідовними кадрами. Матриця переходу потім оцінюється з використанням відслідковуються особливих точок, і ця матриця переходу уточнюється за допомогою алгоритму ICP з використанням карт глибини. З іншого боку, були запропоновані методи відстеження руху камери на основі фотометричної послідовності [22, 23]. Це фотометрическое відстеження руху камери, також використовується для створення щільних методів vSLAM на основі монокулярних камер [20, 21].

Для отримання геометрично послідовної карти в алгоритмах RGB-D VSLAM використовуються оптимізація графа положення і оптимізація графа деформації. У Kerl Group використовували оптимізацію графа положення для зменшення накопичуваної помилки [23]. Ця оптимізація графа положення майже така ж, як замикання петлі в монокулярних алгоритмах VSLAM. Програмісти в Whelan Group використовували оптимізацію графа положення для оптимізації руху камери і оптимізації графіка деформації для уточнення карти, відповідно. На відміну від інших робіт [23], розрахункова карта також уточнюється. Оптимізація графа деформації часто використовується для певних кадрів, а рух камери оцінюється шляхом зіставлення зображення RGB-D з відновленої моделлю. Вони показали геометрично послідовну модель, яка може бути отримана з використанням оптимізації графа деформації якомога частіше.

Зверніть увагу, що API-інтерфейси RGB-D SLAM надаються в таких користувальницьких пристроях, як Google Tango2 і Structure Sensor3. Зокрема, Google Tango забезпечує стабільний результат оцінки шляхом об'єднання всієї інформації з датчика.

Як зазначено вище, структура алгоритмів VSLAM складається з п'яти модулів: ініціалізація, локалізація, картографування, релокалізація і глобальна оптимізація карт. Оскільки кожен алгоритм vSLAM використовує різні методології для кожного модуля, особливості алгоритму vSLAM сильно залежать від використання даних методологій. Тому важливо розуміти кожен модуль алгоритму vSLAM, щоб знати його продуктивність, переваги та обмеження.

Слід зазначити, що відстеження і картографування (ТАМ) використовується замість використання локалізації та картографування. ТАМ вперше було використано в Parallel Tracking and Mapping (PTAM) [15], оскільки локалізація і картографування не виконуються одночасно в традиційному способі. Відстеження виконується в кожному кадрі з одним потоком, тоді як картографування виконується в певний момент часу з іншим потоком. Після того, як був запропонований PTAM, більшість алгоритмів VSLAM успадковують структуру ТАМ. Тому в цій роботі розглядається ТАМ.

На рисунку 1.11 наведено короткий опис функціональних методів. MonoSLAM був розроблений в 2003 році [26]. Як відстеження, так і картографування реалізується послідовно і одночасно з використанням EKF. PTAM був розроблений в 2007 році [15]. Вони запропонували розділити відстеження та картографування на різні потоки на процесорі. Цей багато-підхід дозволяє обробляти тисячі точок на карті. У великій навколишньому середовищу важко досягти оптимального результату при відображенні глобальної карти і положення камери через проблеми локального мінімуму в ВА. Щоб уникнути цієї проблеми, перед ВА можна використовувати виявлення замкнутої петлі і оптимізацію графа положення. ORB-SLAM включає в себе багатопоточний відстеження, зіставлення і виявлення замкнутої петлі, а карта оптимізована з використанням графа положення і ВА, і це можна розглядати як пакет моноблочних VSLAM «все-в-одному». Оскільки ORB-SLAM є проектом з відкритим вихідним кодом, ми можемо легко використовувати всю систему VSLAM в нашій локальному середовищі.

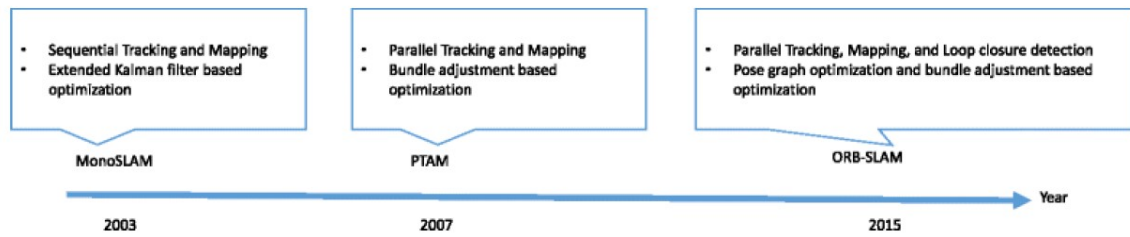


Рисунок 1.11 - Короткий опис функціональних методів

Функціональні алгоритми VSLAM, засновані на точках, зазвичай використовують рукописні детектори і дескриптори функцій і можуть забезпечити стабільні результати оцінки в добре текстурованих середовищах. Проте, важко обробляти вигнуті краю і інші складні форми, використовуючи такі функції. У деяких особливих випадках, таких як погано текстурирована середовище, лінійні функції використовувалися в якості функцій зображення. Крім того, функціональні точки і кромки об'єднуються для забезпечення надійної оцінки по відношенню до розмитого вхідного зображення.

Прямі методи можна класифікувати по щільності карт. Щільні методи створюють щільну карту, розраховану таким чином, що значення глибини оцінюються для кожного пікселя в кожному ключовому кадрі. Ці методи можуть бути корисні для 3D-моделювання в реальному часі з використанням графічного процесора.

На відміну від щільних методів, полуплотные [21] і рідкісні методи фокусуються на додатках, заснованих на відстеженні положення датчиків. Ці методи можуть виконуватися в режимі реального часу на центральних процесорах.

В ході дослідження були представлені останні алгоритми VSLAM в ключовий період з 2010 по 2018 рік. В основному, всі алгоритми VSLAM складаються з ініціалізації, оцінки руху камери, оцінки 3D-структури, глобальної оптимізації і релокалізації. Останнім часом «прямі методи» є активним полем досліджень в монокулярном VSLAM. RGB-D VSLAM був розроблений також в останні роки, так як багато для користувача RGB-D камери зараз можна придбати за низькою ціною. Незважаючи на те, що алгоритми vSLAM розроблялися з 2003 року, vSLAM як і раніше є активним полем досліджень.

У таблиці 1.1 наведено короткий порівняння представлених методів. Кожен алгоритм має різні характеристики. Необхідно вибрати відповідний алгоритм, розглядаючи мету його використання.

Таблица 1.1 - Порівняння представлених алгоритмів

Метод	Тип метода	Плотность карты	Глобальная оптимизация	Замыкание петель
Mono-SLAM	Функциональный	Разреженная	Нет	Нет
PTAM	Функциональный	Разреженная	Есть	Нет
ORB-SLAM	Функциональный	Разреженная	Есть	Есть
DTAM	Прямой	Плотная	Нет	Нет
LSD-SLAM	Прямой	Полу-плотная	Есть	Есть
SVO	Полу-прямой	Разреженная	Нет	Нет
DSO	Прямой	Разреженная	Нет	Нет
KinectFusion	RGB-D	Плотная	Нет	Нет
Dense visual SLAM	RGB-D	Плотная	Есть	Есть
ElasticFusion	RGB-D	Плотная	Есть	Есть
SLAM++	RGB-D	Плотная	Есть	Есть

Були розглянуті останні алгоритми vSLAM, засновані тільки на використанні камери. З іншого боку, алгоритми SLAM, які використовують візуальні і інерціальні дані, називаються візуально-інерційних SLAM. Об'єднуючи візуальні і інерціальні дані, ми можемо отримати більш стабільні результати оцінки. Крім того, можливе використання детальної інформації про датчик для вирішення оцінки масштабу і компенсації спотворення затвора. В даний час практично всі смартфони і планшетні пристрої мають камери, GPS, гіроскоп і акселерометр. Таким чином, вважається, що сполучення датчиків - один з напрямків для реалізації надійних і практичних систем VSLAM.

Порівнявши два зображення нерухомого предмета, зроблені з двох різних точок, можна оцінити відстань, а порівняння трьох кадрів дозволяє оцінити і швидкість наближення до об'єкта. Аналізуючи зображення, що отримується з камери під час руху, апарат може виробляти таку оцінку, щоб скоординувати своє положення в просторі.

Однак такі оцінки можуть давати істотну погрішність, що було показано в статті керівника. Тому ми звернули увагу на дослідження Австралійського професора Мандіяма Срінівасан, який вивчав схеми руху медоносних бджіл. Він експериментально довів те, що бджоли здатні плавно знижувати свою швидкість при наближенні до поверхонь для посадки [4-9]. Він припустив, що

бджолам для цього не потрібно знати свою швидкість і відстань до об'єкта. Вони не мають бінокулярного зору, як вищі тварини. Тим не менш, вони здатні здійснювати таку плавну посадку ґрунтуючись тільки на монокулярному зорі, шляхом підтримки постійним оптичного потоку на сітківці ока. Наочна демонстрація представлена на малюнку 1.12 [2].

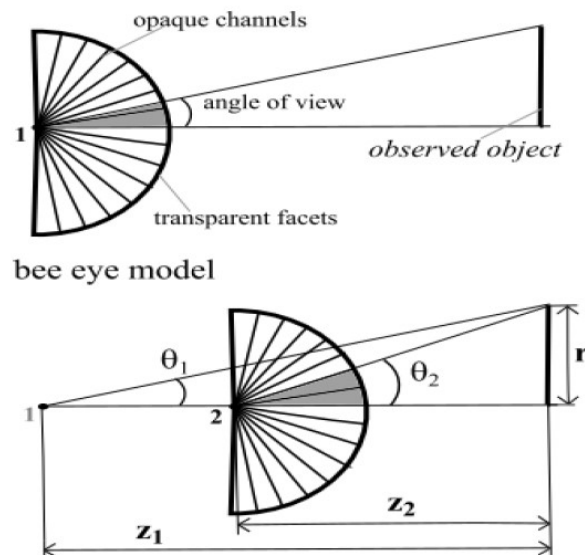


Рисунок 1.12 – Зміна кута огляду при переміщенні

Для задач прогнозу і розвідки треба щоб БПЛА мав можливість на тривалий час зависати над ділянками для точкового дослідження. При потребі безпілотник зобов'язаний швидко поміняти власну траєкторію польоту і володіти величезною маневреністю. Для цих типів БПЛА його потрібно перевозити, тому дрон повинен бути більш малогабаритним і легким. Двигуни дрібного БПЛА частіше застосовуються електронного типу. Крім цього від електродвигуна не залишається термічний відбиток, власне що робить його непримітним для засобів відстеження, в разі його застосування в бойових цілях.

При оцінці всіх переваг і дефектів, ціни і компактності існуючих типів БПЛА, вибір припав на квадрокоптер.

Квадрокоптером – найпопулярніший тип мультироторного безпілотного літального апарата, виконаний за схемою гвинтокрила. Несучі гвинти обертаються в протилежних напрямках. Рама може бути виготовлена з різних матеріалів, наприклад з металу або з полімерних матеріалів. Використовується автономна система керування. Усі датчики та периферійні прилади

розташовуються в центрі рами. Керування квадрокоптером може відбуватися за допомогою пульта дистанційного керування, або за допомогою GPS-приймача з зазначенням маршруту. Ще один з можливих варіантів автономного функціонування – по відеоданим з бортової камери.



Рисунок 1.13 – Зовнішній вигляд деякого квадрокоптеру

Компанія Intel в 2019 році представила навігаційну камеру RealSense Tracking Camera T265. За словами виробника, T265 відкриває новий клас автономних засобів стеження, надаючи розробникам потужний будівельний блок для систем керування та навігації самоврядних пристроїв. У T265 використовується запатентована технологія V-SLAM (visual inertial odometry simultaneous localization and mapping - одночасна локалізації та картографування на основі візуальної інерційної одометра). Особливістю V-SLAM є виконання необхідних обчислень в самій камері, що дозволяє забезпечити високоточне відстеження з малою затримкою, важливе для роботів, безпілотних апаратів, додатків доповненої і віртуальної реальності.



Рисунок 1.14 – Камера Intel RealSense Tracking Camera T265

У камері встановлений процесор машинного зору Intel Movidius Myriad 2, який обробляє всі дані, необхідні для відстеження. До переваг T265 віднесена компактність, низьке енергоспоживання і простота і у використанні. Камеру можна встановлювати навіть невеликі об'єкти, такі, як легкі роботи і дрони, а також підключати до смартфонів або гарнітурам AR. Камера підходить для додатків, де важливо відстежувати місце розташування без доступу до GPS, наприклад, в складських приміщеннях або в віддалених районах.

Дані, необхідні для роботи, Intel RealSense T265 отримує від двох вбудованих надширококутним камер, кожна з яких охоплює приблизно 170 °. На основі цих даних V-SLAM створює і вносить відповідні зміни карту невідомого оточення і місце розташування пристрою в цьому середовищі. Оскільки всі обчислення виконуються безпосередньо в T265, камера не залежить від платформи і може використовуватися навіть на пристроях з дуже скромними обчислювальними можливостями.

T265 можна використовувати разом з камерами Intel RealSense серії D400. Єдина вимога до обладнання - наявність незалежної пам'яті для завантаження, а також наявність підключення USB 2.0 або USB 3.0, здатного забезпечити потужність 1,5 Вт.

1.4 Постановка задач проектування

Завдання проектування полягає в розробці проекту, який буде оптимальним, відповідати всім вимогам, пред'явленим в технічному завданні, а також в розробці планів і документації, яка спростить реалізацію даного проекту в подальшому.

Для полегшення розробки програмного продукту, необхідно застосувати об'єктно-орієнтований підхід при проектуванні системи. Першим етапом є проведення об'єктно-орієнтований аналіз, другим – об'єктно-орієнтоване проектування. Застосувавши цей підхід, буде отримана система з декомпованих елементів, які можна буде легко інтегрувати в сторонні і вже існуючі продукти. Також цей підхід дає можливість швидкої подальшої розробці нових функцій та легкої підтримки актуальності продукту.

2 ДОСЛІДЖЕННЯ ПРОБЛЕМИ І СИНТЕЗ МАТЕМАТИЧНИХ МОДЕЛЕЙ

2.1 Цілі, завдання, структура та вибір шляху вирішення проблеми

Послідовність завдань перельоту в чергову задану точку маршруту або руху по заданих ділянках типових траєкторій – це і є завдання управління польотом квадрокоптера (наприклад - по прямій або дузі кола). Система автоматичного управління побудована як система зі зворотним зв'язком, що здійснює відстеження заданого маршруту. Можна виділити два основних канали – канал управління висотою і канал управління рухом в горизонтальній площині. Зміна сумарної величини тяги забезпечує стабілізацію і управління в вертикальному напрямку. Горизонтальне переміщення апарата відбувається під дією горизонтальної проекції сумарного вектора тяги. В даному варіанті зміна кутів тангажа і крену (при фіксованому куті ристання) може відхилити вектор тяги. Щоб міняти кутове положення, треба досягти диференційованого управління швидкостями обертання гвинтів. Підсистема «система орієнтації та стабілізації» забезпечує необхідні значення кутових параметрів і висоти за рахунок зміни тяги гвинтів, а підсистема «система траєкторного управління» здійснює відстеження заданих траєкторій. Цей варіант не самий найефективнішим. Структура системи управління показана на рис.2.1.

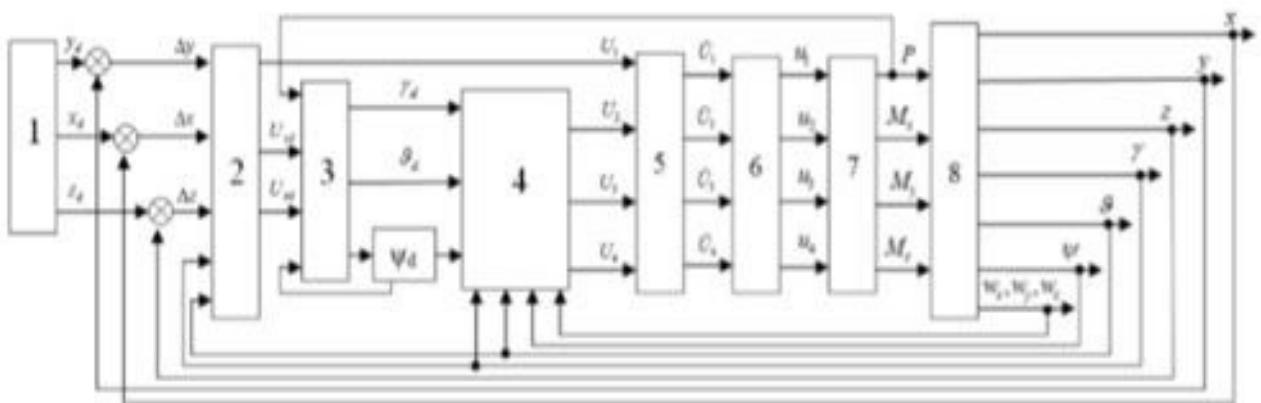


Рисунок 2.1 – Структурна система управління

На рисунку 2.1 прийняті наступні позначення:

- 1 - задана траєкторія;
- 2 - коригувальні пристрої (КУ) підсистеми траєкторного управління;

- 3 - перетворювач координат;
- 4 - регулятор підсистеми орієнтації і стабілізації;
- 5 - розподільник сигналів;
- 6 - обмежувач напруги;
- 7 - модель гвинтокорилої групи;
- 8 - модель квадрокоптера.

Алгоритми роботи підсистем стабілізації і траекторного управління пропонується розраховувати одним з відомих методів, причому краще вибирати ті, які при порівнянній якості є найбільш простими. Тому для кожної з підсистем проведений розрахунок коригуючих пристроїв у вигляді ПДрегуляторів і методом, відомим в літературі під назвою «бекстеппінг» (англ. Backstepping), після чого шляхом порівняння результатів обраний найбільш відповідний регулятор для кожної з підсистем. Щоб уникнути повторів у викладі, розрахунок регуляторів для кожної з підсистем показаний на прикладі одного з методів: «бекстеппінг» - для алгоритму стабілізації кутового положення; ПД-регулятори - для алгоритму відстеження траєкторій. Входами підсистеми орієнтації і стабілізації кутового положення є задаються підсистемою траекторного управління а виходами - параметри кутового руху об'єкта. Керуючий алгоритм цієї підсистеми (блок 4) повинен формувати керуючі сигнали U_2 , U_3 , U_4 , щоб забезпечити створення необхідних моментів M_{Rx} , M_{Ry} , M_{Rz} , за умови, що сумарна тяга гвинтів буде відповідати необхідної для вертикального руху. Якщо знехтувати динамікою і обмеженнями двигунів, гіроскопічними моментами і аеродинамічними моментами ненесучої частини апарату, то сигнали U_2 , U_3 , U_4 повинні бути з точністю до коефіцієнта рівними моментам M_{Rx} , M_{Ry} , M_{Rz} . Тому моделлю об'єкта для алгоритму стабілізації є рівняння при $M_{Rx} = U_2$, $M_{Ry} = U_3$, $M_{Rz} = U_4$. Суть методу «бекстеппінг» полягає в поданні складної системи у вигляді ланцюжка вкладених підсистем, для кожної з яких формуються допоміжні сигнали і складаються залежні від цих сигналів функції Ляпунова. Виконання критеріїв стійкості по Ляпунову при послідовному виборі цих сигналів для кожної підсистеми забезпечує стійкість системи в цілому. Процедура має характер покрокового обходу інтеграторів зворотними зв'язками, звідки - назва «integrator backstepping», або коротко - бекстеппінг (англ. Backstepping). У певних приватних випадках процедура стає регулярною і досить простий. Для кутового руху літального апарату такий випадок можливий при малих кутах тангажу і крену, коли похідні кутів можна вважати рівними відповідним

кутовим швидкостям. Тоді рівняння кутового руху можна приблизно уявити у вигляді трьох підсистем

2.2 Математичні моделі

Синтез системи автоматичного управління полягає у формуванні об'єкта автоматичного управління і в цілому автоматизованих систем управління відповідно до обраного принципу управління за вимогами до якості функціонування.

Для побудови принципу управління системи автоматичного управління було вибрано принцип управління по відхиленню, оскільки він забезпечує більшу точність управління; в розглянутій системі автоматичним об'єктом управління є приміщення; блок автоматичного управління розраховується в процесі планування, що забезпечує необхідний стан заданих показників якості функціонування системи (об'єкт – БПЛА, оптичний потік – параметр; по аналогії літак – об'єкт, швидкість літака – параметр).

Системи автоматичного управління (САУ) призначені для автоматичної зміни одного або декількох параметрів об'єкта управління з метою встановлення необхідного режиму його роботи. Системи автоматичного управління забезпечують підтримання сталості заданих значень регульованих параметрів або їх зміну по заданому закону, а також оптимізує певні критерії якості управління. До такої системи відноситься система стабілізації польотом квадрокоптера. Відносно теорії автоматичного управління (ТАУ), було створено функціональну схему квадрокоптера, що приведена нижче:

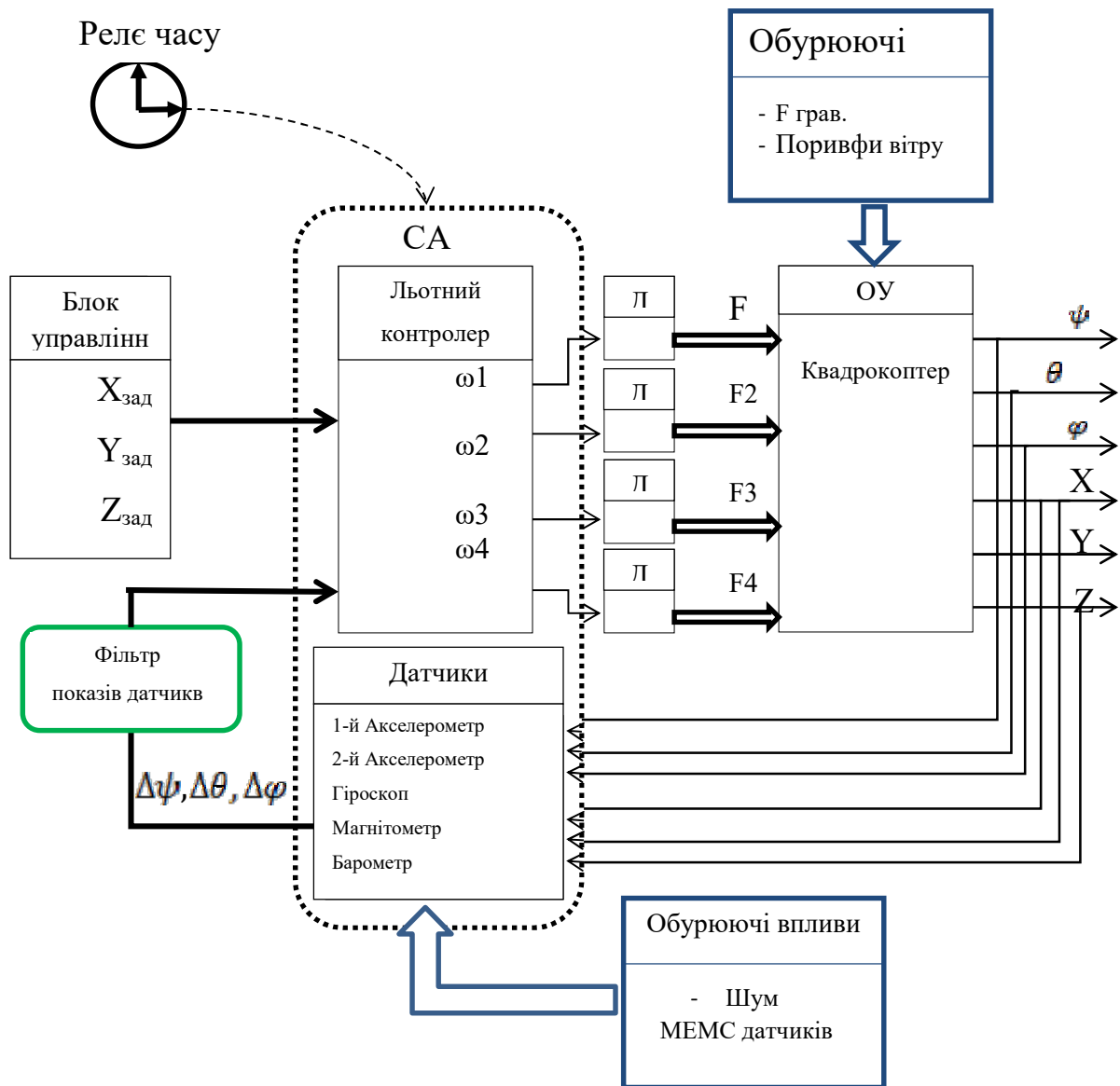


Рисунок 2.2 – Функціональна схема розроблюваного БПЛА

Блок управління являється – це задавач координат. Існує декілька режимів у сучасних квадрокоптерів. Серед них:

1. Manual mode (ручний режим польоту) – управління квадрокоптера оператором. Управління йде через пульт керування.
2. Attitude mode (режим орієнтації в просторі) – використовується акселерометр, для допомоги стабілізувати БПЛА.
3. Стабілізація за допомогою системи GPS – політ за заданими координатами польоту.

При зміні швидкості обертання двигунів задається керуючий вплив.
Схема представлена нижче:

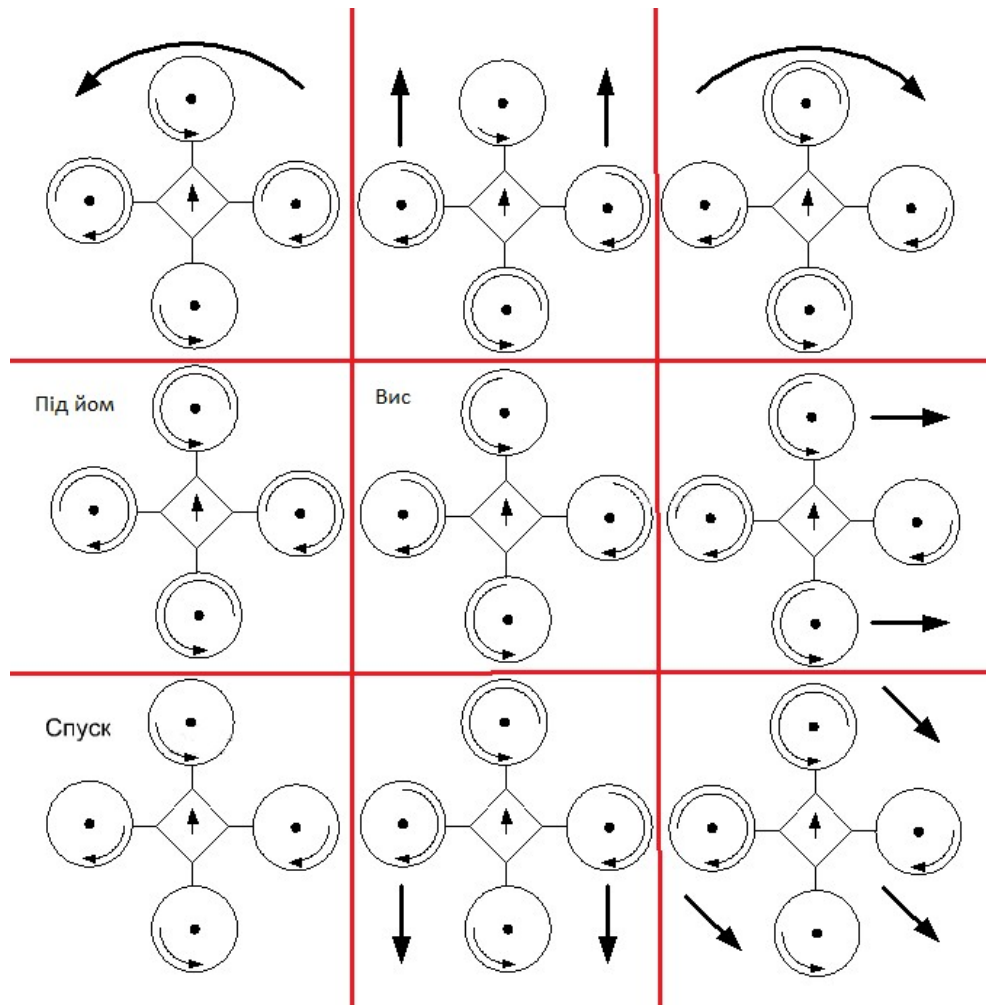


Рисунок 2.3 – Схема рухів квадрокоптера

В систему САУ входить льотний контролер Pixhawk та датчики у каналі зворотного зв'язку для забезпечення стабілізації польоту. Через настільну роботу квадрокоптера виникає необхідність стабілізації польоту. Квадрокоптер не володіє можливостями відновлювати кінематичні параметри самостіно, без участі оператора, тому при зовнішньому впливі починається несанкціоноване зниження.

Щоб ефективно стабілізувати в польоті квадрокоптер необхідно:

1. Визначити, в якому стані знаходиться система в даний момент.
2. Мати можливість надавати керуючі впливу на систему.
3. Знати, в якому стані буде підтримуватися система.

Оскільки об'єкт управління володіє досить швидкою динамікою, то це негативно позначитися при використанні некоректної системи управління. Щоб зменшити ризик травм при роботі з об'єктом та не спричини його руйнування, необхідно створити теоретичні основи, що опише досліджуваній об'єкт. Використовували математичний опис системи, як один з найбільш узагальнених підходів до теоретичного опису динамічної системи Далі для практичної роботи приведемо послідовне виведення спрощеної математичної моделі об'єкта.

Нехай положення центру мас квадрокоптера збігається з початком рухомої системи координат $CX_1Y_1Z_1$, а у нерухомій декартовій системі координат описується координатами X, Y, Z , ($OXYZ$ система) (рисунок 2.4).

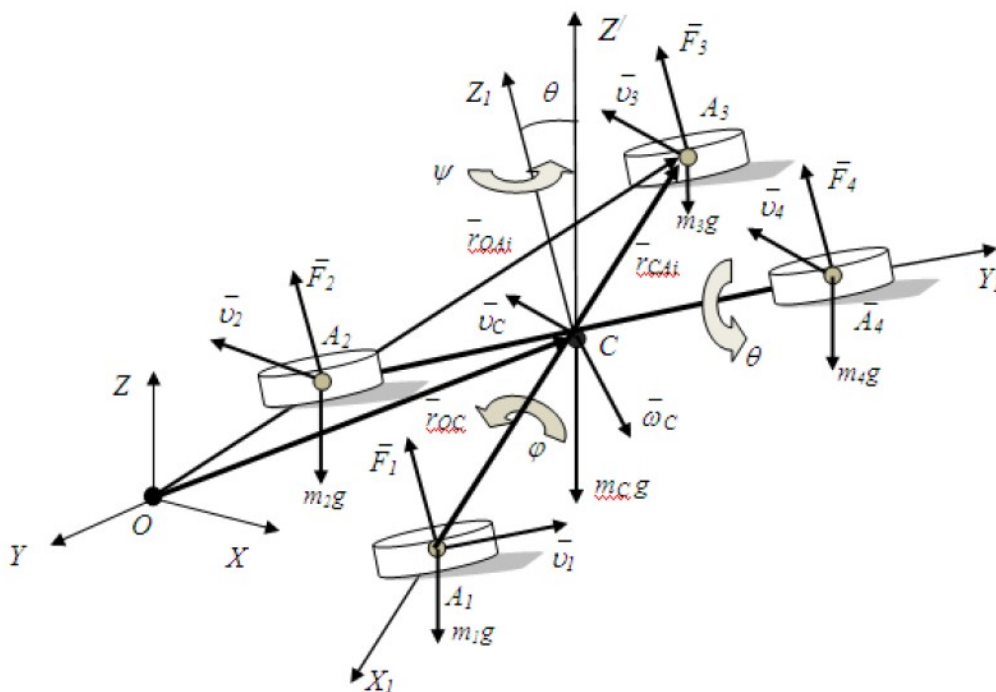


Рисунок. 2.4 – Розрахункова схема квадрокоптера

Орієнтацію в просторі задають кути Ейлера-Крилова, які зазвичай застосовуються в авіаційній техніці при описі руху апарату і складають так звані кути: крену, тангажу і рискання. Вони відповідають наступній послідовності поворотів:

1. Поворот на кут ψ - рискання.
2. Поворот на кут θ - тангаж.
3. Поворот на кут ϕ - крен.

На квадрокоптер при польоті діють аеродинамічні сили несучих гвинтів $\bar{F}_1, \bar{F}_2, \bar{F}_3, \bar{F}_4$, прикладені до центрів мас роторів A_1, A_2, A_3, A_4 відповідно, і сили тяжкості корпусу $m_C g$ і гвинтів $m_i g$ (рисунок. 2.6), причому сили \bar{F}_i , паралельні осі CZ_1 . Положення центру мас квадрокоптера виділяють координати вектора $r_{OC} = [X, Y, Z]^T$. Домовимось в подальшому системи координат $OXYZ$ і $CX_1Y_1Z_1$ розуміти під символами $^{(0)}$ і $^{(1)}$ відповідно. Тоді вектори сил [14]:

$$\bar{F}_i^{(0)} = T_{10} \cdot \bar{F}_i^{(1)} \quad (2.8)$$

$$T_{10} = (\psi, \theta, \varphi) = R(z, \psi) \times R(y, \theta) \times R(x, \varphi) =$$

$$= \begin{bmatrix} \cos\psi \cos\theta & \cos\psi \sin\theta \sin\varphi - \cos\varphi \sin\psi & \sin\psi \sin\theta \sin\varphi + \cos\psi \cos\theta \sin\varphi \\ \sin\psi \cos\theta & \cos\psi \cos\theta \sin\varphi + \sin\psi \sin\theta \sin\varphi & \cos\varphi \sin\psi \sin\theta - \cos\psi \sin\varphi \\ -\sin\theta & \cos\theta \sin\varphi & \cos\varphi \cos\theta \end{bmatrix}$$

Запишемо рівність:

$$\bar{r}_{OA_i}^{(0)} = \bar{r}_{OC}^{(0)} + \bar{r}_{CA_i}^{(0)} \quad (2.9)$$

де

$$\bar{r}_{CA_i}^{(0)} = T_{10} \cdot \bar{r}_{CA_i}^{(1)}$$

Вектори $\bar{r}_{CA_i}^{(1)}$ для точок A_i мають вигляд:

$$\bar{r}_{CA1}^{(1)} = \begin{bmatrix} l \\ 0 \\ 0 \end{bmatrix}; \bar{r}_{CA2}^{(1)} = \begin{bmatrix} 0 \\ -l \\ 0 \end{bmatrix}; \bar{r}_{CA3}^{(1)} = \begin{bmatrix} -l \\ 0 \\ 0 \end{bmatrix}; \bar{r}_{CA4}^{(1)} = \begin{bmatrix} 0 \\ l \\ 0 \end{bmatrix}.$$

де l – відстань від центру мас квадрокоптера C до центру мас роторів A_i . Швидкості точок A_i визначемо, продиференціювавши рівність (2.9) у часі:

$$\bar{v}_{A_i} = \frac{d}{dt} \bar{r}_{OA_i}^{(0)} = \frac{d}{dt} \bar{r}_{OC}^{(0)} + \frac{d}{dt} \bar{r}_{CA_i}^{(0)}.$$

З урахуванням рівності получимо:

$$\bar{v}_{A_i} = \bar{v}_c + \dot{T}_{10} \cdot \overline{r_{CAi}^{(1)}}$$

де $\bar{v}_c = \bar{i}\dot{X} + \bar{j}\dot{Y} + \bar{k}\dot{Z}$ - швидкість центра мас квадрокоптера.

Кількість руху i -ої маси визначимо за формулою:

$$\bar{q}_i = m_i \bar{v}_{A_i} = m_i (\bar{v}_c + \dot{T}_{10} \cdot \overline{r_{CAi}^{(1)}}) \quad (2.10)$$

Зміну кількості руху визначимо з виразу:

$$\frac{d\bar{q}_i}{dt} = m_i \left(\frac{d\bar{v}_c}{dt} + \ddot{T}_{10} \cdot \overline{r_{CAi}^{(1)}} \right) = T_{10} \bar{F}_i^{(1)} \quad (2.11)$$

Вектор кількості руху розглянутої системи, що складається з корпусу і 4 гвинтів, визначимо за формулою:

$$\bar{Q} = m_c \bar{v}_c + \sum_{i=1}^4 m_{A_i} \bar{v}_{A_i} \quad (2.12)$$

Теорема про зміну кількості руху механічної системи в диференціальній формі має вигляд:

$$\frac{d\bar{Q}}{dt} = m_c \frac{d\bar{v}_c}{dt} + \sum m_i \left(\frac{d\bar{v}_c}{dt} + \ddot{T}_{10} \cdot \overline{r_{CAi}^{(1)}} \right) = (m_c + \sum m_i) \frac{d\bar{v}_c}{dt} + \ddot{T}_{10} \sum m_i \overline{r_{CAi}^{(1)}} = T_{10} \bar{F}_i^{(1)} \quad (2.13)$$

У проєкціях на координатні осі рівняння (2.13) набуде вигляду:

$$\begin{cases} (m_c + \sum m_i) \frac{d\bar{v}_c^X}{dt} = \sum F_{ix}^{(0)} \\ (m_c + \sum m_i) \frac{d\bar{v}_c^Y}{dt} = \sum F_{iy}^{(0)} \\ (m_c + \sum m_i) \frac{d\bar{v}_c^Z}{dt} = \sum F_{iz}^{(0)} \end{cases}$$

Тут

$$\sum \bar{F}_i^{(0)} = T_{10} \sum \bar{F}_i^{(1)} = |T_{10}| \begin{vmatrix} \sum F_{ix}^{(1)} \\ \sum F_{iy}^{(1)} \\ \sum F_{iz}^{(1)} \end{vmatrix} = \begin{vmatrix} (\sin \psi \sin \varphi + \cos \psi \cos \varphi \sin \theta) \cdot \sum F_i \\ (\cos \varphi \sin \psi \sin \theta - \cos \psi \sin) \cdot \sum F_i \\ \cos \varphi \cos \theta \cdot \sum F_i \end{vmatrix} \quad (2.14)$$

де

$$\bar{F}_1^{(0)} = \begin{bmatrix} 0 \\ 0 \\ \bar{F}1 \end{bmatrix}, \quad \bar{F}_2^{(0)} = \begin{bmatrix} 0 \\ 0 \\ \bar{F}2 \end{bmatrix}, \quad \bar{F}_3^{(0)} = \begin{bmatrix} 0 \\ 0 \\ \bar{F}3 \end{bmatrix}, \quad \bar{F}_4^{(0)} = \begin{bmatrix} 0 \\ 0 \\ \bar{F}4 \end{bmatrix}, \quad (2.15)$$

$$\sum \bar{F}_i^{(1)} = \begin{bmatrix} 0 \\ 0 \\ \bar{F}1 + \bar{F}2 + \bar{F}3 + \bar{F}4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^4 F_i \end{bmatrix}$$

Тоді рівняння (2.14) з урахуванням (2.15) буде мати вигляд:

$$\begin{cases} (m_C + \sum m_i) \frac{d\bar{v}_C^X}{dt} = (\sin \psi \sin \varphi + \cos \psi \cos \varphi \sin \theta) \cdot \sum F_i \\ (m_C + \sum m_i) \frac{d\bar{v}_C^Y}{dt} = (\cos \varphi \sin \psi \sin \theta - \cos \psi \sin) \cdot \sum F_i \\ (m_C + \sum m_i) \frac{d\bar{v}_C^Z}{dt} = \cos \varphi \cos \theta \cdot \sum F_i \end{cases} \quad (2.16)$$

$$\begin{cases} m\ddot{X} = (\sin \psi \sin \varphi + \cos \psi \cos \varphi \sin \theta) \cdot \sum F_i \\ m\ddot{Y} = (\cos \varphi \sin \psi \sin \theta - \cos \psi \sin) \cdot \sum F_i \\ m\ddot{Z} = \cos \varphi \cos \theta \cdot \sum F_i \end{cases}$$

Представлена система диференціальних рівнянь описує зміну узагальнених координат квадрокоптера X, Y, Z.

Цю систему необхідно доповнити силою аеродинамічного опору.

$$A_x = C_x \frac{\rho v^2}{2} S.$$

де C_x - коефіцієнт аеродинамічної сили,

ρ - щільність повітря, кг/м³;

v - швидкість набігаючого потоку повітря, м/с;

S - площа поверхні апарату, на яку діє набігаючий потік, м².

Наступним завданням буде вирахування обертання роторів квадрокоптера в локальній системі координат $SX_1Y_1Z_1$ (рисунок 2.7, 2.8). Для цього введемо систему координат $A_ix_jy_iz_i$, яка збігається з центром мас m_i роторів.

$$\bar{\Omega}_i = \bar{\omega}_i + \bar{\omega}_c, i = 1 \dots 4;$$

$$\bar{\omega}_i = \bar{i}_i\omega_{ix} + \bar{j}_i\omega_{iy} + \bar{k}_i\omega_{iz}; \quad (2.17)$$

$$\bar{\omega}_c = \bar{i}_1\omega_{cX1} + \bar{j}_1\omega_{cY1} + \bar{k}_1\omega_{cZ1};$$

де $\bar{i}_1, \bar{j}_1, \bar{k}_1$ та $\bar{i}_i, \bar{j}_i, \bar{k}_i$ - одиничні вектори системи координат $SX_1Y_1Z_1$ і $A_ix_jy_iz_i$, - абсолютна кутова швидкість обертання i -ого ротора в системі координат $SX_1Y_1Z_1$; $\bar{\omega}_c, \bar{\omega}_i$ - вектори кутових швидкостей обертання корпусу та i -ого ротора в системі координат $SX_1Y_1Z_1$ і $A_ix_jy_iz_i$ визначаються у вигляді:

$$\bar{\omega}_c = \begin{bmatrix} \omega_{x1} \\ \omega_{y1} \\ \omega_{z1} \end{bmatrix}, \quad \bar{\omega}_c = \begin{bmatrix} 0 \\ 0 \\ \omega_i \end{bmatrix}, \quad \bar{\Omega}_i = \begin{bmatrix} \omega_{x1} \\ \omega_{y1} \\ \omega_i + \omega_{z1} \end{bmatrix} \quad (2.18)$$

або

$$\bar{\Omega}_i = \bar{i}_1\Omega_x + \bar{j}_1\Omega_y + \bar{k}_1\Omega_z \quad (2.19)$$

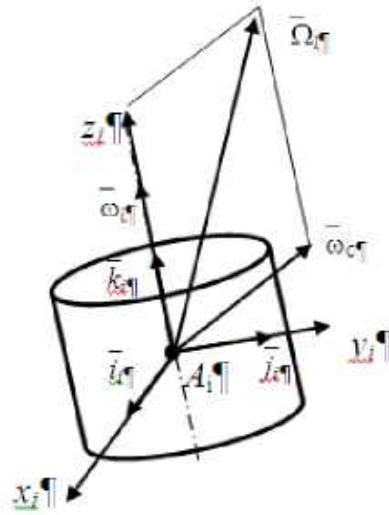


Рисунок. 2.5 – Схема визначення кутової швидкості ротора складного руху

Визначимо момент кількості руху ротора в системі координат $A_i x_i y_i z_i$

$$\bar{L}_{iA_i} = \int_{m_i} (\bar{r}_i \times \bar{v}) dm_i \quad \text{або} \quad \bar{L}_{iA_i} = I_{A_i} \bar{\Omega}_i$$

$$I_{iA_i} = \begin{vmatrix} J_{A_i}^x & 0 & 0 \\ 0 & J_{A_i}^y & 0 \\ 0 & 0 & J_{A_i}^z \end{vmatrix}$$

Тоді кінетичний момент дорівнює:

$$\bar{L}_{iA_i} = \begin{vmatrix} J_{A_i}^x & 0 & 0 \\ 0 & J_{A_i}^y & 0 \\ 0 & 0 & J_{A_i}^z \end{vmatrix} \begin{vmatrix} \omega_{X_1} \\ \omega_{Y_1} \\ \omega_i + \omega_{Z_1} \end{vmatrix} = \begin{vmatrix} J_{A_i}^x \omega_{X_1} \\ J_{A_i}^y \omega_{Y_1} \\ J_{A_i}^z (\omega_i + \omega_{Z_1}) \end{vmatrix}$$

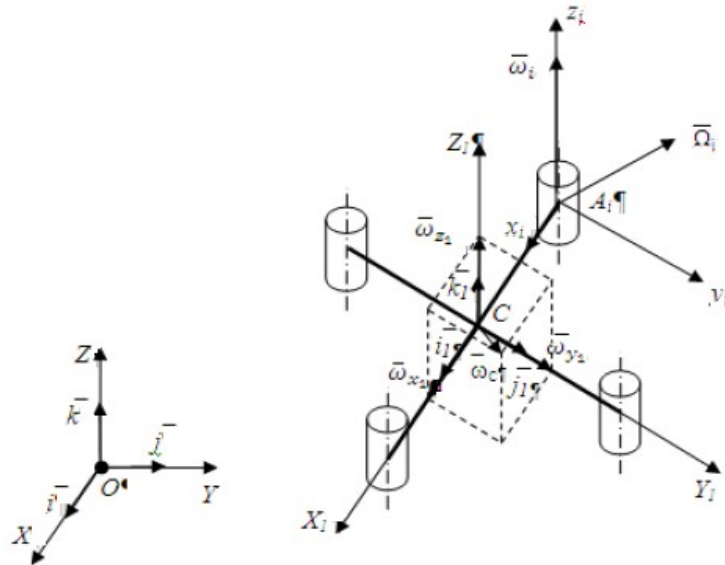


Рисунок. 2.6 – Розрахункова схема визначення кінетичного моменту квадрокоптера

Визначимо момент кількості руху в системі:

$$\bar{L} = \bar{L}_c + \sum \bar{L}_i \quad (2.20)$$

$$\bar{L}_i = \begin{vmatrix} J_{A_i}^x + m_i l^2 & 0 & 0 \\ 0 & J_{A_i}^y + m_i l^2 & 0 \\ 0 & 0 & J_{A_i}^z + m_i l^2 \end{vmatrix} \begin{vmatrix} \omega_{X_1} \\ \omega_{Y_1} \\ \omega_i + \omega_{Z_1} \end{vmatrix} = \begin{vmatrix} (J_{A_i}^x + m_i l^2) \omega_{X_1} \\ (J_{A_i}^y + m_i l^2) \omega_{Y_1} \\ (J_{A_i}^z + m_i l^2) (\omega_i + \omega_{Z_1}) \end{vmatrix}$$

де $\bar{L}_c = I_c \bar{\omega}_c$ - кінетичний момент відносно центру мас квадрокоптера,

$\bar{L}_i = I_i \bar{\Omega}_i = (I_{A_i} + m_i l^2) \bar{\Omega}_i$ - кінетичний момент i -го ротора відносно центру мас квадрокоптера в системі координат $CX_1Y_1Z_1$ (відповідно до теореми Гюйгенса).

Тензори інерції корпусу I_C та i -го ротора I_i з урахуванням того, що головні осі інерції механічної системи є головними центральними осями інерції (всі відцентрові моменти інерції дорівнюють нулю) рівні:

$$I_C = \begin{vmatrix} J_C^{X_1} & 0 & 0 \\ 0 & J_C^{Y_1} & 0 \\ 0 & 0 & J_C^{Z_1} \end{vmatrix}; \quad I_i = \begin{vmatrix} J_{A_i}^x + m_i l^2 & 0 & 0 \\ 0 & J_{A_i}^y + m_i l^2 & 0 \\ 0 & 0 & J_{A_i}^z + m_i l^2 \end{vmatrix} \quad (2.21)$$

тоді

$$\bar{L}_C = \begin{vmatrix} J_C^{X_1} & 0 & 0 \\ 0 & J_C^{Y_1} & 0 \\ 0 & 0 & J_C^{Z_1} \end{vmatrix} \begin{vmatrix} \omega_{X_1} \\ \omega_{Y_1} \\ \omega_{Z_1} \end{vmatrix} = \begin{vmatrix} J_C^{X_1} \omega_{X_1} \\ J_C^{Y_1} \omega_{Y_1} \\ J_C^{Z_1} \omega_{Z_1} \end{vmatrix} \quad (2.22)$$

З урахуванням (2.21), (2.22) вираз (2.20) буде мати вигляд:

$$L = \begin{vmatrix} (J_C^{X_1} + \sum J_{A_i}^x + \sum m_i l^2) \omega_{X_1} \\ (J_C^{Y_1} + \sum J_{A_i}^y + \sum m_i l^2) \omega_{Y_1} \\ (J_C^{Z_1} + \sum J_{A_i}^z + \sum m_i l^2) \omega_{Z_1} + (\sum J_{A_i}^z + \sum m_i l^2) \omega_i \end{vmatrix} = \begin{vmatrix} J^{X_1} \omega_{X_1} \\ J^{Y_1} \omega_{Y_1} \\ J^{Z_1} \omega_{Z_1} + \sum J_i^z \omega_i \end{vmatrix} \quad (2.23)$$

де $J^{X_1} = J_C^{X_1} + \sum J_{A_i}^x + \sum m_i l^2$, $J^{Y_1} = J_C^{Y_1} + \sum J_{A_i}^y + \sum m_i l^2$,

$J^{Z_1} = J_C^{Z_1} + \sum J_{A_i}^z + \sum m_i l^2$.

$\sum J_i^z = J_{A_i}^z + \sum m_i l^2$ - приведені осьові моменти інерції.

Теорема про зміну кінетичного моменту механічної системи:

$$\frac{d\bar{L}}{dt} = \frac{d\bar{L}}{dt} + (\bar{\omega}_C \times \bar{L}) = \sum \bar{M}_C^e$$

$$\frac{d\bar{L}}{dt} = \begin{vmatrix} J^{X_1} \dot{\omega}_{X_1} + \omega_{Y_1} \omega_{Z_1} (J_i^{Z_1} - J_i^{Y_1}) + \omega_{Y_1} \sum J_i^z \omega_i & M_{X_1}^e \\ J_i^{Y_1} \dot{\omega}_{Y_1} + \omega_{X_1} \omega_{Z_1} (J_i^{X_1} - J_i^{Z_1}) - \omega_{X_1} \sum J_i^z \omega_i & M_{Y_1}^e \\ J_i^{Z_1} \dot{\omega}_{Z_1} + J_i^z \dot{\omega}_i + \omega_{X_1} \omega_{Y_1} (J_i^{Y_1} - J_i^{X_1}) & M_{Z_1}^e \end{vmatrix} \quad (2.24)$$

В результаті на підставі (2.16) і (2.24) отримано систему диференціальних рівнянь, що описують рух квадрокоптера:

$$\begin{cases} m\ddot{X} = (\sin \psi \sin \varphi + \cos \psi \cos \varphi \sin \theta) \cdot \sum F_i \\ m\ddot{Y} = (\cos \varphi \sin \psi \sin \theta - \cos \psi \sin \theta) \cdot \sum F_i \\ m\ddot{Z} = \cos \varphi \cos \theta \cdot \sum F_i \\ J^{X_1} \dot{\omega}_{X_1} + \omega_{Y_1} \omega_{Z_1} (J_i^{Z_1} - J_i^{Y_1}) + \omega_{Y_1} \sum J_i^z \omega_i = M_{X_1}^e \\ J_i^{Y_1} \dot{\omega}_{Y_1} + \omega_{X_1} \omega_{Z_1} (J_i^{X_1} - J_i^{Z_1}) - \omega_{X_1} \sum J_i^z \omega_i = M_{Y_1}^e \\ J_i^{Z_1} \dot{\omega}_{Z_1} + J_i^z \dot{\omega}_i + \omega_{X_1} \omega_{Y_1} (J_i^{Y_1} - J_i^{X_1}) = M_{Z_1}^e \end{cases} \quad (2.25)$$

Дану систему (2.25) необхідно вирішувати з кінематичними співвідношеннями, які будуть виражають проекції кутової швидкості тіла на осі зв'язаної системи координат через кутові швидкості кутів тангажу, ристання крену:

$$\begin{cases} \omega_{X_1} = \dot{\varphi} + \dot{\psi} \sin \theta \\ \omega_{Y_1} = \dot{\psi} \cos \theta \cos \varphi + \dot{\theta} \sin \varphi \\ \omega_{Z_1} = \dot{\theta} \cos \varphi - \dot{\psi} \cos \theta \sin \varphi \end{cases} \quad (2.26)$$

або

$$\begin{cases} \dot{\varphi} = \omega_{X_1} - (\omega_{Z_1} \cos \varphi - \omega_{Y_1} \sin \varphi) \operatorname{ctg} \theta \\ \dot{\theta} = \omega_{Z_1} \sin \varphi + \omega_{Y_1} \cos \varphi \\ \dot{\psi} = \frac{1}{\cos \theta} (\omega_{Z_1} \cos \varphi - \omega_{Y_1} \sin \varphi) \end{cases} \quad (2.27)$$

У терміна «оптичний потік» існують кілька визначень. Срінівасон з колегами оптичним потоком вважає швидкість розширення зображення об'єкта на сітківці ока при наближенні до нього по оптичній осі (2.1). Срінівасан висунув гіпотезу, що для здійснення м'якої посадки з плавним зниженням швидкості при наближенні до поверхні потрібно всього лише підтримувати оптичний потік на постійному рівні протягом усього польоту.

Була створена геометрична модель посадки, яка дозволяє визначити оптичний потік, ґрунтуючись на різниці в кутах огляду при спостереженні з різних точок.

Пізніше ця модель була спрощена доцентом ТПУ Хамухінім Олександром Анатолійовичем і приведена до виду [3] - формула обчислення оптичного потоку в рамках моделі:

$$I = \frac{2(\theta_2 - \theta_1)}{\Delta t * \sin 2\theta_1} \quad (2.28)$$

де I – оптичний потік,

θ_1 – кут, яку він обіймав об'єктом на першому кроці,

θ_2 – кут, яку він обіймав на другому кроці.

Виникла ідея використовувати цю модель для задачі автоматизації положення БПЛА. Використовуючи зображення з камери на борту дрона для обчислення оптичного потоку, за допомогою автоматичного регулювання домогтися підтримки його на одному рівні.

Перевагами методу є незалежність від зовнішньої інформації за винятком зображення з камери на борту, а також відсутність випромінювань, які виникають при використанні GPS-навігації і сонара.

Розглядається задача польоту БПЛА від точки старту до моменту зникнення використовуваного навігаційного сигналу GPS або ГЛОНАСС і пошук місцезнаходження по орієнтирам в автономному режимі. Під час польоту БПЛА від точки старту при наявності навігаційного сигналу (до моменту його зникнення) алгоритм будує глобальну карту місцевості, відображеної бортовий відеокамерою, шляхом обчислення географічних координат особливих точок, знайдених на кадрах. Після зникнення сигналу за допомогою цих точок здійснюється навігація БПЛА для його повернення додому. Для побудови глобальної карти місцевості використовуються в повному обсязі не кадри, а тільки так звані перекриваються ключові кадри. В

алгоритмі в якості ключового використовувався кожен μ -й кадр (з наперед заданими μ). Основна вимога при виборі μ полягає в тому, що розмір в пікселях області, видимої на всіх перекриваються кадрах, повинен бути не менше заданої частини кадру, наприклад чверті, третини і т. д.

Нехай $\Omega = \{I_1, \dots, I_n\}$ – послідовність ключових кадрів, отриманих з бортовою камери БПЛА при польоті в умовах присутності зовнішніх навігаційних сигналів GPS або ГЛОНАСС. Для кожного кадру $I_k \in \Omega$ позначимо через T_k і R_k відповідно вектор переміщення і матрицю повороту, що визначають положення камери в просторі відносно нерухомої глобальної системи координат в той момент, коли був зроблений кадр I_k . Точка π на кадрі I_k , відповідна 3D-якій точці $X = (x, y, z)$ в просторі з проектними координатами $X = (x, y, z, 1)$, яка є функцією від змінних R_k, T_k, X визначається відповідно до виразу (2.29):

$$\pi(\mathbf{R}_k, T_k, \tilde{X}) = \frac{1}{z} \cdot \mathbf{K} \cdot \mathbf{R}_k \cdot (\mathbf{I}, -T_k) \cdot \tilde{X},$$

де \mathbf{K} - калібрувальна матриця бортовий камери БПЛА,

\mathbf{I} – одинична матриця.

Опис алгоритму автономної візуальної навігації.

Запропонований алгоритм візуальної навігації складається з трьох великих блоків кроків. На рис. 2.7 приведена блок-схема алгоритму, на якій політ при наявності GPS - перший блок кроків, другий - політ без GPS, а третій - навігація БПЛА по траєкторії.

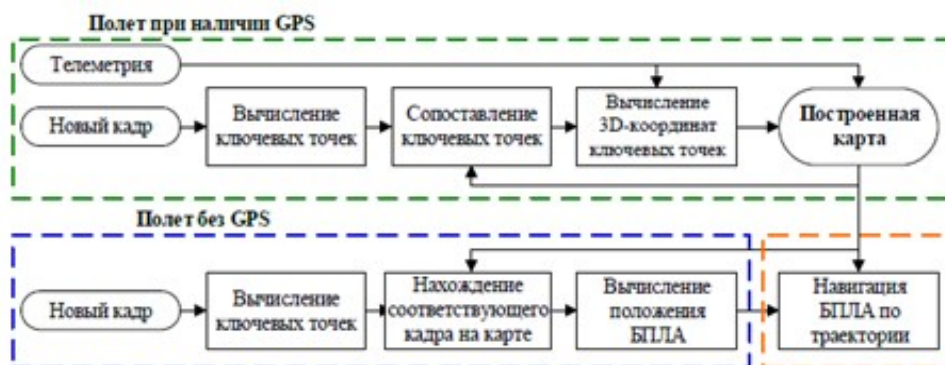


Рисунок 2.7 – Схема алгоритму автономної візуальної навігації

Завданням першого блоку алгоритму є визначення для кожного кадру $I_k \in \Omega$ просторових координат точок ландшафту $X^k = \{X_1^k, X_2^k \dots\}$, відповідних знайденим на ньому особливим точкам $u^k = \{u_1^k, u_2^k \dots\}$.

Для знаходження ключових точок на кадрах використовувалися як відомі детектори особливих точок SURF, SIFT, AKAZE, так і спеціально розроблений детектор екстремальних моментів EMD [9]. Проведені експерименти показали, що детектори SURF і EMD забезпечують знаходження більшої кількості інформативних ключових точок і тому краще підходять для вирішення даної задачі.

Основним етапом першого блоку алгоритму є знаходження безлічі 3D-точок ландшафту $\Theta = \{X_1 \dots X_m\}$, видимих на декількох ключових кадрах. Спочатку на кожному новому кадрі I_k шукаються ключові точки, u^k які зіставляються з ключовими точками u^{k-1} з попереднього кроку. Далі для кожної зіставленої пари ключових точок $\{u_i^{k-1}, u_j^k\}$ точці u_j^k ставиться у відповідність точка X_l , якщо раніше вже було встановлено відповідність між X_l і u_i^{k-1} , або обом ключових точках ставиться у відповідність нова 3D-точка, яка додається в множество Θ . В результаті для кожної 3D-точки $X_l \in \Theta$ формується набір ключових точок $U^l = \{u_i^s, \dots, u_j^d\}$ з деякої кількості ключових кадрів безлічі Ω , а також для кожного кадру I_k – набір 3D-точок $\tilde{X}^k = X^k \cap \Theta$, яким поставлено у відповідність деякі ключові точки з u^k . Після цього кожна точка X_l знаходиться з умови:

$$\begin{cases} \pi(\mathbf{R}_s, T_s, \tilde{X}_l) - \tilde{u}_i^s = 0, \\ \vdots \\ \pi(\mathbf{R}_d, T_d, \tilde{X}_l) - \tilde{u}_j^d = 0, \end{cases}$$

де \tilde{u}_i^s , \tilde{u}_j^d , \tilde{X}_l – проектні координати точок u_i^s , u_j^d , X_l відповідно. Щоб підвищити точність обчислення координат точок безлічі Θ в пізнішій версії алгоритму, була використана ускладнена схема зіставлення ключових точок на кадрах, в якій ключові точки нового кадру u^k зіставлялися з ключовими точками кількох попередніх кадрів. В результаті було отримано надійний метод відстеження точок ландшафту на ключових кадрах, які бралися з досить великим кроком по часу.

Для знаходження координат точок, X^k відповідних ключових точках кожного кадру $I_k \in \Omega$, спочатку оцінювалися параметри 3D-поверхні першого або другого порядку, апроксимуючої рельєф, шляхом наближення точок з \hat{X}^k , після чого перебували X_i^k як точки перетинання променя, що виходить з оптичного центру камери і проходить через u_i^k зі знайденою 3D-поверхнею.

Другий блок кроків алгоритму автономної візуальної навігації призначений для знаходження просторових координат камери БПЛА під час автономного польоту.

Нехай I^t – кадр, отриманий при поверненні БПЛА в точку старту. Необхідно обчислити вектор трансляції T^t і матрицю повороту R^t камери в момент часу t . Для цього слід знайти спочатку особливі точки u^t на кадрі I^t , а потім отриманий при польоті від точки старту при наявності навігаційних сигналів кадр $I(t) \in \Omega$, між ключовими точками якого і u^t вдається знайти достатню кількість відповідностей. В якості початкового кадру для пошуку $I(t)$ береться або $I(t-1) \in \Omega$ (кадр, виявлений на попередньому кроці), або останній кадр Ω , якщо I^t – перший кадр, отриманий за умови відсутності сигналів GPS або ГЛОНАСС.

Нехай далі $I(t) = I_k$, а u^t і u^k – два набори відповідних один одному проєктивних координат особливих точок з I^t і I^k , X^k – набір проєктивних координат 3D-точок ландшафту, які відповідають u^k . Тоді R^t можна знайти за допомогою восьмиточечного [11] або п'ятиточкового [12] алгоритму, використовуючи тільки u^t і u^k , після чого вектор зсуву T^t знаходиться наступним чином:

$$T^t = \underset{T}{\operatorname{argmin}} \sum_i \left\| \tilde{u}_i^t - (R^t, T, \tilde{X}_i^k) \right\|^2.$$

Третій блок алгоритму в кожен момент часу t , коли отримана нова оцінка T^t положення БПЛА, знаходить новий вектор D^t напрямку руху БПЛА, щоб здійснити його політ до точки старту вздовж збереженої траєкторії $T = \{T_1, \dots, T_n\}$ польоту літального апарату від точки старту. Ця проста на перший погляд завдання не є тривіальною в разі заплутаного маршруту БПЛА (рис. 2.8).

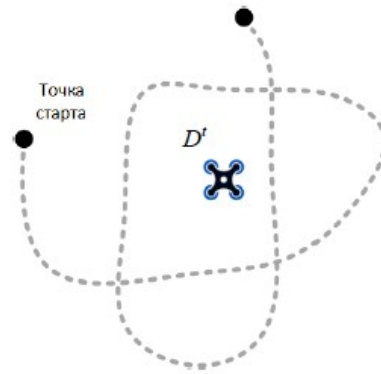


Рисунок 2.8 – Приклад заплутаного маршруту БПЛА

У розглянутому алгоритмі напрямок руху на кожному кроці визначається як $D^t = T(t) - T^t$, де $T(t)$ – деяка спеціально обрана точка на траєкторії T . Для її вибору знаходиться T_k – найближча точка на траєкторії T до позиції T^t , а потім для деякого наперед заданого відстані L (в експериментах були використані значення від 5 до 20 в залежності від швидкості БПЛА і складності маршруту) перевіряється умова $\|T_k - T^t\| > L$.

Якщо умова виконується, то $T(t) = T_k$. В іншому випадку шукається мінімальне число l , для якого виконується умова $\|T_{k-l} - T^t\| > L$, і точка $T(t)$ покладається рівною $T(t) = T_{k-l+1}$.

У разі складної траєкторії T може вийти так, що $T(t) = T_i$ а $T(t-1) = T_j$, де $i > j$. Тоді для запобігання зациклення польоту необхідно покласти $T(t) = T(t-1)$.

Досить велика кількість експериментів показало застосовність розробленого підходу.

Введемо деякі позначення (див. рис. 2.8):

I – число орієнтирів, детектованих системою реєстрації,

K – число ключових кадрів;

X^i – 3D- координати i -го орієнтира ($i = 0, I-1$);

X_c^k – 3D-координати положення пристрою реєстрації для k -го ключового кадру ($k = 0, K-1$);

f^k – фокус пристрої реєстрації для k -го кадру (величина, відома з точністю до діапазону i , можливо, приблизно);

a_1^k – вектор орієнтації пристрою реєстрації для k -го ключового кадру ($l = 0, 1, 2$). Матриця A^k , складена з рядків триплета транспонованих векторів a_1^k , подана в вигляді (трійка кутів $\alpha^k, \beta^k, \gamma^k$ визначає орієнтацію в просторі пристрою реєстрації) [1 – 3, 9]:

$$A^k \equiv \begin{pmatrix} \cos \alpha^k & -\sin \alpha^k & 0 \\ \sin \alpha^k & \cos \alpha^k & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} \cos \beta^k & 0 & -\sin \beta^k \\ 0 & 1 & 0 \\ \sin \beta^k & 0 & \cos \beta^k \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma^k & -\sin \gamma^k \\ 0 & \sin \gamma^k & \cos \gamma^k \end{pmatrix};$$

Де $x^{k,i}$, $y^{k,i}$ – стовпець i рядок k -го кадру, що визначають координати проєкції i -го орієнтира (величини, що визначаються на етапі асоціації даних);

s^x , s^y – вертикальний і горизонтальні розміри осередків ПЗЗ-матриці пристрою реєстрації (заздалегідь відомі величини);

$\delta(k, i) \in \{0, 1\}$ – індикатор присутності на k -м ключовому кадрі зображення i -го орієнтира (величини, що визначаються на етапі асоціації даних);

$\psi(j=1,5)$ – вещественнозначні функції, що визначають приватні критерії (наприклад, відхилення знайдених координат пристрою від відомих або певних сенсорам);

μ_j ($\mu_j \geq 0$, $j=1,6$) – вагові множники, що характеризують відносну важливість відповідного критерію (в тому числі помилки проектування).

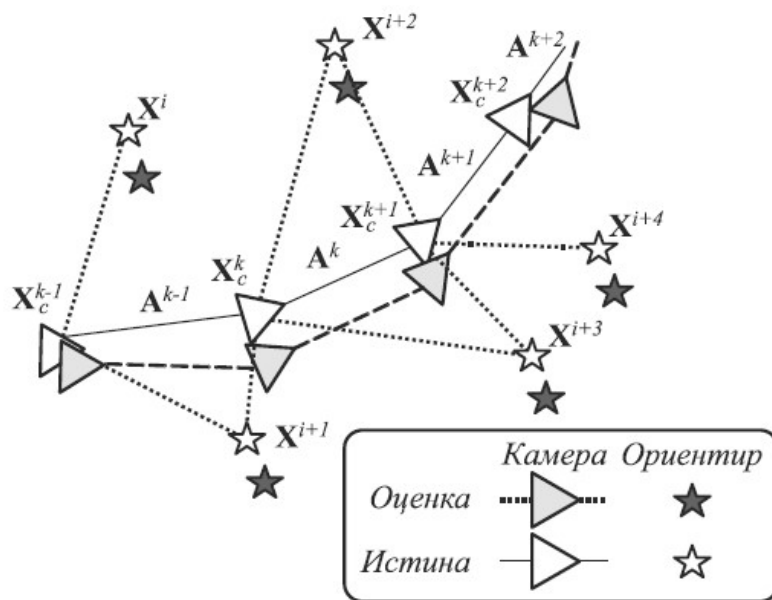


Рисунок 2.9 – Ілюстрація до позначень SLAM, з корекцією позначень

Завданням даної KSLAM-системи є оцінка положень орієнтирів X^i і параметрів станів пристрою реєстрації: координат X_c^k і орієнтації α^k , β^k , γ^k .

Ілюстрація послідовного виконання етапів рішення SLAM-проблеми від кадру до кадру приведена на рис. 2.8. Нижче представлено уточнення щодо реалізації окремих етапів даної і аналізованої KSLAM-системи.

Ініціалізація. Ініціалізація виконується наступним чином:

- стартове положення пристрою реєстрації збігається з початком координат $X_c^0 = (0,0,0)^T$;

- стартова орієнтація пристрою реєстрації покладається «нульовий»: $\alpha^0 = \beta^0 = \gamma^0 = 0$;

- фокус пристрої реєстрації f^0 для стартового кадру визначається або мобільним пристроєм, або задається інтерактивно користувачем. На підставі цієї величини визначається величина $d(f^0)$ потенційної «віддаленості» спостережуваних орієнтирів від пристрою реєстрації (задається таблично при калібрування камери);

- координати орієнтирів, видимих на стартовому кадрі $\{i: \delta(0,i) = 1, i \in I\}$, визначаються як перетин лінії, що проходить через початок координат $X_c^0 = (0,0,0)^T$ і точку з фізичними координатами $(x^{0,i} \quad s_x, y^{0,i} \quad s_y, f^0)$, задающую положення i -го орієнтира на стартовому кадрі, з площиною $Z = d(f^0)$;

- при наявності даних з сенсорів вони виступають як ініціюють значення положення і орієнтації пристрою реєстрації та / або параметри відповідних приватних критеріїв ψ_j .

Асоціація даних. У розглянутій дослідницької KSLAM- системі блок асоціації даних не передбачений. Передбачається безпомилкова асоціація всіх орієнтирів, яка при проведенні експериментальних досліджень (параграф 4 роботи) виконувалася вручну (для натурного експерименту) або автоматично (для модельного експерименту).

Оцінка стану. Виконувалася в рамках оптимізації функціоналу, представленого в пункті «Взаємне уточнення».

Побудова карти / сцени. Виконувалася в рамках оптимізації функціоналу, представленого в пункті «Взаємне уточнення».

Відсів помилок. У розглянутій дослідницької KSLAM-системі блок не передбачений (див. коментар до пункту «Асоціація даних»).

Замикання циклу. У розглянутій дослідницької KSLAM-системі блок не передбачений. Взаємне уточнення. Критерій:

$$\begin{aligned}
\mathfrak{J} = & \mu_1 \frac{1}{2} \sum_{k=0}^{K-1} \sum_{i=0}^{I-1} \delta(k, i) \times \\
& \times \left(\left(\frac{(\mathbf{a}_0^k)^T (\mathbf{X}^i - \mathbf{X}_c^k)}{(\mathbf{a}_2^k)^T (\mathbf{X}^i - \mathbf{X}_c^k)} f^k - x^{k,i} S_x \right)^2 + \right. \\
& \left. + \left(\frac{(\mathbf{a}_1^k)^T (\mathbf{X}^i - \mathbf{X}_c^k)}{(\mathbf{a}_2^k)^T (\mathbf{X}^i - \mathbf{X}_c^k)} f^k - y^{k,i} S_y \right)^2 \right) + \\
& + \mu_2 \sum_{k=0}^{K-1} \Psi_1(f^k) + \mu_3 \sum_{k=0}^{K-1} \Psi_2(\alpha^k, \beta^k, \gamma^k) + \\
& + \mu_4 \sum_{k=0}^{K-1} \Psi_3(\mathbf{X}_c^k) + \mu_5 \sum_{k=0}^{K-1} \sum_{m=k+1}^{K-1} \Psi_4(\|\mathbf{X}_c^k - \mathbf{X}_c^m\|) + \\
& + \mu_6 \sum_{k=0}^{K-1} \Psi_5(\mathbf{X}^i).
\end{aligned}$$

Критерій складається з двох основних складових, а саме:

- помилка проектування (перший доданок)
- неузгодженість значень шуканих параметрів (метричного уявлення сцени і станів пристрої реєстрації монокулярного KSLAM) з відомими / прогнозними / наближеними оцінками (складові 2-6). При відсутності додаткової інформації про завдання відповідні складові виключаються.

Алгоритм рішення відповідної задачі мінімізації зазначеного критерію: градієнтний алгоритм.

2.3 Оцінка результатів моделювання

Модель квадрокоптера в середовищі Simulink зображена на рисунку 2.10.

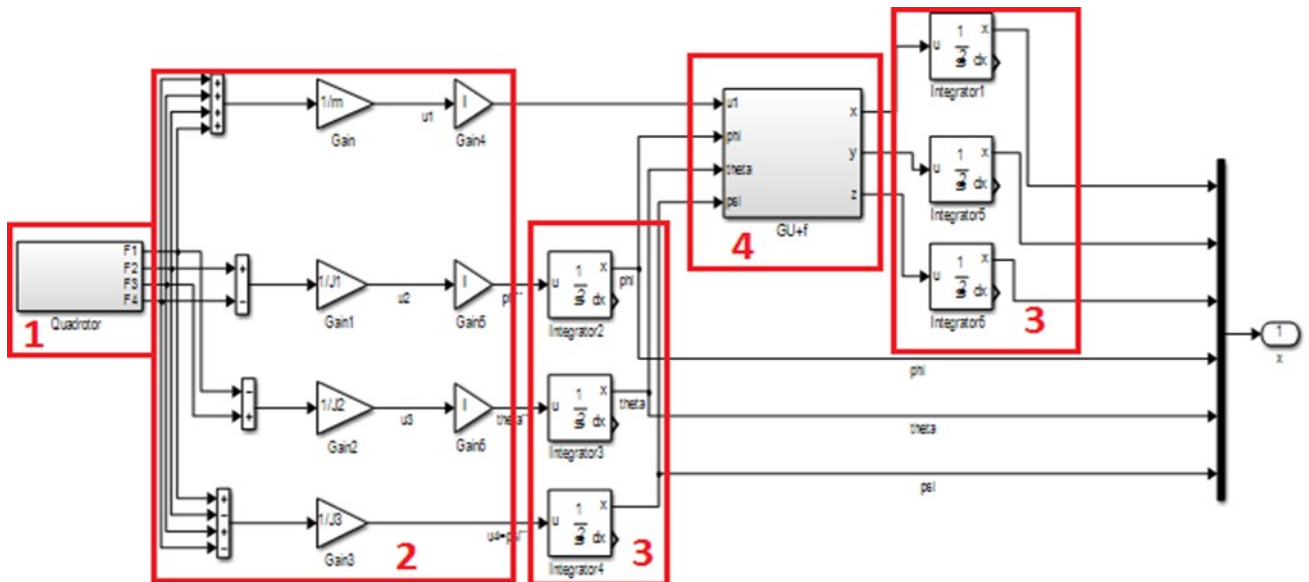


Рисунок 2.10 - Модель квадрокоптера

На рисунку 2.10 виділено такі елементи моделі:

- 1 - підсистема власне квадрокоптера: Quadrotor;
- 2 - обчислення керуючих сигналів U ;
- 3 - блоки інтегрування;
- 4 - підсистема перетворення пов'язаної CO в інерційну: $gu + f$.

Розкриємо вміст підсистем Quadrotor і $gu + f$. На рисунку 2.11 представлена структура підсистеми Quadrotor.

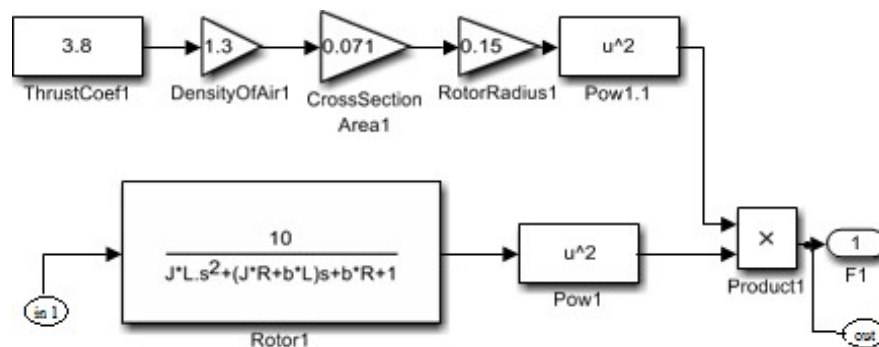


Рисунок 2.11 – Структура підсистеми Quadrotor

У цій підсистемі реалізуються моделі двигунів квадрокоптера, а також тяга, яку вони створюють. На рисунку зображена модель першого двигуна, так

як всі моделі ідентичні. На вхід блоку Rotor1 надходить керуючий вплив-напряга з блоку Step, а на виході отримуємо частоту обертання валу, яка використовується в розрахунку тяги, створюваної двигуном.

Далі розглянемо підсистему $g_u + f$, зображену на рисунку 2.12.

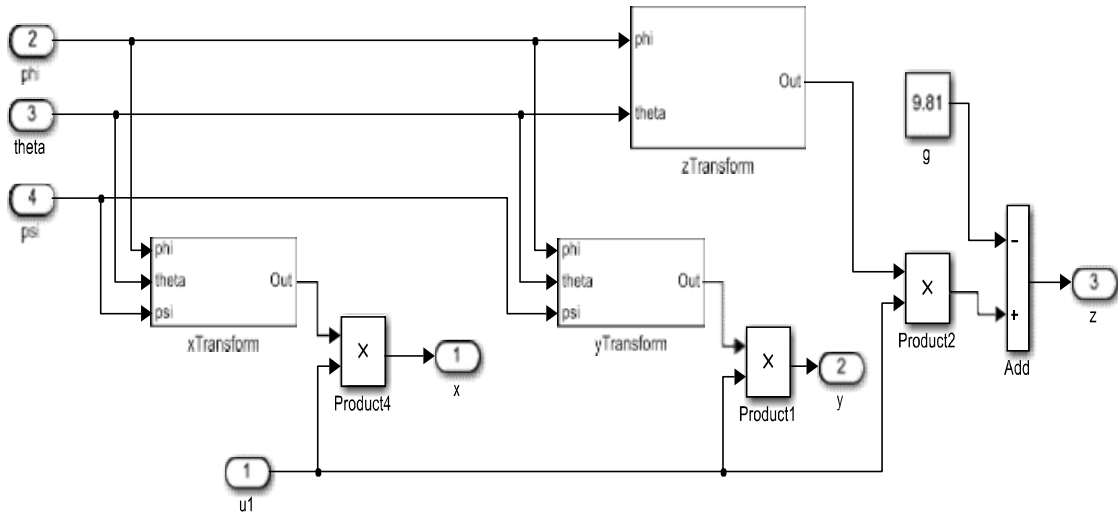


Рисунок 2.12 – Підсистема $g_u + f$

Блоки $xTransform$, $yTransform$ і $zTransform$ реалізують перетворення пов'язаної системи відліку в інерційну.

На рисунку 2.13 наведена структура блоку $xTransform$.

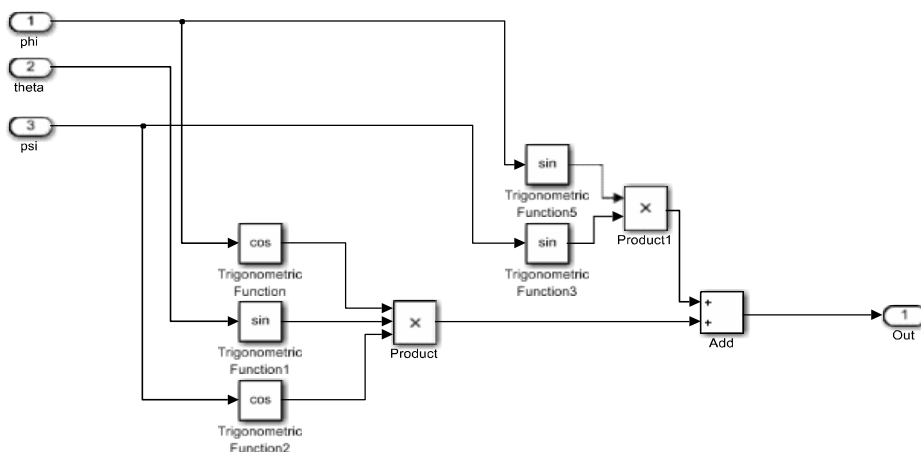


Рисунок 2.13 – Структура блоку $xTransform$

На виході підсистеми $g_u + f$ маємо прискорення вздовж осей інерційної СО, які надходять на інтегратори.

Припустимо, що нам необхідно підняти квадрокоптер на певну висоту строго вертикально і повернути його назад на землю, при тому що зовнішні обурення відсутні. На рисунку 2.14 зображено траєкторія такого руху.

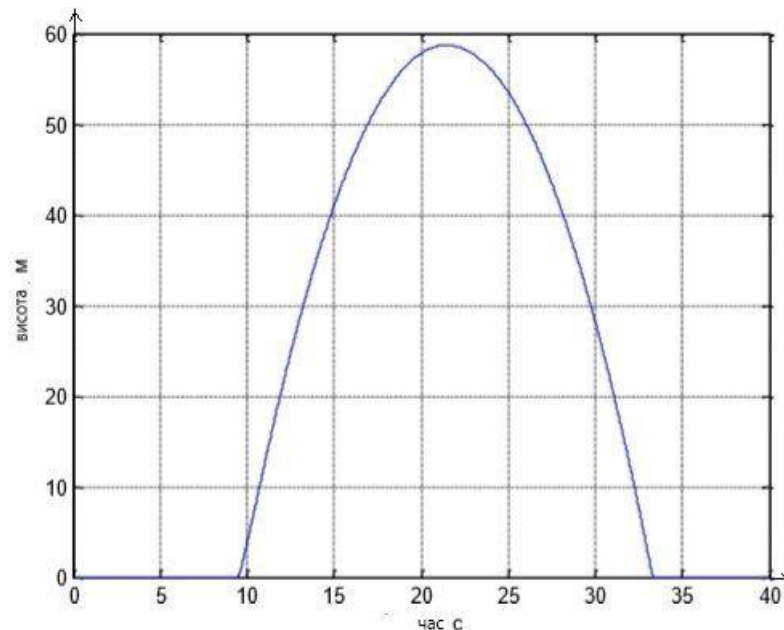


Рисунок 2.14 – Зміна висоти

Збурень немає, тож зміни по осях X і Y відсутні, і не змінюються кути Ейлера. На рисунку 2.14 бачимо, що квадрокоптер на відріжку часу $t \in [9; 22]$ набирає висоту, після чого керуючий вплив встановлюється в 9 і квадрокоптер, під дією тяжіння повертається на землю. Через відсутність СУ, квадрокоптер не зможе компенсувати зміни крену при невеликому збуренні. Це може призвести до відхилення вертикальної траєкторії і згодом до аварії.

СУ без урахування динаміки двигунів:

$$\dot{x} = U - v + fr(x)$$

Схема цієї моделі в середовищі Simulink приведена на рисунках 2.15-2.16.

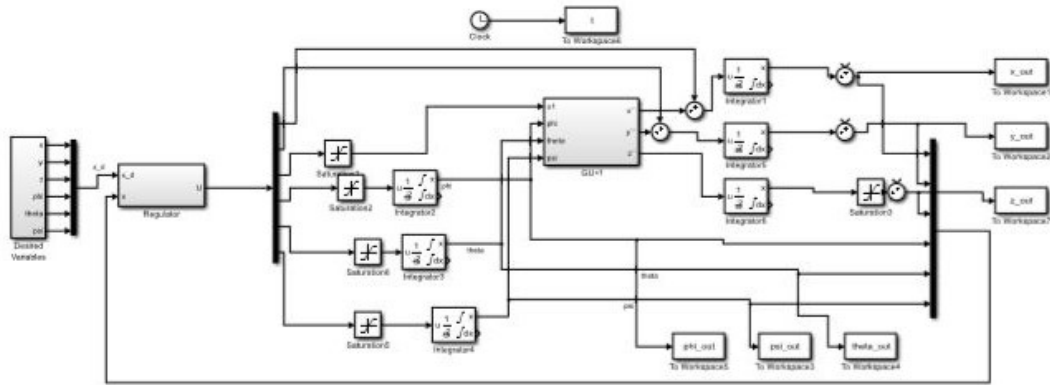


Рисунок 2.15 – Модель СУ

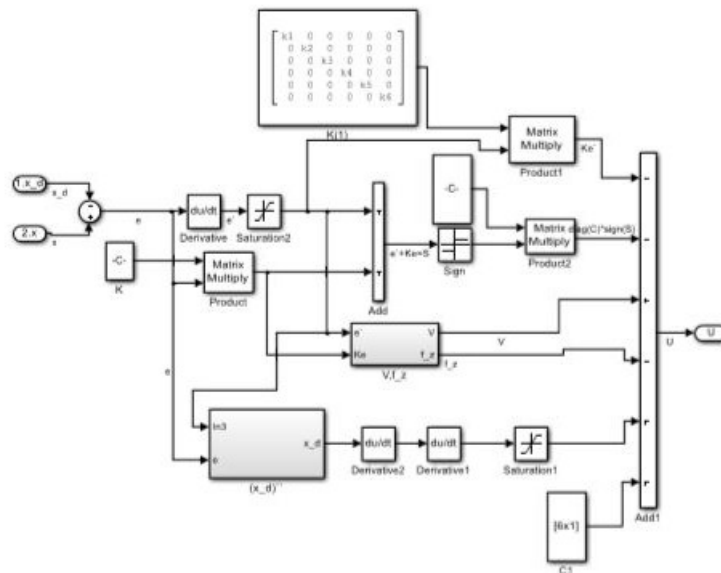


Рисунок 2.16 – Модель регулятора

На рисунку 2.17 зображено траєкторія зльоту квадрокоптера без впливу зовнішніх збурень з точки $(0; 0; 0,2)$ в $(30; -15; 20)$. Вхідний сигнал задавався за допомогою блоку Step, з параметром Step time = 2 с.

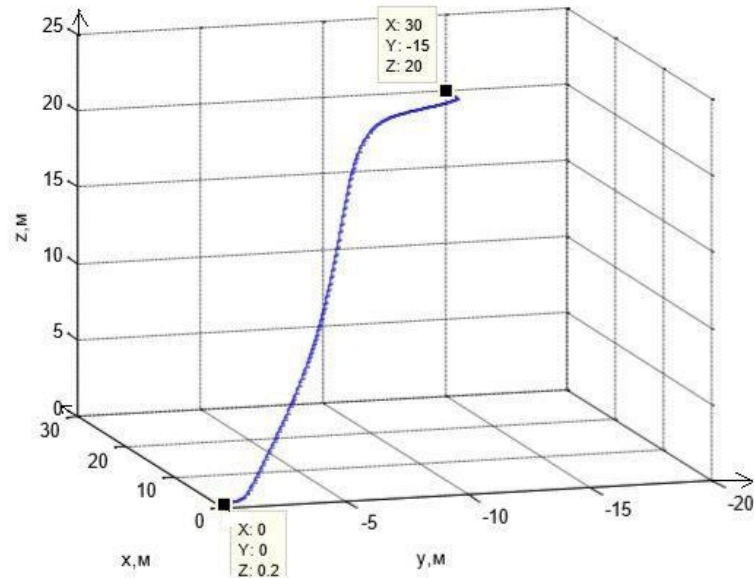


Рисунок 2.17 – Взліт квадрокоптера

Як видно з рисунку 2.17 без зовнішніх збурень квадрокоптер добре відпрацьовує вхідний вплив.

На рис. 2.18 зображена зміна кутів БПЛА.

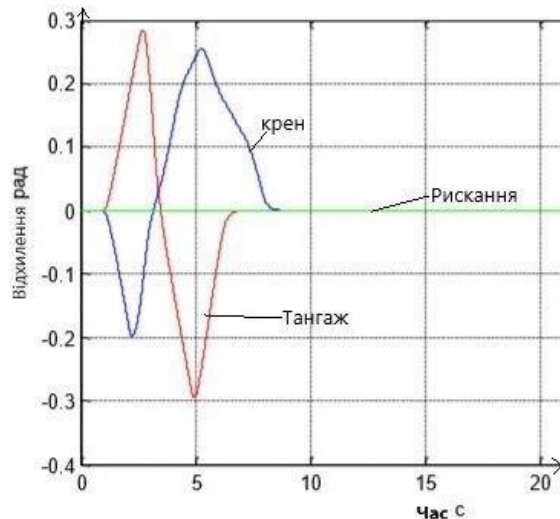


Рисунок 2.18 – Зміна кутів БПЛА

Далі оцінимо вплив зовнішніх збурень – вітру на траєкторію польоту квадрокоптера. Для початку змодельюємо вітер з постійною швидкістю в 5 м, який виникає на 5 секунд польоту і діє на квадрокоптер в площині x-y. Результат на рисунку 2.19.

СУ квадрокоптера дозволяє добре справлятися з постійним вітром: порив на 5 секунд, змістив об'єкт від нормальної траєкторії, але це відхилення було скомпенсовано шляхом зміни крену, внаслідок чого виникла статична помилка, що дозволяє відтворити вхідний вплив при зовнішньому обуренні.

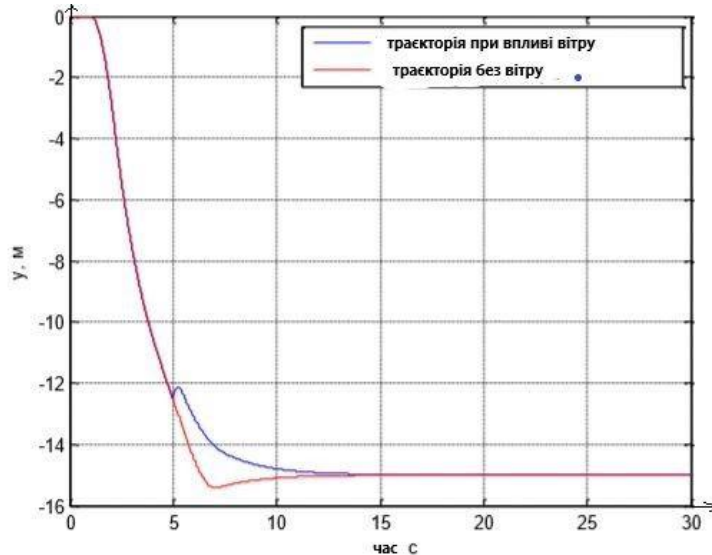


Рисунок 2.19 – Вплив вітру на траєкторію

Властивості СУ відтворювати задаючий вплив при наявності високочастотних збурень залежить від вибору режиму ковзання регулятора. Якщо коефіцієнт малий, то потужності керуючого впливу може бути недостатньо для усунення помилки. Якщо ж фізичні властивості об'єкта дозволяють вибрати більший коефіцієнт посилення, то є можливість досягти мінімальних відхилень або взагалі до точного відтворення.

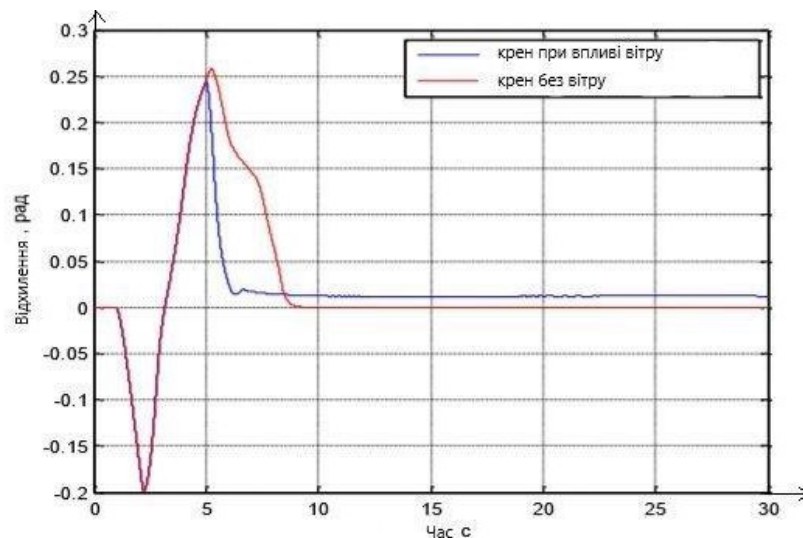


Рисунок 2.20 – Вплив вітру на крен БПЛА

Таблиця 2.1 – Показники якості системи стабілізації кутом крена БПЛА

	$T_{шт}, c$	$\varepsilon_{уст} \%$	$\sigma \%$
Без вплива вітру	9	0	51
При впливі вітру	7	0	49

СУ БПЛА дозволяє добре справлятися з постійним вітром: порив, що виник на 5 секунд, змістив об'єкт від нормальної траєкторії, але це відхилення було скомпенсовано шляхом зміни крену, внаслідок чого виникла статична помилка, що дозволяє відтворити вхідний вплив при зовнішньому обуренні. Аналогічні результати отримані по осі x. Припустимо тепер, що вітер носить періодичний характер, з амплітудою 2 м/с і частотою 1 рад/с, з 15 секунди амплітуда збільшується вдвічі. Обурення діє тільки по осі x. Результат на рис. 2.21-2.22

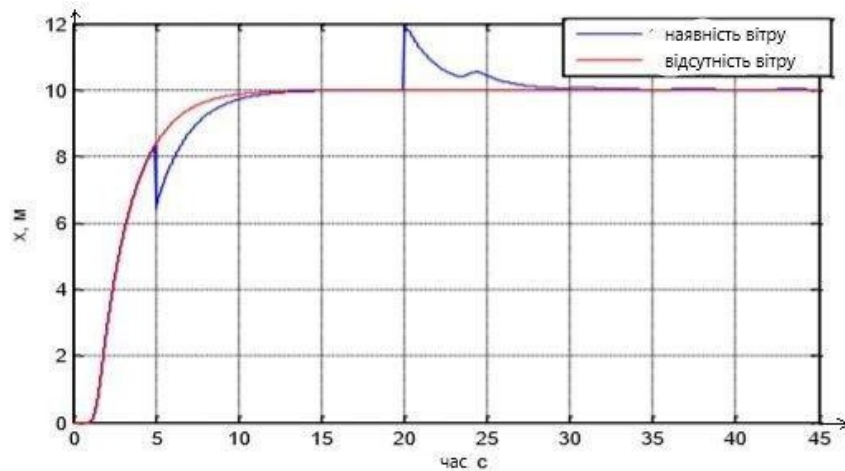


Рисунок 2.21 – Переміщення БПЛА по осі x

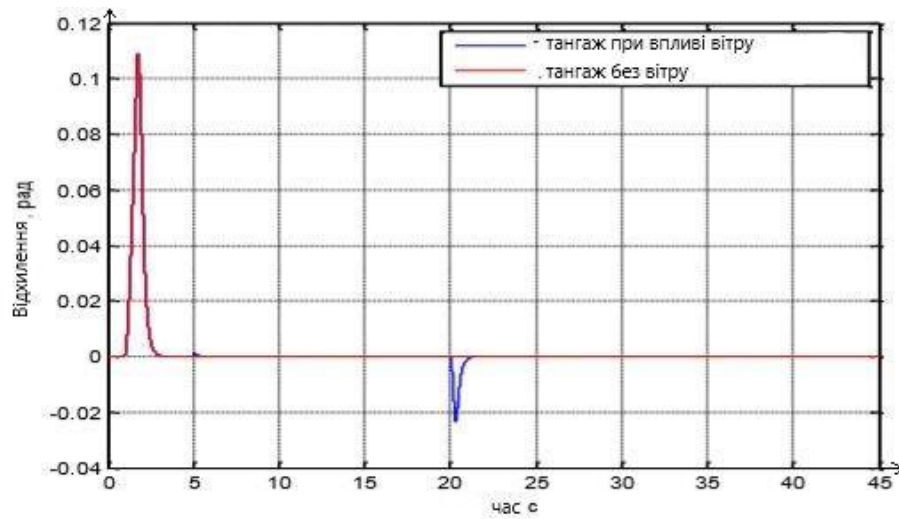


Рисунок 2.22 – Зміна тангажа БПЛА

Таблиця 2.2 – Показники якості системи стабілізації положення БПЛА

	$T_{\text{пн}}, \text{с}$	$\varepsilon_{\text{уст}} \%$	$\sigma \%$
Без вплива вітру	4	0	0
При впливі вітру	2	0	6

Як видно з рисунка 2.21-2.22 гармонійні коливання призводять до відхилення від нормальної траєкторії: найбільше відхилення спостерігається при виникненні вітру (на 5-й і 20-й секунді); коливання після виникнення вітру швидко згасають: СУ реагує на ці обурення, спостерігається дуже малі відхилення по тангажу.

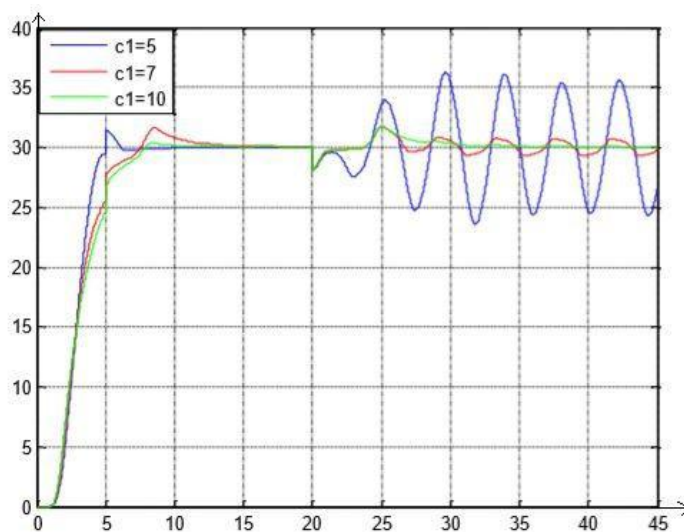


Рисунок 2.23 – Вплив нахилу площини ковзання

Як видно з рис 2.23 збільшення c_1 до 7 дозволило зменшити амплітуду усталених коливань, а при $c_1 = 10$ сталі коливання майже відсутні.

СУ з урахуванням динаміки двигунів.

Модель в Simulink є комбінацією двох попередніх. Визначемо шляхом рішення системи лінійних рівнянь тяги F_i . Тоді система рівнянь буде мати вигляд:

$$\begin{aligned} F_1 + F_2 + F_4 + F_4 &= U_1 m = a \\ 0 + F_2 + 0 - F_4 &= U_2 \cdot J_1 = b \\ -F_1 + 0 + F_3 + 0 &= U_3 \cdot J_2 = c \\ U_4 F_1 - F_2 + F_3 - F_4 &= \rho \cdot J_3 = d \end{aligned}$$

При вирішенні системи отримаємо наступне:

$$\begin{aligned} F_1 + F_2 + F_4 + F_4 &= U_1 m = a \\ 0 + F_2 + 0 - F_4 &= U_2 \cdot J_1 = b \\ -F_1 + 0 + F_3 + 0 &= U_3 \cdot J_2 = c \\ F_1 - F_2 + F_3 - F_4 &= d \end{aligned}$$

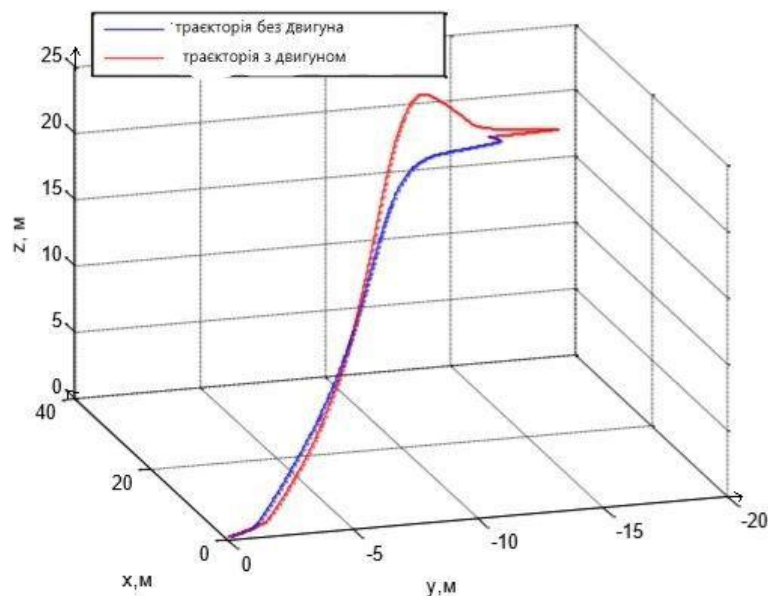


Рисунок 2.24 – Вплив динамічних властивостей двигунів на траєкторію зльоту

Незважаючи на малу інерцію вала двигуна спостерігається істотне відхилення від ідеальної траєкторії.

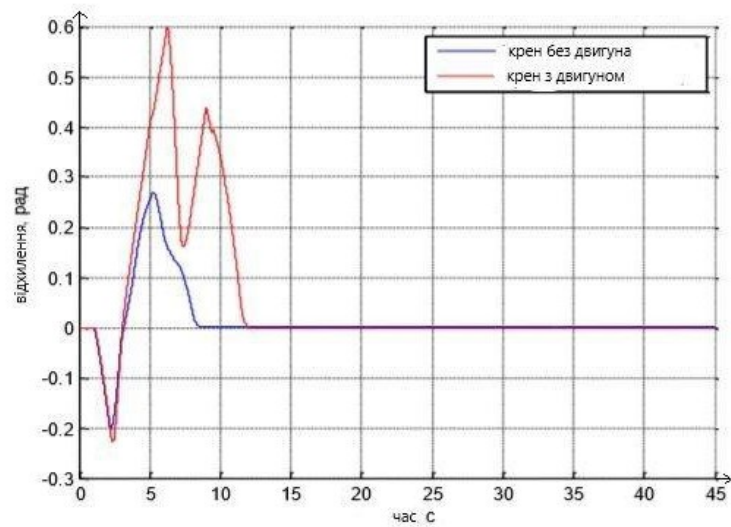


Рисунок 2.25 -Вплив динамічних властивостей двигунів на крен БПЛА

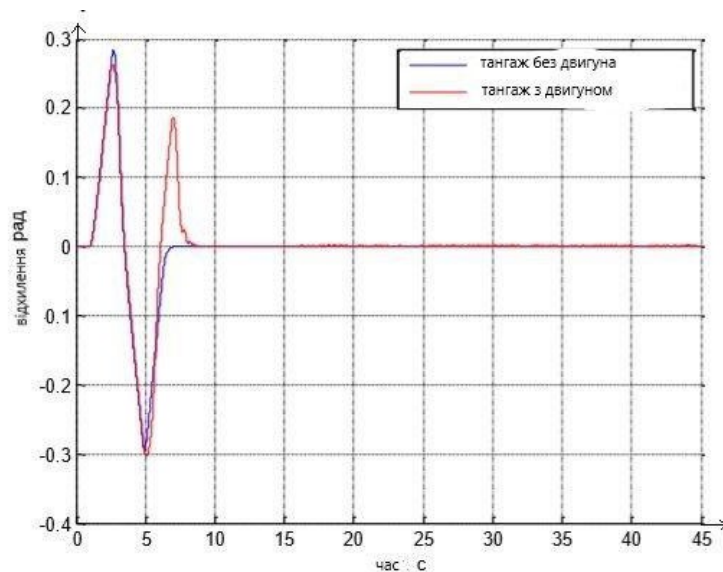


Рисунок 2.26 – Вплив динамічних властивостей двигунів на тангаж БПЛА

Таблиця 2.3 – Показники якості системи стабілізації кутового положення БПЛА

		$T_{\text{пш}}, \text{с}$	$\varepsilon_{\text{уст}} \%$	$\sigma \%$
По крену	Без двигуна	8,5	0	28
	З двигуном	12	0	60
По тангажу	Без двигуна	7	0	30

	3 двигуном	8,5	0	19
--	------------	-----	---	----

Динаміка двигунів привела до збільшення відхилення в пов'язаної СО до 20, що, в свою чергу, викликало збільшення відхилення траєкторії квадрокоптера до 2 м. Щоб позбутися цих відхилень треба налаштувати регулятор, шляхом збільшення рівня вхідного впливу. Тоді «ковзання» з більшою амплітудою дозволить утримати об'єкт управління на потрібній траєкторії. Результати наведені на рисунку 2.27.

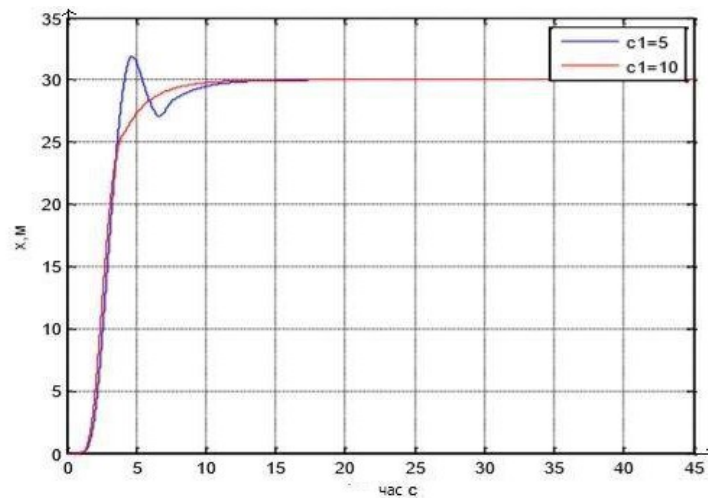


Рисунок 2.27 – Вплив коефіцієнта посилення на траєкторію БПЛА

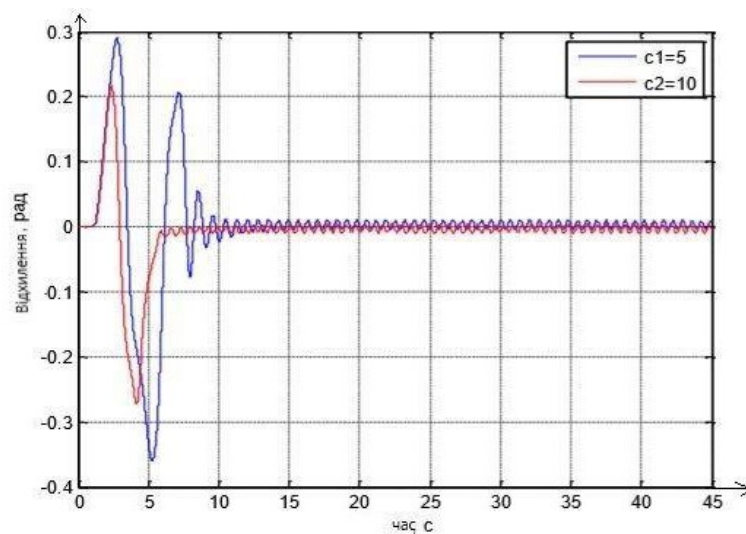


Рисунок 2.28 – Вплив коефіцієнта посилення на тангаж БПЛА

Таблиця 2.4 – Показники якості системи стабілізації кутового положення БПЛА

		$T_{пп}, c$	$\varepsilon_{уст} \%$	$\sigma \%$
По тангажу	$C1=5$	12	10	35
	$C2=10$	12	9	27

В якості останнього експерименту змодельовано політ квадрокоптера з точки А в точку В, де В - точка посадки. При цьому на 4 секунді виникає порив вітру зі швидкістю 5 м/с і тривалістю 1 с (вплив по х-у), а з 5 секунди на квадрокоптер впливає вітер періодичного характеру з амплітудою 2 м/с і частотою 1 рад/с тривалістю 5 секунд (вплив по х). В експерименті враховується динаміка двигунів.

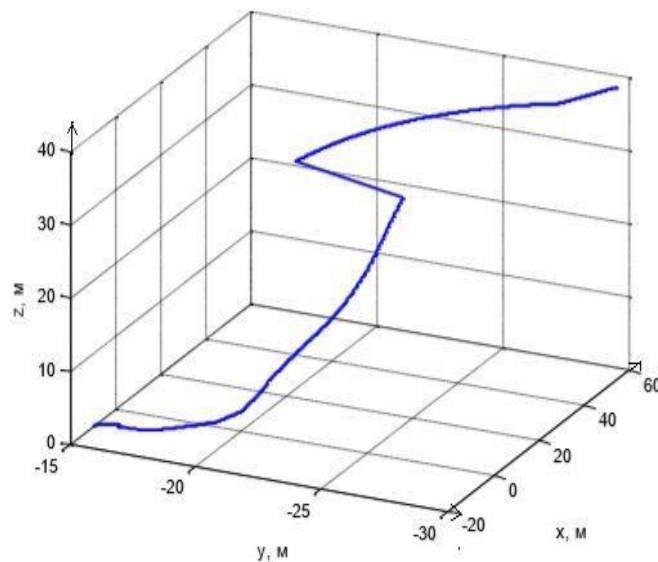


Рисунок 2.29 – Траєкторія польоту БПЛА

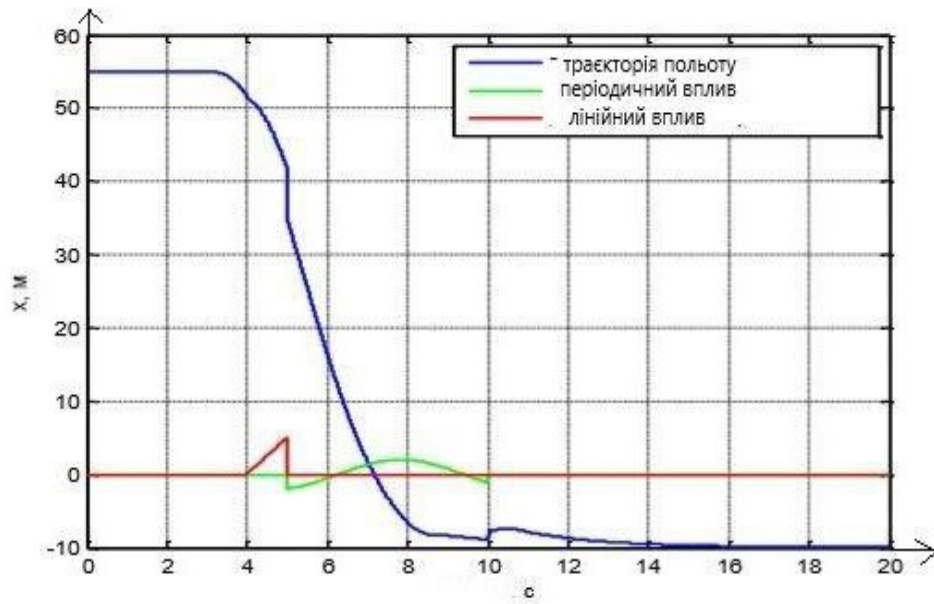


Рисунок 2.30 – Переміщення БПЛА по осі x

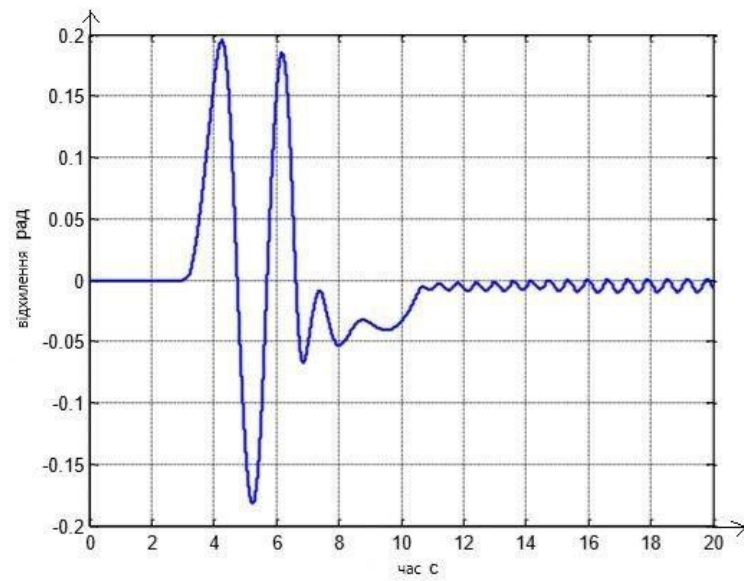


Рисунок 2.31 – Зміна крену БПЛА

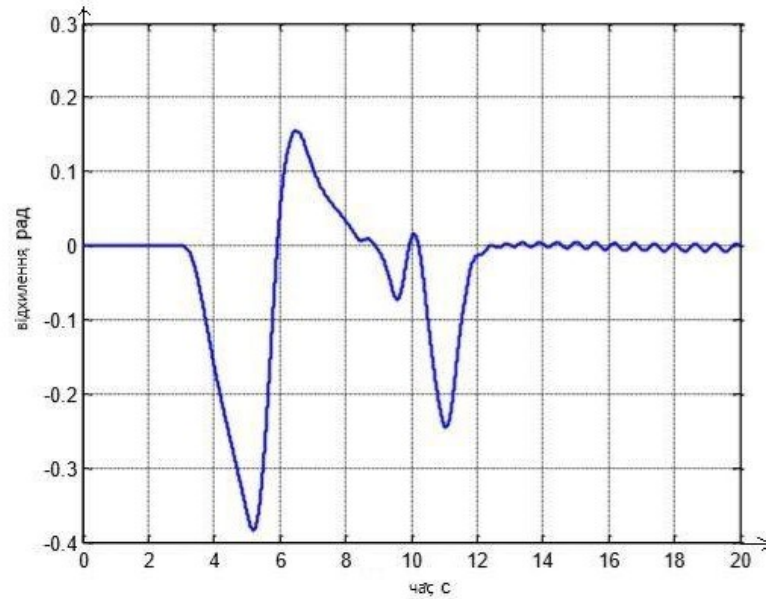


Рисунок 2.32 – Зміна тангажа БПЛА

Таблиця 2.5 – Показники якості системи стабілізації кутового положення БПЛА

	$T_{пп}, с$	$\epsilon_{уст} \%$	$\sigma \%$
По крену	10,5	7	18
По тангажу	12,2	5	38

В цьому експерименті був проведений «комплексний» вплив зовнішніх збурень на ОУ, динамічні властивості двигунів також були враховані, шляхом включення постійного струму в модель.

3 ПРОЕКТУВАННЯ СИСТЕМИ

3.1 Алгоритми реалізації моделей

Робототехнічна платформа складається з платформи, на якій знаходиться вся бортова апаратура, яка веде в рух робота. Рух виконується за допомогою 2 двигунів, що впливають на одне з двох ведучих коліс робота. Двигуни запускаються від сигналів, що подаються з драйвера двигунів постійного струму, які живляться від акумулятора. Основним сигналом для пускача є сигнал, що подається з мікропроцесорного контролера, який, на основі перетвореної інформації з датчика типу лідар, виробляє команди, що впливають на виконавчі механізми відповідно до алгоритму. У ролі візуального датчика можуть виступати камери видимого або інфрачервоного діапазону, стереокамери і камери глибини. За допомогою персонального комп'ютера і програмного забезпечення зображення, отримане камерами, перетвориться в зміну координат по ширині і довжині.

В якості методу визначення місцезнаходження вибираємо метод SLAM (Simultaneous Localization and Mapping) - метод одночасної локалізації і побудови карти.

Основна ідея алгоритму SLAM полягає в тому, що, перебуваючи в деякому положенні, БПЛА починає будувати карту навколишнього простору. Зробивши перші виміри відстаней, він їх запам'ятовує і рухається в напрямку інших об'єктів. Після того, як карта вважається добудованою, в деякому приміщенні знайдені його особливості та відстань до них виміряна БПЛА повертається на вихідну позицію. Швидше за все, він потрапить в інше місце, а не в те, в якому знаходився спочатку. Відбувається це через помилки одометра. Далі БПЛА починає друге коло вимірювань, на якому помилка скорочується.

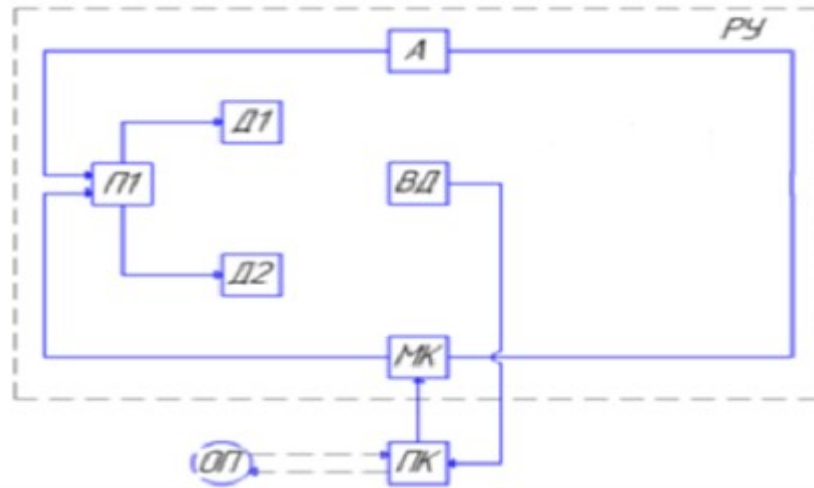


Рисунок 3.1 – Структурна схема РК

де Д1, Д2 – перший і другий двигуни;

А – акумулятор;

П1 – драйвер двигуна;

ВД – візуальний датчик;

МК – мікропроцесорний контролер;

ПК – персональний комп'ютер;

ОП – оперативний персонал;

РУ – робототехнічні пристрої.

Опис функціонування АСР:

1) сигнал з ВД надходить на ПК, де обробляється, згідно з алгоритмом і зберігається у вигляді матриці відстаней до точок навколишнього простору;

2) перетворений сигнал надходить на МК, де виробляється команда з управління виконавчими механізмами відповідно до одним із законів регулювання;

3) пускач П1, харчуючись від акумулятора, включає або вимикає двигуни по команді від МК;

4) двигуни приводять в рух РУ, змінюючи його координати;

5) пункти 1-4 повторюються до тих пір, поки БПЛА облетить всю задану місцевість і не повернеться в початкову позицію.

Перед налаштуванням середовища, в якій буде вестися розробка, дуже важливо знати обмеження стека навігації для подальшої адаптації обладнання. Існує чотири основних обмеження щодо апаратного забезпечення:

1) Стек був створений з метою охоплення тільки диференціального приводу і голономних роботів, хоча можна використовувати деякі функції з іншими типами роботів, які тут не розглядаються;

2) навігаційний стек передбачає, що БПЛА отримує повідомлення типу `twist_message` зі швидкостями X , Y і Θ і може керувати мобільною базою для досягнення цих швидкостей. Якщо інтерфейс робота не в змозі це зробити, потрібно адаптувати обладнання або просто створити вузол `ros`, який перетворює `twist_message`, що надається стеком навігації, в тип повідомлення, яке найкраще відповідає інтерфейсів БПЛА;

3) інформація про навколишнє середовище збирається з теми типу повідомлень `LaserScan`. Для плоских лазерів, таких як Hokuyo URG або SICK Laser, необхідно публікувати свої дані, для чого, досить встановити вузол `hokuyo`, `sicktoolbox` або аналогічні пакети, в залежності від типу датчика. Крім того, можна використовувати інші датчики, якщо є можливість перетворити їх дані в тип `LaserScan`.

Тільки при дотриманні всіх цих умов для використання стека навігації виконані, можна гарантувати його стабільну роботу. Тут розглядається тільки структура підсистеми навігації програмного навігаційного стека, який представлений на малюнку 3.2.

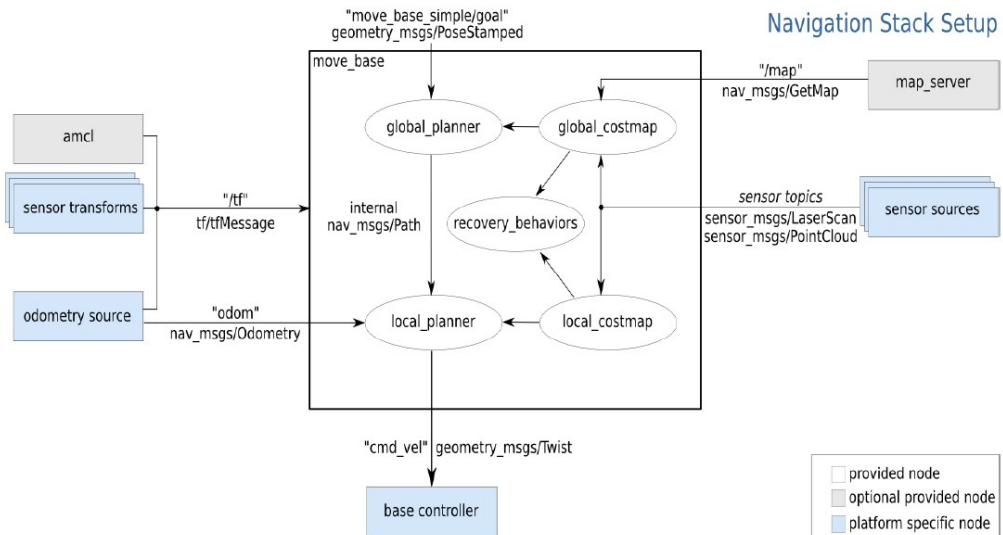


Рисунок 3.2 – Структурна схема підсистеми навігації

AMCL і картографічний сервер. Дані два блоки, відповідають за використання статичної карти. Картографічний сервер містить два вузли: сам

картографічний сервер і допоміжні модулі, спрямовані на спрощення роботи з ним, а також на поліпшення стабільності його роботи. Перший пакет є вузлом ROS, який надає статичні дані карти в якості служби ROS, а група друге виконує функцію зберігача карти, зберігаючи динамічно сгенеровану карту в файл. AMCL не володіє функціоналом для управління картами, це фактично система локалізації, що працює на відомій карті. Ця система локалізації заснована на підході локалізації Монте-Карло: вона випадковим чином розподіляє частки у відомій карті, що представляють можливі місця розташування робота, а потім використовує фільтр частинок для визначення фактичної пози робота.

Google Cartographer.

Google Cartographer, є системою локалізації, але, на відміну від amcl, він працює в невідомій середовищі, виконуючи одночасну локалізацію і картографування (SLAM). Він створює двомірну карту сітки зайнятості з використанням даних про стан робота і лазерних даних (або перетворених даних, таких як дані сенсора Kinect).

Датчики і контролер. Ці блоки системи відносяться до апаратно-програмного взаємодії і, як зазначено, є вузлами, специфічними для платформи. Джерело одометра і блоки базового контролера відносяться до використовуваного роботу, оскільки перший зазвичай публікується з використанням даних колісних енкодерів, а другий відповідає за отримання даних про швидкість з теми `cmd_vel` і гарантує, що рух робота відповідає напрямку вектора даної швидкості.

Локальні і глобальні карти витрат. Локальні і глобальні 2D карти витрат - це теми, які містять інформацію, яка представляє проекцію перешкод в 2D площині (пол), а також радіус інфляції безпеки, область навколо перешкод, які гарантують, що робот не буде стикатися з будь-якими об'єктами, незалежно від того, яка його орієнтація. У той час як глобальна карта витрат представляє всю середу (або величезну її частину), локальна карта витрат, як правило, являє собою вікно прокрутки, яке переміщається в глобальній карті витрат щодо поточної позиції робота.

Локальні і глобальні планувальники. Локальні і глобальні планувальники не працюють однаково. Глобальний планувальник бере поточну позицію робота і мета і відстежує траєкторію більш низької вартості щодо глобальної карти витрат. Однак у локального планувальника є більш цікаве завдання: він працює над локальною картою витрат, і, оскільки локальна карта витрат менше,

вона зазвичай має більшу визначення i , отже, здатна виявляти більше перешкод, ніж глобальна карта витрат. Таким чином, місцевий планувальник відповідає за створення розгортання траєкторії по глобальній траєкторії, яка здатна повернутися до вихідної траєкторії з меншими витратами. Блок управління рухом БПЛА `move_base` - це пакет, який містить в собі локальні і глобальні планувальники і відповідає за їх зв'язок для досягнення мети навігації.

Розробка підсистеми телекерування.

Головним елементом підсистеми управління є вектор управління. Вектор управління призначений для передачі команд з підсистеми високого рівня на підсистему низького рівня. Команди формуються в автоматичному режимі на основі результатів обчислень алгоритму SLAM-навігації, або в режимі ручного управління, коли команди для руху БПЛА формуються безпосередньо оператором. Функціональна блок схема алгоритму роботи вектора управління представлена на рисунку 3.3.

В якості вхідних параметрів для аналізу приймаються дані про стан глобальної та локальної карти, отриманих в програмному пакеті `RTabMap_ROS` на основі датчика візуальної інформації.

Вихідними параметрами є вектори лінійної і кутової швидкості, які і є фактичними командами для здійснення пересування платформи.

Сучасні KSLAM-системи умовно можна представити у вигляді наступного набору блоків/етапів [4 - 7]:

- 1) ініціалізації,
- 2) асоціації даних,
- 3) оцінки положення,
- 4) побудови карти/сцени,
- 5) взаємного уточнення,
- 6) відсіву помилок,
- 7) замикання циклу.

Типова комбінація зазначених блоків представлена на рис. 3.4.

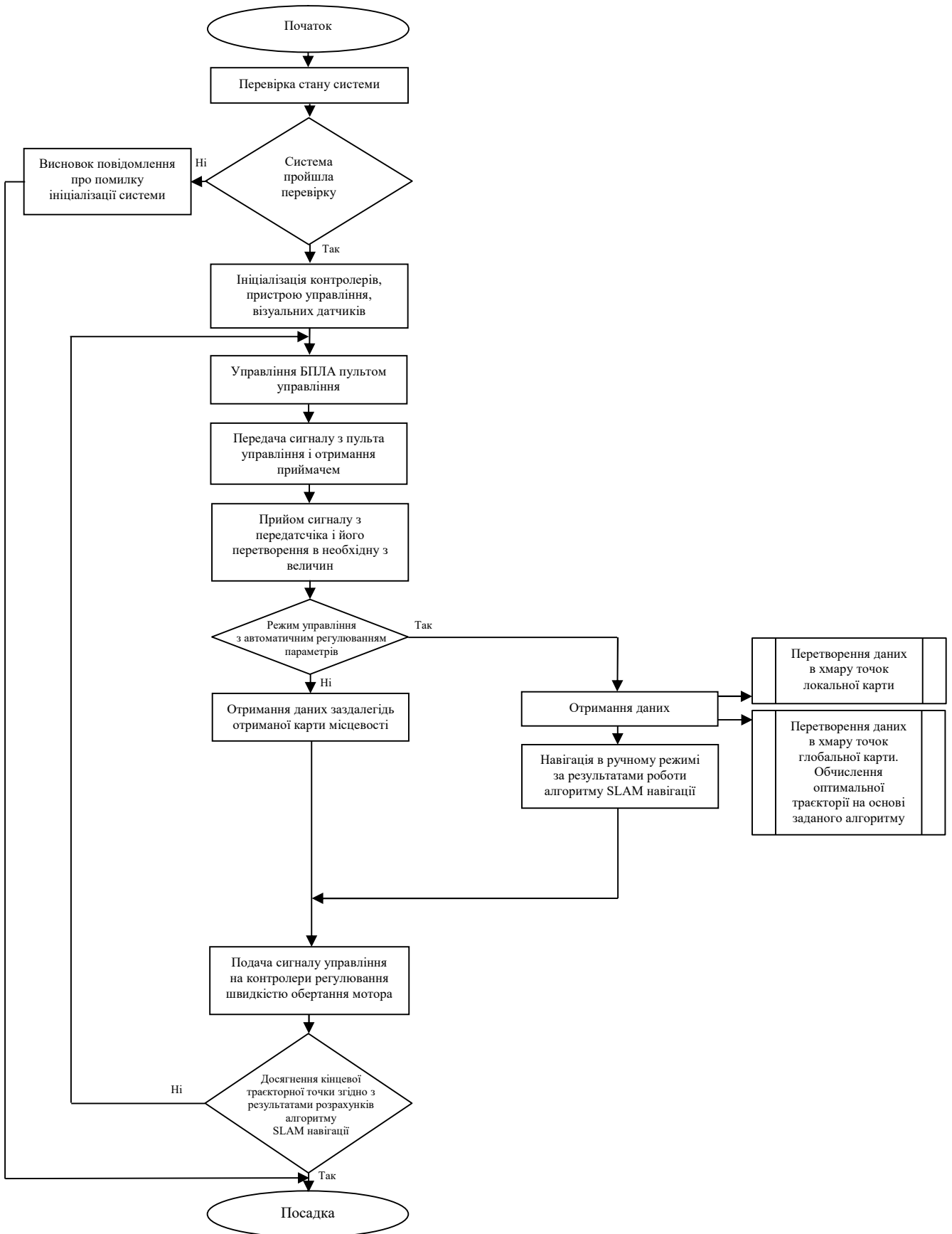


Рисунок 3.3 – Функціональна схема блоку вектора управління підсистеми навігації

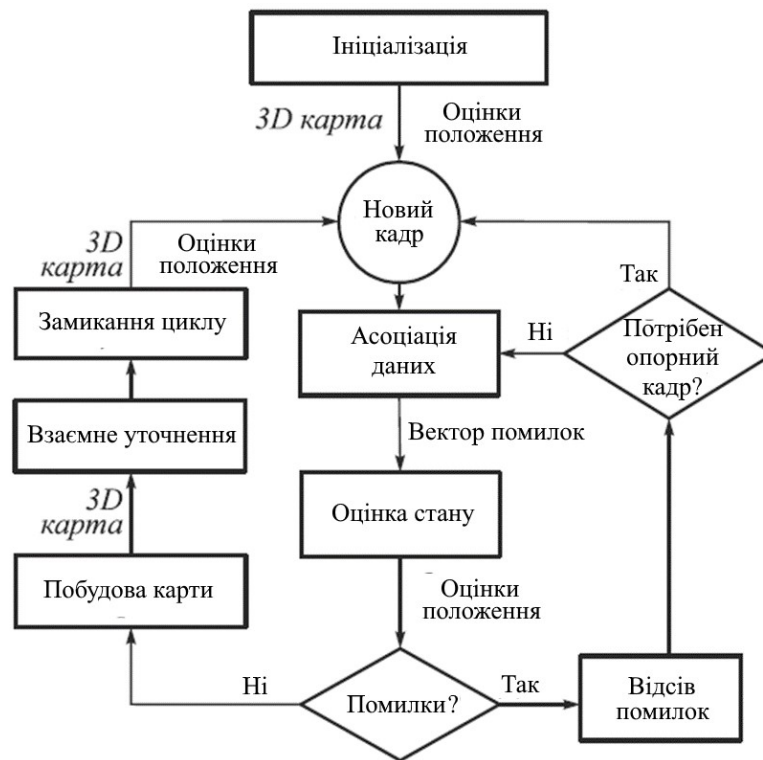


Рисунок 3.4 – Загальна архітектура KSLAM-системи

Блок ініціалізації (англ.: visual initialization) призначений для установки стартового стану.

Блок асоціації даних (англ.: data association) призначений для знаходження відповідності між уже знайденими на попередніх кадрах орієнтирами і орієнтирами, знайденими на поточному кадрі. Підходи, які використовуються для реалізації цього блоку, зазвичай поділяють на прямі (англ.: direct), засновані на ознаках / дескрипторах (англ.: feature-based), і змішані (англ.: hybrid). Методи, що реалізують прямий підхід, в свою чергу, поділяють на «щільні» (англ.: dense) і «напів-щільні» (англ.: semi-dense). Для «щільної» асоціації характерно знаходження відповідностей між усіма точками кадру, в той час як альтернативою є вказівка відповідностей тільки для частини кадру, де присутні різкі перепади яскравості, зазвичай відповідні кордонів і краях об'єктів. На відміну від прямого підходу, що має надзвичайно високу обчислювальну складність, признакова підхід припускає знаходження на зображенні відносно невеликого числа особливих точок (англ.: keypoints, feature points), набір ознак яких - дескриптор - характеризує локальну околиця

відповідної точки. Існує цілий ряд детекторів (Hessian corner detector, Harris detector, ShiTomashi corners, FAST і ін.) І дескрипторів (SIFT, SURF, HOG, ORB і ін.) Особливих точок, які можуть бути використані для вирішення цього завдання. Огляд деяких з них представлений в роботах [7, 10]. Зіставлення в цьому випадку проводиться на основі пошуку сполучених пар точок на різних кадрах з найбільш близькими дескрипторами.

Очевидно, що невірно знайдені асоціації, встановлені відповідним блоком, призводять до принципової неможливості вірного рішення SLAM-проблеми. Для виключення невірно знайдених асоціацій і відповідних (аномальних) помилок в KSLAM-системах передбачений блок відсіву помилок (англ. : failure recovery). Одним з типових варіантів усунення невірно знайдених асоціацій є скорочення (спрямоване або випадкове) числа аналізованих пар асоціацій, що реалізується алгоритмами, типу RANSAC [11] або пошуку найбільшої спільної підсистеми [9].

Блок оцінки стану (англ.: pose estimation) призначений для отримання оцінки стану засобу спостереження, як правило, його положення і орієнтації в деякій системі координат (наприклад, пов'язаної з початковим положенням камери).

Даний блок зазвичай реалізується з використанням передбачення положення і подальшого уточнення положення. Передбачення положення може виконуватися на основі аналізу моделі руху (наприклад, припускаючи прямолінійність або гладкість траєкторії), на основі інформації з допоміжних датчиків або систем управління, а також шляхом агрегації цих двох підходів. Особливий інтерес представляє другий і третій підходи. Уточнення положення, отриманого на основі передбачення, проводиться за рахунок «триангуляції» точки зйомки щодо відомих і асоційованих орієнтирів.

Знайдене в такий спосіб положення камери є попередніми (оскільки положення орієнтирів в подальшому також коригується) і уточнюється на етапі взаємного уточнення.

Блок побудови карти/сцени (англ.: map generation) відповідальний за побудову уявлення тієї частини спостерігається сцени, яка до поточного ключового кадру не спостерігалася. Зазвичай розрізняють два варіанти подання сцени: метричний і топологічний. При метричному поданні кожен орієнтир отримує явні метричні координати в обраній SLAM-системою системі координат, що обчислюються на підставі спостережуваних проекцій орієнтирів на ключових кадрах. В силу існуючих помилок визначення на знімку координат

проекцій орієнтирів, дискретності самого зображення, а також помилок у визначенні координат системи реєстрації метричні координати орієнтирів визначаються також з помилкою. Власне мінімізація помилки, одержуваної як сума відхилень реально зареєстрованих координат проекцій від координат проекцій пропонованого рішення, і є типовою критерій для побудови карти. Топологічний уявлення сцени передбачає побудову графа, вузли якого відповідають зареєстрованим ключовим кадрам, а дуги відповідають знайденим асоціаціям за орієнтирами. Зазначені подання також можуть використовуватися спільно.

Блок взаємного уточнення (англ. : BA - bundle adjustment / pose graph optimization / map maintenance) призначений для одночасного уточнення метричного уявлення реєструється сцени і параметрів станів пристрої реєстрації. Різні алгоритми, що реалізують цей блок, використовують різні критерії: глобальну помилку по всіх кадрах, локальну помилку по підмножеству кадрів і т.п. В силу об'єктивної нелінійності ключового критерію, що мінімізує середньоквадратичне помилку проектування орієнтирів, найбільш типовими алгоритмами рішення BA-завдання є: градієнтний алгоритм, алгоритм Ньютонна, алгоритм Левенберга - Маркуада [12 - 13].

Блок замикання циклу (англ. : loop closure) використовується в тих випадках, коли на практиці здійснюється повернення системи реєстрації (робота) на позицію, з якої був зареєстрований ключовий кадр. В цьому випадку виникає додаткова можливість уточнення всіх шуканих параметрів KSLAM-проблеми за рахунок обліку замкнутості траси пристрої реєстрації.

3.2 Вибір засобів реалізації

Для того, щоб цей проект функціонував ефективно і добре відповідав своєму призначенню, йому потрібно задовольнити вимоги, визначені проблемою. Оскільки рама є несучою конструкцією, яка складається з поєднання лінійних елементів, вона повинна витримувати навантаження, забезпечувати міцність і стійкість БПЛА.

В якості матеріалів використовують безліч матеріалів: пластмаси, скловолокно, алюміній і карбон. Найкращим варіантом є рама з карбону, оскільки він міцний, легкий та дорожчий.

Основним параметром рами для квадрокоптера – це її розмір (відстань між діагонально розташованими двигунами). Залежно від цієї відстані можна розділити квадрокоптери на «класи» (табл. 3.1).

Таблиця 3.1 – Класи рам квадрокоптера

Розмір	Корисне навантаження	Призначення	Дальність польоту (залежить від апаратури)
≤ 200 мм	---	іграшкові	≤ 30 м
≥ 250 мм	відеокамера початкового рівня, GPS	спортивні, для освоєння професійної техніки	≤ 50 м
≥ 350 мм	відеокамера середнього рівня, GPS	Комерційне застосування, розваги	≤ 1000 м
≥ 550 мм	Високоякісна фото/відео камера, GPS	Геологічна розвідка, моніторинг	≥ 5000 м

Для довгочасного моніторингу місцевості необхідна рама останнього класу, яка матиме високу міцність при відносно малій масі, та матиме можливість встановлення на ній високоякісної апаратури для відеозйомки.

Було обрано раму Tarot FY650 "IRON MAN" TL65B01. Дана рама складається та дає високу мобільність. Малі габарити і велика міцність дозволяє літати в глишині, що важливо при аерофотозйомці, картографуванні, розвідки місцевості. Передбачено регулювання центру тяжіння в залежності від ваги встановленої фотокамери, відеокамери.



Рисунок 3.5 – Рама квадрокоптера

Характеристики:

- діаметр рами 650мм;
- діаметр лучів: 16мм;
- вага пустої рами: 476г.

З урахуванням маси, швидкості і тривалості польоту, енергоспоживання і розмірів, підбирається електродвигун для БПЛА. На квадрокоптер можна «поставити» два типи електродвигунів – колекторний тип і бесколекторний. Колекторні двигуни ставляться на невеликі і дешеві моделі. Вибір пав на бесколекторний двигун постійного струму (БДПС), бо система повинна мати велику зносостійкість.

Використання БДПС краще, ніж колекторний двигун, і це обумовлено рядом причин:

- більш простою конструкцією, меншими вагою і розміром;
- вище ККД і показник потужності на кілограм власної ваги;
- більш широкий діапазон швидкості обертання гвинтів;
- менший знос за рахунок відсутності щітково-колекторного вузла;
- БДПС практично не створюють радіоперешкод.

Недолік БДПС є електронний блок управління, який управляє кількістю оборотів двигуна.

При виборі двигунів слід керуватися наступними параметрами:

- споживаний струм (вимірюється в амперах [А]);
- Kv-rating (показник обертів осі двигуна за одну хвилину при певному напруженні);
- підйомна сила (вимірюється в Ньютонах [Н] або кілограмах [Кг]);
- ККД.

Показник обертів:

$$\text{RPM} = K_v \cdot U \quad (3.1)$$

Мінімальна підйомна сила моторів:

$$F_{\text{під}} = \frac{m_{\text{кк}} \cdot 2}{4} \quad (3.2)$$

де $m_{\text{кк}}$ - маса квадрокоптера.

Обраний двигун T-Motor MN3510-15 зі своїми параметрами і характеристиками дозволяє при комп'ютерному моделюванні наблизитись до реального об'єкта.

Таблиця 3.2 – Характеристики двигуна T-Motor MN3510-15.

Виробник	T-Motor	Номінальний струм	12.3 А
Модель	MN3510-15	Струм холостого ходу	0.5 А
Клас	Безколекторний	Макс. тривалий струм	22 А
Вид	outrunner	Макс. тривала потужність	495 Вт
Конфігурація	12N14P	Опір	65 мОм
Кількість полюсів	14	Вал	Гладкий вал
KV	630	Діаметр валу	4 мм
Макс. тяга	1250 — 1850 г	Кріплення мотору	25x25 мм
Живлення	11.1 — 14.8 В	Розміри	28,5x41,8x41,8 мм
		Вага	97 г (без дротів)

Самі по собі двигуни не створюють підйомну силу – необхідно доповнити їх пропелерами. Пропелери перетворюють енергію обертання вала двигуна в підйомну силу.

Одними з найважливіших характеристик пропелера є його розмір і кут нахилу лопатей, що вказує на скільки піднявся б пропелер за один оборот навколо своєї осі. Розміри величин вказуються в дюймах. При збільшенні розміру пропелера збільшується підйомна сила, однак зростає момент на валу двигуна і збільшується споживчий струм. Установка великих гвинтів з великим кроком підійде до розроблюваного квадрокоптера, адже вони дадуть стабільність польоту. Обрані карбонові пропелери 1540 Carbon Fiber Propeller CW&CCW.

Акумуляторна батарея (АКБ) являється елементом живлення усіх систем БПЛА та двигунів, що працюють від постійного струму.

При виборі АКБ необхідно враховувати наступні критерії:

- Ємність батареї ;
- Струмівіддача ;
- Структура батареї ;
- Вага та габарити ;

Найкращі характеристики мають літій-полімерні акумулятори (LiPo - акумулятори). Вони мають низький саморозряд, відсутність ефекту пам'яті, великий діапазон робочих температур, малий перепад напруги в міру розряду.

Недоліки - це не найвища щільність заряду, мала кількість робочих циклів (800-900) і підвищена пожежонебезпека.

LiPo-акумулятори складаються з однієї або декількох відсіків, з'єднаними послідовно. Кожен елемент забезпечує номінальну напругу 3.7В, і відповідає маркуванню 1S. Відповідно, чим більше відсіків, тим більша напруга. Тому позначення напруги рівня 1S =3.7В, 2S =7.4 В, 3S = 11.1 В, 4S = 14.8 В і т.д. Рівень вихідної напруги напряму впливає на кутову швидкість встановлених електродвигунів.

Довготривалість перебування квадрокоптера у повітрі залежить не тільки від вхідного струму з АКБ, а й від вихідного струму, точніше за рахунок його зменшення. Занадто низька токовіддача (позначається індексом «С») може призвести до пошкодження АКБ, а занадто великий індекс «С» може зменшити польотний час.

В якості АКБ було обрано акумулятор Dinogy Li-Pol 11000mAh 4S 25C 40x61x179мм T-Plug (рис. 3.6) з характеристиками, наведеними в таблиці 3.3.



Рисунок 3.6 – Акумулятор Dinogy Li-Pol 11000mAh 4S 25C

Таблиця 3.3 – Характеристики Акумулятора Dinogy Li-Pol

Тип	Li-Po	Струмвіддача	25 С
Ємність	11000 мАч	Струм заряду	5 С
Напруга	14.8 В	Конектор	T-Plug
Конфігурація	4S	Балансуючий роз'єм	JST-XH
Розміри	190x59x38 мм	Вага	879 г

Усі польотні батареї LiPo йдуть з набором дротів та роз'ємів. Через силовий кабель виконується підключення батареї до квадрокоптера або виходу

зарядного пристрою. Вибір з'єднувача для силового дроту залежить від максимального вихідного току АКБ. В якості з'єднувачів для балансуєчого дрота зазвичай використовують роз'єм JST-XH.

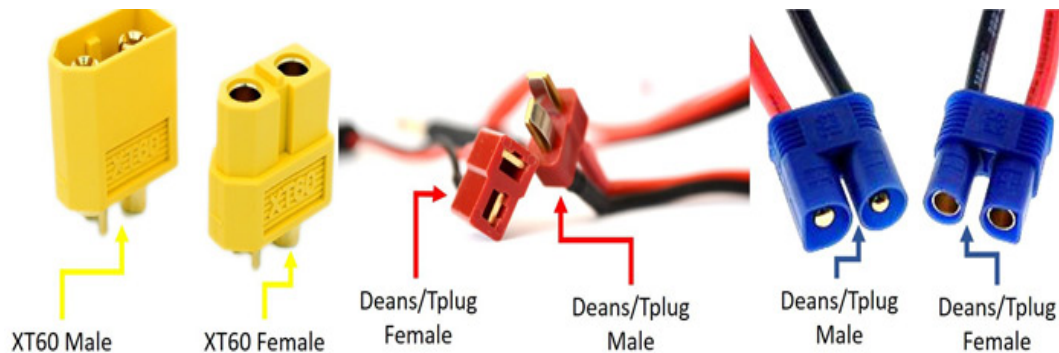


Рисунок 3.7 – Типи з'єднувачів для АКБ

Був обраний силовий модуль, для комутації АКБ з ECS APM ВЕС 3А XT60 plug (рис.3.8).

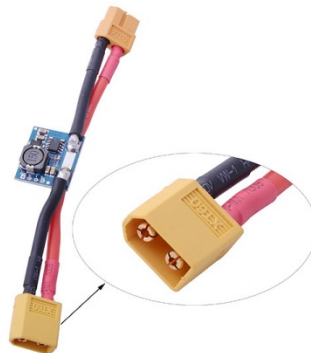


Рисунок 3.8 – Силовий модуль APM ВЕС 3А XT60 plug

При виборі регулятора швидкості необхідно враховувати: максимальний споживаний двигуном струм, а також напругу акумуляторної батареї. Слід вибирати акумулятор з максимальним пропускним струмом на 10-20% вище, споживаного струму двигунами. Номінальна напруга повинна дорівнювати або бути більше, ніж напруга акумуляторної батареї.

Оскільки максимальній споживчий струм двигуна дорівнює 22А, а напруга АКБ становить 14.8В, було обрано ESC HOBBYWING SKYWALKER Quattro 25А.



Рисунок 3.9 – Регулятор ходу (ECS)

Характеристики регулятора ходу:

- напруга 7.4 — 14.8 В;
- номінальний струм 25 А;
- піковий струм 30 А;
- частота PWM 50-432 Гц;
- охолодження пасивне;
- розмір 70x62x11 мм;
- вага 112 г;
- вихідна напруга 5В;
- вихідний струм 3А.

Льотний контролер – це електронний пристрій, що складається з плати із процесором та вмонтованими датчиками, що керують польотом БПЛА.

Метою роботи є побудова такого БПЛА, який здатний виконувати основні польотні режими:

- 1) Stabilize – зліт і посадка; для утримання горизонту задіяні гіроскоп і акселерометр, компас використовується для контролю і корекції.
- 2) AltHold – режим утримання висоти; додається використання барометра, який сприяє утриманню висоти по тиску повітря.
- 3) Land – режим автоматичної посадки в поточному положенні; барометр використовується для контролю висоти.
- 4) Simple – режим, який дозволяє не враховувати орієнтацію БПЛА щодо пілота; найважливіший компас.
- 5) Loiter – режим утримання точки по координаті і висоті; використовується GPS; підходить для відеозйомки і фото.
- 6) Auto – політ по запланованим точкам; шлях може створюватися через ПЗ наземної станції перед польотом вручну.

7) Failsafe – режим порятунку, який в разі втрати зв'язку з наземною станцією відправляє БПЛА в точку, де зроблений запуск двигунів або іншу заплановану точку.

Обрано льотний контролер PixHawk нового покоління на базі 32-бітного процесора STM32 з відкритим вихідним кодом ArduPilot. PixHawk має більшу ефективність у порівнянні з ArduPilot за рахунок більш продуктивного процесорного ядра та нових функцій, а саме «чорний ящик» (запис польотної інформації на SD-карту), самонавчання польоту та адаптивні фільтри.

Характеристики:

- Розмір: 81x44x15mm
- Вага: 33.1g

Мікропроцесор:

- 32-bit STM32F427 Cortex M4 ядро з FPU
- 168 MHz / 256 KB RAM / 2 Мб Flash
- 32 біт STM32F103 failsafe co-processor

Датчики:

- ST Micro L3GD20 3-вісь, 16-bit гіроскоп
- ST Micro LSM303D 3-вісь, 14-ти розрядний акселерометр / магнітометр

- Invensense MPU 6000 акселерометр / гіроскоп з 3 осями
- Барометр MEAS MS5611

Інтерфейси:

- 5x UART (послідовні порти)
- 2x CAN
- Spektrum DSM / DSM2 / Супутник DSM-X® сумісний вхід до DX8
- Futaba S.BUS® сумісний вхід і вихід
- PPM сигнал
- RSSI (PWM або напруга) вхід
- I2C
- SPI
- 3.3 і 6.6-вольт входи АЦП
- Зовнішній порт mUSB

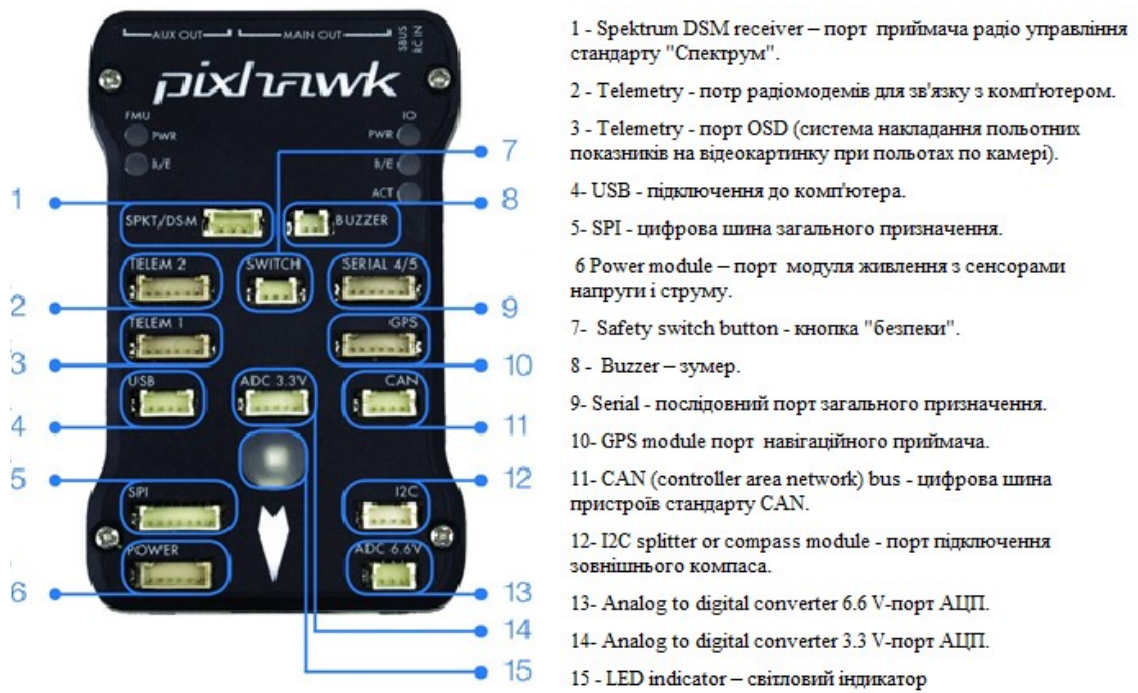


Рисунок 3.10 – Льотний контролер PixHawk

Функціональні схеми акселерометра і магнітометра ST Micro LSM303D, гіроскопа ST Micro L3GD20, барометра MEAS MS5611, яким обладнаний льотний контролер PixHawk приведені на рис.3.11-3.13.

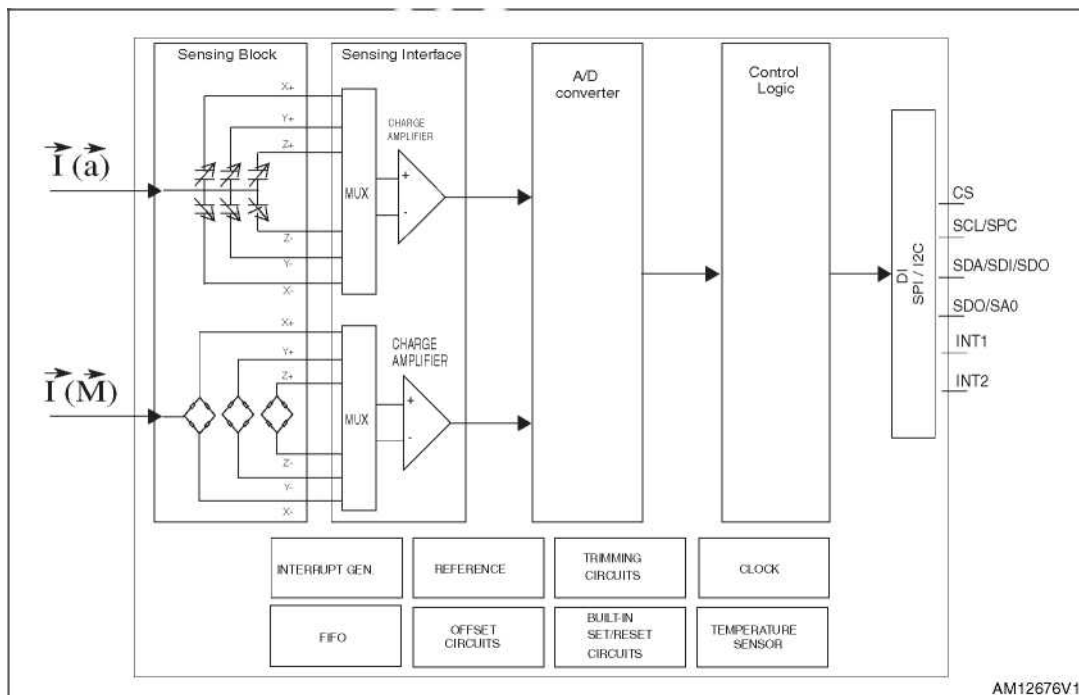


Рисунок 3.11 – Акселерометр і магнітометр ST Micro LSM303D

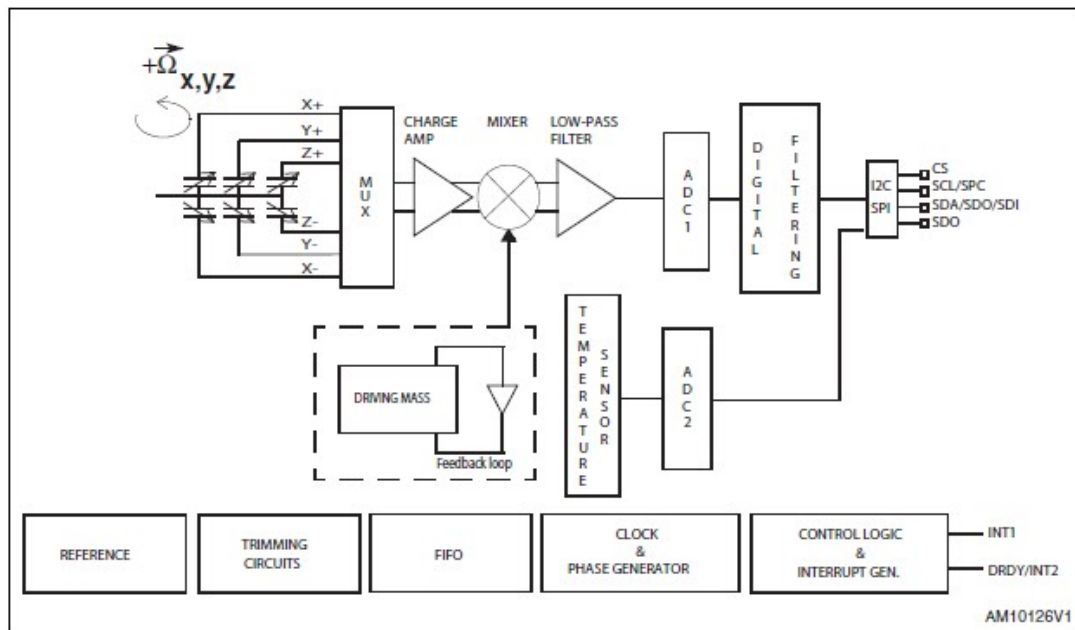


Рисунок 3.12 – Гіроскоп ST Micro L3GD20

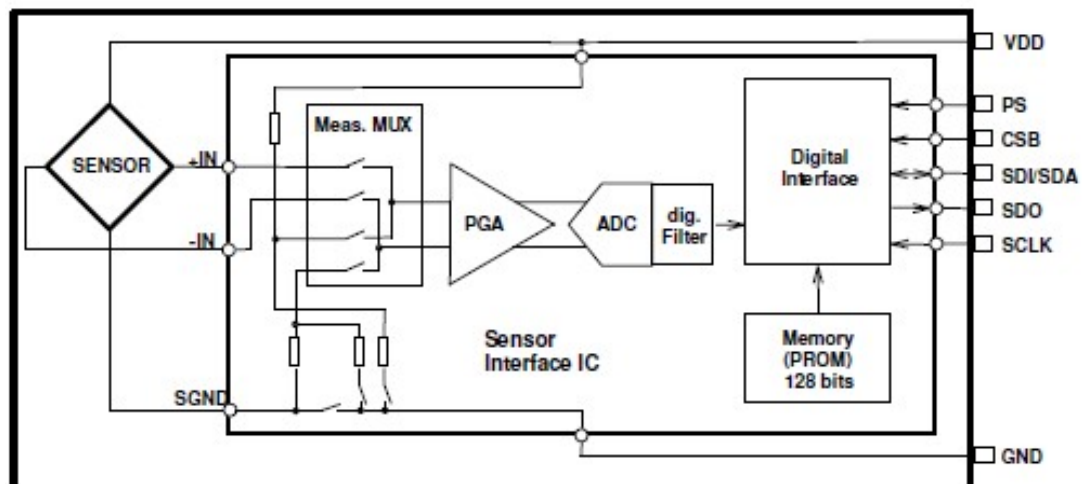


Рисунок 3.13 – Барометр MEAS MS5611

Архітектура льотного контролеру.

PX4 розробляє серію апаратних модулів, які використовуються для керування транспортними засобами. Це ручне і автоматичне керування безпілотних літальних апаратів та інших систем автоматичного керування. Усі ЛА на основі PX4 мають одну кодову базу.

Архітектура PX4 складається з двох основних рівнів:

- верхня частина – проміжне ПЗ являє собою загальний рівень роботизування, може підтримувати будь-який тип автономного робота;

- нижня частина – стек польоту складається з системи оцінки та управління польотом.

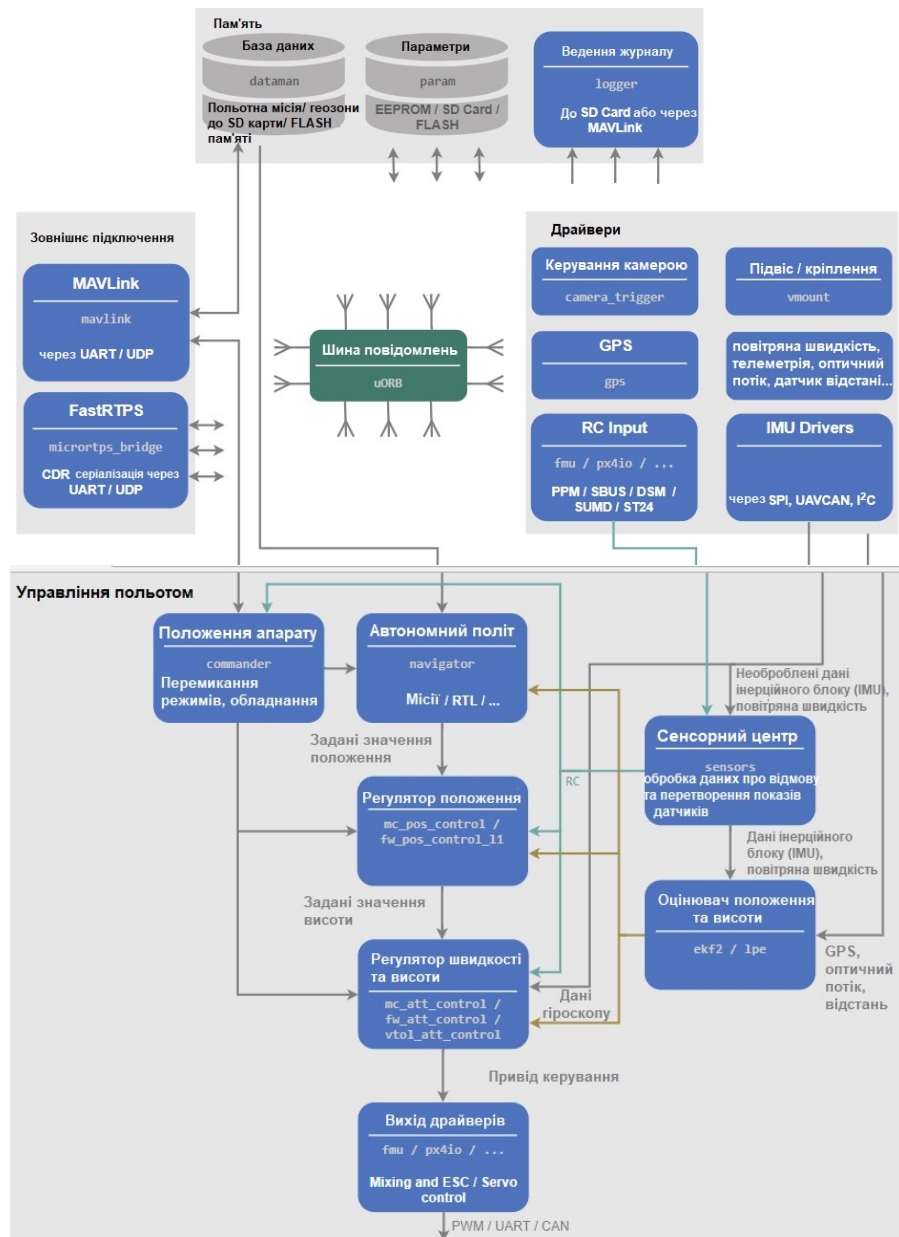


Рисунок 3.14 – Схема архітектури програмного забезпечення високого рівня

Стек польоту складається з набір алгоритмів наведення, навігації і управління для автономних БПЛА. Він включає в себе оцінювач положення (астиматор) та контролери для БПЛА з нерухомим крилом або мультироторних типів.

На рис.3.15 показана схема загального вигляду стека польоту. Вона містить зв'язок від датчиків, входу RC і автономного управління польотом.



Рисунок 3.15 – Схема стека польоту

Оцінювач висоти та позиції приймає входи датчиків, об'єднує їх і обчислює стан БПЛА (наприклад, положення від датчиків IMU).

Контролер позиції приймає задане значення і вимір або оцінює стан (змінна процесу) в якості вхідних даних. Мета контролера: значення змінної процесу повинно відповідати заданій точці, де вихідний сигнал служить поправкою. Наприклад, контролер позиції в якості входів приймає задані значення положення, змінна процесу - це поточне оцінене положення, а вихід - це задане значення положення і тяги, яке переміщує БПЛА в необхідне положення.

Мікшер приймає команди (наприклад, повертання праворуч) і переводить їх в окремі команди двигуна в обмеженнях. Цей перехід специфічний для типу БПЛА і залежить від обертальної інерції двигунів або розташування двигуна щодо центра ваги.

Приймач супутникової навігації потрібен для:

- здійснення автоматичного польоту по заданому маршруту;
- визначення поточної відстані до точки зльоту, швидкості і висоти польоту при польотах по FPV;
- забезпечення автоматичного повернення до точки зльоту при втраті сигналу управління.

Для забезпечення супутникової навігації було обрано GPS Ublox Neo m8n 8N (рис.3.16).



Рисунок 3.16 – GPS Ublox Neo m8n 8N

Відеообладнання БПЛА складається з наступних компонентів:

- відеокамери, курсова та основна;
- відеокомутатор;
- відеопередавачі;
- модуль накладання інформації на зображення (OSD);
- стабілізований підвіс камери;
- всенапрявлена антена;
- джерело живлення відеообладнання.

Наземна частина відеоканалу:

- відеоприймачі;
- антени, всенапрявлена і спрямована;
- система управління положенням антени (трекер);
- монітор або відеоокуляри;
- відеорекордер;
- джерело живлення.

У професійній зйомці для польотів по FPV технології використовують дві камери: курсову і основну. FPV – (англ. «First Person View» - вид від першої особи) спосіб управління БПЛА за допомогою відеокамери на борту. Відео в реальному часі передається на пульт управління і дозволяє керувати БПЛА поза межами огляду. Курсова камера дає зображення середньої якості. Вона монтується на рамі БПЛА в фіксованому положенні, зручному для оператора і направляється злегка під кутом до землі. Камера імітує вид з кабіни пілота і дає уявлення куди і під яким кутом летить БПЛА, чи є перешкоди на шляху. На зображення з цієї камери, як правило, накладається телеметрійна інформація (OSD) [4]. Однак, для розроблюваної системи прийнято рішення у виборі однієї камери Xiaomi Yi Sport Black (рис.3.17) на керованому

стабілізованому підвісі Tarot T4-3D (рис.3.18). Такий підхід більш економічний, відпадає потреба у відеоконвертері, звільняється один з радіоканалів.



Рисунок 3.17 – Камера Xiaomi Yi Sport Black

Характеристики камери:

- матриця CMOS 1/2.3", Sony Exmor R, 16 Мп;
- кут огляду 155°;
- режим відеозйомки 1080p 60 кадрів/сек; 1080p 30 кадрів/сек; 720p 120 кадр/с; 480p 240 кадр/с;
- режим фотоз'йомки 16 Мп (4608x3456);
- живлення Замінний, Li-Po, 1010 мА*г;
- ПЗП microSD (до 64 ГБ);
- вага 72 г.



Рисунок 3.18 – Трьохосьовий підвіс Tarot T4-3D

Характеристики трьохосьового підвісу:

- точність $\pm 0.02^\circ$;
- кут повороту -120° to $+ 15^\circ$;
- максимальна швидкість повороту $\pm 200^\circ/s$;
- робоча температура $-20^\circ\text{C} \sim 50^\circ\text{C}$;

- напруга живлення 2S ~ 6S LiPo;
- розмір 99мм×88.5мм×105.6мм;
- вага 178 г.

Система FPV використовує приймачі та передавачі відеоданих. Слід оперувати наступними частотами:

- 900 мГц;
- 1.2 гГц;
- 1.3 гГц;
- 2.4 гГц;
- 5.8 гГц.

Проникаюча здатність більше там, де нижче частота, але у таких антен більші геометричні розміри. Найоптимальніші антени, які працюють на частоті 2.4 гГц.

Один з важливих показників - потужність. Чим більше потужність, тим більше радіус дії. Щоб збільшити дальності вдвічі, потужність повинна бути збільшена в чотири рази. Також, слід враховувати легальність використання в країні або регіоні обраної FPV частоти і потужності.

Система телеметрії (OSD) необхідна для передачі з борту важливої технічної інформації в режимі реального часу, а також, якщо потрібно, для запису цієї інформації в пам'ять. Склад інформації залежить від потреб оператора та призначення системи:

- режим роботи польотного контролера;
- поточні координати GPS;
- кількість доступних супутників;
- дальність до точки старту;
- висота;
- час польоту;
- лінійна швидкість;
- вертикальне прискорення;
- показання акселерометрів (крен);
- напруга батарей;
- споживаний струм;
- температура батарей;
- показання компаса;
- обороти двигунів.

На рис.3.19 приведена структурна схема реалізації телеметрії з використанням OSD:



Рисунок 3.19 – Структурна схема застосування телеметрії з OSD

Зовнішні дані в модуль OSD можуть надходити через послідовний інтерфейс безпосередньо від зовнішніх датчиків, або одночасно з різних джерел.

Мікроконтролер модуля (MCU) обробляє дані і завантажує їх в мікросхему - відеоміксер (V-MIX), який накладає графічну і цифробуквену інформацію на зображення. Структурну схему модуля OSD приведено на рис.3.20.

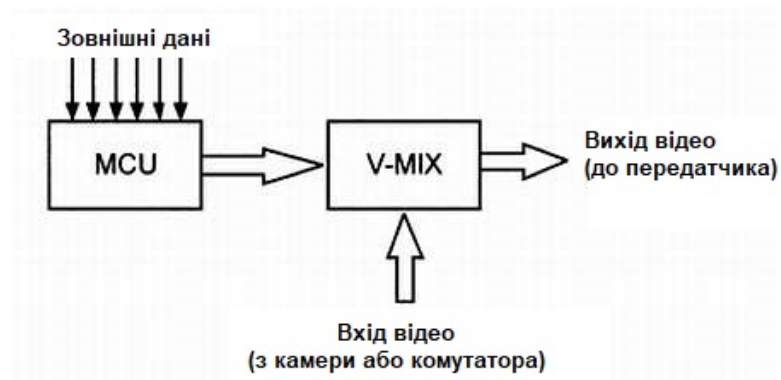


Рисунок 3.20 – Структурна схема модуля OSD

Схема з'єднань елементів БПЛА побудована на основі вибору конструкторських елементів приведена на рис.3.21.

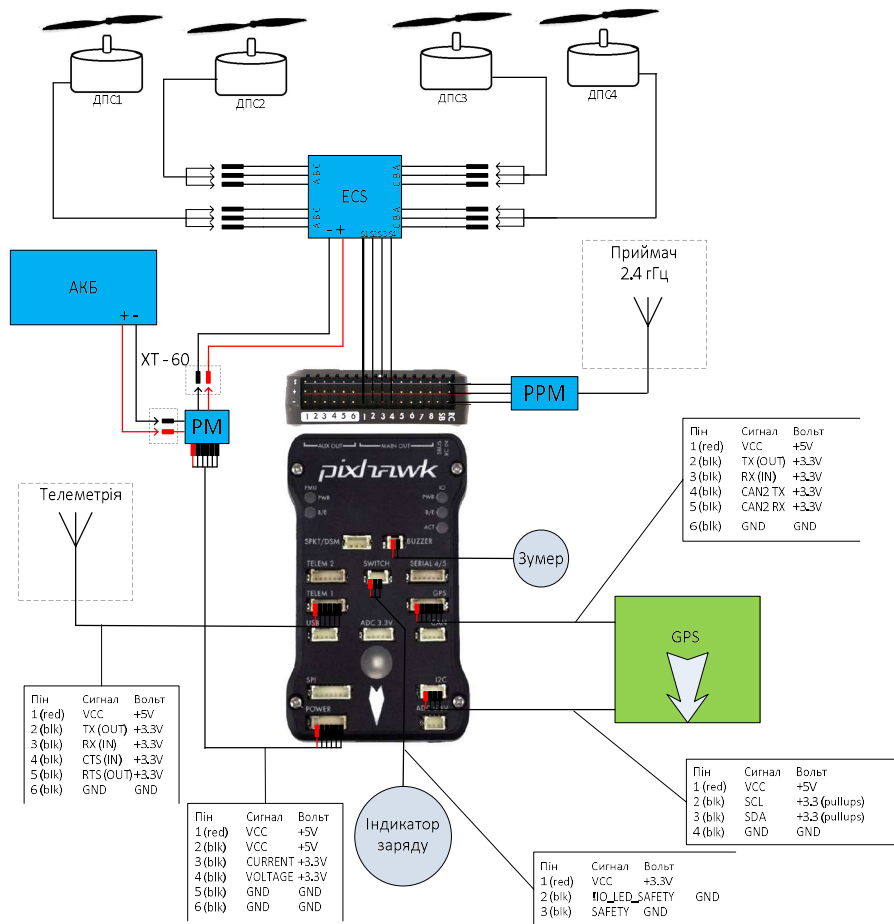


Рисунок 3.21 – Схема з'єднань елементів БПЛА

Проведен синтез конструкторських частин БПЛА. Елементи БПЛА пропонується придбати як окремі модулі. Обґрунтовано вибір механічної частини БПЛА – рами та двигунів, в якості керуючого пристрою вибрано льотний контролер з високопродуктивним мікропроцесором, підібрано систему електроживлення та обладнання для моніторингу.

4 РЕАЛІЗАЦІЯ СИСТЕМИ

4.1 O пис програми

Назва проекту «UAV coordinates». Сучасне програмне забезпечення пред'являє певні вимоги до обчислювальної потужності центрального процесора і обсягу оперативної пам'яті. Для більш ефективної роботи обрано середовище Windows.

Програмне забезпечення складається з безлічі підпрограм, об'єднаних в єдину обчислювальну систему. Для створення ізольованої користувальницької середовища, спрямованої на вирішення завдання для обробки знімків, отриманих з безпілотного літального апарату була підготовлена система на базі Windows, що містить все необхідне програмне забезпечення і готова для використання.

MATLAB (від англ. «Matrix Laboratory») - пакет прикладних програм для вирішення задач технічних обчислень. Пакет використовують понад мільйон інженерних і наукових працівників, працює на більшості сучасних операційних систем, включаючи Linux, Mac OS, Solaris і Windows.

Мова MATLAB є високорівневим інтерпретується мовою програмування, що включає засновані на матрицях структури даних, широкий спектр функцій, інтегроване середовище розробки, об'єктно-орієнтовані можливості і інтерфейси до програм, написаних на інших мовах програмування.

Програми, написані на MATLAB, бувають двох типів - функції і скрипти. Функції мають вхідні і вихідні аргументи, а також власний робочий простір для зберігання проміжних результатів обчислень і змінних. Скрипти ж використовують загальний робочий простір. Як скрипти, так і функції зберігаються у вигляді текстових файлів і компілюються в машинний код динамічно. Існує також можливість зберігати так звані pre-parsed програми - функції і скрипти, оброблені в вид, зручний для машинного виконання. У загальному випадку такі програми виконуються швидше звичайних, особливо якщо функція містить команди побудови графіків.

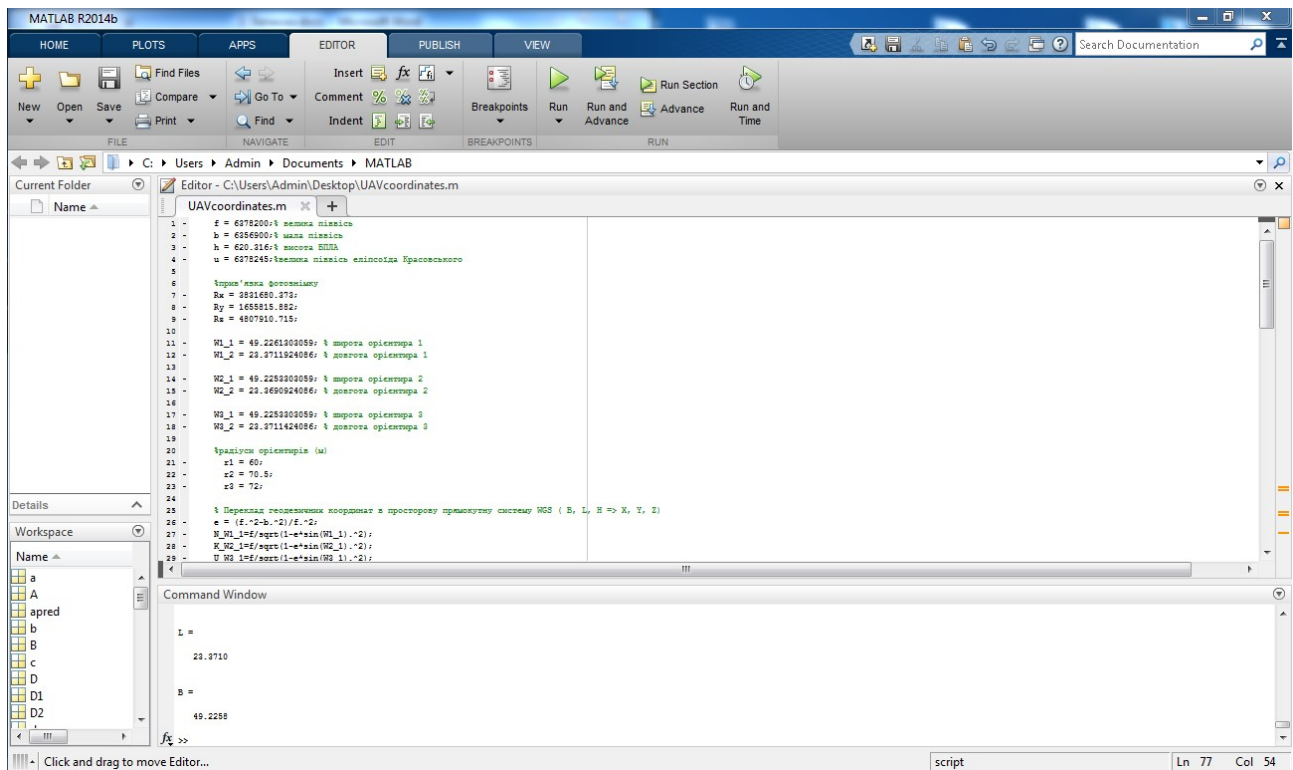


Рисунок 4.1 – Головне вікно програми

Фотознімок зроблений з бортової камери, на якому будуть розглядатися орієнтири для знаходження координат БПЛА.



Рисунок 4.2 – Фотознімок з бортової камери

Вхідні дані для програми:

f , b , h , u – велика піввісь, мала піввісь, висота БПЛА, велика піввісь еліпсоїда Красовського відповідно. Прив'язка фотознімка задаються у просторовій системі координат WGS (X , Y). Широта і довгота орієнтирів задаються у геодезичних координатах (B , H).

```
close all
clear all

% Читання вихідного зображення і виведення його на екран
I=imread('map.PNG');
figure,imshow(I),title('Фотозображення місцевості');
figure; imshow(I),title('Відображення центрів орієнтирів');
hold on;
figure1 = plot(192,170,'r*'); %центр орієнтира 1
figure2 = plot(529,123,'r*'); %центр орієнтира 2
figure3 = plot(452,284,'r*'); %центр орієнтира 3
figure4 = plot(395,201,'b*'); %правдоподібне місцезнаходження БПЛА

%-----

f = 6378200; %велика піввісь
b = 6356900; %мала піввісь
h = 620.316; %висота БПЛА (м)
u = 6378245; %велика піввісь еліпсоїда Красовського

%прив'язка фотознімку
Rx = 3831680.373;
Ry = 1655815.882;
Rz = 4807910.715;

W1_1 = 49.2261303059; %широта орієнтира 1
W1_2 = 23.3711924086; %довгота орієнтира 1

W2_1 = 49.2253303059; %широта орієнтира 2
W2_2 = 23.3690924086; %довгота орієнтира 2

W3_1 = 49.2253303059; %широта орієнтира 3
W3_2 = 23.3711424086; %довгота орієнтира 3

Переводимо геодезичні координати ( $B,L,H$ ) в просторову прямокутну систему WGS ( $X,Y,Z$ ).

% Переклад геодезичних координат в просторову прямокутну систему
WGS ( B, L, H => X, Y, Z)
e = (f.^2-b.^2)/f.^2;
N_W1_1=f/sqrt(1-e*sin(W1_1).^2);
K_W2_1=f/sqrt(1-e*sin(W2_1).^2);
U_W3_1=f/sqrt(1-e*sin(W3_1).^2);

A(1,1) = (N_W1_1+h)*cos(W1_1)*cos(W1_2);
```

```

A(1,2) = (N_W1_1+h)*cos(W1_1)*sin(W1_2);
A(1,3) = (((b.^2)/(f.^2))*N_W1_1+h)*sin(W1_1);

A(2,1) = (K_W2_1+h)*cos(W2_1)*cos(W2_2);
A(2,2) = (K_W2_1+h)*cos(W2_1)*sin(W2_2);
A(2,3) = (((b.^2)/(f.^2))*K_W2_1+h)*sin(W2_1);

A(3,1) = (U_W3_1+h)*cos(W3_1)*cos(W3_2);
A(3,2) = (U_W3_1+h)*cos(W3_1)*sin(W3_2);
A(3,3) = (((b.^2)/(f.^2))*U_W3_1+h)*sin(W3_1);

```

Початок відліку для рішення розрахунку координат БПЛА і задання точності обчислень.

```

% Початок відліку для рішення задачі розрахунку координат позиції
БПЛА
X0 = 0; Y0 = 0; Z0 = 0; %центр землі
%точність обчислень
eps1 = 3;
dX = eps1 * 3;
dY = eps1 * 3;
dZ = eps1 * 3;
apred = [X0 Y0 Z0];
da = [eps1*3 eps1*3 eps1*3];
% Abs (apred)
while da > [eps1 eps1 eps1];
a = [X0 Y0 Z0];
%a1 = [X1 Y1 Z1];
%a2 = [X2 Y2 Z2];
%a3 = [X3 Y3 Z3];
R = [a; a; a];

R1 = sqrt ((A(1,1)).^2 + (A(1,2)).^2 + (A(1,3)).^2);
R2 = sqrt ((A(2,1)).^2 + (A(2,2)).^2 + (A(2,3)).^2);
R3 = sqrt ((A(3,1)).^2 + (A(3,2)).^2 + (A(3,3)).^2);

D1=A(1,1)/R1
D2=A(2,1)/R2
D3=A(3,1)/R3
for j = 1 : 3
    D1 = A1 / R1;
    D2 = A2 / R2;
    D3 = A3 / R3;
end;
end;

```

Після знаходження радіусів для орієнтирів, потрібно знайти точку перетину трьох кіл (X і Y).

$$(X_1 - X)^2 + (Y_1 - Y)^2 = R_1^2,$$

$$(X_2 - X)^2 + (Y_2 - Y)^2 = R_2^2,$$

$$(X_3 - X)^2 + (Y_3 - Y)^2 = R_3^2;$$

де $X_1, Y_1, X_2, Y_2, X_3, Y_3$ – координати орієнтирів, R_1, R_2, R_3 – відстань до орієнтирів.

Після застосування формул скороченого множення многочленів, отримаємо систему виду:

$$X_1^2 - 2X_1X + X^2 + Y_1^2 - 2Y_1Y + Y^2 = R_1^2, \quad (1)$$

$$X_2^2 - 2X_2X + X^2 + Y_2^2 - 2Y_2Y + Y^2 = R_2^2, \quad (2)$$

$$X_3^2 - 2X_3X + X^2 + Y_3^2 - 2Y_3Y + Y^2 = R_3^2, \quad (3)$$

Відніmemo від першого рівняння друге і від другого рівняння третього, отримаємо:

$$X_1^2 - X_2^2 - 2X_1X + 2X_2X + Y_1^2 - Y_2^2 - 2Y_1Y + 2Y_2Y = R_1^2 - R_2^2,$$

$$X_2^2 - X_3^2 - 2X_2X + 2X_3X + Y_2^2 - Y_3^2 - 2Y_2Y + 2Y_3Y = R_2^2 - R_3^2;$$

Далі висловлюємо X і Y:

$$2Y(Y_2 - Y_1) = R_1^2 - R_2^2 - X_1^2 + X_2^2 + 2X_1X - 2X_2X - Y_1^2 + Y_2^2,$$

$$2Y(Y_3 - Y_2) = R_2^2 - R_3^2 - X_2^2 + X_3^2 + 2X_2X - 2X_3X - Y_2^2 + Y_3^2;$$

$$2X(X_2 - X_1) = R_1^2 - R_2^2 - Y_1^2 + Y_2^2 + 2Y_1Y - 2Y_2Y - X_1^2 + X_2^2,$$

$$2X(X_3 - X_2) = R_2^2 - R_3^2 - Y_2^2 + Y_3^2 + 2Y_2Y - 2Y_3Y - X_2^2 + X_3^2;$$

$$Y = (R_1^2 - R_2^2 - X_1^2 + X_2^2 + 2X_1X - 2X_2X - Y_1^2 + Y_2^2) / (2(Y_2 - Y_1)),$$

$$Y = (R_2^2 - R_3^2 - X_2^2 + X_3^2 + 2X_2X - 2X_3X - Y_2^2 + Y_3^2) / (2(Y_3 - Y_2));$$

$$X = (R_1^2 - R_2^2 - Y_1^2 + Y_2^2 + 2Y_1Y - 2Y_2Y - X_1^2 + X_2^2) / (2(X_2 - X_1)),$$

$$X = (R_2^2 - R_3^2 - Y_2^2 + Y_3^2 + 2Y_2Y - 2Y_3Y - X_2^2 + X_3^2) / (2(X_3 - X_2));$$

Прирівняємо відповідні координати об'єкта, отримаємо систему виду:

$$\begin{aligned} & (\mathbf{R}_1^2 - \mathbf{R}_2^2 - \mathbf{X}_1^2 + \mathbf{X}_2^2 + 2\mathbf{X}_1\mathbf{X} - 2\mathbf{X}_2\mathbf{X} - \mathbf{Y}_1^2 + \mathbf{Y}_2^2) / (2(\mathbf{Y}_2 - \mathbf{Y}_1)) = \\ & = (\mathbf{R}_2^2 - \mathbf{R}_3^2 - \mathbf{X}_2^2 + \mathbf{X}_3^2 + 2\mathbf{X}_2\mathbf{X} - 2\mathbf{X}_3\mathbf{X} - \mathbf{Y}_2^2 + \mathbf{Y}_3^2) / (2(\mathbf{Y}_3 - \mathbf{Y}_2)), \end{aligned}$$

$$\begin{aligned} & (\mathbf{R}_1^2 - \mathbf{R}_2^2 - \mathbf{Y}_1^2 + \mathbf{Y}_2^2 + 2\mathbf{Y}_1\mathbf{Y} - 2\mathbf{Y}_2\mathbf{Y} - \mathbf{X}_1^2 + \mathbf{X}_2^2) / (2(\mathbf{X}_2 - \mathbf{X}_1)) = \\ & = (\mathbf{R}_2^2 - \mathbf{R}_3^2 - \mathbf{Y}_2^2 + \mathbf{Y}_3^2 + 2\mathbf{Y}_2\mathbf{Y} - 2\mathbf{Y}_3\mathbf{Y} - \mathbf{X}_2^2 + \mathbf{X}_3^2) / (2(\mathbf{X}_3 - \mathbf{X}_2)); \end{aligned}$$

Знаходимо координати об'єкта:

$$\begin{aligned} & 2(\mathbf{Y}_3 - \mathbf{Y}_2) (\mathbf{R}_1^2 - \mathbf{R}_2^2 - \mathbf{X}_1^2 + \mathbf{X}_2^2 + 2\mathbf{X}_1\mathbf{X} - 2\mathbf{X}_2\mathbf{X} - \mathbf{Y}_1^2 + \mathbf{Y}_2^2) = \\ & = 2(\mathbf{Y}_2 - \mathbf{Y}_1) (\mathbf{R}_2^2 - \mathbf{R}_3^2 - \mathbf{X}_2^2 + \mathbf{X}_3^2 + 2\mathbf{X}_2\mathbf{X} - 2\mathbf{X}_3\mathbf{X} - \mathbf{Y}_2^2 + \mathbf{Y}_3^2), \\ & 2(\mathbf{X}_3 - \mathbf{X}_2) (\mathbf{R}_1^2 - \mathbf{R}_2^2 - \mathbf{Y}_1^2 + \mathbf{Y}_2^2 + 2\mathbf{Y}_1\mathbf{Y} - 2\mathbf{Y}_2\mathbf{Y} - \mathbf{X}_1^2 + \mathbf{X}_2^2) = \\ & = 2(\mathbf{X}_2 - \mathbf{X}_1) (\mathbf{R}_2^2 - \mathbf{R}_3^2 - \mathbf{Y}_2^2 + \mathbf{Y}_3^2 + 2\mathbf{Y}_2\mathbf{Y} - 2\mathbf{Y}_3\mathbf{Y} - \mathbf{X}_2^2 + \mathbf{X}_3^2); \end{aligned}$$

$$\begin{aligned} & 2(\mathbf{Y}_3 - \mathbf{Y}_2)(2\mathbf{X}_1\mathbf{X} - 2\mathbf{X}_2\mathbf{X}) - 2(\mathbf{Y}_2 - \mathbf{Y}_1)(2\mathbf{X}_2\mathbf{X} - 2\mathbf{X}_3\mathbf{X}) = \\ & = 2(\mathbf{Y}_2 - \mathbf{Y}_1) (\mathbf{R}_2^2 - \mathbf{R}_3^2 - \mathbf{X}_2^2 + \mathbf{X}_3^2 - \mathbf{Y}_2^2 + \mathbf{Y}_3^2) - 2(\mathbf{Y}_3 - \mathbf{Y}_2) (\mathbf{R}_1^2 - \mathbf{R}_2^2 - \mathbf{X}_1^2 + \mathbf{X}_2^2 - \mathbf{Y}_1^2 + \\ & \quad \mathbf{Y}_2^2), \end{aligned}$$

$$\begin{aligned} & 2(\mathbf{X}_3 - \mathbf{X}_2)(2\mathbf{Y}_1\mathbf{Y} - 2\mathbf{Y}_2\mathbf{Y}) - 2(\mathbf{X}_2 - \mathbf{X}_1)(2\mathbf{Y}_2\mathbf{Y} - 2\mathbf{Y}_3\mathbf{Y}) = \\ & = 2(\mathbf{X}_2 - \mathbf{X}_1) (\mathbf{R}_2^2 - \mathbf{R}_3^2 - \mathbf{Y}_2^2 + \mathbf{Y}_3^2 - \mathbf{X}_2^2 + \mathbf{X}_3^2) - 2(\mathbf{X}_3 - \mathbf{X}_2) (\mathbf{R}_1^2 - \mathbf{R}_2^2 - \mathbf{Y}_1^2 + \mathbf{Y}_2^2 - \mathbf{X}_1^2 + \\ & \quad \mathbf{X}_2^2); \end{aligned}$$

$$\begin{aligned} & 4\mathbf{X}(\mathbf{Y}_3 - \mathbf{Y}_2)(\mathbf{X}_1 - \mathbf{X}_2) - 4\mathbf{X}(\mathbf{Y}_2 - \mathbf{Y}_1)(\mathbf{X}_2 - \mathbf{X}_3) = \\ & = 2(\mathbf{Y}_2 - \mathbf{Y}_1) (\mathbf{R}_2^2 - \mathbf{R}_3^2 - \mathbf{X}_2^2 + \mathbf{X}_3^2 - \mathbf{Y}_2^2 + \mathbf{Y}_3^2) - 2(\mathbf{Y}_3 - \mathbf{Y}_2) (\mathbf{R}_1^2 - \mathbf{R}_2^2 - \mathbf{X}_1^2 + \mathbf{X}_2^2 - \mathbf{Y}_1^2 + \\ & \quad \mathbf{Y}_2^2), \end{aligned}$$

$$\begin{aligned} & 4\mathbf{Y}(\mathbf{X}_3 - \mathbf{X}_2)(\mathbf{Y}_1 - \mathbf{Y}_2) - 4\mathbf{Y}(\mathbf{X}_2 - \mathbf{X}_1)(\mathbf{Y}_2 - \mathbf{Y}_3) = \\ & = 2(\mathbf{X}_2 - \mathbf{X}_1) (\mathbf{R}_2^2 - \mathbf{R}_3^2 - \mathbf{Y}_2^2 + \mathbf{Y}_3^2 - \mathbf{X}_2^2 + \mathbf{X}_3^2) - 2(\mathbf{X}_3 - \mathbf{X}_2) (\mathbf{R}_1^2 - \mathbf{R}_2^2 - \mathbf{Y}_1^2 + \mathbf{Y}_2^2 - \mathbf{X}_1^2 + \\ & \quad \mathbf{X}_2^2); \end{aligned}$$

$$X = 2((Y_2 - Y_1) (R_2^2 - R_3^2 - X_2^2 + X_3^2 - Y_2^2 + Y_3^2) - (Y_3 - Y_2) (R_1^2 - R_2^2 - X_1^2 + X_2^2 - Y_1^2 + Y_2^2)) / 4((Y_3 - Y_2)(X_1 - X_2) - 4X(Y_2 - Y_1)(X_2 - X_3)),$$

$$Y = 2((X_2 - X_1) (R_2^2 - R_3^2 - Y_2^2 + Y_3^2 - X_2^2 + X_3^2) - (X_3 - X_2) (R_1^2 - R_2^2 - Y_1^2 + Y_2^2 - X_1^2 + X_2^2)) / 4((X_3 - X_2)(Y_1 - Y_2) - 4Y(X_2 - X_1)(Y_2 - Y_3));$$

$$X = ((Y_2 - Y_1) (R_2^2 - R_3^2 - X_2^2 + X_3^2 - Y_2^2 + Y_3^2) - (Y_3 - Y_2) (R_1^2 - R_2^2 - X_1^2 + X_2^2 - Y_1^2 + Y_2^2)) / 2((Y_3 - Y_2)(X_1 - X_2) - 4X(Y_2 - Y_1)(X_2 - X_3)),$$

$$Y = ((X_2 - X_1) (R_2^2 - R_3^2 - Y_2^2 + Y_3^2 - X_2^2 + X_3^2) - (X_3 - X_2) (R_1^2 - R_2^2 - Y_1^2 + Y_2^2 - X_1^2 + X_2^2)) / 2((X_3 - X_2)(Y_1 - Y_2) - 4Y(X_2 - X_1)(Y_2 - Y_3)).$$

Для точки перетину трьох кіл (X, Y) наступний код:

```
% кругова латерація
r1 = sqrt((A(1,1)-X0).^2 + (A(1,2)-Y0).^2 + (A(1,3)-Z0).^2)
r2 = sqrt((A(2,1)-X0).^2 + (A(2,2)-Y0).^2 + (A(2,3)-Z0).^2)
r3 = sqrt((A(3,1)-X0).^2 + (A(3,2)-Y0).^2 + (A(3,3)-Z0).^2)

X = ((A(2,2)-A(1,2)) * (r2.^2-r3.^2-A(2,1).^2+A(3,1).^2-
A(2,2).^2+A(3,2).^2) - (A(3,2)-A(2,2)) * (r1.^2-r2.^2-
A(1,1).^2+A(2,1).^2-A(1,2).^2+A(2,2).^2)) / (2 * ((A(3,1)-
A(2,1)) * (A(1,2)-A(2,2)) - (A(2,1)-A(1,1)) * (A(2,2)-A(3,2))))

Y = ((A(2,1)-A(1,1)) * (r2.^2-r3.^2-A(2,2).^2+A(3,2).^2-
A(2,1).^2+A(3,1).^2) - (A(3,1)-A(2,1)) * (r1.^2-r2.^2-
A(1,2).^2+A(2,2).^2-A(1,2).^2+A(2,1).^2)) / (2 * ((A(3,1)-
A(2,1)) * (A(1,2)-A(2,2)) - (A(2,1)-A(1,1)) * (A(2,2)-A(3,2))))
```

Оскільки точка перетину в просторовій прямокутній системі WGS, переведемо її в геодезичні координати:

```
% Переклад просторової прямокутної системи WGS в геодезичні
координати (X, Y, Z => B, L, H)
% допоміжна величина D
D = sqrt(X.^2+Y.^2);
%аналізуємо значення D наступним чином (D>0)
La=asin(Y/D);
H = D - u %висота
L = 2*pi - La %довгота
%для широти (B) знайдемо додаткові величини r, c, s1
r = sqrt(Rx.^2 + Ry.^2 + Rz.^2);
c = asin(Rz/r);
s1 = 0;
```

```
b = c + s1;
B = b
```

4.2 Керівництво користувача

Для запуску MATLAB потрібно відкрити його на «робочому столі». Створити новий проект, натиснувши на іконку «New Script» зверху в лівому куті. Після чого відкривається «Editor – Untitled» для нового проекту, у якому вводиться код програми.

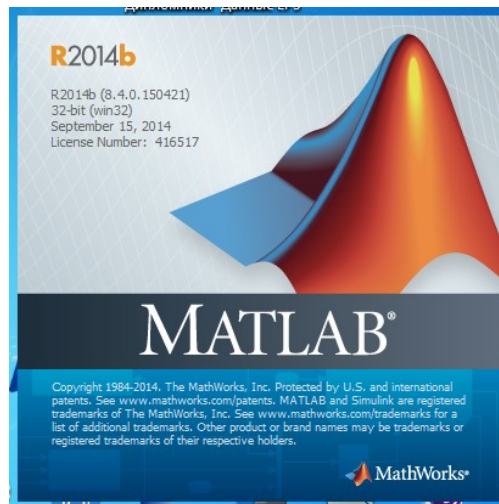


Рисунок 4.3 – Запуск програми Matlab

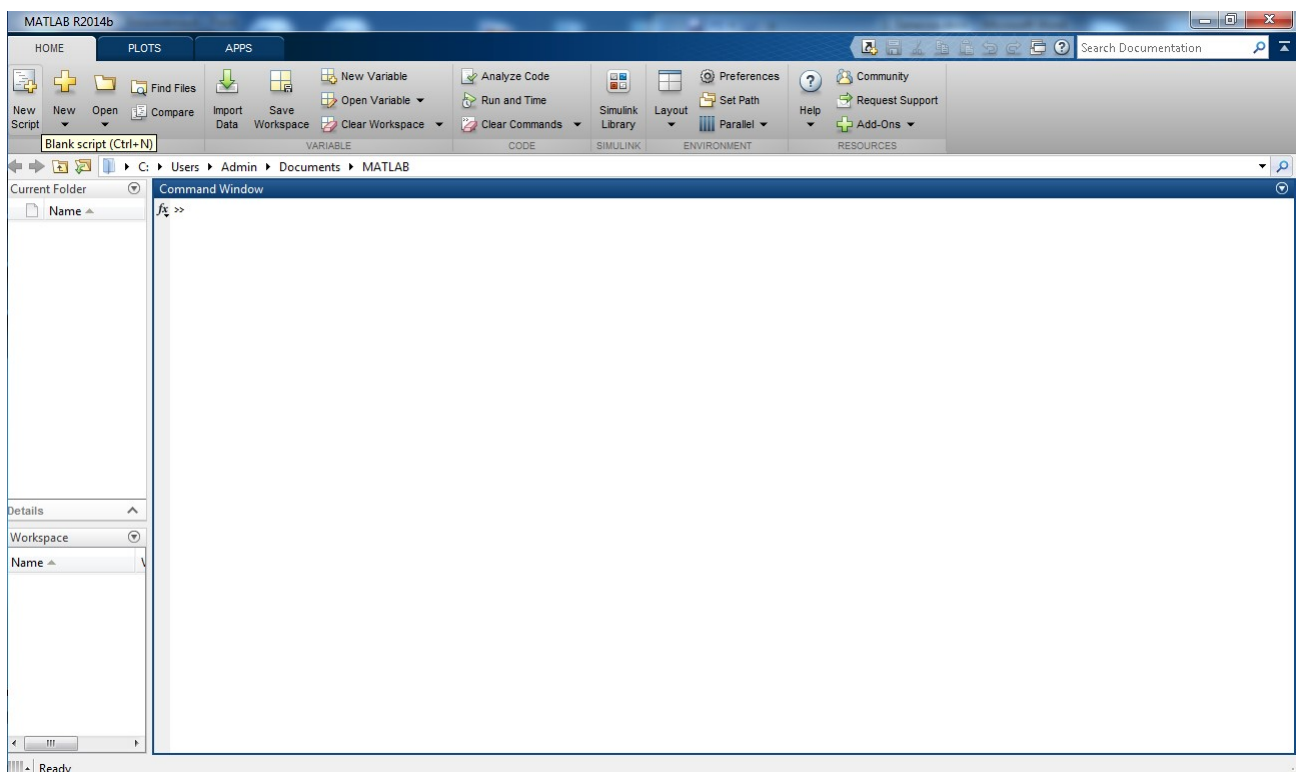


Рисунок 4.4 – Вікно програми Matlab

Для запуску коду треба вибрати іконку «Run» або натиснути клавішу F5.

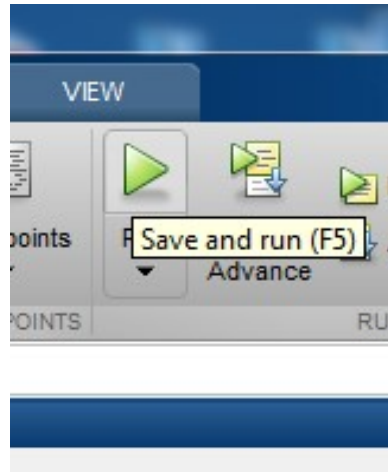


Рисунок 4.6 – Іконка «Run»

Вихідні дані можна побачити в «Command Window».

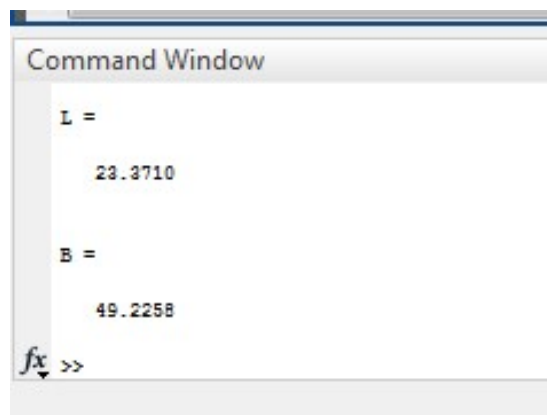


Рисунок 4.7 – Командне вікно

4.3 Технологія тестування програмного забезпечення

Етапом тестування була проведена перевірка функціональності та пошук дефектів в компонентах програми, котрі можуть бути протестовані окремо.

5 ЕКСПЕРЕМЕНТАЛЬНО-ПРАКТИЧНА ЧАСТИНА

5.1

Результати експериментального тестування

Для тестування по пункто буде наведена послідовність операцій.

1) Відкрити пакет прикладних програм Matlab (R2014b). Створити новий проект «New Script». Результат приведений на рисунку 5.1.

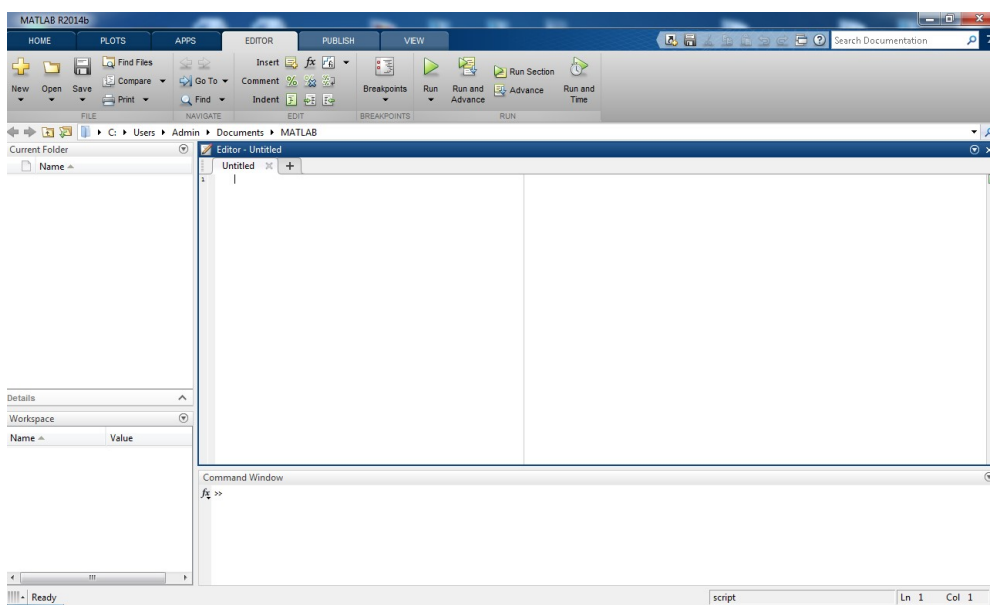


Рисунок 5.1 – Вікно новоствореного проекту

2) У вікно Editor – Untitled написати код програми визначення координат БПЛА за візуальними орієнтирами (див. ДОДАТОК Б).

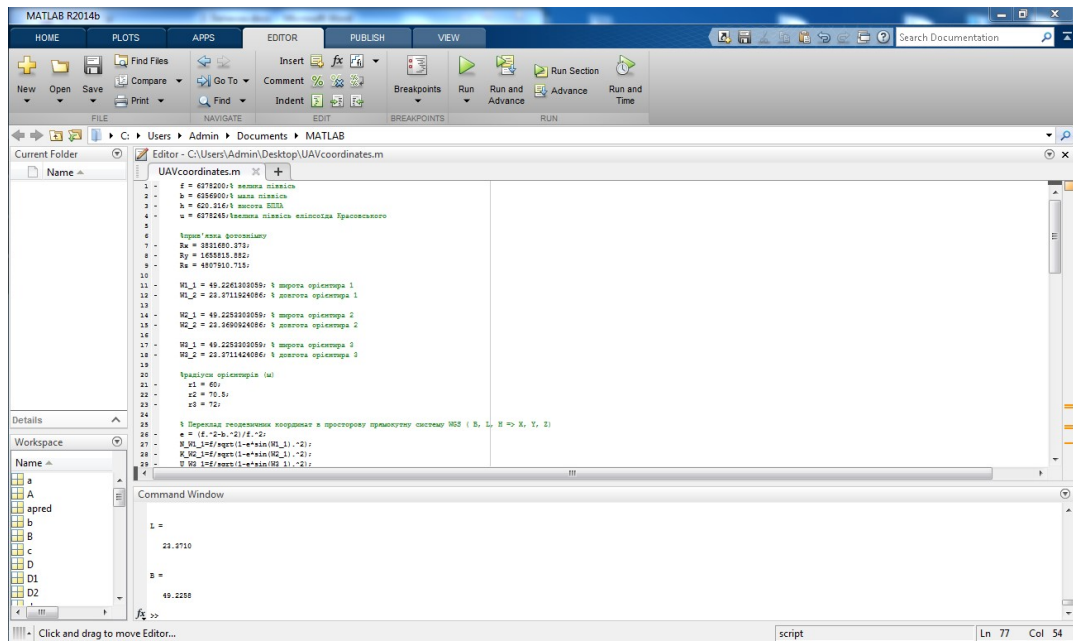


Рисунок 5.2 – Код программы UAVcoordinates

3) Натиснути клавішу F5. У Command Window будуть виведені геодезичні координати (B, L, H).



Рисунок 5.3 – Результат коду програми

4) У вікні «Figure 2» буде зображено центри орієнтирів (зірка червоного кольору), правдоподібне місцезнаходження БПЛА (зірка синього кольору) і реальне місцезнаходження БПЛА (зірка зеленого кольору).



Рисунок 5.4 – Позначення орієнтирів на карті

Були перевірені компоненти системи. Обрані програмні компоненти пройшли успішне тестування, показання працездатності задовольняють вимогам до побудови системи.

6. ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ

6.1 Опис продукту

Основною метою розробки економічної частини полягає у розрахунку собівартості, ціни, трудомісткості робіт при створенні даного програмного продукту.

Розроблюваний продукт – автоматизована система визначення координат БПЛА за візуальними орієнтирами. Для роботи системи потребується комп'ютер операційною системою Windows, доступ до мережі Інтернет та безпілотний літальний апарат.

6.2 Сегментування ринку

Сегментування ринку - це процес поділу споживачів на групи з урахуванням різних принципів і факторів сегментації. У процесі сегментації виділяють сегменти ринку. Основною характеристикою сегмента є ємність - кількість виробів, які можуть бути продані в рік.

Повна потреба сегменту розраховується з урахуванням конкретних особливостей товару і сегментів. Для багатьох видів товарів індивідуального і промислового призначення $S_{1пол.і}$ можна розрахувати по формулі 6.1.

$$S_{1пол.і} = N_1 \cdot Q_1 \cdot m_1; \quad (6.1)$$

де N_1 – кількість підприємств споживачів виробу в і-му сегменті;

Q_1 – середня річна програма виробів в і-му сегменті, для яких буде постачатись розглянутий товар;

m_1 – кількість комплектуючих виробів, йдучих в один виріб (1 шт.)

Таблиця 6.1 – Сегментування і розрахунок повної ємності ринку

Виробник	Завод-виробник			
	N_1	Q_1 , шт	m_1 , шт	S_1 , шт/рік
«Leica Geosystems AG»	1	20	2	40
Spaitech	1	50	2	100

Атлон Авіа	1	45	1	45
	3	105	5	185

Як видно з таблиці 7.1 повна ємність ринку складає 185 шт./рік.

6.3 Аналіз конкурентоспроможності

Конкурентоспроможність товару – це ступінь його відповідності обраному ринку з комерційних, технічних і економічних показників, забезпечуючи можливість збуту товару на цьому ринку. Це ті характеристики, які вигідно вирізняють даний товар від товарів-конкурентів.

Проведемо аналіз конкурентоспроможності нового виробу, за допомогою узагальненого показника якості, який враховує всю систему основних одиничних показників, і характерний для даного пристрою.

У процесі аналізу ринку збуту були виявлені основні конкуренти на українському ринку. Головною відмінністю їх виробів від розроблюваної системи є те, що їх аналогові системи, а не цифрові. Так само відмінність у технічних характеристиках.

При порівнянні варіантів використовується наступний порядок розрахунку.

1. Виділяються еталонні значення окремих показників якості і формується гіпотетичний варіант. Гіпотетичний варіант - штучно створений варіант, в якому кожен показник має найкраще можливе значення.

2. Кожному показнику якості присвоюємо коефіцієнт вагомості (в сумі 1).

Розраховується відносне значення одиничних показників шляхом порівняння P_{ij} із $P_{i \text{ гин}}$ з урахуванням того, що для гіпотетичного варіанти відносні показники дорівнюють одиниці ($K_{i \text{ гин}} = 1$), для інших $K_{ij} \leq 1$.

$$K_{ij} = \frac{P_{ij}}{P_{i \text{ гин}}}; \quad (6.2)$$

де P_{ij} – абсолютні значення і-го показника для j-го варіанту;

$P_{i \text{ гин}}$ – абсолютні значення і-го показника для гіпотетичного варіанти

Отримані результати наведені в табл. 6.2.

Таблиця 6.2 - Аналіз конкурентоспроможності

Показники якості	Коефіцієнт вагомості	Абсолютні значення показників якості				Відносні значення показників якості							
		1	2	3	4	1		2		3		4	
		Розроблений пристрій	A1-СМ Furia	Leica Aibot	Гіпотетичний	K _{i1}	K _{i1} *b _i	K _{i2}	K _{i2} *b _i	K _{i3}	K _{i3} *b _i	K _{i4}	K _{i4} *b _i
Швидкість обробки даних	0.4	10	9	8	10	1.0	0.4	0.9	0.32	0.8	0.38	1.00	0.4
Здатність підлаштуватися під зміну параметрів	0.2	8	8	8	10	0.8	0.2	0.8	0.17	0.8	0.14	1.00	0.2
Швидкодія	0.2	9	9	7	10	0.9	0.17	0.9	0.17	0.7	0.16	1.00	0.2
Складність реалізації	0.1	9	9	5	10	0.9	0.09	0.9	0.09	0.5	0.05	1.00	0.1
Напрацювання на відмову	0.1	8	7	9	10	0.8	0.08	0.7	0.08	0.9	0.09	1.00	0.1
	1						0.94		0.83		0.82		1.00

Виходячи з результатів порівняльного аналізу, найбільше значення показника якості має розроблений продукт. Найближчим конкурентом є продукція PLX Devices. Рівень якості пропонованого виробу порівняно з цим конкурентом становить:

$$Y_{н-б} = Y_{1-2} = \frac{K_1^0}{K_2^0} = \frac{0.94}{0.83} = 1.13.$$

Порівняння виробів, присутніх на українському ринку, показало, що розроблений продукт забезпечує найбільшу швидкість обробки даних.

За отриманими рівнем якості можна зробити висновок, що становище на ринку буде стійким.

6.4 Розрахунок собівартості і ціни

Собівартість продукції складається з ряду найменувань витрат, а саме витрати на основні матеріали, комплектуючі вироби, пряма і додаткова заробітні плати, витрати на утримання та експлуатацію обладнання, утримання транспорту, а також ряд загальнодержавних податків і відрахувань.

В створенні даної системи приймають участь такі особи:

- 1) програміст;
- 2) керівник проекту;
- 3) інженер-тестувальник.

Виконавці робіт та їх заробітні плати наведені в табл. 6.3. Кількість робочих днів у місяці 20.

Таблиця 6.3 – Склад виконавців роботи

Посади	Посадкові оклади, грн.	
	Місячні	Денні
Програміст	12000	600
Керівник проекту	15000	750
Інженер-тестувальник	7000	350

Перелік робіт, які повинні бути виконані кожним працівником, а так їхня тривалість, наведені в табл. 6.4.

Розрахунок собівартості робіт починається з розрахунку основної заробітної плати (ЗП). Обчислення ЗП розробників лабораторного стенду формується з урахуванням трудовитрат, кількості виконавців і середньоденної ЗП. Для цього кількість днів, відпрацьованих окремими виконавцями за стадіями, множимо на їх денні оклади:

$$\text{ОЗП} = 600 \cdot 36 + 750 \cdot 31 + 350 \cdot 36 = 57450 \text{ грн.}$$

Додаткова заробітна плата складе:

$$\text{ДЗП} = \frac{20\% \cdot 57450}{100\%} = 11490 \text{ грн.}$$

Таблиця 6.4 – Розрахунок трудомісткості робіт

Вид робіт	Тривалість, дні	Трудомісткість, чол/дні	Виконавці		
			Програміст	Керівник проекту	Інженер-тестувальник
Постановка задачі	2	2	-	+	-
Розробка графіку роботи	1	1	-	+	-
Розробка ТЗ	2	2	-	+	-
Погодження та затвердження ТЗ	3	10	+	+	+
Аналіз існуючих засобів рішення	3	10	+	+	+
Розробка алгоритму	4	6	+	-	-
Реалізація алгоритму	12	12	+	-	-
Розробка структурної схеми	4	8	-	+	+
Створення робочого макету	8	16	-	+	+
Тестування	5	10	+	-	+
Приймально-здавальні роботи	7	16	+	+	+
Випуск робочої документації	8	16	+	-	+
Випробування і здача продукту	7	21	+	+	+
Всього	66	130	36	31	36

Розрахуємо вартість закупних виробів необхідних для виготовлення системи.

Таблиця 6.5 – Перелік закупних виробів

Матеріал	Кількість, шт.	Ціна, грн.	Вартість, грн.	Призначення
Рама Tarot FY650 "IRON MAN" TL65B01	1	4200	4200	Створення робочого макету
Акумулятор Dinogy Li-Pol 11000mAh 4S 25C	1	3850	3850	Створення робочого макету
Регулятор ходу qattro 25Ax4	4	1420	5680	Створення робочого макету
Польотний контролер PixHawk	1	3800	3800	Створення робочого макету
Камера Xiaomi Yi Sport Black	1	1600	1600	Створення робочого макету
GPS Ublox Neo m8n 8N	1	660	660	Створення робочого макету
Друк документації	100	1.20	120	Випуск робочої документації
Доступ до інтернету	2	100	200	Аналіз існуючих засобів рішення
Всього			20110	

Таблиця 6.4 – Перелік основних засобів

Обладнання	Кількість, шт.	Вартість, грн.
Комп'ютер	3	21000
Стіл	3	2400
Стілець	3	2000
Всього		25400

Норма амортизації становить 25% від основних коштів, для виготовлення 1 виробу, що розраховується по формулі 6.3.

$$A_m = \frac{OC \cdot 0.25 \cdot D_p}{D_r} \quad (6.3)$$

де **OC** – вартість обладнання;

D_p – тривалість робочих днів;

D_r – кількість робочих днів у році.

Для виконання робіт, пов'язаних з виробництвом системи, необхідно обладнання вартістю 25400 грн. Так, як трудомісткість становить 79 днів, то амортизація обчислюється наступним чином:

$$A_m = \frac{25400 \cdot 0.25 \cdot 47}{240} = 1243,54$$

Розрахунок собівартості і договірної ціни наведено в табл. 6.5.

Таблиця 6.5 - Розрахунок собівартості і ціни виробу за статтями

№	Статті	Сума, грн	Примітки
1	Основна заробітна плата (ОЗП)	57450	$ОЗП = \sum N_i \cdot ЗП_{cp}$
2	Додаткова заробітна плата (ДЗП)	11490	20% від ОЗП
3	Єдиний соціальний внесок	15166,8	22% від (ОЗП+ДЗП)
4	Матеріали та куплені вироби	20110	Из табл. 6.5
5	Амортизація	1243,54	$A_m = \frac{OC \cdot 0.25 \cdot D_p}{D_r}$
6	Витрати на утримання обладнання	5745	10% від вартості основних засобів
7	Додаткові витрати	22980	40% від ОЗП
8	Виробнича вартість (С)	134185,34	п.1+п.2+п.3+...+п.7
9	Адміністративні витрати	28852,5	45% від ОЗП
10	Витрати на збут	3354,63	2.5% від п.8

11	Собівартість власних робіт	300577,81	п.1+п.2+п.3+...+п.10
12	Прибуток (П)	60115,56	20% від п.11
13	Ціна без ПДВ	360693,37	П+п.11
14	ПДВ	72138,67	20% від ціни без ПДВ
15	Ціна з ПДВ	432832,04	п.13+п.14

Таким чином, собівартість власних робіт з розроблення програмного продукту складає – 300 578 грн., а його ціна з врахуванням ПДВ – 432 832,04.

Кількість замовлених екземплярів має бути не менше ніж 185 шт.

Виробничу собівартість одного екземпляру програмного продукту (ВС) визначають за формулою 6.4.

$$BC_0 = \frac{BC}{KPP}; \quad (6.4)$$

де ВС – виробничі витрати;

КПП – кількість замовлених екземплярів.

$$BC_0 = \frac{134\,185,34}{185} = 725.$$

Повна собівартість одного екземпляру складається з суми виробничої собівартості (BCП), адміністративних витрат (AB) і витрат на збут (BЗ), які приходяться на один екземпляр виробу:

$$CП_0 = BC_0 + AB_0 + BЗ_0; \quad (6.5)$$

$$CП_0 = 725 + 155,9 + 18,1 = 899.$$

Адміністративні витрати AB_0 , які приходяться на один екземпляр виробу, визначається формулою 6.6.

$$AB_0 = \frac{AB}{KPP}; \quad (6.6)$$

$$AB_0 = \frac{28852,5}{185} = 155,96 \text{ грн.}$$

Витрати на збут VZ_0 , які приходяться на один екземпляр виробу, визначається формулою:

$$VZ_0 = \frac{B3}{KPP}; \quad (6.7)$$

$$VZ_0 = \frac{3354,63}{185} = 18,1 \text{ грн.}$$

Таким чином, $СП_0 = 725 + 155,9 + 18,1 = 899$.

Рентабельність продукції (норма прибутку) – це відношення загальної суми прибутку до витрат виробництва і реалізації продукції (відносна величина прибутку, що припадає на 1 грн. поточних витрат):

$$P_{\pi} = \frac{Ц-BC}{BC} \cdot 100\%; \quad (6.8)$$

де Ц – ціна одиниці продукції;

С – собівартість одиниці продукції.

$$P_{\pi} = \frac{432\ 832 - 300\ 577,81}{300\ 577,81} \cdot 100\% = 44\%.$$

Величина оптової ціни одного виробу ЦПП (без врахування ПДВ):

$$ЦПП = СП \cdot \left(1 + \frac{P_{\pi}}{100}\right); \quad (6.7)$$

де P_{π} – коефіцієнт рентабельності.

$$ЦПП = 899 \cdot \left(1 + \frac{44}{100}\right) = 1\ 294,6.$$

При впровадженні у виробництво продукту важливо знати чи стане цей виробничий процес рентабельним і чи буде він приносити бажаний прибуток. Для цього необхідно визначити точку беззбитковості (ТБ) і зобразити її графічно.

Для підтвердження стійкості проекту необхідно, щоб значення ТБ було менше значень нормальних обсягів виробництва. Чим далі від них значення ТБ (у відсотковому співвідношенні), тим стійкіше проект.

Дохід від реалізації виробів знаходимо множенням ціни одного ПП на кількість замовлених екземплярів ПП:

$$ДР = ЦПП \cdot КПП; \quad (6.8)$$

$$ДР = 1\,294,6 \cdot 185 = 99,457.$$

Аналітичний розмір критичної програми (РКП) розраховують діленням постійних витрат $РП_{ост}В$ на різницю між ціною одного виробу ЦПП і змінними витратами, які приходяться на один виріб ($З_мВ_0$), тобто:

$$РКП = \frac{РП_{ост}В}{ЦПП - З_мВ_0}. \quad (6.9)$$

Річні постійні витрати $РП_{ост}В$ складаються із суми наступних витрат:

$$РП_{ост}В = ВУО + А_м + ДВ + АВ + ВЗ; \quad (6.10)$$

де ВУО – витрати на утримання обладнання;

$А_м$ – амортизація;

ДВ – додаткові витрати;

АВ – адміністративні витрати;

ВЗ – витрати на збут.

$$РП_{ост}В = 5745 + 1243,54 + 22980 + 28852,5 + 3354,63 = 62\,176 \text{ грн.}$$

Річні змінні витрати $РЗ_мВ$ складаються із суми наступних витрат:

$$РЗ_мВ = ВМ + ФОП + ЄСВФОП; \quad (6.11)$$

де $ВМ$ – матеріали та куплені вироби;
 $ФОП$ – фонд оплати праці;
 $ЄСВФОП$ – єдиний соціальний внесок.

$$P_{3_{мВ}} = 20110 + (57450 + 11490) + 15166,8 = 104\,216,8 \text{ грн.}$$

Змінні витрати, які приходяться на один виріб, визначаємо діленням річних змінних витрат на річну програму випуску виробу:

$$Z_{мВ_0} = \frac{P_{3_{мВ}}}{КПП}. \quad (6.12)$$

$$Z_{мВ_0} = \frac{104\,216,8}{185} = 563,33 \text{ грн.}$$

$$РКП = \frac{104\,216,8}{1\,294,6 - 563,33} = \frac{104\,216,8}{731,27} = 142,51 \approx 143 \text{ шт.}$$

Річний дохід в точці беззбитковості:

$$ДР_{БЗ} = 1\,294,6 \cdot 143 = 185\,127,8 \text{ грн}$$

Графічне представлення точки беззбитковості представлено на рис. 6.1.

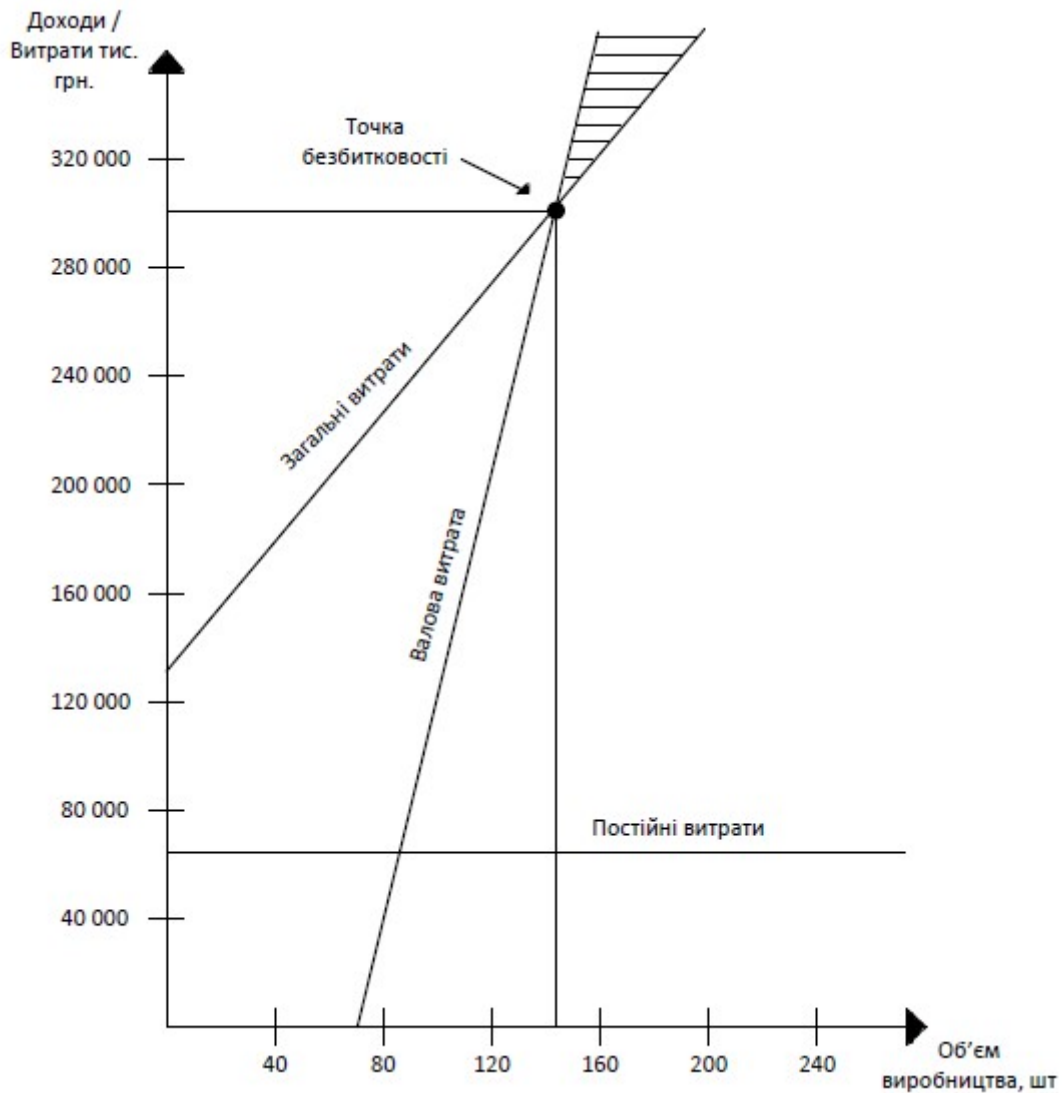


Рисунок 6.1 – Графік безбитковості

Графік безбитковості показує, що при об'ємі продаж 143 шт., дохід від реалізації продукції перевищує загальні витрати, таким чином, в зоні між ними знаходиться прибуток.

6.5 Висновки

В даному розділі був проведений розрахунок ціни і собівартості регулятора. Ціна продукту без ПДВ - 360 693 грн, ціна з ПДВ - 432 832 грн. Собівартість становить 300 578 грн. Розрахунок проведений з врахуванням всіх необхідних трудовитрат, ПДВ – 72 138,67 грн, відрахування в єдиний соціальний фонд 15 166,8 грн. Було встановлено, що рентабельність становить 44%, що є високим показником.

Також був побудований графік беззбитковості, на якому видно при якому обсязі продажів програмного продукту досягається точка беззбитковості, а саме 143 шт.

ВИСНОВКИ

В даній роботі спроектован та реалізован програмний продукт – автоматизована системи визначення координат БПЛА за візуальними орієнтирами.

Існуючі технічні засоби БПЛА і технології обробки геопросторових даних дозволяють вирішувати широкий спектр завдань. В першу чергу визначення власного місця розташування. При виконанні даної роботи було проведено аналіз існуючих рішень, відповідно до котрого були знайдені декілька рішень, які досить складні в освоєнні.

В ході вирішення зазначених у введенні завдань вдалося досягти мети даної дипломної роботи, а саме створення програмного засобу, що дозволяє визначати координати беспілотного літального апарату за візуальними орієнтирами.

Для досягнення мети в ході виконання дипломної роботи були вирішені наступні завдання: обрані всі необхідні елементи для розробки системи та отримані математичні моделі; розроблено функціональну схему для системи автоматичного управління; побудована структурна схема; побудовані характеристики САУ; визначені найкращі засоби реалізації БПЛА, які забезпечать задані показники якості; проведено експеримент. Спроектован інтерфейс і сформовано керівництво користувача.

В економічній частині проведений розрахунок собівартості і ціни продукту, виявлена висока конкурентоспроможність розроблюваного ПО.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кулік А.С. Конспект лекцій по курсу «Теорія автоматичного управління» / А.С. Кулік – Харков, ХАІ, 2014 – 2015.
2. Кулік А.С. Методы расчёта цифровых систем управления. – Харьков: Харьков авиац. ин-т, 1989. – 97 с.
3. Пославский С. И. Архитектура SLAM навигации программного продукта Octomap // Инноватика-2018. Сборник материалов XIV Международной школы-конференции студентов, аспирантов и молодых ученых. г. Томск, с. 117-120.
4. Пославский С.И., Окунский М.В. Построение карт для мобильного робототехнического объекта при помощи визуальной одометрии // Когнитивная робототехника: материалы II международной конференции / под ред. В.И. Сырямкина, А.В. Юрченко; Национальный исследовательский Томский государственный университет, с. 28.
5. Хорн, Б.К.П. Зрение роботов / Б.К.П. Хорн; пер. с англ. – М.: МИР, 1989. – 489 с.
6. Калинов И.А. Создание малого автономного беспилотного летательного аппарата на базе мультиротора с использованием алгоритма полупрямой визуальной одометрии // Интеллектуальные системы, управление и мехатроника-2016: матер. Всерос. научн.-технич. конф. Севастополь: Изд-во СевГУ, 2016. С. 150–154.
7. Ким Н.В. Использование методов анализа ситуации при решении целевых задач беспилотных летательных аппаратов.// Техническое зрение в системах управления 2012. Сб. тр. научно-техн. конф. М.: ИКИ РАН, 2012. С. 10 – 15.
8. Волков В.Ю. Методы дискретной фильтрации и задачи обработки изображений в радиотехнических системах наблюдения. – СПб.: СПбГУТ, 2013.
9. Форсайт, Д. Компьютерное зрение. Современный подход / Д. Форсайт, Ж. Понс; пер. с англ. – М.: Вильямс, 2004. – 928 с. – ISBN: 0-13-085198-1.
10. Шапиро, Л. Компьютерное зрение / Л. Шапиро, Д. Сток-ман; пер. с англ. – М.: Бином, Лаборатория знаний, 2015. – 763 с.

11. Hartley, R. I. In defense of the eight-point algorithm / R. I. Hartley // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1997. – Vol. 19, no. 6. – P. 580–593.

12. Nister, D. An efficient solution to the five-point relative pose problem / D. Nister // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2004. – Vol. 26, no. 6. – P. 756–770.

13. Котов, А.П. Технология оперативной реконструкции трёхмерных сцен по разноракурсным изображениям / А.П. Котов, В.А. Фурсов, Е.В. Гошин // Компьютерная оптика. – 2015. – Т. 39, № 4. – С. 600-605. – DOI: 10.18287/0134-2452-2015-39-4-600-605.

14. Попов Н.И., Емельянова О.В., Яцун С.Ф. Моделирование динамики полета квадрокоптера.

15. Гошин, Е.В. Решение задачи автокалибровки камеры с использованием метода согласованной идентификации / Е.В. Гошин, В.А. Фурсов // Компьютерная оптика. – 2012. – Т. 36, № 4. – С. 605-611

16. Мясников, В.В. Модельно-ориентированный дескриптор поля градиента как удобный аппарат распознавания и анализа цифровых изображений/ В.В. Мясников // Компьютерная оптика. – 2012. – Т. 36, № 4. – С. 596-604.

17. Девятериков, Е.А. Система технического стереозрения для измерения координат мобильного робота [Текст] / Е.А. Девятериков, Б.Б. Михайлов // Техническое зрение в системах управления 2012: Сб. трудов НТК. –М., 2012. – С. 72–78.

18. Девятериков, Е.А. Использование данных визуального одометра в системе управления мобильного робота [Текст] / Е.А. Девятериков, Б.Б. Михайлов // Экстремальная робототехника 2012: Сб. трудов НТК. –СПб., 2012 – С. 155–163.

19. Фетисов В. С., Артемьев А. Е., Муфаззалов Д. Ф. Автоматические сервисные станции для обслуживания электрических беспилотных летательных аппаратов: монография / Уфимск. гос. авиац. техн. ун-т. – М. : «Издательство «Инновационное машиностроение», 2017. – 253 с. ISBN 978-5-6040042-6-5.

20. Микроконтроллерные системы: структуры и практическое применение/В.Г. Джулгаков, В.В.Нарожный, К.И. Руденко, А.Н. Таран. – Учеб. пособие. – Харьков: Нац. аэрокосм. ун-т «Харьк авиац. ин-т», 2003. – 126 с.;

21. Залесский, Б. А. Параллельная версия детектора экстремальных особых точек изображений / Б. А. Залесский, Ф. С. Троцкий//Информатика. – 2018. – Т. 15, № 2. – С. 55–63.

22. Ориентация и навигация подвижных объектов: современные информационные технологии. / Под ред. Б.С. Алешина, К.К. Веремеенко, А.И. Черноморского. – М.: Физматлит, 2006, 424 стр.

23. Современные информационные технологии в задачах навигации и наведения беспилотных маневренных летательных аппаратов. / под ред. Красильщикова М.Н., Себрякова Г.Г. – М.: Физматлит, 2009, 556 стр.

24. Визильтер Ю.В., Желтов С.Ю., Бондаренко А.В., Осоков М.В., Моржин А.В. «Обработка и анализ изображений в задачах машинного зрения», 2010. 689-М.: Физматкнига.

25. Управление и наведение беспилотных маневренных летательных аппаратов на основе современных информационных технологий / Под ред. М.Н. Красильщикова и Г.Г. Себрякова. – М.: Физматлит, 2003, 280 стр.

26. Порядок оформления учебных и научно-исследовательских документов / В.Н. Павленко, А.С. Набатов, И.М. Тараненко. – Учеб. пособие. – Харьков: Нац. аэрокосм. ун-т «Харьк. авиац. ин-т», 2007. – 65 с.

27. Соколов, С.М. Богуславский, А.А. Интеллектуальные алгоритмы обработки изображений для решения задачи распознавания в реальном времени бортовыми системами // Тезисы докладов научнотехнической конференции-семинара. Техническое зрение в системах управления мобильными объектами. – М.: КДУ. – 2011. – С.86-88.

28. Система навигации группы БЛА на основе маркеров / К. С. Яковлев, В. В. Хитьков, М. И. Логинов, А. В. Петров // Робототехника и техническая кибернетика.— 2014.— № 3.— С. 44–48.

29. Абрамов Н.С., Хачумов М.В. Моделирование управления бортовой видеокамерой беспилотного летательного аппарата // Авиакосмическое приборостроение, №3, 2014, с. 9-16

30. Каримов А.Х. Цели и задачи, решаемые беспилотными авиационными комплексами нового поколения // Труды МАИ. Вып. № 47. 2011. С. 1–9; URL: <http://www.mai.ru/science/trudy/published.php?ID=26767>.

31. Спеціалізована БД "Винаходи (корисні моделі) в Україні" [Електронний ресурс]. – Режим доступу: <http://base.uipv.org/searchINV/>, вільний.