

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

Факультет систем управління літальних апаратів

Кафедра систем управління літальних апаратів

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломної роботи

магістра

(освітньо-кваліфікаційний рівень)

на тему Розробка програмного забезпечення для попередньої
обробки відеоданих в комплексі задач інтелектуальних транспортних систем

ХАІ.301.361.24О.272.00183014 ПЗ

Виконав: студент 2 курсу, групи 361

Галузь знань 27 «Транспорт»

Спеціальність

272 “Авіаційний транспорт”

Освітня програма

“Інтелектуальні транспортні системи”

Щербина Олексій Сергійович

(прізвище та ініціали студента)

Керівник Краснов Л. О.

(прізвище та ініціали)

Рецензент Жученко О. С.

(прізвище та ініціали)

Харків – 2024

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

Факультет __ систем управління літальних апаратів _____
(повне найменування)
Кафедра __ систем управління літальних апаратів _____
(повне найменування)
Рівень вищої освіти _____ другий (магістерський) _____
Галузь знань _____ 27 «Транспорт» _____
(код та найменування)
Спеціальність _____ 272 «Авіаційний транспорт» _____
(код та найменування)
Освітня програма _____ Інтелектуальні транспортні системи _____
(найменування)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Костянтин ДЕРГАЧОВ
(підпис) (ім'я та прізвище)
«__» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

_____ Щербини Олексія Сергійовича _____
(прізвище, ім'я та по батькові)

1. Тема кваліфікаційної роботи «Розробка програмного забезпечення для попередньої обробки відеоданих в комплексі задач інтелектуальних транспортних систем»

керівник кваліфікаційної роботи

канд. техн. наук, доцент Краснов Леонід Олександрович

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом Університету № 1968-уч від «06» листопада 2023 року.

2. Термін подання здобувачем кваліфікаційної роботи «08» січня 2024 р.

3. Вихідні дані до роботи Математична модель руху БПЛА, вимоги до функціонування системи та проведення аерофотозйомки

4. Зміст пояснювальної записки (перелік завдань, які потрібно розв'язати) _____

Вступ, аналіз проблеми, опис об'єкту, математичний опис об'єкту, синтез середовища для моделювання, конструкторська частина, розробка дослідницького процесу, експериментальна частина, економічна частина, заключення, список використаних джерел

5. Перелік графічного матеріалу 10 плакатів формату А1 у вигляді слайдів

6. Консультанти розділів кваліфікаційної роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|--------------------------|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| Стан проблеми | к.т.н., доц. Краснов Л.О. | 12.09.23 | 10.01.24 |
| Аналіз даної задачі | к.т.н., доц. Краснов Л.О. | 12.09.23 | 10.01.24 |
| Конструктор. частина | к.т.н., доц. Краснов Л.О. | 12.09.23 | 10.01.24 |
| Дослідницька частина | к.т.н., доц. Краснов Л.О. | 12.09.23 | 10.01.24 |
| Експ.-практ. частина | к.т.н., доц. Краснов Л.О. | 12.09.23 | 10.01.24 |
| Економічне обґрунтування | к.т.н., доц. Краснов Л.О. | 12.09.23 | 10.01.24 |

Нормоконтроль _____ Краснов Л.О. « 16 »
01 2024 р. (підпис) (ім'я та прізвище)

7. Дата видачі завдання « 12 » вересня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

| № п/п | Назва етапів кваліфікаційної роботи | Строк виконання етапів кваліфікаційної роботи | Примітка |
|-------|--|---|----------|
| 1 | Формулювання теми проекту. | 08.09.2023 | |
| 2 | Розробка технічного завдання | 12.09.2023 | |
| 3 | Огляд стану проблеми і патентний пошук. Математичний опис системи управління. | 06.10.2023 | |
| 4 | Аналіз предметної області | 20.10.2023 | |
| 5 | Конструкторська частина роботи. Дослідницька частина роботи. Економічне обґрунтування розробки | 22.12.2023 | |
| 6 | Експериментально-практична частина | 26.12.2023 | |
| 7 | Оформлення розрахунково-пояснювальної записки і графічного матеріалу. Попередній захист роботи | 08.01.2024 | |
| 8 | Рецензування проекту. Захист проекту в ЕК | | |

Здобувач _____ Олексій ЩЕРБИНА
(підпис) (ім'я та прізвище)

Керівник кваліфікаційної роботи _____ Леонід КРАСНОВ
(підпис) (ім'я та прізвище)

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

Кафедра систем управління літальних апаратів (№301)

«ЗАТВЕРДЖУЮ»

Завідуючий кафедрою 301

к.т.н., с.н.с., доцент

Костянтин ДЕРГАЧОВ

«__» _____ 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на дипломне проектування

Щербини Олексія Сергійовича

- 1 Тема роботи: Розробка програмного забезпечення для попередньої обробки відеоданих в комплексі задач інтелектуальних транспортних систем
 затверджена наказом по університету від «06» листопада 2023 р. № 1968-уч .
- 2 Строк здачі студентом закінченої роботи « 08 » _____ січня _____ 2024 р.
- 3 Область застосування розробки: робота з нестійкими космічними об'єктами
- 4 Початкові дані для розроблювальної системи
 - 4.1 Призначення і мета створення системи: створення ефективної системи визначення польоту для проведення обробки відеоданих
 - 4.2 Загальні відомості завдання на дипломний проект, програмні аналоги для відображення керованих рухомих графічних об'єктів _____
- 5 Технічні вимоги до каналів системи управління
 - 5.1 Питання, що підлягають розробці: формування математичної моделі, вибір виконавчого та вимірювальних пристроїв
 - 5.2 Режим роботи системи (безперервний, циклічний, одноразової дії): автоматичний
- 6 Умови експлуатації системи:
 - 6.1 Кліматичні вимоги до експлуатації (температура середовища, у якій буде працювати система управління, її вологість, вміст хімічно активних компонентів і т.ін.): температура середовища ± 20 градусів за Цельсієм.
 - 6.2 Механічні вимоги (вібрація, тряска, можливі перекося, удари, нахили і т.ін.): можливі вібрація, удари, нахили.
 - 6.3 Наявність перешкод (електричні наведення радіоперешкоди, магнітні впливи): можна знехтувати
- 7 Додаткові функції, реалізовані системою (сигналізація про несправності, реєстрація необхідної інформації, самоконтроль самої системи і т.ін.): Дана система не має додаткових функцій.
- 8 Обсяг виконуваних розроблювачем робіт
 - 8.1 Етапи проведення роботи: I етап: Оцінка стану проблеми.
II етап: Вербальний і математичний опис об'єкту
III етап: Синтез алгоритму програмного забезпечення
 - 8.2 Обсяг розробки по кожному етапу: _____
 - 1) стан проблеми і постановка задачі проектування (ст) _____
 - 2) опис задачі та математичний опис (ст) _____

3) конструкторська частина (ст) _____

4) розробка дослідницького процесу (ст) _____

5) економічна частина (ст) _____

9. Параметри обладнання системи

9.1. Габарити розміри визначаються у процесі проектування _____

9.2. Маса визначається в процесі проектування _____

9.3. Вимоги до конструктивного виконання і розміщенню визначаються в процесі проектування _____

9.4. Інші вимоги не передбачаються _____

10. Вимоги безпеки визначаються в процесі проектування _____

11. Дослідницька частина

11.1. Розробка дослідницького процесу розробка технології дослідження контролера з периферійним обладнанням _____

11.2. Умови і вимоги технологія перевірки апаратних засобів _____

11.3. Очікуваний результат створення повністю працездатної та повнофункціональної моделі _____

12. Економічна частина

12.1. Розробити (розрахувати, отримати): трудомісткість виконання роботи, розрахунок витрат на виконання роботи, договірна ціна роботи. _____

12.2. Умови і вимоги одиничне виробництво, розрахунок проводиться за статтями калькуляції _____

12.3. Очікуваний результат собівартість розробки програмного забезпечення грн. з урахуванням усіх витрат _____

13. Перелік графічних матеріалів та їх формат 10 плакати формату А4 у слайдах:

1) опис об'єкту автоматичного управління, математична модель об'єкту, результат дослідження об'єкту.

2) середовище розробки.

3) дослідницька частина, економічна частина.

4) висновок.

14. Мова підготовки пояснювальної записки (захисту) українська _____

Керівник роботи
к.т.н., доц. Краснов Л.О.
(П.І.Б.)

Прийняв до виконання
Щербина О.С.
(П.І.Б. студента)

«12» вересня 2023 р.

«12» вересня 2023 р.

Погоджено з питань:

конструкції
к.т.н., доц. Краснов Л.О.
(П.І.Б.)

дослідницької частини
к.т.н., доц. Краснов Л.О.
(П.І.Б.)

«12» вересня 2023 р.

«12» вересня 2023 р.

економіки
к.т.н., доц. Краснов Л.О.

(П.І.Б.)

«12» вересня 2023 р.

РЕФЕРАТ

85 с., 30 рис., 7 табл.

У роботі розглянуті питання, пов'язані з визначенням кутового положення безпілотного літального апарату за даними відеозйомки, що проводиться на борту безпілотного літального апарату. Для досягнення цілей роботи проведено аналіз існуючих рішень для визначення траєкторій руху, складено необхідні математичні моделі, проведено процес кутового переміщення ЛА, виконано оцінку характерних кутів літального апарату.

ТЕХНІЧНИЙ ЗІР, БЕЗПІЛОТНИЙ ЛІТАЛЬНИЙ АПАРАТ, КУТОВЕ
ПОЛОЖЕННЯ, НАВІГАЦІЙНИЙ ВИМІРЮВАЧ

СПИСОК СКОРОЧЕНЬ

Об'єкт АКТ – об'єкт аерокосмічної техніки

МБПЛА - малогабаритний безпілотний літальний апарат

БЖД – безпека життєдіяльності

ТЗ – Технічне завдання

КЕСН – кореляційно-екстремальна система навігації

ЛА – літальний апарат

ПП – програмний продукт

ПЗ – програмне забезпечення

ЗМІСТ

| | |
|--|----|
| ВСТУП..... | 10 |
| 1 СТАН ПРОБЛЕМИ ВИЗНАЧЕННЯ КУТОВОГО ПОЛОЖЕННЯ МАЛОГАБАРАРИТНИХ БЕЗПЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ..... | 11 |
| 1.1 Загальна характеристика проблеми та існуючі підходи до її вирішення | 11 |
| 1.2 Аналіз технічного завдання | 12 |
| 1.3 Огляд науково-технічної літератури та патентів | 12 |
| 2 ДОСЛІДЖЕННЯ ПРОБЛЕМИ І СИНТЕЗ МАТЕМАТИЧНИХ МОДЕЛЕЙ | 17 |
| 2.1 Цілі, завдання, структура та вибір шляху вирішення проблеми..... | 17 |
| 2.2 Математичні моделі | 19 |
| 2.3 Оцінка результатів моделювання | 24 |
| 3 ПРОЕКТУВАННЯ СИСТЕМИ ВИЗНАЧЕННЯ КУТОВОГО ПОЛОЖЕННЯ ЛА | 28 |
| 3.1 Алгоритми реалізації моделей | 28 |
| 3.2 Вибір засобів реалізації..... | 29 |
| 3.3 Проектування інтерфейсу та функціональних модулів | 36 |
| 4 ДОСЛІДНИЦЬКА ЧАСТИНА..... | 43 |
| 4.1 Завдання та методи проведення досліджень..... | 43 |
| 4.2 Програма проведення досліджень | 44 |
| 4.3 Результати досліджень | 44 |
| 4.5 Висновки за розділом | 45 |
| 5 ЕКСПЕРИМЕНТАЛЬНО-ПРАКТИЧНА ЧАСТИНА | 47 |
| 5.1 Завдання та засоби виконання експериментальної розробки | 47 |
| 5.2 Опис лабораторної установки..... | 47 |
| 5.3 Характеристика розробленого програмного забезпечення..... | 49 |
| 5.4 Результати експериментальних досліджень | 51 |
| 5.4 Висновки за розділом | 58 |
| 6 ЕКОНОМІЧНА ЧАСТИНА | 59 |
| 6.1 Опис програмного продукту..... | 59 |
| 6.2 Сегментація ринку..... | 59 |
| 6.3 Розрахунок собівартості та ціни виготовлення програмного забезпечення | 61 |
| 6.4 Розрахунок точки беззбитковості | 67 |
| 6.5 Висновок | 69 |
| ЗАКЛЮЧЕННЯ..... | 70 |

ВСТУП

Актуальність теми. Забезпечення безпеки польотів безпілотної авіації, а також контроль траєкторії польоту та кутової орієнтації щодо власного центру мас є актуальною науковою проблемою. Процес визначення положення, швидкості та орієнтації МБПЛА, на основі вимірювальних даних включає рішення двох взаємопов'язаних завдань, одна з яких — власне навігаційна, пов'язана з визначенням положення центру мас МБПЛА у просторі, а друга – визначення кутового положення та кутових швидкостей у обраній системі координат відносно власного центру мас. При цьому вирішення другого завдання неможливе без попереднього рішення першого.

Метою проведених досліджень є розробка та дослідження методів визначення кутового положення малогабаритного безпілотної літального апарату.

Для досягнення цілей роботи повинні бути вирішені такі завдання:

- аналіз існуючих систем та методів вирішення задачі навігації;
- Формування математичних моделей процесу візуальної навігації безпілотної літального апарату;
- моделювання процесу зміни кутового положення ЛА та визначення характерних кутів;
- експериментальне дослідження процесів визначення кутового положення БПЛА при моделюванні його руху.

Як методи дослідження застосовується метод системного підходу, формування навігаційних систем, при розробці програмного забезпечення застосовувалася концепція об'єктно-орієнтованого програмування. Як засоби програмування використовувалося середовище моделювання MatLab, при цьому вихідні відеофайли для аналізу кодувалися у форматі avi.

Як основні технології використовувані в роботі використовувалися значення пакети – розширення MATLAB – Image Processing Toolbox, Vision System Toolbox та Computer Vision System Toolbox.

1 СТАН ПРОБЛЕМИ ВИЗНАЧЕННЯ КУТОВОГО ПОЛОЖЕННЯ МАЛОГАБАРИТНИХ БЕЗПЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ

1.1 Загальна характеристика проблеми та існуючі підходи до її вирішення

Оптичні системи вилучення інформації знайшли широке застосування в системах космічного моніторингу земної та водної поверхні для локалізації розливів нафтопродуктів у морських та річкових акваторіях, у системах льодової розвідки, визначення параметрів атмосфери, температури та солоності води, вимірювання електрофізичних параметрів підстилаючих поверхонь та ін. [1- 4].

В даному розділі як об'єкт АКТ виступає малогабаритний безпілотний літальний апарат (МБПЛА), що здійснює автоматичний політ.

Забезпечення безпеки польотів МБПЛА, а також контроль траєкторії польоту та кутової орієнтації щодо власного центру мас є актуальною науковою проблемою. Процес визначення положення, швидкості та орієнтації МБПЛА, на основі вимірювальних даних включає рішення двох взаємопов'язаних завдань, одна з яких — власне навігаційна, пов'язана з визначенням положення центру мас МБПЛА у просторі, а друга – визначення кутового положення та куткових швидкостей у обраній системі координат відносного власного центру мас. При цьому вирішення другого завдання неможливе без попереднього рішення першого.

Для діагностування втрати працездатності системи визначення кутового положення МБПЛА пропонується використовувати додатковий вимірювальний канал, який використовує технології «машинного зору», включаючи методи теорії обробки зображень та розпізнавання образів. Реалізація та створення таких додаткових діагностичних каналів є перспективним напрямом при синтезі бортових високоточних інтегрованих систем навігації літальних апаратів [1].

1.2 Аналіз технічного завдання

Відповідно до технічного завдання як малогабаритного безпілотного літального апарату використовувалася модель літака типу Байрактар, яка найбільш повно відповідає вимогам технічного завдання. Під час проведення експериментального дослідження відеокамера жорстко закріплювалася на масогабаритному макеті літального апарату, запис відеоінформації про еволюції моделі безпілотного літального апарату проводився у форматі avi.

Практична застосування технологій «машинного зору» значною мірою заснована на досягненнях в галузі мікроелектроніки та розробках сенсорних бортових технічних пристроїв у видимому, інфрачервоному та радіолокаційному діапазонах випромінювання електромагнітного спектру, а також на можливості реалізації складних алгоритмів комплексної обробки поточної інформації від кількох джерел у сучасних обчислювальних архітектурах.

Особливу важливість ця проблема має для стабілізації рухів МБПЛА, визначальним фактором для якої є надійна оцінка кутів тангажу та нахилу в реальному часі. Аналіз прикладних завдань дозволяє сформулювати вимоги до точності оцінки кутів на рівні $3\delta < 4-5^\circ$ (тангаж), $3\delta < 3^\circ$ (крен) [2].

Передбачається, що діагностування системи визначення кутового положення МБПЛА буде проводитись за допомогою системи спостереження, що ґрунтуються на застосуванні методів аналізу відеоінформації. Для реалізації системи діагностування необхідний вимірювач навігаційної інформації, що дозволяє визначити кутове положення МБПЛА з використанням даних.

Таким чином, метою роботи є розробка навігаційного вимірювача 2-го рівня, що дозволяє визначити кутове положення МБПЛА щодо центру мас, в реальному часі за даними.

1.3 Огляд науково-технічної літератури та патентів

Оптичні системи застосовуються для вирішення військових завдань: для наведення та самонаведення літальних апаратів (ЛА) на наземні, морські та повітряні цілі; для виявлення низьколітаючих об'єктів на тлі неба та земної поверхні; для космічної та повітряної радіорозвідки у рамках розвідувально-ударних комплексів [5-11]. Багато завдань військового та народногосподарського значення вирішуються завдяки комплексному застосуванню оптичних та інших систем (радіолокаційних, інфрачервоних, оптичних).

На особливу увагу заслуговує питання застосування оптичних систем для навігації ЛА різного призначення за наземними орієнтирами природного та штучного характеру [4].

Системи навігації літальних апаратів – космічних апаратів, літаків, ракет, безпілотних літальних апаратів, функціонують на основі автономного визначення (числення) координат за сигналами систем, що накопичують помилки з часом (інерційні, курсо-доплерівські та ін. системи), та систематичної чи епізодичної корекції траєкторії польоту ЛА за сигналами радіонавігаційних систем, бортових радіолокаційних станцій та пристроїв візування орієнтирів [1-4].

По можливості початкового визначення розрізняють такі методи навігації [4]:

- позиційна навігація, що дозволяє визначати поточну позицію об'єкта самостійно, без залучення даних про попередню позицію;
- Непозиційна навігація, яка не дозволяє визначати поточну позицію об'єкта самостійно.

Особливо поширені методи непозиційної навігації, зокрема інерційної. Отримання непозиційної інформації забезпечує автономність навігації чи підвищує рівень її автономності.

До методів непозиційної навігації відносять:

- метод обчислення шляху;
- оглядово-порівняльний, так званий кореляційно-екстремальний метод.

Засоби та особливості інерційної навігації. Основною системою навігації у системі управління ЛА є інерційна навігаційна система (ІНС).

Поточну позицію об'єкта навігації в ІНС встановлюють опосередковано за його початковою позицією та результатом дворазового інтегрування вектора прискорення або одноразового інтегрування вектора швидкості.

Для цього необхідно:

- вимірювати зазначені вектори або один із них за допомогою акселерометрів або дистанційних вимірювачів швидкості;
- Зберігати інформацію про початкову орієнтацію об'єкта.

Інформацію одержують:

- безпосередньо від позиційних гіроскопів – датчиків кутових відхилень;
- в результаті інтегрування кутових швидкостей, що видаються швидкісними (диференціюючими) гіроскоп-датчиками кутових швидкостей.

Системи, що реалізують оглядово-порівняльний метод навігації. КЕСН здійснює корекцію (прив'язку) траєкторії руху ЛА в обраних заздалегідь точках шляхом поєднання, в окремому випадку двовимірному зображенні кожного району прив'язки, або зображення самого об'єкта навігації на певному тлі, що називається поточним зображенням (ТІ), з еталонним зображенням (ЕІ), що зберігається в пам'яті бортового спецобчислювача (СВ) [1-3]. Результатом поєднання зображень є побудова вирішальної функції (РФ) шляхом обчислення міри подібності ТІ з кожним зі зрушених фрагментів ЕІ та пошук такого значення зсуву, при якому досягається екстремум РФ. Обчислювані таким чином помилки визначення координат об'єкта використовуються для корекції помилок ІНС.

Як інформативні ознаки КЕСН використовують поверхневі поля (поле рельєфу, оптичного, радіолокаційного і радіотеплового контрасту) [1,4]. Як об'єкти – орієнтири навігації можуть служити:

- .- спеціально збудовані об'єкти - орієнтири навігації;
- окремі промислові будівлі;
- околиці міст;
- залізничні мости;
- іригаційні споруди (греблі, греблі);
- злітно-посадкові смуги аеродромів;

- автостради;
- закрути акваторій річок, берег моря.

Аналіз складу, структури та характеристик орієнтирів навігації дозволили виявити основні особливості даних об'єктів:

- Об'єкти є малорозмірними, з великим вмістом металу;
- об'єкти є майданними, великогабаритними;
- Об'єкти є протяжними;
- об'єкти є водні перепони;
- об'єкти сусідять з природними та штучними водоймами;
- Об'єкти відрізняються наявністю складних залізобетонних конструкцій.

Стаціонарні об'єкти мають строго певні геометричну форму і розміри, мають набір стійких інформативних ознак. До того ж стаціонарні об'єкти прив'язані до координат географічної карти. Це дозволяє ідентифікувати об'єкти за допомогою оптичних датчиків видимого, інфрачервоного та міліметрового діапазонів електромагнітних хвиль (ІЧ та ММД).

Використання як орієнтири навігації рухомих об'єктів, не оснащених випромінюючою навігаційною апаратурою, вимагає врахування ряду особливостей як самих об'єктів, так і апаратури КЕСН, що здійснює навігацію ЛА за даними об'єктами - орієнтири. Насамперед, внаслідок обмежень на габарити бортових антен ЛА, рухомі об'єкти є малорозмірними. (В порівнянні з розмірами дозволених елементів на земній поверхні). Хоча відносна швидкість переміщення рухомих наземних об'єктів (порівняно зі швидкістю польоту навігаційних ЛА) невелика, проте рухомі об'єкти можуть змінювати своє місце розташування і не прив'язані до конкретної карти місцевості, що ускладнює завчасне створення ЕІ об'єктів на різних фонах земної поверхні.

Особливістю рухомих наземних об'єктів є також те, що вони можуть об'єднуватись у групи (наприклад, колони автотранспорту). При цьому необхідний оперативний аналіз таких факторів, як кількість об'єктів у групі, порядок побудови, швидкість та напрямок пересування окремих об'єктів та всієї групи загалом.

Аналіз та облік перерахованих особливостей, а також аналіз наявності стійких інформативних ознак об'єктів (характеристик випромінювання та відображення у різних діапазонах електромагнітних хвиль) дозволяє здійснювати надійне виявлення та ідентифікацію рухомих наземних об'єктів пасивно-активними КЕШ.

Зарубіжними аналогами кореляційно-екстремальних систем навігації ЛАТЕРКОМ і DJSMAC видимого діапазону здійснюють навігацію літаків і крилатих ракет Tomahawk [5,6], Навігація керованої ракети UGSP по наземних орієнтирів здійснюється в комбінованому режимі за допомогою активної радіолокаційної (РЛК) та пасивної радіометричної (РМ) КЕШ 3-мм діапазону . Навігація ракети середньої дальності Pershing-2 (нині ракета "Герма") здійснюється за допомогою радіолокаційної КЕШ сантиметрового діапазону RADAG [8-11]. Відомі розробки пасивних радіометричних КЕШ міліметрового діапазону RAC, DJRAC, MICRAD для навігації літаків, крилатих та балістичних ракет [6,12,13].

Сучасні тенденції побудови КЕШ[13] спрямовані на створення одноканальних скануючих та багатоканальних пасивних навігаційних систем. Пасивні КЕШ вигідно відрізняються від активних РЛК систем більш високою скритністю функціонування та відсутністю замирань відбитого від наземних об'єктів сигналу.

Недоліки властиві сучасним системам навігації ЛА.В першу чергу, існує велика невизначеність застосування КЕШ ЛА по об'єктах, розташованих на складному, неоднорідному та слабо контрастному фоні земної поверхні. Невизначеність також полягає в необхідності виявляти та ідентифікувати слабконтрастні та рухливі об'єкти в умовах складної перешкодової, погодної обстановки та обмеженої оптичної видимості. До того ж вимагає якісної та кількісної оцінки та врахування впливу різного роду спотворень поточних зображень наземних об'єктів у КЕШ.

2 ДОСЛІДЖЕННЯ ПРОБЛЕМИ І СИНТЕЗ МАТЕМАТИЧНИХ МОДЕЛЕЙ

2.1 Цілі, завдання, структура та вибір шляху вирішення проблеми

У навігаційному вимірнику, що розробляється, реалізуються методи машинного розпізнавання і технології «машинного зору» наступним чином.

Виконується зіставлення поточного зображення та еталонного модельного опису сцени спостереження з подальшою локалізацією на поточному зображенні заданих об'єктів сцени та визначенням значень поточних координат цих об'єктів з метою формування сигналів неузгодженості функціонування бортової системи визначення кутового положення та існуючої апріорної та апостеріорної інформації, що розглядається.

Апріорна інформація включає, як правило, такі дані: тип та кількість об'єктів сцени, їх взаємне розташування, геометричні та яскраві характеристики, характеристики двовимірних полів (поля рельєфу, оптичного розмаїття тощо).

Для реалізації у навігаційному вимірнику можуть бути використані алгоритми обробки відеоінформації, класифікація яких представлена на рис.2.1.

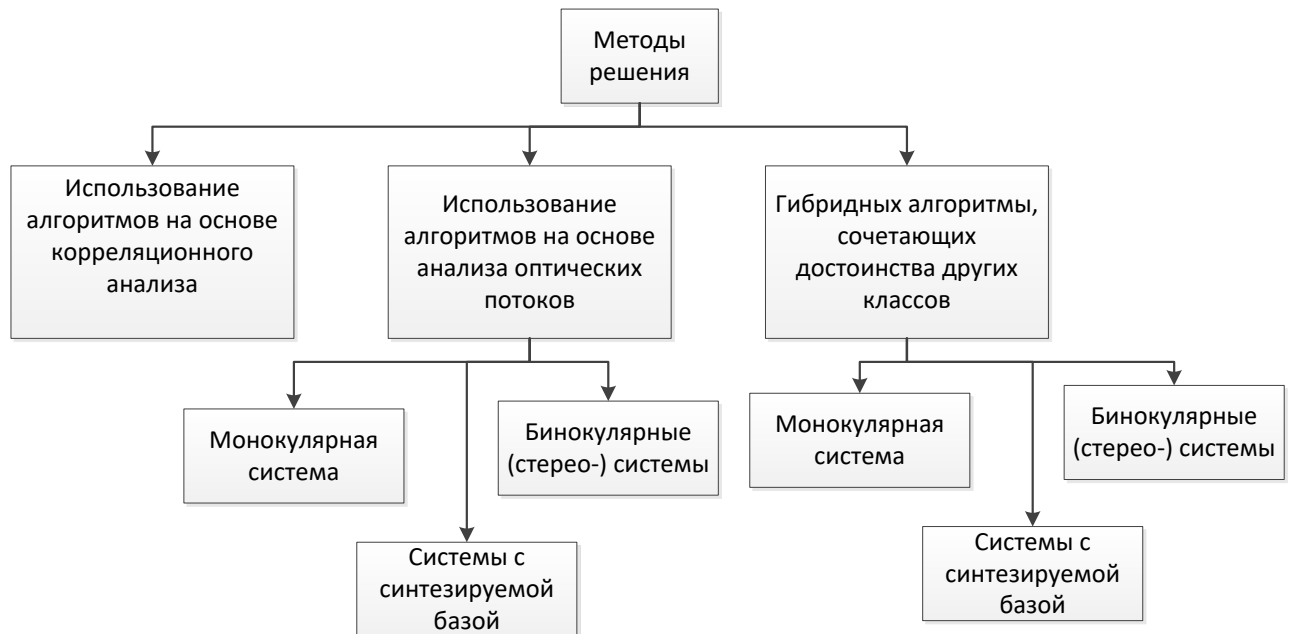


Рисунок 2.1 – Класифікація алгоритмів обробки відеоінформації для реалізації у навігаційному вимірнику

Як видно із класифікації на рис. 2.1, існує кілька підходів до розрахунку параметрів відносного руху камери (або системи з камер), а отже, і її орієнтування. Кожен з цих підходів має переваги та недоліки.

Складність використання дискретного та диференціального підходів у реальних умовах полягає у їхній високій чутливості до шумів у прийнятій відеоінформації. У свою чергу, використання методу найменших квадратів потребує обчислювальних ресурсів, які не забезпечують сучасні бортові процесори.

Оцінка кутів орієнтації (крену, тангажу та нишпорення) МБПЛА в реальному часі на основі аналізу загального оптичного потоку та відповідного поля руху є надзвичайно складним завданням.

Один з можливих способів подолання сформульованої труднощі полягає у використанні поля руху, що формується на основі оцінок зміщень не точок, а окремих блоків, на які попередньо розбивається зображення.

Відносні усунення блоків на двох кадрах відеопослідовності характеризуються векторами поля руху, які перебувають на підставі оцінки положення відповідних блоків на кожному кадрі за допомогою кореляційних функцій.

Не один з алгоритмів класу кореляційних або визначення оптичних потоків не може незалежно використовуватися для ефективного вирішення задачі орієнтації МБПЛА в реальному часі.

Таким чином, для вирішення задачі визначення кутів орієнтації МБПЛА, необхідний гібридний алгоритм, що поєднує в собі переваги як алгоритмів кореляційного аналізу, так і аналізів оптичних потоків. Такий алгоритм має бути побудований на основі наступних принципів:

- Необхідність забезпечення введення послідовності відео як окремих кадрів;
- Необхідність виділення характерної точки (або об'єкта з відомими геометричними розмірами);
- необхідність зменшення («Звуження») кадру до «проблемної» області, що гарантовано містить об'єкт у наступних кадрах;
- необхідність визначення відстані від відеокамери до характерної точки або об'єкта, відомими геометричними об'єктами;
- безпосередня оцінка кутового зміщення будівельної осі об'єкта по крену, тангажу та нишпоренню щодо фокусного об'єкта.

2.2 Математичні моделі

Для розв'язання задачі визначення кутового положення МБПЛА необхідно розв'язати задачу позиціонування МБПЛА на основі відеозображення. Розглянемо методи та алгоритми, необхідні для вирішення задачі позиціонування МБПЛА у просторі щодо навколишніх об'єктів.

Розв'язання залишеної задачі може бути виконане у двох варіантах:

- на МБПЛА закріплюється відеокамера, яка знімає сцену в навколишньому просторі. За знімками камери необхідно аналізувати зміни у просторі положення її носія;
- нерухома зовнішня відеокамера робить огляд області простору, в якій присутній об'єкт, що рухається, і оцінює параметри зміни його положення.

Основне обмеження: відеокамера – єдиний сенсор сприйняття навколишнього простору.

Для вирішення завдання у послідовному потоці кадрів відео необхідно обчислювати, як і наскільки перемістився МБПЛА. Щоб уникнути надлишкових обчислень, необхідно відстежити на поточному зображенні лише деякі точки, зміна положення яких у наступному кадрі визначить зміну положення МБПЛА. Такі точки називатимемо опорними. Оскільки опорні точки на отримуваних зображеннях мають дві координати (зображення є плоским), а опорна точка у просторі має 3 координати, далі необхідно отримати алгоритм перетворення з метою оцінки зміни її просторового становища заданої системі координат.

Можна виділити три основні етапи роботи.

Спочатку потрібно визначити внутрішні характеристики камери, тобто. провести внутрішнє калібрування з прийнятною точністю. Зазначимо, що цю процедуру достатньо провести лише один раз.

Другим етапом є вирішення задачі визначення опорних точок на зображеннях сцени, що реєструється. Це завдання може бути вирішена різними способами, але важливо забезпечити необхідну точність та якість їх детектування.

Заключним етапом роботи є визначення зміни просторового положення об'єкта шляхом пошуку невідомих коефіцієнтів матриці обертання та вектора зсуву заданої системи координат.

Таким чином, завдання визначення зміни положення у просторі мобільного пристрою розбивається на три частини:

- попереднє визначення внутрішніх параметрів камери;
- знаходження відповідних опорних точок на зображеннях;
- розрахунок матриці поворотів та вектора зсуву камери в заданій системі координат.

Різні точки простору предметів відображаються оптичною системою камери в просторі зображень на різних відстанях від фокальної площини. Але якщо відстань між камерою і сценою, що спостерігається, значно перевищує фокусну відстань оптичної системи, можна вважати, що зображення будується в її фокальній площині. В цьому випадку можна скористатися проєктивною моделлю камери, в якій зображення тривимірного об'єкта

виходить проектуванням його у фокальну площину (площину зображення) через єдину точку, яка називається оптичним центром. Пряма лінія, перпендикулярна площині зображення і проходить через цю точку, називається оптичною віссю камери, а точка перетину оптичної осі з площиною зображення головною точкою.

Визначимо у тривимірному просторі ортогональну праву систему координат, початок якої збігається з оптичним центром, вісь з оптичною віссю камери. Така система називається стандартною системою координат камери. Нехай площина зображення знаходиться на відстані оптичного центру. У цій площині задаємо систему координат з початком в головній точці і осями і паралельними осям і відповідно (рис. 2.1). $OXYZ$ $oxyz$ $OXOY$

У стандартній системі координат проекцією точки тривимірного простору з координатами ϵ точка в площині зображення з координатами, причому $M(X, Y, Z) m(x, y) x = fX/Z y = fY/Z$

Для повного опису камери слід врахувати, що для реєстрації зображення в площині зображення камери міститься фотоприймач.

Загалом вимір координат у фотоприймачі здійснюється в одиницях, відмінних від одиниць, що задають координати в стандартній системі.

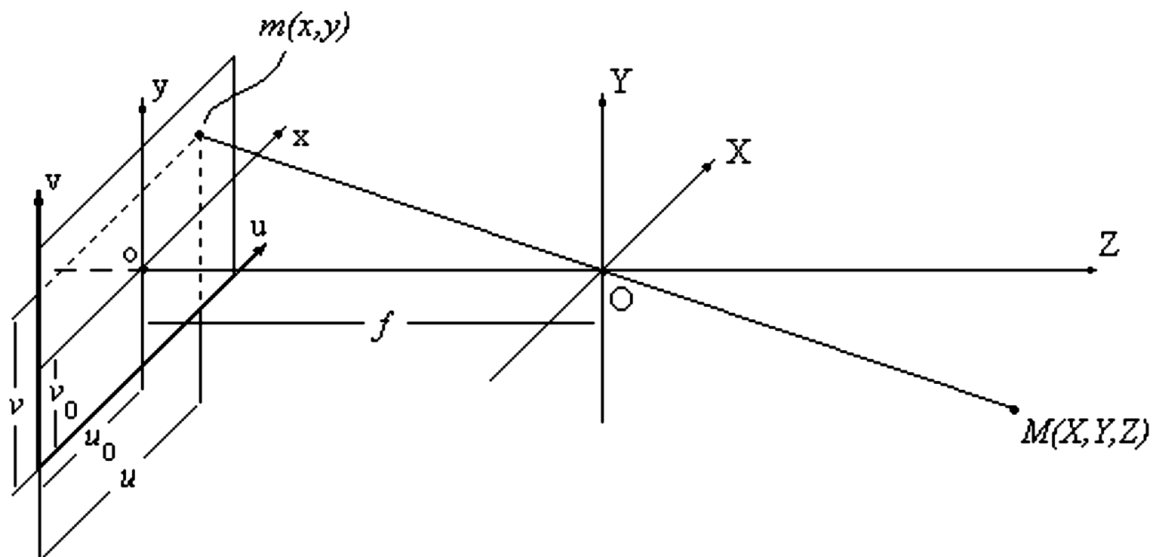


Рисунок 2.1 - Система координат камери на борту ЛА

Тому для повного опису камери необхідно виразити координати точки в одиницях природних фотоприймача. У досить загальному для будь-яких фотоприймачів вигляді (рис. 2.2) це може виглядати як m

$$u = \frac{x}{w} + u_0, \quad v = \frac{y}{h} + v_0,$$

де u – координати головної точки щодо початку координат фотоприймача (у природних координатах фотоприймача); i – масштаби вздовж осей i (наприклад, відстані між осередками матричного фотоприймача вздовж рядків та стовпців). (u_0, v_0) *whoxoy*

У новій системі координати проекції точки набудуть вигляду: m

$$u = \frac{fX}{wZ} + u_0, \quad v = \frac{fY}{hZ} + v_0. \quad (2.1)$$

Введемо тривимірний вектор M , що відповідає точці (X, Y, Z) , і двовимірний вектор m , що відповідає точці (u, v) . Визначимо також вектор однорідних внутрішніх координат камери. Використовуючи ці позначення, співвідношення (1.1) можна представити у компактному векторно-матричному записі: $M = (X, Y, Z)^T$, $Mm = (x, y)^T$, $mv = (u, v, 1)^T$

$$Zv = AM, \quad (2.2)$$

$$\text{де } A = \begin{bmatrix} f/w & 0 & u_0 \\ 0 & f/h & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

– матриця, відома під назвою матриці внутрішніх параметрів камери, оскільки вона містить лише параметри оптичної системи та фотоприймача камери.

Далі матриця внутрішніх параметрів використовується на вирішення завдання визначення просторового становища камери у процесі розрахунку параметрів руху об'єкта спостереження.

Практична реалізація процедури визначення внутрішніх параметрів камери не викликає труднощів і може бути успішно виконана за допомогою програми Camera Calibration Toolbox for Matlab. Ця програма дозволяє провести калібрування з використанням спеціального плоского шаблону, на якому зображено шаховий патерн. Необхідно провести відеозйомку цього шаблону в різних ракурсах (під різними кутами), а потім після програмної обробки отримати наступний набір внутрішніх параметрів камери:

- фокусна відстань: фокусна відстань у пікселях зберігається у векторі розмірністю $2 \times 1; f$
- головна точка: основні координати головної точки збережені як вектора розмірністю 2×1 пікселя; cc
- коефіцієнт спотворення: коефіцієнт, що визначає кут між осями, зберігається в скалярі ; α_c
- спотворення: коефіцієнти спотворення зображення (радіальні та тангенціальні спотворення) зберігаються у вигляді вектора розмірністю 5×1 пікселів. k_s

Розглянемо просторове переміщення ЛА, подане на рис.2.2.

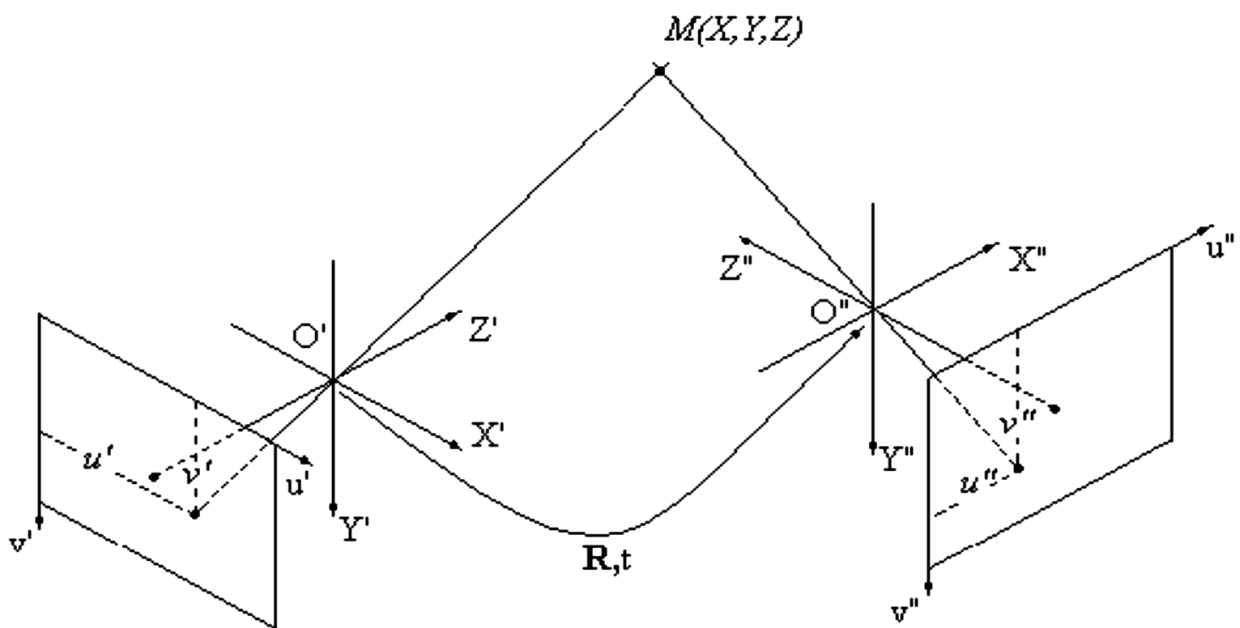


Рисунок 2.2 – Просторове положення відеокамери

Рівняння просторового положення камери має вигляд:

$$M' = R'M + t' \quad (2.3)$$

де

$R = R''R'^T$ - ортогональна матриця, що описує орієнтацію системи координат другого положення камери щодо першого

$t = -R''R'^T t' + t''$ - Вектор трансляції, що визначає положення оптичного центру другого положення камери в системі координат першого.

При цьому матриця направляючих косінусів може бути визначена в наступному вигляді:

$$R = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

α , β , γ – кути крену, тангажу та курсу

В результаті отримано модель визначення просторового становища відеокамери.

2.3 Оцінка результатів моделювання

Рух – це усунення у кадрі одних об'єктів щодо інших. У завданнях технічного зору виділяється кілька різних випадків руху:

- нерухома камера - постійне тло;
- камера, що рухається, - відносно постійне тло;
- камера, що рухається - постійно змінюється фон.

Рухливість камери та ступінь мінливості фону багато в чому обумовлює вибір методів розв'язання задач, пов'язаних із рухом. Очевидно, що наявність нерухомої камери та постійного фону – найпростіша ситуація, т.к. можна визначити рух пікселів, інтенсивність яких мало змінюється щодо інтенсивності фонових пікселів. Найчастіше зазначені умови створюються

розробки систем відеоспостереження, де досить оглядати деяку фіксовану область простору.

Камера, що рухається, на відміну від нерухомої сама є джерелом зміни сцени і фону внаслідок власного зміщення. Але при цьому з'являється можливість отримати більше інформації з отриманих даних.

Найбільш складні умови формуються при використанні камери, що рухається, коли практично неможливо виділити постійне тло. Системи автоматичного управління автономними роботами є типовими прикладами систем обробки даних, одержаних у зазначених умовах.

Наявність великої кількості об'єктів, які переміщуються у кадрі, а й перекривають одне одного, значною мірою ускладнює завдання визначення руху переважають у всіх трьох перелічених випадках.

Основна особливість завдання виявлення об'єкта, що спостерігається в кадрі полягає в необхідності забезпечити суттєве перевищення його яскравості над яскравістю фону.

Тут можуть бути використані два основні підходи:

- У разі постійного та рівномірного фону можна забезпечити високу яскравість об'єкта за рахунок виділення відповідної колірної компоненти кольорового кадру;

- При динамічному характері зміни яскравості фону найбільш простий підхід до вирішення цієї задачі полягає в тому, щоб використовувати механізм віднімання фону з кадру відео (background subtraction). Процедура віднімання передбачає, що для даного відео побудована адекватна модель фону, що враховує і наявність у кадрі фрагментів з більшою яскравістю, ніж яскравість об'єкта, що спостерігається. Цей крок включає попіксельне віднімання інтенсивності кадру відео і фонового зображення.

Існує кілька типів алгоритмів виявлення та оцінки руху об'єкта в кадрі. Це алгоритми зіставлення блоків, алгоритми, що ґрунтуються на аналізі оптичного потоку (по Лукасу-Канаді) [1–3]. Однак найпростішим у обчислювальному відношенні є кореляційно-екстремальний алгоритм, що передбачає обчислення взаємно-кореляційної функції кожного фрагмента кадру з шаблоном, пошук її максимуму, подальшу бінаризацію з відсіканням

по порозі яскравості. Це дозволяє локалізувати положення об'єкта, що спостерігається в кадрі з точністю до пікселя.

Для надійної роботи кореляційних алгоритмів необхідно забезпечити суттєве перевищення у кадрі яскравості об'єкта, що спостерігається по відношенню до яскравості фону. Для цього попередньо виділяється потрібна колірна компонента кадру, найкраще відповідна кольору об'єкта, що спостерігається, а потім по положенню максимуму функції взаємної кореляції цієї компоненти з шаблоном, розміри якого зіставні з розмірами об'єкта, визначається його положення в кадрі.

Шаблон зазвичай вибирається у вигляді квадратної одиничної матриці виду $P = \text{ones}(n, n)$. Це усуває надмірність обчислень при формуванні взаємно кореляційної функції об'єкта, що спостерігається, з шаблоном і дозволяє забезпечити прийнятну точність локалізації об'єкта спостереження в кадрі.

Взаємна кореляційна функція двох зображень описується співвідношенням: $K(m, n)$

$$K(n, m) = \sum_{i=0}^N \sum_{j=0}^M [A(i, j)B(i, j)], \quad (2.4)$$

де - двомірні масиви чисел, що визначають яскравість пікселів зображення кадру розміром i шаблону розміром точок; – координати точки, на яку обчислюється значення кореляційної функції. $A(i, j)B(i, j)M \times Np \times rp, m$

У Matlab ця процедура виконується за допомогою стандартної функції, яка обчислює двовимірну взаємну функцію кореляційну функцію двох зображень за вибірками, заданими в матрицях i . Для точної локалізації положення об'єкта в кадрі далі необхідно нормувати функцію взаємної кореляції на одиницю та провести її бінаризацію з високим порогом відсікання по яскравості. Для визначення положення об'єкта з точністю до пікселя значення порога зазвичай встановлюють від 0,9 до 0,999. $k = \text{xcorr2}(a, b)ab$

Далі необхідно визначити координати точки максимуму кореляційної функції у системі координат кадру відео. Оскільки початок координат кадру відео відповідає лівому верхньому куту знімка, потрібно методом

паралельного перенесення перейти до центральної прямокутної системи координат з початком у точці, координати якої дорівнюють половині розміру кадру. Такий перехід, що забезпечує зручність розрахунку та відображення параметрів траєкторії та обчислення потрібних показників (дальності, кутових координат та швидкості), труднощів не викликає.

До теперішнього часу досягнуто достатньої ефективності та стійкості роботи цих алгоритмів у лабораторних умовах при аналізі тестових відеопослідовностей та незначних змін освітленості сцени. У той же час, використання та розміщення відеореєстраторів на рухомих носіях та робота в природних умовах призводить до суттєвої зміни освітленості залежно від часу доби, стану атмосфери, опадів, туману, вологості, сили вітру. Це породжує погіршення якості виявлення об'єкта спостереження у разі зниження яскравості об'єкта стосовно фону. Такі фактори створюють перешкоди, що призводять до помилкових результатів розпізнавання руху з неприпустимим рівнем аномальних помилок при оцінці параметрів траєкторії. Дія шумів в умовах відеозйомки на натурі призводить до суттєвого погіршення відношення яскравості об'єкта спостереження до яскравості фону.

В рамках цієї теми для оцінки впливу шумів на якість роботи алгоритмів було проведено дослідження їх завадостійкості. Для цього використовувалося моделювання Matlab з накладенням на вихідні зображення кадрів відео різного виду шумів (гаусових, імпульсних і мультиплікативних).

3 ПРОЕКТУВАННЯ СИСТЕМИ ВИЗНАЧЕННЯ КУТОВОГО ПОЛОЖЕННЯ ЛА

3.1 Алгоритми реалізації моделей

Структурна схема алгоритму обробки відеоданих представлена на рис.

3.1.

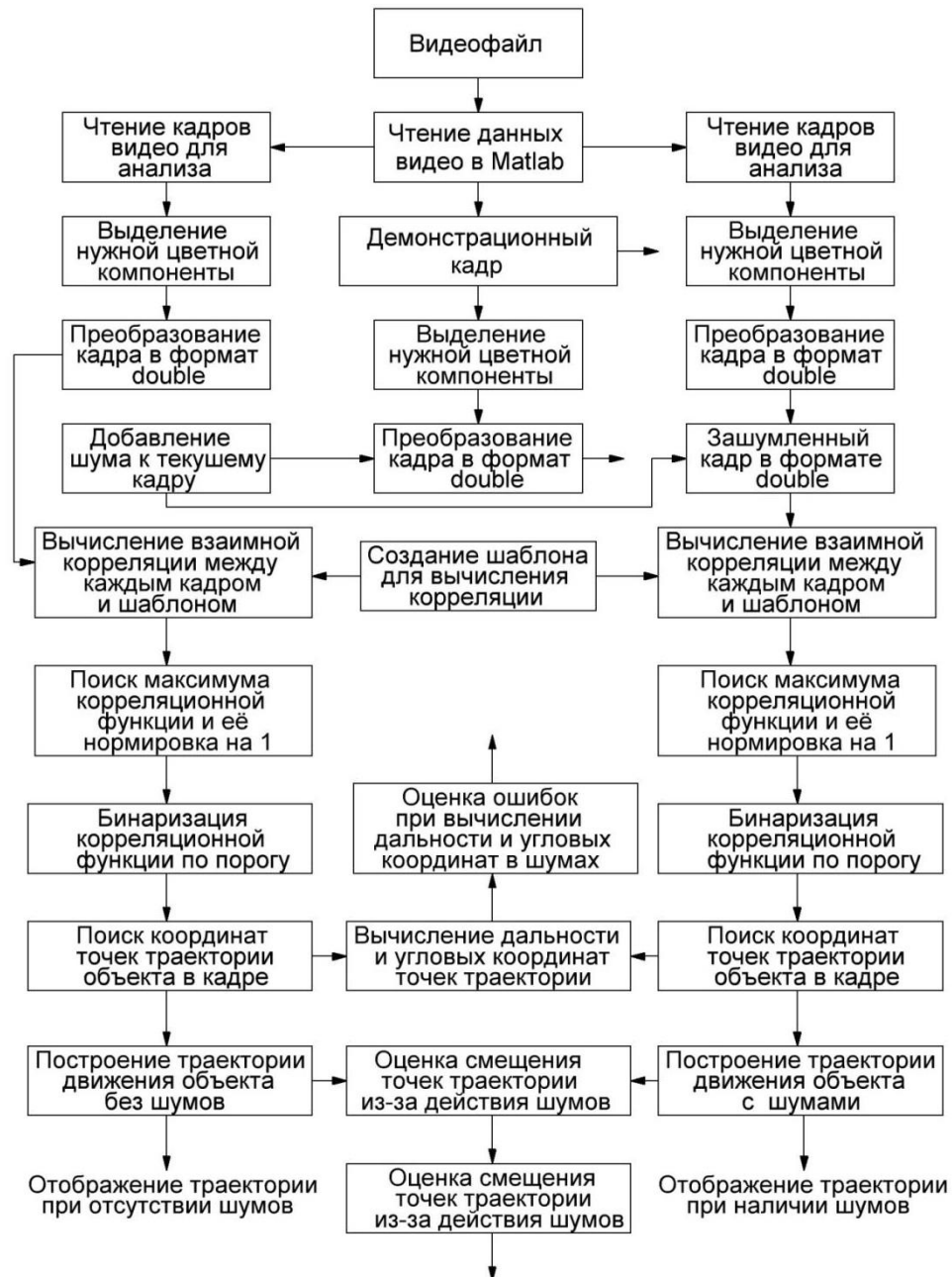


Рисунок 3.1 – Структура алгоритму оцінки руху за відеоданими методами кореляційного аналізу зображень

Ця схема відображає основні перетворення вхідної відеопослідовності. Алгоритм передбачає введення шумової компоненти та кількісний аналіз характеристик траєкторії руху залежно від рівня завадової компоненти.

3.2 Вибір засобів реалізації

Одним з найбільш доступних та конструктивних підходів для вирішення комплексу задач з аналізу та обробки зображень є використання програми MATLAB з програмами Image Processing Toolbox, пакета Computer Vision System Toolbox у додатку Simulink, пакетів Image Acquisition Toolbox та Wavelet Toolbox.

Пакет Image Processing Toolbox надає великий набір засобів для цифрової обробки та аналізу зображень. Будучи тісно пов'язаним із середовищем розробки додатків MATLAB, Image Processing Toolbox звільняє від виконання тривалих операцій кодування та налагодження алгоритмів, дозволяючи зосередити зусилля на вирішенні основного наукового чи практичного завдання. Основні властивості пакета:

- відновлення та виділення деталей зображень;
- робота з виділеною ділянкою зображення;
- аналіз зображення;
- лінійна фільтрація;
- перетворення зображень;
- геометричні перетворення;
- збільшення контрастності важливих деталей;
- бінарні перетворення;
- обробка зображень та статистика;
- колірні перетворення;
- зміна палітри;
- перетворення типів зображень.

Пакет Image Processing Toolbox дає великі можливості для створення та аналізу графічних зображень у середовищі MATLAB. Він забезпечує

надзвичайно гнучкий інтерфейс, що дозволяє маніпулювати зображеннями, інтерактивно розробляти графічні картини, візуалізувати набори даних та анотувати результати для технічних описів. Гнучкість, з'єднання алгоритмів пакета з такою особливістю MATLAB, як матрично-векторний опис роблять пакет дуже вдало пристосованим для вирішення практично будь-яких завдань з розробки та представлення графіки. У MATLAB входять спеціально розроблені процедури, що дозволяють підвищити ефективність графічної оболонки.

Це дозволяє користувачеві витратити значно менше часу та сил на створення стандартних графічних зображень і, таким чином, сконцентрувати зусилля на аналізі важливих деталей та особливостей зображень.

MATLAB та пакет Image Processing Toolbox максимально пристосовані для розвитку, впровадження нових ідей та методів користувача. Для цього є набір пакетів, що сполучаються, спрямованих на вирішення всіляких специфічних завдань.

Є дуже широке коло завдань, які користувачі вирішують за допомогою даного пакету, наприклад, космічні дослідження, військові розробки, астрономія, медицина, біологія, робототехніка, матеріалознавство, генетика і т.д.

Пакет Computer Vision System Toolbox. Одним з найбільш продуктивних методів обробки відео та зображень у MATLAB є використання пакета Computer Vision System Toolbox у додатку Simulink. Цей пакет містить велику бібліотеку блоків для обробки відео та зображень. Основні з них:

- блоки аналізу та покращення зображень (Analysis & Enhancement),
- фільтрації (Filtering),
- геометричних перетворень (Geometric Transformations),
- морфологічних перетворень (Morphological Operations).

Пакет Image Acquisition Toolbox дозволяє безпосередньо підключати, налаштовувати та керувати засобами формування зображень та потокового відео. Це значно розширює середовище технічних обчислень MATLAB. Разом з іншими програмами, такими як Image Processing Toolbox та Computer

Vision System Toolbox, забезпечує високий рівень проведення аналізу та обробки даних.

Пакет Wavelet Toolbox надає користувачеві повний набір програм для дослідження багатовимірних нестационарних явищ за допомогою вейвлетів (коротких пакетів). Порівняно нещодавно створені методи пакету Wavelet розширюють можливості користувача в тих областях, де зазвичай застосовується техніка Фур'є-розкладання. Він може бути корисним для таких додатків, як обробка мовлення та аудіо-сигналів, телекомунікації, геофізика, фінанси та медицина. Основні властивості пакета:

- удосконалений графічний інтерфейс користувача і набір команд для аналізу, синтезу, фільтрації сигналів і зображень;
- перетворення багатовимірних безперервних сигналів;
- дискретне перетворення сигналів;
- декомпозиція та аналіз сигналів та зображень;
- широкий вибір базисних функцій, включаючи корекцію граничних ефектів;
- пакетна обробка сигналів та зображень;
- аналіз пакетів сигналів, що базується на ентропії;
- фільтрація з можливістю встановлення жорстких та нежорстких порогів;
- оптимальне стиснення сигналів та зображень.

Користуючись цим пакетом, можна аналізувати такі особливості, які упускають інші методи аналізу сигналів, тобто тренди, викиди, розриви похідних високих порядків. Пакет дозволяє стискати та фільтрувати сигнали без явних втрат навіть у тих випадках, коли потрібно зберегти і високо-і низькочастотні компоненти сигналу. Є алгоритми стиснення та фільтрації та для пакетної обробки сигналів. Програми стиснення виділяють мінімальну кількість коефіцієнтів, що становлять вихідну інформацію найбільш точно, що дуже важливо для наступних стадій роботи системи стиснення. У пакет включені такі базисні набори вейвлетів: біортогональний, Хаара, «Мексиканська капелюх», Майєра та ін. Детальний опис пояснює принципи

роботи з методами пакета, а наведені приклади дозволяють отримати навички практичної роботи.

Image Processing Toolbox є набір функцій, які розширюють можливості числових обчислень у середовищі MATLAB. Програма підтримує різні операції обробки зображень, включаючи:

- просторові перетворення зображень;
- морфологічні операції;
- ковзну та блокову обробку;
- різні види лінійної та нелінійної фільтрації;
- аналіз та покращення зображень;
- відновлення зображень;
- видалення розмитостей;
- обробку галузі інтересу.

Багато функцій програми подаються в системі MATLAB у вигляді М-файлів. Вони реалізовані найбільш відомі алгоритми обробки зображень. Також є можливість перегляду програмних кодів функцій, які реалізує дані алгоритми.

type function_name

Усі М-файли програми Image Processing Toolbox мають однакові властивості і можуть використовуватися в комбінації з М-файлами інших програм, таких як, наприклад, Signal Processing Toolbox та Wavelet Toolbox.

Пакет Image Processing Toolbox працює з такими типами зображень:

- напівтонові зображення (зображення у градаціях сірого);
- двійкові (чорно-білі) зображення;
- кольорові RGB зображення.

Напівтонові зображення - це матриця, елементи якої представлені у вигляді числових значень класів uint8 [0,255], або uint16 [0,65535].

Бінарні (двійкові) зображення в MATLAB є матрицею логічних елементів 0 або 1. Для перетворення числових масивів у логічні служить функція logical - B=logical(A).

Повнокольорові зображення – такий тип зображень, де кожен піксель описується трьома значеннями червоної, синьої та зеленої складових. Система MATLAB запам'ятовує повнокольорові зображення як масиву даних з розмірністю $m \times n \times 3$. У цьому масиві зберігається кожна компонента кольору кожного окремого пікселя.

Графічний файловий формат запам'ятовує повнокольорові зображення як 24-бітові дані, де червона, зелена та синя компоненти представлені 8 бітами кожна. У сумі це дає близько 16 мільйонів кольорів.

Повнокольорові зображення можуть бути представлені у форматі `uint8`, `uint16`, `single` або `double`.

Зведемо основні дані різних типів зображень до таблиці 3.1.

Таблиця 3.1 – Порівняння основних типів зображень

| Тип зображення | Опис |
|--------------------------|--|
| Бінарне (Binary) | Логічний масив, що містить лише одиниці та нулі, які інтерпретуються як чорний та білий колір відповідно. Також існують так звані дворівневі зображення, пікселі якого містять лише два рівні інтенсивності, не обов'язково 1 і 0. |
| Індексне (Indexed) | Масив у форматі <code>logical</code> , <code>uint8</code> , <code>uint16</code> , <code>single</code> чи <code>double</code> , значення пікселів якого є індексами з палітри. Палітра є масивом з розмірністю $m \times 3$, який представлений у форматі <code>double</code> . Також відомі псевдокольорові зображення. <i>Примітка.</i> Для масивів, представлених у форматі <code>single</code> чи <code>double</code> , значення перебувають у діапазоні $[1, p]$. Для масивів, які представлені у форматі <code>logical</code> , <code>uint8</code> або <code>uint16</code> , діапазон значень становить $[0, p-1]$. |
| Напівтоновий (Grayscale) | Масив зображення у форматі <code>uint8</code> , <code>uint16</code> , <code>int16</code> , <code>single</code> або <code>double</code> . Значення пікселів описують значення інтенсивності зображення. Вони також відомі як яскраві зображення. <i>Примітка.</i> Для зображень у форматі <code>single</code> або <code>double</code> значення пікселів знаходяться в діапазоні $[0, 1]$. Якщо зображення представлені у форматі <code>uint8</code> , значення пікселів знаходяться в діапазоні $[0, 255]$. Якщо зображення представлені у форматі <code>uint16</code> , то значення пікселів знаходяться в діапазоні $[0, 65535]$. Для формату <code>int16</code> значення |

| Тип зображення | Опис |
|----------------------------|--|
| | можуть бути в діапазоні $[-32768, 32767]$. |
| Повнокольорове (Truecolor) | <p>Зображення представляють масив з розмірністю $m \times n \times 3$ у форматі <code>uint8</code>, <code>uint16</code>, <code>single</code> або <code>double</code>. Значення пікселя дорівнює значенню інтенсивності. Ці зображення відомі ще як зображення RGB.</p> <p><i>Примітка.</i> Для зображень у форматі <code>single</code> або <code>double</code> значення пікселів знаходяться у діапазоні $[0, 1]$. Якщо зображення представлені у форматі <code>uint8</code>, значення пікселів знаходяться в діапазоні $[0, 255]$. Якщо зображення представлені у форматі <code>uint16</code>, то значення пікселів знаходяться в діапазоні $[0, 65535]$.</p> |

У піксельній координатній системі піксель є дискретною областю, яка однозначно визначається парою координат, наприклад $(5, 2)$. З цього випливає, що позначення координат у вигляді $(5.3, 2.2)$ немає сенсу.

Однак у деяких випадках для деяких піксельних систем таке уявлення може бути виправданим. У системі просторових координат локалізація елементів зображення описується значеннями x і y (не r і s як піксельної системи координат).

На зображенні (рис. 3.1) продемонстровано просторову систему координат, яка використовується для представлення зображень. Зазначимо, що змінна y збільшується вниз.

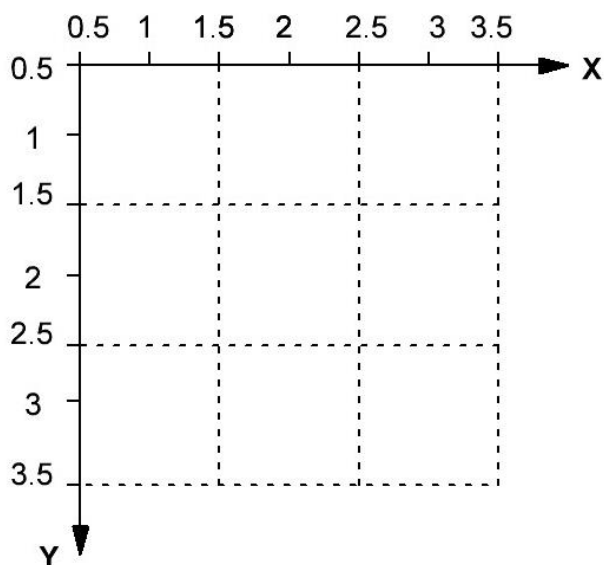


Рисунок 3.1 – Просторова система координат

Просторова система координат у деяких випадках відповідає піксельній координатній системі. Наприклад, у просторовій та піксельній системах координат центри збігаються для всіх пікселів зображення.

Зазначимо деякі суттєві відмінності. У піксельних координатах верхній лівий кут зображення має координати (1,1), а просторової системі координат ці індекси рівні відповідно (0.5,0.5). Ще одна істотна різниця між ними полягає в тому, що піксельна система координат є дискретною, а просторова система координат безперервна. Однак, у більшості випадків, якщо кажуть, що піксель у лівому верхньому куті приймає координати (1,1), то мають на увазі просторову систему координат.

Далі, якщо мова йде про піксельну систему координат, то будуть використовуватися індекси (r, c), а при роботі з просторовою системою координат використовуватимуться індекси (x, y).

Використання нестандартних просторових координатних систем. За умовчанням просторова система координат сприймається як піксельна. Наприклад, координати центрального пікселя в 5 рядку та 3 стовпці мають такі просторові координати $x=3$, $y=5$. (Запам'ятаємо, що порядок координат обернений.). Деякі функції переважно працюють із просторовими координатами, інші з піксельними.

Також у деяких випадках є потреба працювати з нестандартними системами просторових координат.

Для визначення просторової координатної системи необхідно описати дані зображення XData та Ydata. Ці властивості будуть описані у вигляді двоелементного вектора і будуть задавати діапазон координат зображення, що відображається. За умовчанням для зображення A XData є діапазоном [1 size(A,2)], а YData - [1 size(A,1)].

Наприклад, якщо зображення A складається з 100 рядків і 200 стовпців, то за умовчанням XData дорівнює [1200], а YData - [1100]. Значення цих векторів є координатами центрів пікселів. Насправді ж діапазон відображуваних координат трохи більший. Якщо XData дорівнює [1200], то діапазон вздовж x-осі зображення дорівнює [0.5 200.5].

Продемонструємо процедуру візуалізації зображення з використанням XData та YData (рис. 3.2).

```
A = magic(5);
```

```
x = [19.5 23.5];
```

```
y = [8.0 12.0];
```

```
image(A, 'XData',x, 'YData',y), axis image, colormap(jet(25))
```

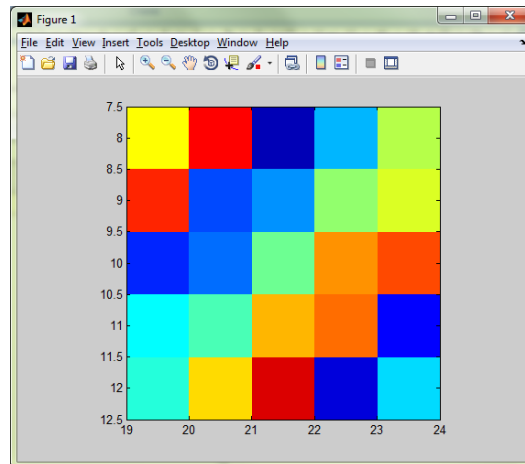


Рисунок 3.2 - Візуалізація зображення з використанням XData та YData

Для більш детальної інформації щодо синтаксису системи просторових координат див. опис функції `imshow`.

3.3 Проектування інтерфейсу та функціональних модулів

Тепер розглянемо питання візуалізації ліченого зображення. Додаток включає дві функції відображення зображень: `imshow` та `imtool`. Функція `imshow` є фундаментальною функцією відображення зображень. Функція `imtool` запускає інструментарій Image Tool, який є інтегрованим середовищем для візуалізації зображень і виконує деякі операції з обробки зображень. Засіб відображення зображень Image Tool включає низку додаткових функцій, які дають можливість надавати інформацію про конкретний піксель або зображення в цілому, підвищувати контраст зображення тощо.

Для візуалізації індексного зображення застосовуються функції `imshow` або `imtool`, які використовують опис матриці зображення та палітри. Для представлення індексного зображення в робочому просторі використовується

змінна під назвою `X` для представлення матриці індексного зображення і змінна `map` для представлення панелі `imshow(X,map)` або `imtool(X,map)`

Для кожного пікселя `X` функція використовує при відображенні той колір, який відповідає відповідному рядку `map`. Якщо матриця даних зображення представлена у форматі `double`, значення першого відліку відображається першим рядком `colormap`, значення другого відліку – другим рядком і т.д. Однак, якщо дані матриці зображень представлені у форматі `uint8` або `uint16`, то значення нульового відліку відображається першим рядком палітри, значення першого відліку – другим рядком і т.д. Це автоматично визначається функціями `imtool` та `imshow`.

Якщо кількість кольорів на панелі більша, ніж на зображенні, тоді решта кольорів палітри функція ігнорує. Якщо панель містить менше кольорів, ніж присутня на зображенні, функція встановлює всім пікселям зображення, значення яких вийшли за діапазон палітри, значення останнього кольору на панелі. Наприклад, при візуалізації зображення, яке представлено у форматі `uint8` та містить 256 кольорів, використовується палітра, яка містить 16 кольорів. У цьому випадку всі пікселі, значення яких більше 15 відобразатимуться останнім кольором палітри.

Для відображення напівтонових зображень використовується функція `imshow` або `imtool` з описом назви матриці зображення як аргумент. Далі будуть використовуватись різні імена (`I`) для представлення напівтонових зображень у робочому просторі.

`imshow(I)` або `imtool(I)`

Обидві функції відображають зображення з масштабованими значеннями інтенсивності, які представлені відповідними індексами на панелі.

Якщо параметр `I` представлений у форматі `double`, пікселі зі значенням 0.0 відображаються як чорні, пікселі зі значенням 1.0 відображаються як білі, а пікселі із проміжними значеннями відображаються різними відтінками сірого. Якщо матриця зображення `I` представлена у форматі `uint8`, то пікселі зі значеннями 255 відображаються білими. Коли матриця зображення `I`

представлена у форматі uint16, то пікселі зі значеннями 65535 також відображаються білими.

Напівтонові зображення, як і індексні, які використовують палітру RGB, у більшості випадків не потребують опису палітри. Система MATLAB відображає напівтонові зображення за допомогою напівтонової палітрової системи (де $R=G=B$). За замовчуванням, число відліків сірого на палітрі дорівнює 256 для системи з поданням 24 біти на піксель.

У системі MATLAB бінарні зображення надаються у форматі logical. Бінарні зображення містять пікселі зі значенням лише нуль та одиниця. Пікселі зі значенням 0 відображаються як чорні, а пікселі зі значенням 1 – як білі.

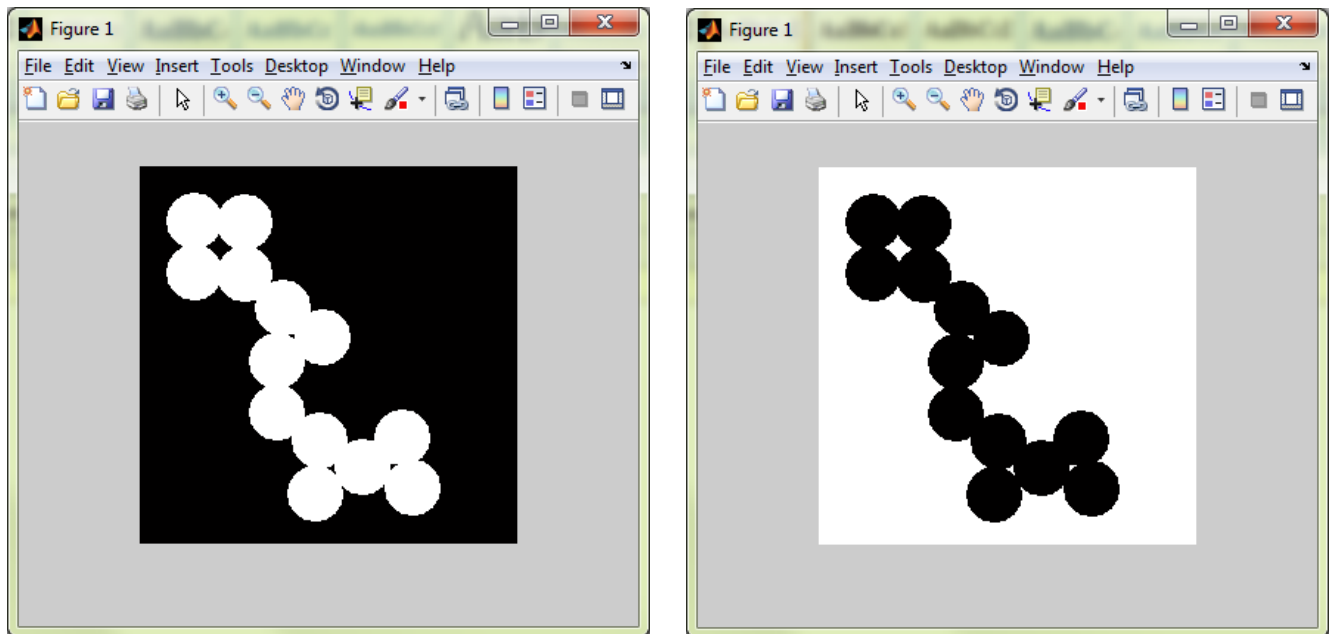
Для того, щоб програма інтерпретувала зображення як бінарне, необхідно, щоб це зображення було представлене у форматі logical.

Для візуалізації бінарних зображень використовується одна з функцій imshow або imtool з описом матриці зображення як аргумент. Розглянемо приклад зчитування бінарного зображення робочий простір MATLAB і наступну візуалізацію зображення. Далі використовуватимемо назву BW для представлення бінарних зображень у робочому просторі.

```
BW = imread('circles.png');  
imshow(BW) або imtool(BW)
```

У деяких випадках зручності бінарні зображення необхідно інвертувати, тобто. пікселі зі значенням 0 відобразатимуться як білі, а пікселі зі значенням 1 відобразатимуться як чорні. Для цього необхідно використовувати оператор NOT (~). Наприклад,
imshow(~BW) або imtool(~BW)

Приклад візуалізації прямого та інвертованого зображень наведено на рис. 3.3.



а б

Рисунок 3.3 - Бінарні зображення: а) пряме; б) інвертоване.

Також існує можливість відображення бінарного зображення за допомогою синтаксису палітри індексних зображень. Наприклад, розглянемо команду, в якій задається палітра, в результаті використання якої пікселі з нульовими значеннями відобразатимуться червоним кольором, а зі значенням 1 – синім кольором

```
(imshow(BW,[1 0 0; 0 0 1]) або imshow(BW,[1 0 0; 0 0 1])
```

Повнокольорові зображення, які називають RGB зображеннями, представляють значення кольорів безпосередньо, без використання палітри. Повнокольорові зображення є тривимірним масивом $m \times n \times 3$. Для кожного пікселя (r,c) на зображенні колір представляється триплетом $(r,c,1:3)$.

Під час візуалізації повнокольорових зображень використовуються функції `imshow` або `imshow` з описом матриці зображення як аргумент. Розглянемо приклад зчитування повнокольорового зображення у робочий простір MATLAB та його подальшу візуалізацію. Далі назву RGB будемо використовувати для представлення повнокольорових зображень у робочому просторі.

```
RGB = imread('C:\Users\krasnov.MEDICA\Desktop\6.jpg');
```

```
imshow(RGB)
```

або

imtool(RGB)

Система, яка використовує уявлення 24 біта на піксель, може відображати повнокольорові зображення прямо, тобто. по 8 біт на подання червоної, зеленої та синьої складової відповідно. У системах з меншою кількістю кольорів, функція `imshow` відображає зображення за допомогою методу моделювання кольорів (`dithering`).

Якщо при візуалізації кольорового зображення воно відображається як чорно-біле, тоді необхідно перевірити, чи це зображення є індексним. Для індексних зображень необхідно описати палітру та зв'язати її із зображенням. У практиці обробки медичних зображень часто виникають завдання перетворення повнокольорових зображень на напівтонові. Це може бути потрібне, наприклад, для проведення спектрального аналізу або інших завдань. Процедура перетворення виконується за допомогою функції `I=rgb2gray(RGB)`. Покажемо це з прикладу (рис. 3.4).

```
>> RGB=imread('C:\8.jpg');
```

```
>> imshow(RGB)
```

```
>> I = rgb2gray (RGB);
```

```
>> figure,imshow(I)
```

```
>> whos
```

```
Name Size Bytes Class Attributes
```

```
I 190x262 49780 uint8
```

```
RGB 190x262x3 149340 uint8
```

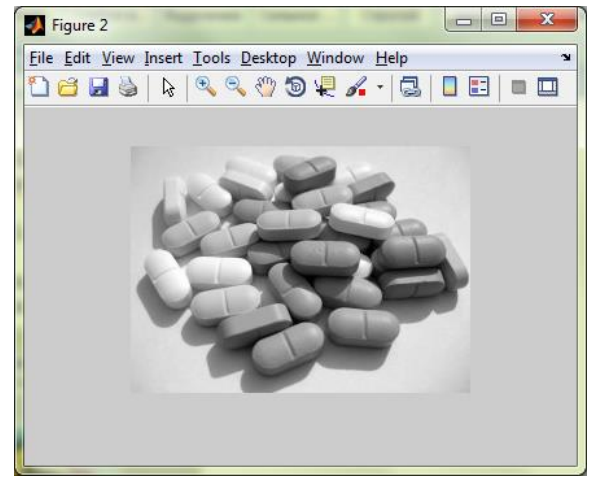
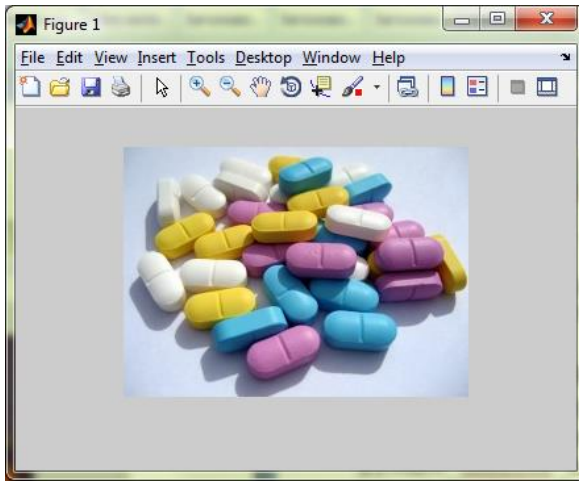
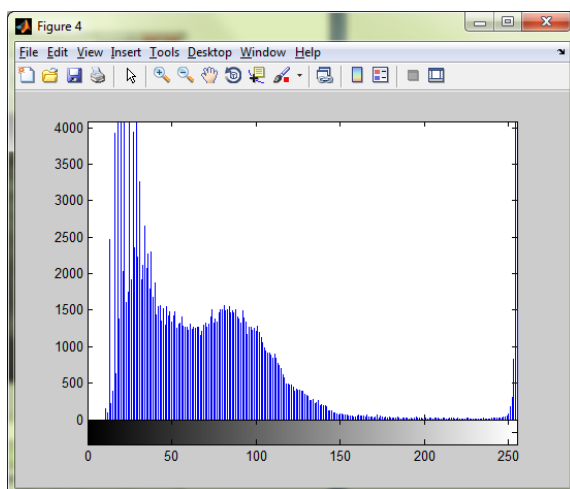



Рисунок 3.4 – Перетворення повнокольорового зображення на напівтонове

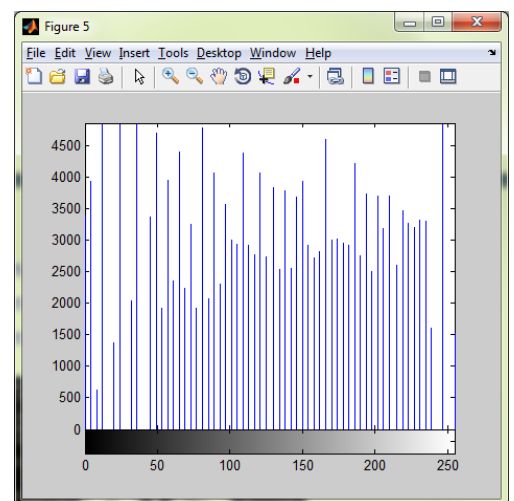
За допомогою команди `whos` нескладно помітити зміни у форматі даних після виконаного перетворення.

Покращення контрасту зображення, що обробляється. Функція `histeq` збільшує контраст зображення шляхом розтягування значень інтенсивності динамічного діапазону. Для отримання більш детальної інформації див. опис функції `histeq`. Відобразимо покращене зображення

```
>> I=imread('C:\129.jpg');      >> imshow(I)
>> I2=rgb2gray(I);            >> figure, imshow(I2)
>> I3 = histeq(I2);           >> figure, imshow(I3)
>> figure, imhist(I2)         >> figure, imhist(I3)
```



а)



б)

Рисунок 3.5 - Гістограми розподілу яскравості вихідного (а) та контрастованого (б) зображень

Функція `imhist` створює гістограму перетвореного зображення. Це дозволяє порівнювати гістограми до та після вирівнювання. Зазначимо, що діапазон інтенсивностей вихідного зображення досить вузьким (рис. 2.9. а). Він не покриває весь діапазон $[0, 255]$, що є однією з причин низької контрастності зображення.

Для запису зображення І файл на диск використовується функція `imwrite`. Якщо ця функція включає назву та розширення `'.png'`, то функція `imwrite` записує зображення у форматі PNG (Portable Network Graphics). Також існує можливість записати зображення в інших форматах, наприклад `tif` або `jpg`.

```
imwrite (I, 'rout2.png');
```

Список доступних форматів можна переглянути в описі функції `imwrite`.

4 ДОСЛІДНИЦЬКА ЧАСТИНА

4.1 Завдання та методи проведення досліджень

Для проведення досліджень розроблено конфігурацію БІНС з 2-ма інерційними блоками GY-87 (3-х осьовий гіроскоп, 3-х осьовий акселерометр, 3-х осьовий магнітометр, датчик тиску і датчик температури). Один із вимірювальних модулів розташований неортогонально щодо пов'язаної системи координат. Основу такого інерціального модуля становить МЕМС датчик MPU6050 (цифровий 3-осьовий акселерометр і 3-осьовий гіроскоп - ДУС).

Неортогонально розташований інерційний блок встановлений на бічній межі зрізаного тетраедра. Кінематична схема осей чутливості акселерометрів та ДУС (гіроскопів) у приладовій системі координат $Ox_1y_1z_1$ представлені на рис. 4.1.

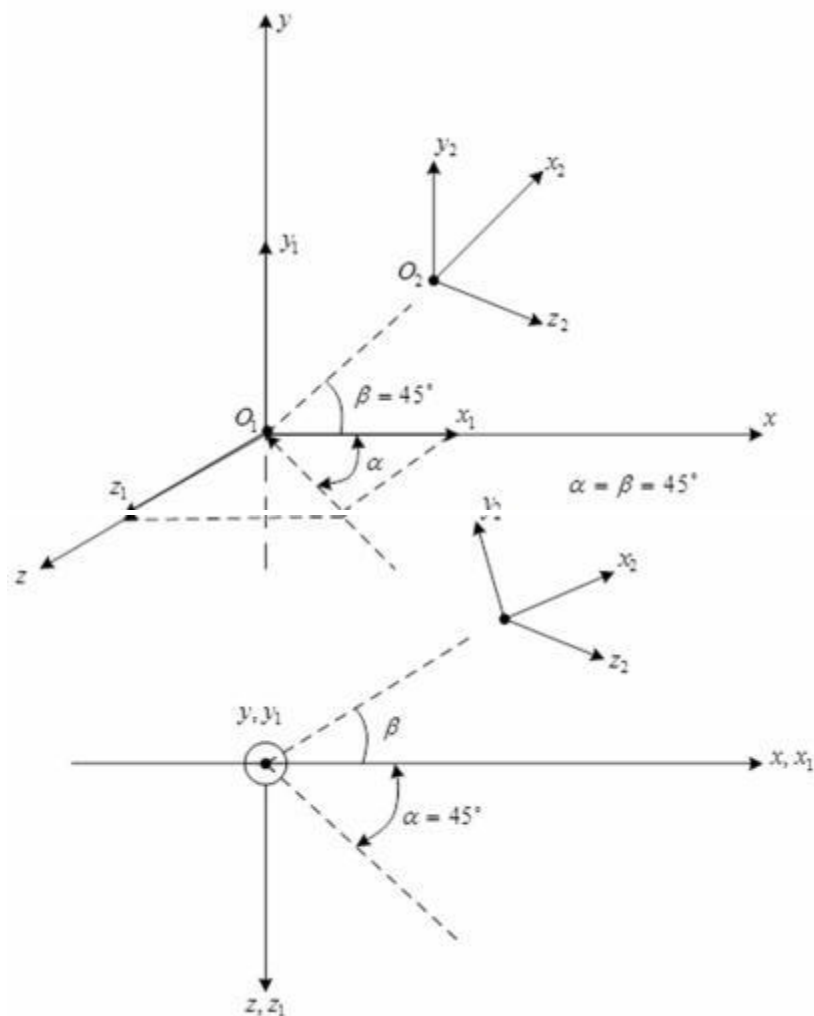


Рисунок 4.1 - Кінематична схема розташування осей чутливості вимірювачів БІНС

4.2 Програма проведення досліджень

Структурну схему експериментальної БІНС представлено на рис. 5.2.

Виходячи із завдань розв'язуваних БІНС в експериментальні установці, на мікроконтролер покладаються функції організації отримання вимірювальної інформації з датчиків (ДУСів, акселерометрів, магнітометрів, датчиків тиску) та її запис на micro SD card для подальшої обробки на комп'ютері.

На ПК покладаються завдання моделювання ідеальної роботи БІНС, розв'язання навігаційної задачі за даними різних вимірювачів, обчислення параметрів орієнтації за показаннями вимірювального блоку, порівняння результатів розв'язання навігаційної задачі з моделлю оцінки отриманих результатів. Крім того, на ПК відпрацьовуються питання пов'язані з дослідженням алгоритмів обробки (у тому числі і попередньої фільтрації) вимірювальної інформації.

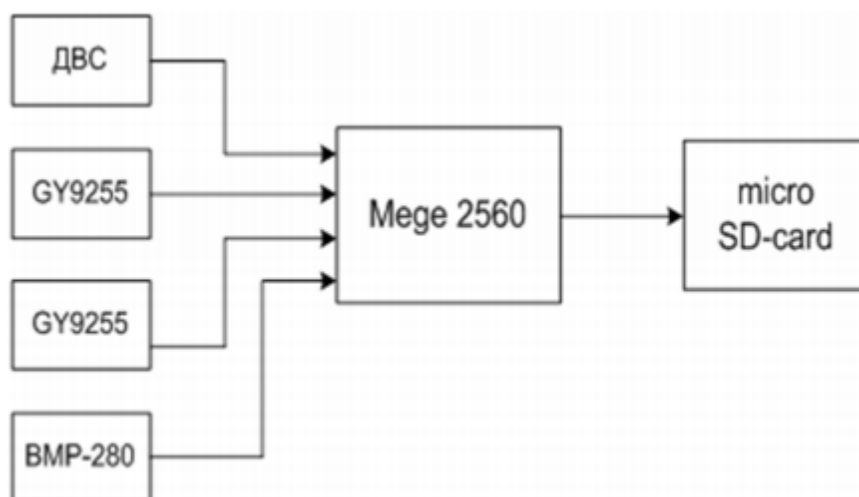


Рисунок 4.2 - Структурна схема експериментальної БІНС

4.3 Результати досліджень

Використання вимірювально модуля з ортогональним розташуванням осей чутливих елементів можливість оцінки точності результатів роботи БІНС.

Зовнішній вигляд експериментальної БІНС представлено на рис. 4.3.

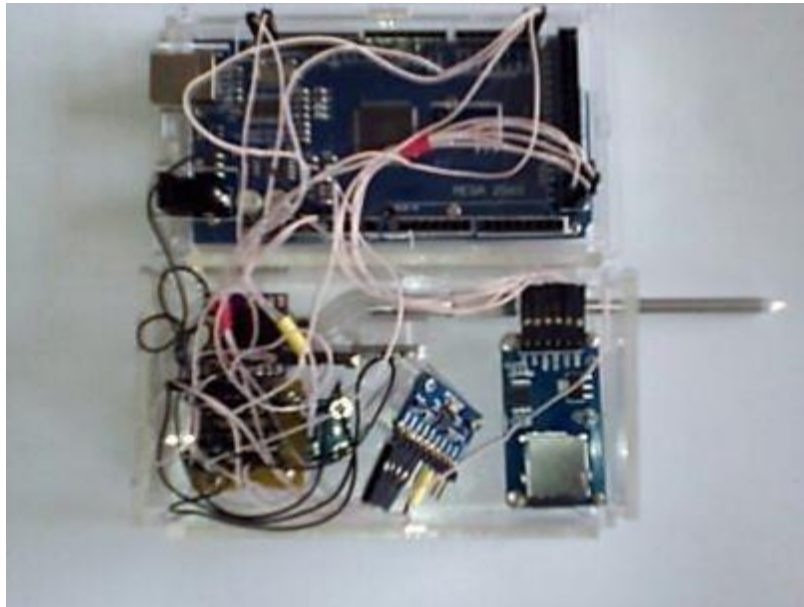


Рисунок 4.3 – Зовнішній вигляд експериментальної БІНС

Склад експериментальної БІНС:

1. GY9255 – вимірювальний модуль (3-х осьовий акселерометр + 3-х осьовий ДУС + 3-х осьовий магнітометр; тип підключення: шина I2C

Розрядність даних: 16 біт АЦП, 16 біт дані

Діапазони гіроскопа: $\pm 250/500/1000/2000^\circ/\text{сек}$

2. ДВС – модуль MPXV 7002 DR – датчик диференціального тиску + Трубка Пітто. Макс. допустимий тиск 8 кПа; чутливість 1 мВ/кПа; похибка 2,5%;

3. BMP280 – датчик атмосферного тиску;

4. MicroSD – об'єм пам'яті 18 Гб

5. Мікроконтролер ATmega2560:

Тактова частота: 16 МГц;

Портів введення-виведення загального призначення: 54;

Розрядність АЦП: 10біт;

Flash-пам'ять: 256 Кб;

EEPROM-пам'ять: 4 Кб.

4.5 Висновки за розділом

За експериментальними вихідними даними вимірювального модуля отримані моделі акселерометра та гіроскопа. Тому розроблена модель БСО відповідає характеристикам використовуваних MEMS датчиків.

Розроблено та виготовлено модель БНС з надлишковим числом вимірювачів за рахунок неортогонального розташування вимірювального модуля

Розроблено модель БНС у середовищі Matlab Simulink. Це дає можливість проводити моделювання різних варіантів алгоритмів розв'язання навігаційної задачі (з використанням кутів Ейлера-Крилова, напрямних косинусів, рішення узагальненого рівняння Пуассона та з використанням параметрів Родріга-Гамільтона).

5 ЕКСПЕРИМЕНТАЛЬНО-ПРАКТИЧНА ЧАСТИНА

5.1 Завдання та засоби виконання експериментальної розробки

Метою експериментального дослідження є натурне підтвердження розроблених математичних моделей визначення кутового становища ЛА. З цією метою проводився експеримент, що полягає в напівнатурній імітації кутових еволюцій ЛА та визначенні кутового положення ЛА за даними оптичних відеодатчиків.

Експеримент включав проведення двох етапів.

Перший етап – це вимірювання кутового положення ЛА за даними відеоданим оптичного датчика закріпленого на борту безпілотного літального апарату.

Другий етап полягав у визначенні кутового положення безпілотного літального апарату за даними відеодатчика, розташованого в прямій видимості ЛА.

5.2 Опис лабораторної установки

Розглянутий безпілотний літальний апарат є високопланом, побудованим за нормальною двокільовою аеродинамічною схемою, з нерухомим відносно корпусу крилом. Він призначений для цілеспрямованого керованого руху в межах атмосфери на різних, у тому числі і малих, висотах польоту на основі використання аеродинамічного принципу польоту та створення аеродинамічних моментів, що управляють.

Наведемо коротку характеристику об'єкта, що аналізується.

Для проведення експерименту модель була закріплена на керованому тривимірному підвісі (рис. 5.1).

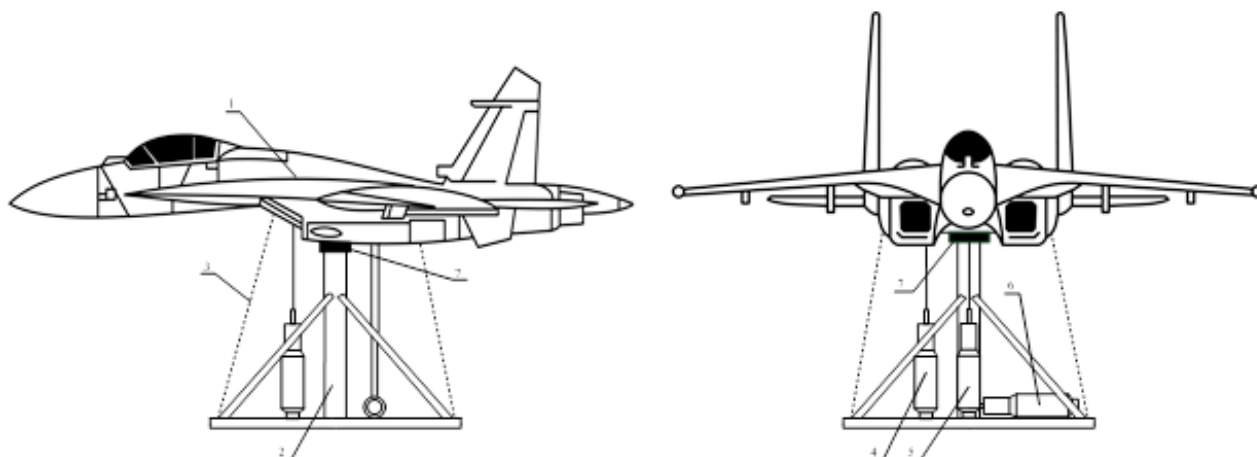


Рисунок 5.1 – Кінематична схема тривимірного підвісу стенду

Підвіс складається з металевого каркаса (2), на якому закріплені БПЛА (1). Нейтральне положення об'єкта управління забезпечується завдяки балансувальним пружинам (3), зміна кутів тангажу, крену та нищпорення моделі відбувається за допомогою приводів РАУ-107А (4,5,6). Також на підвісі розташований блок потенціометричних датчиків (7), який дозволяє забезпечити зворотний зв'язок по кутовому положенню БПЛА щодо трьох осей координат.

Вимірювальний блок на борту БПЛА складається з гіроскопічних датчиків кутів (ДУ) курсу, крену та тангажу, датчиків кутових швидкостей (ДУС) (по два датчики, щодо кожної з осей пов'язаної з об'єктом системи координат) і розташований у центрі мас літального апарату.

Для збору та перетворення до цифрового вигляду вихідних сигналів аналогових датчиків служить модуль аналого-цифрового перетворення.

Основний контролер здійснює обмін даними між моделлю та персональним комп'ютером-сервером та призначений для управління рульовими агрегатами та бортовими сервоприводами, розташованими на підвісі. Він виступає у ролі обчислювального пристрою та задає закон кутового руху БПЛА.

Передбачено двосторонній обмін даними між персональним комп'ютером-сервером та комп'ютерами-клієнтами по каналу бездротового зв'язку,

організованого за допомогою відповідної системи. Клієнти одержують дані з датчиків у реальному масштабі часу.

Ручне керування положенням моделі здійснюється за допомогою джойстика, підключеного до комп'ютера-сервера.

5.3 Характеристика розробленого програмного забезпечення

Для зчитування зображень використовується функція `imread`. Покажемо на прикладі зчитування одного зображення `rout.tif`, яке включене до програми Image Processing Toolbox, та його запам'ятовування у вигляді масиву `I`.

```
I = imread('rout.tif');
```

Функція `imread` зчитує дані з графічного формату, представленого як TIFF (Tagged Image File Format). Список всіх форматів, що підтримуються, можна знайти в описі функції `imread`. Деякі найбільш відомі формати, що підтримуються в MATLAB, наведені нижче:

- BMP Windows Bitmap
- GIF Graphics Interchange Format
- HDF Hierarchical Data Format
- JPEG, JPG Joint Photographic Experts Group
- PBM Portable Bitmap
- PCX Paintbrush
- PGM Portable Graymap
- PNG Portable Network Graphics
- TIFF, TIF Tagged Image File Format

Для зчитування зображення, розташованого в певній конкретній папці, потрібно в явному вигляді вказати повний шлях до цієї директорії, наприклад:

```
F=imread('C:\Users\krasnov.MEDICA\Desktop\MED_IMAGE\phono.jpg');
```

У першому прикладі функція `imread` зчитує дані із графічного формату TIFF, у другому – JPEG.

Функція `imread` дозволяє зчитувати зображення з графічних файлів різних форматів та з різною глибиною кольору (кількість біт на піксель у кожній колірній компоненті). Більшість форматів використовує для запам'ятовування пікселів 8 біт на піксель.

Якщо ці дані зчитуються на згадку, то система MATLAB запам'ятовує їх у форматі `uint8`.

Для файлових форматів PNG і TIFF, які підтримують 16-бітові дані, система MATLAB запам'ятовує зображення у форматі `uint16`.

Для індексних зображень функція `imread` завжди зчитує палітру `colormap` у матрицю, яка представлена у форматі `double`. Наприклад, уявімо код, який зчитує зображення в робочий простір MATLAB у форматі RGB.

```
RGB = imread('football.jpg');
```

За допомогою цього коду зчитується індексне зображення з відповідною палітрою `colormap` у робочий простір MATLAB у вигляді двох окремих змінних.

```
[X,map] = imread('trees.tif');
```

У цьому прикладі функція `imread` при зчитуванні використовує формат, вказаний у змісті файлу. Також формат можна вказати у функції `imread` як аргумент. Для отримання іншої інформації, наприклад, щодо глибини кольоровості, див. опис функцій `imread` і `imformats`.

Функція `imread` передає дані зображення робочий простір. У робочому просторі може відображатись інформація про всіх змінних, які були створені в MATLAB протягом одного сеансу роботи. Функція `imread` повертає дані зображення змінної `I`, яка є масивом у форматі `uint8` з розмірністю 291x240 елементів. Система MATLAB може відображати також масиви та інших форматів – `uint8`, `uint16` або `double`.

Існує можливість отримання інформації про всіх змінних, які знаходяться у робочому просторі. Для цього використовується команда `whos`. Наприклад:

```
>> RGB = imread('football.jpg');
```

```
>> whos
```

Name Size Bytes Class Attributes

BW 256x256 65536 logical

RGB 256x320x3 245760 uint8

Повний текст програми обробки відео представлений у додатку А.

5.4 Результати експериментальних досліджень

Відеокамера, що стежить, розміщується так, що в кадрі відеозапису знаходиться сама модель МБПЛА. На моделі виділено 2 характерні точки зеленого та червоного кольору, що імітують посадкові вогні МБПЛА (рис.5.2).

Потім модель здійснювала вільні кутові еволюції у просторі щодо власного центру мас. Під час експерименту фіксувався відеозапис кутових переміщень моделі МБПЛА.

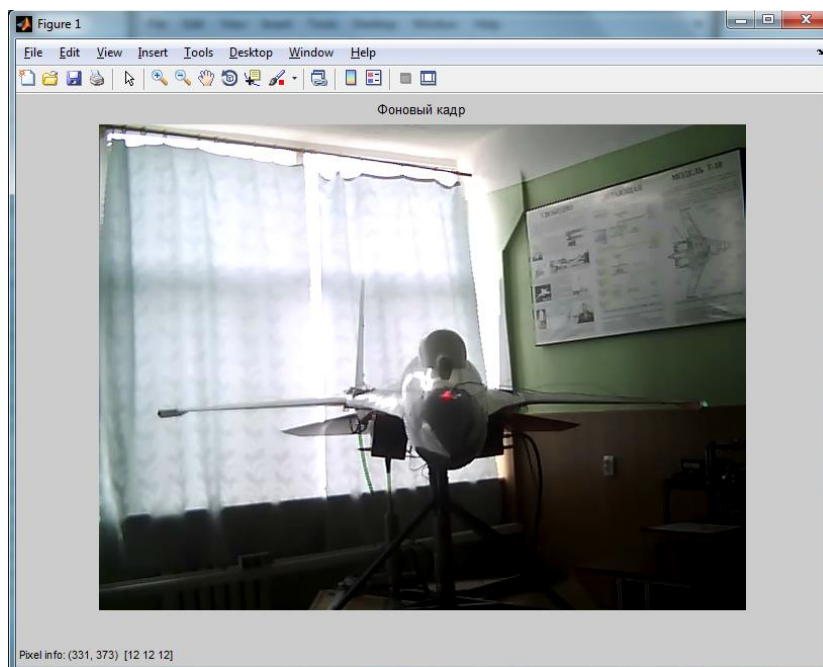


Рисунок 5.2 – Схема проведення експерименту із відеокамерою, що слідкує.

Далі відеозапис було опрацьовано за допомогою програми, що реалізує алгоритм, описаний у п.3.1 з метою визначення кутів положення МБПЛА. Результат оцінки кута крену представлений на рис.5.2.

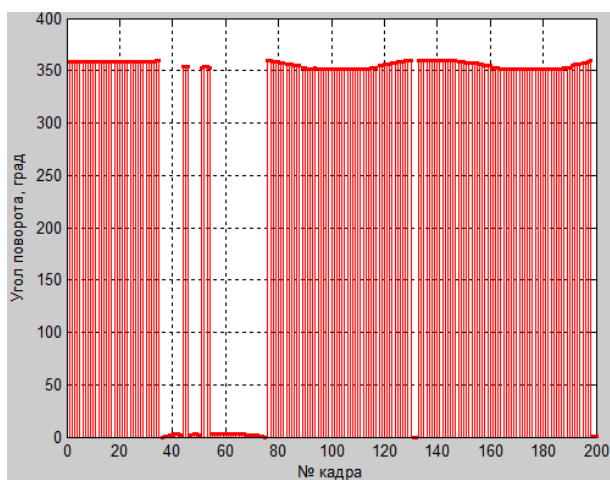


Рисунок 5.2 -Результат визначення кута крену МБПЛА в процесі еволюцій за відеоданим запису польоту

На експериментальну модель нерухомо в точці центру мас була закріплена відеокамера, так що в кадрі були характерні точки, оскільки показано на рис.5.3.

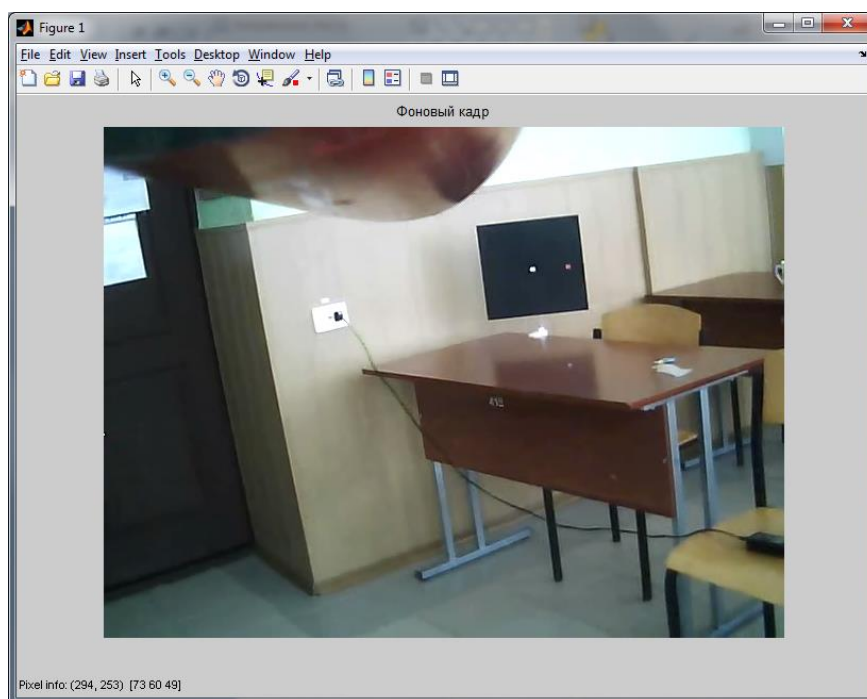


Рисунок 5.3 – Спостереження опорних точок визначення кутового переміщення МБПЛА

Відповідно до програми експерименту МБПЛА здійснював еволюції у просторі – змінював своє кутове становище. Метою експерименту було визначення кутового положення МБПЛА щодо переміщення опорних точок у кадрі відеокамери. В результаті оцінювалося кутове положення МБПЛА за даними відеовимірювань, за допомогою алгоритму, описаного в п.3.1.

Результати розв'язання задачі – визначення кута нахилу залежно від номера відеокадра представлені на рис. 5.4.

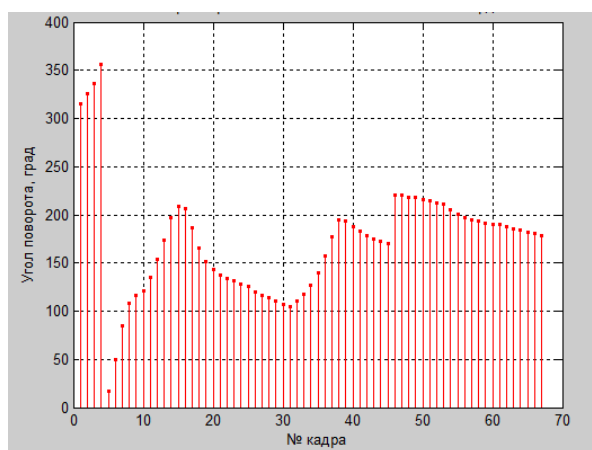


Рисунок 5.4 -Результат визначення кута крену МБПЛА в процесі еволюцій за відеоданим запису польоту

Аналогічно визначалися значення кутів тангажу та крену МБПЛА.

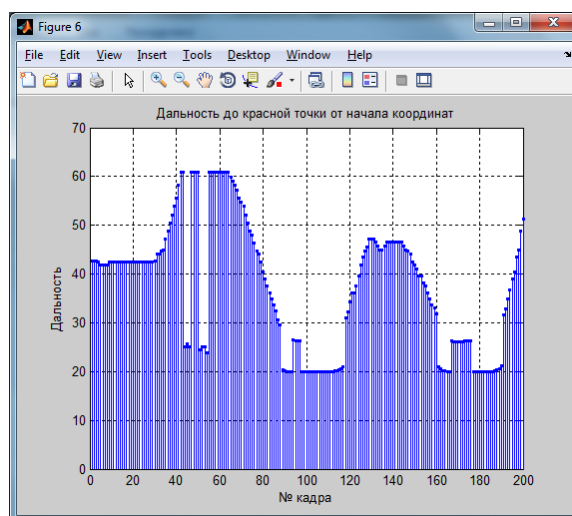


Рисунок 5.5 -Результат визначення кута крену МБПЛА в процесі еволюцій за відеоданим запису польоту

Інші опорні характеристики подано на графіках 5.6–5.7.

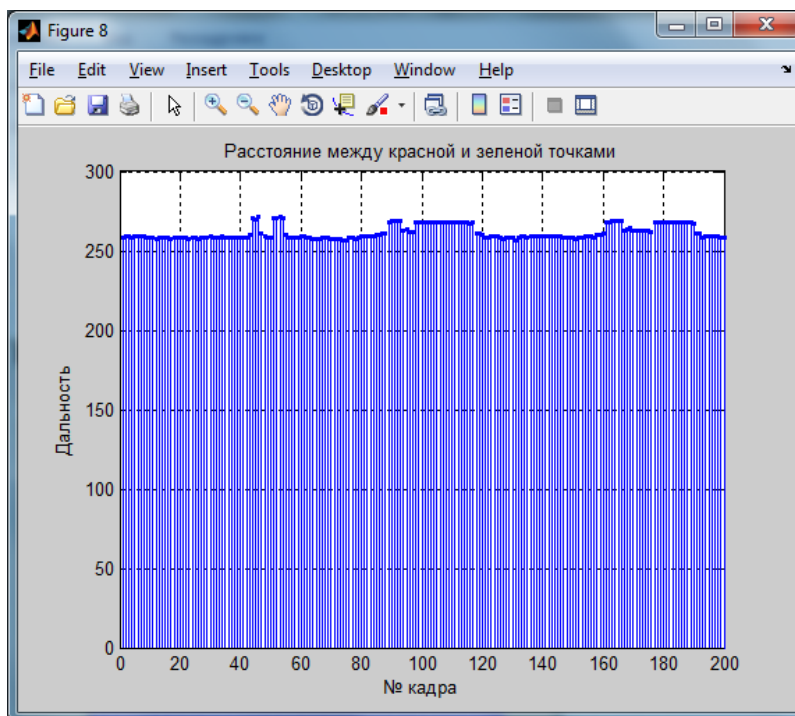


Рисунок 5.6 – Процедура видалення опорних точок від початку координат прототипу літака СУ-27

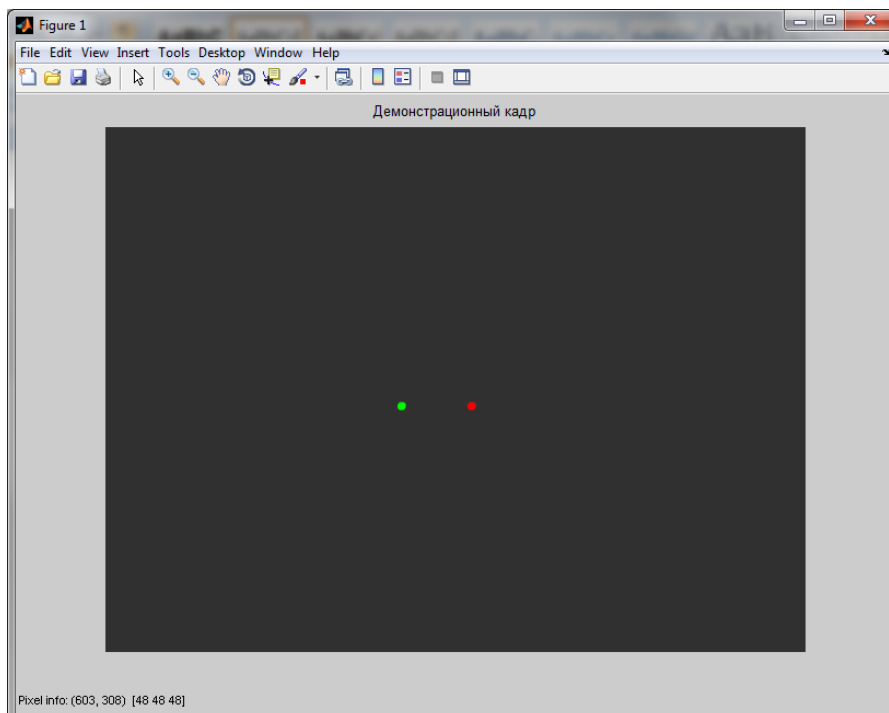


Рисунок 5.7 – Візуалізація розміщення опорних точок

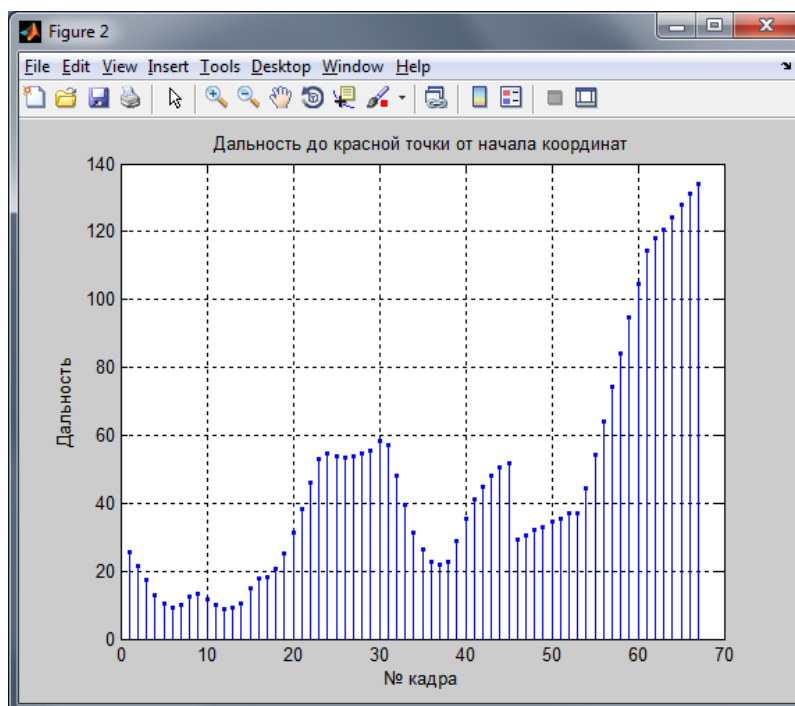


Рисунок 5.8 – Процедура видалення опорних точок від початку координат прототипу літака СУ-27 (дослід1)

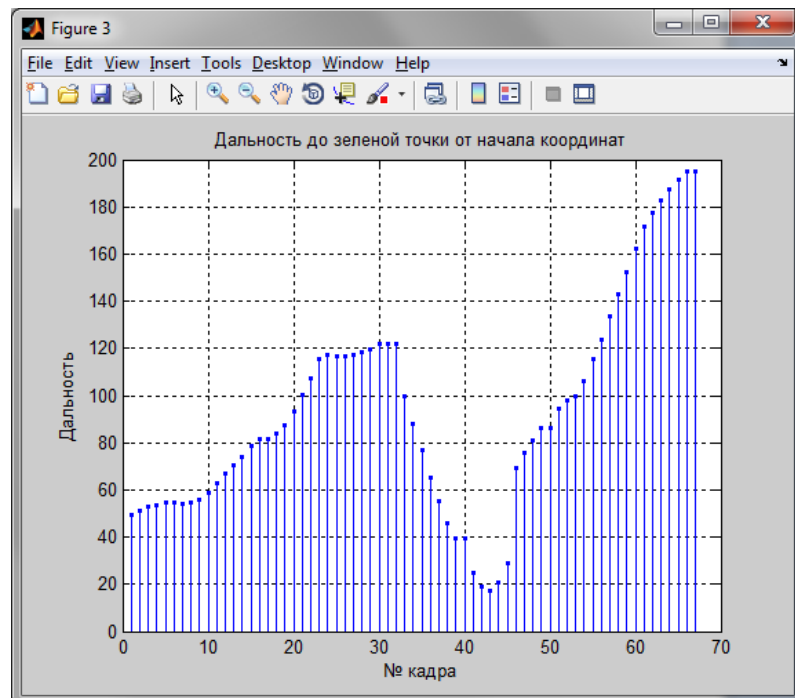


Рисунок 5.9 - Процедура видалення опорних точок від початку координат прототипу літака СУ-27 (дослід2)

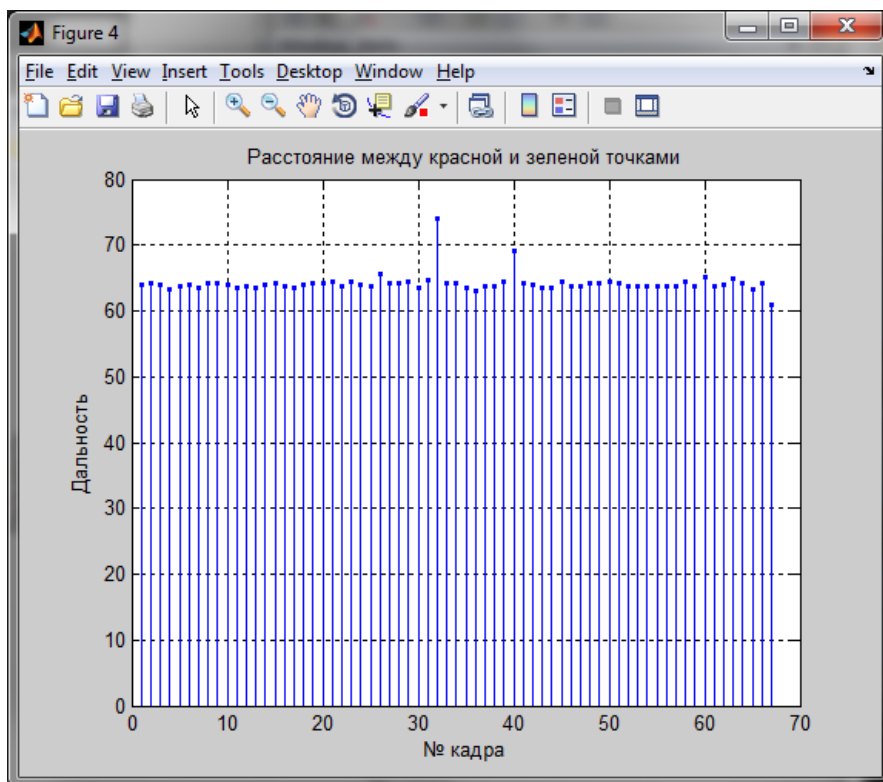


Рисунок 5.9 - Процедура видалення опорних точок від початку координат прототипу літака СУ-27 (дослід 3)

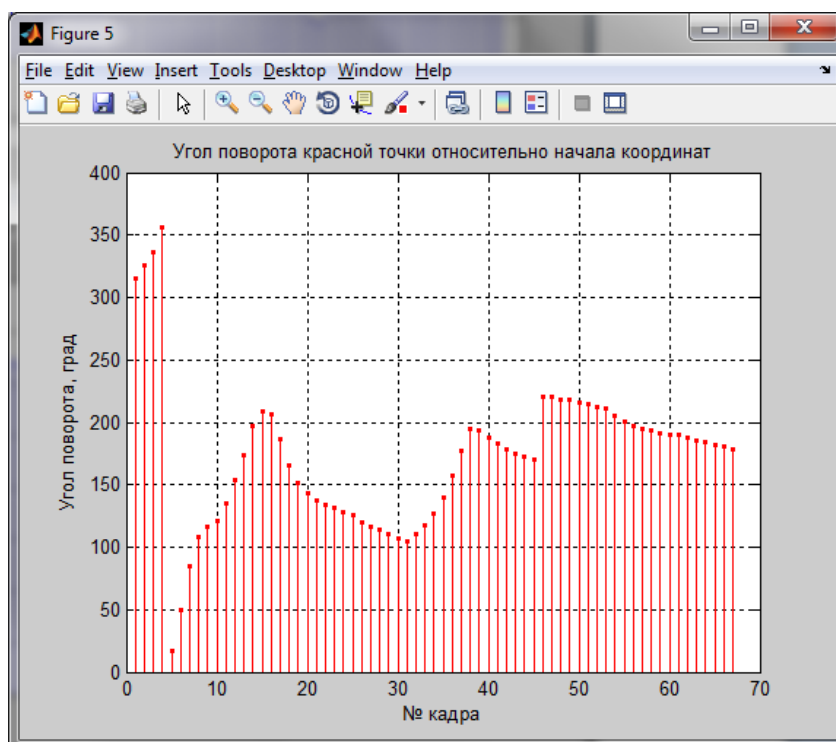


Рисунок 5.10 – Процедура видалення опорних точок від початку координат прототипу літака СУ-27 (дослід 4)

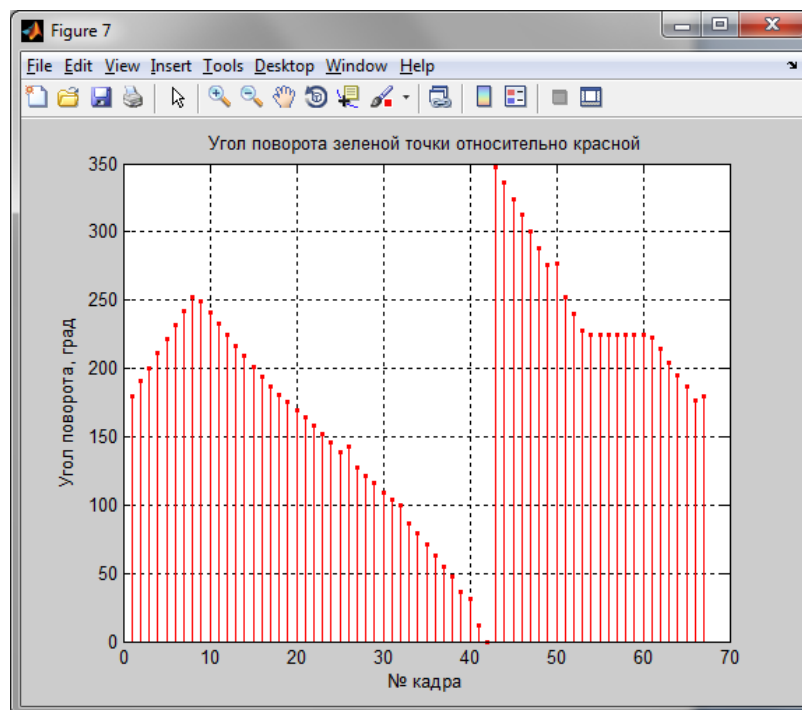


Рисунок 5.11 – Процедура видалення опорних точок від початку координат прототипу літака СУ-27 (дослід 5)

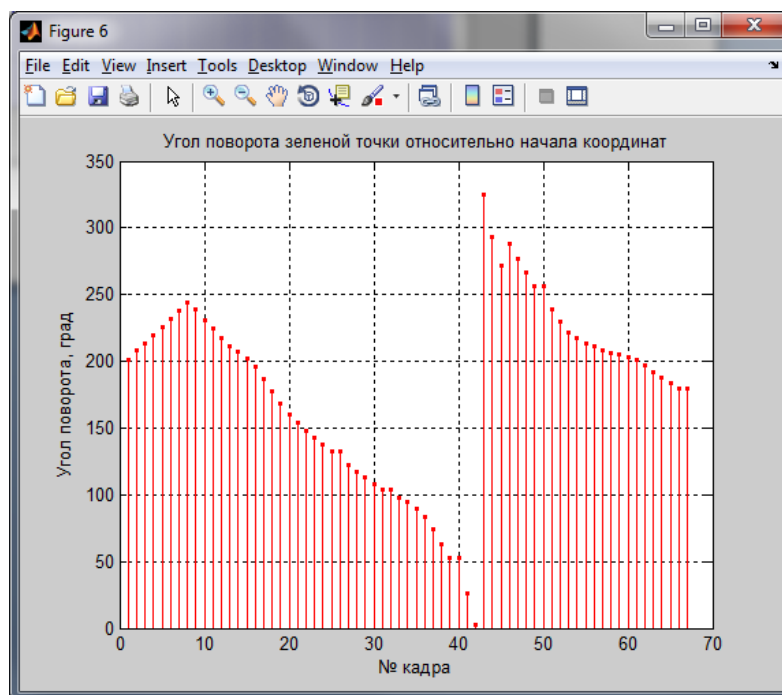


Рисунок 5.12 – Процедура видалення опорних точок від початку координат прототипу літака СУ-27 (дослід 6)

Таким чином, розроблений алгоритм продемонстрував у напів-натурній імітації кутових еволюцій ЛА та визначенні кутового положення ЛА за даними оптичних відео-датчиків роботу спроможність.

5.4 Висновки за розділом

У розділі було запропоновано моделі та методи оцінки кутового положення МБПЛА за даними відеозйомки, які можуть бути застосовані в реальному часі та реалізовані у вигляді програмного забезпечення навігаційного обчислювача у додатковому каналі діагностування працездатності систем кутової орієнтації МПВЛА.

Проведені дослідження показують доцільність застосовуваного підходу та аргументують вірність математичних припущень, що відкриває можливість побудови нових оптичних систем навігації, призначених для реалізації у спецобчислювачах другого рівня.

6 ЕКОНОМІЧНА ЧАСТИНА

6.1 Опис програмного продукту

Програмний продукт є дуже корисним інструментом для споживачів авіаційної промисловості, тому що він дозволяє дізнаватися кутове положення ЛА, на якому закріплена відеокамера на підставі відео ряду, який вона знімає. Застосування оптичного зору є перспективним напрямом авіаційної промисловості. Вказана розробка може використовуватися для діагностування втрати працездатності системи визначення кутового положення МБПЛА шляхом реалізації додатково вимірювального каналу, що використовує технології «машинного» зору.

Система має необхідні компоненти її повного функціонування. Дані виводяться на наземну станцію, саме на екран споживача.

Для нормального функціонування системи необхідне виконання наступних факторів та засобів:

- комп'ютер IBM PC сумісний;
- процесор Intel Pentium 1000Гц та вище;
- оперативної пам'яті щонайменше 512 Мбайт;
- розмір вільного місця на твердому диску 50 Мбайт;
- наявність пристроїв для введення інформації (миша, клавіатура);
- операційна система Windows XP, Vista, 7, 8.

6.2 Сегментація ринку

Сегментація ринку – це процес поділу споживачів на групи з урахуванням різних принципів та факторів сегментації. У процесі сегментації виявляються сегменти ринку. Сегмент – це група споживачів, які однаково реагують на товар. Сегмент ринку оцінюють рядом характеристик (описують профіль сегмента): число

можливих споживачів, ємність ринку, можливі темпи зростання за роками, споживчі ціни, прибутковість продажу тощо.

Загальний порядок сегментування ринку можна так:

- 1) виявлення основних принципів і чинників сегментування для виробу (насамперед розглядаються географічні чинники, та був демографічні та інші);
- 2) сегментування ринку: визначення складу груп споживачів, розподіл ринку на сегменти, опис профілю кожного сегмента, розрахунки річної ємності сегмента та всього ринку;
- 3) оформлення результатів сегментування як таблиць;
- 4) аналіз інформації про сегменти, вибір сегментів для подальшого аналізу (сегменти з незначною ємністю можна далі не розглядати);
- 5) позиціонування товару у вибраних сегментах, визначення цільової ємності.

Споживачами системи, що розробляється, є різні авіаційні підприємства України з виробництва БПЛА, а також Міністерство оборони України.

Основною характеристикою сегмента є ємність – кількість копій програмного забезпечення, які можуть бути продані за рік. ПЗ тісно пов'язане з конструктивною особливістю БПЛА і може бути встановлено тільки для машин з передбаченими конструктивними особливостями, що дуже знижує кількість копій, що продаються.

Розрахунки ємності ринку виконуються після визначення складу сегментів і починають із визначення повної потреби в товарі:

$$S_{\text{полн}} = \sum_{i=1}^L S_{1_{\text{полн}_i}}, \quad (6.1)$$

де повна потреба у товарі, шт/рік; $S_{\text{полн}}$ –

повна потреба одного сегмента, шт/рік; $S_{1_{\text{полн}_i}}$ –

i – номер сегмента;

L – кількість сегментів, прим.

Повна потреба сегмента розраховується з урахуванням конкретних особливостей товару та сегментів. Для багатьох видів товарів індивідуального та промислового призначення можна розрахувати за формулою: $S_{1_{\text{полн}_i}}$

$$S_{1_{\text{полн}_i}} = N_1 \cdot Q_1 \cdot m_1, \quad (6.2)$$

де кількість підприємств споживачів виробу в i -му сегменті; N_1 – середня річна програма виробів в i -му сегменті, для яких поставлятиметься товар; Q_1 – кількість комплектуючих виробів, що йдуть в один виріб – споживач (1 шт.). m_1 –

Сегментування та розрахунок ємності ринку представлений у табл. 6.1.

Таблиця 6.1 – Сегментування та розрахунок повної ємності ринку

| | Завод виробник | | | |
|------------------------------|----------------|--------|--------|-----------|
| | N1 | Q1, шт | m1, шт | S1 шт/рік |
| Міністерство оборони України | 1 | 5 | 1 | 5 |
| МНС України | 1 | 4 | 1 | 4 |
| ДП «АНТОНОВ» | 1 | 3 | 1 | 3 |
| ДП ХМЗ «ФЕД» | 1 | 2 | 1 | 2 |
| ХДАВП | 1 | 2 | 1 | 2 |
| Разом | 5 | 16 | 1 | 16 |

З таблиці 6.1 видно, повна ємність ринку становить 16 прим/рік.

6.3 Розрахунок собівартості та ціни виготовлення програмного забезпечення

Для коректної експлуатації програмного продукту та керування всіма необхідними роботами потрібен керівник.

Для зміни та коригування роботи програмного продукту та його підсистем слід враховувати наявність програміст-аналітика.

Тривалість робочих днів на місяць у середньому становить 23 дні. Склад виконавців показаний у таблиці 6.2. Розрахунок трудомісткості робіт показаний у таблиці 6.3.

Таблиця 6.2 - Склад виконавців роботи

| Посади | Посадові оклади, грн. | |
|------------|-----------------------|-------|
| | Місячні | Денні |
| Керівник | 7360 | 320 |
| Програміст | 8050 | 350 |

Таблиця 6.3 - Розрахунок трудомісткості робіт

| Вид робіт | Тривалість днів | Трудомісткість, чол/дн. | Виконавець | |
|---|-----------------|-------------------------|--------------|------------|
| | | | Руководитель | Програміст |
| Технічне завдання (ТЗ) | | | | |
| Постановка задачі | 1 | 1 | + | |
| Вибір платформи розробки ПЗ | 1 | 2 | + | + |
| Розробка графіка ходу робіт | 1 | 1 | + | |
| Визначення вимог до ПЗ | 1 | 2 | + | + |
| Розробка ТЗ | 3 | 3 | + | |
| Погодження та затвердження ТЗ | 3 | 3 | + | |
| Проектування методичного забезпечення | | | | |
| Розробка інтерфейсу | 2 | 4 | + | + |
| Розробка зв'язків між елементами інтерфейсу | 1 | 2 | + | + |

Продовження таблиці 6.3

| Вид робіт | Тривалість днів | Трудомісткіст ь, чол/дн. | Виконавець | |
|--|--------------------|-----------------------------|-------------------|----------------|
| | | | Руково- дитель | Програмі ст |
| Визначення методів випробувань системи | 17 | 17 | | + |
| Тестування програмного забезпечення | 8 | 8 | | + |
| Впровадження | | | | |
| Налагодження та випробування продукту | 3 | 6 | + | + |
| Додаткове налагодження, коригування | 2 | 4 | + | + |
| Випробування та здавання продукту в експлуатацію | 1 | 2 | + | + |
| Разом | 44 | 55 | 19 | 36 |

Отже загальна тривалість робіт становить 44 днів, загальна трудомісткість 55 чол/дней. Трудомісткість керівника – 19 днів, програміста-аналітика – 36 днів.

Обчислимо основну заробітну плату розробників ПЗ для визначення кутового положення ЛА з відеозображення, з урахуванням трудовитрат, кількості виконавців та середньоденної заробітної плати. Для цього кількість днів, відпрацьованих окремими виконавцями, помножимо на їх оклади:

$$\text{ОЗП} = \sum_{i=1}^s (N_i \cdot \text{ЗП}), \quad (6.3)$$

де кількість днів, відпрацьованих і-м виконавцем, дн. N_i –

ЗП – денні оклади і-х виконавців, грн.

Таким чином, основна заробітна плата становить:

$$\text{ОЗП} = (32019) + (35036) = 18680 \text{ (грн.)} \cdot \cdot$$

Проведемо розрахунок додаткової заробітної плати, яка становить 15% від основної заробітної плати:

$$\text{ДЗП} = 15\% \text{ ОЗП} = 0.15 \cdot 18680.00 = 2802 \text{ (грн.)} \cdot \cdot$$

Проведемо розрахунок єдиного соціального внеску:

$$\text{ЄСВ} = 22\% (\text{ОЗП} + \text{ДЗП}) = 0.22 (18680 + 2802) = 4726.04 \text{ (грн.)} \cdot + \cdot$$

Розрахуємо вартість матеріалів та комплектуючих, необхідні написання програми таблиці 6.4. Ціни вказані у гривнях.

Таблиця 6.4. Матеріали та комплектуючі

| Матеріали | Кількість | Ціна, грн. | Сума грн. | Призначення |
|---------------------------------|-----------|------------|-----------|--|
| Накопичувач (флешка) 4 Гб | 1 шт. | 110 | 110 | Зберігання резервних копій, перенесення програми |

Продовження таблиці 6.4

| Матеріали | Кількість | Ціна, грн. | Сума грн. | Призначення |
|-----------------------|-----------|------------|-----------|--------------------|
| Папір (пачка, 500 л.) | 1 шт. | 90 | 90 | Документація, друк |
| Друк документації | 250 л. | 0,75 | 112.5 | Різний друк |
| Разом | | | 312.5 | |

Також розрахуємо амортизацію (A_m), виходячи з вартості основних засобів.

$$A_m = \frac{25\% \cdot \text{стоимость ОС} \cdot \text{длительность разработки}}{12 \cdot 23} \quad (6.4)$$

Вартість основних засобів наведена у таблиці 6.5.

Таблиця 6.5 - Основні засоби

| Елемент | Кількість | Ціна, грн. | Сума грн. |
|-------------|------------|------------|-----------|
| Ноутбук | 2 шт. | 4499.00 | 8998.00 |
| Стілець | 2 шт. | 560.00 | 1120.00 |
| Стіл | 2 шт. | 779.00 | 1558.00 |
| Наст. лампа | 2 шт. | 436.00 | 872.00 |
| Разом | 12548 грн. | | |

Отже:

$$A_m = \frac{0.25 \cdot 12548 \cdot 44}{12 \cdot 23} = 500.10 \text{ (грн.)} \quad (6.5)$$

Накладні витрати приймемо у розмірі 20% від фонду заробітної плати. Розрахунок собівартості та договірної ціни наводимо в таблиці 6.6.

Таблиця 6.6 - Розрахунок собівартості та ціни ПП

| № п/п | Статті | Витрати, грн |
|-------|--|--------------|
| 1 | Основна заробітна плата | 18680 |
| 2 | Додаткова заробітна плата (15% від ОЗП) | 2802 |
| 3 | Відрахування на єдиний соціальний внесок (22% від ЗП (ЗП = ОЗП+ДПЗ)) | 4726.04 |
| 4 | Накладні витрати (20% від ЗП) | 4296.4 |
| 5 | Матеріали та комплектуючі | 312.5 |
| 6 | Амортизація | 500.10 |
| 7 | Разом собівартість ПП | 31317.04 |
| 8 | Прибуток (20%) | 6263.41 |
| 9 | Ціна без ПДВ | 37580.45 |
| 10 | ПДВ (20%) | 7516.09 |
| 11 | Ціна ПП | 45096.54 |

Таким чином, ціна ПЗ для визначення кутового положення ЛА з відеозображення без урахування ПДВ – 37580.45 грн., а з урахуванням – 45096.54 грн.

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

$$P_{\pi} = \frac{Ц - С}{С} \cdot 100\% = \frac{45096.54 - 31317.04}{31317.04} = 44\% \quad (6.8)$$

6.4 Розрахунок точки беззбитковості

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

Для підтвердження стійкості проекту необхідно, щоб точка беззбитковості була меншою за значення номінальних обсягів виробництва. Чим далі від них значення точки беззбитковості (у відсотковому співвідношенні), тим стійкіший проект. Проект зазвичай визнається стійким.

Точку беззбитковості розраховують за такою формулою:

$$N_6 = \frac{K}{C - S}, \quad (6.9)$$

де K - умовно постійні витрати, що приймаються рівними ціні теми;

C – вартість виробу;

S – собівартість виробу.

Ціна теми визначається за такою формулою:

$$C_{\text{темы}} = C_{\text{темы}} + P_{\text{темы}} = 31317.04 + 6263.41 = 37580.45 \text{ (грн)}, \quad (6.10)$$

де – загальний кошторис витрат (собівартість) теми, грн; $C_{\text{темы}}$

– плановий прибуток, який забезпечує рентабельну роботу безпосередніх виконавців теми, грн. $P_{\text{темы}}$

Розрахована ціна є переддоговірною ціною розробника – це мінімально допустима ціна, яка враховує кошторис витрат на розробку теми та прибуток, розрахований за встановленим коефіцієнтом рентабельності.

При остаточному призначенні ціни теми необхідно врахувати надбавки, пов'язані зі збутом виробу. Податок на додану вартість приймається 20% від ціни

теми. Таким чином, точка беззбитковості дорівнює:

$$N_6 = \frac{37580.45}{45096.54 - 31317.04} = 2.73 \approx 3 \text{ (копии)} \quad (6.11)$$

Таким чином, показник беззбитковості дорівнює 3 копії. Це означає, що реалізація 3 копій програмного забезпечення забезпечить беззбитковість проекту.

Графічне уявлення точки беззбитковості наведено на рис. 6.1.

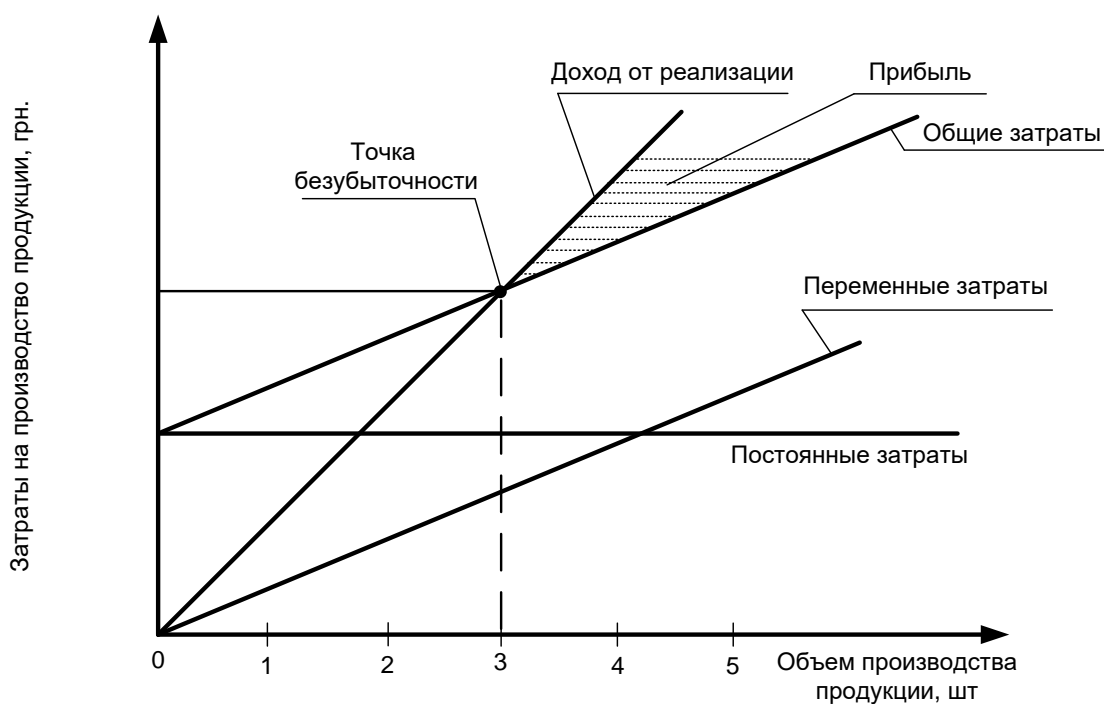


Рисунок 6.1 – Графік беззбитковості

Графік беззбитковості показує, що при обсязі продажів 3 копії ПЗ, дохід від реалізації продукції перевищує загальні витрати, таким чином, у зоні між ними ми отримуємо прибуток

6.5 Висновки

У цьому розділі проведено обґрунтування економічної доцільності розробки лабораторного стенду. Були розглянуті такі етапи просування цього товару:

- сегментація ринку;
- визначено трудомісткість етапів, розстановку виконавців.чином загальна тривалість робіт становить 44 днів, загальна трудомісткість 55 чол/днів. Трудомісткість керівника – 19 днів, програміста-аналітика – 36 днів.

- проведено розрахунок заробітної плати, визначено ціну теми. Основна заробітна плата дорівнює 18680 грн., а ціна теми 37580.45 грн. Прибуток від продукції становила 44 %;

- проведено розрахунок собівартості та ціни ПЗ. Вона становитиме без ПДВ – 37580.45 грн., а з урахуванням ПДВ – 45096.54 грн.;

- побудовано графік беззбитковості виробництва.

Для досягнення точки беззбитковості необхідно, щоб обсяг продажів $N_{пр}$ дорівнював 3 шт. Про це можна судити з побудованого графіку беззбитковості. Отримання прибутку здійснюється після реалізації продукції у кількості 3 прим. Отже, отримання прибутку можливо, коли обсяг продажів досягне точки беззбитковості.

ЗАКЛЮЧЕННЯ

У рамках виконання магістерської роботи було запропоновано моделі та методи оцінки кутового положення МБПЛА за даними відеозйомки, які можуть бути застосовані в реальному часі та реалізовані у вигляді програмного забезпечення навігаційного обчислювача у додатковому каналі діагностування працездатності систем кутової орієнтації МПВЛА.

Розроблено алгоритмічне та програмне забезпечення в середовищі MATLAB, що дозволяє проводити оцінку кутового переміщення малогабаритного безпілотного літального апарату за відео, що описує еволюцію ЛА.

Проведені дослідження показують доцільність застосовуваного підходу та аргументують вірність математичних припущень, що відкриває можливість побудови нових оптичних систем навігації, призначених для реалізації у спецобчислювачах другого рівня.

У роботі виконано техніко-економічну оцінку запропонованого рішення у вигляді програмного проекту. Зроблено висновок про економічну доцільність запропонованих рішень, так само у роботі виконано оцінку впливу небезпечних факторів на розробника програмного забезпечення. У результаті отриманий результат свідчить про безпеку запропонованого проекту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мартинова Л. А., Визначення координат та параметрів руху об'єкта на основі обробки зображень [Текст] / Л. А. Мартинова, А. В. Корякін, К.В. Ланцов, В.В. Ланців. Комп'ютерна оптика, 2012, том 36 №2 - С. 266 - 273.
2. Нгуєн Ван Чіонг, Тропченко А.А. Ієрархічний адаптивний алгоритм шаблонного пошуку для оцінки руху під час аналізу відеопослідовності // Науково-технічний вісник інформаційних технологій, механіки та оптики. 2016. Т. 16. № 3. С. 474-481.
3. Блаттер К. Вейвлет аналіз. Основи теорії. К: Техно-сфера, 2004.
4. Гонсалес, Р. Цифрова обробка зображень у середовищі matlab [Текст]/Р. Гонсалес, В. Вудс, С. Еддінс; пров. з англ. В.В. Чепижова. - К.: Техносфера, 2006. - 512 с.
5. Добеши І. Десять лекцій з вейвлетів. М.: НДЦ "Регулярна та хаотична динаміка", 2004.
6. Лайонс, Річард. Цифрова обробка сигналів [Текст]/Річард Лайонс; пров. з англ., за ред. А.А. Брітова. - 2-ге вид. - К.: БІНОМ-Прес, 2007. - 653 с.
7. Солоніна А.І., Арбузов С.М. Цифрове оброблення сигналів. Моделювання у MATLAB. - Львів: 2008.
8. Яковлєв О.М. Введення у вейвлет-перетворення. - Чернігів: НДТУ, 2003.
9. Dorf, R. C. Modern Control Systems [Текст] / R. C. Dorf, R. H. Bishop. – 13th Edition. – London : Pearson, 2017. – 1032 p.
10. Кулик, А. С. Концепция активной отказоустойчивости спутниковых систем ориентации и стабилизации [Текст] / А. С. Кулик // Радіоелектронні і комп'ютерні системи. – 2009. – № 2 (36). – С. 101–108.
11. Кулик, А. С. Становление и развитие рационального управления объектами в нештатных ситуациях [Текст] / А. С. Кулик // Радіоелектронні і комп'ютерні системи. – 2007. – № 5 (24). – С. 8–15.
12. Лойцянский, Л. Г. Курс теоретической механики: В 2-х томах [Текст]. Т. 2. Динамика. – 6-е изд. перераб. и доп. / Л. Г. Лойцянский, А. И. Лурье. – М. : Наука, Главная редакция физико-математической литературы, 1983. – 640 с.

13. Пасічник, С. М. Конспект лекцій з курсу «Основи моделювання систем авіоніки» [Електронний ресурс] / Режим доступу: <https://drive.google.com/drive/folders/18G6NILs94mXXFF5eKw1ihFLTWteRThFR>. – 09.12.2022 р.
14. Archived:APM 2.5 and 2.6 Overview [Електронний ресурс] / Режим доступу: <https://ardupilot.org/copter/docs/common-arms25-and-26-overview.html>. – 09.12.2022 р.
15. Кулік, А. С. Конспект лекцій з курсу «Теорія автоматичного керування» [Електронний ресурс] / Режим доступу: https://drive.google.com/drive/folders/1uSEeqs2-U_JC2KRIsdcmsgJy_NtzfWV2. – 09.12.2022 р.
16. Дорф, Р. Современные системы управления [Текст] / Р. Дорф, Б. Бишоп. – М. : Лаборатория базовых знаний, 2002. – 832 с.
17. Варталян, В. М. Методи ціноутворення: теорія і практикум: навч. посіб. [Текст] / В. М. Варталян, М. А. Голованова, С. М. Мельніков. – Х. : Нац. аерокосм. ун-т ім. М. Є. Жуковського «Харк. авіац. ін-т», 2010. – 224 с.

ДОДАТОК А

Лістинг програми оцінки параметрів траєкторії руху об'єкта в умовах дії перешкод

```

clc,close all,clear all
% Зчитування відеофайлу
Video = mmreader('C:\Users\krasnov.MEDICA\Desktop\blue_ball.avi');
% Визначення характеристик лічених відеоданих
width = Video.Width;
height = Video.Height;
frameRate = Video.FrameRate;
numOfFrames = Video.NumberOfFrames;

% Читання демонстраційного кадру
frameNo = 20;
size(frameNo)
F = read (Video, frameNo);
figure,imshow(F),title('Демонстраційний кадр'),impixelinfo
F1=rgb2gray(F);
F2=im2double(F1);
figure,imshow(F2),title('Демонстраційний кадр у форматі double'),impixelinfo

% Завдання дисперсії гаусових шумів
q=0.1

% Додавання шуму гауса до демонстраційного кадру
F3 = imnoise(F2,'gaussian',0,q);
figure,imshow(F3),title('Зашумлений демонстраційний кадр у форматі
double'),impixelinfo

% Створення нульових кадрів
A=zeros(480,640);
B = zeros (480,640);

% Створення шаблону для обчислення кореляції
P=ones(25,25);
% figure,imshow(P),title('Шаблон у форматі double'),impixelinfo

% Читання та обробка поточних кадрів
% Номер початкового кадру
u=15;
% Номер кінцевого кадру
z=120;
% Кількість оброблюваних кадрів
W=z-u+1

```

```

for k=u:z
% Читання поточного кадру
I = read (Video, k);

% Виділення синьої частини поточного кадру
I1=I(:, :, 3);
% figure, imshow(I1), title('Синя компонента поточного кадру'), impixelinfo

% Перетворення синьої компоненти поточного кадру у формат double
I2=im2double(I1);
% figure, imshow(I2), title('Синя компонента поточного кадру у форматі
double'), impixelinfo

% Додавання шуму гауса до поточного кадру
I3 = imnoise(I1, 'gaussian', 0, q);

% Перетворення зашумленого поточного кадру у формат double
I4=im2double(I3);
% figure, imshow(I4), title('Зашумлена синя компонента поточного кадру у форматі
double'), impixelinfo

% Обчислення взаємної кореляції кожного кадру із шаблоном
K=xcorr2(I2,P);
K1=xcorr2(I4,P);

% Обчислення максимумів кореляційних функцій
C=max(max(K));
C1=max(max(K1));

% Нормування кореляційних функцій на одиницю
L=K./C;
L1 = K1. / C1;
% figure, imshow(L), title('Нормована взаємнокореляційна функція синьої
компоненти')
% figure, imshow(L1), title('Нормована взаємнокореляційна функція зашумленої
синьої компоненти')

% Бінарізація взаємно кореляційних функцій з порога
BW=im2bw(L,0.9999);
BW1=im2bw(L1,0.9999);
% figure, imshow(BW), title('Бінарізація по порозі синій компоненти')
% figure, imshow(BW1), title('Бінарізація по порозі зашумленої синьої компоненти')

```

```

% Нормування розміру зображень кореляційних функцій
Q=imcrop(BW,[0,0,640,480]);
Q1=imcrop(BW1,[0,0,640,480]);
% figure,imshow(Q),title('Нормування зображення кореляційної функції'),imshowinfo
% figure,imshow(Q1),title('Нормування зображення кореляційної функції')

% Визначення координат точок незашумленої траєкторії
% Умова пошуку -> Q (m, n) = 1
for m=1:480
for n=1:640
if Q(m,n)==1
x0=n;
y0=m;
end;
end;
end;
% Визначення координат точок зашумленої траєкторії
% Умова пошуку -> Q1 (m, n) = 1
for m=1:480
for n=1:640
if Q1(m,n)==1
x1=n;
y1=m;
end;
end;
end;

% Оцінка лінійного зміщення точок траєкторії через шуми
dx=abs(x1-x0);
dy=abs(y1-y0);
DX = (dx) ^ 2;
DY = (dy) ^ 2;
DT = sqrt (DX + DY);
DT1 = DT;
V(k)=(DT1)^2;

% Оцінка координат точок незашумленої траєкторії щодо початку координат
XA = 321; YA = 241;
DX0 = (x0-XA);
DY0=(y0-YA);
RX = (DX0) ^ 2;
RY = (DY0) ^ 2;

```

```
% Номер точки траєкторії
```

```
Nom=k
```

```
% Обчислення дальності від початку координат до поточної точки незашумленої траєкторії
```

```
Range=sqrt(RX+RY)
```

```
% Обчислення кутової координати точок незашумленої траєкторії
```

```
Sangle=abs((DY0)/(Range));
```

```
Angle1=asin(Sangle)*180/pi;
```

```
if (y0<YA)
```

```
if (x0>=XA)
```

```
Angle=Angle1
```

```
end;
```

```
if (x0<XA)
```

```
Angle=180-Angle1
```

```
end;
```

```
end;
```

```
if (y0>YA)
```

```
if (x0>=XA)
```

```
Angle=360-Angle1
```

```
end;
```

```
if (x0<XA)
```

```
Angle=180+Angle1
```

```
end;
```

```
end;
```

```
if (y0==0)
```

```
if (x0>=XA)
```

```
Angle=0
```

```
end;
```

```
if (x0<XA)
```

```
Angle=180
```

```
end;
```

```
end;
```

```
% Оцінка координат точок зашумленої траєкторії щодо початку координат
```

```
XA = 321; YA = 241;
```

```
DX0 = (x1-XA);
```

```
DY0=(y1-YA);
```

```
RX = (DX0) ^ 2;
```

```
RY = (DY0) ^ 2;
```

```
% Обчислення дальності від початку координат до поточної точки зашумленої траєкторії
RangeN=sqrt(RX+RY)
```

```
% Обчислення кутової координати точок зашумленої траєкторії
Sangle=abs((DY0)/(RangeN));
Angle1=asin(Sangle)*180/pi;
if (y1<YA)
if (x1>=XA)
AngleN=Angle1
end;
if (x1<XA)
AngleN=180-Angle1
end;
end;
```

```
if (y1>YA)
if (x1>=XA)
AngleN=360-Angle1
end;
if (x1<XA)
AngleN=180+Angle1
end;
end;
```

```
if (y1==0)
if (x1>=XA)
AngleN=0
end;
if (x1<XA)
AngleN=180
end;
end;
```

```
% Оцінка кутового зміщення точок траєкторії через дії шумів
dangle=abs(Angle-AngleN);
V1(k)=(dangle)^2;
```

```
% Оцінка зміщення точок траєкторії по дальності через дії шумів
drange=abs(Range-RangeN);
V2(k)=(drange)^2;
```

```
% Перетворення точок траєкторії руху кульки у формат double
```

```

Q2=im2double(Q);
Q3=im2double(Q1);

% Побудова траєкторій руху кульки за відсутності та наявності шумів
A = A + Q2;
B = B + Q3;
end;

% Оцінка дисперсії лінійного зміщення точок траєкторії, обумовленої дією шумів
V(~V)=[];
MSEpoint=sum(V)/W
% Оцінка кутового зміщення точок траєкторії через дії шумів
V1(~V1)=[];
MSEangle=sum(V1)/W
% Оцінка зміщення точок траєкторії по дальності через дії шумів
V2(~V2)=[];
MSErange=sum(V2)/W
% Оцінка SKO лінійного відхилення точок траєкторії через шуми
SKOpoint = sqrt (MSEpoint)
% Оцінка SKO кутового відхилення точок траєкторії через шуми
SKOangle=sqrt(MSEangle)
% Оцінка SKO зміщення точок траєкторії по дальності через дії шумів
SKOrange=sqrt(MSErange)

% Відображення траєкторії руху
figure,imshow(A),title('Траєкторія руху'),impixelinfo
figure,imshow(B),title('Траєкторія руху'),impixelinfo

```

ДОДАТОК Б визначення кутового положення ЛА за точками екрану

```
clc,close all,clear all
% Зчитування відеофайлу
Video = mmreader('C:\Users\krasnov.MEDICA\Desktop\7_new.avi');
% Визначення характеристик лічених відеоданих
width = Video.Width;
height = Video.Height;
frameRate = Video.FrameRate;
numOfFrames = Video.NumberOfFrames;

% Читання демонстраційного кадру
frameNo = 1;
size(frameNo)
F = read (Video, frameNo);
figure,imshow(F),title('Демонстраційний кадр'),impixelinfo

% Створення шаблону для обчислення кореляції
P=ones(6,6);
% figure,imshow(P),title('Шаблон у форматі double'),impixelinfo

% ЧИТАННЯ І ОБРОБКА ПОТОЧНИХ КАДРІВ
% Номер початкового кадру
u=1;
% Номер кінцевого кадру
z=270;
% Кількість оброблюваних кадрів
W=z-u+1
% Читання кадрів через (1:2:3: ...)
v=4

pom = 0;
for i=u:v:zv
pom = pom+1
% Читання поточного кадру
I = read(Video,i);
% figure,imshow(I),title('Поточний кадр'),impixelinfo
% Виділення червоної компоненти поточного кадру
I1=I(:, :, 1);
% figure,imshow(I1),title('Червона точка'),impixelinfo
% Виділення зеленої компоненти кадру
I2=I(:, :, 2);
% figure,imshow(I2),title('Зелена точка'),impixelinfo
```

```

% Перетворення зображень кольорових точок у формат double
I3=im2double(I1);
% figure,imshow(I3),title('Червона точка у форматі double'),impixelinfo
I4=im2double(I2);
% figure,imshow(I4),title('Зелена точка у форматі double'),impixelinfo

% Обчислення взаємної кореляції зображень червоної точки із шаблоном
K=xcorr2(I3,P);
% figure,imshow(K),title('Взаємна кореляція зображення червоної точки з
шаблоном'),impixelinfo
% Обчислення взаємної кореляції зображень червоної точки із шаблоном
K1=xcorr2(I4,P);
% figure,imshow(K),title('Взаємна кореляція зображення червоної точки з
шаблоном'),impixelinfo

% Обчислення максимумів кореляційних функцій
C=max(max(K));
C1=max(max(K1));

% Нормування кореляційних функцій на одиницю
L=K./C;
L1 = K1. / C;
% figure,imshow(L),title('Нормована взаємнокореляційна функція червоної
точки'),impixelinfo
% figure,imshow(L1),title('Нормована взаємно-кореляційна функція зеленої
точки'),impixelinfo

% Бінаризація взаємнокореляційних функцій з порога яскравості
BW=im2bw(L,0.999);
BW1=im2bw(L1,0.99);
% figure, imshow(BW),title('Бінаризація по порозі червоної точки'),impixelinfo
% figure, imshow(BW1),title('Бінаризація по порозі зеленої точки'),impixelinfo

% Нормування розміру зображень кореляційних функцій
Q=imcrop(BW,[0,0,640,480]);
Q1=imcrop(BW1,[0,0,640,480]);
% figure,imshow(Q),title('Нормування зображення кореляційної функції для
червоної точки'),impixelinfo
% figure,imshow(Q1),title('Нормування зображення кореляційної функції для зеленої
точки'),impixelinfo

% ВИЗНАЧЕННЯ КООРДИНАТ ОПОРНИХ ТОЧОК
% Визначення координат червоної точки

```



```

% Умова пошуку -> Q (m, n) = 1
for m=1:480
for n=1:640
if Q(m,n)==1
x_red=n;
y_red=m;
end;
end;
end;

% Визначення координат зеленої точки
% Умова пошуку -> Q1 (m, n) = 1
for m=1:480
for n=1:640
if Q1(m,n)==1
x_green = n;
y_green = m;
end;
end;
end;

% РІШЕННЯ ЗАВДАННЯ ПЕРЕТВОРЕННЯ СИСТЕМИ КООРДИНАТ
% Визначення положення початку системи координат кадру оху
x_o=320;
y_o=240;
% Визначення положення точки центру мас об'єкта (червоної точки) щодо початку
системи координат оху
x_1 = x_red;
y_1=y_red;
dx = x_1-x_o;
dy=y_1-y_o;
% Визначення положення зеленої точки щодо початку системи координат оху
x_2 = x_green;
y_2=y_green;
dx1 = x_2-x_o;
dy1=y_2-y_o;

% Визначення положення зеленої точки щодо центру мас об'єкта (червоної точки)
dx2 = x_2-x_1;
dy2=y_2-y_1;

% Оцінка видалення червоної точки від початку системи координат оху
Range_red_point(nom)=sqrt((dx)^2 +(dy)^2);

```

```
% Оцінка видалення зеленої точки від початку системи координат оху
Range_green_point(nom)=sqrt((dx1)^2 +(dy1)^2);
```

```
% Оцінка видалення зеленої точки від червоної точки (центру мас)
Range_red_green(nom)=sqrt((dx2)^2 +(dy2)^2);
```

```
% Оцінка кутового положення червоної точки щодо системи координат оху
R1(nom)=(dy)/(Range_red_point(nom));
```

```
beta(nom)=asin(R1(nom))*180/pi;
```

```
if (dy==0)
```

```
if (dx>=0)
```

```
alpha (nom) = 0
```

```
end;
```

```
if (dx<0)
```

```
alpha (nom) = 180
```

```
end;
```

```
end;
```

```
if (dx==0)
```

```
if (dy>=0)
```

```
alpha (nom) = 90
```

```
end;
```

```
if (dy<0)
```

```
alpha (nom) = 270
```

```
end;
```

```
end;
```

```
if (x_1>x_o)
```

```
if (y_1<=y_o)
```

```
alpha(nom)=0-beta(nom)
```

```
end;
```

```
if (y_1>y_o)
```

```
alpha(nom)=360-beta(nom)
```

```
end;
```

```
end;
```

```
if (x_1<x_o)
```

```
alpha(nom)=180+beta(nom)
```

```
end;
```

```
% Оцінка кутового положення зеленої точки щодо системи координат оху
```

```
R2(nom)=(dy1)/(Range_green_point(nom));
```

```
beta1(nom)=asin(R2(nom))*180/pi;
```

```

if (dy1==0)
if (dx1>=0)
alpha1(nom)=0
end;
if (dx1<0)
alpha1(nom)=180
end;
end;

```

```

if (dx1==0)
if (dy1>=0)
alpha1(nom)=90
end;

```

```

if (dy1<0)
alpha1(nom)=270
end;
end;

```

```

if (x_2>x_o)
if (y_2<=y_o)
alpha1(nom)=0-beta1(nom)
end;
if (y_2>y_o)
alpha1(nom)=360-beta1(nom)
end;
end;

```

```

if (x_2<x_o)
alpha1(nom) = 180+beta1(nom)
end;

```

% Оцінка кутового положення зеленої точки щодо червоної (центр мас)

```
R3(nom)=(dy2)/(Range_red_green(nom));
```

```
beta2(nom)=asin(R3(nom))*180/pi;
```

```

if (dy2==0)
if (dx2>=0)
alpha2(nom)=0
end;
if (dx2<0)
alpha2(nom)=180
end;
end;
if (dx2==0)

```

```

if (dy2 >= 0)
alpha2(nom)=90
end;
if (dy2 < 0)
alpha2(nom)=270
end;
end;
if (x_2 > x_1)
if (y_2 <= y_1)
alpha2(nom)=0-beta2(nom)
end;
if (y_2 > y_1)
alpha2(nom)=360-beta2(nom)
end;
end;

if (x_2 < x_1)
alpha2(nom)=180+beta2(nom)
end;
end

for l=1:1:nom
xxx(l)=l;
end;
% ВІДОБРАЖЕННЯ РЕЗУЛЬТАТІВ РОЗРАХУНКУ
figure,subplot(1,1,1),stem(xxx,Range_red_point,'b','s','MarkerFaceColor','b','MarkerSize',
2),grid,title('Дальність до червоної точки від початку координат ');
xlabel('№ кадру'); ylabel('Дальність');
figure,subplot(1,1,1),stem(xxx,Range_green_point,'b','s','MarkerFaceColor','b','MarkerSize',
2),grid,title('Далекість до зеленої точки від початку координат ');
xlabel('№ кадру'); ylabel('Дальність');
figure,subplot(1,1,1),stem(xxx,Range_red_green,'b','s','MarkerFaceColor','b','MarkerSize',
2),grid,title('Відстань між червоною та зеленою крапками');
xlabel('№ кадру'); ylabel('Дальність');
figure,subplot(1,1,1),stem(xxx,alpha,'r','s','MarkerFaceColor','r','MarkerSize',2),grid,title('
Кут повороту червоної точки відносно початку координат '); xlabel('№ кадру');
ylabel('Кут повороту, град');
figure,subplot(1,1,1),stem(xxx,alpha1,'r','s','MarkerFaceColor','r','MarkerSize',2),grid,title('
Кут повороту зеленої точки відносно початку координат '); xlabel('№ кадру');
ylabel('Кут повороту, град');
figure,subplot(1,1,1),stem(xxx,alpha2,'r','s','markerFaceColor','r','MarkerSize',2),grid,title('
Кут повороту зеленої точки відносно червоний'); xlabel('№ кадру'); ylabel('Кут
повороту, град');

```

