

TODO

- 1. Посатавити номер наказу та дату видачи наказу;
- 2. Уточнити номер НДР;
- 3. Перевірити Технічне завдання;
- 4. Перевірити нумерацію сторінок;
- 5. Перевірити і додати джерела до списку літератури;
- 6. Можливо зменшити шрифт в прізвищах.

					НАУ 11 54 75 000 ПЗ			
Зм.	Лист	№ докум.	Підп.	Дата	Інтегрована інерціально-супутникова система навігації, що базується на принципах комплексної обробки інформації з використанням калманівської фільтрації	Лит.	Аркуш	Аркушів
Розробив	НовікМ.В.							
Перевірів	Мар'ясоваТ.І.						1	152
Консульт.	ФіляшкінМ.К.					ІАСУ 608		
Н. контр.	КозловаА.П.							
Затвердив	СинєглазовВ.М.							

# Технічне завдання

## 1. Найменування та галузь застосування

Комітет ІКАО з перспективних навігаційних систем (FANS- Future Air Navigation System) прийняв рішення про обов'язкове використання систем супутникової навігації в сполученні з ІНС. Тому в даний час у всіх галузях авіації основним інформаційним ядром сучасного навігаційного комплексу є інтегрована інерціально-супутникова система навігації (ICSH).

В теперішній час визнано, що одним з основних шляхів вдосконалення навігаційного обладнання є створення комплексних навігаційних систем, що інтегрують командні прилади в єдиний блок, здатний автоматично вирішувати задачу навігації ЛА. Сутність комплексування полягає у використанні інформаційної та структурної надмірності для підвищення точності, надійності та завадостійкості інформації при вимірюванні одних і тих же або функціонально зв'язаних навігаційних параметрів. Інформаційна надмірність полягає в тому, що забезпечується отримання однорідної інформації від декількох навігаційних датчиків різної фізичної природи з наступною сумісною обробкою цієї інформації в спеціалізованому обчислювачі. Надмірність структури комплексу забезпечує його працездатність при відмові, особливо короткотривалій, одного із датчиків.

Найбільш привабливим для розв'язання цієї задачі є залучення калманівської фільтрації. Фільтр Калмана призначений для ідентифікації (оцінювання) змінних стану системи за даними вимірювання вихідних сигналів цієї системи, які містять похибки вимірювання (вимірювальний шум). Ідентифікація оптимальна в тому сенсі, що сума квадратів похибок оцінювання змінних стану в будь-який момент часу має найменше з можливих значень. Похибка оцінювання це різниця між оцінкою фільтра й дійсним значенням змінних стану системи при наявності в системі детермінованих і випадкових похибок вимірювань. Отже, фільтр Калмана призначений для найкращого відновлення змінних стану, тобто для оптимального приглушення вимірювальних шумів.

					НАУ 11 54 75 000 ПЗ	Аркул
Зм.	Лист	№ докум.	Підп.	Дата		2

## 2. Підстава до розробки

Наказ по Національному авіаційному університету №1111/ст від «20» жовтня 2010 р.

## 3. Мета та призначення розробки

Основною метою роботи є аналіз та вибір схеми комплексної інерціально-супутникової навігаційної системи та схем оцінювання та корекції в цій системі і, як наслідок, розробка слабо зв'язаної схеми інтеграції, що базується на принципах комплексної обробки інформації з використанням калманівської фільтрації, дослідження ступеню впливу похибок датчиків первинної інформації безплатформної інерціальної системи (БІНС) та супутникової навігаційної системи (СНС) на стійкість фільтра Калмана, точнісні характеристики числення навігаційних параметрів і динаміку зміни похибок, впливу перерв у роботі СНС на траєкторний рух ЛА, моделювання зміни похибок комплексної інерціально-супутникової навігаційної системи.

## 4. Технічні вимоги

Основні технічні вимоги:

- точність визначення навігаційних параметрів:  
координат (СКВ), м ..... 15  
висоти (СКО), м .....  $0 \div 20$
- характеристики повинні зберігатися при:  
швидкості, м/с до ..... 300
- час визначення (холодний старт), хв .....  $< 2$
- частота відновлення координат,  $\text{с}^{-1}$  .....  $< 1$

					НАУ 11 54 75 000 ПЗ	Аркуш
Зм.	Лист	№ докум.	Підп.	Дата		3

5. масса, кг ..... <1
6. автоматичне, безперервне, всепогодне визначення поточних 3D координат місця розташування, вектора шляхової швидкості і шляхового кута ЛА.
7. автоматичний тестовий контроль функціонування блоків і вузлів апаратури, індикація блоків, що відмовили
8. стійке визначення навігаційних параметрів при русі з лінійними прискореннями і при стрибкоподібних змінах прискорення
9. БІНС повинна забезпечити визначення координат на протязі 60 с
10. взаємна корекція СНС та БІНС
11. підтримка СНС від БІНС для зменшення часу повторного запуску (“гарячого старту”) при короткочасних перервах у роботі СНС

#### Вимоги до засобів захисту

1. робоча температура, С .....-40...+60
2. робоча вологість (25 С) .....98

#### Додаткові вимоги

1. швидкість польоту ЛА, м/с ..... 400
2. максимальний кут крену ЛА,град ..... 450
3. максимальний кут тангажу ЛА,град ..... 200

### 5. Джерела розробки

1. Хоздоговірна науково-дослідна робота № 201-Хд04 “Ресурс”: “Розробка попередніх алгоритмів роботи інерціально-супутникової навігаційної системи та інформаційного зв’язку з літаком-носієм ”.
2. М.К. Філяшкін В.О. Рогожин, А.В. Скрипець, Т.І. Лукінова Інерціально-супутникові навігаційні системи. – К.: Вид-во НАУ, 2009. – 306 с.

3. Науково-дослідна робота № 396 ДБ-07 : Методика побудови комплексної навігаційної системи на основі спрощеного варіанту безплатформної інерціальної та високоточної супутникової навігаційних систем

## 6. Стадії та етапи розробки

Проведення аналізу та вибору навігаційного забезпечення ЛА, схеми комплексної інерціальної-супутникової системи навігації та застосування калманівської фільтрації для оцінки навігаційних даних, розробка слабо зв'язаної схеми інтеграції.

Розробка алгоритмів роботи комплексної навігаційної системи, дослідження ступеню впливу похибок датчиків первинної інформації безплатформної інерціальної системи (БІНС) та супутникової навігаційної системи (СНС) на точнісні характеристики числення навігаційних параметрів і динаміку зміни похибок, впливу перерв у роботі СНС на траєкторний рух ЛА.

Розробка програми моделювання помилок комплексної інерціальної-супутникової системи навігації з використанням фільтра Калмана. Пропозиція щодо удосконалення запропонованої навігаційної системи, шляхом модифікації оптимального фільтра для поліпшення його стійкості.

## 7. Порядок контролю та приймання

Контроль за ходом виконання календарного плану дипломної роботи протягом всього періоду дипломного проектування здійснює керівник дипломного проектування. Керівник визначає строки виконання та почерговість кожної стадії розробки дипломного проекту, проведення розрахункових та дослідницьких робіт, виконання графічних робіт, кінцевого оформлення дипломного проекту та подачі проекту до попереднього захисту на провідній кафедрі. Допуск до захисту у державній екзаменаційній комісії відбувається з дозволу завідувача кафедри після попереднього захисту.

					НАУ 11 54 75 000 ПЗ	Аркул
Зм.	Лист	№ докум.	Підп.	Дата		5

Приймання здійснюється на підставі захисту дипломної роботи ДЕК Інституту аерокосмічних систем управління.

Термін здачі дипломної роботи: «09» лютого 2011 р.

					НАУ 11 54 75 000 ПЗ	Аркул
Зм.	Лист	№ докум.	Підп.	Дата		6

# РЕФЕРАТ

Пояснювальна записка до дипломного проекту «Інтегрована інерціально-супутникова система навігації, що базується на принципах комплексної обробки інформації з використанням калманівської фільтрації»: стор. — 152 , рис. — 40, використаних джерел — 21.

ІНЕРЦІАЛЬНА НАВІГАЦІЙНА СИСТЕМА, МЕТОДИ КОМПЛЕКСНОЇ ОБРОБКИ ІНФОРМАЦІЇ, ФІЛЬТР КАЛМАНА, КОМП'ЮТЕРНО ІНТЕГРОВАНІЙ КОМПЛЕКС.

Об'єкт дослідження — методи та алгоритми комплексної обробки інформації, принципи побудови інтегрованих навігаційних комплексів, на базі процедури оптимальної калманівської фільтрації.

Мета диплому — аналіз та вибір схеми комплексної інерціально-супутникової навігаційної системи та схем оцінювання та корекції в цій системі і, як наслідок, розробка слабо зв'язаної схеми інтеграції, дослідження ступеню впливу похибок датчиків первинної інформації безплатформної інерціальної системи та точнісні характеристики числення навігаційних параметрів і динаміку зміни похибок, впливу перерв у роботі СНС на траекторний рух ЛА, моделювання зміни похибок комплексної інерціально-супутникової навігаційної системи.

Метод дослідження — математичне моделювання.

Розроблений алгоритм авіаційного бортового навігаційного комплексу, що включає безплатформенну інерціальну навігаційну систему, супутникову навігаційну систему та баровисотомір, дозволяє ефективно оцінити навігаційні параметри, залишивши переваги кожної із підсистем і значно знизити вплив їх недоліків.

Матеріали дипломного проекту рекомендується використовувати при проведенні наукових досліджень та у навчальному процесі.

					НАУ 11 54 75 000 ПЗ	Аркул
Зм.	Лист	№ докум.	Підп.	Дата		7

# Зміст

Перелік позначень	10
Вступ	11
<b>1 Аналіз та вибір навігаційного забезпечення</b>	<b>14</b>
1.1 Аналіз та вибір структури та варіанту комплексування інтегрованої інерціально-супутникової системи . . . . .	16
1.2 Аналіз та вибір варіанта супутникової навігаційної системи . . . . .	27
1.3 Вибір варіанту побудови інерціальної навігаційної системи . . . . .	34
1.4 Оцінка орієнтовних значень похибок вимірників первинної інформації БІНС . . . . .	52
<b>2 Аналіз та вибір методів оцінювання та корекції в комплексній інерціально-супутниковій системі</b>	<b>59</b>
2.1 Огляд методів оптимальної обробки інформації . . . . .	60
2.2 Рекурентний фільтр Калмана . . . . .	65
2.3 Алгоритм фільтра Калмана . . . . .	68
2.4 Проектування фільтра Калмана . . . . .	69
<b>3 Розробка алгоритмів оптимального комплексування в інерціально-супутникових систем навігації</b>	<b>71</b>
3.1 Моделі похибок інерціальних навігаційних систем . . . . .	72
3.2 Математичні моделі похибок супутникової системи навігації . . . . .	80
3.3 Математичні моделі похибок барометричного висотоміра . . . . .	82
3.4 Розробка алгоритмів оптимальної комплексної обробки навігаційної інформації . . . . .	83
<b>4 Дослідження розроблених алгоритмів</b>	<b>91</b>
4.1 Дослідження моделі похибок БІНС . . . . .	91
4.2 Рівняння траєкторії ЛА . . . . .	94
4.3 Моделювання алгоритмів калманівської фільтрації . . . . .	97



<b>5</b>	<b>Дослідження алгоритму оптимальної фільтрації за допомогою розробленого програмного забезпечення</b>	<b>102</b>
5.1	Опис вхідних та вихідних даних . . . . .	103
5.2	Опис графічного інтерфейсу користувача . . . . .	104
5.3	Опис структури програми . . . . .	105
<b>6</b>	<b>Охорона праці</b>	<b>107</b>
6.1	Вступ . . . . .	107
6.2	Опис робочого місця . . . . .	107
6.3	Шум . . . . .	109
6.4	Освітлення . . . . .	109
6.5	Електробезпека . . . . .	112
6.6	Забезпечення пожежної безпеки в розроблювальному проекті . .	114
6.7	Висновок . . . . .	117
<b>7</b>	<b>Охорона навколишнього середовища</b>	<b>118</b>
7.1	Дослідження екологічного впливу аіадійного транспортного комплексу . . . . .	118
7.2	Аналіз впливу шуму повітряних суден на навколишнє середовище	121
7.3	Аналіз впливу радіохвиль на навколишнє середовище . . . . .	122
7.4	Характеристика ПК як джерела забруднення . . . . .	123
7.5	Вплив на здоров'я користувача електромагнітних полів ПК . . .	123
7.6	Комп'ютер як джерело електростатичного поля . . . . .	124
7.7	Комп'ютер як джерело рентгенівського випромінювання . . . . .	124
7.8	Оптимальні шляхи утилізації ЕОМ . . . . .	125
7.9	Рекомендації щодо зменшення негативного впливу ПК на здоров'я людини та навколишнє середовище . . . . .	126
	<b>Висновки</b>	<b>128</b>
	<b>Перелік посилань</b>	<b>129</b>
	<b>Додаток А Код програми jRKalman</b>	<b>131</b>

# Перелік позначень

FANS Future Air Navigation System

GPS global positioning system

IMU Inertial Measurement Unit

ІНС інерціальна навігаційна система

ІССН інтегрована інерціально-супутникова система навігації

АБ антенний блок

БІНС безплатформенна інерціальна навігаційна система

ГЛОНАСС глобальна навігаційна супутникова система

ДПІ датчики первинної інформації

КВК командно-вимірювальний комплекс

КСЗ контур слідкування за затримкою

КСФ контур слідкування за фазою

ЛА літальний апарат

НС навігаційний супутник

ПІ прийомоіндикатор

ПКА підсистема космічних апаратів

ПКК підсистема контролю і керування

ПНК пілотажно-навігаційний комплекс

СНС супутникова навігаційна система

СРНС супутникова радіонавігаційна система

СФ смуговий фільтр

ФК фільтр Калмана

# ВСТУП

З розвитком та вдосконаленням літальних апаратів, ускладненням та розширенням виконуваних ними польотних завдань, розвивались та вдосконалювались пілотажно-навігаційні прилади та системи, які з впровадженням в склад бортового обладнання високопродуктивної обчислювальної техніки стало об'єднувати в пілотажно-навігаційні комплекси (ПНК).

ПНК є логічним наслідком еволюції систем навігації та управління і являє собою якісно новий ступінь в автоматизації літаководіння. В склад ПНК сучасного літального апарату будь-якого класу входять декілька навігаційних систем, зокрема інерціальні (ІНС ) та супутникові (СНС ) системи навігації. Завдяки різній фізичній природі та різним принципам формування навігаційного алгоритмічного забезпечення, ІНС та СНС добре доповнюють одна одну, що природно визначило їхню інтеграцію в складі сучасних ПНК, у якості інтегрованих інерціально-супутникових систем навігації (ІССН ). Доцільність спільного використання ІНС та СНС дозволяє, з одного боку, обмежити зростання похибок ІНС (головний недолік цієї системи) а, з іншого боку, знизити шумову складову похибок СНС, підвищити темп видачі інформації бортовим споживачам, істотно підняти рівень завадозахищеності (недоліки СНС), крім того забезпечується висока інформативність інтегрованої системи.

В результаті комплексування ІНС та СНС досягаються:

- підвищення точності визначення координат, висоти, швидкості і часу споживача;
- уточнення кутів орієнтації (курсу, крену і тангажа);
- оцінка й уточнення параметрів калібрування навігаційних датчиків, таких, як дрейфи гіроскопів, масштабні коефіцієнти, зсуви акселерометрів тощо;
- забезпечення на цій основі безперервності навігаційних визначень на всіх етапах руху, у тому числі і при тимчасовій непрацездатності приймача СНС у випадках впливу завад або енергійних маневрів ЛА.

Вищезазначені причини призводять до необхідності застосування інтегрованих інерціально-супутникових систем для навігації і керування ЛА практично всіх

типів. Тому комітет міжнародної організації цивільної авіації (ІКАО) з майбутніх навігаційних систем (FANS Future Air Navigation System ) прийняв рішення про обов'язкове використання систем супутникової навігації в поєднанні з ІНС.

Інтеграція інерціальної та супутникової систем реалізується шляхом комплексування двох систем. При вирішенні задачі комплексної обробки інформації в інерціально-супутникових системах навігації найбільш привабливим і розповсюдженим є, безумовно, калманівська фільтрація.

Фільтр калмана (ФК ) — ефективний рекурсивний фільтр оцінюючий вектор стану системи ряд неповних і зашумлених вимірів.

ФК призначений для рекурсивного дооцінювання вектора стану апріорно відомої динамічної системи, для розрахунку поточного стану системи необхідно знати поточні виміри, а також попередній стан фільтра. Таким чином фільтр Калмана, як і безліч інших рекурсивних фільтрів, реалізовані в часовому представленні, а не в частотному. Дана особливість відрізняє його від пакетних фільтрів, які вимагають в поточний такт роботи знати історію змін і/або оцінок.

В ряді випадків, кількість параметрів, що задають стан об'єкта, більше, ніж кількість спостерігаємих параметрів, доступно для вимірів. За допомогою моделі об'єкта по ряду доступних вимірів фільтр Калмана дозволяє отримати оцінку всього вектора внутрішнього стану об'єкта.

Оцінці за допомогою фільтра Калмана доступні лише змінні стану, які спостерігаються за результатами вимірювань вихідних сигналів. Якщо вектор стану спостережений не повністю, то можна замість звичайного фільтра, що ідентифікує весь вектор стану, синтезувати так званий редукований фільтр Калмана, тобто фільтр, який оцінює лише деякі змінні стану.

В результаті роботи фільтра обчислюється оцінка поточних похибок ІНС у визначенні координат, швидкостей, кутів орієнтації, а також оцінки похибок її акселерометрів і гіроскопів. На основі отриманих оцінок коригуються показання ІНС і її вимірювальних елементів.

Загальною вимогою для організації процесу комплексування є наявність математичних моделей підсистем, що підлягають комплексуванню. Сучасний стан обчислювальної техніки, знань в області інерціальної та супутникової навігації дозволяють скласти досить повні й адекватні моделі цих систем.

При побудові інтегрованих навігаційних систем широке поширення одер-

жав прийом, заснований на формуванні різницевих вимірів, зі складу яких виключаються шукані параметри. З використанням різницевих вимірів вирішується задача оцінювання похибок однієї підсистеми на фоні похибок іншої підсистеми. Цей прийом найчастіше називають методом одержання інваріантних оцінок. При реалізації такого методу використовуються лінійні моделі еволюції похибок підсистем і не потрібно введення в загальному випадку нелінійних моделей еволюції самих шуканих навігаційних параметрів, що істотно спрощує побудову алгоритмів комплексної обробки навігаційної інформації і дає можливість застосування добре освоєних процедур оптимальної лінійної калмановської фільтрації.

Таким чином для реалізації процедур оптимального комплексування інерціальної та супутникової систем навігації необхідно мати моделі еволюцій похибок окремих підсистем комплексу.

Отже розробка та дослідження працездатності алгоритмів роботи інтегрованих інерціально-супутникових систем для навігації і керування ЛА, оцінка ступіню впливу похибок датчиків первинної інформації безплатформної інерціальної системи (БІНС) та похибок супутникової навігаційної системи (СНС) на точнісні характеристики числення навігаційних параметрів і динаміку зміни похибок комплексної системи, оцінка впливу перерв у роботі СНС на траєкторний рух ЛА при польоті за складним маршрутом складають зміст магістерської роботи. Саме тому тема роботи є досить актуальною на сьогоднішній час. Робота виконувалася у рамках держбюджетної НДР 598ДБ09 “Методики створення інерціально супутникової навігаційної системи на основі нанотехнологічної БІНС” [TODO перевірити номер НДР]

# 1 Аналіз та вибір навігаційного забезпечення

Відповідно до розробленої у НДР „”( [TODO додати номер НДР]) методики побудови комплексної навігаційної системи, задача створення інерціально-супутникової системи навігації для визначення координат місцеположення рухомого об'єкта, передбачає:

Обґрунтування, та вибір (або розробка) структури і варіанту комплексування інтегрованої інерціально-супутникової системи на основі аналізу класу і технічних характеристик ЛА, з урахуванням діапазонів кутів крену та тангажа, ударних навантажень, максимальних швидкостей та прискорень, орієнтуючись на тип точності основних навігаційних засобів (комплекс середньої точності, низької вартості та малих габаритів і маси), масогабаритні характеристики, споживану потужність та вартість обладнання.

Формулювання основних характеристик комплексної системи у вигляді граничних погрішностей навігаційних визначень;

Для обраного варіанту інтегрованої інерціально-супутникової системи обирається з представленого на ринку модельного ряду авіаційних прийо́моіндикаторів супутникових систем потрібний за технічними характеристиками та розв'язуваними задачами тип прийо́моіндикатора, або формулюються технічні вимоги на розробку такого прийо́моіндикатора.

Для обраного варіанту інерціально-супутникової системи обирається схема інерціальної навігаційної системи;

На основі аналізу варіантів побудови датчиків первісної інформації БІНС і основних характеристик інерціальних датчиків, використовуючи методику попереднього оцінювання точностних характеристик ДПІ БІНС, обираються типи датчиків первісної інформації БІНС;

Ґрунтуючись на аналізі типових польотних завдань, що виконує даний клас ЛА або БПЛА, обираються варіанти систем координат, в яких повинні формулюватися кінематичні рівняння алгоритмів БІНС;

					НАУ 11 54 75 000 ПЗ			
Зм.	Лист	№ докум.	Підп.	Дата	Інтегрована інерціально-супутникова система навігації, що базується на принципах комплексної обробки інформації з використанням калманівської фільтрації	Лит.	Аркуш	Аркушів
Розробив	НовікМ.В.						14	152
Перевірив	Мар'ясоваТ.І.					ІАСУ 608		
Консульт.	ФіляшкінМ.К.							
Н. контр.	КозловаА.П.							
Затвердив	СинеглазовВ.М.							

Для обраних варіантів систем координат розробляються кінематичні рівняння алгоритмів БІНС, в яких при завданні орієнтацію зв'язаної системи координат відносно опорної за рішенням проектанта можуть бути використані алгоритми із застосуванням направляючих косинусів, кутів Ейлера, компонентів векторів кінцевого повороту й орієнтації, параметрів Родріга-Гамільтона, параметрів Келі-Клейна;

Після прийняття рішення про корекцію вертикального каналу інтегрованої інерціально-супутникової системи від вимірника барометричної висоти обирається тип баровисотоміра, або формулюються технічні вимоги на розробку такого вимірника, обирається тип датчика статичного тиску, розробляються алгоритми обчислення барометричної висоти за інформацією від датчика статичного тиску;

Випереджаючи розробку алгоритмів комплексної обробки навігаційної інформації, виконується розробка математичних моделей похибок датчиків, БІНС, СНС та вимірника барометричної висоти. Причому, якщо за моделі похибок датчиків БІНС, СНС та вимірника барометричної висоти можна застосовувати відомі математичні моделі, то математичну модель похибок БІНС потрібно формулювати базуючись на розроблених алгоритмах ідеальної роботи БІНС;

Після прийняття рішення про застосування в інтегрованої інерціально-супутникової системі оптимальної схеми комплексування виконується розробка алгоритмів комплексної обробки навігаційної інформації від БІНС, СНС та вимірника барометричної висоти на базі процедур оптимальної дискретної калманівської фільтрації залежно від обраного варіанту схем інтегрування: для слабкозв'язаної або сильнозв'язаної системи, з метою оцінювання похибок і введення поправок у вихідні дані БІНС.

## 1.1 Аналіз та вибір структури та варіанту комплексування інтегрованої інерціально-супутникової системи

Комплексна вторинна обробка інформації дає значний позитивний ефект тоді, коли відповідні вимірники працездатні, тобто на виходах радіонавігаційних вимірників, зокрема на виході корелятора приймача СНС, є досить "хороші" (природно що, і в таких випадках існують похибки слідування) сигнали, сформовані в результаті первинної обробки інформації. Реальні умови застосування апаратури навігаційних споживачів показують, що багато вимірників, і, насамперед, радіотехнічні супутникові системи навігації, далеко не завжди знаходяться в працездатному стані. У реальних умовах нерідко спостерігається зрив супроводу сигналів супутників, зокрема, через багатопроменевість і дію інших завад, захоплення похибкових сигналів і т.п.

В теперішній час розроблені схеми можливого комплексування СНС і ІНС у чотирьох основних варіантах:

1. роздільна схема;
2. слабо зв'язана схема;
3. жорстко зв'язана схема;
4. глибоко інтегрована схема.

Тут і в подальшому під СНС мається на увазі інтегрована СНС ГЛОНАСС/GPS. Перший варіант – це роздільна або розімкнута схема рис.1.1.1. Це найбільш простий варіант спільного використання ІНС і СНС. Тут обидві системи працюють незалежно одна від одної, але, оскільки похибки ІНС з часом зростають, то необхідно періодично або безперервно проводити корекцію ІНС за даними СНС. Для демпфірування вертикального каналу ІНС може бути застосована інформація від системи повітряних сигналів (СПС).

Періодична корекція може зводитися до періодичного перезапуску алгоритму ІНС із новими початковими умовами за координатами та швидкістю, дані про які надходять від приймача СНС. Безперервна корекція процедурно



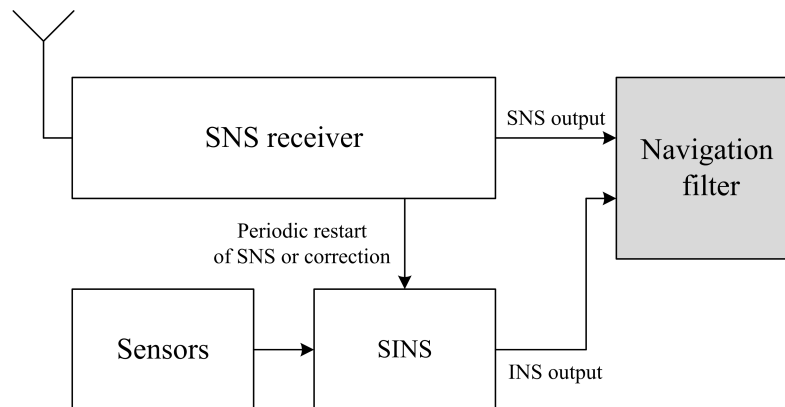


Рисунок 1.1.1 – Розімкнута схема

може бути оформлена як одночасна позиційна та швидкісна корекції ІНС за сигналами СНС. Така архітектура комплексування на етапі розв’язання навігаційної задачі (на етапі вторинної обробки інформації) забезпечує незалежність систем (крім моментів перезапуску або корекції) й інформаційну надмірність сукупної структури. Вихідна інформація двох систем може піддаватися комплексній обробці з використанням калманівської фільтрації.

В цілому комплексна система має більш високу точність як за координатами та швидкістю, так і за кутовою орієнтацією. При цьому зберігається можливість одержувати позиційну, швидкісну та кутову інформацію (у тому числі про перевантаження та кутову швидкість), необхідну для цілей пілотування та навігації з високою частотою, притаманною ІНС.

Крім того, для створення архітектури такої інтегрованої ІССН потрібні мінімальні зміни в апаратних засобах і програмному забезпеченні вже існуючого обладнання ЛА.

Наступною за глибиною зв’язку ІНС і СНС є слабо зв’язана система. Тут інерціальна система та приймач СНС як і раніше виробляють незалежні навігаційні вимірювання, однак з’являється з’єднувальний блок – обчислювач ІНС СНС, у якому формується оцінка координат і швидкості польоту, виробляється корекція даних, отриманих від ІНС (рис. 1.1.2).

В цій схемі функціональний розподіл підсистем може супроводжуватися їхнім фізичним поділом: приймач СНС, ІНС і навігаційний обчислювач конструктивно оформляються у вигляді закінчених роздільних блоків, між якими організовані відповідні інформаційні зв’язки, що не вимагають, як правило, високих швидкостей передачі даних. Зрозуміло, усі три перелічені компоненти



У СНС (див. рис. 1.1.2) сигнал, прийнятий антенним блоком, є сигналом несучої частоти, модульованим за амплітудою псевдовипадковим сигналом тривалістю  $dt \approx 1$  мксек (або 300 м еквівалентної довжини коду). Вхідні сигнали демодулюються і подаються на корелятори. Інформація з кореляторів передається в контури слідкування за фазою (КСФ) і затримкою (КСЗ). Контур слідкування за затримкою видає командні сигнали, які здійснюють затримку або випередження сигналів на виході корелятора (див.  $[+, -]$  на рис. 1.1.2) доти, поки на виході корелятора не з'явиться сигнал максимальної величини, а різниця сигналів корелятора на попередньому і поточному кроках не буде дорівнювати нулю. Це означає „захоплення” сигналу супутника, а величина отриманої при цьому затримки вважається часом поширення сигналу від супутника до приймача і використовується для обчислення псевдодальності  $\dot{R}_i$  до конкретного супутника. Синфазна та квадратурна складові сигналів несучої частоти

( $IQ$  відповідно – на рис. 1.1.2) подаються в контур слідкування за фазою несучої частоти (КСФ). Арктангенс пропорційний амплітуді квадратурного ( $Q$ ) сигналу до синфазного ( $I$ ) є похибкою КСФ. Цей сигнал похибки подається у вигляді зворотного зв'язку в корелятор, здійснюючи фазове автопідстроювання його частоти. Різниця частот опорного і прийнятого сигналів пропорційна швидкості зміни псевдодальності  $\dot{R}_i$ . При цьому контур КСФ має астатизм 3-го порядку, що дозволяє відслідковувати сигнали з постійним прискоренням (другої похідної від псевдодальності). Якщо цей контур захоплює і стежить за фазою, він подає коригувальний сигнал  $\Delta R$  у контур КСЗ, підвищуючи тим самим точність визначення псевдодальності  $R$ .

Інформація про вимірюванні псевдодальності  $R$  і псевдошвидкості  $\dot{R}_i$  використовується в алгоритмах розв'язання навігаційних задач для отримання координат і швидкості споживача, а також виправлень до еталона часу та частоти приймача СНС. При наявності надмірності з метою підвищення точності зчислення навігаційних параметрів здійснюється їхнє спільне оцінювання, зокрема з використанням оптимальної калманівської фільтрації.

Робота супутникової системи коригується від ІНС на етапі “холодного” і “гарячого” стартів. Тут приймач СНС використовує інформацію від ІНС тільки з метою більш надійного та швидкого відновлення захоплення сигналу у випадку його втрати. На схемі це показано зв'язком вихідного блоку ІНС і корелятора. Передана по цьому каналу інформація про обчислені координати та швидкість ЛА у випадку втрати слідкування дозволяє розрахувати оцінки передбачуваної затримки сигналу  $\tau$  та доплерівського зсуву частоти несучої  $f_{\text{доп}}$ , що суттєво знижує час пошуку та захоплення сигналу. В результаті значно знижується час відновлення роботи приймача після втрати сигналу, тобто тут в деякому сенсі реалізоване об'єднання ІНС і СНС не тільки на рівні вторинної обробки інформації, а й на рівні первинної обробки радіосигналів.

У блоці ІНС на рис. 1.1.2 показана структура безплатформної інерціальної системи. Блок датчиків видає вектори кутових швидкостей  $\omega$  та лінійних прискорень  $a$ . У блоці „кінематика обертового руху” виконується інтегрування кінематичних рівнянь кутового руху та формується матриця напрямних косинусів  $B$  за інформацією датчиків кутових швидкостей. Матриця напрямних косинусів  $B$  разом із даними акселерометрів використовується в блоці інтегрування кінематичних рівнянь поступального руху – блок “кінематика поступаль-

ного руху”. На виході цього блоку формуються координати та швидкості ЛА у вибраній навігаційній системі.

У середній частині рис. 1.1.2 зображено з’єднувальний блок – обчислювач ІНС СНС, що копіює алгоритм безплатформної ІНС, здійснює в блоці „компенсатор похибок датчиків” компенсацію похибок датчиків відповідно до моделей цих похибок та реалізує безпосередньо комплексування ІНС і СНС. Оцінка параметрів, що характеризують фазові координати руху ЛА, реалізується в польоті за результатами, наприклад, розширеної калманівської фільтрації сигналів ІНС і СНС у блоці ФК. За результатами оцінювання здійснюється позиційна та швидкісна корекція копії алгоритмів безплатформної ІНС. Корекція самої ІНС у слабо-зв’язаних системах не передбачається. Але в ІНС передбачається можливість компенсації інструментальних похибок вимірювальних елементів за апріорними даними (наприклад, за паспортними даними системи) або за значеннями оцінок цих похибок, що отримані в обчислювачі ІНС СНС. В результаті в основний алгоритм ІНС передаються скориговані показання датчиків кутової швидкості і акселерометрів.

Як видно, у слабо зв’язаній системі навігаційні параметри, так само як і в роздільній схемі, виробляються незалежно як у ІНС так і в СНС, причому, як уже відзначалося, до складу приймача включена схема оцінювання (як правило, фільтр Калмана). Така схема зветься „каскадною” через два послідовно включених фільтри Калмана. Достоїнством такої схеми є висока надійність інтегрованої системи, а недоліком – взаємна кореляція похибок оцінок першого фільтра (фільтра супутникового приймача) і їх відмінність від білих шумів. Надходячи з виходу СНС на вхід другого фільтра Калмана, і стаючи для нього шумами вимірювань, вони порушують умови оптимальної роботи цього фільтра. Крім цього, у такій схемі необхідно здійснювати заходи синхронізації вимірювань ІНС і приймача СНС.

Підвищений рівень автономності ІНС (передбачається, що підсистема ІНС може працювати автономно протягом 1-ї години) вимагає значної точності інерціальних датчиків (датчиків кутових швидкостей і акселерометрів) і застосування досить складних алгоритмів інерціальної навігації. Тому такі системи досить дорогі та складні. Такі системи доцільно застосовувати в ПНК високої та середньої точності, але, наприклад для БПЛА, вони занадто дорогі.

У літературі можна знайти ділення слабо зв’язаних схем на три типи:

					НАУ 11 54 75 000 ПЗ	Аркуш
Зм.	Лист	№ докум.	Підп.	Дата		20

стандартну, агресивну і так звану *MAGR*-схему (*Military Airborne GPS Receiver*). Відмінність „агресивної” схеми від стандартної полягає в тому, що в ній використовується інформація БІНС про прискорення для екстраполяції навігаційних вимірювань приймача СНС в період між супутниковими вимірюваннями. *MAGR* - схема фірми *Rockwell* використовує інерціальні вимірювання в контурі слідкування за кодом СНС-приймача при провалі „захоплення” у контурі слідкування за несучою частотою. У цьому випадку можна говорити про повноцінне комплексування як на рівні вторинної обробки інформації, так й на рівні первинної обробки інформації.

Третій варіант інтеграції систем – жорстко зв’язана схема (рис. 1.1.3). У жорстко зв’язаних системах ступінь автономності ІНС значно менший, ніж у слабо зв’язаних системах: допускається автономна робота протягом від декількох секунд до декількох десятків секунд. Практично в цих системах ІНС найчастіше є додатком для СНС. Основна навігаційна інформація виробляється в СНС, у той час як ІНС інтерполює значення навігаційних параметрів у період між двома сусідніми тактами надходження інформації від СНС, а також забезпечує навігаційною інформацією системи керування польотом при короточасній втраті сигналів від супутників.

ІНС у жорстко зв’язаних системах забезпечує „сирі вимірювання”. Блок датчиків видає вектори кутових і лінійних координат.

Компенсація похибок датчиків відповідно до моделей цих похибок виконується в блоці компенсатора похибок від розширеного фільтра Калмана. Інтегрування кінематичних рівнянь обертового руху та поступального руху виконується з урахуванням скоригованих координат. Тобто в у жорстко зв’язаних системах виконується одночасно процедури оцінювання (фільтрації) і коригування ІНС.

Приймач СНС функціонує аналогічно описаному вище варіанту слабо зв’язаної схеми. Відмінністю даної структури від попередніх є відсутність у складі приймача фільтра Калмана. У жорстко зв’язаній схемі ІНС і приймач лише забезпечують склад вимірювань для загального обчислювального блоку, в якому реалізований єдиний фільтр Калмана. Вимірювання для фільтра в жорстко зв’язаних системах будуються за різницею псевдодальностей або/і швидкостей зміни псевдодальностей, визначених, з одного боку, в ІНС за обчисленими координатами ЛА й ефемеридами супутника, і вимірюваними приймачем-

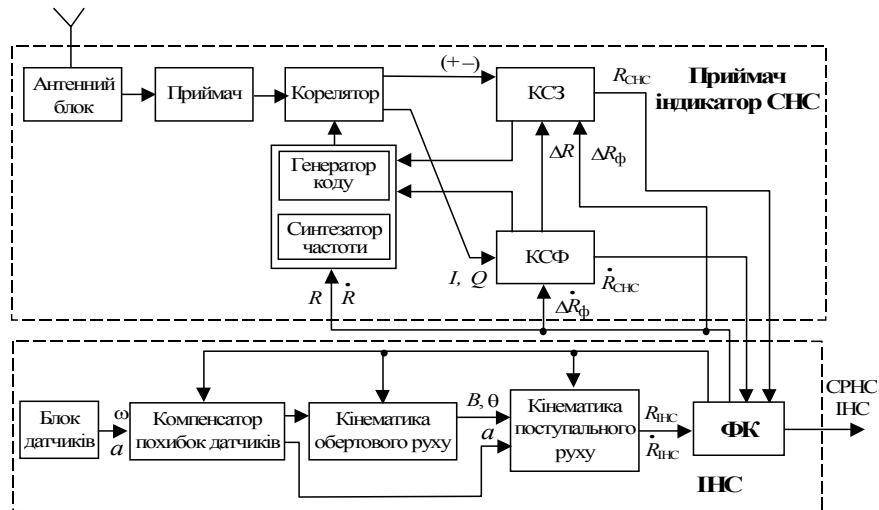


Рис. 10.3.

Рисунок 1.1.3 – Жорстко зв’язана схема інтегрування

індикатором СНС, з іншого. При цьому за навігаційну систему координат ІНС доцільно вибрати ту систему координат, в якій працює СНС.

Фільтр Калмана, на відміну від попереднього випадку, повинен бути дуже швидкодіючим. Це пов’язано з тим, що зв’язок блока фільтра Калмана з контурами приймача СНС значно більш жорсткий, ніж у попередньому випадку, оскільки відмінною рисою жорстко зв’язаної схеми є використання контурами слідування за затримкою і фазою інформації про розрахункові псевдодальності і псевдошвидкості (або про їхні збільшення), які надходить саме від фільтра Калмана. Використання цієї інформації дозволяє істотно поліпшити стійкість слідування і знизити час відновлення роботи приймача у випадку втрати сигналів супутників. Необхідно, щоб ці дані надходили з високою швидкістю так, щоб період часу між вимірюваннями в підсистемі СНС був розбитий на велику кількість підінтервалів з метою корекції контурів слідування. Це потрібно для того, щоб постачати контуру слідування інформацію навіть тоді, коли вхідний сигнал приймача відсутній або подавлений завадами, тобто тут реалізоване повномасштабне комплексування ІНС/СНС і на рівні первинної обробки інформації.

Жорстко зв’язані системи мають більшу точність при тих самих інерціальних датчиках у порівнянні зі слабо зв’язаними системами. У цих системах за рахунок додаткових сигналів корекції від ІНС смуга пропускання контурів слідування СНС може бути значно зменшена. При цьому зростає завадостійкість цих систем і зменшується ймовірність втрати сигналів, що відслідковують

ться. До того ж застосування фільтра Калмана, що відновлює повний вектор стану, включаючи псевдодальність  $R$  і швидкість її зміни  $\dot{R}$ , навіть при неповних вимірюваннях, дозволяє СНС працювати навіть при кількості видимих супутників менше 4-х. Якщо кількість цих супутників більше 4-х, то фільтр Калмана здійснює комплексування інформації, що надходить від видимих супутників. Однак, наявність лише одного фільтра Калмана призводить до втрати надмірності системи, тому що стає доступним лише одне спільне рішення.

Як і у слабо зв'язаних системах тут передбачено коригування СНС від коректованої ІНС на етапах „холодного” та „гарячого” стартів, а відновлені значення псевдодальності  $\Delta R_D$  і швидкості її зміни  $\Delta \dot{R}_D$ , надходячи в контури слідкування за затримкою КСЗ та за фазою КСФ сигналу СНС, забезпечують процедуру інтерполяції.

Таким чином, основні відмінності жорстко зв'язаної схеми від слабо зв'язаної полягають у наступному:

1. використання вихідної інформації ІНС про прискорення в контурі слідкування за кодом і доплерівським зсувом несучої частоти, що дозволяє звузити смугу пропускання контурів слідкування і підвищити швидкодію та точність настроювання;
2. використання вимірювань псевдодальностей та псевдошвидкостей (а не координат і швидкостей) для оцінювання похибок ІНС.

Як вже було зазначено, жорстко зв'язані системи забезпечують більш високу точність розв'язання навігаційної задачі в порівнянні зі слабо зв'язаними системами. До інших переваг такої схеми можна віднести:

1. відсутність проблем взаємної кореляції шумів вимірювань та їхніх відмінностей від білих шумів;
2. відсутність проблеми синхронізації вимірювань ІНС і СНС, оскільки використовується один формувач тактових частот;
3. можливість виявлення та відбраковування схиблених вимірювань псевдодальностей за їхніми передбачуваними значеннями, сформованими з використанням даних від ІНС.

До недоліків жорстко зв'язаних систем можна віднести:

1. необхідність розробки спеціальної апаратури споживача (приймача-індикатора СНС);
2. використання складних співвідношень для вимірювань;
3. зниження надійності, оскільки відмова ІНС призводить до відмови системи в цілому;
4. відсутність надмірності, що ускладнює рішення задач діагностики та контролю.

Два останні недоліки можна усунути, використовуючи фільтр Калмана в приймачі СНС і перераховуючи навігаційну інформацію скоригованої ІНС у навігаційну систему координат споживача. Таке рішення створює деякий проміжний варіант між слабко і жорстко зв'язаними схемами – варіант інерціально-супутникової системи середньої інтеграції (рис.1.1.4) .

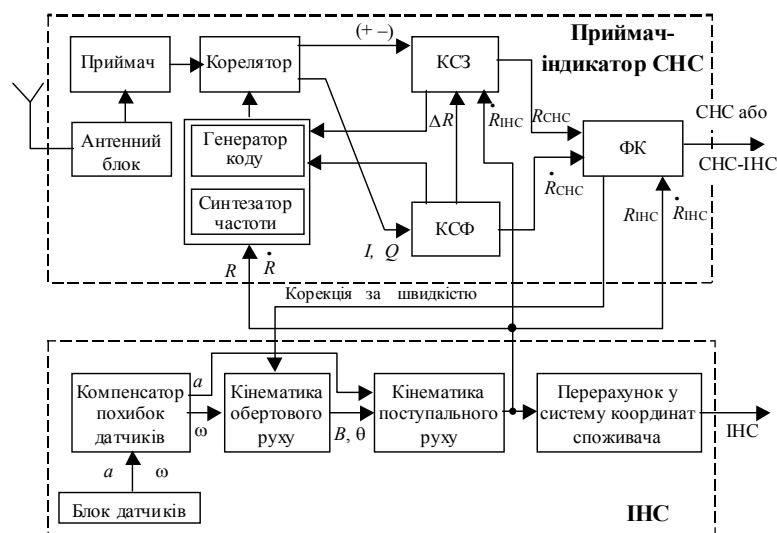


Рис.10.4

Рисунок 1.1.4 – Система середнього типу інтеграції

Система, що зображена на рис. 1.1.4, надає два навігаційних рішення: одне на виході блоку СНС, інше – на виході ІНС. Блоки, що зображені на схемі рис. 1.1.4, мають той же зміст, що і на попередніх схемах. ІНС може забезпечувати розв'язання навігаційної задачі навіть при відсутності сигналів від СНС. Крім того, передбачений режим підтримки роботи СНС від ІНС за рахунок поліпшення стійкості слідкування. Блок КЗФ – блок слідкування за фазою несучої частоти, зазвичай, більш уразливий для природних або штучних завад. Тому, якщо цей блок слідкування втратив „захоплення” фази і не виконує



функцію підтримки слідування КЗС, тобто працює тільки блок КЗС - блок слідування за затримкою, то ІНС заміняє відсутній сигнал  $\Delta R$  на сигнал  $\dot{R}$ , підтримуючи, таким чином, роботу супутникової системи без збоїв.

ІНС у цьому випадку, так само як і у всіх інших, використовується також і для екстраполяції сигналів положення  $R$  і швидкості  $\dot{R}$  між двома вимірюваннями СНС.

Оскільки у фільтрі Калмана відновлюється цілком весь вектор стану ЛА, то змінні кутової орієнтації використовуються для корекції алгоритмів інтегрування кінематичних рівнянь кутового руху, тобто здійснюється корекція за швидкістю.

Крім розглянутих варіантів структур комплексної системи, існують ще й інші варіанти, що побудовані як за принципом слабкої, так і жорсткої інтеграції. Але при цьому слід мати на увазі, що ці варіанти вимагають значно більш складного і дорогого математичного забезпечення в порівнянні з уже розглянутими варіантами структур.

Так звані глибоко інтегровані системи є ще більш складними і менш гнучкими з огляду організації їхньої структури, мають жорстку організацію зв'язків і єдиний вихід (рис. 1.1.5).

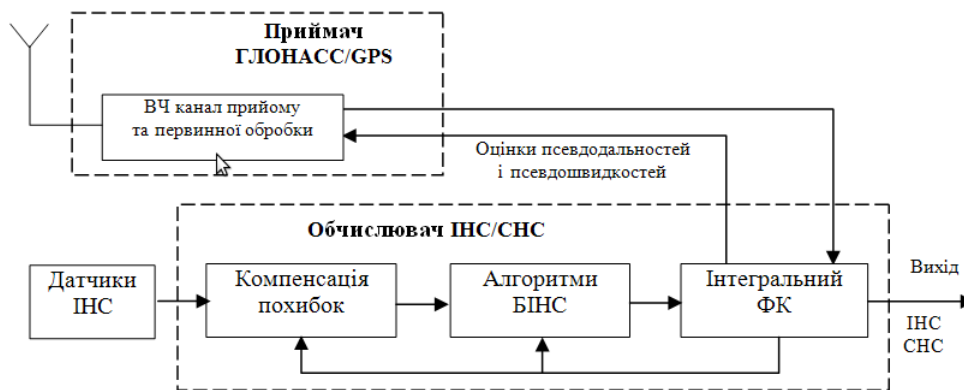


Рисунок 1.1.5 – Глибоко інтегрована схема

Обчислювач ІНС/СНС реалізує алгоритми безплатформної ІНС й алгоритми оптимальної оцінки параметрів. Всі оцінки виробляються в інтегральному фільтрі Калмана, а приймач СНС ГЛОНАСС/GPS ще більш спрощується. У цій схемі він складається тільки з високочастотного каналу прийому і первинної обробки інформації, що включає високочастотний прийомний тракт, генератор коду, корелятори і схему „захоплення”. Виходи кореляторів є входами для ін-

тегрального фільтра Калмана, де обчислюються не тільки похибки ІНС, але й оцінки псевдодальностей і псевдошвидкостей, які передаються в приймач для поліпшення характеристик „захоплення” сигналу. Таким чином, традиційні контури слідування за кодом і доплерівською частотою включаються в загальний інтегральний фільтр комплексної системи. У такій схемі фільтр повинен мати двадцятий-сороковий порядок, і для його реалізації потрібна БЦОМ із високою швидкодією.

Усі перераховані схеми комплексування СНС і ІНС (крім першої), одержують на виході фільтра Калмана оцінки інструментальних похибок ІНС (похибки зсуву нулів гіроскопів і акселерометрів, похибки масштабних коефіцієнтів і т. ін.), які використовуються для корекції інерціальних датчиків. Тому при перервах надходження даних із приймача отримані раніше оцінки похибок ІНС і її вимірювальних елементів дозволяють поліпшити точнісні характеристики ІНС в автономному режимі.

В табл. 1.1.1 підсумовані основні особливості перелічених схем комплексних систем.

Перші дві з приведених структур інтегрованих систем можуть бути реалізовані з використанням існуючих супутникових приймачів та інерціальних систем. Разом з тим жорстко зв'язана і особливо глибоко інтегрована схеми в обов'язковому порядку потребують розробки спеціальних приймачів і обчислювачів супутникової навігації для забезпечення корекції обох контурів спостереження від інерціальної системи навігації, а також створення спеціалізованих датчиків для інерціальних систем, виготовлених на одній технологічній та конструктивній базі. При цьому можуть бути використані самі передові технології, наприклад мікромеханічні датчики. Це дозволяє одержати інтегровані системи менших габаритів, маси, енергоспоживання. Але з точки зору розробника ці обставини є певним недоліком таких систем.

Об'єктом, на який передбачається встановлювати інтегровану навігаційну систему є пасажирський середньомагістральний літак українського виробництва, через це обираємо слабкозв'язану схему комплексування, оскільки архітектура такої інтегрованої КІССН потребує мінімальної зміни в апаратних засобах і програмному забезпеченні складових систем комплексної системи. Це дає можливість використовувати надійні, покупні і уніфіковані блоки системи і легко розширювати навігаційне забезпечення додаючи нове обладнання. До того ж ви-

Таблиця 1.1.1 – Особливості схем комплексування

Тип системи	Основні особливості
Роздільна	Надмірність, обмеженість похибок оцінок місця розташування і швидкості, наявність інформації про орієнтацію і кутову швидкість, висока швидкість видачі інформації, мінімальні зміни в бортовій апаратурі
Слабко зв'язана	Усі перераховані особливості роздільних систем, плюс більш швидке відновлення слідкування за кодом і фазою сигналів СНС, виставлення та калібрування БІНС у польоті, як наслідок – підвищена точність під час відсутності сигналу СНС
Жорстко зв'язана	Подальше поліпшення точності і калібрування, підвищена стійкість слідкування за сигналами СНС при маневрах ЛА, підвищена завадостійкість
Глибоко інтегрована	Достоїнства: єдиний фільтр усуває проблему “каскадного” включення фільтрів, компактність, знижені вимоги з енергозабезпечення. Недоліки: вектор стану містить до 40 компонентів, тому фільтр складно реалізувати; необхідність розробки спеціальних датчиків

хідна інформація двох систем може просто піддаватися комплексній обробці з використанням тих чи інших алгоритмів оптимальної фільтрації. Окрім цього структурна надмірність надає більшу надійність системи: вихід однієї підсистеми з ладу не впливає на роботу іншої (на відміну з жорстко зв'язаною схемою).

Отже після вибору методу комплексування, необхідно визначитись з варіантами супутникової та інерціальної навігаційної системи, які б оптимально підходили під вибрану архітектуру побудови.

## 1.2 Аналіз та вибір варіанта супутникової навігаційної системи

На сьогодні має сенс розглядати лише дві супутникові навігаційні системи : GPS (Global Positioning System), ГЛОНАСС (Глобальна Навігаційна Супутникова Система).

Двадцять чотири супутники системи GPS знаходяться на 12-годинних орбітах висотою 20146 км із нахиленням орбіти, рівним 55. Таким чином, у будь-якій крапці земної кулі в межах прямої видимості мається не менш чотирьох супутників у конфігурації, сприятливої для місцевизначення.

Система заснована на обчисленні відстані від користувача до супутника за обмірюваним часом від передачі сигналу супутником до прийому цього сигналу користувачем.

Глобальна Навігаційна Супутникова Система (ГЛОНАСС) – це технології російських конструкторів і вчених. Вона складається з 21 супутників, що, знаходячись у заданих крапках на високих орбітах, безупинно випромінюють у бік Землі спеціальні навігаційні сигнали. Будь яка людина або транспортний засіб, оснащений спеціальним приладом для прийому й обробки цих сигналів, можуть з високою точністю в будь-якій крапці Землі і навколоземного простору визначити власні координати і швидкість руху, а також здійснити прив'язку до точного часу.

У складі сучасної супутникової радіонавігаційної системи (СРНС) типу ГЛОНАСС і GPS функціонують три основні підсистеми:

1. Підсистема космічних апаратів (ПКА), що складається з навігаційних супутників (НС) (мережа навігаційних супутників - космічний сегмент). ПКА СРНС складається з визначеного числа навігаційних супутників. Основні функції НС — формування і випромінювання радіосигналів, необхідних для навігаційних визначень споживачів СРНС, контролю бортових систем супутника підсистемою контролю і керування СРНС. Відповідні характеристики сигналів НС і способи їхньої обробки дозволяють проводити навігаційні виміри з високою точністю.

2. Підсистема контролю і керування (ПКК) (наземний командно-вимірювальний комплекс (КВК)) - сегмент керування. ПКК являє собою комплекс наземних засобів, що забезпечують спостереження і контроль за траєкторіями руху НС, якістю функціонування їхньої апаратури, керування режимами її роботи і параметрами супутникових радіосигналів, складом, обсягом і дискретністю переданої із супутників навігаційної інформації та ін.
3. Апаратура споживачів (АС) СРНС (приймоіндикатори (ПІ)) - сегмент споживачів. Апаратура споживачів призначена для визначення просторових координат, вектора швидкості, часу й інших навігаційних параметрів у результаті прийому й обробки радіосигналів багатьох навігаційних супутників.

На вхід ПІ надходять сигнали від НС, що знаходяться в зоні радіо видимості. Оскільки для рішення навігаційної задачі необхідно вимірити псевдодальності і псевдошвидкості відносно, як мінімум, чотирьох НС, то ПІ повинний бути багатоканальним (більш 24 у сполучених ГЛОНАСС і GPS ).

Сучасні ПІ є аналого-цифровими системами, що здійснюють аналогову і цифрову обробку сигналів. Перехід на цифрову обробку здійснюється на одній із проміжних частот, при цьому має місце тенденція до підвищення цієї проміжної частоти.

Основа типового варіанту ПІ – два конструктивно роздільних блоків: антенний блок (АБ) та прийомообчислювач (ПО), які призначені для прийому й обробки навігаційних сигналів супутників з метою визначення необхідної споживачам інформації (просторово-тимчасових координат, напрямки і швидкості і т.п.).

В антенному блоці (рис. 1.2.1) сукупність сигналів НС, прийнятих антенною, попередньо підсилюється і фільтрується по всій смузі несучих частот НС у попередньому підсилювачі (ПП) зі смуговим фільтром (СФ).

Прийомообчислювач виконаний у вигляді блоку, у якому розташовані модулі вторинних джерел живлення і плати — прийомокорелятора, навігаційного обчислювача та інтерфейсного пристрою (рис. 1.2.2). Вхід ПО через фідерну лінію з'єднаний з виходом антенного блоку. В аналоговому приймачі АП сигнали підсилюються, фільтруються і переносяться з несучої частоти на проміжну



навігаційні параметри (дальність, її похідні й ін.) щодо відповідних НС, а потім розраховуються вторинні — географічна широта, довгота, висота споживача і т.д.

Вектор швидкості споживача обчислюють шляхом обробки результатів вимірів доплерівських зсувів частоти сигналів НС з урахуванням відомого вектора швидкості супутника.

Інтерфейсний пристрій (ІП) призначений для забезпечення взаємодії прийомоіндикатора з зовнішніми пристроями такими, наприклад, як пульт керування й індикації (ПКІ). Додатково до складу ІП входять два підсилювачі П, що формують ознаку відмови ПІ і сигнали дискретного керування, а також 8-розрядний регістр Рг, що приймає сигнали дискретного керування. Цей регістр доступний для читання з боку НО. Останній, у залежності від інформації, що знаходиться в регістрі, вибирає той або інший режим роботи.

Таким чином, основною операцією, що виконуваної в СНС за допомогою космічного сегменту, сегменту керування та сегменту споживача, є визначення просторових координат місця розташування споживачів і часу, тобто просторово-тимчасових координат (ПТК). Як було показано, цю операцію здійснюють відповідно до концепції незалежної навігації, що передбачає обчислення шуканих навігаційних параметрів безпосередньо в апаратурі споживача. У рамках цієї концепції в СРНС обраний позиційний спосіб визначення місця розташування споживачів на основі беззапитних (пасивних) далекомірних вимірів по сигналах декількох навігаційних штучних супутників Землі з відомими координатами. Висока точність визначення місця розташування споживачів обумовлена багатьма факторами, включаючи взаємне розташування супутників і параметри їхніх навігаційних сигналів. Структура космічного сегмента забезпечує для споживача постійну видимість необхідного числа супутників.

Використання СНС в інтересах місцезнаходження і навігації рухливих об'єктів, а також у рішенні спеціальних задач (спостереження, аерофотознімання, пошук корисних копалин, пошук і порятунк транспортних засобів, що терплять нещастя, і людей) висуває високі вимоги.

Вимоги до точнісних характеристик, таких як середньоквадратичне відхилення помилки (СКП) визначення навігаційних параметрів, показників надійності навігаційного забезпечення, тощо наступні:

- доступність (готовність), мірою якої є імовірність працездатності СРНС перед виконанням тієї або іншої задачі та у процесі її виконання. Чисельні значення доступності складають 0,95.... . 0,997;
- цілісність, мірою якої є імовірність виявлення відмови протягом часу, рівному заданому або менше. Вимоги до цілісності для маршрутних польотів складає 0,999;
- безперервність обслуговування, мірою якої служить імовірність працездатності системи протягом найбільш відповідальних відрізків часу. На етапах заходу на посадку вимоги до безперервності обслуговування складають  $10^{-5}$  . . . . .  $10^{-4}$  для проміжків часу від 15 до 150 с.

Основні навігаційні параметри, що визначаються в СРНС – дальність і радіальна швидкість. Відповідними їм радіонавігаційними параметрами (параметрами радіосигналу) служать затримка  $t$  сигналу і доплерівський зсув частоти  $f_{\text{доп}}$ . Оскільки головною вимогою до СРНС є висока точність виміру навігаційних параметрів, отже, й основною вимогою до радіосигналів так само є висока точність виміру затримки  $t$  сигналу і доплерівського зсуву частоти  $f_{\text{доп}}$ .

Вимоги до підвищення точності затримки сигналу і доплерівського зсуву частоти суперечливі. Для підвищення точності виміру затримки необхідно розширювати спектр сигналу, а для підвищення точності виміру доплерівського зсуву частоти – збільшувати тривалість сигналу.

Дане протиріччя вирішується при вирішенні задачі спільної оцінки  $t$  та  $f_{\text{доп}}$ .

Підвищення точності спільних оцінок затримки сигналу і доплерівського зсуву частоти можна досягти за рахунок збільшення так званої бази сигналу –  $B$  (добуток ефективної тривалості сигналу на ефективну ширину спектра сигналу) і основною вимогою до радіосигналів у СРНС є збільшення бази сигналу  $\gg 1$ . Такі сигнали називають шумоподібними. Відомо, що стійкість до перешкод радіотехнічної системи визначається значенням бази сигналу, а для більшості ЛА скритність і перешкодозахищеність є одним з визначальних вимог.

Інша істотна вимога — забезпечення багатостанційного доступу. При визначенні навігаційних параметрів у споживача повинна бути можливість одночасного доступу до сигналів від різних супутників. Проблема багатостанційного



доступу вирішується шляхом тимчасового, частотного або кодового поділу сигналів, наприклад, у супутниковій навігаційній системі GPS використовується кодовий поділ, у СРНС ГЛОНАСС - частотний.

З результатів аналізів стає очевидно, що не має принципової різниці між супутниковими навігаційними системами GPS та ГЛОНАСС.

В залежності від області використання апаратура споживача (АС) має свої особливості, тому виробники АС завжди вказують на область застосування відповідного зразка. Крім основних блоків, таких, як антена, приймач, індикатор, АС може містити допоміжні, що забезпечують виконання спеціальних сервісних функцій, наприклад, діагностику вузлів транспортного засобу, зв'язок з диспетчерським пунктом і т.п.

З огляду на, те що супутникова система навігації буде працювати в комплексі з інерціальною системою навігації, то навряд варто встановлювати на борт ЛА повний комплект супутникової системи. Досить обмежитися прийомо-індикатором і сигнальним процесором, думаючи, що алгоритми рішення навігаційної задачі будуть вирішуватися в спільному процесорі інерціально - супутникової системи навігації.

Виходячи з вищенаведеного, а також враховуючи умови застосування ЛА та вимоги ТЗ можна сформулювати вимоги, яким повинний задовольняти обраний тип прийомоіндикатора СРНС.

Розв'язувані задачі:

- автоматичне, безперервне, глобальне, всепогодне визначення поточних 3D-координат місця розташування, вектора шляхової швидкості шляхового кута ЛА при роботі: по сигналу стандартної точності частотного діапазону L1 ГЛОНАСС; по сигналі З/А-коду GPS; при спільній обробці вищевказаних сигналів;
- видача поточних 3D-координат місця розташування ЛА, що є складовими вектора швидкості і шляхового кута в системі координат СК-42 або ПЗ-90 у географічному форматі, а також ознак режиму роботи апаратури;
- стійке визначення навігаційних параметрів при русі з лінійними прискореннями і при стрибкоподібних змінах прискорення;
- можливість переключення з антени носія на антену ЛА;

- інтегральна оцінка очікуваної точності визначення поточних координат місця розташування;
- автоматичний вибір оптимального з погляду очікуваної точності сузір'я НС ГЛОНАСС і GPS при роботі в сполученому режимі;
- автоматичне рішення навігаційної задачі в географічній системі координат:

З огляду на зазначені вимоги можна запропонувати СНС типу — SPIRIT 24 Channel GPS+GLONASS Receiver DuoStar-1000 (рис.1.2.3). Приймач працює одночасно з системами GPS та ГЛОНАСС, добре себе зарекомендував в роботі на рухомих динамічних об'єктах (із значними прискореннями та різкими поштовхами), високим рівнем вібрацій та в широкому діапазоні температурних умов.

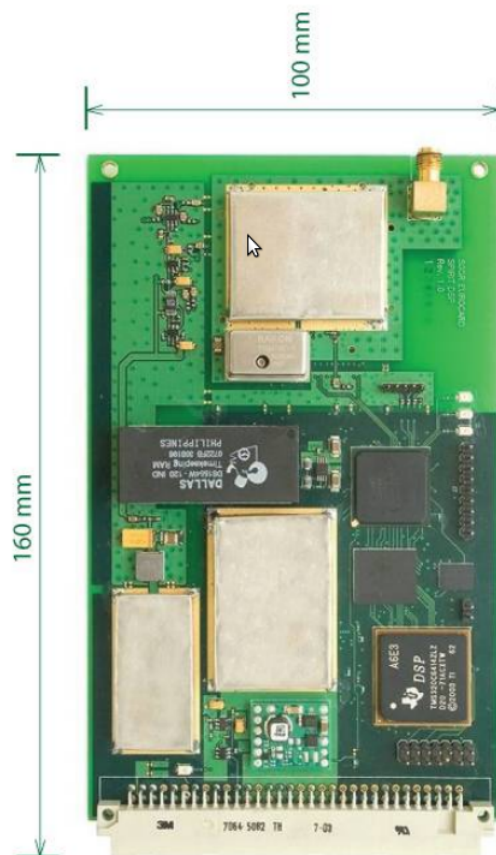


Рисунок 1.2.3 – СНС приймач SPIRIT DuoStar-1000

Комбінування GPS та ГЛОНАСС дає можливість використовувати практично до 56 супутників (32 GPS та 24 ГЛОНАСС). В цьому випадку приймач

подвійної системи може використовувати всі видимі супутники (до 24), що позитивно позначається на надійність та стабільність роботи в місцях з обмеженою видимістю.

Таблиця 1.2.1 – SPIRIT 24 Channel GPS+GLONASS Receiver DuoStar-1000

Параметр	Значення
Частоти	GPS L1 та ГЛОНАСС L1
Кількість каналі	24
Протоколи передачі	NMEA 0183 v.2.3, RTCM SC104 v.2.2
Позиційна точність	5 м
Точність визнач. часу	30 нс.
Точність визнач. швидкості	0.05 м/с
Динаміка	20g
Гарячий старт	1 с
Холодний старт	30 с
Частота вихідного сигналу	10 Гц

Наступним кроком є вибір інерціальної навігаційної системи, яка разом з СНС є ядром комплексної навігаційної системи. Тільки поєднання цих двох підсистеми, дає можливість задовольнити вимоги точності та надійності.

### 1.3 Вибір варіанту побудови інерціальної навігаційної системи

В інерціальній навігаційній системи (ІНС) інформацію про швидкість і координати одержують шляхом інтегрування сигналів, що відповідають прискоренням ЛА. Інформація про прискорення надходить від розташованих на борту ЛА акселерометрів. Процедура інтегрування векторних величин, якими є прискорення і швидкості ЛА, забезпечується шляхом відтворення (моделювання) на борті ЛА відповідної системи координат. З цією метою найчастіше використовують гіростабілізатори або гіроскопічні датчики кутової швидкості разом з обчислювачем.

Наявність похибок датчиків ІНС у свою чергу приводить до похибок у визначенні навігаційних координат руху ЛА, от чому при створенні ІНС намагаються зменшити величину похибок первинних датчиків. Перевагами інерціальних систем перед іншими системами навігації є їхня повна автономність, абсолютна перешкодозахищеність, а також висока інформативність. У залежності від способів розташування акселерометрів на ЛА розрізняють платформні і безплатформні ІНС. У першому випадку акселерометри встановлюються на гіростабілізуючій платформі, у другому безпосередньо на корпусі ЛА або в спеціальному блоці чуттєвих елементів, при цьому осі чутливості акселерометрів не змінюють орієнтацію відносно напрямку осей, зв'язаних з ЛА. Серед платформних ІНС розрізняють ІНС з некоректованою платформою та ІНС з горизонтальною платформою.

У ІНС з некоректованою платформою осі платформи, а також акселерометри, що установлені на цій платформі, не обертаються в інерціальному просторі. ІНС з горизонтальною платформою у свою чергу класифікують як ІНС із вільною в азимуті платформою (платформа розташовується відносно точки світового простору – відносно зірки) та ІНС з корегованою в азимуті платформою (платформа стабілізується відносно меридіана – „направлена” на північ). По ролі обчислювача у визначенні кутових і лінійних координат прийнято розрізняти геометричні, напіваналітичні та аналітичні ІНС. У геометричних ІНС основним елементом служить гіростабілізатор, що відтворює напрямок осей інерціальної системи відліку, і платформа з акселерометрами, осі чутливості яких відтворюють деякі напрямки в площині обрію і напрямок місцевої вертикалі. Роль обчислювача в такій ІНС мінімальна і зводиться до забезпечення корекції заданого положення платформи. Інформація про координати знімається з кутомірних пристроїв гіростабілізатора і платформи.

До напіваналітичних систем відносять системи з горизонтальною платформою. У цих системах гіроплатформа з акселерометрами відтворює напрямок нормальної (рухливої) системи відліку. З кутомірних пристроїв гіростабілізатора знімається інформація про кути крену, тангажу, курсу ЛА. Обчислювач ІНС вирішує задачу визначення кінематичних параметрів руху центра мас ЛА і видає сигнали для корекції гіростабілізатора. До аналітичних ІНС відносять безплатформні ІНС та ІНС з акселерометрами на некоректованому або вільному гіростабілізаторі. Обчислювач ІНС у даному випадку виконує найбільший

обсяг обчислень. Крім визначення кінематичних параметрів руху центра мас ЛА він визначає кутову орієнтацію нормальної рухливої системи координат відносно інерціальної і кутову орієнтацію зв'язаної рухливої системи координат щодо нормальної.

Побудова прецизійних і одночасно надійних гіроплатформ являє собою складну технічну задачу. Тому останнім часом усе більше уваги приділяється розробці так званих безплатформних ІНС (БІНС), у яких датчики акселерометрів жорстко зв'язані з корпусом ЛА. Такі системи мають у своєму складі гіроскопічні прилади, але головною задачею цих пристроїв є забезпечення обчислювачів БІНС інформацією про кутове положення ЛА, а так само про положення осей чутливості акселерометрів відносно обраної навігаційної системи координат. Відсутність горизонтальної платформи вимагає виділяти з показань акселерометрів сигнали, що є прискореннями ЛА, тобто обчислювачі БІНС аналітично визначають напрямок вертикалі. При цьому точність зазначеного моделювання визначається точністю роботи обчислювача і, природно, точністю датчиків первинної навігаційної інформації. До числа потенційних переваг безплатформних інерціальних навігаційних систем БІНС у порівнянні з платформними ІНС можна віднести:

- менші розміри, вага й енергоємність;
- істотне спрощення механічної частини системи ;
- відсутність обмежень по кутах розвороту;
- скорочення часу початкової виставки.

Тому, навіть за певних труднощів, що виникають при створенні БІНС, таких як:

- розробка датчиків інформації із широким діапазоном вимірів і прийнятною точністю в умовах їхнього твердого кріплення на борті ЛА;
- розробка БЦВМ, що мають достатню швидкодію.

У роботі розглядатиметься безплатформна інерціальна система. В залежності від способу визначення кутового положення об'єкта в інерціальному просторі можливі наступні основні варіанти схеми БІНС:

Перший варіант передбачає наявність у БІНС шести акселерометрів рознесених по осям об'єкта на відстань (для виміру кутових прискорень) і обчислювального пристрою (ОП);

Другий варіант включає три лінійних акселерометри і три вимірники кутової швидкості руху об'єкта щодо центра мас, встановлених в центрі мас об'єкта, а також ОП.

Третій варіант передбачає наявність трьох лінійних акселерометрів, і вимірника кутового положення об'єкта в інерціальному просторі, встановлених у центрі мас об'єкта, і ОП.

Стосовно розглянутого класу ЛА використання БІНС першого варіанту зустрічає складності реалізації через малу вимірювальну базу визначення кутових прискорень об'єкта за допомогою акселерометрів. До того ж, похибки БІНС цього варіанту у визначенні координати, обумовлені помилками виміру кутових прискорень, має три складових: одна з них постійна, інша наростає пропорційно квадратові часу руху, а третя змінюється з періодом Шулера. Звідси ясно, що цей варіант схеми може бути застосований тільки при досить точних акселерометрах і для об'єктів, що здійснюють політ протягом нетривалого часу.

Реалізація третього варіанта БІНС припускає наявність у складі навігаційної системи триступеневого гіроскопічного вимірника кутових положень (електростатичні гіроскопи, гіроскопи, що динамічно з'являються у великій кількості) – досить дорогі прецизійні прилади.

За результатами аналізу можна зробити висновок, що в даній роботі доцільно використовувати БІНС, що побудована на трьох акселерометрах і трьох вимірниках кутової швидкості, тобто БІНС другого класу за вище приведеною класифікацією. Найбільш поширеними й перспективними у використанні в якості чутливих елементів є лазерні кільцеві гіроскопи.

Під польотним калібруванням розуміють метод підвищення роботи БІНС шляхом оцінки у польоті систематичних складових похибок БІНС та їх компенсації. Для виконання такої оцінки необхідно порівнювати вихідну інформацію БІНС з еталонною навігаційною інформацією і, маючи модель помилок БІНС, виконати оцінку параметрів цієї моделі за різницею між вихідною інформацією БІНС та еталонною інформацією.

З урахуванням того, що БІНС працює у складі комплексної ІССН необхідно обрати спільну навігаційну систему координат (СК) й для обраної СК

розробити алгоритми розв'язку кінематичних рівнянь числення навігаційних параметрів. З урахуванням того, що СНС частіше за все працює в географічній системі координат алгоритми роботи БІНС також слід формувати в цієї системі координат.

Алгоритм функціонування БІНС містить у собі сукупність аналітичних залежностей, які дозволяють за вимірювань алгоритми роботи функціонування БІНСним значенням уявного прискорення й абсолютної кутової швидкості ЛА безперервно визначати поточне значення координат місця розташування, складові шляхової швидкості та кутове положення ЛА в обраній навігаційній системі координат.

В алгоритмах роботи трикомпонентної БІНС, як і в алгоритмах платформної ІНС, точність зчислення навігаційних параметрів досягається за рахунок виключення із сигналів уявного прискорення, яке вимірюють акселерометри, складові прискорення сили ваги і коріолісового прискорення. Але вплив цих складових компенсується на відміну від платформної ІНС тільки аналітично.

Кінематичні рівняння інерціальної навігації в основному визначаються вибраною системою координат, тобто навігаційним базисом, в якому визначаються навігаційні параметри (координати і проекції швидкості). У свою чергу, вибір навігаційного базису залежить від типу літального апарата, особливостей його траєкторного руху, характеру розв'язуваних задач.

Наприклад, для БІНС, що інтегруються зі супутниковими навігаційними системами, можна застосовувати інерціальну систему координат, яка використовується супутниковою системою навігації. При цьому, позиційну інформацію одержують у формі декартових прямокутних координат, швидкісну – у формі проекцій абсолютної швидкості на осі вибраної інерціальної системи координат, а інформацію про кутову орієнтацію – у вигляді відповідної матриці або трьох кутів орієнтації ЛА відносно вибраного базису. Подальше перерахування отриманих координат в обертову систему координат ПЗ-90 (WGS-84) здійснюється за алгоритмами супутникової системи навігації.

Для БІНС літальних апаратів, які здійснюють рух в атмосфері Землі, найбільш часто використовуються обертові системи координат з базовою площиною місцевого горизонту і певною орієнтацією горизонтальних осей в азимуті. Під орієнтацією осей в азимуті розуміється можливість їхньої орієнтації,

наприклад, за сторонами світу, коли дві горизонтальні осі спрямовані в східному і північному напрямках. При цьому позиційну інформацію визначають широтою  $\varphi$ , довготою  $\lambda$  і висотою  $h$ , що виміряні на еліпсоїді Красовського або на еліпсоїді міжнародної системи WGS-84, швидкість визначають проекціями на східну  $V_E$ , північну  $V_N$  і вертикальну осі  $V_H$ , якщо за навігаційну систему вибрана система з орієнтацією осей за сторонами світу, або проекціями на осі горизонтального базису з іншою орієнтацією. Орієнтація при цьому визначається кутами крену, тангажа і справжнього курсу.

Типову схему побудови БІНС зображено на рис.1.3.1. Цей варіант реалізує алгоритм системи, яка працює в обертовій земній системі координат.

Датчики первинної інформації БІНС – датчики кутової швидкості й акселерометри встановлюються жорстко на ЛА. Складні умови роботи датчиків інформації призводять до появи значних похибок, тому в алгоритмах роботи БІНС бажано здійснити аналітичну компенсацію похибок вимірників (здійснювати їх польотне калібрування), перш ніж ці сигнали будуть використані для розрахунку параметрів орієнтації і для визначення складових уявного прискорення уздовж навігаційних осей.

Для корекції показань датчиків первинної інформації необхідна математична модель вимірника, в якій, зазвичай, враховують: нелінійність; неспіввідповідність осей датчиків; дрейф; викривлення масштабного коефіцієнта. Сигнали

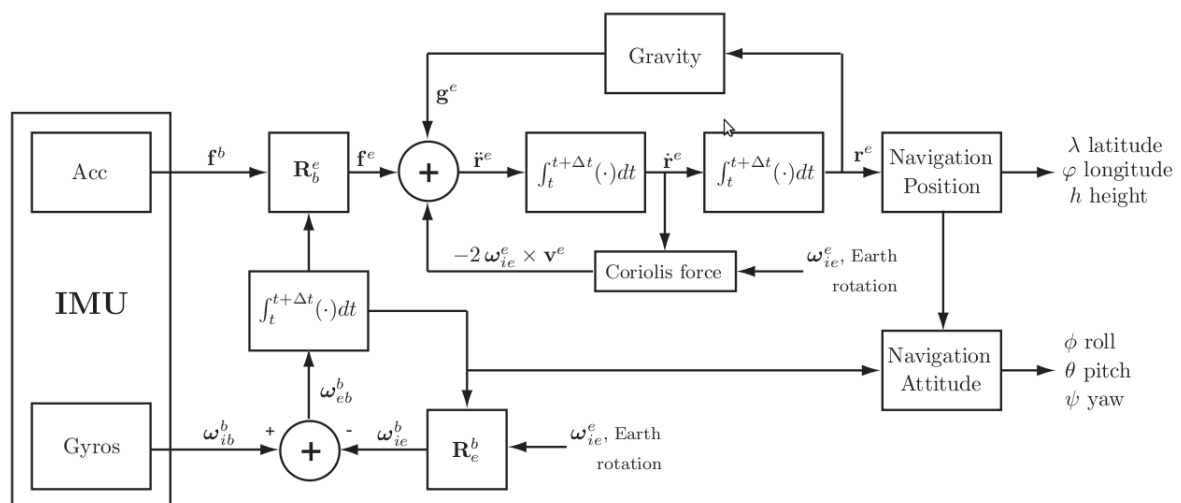


Рисунок 1.3.1 – Алгоритм роботи БІНС

$\omega_{x,y,z}$  з виходу аналітичного компенсатора похибок використовуються для обчислення параметрів матриці напрямних косинусів  $B$ , яка визначає зв'язок між



двома системами координат. Оскільки матриця напрямних косинусів  $B$  визначається між зв'язаними з ЛА осями й осями обертової навігаційної системи координат, то при розрахунках параметрів матриці  $B$  необхідно залучити обчислені проекції вектора кутової швидкості навігаційної системи координат, що відображено на схемі додатковими зв'язками, які враховують кутову швидкість, що виникає при обльоті сферичної Землі ( $\dot{\lambda}, \dot{h}, \dot{\varphi}$ , і кутову швидкість обертання самої Землі ( $\Omega_3$ ).

Перетворення складових уявного прискорення  $a_{x,y,z}$  від осей ЛА до осей навігаційної системи координат здійснюється за допомогою матриці напрямних косинусів  $B$ . Навігаційний обчислювач вирішує задачі, властиві всім платформним системам, оскільки на вході цього обчислювача сформовані проекції уявного прискорення на осі навігаційної системи координат і нічого принципово нового в розв'язанні цієї задачі немає. На виході БІНС формуються радіус-вектор місця розташування ЛА, вектор швидкості, а також кути орієнтації ЛА.

В окремому випадку, коли за навігаційний базис вибраний горизонтальний орієнтований за сторонами світу тригранник, на виході системи будуть сформовані географічні (геодезичні) координати радіуса-вектора місця розташування  $\lambda, \varphi, H$ , проекції відносної швидкості руху  $V_N, V_E, V_H$ , а також кути орієнтації ЛА в географічній системі координат – справжній курс  $\psi$ , тангаж  $\vartheta$  і крен  $\gamma$ .

Обсяг обчислень у БІНС значний. Це пояснюється в основному тим фактом, що БЦОМ розв'язує задачі, які пов'язані з динамікою обертання ЛА, а також з динамікою поступального руху ЛА. Поступальні швидкості ЛА відносно малі. Наприклад, швидкість при польоті ЛА в напрямку на північ 1100 км/год відповідає швидкості зміни широти усього на 10 град/год.

Таким чином, інтегрування для одержання швидкості і місця розташування можуть здійснюватися досить точно з використанням дуже простих методів чисельного інтегрування при низькій частоті повторення в типовому випадку 10...20 Гц .

Кутові швидкості ЛА в типовому випадку за величиною на кілька порядків більші поступальних швидкостей. Зокрема, для маневрених ЛА кутові швидкості обертання можуть складати сотні градусів за секунду. В результаті цього інтегрування кутового положення в БІНС зв'язано з жорсткими вимогами до БЦОМ.

Оскільки для забезпечення високої точності інерціальної навігації потрібно, щоб похибки інтегрування кутового положення обмежувалися декількома частками кутової хвилини, необхідно застосовувати алгоритми інтегрування більш високого порядку при типових частотах повторення 80...50 Гц.

З огляду на вище сказане, наведемо варіант побудови алгоритмів БІНС для випадку, коли за навігаційний базис вибраний горизонтальний орієнтований за сторонами світу тригранник.

### Алгоритми БІНС, яка працює в географічній системі координат

За навігаційний тригранник візьмемо тригранник  $NHE$ , зв'язаний з землею поверхнею. Виберемо наступний напрямок осей  $NHE$  (рис. 1.3.2):

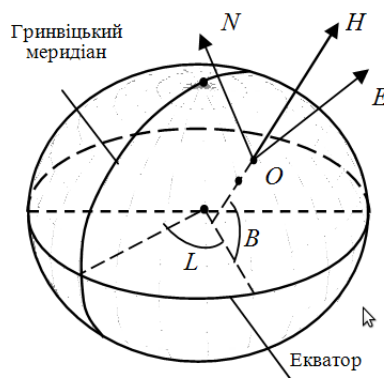


Рисунок 1.3.2 – Системи координат

$OH$  – збігається з вертикаллю;

$ON$  – дотична до меридіана;

$OE$  – утворює праву трійку.

В алгоритмах БІНС, зазвичай, виділяють динамічні та кінематичні рівняння. Динамічні рівняння реалізують трикомпонентну схему БІНС, у якій географічні координати  $\lambda, \varphi, H$  визначаються інтегруванням рівнянь вигляду

$$\begin{aligned}\dot{\lambda} &= \frac{V_E}{(R_2+H) \cos \varphi}; \\ \dot{\varphi} &= \frac{V_N}{R_1+H}; \\ \dot{H} &= V_H,\end{aligned}$$

де  $V_N, V_E$  – північна та східна проекції шляхової швидкості (проекції на осі системи координат  $NHE$  (див. рис. 1.3.2);

$R_1, R_2$  – два радіуси кривизни земного сфероїда (еліпсоїда обертання);  
 $R_1$  – радіус кривизни меридіонального перетину еліпсоїда (площиною  $HN$ );  
 $R_2$  – радіус кривизни перетину еліпсоїда площиною  $HE$  (площиною першого вертикала);

$$R_1 = \frac{a(1 - e^2)}{(1 - e^2 \sin(\varphi)^2)^{\frac{3}{2}}}; R_2 = \frac{a}{\sqrt{1 - e^2 \sin(\varphi)^2}}.$$

де  $a$  — велика піввісь еліпсоїда ( $a = 6378388$  м);

$e$  — ексцентриситет еліпсоїда ( $e^2 = 6,73 \cdot 10^{-3}$ );

$H$  — висота польоту.

Тут можна застосовувати такі ж спрощення, що й у платформних інерціальних системах. Зокрема, функції  $\frac{1}{R_1+H}$  та  $\frac{1}{R_2+H}$  з точністю до членів порядку малості  $10^{-5}$  можна представити в наступному вигляді:

$$\begin{aligned} \frac{1}{R_1+H} &= \frac{1}{a} [1 - e^2 - \frac{H}{a} - \frac{3}{2}e^2 \sin^2 \varphi - 2e^2 \frac{H}{a} + 3e^2 \frac{H}{a} \sin^2 \varphi + (\frac{H}{a})^2 + \\ &+ e^4 (1 - 3 \sin^2 B + \frac{3}{8} \sin^4 \varphi)]; \\ \frac{1}{R_2+H} &= \frac{1}{a} [1 - \frac{H}{a} - \frac{1}{2}e^2 \sin^2 \varphi + (\frac{H}{a})^2 + e^2 \frac{H}{a} \sin^2 \varphi + \\ &+ e^4 (\frac{1}{4} \sin^2 \varphi - \frac{3}{8} \sin^2 \varphi)] \end{aligned}$$

Якщо у формулах  $\frac{1}{R_1+H}$  та  $\frac{1}{R_2+H}$  зберегти лише члени порядку малості  $10^{-2}$ , то вони приймуть вигляд

$$\begin{aligned} \frac{1}{R_1+H} &\approx \frac{1}{a} [1 - e^2 - \frac{H}{a} - \frac{3}{2}e^2 \sin(B)^2]; \\ \frac{1}{R_2+H} &\approx \frac{1}{a} [1 - \frac{H}{a} - \frac{1}{2}e^2 \sin(\varphi)^2]. \end{aligned} \quad (1.3.1)$$

Слід відзначити, що використання спрощень (1.3.1) може призвести до похибок, порівняних з похибками високоякісних гіроскопічних вимірників, які використовуються в БІНС.

Складові шляхової швидкості ЛА  $V_E$ ,  $V_N$ ,  $V_H$  одержують в результаті інтегрування проекцій сигналів акселерометрів, виключаючи із них складові коріолісового прискорення і прискорення сили ваги:

$$\begin{aligned} \dot{V}_E &= a_E - (V_N \omega_{H_\Sigma} - V_H \omega_{N_\Sigma}) + g_E; \\ \dot{V}_H &= a_H - (V_E \omega_{N_\Sigma} - V_N \omega_{E_\Sigma}) + g_H; \\ \dot{V}_N &= a_N - (V_H \omega_{E_\Sigma} - V_E \omega_{H_\Sigma}) + g_N, \end{aligned} \quad (1.3.2)$$

де  $a_{E,H,N}$  — проекції уявного прискорення ЛА, вимірювані акселерометрами, на осі навігаційного тригранника;

$g_{E,H,N}$  — проекції вектора прискорення сили ваги, які враховують прискорення земного тяжіння, і прискорення, що викликається відцентровою силою інерції і зв'язане з обертанням Землі;

складові в дужках — проекції коріолісового прискорення на осі навігаційного тригранника;

$\omega_{E\Sigma}, \omega_{H\Sigma}, \omega_{N\Sigma}$  — проекції кутової швидкості навігаційного тригранника відносно інерціального простору, які враховують проекції кутової швидкості обертання Землі  $\Omega_E, \Omega_H, \Omega_N$  і складові відносної кутової швидкості навігаційного тригранника, які обумовлені рухом ЛА відносно Землі  $\omega_{E_V}, \omega_{H_V}, \omega_{N_V}$ :

$$\omega_{N\Sigma} = \omega_{N_V} + 2\Omega_N; \omega_{H\Sigma} = \omega_{H_V} + 2\Omega_H; \omega_{E\Sigma} = \omega_{E_V} + 2\Omega_E.$$

У свою чергу, складові відносної кутової швидкості навігаційного тригранника і швидкості обертання Землі визначаються співвідношеннями

$$\begin{aligned}\omega_{E_V} &= -\frac{V_N}{R_1+H} = -\dot{\varphi}; \\ \omega_{H_V} &= \frac{V_E}{(R_2+H)} \operatorname{tg} \varphi = \dot{\lambda} \sin \varphi; \\ \omega_{N_V} &= \frac{V_E}{(R_2+H)} = \dot{\lambda} \cos \varphi;\end{aligned}$$

$$\Omega_N = \Omega_3 \cos \varphi; \Omega_H = \Omega_3 \sin \varphi; \Omega_E = 0,$$

де  $\Omega_3$  — кутова швидкість обертання Землі ( $\Omega_3 = 7,27 \cdot 10^{-5}$  рад/с).

Детермінована математична модель прискорення сили ваги існує тільки для нормальної складової поля сили ваги, що відповідає земному еліпсоїду з рівномірним розподілом мас в об'ємі цієї фігури. Градієнт цього поля в будь-якій точці, що належить поверхні еліпсоїда, спрямований за нормаллю до неї і розташований у площині меридіонального перетину. Оскільки точка місцеположення ЛА не належить поверхні Землі, то вектор градієнта нормального поля сили ваги  $\bar{g}$  в цій точці не буде спрямований за лінією нормалі, опущеної з неї до поверхні земного еліпсоїда (вісь  $ОН$ ). Разом з тим, цей вектор буде розташований у площині меридіана точки  $O$ , тобто в площині  $НОН$ . Тоді, використовуючи потенційну функцію нормального поля тяжіння земного сфероїда, з точністю до членів порядку малості  $10^{-5}$  співвідношення для проекцій складових поля

сили ваги  $\bar{g}$  мають такий вигляд:

$$\begin{aligned} g_E &= 0; \\ g_N &= \frac{1}{2}g\left[\frac{H}{a}(e^2 - 5q) + qe^2 \sin^2 \varphi\right] \sin^2 \varphi; \\ g_H &= -g\left\{1 - 2\frac{H}{a} - (e^2 + 2q - 3\frac{H}{a})\frac{H}{a} + \left[\frac{1}{2}(5q - e^2) - \frac{1}{8}e^4 + \frac{17}{18}qe^2 + \right. \right. \\ &\quad \left. \left. + (3e^2 - 5q)\frac{H}{a}\right] \sin^2 B - \frac{1}{2}qe^2 \sin^4 \varphi + \frac{1}{16}e^2(\frac{1}{2}e^2 - 7q) \sin^2 2\varphi\right\}, \end{aligned}$$

де  $g = 9,78049\text{м/с}^2$  прискорення сили ваги на екваторі;

$q = \Omega_3^2 a/g = 0,00346775$  — відношення відцентрової сили, обумовленої обертанням Землі, до сили ваги на екваторі.

З точністю до величин порядку малості  $10^{-4}$  співвідношення для проєкцій складових поля сили ваги  $\bar{g}$  декілька спрощуються:

$$\begin{aligned} g_E &= 0; \\ g_N &= g \sin 2\varphi + \frac{5}{2}q \sin^2 B \frac{H}{a} (\frac{e^2}{2} - 2q); \\ g_H &= -g \left[ 1 - \frac{e^2}{2} \sin^2 \varphi + \frac{3}{2}q \sin^2 \varphi + e^4 (-\frac{1}{8} \sin^2 \varphi + \frac{1}{32} \sin^2 2\varphi) + \right. \\ &\quad \left. + e^2 q (-\frac{17}{28} \sin^2 \varphi - \frac{5}{16} \sin^2 2\varphi) + \frac{H}{a} e^2 (3 \sin^2 \varphi - 1) + \right. \\ &\quad \left. + \frac{Hq}{a} (-1 - 6 \sin^2 \varphi) - 2\frac{H}{a} + 3\frac{H^2}{a^2} \right], \end{aligned}$$

а при малих значеннях висоти ( $\leq 100$  км) проєкції вектора  $\bar{g}$  на осі  $NHE$ , якщо в них зберегти лише члени порядку малості  $10^{-2}$ , взагалі мають простий вигляд:

$$\begin{aligned} g_E &= 0; \\ g_N &= 0; \\ g_H &= -g(1 + 5,2884 \cdot 10^{-3} \sin^2 \varphi) \left[ 1 - \frac{2H}{a} (1 - e \sin^2 \varphi) \right] \end{aligned}$$

Є й інші форми запису даної складової.

При розв'язанні кінематичних рівнянь розраховуються проєкції  $a_{E,H,N}$  уявного прискорення ЛА на осі навігаційного тригранника  $NHE$  за показаннями акселерометрів зі зв'язаної з ЛА системи координат  $XYZ$  з використанням матриці напрямних косинусів **B**

$$\begin{bmatrix} a_N \\ a_H \\ a_E \end{bmatrix} = B \begin{bmatrix} a_{x_{\text{ЛА}}} \\ a_{y_{\text{ЛА}}} \\ a_{z_{\text{ЛА}}} \end{bmatrix}$$

Матриця напрямних косинусів має такий вигляд:

$$B = \left[ \begin{array}{c|c|c} \cos \psi \cos \vartheta & \sin \psi \sin \gamma - \cos \psi \sin \vartheta \cos \gamma & \sin \psi \cos \gamma + \sin \gamma \sin \vartheta \cos \psi \\ \hline \sin \vartheta & \cos \vartheta \cos \gamma & -\cos \vartheta \sin \gamma \\ \hline -\sin \psi \cos \vartheta & \cos \psi \sin \gamma + \sin \psi \sin \vartheta \cos \gamma & \cos \psi \cos \gamma - \sin \psi \sin \vartheta \sin \gamma \end{array} \right]$$

де  $\gamma$ ,  $\vartheta$ ,  $\psi$  – кути крену, тангажа і рискання. Кут рискання відрізняється від географічного курсу  $\psi_{\Gamma}$  знаком, тобто  $\psi_{\Gamma} = \psi$ .

Матриця напрямних косинусів **B** може бути отримана в різні способи. Наведемо приклади деяких з них.

Знайти матрицю **B** можна в результаті розв’язання узагальненого рівняння Пуассона за інформацією про кутову швидкість ЛА відносно інерціального простору  $\omega_{\text{ЛА}}$  і кутову швидкість навігаційної системи координат відносно інерціального простору  $\omega_{NHE}$ , яка враховує кутову швидкість обертання Землі і кутову швидкість, обумовлену обльотом ЛА сферичної Землі

$$\dot{B} = B\omega_{\text{ЛА}} - \omega_{NHE}B$$

де

$$\omega_{\text{ЛА}} = \begin{bmatrix} 0 & -\omega_{z_{\text{ЛА}}} & \omega_{y_{\text{ЛА}}} \\ \omega_{z_{\text{ЛА}}} & 0 & -\omega_{x_{\text{ЛА}}} \\ -\omega_{y_{\text{ЛА}}} & \omega_{x_{\text{ЛА}}} & 0 \end{bmatrix};$$

$$\omega_{NHE} = \begin{bmatrix} 0 & -(\omega_{E_V} + \Omega_E) & (\omega_{H_V} + \Omega_H) \\ (\omega_{E_V} + \Omega_E) & 0 & -(\omega_{N_V} + \Omega_N) \\ -(\omega_{H_V} + \Omega_H) & (\omega_{N_V} + \Omega_N) & 0 \end{bmatrix};$$

$\omega_{x_{\text{ЛА}}}$ ,  $\omega_{y_{\text{ЛА}}}$ ,  $\omega_{z_{\text{ЛА}}}$  – кутові швидкості ЛА відносно зв’язаних осей, вимірювані датчиками кутової швидкості;

$\omega_{E_V}$ ,  $\omega_{H_V}$ ,  $\omega_{N_V}$  були визначені раніше.

За елементами матриці **B** визначаються кути орієнтації ЛА: крен  $\gamma$ , тангаж  $\vartheta$  рискання (курс)  $\psi$ :

$$\begin{aligned} \gamma &= \arctg\left(\frac{-b_{23}}{b_{22}}\right) = \arcsin\left(\frac{-b_{23}}{\sqrt{1-b_{21}^2}}\right) = \arccos\left(\frac{b_{22}}{\sqrt{1-b_{21}^2}}\right); \\ \vartheta &= \arctg\left(\frac{b_{21}}{\sqrt{b_{22}^2+b_{33}^2}}\right) = \arcsin(b_{21}) = \arccos\left(\sqrt{1-b_{21}^2}\right) \\ \psi &= -\arctg\left(\frac{b_{31}}{b_{11}}\right) = \arcsin\left(\frac{-b_{31}}{\sqrt{1-b_{21}^2}}\right) = \arccos\left(\frac{b_{11}}{\sqrt{1-b_{21}^2}}\right). \end{aligned} \quad (1.3.3)$$

Інший алгоритм отримання матриці напрямних косинусів припускає її формування безпосередньо за кутами  $\gamma, \vartheta, \psi$ . Кінематичні співвідношення між кутами

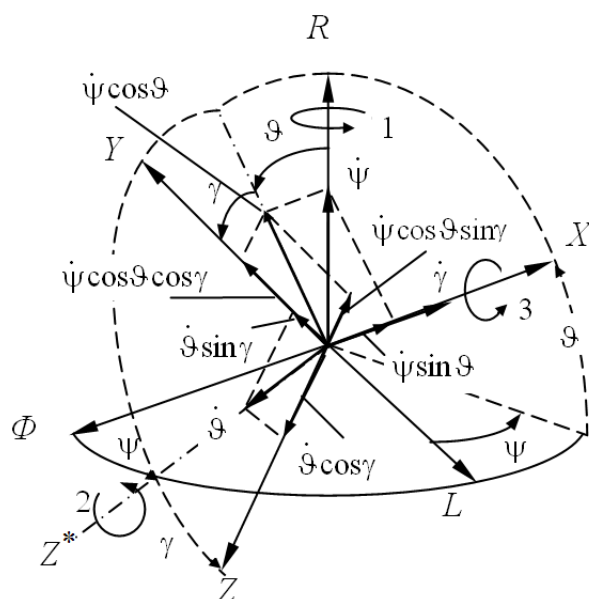


Рисунок 1.3.3 – Кінематичні співвідношення між кутами

$\gamma, \vartheta, \psi$  і проекціями вектора абсолютної кутової швидкості на осі зв'язаної системи координат  $\omega_{x\Sigma}, \omega_{y\Sigma}, \omega_{z\Sigma}$  можна одержати з рис. 1.3.3, на якому показано перетворення навігаційної системи координат  $OLR\Phi$  у зв'язану  $OXYZ$  шляхом трьох поворотів: 1 навколо осі  $OR$ ; 2 навколо проміжної осі  $OZ^*$ ; 3 навколо осі  $OX$ .

Звичайно, що кутові швидкості  $\dot{\psi}, \dot{\vartheta}, \dot{\gamma}$ , які спрямовані уздовж відповідних осей, є складовими абсолютної кутової швидкості ЛА.

Проектуючи  $\dot{\psi}, \dot{\vartheta}, \dot{\gamma}$  на осі зв'язаної системи координат, отримаємо:

$$\begin{aligned}\omega_{x\Sigma} &= \dot{\gamma} + \dot{\psi} \sin \vartheta; \\ \omega_{y\Sigma} &= \dot{\vartheta} \sin \gamma + \dot{\psi} \cos \vartheta \cos \gamma; \\ \omega_{z\Sigma} &= \dot{\vartheta} \cos \gamma - \dot{\psi} \cos \vartheta \sin \gamma.\end{aligned}$$

Розв'язуючи ці співвідношення, одержимо такі кінематичні рівняння:

$$\begin{aligned}\dot{\psi} &= (\omega_{y\Sigma} \cos \gamma - \omega_{z\Sigma} \sin \gamma) \sec \vartheta; \\ \dot{\gamma} &= \omega_{x\Sigma} + \tan \vartheta (\omega_{z\Sigma} \sin \gamma - \omega_{y\Sigma} \cos \gamma); \\ \dot{\vartheta} &= \omega_{y\Sigma} \sin \gamma + \omega_{z\Sigma} \cos \gamma.\end{aligned}$$

У свою чергу

$$\omega_{y\Sigma} = \omega_{y_{\text{ЛА}}} - \omega_{y_{\text{NHE}}};$$

$$\omega_{x\Sigma} = \omega_{x_{\text{ЛА}}} - \omega_{x_{\text{NHE}}};$$

$$\omega_{z\Sigma} = \omega_{z_{\text{ЛА}}} - \omega_{z_{\text{NHE}}}.$$

де  $\omega_{y_{\text{ЛА}}}, \omega_{x_{\text{ЛА}}}, \omega_{z_{\text{ЛА}}}$  – проекції кутової швидкості ЛА відносно інерціального простору на осі зв'язаної системи координат, вимірювані датчиками кутових швидкостей;

$\omega_{y_{\text{NHE}}}, \omega_{x_{\text{NHE}}}, \omega_{z_{\text{NHE}}}$  – проекції кутової швидкості навігаційного тригранника відносно інерціального простору на осі зв'язаної системи координат, які враховують проекції кутової швидкості обертання Землі  $\Omega_H, \Omega_E, \Omega_N$  і складові відносної кутової швидкості навігаційного тригранника, що обумовлені рухом ЛА відносно Землі  $\omega_{H_V}, \omega_{E_V}, \omega_{N_V}$ .

Ці проекції кутової швидкості визначаються в результаті розв'язання матричного рівняння

$$\begin{bmatrix} \omega_{x_{\text{NHE}}} \\ \omega_{y_{\text{NHE}}} \\ \omega_{z_{\text{NHE}}} \end{bmatrix} = B^T \begin{bmatrix} \omega_{N_V} + \Omega_N \\ \omega_{H_V} + \Omega_H \\ \omega_{E_V} + \Omega_E \end{bmatrix}.$$

Перевагою такого підходу до визначення кутів орієнтації ЛА (інтегруванням диференціальних рівнянь, що описують швидкості зміни кутів Ейлера, а не за арктангенсами відношення елементів матриці напрямних косинусів) є відсутність обмежень  $\gamma \pm 90^\circ$ , що особливо важливо при визначенні курсу ЛА на віражах.

Тривимірні матриці напрямних косинусів досить зручні для обчислень у бортовій ЦОМ. Однак формування матриці **B** з використанням тригонометричних функцій вимагає значних обчислювальних витрат.

Для визначення орієнтації ЛА можна використовувати не тільки напрямні косинуси, але і параметри Родрига-Гамільтона у формі кватерніонів. Достоїнство методу кватерніонів полягає в тому, що він дозволяє описувати перехід від однієї системи координат до іншої за допомогою всього лише чотирьох чисел, а не 9 напрямних косинусів.

Кватерніонний метод ґрунтується на теоремі Ейлера, яка доводить, що будь-який поворот однієї системи координат відносно іншої можна подати, як поворот на деякий кут навколо однієї нерухомої осі.



Кватерніон є компактною формою запису орієнтації зазначеної осі (векторна частина кватерніона  $\lambda_1, \lambda_2, \lambda_3$ ) і кута повороту (скалярна частина кватерніона  $\lambda_0$ ) відповідно до теореми Ейлера.

Застосування кватерніонів дозволяє подати ортогональні перетворення у формі множення кватерніонів. Дії над кватерніонами допускають матричні операції з використанням симетризованих матриць, що дуже зручно при створенні програм бортових обчислювачів.

Відповідно до теореми Ейлера-Шаля усяке переміщення твердого тіла, яке має нерухому точку, можна зобразити як результат повороту навколо незмінного напрямку (ейлерової осі) на певний кут  $\varphi$ . Якщо зв'язати з розглянутим твердим тілом правий ортогональний координатний тригранник, то параметри Родрига-Гамільтона  $\lambda_0, \lambda_1, \lambda_2, \lambda_3$ , що однозначно характеризують згадані переміщення, можна задати такими виразами:

$$\lambda_1 = \frac{l_1 \sin \varphi}{2}; \lambda_2 = \frac{l_2 \sin \varphi}{2}; \lambda_3 = \frac{l_3 \sin \varphi}{2}; \lambda_0 = \frac{\cos \varphi}{2},$$

де  $l_1, l_2, l_3$  — косинуси кутів, утворених ейлеровою віссю з осями тригранника в його вихідному та кінцевому положенні.

Зв'яжемо з ЛА, на якому встановлена БІНС, ортонормований базис **Е** — праву трійку взаємно ортогональних одиничних векторів  $e_1, e_2, e_3$ . Орієнтацію базису **Е** відносно ортонормованого інерціального базису **І**, складеного з ортів  $i_1, i_2, i_3$ , охарактеризуємо параметрами Родрига-Гамільтона  $\lambda_0, \lambda_1, \lambda_2, \lambda_3$ . Матриця напрямних косинусів, що обчислена за параметрами Родрига-Гамільтона (кватерніонами), має такий вигляд:

$$B = \begin{bmatrix} 1 - 2(\lambda_2^2 + \lambda_3^2) & 2(\lambda_1\lambda_2 - \lambda_0\lambda_3) & 2(\lambda_1\lambda_3 + \lambda_0\lambda_2) \\ 2(\lambda_1\lambda_2 + \lambda_0\lambda_3) & 1 - 2(\lambda_1^2 + \lambda_3^2) & 2(\lambda_2\lambda_3 - \lambda_0\lambda_1) \\ 2(\lambda_1\lambda_3 - \lambda_0\lambda_2) & 2(\lambda_2\lambda_3 + \lambda_0\lambda_1) & 1 - 2(\lambda_1^2 + \lambda_2^2) \end{bmatrix}.$$

Вимірники кутової швидкості, що входять до складу БІНС, вимірюють координати  $\omega_x, \omega_y, \omega_z$  вектора  $\bar{\Omega}$  абсолютної кутової швидкості базису **Е**, що задані в цьому базисі. Необхідно, знаючи значення параметрів Родрига-Гамільтона в момент часу  $t = t_0$  і використовуючи сигнали вимірників кутової швидкості, обчислювати параметри Родрига-Гамільтона при  $t > t_0$ . У початковий момент часу за інформацією про кути крену тангажа і курсу можна розрахувати вихі-

дні значення параметрів Родрига-Гамільтона:

$$\begin{aligned}\lambda_{0_0} &= \sin(\gamma_0/2) \sin(\vartheta_0/2) \sin(\psi_0/2) + \cos(\gamma_0/2) \cos(\vartheta_0/2) \cos(\psi_0/2); \\ \lambda_{1_0} &= -\sin(\vartheta_0/2) \sin(\psi_0/2) \cos(\gamma_0/2) + \sin(\gamma_0/2) \cos(\vartheta_0/2) \cos(\psi_0/2); \\ \lambda_{2_0} &= \sin(\gamma_0/2) \cos(\vartheta_0/2) \sin(\psi_0/2) + \sin(\vartheta_0/2) \cos(\gamma_0/2) \cos(\psi_0/2); \\ \lambda_{3_0} &= \sin(\psi_0/2) \cos(\gamma_0/2) \cos(\vartheta_0/2) - \sin(\gamma_0/2) \sin(\vartheta_0/2) \cos(\psi_0/2).\end{aligned}$$

Поточні значення параметрів  $\lambda_0, \lambda_1, \lambda_2, \lambda_3$  можна визначити, знаючи проекції кутової швидкості ЛА  $\omega_x, \omega_y, \omega_z$  на зв'язаній осі  $XYZ$ , шляхом розв'язання лінійного диференціального рівняння зі змінними коефіцієнтами. У цьому випадку параметри  $\lambda_0, \lambda_1, \lambda_2, \lambda_3$  кватерніона описують положення осей ЛА  $XYZ$  відносно інерціального простору:

$$\dot{\lambda} = \frac{1}{2} \Omega(t) \cdot \lambda(t)$$

де  $\Omega(t)$  – кососиметрична  $(4 \times 4)$ -матриця, яка відповідає вектору  $\omega = [\omega_x \omega_y \omega_z]^T$

$$\Omega(t) = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}; \lambda = \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}.$$

Цей вираз є кватерніонним однорідним лінійним диференціальним рівнянням першого порядку зі змінним коефіцієнтом у вигляді гіперкомплексного числа з дійсною частиною, що дорівнює нулю. У скалярній формі це рівняння має такий вигляд:

$$\begin{aligned}\dot{\lambda}_0 &= -0,5(\omega_x \lambda_1 + \omega_y \lambda_2 + \omega_z \lambda_3); \\ \dot{\lambda}_1 &= -0,5(\omega_x \lambda_0 + \omega_z \lambda_2 + \omega_y \lambda_3); \\ \dot{\lambda}_2 &= -0,5(\omega_y \lambda_0 + \omega_z \lambda_1 + \omega_x \lambda_3); \\ \dot{\lambda}_3 &= -0,5(\omega_z \lambda_0 + \omega_y \lambda_1 + \omega_x \lambda_2).\end{aligned}$$

Динаміка зміни параметрів кватерніона у випадку, коли кватерніон характеризує взаємне положення зв'язаних з ЛА осей  $XYZ$  і оберткових навігаційних

осей  $NHE$ , описується рівняннями

$$\begin{bmatrix} \dot{\lambda}_0 \\ \dot{\lambda}_1 \\ \dot{\lambda}_2 \\ \dot{\lambda}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_{x\Sigma} & -\omega_{y\Sigma} & -\omega_{z\Sigma} \\ \omega_{x\Sigma} & 0 & \omega_{z\Sigma} & -\omega_{y\Sigma} \\ \omega_{y\Sigma} & -\omega_{z\Sigma} & 0 & \omega_{x\Sigma} \\ \omega_{z\Sigma} & \omega_{y\Sigma} & -\omega_{x\Sigma} & 0 \end{bmatrix} \cdot \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \quad (1.3.4)$$

У свою чергу

$$\omega_{x\Sigma} = \omega_{x_{\text{ЛА}}} - \omega_{x_{NHE}}; \omega_{y\Sigma} = \omega_{y_{\text{ЛА}}} - \omega_{y_{NHE}}; \omega_{z\Sigma} = \omega_{z_{\text{ЛА}}} - \omega_{z_{NHE}},$$

де  $\omega_{y_{\text{ЛА}}}$ ,  $\omega_{x_{\text{ЛА}}}$ ,  $\omega_{z_{\text{ЛА}}}$  – проекції кутової швидкості ЛА відносно інерціального простору на осі зв'язаної системи координат, вимірювані датчиками кутових швидкостей;

$\omega_{x_{NHE}}$ ,  $\omega_{y_{NHE}}$ ,  $\omega_{z_{NHE}}$  – проекції кутової швидкості навігаційної системи координат відносно інерціального простору на осі зв'язаної системи координат, що визначаються в результаті розв'язання матричного рівняння

$$\begin{bmatrix} \omega_{x_{NHE}} \\ \omega_{y_{NHE}} \\ \omega_{z_{NHE}} \end{bmatrix} = B^T \begin{bmatrix} \omega_{N_V} + \Omega_N \\ \omega_{H_V} + \Omega_H \\ \omega_{E_V} + \Omega_E \end{bmatrix}.$$

Ці складові розраховуються й у раніше розглянутих алгоритмах.

У скалярній формі рівняння (1.3.4) мають вигляд:

$$\begin{aligned} \dot{\lambda}_0 &= -0,5(\omega_{x\Sigma}\lambda_1 + \omega_{y\Sigma}\lambda_2 + \omega_{z\Sigma}\lambda_3); \\ \dot{\lambda}_1 &= -0,5(\omega_{x\Sigma}\lambda_0 + \omega_{z\Sigma}\lambda_2 + \omega_{y\Sigma}\lambda_3); \\ \dot{\lambda}_2 &= -0,5(\omega_{y\Sigma}\lambda_0 + \omega_{z\Sigma}\lambda_1 + \omega_{x\Sigma}\lambda_3); \\ \dot{\lambda}_3 &= -0,5(\omega_{z\Sigma}\lambda_0 + \omega_{y\Sigma}\lambda_1 + \omega_{x\Sigma}\lambda_2). \end{aligned}$$

Матрицю  $B$  перерахування зі зв'язаної в географічну систему координат можна також отримати шляхом перемножування двох матриць, з яких одна перераховує зі зв'язаних у інерціальні осі, друга – з інерціальних у географічні. Кожна з двох матриць також обчислюється на основі параметрів Родрига-Гамільтона, які у свою чергу визначаються чисельним алгоритмом другого порядку, побудованим на основі методу послідовних наближень Пікара:

$$B = C^T A$$

$$A = \begin{bmatrix} 1 - 2(\lambda_2^2 + \lambda_3^2) & 2(\lambda_1\lambda_2 - \lambda_0\lambda_3) & 2(\lambda_1\lambda_3 + \lambda_0\lambda_2) \\ 2(\lambda_1\lambda_2 + \lambda_0\lambda_3) & 1 - 2(\lambda_1^2 + \lambda_3^2) & 2(\lambda_2\lambda_3 - \lambda_0\lambda_1) \\ 2(\lambda_1\lambda_3 - \lambda_0\lambda_2) & 2(\lambda_2\lambda_3 + \lambda_0\lambda_1) & 1 - 2(\lambda_1^2 + \lambda_2^2) \end{bmatrix};$$

$$\begin{aligned} \lambda_0^{(k+1)} &= \lambda_0^{(k)} - \lambda_0^{(k)} e / .8 - 0,5(\lambda_1^{(k)} \Delta\beta_x + \lambda_2^{(k)} \Delta\beta_y + \lambda_3^{(k)} \Delta\beta_z); \\ \lambda_1^{(k+1)} &= \lambda_1^{(k)} - \lambda_1^{(k)} e / .8 - 0,5(\lambda_0^{(k)} \Delta\beta_x + \lambda_3^{(k)} \Delta\beta_y + \lambda_2^{(k)} \Delta\beta_z); \\ \lambda_2^{(k+1)} &= \lambda_2^{(k)} - \lambda_2^{(k)} e / .8 - 0,5(\lambda_3^{(k)} \Delta\beta_x + \lambda_0^{(k)} \Delta\beta_y + \lambda_1^{(k)} \Delta\beta_z); \\ \lambda_3^{(k+1)} &= \lambda_3^{(k)} - \lambda_3^{(k)} e / .8 - 0,5(\lambda_2^{(k)} \Delta\beta_x + \lambda_1^{(k)} \Delta\beta_y + \lambda_0^{(k)} \Delta\beta_z), \end{aligned} \quad (1.3.5)$$

де  $e = \Delta\beta_x^2 + \Delta\beta_y^2 + \Delta\beta_z^2$ ;

$$\Delta\beta_x = \int_{t_k}^{t_k+1} \omega_{x_{\text{ЛА}}} dt; \Delta\beta_y = \int_{t_k}^{t_k+1} \omega_{y_{\text{ЛА}}} dt; \Delta\beta_z = \int_{t_k}^{t_k+1} \omega_{z_{\text{ЛА}}} dt;$$

$\Delta\beta_x, \Delta\beta_y, \Delta\beta_z$  – збільшення інтегралів від проекцій абсолютної кутової швидкості ЛА на осі чутливості гіроскопів (показання датчиків кутової швидкості БІНС, які вимірюють не проекції кутових швидкостей, а збільшення кутів повороту навколо своїх осей чутливості, тобто показання інтегруючих датчиків кутової швидкості):

$$C = \begin{bmatrix} 1 - 2(\mu_2^2 + \mu_3^2) & 2(\mu_1\mu_2 - \mu_0\mu_3) & 2(\mu_1\mu_3 + \mu_0\mu_2) \\ 2(\mu_1\mu_2 + \mu_0\mu_3) & 1 - 2(\mu_1^2 + \mu_3^2) & 2(\mu_2\mu_3 - \mu_0\mu_1) \\ 2(\mu_1\mu_3 - \mu_0\mu_2) & 2(\mu_2\mu_3 + \mu_0\mu_1) & 1 - 2(\mu_1^2 + \mu_2^2) \end{bmatrix};$$

$$\begin{aligned} \mu_0^{(k+1)} &= \mu_0^{(k)} - 0,5 \left( \mu_1^{(k)} \Omega_x + \mu_2^{(k)} \Omega_y + \mu_3^{(k)} \Omega_z \right) dt; \\ \mu_1^{(k+1)} &= \mu_1^{(k)} - 0,5 \left( \mu_0^{(k)} \Omega_x + \mu_3^{(k)} \Omega_y + \mu_2^{(k)} \Omega_z \right) dt; \\ \mu_2^{(k+1)} &= \mu_2^{(k)} - 0,5 \left( \mu_3^{(k)} \Omega_x + \mu_0^{(k)} \Omega_y + \mu_1^{(k)} \Omega_z \right) dt; \\ \mu_3^{(k+1)} &= \mu_3^{(k)} - 0,5 \left( \mu_2^{(k)} \Omega_x + \mu_1^{(k)} \Omega_y + \mu_0^{(k)} \Omega_z \right) dt \end{aligned}$$

де  $\Omega_x = \omega_{N_V} + \Omega_N$ ;  $\Omega_y = \omega_{H_V} + \Omega_H$ ;  $\Omega_z = \omega_{E_V} + \Omega_E$  – проекції абсолютної кутової швидкості географічного базису на його осі.

До переваг цього методу побудови матриці орієнтації відноситься гарантована ортогональність матриці орієнтації, обчисленої за співвідношеннями (1.3.5). Крім цього, практика показує, що обчислення з використанням параметрів Родрига-Гамільтона дає найменші обчислювальні витрати в порівнянні з іншими методами за умови забезпечення однакових точностних характеристик. Разом з тим,

визначення матриці  $B$  через параметри Родрига-Гамільтона призводить до необхідності рішення двох однотипних систем лінійних диференціальних рівнянь четвертого порядку кожна.

За елементами матриці  $B$  відповідно до (1.3.3) визначаються кути орієнтації ЛА: крен  $\gamma$ , тангаж  $\vartheta$  та рискання (курс)  $\psi$ :

Після знаходження матриці  $B$  система рівнянь для проведення навігаційних розрахунків замикається.

Алгоритм проведення навігаційних розрахунків у випадку формування матриці напрямних косинусів безпосередньо за кутами  $\gamma$ ,  $\vartheta$ ,  $\psi$  можна представити у вигляді (1.3.6)...(1.3.14).

### Швидкий темп

$$\begin{aligned}\omega_{y\Sigma} &= \omega_{y_{\text{ЛА}}} - \omega_{y_{\text{NHE}}}; \\ \omega_{x\Sigma} &= \omega_{x_{\text{ЛА}}} - \omega_{x_{\text{NHE}}}; \\ \omega_{z\Sigma} &= \omega_{z_{\text{ЛА}}} - \omega_{z_{\text{NHE}}}.\end{aligned}\tag{1.3.6}$$

$$\begin{aligned}\dot{\psi} &= (\omega_{y\Sigma} \cos \gamma - \omega_{z\Sigma} \sin \gamma) \sec \vartheta; \\ \dot{\gamma} &= \omega_{x\Sigma} + \operatorname{tg} \vartheta (\omega_{z\Sigma} \sin \gamma - \omega_{y\Sigma} \cos \gamma); \\ \dot{\vartheta} &= \omega_{y\Sigma} \sin \gamma + \omega_{z\Sigma} \cos \gamma; \\ \psi_{\Gamma} &= -\psi.\end{aligned}\tag{1.3.7}$$

$$B = \begin{bmatrix} \cos \psi \cos \vartheta & \sin \psi \sin \gamma - \cos \psi \sin \vartheta \cos \gamma & \sin \psi \cos \gamma + \sin \psi \cos \vartheta \sin \gamma \\ \sin \vartheta & \cos \vartheta \cos \gamma & -\cos \vartheta \sin \gamma \\ -\sin \psi \cos \vartheta & \cos \psi \sin \gamma + \sin \psi \sin \vartheta \cos \gamma & \cos \psi \cos \gamma - \sin \psi \sin \vartheta \sin \gamma \end{bmatrix}.\tag{1.3.8}$$

### Середній темп

$$\begin{bmatrix} a_N \\ a_H \\ a_E \end{bmatrix} = B \begin{bmatrix} a_{x_{\text{ЛА}}} \\ a_{y_{\text{ЛА}}} \\ a_{z_{\text{ЛА}}} \end{bmatrix}\tag{1.3.9}$$

$$\begin{aligned}\dot{V}_E &= a_E - V_N(\omega_{H_V} + 2\Omega_H) + V_H(\omega_{N_V} + 2\Omega_N); \\ \dot{V}_H &= a_H - V_E(\omega_{N_V} + 2\Omega_N) + V_N\omega_{E_V} + g_H; \\ \dot{V}_N &= a_N - V_H\omega_{E_V} + V_E(\omega_{H_V} + 2\Omega_H).\end{aligned}\tag{1.3.10}$$

### Повільний темп

$$\begin{aligned}\dot{\lambda} &= \frac{V_E}{(R_2+H) \cos B}; \\ \dot{\varphi} &= \frac{V_N}{R_1+H}; \\ \dot{H} &= V_H.\end{aligned}\tag{1.3.11}$$

$$\begin{aligned}
\omega_{E_V} &= -\dot{\varphi}; \\
\omega_{H_V} &= \dot{\lambda} \sin \varphi; \\
\omega_{N_V} &= \dot{\lambda} \cos \varphi; \\
\Omega_N &= \Omega_3 \cos \varphi; \\
\Omega_H &= \Omega_3 \sin \varphi.
\end{aligned} \tag{1.3.12}$$

$$\begin{bmatrix} \omega_{x_{NHE}} \\ \omega_{y_{NHE}} \\ \omega_{z_{NHE}} \end{bmatrix} = B^T \begin{bmatrix} \omega_{N_V} + \Omega_N \\ \omega_{H_V} + \Omega_H \\ \omega_{E_V} + \Omega_E \end{bmatrix}. \tag{1.3.13}$$

$$\begin{aligned}
\frac{1}{(R_1+H)} &\approx \frac{1}{a} \left[ 1 - e^2 - \frac{H}{a} - \frac{3}{2}e^2 \sin^2 \varphi \right]; \\
\frac{1}{(R_2+H)} &\approx \frac{1}{a} \left[ 1 - \frac{H}{a} - \frac{1}{2}e^2 \sin^2 \varphi \right]; \\
g_H &= -g \left( 1 + 5,2884 \cdot 10^{-3} \sin^2 \varphi \right) \left[ 1 - \frac{2H}{a} (1 - e \sin^2 \varphi) \right].
\end{aligned} \tag{1.3.14}$$

У випадку недостатньої швидкодії бортового процесора навігаційного обчислювача алгоритм роботи БІНС може бути розділений за необхідною швидкістю розрахунку (за тривалістю періоду дискретизації) на два або навіть на три рівні, що характеризують відповідно швидкий, середній і повільний темпи розрахунків.

Для більш коректного вибору датчиків необхідно оцінити орієнтовні значення похибок БІНС, в залежності від параметрів ДПІ.

## 1.4 Оцінка орієнтовних значень похибок вимірників первинної інформації БІНС

Датчики первинної інформації БІНС – датчики кутової швидкості й акселерометри встановлюються жорстко на ЛА. Тяжкі умови роботи датчиків інформації призводять до появи значних похибок, тому в алгоритмах роботи БІНС бажано здійснити аналітичну компенсацію похибок вимірників (здійснювати їх польотне калібрування), перш ніж ці сигнали будуть використані для розрахунку параметрів орієнтації і для визначення складових уявного прискорення уздовж навігаційних осей.

Інструментальні похибки ІНС визначаються погрішностями акселерометрів, вимірників кутової швидкості або кута, а також погрішностями обчислювального пристрою. Очевидно, при застосуванні обчислювального пристрою досить високої точності похибки, ІНС будуть визначатися головним чином погрішностями первинних вимірювальних датчиків, що входять у систему.

Якщо акселерометри ІНС вимірюють прискорення  $a_x$  і  $a_y$  з погрішностями  $\Delta a_x$  і  $\Delta a_y$ , то, це приведе до помилки у визначенні координати  $\Delta \lambda_y$ .

Приладові значення зазначених параметрів (зі значком «\*»)

$$\left. \begin{aligned} a_{\xi}^* &= a_{\xi} + \Delta a_{\xi}; & a_x^* &= a_x + \Delta a_x; & a_y^* &= a_y + \Delta a_y; \\ \dot{\lambda}_y^* &= \dot{\lambda}_y + \Delta \dot{\lambda}_y; & \lambda_y^* &= \lambda_y + \Delta \lambda_y; \\ \ddot{\vartheta}^* &= \ddot{\vartheta} + \Delta \ddot{\vartheta}; & \dot{\vartheta}^* &= \dot{\vartheta} + \Delta \dot{\vartheta}; & \vartheta^* &= \vartheta + \Delta \vartheta. \end{aligned} \right\} \quad (1.4.1)$$

Підставивши значення цих параметрів у перші рівняння систем і зробивши відповідні перетворення наступне рівняння похибок:

$$\Delta \ddot{\lambda}_y + \frac{(a_{\eta} + g_0)}{R_3} \Delta \lambda_y = \frac{1}{R_3} [a_x \cos(\lambda_y - \vartheta') + a_y \sin(\lambda_y - \vartheta')] \quad (1.4.2)$$

Як видно, ліва частина рівняння (1.4.2) є (при  $a_{\eta} = 0$ ) рівнянням маятника Шулера, а права – збурюючим впливом.

Координата  $\lambda_y$  і кут  $\vartheta'$  у процесі руху безупинно змінюються, тому права частина рівняння (1.4.2) буде теж змінною в часі.

З огляду на вираз і те, що при автоматичному керуванні рухом кут відхилення об'єкта від площини горизонту досить малий, а також вважаючи

$$\Delta a_x = \Delta a_y = \Delta a$$

у першому наближенні одержимо

$$\Delta \ddot{\lambda}_y + \frac{1}{R_3} (a_{\eta} + g_0) \Delta \lambda_y \cong \frac{\Delta a}{R_3} \quad (1.4.3)$$

При  $a_{\eta} = 0$ ,  $\Delta a = \text{const}$  рішення рівняння (1.4.3) буде наступним:

$$\Delta \lambda_y \cong \frac{\Delta a}{g_0} \left( 1 - \cos \left( \sqrt{\frac{g_0}{R_3}} \cdot t \right) \right) \quad (1.4.4)$$

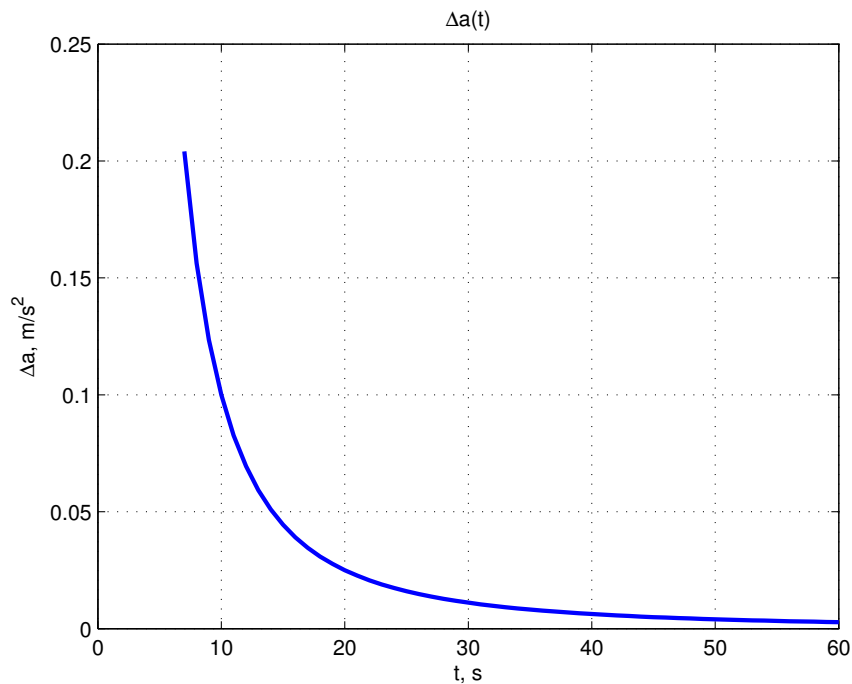


Рисунок 1.4.1 – Графік залежності значень похибок акселерометра від часу

З виразу (1.4.4) видно, що помилка ІНС у визначенні координати  $\lambda_y$ , обумовлена похибкою акселерометрів, буде мати як постійну, так і змінну складові. Найбільше значення похибки не перевищить  $\Delta\lambda_y \leq 2\frac{\Delta a}{g_0}$ .

#### *Оцінка помилки акселерометрів*

За допомогою (1.4.4) можуть бути отримані орієнтовані формули для розрахунку точнісних вимог пропонованих до датчиків первинної інформації – акселерометрам.

$$\Delta a \cong \frac{\Delta\lambda_y g_0}{\left(1 - \cos\left(\sqrt{\frac{g_0}{R_3}} \cdot t\right)\right)}. \quad (1.4.5)$$

Як випливає з (1.4.5) вимоги до точнісних характеристик акселерометрів залежать від проміжків часу автономної роботи БІНС у складі комплексної інерціально-супутникової системи навігації. Виходячи з вимог до точності визначення координат (СКВ 5 м) отримані орієнтовані значення похибок акселерометра, у залежності від очікуваних перерв у роботі супутникової системи навігації. Розрахункові значення точнісних вимог пропонованих до датчиків первинної інформації, зокрема акселерометрів відображені на графіку рис. 1.4.1

#### *Оцінка помилки датчика кутової швидкості*



Якщо вимірник кутової швидкості об'єкта має погрішність  $\Delta\vartheta'$ , то приладове значення кутової швидкості

$$\dot{\vartheta}'^* = \dot{\vartheta}' - \Delta\dot{\vartheta}'.$$

При цьому, будуть мати місце помилки у визначенні інших параметрів.

Підставляючи значення параметрів  $\dot{\vartheta}'^*$  і  $\lambda_y^*$  рівняння (1.4.3), після перетворень з врахуванням другого рівняння системи (1.4.1) одержимо

$$\Delta\ddot{\lambda}_y + \frac{a_\eta + g_0}{R_3}\Delta\lambda_y = -\frac{a_\eta + g_0}{R_3}\Delta\vartheta' \quad (1.4.6)$$

Як видно ліва частина рівняння (1.4.6) і в цьому випадку (при  $a_\eta = 0$ ) представляється рівняння маятника Шулера, а права частина – фактор, що викликається, обумовленими погрішностями у вимірі  $\vartheta'$  кута.

Якщо вважати погрішність  $\Delta\dot{\vartheta}' = \Delta\dot{\vartheta}'_0 = const$ , то  $\Delta\vartheta' = \Delta\dot{\vartheta}'_0 t$ , при цьому рішення рівняння (1.4.6) буде (якщо  $a_\eta = 0$ ) наступної:

$$\Delta\lambda_y = \Delta\dot{\vartheta}'_0 \left( \sqrt{\frac{R_3}{g_0}} \sin \sqrt{\frac{g_0}{R_3}} \cdot t - t \right) \quad (1.4.7)$$

Як видно з виразу (1.4.7), погрішність у визначенні координати  $\lambda_y$ , обумовлена постійною помилкою вимірника кутової швидкості, у першому наближенні має дві складові (рис. 2.5,б), одна з яких росте пропорційно часу польоту

$$\Delta\lambda_{y0} = \Delta\dot{\vartheta}'_0 t,$$

а інша змінюється з періодом маятника Шулера

$$\Delta\lambda_y = \Delta\dot{\vartheta}'_0 \sqrt{\frac{R_3}{g_0}} \sin \sqrt{\frac{g_0}{R_3}} \cdot t$$

Аналогічно (1.4.5) можуть бути отримані орієнтовані формули для розрахунку точносних вимог пропонованих до вимірників кутових швидкостей.

$$\Delta\dot{\vartheta}'_0 = \frac{\Delta\lambda_y}{\left( \sqrt{\frac{R_3}{g_0}} \sin \left( \sqrt{\frac{g_0}{R_3}} \cdot t \right) - t \right)}$$

Виходячи з вимог пропонованих до точносних характеристик визначення координат (СКО  $\approx 5$ м) отримані орієнтовані значення похибок вимірникам

кутових швидкостей, у залежності від очікуваних перерв у роботі супутникової системи навігації. Розрахункові значення точнісних вимоги пропонованих до датчиків первинної інформації, зокрема вимірникам кутових швидкостей відображені на графіку рис.1.4.2.

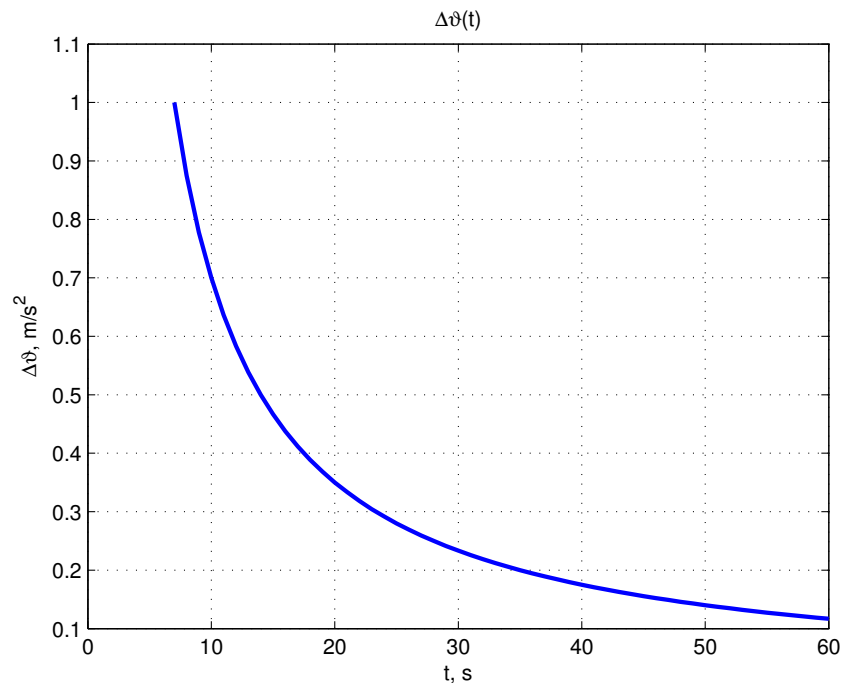


Рисунок 1.4.2 – Графік залежності значень похибок ДКШ від часу

Для БІНС розглянутого класу основний внесок у похибки визначення координат вносять датчики первинної інформації. Необхідно відзначити, що методичні похибки, у тому числі похибки, зв'язані зі спрощеннями кінематичних рівнянь БІНС, похибками моделювання форми Землі і похибками моделі гравітаційного поля, повинні бути не більше похибок, внесених датчиками первинної інформації.

Багато складові вихідні похибки залежать від параметрів траєкторії й умов роботи, коефіцієнти моделі похибок істотно залежать від рівня вібрації і температури. Тому для більш детального дослідження точнісних характеристик БІНС необхідна вихідна інформація про аеродинамічні й інерційно масові характеристики літака, а також параметри траєкторії. У цьому випадку можна буде провести детальні статистичні дослідження точнісних характеристик з урахуванням впливу динамічних похибок датчиків первинної інформації.

Однак при моделюванні враховувалися тільки деякі складові:

1. систематичні;
2. випадкові складові;

Випадкові складові і перекручування масштабного коефіцієнта моделювалися з використанням генераторів "білого шуму" і формуючих фільтрів. При цьому вважалося, що кожен чуттєвий елемент цілком визначається значеннями цих складових, а самі ці складові змінюються таким чином, що при збільшенні одного з них зростають і всі інші.

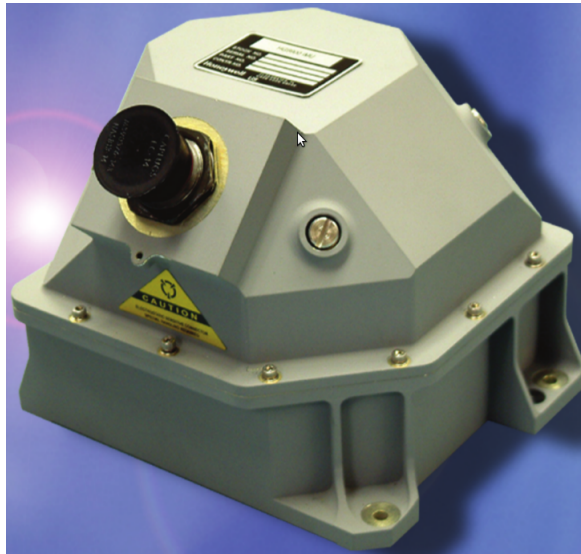


Рисунок 1.4.3 – ДПІ Honeywell HG9900IMU

В роботі пропонується датчики первинної інформації *HG9900IMU* (рис.1.4.3), які базуються на лазерних гіроскопах *Honeywell GG1320AN01* та кварцових акселерометрах *Honeywell QA-2000*. Всі датчики знаходяться в протиударному герметичному корпусі. Параметри ДПІ показані в таблицях 1.4.2, 1.4.3.

Таблиця 1.4.1 – Параметри IMU HG9900

Процесор	TI DSP TMS320VC33 (60 Mips)
Пам'ять	128 Kbytes SRAM, 512 Kbytes Flash EEPROM
Ввід/вивід	SDLC RS-422
Частота	300 Hz
Живлення	5, $\pm 15$ Vdc input
Потужність	<10W
Вага	2.948 кг
Температурний діапазон	$-54^{\circ}\text{C}$ до $+71^{\circ}\text{C}$
Розміри	$5.5 \times 6.4 \times 5.34\text{inch}$

Таблиця 1.4.2 – Параметри гіроскопів GG1320AN01

Дрейф	<0.003 deg/hr
Випадкове блукання	<0.002 deg/ $\sqrt{hr}$
Коефіцієнт масштабування	<5.0 PPM

Таблиця 1.4.3 – Параметри акселерометрів QA-2000

Дрейф	<25 $\mu g$
Випадкове блукання	<0.002 deg/ $\sqrt{hr}$
Коефіцієнт масштабування	<100 PPM

Іншою перевагою вказаних датчиків є присутність, температурних сенсорів, що за допомогою спеціально створеного алгоритму, компенсує температурний дрейф параметрів (наприклад масштабний коефіцієнт, не ортогональність осей).

## 2 Аналіз та вибір методів оцінювання та корекції в комплексній інерціально-супутниковій системі

Основними задачами пілотажно-навігаційних комплексів (ПНК) як постачальника інформаційного забезпечення польоту ЛА є сумісна обробка навігаційної інформації, яка надходить на борт ЛА та забезпечення високої надійності функціонування бортових систем та комплексів ЛА і взагалі безпеки польоту за рахунок резервування джерел інформації. Висока ефективність використання інформації, яка надходить на борт ЛА, забезпечується застосуванням різних методів її обробки.

Найкращі результати підвищення якісних характеристик вимірювальних комплексів досягаються в системах зі структурною надмірністю, коли існує можливість отримання пілотажно-навігаційної інформації паралельно декількома способами з використанням інформації від приладів та вимірювальних систем, що входять до складу ПНК. Отримана таким чином інформація комплексується.

В існуючих ПНК широке розповсюдження знайшли такі способи сумісної обробки інформації, що надходять від декількох вимірників, як взаємна компенсація і фільтрація похибок вимірювальних приладів, що вимірюють один і той самий навігаційний параметр та оптимальне оцінювання вектора стану з використанням апріорної інформації про контрольований процес та поточні вимірювання.

Методи оптимальної обробки інформації в ПНК використовуються з метою отримання оцінок вектора стану повітряного судна (або деякої частини цього вектора) в умовах впливу випадкових збурень і завад на процес вимірювання. При цьому оцінюються не самі параметри польоту, а їхні похибки. За оптимальної обробки пілотажно - навігаційної інформації в ПНК найважливішим процесом є процес отримання оптимальних оцінок. В основу алгоритмів

					<i>НАУ 11 54 75 000 ПЗ</i>			
Зм.	Лист	№ докум.	Підп.	Дата				
Розробив	НовікМ.В.				Інтегрована інерціально-супутникова система навігації, що базується на принципах комплексної обробки інформації з використанням калманівської фільтрації	Лит.	Аркуш	Аркушів
Перевірив	Мар'ясоваТ.І.						61	152
Консульт.	ФіляшкінМ.К.					<i>IACU 608</i>		
Н. контр.	КозловаА.П.							
Затвердив	СинеглазовВ.М.							

отримання оптимальних оцінок можуть бути покладені такі методи обробки інформації [6]:

- метод найменших квадратів;
- метод максимуму правдоподібності;
- рекурентний неоптимальний фільтр;
- оптимальний фільтр Калмана.

## 2.1 Огляд методів оптимальної обробки інформації

Метод найменших квадратів (МНК) застосовується для одержання оптимальних оцінок при обробці накопичених вимірювань. Якщо виконано  $m$  вимірювань координат (параметрів) системи  $\dot{X}(t) = A(t)X(t) + B(t)u_x(t)$ , тоді

$$Z = HX + Vz \quad (2.1.1)$$

де  $H$  – матриця спостереження;

$Z$ ,  $X$  і  $Vz$  – вектори, компонентами яких є реалізація вектора вимірювання  $Z$  і, вектора стану системи  $X$  і вектора помилок вимірювання  $Vz_i$  відповідно, причому  $i = \overline{1, m}$ .

Необхідно за спостереженнями  $Z$  і заданою матрицею спостереження щонайкраще оцінити стан вектора  $X$ . Критерієм такої оцінки за МНК служить функціонал  $J = \sum_{i=1}^m V_{z_i}^T V_{z_i}$  який мінімізує суму квадратів помилок вимірювання  $Vz_i$ . У матричному виді цей вираз запишеться так:

$$J = |V_{z_1} V_{z_2} \dots V_{z_m}| \begin{vmatrix} V_{z_1} \\ V_{z_2} \\ \vdots \\ V_{z_m} \end{vmatrix}$$

або з урахуванням (2.1.1)

$$J = (Z - HX)^T (Z - HX) \quad (2.1.2)$$

Оцінку  $\hat{X}$ , вектора стану системи можна одержати шляхом розв'язання рівняння  $\frac{\partial J}{\partial \hat{X}} = 0$ .

З урахуванням рівняння (2.1.2) маємо

$$H^T(Z - H\hat{X}) + (Z - H\hat{X})^T H = 0 \quad (2.1.3)$$

Доданки виразу (2.1.3) рівні між собою, оскільки є добутками транспонованих відносно один до одного однакових співмножників  $H$  і  $(Z - H\hat{X})$ . Отже, тільки рівність нулю кожного з цих двох доданків забезпечує рівність нулю виразу (2.1.3). Отже

$$H^T(Z - H\hat{X}) = 0$$

Відтепер можна сформулювати необхідні і достатні умови одержання оптимальних оцінок  $\hat{X}$  вектора стану системи за методом найменших квадратів у вигляді основних положень, виконання яких передбачає:

- наявність накопичених спостережень  $Zi, i = \overline{1, m}$ ;
- знання матриці спостережень ;
- не особливість матриці  $H^T H$ , тобто  $H^T H \neq 0$ .

Отримання оцінки  $\hat{X}$  зв'язано з накопиченням спостережень  $Zm$  у наслідок чого нова оцінка параметра не збігається за часом з його поточним значенням на час, необхідний для накопичення спостережень. Тому даний алгоритм для оцінки використовують лише у випадку виміру того самого параметра одночасно кількома датчиками.

Метод найменших квадратів застосовується в тому випадку, коли надлишок інформації виходить за рахунок рівно точних вимірювань різними датчиками інформації. При цьому, вимоги щодо спектрального складу помилок датчиків не відзначаються. Відповідно до цього методу, має місце мінімізація суми квадратів помилок всіх вимірювань.

#### *Алгоритм оцінювання за методом максимуму правдоподібності*

Алгоритм оцінювання за методом максимуму правдоподібності як і алгоритм оцінювання за МНК потребує накопичення вимірювань, тобто наявності вектора спостережень. Передбачається, що похибки вимірювання розподілені

за нормальним законом. Тоді щільність розподілу ймовірностей вектора  $V_{z_m}$  має вигляд:

$$P(V_{z_m}) = \frac{1}{\sqrt{(2\pi)^m |R_z|}} \exp \left[ -\frac{1}{2} V_{z_m}^T R_z^{-1} V_{z_m} \right], \quad (2.1.4)$$

де  $R_z$  – кореляційна матриця похибок вимірювання;

$|R_z|$  – визначник матриці  $R_z$ .

Використання алгоритму оцінок за методом максимуму правдоподібності передбачує виконання умови  $|R_z| \neq 0$ , тобто матриця  $R_z$  не повинна бути особливою. Підставивши (2.1.1) у (2.1.4), отримаємо вираз для функції правдоподібності.

$$\psi(X) = \frac{1}{\sqrt{(2\pi)^m |R_z|}} \exp \left[ -\frac{1}{2} (Z_m - H X_m)^T R_z^{-1} (Z_m - H X_m) \right],$$

яка являє собою щільність розподілу помилок вимірювання.

Необхідно обрати таку оцінку  $\hat{X}_m$ , при якій функція правдоподібності  $\psi(X)$  перетворюється в максимум, що відповідає мінімуму квадратів відхилень виміряних координат вектора  $X$  від їхнього дійсного значення. Для цього необхідно, щоб  $\frac{\partial \psi(X)}{\partial X} = 0$ .

На практиці зручніше обчислювати максимум не самої функції правдоподібності, а її логарифма, тобто

$$\ln \psi(X) = \ln \frac{1}{\sqrt{(2\pi)^m |R_z|}} - \frac{1}{2} (Z_m - H X_m)^T R_z^{-1} (Z_m - H X_m). \quad (2.1.5)$$

Узявши в рівнянні (2.1.5) похідні за компонентами вектора  $X_m$  і прирівнюючи їхню суму до нуля, одержимо:

$$\frac{1}{2} H^T R_z^{-1} (Z_m - H \hat{X}_m) + \frac{1}{2} H R_z^{-1} (Z_m - H \hat{X}_m)^T = 0 \quad (2.1.6)$$

Зауважимо, що як і для формули оцінки вимірювань за методом найменших квадратів, один із доданків виразу (2.1.6) є транспонованим відносно іншого. Отже, доданки цього виразу рівні між собою, вони не можуть бути від'ємні, тому кожний з них дорівнює нулю.

Припустимо, що

$$H^T R_z^{-1} (Z_m - H \hat{X}_m) = 0$$

тоді

$$\hat{X}_m = (H^T R_z^{-1} H)^{-1} H^T R_z^{-1} Z_m \quad (2.1.7)$$



Вираз (2.1.7) стає вихідним для розробки алгоритму отримання оптимальних оцінок за методом максимуму правдоподібності.

Для визначення цих оцінок необхідно:

- накопичити  $m$  спостережень –  $Zm$ ;
- знати кореляційну матрицю  $Rz$  похибок вимірника;
- знати матрицю зв'язків спостереження  $H$ .

Як і для алгоритму оцінок за МНК отримання оцінки  $\hat{X}_m$  пов'язано з накопиченням вимірювань  $Zm$ , тому цей метод, як і МНК, можна використовувати лише при вимірюванні одного параметра декількома системами. В іншому випадку нова оцінка помилок ПНК не буде співпадати з поточним значенням помилок на час, що дорівнює часу накопичення спостережень.

Алгоритм оцінювання за методом максимуму правдоподібності, як і алгоритм оцінювання за методом найменших квадратів, потребує накопичення вимірювань, тобто наявності вектора спостережень. Цей метод пропонує максимізацію функції правдоподібності, що відповідає мінімуму квадратів помилок. При цьому, передбачається, що похибки вимірювання розподілені за нормальним законом.

### *Рекурентний метод обробки інформації*

Рекурентний метод обробки інформації дозволяє отримати оцінку параметра після кожного дослід. За результат оцінки вимірюваного параметра  $m$  при проведенні  $m$  спостережень візьмемо:

$$\hat{x}_m = \frac{1}{m} \sum_{i=1}^m z_i,$$

де  $z_i = x + v_{z_i}$ ,  $i = \overline{1, m}$ ;

$x$  – вимірюваний параметр;

$v_{z_i}$  – похибка  $i$ -го спостереження.

Тоді на черговому  $(i + 1)$ -му кроці вимірювань значення оцінки  $\hat{x}_{m+1}$ , має вигляд:

$$\hat{x}_{m+1} = \frac{(\sum_{i=1}^m z_i) + z_{m+1}}{m + 1} = \frac{m}{m + 1} \left( \frac{1}{m} \sum_{i=1}^m z_i \right) + \frac{1}{m + 1} z_{m+1} \quad (2.1.8)$$

або

$$\hat{x}_{m+1} = \frac{m}{m+1} \hat{x}_m + \frac{1}{m+1} z_{m+1}$$

де  $z_{m+1}$  – останнє  $(m+1)$ -ше спостереження.

Виконавши певні перетворення отримаємо:

$$\hat{x}_{m+1} = \hat{x}_m + \frac{1}{m+1} (z_{m+1} - \hat{x}_m)$$

або, позначивши  $\frac{1}{m+1} = k$ ,

$$\hat{x}_{m+1} = \hat{x}_m + k (z_{m+1} - \hat{x}_m). \quad (2.1.9)$$

Отже, оцінку  $\hat{x}_{m+1}$  можна отримати з попередньої оцінки  $\hat{x}_m$  шляхом складання її з різницею між новим спостереженням  $z_{m+1}$  та попередньою оцінкою, помноженою на коефіцієнт ваги  $k$ . У цьому випадку зникає необхідність зберігати  $m$  спостережень, отриманих на попередніх кроках вимірювання, оскільки вся попередня інформація об'єднана в апіорній оцінці  $\hat{x}_m$ .

Рекурентний алгоритм (2.1.9) зв'язує поточне значення оцінки  $\hat{x}_{m+1}$  з її попереднім значенням  $\hat{x}_m$

Різниця  $(z_{m+1} - \hat{x}_m)$  стає показником цінності інформації, яку отримують при проведенні  $z_{m+1}$ -го спостереження. Дійсно, якщо ця різниця близька до нуля, то зафіксоване спостереження  $z_{m+1}$  не несе будь-якої нової інформації у порівнянні з апіорною, і в цьому випадку  $\hat{x}_{m+1} \cong \hat{x}_m$ . Навпаки, при великій різниці  $(z_{m+1} - \hat{x}_m)$  з урахуванням вагового коефіцієнта здійснюється суттєве уточнення оцінки  $\hat{x}_m$ , отриманої на попередньому кроці розрахунків.

Але коефіцієнт  $k = \frac{1}{m+1}$  отримано без використання критерію оптимальності, тому оцінка  $\hat{x}_{m+1}$  також не є оптимальною, що знижує цінність даного методу обробки інформації.

Алгоритм неперервного оптимального фільтра Калмана об'єднує розв'язання двох задач: спостереження та фільтрації, враховуючи характеристики контрольованого процесу, мінімізує помилку оцінювання. З іншого боку ФК дозволяє оцінити навігаційні параметри, що безпосередньо не спостерігаються, а відносна простота реалізації алгоритму в БЦОМ робить цей метод комплексної обробки інформації найбільш привабливим. Детальніше алгоритм роботи, розглянемо в наступному розділі.

## 2.2 Рекурентний фільтр Калмана

Оптимальний фільтр Калмана [14] — ефективна і гнучка процедура для комплексної обробки інформації із зашумлених датчиків для оцінки стану стохастичної системи. В більшості випадків, кількість параметрів, що задають стан об'єкту, більше, ніж кількість спостерігаємих параметрів, доступних для вимірювання. За допомогою моделі об'єкту по ряду доступних вимірів фільтр Калмана дозволяє отримати оцінку внутрішнього стану системи.

ФК — призначений для рекурсивного дооцінювання вектора стану апріорно відомої динамічної системи, для розрахунку поточного стану системи необхідно знати поточні виміри, а також попередній стан самого фільтра. Таким чином фільтр Калмана, як і інші рекурсивні фільтри, реалізований в часовому представленні, а не в частотному.

Фільтра включає 2 типа змінних:

1. Вектор стану системи, включає наступні компоненти:

- Змінні, які безпосередньо нас цікавлять (необхідно знайти, наприклад швидкість, прискорення)
- Змінні, які безпосередньо не використовуються, але необхідні для процесу оцінювання. В загальному випадку не важливо знати їхні значення, але необхідно визначити їх для покращення точності оцінки.
- Фільтр Калмана у специфічних задачах має включати всі ті змінні динаміки системи, які можуть бути виміряні ДПП.

2. Коваріаційна матриця: міра невизначеності оцінювання. Рівняння, що використовуються для отримання коваріаційної матриці (рівняння Ріккатті) та управління невизначеністю, визначають як шум датчиків та динаміка невизначеності впливають на невизначеність стану системи, що оцінюється.

За допомогою отримання невизначеності власне системи і невизначеності у відповідних показника датчиків, ФК дає можливість комплексувати данні

з усіх ДПІ “отимально”, в сенсі, що результуюча оцінка мінімізує квадратичну функцію помилки оцінки, включаючи мінімальне середньо квадратичне відхилення будь-якої лінійної комбінації помилок оцінювання. Коефіцієнт підсилення Калмана — оптимальна зважена матриця, що поєднує данні ДПІ з попередньою оцінкою, для отримання нової апостеріорної оцінки.

### *Рівняння динамічної системи*

Фільтр Калмана намагається оцінити стан змінної  $x \in \mathbb{R}^n$  дискретно управляемого процесу, що описується лінійними стохастичними диференціальним рівняннями:

$$x_k = \Phi x_{k-1} + w_{k-1}, \quad (2.2.1)$$

З вимірюваннями  $Z \in \mathbb{R}^m$ :

$$z_k = H x_k + v_k, \quad (2.2.2)$$

Випадкові змінні  $w_k$  та  $v_k$  представляють шум системи (process noise) та шум вимірювань (measurement noise), які вважаються незалежними один від одного, білими з нормальним розподілом шумами.

$$p(w) \sim N(0, Q), \quad (2.2.3)$$

$$p(v) \sim N(0, R). \quad (2.2.4)$$

На практиці, коваріаційна матриця шуму системи  $Q$  та коваріаційна матриця шуму вимірювань  $R$  можуть змінюватись на кожному кроці вимірювань, але для простоти, в нашому випадку, вони постійні.

Матриця  $\Phi$  розміром  $n \times n$  в диференціальному рівнянні (2.2.1) проводить залежність між попереднім станом системи  $k - 1$  та поточним станом  $k$ , при наявності вхідного сигналу чи шуму. Матриця  $(H)$   $m \times n$  в рівнянні (2.2.2) відповідає вимірюванням  $z_k$ .

### *Отримання рівнянь фільтра*

Позначимо  $\hat{x}(-) \in \mathbb{R}^n$  — апіорна оцінка на кроці  $k$  інформація про рух системи до кроку  $k$ , та  $\hat{x}(+) \in \mathbb{R}^n$  апостеріорна оцінка на кроці  $k$  за допомогою

вимірювань  $z_k$ . Відповідно апіорна оцінка коваріації помилки:

$$P_k(-) = E[(x_k - \hat{x}_k(-))(x_k - \hat{x}_k(-))^T] \quad (2.2.5)$$

та апостеріорна коваріаційна матриця помилок:

$$P_k(+) = E[(x_k - \hat{x}_k(+))(x_k - \hat{x}_k(+))^T] \quad (2.2.6)$$

При виведенні рівнянь фільтра Калмана, наша ціль – знайти рівняння, що розраховують апостеріорну оцінку стану  $\hat{x}_k(+)$  як лінійну компенсацію апіорної оцінки стану  $\hat{x}_k(-)$  та взваженої різниці між безпосередньо вимірами  $z_k$  та їхнім прогнозом  $H\hat{x}_k$ , як показано нижче (2.2.7)

$$\hat{x}_k(+) = \hat{x}_k(-) + K [z_k - H\hat{x}_k(-)] \quad (2.2.7)$$

Різниця  $(z_k - H\hat{x}_k(-))$  – називається інноваціями (*innovation*) або залишками (residual). Залишки, як відображення дисперсії між прогнозованими вимірами  $H\hat{x}_k(-)$  та безпосередніми вимірами  $z_k$ . Залишок, який рівний нулю, означає, що обоє вимірів знаходяться в повному узгодженні.

Матриця  $K$   $n \times m$  визначає коефіцієнт підсилення або фактор змішування, який мінімізує апостеріорну помилку оцінки (2.2.6). Мінімізація може бути закінчена підстановкою (2.2.7) у подане вище визначення для коваріації (2.2.6), виконуючи необхідні дії та взявши похідну по  $K$ , встановивши результат рівним нулю, а потім вирішуючи відносно  $K$  рівняння. Детальне виведення рівнянь описане в [7]. Результат може бути отриманий в наступній формі:

$$K_k = P_k(-)H^T(H P_k(-)H^T + R)^{-1} = \frac{P_k(-)H^T}{H P_k(-)H^T + R} \quad (2.2.8)$$

Зважаючи на (2.2.8), якщо коваріаційна матриця помилок вимірювання  $R$  прямує до нуля, коефіцієнт підсилення  $K$  взважає залишки більш сильно. Наприклад:

$$\lim_{R_k \rightarrow 0} K_k = H^{-1} \quad (2.2.9)$$

Якщо апіорна оцінка коваріаційної матриці  $P_k(-)$  наближається до нуля, коефіцієнт підсилення  $K$  майже не взважає залишки.

$$\lim_{P_k(-) \rightarrow 0} K_k = 0 \quad (2.2.10)$$

З іншого боку, взважування за допомогою  $K$ , у випадку коли коваріація помилки вимірювань  $R$  наближається до нуля, то безпосереднім вимірам  $z_k$  довіряється все більше і більш, коли прогнозовані виміри  $H\hat{x}_k(-)$  враховуються все менше. Але якщо апріорна коваріація помилки  $P_k(-)$  наближається до нуля то спостереження  $z_k$  майже не враховуються, коли відповідно прогнозовані виміри  $H\hat{x}_k(-)$  стають більш “впливовими”.

## 2.3 Алгоритм фільтра Калмана

ФК знаходить стан системи за допомогою форми зворотнього зв'язку: фільтр оцінює стан процесу в один момент часу, а потім отримує зв'язок в формі зашумлених вимірів. Такі рівняння для фільтра розділяються на дві групи: прогноз та корекція на основі вимірів. Прогноз відповідає за екстраполяцію в часі поточного стану системи та її коваріації для отримання апріорної оцінки для наступного кроку. Рівняння корекції необхідні для зворотнього зв'язку – використовують нові виміри і апріорні оцінки для отримання апостеріорних оцінок.

Для фільтра не важливо з якого кроку починати, так і рівняння можуть бути представлені з точки зору прогнозу та корекції Рис.2.3.1

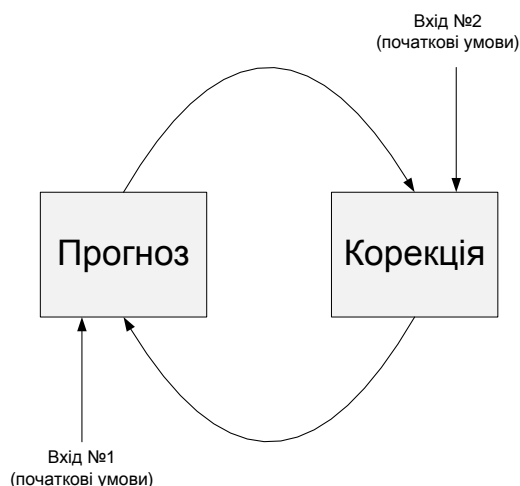


Рисунок 2.3.1 – Цикл функціонування фільтра Калмана

Перше завдання при корекції – розрахувати коефіцієнт підсилення Калмана,  $K_k$ / Наступний крок, безпосередньо отримати виміри  $z_k$ , а потім генеру-

Таблиця 2.3.1 – Рівняння дискретного фільтра Калмана

Рівняння / Формула		
Модель динаміки системи	$x_k = \Phi x_{k-1} + w_{k-1},$	(2.3.1)
Модель вимірювань	$z_k = H x_k + v_k$	(2.3.2)
Початкові умови	$E[x_o] = \hat{x}_0, E[x_o x_o^T] = P_0$	(2.3.3)
Припущення незалежності	$E[w_k v_j^T] = 0$ для всіх k та j	(2.3.4)
Екстраполяція оцінки (прогноз)	$\hat{x}_k(-) = \Phi \hat{x}_{k-1}(+)$	(2.3.5)
Екстраполяція коваріації	$P_k(-) = \Phi_{k-1} P_{k-1}(+) \Phi_{k-1}^T + Q_{k-1}$	(2.3.6)
Корекція оцінки	$\hat{x}_k(+) = \hat{x}_k(-) + K_k [z_k - H_k \hat{x}_k(-)]$	(2.3.7)
Корекція оцінки коваріації	$P_k(+) = [I - K_k H_k] P_k(-)$	(2.3.8)
Коефіцієнт підсилення Калмана	$K_k = P_k(-) H_k^T [H_k P_k(-) H_k^T + R_k]^{-1}$	(2.3.9)

вати апостеріорну оцінку за допомогою (2.2.7). Останній крок – це отримання апостеріорної коваріаційної матриці оцінки помилки.

Після кожного разу екстраполяції та корекції, процес повторюється, попередні апостеріорні оцінки використовуються для прогнозу нових апіорних оцінок. Ця рекурсивна природа є найбільш важливою особливістю фільтра Калмана, це робить його більш практичним, в порівнянні з іншими фільтрами (наприклад фільтр Віннера)

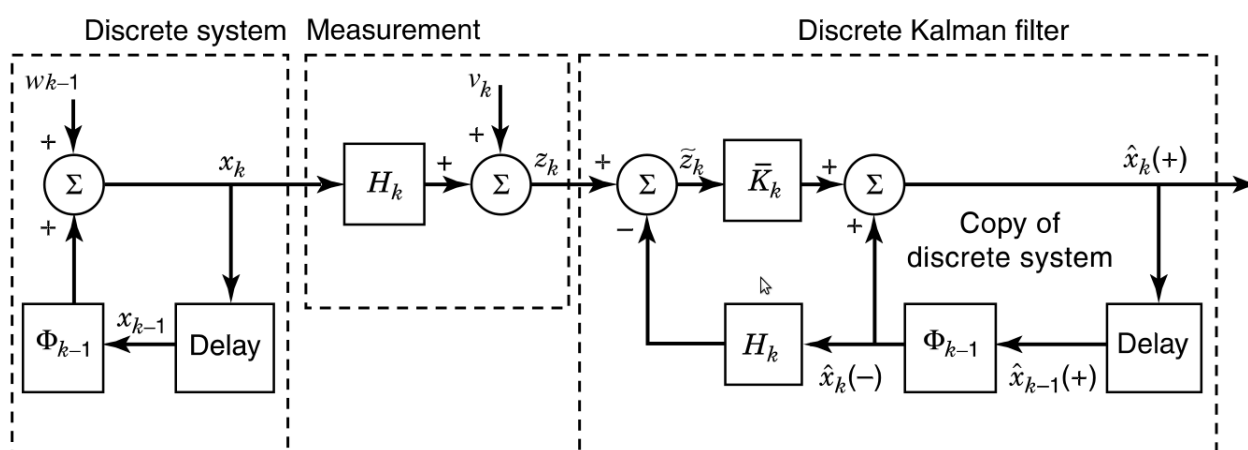


Рисунок 2.3.2 – Блок діаграма системи, моделі вимірювань та дискретного ФК

## 2.4 Проектування фільтра Калмана

В практичній реалізації фільтра, коваріація вимірюваного шуму  $R$  вимірюється звичайно до використання фільтра. Знаходження цієї матриці в загальному випадку практична задача, виміряні величини дають можливість визначити дисперсію шуму, що діє на данні датчиків.

Визначення коваріації шуму системи  $Q$  в загальному більш складна задача, просто не має можливості безпосередньо спостерігати процес який оцінюється. Інколи відносно проста модель процесу може дати прийнятний результат, якщо введена достатня величина невизначеності процесу через вибір матриці  $Q$ .

В іншому випадку, є чи немає раціональної бази для вибору параметрів, часто краща продуктивність (з точки зору статистики) може бути отримана за допомогою налаштування параметрів фільтра  $Q$  та  $R$ .

Асиметрія коваріаційної матриці  $P$  один з факторів, що впливає на чисельну нестійкість рівняння Рікатті. Якщо не використовується фільтр з квадратним коренем, тоді ця матриця може бути “симетризована” просто за наступною формулою:

$$P = \frac{1}{2}(P + P^T) \quad (2.4.1)$$

Цей метод використовується протягом довгого часу і добре себе зарекомендував.

При проектуванні фільтра, корекція коваріаційної матриці (2.3.6) та (2.3.8) має бути перевірена не тільки на симетрію але й на додатно визначеність. Якщо ці умови не будуть виконуються, це свідчить про помилки в програмі або матриця погано обумовлена. Для усунення проблеми обумовленості використовується інше рівняння для  $P_k(+)$ , яке називається формою Джозефа [?], яка показна на наступному рівнянні:

$$P_k(+) = [I - K_k H_k] P_k(-) [I - K_k H_k]^T + K_k R_k K_k^T \quad (2.4.2)$$

З рівняння видно, що права частина є сумою двох симетричних матриць. Перша додатно визначена інша не від’ємно визначена, що робить  $P_k(+)$  додатно визначеною матрицею.

Безперечно саме фільтр Калмана найбільш привабливий при розв’язанні



задачі комплексної обробки інформації в інерціально-супутникових системах навігації.

### 3 Розробка алгоритмів оптимального комплексування в інерціально-супутникових систем навігації

При побудові інтегрованих навігаційних систем широке поширення одержав прийом, заснований на формуванні різницевих вимірів, зі складу яких виключаються шукані параметри. З використанням різницевих вимірів вирішується задача оцінювання похибок однієї підсистеми на фоні похибок іншої підсистеми. Цей прийом найчастіше називають методом одержання інваріантних оцінок. При реалізації такого методу використовуються лінійні моделі еволюції похибок підсистем і не потрібно введення в загальному випадку нелінійних моделей еволюції самих шуканих навігаційних параметрів, що істотно спрощує побудову алгоритмів комплексної обробки навігаційної інформації і дає можливість застосування добре освоєних процедур оптимальної лінійної калмановської фільтрації.

В інтегрованих інерціально-супутникових системах широке поширення одержали дві схеми реалізації інваріантного методу, що відрізняються правилом формування різницевих вимірів і складом оцінюваного вектора стану.

В одній з цих схем, що одержала назва слабкозв'язаної, виміру для задачі фільтрації формуються шляхом утворення різниць координат і компонентів швидкості, вироблюваних ІНС і СНС, а у вектор стану включаються похибки ІНС і СНС. У такій схемі можливість використання вимірювальної інформації виникає при наявності сигналів не менш, ніж від чотирьох навігаційних супутників.

Для другої схеми, названої жорсткозв'язаної, різницеві виміри формуються шляхом зіставлення безпосередньо обмірюваних СНС і обчислених за показниками ІНС псевдодальностей і псевдошвидкостей. Для такої схеми у вектор стану включаються похибки ІНС, похибки виміру в СНС псевдодальностей і псевдошвидкостей для кожного з видимих навігаційних супутників, а також

					<i>НАУ 11 54 75 000 ПЗ</i>			
Зм.	Лист	№ докум.	Підп.	Дата				
Розробив	НовікМ.В.				Інтегрована інерціально-супутникова система навігації, що базується на принципах комплексної обробки інформації з використанням калманівської фільтрації	Лит.	Аркуш	Аркушів
Перевірив	Мар'ясоваТ.І.						74	152
Консульт.	ФіляшкінМ.К.					<i>IACU 608</i>		
Н. контр.	КозловаА.П.							
Затвердив	СинеглазовВ.М.							

зрушення і дрейф бортової шкали часу приймача СНС. У цьому випадку розмірність вектора стану залежить від числа видимих навігаційних супутників, а вимірювальна інформація може бути використана навіть від одного супутника. Розмірність вектора вимірів визначається числом видимих супутників (кожному супутникові відповідає два компоненти вектора вимірів). Для забезпечення можливості використання алгоритмів оптимальної лінійної фільтрації в жорсткозв'язаній схемі використовуються лінеаризовані представлення залежностей похибок в розрахованих по показанням ІНС оцінках псевдодальностей і псевдошвидкостей від похибок ІНС.

В обох із зазначених схем реалізації інваріантного методу одержувані за допомогою процедури оптимальної лінійної фільтрації оцінки похибок ІНС використовуються для внесення виправлень у вироблювані ІНС поточні значення вихідних навігаційних параметрів.

Для реалізації процедур оптимального комплексування інерціальної та супутникової систем навігації необхідно мати моделі еволюцій похибок окремих підсистем комплексу.

### 3.1 Моделі похибок інерціальних навігаційних систем

Рівняння похибок БІНС описують збурений режим роботи системи і є основою при аналізі її точності, при організації корекції, при побудові оптимальних навігаційних алгоритмів.

Матриця переходу від зв'язаної СК до географічної СК  $B(\psi, \vartheta, \gamma)$  має вид:

$$B(\psi, \vartheta, \gamma) = \begin{pmatrix} \sin \psi \cos \vartheta & \cos \psi \sin \gamma - \sin \psi \cos \gamma \sin \vartheta & \cos \psi \cos \gamma + \sin \psi \sin \gamma \sin \vartheta \\ \cos \psi \cos \vartheta & -\sin \psi \sin \gamma - \cos \psi \cos \gamma \sin \vartheta & -\sin \psi \cos \gamma + \cos \psi \sin \gamma \sin \vartheta \\ \sin \vartheta & \cos \gamma \cos \vartheta & -\sin \gamma \cos \vartheta \end{pmatrix}, \quad (3.1.1)$$

де  $\psi(t)$ ,  $\vartheta(t)$ ,  $\gamma(t)$  - кути курсу, тангажа та крену ЛА відповідно.

Матриця переходу від географічної СК до рухомої екваторіальної СК  $Q(\varphi)$  має вигляд:

$$Q(\varphi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi \\ 0 & -\sin \varphi & \cos \varphi \end{pmatrix},$$

де  $\varphi$ - географічна широта.

Матриця переходу від зв'язаної системи координат до рухомої екваторіальної СК  $C(\psi, \vartheta, \gamma, \varphi)$  задовольняє співвідношенням виду:

$$C(\psi, \vartheta, \gamma, \varphi) = Q(\varphi) \cdot B(\psi, \vartheta, \gamma).$$

При розв'язанні задач повітряної навігації як основні навігаційні параметри ЛА можна розглядати поточні географічні координати ( довготу  $\lambda$ , широту  $\varphi$  и висоту над поверхнею земного еліпсоїда  $H$ ), проекції шляхової швидкості  $V_E, V_N, V_h$ , а також елементи матриці переходу  $B(\psi, \vartheta, \gamma)$ , що характеризує орієнтацію ЛА у просторі.

Вказані навігаційні параметри задовольняє таким диференціальним рівнянням:

$$\left. \begin{aligned} \dot{\lambda} &= \frac{V_E(t)}{(R_1+h) \cos \varphi(t)} \\ \dot{\varphi} &= \frac{V_N(t)}{(R_2+h)} \\ \dot{h} &= V_h(t) \end{aligned} \right\}; \quad (3.1.2)$$

$$\dot{B} = B\Omega_c - \Omega_\Gamma B; \quad (3.1.3)$$

$$\dot{\bar{V}} = B\bar{a}_c - \Delta\bar{n}(t) + \bar{g}_T, \quad (3.1.4)$$

де (3.1.2) – рівняння для числення географічних координат;

(3.1.3) – матричне рівняння Пуассона для визначення матриці направляючих косинусів  $B(\psi, \vartheta, \gamma)$ ;

(3.1.4) – векторне рівняння відносно проекцій шляхової швидкості ЛА  $\bar{V} = (V_E, V_N, V_h)^T$ ;  $\bar{a}_c(t) = (a_{x1}(t), a_{y1}(t), a_{z1}(t))^T$  – вектор проекцій уявного прискорення початку зв'язаної СК на її осі;

$$\Omega_c = \begin{pmatrix} 0 & -\omega_{z1} & \omega_{y1} \\ \omega_{z1} & 0 & -\omega_{x1} \\ -\omega_{y1} & \omega_{x1} & 0 \end{pmatrix};$$

$$\Omega_\Gamma = \begin{pmatrix} 0 & -(\dot{\lambda} + u) \sin \varphi & (\dot{\lambda} + u) \cos \varphi \\ (\dot{\lambda} + u) \sin \varphi & 0 & \dot{\varphi} \\ -(\dot{\lambda} + u) \cos \varphi & -\dot{\varphi} & 0 \end{pmatrix};$$

$\omega_{x1}, \omega_{y1}, \omega_{z1}$  – проекції абсолютної кутової швидкості зв’язаної з ЛА СК на її осі;  $u$  – кутова швидкість обертання Землі;

$R_1$  и  $R_2$  – головні радіуси кривизни обраного земного еліпсоїда;

$$R_1 = a [1 - e^2 \sin^2 \varphi(t)]^{-\frac{1}{2}};$$

$$R_2 = a (1 - e^2) [1 - e^2 \sin^2 \varphi(t)]^{-\frac{3}{2}};$$

$a, e$  – велика піввісь и ексцентриситет земного еліпсоїда;

$\bar{g}_T = (g_{TE}, g_{TN}, g_{Th})^T$  – вектор проекцій прискорення сили ваги на осі географічної СК;

$\Delta \bar{n} = (\Delta n_E, \Delta n_N, \Delta n_h)^T$  – вектор проекцій суми переносного и кориолісова прискорень на осі географічної СК;

$$\Delta n_E = \frac{V_E V_h}{R_1 + h} - \frac{V_E V_N}{R_1 + h} t g \varphi + 2u (V_h \cos \varphi - V_N \sin \varphi);$$

$$\Delta n_N = \frac{V_N V_h}{R_2 + h} + \frac{V_E^2}{R_1 + h} t g \varphi + 2u V_E \sin \varphi;$$

$$\Delta n_h = -\frac{V_E^2}{R_1 + h} - \frac{V_N^2}{R_2 + h} - 2u V_E \cos \varphi;$$

$\bar{g}_T = [0, 0, g_e]^T$  – вектор проекцій нормального прискорення сили ваги на осі географічної СК  $g_e = \mu / a^2$ ,  $\mu = 398600, 44 \cdot 10^9 \text{ [м}^3/\text{с}^2]$

Маючи інформацію про вихідні координати та проекції шляхової швидкості ЛА, про вихідну матрицю орієнтації  $B_0$  (її визначення є предметом задачі початкового виставлення БІНС), а також про моделі прискорення сили ваги  $g^T(\varphi, \lambda, h)$ , на основі рівнянь (3.1.2)÷(3.1.4) с використанням поточних показів ДУС и акселерометрів можна отримати поточні значення шуканих навігаційних параметрів ЛА.

При точному завданні вихідних умов и при точній моделі прискорення сили ваги, а також при відсутності похибок інерціальних ДПП и похибок обчислення в наслідок інтегрування рівнянь (3.1.2)÷(3.1.4) будуть отримані істинні значення основних навігаційних параметрів ЛА.

Похибки завдання вихідних координат и проекцій шляхової швидкості ЛА, похибки початкового виставлення, аномальні варіації прискорення сили ваги, похибки інерціальних ДПП, методичні похибки алгоритмів обчислення и похибки через кінцеву довжину розрядній сітці обчислювача (похибки округлення) будуть приводити до похибок визначення шуканих навігаційних параметрів ЛА.

У лінійному наближенні еволюція похибок БІНС у визначенні основних

навігаційних параметрів у часі може бути описана лінійними диференціальними рівняннями похибок.

Рівняння похибок БІНС у визначенні координат впливає з динамічних рівнянь числення координат, що наведені в алгоритмах БІНС і мають вигляд:

$$\begin{aligned}\Delta \dot{R}_E &= \Delta V_E(t) \cdot \frac{R_3}{R \cos \varphi(t)} + \Delta R_N(t) \frac{V_E(t) \sin \varphi(t)}{R_3 R \cos^2 \varphi(t)} - \Delta h(t) \frac{R_3 V_E(t)}{R^2 \cos \varphi(t)}; \\ \Delta \dot{R}_N &= \Delta V_N(t) \cdot \frac{R_3}{R} - \Delta h(t) \frac{R_3 V_N(t)}{R^2}; \\ \Delta \dot{h} &= \Delta V_h(t);\end{aligned}\tag{3.1.5}$$

де  $\Delta R_E(t) = \Delta \lambda(t) R_3$ ,  $\Delta R_N(t) = \Delta \varphi(t) R_3$  – похибка БІНС у визначенні приведених координат місцезнаходження ЛА;

$\Delta \lambda(t), \Delta \varphi(t), \Delta H(t)$  – похибки БІНС у визначенні географічних координат;  
 $\Delta V_E(t), \Delta V_N(t), \Delta V_H(t)$  – похибки БІНС у визначенні проекції шляхової швидкості ЛА;

$R = R_3 + H$ ;  $R_3$  – радіус земної сфери;

Еволюція похибок БІНС у визначенні проекції шляхової швидкості ЛА  $\Delta V_E(t)$ ,  $\Delta V_N(t), \Delta V_h(t)$ , також може бути отримана з динамічних рівнянь числення шляхової швидкості в алгоритмах БІНС, і описується наступною системою рівнянь:

$$\begin{aligned}\Delta \dot{V}_E &= a_N \alpha_h - a_h \alpha_N + \sum_{i=1}^3 b_{1,i} \Delta a_i - \Delta V_h U(t) \cos \varphi + \Delta V_N U(t) \sin \varphi + \\ &+ \frac{\Delta R_N}{R_3} (U(t) (V_h \sin \varphi + V_N \cos \varphi)) - \left( \frac{\Delta V_E}{R \cos \varphi} + \frac{V_E \sin \varphi}{R \cos^2 \varphi} \frac{\Delta R_N}{R_3} \right) \times \\ &\times (V_h \cos \varphi - V_N \sin \varphi) + \frac{\Delta h V_E}{R^2} (V_h - V_N \tan \varphi); \\ \Delta \dot{V}_N &= -a_E \alpha_h + a_h \alpha_E + \sum_{i=1}^3 b_{2,i} \Delta a_i - \Delta V_E U(t) \sin \varphi - \Delta V_h \dot{\varphi}(t) - \\ &- \frac{\Delta R_N}{R_3} V_E U(t) \cos \varphi - \frac{\Delta V_N}{R} V_h - \left( \frac{\Delta V_E}{R \cos \varphi} + \frac{V_E \sin \varphi}{R \cos^2 \varphi} \frac{\Delta R_N}{R_3} \right) V_E \sin \varphi + \\ &+ \frac{\Delta h}{R^2} (V_E^2 \tan \varphi + V_N V_h); \\ \Delta \dot{V}_h &= a_E \alpha_N - a_N \alpha_E + \sum_{i=1}^3 b_{3,i} \Delta a_i + \Delta V_E U(t) \cos \varphi + \Delta V_N \dot{\varphi}(t) - \\ &- \frac{\Delta R_N}{R_3} V_E U(t) \sin \varphi + \frac{\Delta V_N}{R} V_N + \left( \frac{\Delta V_E}{R \cos \varphi} + \frac{V_E \sin \varphi}{R \cos^2 \varphi} \frac{\Delta R_N}{R_3} \right) V_E \cos \varphi + \\ &+ g_e \left( -\frac{2 \Delta h}{a} + \frac{3}{2} e^2 \sin \varphi \cos \varphi \frac{\Delta R_N}{R_3} \right) - \frac{\Delta h}{R^2} (V_E^2 + V_N^2),\end{aligned}\tag{3.1.6}$$

де  $b_{ij}$  ( $i, j=1, 2, 3$ ) – елементи матриці направляючих косинусів **B**;

$\Delta a_i$  ( $i=1, 2, 3$ ) – приведені похибки акселерометрів БІНС (з урахуванням похибок чисельного інтегрування рівняння у бортовому обчислювачі);

$a_H, a_E, a_N$  – поточні значення проекцій уявного прискорення початку зв’язаної СК на осі географічної СК;

$\alpha_H, \alpha_E, \alpha_N$  – похибки моделювання в БІНС орієнтації географічного координатного тригранника ( $\alpha_E$ ;

$\alpha_N$  – похибки побудови вертикалі,  $\alpha_H$  – азимутальна похибка);

$R = R_3 + H$  – поточна висота;

$$U(t) = 2\Omega_3 + \dot{\lambda}(t); \dot{\varphi}(t) = \frac{V_N}{R}; \dot{\lambda}(t) = \frac{V_E}{R \cos \varphi}.$$

Аналіз показує, що еволюція параметрів  $\alpha_h, \alpha_E, \alpha_N$  у часі описується наступною системою рівнянь:

$$\begin{aligned} \dot{\alpha}_E &= -\omega_N \alpha_h + \omega_h \alpha_N - \frac{\Delta V_N}{R} - \sum_{i=1}^3 b_{1,i} \varepsilon_i, \\ \dot{\alpha}_N &= -\omega_h \alpha_E + \omega_E \alpha_h + \frac{\Delta V_E}{R} - u \sin \varphi \frac{\Delta R_N}{R} - \sum_{i=1}^3 b_{2,i} \varepsilon_i, \\ \dot{\alpha}_h &= -\omega_E \alpha_N + \omega_N \alpha_E + \frac{\Delta V_E}{R} \tan \varphi + \left( u \cos \varphi + \frac{V_E}{R \cos^2 \varphi} \right) \frac{\Delta R_N}{R} - \sum_{i=1}^3 b_{3,i} \varepsilon_i, \end{aligned} \quad (3.1.7)$$

$$\text{де } \omega_E = -\dot{\varphi}(t), \omega_N = \left[ u + \dot{\lambda}(t) \right] \cos \varphi, \omega_h = \left[ u + \dot{\lambda}(t) \right] \sin \varphi,$$

$$\dot{\lambda} = \frac{V_E}{R \cos \varphi}; \dot{\varphi} = \frac{V_N}{R}; \varepsilon_i \text{ (i=1,2,3)} - \text{приведені похибки ДУС БІНС};$$

Аналіз показує, що похибки моделювання географічного тригранника  $\alpha_h, \alpha_E, \alpha_N$  зв’язані з похибками визначення координат  $\Delta R_N, \Delta R_E$  і похибками моделювання орієнтації рухливої екваторіальної СК  $\delta_\xi, \delta_\eta, \delta_\zeta$  такими співвідношеннями:

$$\begin{aligned} \alpha_E &= \delta_\xi - \frac{\Delta R_N}{R_3}; \\ \alpha_N &= \delta_\eta \cos \varphi - \delta \sin \varphi + \frac{\Delta R_E}{R_3} \cos \varphi; \\ \alpha_h &= \delta_\eta \sin \varphi - \delta_\zeta \cos \varphi + \frac{\Delta R_E}{R_3} \sin \varphi. \end{aligned}$$

Еволюція в часі похибок моделювання рухливої екваторіальної СК  $\delta_\xi, \delta_\eta, \delta_\zeta$  описується більш простими, ніж (3.1.7), рівняннями:

$$\begin{aligned} \dot{\delta}_\xi &= -(u + \dot{\lambda}) \delta_\zeta - \varepsilon_\zeta(t) \\ \dot{\delta}_\eta &= -\varepsilon_\eta(t) \\ \dot{\delta}_\zeta &= -(u + \dot{\lambda}) \delta_\xi - \varepsilon_\xi(t) \end{aligned} ;$$

$$\text{де } \dot{\lambda} = \frac{V_E(t)}{R \cos(t)}.$$

Якщо ввести в розгляд інерціальну прямокутну геоцентричну СК  $\xi_u \eta_u \zeta_u$ , вісь  $\eta_u$  якої збігається з віссю  $\zeta$ , а вісь  $\xi_u$  у момент  $t = 0$  лежить у площині Гринвіцького меридіана, то можна сказати, що похибки моделювання орієнтації такої СК  $\delta_{\xi_u}$ ,

$\delta_{\eta_u}, \delta_{\zeta_u}$  зв'язані з параметрами  $\delta_\xi, \delta_\eta, \delta_\zeta$  співвідношеннями виду:

$$\begin{aligned}\delta_\xi &= \delta_{\xi_u} \cos \lambda_* - \delta_{\zeta_u} \sin \lambda_* \\ \delta_\eta &= \delta_{\eta_u} \\ \delta_\zeta &= \delta_{\xi_u} \sin \lambda_* - \delta_{\zeta_u} \cos \lambda_*\end{aligned};$$

де  $\lambda_* = \omega t + \lambda(t)$ .

Рівняння, що описують еволюцію в часі похибок моделювання інерціальної СК  $\delta_{\xi_u}, \delta_{\eta_u}, \delta_{\zeta_u}$  виявляється досить простими:

$$\dot{\delta}_{\xi_u} = -\varepsilon_{\xi_u}(t); \dot{\delta}_{\eta_u} = -\varepsilon_{\eta_u}(t); \dot{\delta}_{\zeta_u} = -\varepsilon_{\zeta_u}(t),$$

$$\text{де } \begin{pmatrix} \varepsilon_{\xi_u} \\ \varepsilon_{\eta_u} \\ \varepsilon_{\zeta_u} \end{pmatrix} = \Delta \mathbf{C}(t) \mathbf{C}(t) \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{pmatrix};$$

$$\Delta \mathbf{C}(t) = \begin{pmatrix} \cos \lambda_* & -\sin \lambda_* & 0 \\ \sin \lambda_* & \cos \lambda_* & 0 \\ 0 & 0 & 1 \end{pmatrix} - \text{матриця переходу від рухливої еква-}$$

торіальної СК до інерціальної СК.

Таким чином, у моделі похибок БІНС можливе використання принаймні трьох груп параметрів, що характеризують похибки моделювання орієнтації СК:

$$\{\alpha_E, \alpha_N, \alpha_h\}, \{\delta_\xi, \delta_\eta, \delta_\zeta\}, \{\delta_{\xi_u}, \delta_{\eta_u}, \delta_{\zeta_u}\}.$$

Надалі в роботі використовуються параметри  $\alpha_E, \alpha_N, \alpha_h$ , що характеризують похибки моделювання географічної СК і мають найбільш наочну фізичну інтерпретацію. Цим параметрам відповідають рівняння еволюції (3.1.7).

Для замикання системи рівнянь похибок БІНС (3.1.5), (3.1.6), (3.1.7) необхідно вказати моделі еволюції приведених похибок ДПІ.

З урахуванням вигляду моделі еволюції похибок ДПІ, рівняння похибок БІНС (3.1.5), (3.1.6), (3.1.7) можуть бути замкнені наступними рівняннями відносно  $C_\omega, C_a, C_\varepsilon, D_a, \bar{\varepsilon}_A, \Delta \bar{a}_c$ :

$$\begin{aligned}\dot{C}_\omega &= \xi_{A\omega}(t); \dot{C}_a = \xi_{Aa}(t); \dot{C}_\varepsilon = \xi_{A\varepsilon}(t); \\ \dot{D}_a &= \xi_{Da}(t); \dot{\bar{\varepsilon}}_c = \bar{\xi}_A(t); \Delta \dot{\bar{a}}_c = \bar{\xi}_{\Delta a}(t),\end{aligned}\tag{3.1.8}$$

де  $\xi_{A\omega}(t); \xi_{Aa}(t);$

$\xi_{A\varepsilon}(t); \xi_{Da}(t); \bar{\xi}_A(t); \bar{\xi}_{\Delta a}(t)$  – білошумні збурення відповідної розмірності, які характеризують дрейф квазістаціонарних параметрів моделі ДПІ.



Повертаючись до моделей похибок БІНС відзначимо, що коли вектор-стовпець похибок БІНС  $\bar{X}(t)$  прийняти у вигляді:

$$\bar{X} = (\Delta R_E, \Delta R_N, \Delta h, \Delta V_E, \Delta V_N, \Delta V_h, \alpha_E, \alpha_N, \alpha_h, \varepsilon_{c1}, \varepsilon_{c2}, \varepsilon_{c3}, \Delta a_{c1}, \Delta a_{c2}, \Delta a_{c3},)^T,$$

то модель еволюції похибок БІНС може бути подана у компактній формі

$$\dot{\bar{X}} = F\bar{X}(t) + G\bar{\xi}(t), \quad (3.1.9)$$

де  $F$  та  $G$  – матриці  $15 \times 15$  і  $15 \times 21$  відповідно;  $\bar{\xi}(t)$  – вектор-стовпець розмірності 21, компонентами якого є незалежні Гауссівські «білі» шуми з нульовими середніми значеннями і одиничними дисперсіями.

Відмінні від нуля елементи матриці  $F$  мають вигляд:

$$\begin{aligned} f_{1,2} &= \frac{\dot{\lambda}}{R_3} tg\varphi; f_{1,3} = \frac{-\dot{\lambda}R_3}{R}; f_{1,4} = \frac{R_3}{R \cos \varphi}; \\ f_{2,3} &= \frac{-\dot{\varphi}R_3}{R}; f_{2,5} = \frac{R_3}{R}; f_{3,6} = 1; \\ f_{4,2} &= \frac{2u+\dot{\lambda}}{R_3} (V_h \sin \varphi + V_N \cos \varphi) - \frac{\dot{\lambda}}{R_3} tg\varphi (V_h \cos \varphi - V_N \sin \varphi); \\ f_{4,3} &= \frac{V_E}{R^2} (V_h - V_N tg\varphi); \\ f_{4,4} &= \frac{V_N \sin \varphi - V_h \cos \varphi}{R \cos \varphi}; \\ f_{4,5} &= (2u + \dot{\lambda}) \sin \varphi; f_{4,6} = - (2u + \dot{\lambda}) \cos \varphi; \\ f_{4,8} &= -a_h; f_{4,9} = a_N; f_{4,13} = b_{1,1}; f_{4,14} = b_{1,2}; f_{4,15} = b_{1,3}; \\ f_{5,2} &= -\frac{2u+\dot{\lambda}}{R_3} V_E \cos \varphi - \frac{V_E^2}{RR_3} tg^2\varphi; \\ f_{5,3} &= \frac{V_E^2 tg\varphi + V_h V_N}{R^2}; \\ f_{5,4} &= - (2u + \dot{\lambda}) \sin \varphi; f_{5,5} = -\frac{V_h}{R}; \\ f_{5,6} &= -\dot{\varphi}(t); f_{5,7} = a_h; f_{5,9} = -a_E; f_{5,13} = b_{2,1}; f_{5,14} = b_{2,2}; f_{5,15} = b_{2,3}; \\ f_{6,2} &= -2u \frac{V_E \sin \varphi}{R} + \frac{3g_e}{2R_3} e^2 \sin \varphi \cos \varphi; \\ f_{6,3} &= -\frac{2g_e}{a} - \frac{V_E^2 + V_N^2}{R^2}; f_{6,4} = (2u + \dot{\lambda}) \cos \varphi; \\ f_{6,5} &= \dot{\varphi}(t) + \frac{V_N}{R}; f_{6,7} = -a_N; f_{6,8} = a_E; f_{6,13} = b_{3,1}; f_{6,14} = b_{3,2}; f_{6,15} = b_{3,3}; \\ f_{7,5} &= -\frac{1}{R}; f_{7,8} = \omega_h; f_{7,9} = -\omega_N; \\ f_{7,10} &= -b_{1,1}; f_{7,11} = -b_{1,2}; f_{7,12} = -b_{1,3}; \\ f_{8,2} &= -\frac{u}{R} \sin \varphi; f_{8,4} = \frac{1}{R}; f_{8,7} = -\omega_h; f_{8,9} = \omega_E; \\ f_{8,10} &= -b_{2,1}; f_{8,11} = -b_{2,2}; f_{8,12} = -b_{2,3}; \\ f_{9,2} &= \frac{1}{R_3} (u \cos \varphi + \frac{\dot{\lambda}}{\cos \varphi}); \\ f_{9,4} &= \frac{tg\varphi}{R}; f_{9,7} = \omega_N; f_{9,8} = -\omega_E; \\ f_{9,10} &= -b_{3,1}; f_{9,11} = -b_{3,2}; f_{9,12} = -b_{3,3}. \end{aligned}$$

(3.1.10)

Зм.	Лист	№ докум.	Підп.	Дата

HAУ 11 54 75 000 ПЗ

Аркун

82

[illegible]

Відрізні від нуля елементи матриці  $G(15 \times 21)$  задовольняють таким співвідношенням:

$$\begin{aligned} g_{i,i} &= \sigma_i, & i &= 1, \dots, 15; \\ g_{i+3,j+18} &= b_{i,j} \sigma_a, & i &= 1, 2, 3, j = 1, 2, 3; \\ g_{i+6,j+15} &= -\sigma_\omega b_{i,j} & i &= 1, 2, 3, j = 1, 2, 3; \end{aligned} \quad (3.1.11)$$

де  $\sigma_1 \div \sigma_{15}$  – середньоквадратичні значення (СКЗ) білошумних збурень, що характеризують вплив різних факторів ( $\sigma_1 \div \sigma_3$  – похибок численного інтегрування рівняння (3.1.2);

$\sigma_4 \div \sigma_6$  – підсумковий ефект аномалій гравітаційного поля и похибок численного інтегрування рівняння (3.1.4),

$\sigma_7 \div \sigma_9$  – похибок численного інтегрування рівняння для параметрів орієнтації (3.1.3);

$\sigma_{10} \div \sigma_{15}$  – випадкового дрейфу квазістаціонарних зведених погрішностей ДП  $\bar{\varepsilon}_A$  и  $\Delta \bar{0}_A$ );

$\sigma_a, \sigma_\omega$  – СКЗ білошумних складових погрішностей акселерометрів і ДКШ БІНС.

Елементи матриць **F** и **G**, що впливає з аналізу співвідношень (3.1.10) и (3.1.11), залежать від поточних значень навігаційних параметрів польоту ЛА.

Безперервної моделі еволюції похибок БІНС (3.1.9) відповідає такий дискретний аналог:

$$\bar{X}_{k+1} = \Phi_k \bar{X}_k + G_k \bar{\xi}_k,$$

де  $\Phi_k = E + F(t_k) \Delta t$ ,  $G_k = G(t_k) \cdot \Delta t$ ;

$\Delta t$  – крок дискретизації часу;

$E$  – одинична матриця  $15 \times 15$ .

## 3.2 Математичні моделі похибок супутникової системи навігації

Для опису похибок СНС у визначенні координат і проекцій шляхової швидкості ЛА пропонується використовувати математичні моделі, що містять

Марківські і гаусовські складові похибок:

$$\begin{aligned}\Delta R_{Es,k} &= \Delta R_{Ec,k} + \frac{\sigma_{Rs}}{\cos \varphi_k} \eta_{REs,k} + \frac{\sigma_{\delta Rs}}{\cos \varphi_k} \eta_{\delta RE,k}; \\ \Delta R_{Ns,k} &= \Delta R_{Nc,k} + \sigma_{Rs} \eta_{RNs,k} + \sigma_{\delta Rs} \eta_{\delta RN,k}; \\ \Delta H_{s,k} &= \Delta H_{c,k} + \sigma_{Hs} \eta_{Hs,k} + \sigma_{\delta Hs} \eta_{\delta H,k}; \\ \Delta V_{ls,k} &= \Delta V_{lc,k} + \sigma_{Vs} \eta_{Vls,k} + \sigma_{\delta Vs} \eta_{\delta Vls,k}, \text{ при } l = E, N, H;\end{aligned}\quad (3.2.1)$$

де  $\Delta R_{ls,k}$ , ( $l=E,N$ );  $\Delta H_{s,k}$ ;  $\Delta V_{ls,k}$  ( $l=E,N,H$ ) – похибки СНС у визначенні приведених координат, висоти і складових шляхової швидкості ЛА;

$\Delta R_{lc,k}$  ( $l=E,N$ );  $\Delta H_{c,k}$ ;  $\Delta V_{lc,k}$  ( $l=E,N,H$ ) – корельовані (Марківські) складові похибок СНС;

$\sigma_{Rs}$ ,  $\sigma_{Hs}$ ,  $\sigma_{Vs}$  – СКЗ білошумових складових похибок СНС;

$\sigma_{\delta Rs}$ ,  $\sigma_{\delta Hs}$ ,  $\sigma_{\delta Vs}$  – СКЗ додаткових білошумових складових похибок СНС, що виникають тільки за умови, що  $t_k$  – момент зміни сузір'я навігаційних супутників;

$\eta_{Rls,k}$ ,  $\eta_{\delta Rls,k}$ , ( $l=E,N$ );  $\eta_{Hs,k}$ ,  $\eta_{\delta Hs,k}$ ;  $\eta_{Vls,k}$ ,  $\eta_{\delta Vls,k}$  ( $l=E,N,H$ ) – стандартні білі дискретні шуми зі СКЗ.

Корельовані складові похибок СНС описуються наступними співвідношеннями:

$$\begin{aligned}\Delta R_{Ec,k} &= W_R \Delta R_{Ec,k-1} + q_R \frac{\sigma_{Rc}}{\cos \varphi_k} \eta_{REc,k} + \frac{\sigma_{\delta Rc}}{\cos \varphi_k} \eta_{\delta REc,k}; \\ \Delta R_{Nc,k} &= W_R \Delta R_{Nc,k-1} + q_R \sigma_{Rc} \eta_{RNc,k} + \sigma_{\delta Rc} \eta_{\delta RNc,k}; \\ \Delta H_{c,k} &= W_R \Delta H_{c,k-1} + q_R \sigma_{Hc} \eta_{Hc,k} + \sigma_{\delta Hc} \eta_{\delta Hc,k}; \\ \Delta V_{lc,k} &= W_V \Delta V_{lc,k-1} + q_V \sigma_{Vc} \eta_{Vlc,k} + \sigma_{\delta Vc} \eta_{\delta Vlc,k}, \text{ при } l = E, N, H,\end{aligned}\quad (3.2.2)$$

де

$$\begin{aligned}W_R &= e^{-(\lambda_s V_{\text{ш}} + \lambda_{st}) \Delta t}; q_R = [1 - \exp(-2(\lambda_s V_{\text{ш}} + \lambda_{st}) \Delta t)]^{0,5}; \\ W_V &= e^{-\lambda_v \Delta t}; q_V = [1 - \exp(-2\lambda_v \Delta t)]^{0,5};\end{aligned}$$

$\lambda_s$  – показник просторової кореляції похибки СНС за координатами;

$\lambda_v$ ,  $\lambda_{st}$  – показник часової кореляції похибок СНС за швидкістю та за координатами;

$V_{\text{ш}}$  – шляхова швидкість ЛА;

$\Delta t$  – дискрета оновлення вихідної інформації СНС у часі;

$\sigma_{Rc}$ ,  $\sigma_{Hc}$ ,  $\sigma_{Vc}$  – СКЗ корельованих складових похибок СНС;

$\sigma_{\delta Rc}$ ,  $\sigma_{\delta Hc}$ ,  $\sigma_{\delta Vc}$  – СКЗ додаткових гаусовських збурень у моменти зміни сузір'я навігаційних супутників;  $\eta_{Rlc,k}$ ,  $\eta_{\delta Rlc,k}$  ( $l=E,N$ ),  $\eta_{Hc,k}$ ,  $\eta_{\delta Hc,k}$ ,  $\eta_{Vlc,k}$

$\eta_{\delta V l c, k}$  ( $l=E, N, H$ ) – стандартні центровані дискретні білі шуми з одиничною інтенсивністю.

Для стандартного режиму СНС типу GPS NAVSTAR можуть бути рекомендовані наступні значення параметрів моделей (3.2.1), (3.2.2):

$$\begin{aligned}\lambda_s &= 4 \cdot 10^{-6} \text{ м}^{-1}; \lambda_{st} = 5 \cdot 10^{-4} \text{ с}^{-1}; \lambda_V = (0,0017 \div 0,05) \text{ с}^{-1}; \\ \sigma_{Rs} &= (1 \div 3) \text{ м}; \sigma_{Hs} = (1,5 \div 4) \text{ м}; \sigma_{Vs} = (0,01 \div 0,05) \text{ м/с}; \\ \sigma_{\delta Rs} &= (1 \div 4) \text{ м}; \sigma_{\delta Vs} = (0,02 \div 0,2) \text{ м/с}; \sigma_{Rc} = (5 \div 7) \text{ м}; \\ \sigma_{Hc} &= (7 \div 10) \text{ м}; \sigma_{Vc} = (0,02 \div 0,3) \text{ м/с}; \sigma_{\delta Rc} = (2 \div 5) \text{ м}; \\ \sigma_{\delta Vc} &= (0,01 \div 0,02) \text{ м/с}; \sigma_{\delta Hc} = (3 \div 7) \text{ м}.\end{aligned}$$

### 3.3 Математичні моделі похибок барометричного висотоміра

Похибка барометричного висотоміра (БВ) у визначенні абсолютної висоти ЛА може бути описана співвідношенням вигляду:

$$\Delta h(t_k) = \Delta h_{\text{вс}} + \sigma_h \eta_{n,k}, \quad (3.3.1)$$

де  $\Delta h_{\text{вс}}$  – квазістаціонарна похибка виміру барометричної висоти, що обумовлена неточністю початкової виставки, а також змінами температури та тиску атмосфери за час польоту;

$\sigma_h$  – СКЗ флюктуаційної складової похибки, що обумовлена пульсаціями тиску й іншими факторами;

$\eta_{n,k}$  – дискретний білий шум з одиничною інтенсивністю.

У свою чергу дискретна модель еволюції квазістаціонарної похибки БВ може бути представлена в наступному вигляді:

$$\Delta h_{c,k} = \Delta h_{c,k-1} + \sigma_{\xi A} \xi_{k-1}, \quad (3.3.2)$$

де  $\sigma_{\xi A}$  – заданий параметр;

$\xi_{k-1}$  – стандартний дискретний білий шум з одиничною інтенсивністю.

Аналіз показує, що для моделі похибок БВ (3.3.1), (3.3.2) можна рекомендувати наступні значення параметрів:

$$\begin{aligned}\sigma_h &= (0,5 \div 1)_m; \\ \sigma_{\xi c} &= (0,05 \div 0,02)_m; \\ \sigma_{\Delta h c,0} &= (3 \div 5)_m;\end{aligned}$$

де  $\sigma_{\Delta h c,0}$  – СКЗ похибки  $\Delta h c$  у початковий момент часу.

### 3.4 Розробка алгоритмів оптимальної комплексної обробки навігаційної інформації

Загальною вимогою для організації процесу комплексування є наявність математичних моделей підсистем, що підлягають комплексуванню. Сучасний стан обчислювальної техніки, знань в області інерціальної та супутникової навігації дозволяють скласти досить повні й адекватні моделі цих систем. У комплексі системи описуються на рівні їхніх похибок. Таким чином, для нормальної роботи комплексу потрібний адекватний опис похибок підсистем, включаючи неконтрольовані джерела похибок. Розробка алгоритмів комплексної обробки навігаційної інформації здійснюватиметься з використанням моделей похибок ІНС ((3.1.5), (3.1.6), (3.1.7)), СНС ((3.2.1)) та баровисотоміра ((3.3.1)-(3.3.2)).

При розгляді слабкозв'язаної схеми інваріантного алгоритму комплексної обробки навігаційної інформації для розглянутого складу навігаційних підсистем рекомендується використовувати розширений вектор стану, що включає 22 компоненти, у тому числі: 15 компонент – помилки нанотехнологічної БІНС, одна – систематична помилка БВ, 6 компонент – корельовані помилки

СНС у визначенні координат і проекцій швидкості:

$$\bar{X} = \begin{bmatrix} \Delta R_E \\ \Delta R_N \\ \Delta h \\ \Delta V_E \\ \Delta V_N \\ \Delta V_h \\ \alpha_E \\ \alpha_N \\ \alpha_h \\ \varepsilon_{c1} \\ \varepsilon_{c2} \\ \varepsilon_{c3} \\ \Delta a_{c1} \\ \Delta a_{c2} \\ \Delta a_{c3} \\ \Delta h_{\text{БВ}} \\ \Delta R_{Ec} \\ \Delta R_{Nc} \\ \Delta h_c \\ \Delta V_{Ec} \\ \Delta V_{Nc} \\ \Delta V_{hc} \end{bmatrix} = \begin{bmatrix} \text{Помилка приведенної координати Схід} \\ \text{Помилка приведенної координати Північ} \\ \text{Помилка по висоті} \\ \text{Помилка по швидкості Схід} \\ \text{Помилка по швидкості Північ} \\ \text{Помилка по вертикальній швидкості} \\ \text{Помилка моделювання географіч. тригранника Схід} \\ \text{Помилка моделювання географіч. тригранника Північ} \\ \text{Помилка моделювання географіч. тригранника Висота} \\ \text{Дрейф гіроскопа Схід} \\ \text{Дрейф гіроскопа Північ} \\ \text{Дрейф гіроскопа висота} \\ \text{Дрейф акселерометра Схід} \\ \text{Дрейф акселерометра Північ} \\ \text{Дрейф акселерометра по висоті} \\ \text{Помилка баровисотоміра} \\ \text{Корельована помилка координати СНС Схід} \\ \text{Корельована помилка координати СНС Північ} \\ \text{Корельована помилка координати СНС по висоті} \\ \text{Корельована помилка швидкості СНС Схід} \\ \text{Корельована помилка швидкості СНС Північ} \\ \text{Корельована помилка швидкості СНС по висоті} \end{bmatrix} \quad (3.4.1)$$

Дискретна модель еволюції вектора стану  $\bar{X}_p$ , що отримується на основі (3.3.2), (3.1.7), (3.2.1), має вигляд:

$$\bar{X}_{p,k+1} = \Phi_{p,k} \bar{X}_{p,k} + G_{p,k} \bar{\xi}_k \quad (3.4.2)$$

де  $\Phi_{p,:} = E + F_{p,k} \Delta t$ ;

$\bar{\xi}_k$  – 34-мірний вектор центрованих гаусових дискретних білих шумів з одиничною інтенсивністю;

$$F_{p,k} = \begin{pmatrix} F_k & . & . & . & . & . & . & . & . \\ . & 0 & . & . & . & . & . & . & . \\ . & . & W_R & . & . & . & . & . & . \\ . & . & . & W_R & . & . & . & . & . \\ . & . & . & . & W_R & . & . & . & . \\ . & . & . & . & . & W_V & . & . & . \\ . & . & . & . & . & . & W_V & . & . \\ . & . & . & . & . & . & . & W_V & . \\ . & . & . & . & . & . & . & . & W_V \end{pmatrix};$$

$$G_{p,k} = \begin{pmatrix} G_k & . & . \\ . & \sigma_{\text{БВ}} \sqrt{\Delta t} & . \\ . & . & G_{s,k} \end{pmatrix};$$

$$G_{S,k} = \begin{pmatrix} \frac{q_R \sigma_{Rc}}{\cos \varphi_k} & . & . & . & . & . & \frac{\mu \sigma_{\delta Rc}}{\cos \varphi_k} & . & . & . & . & . \\ . & q_R \sigma_{Rc} & . & . & . & . & . & \mu \sigma_{\delta Rc} & . & . & . & . \\ . & . & q_R \sigma_{hc} & . & . & . & . & . & \mu \sigma_{\delta hc} & . & . & . \\ . & . & . & q_V \sigma_{Vc} & . & . & . & . & . & \mu \sigma_{\delta Vc} & . & . \\ . & . & . & . & q_V \sigma_{Vc} & . & . & . & . & . & \mu \sigma_{\delta Vc} & . \\ . & . & . & . & . & q_V \sigma_{Vc} & . & . & . & . & . & \mu \sigma_{\delta Vc} \end{pmatrix}$$

$\mu = 1$  в момент зміни сузір'я  $t_k^*$ ,  $\mu = 0$  в будь який інший момент  $t_k$ .

До складу вектора спостережень пропонується включити 8 компонент, у тому числі різницю оцінок висоти, видаваних нанотехнологічної БІНС і БВ, 3 різниці координат і 3 різниці складові швидкості, вироблюваних нанотехнологічної БІНС і СНС відповідно, а також різниця оцінок висоти, видаваних БВ і СНС відповідно:

$$\bar{Y}_k = \begin{pmatrix} \tilde{h}_k - \tilde{h}_{\text{БВ},k}, \\ \tilde{R}_{E,K} - \tilde{R}_{ES,k}, \\ \tilde{R}_{N,K} - \tilde{R}_{NS,k}, \\ \tilde{h}_k - \tilde{h}_{s,k}, \\ \tilde{V}_{E,k} - \tilde{V}_{ES,k}, \\ \tilde{V}_{N,k} - \tilde{V}_{NS,k}, \\ \tilde{V}_{h,k} - \tilde{V}_{hS,k}, \\ \tilde{h}_{\text{БВ}} - \tilde{h}_{s,k} \end{pmatrix} \quad (3.4.3)$$

Рівняння спостережень у компактній формі має вигляд:



$$\bar{Y}_k = H\bar{X}_{p,k} + Q_{p,k}\bar{\eta}_k,$$

де  $\bar{\eta}_k$  – 13-мірний вектор стандартних центрованих гаусових дискретних білих шумів з одиничною інтенсивністю;

$$H = \begin{pmatrix} . & . & 1 & . & . & . & \dots & -1 & . & . & . & . & . & . \\ 1 & . & . & . & . & . & \dots & . & -1 & . & . & . & . & . \\ . & 1 & . & . & . & . & \dots & . & . & -1 & . & . & . & . \\ . & . & 1 & . & . & . & \dots & . & . & . & -1 & . & . & . \\ . & . & . & 1 & . & . & \dots & . & . & . & . & -1 & . & . \\ . & . & . & . & 1 & . & \dots & . & . & . & . & . & -1 & . \\ . & . & . & . & . & 1 & \dots & . & . & . & . & . & . & -1 \\ . & . & . & . & . & . & \dots & 1 & . & . & . & -1 & . & . \end{pmatrix}$$

$$Q_{p,k} = \begin{pmatrix} \sigma_{\text{БВ}} & . & . & . & . & . & . & . & . & . & . & . & . & . \\ . & \frac{\sigma_{Rs}}{\cos \varphi_k} & . & . & . & . & . & \frac{\mu\sigma_{\delta Rs}}{\cos \varphi_k} & . & . & . & . & . & . \\ . & . & \sigma_{Rs} & . & . & . & . & . & \mu\sigma_{\delta Rs} & . & . & . & . & . \\ . & . & . & \sigma_{hs} & . & . & . & . & . & \mu\sigma_{\delta Rs} & . & . & . & . \\ . & . & . & . & \sigma_{Vs} & . & . & . & . & . & \mu\sigma_{\delta Vs} & . & . & . \\ . & . & . & . & . & \sigma_{Vs} & . & . & . & . & . & \mu\sigma_{\delta Vs} & . & . \\ . & . & . & . & . & . & \sigma_{Vs} & . & . & . & . & . & \mu\sigma_{\delta Vs} & . \\ \sigma_{\text{БВ}} & . & . & \sigma_{hs} & . & . & . & . & . & \mu\sigma_{\delta Rs} & . & . & . & . \end{pmatrix};$$

$\mu = 1$  в момент зміни сузір'я  $t_k^*$ ,  $\mu = 0$  в будь який інший момент  $t_k$ .

Для оцінки вектора стану системи (3.4.1) за спостереженнями (3.4.3) пропонується використовувати процедуру дискретного оптимального фільтра Калмана, у якій екстраполяція оцінки вектора стану  $\hat{\bar{X}}_{p,k-1}$  і коваріаційної матриці помилок оцінки  $P_{k-1}$  здійснюється відповідно до формул:

$$\begin{aligned} \hat{\bar{X}}_{p,k} &= \Phi_{p,k-1} \hat{\bar{X}}_{p,k-1}^+, \\ P_k &= \Phi_{p,k-1} P_{k-1}^+ \Phi_{p,k-1}^T + G_{p,k-1} G_{p,k-1}^T; \end{aligned} \quad (3.4.4)$$

а корекція виконується згідно співвідношень виду:

$$\begin{aligned} \hat{\bar{X}}_{p,k}^+ &= \hat{\bar{X}}_{p,k} + K_k(\bar{Y}_k - H\hat{\bar{X}}_{p,k}) \\ P_k^+ &= (E - K_k H) P_k (E - K_k H)^T + K_k Q_{p,k} Q_{p,k}^T K_k^T \end{aligned} \quad (3.4.5)$$

де верхній індекс «+» є ознака корекції, виконаної на відповідному кроці;  
 $K_k = P_k H^T (H P_k H^T + Q_{p,k} Q_{p,k}^T)^{-1}$  – матричний коефіцієнт підсилення фільтра.

Процедура (3.4.4), (3.4.5) може бути доповнена операцією обмеження знизу значень діагональних елементів матриці коваріації  $P_k^+$ .

$P_{k,i}^+$  якщо  $P_{k,i}^+ \geq \gamma_i$ ;

$$\hat{P}_{k,i}^+ = \begin{cases} P_{k,i}^+, & \text{при } P_{k,i}^+ \geq \gamma_i \\ \gamma_i, & \text{при } P_{k,i}^+ < \gamma_i \end{cases},$$

де  $P_{k,i}^+$  –  $i$ -й діагональний елемент матриці  $P_k^+$ ;

$\gamma_i$  якщо  $i = 1, \dots, Nr$  – задані нижні границі значень діагональних елементів.

Як відзначалося вище, при комплексній обробці навігаційної інформації необхідно здійснювати алгоритмічний контроль цілісності СНС. Можна вказати, принаймні, два підходи до розв’язання задачі контролю цілісності СНС. Перший підхід зводиться до контролю за допуском вихідної позиційної і швидкісної інформації СНС. З цією метою здійснюється порівняння поточних показань СНС з географічних координат і проекцій шляхової швидкості з відповідними оцінками зазначених навігаційних параметрів, екстрапольованих з використанням навігаційних рівнянь (3.1.2) з попереднього кроку (приймається гіпотеза про те, що оцінки навігаційних параметрів на попередньому кроці достовірні). Для такого підходу значення допусків можуть бути встановлені з урахуванням енергетичних можливостей ЛА.

Другий підхід впливає з теоретичних моделей процесу оптимальної калмановської фільтрації і передбачає аналіз характеристик так називаної оновленої послідовності спостережень

$$\Delta \bar{Y}_{j,k} = \bar{Y}_{j,k} - H \bar{X}_{p,k}, \quad j = 1, 2, \quad (3.4.6)$$

де  $\bar{Y}_{j,k}$  ( $j = 1, 2$ ) – підвектори вектора спостережень  $\bar{Y}_{p,k}$ , які відповідають позиційної (компоненти  $2 \div 4$ ) і швидкісний (компоненти  $5 \div 7$ ) вихідної інформації СНС;

$h_{1,i,j} = h_{i+1,j}$  ( $i = 1, 2, 3, j = 1, \dots, 22$ );  $h_{2,i,j} = h_{i+n,j}$  ( $i = 1, 2, 3, j = 1, \dots, 22$ ).

Рішення про відмовлення позиційного або швидкісного каналів СНС при-

ймається на основі аналізу умов нормальної роботи фільтра:

$$\frac{Sp(\Delta\bar{Y}_{j,k}\Delta\bar{Y}_{j,k}^T)}{Sp(H_j P_k H_j^T + R_j)} < \delta, \quad j = 1, 2, \quad (3.4.7)$$

де  $Sp()$  – символ сліду матриці;

$\delta$  – задана константа ( $\delta \geq 10$ );

$R_j$  ( $j = 1, 2$ ) – коваріаційні матриці відповідних підвекторів випадкових помилок вимірів.

Якщо умови не виконуються на  $k$ -му кроці для будь якого  $j$ , то відповідний підвектор спостережень ігнорується при обробці інформації на цьому кроці.

Одержувані з виходу фільтра оцінки помилки БІНС використовуються для виправлення вихідних навігаційних параметрів БІНС. Алгоритм виправлення оцінок координат і проекцій швидкості має вигляд:

$$\begin{aligned} h_i^+ &= h_i^- - \Delta\hat{h}_i; \\ \varphi_i^+ &= \varphi_i^- - \frac{\Delta\hat{R}_{Ni}}{R_3}; \\ \lambda_i^+ &= \lambda_i^- - \frac{\Delta\hat{R}_{Ei}}{R_3}; \\ V_{l,i}^+ &= V_{l,i}^- - \Delta\hat{V}_l, l = E, N, h, \end{aligned} \quad (3.4.8)$$

де верхніми індексами «-» і «+» позначені оцінки вихідних навігаційних параметрів до виправлення і після виправлення відповідно;

$\Delta\hat{R}_{Ei}$ ,  $\Delta\hat{R}_{Ni}$ ,  $\Delta\hat{h}_i$ ,  $\Delta\hat{V}_E$ ,  $\Delta\hat{V}_N$ ,  $\Delta\hat{V}_h$  – поточні оцінки помилок нанотехнологічної БІНС, одержувані на виході фільтра.

Виправлення одержуваної в нанотехнологічній БІНС оцінки матриці орієнтації  $B_i$  виконується за допомогою наступної процедури:

$$\widehat{B}_i^+ = \Delta B_i \widehat{B}_i^-, \quad (3.4.9)$$

$$\text{де } \Delta B_i = \begin{pmatrix} 1 & -\hat{\alpha}_{h,i} & \hat{\alpha}_{N,i} \\ \hat{\alpha}_{h,i} & 1 & -\hat{\alpha}_{E,i} \\ -\hat{\alpha}_{N,i} & \hat{\alpha}_{E,i} & 1 \end{pmatrix},$$

$\hat{\alpha}_{E,i}$ ,  $\hat{\alpha}_{N,i}$ ,  $\hat{\alpha}_{h,i}$  – поточні оцінки помилок БІНС у визначенні орієнтації географічної системи координат, одержувані на виході фільтра.

Після виконання операції (3.4.9) варто перевіряти умови ортогональності матриці  $\widehat{B}_i^+$  і при необхідності робити ортогоналізацію оцінки матриці направляючих косинусів  $\widehat{B}_i^+$ , наприклад, за допомогою процедури, запропонованої в роботі [2].

Як відзначалося вище, для випадку грубих ДПІ роботу нанотехнологічної БІНС необхідно періодично коректувати. Період корекції  $T_{\text{кор}}$  може вибиратися з умови:

$$\Delta\alpha(T_{\text{кор}}) = \Delta\alpha_{\text{доп}},$$

де  $\Delta\alpha(T_{\text{кор}})$  – оцінка максимальної помилки моделювання орієнтації осей географічної системи координат у нанотехнологічній БІНС;

$\Delta\alpha_{\text{доп}}$  – припустиме значення помилки, що забезпечує збереження лінійності моделі еволюції помилок нанотехнологічної БІНС.

Аналіз показує, що для значень  $T_{\text{кор}}$ , які задовольняють умові  $T_{\text{кор}} \ll T_{\text{ш}}$  ( $T_{\text{ш}} = 84,4$  хв – період маятника Шулера), для оцінки  $\Delta\alpha(T)$  може бути використана формула виду:

$$\Delta\alpha_{\text{доп}} = \varepsilon T_{\text{кор}} + \Delta\alpha^*(T_{\text{кор}}),$$

$$\text{де } \alpha^*(T_{\text{кор}}) = \left[ (\Delta\alpha_0 g + \Delta a) \frac{T_{\text{кор}}^2}{2} + g \frac{\varepsilon T_{\text{кор}}^3}{6} \right] R_3^{-1};$$

$\Delta\alpha_0$  – максимальне значення помилки початкової виставки нанотехнологічної БІНС ;

$\Delta a, \varepsilon$  – максимальне значення помилки інерціальних ДПІ нанотехнологічної БІНС.

У момент корекції роботи БІНС виконуються наступні операції:

- вносяться виправлення в обчислені значення оцінок координат, проєкцій швидкості і матриці орієнтації  $B$  у відповідності з формулами (3.4.8), (3.4.9);
- обновляються оцінки приведених помилок датчиків нанотехнологічної БІНС за формулами:

$$\begin{aligned} \varepsilon_{i,l}^* &= \varepsilon_{i,l-1}^* + \widehat{\varepsilon}_{i,l}^* \\ \Delta a_{i,l}^* &= \Delta a_{i,l-1}^* + \Delta \widehat{a}_{i,l}, \end{aligned} \quad (i = 1, 2, 3),$$

де  $l$  – номер точки корекції ( $l = 1, 2, \dots$ );  $\varepsilon_{i,0}^* = \Delta a_{i,0}^* = 0$ ,  $i = 1, 2, 3$ ;  
 $\widehat{\varepsilon}_{i,e}, \Delta \widehat{a}_{i,e}$  – оцінки помилок у точці корекції нанотехнологічної БІНС;

- онулюються компоненти вектора стану  $X_p$   $1 \div 15$ , що відповідають помилкам нанотехнологічної БІНС.

Поточні оцінки приведених помилок ДКШ і акселерометрів  $\varepsilon_i^*$ ,  $\Delta a_i^*$  ( $i = 1, 2, 3$ ) використовуються в обчислювальних алгоритмах БІНС для внесення в показання ДПІ виправлень виду:

$\Delta \alpha_i = \Delta t_{\text{оптим}} \varepsilon_i^*$  і  $\Delta v_i = \Delta t_{\text{оптим}} \Delta a_i^*$  ( $i = 1, 2, 3$ ), де  $\Delta t_{\text{оптим}}$  – крок опитування ДПІ.

					НАУ 11 54 75 000 ПЗ	Аркул
Зм.	Лист	№ докум.	Підп.	Дата		93

## 4 Дослідження розроблених алгоритмів

### 4.1 Дослідження моделі похибок БІНС

Перевіримо модель та впливи різних складових похибок, на модель БІНС, зобразимо їх в залежності від часу. Моделювання проведемо над стаціонарно закріпленою БІНС. Для початку перевіримо випадок коли координатний тригранник має невеликий нахил, помилку  $\Delta\alpha_E$ . Це призведе до того, що на горизонтальний акселерометр подіє прискорення  $-g\alpha_E$ . Виміряне прискорення спричинить, до того, що після двох інтеграторів, буде здаватись, що система має швидкість і відповідно рухається. Це спричинить момент на гіроскопах в напрямку зменшення зміщення помилки координатного тригранника, але коли акселерометр стає зрівноваженим система матиме значну швидкість, що продовжить коливання. Це нагадує маятник, коли відхиляють підвіс і дають йому коливатись

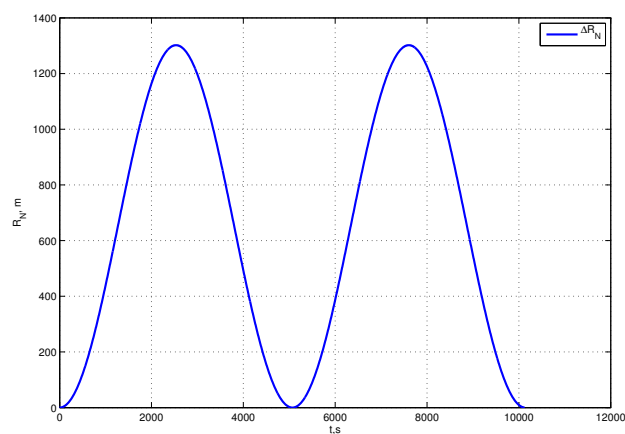


Рисунок 4.1.1 – Еволюція похибки за умови, похибки координатного тригранника  $10^{-3}rad$

На рис. 4.1.1 зображено результат моделювання руху похибки ІНС яка зумовлена початковим зміщенням координатного тригранника на  $10^{-3}rad$ . Мо-

					НАУ 11 54 75 000 ПЗ			
Зм.	Лист	№ докум.	Підп.	Дата	Інтегрована інерціально-супутникова система навігації, що базується на принципах комплексної обробки інформації з використанням калманівської фільтрації	Лит.	Аркуш	Аркушів
Розробив	НовікМ.В.						94	152
Перевірив	Мар'ясоваТ.І.							
Консульт.	ФіляшкінМ.К.							
Н. контр.	КозловаА.П.							
Затвердив	СинеглазовВ.М.					ІАСУ 608		

жна зазначити, що помилка коливається з частотою Шулера, з періодом 84.4 хвилини. Найбільша помилка приблизно 1300 метрів і досягається приблизно за 42.2 хвилини роботи системи.

Розглянемо еволюцію похибок при наявності дрейфа гіроскопа. Ефект дрейфа гіроскопа позначається на нахилі координатного тригранника, в результаті виникає помилка прискорення. Швидкість і координата коливаються з частотою Шулера. Але цього разу швидкість коливається не навколо нуля, отже помилка координати утворюється як сума лінійної наростаючої та гармонічної функції.

Яскраво виражені коливання Шулера та лінійно наростаюча функція на рис.4.1.2 як загальна помилка для стаціонарно закріпленої ІНС з дрейфом вертикального гіроскопа на  $0.01^\circ/h$ . Після 1 години роботи похибка по координаті приблизно 1300 метрів. Якщо гіроскоп менш точний то його дрейф спричиняє похибку 1600 метрів за 10 хвилин. Зрозуміло, що на не дорогих ДПІ похибка зростає до 1500 метрів за 1 хвилину.

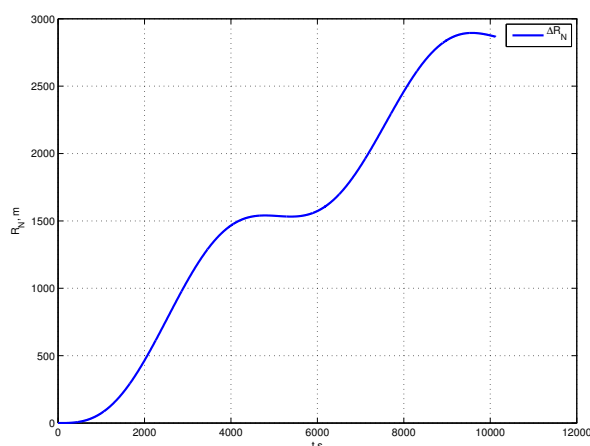


Рисунок 4.1.2 – Еволюція похибки за умови, дрейфу гіроскопа  $0.01deg/h$

При наявності помилки по швидкості  $\Delta V_E = 1$  м/с позиційна похибка буде обмеженою і коливатиметься з частотою Шулера рис.4.1.3.

Далі порівняємо ефект впливу помилок різного типу в стаціонарно закріплених ІНС. На рис. 4.1.4. На малюнку зображено вплив кожного виду помилки: помилку початкової виставки як помилку нахилу координатного тригранника та помилки ДПІ як дрейфи акселерометра та гіроскопа. Також результуючу помилку по координаті, як суму перерахованих вище.

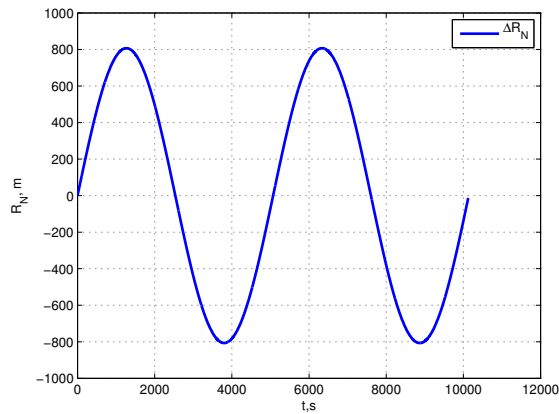


Рисунок 4.1.3 – Еволюція похибки при початковій похибці по швидкості 1 м/с

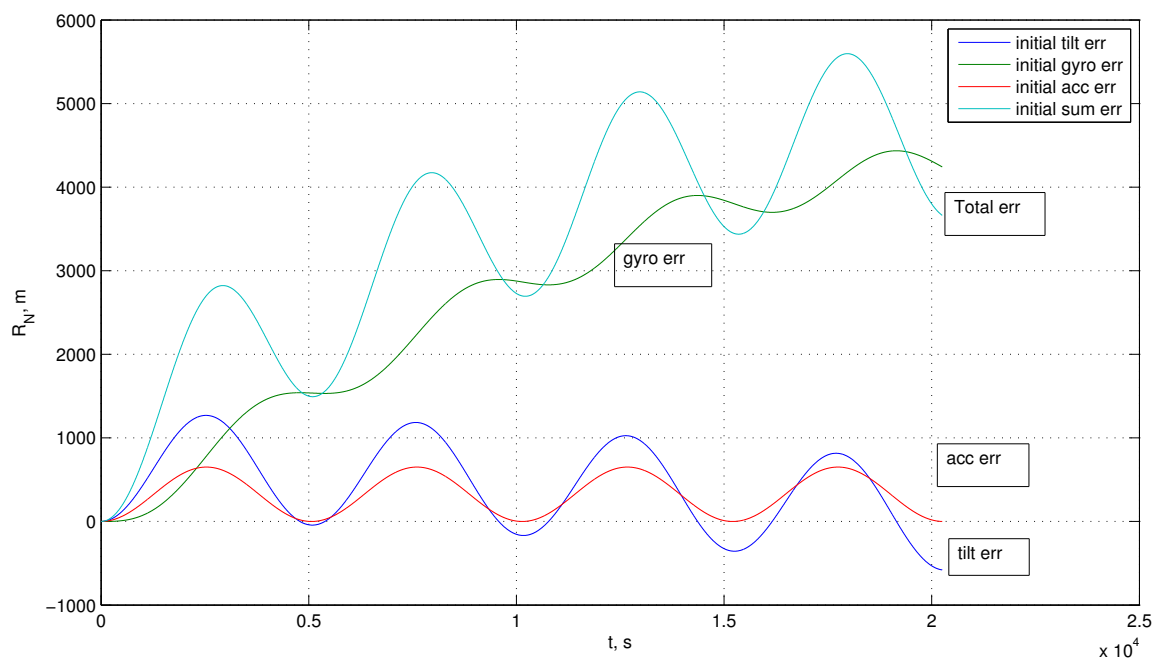


Рисунок 4.1.4 – Еволюція сумарної похибки по координаті за умови, дрейфу гіроскопа  $0.01 \text{ deg/h}$ , похибки координатного тригранника  $10^{-3} \text{ rad}$ , та зміщенням акселерометра  $10^{-4} \text{ rad}$

Можна зауважити, що за виключенням початкової виставки, сумарна похибка по координаті переважно визначається дрейфом гіроскопа. Дрейф інтегрується один раз, коли розраховується кут для перетворення прискорення, який потім інтегрується вдруге для отримання позиційної похибки.



## 4.2 Рівняння траєкторії ЛА

Еталонні параметри руху ЛА у функції часу задавалися аналітично, що дозволило уникнути методичних похибок у процесі моделювання.

Географічні координати описуються наступними рівняннями:

$$\begin{aligned}\varphi(t) &= \varphi_0 + K_\varphi t + \Delta_\varphi \sin(\omega_\varphi t + \delta_\varphi); \\ \lambda(t) &= \lambda_0 + K_\lambda t + \Delta_\lambda \sin(\omega_\lambda t + \delta_\lambda); \\ h(t) &= h_0 - \Delta h \cos(\omega_h t + \delta_h); \\ \omega_\varphi &= 2\pi/T_\varphi; \\ \omega_\lambda &= 2\pi/T_\lambda; \\ \omega_h &= 2\pi/T_h;\end{aligned}\tag{4.2.1}$$

Складові швидкості руху відносно Землі:

$$\begin{aligned}V_E(t) &= \dot{\lambda}(t) [R_1(\varphi) + h(t)] \cos \varphi(t); \\ V_N(t) &= \dot{\varphi}(t) [R_2(\varphi) + h(t)]; \\ V_h(t) &= \dot{h}(t); \\ R_1(\varphi) &= \frac{a}{\sqrt{1-e^2 \sin^2 \varphi}}; \\ R_2(\varphi) &= R_1(\varphi) \frac{1-e^2}{1-e^2 \sin^2 \varphi};\end{aligned}\tag{4.2.2}$$

Де  $a$  та  $e^2$  велика піввісь та квадрат першого ексцентриситету земного еліпсоїда:

- для: СК-42:  $a = 6378245\text{м}$ ,  $e^2 = 6,69342749 \cdot 10^{-3}$
- для: ПЗ-90:  $a = 6378136\text{м}$ ,  $e^2 = 6,69436619 \cdot 10^{-3}$
- для: WGS-84:  $a = 6378137\text{м}$ ,  $e^2 = 6,6943799901413 \cdot 10^{-3}$

Складові уявного прискорення:

$$\begin{aligned}a_E(t) &= \dot{V}_E(t) - q(t) \sin \varphi(t) V_N(t) + q(t) \cos(t) V_h(t); \\ a_N(t) &= \dot{V}_N(t) + q(t) \sin \varphi(t) V_E(t) + \dot{\varphi}(t) V_h(t); \\ a_h(t) &= \dot{V}_h(t) - q(t) \cos \varphi(t) V_E(t) - \dot{\varphi}(t) V_N(t) + g(h, \varphi); \\ q(t) &= \dot{\lambda}(t) + 2\omega_3; \\ g(h, \varphi) &= g_e [1 - 2 \frac{h(t)}{a} + \frac{3}{4} e^2 \sin^2 \varphi(t)];\end{aligned}\tag{4.2.3}$$

де  $g_e = \frac{\mu}{a^2}$ ,  $\mu = 398600.44 \cdot 10^9$

$\omega_3$  – кутова швидкість обертання Землі.

$$\begin{aligned}\dot{V}_E(t) &= \ddot{\lambda}(t)[R_1(\varphi) + h(t)] \cos(t) + \\ &+ \dot{\lambda}(t)[\dot{R}_1(\varphi) + V_h(t)] \cos(t) - \\ &- \dot{\lambda}(t)[R_1(\varphi) + h(t)] \sin \varphi \cdot \dot{\varphi}(t); \\ \dot{V}_N(t) &= \ddot{\varphi}(t)[R_2(\varphi) + h(t)] + \dot{\varphi}(t)[\dot{R}_2(\varphi) + V_h(t)]; \\ \dot{V}_h(t) &= \ddot{h}(t); \\ \dot{R}_1(t) &= \frac{ae^2 \sin \varphi(t) \cdot \cos y(t) \cdot \dot{\varphi}(t)}{(1-e^2 \sin^2 \varphi)^{3/2}}; \\ \dot{R}_2 &= \dot{R}_1(t) \frac{1-e^2}{1-e^2 \sin^2 \varphi} + R_1(\varphi) \frac{(1-e^2)2e^2 \sin \varphi \cdot \cos \varphi \cdot \dot{y}(t)}{(1-e^2 \sin^2 \varphi)^2};\end{aligned}\tag{4.2.4}$$

Кути тангажа, курса та крена (вважається, що поздовжня ось об'єкта спів-падає з вектором відносної швидкості, а кут крена пропорційний боковому прискоренню) визначаються з наступних співвідношень:

$$\begin{aligned}\vartheta(t) &= \arctg[V_h(t)/V_r(t)]; \\ \psi(t) &= \arctg[V_E(t)/V_N(t)]; \\ \gamma(t) &= K_\gamma \frac{V_N(t)\dot{V}_E(t) - V_E(t)\dot{V}_N(t)}{V_r(t) \cos v(t)},\end{aligned}\tag{4.2.5}$$

де  $V_r(t) = \sqrt{V_N^2(t) + V_E^2(t)}$

$K_\gamma \approx 0.02$  рад/(м/с<sup>2</sup>).

Супутникова навігаційна система надає вимірювання швидкостей і координат, що є не достатнім для оцінки параметрів, які безпосередньо не спостерігаються (дрейфи гіроскопів і зміщення акселерометрів), але при виконанні певних маневрів, рівень спостережності збільшується. Наприклад, в роботі [21] проводиться аналіз спостережності стану змінної в часі системи в залежності від маневрів ЛА, з чого можна зробити висновок, що обертання ЛА навколо трьох осей має найбільший рівень спостережності. Отже дані при спостереженні дрейфів ДПІ може бути використано для їх польотного калібрування.

Беручи до уваги вище зазначене, пропонується наступна тестова траєкторія рисунку 4.2.1:

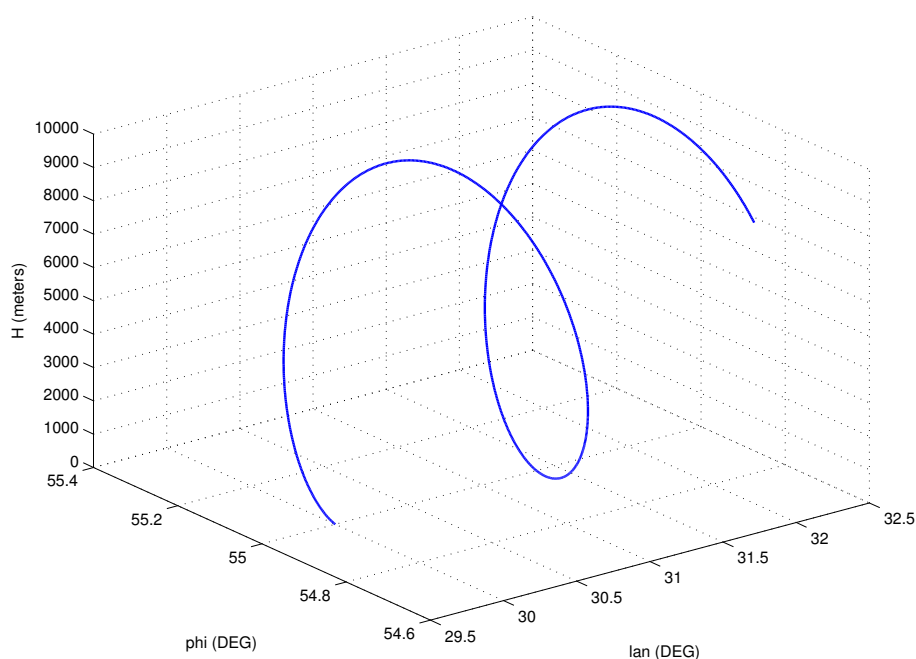


Рисунок 4.2.1 – Траєкторія руху ЛА

Кути курсу, крену та тангажа мають наступну залежність від часу рис.4.2.2:

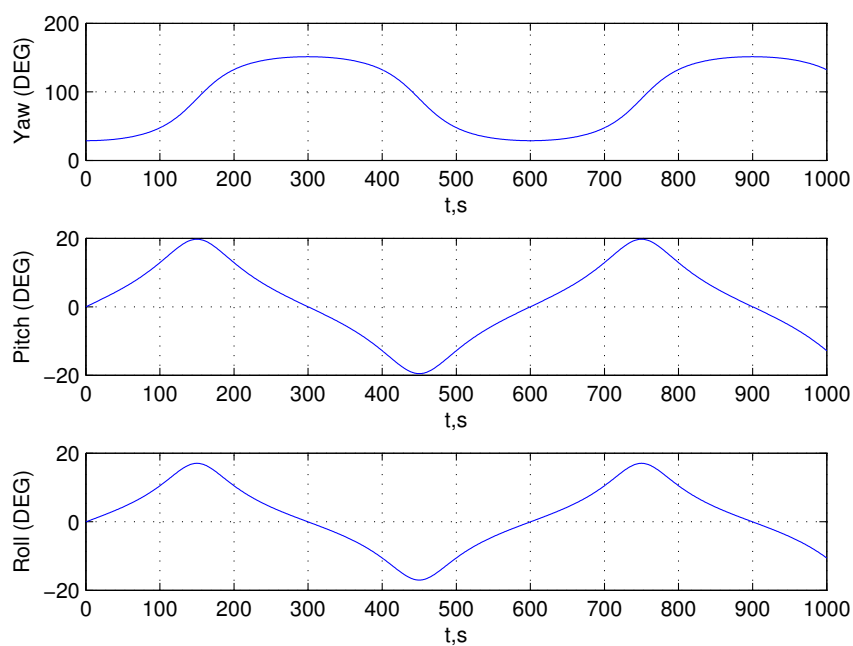


Рисунок 4.2.2 – Залежність курсу, тангажа та крена від часу

- $\varphi_0 = 55^\circ$   $\lambda_0 = 30^\circ$  – координати передмістя Києва;
- $T_\varphi = 600$ ;  $T_\lambda = 600$ ;  $T_h = 600$ ;

- $h_0 = 5000$ ;  $K_\varphi = 0$ ;  $K_\lambda = 0.00004$ ;
- $\Delta\varphi = 0.004$ ;  $\Delta\lambda = 0$ ;  $\Delta h = h_0$ ;
- $\delta\varphi = 0$ ;  $\delta\lambda = 0$ ;  $\delta h = 0$  ;

### 4.3 Моделювання алгоритмів калманівської фільтрації

Нижче представлені результати імітаційного моделювання алгоритмів калманівської фільтрації при побудові інваріантного алгоритму комплексної обробки інформації на прикладі польоту ЛА за заданою траєкторією. На рис 4.3.1...4.3.5 представлено еволюції похибок оцінювання: координат та висоти польоту (рис.4.3.1), складових швидкості (рис.4.3.2), параметрів кутової орієнтації (рис. 4.3.3), вихідних сигналів гіроскопів (рис. 4.3.4), акселерометрів (рис. 4.3.5).

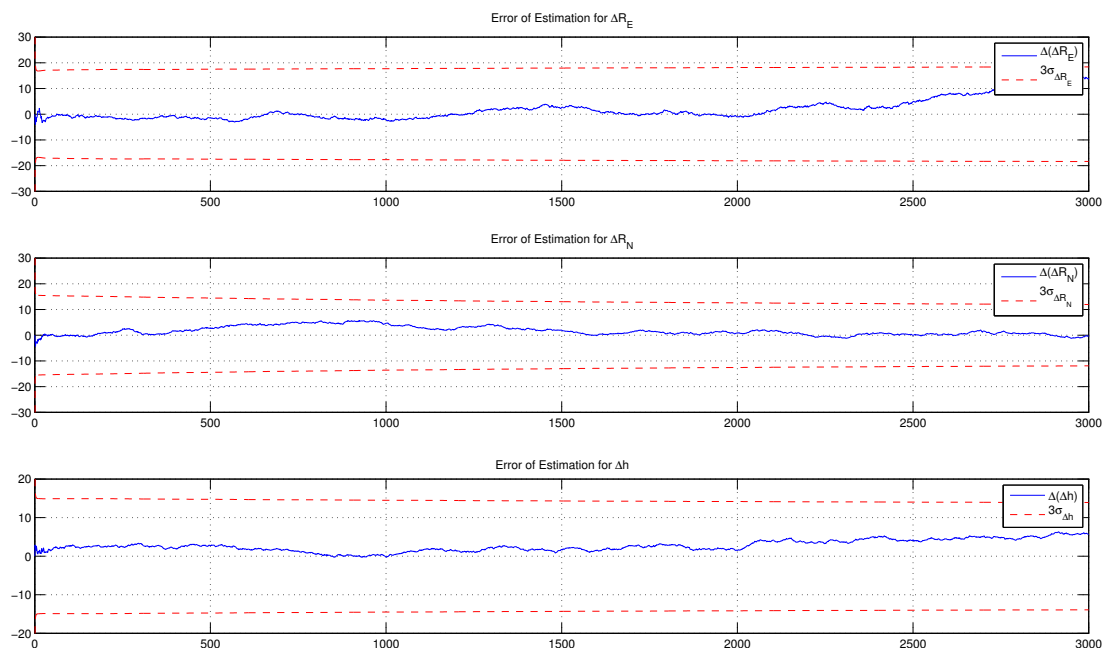


Рисунок 4.3.1 – Еволюції похибок координат

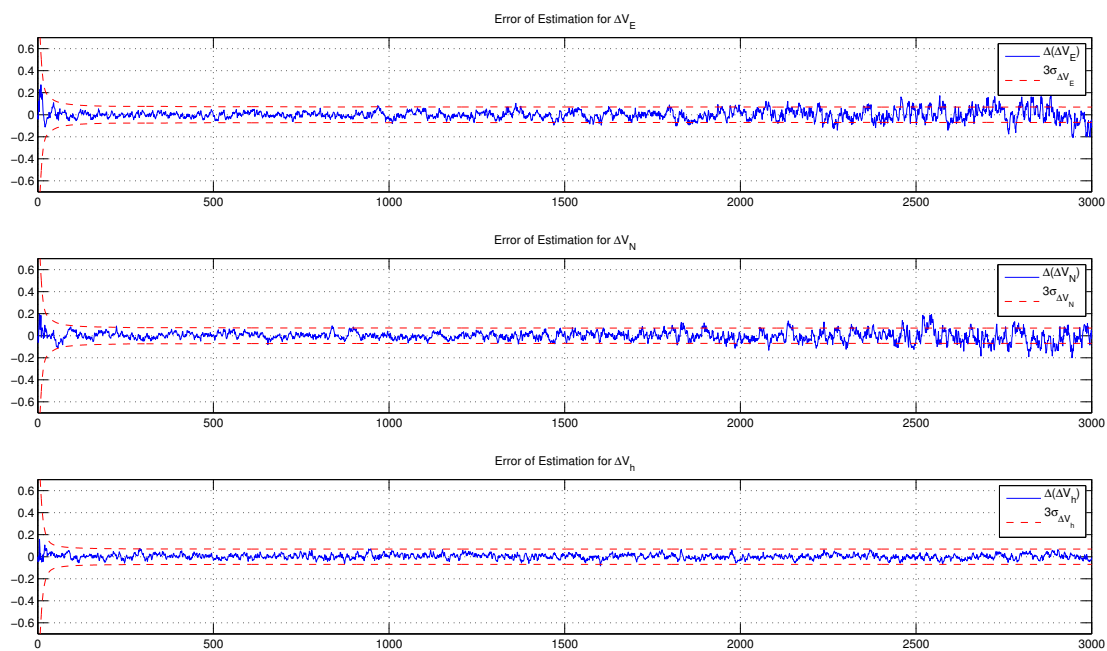


Рисунок 4.3.2 – Еволюції похибок швидкостей

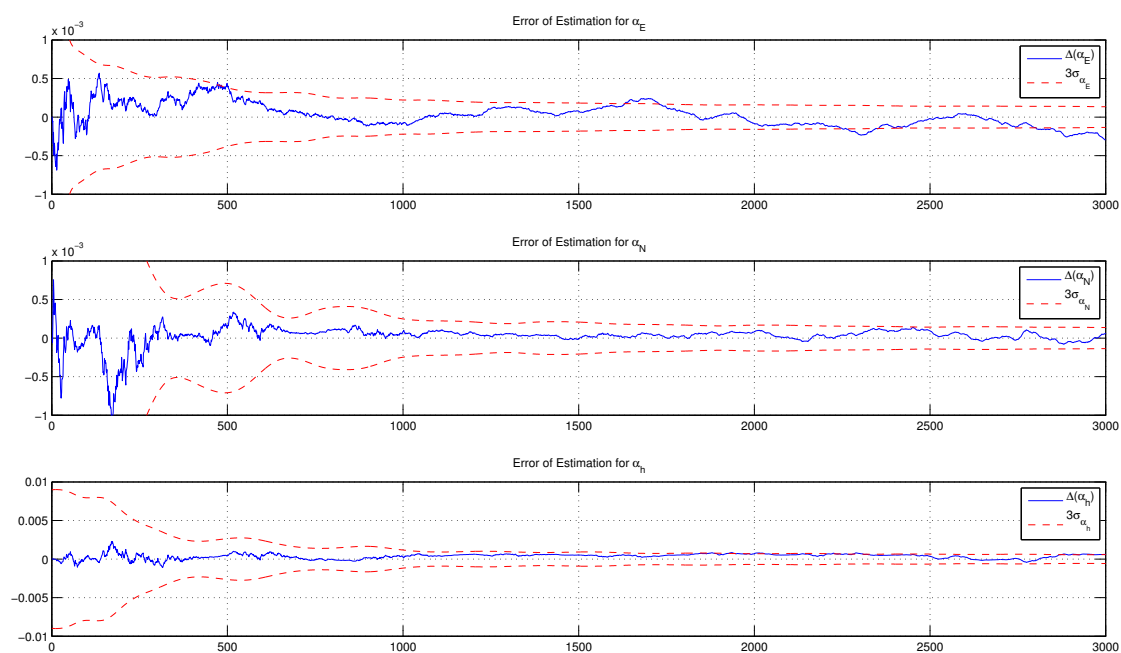


Рисунок 4.3.3 – Еволюції похибок параметрів кутової орієнтації

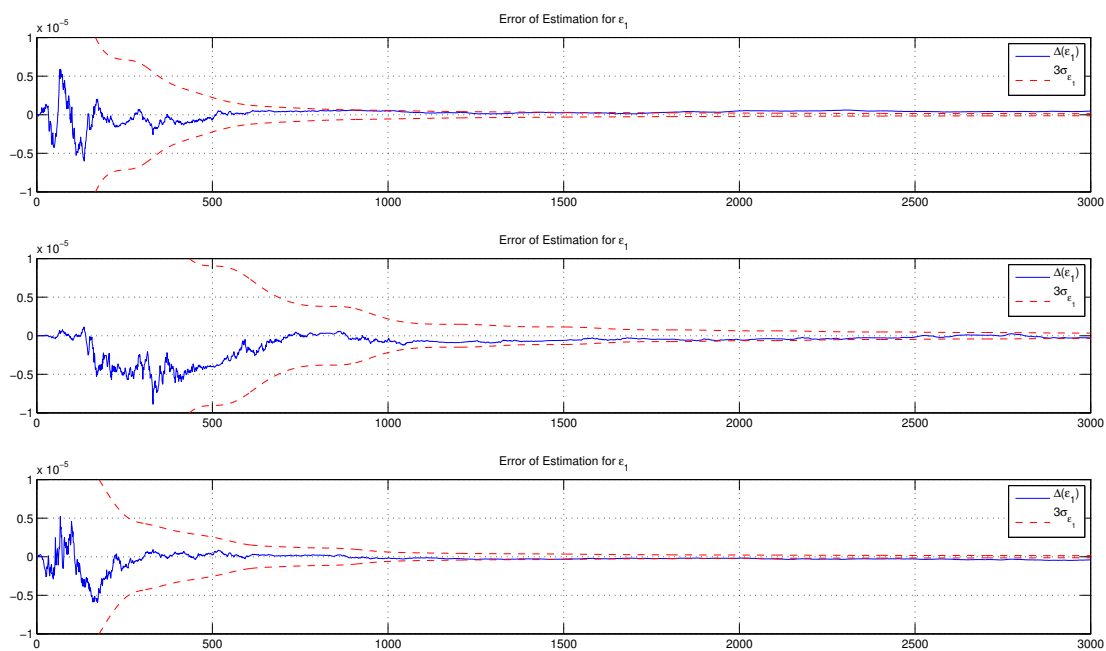


Рисунок 4.3.4 – Еволюції похибки оцінки дрейфів гіроскопів

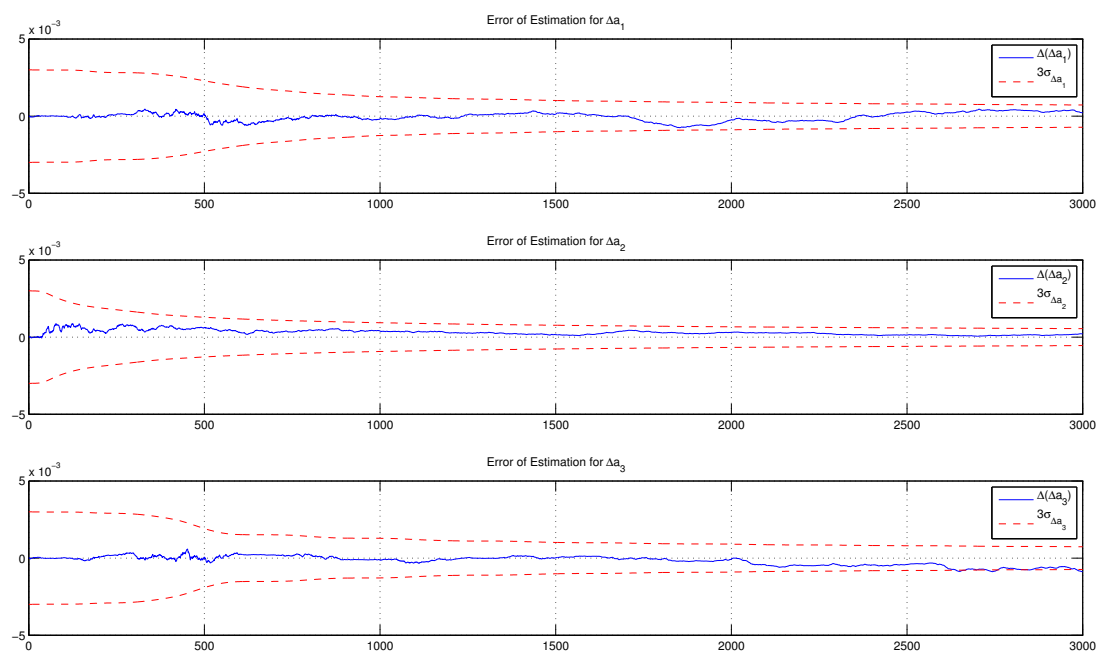


Рисунок 4.3.5 – Еволюції похибки оцінки дрейфів акселерометрів

Похибки оцінювання навігаційних параметрів, що спостерігаються, – ко-

ординат та складових швидкості сходяться до рівня корельованих складових похибок СНС, при цьому наявність корекції від барометричного висотоміра забезпечує стійкість вертикального каналу інтегрованої системи навігації. Одночасно оцінюються параметри, що не спостерігаються – параметри кутової орієнтації, а також квазістаціонарні складові похибок ДПІ, які можна використовувати для польотного калібрування ДКШ та акселерометрів.

Помилки оцінки крену, курсу і тангажу показані на рисунку 4.3.6

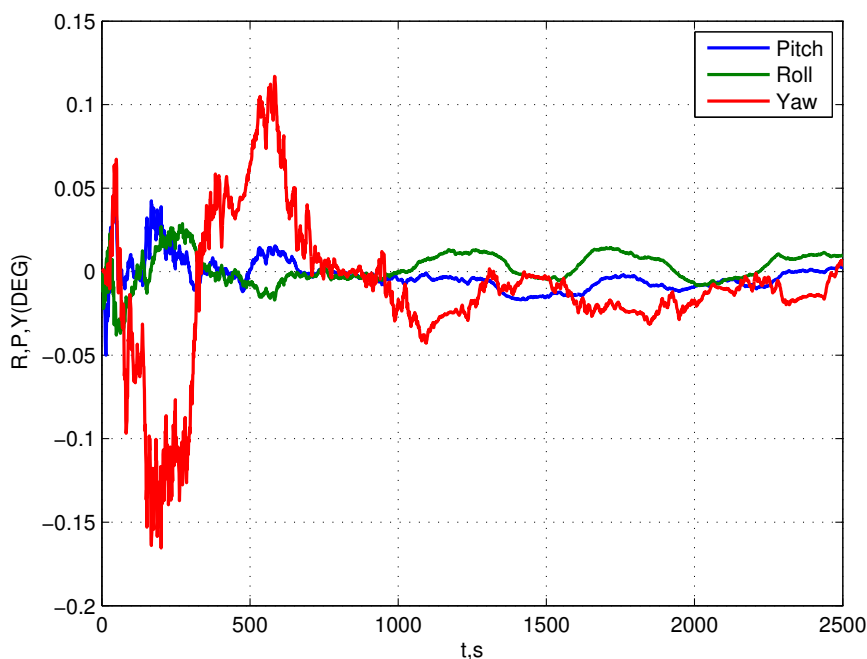


Рисунок 4.3.6 – Похибки оцінки кутів курсу, крену тангажу

При вимкненій супутниковій системі, але з фільтром Калмана та барометричним висотоміром, показання БІНС будуть наступними рис.4.3.7. Екстраполяція помилок за допомогою фільтра Калмана, значно покращує, результати навігаційної системи. В той час як автономно ІНС уходить, досить швидко, фільтр підтримує на прийнятному рівні  $\pm 10$  м на протязі 100 с. :

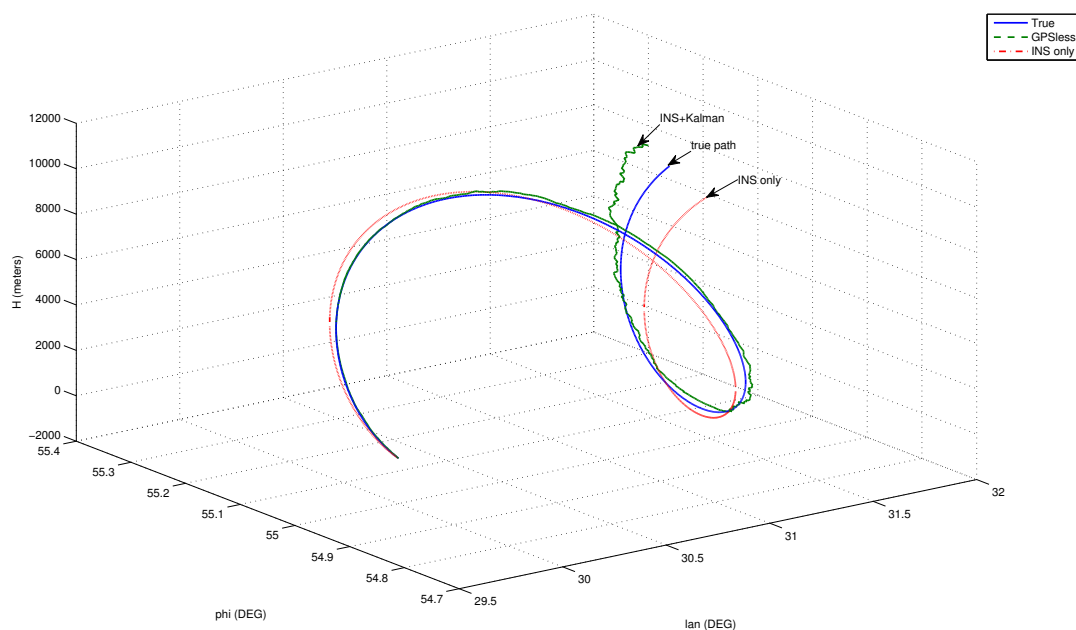


Рисунок 4.3.7 – Траєкторія руху ЛА з вимкненою СНС

Результати моделювання занесені до таблиці 4.3.1, де вказані середньоквадратичні помилки по: координатам, швидкостям, орієнтації, дрейфів гіроскопів та акселерометрів після роботи системи на протязі 800с.

Таблиця 4.3.1 – Середньоквадратичні помилки:

N	East	North	Height
Координати, м	5.8792050244	4.6476224404	4.8677711489
Швидкості, м/с	0.0236254078	0.0235478062	0.0231813797
Орієнтація, рад	8.42E-005	0.000133569	0.0004735418
Дрейф ДКШ, рад/с	2.50E-007	1.28E-006	3.80E-007
Акселером, g	0.00005007264	0.0000344999	0.00004686141

Аналіз результатів моделювання доказує працездатність розроблених алгоритмів комплексування. Дані таблиці 4.3.1 відповідають критеріям заданим в технічному завданні. Помилка по координаті не перевищує 5 метрів (1 сигма), по швидкості не більше 0.03 м/с, що відповідає рівню точності СНС.



## 5 Дослідження алгоритму оптимальної фільтрації за допомогою розробленого програмного забезпечення

В роботі проведено дослідження розроблених алгоритмів оптимальної калманівської фільтрації, в результаті розроблено програмне забезпечення. Дослідження проведені шляхом моделювання еволюцій похибок БІНС та СНС, та їх оцінки за допомогою оптимального рекурентного фільтра Калмана. Програмне розроблене на об'єктно-орієнтовній мові програмування Java, основною перевагою якої є незалежність від архітектури.

Розробка проведена в середовищі Java SE 6 (1.6.0). У офіційній реалізації, Java програми компілюються у байткод, який при виконанні інтерпретується віртуальною машиною для конкретної платформи. Sun Microsystems надає компілятор Java та віртуальну машину Java, які задовольняють специфікації Java Community Process, під ліцензією GNU General Public License.

Під «незалежністю від архітектури» мається на увазі те, що програма, написана на мові Java, працюватиме на будь-якій підтримуваний апаратній чи системній платформі без змін у початковому коді та перекомпіляції.

Цього можна досягти, компілюючи початковий Java код у байт-код, який являє собою спрощені машинні команди. Потім програму можна виконати на будь-якій платформі, що має встановлену віртуальну машину Java, яка інтерпретує байткод у код, пристосований до специфіки конкретної операційної системи і процесора. Зараз віртуальні машини Java існують для більшості процесорів і операційних систем, в тому числі різноманітних версій GNU/Linux, Microsoft Windows, Apple Mac OS, мобільних платформ, наприклад Google Android та багато інших.

В програмі використані наступні бібліотеки:

– JMathPlot - побудова графіків;

					<i>НАУ 11 54 75 000 ПЗ</i>			
Зм.	Лист	№ докум.	Підп.	Дата				
Розробив	НовікМ.В.				Інтегрована інерціально-супутникова система навігації, що базується на принципах комплексної обробки інформації з використанням калманівської фільтрації	Лит.	Аркуш	Аркушів
Перевірив	Мар'ясоваТ.І.						105	152
Консульт.	ФіляшкінМ.К.					<i>IACU 608</i>		
Н. контр.	КозловаА.П.							
Затвердив	СинеглазовВ.М.							

- JMathArray(JAMA) - математичні розрахунки;
- JMathIO - зберігання/читання даних в ASCII форматі;
- Swing - розробка графічного інтерфейсу.

Математичні обчислення проводяться за допомогою спеціалізованої бібліотеки JAMA. Це програмна бібліотека для розв'язання задач лінійної алгебри. Ця бібліотека створена Національним інститутом стандартів і технологій США і схожа по функціональності з LAPACK. Існують версії JAMA для мов програмування C++ та Java. Іншою перевагою є те, що ця бібліотека розповсюджується вільно, з багатою документацією та джерельними кодами.

Swing – інструментарій для створення графічного інтерфейсу користувача (GUI) мовою програмування Java. Це частина бібліотеки базових класів Java (JFC, Java Foundation Classes). Дозволяє зручна та швидко створити інтерфейс користувача будь якої складності.

JMathPlot, JMathIO – зручні бібліотеки, що розповсюджуються під ліцензією BSD, для виводу графічної інформації: гістограм, 2D та 3D графіків.

## 5.1 Опис вхідних та вихідних даних

Вхідними даними для програми є параметри інтегрування, параметри руху ЛА, параметри БІНС: початкові похибки гіроскопів, акселерометрів, барометричного висотоміра та параметри СНС: випадкові та корельовані складові похибок по кожному каналу.

Через дружній графічний інтерфейс, користувач має можливість ввести або змінити наступні данні:

- час моделювання;
- крок розрахунку;
- параметри траєкторії руху ЛА:
- початкові координати;
- лінійні і гармонічні складові траєкторії по кожному каналу;

- дрейф та випадкове блукання гіроскопа;
- похибка барометричного висотоміра;
- кореляційна матриця системи
- кореляційні похибки СНС;
- випадкові похибки СНС;

Вихідні данні представлені у вигляді графіків, або існує можливість зберегти їх в текстовому файлі, яким можна потім наприклад завантажити в середовище Octave чи Matlab. Отже на виході отримується наступна інформація:

- оцінки координат;
- оцінки швидкостей;
- оцінки зміщення навігаційного тригранника;
- оцінки дрейфів гіроскопів;
- оцінка зміщення акселерометрів.

За допомогою графічного інтерфейсу, користувач має можливість наочно отримати результати розрахунків, або імпортувати в іншу підпрограму чи середовище розробки.

## 5.2 Опис графічного інтерфейсу користувача

Інтерфейс програми складається з вікна програми, яке включає поля для введення параметрів траєкторії руху ЛА, меню та закладки з графічною інформацією. На Рис.5.2.1 зображено головне вікно, з графіком траєкторії руху ЛА.

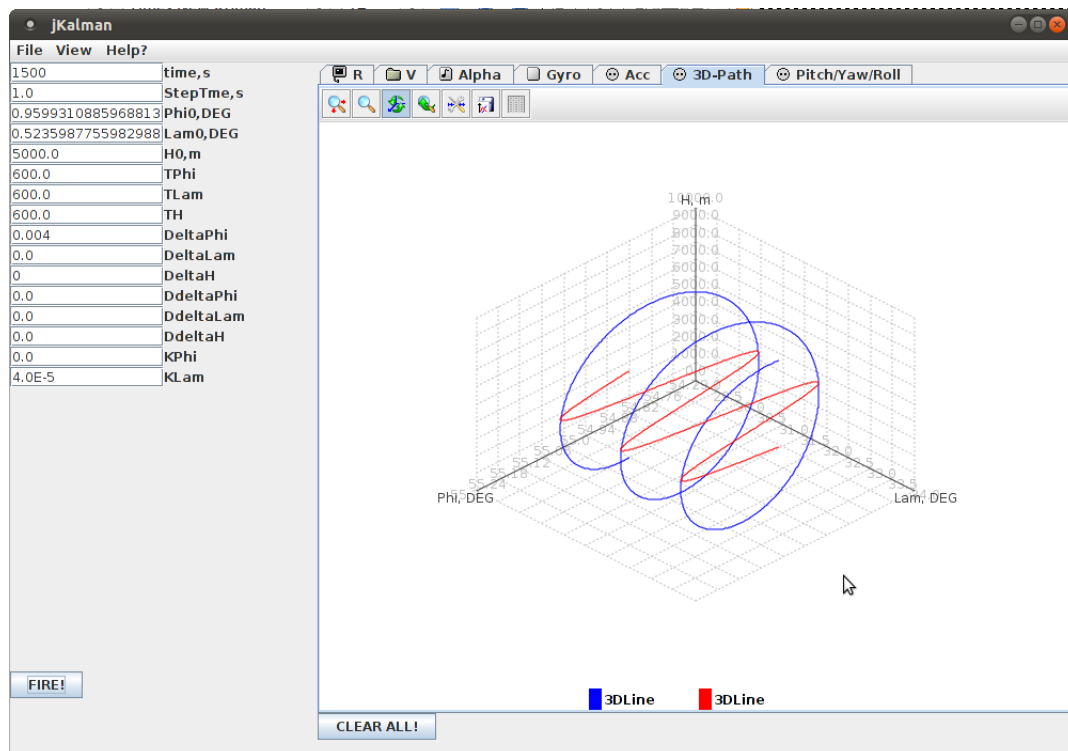


Рисунок 5.2.1 – Головне вікно програми

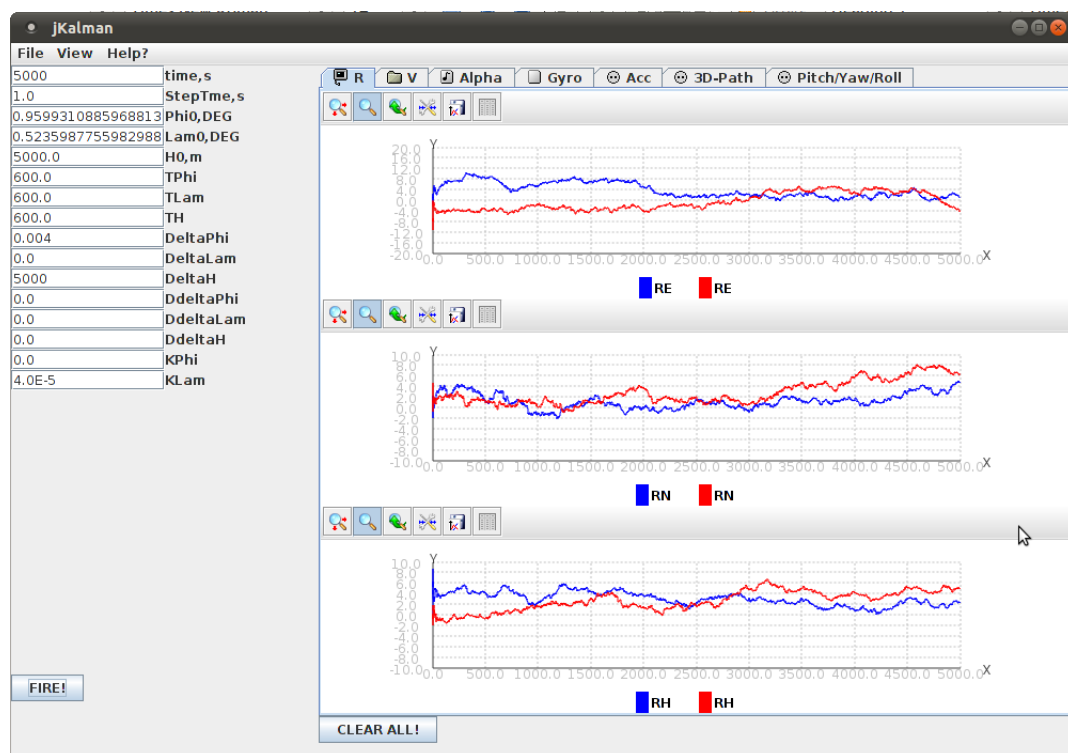


Рисунок 5.2.2 – Помилка оцінки координат

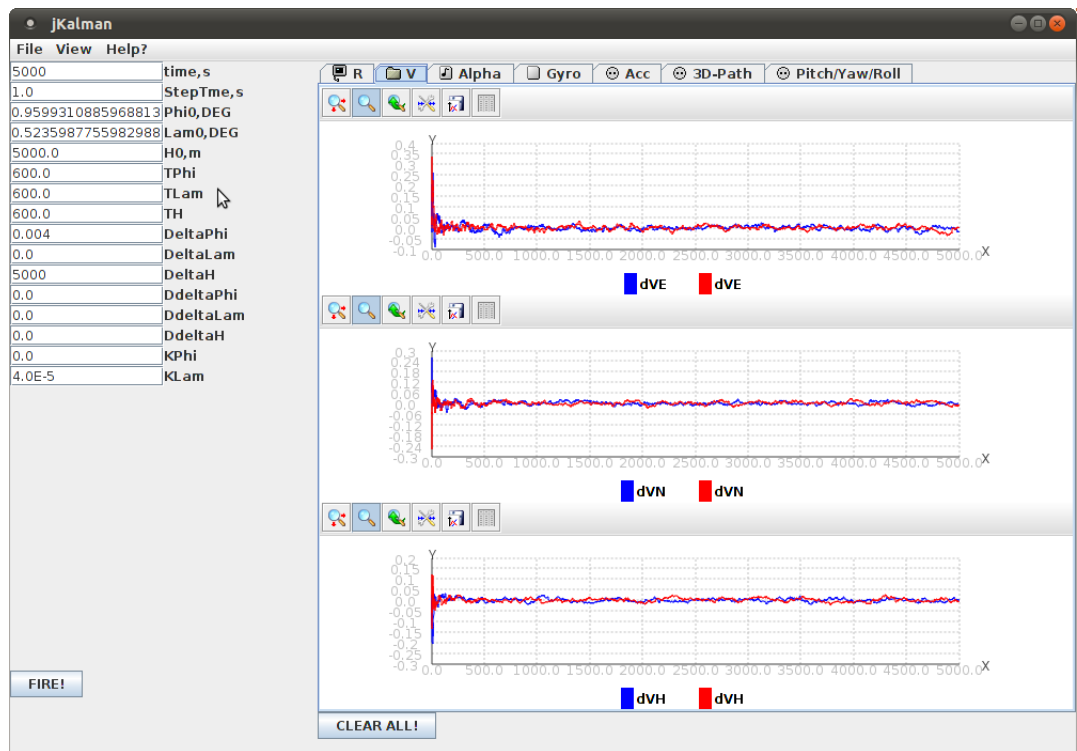


Рисунок 5.2.3 – Помилка оцінки швидкостей

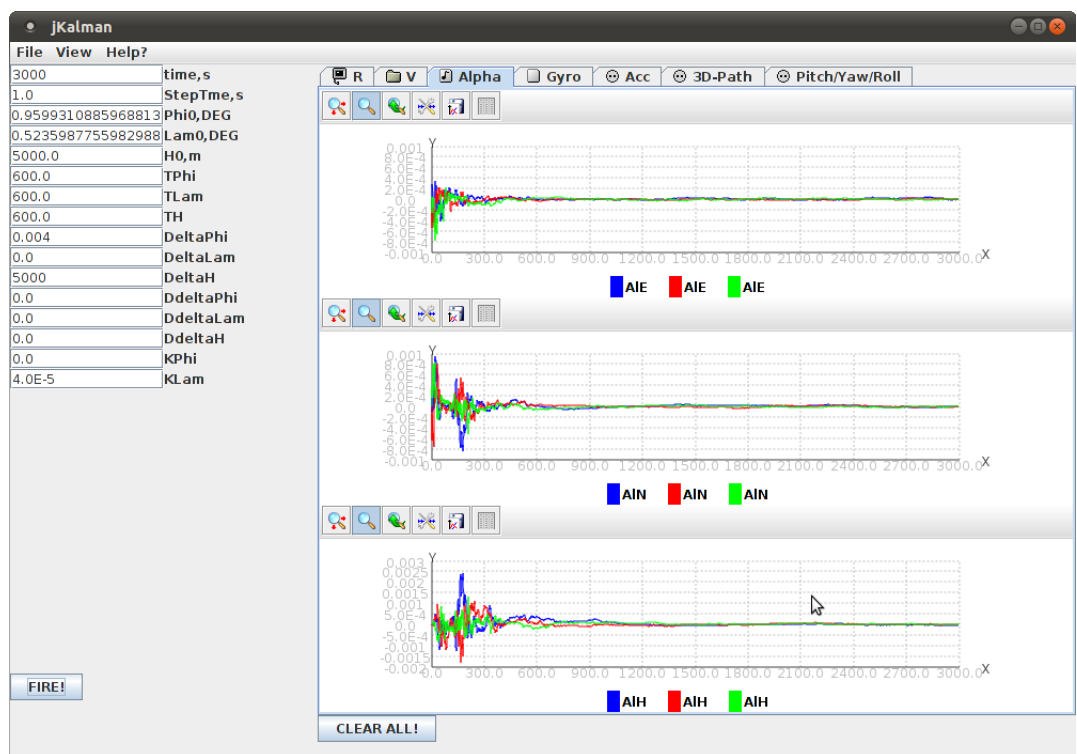


Рисунок 5.2.4 – Помилка оцінки координатного тригранника

Графічна інформація зображається на спеціальних закладках з панеллю налаштування графіка. За допомогою панелі можна провести: зміщення,

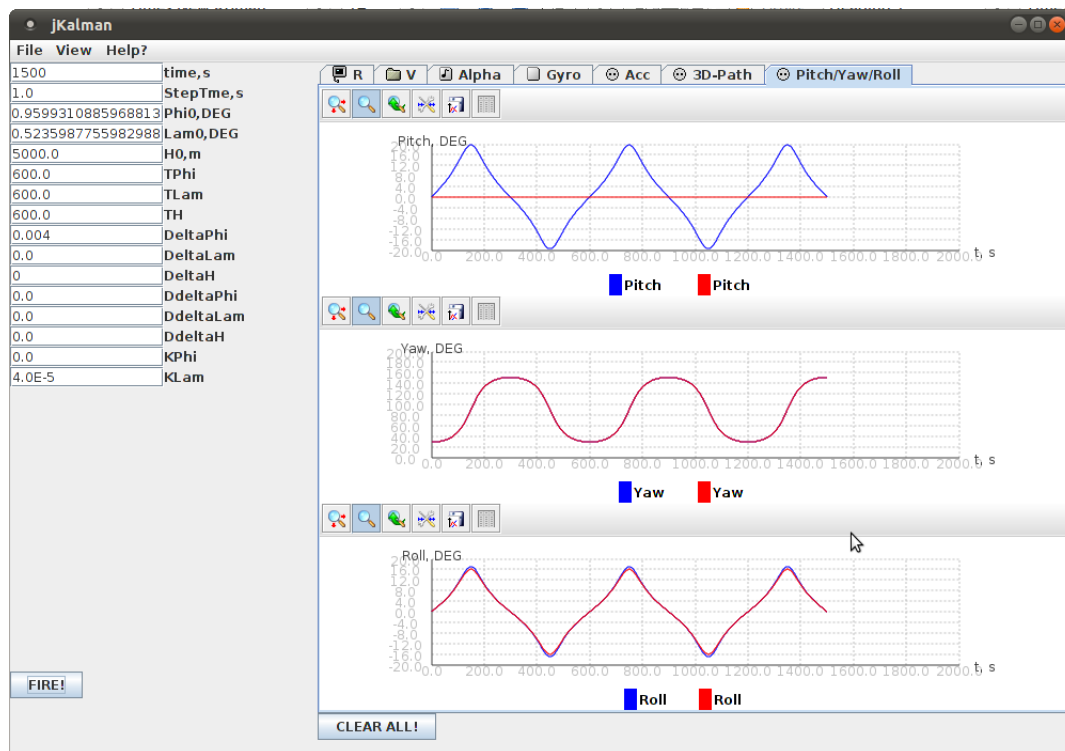


Рисунок 5.2.5 – Еволюція в часі кутового положення ЛА

масштабування, поворот (для тривимірних зображень), настроїти колір та видимість лінії (Рис.5.2.6).

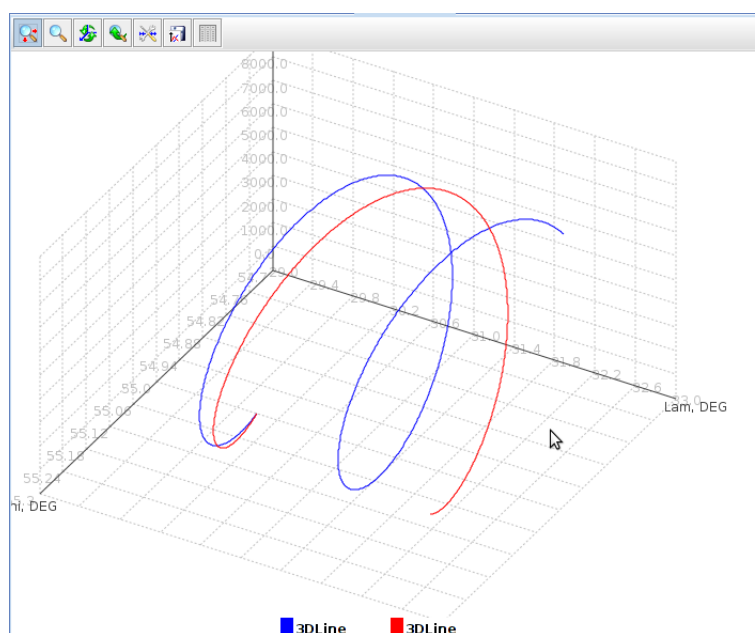


Рисунок 5.2.6 – Панель настроювання графіка

На рис.5.2.7 показане вікно редагування даних, за на основі яких будується графічне відображення. Звідси надається можливість скопіювати данні в буфер обміну, колір та видимість даних.

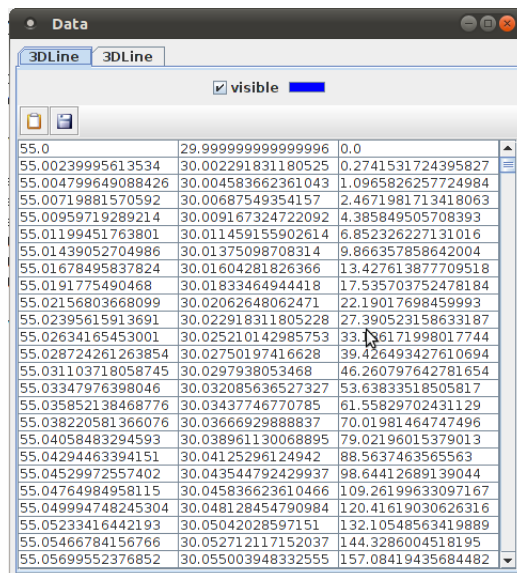


Рисунок 5.2.7 – Панель настроювання графіка

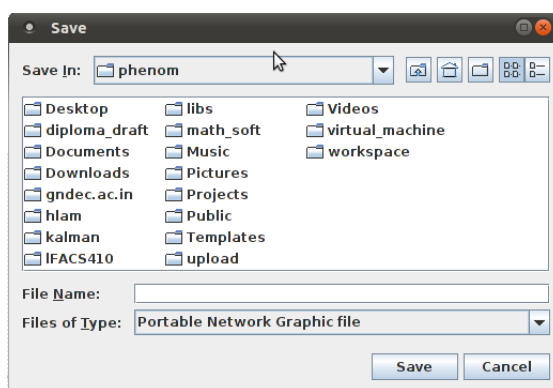


Рисунок 5.2.8 – Панель настроювання графіка

Колір можливо змінити через спеціалізовані елементи керуванні за допомогою наступних діалогів:

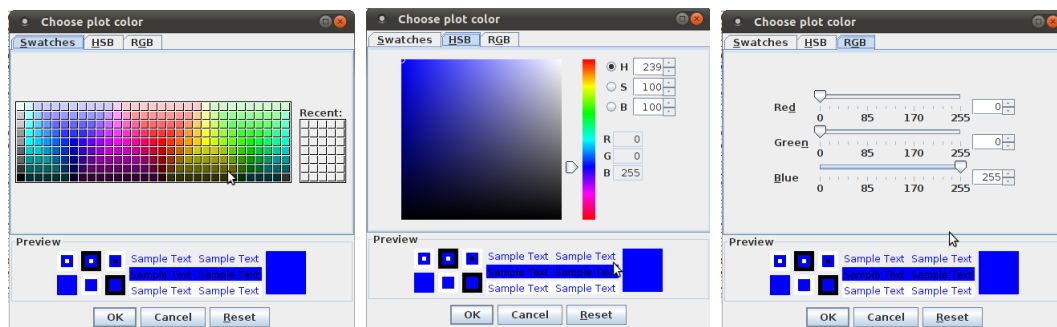


Рисунок 5.2.9 – Панель настроювання графіка

## 5.3 Опис структури програми

Програмне забезпечення створене на мові програмування Java, яка є об'єктно орієнтовною мовою, основою якої є класи. В роботі роблені наступні класи:

- FlightPathGen.class;
- RunISNS.class;
- jRKalman.class;
- InputsPanel.class;
- FlightPropPanel.class;

FlightPathGen.class – генератор траєкторії руху ЛА та допоміжних функцій, крім конструктора включає наступні методи:

- ComputePhiLamH() – розраховує еволюцію координат;
- ComputeR1R2() – розраховує параметри моделі Землі;
- ComputeV() – розраховує швидкості руху ЛА;
- ComputeA() – розраховує прискорення ЛА;
- ComputePRY() – розраховує еволюцію кутового прискорення ЛА;

RunISNS.class – симулятор роботи БІНС, точніше формує необхідні матриці для вирішення рівнянь еволюції системи.

- ComputeF() – формує матрицю динаміки системи;
- ComputeH() – формує матрицю спостережень;
- ComputeB() – формує матрицю напрямних косинусів;
- ComputeQ() – формує матрицю шумів динаміки системи;
- ComputeR() – формує матрицю шумів датчиків первинної інформації;

jRKalman – містить статичні методи, необхідні для роботи програми: містить функцію `main(String[] args)`, відповідні методи для інтегрування рівнянь еволюції похибок та функції вирішення основних рівнянь фільтра Калмана.



InputsPanel.class, FlightPropPanel.class – формують необхідні елементи графічного інтерфейсу користувача: панелі зміни параметрів, та елементи для побудови необхідних графіків.

					НАУ 11 54 75 000 ПЗ	Аркул
Зм.	Лист	№ докум.	Підп.	Дата		113

## 6 Охорона праці

### 6.1 Вступ

В дипломній роботі розробляється інерціально-супутникова навігаційна система, що базується на основі комплексної обробки інформації з використанням фільтра Калмана. Розробкою та відпрацюванням алгоритмів роботи навігаційної системи, налаштуванням обладнання та калібрування датчиків займаються інженери програмісти та радіотехніки лабораторії. Отже суб'єктом є інженер програміст лабораторії, функціональним зобов'язаннями якого є програмування навігаційних алгоритмів для бортової обчислюваної машини, засобом праці є персональний комп'ютер та модулі навігаційного обладнання: датчики навігаційної системи (мікромеханічні чи лазерні акселерометри та гіроскопи), бортовий обчислювач навігаційної інформації.

До роботи з навігаційним обладнанням та ЕОМ допускаються працівники, що не мають медичних протипоказань, пройшли вчасно періодичний медичний огляд, інструктаж і навчання правилам техніки безпеки і виробничої санітарії.

Основним місцем роботи інженера програміста є лабораторія навігаційного обладнання авіаційного підприємства чи науково-дослідного інституту. Періодично місцем роботи може бути літак, де встановлено навігаційне обладнання (налаштування, тестування, випробування) або ЗПС, у випадку, якщо розробляємі пристрої встановлюються на БПЛА.

### 6.2 Опис робочого місця

Для приміщення лабораторії вибрана площа 30 м<sup>2</sup>, з висотою стелі – 3м. Виробничі будівлі та приміщення споруджуються згідно з вимогами бу-

					НАУ 11 54 75 000 ПЗ			
Зм.	Лист	№ докум.	Підп.	Дата	Інтегрована інерціально-супутникова система навігації, що базується на принципах комплексної обробки інформації з використанням калманівської фільтрації	Лит.	Аркуш	Аркушів
Розробив	НовікМ.В.						114	152
Перевірив	Мар'ясоваТ.І.					ІАСУ 608		
Консульт.	ФіляшкінМ.К.							
Н. контр.	КозловаА.П.							
Затвердив	СинеглазовВ.М.							

дівельних і санітарних норм. Об'єм виробничих приміщень для програмістів, операторів відеотермінальних пристроїв на одного працівника складає 19,5 м<sup>2</sup>, площа приміщень — 6 м<sup>2</sup> з урахуванням максимального числа працівників в одну зміну.

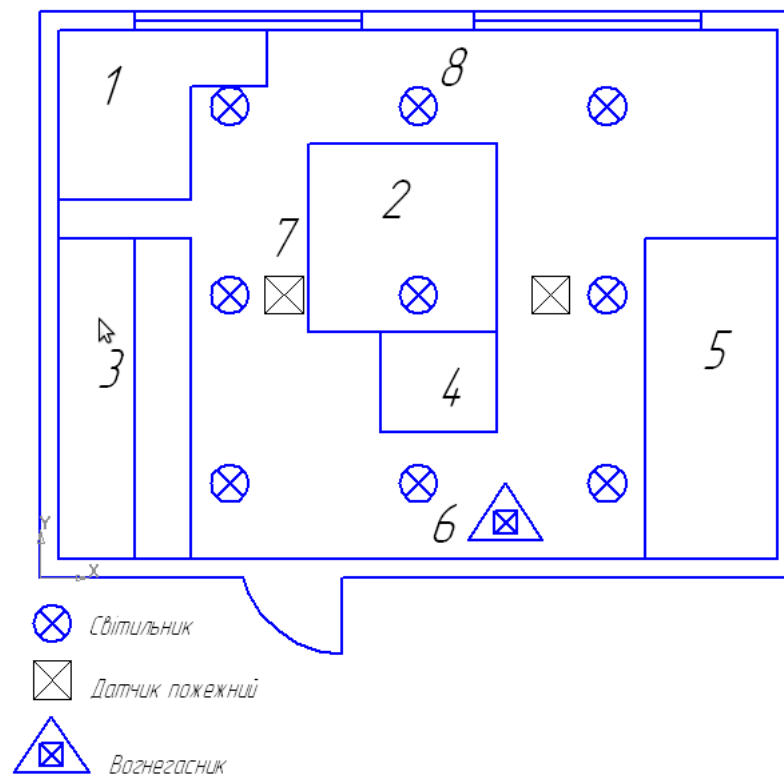


Рисунок 6.2.1 – План робочого приміщення

1. Робоче місце №1;
2. Координатна платформа;
3. Робоче місце №2;
4. Стіл з ПК приєднаний до мережі та до вимірювальної апаратури;
5. Шафа з допоміжним обладнанням та витратними матеріалами;
6. Вогнегасник;
7. Датчик пожежний;
8. Світильник.

В приміщенні розташовані два робочих місця. Перше обладнане ПК з рідкокристалічним дисплеєм, приєднане до локальної мережі. На столі додатково встановлені програматори, осцилограф, лабораторний блок живлення та

мультиметр, телефон, принтер. Друге робоче місце обладнане паяльною станцією, контрольно-вимірною апаратурою, лабораторним блоком живлення, та додатковими розетками з мережею живлення 115В 400Гц. ПК, приєднаний до локальної мережі Ethernet. Над столом знаходиться полиця з радіoeлементами. Всі прилади мають бути заземлені.

Трудова діяльність інженера програміста відбувається у певному виробничому середовищі, де діють такі шкідливі фактори:

1. Понижений рівень штучного освітлення
2. Шум
3. Електричний струм
4. Пожежонебезпека
5. Мікроклімат

## 6.3 Шум

Допустимі рівні звукового тиску у октавних смугах частот, еквівалентні рівні звуку на робочому місці регламентовані ДСН 3.3.6.037-99, для інженера показані в таблиці 6.3.1.

Таблиця 6.3.1 – Нормовані рівні звукового тиску та рівень шуму на робочому місці

Рівні звукового тиску (дБ) в октавних смугах з серединами геометричними частотами, Гц									Рівень звуку, дБА
31.5	63	125	250	500	1000	2000	4000	8000	50
86	71	61	54	49	45	42	40	38	

Методи вимірювання шуму, інфразвуку та ультразвуку регламентовано ДСН 3.3.6.037-99. Вплив шуму на людину, його вимірювання та оцінювання, для

присутньої обчислюваної та офісної техніки здійснюється відповідно до ДСТУ ISO 7779:2005, а присутні кондиціонер за ДСТУ 3010-95.

Якщо лабораторія знаходиться не далеко від аеродрому, необхідна додаткова звукоізоляція. У якості звукоізолюючих матеріалів, які застосовують у конструкціях перекриттів для зниження передачі структурного (ударного) звуку переважно використовують мати та плити із скляного та мінерального волокна, м'які плити з деревних стружок, картон, гуму, утеплений лінолеум.

## 6.4 Освітлення

Особу увагу необхідно приділити важливому з точки зору виробничої санітарії питанню освітлення на робочому місці. Основним документом, який регламентує норми освітленості є ДБН.В.2.5-28-2006 «Природне і штучне освітлення».

Освітлення на робочому місці повинно бути поєднаним (штучне та природне світло). Природне освітлення повинно бути боковим. При виконанні робіт з категорії високої зорової точності коефіцієнт природної освітленості повинен відповідати нормативним рівням по ДБН.В.2.5-28-2006 (не нижче 1,5), при зоровій роботі середньої точності – не нижче 1.

Освітлення повинно бути достатнім, щоб очі без зайвого напруження могли розрізняти деталі, що розглядаються; стабільним – для цього напруга в електричній мережі не повинна коливатися більше ніж на 4 %; рівномірно розподіленим по робочих поверхнях, щоб очам не доводилося потрапляти з дуже темного місця у світле і навпаки; таким, що не викликає сліпучої дії на око людини як самого джерела світла, так і від відбиваючих поверхонь, що знаходяться в полі зору робітника. Зменшення віддзеркалювання джерел світла досягається шляхом застосування світильників; таким, щоб не викликати різкі тіні на робочих місцях (цього можна досягти при правильному розташуванні світильників); безпечним – не призводити до вибуху, пожежі у виробничих приміщеннях.

Для створення сприятливих умов зорової роботи освітлення робочих приміщень задовольняються наступні умови:

					НАУ 11 54 75 000 ПЗ	Аркул
Зм.	Лист	№ докум.	Підп.	Дата		117

1. рівень освітленості робочих місць відповідає гігієнічним нормам для даного виду роботи;
2. забезпечена рівномірність та часова стабільність рівня освітленості у приміщенні, відсутні різкі контрасти між освітленою робочою зоною та навколишнім простором;
3. відсутні різкі та рухомі тіні;
4. у полі зору предмета немає сліпучого блиску
5. штучне світло за спектральним складом наближається до природного

Розрахунок виробничого освітлення зроблено за методом використання світлового потоку. За цим методом світловий потік однієї лампи (у люменах) визначається за формулою:

$$F_n = \frac{E_n S k Z}{n \eta} \quad (6.4.1)$$

де  $E_n$  – нормована освітленість для проєктованих ділянок, цехів, лабораторій;

$S$  – площа приміщення, у якому проєктується виробниче освітлення, м<sup>2</sup>;

$k$  – коефіцієнт запасу світлового потоку. Він приймається: для люмінесцентних ламп при малому виділенні пилу, диму, кіптяви - 1,5, при середньому і великому виділенні відповідно - 1,8 й 2,0; для ламп накаливання при малому виділенні пилу, диму, кіптяви - 1,3, при середньому й великому, відповідно - 1,5 й 1,7;

$Z$  – поправочний коефіцієнт, що відбиває відношення, приймається при найвигіднішому розташуванні світильників, коли світловий потік використовується для освітлення робочої зони найбільш раціонально, рівним 1,1-1,2;

$n$  – число ламп в приміщенні;

$\eta$  – коефіцієнт використання світового потоку від світильника, що показує, яка частина світлового потоку лампи досягає освітлюваної поверхні, у тому числі завдяки відбиттю світлового потоку від стін, стелі й робочої поверхні.

Коефіцієнт  $\eta$ , що залежить від показника геометричних розмірів приміщення і коефіцієнтів відбиття стін, стелі і приміщення, обчислений для різних типів світильників.

Показник приміщення:

$$i = \frac{a \times b}{H(a + b)} = \frac{5 \times 6}{1,925(6 + 5)} \quad (6.4.2)$$

де  $a$  та  $b$  – довжина й ширина освітлюваного приміщення, м;

$H$  – висота підвісу світильників над робочою поверхнею, м.

Висота підвісу знаходить з наступної формули:

$$H = h_b - N - h_c \quad (6.4.3)$$

де  $h_b$  – висота приміщення, м;

$h$  – висота світильника, м;

$N$  – висота робочої поверхні, м.

Приміщення лабораторії має наступні геометричні формули: довжина робочого кабінету складає 6 м; ширина - 5 м; висота - 3 м. Визначимо висоту підвісу світильників, підставив вихідні значення в формулу 6.4.3:

$$H = 3,0 - 0,275 - 0,8 = 1,925(m). \quad (6.4.4)$$

тоді індекс приміщення:

$$i = \frac{5 \times 6}{1,925(6 + 5)} = 1.41676$$

Далі визначимо значення показника приміщення, підставляючи в формулу значення : По показнику приміщення та коефіцієнтам світлового потоку від підлоги – 10% (0,1), від стін – 30% (0,3) та від стелі – 80% (0,8) визначаємо для світлодіодної лампи ML-T8-13W/0.6-SMD значення коефіцієнта використання світлового потоку ( $\eta$ ).  $\eta = 0,69$ ; коефіцієнт запасу світлового потоку  $k = 1,25$ ; поправочний коефіцієнт  $Z = 1,2$ .

Норма (мінімум) освітленості при проведенні середньо точних робіт складає 400 лк. Світловий потік лампи ML-T8-13W/0.6-SMD складає 1950 лм. З формули 6.4.1 виразимо число ламп в приміщенні та підставляючи відомі значення в вираз одержимо

$$n = \frac{400 \times 30 \times 1.25 \times 1.2}{1750 \times 0.69} = 18 \quad (6.4.5)$$

Округляючи значення до більшої цілої цифри, отримуємо, що вимагається 18 ламп. Якщо в світильник 2 лампи то нам необхідно 9 світильників, які необхідно розмістити рівномірно у три ряди по три світильника.

Приміщення, призначені для роботи ПК, повинні мати природне освітлення. Орієнтація вікон повинна бути на північ або на північний схід, вікна повинні мати жалюзі, які можна регулювати, або штори.

## 6.5 Електробезпека

Відповідно до НАОП 0.00-1.28-10 Правила охорони праці під час експлуатації електронно-обчислювальних машин, ЕОМ з ВДТ і ПП, інше устаткування (апарати управління, контрольно-вимірювальні прилади, світильники), електропроводи та кабелі за виконанням і ступенем захисту мають відповідати класу зони за НПАОП 40.1-1.01-97, мати апаратуру захисту від струму короткого замикання та інших аварійних режимів.

Під час монтажу та експлуатації ліній електромережі необхідно повністю унеможливити виникнення електричного джерела загоряння внаслідок короткого замикання та перевантаження проводів, обмежувати застосування проводів з легкозаймистою ізоляцією і, за можливості, застосовувати негорючу ізоляцію.

Під час ремонту ліній електромережі шляхом зварювання, паяння та з використанням відкритого вогню необхідно дотримуватися НАПБ А.01.001-2004.

Лінія електромережі для живлення ЕОМ з ВДТ і ПП виконується як окрема групова трипровідна мережа шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Не допускається використовувати нульовий робочий провідник як нульовий захисний провідник.

Нульовий захисний провідник прокладається від стійки групового розподільного щита, розподільного пункту до розеток електроживлення.

Не допускається підключати на щиті до одного контактного затискача нульовий робочий та нульовий захисний провідники.

Площа перерізу нульового робочого та нульового захисного провідника в груповій трипровідній мережі має бути не менше площі перерізу фазового провідника. Усі провідники мають відповідати номінальним параметрам мережі та



навантаження, умовам навколишнього середовища, умовам розподілу провідників, температурному режиму та типам апаратури захисту, вимогам НПАОП 40.1-1.01-97.

У приміщенні, де одночасно експлуатуються понад п'ять ЕОМ з ВДТ і ПП, на помітному та доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення.

ЕОМ з ВДТ і ПП повинні підключатися до електромережі тільки за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення. У штепсельних з'єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного провідника. Їхня конструкція має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним.

Не допускається підключати ЕОМ з ВДТ і ПП до звичайної двопровідної електромережі, в тому числі - з використанням перехідних пристроїв.

Електромережі штепсельних з'єднань та електророзеток для живлення ЕОМ з ВДТ і ПП потрібно виконувати за магістральною схемою, по 3-6 з'єднань або електророзеток в одному колі.

Штепсельні з'єднання та електророзетки для напруги 12 В та 42 В за своєю конструкцією мають відрізнятися від штепсельних з'єднань для напруги 127 В та 220 В. Штепсельні з'єднання та електророзетки, розраховані на напругу 12 В та 42 В, мають візуально (за кольором) відрізнятися від кольору штепсельних з'єднань, розрахованих на напругу 127 В та 220 В.

Індивідуальні та групові штепсельні з'єднання та електророзетки необхідно монтувати на негорючих або важкогорючих пластинах з урахуванням вимог НПАОП 40.1-1.01-97 та НАПБ А.01.001-2004.

Електромережу штепсельних розеток для живлення ЕОМ з ВДТ і ПП при розташуванні їх уздовж стін приміщення прокладають по підлозі поруч зі стінами приміщення, як правило, в металевих трубах і гнучких металевих рукавах, а також у пластикових коробах і пластмасових рукавах з відводами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. При розміщенні в приміщенні до п'яти ЕОМ з ВДТ і ПП

допускається прокладання трипровідникового захищеного проводу або кабелю в оболонці з негорючого чи важкогорючого матеріалу по периметру приміщення без металевих труб та гнучких металевих рукавів.

При організації робочих місць операторів електромережу штепсельних розеток для живлення ЕОМ з ВДТ і ПП у центрі приміщення прокладають у каналах або під знімною підлогою в металевих трубах або гнучких металевих рукавах. При цьому не допускається застосовувати провід і кабель в ізоляції з вулканізованої гуми та інші матеріали, які містять сірку.

## 6.6 Забезпечення пожежної безпеки в розроблювальному проекті

Пожежна безпека забезпечена у відповідності з НАПБ А.01.001-2004 «Правила пожежної безпеки в Україні», який є обов'язковим для виконання всіма підприємствами не залежно від форми власності. Правила встановлюють загальні вимоги з пожежної безпеки. Забезпечуючи пожежну безпеку, слід також керуватись ПУЕ та НПАОП 40.1-1.32-01 «Правила побудови електроустановок. Електрообладнання спеціальних установок» та інших нормативних документів, що стосуються штучного освітлення і електротехнічних пристроїв, а також вимог нормативно-технічної експлуатаційної документації заводу-виробника.

Робоче приміщення лабораторії за класифікацією пожежонебезпечності має відноситись до категорії Д.

Забезпечення пожежної безпеки в лабораторії досягається за рахунок застосування мір пожежної профілактики й активного пожежного захисту, тобто комплексу мір попередження виникнення пожеж або зменшення їх наслідків. Причинами виникнення пожежі електроустаткування можуть бути:

1. перевантаження проводів;
2. неякісне виконання з'єднань електропроводки;
3. перевантаження різних електричних пристроїв;
4. коротке замикання

5. контакт горючих речовин з нагрівальними пристроями.

Джерела електричної енергії (розподільчі пристрої, трансформатори) розташовувані у відокремлених приміщеннях.

Освітлювальну електричну мережу виконано згідно вимог ПЕУ – правилам устрою електроустановок для пожежонебезпечних зон. Прокладання кабелю через перекриття, стіни, фальшпідлогу здійснено в сталевих трубах з наповнювачем з негорючих матеріалів. Аварійні мережі освітлення, дистанційного та автоматичного пуску протипожежних систем та сигналізації прокладено окремо від силових та інших електричних комунікацій, а при сумісному прокладанні їх розділено перегородками з негорючих матеріалів (метал, гетинакс).

Повітропроводи виконані з негорючих матеріалів. Система вентиляції обладнана пристроєм, що забезпечує автоматичне її відключення, а також перекриття повітропроводів лабораторії автоматичними заслінками в разі виникнення пожежі. Кабельні вертикальні шахти розділені по поверхах діафрагмами з негорючих матеріалів.

Ефективність застосування вогнегасника, у першу чергу пов'язана з правильним вибором його типу залежно від класу пожежі, яку не необхідно погасити. Основні вимоги до оснащення об'єктів вогнегасниками регламентуються НАПБ Б.03.001-2004 Типові норми належності вогнегасників. При експлуатації вогнегасників слід керуватись НАПБ Б.01.008-2004 Правила експлуатації вогнегасників.

Для гасіння та локалізації пожежі до прибуття пожежних підрозділів використовуються ручні вогнегасники. У приміщенні необхідний 1 вуглекислотний вогнегасник типу ВВ (ВВ-2, ВВ-5, ВВ-8). Застосування вуглекислотних вогнегасників зумовлено тим, що вони можуть використовуватися для гасіння дорогого обладнання, яке знаходиться під напругою до 1000 В.

Для виявлення пожежі використовують пожежно-охоронну сигналізацію, у відповідності до ДСТУ EN54-2:2003. Пожежні сповіщувачі використовуються для формування командного імпульсу автоматичного пуску системи автоматичного пожежегасіння. Кількість теплових пожежних сповіщувачів визначається за таблицею і для приміщення розмірами 6 x 5 x 3 м становить 2. Температура спрацювання сповіщувачів встановлюється не менше ніж на 20°C вище максимальної припустимої температури в приміщенні.

Для виявлення пожежі, замість старих точкових пристроїв пропонується встановити датчики Honeywell Notifier SFAPT-453(A) (Acclimate Plus™ Multi-Sensor Low-Profile Smoke Intelligent Detector). Датчик використовує аналізатор диму та комбінацію фотоелектричних та температурних сенсорів з вмонтованим для підвищення імунітету до фальшивого спрацювання. Пристрій обладнаний мікропроцесором для обробки інформації, в результаті він налаштовує чутливість автоматично не залежно від оператора контрольної панелі та проводить самотестування. Іншою перевагою, даного типу датчика є, його безпосередня зв'язаність з системою кондиціонування та вентиляції. В разі пожежі вимикається та блокується кондиціонер, і закриваються заслонки вентиляційної системи.

Дані з сенсорів подаються на загальну панель керування пожежно-охоронної системи сигналізації, а далі на пост чергового пожежної частини. Оператор або панель керування автоматично приймають рішення, щодо вимкнення постачання електроенергії до приміщення лабораторії

Евакуація здійснюється відповідно до НАПБ А.01.003-2009 "Правила улаштування та експлуатації систем оповіщення про пожежу та управління евакуацією людей в будинках та спорудах". Комплекс виробничих приміщень має два евакуаційних виходи. Двері на шляхах евакуації мають відчинятися у напрямку виходу зі споруди, ширина шляхів – не менше 1м, а ширина дверей – 0.8м. Шляхи евакуації показані на рисунку 6.6.1.

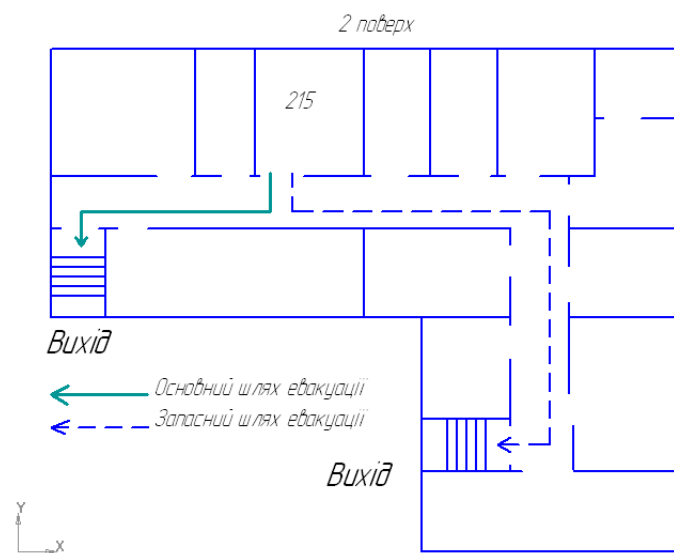


Рисунок 6.6.1 – План евакуації

## 6.7 Висновок

В роботі проаналізовано основні небезпечні чинники, можна відзначити, що при дотриманні правил безпеки і виробничої санітарії обладнання навігаційної системи не є пожежонебезпечним. Запропоновані світлодіодні світильники мають строк служби 50 тисяч годин, що значно краще ніж у люмінесцентних ламп, де строк рівний 10 - 20 тисяч годин, і крім того залежить від кількості переключень. З іншого боку світильники є економічнішими на 44.4 % (світлодіодна лампа 20 +/- 1 Вт, люмінесцентна 36 +/-1Вт), більш ударостійкі, не містять токсичних речовин і не мають спеціальних вимог щодо утилізації. Ці лампи створюють оптимальні умови для зорової роботи інженера програміста, а порівняно не висока температура нагрівання підвищує рівень пожежної безпеки.

					НАУ 11 54 75 000 ПЗ	Аркул
Зм.	Лист	№ докум.	Підп.	Дата		125

## 7 Охорона навколишнього середовища

### 7.1 Дослідження екологічного впливу авіаційного транспортного комплексу

У результаті авіатранспортних перевезень відбувається забруднення ґрунтів, водних об'єктів та атмосфери, а сама специфіка впливу повітряного транспорту на довкілля виявлена в значній шумовій дії та значних викидах різноманітних забруднюючих речовин.

Негативна дія різних авіаційних джерел шуму, в першу чергу, здійснюється на операторів, інженерів та техніків виробничих підрозділів. Так історично склалося, що аеропорти розташовані поблизу густозаселених районів міста. Тому з ростом міст та інтенсифікацією авіатранспортних процесів постає серйозна проблема співіснування міста та аеропорту. Населення авіаміста та розташованих поблизу селищ відчувають шум від літаків, що пролітають. У меншій мірі відчувають шум персонал аеропортів, авіапасажири та відвідувачі.

Крім шуму авіація призводить до електромагнітного забруднення середовища. Його викликає радіолокаційна та радіонавігаційна техніка. Аеропорти України здійснюють вплив на довкілля через стаціонарні джерела прямої та непрямої дії на навколишнє середовище, які розташовані в авіатехнічній базі, аеро-вокзальному комплексі з привокзальною площею, складах паливно-мастильних матеріалів, котельних, сміттєспалювальних станціях. Кількість шкідливих речовин, які потрапили у 2000 році в атмосферу від стаціонарних джерел в аеропортах, склала 23,1 тисяч тонн. Разом з викидами забруднюючих речовин парк літаків споживає у великій кількості кисень.

В аеропортах накопичуються тверді та рідкі відходи споживання та виробництва. У багатьох випадках ці відходи безпечні у санітарно-гігієнічному співвідношенні. Об'єми накопичення твердих відходів у 2000 році склали: виробничі відходи — 43 тис. т; побутові відходи — 79,9 тис. т; відходи, які вида-

					НАУ 11 54 75 000 ПЗ			
Зм.	Лист	№ докум.	Підп.	Дата	Інтегрована інерціально-супутникова система навігації, що базується на принципах комплексної обробки інформації з використанням калманівської фільтрації	Лит.	Аркуш	Аркушів
Розробив	НовікМ.В.						126	152
Перевірив	Мар'ясоваТ.І.							
Консульт.	ФіляшкінМ.К.							
Н. контр.	КозловаА.П.							
Затвердив	СинєглазовВ.М.					ІАСУ 608		

ляються з літаків міжнародних авіаліній, — 2,1 тис. т. Відходами у аеропортах зайнято спеціальні приміщення площею до 3,3 тис. кв.м, а площа відкритих сховищ (звалищ) складає 118,7 тис. кв.м, з них тільки 18% спеціально підготовлені для зберігання та накопичення відходів.

У цивільній авіації авіаремонтні заводи та аеропорти із спецавтотранспортом є найбільш інтенсивними джерелами забруднення природної води. Стічні води авіаремонтних підприємств та аеропортів складаються з виробничих і господарсько-побутових стічних вод та поверхневих стоків.

Кількість стічних вод і їх склад змінюються протягом доби, тижня, місяця. Для ряду виробничих процесів характерний залповий скид сильно концентрованих стічних вод. Найбільшу небезпеку для водних об'єктів становлять стоки з території аеропорту: передангарного та доводневого майданчиків, складів паливо-мастильних матеріалів, майданчиків для миття.

Поверхневі стоки з територій транспортних підприємств містять рідкі нафтопродукти, залишки миючих, дезінфікуючих, антиобмерзаючих і протиожеледних реагентів, формувальних сумішей, розчинів, використовуваних у металообробці, відпрацьовані електроліти акумуляторних батарей, продукти руйнування штучних покрівель і зносу шин.

Атмосферні опади, потоки дощових та талих вод також поглинають частину димових газів котелень, шкідливих викидів авто- та авіатранспорту, які осідають на аеродромі.

У пришляховому просторі при зльоті літака приблизно 50% викидів у вигляді мікрочастинок відразу розсіюється на прилеглих до аеропорту територіях. Нагромадження забруднюючих речовин у пришляховій смузі призводить до забруднення екосистем і робить ґрунти на прилеглих територіях непридатними до сільськогосподарського використання.

Токсичні забруднюючі речовини з пересувних і стаціонарних джерел поділяються за ступенями небезпеки на 4 класи:

1. надзвичайно небезпечні (тетраетилсвинець, свинець, ртуть та ін.);
2. високо небезпечні (марганець, мідь, сірчана кислота, хлор та ін.);
3. помірно небезпечні (ксилол, метиловий спирт та ін.);
4. малонебезпечні (аміак, бензин паливний, газ, оксид вуглецю, скипидар, ацетон та ін.).

В авіації існують обмежуючі норми припустимих викидів (МДВ), які встановлені ІКАО[15], а в деяких країнах також прийняті національні норми на чотири основні шкідливі компоненти: СО, СН, М та частинки сажі (дим).

При нормуванні та виявленні викиду шкідливих речовин беруться до уваги всі маневри літака та відповідні режими роботи двигуна, які відбуваються в зоні аеропорту на висоті польоту до 1 км. Для визначення кількісних та якісних показників речовин потрібно знати етапи, на яких вони відбуваються, розподіл використаних режимів роботи двигуна по етапам, а також їх довготривалість за злітно — посадочний цикл.

Для оцінки кількості викидів шкідливих речовин в атмосферу вводять поняття індексу викиду за злітно - посадочний цикл,  $E_1$  - це відношення кількості грамів шкідливої речовини до 1 кг згорілого палива.

Утворення окису азоту протікає при достатньо високих температурах ( $T=2000\text{ K}$ ). Інтенсивність цього процесу значно збільшується зі зростанням значень температури і часу перебування суміші в камері згорання. Тому максимальний викид окисів азоту спостерігається на злітному режимі роботи двигуна.

В якості контрольного параметру емісії приймають відношення маси забруднюючої речовини у грамах, яка виділяється за стандартний цикл, до злітної тяги двигуна в ньютонах. Знаючи індекс емісії на кожному режимі роботи двигуна, можна, шляхом складання, визначити масу емісії за весь цикл.

Таким чином, авіація є джерелом досить широкого спектру факторів негативного впливу на довкілля. У зв'язку з цим своєчасною і актуальною задачею є розробка і впровадження державних нормативних актів, що регламентували б розташування населених пунктів поблизу аеропортів, а також є доцільною розробка заходів та рекомендацій щодо зниження негативного впливу авіатранспортних процесів на довкілля.



## 7.2 Аналіз впливу шуму повітряних суден на навколишнє середовище

Людина завжди жила в світі звуків і шуму. Звуком називають такі механічні коливання зовнішнього середовища, які сприймаються слуховим апаратом людини (від 16 до 20 000 коливань в секунду). Коливання більшої частоти називають ультразвуком, меншою, — інфразвуком. Шум — це набір звукових коливань приблизно однакової амплітуди широкого спектру.

Авіаційний шум в силу своїх особливостей займає окреме місце серед транспортних джерел шуму внаслідок підвищених рівнів звуку (95-100 дБ поблизу кордону аеропорту), широкосмугового спектрального складу.

Авіаційний шум несприятливо впливає на широке коло осіб, які безпосередньо пов'язані з діяльністю цивільної авіації: льотно-технічний склад, працівників підприємств цивільної авіації та авіапасажирів, а також населення, що проживає поблизу аеропортів. Несприятливий вплив шуму на людину пов'язаний з загальним роздратуванням, перешкодами розмові, неможливістю заснути, неможливістю зосередитись для виконання конкретної роботи, а при тривалому впливі шуму — втратою слуху та здоров'я. Такий вплив залежить від реакції людини на шум та фізичних характеристик шуму — інтенсивності та спектру, а також тривалості впливу.

Розрізняють три типи критеріїв оцінки подразнюючого впливу шуму:

1. максимальні рівні шуму з урахуванням психофізіологічної реакції людини на шум;
2. ефективні рівні шуму, що характеризуються впливом шуму при польоті літака з урахуванням часу його звучання,
3. критерії сумарного впливу шуму, що враховують не тільки максимальні рівні шуму при кожному прольоті, а також їх кількість за певний час доби.

Були проведені дослідження, по даним яких слід очікувати, що максимальні зони зашумлення будуть спостерігатись при зльотах та прольотах по трасам літаків Ту-154 та Іл-86.

Для зниження шуму використовується обладнання бар'єру (екрану) на шляху розповсюдження шуму. Для цього використовуються спеціальні конструкції, земляні відкопи, будівлі нежитлового призначення, а також смуги зелених насаджень.

### 7.3 Аналіз впливу радіохвиль на навколишнє середовище

З того часу, коли почалося практичне використання радіо, люди почали спостерігати шкідливий вплив радіохвиль на організми живих істот, у тому числі й людей.

Радіохвилі – це електромагнітні коливання, що розповсюджуються в просторі із швидкістю світла (300 000 км/сек).

Радіохвилі переносять через простір енергію, що випромінюється генератором електромагнітних коливань. Електромагнітне випромінювання характеризується частотою, довжиною хвилі і потужністю переносної енергії. Частота електромагнітних хвиль показує, скільки разів в секунду змінюється у випромінювачі напрям електричного струму і, отже, скільки разів в секунду змінюється в кожній точці простору величина електричного і магнітного полів.

Оточуюче нас середовище завжди перебувало під впливом електромагнітних полів. Ці поля називаються фоновим випромінюванням та спричинені природою. З розвитком науки й техніки фонове випромінювання значно підсилювалося. Тому електромагнітні поля, які можна віднести до антропогенних, значно перевищують природний фон і останнім часом перетворилися на небезпечний екологічний чинник.

Як відомо, основний принцип роботи нервової системи людини - передача електромагнітних імпульсів від однієї клітки до іншої. Але ж людина живе в світі, насиченому електромагнітними полями, постійно піддаючись їх шкідливій дії, їх створюють будь-які електричні прилади, теле- і радіоантени, тролейбуси і трамваї. Але найбільшу частину шкідливої дії людина отримує у себе удома або на своєму робочому місці.

## 7.4 Характеристика ПК як джерела забруднення

Усі елементи, які є складовими частинами персонального комп'ютера (ПК), такі, як системний блок, різні пристрої введення/виведення інформації, засіб візуального відображення інформації, формують складний електромагнітний стан на робочому місці користувача, що вносить свій негативний внесок на навколишнє середовище.

Основними факторами несприятливого впливу роботи з ПК є ергономічні параметри екрана монітора (зниження контрасту зображення в умовах інтенсивного зовнішнього освітлення, дзеркальні відблиски від передньої поверхні екранів моніторів, наявність мерехтіння зображення на екрані монітора). Випромінювальні характеристики монітора:

1. електромагнітне поле монітора в діапазоні частот 20 Гц–1000 МГц;
2. статичний електричний заряд на екрані монітора;
3. ультрафіолетове випромінювання в діапазоні 200–400 нм;
4. інфрачервоне випромінювання в діапазоні 1050 нм – 1 мм;
5. рентгенівське випромінювання  $> 1,2$  КеВ.

## 7.5 Вплив на здоров'я користувача електромагнітних полів ПК

Вплив електромагнітних полів на людину має негативні наслідки для життєво важливих систем людини і може стати причиною важких захворювань. Адже на біологічну реакцію людини впливають такі параметри електромагнітних полів ЕОМ, як інтенсивність і частота випромінювання, тривалість опромінення і модуляція сигналу, частотний спектр і періодичність дії.

Так, деякі дослідження показали, що навіть при короткочасній роботі (45 хвилин), в організмі користувача, під впливом електромагнітного випромінювання монітора відбуваються значні зміни гормонального стану і специфічні

зміни біострумів мозку. А збільшення часу користування ПК стає причиною різних важких захворювань. Згідно статистики, у працюючих за монітором від 2 до 6 годин на добу функціональні порушення центральної нервової системи відбуваються в середньому в 4,6 рази частіше, ніж у контрольних групах, хвороби серцевосудинної системи — у 2 рази частіше, хвороби верхніх дихальних шляхів — у 1,9 рази частіше, хвороби опорно-рухового апарата - у 3,1 рази частіше. Як результат — при восьмигодинній роботі на протязі 4 місяців спостерігається зниження імунітету на 95%.

## 7.6 Комп'ютер як джерело електростатичного поля

Кожен персональний комп'ютер включає засіб візуального відображення інформації - монітор. Як правило, це пристрій на основі електронно-променевої трубки. Люди, що працюють з монітором, здобувають електростатичний потенціал. Електростатичне поле (Естп) створюється накопиченням електростатичного заряду на екрані кінескопа при роботі монітора. Розкид електростатичних потенціалів користувачів коливається в діапазоні від -3 до +5 КВ. Крім того, внеском у загальне електростатичне поле являються клавіатури, що електризуються від тертя поверхні, і миші. Експерименти показують, що навіть після роботи з клавіатурою, електростатичне поле швидко зростає з 2 до 12 КВ/м. На окремих робочих місцях в області рук реєструвалися напруженості статичних електричних полів більш 20 КВ/м.

## 7.7 Комп'ютер як джерело рентгенівського випромінювання

Крім причиною створення електростатичного поля, є джерелом рентгенівського, бета - і гамма-випромінювань. Таке випромінювання виникає при

роботі монітора за рахунок гальмування пучка електронів і як характеристичне випромінювання атомів матеріалів кінескопа. Спектр рентгенівського випромінювання є безперервним, максимальна енергія якого - 20 KeV. Джерелом бета-, гамма-випромінювання, які присутні і при включеному і при виключеному моніторі, є радіоактивний розпад ядер сімейств урану і торію, а також ядер калію-40. Спектральний склад гамма-випромінювання переважно складається з набору моноенергетичних ліній. спектральний склад бета- випромінювання безперервний, а його максимальна енергія -1.3 MeV.

Шкідливий вплив на людину дії іонізуючих випромінювань може призвести до помутніння кришталика ока. Для запобігання такої шкоди здоров'ю людини, у моніторах була знижена анодна напруга, а в скло моніторів доданий свинець. Серед вказаних вище негативних впливів ПК на здоров'я людини, можна назвати ще й шум в приміщеннях, обладнаних комп'ютерами, рівень якого в таких приміщеннях іноді досягає 85 дБ. Одними з джерел шуму є принтери, техніка й обладнання для кондиціонування повітря, у самих ПЕОМ - вентилятори систем охолодження і трансформатори.

## 7.8 Оптимальні шляхи утилізації ЕОМ

Сьогодні ми не можемо уявити собі наше життя без персонального комп'ютера (ПК), електроніки, мобільного телефону. ПК усюди – вдома, на роботі, школі, ВНЗі, аптеці, банку, на вокзалі, магазині, лікарні й т.д. Бази даних, потоки інформації, Internet – звичні слова й поняття.

Кількість електроніки росте з кожним роком. Стрімко застаріває техніка, їй на зміну приходять могутніші, більш сучасні ПК й оргтехніка. Людство, хоче воно цього чи ні, втягнуто в постійний процес модернізації й заміни електронної техніки. Ми радуємося новим моделям персональної техніки з новими можливостями. Ми даруємо один одному сучасні, престижні марки мобільних телефонів.

Поступово виникає проблема: а що робити зі старою технікою, що морально застаріла або по тим чи іншим причинам вийшла з ладу. Виникає по-

няття "електронне сміття". Про величини цього сміття ми можемо тільки догадуватися.

Екологи б'ють тривогу, грозять санкціями провідним виробникам електроніки, якщо ті не вживуть заходів по утилізації персональної й іншої техніки.

За період з 1991 року по теперішній час в Україну завезене різними постачальниками близько 10 млн. одиниць (близько 400000 т) персональної й оргтехніки (це по самих скромних підрахунках), мобільних телефонів – 37-40 млн. шт. (близько 4800 т). США, Європа, Японія починають задихатися від електронного сміття. У цих країнах скопалося більше сотні мільйонів тільки кінескопів і моніторів.

Таким чином, звичайний персональний комп'ютер є складом як коштовних металів – міді, срібла й золота, так і небезпечних матеріалів – кадмію, свинцю, цинку, нікелю, ртуті. До них додаються пластмаси, індикатори, монітори на рідких кристалах, батареї – усього понад 90 компонентів. Однак вартість переробки й витяг коштовних металів перевищує вартість самих металів. Але людство приречене на витрати по утилізації "електронного сміття".

## **7.9 Рекомендації щодо зменшення негативного впливу ПК на здоров'я людини та навколишнє середовище**

Перш за все, необхідно скорочувати час роботи за комп'ютером, або якнайчастіше робити перерву в роботі.

Серед основних правил, які слід пам'ятати при роботі з ПК, є те, що не слід залишати комп'ютер включеним на тривалий час, якщо він не використовується, рекомендується використовувати "сплячий режим" для монітора. У зв'язку з тим, що електромагнітне випромінювання максимальне збоку монітора, необхідно розташовувати монітор таким чином, щоб він не випромінював на сусідні робочі місця. Оптимальною відстанню розташування монітора від користувача є більш 1,2 м, критичною - 1,2 м. На даний час, широкого розповсюдження набувають рідинно кристалічні монітори, випромінювання яких

значно менше, ніж у моніторів з електроннопроменевою трубкою. Також, комп'ютер повинен бути заземлений, при наявності захисного екрана, його теж варто заземлити.

В Україні безпека рівнів іонізуючих випромінювань комп'ютерних моніторів регламентується нормами [16]. Стандарти обмежують потужність дози рентгенівського випромінювання величиною 100 мкР/год на відстані 5 см від поверхні екрана монітора і встановлюють для населення межа річної еквівалентної дози випромінювань на кришталік ока рівний 15 мЗв. Потужність дози гамма-випромінювання на відстані 5 см від екрана монітора незначна (0.03-0.1 мкР/год) і складає 0.5% від потужності дози тіла, щільність потоку бета-випромінювання на відстані 5 см від екрана монітора може складати 0.2-0.5 част/см<sup>2</sup>, максимальна потужність дози рентгенівського випромінювання на відстані 5 см від екрана монітора порівнянна з фоном і не перевищує 5-15 мкР/год.

Звідси випливає, що дана відстань від екрану монітора є оптимальною для людини при роботі з ПК і потужність еквівалентної дози випромінювань за такої відстані складе 0.3-0.4 мкЗв/год. І оскільки накопичена хрусталиком ока річна еквівалентна доза ( 0.7 мЗв) у 20 разів менше припустимого нормами [16] значення, це свідчить про радіаційну безпеку комп'ютерних моніторів.

Заради мінімізації забруднення навколишнього середовища через шкідливий вплив виробництва на нього, в даний час широко використовуються такі методи вирішення проблеми екологічності виробництва, суть яких зводиться до обмеження кількості забруднюючих речовин. Вирішення проблеми екологічної безпеки експлуатації мікросхем від використання електроенергії на сьогоднішній день складається в раціональному використанні енергії, застосуванні нетрадиційних методів її вироблення. Для запобігання радіозабруднення, апаратуру розміщують в екранованому корпусі.

# Висновки

Результатами виконання роботи є

1. розроблений принцип діагностування САУ на базі єдиного методологічного підходу;
2. опрацьовані та обґрунтовані основні алгоритми для використання в автоматизованих системах діагностування САУ;
3. розроблене програмне забезпечення, що дає змогу автоматизувати процес оптимізації параметрів за умови наявності математичної моделі.

Для покращення роботи в майбутньому запропоновано наступні вдосконалення:

1. перевести всі напіваавтоматичні дії, виконувані в середовищі розробленої програми, в автоматичні з можливістю втручання користувача в хід роботи;
2. реалізувати індикацію основних показників якості системи у головному вікні;
3. виконати поділ діаграми стійкості на зони з окремим виділенням зони оптимальних регулювань згідно заданих критеріїв;
4. додати можливість створення власних моделей систем для подальшого їх дослідження а також можливість збереження результатів роботи та експорт у загальновживані формати;
5. забезпечити роботу з багатьма моделями одночасно;
6. реалізувати порівняльне дослідження шляхом індикації показників системи з заданими параметрами з попередніми налаштуваннями математичної моделі системи;



## Перелік посилань

1. М.К. Філяшкін В.О. Рогожин, А.В. Скрипець, Т.І. Лукінова Інерціально-супутникові навігаційні системи. - К.: Вид-во НАУ, 2009. - 306 с.
2. Ільїн О.Ю., Філяшкін М.К., Черних Ю.О. Пілотажно-навігаційні системи та комплекси. - К.: Вид-во КІ ВПС, 1999. - 335 с.
3. Интегрированные инерциально-спутниковые навигационные системы // Под ред. В.А. Пешехонова. - С.-Петербург: 2001. - 235 с.
4. Интегрированные комплексы на базе ИНС и приемника «Навстар» // Новости зарубежной науки и техники, Серия «авиационные системы». ГосНИИ-АС, 1995, №10-12.
5. Кузовков Н.Т., Салычев О.С. Инерциальная навигация и оптимальная фильтрация. - М.: Машиностроение, 1988. - 216 с.
6. Рогожин В.О., Синеглазов В.М., Філяшкін М.К. Пілотажно-навігаційні комплекси повітряних суден -К.: Вид-во НАУ, 2005. - 316с.
7. Grewal M. S. Global Positioning Systems, Inertial Navigation and Integration / M. S. Grewal, L.R.Weill, A. P. Andrews. - A John Wiley and Sons, Inc., Publication, 2007. - 525 p.
8. Соловьев Ю.А. Системы спутниковой навигации - М.: ЭКО-ТРЕНДЗ, 2000. - 270 с.
9. J. Zander, B. Slimane, and L. Ahlin, Principles of Wireless Communications.// Stockholm: Royal Institute of Technology, 2005.
10. A. El-Rabbany, Introduction to GPS - The Global Positioning System // London: Artech House, 2002
11. USCG Navigation Center, GPS Standard Positioning Service - Performance Standard, // USCG Navigation Center, Oct. 2001.
12. B. W. Parkinson and J. J. Spilker, Global Positioning System: Theory and Applications. Progress in Astronautics and Aeronautics, 1996, vol. 163.
13. R. S. Bucy and P. D. Joseph, Filtering for Stochastic Processes, with Applications to Guidance, Wiley, New York, 1968.

14. R. E. Kalman, A new approach to linear filtering and prediction problems, ASME Journal of Basic Engineering, Vol. 82, pp. 34-45, 1960.
15. R. E. Kalman, New methods in Wiener filtering, in Proceeding of the First Symposium on Engineering Applications of Random Function Theory and Probability, Wiley, New York, 1963.
16. Авиационные приборы и навигационные системы // Под ред. Бабича О.А. - М.: Изд-во ВВИА им. проф. Н.Е. Жуковского, 1981. - 648 с.
17. Бабич О.А. Обработка информации в навигационных комплексах. - М.: Машиностроение, 1988. - 212 с.
18. Власенко А. В. Интегральные гироскопы iMEMS - датчики угловой скорости фирмы Analog Devices (Интернет-издание), 2006.
19. Воробьев В.Г., Глухов В.В., Кадышев И.К. Авиационные приборы, информационно-измерительные системы и комплексы. - М.: Транспорт, 1992. - 399 с.
20. Глобальная спутниковая навигационная система ГЛОНАСС // под ред. В.Н. Харисова, А.И.Петрова, В.А.Болдина. - М.: ИПРЖР, 1998. - 400 с.
21. I. Rhee, M.F. Abdel-Hafez, and J.L. Speyer, Observability of an integrated GPS/INS during maneuvers. IEEE Transactions on Aerospace Electronic Systems, Apr. 2004, pp. 526-535

Додаток А

Код програми jRKalman

Математико-графічне ядро програми

Listing 1: FlightPathGen.java

```
1/**
 *
 */
package jKalman;
5
/**
 * @author Mykola Novik
9 * java -Dsun.java2d.pmosffscreen=false
 */
import javax.swing.*;
import java.awt.*;
13//import java.awt.event.*;

//import java.util.*;

17//import java.awt.Color;
//import java.awt.LayoutManager;
//import java.io.*;
//import math lib
21//import java.lang.Math.*;

// import math array lib
//import static org.math.array.DoubleArray.*;
25//import static org.math.array.LinearAlgebra.*;
//import static org.math.array.StatisticSample.*;

// import input/output library
29//import static org.math.io.files.ASCIIFile.*;
//import static org.math.io.parser.ArrayString.*;
// import graph plot library
import org.math.plot.*;
33
public class FlightPathGen {

37 // Radius of Earth, m
final static double EARTH_RADIUS = 6378137 ;
// Angular velocity of Earth, rad/sec
final static double EARTH_SPEED = 7.292115E-5;
41 // Const for WGS84 e^2
final static double E2 = 6.6943799901413E-3;
final static double MIU = 398600.44E9; // m^3/c^2
final static double GE = MIU/(Math.pow(EARTH_RADIUS,2));

45
double simTime ;
// Step time, s
double stepTime ;
49 int stepCount ;
double [] t;
// Parameters of motion of aircraft
// Start point on surface of Earth, lat, long, height
53 double Phi0 ;
double Lam0 ;
double H0 ;

57 // Parameter of oscilation of flight path
double TPhi ;
double TLam ;
double TH ;

61
```

```

        double kPhi ;
        double kLam ;
        double deltaPhi ;
65      double deltaLam ;
        double deltaH ;

        double ddeltaPhi ;
69      double ddeltaLam ;
        double ddeltaH ;

        double WPhi ;
73      double WLam ;
        double WH ;

        double[] Phi;
77      double[] Lam;
        double[] H;
        double[] dPhi;
        double[] dLam;
81      double[] dH;
        double[] ddPhi;
        double[] ddLam;
        double[] ddH;
85      double[] R1;
        double[] R2;
        double[] dR1;
        double[] dR2;
89      double[] VE;
        double[] VN;
        double[] VH;
        double[] VR;
93      double[] dVE;
        double[] dVN;
        double[] dVH;
        double[] AE;
97      double[] AN;
        double[] AH;
        double[] q;
        double[] g;
101     double[] Pitch;
        double[] Yaw;
        double[] Roll;

105     FlightPathGen(){
        ///////////////////////////////////////////////////////////////////
        ///////////////////////////////////////////////////////////////////
        ///////////////////////////////////////////////////////////////////
109     simTime = 1000 ;
        // Step time, s
        stepTime = 1;
        stepCount = (int) (simTime/stepTime);

113     // Parameters of motion of aircraft
        // Start point on surface of Earth, lat, long, height
        Phi0 = 55*(Math.PI/180) ;
117     Lam0 = 30*(Math.PI/180) ;
        H0 = 5000;

        // Parameter of oscilation of flight path
121     TPhi = 600;
        TLam = 600;
        TH = 600;

125     kPhi = 0;
        kLam = 0.00004;
        deltaPhi = 0.004;
        deltaLam = 0;
129     deltaH = H0;

        ddeltaPhi = 0;
        ddeltaLam = 0;
133     ddeltaH = 0;

        WPhi = 2*Math.PI/TPhi;
        WLam = 2*Math.PI/TLam;
137     WH = 2*Math.PI/TH;
        t = loadTimeArray(simTime,stepTime);
        InitPathVars();

141 }
        FlightPathGen(double simtime, double steptime,

```

```

double phi0,double lam0,double h0,
double tphi, double tlam,double th,
double kphi, double klam,
double deltaphi, double deltalam, double deltah,
double ddeltaphi, double ddeltalam, double ddeltah){
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
simTime = simtime ;
// Step time, s
stepTime = steptime;
stepCount = (int) (simTime/stepTime);

// Parameters of motion of aircraft
// Start point on surface of Earth, lat, long, height
Phi0 = phi0 ;
Lam0 = lam0 ;
H0 = h0;

// Parameter of oscilation of flight path
TPhi = tphi;
TLam = tlam;
TH = th;

kPhi = kphi;
kLam = klam;
deltaPhi = deltaphi;
deltaLam = deltalam;
deltaH = deltah;

ddeltaPhi = ddeltaphi;
ddeltaLam = ddeltalam;
ddeltaH = ddeltah;

WPhi = 2*Math.PI/TPhi;
WLam = 2*Math.PI/TLam;
WH = 2*Math.PI/TH;
InitPathVars();
t = loadTimeArray(simTime,stepTime);
}

/*
FlightPathGen(double[] time, double[] pathPos0,
double[] pathTpos,double[] kPos,
double[] deltaPos, double[] ddeltaPos ){
simTime = time[0];
// Step time, s
stepTime = time[1];
stepCount = (int) (simTime/stepTime);

// Parameters of motion of aircraft
// Start point on surface of Earth, lat, long, height
Phi0 = pathPos0[0]*(Math.PI/180) ;
Lam0 = pathPos0[1]*(Math.PI/180) ;
H0 = pathPos0[2];

// Parameter of oscilation of flight path
TPhi = pathTpos[0];
TLam = pathTpos[1];
TH = pathTpos[2];

kPhi = kPos[0];
kLam = kPos[1];
deltaPhi = deltaPos[0];
deltaLam = deltaPos[1];
deltaH = deltaPos[2];

ddeltaPhi = ddeltaPos[0];
ddeltaLam = ddeltaPos[1];
ddeltaH = ddeltaPos[2];

WPhi = 2*Math.PI/TPhi;
WLam = 2*Math.PI/TLam;
WH = 2*Math.PI/TH;
InitPathVars();
t = loadTimeArray(simTime,stepTime);
}
*/

```

```

225 void InitPathVars() {

    Phi = new double[t.length];
    Lam = new double[t.length];
229 H = new double[t.length];
    dPhi = new double[t.length];
    dLam = new double[t.length];
    dH = new double[t.length];
233 ddPhi = new double[t.length];
    ddLam = new double[t.length];
    ddH = new double[t.length];
    R1 = new double[t.length];
237 R2 = new double[t.length];
    dR1 = new double[t.length];
    dR2 = new double[t.length];
    VE = new double[t.length];
241 VN = new double[t.length];
    VH = new double[t.length];
    VR = new double[t.length];
    dVE = new double[t.length];
245 dVN = new double[t.length];
    dVH = new double[t.length];
    AE = new double[t.length];
    AN = new double[t.length];
249 AH = new double[t.length];
    q = new double[t.length];
    g = new double[t.length];
    Pitch = new double[t.length];
253 Yaw = new double[t.length];
    Roll = new double[t.length];

}

257 static double[] loadTimeArray(double simT, double stepT){
    int stepCount = (int) (simT/stepT);
    double[] Time = new double[stepCount];
261 for (int i=0; i<stepCount; i++){
        Time[i] = i*stepT;
    }
    return Time;
265 }

void ComputePhiLamH () {
    /* phi = phi0 + Kphi*t + delta_phi*sin(w_phi*t + ddelta_phi);
    * lam = lam0 + Klam*t + delta_lam*sin(w_lam*t + ddelta_lam);
269 * h = h0 - delta_h0*cos(w_h*t+ddelta_h);
    * dphi = Kphi + delta_phi*w_phi*cos(w_phi*t + ddelta_phi);
    * dlam = Klam + delta_lam*w_lam*cos(w_lam*t + ddelta_lam);
273 * dh = delta_h0*w_h*sin(w_h*t+ddelta_h);
    * ddphi = - delta_phi*(w_phi^2)*sin(w_phi*t + ddelta_phi);
    * ddlam = - delta_lam*(w_lam^2)*sin(w_lam*t + ddelta_lam);
    * ddh = delta_h0*(w_h^2)*cos(w_h*t+ddelta_h);
277 * */
    WPhi = 2*Math.PI/TPhi;
    WLam = 2*Math.PI/TLam;
    WH = 2*Math.PI/TH;

281 for (int i=0; i<t.length; i++){
        Phi[i] = Phi0 + kPhi*t[i] + deltaPhi*Math.sin(WPhi*t[i] + ddeltaPhi);
        Lam[i] = Lam0 + kLam*t[i] + deltaLam*Math.sin(WLam*t[i] + ddeltaLam);
285 H[i] = H0 - deltaH*Math.cos(WH*t[i] + ddeltaH);

        dPhi[i] = kPhi + deltaPhi*WPhi*Math.cos(WPhi*t[i] + ddeltaPhi);
        dLam[i] = kLam + deltaLam*WLam*Math.cos(WLam*t[i] + ddeltaLam);
289 dH[i] = deltaH*WH*Math.sin(WH*t[i] + ddeltaH);

        ddPhi[i] = - deltaPhi*(Math.pow(WPhi,2))*Math.sin(WPhi*t[i] + ddeltaPhi);
        ddLam[i] = - deltaLam*(Math.pow(WLam,2))*Math.sin(WLam*t[i] + ddeltaLam);
293 ddH[i] = deltaH*(Math.pow(WH, 2)*Math.cos(WH*t[i] + ddeltaH));

    }
}

297 void ComputeR1R2() {
    /* a = 6378137; % m
    * e2 = 6.66943799901413*10^-3;
    * R1 = a./(1-(e2)*sin(phi).^2).^0.5;
301 * R2 = R1.*((1-e2)./(1-e2*(sin(phi).^2)));
    * dR1 = (a*e2*sin(phi).*cos(phi).*dphi)./(1-e2*sin(phi).^2).^1.5;
    * dR2 = dR1.*((1-e2)./(1-e2*sin(phi).^2)) + ...
    * R1.*((1-e2)*2*e2.*sin(phi).*cos(phi).*dphi)./(1-e2.*sin(phi).^2).^2;

```

```

305      * */
      for (int i=0; i<t.length; i++){
          R1[i] = EARTH_RADIUS/Math.sqrt(1-E2*Math.pow(Math.sin(Phi[i]),2));
          R2[i] = (R1[i]*(1-E2))/(1-E2*Math.pow(Math.sin(Phi[i]),2));
309      dR1[i] = (EARTH_RADIUS*E2*Math.sin(Phi[i])*Math.cos(Phi[i])*dPhi[i])/Math.pow(1-E2*Math.pow(Math.sin(Phi[i]),2),1.5);
          dR2[i] = dR1[i]*((1-E2)/(1-E2*Math.pow(Math.sin(Phi[i]),2))) + R1[i]*((1-E2)*2*E2*Math.sin(Phi[i])*Math.cos(Phi[i])*dPhi[i])/Math.pow(1-E2*Math.pow(Math.sin(Phi[i]),2),2);

      }
313 }
void ComputeVenhr(){
    /*
    * V_N = dphi.*(R2 +h);
317    * V_E = dlam.*(R1 +h).*cos(phi);
    * V_h = dh;

    * dV_N = ddphi.*(R2 + h) + dphi.*(dR2 + dh);
321    * dV_E = ddlam.*(R1 + h).*cos(phi) + dlam.*(dR1 + dh).*cos(phi)-...
    * dlam.*(R1 + h).*sin(phi).*dphi;
    * dV_h = ddh;
    q = dlam + 2*wz;
325    g = get_g(phi,h);
    */

    for (int i=0; i<t.length; i++){
329        VN[i] = dPhi[i]*(R2[i] +H[i]);
        VE[i] = dLam[i]*(R1[i] +H[i])*Math.cos(Phi[i]);
        VH[i]= dH[i];
        dVN[i] = ddPhi[i]*(R2[i] + H[i]) + dPhi[i]*(dR2[i] + dH[i]);
333        dVE[i] = ddLam[i]*(R1[i] + H[i])*Math.cos(Phi[i]) + dLam[i]*(dR1[i] + dH[i])*Math.cos(Phi[i]) - dLam[i]*(R1[i] + H[i])*Math.sin(Phi[i])*dPhi[i];
        dVH[i] = ddH[i];
        VR[i] = Math.sqrt(Math.pow(VE[i], 2) + Math.pow(VN[i], 2));

    }
337 }
void ComputeAenh(){
    /*
    * q = dlam + 2*wz;
341    * g = get_g(phi,h);
    * a_N = dV_N + q.*sin(phi).*V_E + dphi.*V_h;
    * a_E = dV_E - q.*sin(phi).*V_N+q.*cos(phi).*V_h;
    * a_h = dV_h - q.*cos(phi).*V_E - dphi.*V_N + g;
345    */

    for (int i=0; i<t.length; i++){

349        q[i] = dLam[i] + 2*EARTH_SPEED;
        g[i] = (MIU/Math.pow(EARTH_RADIUS,2))*(1 - 2*(H[i]/EARTH_RADIUS)+0.75*E2*Math.pow(Math.sin(Phi[i]),2));

        AE[i] = dVE[i] - q[i]*Math.sin(Phi[i])*VN[i]+q[i]*Math.cos(Phi[i])*VH[i];
353        AN[i] = dVN[i] + q[i]*Math.sin(Phi[i])*VE[i] + dPhi[i]*VH[i];
        AH[i] = dVH[i] - q[i]*Math.cos(Phi[i])*VE[i] - dPhi[i]*VN[i] + g[i];

    }
}
357 void ComputePRY(){
    /*
    * pitch = atan2(V_h,V_r);
    * yaw = atan2(V_E,V_N);
361    * Kg = 0.1; % rad/(m/s^2)
    * roll = Kg*(V_N.*dV_E - V_E.*dV_N)./(V_r.*cos(pitch));
    * pitch = atan2(V_h,V_r);*/
    for (int i=0; i<t.length; i++){
365        Pitch[i] = Math.atan2(VH[i],VR[i]);
        Yaw[i] = Math.atan2(VE[i],VN[i]);
        double Kg = 0.1; // rad/(m/s^2)
        Roll[i] = Kg*(VN[i]*dVE[i] - VE[i]*dVN[i])/(VR[i]*Math.cos(Pitch[i]));
369    }

    }

}

```

Listing 2: RunISNS.java

```

package jKalman;
//package flightpath;

```

4

					НАУ 11 54 75 000 ПЗ	Архив
						143
Зм.	Лист	№ докум.	Підп.	Дата		

```

import static org.math.io.parser.ArrayString.printDoubleArray;
import java.util.Random;

8public class RunISNS {
    double[][] Fisns;// = new double[22][22];
    double[][] Risns;// = new double[8][13];
    double[][] Qisns;// = new double[22][28];
12    double[][] Pp = new double[22][22];
    double[][] Pm = new double[22][22];
    double[][] Xp = new double[22][1];
    double[][] Xm = new double[22][1];
16    double[][] Z = new double[8][1];
    double[][] H;// = new double[8][22];
    double[][] B;//q = new double[3][3];
    double lam_s;
20    double lam_st;
    double lam_v;
    int miu = 0;
    double[] SigmaI = {0.001,0.001,0.001,
24        1E-4,1E-4,1E-4,
        1E-8,1E-8,1E-8,
        1E-10,1E-10,1E-10,
        1E-7,1E-7,1E-7};
28
    double SigmaA;
    double SigmaW;

32
    // elements of R - matrix of SNS
    // errors of SNS
    double SigmaRS;
36    double SigmadRS;
    double SigmaHS;
    double SigmaVS;
    double SigmadVS;
40    double SigmaBa;
    // correlated errors of SNS
    double SigmaRC;
    double SigmadRC;
44    double SigmaVC;
    double SigmadVC;
    double SigmaHC;
    double SigmadHC;
48

    RunISNS(){
        super();
        SigmaA = 1E-3;
52        SigmaW = 1E-6;

        lam_s = 4E-6; // m^-1
        lam_st = 5E-4; // c^-1
56        lam_v = 0.01; // c^-1 (0.0017 - 0.05)
        // elements of R - matrix of SNS
        // errors of SNS
        SigmaRS = 3; // m (1-3)
60        SigmadRS = 4; // m (1-4)
        SigmaHS = 4; // m (1.5-4)
        SigmaVS = 0.04; // m/s (0.001-0.05)
        SigmadVS = 0.2; // v/s (0.02-0.2)
64        SigmaBa = 10; // m
        // correlated errors of SNS
        SigmaRC = 7; // m (5-7)
        SigmadRC = 5; // m (2-5)
68        SigmaVC = 0.3; // m/s (0.02-0.3)
        SigmadVC = 0.02; // m/s (0.01-0.02)
        SigmaHC = 10; // m (7-10)
        SigmadHC = 7; // m (3-7)
72

76    }

    double[][] ComputeF(double phi,double[][] b,double h,double V_E,double V_N,double V_h, double a_E, double a_N
    , double a_h, double dt, double t){
80    double[][] F = Zeros(22);

    double R = FlightPathGen.EARTH_RADIUS + h;
    double Rz = FlightPathGen.EARTH_RADIUS;
84    double dlam1 = V_E/(R*(Math.cos(phi)));

```



```

double dphi1 = V_N/R;
double u = FlightPathGen.EARTH_SPEED;
double w_1E = -dphi1;
88 double w_1N = (u+dlam1)*Math.cos(phi);
double w_1h = (u+dlam1)*Math.sin(phi);
double e2 = FlightPathGen.E2;
double ge = FlightPathGen.GE;
92 double a = FlightPathGen.EARTH_RADIUS;

for(int i=0;i<22;i++)
    for(int j=0;j<22;j++)
96 F[i][j] = 0;

F[0][1] = (dlam1/Rz)*Math.tan(phi);
100 F[0][2] = -(dlam1*Rz)/R;
F[0][3] = Rz/(R*Math.cos(phi));

F[1][2] = -dphi1*(Rz/R);
104 F[1][4] = -Rz/R;

F[2][5]=1;

108 F[3][1] = ((2*u+dlam1)/Rz)*(V_h*Math.sin(phi)+V_N*Math.cos(phi))- (dlam1/Rz)*Math.tan(phi)*(V_h*Math.cos(phi)-V_N*Math.sin(phi));
F[3][2] = (V_E/(R*R))*(V_h-V_N*Math.tan(phi));
F[3][3] = -(V_h*Math.cos(phi)-V_N*Math.sin(phi))/(R*Math.cos(phi));
F[3][4] = (2*u+dlam1)*Math.sin(phi);
112 F[3][5] = -(2*u+dlam1)*Math.cos(phi);
F[3][7] = -a_h;
F[3][8] = a_N;
F[3][12] = b[0][0];
116 F[3][13] = b[0][1];
F[3][14] = b[0][2];

F[4][1] = -(2*u + dlam1)*V_E*Math.cos(phi)*(1/Rz) - ((V_E*V_E)*Math.pow(Math.tan(phi),2))/(Rz*R);
120 F[4][2] = ((Math.pow(V_E,2))*Math.tan(phi)+V_N*V_h)*Math.pow(R,-2);
F[4][3] = -2*(u+dlam1)*Math.sin(phi)-V_E*Math.tan(phi)*(1/R);
F[4][4] = -V_h/R;
F[4][5] = -dphi1;
124 F[4][6] = a_h;
F[4][8] = -a_E;
F[4][12] = b[1][0];
F[4][13] = b[1][1];
128 F[4][14] = b[1][2];

F[5][1] = - V_E*(2*u+dlam1)*Math.sin(phi)*(1/Rz)+ ((Math.pow(V_E,2))*Math.tan(phi))/(Rz*R)+ 1.5*(ge/Rz)*e2*Math.sin(phi)*Math.cos(phi);
F[5][2] = -(ge*2)/a - (Math.pow(V_E,2)+Math.pow(V_N,2))/(R*R);
132 F[5][3] = 2*(u+dlam1)*Math.cos(phi)+V_E/R;
F[5][4] = dphi1+V_N/R;
F[5][6] = -a_N;
F[5][7] = a_E;
136 F[5][12] = b[2][0];
F[5][13] = b[2][1];
F[5][14] = b[2][2];

140 F[6][4] = -(1/R);
F[6][7] = w_1h;
F[6][8] = -w_1N;
F[6][9] = -b[0][0];
144 F[6][10] = -b[0][1];
F[6][11] = -b[0][2];

F[7][1] = -u*Math.sin(phi)*(1/Rz);
148 F[7][3] = (1/R);
F[7][6] = -w_1h;
F[7][8] = w_1E;
F[7][9] = -b[1][0];
152 F[7][10] = -b[1][1];
F[7][11] = -b[1][2];

F[8][1] = ((u*Math.cos(phi)+dlam1/Math.cos(phi))*(1/Rz));
156 F[8][3] = (Math.tan(phi)/R);
F[8][6] = w_1N;
F[8][7] = -w_1E;
F[8][9] = -b[2][0];
160 F[8][10] = -b[2][1];
F[8][11] = -b[2][2];

// SNS part of matrix

```

```

164      double V_r = Math.sqrt(Math.pow(V_E, 2) + Math.pow(V_N, 2));
      double W_R = Math.exp(-(lam_s*V_r+lam_st)*dt);
      double W_V = Math.exp(-lam_v*t);

168      F[16][16] = W_R;
      F[17][17] = W_R;
      F[18][18] = W_R;
      F[19][19] = W_V;
172      F[20][20] = W_V;
      F[21][21] = W_V;

      // convert system from continuous to discrete time
176      for (int i=0; i<16; i++){
          for (int j=0; j<16; j++){
              F[i][j] = F[i][j]*dt;
              if (i==j)
180                  F[i][j] = F[i][j] + 1;
          }

184      //System.out.println("F = \n" + printDoubleArray(F));
      return F;
    }

188    double[][] ComputeB(double pitch, double yaw, double roll){
      double[][] b = new double[3][3];
      b[0][0] = Math.sin(yaw)*Math.cos(pitch);
      b[1][0] = Math.cos(yaw)*Math.cos(pitch);
192      b[2][0] = Math.sin(pitch);

      b[0][1] = Math.cos(yaw)*Math.sin(roll)-Math.sin(yaw)*Math.cos(roll)*Math.sin(pitch);
      b[1][1] = -Math.sin(yaw)*Math.sin(roll)-Math.cos(yaw)*Math.cos(roll)*Math.sin(pitch);
196      b[2][1] = Math.cos(roll)*Math.cos(pitch);

      b[0][2] = Math.cos(yaw)*Math.cos(roll)+Math.sin(yaw)*Math.sin(roll)*Math.sin(pitch);
      b[1][2] = -Math.sin(yaw)*Math.cos(roll)+Math.cos(yaw)*Math.sin(roll)*Math.sin(pitch);
200      b[2][2] = -Math.sin(roll)*Math.cos(pitch);
      // System.out.println("b = \n" + printDoubleArray(b));
      return b;
    }

204    double[][] ComputeQ(double[][] b, double phi, double t, double dt, double V_E, double V_N){

      double[][] Q = Zeros(22,34);
208      // INS 15x21
      for (int i=0; i<SigmaI.length; i++){
          {
              Q[i][i] = SigmaI[i];
212          }
          for (int i=0; i<3; i++){
              for (int j=0; j<3; j++){
                  Q[i+3][j+18] = b[i][j]*SigmaA;
216                  Q[i+6][j+15] = -b[i][j]*SigmaW;
              }
          }
          // Baro altimeter
220      Q[15][21] = SigmaBa*Math.sqrt(dt);
          // SNS 6x12
      double V_r = Math.sqrt(Math.pow(V_E, 2) + Math.pow(V_N, 2));
      double q_R = Math.sqrt(1-Math.exp((-2*(lam_s*V_r+lam_st))*dt));
224      double q_V = Math.sqrt(1-Math.exp(-2*lam_v*dt));

      Q[16][22] = q_R*SigmaRC/Math.cos(phi) ;
      Q[17][23] = q_R*SigmaRC ;
228      Q[18][24] = q_R*SigmaHC ;
      Q[19][25] = q_V*SigmaVC ;
      Q[20][26] = q_V*SigmaVC ;
      Q[21][27] = q_V*SigmaVC ;

232

      // System.out.println("Q = \n" + printDoubleArray(Q));
      return Q;
    }

236    double[][] ComputeH(){
      H = Zeros(8,22);
      H[0][2] = 1; H[0][15] = -1;
240      H[1][0] = 1; H[1][16] = -1;
      H[2][1] = 1; H[2][17] = -1;
      H[3][2] = 1; H[3][18] = -1;
      H[4][3] = 1; H[4][19] = -1;
244      H[5][4] = 1; H[5][20] = -1;

```

```

H[6][5] = 1; H[6][21]=-1;
H[7][15] = 1; H[7][18]=-1;
//System.out.println("H = \n" + printDoubleArray(H));
248 return H;
}
double[][] CouputeR(double phi){
double[][] R = Zeros(8,13);
252 R[0][0] = SigmaBa;
R[1][1] = SigmaRS/Math.cos(phi);
R[2][2] = SigmaRS;
R[3][3] = SigmaHS;
256 R[4][4] = SigmaVS;
R[5][5] = SigmaVS;
R[6][6] = SigmaVS;
R[7][0] = SigmaBa;R[7][3] = SigmaHS;
260 // System.out.println("R = \n" + printDoubleArray(R));
return R;
}
double[][] InitPp(){
264 // cov state of SDINS
//Pp_ins = [100 100 50 5 5 5 10^-3 10^-3 3*10^-3 ...
// 10^-5 10^-5 10^-5 10^-5 10^-3 10^-3 10^-3];
// cov state of baro altimeter
268 //Pp_ba = 10;
// cov state of SNS
//Pp_sns = [5 5 5 0.3 0.3 0.3];10^-
double[][] Pp;
272 double[] Pdiag = {Math.pow(100,2), Math.pow(100,2), Math.pow(50,2), Math.pow(5,2), Math.pow(5,2), Math.
pow(5,2),
Math.pow(1E-3,2), Math.pow(1E-3,2), Math.pow(3*1E-3,2), Math.pow(1E-5,2), Math.pow(1E-5,2), Math.
pow(1E-5,2),
Math.pow(1E-3,2), Math.pow(1E-3,2), Math.pow(1E-3,2),Math.pow(10,2),Math.pow(5,2), Math.pow(5,2),
Math.pow(5,2), Math.pow(0.3,2), Math.pow(0.3,2), Math.pow(0.3,2)};
Pp = Diag(Pdiag);
276 //System.out.println("Pp = \n" + printDoubleArray(Pp));
return Pp;
}
280 double[][] InitXp(){

double[][] Xp = {{120}, {-100}, {70}, {5}, {-5}, {1},
284 {1E-3}, {1E-3}, {3*1E-3}, {1E-4}, {-5*1E-4}, {7*1E-4},
{8*1E-2}, {-1E-2}, {-1E-2}, {15}, {2}, {5}, {7}, {0.01}, {0.02}, {0.01}};

//System.out.println("InitXp = \n" + printDoubleArray(Xp));
288 return Xp;
}
double[][] RandomKsi(int size){
292 double [][] Ksi = new double [size][1];
Random r = new Random();
for (int i=0; i< size; i++)
Ksi [i][0] = r.nextGaussian();
296 return Ksi;
}

public static double[][] Zeros(int n, int m){
300 double[][] z = new double[n][m];
for(int i=0;i<n;i++)
for(int j=0;j<m;j++)
z[i][j] = 0;
304 return z;
}
public static double[][] Zeros(int n){
double[][] z = new double[n][n];
308 for(int i=0;i<n;i++)
for(int j=0;j<n;j++)
z[i][j] = 0;
return z;
312 }
public static double[][] Diag (double[] diag){
double[][] matrix = Zeros(diag.length);
for(int i=0;i<diag.length;i++)
316 matrix[i][i] = diag[i];
return matrix;
}
320
}

```

# Графічний інтерфейс програми

Listing 3: SetDialog.java

```
package jKalman;
2
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
6
import javax.swing.Box;
import javax.swing.JButton;
import javax.swing.JDialog;
10import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
14import javax.swing.SwingConstants;

public class SetDialog extends JDialog {
    public SetDialog(JFrame parent) {
18        super(parent, "ISNS_SETTINGS", true);

        Box b = Box.createVerticalBox();
        b.add(Box.createGlue());
22        b.add(new JLabel("Settings_fro_INS"));

        b.add(Box.createGlue());

26        getContentPane().add(b, "Center");

        JPanel p2 = new JPanel();
        setLayout(new GridLayout(5, 4));
30

        JButton ok = new JButton("Ok");

        getContentPane().add(p2, "South");
34

        ok.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                setVisible(false);
38            }
        });

        setSize(600, 200);
42

        JTextField TfSigmaA = new JTextField(String.valueOf("1E-3"));
        JTextField TfSigmaW = new JTextField(String.valueOf("1E-6"));
        JTextField TfSigmaI1 = new JTextField(String.valueOf("0.001"));
46        JTextField TfSigmaI2 = new JTextField(String.valueOf("1E-4"));
        JTextField TfSigmaI3 = new JTextField(String.valueOf("1E-8"));
        JTextField TfSigmaI4 = new JTextField(String.valueOf("1E-10"));
        JTextField TfSigmaI5 = new JTextField(String.valueOf("1E-7"));
50        JTextField TfSigmaRS = new JTextField(String.valueOf("3"));
        JTextField TfSigmaHS = new JTextField(String.valueOf("4"));
        JTextField TfSigmaVS = new JTextField(String.valueOf("0.04"));
        JTextField TfBa = new JTextField(String.valueOf("10"));
54        JTextField TfSigmaRC = new JTextField(String.valueOf("7"));
        JTextField TfSigmaVC = new JTextField(String.valueOf("0.3"));
        JTextField TfSigmaHC = new JTextField(String.valueOf("10"));

58        JLabel LbSigmaA = new JLabel("SigmaA", SwingConstants.LEFT);
        JLabel LbSigmaW = new JLabel("SigmaW", SwingConstants.LEFT);
        JLabel LbSigmaI1 = new JLabel("SigmaI_1-3", SwingConstants.LEFT);
        JLabel LbSigmaI2 = new JLabel("SigmaI_4-6", SwingConstants.LEFT);
62        JLabel LbSigmaI3 = new JLabel("SigmaI_7-9", SwingConstants.LEFT);
        JLabel LbSigmaI4 = new JLabel("SigmaI_9-12", SwingConstants.LEFT);
        JLabel LbSigmaI5 = new JLabel("SigmaI_13-15", SwingConstants.LEFT);
        JLabel LbSigmaRS = new JLabel("SigmaRS", SwingConstants.LEFT);
66        JLabel LbSigmaHS = new JLabel("SigmaHS", SwingConstants.LEFT);
        JLabel LbSigmaVS = new JLabel("SigmaVS", SwingConstants.LEFT);
        JLabel LbBa = new JLabel("SigmaBa", SwingConstants.LEFT);
        JLabel LbSigmaRC = new JLabel("SigmaRC", SwingConstants.LEFT);
70        JLabel LbSigmaVC = new JLabel("SigmaVC", SwingConstants.LEFT);
        JLabel LbSigmaHC = new JLabel("SigmadHC", SwingConstants.LEFT);
    }
}
```

```

74    p2.add(LbSigmaA);    p2.add(TfSigmaA);
    p2.add(LbSigmaW);    p2.add(TfSigmaW);
    p2.add(LbSigmaI1);   p2.add(TfSigmaI1);
    p2.add(LbSigmaI2);   p2.add(TfSigmaI2);
78    p2.add(LbSigmaI3);   p2.add(TfSigmaI3);
    p2.add(LbSigmaI4);   p2.add(TfSigmaI4);
    p2.add(LbSigmaI5);   p2.add(TfSigmaI5);
    p2.add(LbSigmaRS);   p2.add(TfSigmaRS);
82    p2.add(LbSigmaHS);   p2.add(TfSigmaHS);
    p2.add(LbSigmaVS);   p2.add(TfSigmaVS);
    p2.add(LbBa);        p2.add(TfBa);
    p2.add(LbSigmaRC);   p2.add(TfSigmaRC);
86    p2.add(LbSigmaVC);   p2.add(TfSigmaVC);
    p2.add(LbSigmaHC);   p2.add(TfSigmaHC);
    //p2.add(TextfieldKPhi);    add(label_KPhi);
    // p2.add(TextfieldKLam);    add(label_KLam);
90
    p2.add(ok);

94
}

}

```

## Listing 4: InputsPanel.java

```

package jKalman;

3import java.awt.*;
import javax.swing.*;

/** Panel to hold input textfields. */
7public class InputsPanel extends JPanel {
    JTextField fTextfieldTop;

    /** Constructor builds panel with labels and text fields. */
11    InputsPanel(String label_str_top, String init_top) {

        // Set the layout with 2 rows by 2 columns
        setLayout(new GridLayout(1, 2));

15        // Create two text fields with the initial values
        fTextfieldTop = new JTextField(init_top);
        // fTextfieldTop.setSize(300, 20);

19        // Create the first label and right justify the text
        JLabel label_top = new JLabel(label_str_top, SwingConstants.RIGHT);

23        // Insert the label and textfield into the top grid row
        add(label_top);
        add(fTextfieldTop);

27    } // ctor

} // class InputsPanel

```

## Listing 5: FlightPropPanel.java

```

package jKalman;

3import java.awt.*;

import javax.swing.*;

7public class FlightPropPanel extends JPanel {

    JTextField TextfieldTime;
    JTextField TextfieldStepTme;
11    JTextField TextfieldPhi0;
    JTextField TextfieldLam0;
    JTextField TextfieldH0;

15    JTextField TextfieldTPhi;

```

```

        JTextField TextfieldTLam;
        JTextField TextfieldTH;
19
        JTextField TextfieldDeltaPhi;
        JTextField TextfieldDeltaLam;
        JTextField TextfieldDeltaH;
23
        JTextField TextfieldDdeltaPhi;
        JTextField TextfieldDdeltaLam;
        JTextField TextfieldDdeltaH;
27
        JTextField TextfieldKPhi;
        JTextField TextfieldKLam;
        JButton ButtonCalc;
31

35  /** Constructor builds panel with labels and text fields. */
    FlightPropPanel(double [] ParamData) {

39
        TextfieldTime = new JTextField(String.valueOf(ParamData[0]));
        TextfieldStepTme = new JTextField(String.valueOf(ParamData[1]));

43
        TextfieldPhi0 = new JTextField(String.valueOf(ParamData[2]));
        TextfieldLam0 = new JTextField(String.valueOf(ParamData[3]));
        TextfieldH0 = new JTextField(String.valueOf(ParamData[4]));

47
        TextfieldTPhi = new JTextField(String.valueOf(ParamData[5]));
        TextfieldTLam = new JTextField(String.valueOf(ParamData[6]));
        TextfieldTH = new JTextField(String.valueOf(ParamData[7]));

51
        TextfieldDeltaPhi = new JTextField(String.valueOf(ParamData[8]));
        TextfieldDeltaLam = new JTextField(String.valueOf(ParamData[9]));
        TextfieldDeltaH = new JTextField(String.valueOf(ParamData[10]));

55
        TextfieldDdeltaPhi = new JTextField(String.valueOf(ParamData[11]));
        TextfieldDdeltaLam = new JTextField(String.valueOf(ParamData[12]));
        TextfieldDdeltaH = new JTextField(String.valueOf(ParamData[13]));

59
        TextfieldKPhi = new JTextField(String.valueOf(ParamData[14]));
        TextfieldKLam = new JTextField(String.valueOf(ParamData[15]));

        JLabel label_Time = new JLabel("time,s", SwingConstants.LEFT);
63        JLabel label_StepTme = new JLabel("StepTme,s", SwingConstants.LEFT);

        JLabel label_Phi0 = new JLabel("Phi0,DEG", SwingConstants.LEFT);
        JLabel label_Lam0 = new JLabel("Lam0,DEG", SwingConstants.LEFT);
67        JLabel label_H0 = new JLabel("H0,m", SwingConstants.LEFT);

        JLabel label_TPhi = new JLabel("TPhi", SwingConstants.LEFT);
        JLabel label_TLam = new JLabel("TLam", SwingConstants.LEFT);
71        JLabel label_TH = new JLabel("TH", SwingConstants.LEFT);

        JLabel label_DeltaPhi = new JLabel("DeltaPhi", SwingConstants.LEFT);
        JLabel label_DeltaLam = new JLabel("DeltaLam", SwingConstants.LEFT);
75        JLabel label_DeltaH = new JLabel("DeltaH", SwingConstants.LEFT);

        JLabel label_DdeltaPhi = new JLabel("DdeltaPhi", SwingConstants.LEFT);
        JLabel label_DdeltaLam = new JLabel("DdeltaLam", SwingConstants.LEFT);
79        JLabel label_DdeltaH = new JLabel("DdeltaH", SwingConstants.LEFT);

        JLabel label_KPhi = new JLabel("KPhi", SwingConstants.LEFT);
        JLabel label_KLam = new JLabel("KLam", SwingConstants.LEFT);
83        // Set the layout with 16 rows by 2 columns
        setLayout(new GridLayout(30, 2));

        //ButtonCalc = new JButton("Calculate");
87

91

        // Create the first label and right justify the text
95        //JLabel label_top = new JLabel(label_str_top, SwingConstants.LEFT);

        // Insert the label and textfield into the top grid row

```

```

    add(TextfieldTime); add(label_Time);
99    add(TextfieldStepTme); add(label_StepTme);
    add(TextfieldPhi0); add(label_Phi0);
    add(TextfieldLam0); add(label_Lam0);
    add(TextfieldH0); add(label_H0);
103    add(TextfieldTPhi); add(label_TPhi);
    add(TextfieldTLam); add(label_TLam);
    add(TextfieldTH); add(label_TH);
    add(TextfieldDeltaPhi); add(label_DeltaPhi);
107    add(TextfieldDeltaLam); add(label_DeltaLam);
    add(TextfieldDeltaH); add(label_DeltaH);
    add(TextfieldDdeltaPhi); add(label_DdeltaPhi);
    add(TextfieldDdeltaLam); add(label_DdeltaLam);
111    add(TextfieldDdeltaH); add(label_DdeltaH);
    add(TextfieldKPhi); add(label_KPhi);
    add(TextfieldKLam); add(label_KLam);
    //add(ButtonCalc);
115

    }
119 } // class InputsPanel

```

## Точка входу в програму

### Listing 6: jRKalman.java

```

package jKalman;

import static org.math.array.DoubleArray.identity;
4import static org.math.array.DoubleArray.transpose;
import static org.math.array.LinearAlgebra.inverse;
import static org.math.array.LinearAlgebra.minus;
import static org.math.array.LinearAlgebra.plus;
8import static org.math.array.LinearAlgebra.times;

import static org.math.io.files.ASCIIFile.*;
import static org.math.io.parser.ArrayString.*;
12
import static org.math.array.DoubleArray.*;
import static org.math.array.LinearAlgebra.*;

16import javax.swing.*;

import java.io.*;
import org.math.plot.Plot2DPanel;
20import org.math.plot.Plot3DPanel;

import java.awt.*;
import java.awt.event.*;
24
/*
 * String str1 = fInputsPanel.fTextfieldTop.getText ();
 * val1 = Double.parseDouble (str1).getText ();
28 */
/**
 *
 * @author Mykola Novik java -Dsun.java2d.pmoffscreen=false
32 */
// implements ActionListener
public class jRKalman {
    final static boolean shouldFill = true;
36    final static boolean shouldWeightX = true;
    final static boolean RIGHT_TO_LEFT = false;

    // time variables
40    static double simTime = 1000;
    static double stepTime = 1;
    static double[] t = FlightPathGen.loadTimeArray(simTime, stepTime);
    // Start coordinates
44    static double Phi0 = 55 * (Math.PI / 180);
    static double Lam0 = 30 * (Math.PI / 180);
    static double H0 = 5000;
    // Period of trajectory
48    static double TPhi = 600;

```

```

static double TLam = 600;
static double TH = 600;
// linear part of eq
52 static double kPhi = 0;
static double kLam = 0.00004;
// delta Phi
static double deltaPhi = 0.004;
56 static double deltaLam = 0;
static double deltaH = H0;
// delta phase
static double ddeltaPhi = 0;
60 static double ddeltaLam = 0;
static double ddeltaH = 0;

static double WPhi = 2 * Math.PI / TPhi;
64 static double WLam = 2 * Math.PI / TLam;
static double WH = 2 * Math.PI / TH;

// store Data
68 static double[] Phi;
static double[] Lam;
static double[] H;
static double[] Pitch;
72 static double[] Yaw;
static double[] Roll;
static double[] VE;
static double[] VN;
76 static double[] VH;
static double[] AE;
static double[] AN;
static double[] AH;
80 //

FlightPathGen FP;
static FlightPropPanel FPset;
84 // ////////////////////////////////////////
// GUI
static JFrame frame;
// PLOTS
88 static Plot2DPanel plot11;
static Plot2DPanel plot12;
static Plot2DPanel plot13;

92 static Plot2DPanel plot21;
static Plot2DPanel plot22;
static Plot2DPanel plot23;

96 static Plot2DPanel plot31;
static Plot2DPanel plot32;
static Plot2DPanel plot33;

100 static Plot2DPanel plot41;
static Plot2DPanel plot42;
static Plot2DPanel plot43;

104 static Plot2DPanel plot51;
static Plot2DPanel plot52;
static Plot2DPanel plot53;

108 static Plot3DPanel path3d;
static Plot2DPanel pathPitch;
static Plot2DPanel pathYaw;
static Plot2DPanel pathRoll;
112

static double[][] XP;
static double[][] XR;
static double[][] XD;
116

static public void InitPlots() {
// Plots for TAB 1
plot11 = new Plot2DPanel();
120 plot11.addLegend("SOUTH");
plot12 = new Plot2DPanel();
plot12.addLegend("SOUTH");
plot13 = new Plot2DPanel();
124 plot13.addLegend("SOUTH");

plot21 = new Plot2DPanel();
plot21.addLegend("SOUTH");
128 plot22 = new Plot2DPanel();
plot22.addLegend("SOUTH");

```



```

        plot23 = new Plot2DPanel();
        plot23.addLegend("SOUTH");
132
        plot31 = new Plot2DPanel();
        plot31.addLegend("SOUTH");
        plot32 = new Plot2DPanel();
136        plot32.addLegend("SOUTH");
        plot33 = new Plot2DPanel();
        plot33.addLegend("SOUTH");

        plot41 = new Plot2DPanel();
        plot41.addLegend("SOUTH");
        plot42 = new Plot2DPanel();
        plot42.addLegend("SOUTH");
144        plot43 = new Plot2DPanel();
        plot43.addLegend("SOUTH");

        plot51 = new Plot2DPanel();
148        plot51.addLegend("SOUTH");
        plot52 = new Plot2DPanel();
        plot52.addLegend("SOUTH");
        plot53 = new Plot2DPanel();
152        plot53.addLegend("SOUTH");

        pathPitch = new Plot2DPanel();
        pathPitch.addLegend("SOUTH");
156        pathPitch.setAxisLabel(0, "t, °s");
        pathPitch.setAxisLabel(1, "Pitch, °DEG");

        pathYaw = new Plot2DPanel();
160        pathYaw.addLegend("SOUTH");
        pathYaw.setAxisLabel(0, "t, °s");
        pathYaw.setAxisLabel(1, "Yaw, °DEG");

        pathRoll = new Plot2DPanel();
164        pathRoll.addLegend("SOUTH");
        pathRoll.setAxisLabel(0, "t, °s");
        pathRoll.setAxisLabel(1, "Roll, °DEG");
168

        // 3D PLOT
        path3d = new Plot3DPanel();
        path3d.addLegend("SOUTH");
172        path3d.setAxisLabel(0, "Phi, °DEG");
        path3d.setAxisLabel(1, "Lam, °DEG");
        path3d.setAxisLabel(2, "H, m");
    }
176
    static public void ClearAllPlots()
    {
        plot11.removeAllPlots();
        plot12.removeAllPlots();
180        plot13.removeAllPlots();
        plot21.removeAllPlots();
        plot22.removeAllPlots();
184        plot23.removeAllPlots();
        plot31.removeAllPlots();
        plot32.removeAllPlots();
        plot33.removeAllPlots();
188        plot41.removeAllPlots();
        plot42.removeAllPlots();
        plot43.removeAllPlots();
        plot51.removeAllPlots();
192        plot52.removeAllPlots();
        plot53.removeAllPlots();
        pathPitch.removeAllPlots();
        pathYaw.removeAllPlots();
196        pathRoll.removeAllPlots();
        // 3D PLOT
        path3d.removeAllPlots();
200
    }

    static protected JPanel createInnerPanel31(String text) {
        JPanel jplPanel = new JPanel();
204        // JLabel jlbDisplay = new JLabel(text);
        // jlbDisplay.setHorizontalAlignment(JLabel.CENTER);
        jplPanel.setLayout(new GridLayout(3, 1));
        // jplPanel.add(jlbDisplay);
208        return jplPanel;
    }

```

```

212     static protected JPanel createInnerPanel11(String text) {
        JPanel jplPanel = new JPanel();

        // JLabel jlbDisplay = new JLabel(text);
        // jlbDisplay.setHorizontalAlignment(JLabel.CENTER);
216     jplPanel.setLayout(new GridLayout(1, 1));
        // jplPanel.add(jlbDisplay);
        return jplPanel;
    }

220     public static void addComponentsToPane(Container pane) {
        if (RIGHT_TO_LEFT) {
            pane.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);
224        }
        // Icons set
        ImageIcon icon1 = new ImageIcon("pics/1.png");
        ImageIcon icon2 = new ImageIcon("pics/2.png");
228        ImageIcon icon3 = new ImageIcon("pics/3.png");
        ImageIcon icon4 = new ImageIcon("pics/4.png");
        ImageIcon icon5 = new ImageIcon("pics/5.png");

232        // Math Plot 1
        // double[] x1 = { 1, 2, 3, 4, 5, 6, 7, 8 };
        // double[] y1 = { 1, 4, 9, 16, 25, 36, 49, 64 };

236        InitPlots();

        // Create Tabs
        JTabbedPane TabPanel = new JTabbedPane();

240        JPanel jplInnerPanel1 = createInnerPanel31("");
        TabPanel.addTab("R", icon1, jplInnerPanel1, "Tab_1");
        jplInnerPanel1.add(plot11);
244        jplInnerPanel1.add(plot12);
        jplInnerPanel1.add(plot13);

        JPanel jplInnerPanel2 = createInnerPanel31("");
248        TabPanel.addTab("V", icon2, jplInnerPanel2);
        jplInnerPanel2.add(plot21);
        jplInnerPanel2.add(plot22);
        jplInnerPanel2.add(plot23);

252        JPanel jplInnerPanel3 = createInnerPanel31("");
        TabPanel.addTab("Alpha", icon3, jplInnerPanel3, "Tab_3");
        jplInnerPanel3.add(plot31);
256        jplInnerPanel3.add(plot32);
        jplInnerPanel3.add(plot33);

        JPanel jplInnerPanel4 = createInnerPanel31("");
260        TabPanel.addTab("Gyro", icon4, jplInnerPanel4);
        jplInnerPanel4.add(plot41);
        jplInnerPanel4.add(plot42);
        jplInnerPanel4.add(plot43);

264        JPanel jplInnerPanel5 = createInnerPanel31("");
        TabPanel.addTab("Acc", icon5, jplInnerPanel5);
        jplInnerPanel5.add(plot51);
268        jplInnerPanel5.add(plot52);
        jplInnerPanel5.add(plot53);

        JPanel jplInnerPanel6 = createInnerPanel11("");
272        TabPanel.addTab("3D-Path", icon5, jplInnerPanel6);
        // TabPanel.setSelectedIndex(0);
        jplInnerPanel6.add(path3d);

276        JPanel jplInnerPanel7 = createInnerPanel31("");
        TabPanel.addTab("Pitch/Yaw/Roll", icon5, jplInnerPanel7);

        jplInnerPanel7.add(pathPitch);
280        jplInnerPanel7.add(pathYaw);
        jplInnerPanel7.add(pathRoll);

        // Create menu
284        JMenuBar mainMenuBar = new JMenuBar();
        JMenu menuFile = new JMenu("File");
        JMenuItem pTextMenuItem1 = new JMenuItem("Compute_Flight_Path",
            KeyEvent.VK_T);
288        JMenuItem FileMenuItemOpen = new JMenuItem("Open_Data", KeyEvent.VK_T);
        JMenuItem FileMenuItemSave = new JMenuItem("Save_Data", KeyEvent.VK_T);
        JMenuItem FileMenuItemQuit = new JMenuItem("Quit", KeyEvent.VK_T);
        pTextMenuItem1.addMouseListener(new MouseAdapter() {

```

```

292         public void mousePressed(MouseEvent me) {
                GenerateTrajectory();

                };
296     });

    FileMenuItemQuit.addMouseListener(new MouseAdapter() {
300         public void mousePressed(MouseEvent me) {
                System.exit(0);
            };
        });
304    FileMenuItemOpen.addMouseListener(new MouseAdapter() {
        public void mousePressed(MouseEvent me) {
            //JFileChooser fc = new JFileChooser();
            //fc.showOpenDialog(frame);
308            //fc.setVisible(true);
            SetDialog f = new SetDialog(frame);
            f.show();
        };
    });
312    FileMenuItemSave.addMouseListener(new MouseAdapter() {
        public void mousePressed(MouseEvent me) {
            JFileChooser fc = new JFileChooser();
316            fc.showSaveDialog(frame);
            fc.setVisible(true);
        };
    });

320    // pTextMenuItem3.addActionListener();

    menuFile.add(pTextMenuItem1);
324    menuFile.add(FileMenuItemOpen);
    menuFile.add(FileMenuItemSave);
    menuFile.add(FileMenuItemQuit);

328    JMenu menuView = new JMenu("View");
    JMenu menuHelp = new JMenu("Help?");
    mainMenuBar.add(menuFile);
    mainMenuBar.add(menuView);
332    mainMenuBar.add(menuHelp);

    // JButton button;
    JButton CalcButton = new JButton("FIRE!");
336    CalcButton.addMouseListener(new MouseAdapter() {
        public void mousePressed(MouseEvent me) {
            GenerateTrajectory();

            };
340    });
    JButton ClearButton = new JButton("CLEAR_ALL!");
    ClearButton.addMouseListener(new MouseAdapter() {
344        public void mousePressed(MouseEvent me) {
            ClearAllPlots();

        };
348    });

352    pane.setLayout(new GridBagLayout());
    GridBagConstraints c = new GridBagConstraints();

    // c.fill = GridBagConstraints.HORIZONTAL;
356    if (shouldFill) {
        // natural height, maximum width
        c.fill = GridBagConstraints.BOTH;
360        // c.fill = GridBagConstraints.VERTICAL;
    }
    // button = new JButton("Button 1");
    if (shouldWeightX) {
364        c.weightx = 0.3;
    }

    c.fill = GridBagConstraints.HORIZONTAL;
368    c.gridx = 0;
    c.gridy = 0;
    c.gridwidth = 2;
    c.weightx = 0;
372    pane.add(mainMenuBar, c);

```

```

// Input Panel
double[] PropData = { simTime, stepTime, Phi0, Lam0, H0, TPhi, TLam,
376     TH, deltaPhi, deltaLam, deltaH, ddeltaPhi, ddeltaLam, ddeltaH,
        kPhi, kLam };
FPset = new FlightPropPanel(PropData);
// InputsPanel fInputsPanel1 = new InputsPanel("time, sec ", "100");
380 c.fill = GridBagConstraints.HORIZONTAL;
c.weightx = 0;
c.gridx = 0;
c.gridy = 1;
384 c.gridwidth = 1;
pane.add(FPset, c);

InputsPanel fInputsPanel2 = new InputsPanel("Step, _sec_", "1");
388 // c.fill = GridBagConstraints.HORIZONTAL;
c.weightx = 0;
c.gridx = 0;
c.gridy = 1;
392 pane.add(fInputsPanel2, c);

// c.fill = GridBagConstraints.HORIZONTAL;
c.fill = GridBagConstraints.BOTH;
396 c.gridx = 1;
c.gridy = 1;
// c.gridwidth = 1;
c.gridheight = 2;
400 c.weightx = 1;
c.weighty = 1;
pane.add(TabPanel, c);

404 c.fill = GridBagConstraints.NONE;
c.anchor = GridBagConstraints.FIRST_LINE_START;
c.weightx = 0;
c.gridx = 0;
408 c.gridy = 2;
pane.add(CalcButton, c);

c.fill = GridBagConstraints.NONE;
412 c.anchor = GridBagConstraints.FIRST_LINE_START;
c.weightx = 0;
c.gridx = 1;
c.gridy = 3;
416 pane.add(ClearButton, c);

}

420

public static void PlotKalmanXP() {

424     double[] DRE = CopyRowArr(XP, 0, t.length);
    double[] DRN = CopyRowArr(XP, 1, t.length);
    double[] DRH = CopyRowArr(XR, 2, t.length);
    plot11.addLinePlot("RE", t,DRE);
    plot12.addLinePlot("RN", t,DRN);
    plot13.addLinePlot("RH", t,DRH);

428

    double[] DVE = CopyRowArr(XP, 3, t.length);
    double[] DVN = CopyRowArr(XP, 4, t.length);
    double[] DVH = CopyRowArr(XP, 5, t.length);
    plot21.addLinePlot("dVE", t,DVE);
    plot22.addLinePlot("dVN", t,DVN);
    plot23.addLinePlot("dVH", t,DVH);

432

    double[] AIE = CopyRowArr(XP, 6, t.length);
    double[] AIN = CopyRowArr(XP, 7, t.length);
    double[] AIH = CopyRowArr(XP, 8, t.length);
    plot31.addLinePlot("AIE", t,AIE);
    plot32.addLinePlot("AIN", t,AIN);
    plot33.addLinePlot("AIH", t,AIH);

440

    double[] GyE = CopyRowArr(XP, 9, t.length);
    double[] GyN =CopyRowArr(XP, 10, t.length);
    double[] GyH = CopyRowArr(XP, 11,t.length);
    plot41.addLinePlot("GyE", t,GyE);
    plot42.addLinePlot("GyN", t,GyN);
    plot43.addLinePlot("GyH", t,GyH);

444

    double[] AccE = CopyRowArr(XP, 12, t.length);

```

```

double[] AccN =CopyRowArr(XP, 13, t.length);
double[] AccH = CopyRowArr(XP, 14,t.length);
456 plot51.addLinePlot("AccE", t,AccE);
plot52.addLinePlot("AccN", t,AccN);
plot53.addLinePlot("AccH", t,AccH);

460
// double[][] Xdiff = minus(XP,XR);
// System.out.println("RE = \n" + printDoubleArray(RE));
// double[] RE = CopyRowArr(Xdiff, 1, FP.t.length);
464
}
public static void PlotKalmanXR() {

468 double[] DRE = CopyRowArr(XR, 0, t.length);
double[] DRN = CopyRowArr(XR, 1, t.length);
double[] DRH = CopyRowArr(XR, 2, t.length);
plot11.addLinePlot("RE", t,DRE);
472 plot12.addLinePlot("RN", t,DRN);
plot13.addLinePlot("RH", t,DRH);

double[] DVE = CopyRowArr(XR, 3, t.length);
476 double[] DVN = CopyRowArr(XR, 4, t.length);
double[] DVH = CopyRowArr(XR, 5, t.length);
plot21.addLinePlot("dVE", t,DVE);
plot22.addLinePlot("dVN", t,DVN);
480 plot23.addLinePlot("dVH", t,DVH);

double[] AIE = CopyRowArr(XR, 6, t.length);
double[] AIN = CopyRowArr(XR, 7, t.length);
484 double[] AIH = CopyRowArr(XR, 8, t.length);
plot31.addLinePlot("AIE", t,AIE);
plot32.addLinePlot("AIN", t,AIN);
plot33.addLinePlot("AIH", t,AIH);
488

double[] GyE = CopyRowArr(XR, 9, t.length);
double[] GyN =CopyRowArr(XR, 10, t.length);
492 double[] GyH = CopyRowArr(XR, 11,t.length);
plot41.addLinePlot("GyE", t,GyE);
plot42.addLinePlot("GyN", t,GyN);
plot43.addLinePlot("GyH", t,GyH);
496

double[] AccE = CopyRowArr(XR, 12, t.length);
double[] AccN =CopyRowArr(XR, 13, t.length);
double[] AccH = CopyRowArr(XR, 14, t.length);
500 plot51.addLinePlot("AccE", t,AccE);
plot52.addLinePlot("AccN", t,AccN);
plot53.addLinePlot("AccH", t,AccH);

504
// double[][] Xdiff = minus(XP,XR);
// System.out.println("RE = \n" + printDoubleArray(RE));
// double[] RE = CopyRowArr(Xdiff, 1, FP.t.length);
508
}
public static void PlotKalmanXD() {

512 double[] DRE = CopyRowArr(XD, 0, t.length);
double[] DRN = CopyRowArr(XD, 1, t.length);
double[] DRH = CopyRowArr(XD, 2, t.length);
plot11.addLinePlot("RE", t,DRE);
516 plot12.addLinePlot("RN", t,DRN);
plot13.addLinePlot("RH", t,DRH);

double[] DVE = CopyRowArr(XD, 3, t.length);
520 double[] DVN = CopyRowArr(XD, 4, t.length);
double[] DVH = CopyRowArr(XD, 5, t.length);
plot21.addLinePlot("dVE", t,DVE);
plot22.addLinePlot("dVN", t,DVN);
524 plot23.addLinePlot("dVH", t,DVH);

double[] AIE = CopyRowArr(XD, 6, t.length);
double[] AIN = CopyRowArr(XD, 7, t.length);
528 double[] AIH = CopyRowArr(XD, 8, t.length);
plot31.addLinePlot("AIE", t,AIE);
plot32.addLinePlot("AIN", t,AIN);
plot33.addLinePlot("AIH", t,AIH);
532

double[] GyE = CopyRowArr(XD, 9, t.length);

```

```

536     double[] GyN =CopyRowArr(XD, 10, t.length);
    double[] GyH = CopyRowArr(XD, 11,t.length);
    plot41.addLinePlot("GyE", t,GyE);
    plot42.addLinePlot("GyN", t,GyN);
    plot43.addLinePlot("GyH", t,GyH);

540
    double[] AccE = CopyRowArr(XD, 12, t.length);
    double[] AccN =CopyRowArr(XD, 13, t.length);
    double[] AccH = CopyRowArr(XD, 14,t.length);
544    plot51.addLinePlot("AccE", t,AccE);
    plot52.addLinePlot("AccN", t,AccN);
    plot53.addLinePlot("AccH", t,AccH);

548
    // double[][] Xdiff = minus(XP,XR);
    // System.out.println("RE = \n" + printDoubleArray(RE));
    // double[] RE = CopyRowArr(Xdiff, 1, FP.t.length);

552
}
public static void CalcKalman() {

556    RunISNS Sys = new RunISNS();

    double[][] Z;
    double[][] K;
560    double[][] B;// = Sys.ComputeB(Pitch[0], Yaw[0], Roll[0]);
    double[][] F;// = Sys.ComputeF(Phi[0], B,H[0], VE[0],VN[0],VH[0],
        // AE[0],AN[0],AH[0],stepTime,t[0]);
    double[][] Q;// =
564    // Sys.ComputeQ(B,FP.Phi[0],FP.t[0],FP.stepTime,FP.VE[0],FP.VN[0]);
    double[][] Ho;// = Sys.ComputeH();
    double[][] R;// = Sys.CouputeR(FP.Phi[0]);
    double[][] Pm;
568    double[][] Pp = Sys.InitPp();

    double[][] Xm;
    XP = new double[22][t.length];
572    XR = new double[22][t.length];

    // double[][] Ksi = Sys.RandomKsi(34);
    double[][] Xp = Sys.InitXp();
576    double[][] Xt = Sys.InitXp();
    int k = 1;

    XR = PasteCol(XR, 0, Xt);
580    XP = PasteCol(XP, 0, Xp);
    Ho = Sys.ComputeH();

    for (k = 1; k < t.length; k++) {
584        B = Sys.ComputeB(Pitch[k - 1], Yaw[k - 1], Roll[k - 1]);
        F = Sys.ComputeF(Phi[k - 1], B, H[k - 1], VE[k - 1], VN[k - 1],
            VH[k - 1], AE[k - 1], AN[k - 1], AH[k - 1], stepTime,
            t[k - 1]);
588
        Q = Sys.ComputeQ(B, Phi[k - 1], t[k - 1], stepTime, VE[k - 1],
            VN[k - 1]);
        R = Sys.CouputeR(Phi[k - 1]);
        Xt = plus(times(F, Xt), times(Q, Sys.RandomKsi(34)));
592        // Xt = times(F,Xt);
        XR = PasteCol(XR, k, Xt);

596
        Xm = times(F, Xp);

        Pm = plus(times(times(F, Pp), transpose(F)), times(Q, transpose(Q)));

600
        Z = plus(times(Ho, Xt), times(R, Sys.RandomKsi(13)));

        K = times(times(Pm, transpose(Ho)), inverse(plus(times(times(Ho, Pm), transpose(Ho)), times(R,
            transpose(R)))));
604

        Pp = plus(times(times(minus(identity(22), times(K, Ho)), Pm),
            transpose(minus(identity(22), times(K, Ho)))),
            times(times(times(K, R), transpose(R)), transpose(K)));
608        Xp = plus(Xm, times(K, minus(Z, times(Ho, Xm))));
        XP = PasteCol(XP, k, Xp);

612    }
    XD = minus(XP,XR);
}

```

```

616     public static void GenerateTrajectory() {

        ReadFlightPathForms();
        InitFlightPath();
620     //ClearAllPlots();
        path3d.addLinePlot("3DLine", Rad2Deg(Phi), Rad2Deg(Lam), H);
        pathPitch.addLinePlot("Pitch", t, Rad2Deg(Pitch));
        pathYaw.addLinePlot("Yaw", t, Rad2Deg(Yaw));
624     pathRoll.addLinePlot("Roll", t, Rad2Deg(Roll));

        CalcKalman();
        //PlotKalmanXP();
628     PlotKalmanXD();

632 }

    public static void ReadFlightPathForms() {
        try {
636             simTime = Double.parseDouble(FPset.TextfieldTime.getText());
            stepTime = Double.parseDouble(FPset.TextfieldStepTme.getText());
            Phi0 = Double.parseDouble(FPset.TextfieldPhi0.getText());
            Lam0 = Double.parseDouble(FPset.TextfieldLam0.getText());
640             H0 = Double.parseDouble(FPset.TextfieldH0.getText());
            TPhi = Double.parseDouble(FPset.TextfieldTPhi.getText());
            TLam = Double.parseDouble(FPset.TextfieldTLam.getText());
            TH = Double.parseDouble(FPset.TextfieldTH.getText());

644             deltaPhi = Double.parseDouble(FPset.TextfieldDeltaPhi.getText());
            deltaLam = Double.parseDouble(FPset.TextfieldDeltaLam.getText());
            deltaH = Double.parseDouble(FPset.TextfieldDeltaH.getText());
648             ddeltaPhi = Double.parseDouble(FPset.TextfieldDdeltaPhi.getText());
            ddeltaLam = Double.parseDouble(FPset.TextfieldDdeltaLam.getText());
            ddeltaH = Double.parseDouble(FPset.TextfieldDdeltaH.getText());
652             kPhi = Double.parseDouble(FPset.TextfieldKPhi.getText());
            kLam = Double.parseDouble(FPset.TextfieldKLam.getText());
            WPhi = 2 * Math.PI / TPhi;
            WLam = 2 * Math.PI / TLam;
            WH = 2 * Math.PI / TH;

656         } catch (NumberFormatException nfe) {
            System.out.println("Improper input");
            JOptionPane.showMessageDialog(null, "FIRE!");
660         }

    }

664     public static void InitFlightPath() {

        FlightPathGen FP = new FlightPathGen();

668         FP.simTime = simTime;
        FP.stepTime = stepTime;
        FP.Phi0 = Phi0;
        FP.Lam0 = Lam0;
672         FP.H0 = H0;
        FP.TPhi = TPhi;
        FP.TLam = TLam;
        FP.TH = TH;
676         FP.kPhi = kPhi;
        FP.kLam = kLam;
        FP.deltaPhi = deltaPhi;
        FP.deltaLam = deltaLam;
680         FP.deltaH = deltaH;
        FP.ddeltaPhi = ddeltaPhi;
        FP.ddeltaLam = ddeltaLam;
        FP.ddeltaH = ddeltaH;
684         FP.WPhi = 2 * Math.PI / TPhi;
        FP.WLam = 2 * Math.PI / TLam;
        FP.WH = 2 * Math.PI / TH;
        FP.t = FlightPathGen.loadTimeArray(simTime, stepTime);

688         FP.InitPathVars();
        FP.ComputePhiLamH();
        FP.ComputeR1R2();
692         FP.ComputeVenhr();
        FP.ComputeAenh();
        FP.ComputePRY();

696         Phi = FP.Phi;

```

```

        Lam = FP.Lam;
        H = FP.H;
        Pitch = FP.Pitch;
700    Yaw = FP.Yaw;
        Roll = FP.Roll;
        VE = FP.VE;
        VN = FP.VN;
704    VH = FP.VH;
        AE = FP.AE;
        AN = FP.AN;
        AH = FP.AH;
708    t = FP.t;
    }

    static double[] Rad2Deg(double[] Rad) {
712        double[] Deg = new double[Rad.length];
        for (int i = 0; i < Rad.length; i++)
            Deg[i] = 180 * Rad[i] / Math.PI;
        return Deg;
716    }

    public static double[][] PasteCol(double[][] M, int Pos, double X[][]) {

720        for (int i = 0; i < 22; i++) {
            M[i][Pos] = X[i][0];
            // System.out.println("i= " + i);
        }
724        return M;
    }

    public static double[] CopyRowArr(double[][] M, int Pos, int len) {
728        double[] array = new double[len];
        for (int i = 0; i < len; i++) {
            array[i] = M[Pos][i];

732            // System.out.println("i= " + i);
        }
        return array;
    }
736

    static double Deg2Rad(double Deg) {
        return Math.PI * Deg / 180;
    }
740

    private static void createAndShowGUI() {
        // Create and set up the window.
        frame = new JFrame("jKalman");
744        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Set up the content pane.
        addComponentsToPane(frame.getContentPane());
748

        // Display the window.
        frame.pack();
        frame.setSize(1000, 700);
752        frame.setVisible(true);
    }

    public static void main(String[] args) {
756        // Schedule a job for the event-dispatching thread:
        // creating and showing this application's GUI.
        javax.swing.SwingUtilities.invokeLater(new Runnable() {
            public void run() {
760                createAndShowGUI();
            }
        });
    }
764
}

```