

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Радіотехнічний факультет
Кафедра радіотехнічних пристроїв**

«На правах рукопису»
УДК 621.396.73

До захисту допущено:
Завідувач кафедри
_____ Сергій ЖУК
«__» _____ 2022р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою

«Радіотехнічні комп'ютеризовані системи»

зі спеціальності 172 «Телекомунікації та радіотехніка»

на тему: «Алгоритм адаптивного супроводження маневруючого БПЛА за даними РЛС з використанням вирішальних статистик відміток»

Виконав:
студент VI курсу, групи РС-11мп
Левченко Володимир Володимирович




Керівник:
к.т.н., доц.
Неуймін Олександр Станіславович



Рецензент:
к.т.н., доц. каф. ПРЕ
Адаменко Юлія Федорівна

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____ 

Київ – 2022 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Радіотехнічний факультет

Кафедра радіотехнічних систем

Рівень вищої освіти – другий (магістерський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Радіотехнічні комп'ютеризовані системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Сергій ЖУК

«__» _____ 2022р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Левченку Володимиру Володимировичу

1. Тема дисертації «Алгоритм адаптивного супроводження маневруючого БПЛА за даними РЛС з використанням вирішальних статистик відміток», науковий керівник дисертації Неуймін Олександр Станіславович, к.т.н., доц.,

затверджені наказом по університету від «15» листопада 2022 р. №3744-с

2. Термін подання студентом дисертації _____ 19 грудня 2022р.

3. Об'єкт дослідження Процес адаптивного супроводження маневруючого БПЛА за даними РЛС з використанням вирішальних статистик відміток

4. Вихідні дані технічні характеристики БПЛА, характеристики РЛС.

5. Перелік завдань, які потрібно розробити: Підготувати розділ по актуальності теми; Виконати огляд алгоритмів супроводження цілей; Розробити алгоритм адаптивного супроводження маневруючого БПЛА; Провести статистичне моделювання розробленого алгоритму; Виконати порівняльний аналіз алгоритмів об'єднання даних у складі алгоритму адаптивного супроводження маневруючого БПЛА; Оформити результати моделювання.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: презентація 13 слайдів

7. Орієнтовний перелік публікацій _____

8. Дата видачі завдання 5 вересня 2022р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз існуючого матеріалу	10.09.2022	
2	Аналіз актуальності обраної теми	21.09.2022	
3	Огляд алгоритмів супроводження цілей	5.10.2022	
4	Розробка алгоритму адаптивного супроводження маневруючого БПЛА	20.10.2022	
5	Статистичне моделювання розробленого алгоритму	10.11.2022	
6	Порівняльний аналіз алгоритмів об'єднання даних у складі алгоритму адаптивного супроводження маневруючого БПЛА	24.11.2022	
7	Оформлення результатів моделювання	1.12.2022	
8	Висновки	10.12.2022	

Студент

Володимир ЛЕВЧЕНКО

Науковий керівник

Олександр НЕУЙМІН

РЕФЕРАТ

Магістерська дисертація на тему «Алгоритм адаптивного супроводження маневруючого БПЛА за даними РЛС з використанням вирішальних статистик відміток» складається з 92 сторінок, що включають в себе 5 розділів, 26 ілюстрації, 13 таблиць, 33 бібліографічне найменування за переліком джерел посилань.

В роботі розроблено алгоритм адаптивного супроводження маневруючого БПЛА та проведено його статистичний аналіз.

Мета дослідження – підвищення ефективності супроводження маневруючого БПЛА у просторі.

Об’єкт дослідження – алгоритм адаптивного супроводження маневруючого БПЛА за даними РЛС з використанням вирішальних статистик відміток.

Предмет дослідження – методи супроводження та оцінки положення БПЛА за даними РЛС з використанням вирішальних статистик.

Синтезовано алгоритм адаптивного супроводження маневруючого БПЛА за даними РЛС з використанням вирішальних статистик відміток.

Результати дисертації рекомендуються для використання в установах, які займаються створенням і покращенням систем виявлення БПЛА для підвищення ефективності виявлення і супроводження цілей, що не потребують значних обчислювальних затрат.

Дана робота має важливе значення для підвищення обороноздатності країни. Передбачається, що за допомогою синтезованого алгоритму будуть створені системи для раннього виявлення об’єктів, забезпечення безпеки аеропортів та інших важливих промислових, військових та цивільних об’єктів.

Ключові слова: виявлення цілі, супроводження цілі, БПЛА, алгоритм, супроводження, РЛС, вирішальні статистики.

ABSTRACT

The master's dissertation on "Algorithm of adaptive tracking of the BPLA maneuver based on radar data using decisive statistical marks" consists of 92 pages, including 5 sections, 26 issues, 13 tables, 33 bibliographic names in the list of references.

In the work, the algorithm of the adaptive escort maneuver performed by the UAV was developed and a statistical analysis was carried out.

The aim of the research is to increase the efficiency of tracking a maneuvering UAV in space.

The object of the research is an algorithm of adaptive tracking by maneuvering means of UAVs for the use of decisive statistical marks.

The subject of the research is methods of conducting and assessing the position of UAVs based on radar data using decisive statistical data.

The algorithm of adaptive tracking of the maneuver performed by the UAV based on PC data using higher statistical outflows was synthesized.

The results of the dissertation are recommended for use in institutions that are engaged in the creation and improvement of the UAV detection system to increase the effectiveness of detecting and tracking targets that do not require significant computing costs.

This work is important for increasing the defense capability of the country. It is assumed that with the help of the synthesized algorithm, systems for early detection of objects, ensuring the security of airports and other important industrial, military and civilian objects will be created.

Keywords: target detection, target tracking, UAV, algorithm, tracking, radar, critical statistics.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	3
ВСТУП	4
1 АНАЛІЗ ІСНУЮСНУЮЧИХ АЛГОРИТМІВ СУПРОВОДЖЕННЯ ЦІЛІ ПО ДАНИМ РАДІОЛОКАТОРА	7
1.1 Огляд алгоритмів супроводження одиночної цілі.....	7
1.2 Огляд алгоритмів супроводження декількох цілей цілі	17
1.3 Супроводження одиночної цілі при наявності завад	28
2 РОЗРОБКА АЛГОРИТМУ АДАПТИВНОГО СУПРОВОДЖЕННЯ БПЛА ЗА ДАНИМИ РЛС З ВИКОРИСТАННЯМ ВИРІШАЛЬНИХ СТАТИСТИК ВІДМІТОК	31
2.1 Синтез алгоритму адаптивного оцінювання параметрів руху маневруючого МБПЛА.....	31
2.2 Розробка алгоритмів супроводження маневруючого МБПЛА з використанням вирішальних статистик відміток	40
3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АЛГОРИТМУ АДАПТИВНОГО СУПРОВОДЖЕННЯ БПЛА ЗА ДАНИМИ РЛС З ВИКОРИСТАННЯМ ВИРІШАЛЬНИХ СТАТИСТИК ВІДМІТОК.....	52
3.1 Розробка модуля генерації траєкторії з використанням waypointTrajectory	52
3.2 Реалізація адаптивного алгоритму супроводження з ймовірнісним об'єднанням даних на мові MATLAB.....	55
4 СТАТИСТИЧНИЙ АНАЛІЗ РОЗРОБЛЕНОГО АЛГОРИТМУ.....	66
4.1 Аналіз алгоритму адаптивного оцінювання параметрів руху маневруючого МБПЛА.....	66

4.2 Аналіз алгоритму супроводження маневруючого МБПЛА з використанням вирішальних статистик відміток	76
5 РОЗРОБКА СТАРТАП ПРОЕКТУ	80
5.1 Етапи розробки стартап-проекту.....	80
5.2 Опис ідеї продукту.....	81
5.3 Технологічний аудит ідеї проекту.....	82
5.4 Аналіз ринкових можливостей запуску стартап-проекту.....	82
5.5 Розроблення ринкової стратегії проекту	84
5.6 Розроблення маркетингової програми стартап-проекту.....	86
5.7 Висновки	87
ВИСНОВКИ.....	89
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	90

ПЕРЕЛІК СКОРОЧЕНЬ

РЛС – радіолокаційна станція.

БПЛА – безпілотний літальний апарат.

PDA – The probabilistic data association.

ЕОМ – електронно-обчислювальна машина.

ВСТУП

У сучасній війні радары відповідають за моніторинг, раннє попередження, виявлення стеження, навігаційне наведення та інші багатофункціональні завдання. Від їх інформаційної здатності безпосередньо залежить боєздатність озброєння та техніки. Останніми роками дослідження супроводу маневруючих цілей є гарячою темою в галузі радіолокації. Ключем до алгоритму супроводу цілі РЛС є можливість точного супроводу цілі, тобто забезпечення точності алгоритму. При цьому слід враховувати час реалізації алгоритму, тобто гарантувати продуктивність у реальному часі в інженерних додатках. Загалом, алгоритм супроводу одиночної цілі РЛС складається з двох частин: одна частина – це оцінка стану, а саме оцінка стану руху цілі в реальному часі різними методами; інша частина описує закон зміни цільового рухомого стану шляхом створення різних моделей.

Алгоритми супроводження надають можливість передбачити майбутнє положення кількох рухомих об'єктів на основі історії окремих положень, які повідомляють системи датчиків.

Радіолокаційне супроводження цілі полягає у використанні радару для спостереження та аналізу заблокованої цілі, отримання швидкості, розташування та іншу інформацію про ціль, а також встановити відповідну динамічну модель для стану руху цілі. Потім комп'ютер прогнозує та оцінює цю інформацію про ціль у наступний момент через ряд методів фільтрації, щоб точно встановити траєкторію цілі. Алгоритм супроводу цілі є важливою частиною обробки радіолокаційної інформації.

Задача виявлення та подальшого супроводження сучасних малорозмірних безпілотних засобів розвідки в зоні проведення бойових дій є актуальною. Для протидії таким апаратам розробляються спеціальні засоби невогневого функціонального ураження, які потребують певних точнісних характеристик координатної інформації. Одним із джерел такої інформації можуть стати існуючі

багатоканальні РЛС супроводження повітряних цілей, що забезпечують високу точність вимірювання дальності та кутових координат. При подоланні системи ППО засоби повітряної розвідки можуть застосовувати різні види маневрування, чим погіршують точність роботи радіотехнічних слідкуючих систем. Таким чином, є необхідність в оцінці точнісних характеристик супроводження цілей багатоканальною РЛС та визначення максимальної дальності її застосування.

Системи автосупроводження РЛС синтезовані на основі моделі руху цілі і моделі спостережень (вимірювань). Використовуючи сучасний пакет моделювання розраховано дисперсію помилки слідкування за дальністю та кутовими координатами. Показано, що існують обмеження за дальністю видачі координат безпілотних літальних апаратів багатоканальною РЛС для засобів функціонального ураження.

Невеликі БПЛА також можуть бути використані не тільки у воєнних цілях, а й для здійснення антисоціальних, небезпечних і навіть злочинних дій, таких як порушення конфіденційності, небезпека зіткнення (з людьми, іншими БПЛА та більшими літальними апаратами) і навіть перевезення заборонених матеріалів і/або вибухівки або біологічних агентів. У результаті зростає інтерес до розробки систем, які можуть виявляти та відстежувати БПЛА. Невеликі БПЛА також здатні здійснювати дуже різноманітні рухи, наприклад зависати, що ускладнює завдання відокремлення їх від нерухомого фону. Крім того, висока маневреність малих БПЛА ускладнює проблему відстеження, оскільки неможливо зробити сильні припущення щодо очікуваного руху БПЛА.

Безпілотний літальний апарат (БПЛА) є популярним літальним засобом як у військовій, так і в цивільній сферах. Радар є ефективним інструментом для спостереження за повітряними цілями та оцінки параметрів цілей і широко використовується для управління повітрям. Однак останніми роками для радарів стало складно стежити за такими невеликими літаючими цілями на низькій висоті. Для виявлення та відстеження БПЛА ключовою проблемою є

придушення перешкод. Оскільки на БПЛА легко впливають перешкоди на землі, часто виникають помилкові тривоги та пропуски виявлення. Крім того, траєкторія БПЛА легко переривається. Звичайні методи придушення перешкод включають індикацію рухомих цілей і виявлення рухомих цілей, які можуть придушувати лише головну пелюстку перешкод і не забезпечують хорошу продуктивність для малих цілей. Алгоритм CLEAN може пригнічувати бічні пелюстки перешкод, але страждає від великого обчислювального навантаження. Методи частотно-часового аналізу, такі як вейвлет-перетворення, короткочасне перетворення Фур'є та емпіричне розкладання моди, також були введені для відділення ефективної цілі від перешкод. Однак виявлення та відстеження невеликих БПЛА в захищених середовищах залишається проблемним.

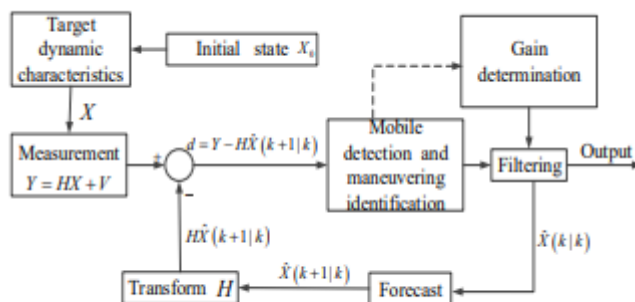
1 АНАЛІЗ ІСНУЮСНУЮЧИХ АЛГОРИТМІВ СУПРОВОДЖЕННЯ ЦІЛІ ПО ДАНИМ РАДІОЛОКАТОРА

1.1 Огляд алгоритмів супроводження одиночної цілі

Алгоритми супроводження надають можливість передбачити майбутнє положення кількох рухомих об'єктів на основі історії окремих положень, які повідомляють системи датчиків.

Радіолокаційне супроводження цілі полягає у використанні радара для спостереження та аналізу заблокованої цілі, отримання швидкості, розташування та іншу інформацію про ціль, а також встановити відповідну динамічну модель для стану руху цілі. Потім комп'ютер прогнозує та оцінює цю інформацію про ціль у наступний момент через ряд методів фільтрації, щоб точно встановити траєкторію цілі. Алгоритм супроводу цілі є важливою частиною обробки радіолокаційної інформації та її основний принцип показаний на рис. 1. Коли ціль маневрує, значення залишку V в вимірюванні збільшиться, цього разу виявлення маневрування базується на зміні V .

Таким чином, комп'ютер визначає підсилення фільтра, коваріаційну матрицю та інші параметри, виходи фільтра виконання функції супроводу маневруючої цілі.



Рисинок 1.1 — Схема радіолокаційного супроводу одиночної цілі

Алгоритм оцінки стану

Оцінка стану є основною задачею супроводу цілі. Здатність досягти точної оцінки цільового стану є необхідною умовою для подальшої обробки даних. Зазвичай оцінку цільового стану можна розділити на два типи: лінійний алгоритм оцінки стану та нелінійний алгоритм оцінки стану.

Алгоритм лінійного оцінювання стану

Найбільш ранній алгоритм оцінки стану слід відстежити на початку дев'ятнадцятого століття. Математик Гаусс запропонував метод найменших квадратів, і цей метод вперше був застосований для спостереження за орбітою планети. У 1940-х роках, виходячи з потреб радіолокаційної системи, Вінер розробив і запропонував фільтр Вінера. Метод використовував рівняння Вінера-Хопфа для розв'язання аналітичного розв'язку оптимальної функції передачі для досягнення стробування та пригнічення сигналу. Однак фільтр Вінера не може працювати з нестационарним випадковим процесом. Він потребує зберігання великої кількості даних у програмі, і це не сприяє розробці. Тоді був запропонований α - β фільтр для аналізу системи відстеження краю сканування. Як фільтр із постійним підсиленням він широко використовується завдяки перевагам невеликих обчислень і простоті інженерної реалізації. У 1960-х роках, з безперервним удосконаленням і розвитком теорії оптимального оцінювання, для дефектів застосування фільтра Вінера американський вчений Калман вивів фільтр Калмана (ФК), який мав віхове значення в історії теорії управління. Алгоритм вводить поняття простору станів, використовує рівняння стану для вираження зв'язку між входом і виходом і оцінює необхідну інформацію з вимірних даних. У 1970-х роках Спейнгарн запропонував лінійний авторегресійний алгоритм, але алгоритм ігнорував вплив технологічного шуму, і його не вдалося застосувати в техніці. Фільтру Калмана не потрібно зберігати всі історичні дані та не вимагає суворої плавності системи. Характеристики ітераційної рекурсії також допомагають комп'ютеру розв'язувати

проблему, тому ФК швидко привернув увагу науковців і застосовується в багатьох галузях, таких як керування навігацією. Фільтр Калмана має просту конструкцію, широко використовується, але через необхідність використання комп'ютерної реалізації призводить до помилки округлення, це, в свою чергу, призводить до нестабільності результатів фільтра. У зв'язку з цим Джульєр і Ульманн запропонували фільтр Калмана розкладу за сингулярним значенням, фільтр Калмана квадратного кореня та інший чисельний надійний фільтр, щоб покращити чисельну стабільність алгоритму [27].

Алгоритм нелінійного оцінювання стану

Хоча фільтр Калмана має багато переваг і застосовується в багатьох областях, він обмежений лінійними системами. У реальних умовах маневрування цілі зазвичай нелінійне. У зв'язку з цим вчені досліджували нелінійну фільтрацію на основі фільтра Калмана. Є чотири основні методи:

1. Вдосконалений алгоритм на основі апроксимації функції.

У 1970-х роках Бьюсі використав розкладання Тейлора першого порядку для лінеаризації нелінійної проблеми на основі ФК і запропонував розширений фільтр Калмана (РФК). РФК має переваги простої структури, легкої реалізації та широкого застосування. Однак через помилку лінеаризації в процесі РФК ефект буде незадовільним або навіть розбіжним, коли ступінь нелінійності високий. Щоб зменшити похибки в процесі лінеаризації та підвищити точність РФК, вчені вдосконалили його та висунули багато нових алгоритмів. Наприклад, Махаланабіс запропонував алгоритм РФК другого порядку (РФКДП). Белл вивів ітераційний алгоритм РФК (ІРФК), використовуючи ітерацію Гаусса-Ньютона, щоб підвищити точність фільтра, але недолік полягав у тому, що його продуктивність конвергенції не покращилася. Галковський і Герчі запропонували алгоритм модифікованого підсилення РФК (МПРФК) [28,29], який стало важко узагальнити через відсутність одноманітності.

2. Покращений алгоритм на основі детермінованої вибірки.

Щоб вирішити проблему величезних похибок у сильній нелінійності РФК, Жульє вперше запропонував алгоритм беззапахового фільтра Калмана (БФК) [30], заснований на беззапаховому перетворенні (БП) у 1997 році. Алгоритм використовує функцію щільності ймовірності для наближення нелінійної функції за допомогою вибираючи $2n+1$ сигма точок, і точність оцінки може досягати другого порядку. У разі однакового обсягу обчислень з РФК, БФК більш точний, ніж РФК, не потребує обчислення матриці Якобі, і його робота більш зручна. З того часу для БФК було розроблено низку вдосконалених алгоритмів. Щоб вирішити проблему нелокальної вибірки та перехресної матриці високого порядку БФК, коли розмірність системи є високою, Джульєр запропонував метод зменшення кількості точок вибірки в 2002 році. Щоб підвищити ефективність алгоритму, Ван вивів алгоритм квадратного корню БФК, який був чисельно стабільним, так що обчислення було значно скорочено. Чжоу Чжань-сінь поєднав генетичний алгоритм із БФК, що покращило метод побудови сигма-точок вибірки та запропонувало адаптивний ваговий підхід алгоритму БФК. З огляду на повільну швидкість збіжності БФК, Зан Р Х, Сіблі Г і Ченг Шуї-їнь запропонували різні форми ітераційного алгоритму БФК (ІБФК) для підвищення швидкості збіжності, але кількість обчислень відповідно зростає. Потім Лю Да-Пен висунув гібридний ітераційний алгоритм БФК, який гарантував швидкість конвергенції та зменшив обчислювальну складність.

3. Вдосконалений алгоритм на основі кубатурного критерію.

У 2009 році Арасаратнам і Хайкін запропонували кубатурний фільтр Калмана (КФК) [31, 32]. Алгоритм базувався на критерії сферично-радіальної кубатури третього порядку, і процес виведення був дуже суворим. КФК має такі переваги, як простота в реалізації, малі обчислення та висока точність. Bazуючись на чудовій продуктивності КФК, вітчизняні та іноземні вчені провели більш поглиблені дослідження та покращили для розширення простору застосування КФК. У 2010 році Арасаратнам і Хайкін запропонували алгоритм ква-

дратного кореня КФК (ККФК) для покращення стабільності КФК і забезпечення невід'ємного детермінізму коваріаційної матриці. Щоб зменшити вплив нелінійності початкового рівняння вимірювання помилки на оцінку стану, Му Джінг запропонував ітераційний алгоритм КФК (ІКФК), поєднавши ітерацію Ньютона-Гауса з КФК, що покращує точність оцінки. Лю Ян висунув КФК трьох порядків і вивів узагальнений алгоритм КФК, а також перевіряв його КФК п'ятого порядку. Було доведено, що вищий порядок може підвищити точність оцінювання алгоритму, але збільшити обчислювальну складність.

4. Покращений алгоритм на основі випадкової вибірки.

Усі перераховані вище алгоритми можуть мати справу лише зі статистичними характеристиками шуму Гаусової нелінійної системи. Для негаусового шуму необхідно реалізувати фільтр частинок (ФЧ), який базується на методі Монте-Карло та Байєсівській оцінці. У 1950-х роках американський вчений Хаммерслі запропонував алгоритм послідовної важливої вибірки (ПВВ). Цей алгоритм використовувався для оцінювання стану у фізиці та статистиці, що заклало теоретичну основу для генерації ФЧ. У 1970-х роках багато експертів вивчали ПВВ, щоб він був розроблений. Експерти також виявили проблему, тобто деградацію частинок. Деградація частинок призводить до марної витрати обчислювальних ресурсів і безпосереднього відхилення результатів. Проблема не була повністю вирішена, поки Гордон не представив алгоритм повторної дискретизації в початковій програмі в 1993 році, до певної міри, щоб вирішити проблему деградації частинок, але значно покращив обчислення. У 1999 році Карпентер вперше запропонував концепцію фільтра частинок (ФЧ) на основі узагальнення алгоритму послідовної вибірки важливості. Відтоді ФЧ вступив у стадію бурхливого розвитку.

Хан представив процес Монте-Карло ланцюга Маркова в ФЧ і запропонував алгоритм МКЛМФЧ. Для методу передискретизації дослідники провели ряд удосконалень, головним чином передискретизацію за кількома шаблонами, системну передискретизацію, ієрархічну передискретизацію, залишкову

передискретизацію тощо. Подібним чином Ошман запровадив генетичний алгоритм повторної дискретизації та запропонував метод генетичного фільтру частинок, який до певної міри покращив точність, але кількість обчислень значно зростає зі збільшенням кількості ітерацій. Щоб вибрати функцію щільності ймовірності важливості, Му Джінг використовує КФК для створення функції щільності важливості, і запропонував алгоритм кубатурного фільтра частот (КФЧ), перевагою якого є короткий час роботи та мала помилка оцінки. Юань Зе-джіань використовував фільтр Гаусса-Ерміта для виробництва фільтра частот Гаусса Ерміта (ГЕФЧ).

Модель руху цілі

Для цільової системи відстеження, щоб забезпечити стабільну та послідовну конвергенцію продуктивності фільтра, ми повинні переконатися, що системна модель спочатку є неупередженою. Тому важливо створити математичну модель для якісної роботи процесу відстеження цілі. Модель руху маневрової цілі поділяється на одиночну модель і мультимодель [33] за кількістю моделей.

Алгоритм одиночної моделі

У 1973 році Бріджвейт і Фрідланд послідовно запропонували модель постійної швидкості (ПШ) і модель постійного прискорення (ПП). Модель ПШ враховує, що ціль здійснює лінійний рух, швидкість є постійною, а прискорення є у формі білого шуму Гауса. Модель ПП припускає, що ціль є лінійним рухом, а прискорення є постійним. Моделі ПШ і ПП є основними моделями для відстеження цілі. Сінгер стверджував, що модель маневрування була корельованою шумовою моделлю, а не звичайною моделлю білого шуму. Він припустив, що прискорення є стохастичним процесом першого порядку нульового значення з експоненціальною автокореляцією, і що функція часової кореляції задовольняє умови експоненціального спадання, таким чином була висунута модель Сінгера. Модель добре справляється з повільним маневруванням як у постійному, так і в рівномірному діапазонах прискорення, але її

недоліком є те, що прискорення дорівнює нулю у будь-який час, і існує серйозна помилка в умовах сильної проблеми маневрування. Хелферті повністю використав ідею моделі Сінгера, узагальнив її на круговий рух і запропонував модель Хелферті. Модель описує рух маневрової цілі шляхом вибору інструкцій, які підкоряються напівмарківському процесу, а величина прискорення визначається ймовірністю переходу. У 1983 році професор Чжоу Хун-жень вважав, що в наступний момент цільове прискорення не може бути повністю поза значенням попереднього моменту, але значення було в поточному прискоренні інтервалу сусідства. Тому він запропонував «поточну» статистичну модель, яку також називають моделлю середнього адаптивного прискорення. Ця модель це лише ще одна модель Сінгера, але вона більш реалістично відображає масштаб маневру цілі. У 1992 році Вастон і Блер створили модель координованого повороту (КП). Спочатку вони припустили, що ціль має здійснювати рівномірний рух по колу, а потім використали співвідношення між швидкістю та прискоренням, щоб вибрати матрицю передачі, параметром якої була кутова швидкість для опису траєкторії цілі, щоб визначити рух цілі. Згодом Бест і Нортон висунули модель дуги на основі моделі КП. Модель є більш загальною. Мацузакі висунув модель САV, яка заснована на характеристиках кругового руху. У цій моделі рух кривої розкладається на рівномірний лінійний рух уздовж напрямку швидкості та рух із постійною кутовою швидкістю навколо двох осей, перпендикулярних до напрямку швидкості. Його часто використовують для опису тривимірного кругового руху просторової цілі. Мехротра зазначив, що обмеження порядку похідної вектора стану є важливою причиною, чому всі моделі не можуть точно охарактеризувати складний рух цілі. Тож Мехротра ввів швидкість зміни прискорення (ривок), коли встановив цільову модель руху, і встановив модель ривка, щоб пристосуватись до більшого маневрування.

Алгоритм мультимоделі

Мультимодельний алгоритм першого покоління. У 1965 році Магіл запропонував перше покоління мультимодельного алгоритму, який базувався на різних моделях фільтрів, що працюють паралельно та потім об'єднують виходи. Робочий процес показано на рис. 2. У порівнянні з одиночною моделлю, коли вибраний набір моделей може охоплювати всі форми руху, алгоритм може досягти найкращого ефекту фільтрації. Однак очевидним недоліком є те, що номер моделі фіксований. Він не врахував перехід між режимами системи і не зміг повністю вибрати відповідну модель. Коли модель більше, ефективність відстеження стає низькою, і між моделями не вистачає взаємодії.

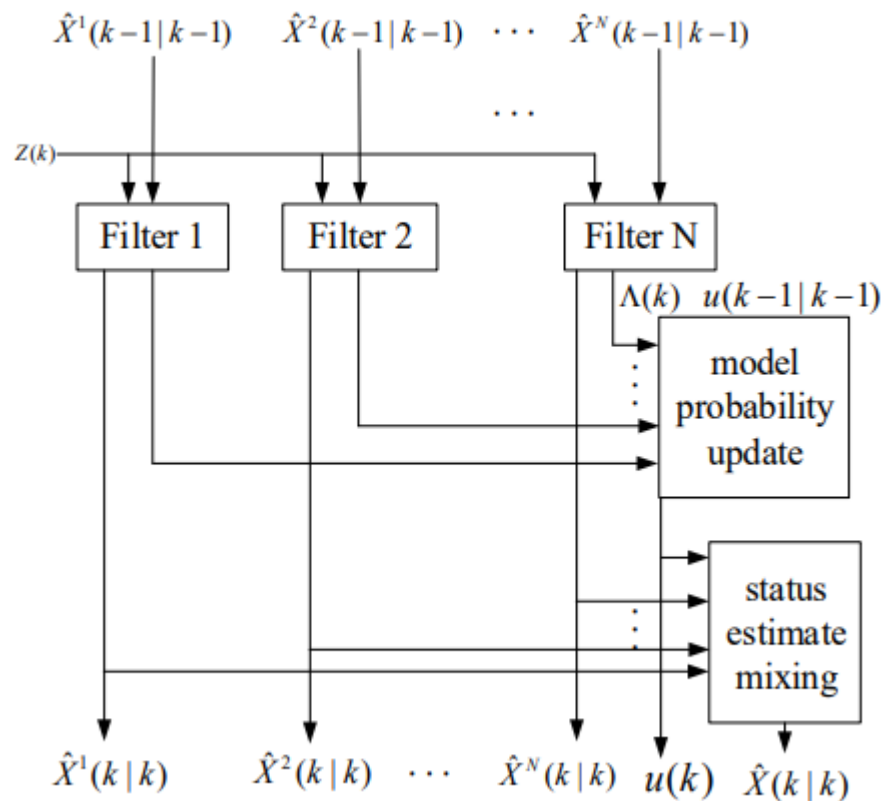


Рисунок 1.2 — Принципова діаграма алгоритму ММ першого покоління

Мультимодельний алгоритм другого покоління. Кількість моделей другого покоління ММ алгоритму все ще фіксований, але вибрана модель має взаємодію. Існують алгоритм УПБ і алгоритм ІММ.

1. Алгоритм УПБ.

У 1970 році Акерсон запропонував Узагальнений Псевдо-Байсіанський алгоритм (УПБ). Алгоритм враховує, що кожна з вибраних моделей відповідає стану руху цілі, і виконує однакові дані для кожної моделі. Однак алгоритм не враховує характеристики кожної моделі та не використовує інформацію про модель, тому його не можна застосовувати широко.

2. Алгоритм ВКМ

У 1988 році Блом і Бар-Шалом запропонували новий алгоритм для апроксимації цільового стану руху шляхом імовірнісного зважування кількох моделей на основі алгоритму УПБ, а саме алгоритму взаємодії кількох моделей (ВКМ). На рис. 3 представлена схематична діаграма алгоритму ВКМ. Алгоритм ВКМ розглядає перехід між моделями як марковський процес і перетворює їх за допомогою ймовірнісної матриці передачі. Алгоритм отримав широке застосування через такі його переваги, як відсутність

необхідність виявлення маневрування, проста структура, помірні обчислення тощо. Вивченню алгоритму ВКМ та його вдосконаленню присвятили багато вчених.

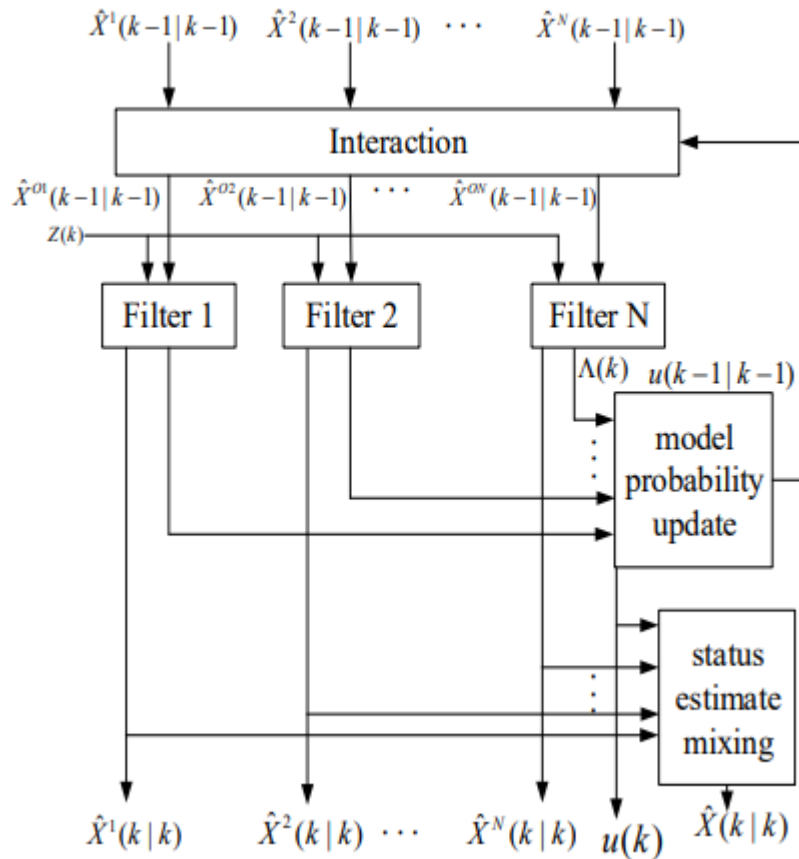


Рисунок 1.3 — Принципова схема алгоритму ВКМ

Мультимодельний алгоритм третього покоління. Незважаючи на те, що ВКМ досяг значного прогресу порівняно з першим поколінням багатомодельного алгоритму, все ще були деякі недоліки. Коли ціль використовує різноманітні методи маневру, маленький набір моделей не може точно описати ситуацію маневру. Крім того, коли кількість моделей збільшується, обчислювальна складність буде значно вище. Крім того, збільшення кількості моделей також посилює конкуренцію між різними моделями, що негативно впливає на продуктивність. Щоб вирішити вищезазначені проблеми, у 1992 році Лі запропонував мультимодельний алгоритм зі змінною структурою (ММЗС) із змінним у часі набором адаптивних перемикачів для заміни фіксованої структури обмежень, у якому ключовою частиною є адаптивний механізм модельного набору. Для проблеми визначення набору моделей-кандидатів і вибору оптимального

набору моделей Лі запропонував чотири алгоритми, які зазвичай використовуються в літературі: перемикання груп моделей (ПГМ), набір імовірних моделей (НІМ), адаптивна сітка (АС) і модельний алгоритм очікування (МАО). ПГМ є першим запропонованим, і представленням є адаптивний алгоритм набору моделей перемикання орграфів (ПО). Його продуктивність краща, ніж у алгоритму ВКМ, але алгоритм може бути спричинений активацією неправильного набору моделей і погіршенням продуктивності. Представником НІМ є адаптивний алгоритм набору моделей активних орграфів, економічна ефективність якого низька та її легко реалізувати. Недоліком є те, що якщо дизайн топології є невідповідним, алгоритм перетворення може бути не в змозі обробити два різних шаблони та призвести до більшої помилки. Алгоритм адаптивної сітки, що взаємодіє з декількома моделями (АСВКМ), є типовим випадком алгоритму адаптивної сітки (АС), запропонованого Жилковим. МАО може генерувати модель, що відповідає шаблону, в реальному часі. Точність оцінки МАО дуже висока, але обсяг обчислень великий, і його відносно важко досягти. Загалом продуктивність алгоритму ММЗС краща, ніж алгоритму ВКМ. Більше того, чим сильніша рухливість цілі, тим більш сприятливі переваги покаже ММЗС.

1.2 Огляд алгоритмів супроводження декількох цілей цілі

Більшість алгоритмів супроводження декількох цілей (СДЦ) передбачає стандартну багатоцільову модель переходу, в якій кожна існуюча ціль x_{k-1} у момент часу $k-1$ або продовжує існувати в момент часу k з імовірністю $P_{S,k|k-1}(x_{k-1})$ і переходить до новий стан x_k з щільністю ймовірності $f_{k|k-1}(x_k/x_{k-1})$, або зникає з ймовірністю $1 - P_{S,k|k-1}(x_{k-1})$. Крім того, випадкова кількість нових цілей може з'явитися з випадкових місць у просторі станів у момент часу k . Передбачається, що кожна ціль з'являється та розвивається незалежно від інших. Різні багатоцільові підходи відстеження використовують різні моделі цільових появ і зникнень. Метою СДЦ є спільне оцінювання за

кожний час спостереження кількості цілей та їхніх траєкторій за даними датчиків. Навіть на концептуальному рівні СДЦ є нетривіальним розширенням відстеження однієї цілі. Дійсно, СДЦ – це набагато більше складний і в теорії, і в практиці.

Найближчий сусід

Декілька цілей додають до задачі відстеження необхідність призначати кожен вхідний звіт про місцезнаходження певного цільового сліду. Найпершим механізмом класифікації звітів було правило найближчого сусіда. Ідея правила полягає в тому, щоб оцінити положення кожного об'єкта під час нового звіту про місцезнаходження, а потім призначити звіт найближчій такій оцінці (рис. 4). Цей інтуїтивно зрозумілий підхід є особливо привабливим, тому що він розкладає проблему відстеження кількох цілей на набір проблем з однією ціллю. Правило найближчого сусіда легко застосувати, коли всі сліди та звіти представлені як точки; однак немає чітких засобів для визначення того, що є «найближчими сусідами» серед слідів і звітів з різними коваріаціями помилок. Наприклад, якщо датчик має дисперсію похибки в 1 см, то ймовірність того, що вимірювання на відстані 10 см одне від одного зроблено одним об'єктом, дорівнює $O(10^{-20})$, тоді як вимірювання з дисперсією в 10 см можуть мати різницю в 20–30 сантиметрів. і можливо відповідають тому самому об'єкту. Тому відповідна міра відстані повинна відображати відносну невизначеність середніх оцінок.

Правило найближчого сусіда має сильну інтуїтивну привабливість, але незабаром з'явилися сумніви і труднощі, пов'язані з ним. Наприклад, перші реалізатори методу виявили проблеми у створенні початкових слідів для кількох цілей. У випадку однієї цілі можна накопичити два звіти для отримання оцінки швидкості, на основі якої можна створити доріжку. Однак для кількох цілей немає очевидного способу вивести такі початкові швидкості. Перші два отримані звіти можуть означати послідовне розташування одного об'єкта або по-

чаткове виявлення двох різних об'єктів. Кожен наступний звіт може бути продовженням відомого сліду або початком нового. Що ще гірше, майже кожен датчик видає певну фонову частоту помилкових звітів, які створюють помилкові сліди. Таким чином, система відстеження потребує додаткового механізму для розпізнавання та видалення слідів, які не отримують жодних підтверджуючих звітів. Ще одна складність із правилом найближчого сусіда стає очевидною, коли звіти неправильно класифікуються, як це неминуче трапляється час від часу, якщо відстежувані об'єкти знаходяться близько один до одного. Помилкове призначення може спричинити дуже повільну збіжність процесу фільтрації за Калманом або повну неспроможність збіжності, і в цьому випадку трек неможливо передбачити. Крім того, доріжки, оновлені за допомогою неправильно призначених звітів (або не оновлені взагалі), погано співвідносяться з наступними звітами, і тому механізм видалення доріжок може бути сприйнятий як фальшивий. Помилково видалені доріжки вимагають наступного запуску доріжки та можливого повторення процесу.

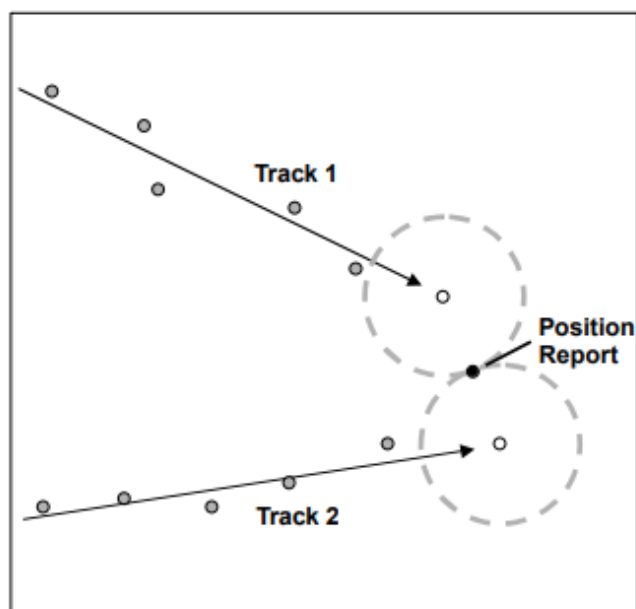


Рисунок 1.4 — Правило найближчого сусіда є, мабуть, найпростішим підходом для визначення того, який відстежуваний об'єкт створив певний звіт датчика. Коли надходить новий звіт про місцезнаходження, усі існуючі сліди проєктуються вперед до часу нового вимірювання. (На цій діаграмі попередні

цільові позиції позначено крапками, а прогнозовані – колами; новий звіт про місцезнаходження позначено.) Потім обчислюється відстань від звіту до кожної прогнозованої позиції, і звіт пов'язується з найближчим треком. Загалом, розрахунок відстані обчислюється для відображення відносної невизначеності (коваріації), пов'язаної з кожним слідом і звітом. У описаній вище ситуації звіт було б віднесено до доріжки 1 виключно на основі його евклідової близькості до звіту. Якщо це призначення є помилковим, це негативно вплине на подальший процес відстеження.

Поділ доріжок слідів і кілька гіпотез

Надійним рішенням проблеми неоднозначності призначення є створення кількох доріжок гіпотез. Згідно з цією схемою, система відстеження не зобов'язана негайно або безповоротно виконувати окреме призначення кожного звіту. Якщо звіт сильно корелює з більш ніж одним слідом, можна створити оновлену копію кожного сліду; наступні звіти можна використовувати для визначення правильного призначення. У міру надходження більшої кількості звітів траєкторія, пов'язана з правильним призначенням, швидко сходиться зі справжньою цільовою траєкторією, тоді як помилково оновлені сліди з меншою ймовірністю будуть корелювати з наступними звітами. Ця базова техніка називається поділом доріжки. Одним із тривожних наслідків є збільшення кількості доріжок, за якими програма має стежити. Розповсюдженням можна керувати за допомогою того самого механізму видалення доріжок, який використовується в алгоритмі найближчого сусіда, який час від часу сканує всі доріжки та усуває ті, які мають низьку ймовірність асоціації з останніми повідомленнями. Більш складний підхід до поділу слідів, який називається відстеженням кількох гіпотез, зберігає історію розгалужень доріжок, тож щойно одна гілка буде підтверджена, альтернативні гілки можна буде відрізати. Поділ слідів у його різних формах є широко застосовуваною стратегією для вирішення неоднозначностей, властивих кореляції слідів зі звітами від кількох цілей. Він

також використовується для мінімізації впливу помилкових звітів під час відстеження однієї цілі. Тим не менш, деякі серйозні труднощі залишаються. По-перше, поділ слідів не розкладає повністю проблему відстеження кількох цілей на незалежні проблеми з однією ціллю, як мала працювати стратегія найближчого сусіда. Наприклад, дві гіпотетичні доріжки можуть збігатися з траєкторією одного об'єкта. Оскільки обидві доріжки дійсні, стандартний механізм видалення доріжок не може видалити жодну з них. Процедура видалення потрібно змінити для виявлення зайвих слідів і, отже, не можна переглядати лише одну доріжку за раз. Цей зв'язок між кількома слідами теоретично викликає занепокоєння; однак досвід показав, що ним можна керувати на практиці за низьких обчислювальних витрат. Друга проблема полягає в тому, що складно вирішити, коли звіт про позицію та прогнозована доріжка корелюють настільки тісно, щоб виправдати створення нової гіпотетичної доріжки. Якщо кореляційний поріг встановлений занадто високим, правильні призначення можуть пропускатися так часто, що завадить збіжності фільтра Калмана. Якщо поріг занадто низький, кількість гіпотез може зростати в геометричній прогресії. Звичайна практика полягає в тому, щоб встановити поріг досить низьким, щоб забезпечити конвергенцію, а потім додати інший механізм для обмеження швидкості генерації гіпотез. Проста стратегія полягає у виборі n гіпотез-кандидатів з найвищими ймовірностями асоціації, де n є максимальною кількістю гіпотез, яку дозволять обмеження обчислювальних ресурсів. Цей «жадібний» метод часто дає хорошу продуктивність. Навіть із цими вдосконаленнями алгоритм відстеження висуває такі величезні вимоги до обчислювальних ресурсів, що великі проблеми залишаються поза межами практичної досяжності. Моніторинг обчислень, щоб побачити, скільки часу витрачається на різні підзадачі, показує, що обчислення ймовірностей асоціації є, безумовно, найбільшою витратою. Програма заплутана проектуванням цільових треків на час звіту про місцезнаходження та обчисленням ймовірностей асоціації. Оскільки це критична частина алгоритму, подальші зусилля були зосереджені на покращенні продуктивності в цій області.

Спільне об'єднання ймовірнісних даних

Спільне об'єднання ймовірнісних даних (JPDA, англ. — Joint probabilistic data association) — це засіб оцінки неманевруючих багатооб'єктних траєкторій із завадами за допомогою одного сканування. JPDA є найпростішим, а також найефективнішим з точки зору обчислень алгоритмом відстеження кількох об'єктів. Однак це не забезпечує ймовірність існування об'єкта, який буде використовуватися як міра якості сліду за допомогою процедури розрізнення хибного сліду.

- JPDA припускає, що кожен слід, який розглядається, слідує за існуючим унікальним об'єктом, тобто ймовірність існування об'єкта для кожного сліду дорівнює одиниці та не оновлюється,

$$p(\chi_{k-1}|\mathbf{Y}^{k-1}) = p(\chi_k|\mathbf{Y}^{k-1}) = p(\chi_k|\mathbf{Y}^k) = 1.$$

- JPDA припускає, що жоден із відстежуваних об'єктів не маневрує, тобто об'єкти слідують одній моделі траєкторії (кількість моделей траєкторії, $M = 1$),

$$\mu_{k-1|k-1}(1) = \mu_{k|k-1}(1) = \mu_{k|k}(1) = 1.$$

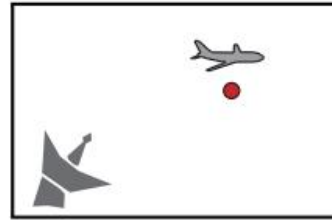
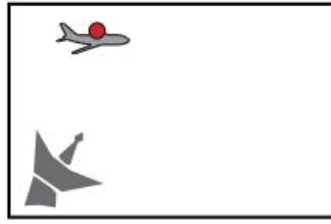
- JPDA — це алгоритм одноразового сканування, тобто всі пробні компоненти сліду об'єднуються в один (кількість попередніх і задніх компонентів сліду дорівнює одному, $C_{k-1} = C_k = 1$), і як апіорний, так і апостеріорна функція щільності ймовірності об'єкта оцінки траєкторії апроксимуються одним гауссовською густини ймовірності для кожної доріжки:

$$p(c_{k-1}) = p(c_k) = 1.$$

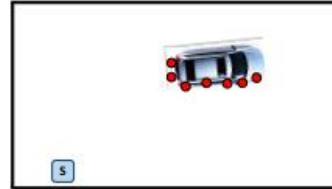
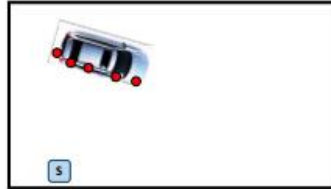
Розширене супроводження цілі

Відстеження об'єкта, який може займати більше однієї комірки датчика, призводить до так званого розширеного відстеження об'єкта або розширеного відстеження цілі. У розширеному відстеженні об'єктів об'єкти викликають різну кількість потенційно зашумлених вимірювань від різних просторово розподілених джерел вимірювань, які також називаються точками відбиття. Форма об'єкта зазвичай невідома і може навіть змінюватися з часом, а метою є

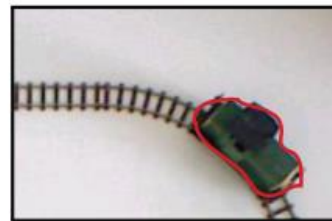
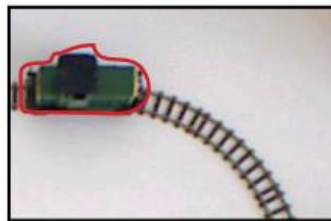
рекурсивне визначення форми об'єкта та його кінематичних параметрів. Через нелінійність кінцевої проблеми оцінки вже відстеження окремого розширеного об'єкта загалом є дуже складною проблемою, для якої потрібні складні методи нелінійної оцінки. Хоча це часто неправильно розуміють, розширене відстеження об'єктів, як визначено вище, принципово відрізняється від типових проблем відстеження контурів у комп'ютерному зорі. У відстеженні контурів на основі візуалізації повне червоно-зелено-синє (RGB, англ. — red-green-blue) зображення доступне в кожному часовому проміжку, а контур виділяється з кожного зображення, яке відстежується з часом. У розширеному відстеженні об'єктів працюється з кількома (зазвичай дво- чи тривимірними) вимірюваннями за крок у часі, тобто розріджена хмара точок. Майже завжди неможливо витягнути форму лише на основі вимірювання одного моменту в часі. Форму об'єкта можна визначити, лише якщо вимірювання протягом кількох часових кроків систематично накопичуються та об'єднуються з урахуванням руху (невідомого) об'єкта та шуму датчика. Ілюстрація різниці між відстеженням точкових об'єктів, розширеним відстеженням об'єктів і контурним відстеженням наведена на рис. 1.



(a) Point object tracking example: Frame 1 (left) & Frame 2 (right)



(b) Extended object tracking example: Frame 1 (left) & Frame 2 (right)



(c) Contour Tracking example: Frame 1 (left) & Frame 2 (right)

Рисунок 1.5 — Ілюстрація різних типів проблем відстеження: а) При відстеженні точкових об'єктів отримується не більше одного вимірювання (червоні маркери) на кадр. б) У розширеному відстеженні об'єкта для кадру отримують численні вимірювання (червоні маркери) від різної кількості джерел вимірювання/центрів відображення. с) При відстеженні контурів один контур (червоний) виділяється з кожного кадру зображення. Отже, можна сказати, що при відстеженні контурів вимірювання є контурами, тоді як при відстеженні розширеного об'єкта вимірюваннями є (декартові) точки. Однак як у розширеному відстеженні об'єкта, так і в відстеженні контуру мета полягає в тому, щоб оцінити форму, тобто контур, на основі отриманих вимірювань.

У багатьох практичних застосуваннях необхідно відстежувати кілька розширених об'єктів, де немає зв'язків вимірювання з об'єктом. На жаль, зв'язування даних стає ще складнішим у відстеженні кількох розширених об'єктів,

оскільки можлива величезна кількість подій зв'язування: потрібно пронумерувати всі можливі розділи набору вимірювань, а потім усі можливі способи призначити клітинки розділу оцінкам об'єкта.

У проблемах відстеження фізичні об'єкти реального світу, що представляють інтерес, завжди мають просторові межі. Це справедливо для відносно великих об'єктів інтересу, таких як кораблі, човни, автомобілі, велосипедисти, люди та тварини, і це вірно для відносно невеликих об'єктів інтересу, таких як клітини. Відмінності між розширеним відстеженням об'єктів і відстеженням точкових об'єктів зумовлені властивостями датчика, особливо роздільною здатністю датчика, ніж властивостями об'єкта, такими як просторовий протяжність. Якщо роздільна здатність відносно розміру об'єктів є достатньо високою, то об'єкт може займати кілька клітинок роздільної здатності. Таким чином, у цьому випадку кожен об'єкт може генерувати декілька виявлень за крок у часі. Іншими словами, залежно від властивостей датчика, зокрема від роздільної здатності датчика, виникатимуть різні типи відстеження об'єктів, і тому корисно розрізняти різні типи проблем відстеження об'єктів. Нижче наведено визначення типів проблем відстеження:

- Відстеження точкового об'єкта:

Кожен об'єкт генерує щонайбільше одне вимірювання за крок у часі, тобто одна клітинка роздільної здатності зайнята об'єктом.

- Розширене відстеження об'єктів:

Кожен об'єкт генерує кілька вимірювань за крок у часі, а вимірювання просторово структуровані навколо об'єктів, тобто об'єкт займає кілька клітинок роздільної здатності.

- Групове відстеження об'єктів:

Кожен об'єкт генерує декілька вимірювань за крок у часі, а вимірювання просторово структуровані навколо об'єкта. Груповий об'єкт складається з двох або більше підоб'єктів, які мають спільний рух. Крім того, об'єкти не відстежуються окремо, а розглядаються як єдине ціле. Таким

чином, груповий об'єкт займає кілька комірок дозволу; кожен підоб'єкт може займати одну або кілька клітинок роздільної здатності.

- Відстеження з багатошляховим розповсюдженням:

Кожен об'єкт генерує кілька вимірювань за крок у часі, які є результатом багатопроменевого розповсюдження. Таким чином, вимірювання не є просторово структурованими навколо об'єкта.

Усі методи супроводу, за винятком супроводу за точковим об'єктом, передбачають можливість багаторазових вимірювань на ціль. Через необхідні відмінності в моделюванні руху та вимірювання ми розрізняємо три підходи до відстеження, а не визначаємо один тип під назвою відстеження з кількома виявленнями.

У розширеному відстеженні об'єктів кожен об'єкт є окремою сутністю, наприклад, автомобіль, літак, людина або тварина. Часто форму можна вважати твердим тілом, однак також можливі протяжні об'єкти з розмірами, що деформуються. У груповому відстеженні об'єктів кожен об'єкт є набором (менших) об'єктів, які мають певну спільну динаміку, але все ще допускають індивідуальну динаміку всередині групи. Наприклад, у групі пішоходів існує загальний груповий рух, але окремі пішоходи також можуть змінювати свої позиції в групі.

Розширене супроводження декількох цілей

Незалежно від типу проблеми відстеження (точкове, розширене, групове тощо), СДЦ є задачею, яка має багато проблем:

- Кількість об'єктів невідома і час змінюється.
- Є пропущені вимірювання, тобто на кожному кроці часу деякі з існуючих об'єктів не дають вимірювань.
- Непропущені об'єкти призводять до невідомої кількості виявлень.
- Існують вимірювання перешкод, тобто вимірювання, які не були спричинені цільовим об'єктом.

- Джерело вимірювання невідоме, тобто джерело кожного вимірювання невідоме. Це часто називають «проблемою асоціації даних»

Приклади розширеного та групового МТТ

1) Фільтри РНД та СРНД: фільтр РНД для розширених об'єктів за моделлю Пуассона[9]. Кілька модельних РНД-фільтрів застосовуються до відстеження автомобілів і велосипедів за припущеними моделями форми прямокутника та палиці, і показано, що використання кількох моделей вимірювання може покращити результати оцінки. Доповнення реалізацій гамма-розподілами дає змогу оцінити невідому швидкість вимірювання Пуассона для кожного об'єкта [10]. Отримані алгоритми потім називаються гамма-гаусовим (GG) або гамма-гаусовим оберненим Вішартом (GGIW) відповідно. Підхід до групового відстеження об'єктів на основі фільтра GM-РНД точкового об'єкта представлено в [11]. Розширений об'єктний РНД-фільтр, представлений у [12], [13], отримано для об'єктної моделі, відмінної від моделі процесу точки Пуассона [14]. Об'єкти моделюються за допомогою кластерного процесу Пуассона, ієрархічного процесу з батьківським процесом і дочірнім процесом. Батьківський процес моделює кількість об'єктів з розподілом Пуассона. Для кожного об'єкта дочірній процес моделює ряд точок відбиття, які генерують вимірювання. Пропонується реалізація, де передбачається, що об'єкт має форму еліпса, а точки відображення розташовані на краю еліпса. Було представлено принаймні два різних фільтри СРНД. Фільтр СРНД для розширених об'єктів, представлений у [15], виведений за припущенням, що «відносно роздільної здатності датчика розширені об'єкти та нерозділені об'єкти не розташовані надто близько, а щільність перешкод не надто велика» [15]. Однак це припущення, яке не можна очікувати в загальному випадку. Фільтр СРНД, здатний обробляти як просторово близькі об'єкти, так і щільний перешкод, представлений у [16]–[19], а також представлена реалізація GGIW. Порівняння показує, що фільтр GGIW-СРНД

перевершує фільтр GGIW-PHD, особливо коли ймовірність виявлення низька та/або щільність перешкод висока. Ціною за підвищення продуктивності є збільшення обчислювальних витрат.

2) Фільтри СВ-MeMVer: розширення фільтра СВ-MeMVer [20] для розширених об'єктів, використовуючи модель вимірювання PPP, було представлено в [21]. Реалізація суміші Гауса представлена в [21], а реалізації послідовного Монте-Карло (SMC) СВMeMVer для розширених об'єктів можна знайти в [22], [23]. Розширений об'єктний фільтр СВ-MeMVer з декількома моделями представлено в [24].

3) Спряжені пріоритети: помічені фільтри MB для розширеного відстеження об'єктів представлені в [25], [26], як фільтр GLMB, так і його наближення фільтр LMB. Представлені реалізації GGIW, і результати моделювання показують, що позначені фільтри MB перевершують своїх колег PHD та CPHD.

1.3 Супроводження одиночної цілі при наявності завад

Представлені алгоритми відстеження об'єктів базуються на таких припущеннях, якщо не вказано інше:

- Ціль:
 - У зоні спостереження один об'єкт. Положення об'єкта є апіорі невідомий.
 - Припускається, що можливі моделі траєкторії цілі відомі, і передбачається, що моделі поширюються як ланцюги Маркова.
 - Під час кожного сканування може бути присутнім одне вимірювання на ціль, що описується ймовірністю виявлення, P_D , яка може бути різною для кожної цілі. Іншими словами, ми припускаємо точкові об'єкти, і вони виробляють нуль або одне вимірювання за сканування випадковим чином.
- Завада:

– Щільність вимірювання перешкод у зоні спостереження апіорі відома, інакше ми оцінюємо її за отриманими вимірюваннями.

- **Вимірювання:**

– Вимірювання виконуються датчиком із нескінченною роздільною здатністю. Іншими словами, кожне вимірювання може мати лише одне джерело. Коли вимірювання використовується для оновлення доріжки, це може бути одне з:

- * вимірювання (виявлення) об'єкта відстеження;

- * вимірювання завад.

– Ми розглядаємо тут тільки вимірювання позиції. Додаткові атрибути вимірювання, такі як амплітуда, доплерівська швидкість тощо, можуть бути включені простим способом за допомогою апіорних функцій щільності ймовірності атрибутів вимірювання.

Фільтр асоціації ймовірнісних даних

Найуспішнішим алгоритмом у класі байєсівських фільтрів усіх сусідів є фільтр асоціації ймовірнісних даних (з англ. — The probabilistic data association (PDA) filter). Алгоритм оновлює оцінку стану об'єкта, використовуючи всі підтвержені вимірювання та їхні відповідні апостеріорні ймовірнісні ваги. Алгоритм PDA отримано шляхом припущення, що щільність передбачення в оптимальній рекурсії Байєса, враховуючи всі минулі спостереження, є щільністю Гауса, хоча, строго кажучи, це суміш Гауса. Ця апроксимація фільтра PDA базується на наступних шести припущеннях:

1. Ціль, яка відстежується, існує, а іншої цілі не існує.

2. Рух цілі підкоряється лінійній статистиці Гауса.

3. Тільки одне вимірювання може бути з цілі інтересу.

4. Шум вимірювання є білим Гаусовим шумом.

5. Об'єкт може виявлятися або не виявлятися весь час і виявляється з ймовірністю виявлення ПД.

6. Припускається, що всі вимірювання, що не здійснюються об'єктами, походять від перешкод, рівномірно розподілених у просторі, і Пуассона, розподілених у часі.

7. Лише вимірювання, які знаходяться в межах близькості до очікуваного вимірювання

(тобто, які підпадають під перевірку) розглядаються для обробки.

2 РОЗРОБКА АЛГОРИТМУ АДАПТИВНОГО СУПРОВОДЖЕННЯ БПЛА ЗА ДАНИМИ РЛС З ВИКОРИСТАННЯМ ВИРІШАЛЬНИХ СТАТИСТИК ВІДМІТОК

2.1 Синтез алгоритму адаптивного оцінювання параметрів руху маневруючого МБПЛА

Для вирішення задачі траєкторної фільтрації маневруючого МБПЛА з розпізнаванням різних видів його руху, а також з інших міркувань доцільно оцінювання параметрів його руху здійснювати в прямокутній системі координат (СК) з початком у точці стояння РЛС. Така СК називається місцевою прямокутною. Модель руху маневруючої цілі у місцевій прямокутній СК може бути представлена у вигляді дискретної динамічної стохастичної системи з випадковою структурою.

$$\mathbf{u}(k) = \mathbf{F}_j(k, k-1)\mathbf{u}(k-1) + \mathbf{G}_j(k)\boldsymbol{\omega}(k), \quad j = \overline{1, M}, \quad (2.1)$$

де $\mathbf{u}(k)$ - вектор стану, що включає параметри руху цілі по осях декартової системи координат; $\mathbf{F}_j(k, k-1)$, $\mathbf{G}_j(k)$ - матриці, що описують різні види руху; $\boldsymbol{\omega}(k)$ - некорельована послідовність гауссівських векторів з одиничною кореляційною матрицею, k - кроку фільтрації.

Для опису типу структури моделі руху цілі, що відповідає певному виду маневру, застосовується змінна перемикач $a_j(k)$, $j = \overline{1, M}$. Вона відноситься до класу ланцюгів Маркова, з матрицею ймовірностей переходів $\Pi_{ij}(k, k-1)$, $i, j = \overline{1, M}$ і початковими ймовірностями $p_i(0)$, $i = \overline{1, M}$, що дозволяє врахувати переходи між різними видами руху цілі у довільні моменти часу.

РЛС вимірюють азимут $\beta_{II}(k)$, кут місця $\varepsilon_{II}(k)$, радіальну дальність $r_{II}(k)$ до БПЛА та радіальну швидкість $\dot{r}_{II}(k)$ в сферичній системі координат. Враховуючи, що при побудові моделі руху цілі використано прямокутну систему координат, рівняння спостереження можна представити у вигляді

$$\begin{aligned}
\beta_{\text{И}}(k) &= \arctg\left(\frac{y(k)}{x(k)}\right) + v_{\beta}(k); \\
\varepsilon_{\text{И}}(k) &= \arctg\left(\frac{z(k)}{\sqrt{x^2(k) + y^2(k)}}\right) + v_{\varepsilon}(k); \\
r_{\text{И}}(k) &= \sqrt{x^2(k) + y^2(k) + z^2(k)} + v_r(k); \\
\dot{r}_{\text{И}}(k) &= \frac{x(k)\dot{x}(k) + y(k)\dot{y}(k) + z(k)\dot{z}(k)}{\sqrt{x^2(k) + y^2(k) + z^2(k)}} + v_{\dot{r}}(k),
\end{aligned} \tag{2.2}$$

де $v_{\beta}(k)$, $v_{\varepsilon}(k)$, $v_r(k)$, $v_{\dot{r}}(k)$ - некорельовані гауссівські помилки вимірювань азимуту, кута місця, радіальної дальності і радіальної швидкості БПЛА в полярній СК з нульовим математичними очікуваннями та дисперсіями складових σ_{β}^2 , σ_{ε}^2 , σ_r^2 , $\sigma_{\dot{r}}^2$.

Вимірювані координати положення БПЛА в прямокутній і сферичній системах координат зв'язані нелінійними рівняннями

$$\begin{aligned}
x_{\text{И}}(k) &= r_{\text{И}}(k) \cos \varepsilon_{\text{И}}(k) \cos \beta_{\text{И}}(k); \\
y_{\text{И}}(k) &= r_{\text{И}}(k) \cos \varepsilon_{\text{И}}(k) \sin \beta_{\text{И}}(k); \\
z_{\text{И}}(k) &= r_{\text{И}}(k) \sin \varepsilon_{\text{И}}(k),
\end{aligned} \tag{2.3}$$

де $x_{\text{И}}(k)$, $y_{\text{И}}(k)$, $z_{\text{И}}(k)$ - вимірювані координати положення БПЛА в прямокутній системі координат.

Зв'язок між координатами положення БПЛА в сферичній і прямокутній системах координат показаний на рис. 2.1.

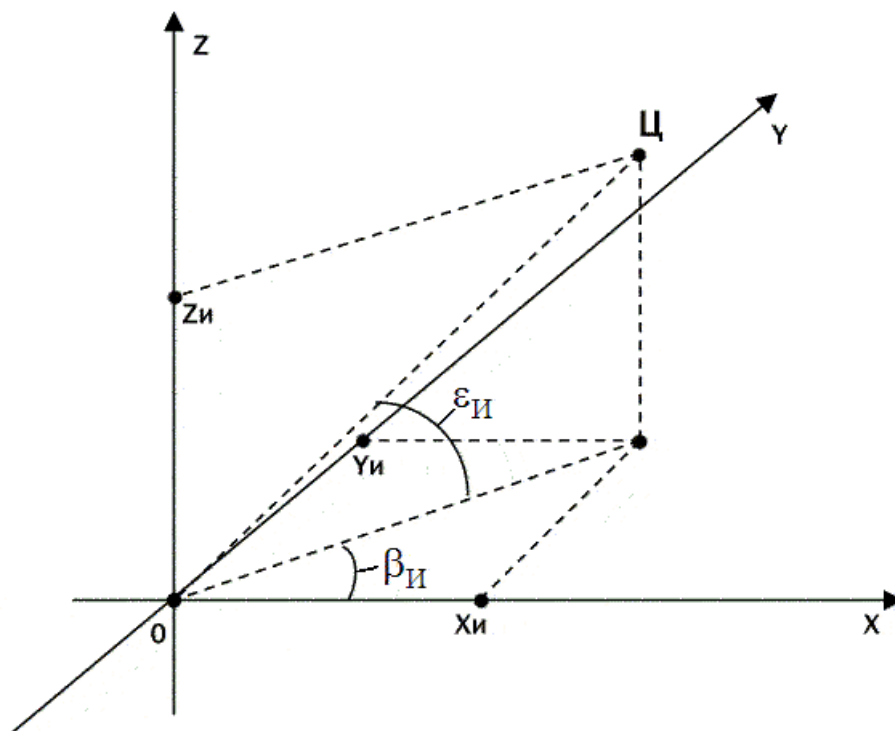


Рисунок 2.1 – Зв'язок між координатами положення БПЛА в сферичній і прямокутній системах координат

Виконавши лінеаризацію виразів (2.3) відносно істинних значень координат БПЛА в сферичній системі координат, рівняння спостереження РЛС в прямокутній системі координат можна привести до вигляду:

$$\begin{aligned} x_{И}(k) &= x(k) + v_x(k); \\ y_{И}(k) &= y(k) + v_y(k); \\ z_{И}(k) &= z(k) + v_z(k); \end{aligned} \tag{2.4}$$

$$\dot{r}_{И}(k) = \frac{x(k)\dot{x}(k) + y(k)\dot{y}(k) + z(k)\dot{z}(k)}{\sqrt{x^2(k) + y^2(k) + z^2(k)}} + v_{\dot{r}}(k),$$

де $v_x(k), v_y(k), v_z(k)$ - помилки вимірювання положення БПЛА у прямокутній СК.

У векторному вигляді рівняння спостереження (2.4) можна представити у вигляді:

$$\mathbf{u}^v(k) = \mathbf{h}(\mathbf{u}(k)) + \mathbf{v}(k), \tag{2.5}$$

де $\mathbf{u}^b(k) = (x^b(k), y^b(k), z^b(k), \dot{r}^b(k))^T$ - вектор спостереження, що включає, виміряні координати положення БПЛА в прямокутній системі координат і радіальну швидкість; $\mathbf{h}(\mathbf{u}(k))$ - нелінійна векторна функція, значенням якої є вектор істинних координат положення БПЛА прямокутної СК і радіальної швидкості цілі $(x(k), y(k), z(k), \dot{r}(k))^T$; $\mathbf{v}(k) = (v_x(k), v_y(k), v_z(k), v_{\dot{r}}(k))^T$ - вектор помилок вимірювання координат положення БПЛА у прямокутній СК та радіальній швидкості цілі з нульовим математичним очікуванням та кореляційною матрицею $\mathbf{R}(k)$.

Кореляційна матриця помилок вимірювання системи відеоспостереження і далекоміра в прямокутній системі координат має вигляд:

$$\mathbf{R}(k) = \begin{bmatrix} \sigma_x^2(k) & \sigma_{xy}^2(k) & \sigma_{xz}^2(k) & 0 \\ \sigma_{xy}^2(k) & \sigma_y^2(k) & \sigma_{yz}^2(k) & 0 \\ \sigma_{xz}^2(k) & \sigma_{yz}^2(k) & \sigma_z^2(k) & 0 \\ 0 & 0 & 0 & \sigma_{\dot{r}}^2(k) \end{bmatrix}. \quad (2.6)$$

При цьому її елементи визначаються по формулах:

$$\begin{aligned} \sigma_x^2(k) &= \sigma_r^2 \cos^2 \varepsilon(k) \cos^2 \beta(k) + r^2(k) \sigma_\varepsilon^2 \cos^2 \beta(k) \sin^2 \varepsilon(k) + \\ &\quad + r^2(k) \sigma_\beta^2 \sin^2 \beta(k) \cos^2 \varepsilon(k); \\ \sigma_y^2(k) &= \sigma_r^2 \cos^2 \varepsilon(k) \sin^2 \beta(k) + r^2(k) \sigma_\varepsilon^2 \sin^2 \beta(k) \sin^2 \varepsilon(k) + \\ &\quad + r^2(k) \sigma_\beta^2 \cos^2 \beta(k) \cos^2 \varepsilon(k); \\ \sigma_z^2(k) &= \sigma_r^2 \sin^2 \varepsilon(k) + \sigma_\varepsilon^2 r^2(k) \cos^2 \varepsilon(k); \\ \sigma_{xy}^2(k) &= 0.5 \sin 2\beta(k) (\cos^2 \varepsilon(k) \sigma_r^2 - r^2(k) \sigma_\beta^2 + \sin^2 \varepsilon(k) r^2(k) \sigma_\varepsilon^2); \\ \sigma_{xz}^2(k) &= 0.5 \sin 2\varepsilon(k) \sin \beta(k) (\sigma_r^2 - r^2(k) \sigma_\varepsilon^2(k)); \\ \sigma_{yz}^2(k) &= 0.5 \sin 2\varepsilon(k) \cos \beta(k) (\sigma_r^2 - r^2(k) \sigma_\varepsilon^2(k)), \end{aligned} \quad (2.7)$$

де σ_r^2 , σ_β^2 , σ_ε^2 - дисперсії помилок вимірювання координат цілі в сферичній системі координат; $r(k)$, $\beta(k)$, $\varepsilon(k)$ - координати цілі в сферичній системі координат.

Компоненти кореляційної матриці помилок вимірювання в прямокутній системі координат залежать від поточного положення БПЛА. Тому вони повинні обчислюватися на кожному кроці спостережень.

Рівняння (2.1), (2.5) є вихідними для синтезу оптимального алгоритму траєкторної фільтрації БПЛА у прямокутній СК. У розглянутій постановці задачі, поряд з оцінкою невідомих параметрів руху об'єкта, має вирішуватись завдання розпізнавання виду його маневру. Результати розпізнавання маневру використовуються для формування оцінок вектора параметрів руху об'єкта. Тому синтезовані з урахуванням рівнянь (2.1), (2.5) алгоритми відносяться до класу адаптивних.

Процес із випадковою структурою $\mathbf{u}(k)$ не належить до класу марківських процесів. Це значно ускладнює синтез алгоритмів оцінювання таких процесів і при використанні байєсівського методу адаптивного оцінювання призводить до фільтрів зі зростаючим числом каналів.

Однак цю проблему можна обійти при розгляді розширеного процесу $(\mathbf{u}(k), a_j(k))$, який належить до класу змішаних марківських процесів у дискретному часі. При цьому оптимальний алгоритм оцінювання безперервного компонента $\mathbf{u}(k)$ є частиною алгоритму оцінювання змішаного процесу $(\mathbf{u}(k), a_j(k))$.

Найбільш повне рішення задачі фільтрації полягає у визначенні апостеріорної щільності ймовірності (ЩЙ) фільтрованого процесу $W(\mathbf{u}(k), a_j(k)) = P(\mathbf{u}(k), a_j(k) / \mathbf{U}_И(k))$, где $\mathbf{U}_И(k) = \mathbf{u}_И(1), \dots, \mathbf{u}_И(k)$ - отримана послідовність вимірювань до k моменту включно. На її основі можуть бути визначені оцінки невідомих параметрів будь-якої функції втрат. На основі

моделі (2.1), (2.5) оптимальний алгоритм фільтрації може бути представлений у вигляді:

$$W_j^*(k) = \sum_{i=1}^M \Pi_{ij}(k, k-1) W_i(k-1); \quad (2.8)$$

$$W_j^*(\mathbf{u}(k)) = \sum_{i=1}^M \Pi_{ij}(k, k-1) W_i(k-1) \times \quad (2.9)$$

$$\times \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Pi(\mathbf{u}(k) / \mathbf{u}(k-1), a_j(k)) W_i(\mathbf{u}(k-1)) d\mathbf{u}(k-1) / W_j^*(k);$$

$$W_j(\mathbf{u}(k)) = P(\mathbf{u}_{II}(k) / \mathbf{u}(k)) W_j^*(\mathbf{u}(k)) / P(\mathbf{u}_{II}(k) / a_j(k), \mathbf{U}_{II}(k-1));, \quad (2.10)$$

$$W_j(k) = P(\mathbf{u}_{II}(k) / a_j(k), \mathbf{U}_{II}(k-1)) W_j^*(k) / P(\mathbf{u}_{II}(k) / \mathbf{U}_{II}(k-1)), \quad (2.11)$$

де $W_j^*(\mathbf{u}(k)) = P(\mathbf{u}(k) / a_j(k), \mathbf{U}_{II}(k-1))$, $W_j(\mathbf{u}(k)) = P(\mathbf{u}(k) / a_j(k), \mathbf{U}_{II}(k))$ - умовні екстрапольована та апостеріорна ПВ безперервного компонента $\mathbf{u}(k)$ при умові $a_j(k)$; $W_j^*(k) = P(a_j(k) / \mathbf{U}_{II}(k-1))$, $W_j(k) = P(a_j(k) / \mathbf{U}_{II}(k))$ - екстраполірована та апостеріорна ймовірності дискретного компонента $a_j(k)$; $P(\mathbf{u}_{II}(k) / \mathbf{u}(k))$ - умовна ПВ, яка визначається за виразами (1); $P(\mathbf{u}_{II}(k) / a_j(k), \mathbf{U}_{II}(k-1))$ - умовна ПВ, яка визначається за допомогою формули:

$$P(\mathbf{u}_{II}(k) / a_j(k), \mathbf{U}_{II}(k-1)) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P(\mathbf{u}_{II}(k) / \mathbf{u}(k)) W_j^*(\mathbf{u}(k)) d\mathbf{u}(k); \quad (2.12)$$

$P(\mathbf{u}_{II}(k) / \mathbf{U}_{II}(k-1))$ - умовна ПВ, що обчислюється на основі співвідношення:

$$P(\mathbf{u}_{II}(k) / \mathbf{U}_{II}(k-1)) = \sum_{i=1}^M P(\mathbf{u}_{II}(k) / a_j(k), \mathbf{U}_{II}(k-1)) W_j^*(k). \quad (2.13)$$

Синтезований алгоритм (2.8)-(2.11) є нелінійним. Рівняння (2.8), (2.11) використовуються для обчислення екстрапольованих $W_j^*(k)$ та апостеріорних $W_j(k)$ ймовірностей дискретних компонентів $a_j(k)$. Рівняння (2.9), (2.10)

використовуються для обчислення умовних екстрапольованих $W_j^*(\mathbf{u}(k))$ та апостеріорних $W_j(\mathbf{u}(k))$ ПВ неперервного компонента $\mathbf{u}(k)$. Особливість синтезованого алгоритму полягає у нерозривній зв'язаності рівнянь фільтрації та екстраполяції дискретного та безперервного компонентів між собою. Алгоритм фільтрації (2.8)-(2.11) визначає структуру оптимального пристрою. Оптимальний фільтр містить M каналів і відноситься до класу пристроїв зі зворотними зв'язками між каналами. Зворотні зв'язку між каналами обумовлені марківською властивістю змінної перемикавання $a_j(k)$. Наявність зворотного зв'язку дозволяє виконувати постійну корекцію каналів, налаштованих на помилкові значення змінної перемикавання $a_j(k)$ від "істинного" каналу, що запобігає їх розбіжності і забезпечує швидке розпізнавання зміни структури процесу $\mathbf{u}(k)$. У синтезованому оптимальному пристрої немає зростання числа каналів, що обумовлено марківською властивістю змішаного процесу.

Практична реалізація синтезованого оптимального алгоритму спільної фільтрації змішаного марківського процесу (2.8)-(2.11) зустрічає значні труднощі. Вони обумовлені вимогою великих обсягів пам'яті і швидкодії обчислювача при виконанні чисельного інтегрування багатовимірних щільностей ймовірностей. Тому представляє значний інтерес синтез квазіоптимальних алгоритмів фільтрації і екстраполяції.

Квазіоптимальний адаптивний алгоритм можна отримати шляхом лінеаризації рівняння вимірювання радіальної швидкості (2.8), а також з використанням гауссівської апроксимації умовних екстрапольованих ЩЙ $W_j^*(\mathbf{u}(k))$, $j = \overline{1, M}$ на кожному кроці k . При цьому квазіоптимальний алгоритм адаптивної фільтрації параметрів руху маневруючого об'єкта у прямокутній системі координат має вигляд:

$$W_j^*(k) = \sum_{i=1}^M \Pi_{ij}(k, k-1) W_i(k-1); \quad (2.14)$$

$$\mathbf{u}_j^*(k) = \sum_{i=1}^M \Pi_{ij}(k, k-1) W_i(k-1) \mathbf{F}_j(k, k-1) \hat{\mathbf{u}}_i(k-1) / W_j^*(k); \quad (2.15)$$

$$\begin{aligned} \mathbf{P}_j^*(k) = & \sum_{i=1}^M \left\{ \mathbf{F}_j(k, k-1) \hat{\mathbf{P}}_i(k-1) \mathbf{F}_j^T(k, k-1) + \mathbf{G}_j(k) \mathbf{G}_j^T(k) + \right. \\ & \left. + (\mathbf{F}_j(k, k-1) \hat{\mathbf{u}}_i(k-1) - \mathbf{u}_j^*(k)) (\mathbf{F}_j(k, k-1) \hat{\mathbf{u}}_i(k-1) - \mathbf{u}_j^*(k))^T \right\} \times \\ & \times \Pi_{ij}^\Gamma(k, k-1) W_i(k-1) / W_j^*(k); \end{aligned} \quad (2.16)$$

$$\hat{\mathbf{u}}_j(k) = \mathbf{u}_j^*(k) + \mathbf{K}_j(k) (\mathbf{u}_{\text{И}}(k) - \mathbf{h}(\mathbf{u}_j^*(k))); \quad (2.17)$$

$$\mathbf{K}_j(k) = \mathbf{P}_j^*(k) \mathbf{H}_j^T(k) (\mathbf{H}_j(k) \mathbf{P}_j^*(k) \mathbf{H}_j^T(k) + \mathbf{R}(k))^{-1}; \quad (2.18)$$

$$\hat{\mathbf{P}}_j(k) = \mathbf{P}_j^*(k) - \mathbf{K}_j(k) \mathbf{H}_j(k) \mathbf{P}_j^*(k); \quad (2.19)$$

$$W_j(k) = P(\mathbf{u}_{\text{И}}(k) / a_j(k), \mathbf{U}_{\text{И}}(k-1)) W_j^*(k) / P(\mathbf{u}_{\text{И}}(k) / \mathbf{U}_{\text{И}}(k-1)), \quad (2.20)$$

де $\mathbf{u}_j^*(k)$, $\hat{\mathbf{u}}_j(k)$, $\mathbf{P}_j^*(k)$, $\hat{\mathbf{P}}_j(k)$ - математичні очікування та кореляційні матриці умовних ПВ $W_j^*(\mathbf{u}(k))$, $W_j(\mathbf{u}(k))$; $\mathbf{K}_j(k)$ - коефіцієнт підсилення j -го каналу квазіоптимального пристрою; $\mathbf{H}_j(k) = \nabla_{\mathbf{u}(k)} \mathbf{h}(\mathbf{u}^*(k))$ - матриця Якобі векторнозначної функції $\mathbf{h}(\mathbf{u}^*(k))$, яка має вигляд:

$$\mathbf{H}_j(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{\dot{x}_j^* r_j^* - \dot{r}_j^* x_j^*}{r_j^{*2}} & \frac{x_j^*}{r_j^*} & 0 & \frac{\dot{y}_j^* r_j^* - \dot{r}_j^* y_j^*}{r_j^{*2}} & \frac{y_j^*}{r_j^*} & 0 & \frac{\dot{z}_j^* r_j^* - \dot{r}_j^* z_j^*}{r_j^{*2}} & \frac{z_j^*}{r_j^*} & 0 \end{bmatrix} \quad (2.21)$$

Умовна ЩЙ $P(\mathbf{u}_{\text{И}}(k) / a_j(k), \mathbf{U}_{\text{И}}(k-1))$ - визначається за формулою:

$$P(\mathbf{u}_{\text{И}}(k) / a_j(k), \mathbf{U}_{\text{И}}(k-1)) = (2\pi)^{-1} \det(\mathbf{D}_j(k))^{-1/2} \exp\{-0.5 \|\mathbf{u}_{\text{И}}(k) - \mathbf{h}(\mathbf{u}_j^*(k))\|_{\mathbf{D}_j(k)}\},$$

2.22(3.49)

де $\mathbf{D}_j(k) = \mathbf{H}_j(k) \mathbf{P}_j^*(k) \mathbf{H}_j^T(k) + \mathbf{R}(k)$.

Квазіоптимальний адаптивний алгоритм (2.14)-(2.20) часто розходиться на початковій ділянці його роботи, що обумовлено нелінійним характером рівняння вимірювання радіальної швидкості. Для усунення ефекту розбіжності запропоновано збільшувати дисперсію помилки вимірювання

радіальної швидкості на величину, пропорційну дисперсії помилки екстраполяції радіальної швидкості $p_{\dot{r},k}^*$ по формулі

$$R'_{44,j,k} = R_{44,j,k} + \gamma p_{\dot{r},j,k}^*, \quad (2.23)$$

де γ - коефіцієнт, що регулює величину додаткової складової дисперсії помилки вимірювання, що вноситься; $p_{\dot{r},j,k}^*$ - дисперсія помилки прогнозу радіальної швидкості, яка обчислюється за формулою

$$p_{\dot{r},j,k}^* = \sum_{m=1}^9 \sum_{n=1}^9 H_{4n,j,k} p_{nm,j,k}^* H_{4m,j,k}. \quad (2.24)$$

Доцільно вважати $\gamma = 0.8$.

Квазіоптимальний алгоритм (2.14)-(2.20) є нелінійним. На відміну від оптимального алгоритму при його реалізації обчислюються лише перший і другий моменти умовних апостеріорних розподілів. При цьому апостеріорна ПВ $w(\mathbf{u}(k))$ при переході на наступний крок фільтрації апроксимується сумою M гаусівських щільностей. Синтезований квазіоптимальний фільтр також включає M каналів і відноситься до класу пристроїв із зворотними зв'язками між каналами. Він відрізняється від оптимального пристрою фільтрації тим, що в блоках прогнозу і фільтрації квазіоптимального пристрою обчислюються тільки математичні очікування $\mathbf{u}_j^*(k)$, $\hat{\mathbf{u}}_j(k)$ і кореляційні матриці $\mathbf{P}_j^*(k)$, $\hat{\mathbf{P}}_j(k)$ умовних щільностей ймовірностей $W_j^*(\mathbf{u}(k))$ і $W_j(\mathbf{u}(k))$, $j = \overline{1, M}$.

Для запуску квазіоптимального адаптивного алгоритму необхідно встановити початкові умови. За наявності двох вимірів у момент часу $k = 0$ и $k = 1$ вектори початкової оцінки однакові та мають вигляд:

$$\hat{\mathbf{u}}_j^T(1) = (x_{\text{И}}(1); \frac{x_{\text{И}}(1) - x_{\text{И}}(0)}{T}; 0; y_{\text{И}}(1); \frac{y_{\text{И}}(1) - y_{\text{И}}(0)}{T}; 0; z_{\text{И}}(1); \frac{z_{\text{И}}(1) - z_{\text{И}}(0)}{T}; 0), j = \overline{1, M}. \quad (2.25)$$

Кореляційні матриці помилок початкової оцінки різні та мають вигляд:

$$\hat{\mathbf{P}}_j(1) = \begin{bmatrix} \sigma_x^2(1) & \frac{\sigma_x^2(1)}{T} & 0 & \sigma_{xy}^2(1) & \frac{\sigma_{xy}^2(1)}{T} & 0 & \sigma_{xz}^2(1) & \frac{\sigma_{xz}^2(1)}{T} & 0 \\ \frac{\sigma_x^2(1)}{T} & \frac{\sigma_x^2(1) + \sigma_x^2(0)}{T^2} + \sigma_{jax}^2 T^2 & 0 & \frac{\sigma_{xy}^2(1)}{T} & \frac{\sigma_{xy}^2(1) + \sigma_{xy}^2(0)}{T^2} & 0 & \frac{\sigma_{xz}^2(1)}{T} & \frac{\sigma_{xz}^2(1) + \sigma_{xz}^2(0)}{T^2} & 0 \\ 0 & 0 & \sigma_{jax}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ \sigma_{xy}^2(1) & \frac{\sigma_{xy}^2(1)}{T} & 0 & \sigma_y^2(1) & \frac{\sigma_y^2(1)}{T} & 0 & \sigma_{yz}^2(1) & \frac{\sigma_{yz}^2(1)}{T} & 0 \\ \frac{\sigma_{xy}^2(1)}{T} & \frac{\sigma_{xy}^2(1) + \sigma_{xy}^2(0)}{T^2} & 0 & \frac{\sigma_y^2(1)}{T} & \frac{\sigma_y^2(1) + \sigma_y^2(0)}{T^2} + \sigma_{jay}^2 T^2 & 0 & \sigma_{yz}^2(1) & \frac{\sigma_{yz}^2(1) + \sigma_{yz}^2(0)}{T^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{jay}^2 & 0 & 0 & 0 \\ \sigma_{xz}^2(1) & \frac{\sigma_{xz}^2(1)}{T} & 0 & \sigma_{yz}^2(1) & \sigma_{yz}^2(1) & 0 & \sigma_z^2(1) & \frac{\sigma_z^2(1)}{T} & 0 \\ \frac{\sigma_{xz}^2(1)}{T} & \frac{\sigma_{xz}^2(1) + \sigma_{xz}^2(0)}{T^2} & 0 & \frac{\sigma_{yz}^2(1)}{T} & \frac{\sigma_{yz}^2(1) + \sigma_{yz}^2(0)}{T^2} & 0 & \frac{\sigma_z^2(1)}{T} & \frac{\sigma_z^2(1) + \sigma_z^2(0)}{T^2} + \sigma_{jaz}^2 T^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{jaz}^2 \end{bmatrix} \quad (2.26)$$

$j = 1, M,$

2.2 Розробка алгоритмів супроводження маневруючого МБПЛА з використанням вирішальних статистик відміток

Як зазначалося вище, супровід МБПЛА здійснюється у складній завадній обстановці, яка характеризується наявністю хибних відміток. Кількість хибних відміток, що надходять на вхід траєкторної обробки, істотно залежить як від інтенсивності завад, так і від ефективності попередньої обробки радіолокаційної інформації, що характеризується ймовірністю хибних тривоги.

При розробці алгоритмів супроводження, як правило, обмежуються розглядом вимірювань у деякій області - стробі супроводження, в яку на черговому такті з високою ймовірністю має потрапити справжній вимір. Строб супроводження є багатовимірним еліпсоїдом, розміри якого визначаються необхідною ймовірністю потрапляння до нього вектора виміру. Зазвичай практично строб супроводження представляють як паралелепіпеда у просторі вимірів. Центр строба збігається з екстраполірованим положенням цілі на даний крок виміру, а розміри залежать від помилок виміру та екстраполяції. Очевидно, що зі збільшенням розмірів строба ймовірність зриву

супроводження маневруючої цілі зменшується, проте при цьому підвищується ймовірність попадання в строб помилкових відміток, що також може призводити до зриву супроводження.

Розроблений квазіоптимальний алгоритм адаптивного оцінювання дозволяє виконати оптимізацію розмірів стробів супроводження на основі отриманих оцінок змінної перемикавання $a_j(k)$. Поточні строби визначаються з використанням характеристик помилок екстраполяції, розрахованих для моделі руху цілі, екстраполірована ймовірність якої максимальна.

Операції вторинної обробки радіолокаційної інформації, такі як стробування, ототожнення відміток та ін, найбільш просто здійснюються у сферичній СК, тому фільтровані параметри траєкторії цілі на кожному кроці оновлення інформації необхідно перераховувати з місцевої прямокутної СК в сферичну. При цьому перерахунок екстрапольованих координат цілі виконується за формулами

$$r_j^*(k) = \sqrt{x_j^*(k)^2 + y_j^*(k)^2 + z_j^*(k)^2}; \quad (2.27)$$

$$\beta_j^*(k) = \arctg\left(\frac{y_j^*(k)}{x_j^*(k)}\right); \quad (2.28)$$

$$\varepsilon_j^*(k) = \arctg\left(\frac{z_j^*(k)}{\sqrt{x_j^*(k)^2 + y_j^*(k)^2}}\right); \quad (2.29)$$

$$\dot{r}_{j,k}^* = \frac{x_{j,k}^* \dot{x}_{j,k}^* + y_{j,k}^* \dot{y}_{j,k}^* + z_{j,k}^* \dot{z}_{j,k}^*}{\sqrt{x_{j,k}^{*2} + y_{j,k}^{*2} + z_{j,k}^{*2}}}. \quad (2.30)$$

Виконуючи лінеаризацію виразів (2.27)-(2.29), можна показати, що дисперсії $p_{rj}^*(k)$, $p_{\beta j}^*(k)$, $p_{\varepsilon j}^*(k)$ помилок екстраполяції координат цілі $r(k), \beta(k), \varepsilon(k)$ у сферичній СК, визначаються за формулами:

$$p_{rj}^*(k) = \dot{r}_{xj}^{*2}(k)p_{xj}^*(k) + \dot{r}_{yj}^*(k)p_{yj}^*(k) + \dot{r}_{zj}^*(k)p_{zj}^*(k) + \\ + 2\dot{r}_{xj}^*(k)\dot{r}_{yj}^*(k)p_{xyj}^*(k) + 2\dot{r}_{xj}^*(k)\dot{r}_{zj}^*(k)p_{xzj}^*(k) + 2\dot{r}_{yj}^*(k)\dot{r}_{zj}^*(k)p_{yzj}^*(k);$$

2.31(3.58)

$$p_{\beta j}^*(k) = \left(\frac{1}{\left(1 + \left(\frac{y_j^*(k)}{x_j^*(k)}\right)^2\right)x_j^*(k)} \right)^2 \left(p_{yj}^*(k) - 2\left(\frac{y_j^*(k)}{x_j^*(k)}\right)p_{xyj}^*(k) + p_{xj}^*(k)\left(\frac{y_j^*(k)}{x_j^*(k)}\right)^2 \right);$$

2.32(3.59)

$$p_{ej}^*(k) = \dot{\varepsilon}_{xj}^{*2}(k)p_{xj}^*(k) + \dot{\varepsilon}_{yj}^{*2}(k)p_{yj}^*(k) + \dot{\varepsilon}_{zj}^{*2}(k)p_{zj}^*(k) + \\ + 2\dot{\varepsilon}_{xj}^*(k)\dot{\varepsilon}_{yj}^*(k)p_{xyj}^*(k) + 2\dot{\varepsilon}_{xj}^*(k)\dot{\varepsilon}_{zj}^*(k)p_{xzj}^*(k) + 2\dot{\varepsilon}_{yj}^*(k)\dot{\varepsilon}_{zj}^*(k)p_{yzj}^*(k),$$

2.33(3.60)

де $p_{xj}^*(k), p_{yj}^*(k), p_{zj}^*(k), p_{xyj}^*(k), p_{xzj}^*(k), p_{yzj}^*(k)$ - елементи кореляційної матриці $\mathbf{P}_j^*(k)$; $\dot{r}_{xj}^*(k), \dot{r}_{yj}^*(k), \dot{r}_{zj}^*(k), \dot{\varepsilon}_{xj}^*(k), \dot{\varepsilon}_{yj}^*(k), \dot{\varepsilon}_{zj}^*(k)$ - визначаються за формулами:

$$\dot{r}_{xj}^*(k) = \frac{x_j^*(k)}{r_j^*(k)}; \quad \dot{r}_{yj}^*(k) = \frac{y_j^*(k)}{r_j^*(k)}; \quad \dot{r}_{zj}^*(k) = \frac{z_j^*(k)}{r_j^*(k)};$$

$$\dot{\varepsilon}_{xj}^*(k) = \frac{-z_j^*(k)x_j^*(k)}{\left(1 + \left(\frac{z_j^*(k)}{r_{xyj}^*(k)}\right)^2\right)r_{xyj}^{*3}(k)}; \quad \dot{\varepsilon}_{yj}^*(k) = \frac{-z_j^*(k)y_j^*(k)}{\left(1 + \left(\frac{z_j^*(k)}{r_{xyj}^*(k)}\right)^2\right)r_{xyj}^{*3}(k)};$$

$$\dot{\varepsilon}_{zj}^*(k) = \frac{1}{\left(1 + \left(\frac{z_j^*(k)}{r_{xyj}^*(k)}\right)^2\right)r_{xyj}^*(k)}; \quad r_{xyj}^*(k) = \sqrt{x_j^*(k)^2 + y_j^*(k)^2}.$$

Дисперсія помилки прогнозу радіальної швидкості $p_{ij}^*(k)$ обчислюється за формулою (2.24).

При супроводженні цілі в умовах пасивних перешкод, при попаданні в строб супроводження поряд з істинною хибних відміток, вирішується завдання відбору позначки для оновлення даних траєкторії.

Перевірка на потрапляння в строб супроводження l -й відмітки $(r_H^l(k), \dot{r}_H^l(k), \beta_H^l(k), \varepsilon_H^l(k))$ виконується шляхом одночасної перевірки наступних умов:

$$\begin{cases} \left| r_H^l(k) - r_j^*(k) \right| \leq \alpha \sqrt{p_{rj}^*(k) + \sigma_r^2} \\ \left| \beta_H^l(k) - \beta_j^*(k) \right| \leq \alpha \sqrt{p_{\beta j}^*(k) + \sigma_\beta^2} \\ \left| \varepsilon_H^l(k) - \varepsilon_j^*(k) \right| \leq \alpha \sqrt{p_{\varepsilon j}^*(k) + \sigma_\varepsilon^2} \\ \left| \dot{r}_H^l(k) - \dot{r}_j^*(k) \right| \leq \alpha \sqrt{p_{\dot{r}j}^*(k) + \sigma_{\dot{r}}^2} \end{cases}, \quad (2.34)$$

Параметр α вибирається виходячи з необхідної ймовірності потрапляння відмітки в строб супроводження. На практиці обирають в інтервалі $\alpha \in [3, 4]$.

Найбільше поширення на практиці отримав метод «найближчий сусід». Він полягає у відборі позначки, найближчої до центру строба як істинної. Правило відбору істинної позначки у стробі, що відповідає j -му типу руху цілі, має вигляд:

$$\min_l d_j^l(k) \quad (2.35)$$

де $d_j^l(k)$ - визначається за формулою:

$$d_j^l(k) = \frac{(r_H^l(k) - r_j^*(k))^2}{p_{rj}^*(k) + \sigma_r^2} + \frac{(\beta_H^l(k) - \beta_j^*(k))^2}{p_{\beta j}^*(k) + \sigma_\beta^2} + \frac{(\varepsilon_H^l(k) - \varepsilon_j^*(k))^2}{p_{\varepsilon j}^*(k) + \sigma_\varepsilon^2} + \frac{(\dot{r}_H^l(k) - \dot{r}_j^*(k))^2}{p_{\dot{r}j}^*(k) + \sigma_{\dot{r}}^2} \quad (2.36)$$

Істинною вважається відмітка, у якої $d_j^l(k)$ мінімально.

При вимірюванні вирішальних статистик позначок також використовується алгоритм ототожнення позначок за максимальним значенням достатньої статистики. Правило відбору істинної позначки у стробі, що відповідає j -му типу руху цілі, має вигляд:

$$\max_l z^l(k), \quad (2.37)$$

де $z^l(k)$ - виміряне значення достатньої статистики l -ї відмітки.

Поточний строб визначається за допомогою характеристик помилок екстраполяції, розрахованих для моделі руху цілі, екстрапольована ймовірність якої є максимальною. За відсутності вимірювань у поточному стробі супроводження, що може бути обумовлено зміною типу руху цілі, здійснюється її допущук у стробі супроводження з більшими розмірами, отриманими для моделі руху цілі з маневром $j = 3$. Після отримання вимірювання та його обробки, у фільтрі приймається рішення про зміну виду руху цілі та подальший супровід виконується на основі відповідних стробів.

Широке використання для оцінювання параметрів руху цілі і її супроводження при наявності хибних відміток знаходить алгоритм з ймовірнісними об'єднанням даних PDA. Він враховує при оцінюванні параметрів руху цілі всі відмітки, що потрапили в строб супроводження. З використанням методу PDA виконано модифікацію синтезованого квазіоптимального адаптивного алгоритму оцінювання параметрів руху маневруючого БПЛА в прямокутній СК (2.14)-(2.20). При цьому рівняння прогнозу параметрів руху БПЛА (2.14)-(2.16) не змінюються. Стробування виміряних відміток відбувається в сферичній СК на основі алгоритму. Координати відміток, що потрапили в строб супроводження перераховуються в прямокутну СК і формується вектор вимірювання $\mathbf{u}_I^T(k) = (\mathbf{u}_1^I(k), \dots, \mathbf{u}_{M(i)}^I(k))$, де $M(k)$ - кількість відміток, що потрапила в строб супроводження. Формується вектор $\mathbf{z}(k) = (z_1(k), \dots, z_m(k))$, що включає вирішальні статистики відміток, які отримані в стробі на k -у огляді.

Також вводяться гіпотези про походження відміток $\mathbf{u}_{\text{II}}(k)$ з вирішальними статистиками $\mathbf{z}(k)$, які потрапили в строб супроводження:

- $\gamma_m(k)$, $m = \overline{1, M_s(k)}$ - гіпотеза, яка полягає в тому, що m -а відмітка належить траєкторії, що супроводжується;

- $\gamma_0(k)$ - гіпотеза, яка полягає в тому, що всі відмітки, які потрапили в строб, є хибними.

На першому етапі розраховуються $\hat{\mathbf{u}}_{jm}(k), \tilde{\mathbf{P}}_j(k)$ - математичні очікування і кореляційні матриці умовних щільностей ймовірностей $W_{jm}(\mathbf{u}(k)) = P(\mathbf{u}(k) / a_j(k), \gamma_m(k), \mathbf{U}_{\text{II}}(k), \mathbf{Z}(k))$, $j = \overline{1, M}$, $m = \overline{1, M_s(k)}$ по формулам:

$$\hat{\mathbf{u}}_{jm}(k) = \mathbf{u}_j^*(k) + \mathbf{K}_{jm}(k)(\mathbf{u}_m^{\text{II}}(k) - \mathbf{h}(\mathbf{u}_j^*(k))); \quad (2.38)$$

$$\mathbf{K}_j(k) = \mathbf{P}_j^*(k) \mathbf{H}_j^T(k) (\mathbf{H}_j(k) \mathbf{P}_j^*(k) \mathbf{H}_j^T(k) + \mathbf{R}_j(k))^{-1}; \quad (2.39)$$

$$\tilde{\mathbf{P}}_j(k) = \mathbf{P}_j^*(k) - \mathbf{K}_j(k) \mathbf{H}_j(k) \mathbf{P}_j^*(k); \quad (2.40)$$

Як випливає з формул (3.64), (3.65) кореляційні матриці $\tilde{\mathbf{P}}_j(k)$ не залежать від гіпотез $\gamma_0(k)$, $\gamma_m(k)$, $m = \overline{1, M_s(k)}$.

Наступним кроком є обчислення умовних апостеріорних імовірностей $W_{m|j}(k) = P(\gamma_m(k) | a_j(k), \mathbf{U}_{\text{II}}(k), \mathbf{Z}(k))$ гіпотез $\gamma_0(k)$, $\gamma_m(k)$, $m = \overline{1, M_s(k)}$ при умові, що має місце відповідний тип руху $a_j(k)$, $m = \overline{1, M}$. Вони розраховуються на основі формули Байєса у вигляді:

$$W_{m|j}(k) = \begin{cases} \frac{f(\mathbf{u}^{\text{v}}(k), \mathbf{z}(k)) | a_j(k), \gamma_0(k), \mathbf{U}^{\text{v}}(k-1), \mathbf{Z}(k-1)) P(\gamma_0(k))}{f(\mathbf{u}^{\text{v}}(k), \mathbf{z}(k)) | a_j(k), \mathbf{U}^{\text{v}}(k-1), \mathbf{Z}(k-1))}, & m = 0, \\ \frac{f(\mathbf{u}^{\text{v}}(k), \mathbf{z}(k)) | a_j(k), \gamma_m(k), \mathbf{U}^{\text{v}}(k-1), \mathbf{Z}(k-1)) P(\gamma_m(k))}{f(\mathbf{u}^{\text{v}}(k), \mathbf{z}(k)) | a_j(k), \mathbf{U}^{\text{v}}(k-1), \mathbf{Z}(k-1))}, & m = \overline{1, M_s(k)}. \end{cases} \quad (2.41)$$

де умовна щільність ймовірності $f(\mathbf{u}^v(k), \mathbf{z}(k)) | a_j(k), \mathbf{U}^v(k-1), \mathbf{Z}(k-1))$ визначається по формулі:

$$\begin{aligned} f(\mathbf{u}^v(k), \mathbf{z}(k)) | a_j(k), \mathbf{U}^v(k-1), \mathbf{Z}(k-1)) = \\ = \sum_{l=0}^{M(k)} f(\mathbf{u}^v(k), \mathbf{z}(k)) | a_j(k), \gamma_l(k), \mathbf{U}^v(k-1), \mathbf{Z}(k-1)) P(\gamma_l(k)) \end{aligned} \quad (2.42)$$

Визначимо складові, що входять у вираз (3.66). Умовна щільність ймовірності $f(\mathbf{u}^v(k), \mathbf{z}(k)) | a_j(k), \gamma_0(k), \mathbf{U}^v(k-1), \mathbf{Z}(k-1))$ має вигляд:

$$f(\mathbf{u}^v(k), \mathbf{z}(k)) | a_j(k), \gamma_0(k), \mathbf{U}^v(k-1), \mathbf{Z}(k-1)) = V_{\text{стр}}^{-M(k)} \prod_{m=1}^{M(k)} f_N^y(z_m(k)). \quad (2.43)$$

де $V(k)$ - об'єм строба супроводження.

Апріорна ймовірність $P(\gamma_0(k))$ розраховується по формулі:

$$P(\gamma_0(k)) = (1 - D_{\text{вх}} P_{\text{стр}}). \quad (2.44)$$

де $D_{\text{вх}}$ - ймовірність первинного виявлення цільової відмітки; $P_{\text{стр}}$ - ймовірність попадання цільової відмітки в строб супроводження.

Умовні щільності ймовірності $f(\mathbf{u}^v(k), \mathbf{z}(k)) | a_j(k), \gamma_m(k), \mathbf{U}^v(k-1), \mathbf{Z}(k-1))$, $m = \overline{1, M_s(k)}$ мають вигляд:

$$\begin{aligned} f(\mathbf{u}^v(k), \mathbf{z}(k)) | a_j(k), \gamma_m(k), \mathbf{U}^v(k-1), \mathbf{Z}(k-1)) = \\ = f(\mathbf{u}_m^v(k) | a_j(k), \gamma_m(k), \mathbf{U}^v(k-1), \mathbf{Z}(k-1)) V_{\text{стр}}^{-M_k+1} f_S^y(z_m(k)) \prod_{j=1, j \neq m}^{M(k)} f_N^y(z_j(k)) \end{aligned} \quad (2.45)$$

де $f(\mathbf{u}_m^v(k) | a_j(k), \gamma_m(k), \mathbf{U}^v(k-1), \mathbf{Z}(k-1))$ умовна щільність ймовірності, яка має вигляд:

$$f(\mathbf{u}_m^v(k) | a_j(k), \gamma_m(k), \mathbf{U}^v(k-1), \mathbf{Z}(k-1)) = \frac{N(\mathbf{u}_m^v(k), \mathbf{h}(\mathbf{u}_j^*(k)), \mathbf{D}_j(k))}{P_{\text{стр}}};$$

$\mathbf{D}_j(k)$ - кореляційна матриця нев'язки вимірювання $\mathbf{u}_m^v(k)$ при умові, що має місце j -й тип руху БПЛА, яка розраховується по формулі:

$$\mathbf{D}_j(k) = \mathbf{H}_j(k) \mathbf{P}_j^*(k) \mathbf{H}_j^T(k) + \mathbf{R}_j(k).$$

Апріорні ймовірності $P(\gamma_m(k))$, $m = \overline{1, M(k)}$ розраховується по формулі:

$$P(\gamma_m(k)) = \frac{D_{\text{вх}} P_{\text{стр}}}{M(k)}. \quad (2.46)$$

Підставивши формули (3.68)-(3.71) в вираз (3.66) отримаємо:

$$W_{m|j}(k) = \begin{cases} \frac{b(k)}{b(k) + \sum_{m=1}^{M_k} e_{jm}(k) \lambda_m(k)}, & m = 0, \\ \frac{e_{jm}(k) \lambda_m(k)}{b(k) + \sum_{m=1}^{M_k} e_{jm}(k) \lambda_m(k)}, & m = \overline{1, M_s(k)}, \end{cases} \quad (2.47)$$

де $b(k)$, $e_{jm}(k)$, $\lambda_m(k)$ розраховуються за формулами:

$$b(k) = M(k) \frac{1 - D_{\text{вх}} P_{\text{стр}}}{D_{\text{вх}} P_{\text{стр}} V(k)}; \quad e_{jm}(k) = \frac{N(\mathbf{u}_{\text{И}}^m(k), \mathbf{h}(\mathbf{u}_j^*(k)), \mathbf{D}_j(k))}{P_{\text{стр}}}$$

$$\lambda_m(k) = \frac{f_S^y(z_m(k))}{f_N^y(z_m(k))};$$

Наступним кроком є осереднення параметрів руху БПЛА по отриманим вимірюванням при умові що має місце відповідний тип руху $a_j(k)$, $m = \overline{1, M}$

$$\hat{\mathbf{u}}_j(k) = \mathbf{u}_j^*(k) W_{0|j}(k) + \sum_{m=1}^{M(k)} \hat{\mathbf{u}}_{jm}(k) W_{m|j}(k); \quad (2.48)$$

$$\hat{\mathbf{P}}_j(k) = \{\mathbf{P}_j^*(k) + (\mathbf{u}_j^*(k) - \hat{\mathbf{u}}_j(k))(\mathbf{u}_j^*(k) - \hat{\mathbf{u}}_j(k))^T\} W_{0|j}(k) + \sum_{m=1}^{M(k)} \{\tilde{\mathbf{P}}_j(k) + (\hat{\mathbf{u}}_{jm}(k) - \hat{\mathbf{u}}_j(k))(\hat{\mathbf{u}}_{jm}(k) - \hat{\mathbf{u}}_j(k))^T\} W_{m|j}(k) \quad (2.49)$$

Апостеріорні ймовірності $W_j(k) = P(a_j(k) | \mathbf{U}_{\text{И}}(k), \mathbf{Z}(k))$ того, що має місце руху БПЛА $a_j(k)$, $m = \overline{1, M}$ розраховується по формулі

$$W_j(k) = P(\mathbf{u}_И(k), \mathbf{z}(k) | a_j(k), \mathbf{U}_И(k-1), \mathbf{Z}(k-1))W_j^*(k) / \quad (2.50)$$

$$/ P(\mathbf{u}_И(k), \mathbf{z}(k) | \mathbf{U}_И(k-1), \mathbf{Z}(k-1))$$

$f(\mathbf{u}^v(k), \mathbf{z}(k) | a_j(k), \mathbf{U}^v(k-1), \mathbf{Z}(k-1))$ - функція правдоподібності вимірювань $\mathbf{u}_И(k), \mathbf{z}(k)$, що визначається по формулі

$$f(\mathbf{u}^v(k), \mathbf{z}(k) | a_j(k), \mathbf{U}^v(k-1), \mathbf{Z}(k-1)) =$$

$$= (b(k) + \sum_{m=1}^{M(k)} e_{jm}(k)\lambda_m(k)) \frac{D_{\text{вх}} P_{\text{стр}} V_{\text{стр}}^{-M_k+1}}{M(k)} \prod_{m=1}^{M(k)} f_N^y(z_m(k)) \quad (2.51)$$

$P(\mathbf{u}_И(k), \mathbf{z}(k) | \mathbf{U}_И(k-1), \mathbf{Z}(k-1))$ - умовна щільність ймовірності вимірювань $\mathbf{u}_И(k), \mathbf{z}(k)$, яка після їх отримання виконує функцію нормуючого множника і розраховується по формулі

$$P(\mathbf{u}_И(k), \mathbf{z}(k) | \mathbf{U}_И(k-1), \mathbf{Z}(k-1)) = \sum_{j=1}^M P(\mathbf{u}_И(k), \mathbf{z}(k) | a_j(k), \mathbf{U}_И(k-1), \mathbf{Z}(k-1))W_j^*(k). \quad (2.52)$$

З врахування наведеного, в цілому алгоритм уточнення прогнозованих параметрів руху маневруючого БПЛА в якому відбувається ймовірнісне отожднення відміток, що потрапили в строб супроводження на основі методу PDA описується рівняннями

$$\hat{\mathbf{u}}_{jm}(k) = \mathbf{u}_j^*(k) + \mathbf{K}_{jm}(k)(\mathbf{u}_m^И(k) - \mathbf{h}(\mathbf{u}_j^*(k))); \quad (2.53)$$

$$\mathbf{K}_j(k) = \mathbf{P}_j^*(k)\mathbf{H}_j^T(k)(\mathbf{H}_j(k)\mathbf{P}_j^*(k)\mathbf{H}_j^T(k) + \mathbf{R}_j(k))^{-1}; \quad (2.54)$$

$$\tilde{\mathbf{P}}_j(k) = \mathbf{P}_j^*(k) - \mathbf{K}_j(k)\mathbf{H}_j(k)\mathbf{P}_j^*(k); \quad (2.55)$$

$$W_{m|j}(k) = \begin{cases} \frac{b(k)}{b(k) + \sum_{m=1}^{M_k} e_{jm}(k)\lambda_m(k)}, & m = 0, \\ \frac{e_{jm}(k)\lambda_m(k)}{b(k) + \sum_{m=1}^{M_k} e_{jm}(k)\lambda_m(k)}, & m = \overline{1, M_s(k)}, \end{cases} \quad (2.56)$$

$$\hat{\mathbf{u}}_j(k) = \mathbf{u}_j^*(k)W_{0|j}(k) + \sum_{m=1}^{M(k)} \hat{\mathbf{u}}_{jm}(k)W_{m|j}(k); \quad (2.57)$$

$$\begin{aligned} \hat{\mathbf{P}}_j(k) = & \{\mathbf{P}_j^*(k) + (\mathbf{u}_j^*(k) - \hat{\mathbf{u}}_j(k))(\mathbf{u}_j^*(k) - \hat{\mathbf{u}}_j(k))^T\} W_{0|j}(k) + \\ & + \sum_{m=1}^{M(k)} \{\tilde{\mathbf{P}}_j(k) + (\hat{\mathbf{u}}_{jm}(k) - \hat{\mathbf{u}}_j(k))(\hat{\mathbf{u}}_{jm}(k) - \hat{\mathbf{u}}_j(k))^T\} W_{m|j}(k) \end{aligned} \quad (2.58)$$

$$W_j(k) = \frac{P(\mathbf{u}_{\text{И}}(k), \mathbf{z}(k) | a_j(k), \mathbf{U}_{\text{И}}(k-1), \mathbf{Z}(k-1))W_j^*(k)}{P(\mathbf{u}_{\text{И}}(k), \mathbf{z}(k) | \mathbf{U}_{\text{И}}(k-1), \mathbf{Z}(k-1))}, \quad (2.59)$$

Як впливає з формул (2.53) (2.59), при визначені осереднених апостеріорних характеристик $\hat{\mathbf{u}}_j(k), \hat{\mathbf{P}}_j(k)$ при справедливості гіпотези $\gamma_0(k)$ використовуються екстрапольовані характеристики

Для проведення моделювання представляє інтерес встановити зв'язок між ймовірністю помилкової тривоги F та просторовою щільністю відміток λ . Кількість відміток у сфері огляду підпорядковується закону розподілу Пуассона з просторовою щільністю хибних позначок λ . Координати хибних позначок області огляду мають рівномірний закон розподілу. Алгоритми виявлення сигналів, як правило, характеризуються ймовірностями помилкової тривоги F . Тому представляє інтерес встановити зв'язок між ймовірністю помилкової тривоги F та просторовою щільністю відміток λ .

Ймовірність появи m помилкових відміток відповідно до закону Пуассона можна подати у вигляді

$$p(m) = e^{-\lambda V} \frac{(\lambda V)^m}{m!} = e^{-FU} \frac{(FU)^m}{m!}, \quad (2.60)$$

де λ - просторова щільність хибних відміток (середнє число в одиниці об'єму); V - об'єм строби супроводження; U - кількість елементів розрізнення у стробі супроводження.

$$U = \frac{V}{\delta r \delta \dot{r} \delta b \delta e}; \quad (2.61)$$

F - мовірність помилкової тривоги в елементі розрізнення.

З (2.60) середня кількість помилкових відміток у стробі супроводження визначається за формулою:

$$\lambda V = F U. \quad (2.62)$$

Зв'язок між дисперсіями помилок вимірювання та розмірами елементів розрізнення

$$\sigma_r^2 = \frac{\delta r^2}{12}; \quad \sigma_{\dot{r}}^2 = \frac{\delta \dot{r}^2}{12}; \quad \sigma_b^2 = \frac{\delta b^2}{12}; \quad \sigma_e^2 = \frac{\delta e^2}{12}. \quad (2.63)$$

Відповідно

$$\delta r = 2\sqrt{3}\sigma_r = 3.465\sigma_r; \quad \delta \dot{r} = 2\sqrt{3}\sigma_{\dot{r}}; \quad \delta b = 2\sqrt{3}\sigma_b; \quad \delta e = 2\sqrt{3}\sigma_e.$$

2.64(3.74)

З (2.62) з урахуванням (2.61), (2.64) отримаємо:

$$\lambda = F \frac{1}{\delta r \delta \dot{r} \delta b \delta e} = F \frac{1}{144\sigma_r \sigma_{\dot{r}} \sigma_b \sigma_e}. \quad (2.65)$$

Відповідно ймовірність помилкової тривоги в елементі дозволу F визначається за формулою

$$F = 144\lambda\sigma_r\sigma_{\dot{r}}\sigma_b\sigma_e. \quad (2.66)$$

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АЛГОРИТМУ АДАПТИВНОГО СУПРОВОДЖЕННЯ БПЛА ЗА ДАНИМИ РЛС З ВИКОРИСТАННЯМ ВИРІШАЛЬНИХ СТАТИСТИК ВІДМІТОК

3.1 Розробка модуля генерації траєкторії з використанням waypointTrajectory

Об'єкт WaypointTrajectory System™ генерує траєкторії, використовуючи задані маршрутні точки. Коли ви створюєте системний об'єкт, ви можете додатково вказати час прибуття, швидкість і орієнтацію в кожній маршрутній точці.

Системний об'єкт waypointTrajectory визначає траєкторію, яка плавно проходить через маршрутні точки. Траєкторія з'єднує маршрутні точки за допомогою інтерполяції, яка припускає, що напрямок сили тяжіння, виражений у системі відліку траєкторії, є постійним. Траєкторія передбачає, що система відліку є фіксованою системою північно-схід-вниз (NED, англ. North-East-Down) або схід-північ-вгору (ENU, англ. East-North-Up). Оскільки інтерполяція траєкторії передбачає, що гравітаційне прискорення, виражене в системі відліку траєкторії, є постійним, waypointTrajectory зазвичай використовується для траєкторії, визначеної в межах області, яка охоплює лише десятки чи сотні кілометрів.

Плоский шлях траєкторії (проекція на площину x-y) складається з кускових кривих клотоїди. Кривизна кривої між двома послідовними маршрутними точками змінюється лінійно залежно від довжини кривої між ними. Напрямок дотичної до шляху в кожній точці маршруту вибирається для мінімізації розривів у кривизні, якщо курс не вказано явно через властивість Course або неявно через властивість Velocity. Після встановлення шляху об'єкт використовує кубічну інтерполяцію Ерміта, щоб обчислити місцезнаходження об'єкта на всьому шляху як функцію часу та пройденої відстані.

Нормальний компонент (z-компонент) траєкторії згодом вибирається так, щоб задовольнити кусковий сплайн зі збереженням форми (РСНІР), якщо шви-

дкість підйому не вказана явно за допомогою властивості ClimbRate або третього стовпця властивості Velocity. Виберіть знак швидкості підйому на основі вибраного ReferenceFrame:

- Коли вибрано опорну систему «ENU», визначення позитивної швидкості підйому призводить до збільшення значення z.
- Коли вибрано опорну систему «NED», визначення позитивної швидкості підйому призводить до зменшення значення z.

Ви можете визначити орієнтацію об'єкта на шляху двома основними способами:

- Якщо вказано властивість Orientation, тоді об'єкт використовує кусково-кубічний кватерніонний сплайн для обчислення орієнтації вздовж шляху як функції часу.
- Якщо властивість Orientation не вказано, поворот об'єкту завжди вирівнюється з траєкторією. Тоді крен і тангаж регулюються значеннями властивостей AutoBank і AutoPitch відповідно.

На рис. 3.1 наведено скріншот генерації траєкторії. Колонки X, Y, Z відповідають за положення точки у відповідних координатах. У колонці TimeofArrival вказують час, який пройшов з початку (в секундах), Vx, Vy, Vz – швидкість цілі в конкретний час (TimeofArrival) у координатах X, Y, Z відповідно. Для коректної роботи останніх трьох колонок потрібно поставити прапорець біля Use Velocities.

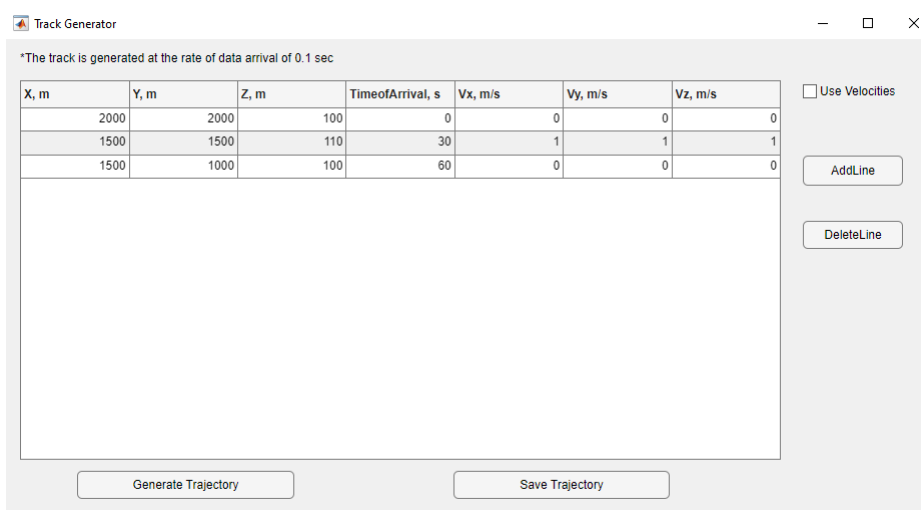


Рисунок 3.1 — Генератор траєкторії

На рис. 3.2 показано приклад згенерованої траєкторії.

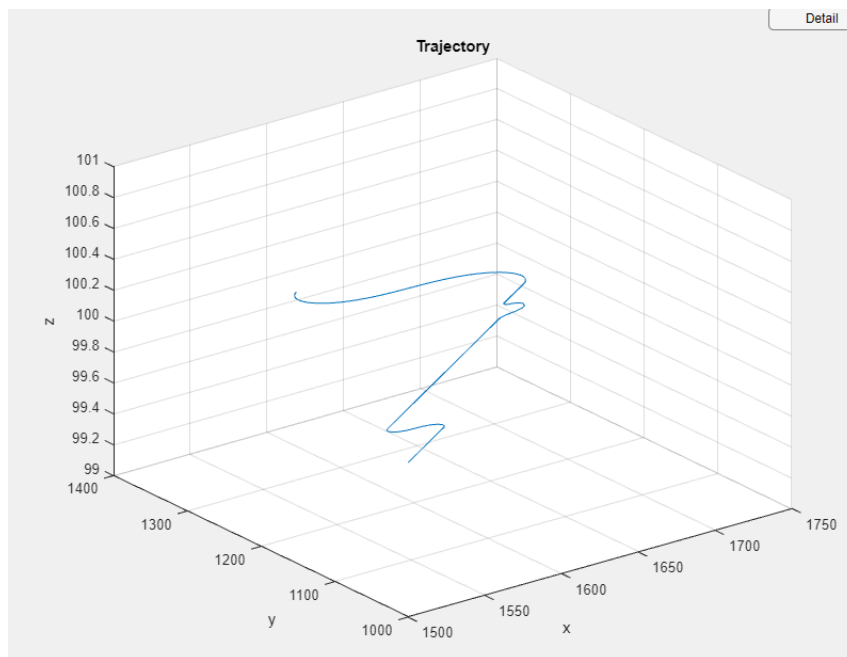


Рисунок 3.2 — Згенерована траєкторія руху в СК XYZ.

На рис. 113 показано залежність зміни параметрів руху цілі по осях XY.

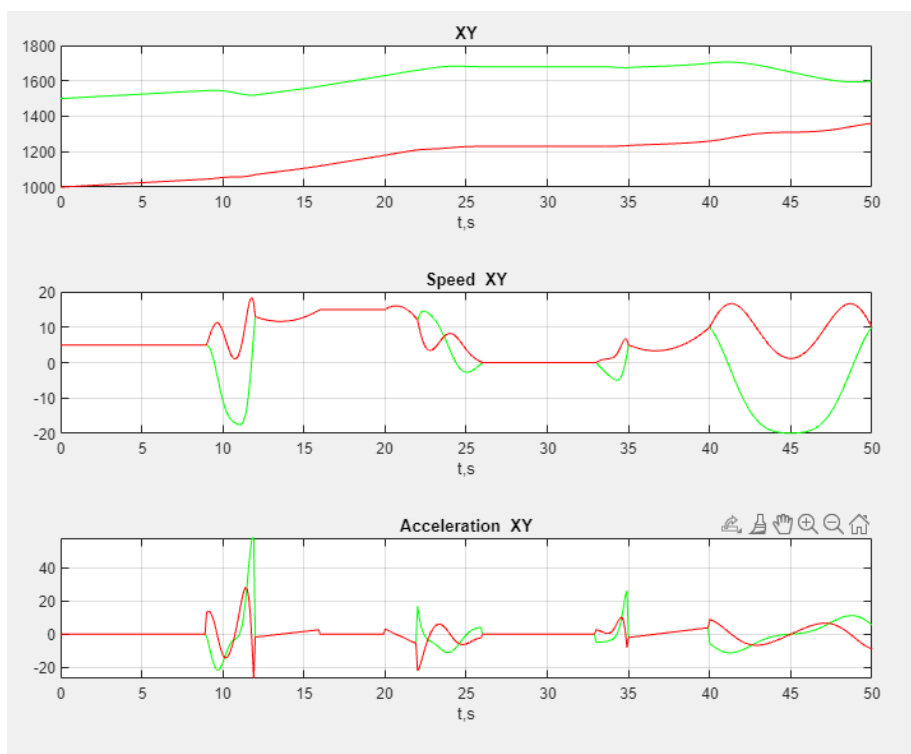


Рисунок 3.3 — залежність зміни параметрів руху цілі по осях XY

3.2 Реалізація адаптивного алгоритму супроводження з ймовірнісним об'єднанням даних на мові MATLAB

Код програми, що реалізовує адаптивний алгоритм супроводження з ймовірнісним об'єднанням даних:

```
classdef ImmAiFilter < BaseFilter
    % The parametric PDA with the Poisson clutter model with spatial density lambda
    properties (Access = protected)
        Ax1;
        Ay1;
        Az1;

        Ax2;
        Ay2;
        Az2;

        Ax3;
        Ay3;
        Az3;

        pr;
        pb;
        pe;
        pvr;

        Uzi;      % prognoz xyz  and vr

        NumHip;
        ks;
        Order;

        Wp;
        Wz;      % predict probability
        PP;

        Pd;      %detection probability
        Pc;      % in gate probability

        Flt;     % False alarm Prob
        noiseModel;
        targetModel;
        optHyp; % optimal Hypothesis
        optHypz;% optimal predict Hypothesis
    end

    methods (Access = public)

        function o = ImmAiFilter(Ax1, Ay1, Az1, Ax2, Ay2, Az2, Ax3, Ay3, Az3, ppp)
            % Call BaseFilter constructor
            o = o@BaseFilter();

            o.Ax1 = Ax1;
            o.Ay1 = Ay1;
```

```

o.Az1 = Az1;

o.Ax2 = Ax2;
o.Ay2 = Ay2;
o.Az2 = Az2;

o.Ax3 = Ax3;
o.Ay3 = Ay3;
o.Az3 = Az3;

o.NumHip = 3;
o.Order = 3;
NumRows = o.Order* o.Order;

o.F = zeros(NumRows, NumRows, o.NumHip);
o.G = zeros(NumRows, o.Order, o.NumHip);
o.Q = zeros(NumRows, NumRows, o.NumHip);

o.Uo = zeros(NumRows, o.NumHip);
o.Po = zeros(NumRows, NumRows, o.NumHip);

%transition probability matrix of a Markov chain
o.PP = [      ppp      0.5*(1-ppp)      0.5*(1-ppp);
        0.5*(1-ppp)      ppp      0.5*(1-ppp);
        0.5*(1-ppp)      0.5*(1-ppp)      ppp ];

%o.H = zeros(4,9);
o.R = zeros(4, 4, o.NumHip);

o.Uzi = zeros( 4, o.NumHip);

o.Pc = 0.994;

o.isMultiModelFilter = true;
end

function o = init(o, RadarMeasureErrors)

T = RadarMeasureErrors.tempo;
[o.pr, o.pb, o.pe, o.pvr] = RadarMeasureErrors.getMErrors();
% init R then Po

o.initF(T);
o.initQ(T);

% first 2 steps for init Filter
%[r, b, e, vr] = Trajectory.getSphericalPosition(1);
Zm1 = RadarMeasureErrors.getMeasurement(1);
Y1 = convertPolToRec(Zm1);
xi1 = Y1(1);
yi1 = Y1(2);
zi1 = Y1(3);
%vri1 = Y1(4);
%[r, b, e, vr] = Trajectory.getSphericalPosition(2);
Zm2 = RadarMeasureErrors.getMeasurement(2);
Y2 = convertPolToRec(Zm2);
xi2 = Y2(1);
yi2 = Y2(2);

```

```

zi2 = Y2(3);
%vri2 = Y2(4);

%o.initUo();
Uo2 = [xi2; (xi2-xi1)/T; 0; yi2; (yi2-yi1)/T; 0; zi2; (zi2-zi1)/T; 0];
Uo1 = [xi2; 0; 0; yi2; 0; 0; zi2; 0; 0];
o.Uo(:, 1) = Uo1;
o.Uo(:, 2) = Uo2;
o.Uo(:, 3) = Uo2;

o.Zm = Zm2;
o.initR();

o.initPo(T);

% target moves
o.Wp = [0.33 0.34 0.33];

o.targetModel = RadarMeasureErrors.targetAmplModel;
o.noiseModel = RadarMeasureErrors.noiseAmplModel;

%todo rewrite
o.Flt = RadarMeasureErrors.noiseAmplModel.getProbIntegr();
o.Pd = RadarMeasureErrors.getDetectionProb();
end

function res = isImmFilter(o)
res = o.isMultiModelFilter;
end

function initF(o, T)
o.F(:, :, 3) = [1 T T^2/2 0 0 0      0 0 0;
               0 1 T      0 0 0      0 0 0;
               0 0 1      0 0 0      0 0 0;
               0 0 0      1 T T^2/2 0 0 0;
               0 0 0      0 1 T      0 0 0;
               0 0 0      0 0 1      0 0 0;
               0 0 0      0 0 0      1 T T^2/2;
               0 0 0      0 0 0      0 1 T;
               0 0 0      0 0 0      0 0 1];
o.F(:, :, 2) = [1 T 0      0 0 0      0 0 0;
               0 1 0      0 0 0      0 0 0;
               0 0 0      0 0 0      0 0 0;
               0 0 0      1 T 0      0 0 0;
               0 0 0      0 1 0      0 0 0;
               0 0 0      0 0 0      0 0 0;
               0 0 0      0 0 0      1 T 0;
               0 0 0      0 0 0      0 1 0;
               0 0 0      0 0 0      0 0 0];
o.F(:, :, 1) = [1 0 0      0 0 0      0 0 0;
               0 0 0      0 0 0      0 0 0;
               0 0 0      0 0 0      0 0 0;
               0 0 0      1 0 0      0 0 0;
               0 0 0      0 0 0      0 0 0;
               0 0 0      0 0 0      0 0 0;
               0 0 0      0 0 0      1 0 0;
               0 0 0      0 0 0      0 0 0;
               0 0 0      0 0 0      0 0 0];

```

end

```
function initQ(o, T)
```

```
o.G(:, :, 3) = [ o.Ax3*T^3/6  0  0;
                 o.Ax3*T^2/2  0  0;
                 o.Ax3*T  0  0;
                 0  o.Ay3*T^3/6  0;
                 0  o.Ay3*T^2/2  0;
                 0  o.Ay3*T  0;
                 0  0  o.Az3*T^3/6;
                 0  0  o.Az3*T^2/2;
                 0  0  o.Az3*T  ];
```

```
o.G(:, :, 2) = [ o.Ax2*T^2/2  0  0;
                 o.Ax2*T  0  0;
                 0  0  0;
                 0  o.Ay2*T^2/2  0;
                 0  o.Ay2*T  0;
                 0  0  0;
                 0  0  o.Az2*T^2/2;
                 0  0  o.Az2*T;
                 0  0  0  ];
```

```
o.G(:, :, 1) = [ o.Ax1*T  0  0;
                 0  0  0;
                 0  0  0;
                 0  o.Ay1*T  0;
                 0  0  0;
                 0  0  0;
                 0  0  o.Az1*T;
                 0  0  0;
                 0  0  0  ];
```

```
for j = 1:o.NumHip
    o.Q(:, :, j) = o.G(:, :, j) * o.G(:, :, j)';
end
```

end

```
function [R] = initR(o)
```

```
ri = o.Zm(1);
bi = o.Zm(2);
ei = o.Zm(3);
```

```
Ry = o.pr^2 * cos(ei)^2 * sin(bi)^2 + ri^2 * o.pe^2 * sin(bi)^2 *
sin(ei)^2 + ri^2 * o.pb^2 * cos(bi)^2 * cos(ei)^2;
Rx = o.pr^2 * cos(ei)^2 * cos(bi)^2 + ri^2 * o.pe^2 * cos(bi)^2 *
sin(ei)^2 + ri^2 * o.pb^2 * sin(bi)^2 * cos(ei)^2;
Rz = o.pr^2 * sin(ei)^2 + ri^2 * o.pe^2 * cos(ei)^2;
Rxy = 0.5 * sin(2*bi) * (cos(ei)^2 * o.pr^2 - ri^2 * o.pb^2 + ri^2 *
o.pe^2 * sin(ei)^2);
Rxz = 0.5 * sin(2*ei) * cos(bi) * (o.pr^2 - ri^2 * o.pe^2);
Ryz = 0.5 * sin(2*ei) * sin(bi) * (o.pr^2 - ri^2 * o.pe^2);
```

```
for j0 = 1:o.NumHip
    o.R(:, :, j0) = [ Rx  Rxy  Rxz  0;
                    Rxy  Ry  Ryz  0;
                    Rxz  Ryz  Rz  0;
                    0  0  0  o.pvr^2  ];
```

end

```

        R = o.R;
    end

%     function initUo(o)
%
%         for j = 2:o.NumHip
%             o.Uo(:, j) = Uo2;
%         end
%     end

function initPo(o, T)
    Rx = o.R(1,1);
    Rxy = o.R(2,1);
    Ry = o.R(2,2);
    Rz = o.R(3,3);
    Ryz = o.R(3,2);
    Rxz = o.R(1,3);

    Ax = o.Ax3;
    Ay = o.Ay3;
    Az = o.Az3;

    Po = zeros(9, 9);

    Po(1,1)=Rx;      Po(1,2)=Rx/T;      Po(1,3)=0;
    Po(1,4)=Rxy;     Po(1,5)=Rxy/T;      Po(1,6)=0;      Po(1,7)=Rxz;
    Po(1,8)=Rxz/T;   Po(1,9)=0;
    Po(2,1)=Po(1,2); Po(2,2)=2*Rx/T^2+Ax^2*T^2; Po(2,3)=0;
    Po(2,4)=Rxy/T;   Po(2,5)=2*Rxy/T^2;      Po(2,6)=0;      Po(2,7)=Rxz/T;
    Po(2,8)=2*Rxz/T^2; Po(2,9)=0;
    Po(3,1)=0;      Po(3,2)=0;      Po(3,3)=Ax^2;    Po(3,4)=0;
    Po(3,5)=0;      Po(3,6)=0;      Po(3,7)=0;      Po(3,8)=0;
    Po(3,9)=0;
    Po(4,1)=Po(1,4); Po(4,2)=Po(2,4);      Po(4,3)=0;
    Po(4,4)=Ry;     Po(4,5)=Ry/T;      Po(4,6)=0;      Po(4,7)=Ryz;
    Po(4,8)=Ryz/T;   Po(4,9)=0;
    Po(5,1)=Po(1,5); Po(5,2)=Po(2,5);      Po(5,3)=0;
    Po(5,4)=Po(4,5); Po(5,5)=2*Ry/T^2+Ay^2*T^2; Po(5,6)=0;      Po(5,7)=Ryz/T;
    Po(5,8)=2*Ryz/T^2; Po(5,9)=0;
    Po(6,1)=0;      Po(6,2)=0;      Po(6,3)=0;      Po(6,4)=0;
    Po(6,5)=0;      Po(6,6)=Ay^2;      Po(6,7)=0;      Po(6,8)=0;
    Po(6,9)=0;
    Po(7,1)=Po(1,7); Po(7,2)=Po(2,7);      Po(7,3)=Po(3,7);
    Po(7,4)=Po(4,7); Po(7,5)=Po(5,7);      Po(7,6)=Po(6,7); Po(7,7)=Rz;
    Po(7,8)=Rz/T;    Po(7,9)=0;
    Po(8,1)=Po(1,8); Po(8,2)=Po(2,8);      Po(8,3)=Po(3,8);
    Po(8,4)=Po(4,8); Po(8,5)=Po(5,8);      Po(8,6)=Po(6,8); Po(8,7)=Po(7,8);
    Po(8,8)=2*Rz/T^2+Az^2*T^2; Po(8,9)=0;
    Po(9,1)=0;      Po(9,2)=0;      Po(9,3)=0;      Po(9,4)=0;
    Po(9,5)=0;      Po(9,6)=0;      Po(9,7)=0;      Po(9,8)=0;
    Po(9,9)=Az^2;

    o.Po(:, :, 3) = Po;
    %-----
    Ax = o.Ax2;
    Ay = o.Ay2;
    Az = o.Az2;

    Po = zeros(9, 9);

```

```

        Po(1,1)=Rx;      Po(1,2)=Rx/T;      Po(1,3)=0;
Po(1,4)=Rxy;      Po(1,5)=Rxy/T;      Po(1,6)=0;      Po(1,7)=Rxz;
Po(1,8)=Rxz/T;      Po(1,9)=0;
        Po(2,1)=Po(1,2); Po(2,2)=2*Rx/T^2+Ax^2*T^2; Po(2,3)=0;
Po(2,4)=Rxy/T;      Po(2,5)=2*Rxy/T^2;      Po(2,6)=0;      Po(2,7)=Rxz/T;
Po(2,8)=2*Rxz/T^2;      Po(2,9)=0;
        Po(3,1)=0;      Po(3,2)=0;      Po(3,3)=Ax^2;      Po(3,4)=0;
Po(3,5)=0;      Po(3,6)=0;      Po(3,7)=0;      Po(3,8)=0;
Po(3,9)=0;
        Po(4,1)=Po(1,4); Po(4,2)=Po(2,4);      Po(4,3)=0;
Po(4,4)=Ry;      Po(4,5)=Ry/T;      Po(4,6)=0;      Po(4,7)=Ryz;
Po(4,8)=Ryz/T;      Po(4,9)=0;
        Po(5,1)=Po(1,5); Po(5,2)=Po(2,5);      Po(5,3)=0;
Po(5,4)=Po(4,5); Po(5,5)=2*Ry/T^2+Ay^2*T^2; Po(5,6)=0;      Po(5,7)=Ryz/T;
Po(5,8)=2*Ryz/T^2;      Po(5,9)=0;
        Po(6,1)=0;      Po(6,2)=0;      Po(6,3)=0;      Po(6,4)=0;
Po(6,5)=0;      Po(6,6)=Ay^2;      Po(6,7)=0;      Po(6,8)=0;
Po(6,9)=0;
        Po(7,1)=Po(1,7); Po(7,2)=Po(2,7);      Po(7,3)=Po(3,7);
Po(7,4)=Po(4,7); Po(7,5)=Po(5,7);      Po(7,6)=Po(6,7); Po(7,7)=Rz;
Po(7,8)=Rz/T;      Po(7,9)=0;
        Po(8,1)=Po(1,8); Po(8,2)=Po(2,8);      Po(8,3)=Po(3,8);
Po(8,4)=Po(4,8); Po(8,5)=Po(5,8);      Po(8,6)=Po(6,8); Po(8,7)=Po(7,8);
Po(8,8)=2*Rz/T^2+Az^2*T^2; Po(8,9)=0;
        Po(9,1)=0;      Po(9,2)=0;      Po(9,3)=0;      Po(9,4)=0;
Po(9,5)=0;      Po(9,6)=0;      Po(9,7)=0;      Po(9,8)=0;
Po(9,9)=Az^2;

```

```
o.Po(:, :, 2) = Po;
```

```
%-----
```

```
Ax = o.Ax1;
```

```
Ay = o.Ay1;
```

```
Az = o.Az1;
```

```
Po = zeros(9, 9);
```

```
Po(1,1)=Rx;
```

```
Po(4,4)=Ry;
```

```
Po(7,7)=Rz;
```

```
o.Po(:, :, 1) = Po;
```

```

%         for j = 1:o.NumHip
%             o.Po(:, :, j) = Po;
%         end
end

```

```
function inith(o)
o.H = zeros(4,9, o.NumHip);
```

```
rz = zeros(o.NumHip);
```

```
vrz = zeros(o.NumHip);
```

```

for j0 = 1 : o.NumHip
o.H(:, :, j0) = zeros(4, 9);
o.H(1,1, j0) = 1;
o.H(2,4, j0) = 1;
o.H(3,7, j0) = 1;

```

```

        rz(j0) = sqrt(o.Uz(1,j0)^2 + o.Uz(4,j0)^2 + o.Uz(7,j0)^2);
        vrz(j0) = (o.Uz(1,j0)*o.Uz(2,j0) + o.Uz(4,j0)*o.Uz(5,j0) +
o.Uz(7,j0)*o.Uz(8,j0)) / rz(j0);

        o.H(4,1,j0) = (o.Uz(2,j0)*rz(j0) - vrz(j0)*o.Uz(1,j0)) / rz(j0)^2;
        o.H(4,2,j0) = o.Uz(1,j0) / rz(j0);

        o.H(4,4,j0) = (o.Uz(5,j0)*rz(j0) - vrz(j0)*o.Uz(4,j0)) / rz(j0)^2;
        o.H(4,5,j0) = o.Uz(4,j0) / rz(j0);

        o.H(4,7,j0) = (o.Uz(8,j0)*rz(j0) - vrz(j0)*o.Uz(7,j0)) / rz(j0)^2;
        o.H(4,8,j0) = o.Uz(7,j0) / rz(j0);

        o.Uzi(:,j0) = [ o.Uz(1,j0), o.Uz(4,j0), o.Uz(7,j0), vrz(j0) ];
    end
end

function [] = updateR(o)

    Rnew = zeros(4,4,o.NumHip);

    for j0 = 1 : o.NumHip
        Rnew(:,j0) = o.H(:,j0) * o.Pz(:,j0) * o.H(:,j0)';

        %TODO: add configuration for Gamma
        gamma = 0.8;

        o.R(4,4,j0) = gamma * Rnew(4,4,j0) + o.R(4,4,j0);%*exp(-1*step)
%!!!!!!!!!!!!
    end
end

function [Uz, Pz] = predict(o)
    N = o.Order^2;

    % extarapolated probability
    for j0 = 1:o.NumHip
        o.Wz(j0) = 0;
        for j1 = 1:o.NumHip
            o.Wz(j0) = o.Wz(j0) + o.PP(j1,j0) * o.Wp(j1);
        end
    end
    % extarapolated position
    for j0 = 1:o.NumHip
        o.Uz(:,j0) = zeros(N,1);
        for j1 = 1:o.NumHip
            o.Uz(:,j0) = o.Uz(:,j0) + o.PP(j1,j0) * o.Wp(j1) * o.F(:,j1) *
o.Uo(:,j1) / o.Wz(j0);
        end
    end

    for j0 = 1:o.NumHip
        o.Pz(:,j0) = zeros(N,N);
        for j1 = 1:o.NumHip
            o.Pz(:,j0) = o.Pz(:,j0) +
(o.F(:,j1)*o.Po(:,j1)*o.F(:,j1)' + ...
o.Q(:,j0) + (o.F(:,j1)*o.Uo(:,j1) - o.Uz(:,j0)) *
(o.F(:,j1)*o.Uo(:,j1) - o.Uz(:,j0))' ) ...
* o.PP(j1,j0) * o.Wp(j1) / o.Wz(j0);
        end
    end
end

```

```

        end
    end

    % Find Maximum probability and optimal Index using prediction
    [maxW, opt_j] = max(o.Wz);
    o.optHypz = opt_j;

    Uz = o.Uz(:, opt_j);
    Pz = o.Pz(:, :, opt_j) ;

%       Uz = o.Uz;
%       Pz = o.Pz;
end

function [Uo, Po, Uz, Pz, extra] = run(o)
    extra = [];
    if isempty(o.Zm)
        o.Uo = o.Uz;
        o.Po = o.Pz;
        o.Wp = o.Wz;

        Uo = o.Uz;
        Po = o.Pz;
        [Uz, Pz] = o.predict();

        extra.maxHyp = o.optHypz; % number from prediction
        extra.probHyp = o.Wz;
        return;
    end

    o.initH(); % calc Uzi
    o.initR();
    o.updateR();

    [numParams, numPlots] = size(o.Zm);
    numParams = numParams -1; % due to amplitude
    innov = zeros(numParams, numPlots, o.NumHip);
    Y=[];
    for i=1:numPlots
        convertedM = convertPolToRec(o.Zm(:,i));
        Y = [Y convertedM];
        for j0 = 1:o.NumHip
            %innovation associated with i measurement
            innov(:,i,j0) = convertedM - o.Uzi(:,j0);
        end
    end

    S = zeros(4,4, o.NumHip);
    K = zeros(9,4, o.NumHip);
    for j0 = 1:o.NumHip
        S(:, :, j0) = o.H(:, :, j0) * o.Pz(:, :, j0) * o.H(:, :, j0)' + o.R(:, :, j0);
        K(:, :, j0) = o.Pz(:, :, j0) * o.H(:, :, j0)' / S(:, :, j0);
    end

    [SumNorm, NormGMx, AmplDensity] = o.calcConditionalProbabilities(S,
    innov);

    gateSize = o.calcGateSize();
    vWeightMeasureExist = zeros(numPlots,o.NumHip);

```

```

vWeightNoMeasure = zeros(1,o.NumHip);
for j0 = 1:o.NumHip
    %denominator and .. are multiplied by Pc related to PDAF
    denominator = ((1 - o.Pd * o.Pc) * numPlots /(gateSize(1,j0) * o.Pd
)) + SumNorm(j0);
    multNormAmplDens = NormGMx(1,:,j0) .* AmplDensity;
    vWeightMeasureExist(:,j0) = multNormAmplDens ./ denominator;
    vWeightNoMeasure(j0) = 1 / (1 + SumNorm(j0) * gateSize(1,j0) * o.Pd /
((1 - o.Pd* o.Pc) * numPlots));
end

%Calc estimation for every plot partial N = o.Order^2;
partUo=zeros(9,numPlots, o.NumHip);
partPo=zeros(9,9, o.NumHip);
for j0 = 1:o.NumHip
    partPo(:, :,j0) = o.Pz(:, :,j0) - K(:, :,j0)* o.H(:, :,j0) *
o.Pz(:, :,j0);
    for i=1:numPlots
        partUo(:, i, j0) = o.Uz(:, j0) + K(:, :,j0) * innov(:,i,j0);
    end
end

%estimation for each Hyp
for j0 = 1:o.NumHip
    o.Uo(:,j0) = vWeightNoMeasure(j0) * o.Uz(:, j0);
    for i=1:numPlots
        o.Uo(:,j0) = o.Uo(:,j0) + partUo(:, i, j0) *
vWeightMeasureExist(i,j0);
    end
end

for j0 = 1:o.NumHip
    innMult = (o.Uz(:,j0) - o.Uo(:,j0)) * (o.Uz(:,j0)- o.Uo(:,j0))';
    o.Po(:, :,j0) = (o.Pz(:, :,j0) + innMult) * vWeightNoMeasure(j0) ;
    for i=1:numPlots
        iinnMult = (partUo(:, i, j0) - o.Uo(:,j0)) * (partUo(:, i, j0) -
o.Uo(:,j0))';
        o.Po(:, :,j0) = o.Po(:, :,j0) + (partPo(:, :,j0)+ iinnMult ) *
vWeightMeasureExist(i,j0);
    end
end

% calc prob of Hyp
p = zeros(1,o.NumHip);

pp=0;
for j0 = 1:o.NumHip

    denominator = ((1 - o.Pd * o.Pc) * numPlots /(gateSize(1,j0) * o.Pd
)) + SumNorm(j0);
    % this can be skipped because it is the same i denominator and
numerator
    %p(1,j0) = denominator * (o.Pd / (gateSize(1,j0)^(numPlots-1) *
numPlots)) ;
    p(1,j0) = denominator;

    for i=1:numPlots
        ampl = o.Zm(5,i);
        p(1,j0) = p(1,j0) * o.noiseModel.getDensityValue(ampl)/o.Flt;
    end
end

```

```

        pp = pp + p(1,j0) * o.Wz(j0); % Sum of probability
    end

    for j0 = 1:o.NumHip
        o.Wp(j0) = p(1,j0)*o.Wz(j0) / pp;
    end

    % find max Probability and index
    [maxW, opt_j0] = max(o.Wp);
    o.optHyp = opt_j0;
    extra.maxHyp = opt_j0;
    extra.probHyp = o.Wp;
    % get optimal estimation
    Uo = o.Uo(:,opt_j0);
    Po = o.Po(:,opt_j0);

    [Uz, Pz] = o.predict();
end

%
function [SumNorm, NormGMx, AmplDensity] = calcConditionalProbabilities(o, S,
innov)

    [numParam, numPlots,~] = size(innov);
    NormGMx = zeros(1, numPlots, o.NumHip); % значение вероятности нормаль-
ного закона
    AmplDensity = zeros(1, numPlots); % значение вероятности нормального за-
кона
    SumNorm = zeros(1, o.NumHip); % Сумма вероятности норм
    for i = 1:numPlots %
        ampl = o.Zm(5,i);
        targetProbDensity = o.targetModel.getDensityValue(ampl);
        noiseProbDensity = o.noiseModel.getDensityValue(ampl);

        % if Flt in window is zero , o.Flt will be 1 due to
        % Hnorm = 0;
        AmplDensity(i) = ( o.Flt * targetProbDensity ) / (noiseProbDensity *
o.Pd);

        for j0 = 1:o.NumHip
            %part = innov(:, i)' * inv(S) *innov(:, i);
            part = (innov(:, i, j0)'/ S(:, :,j0)) *innov(:, i, j0);
            NormGMx(1,i,j0) = exp( -0.5* part ) / ( (2 * pi)^(numParam/2)
*sqrt( det(S(:, :,j0)) ) );

            SumNorm(j0) = SumNorm(j0) + AmplDensity(i) * NormGMx(1,i,j0);
        end
    end
end

%
%
function gateSize = calcGateSize(o)
    ks = 3;
    gateSize = zeros(1,o.NumHip);
    for j0 = 1:o.NumHip
        sx = ks * sqrt(o.Pz(1,1,j0) + o.R(1,1));

```

```
        sy = ks * sqrt(o.Pz(4,4,j0) + o.R(2,2));
        sz = ks * sqrt(o.Pz(7,7,j0) + o.R(3,3));
        gateSize(1,j0) = sx * sy * sz * 8;
    end
    % do we need Vr ? or calc only for opt ??
end
end
end
```

4 СТАТИСТИЧНИЙ АНАЛІЗ РОЗРОБЛЕНОГО АЛГОРИТМУ

4.1 Аналіз алгоритму адаптивного оцінювання параметрів руху маневруючого МБПЛА

Аналіз точносних характеристик синтезованого алгоритму оцінювання параметрів руху БПЛА з розпізнаванням різних видів руху виконаний за допомогою статистичного моделювання на ЕОМ. Модель руху БПЛА описується стохастичною динамічною системою з випадковою структурою (1), яка враховує три основних види руху $M = 3$: зависання $j = 1$, майже рівномірний рух $j = 2$, рух з маневром $j = 3$. Вектор стану $\mathbf{u}_w^T(k) = (x_w(k), \dot{x}_w(k), \ddot{x}_w(k), y_w(k), \dot{y}_w(k), \ddot{y}_w(k), z_w(k), \dot{z}_w(k), \ddot{z}_w(k))$ включає координати положення, швидкості і прискорення по осях місцевої СК.

Матриці, що входять в модель руху мають вигляд:

$$\mathbf{F}_j(k, k-1) = \begin{bmatrix} \mathbf{F}_j^b & 0 & 0 \\ 0 & \mathbf{F}_j^b & 0 \\ 0 & 0 & \mathbf{F}_j^b \end{bmatrix}, \mathbf{G}_j(k) = \begin{bmatrix} \mathbf{G}_j^b & 0 & 0 \\ 0 & \mathbf{G}_j^b & 0 \\ 0 & 0 & \mathbf{G}_j^b \end{bmatrix},$$

де $\mathbf{F}_j^b, \mathbf{G}_j^b, j = \overline{1,3}$ мають вид

$$\mathbf{F}_1^b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{F}_2^b = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{F}_3^b = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{G}_1^b = \begin{bmatrix} a_1 \cdot T \\ 0 \\ 0 \end{bmatrix}, \mathbf{G}_2^b = \begin{bmatrix} \frac{a_2 \cdot T^2}{2} \\ a_2 \cdot T \\ 0 \end{bmatrix}, \mathbf{G}_3^b = \begin{bmatrix} \frac{a_3 \cdot T^3}{6} \\ \frac{a_3 \cdot T^2}{2} \\ a_3 \cdot T \end{bmatrix};$$

a_{1w}, a_{2w}, a_{3w} — СКВ випадкових флуктуацій швидкості, прискорення і швидкості зміни прискорення БПЛА для першого, другого і третього типів руху, відповідно.

Для більшої наглядності моделювались траєкторії руху БПЛА в системі координат камери $X Y Z$. Формування траєкторій БПЛА здійснювалось на основі рівняння (5), яке враховує три основних види руху $M = 3$. Траєкторія включає такти $-1 \leq k \leq 50$. Такти $k = -1, k = 0$ використовуються для формування початкових умов адаптивного алгоритму. На них моделюється рівномірний рух $j = 2$ БПЛА. Такти $1 \leq k \leq 50$ використовуються для оцінки ефективності роботи алгоритму. Для наглядності аналізу ефективності роботи алгоритму кожна тестова траєкторія була розбита на інтервали, на яких моделювався відповідний тип руху:

- перший $1 \leq k \leq 9$ – рівномірний рух $j = 2$;
- другий $10 \leq k \leq 15$ – маневр $j = 3$;
- третій $16 \leq k \leq 20$ – рівномірний рух $j = 2$
- четвертий $21 \leq k \leq 25$ – маневр «скидання швидкості» $j = 3$;
- п'ятий $26 \leq k \leq 33$ – зависання $j = 1$;
- шостий $34 \leq k \leq 39$ – маневр $j = 3$;
- сьомий $40 \leq k \leq 50$ – рівномірний рух $j = 2$.

Початкові параметри руху БПЛА: $x_c(-1) = 400$ м, $\dot{x}_w(-1) = -20$ м/с, $\ddot{x}_c(-1) = 0$ м/с², $y_c(-1) = 800$ м, $\dot{y}_c(-1) = -20$ м/с, $\ddot{y}_c(-1) = 0$ м/с², $z_c(-1) = 100$ м, $\dot{z}_c(-1) = 0$ м/с, $\ddot{z}_c(-1) = 0$ м/с². Темп формування траєкторії БПЛА $T = 1$ секунда.

На рис. 4.1 показано типову тестову траєкторію руху БПЛА в системі координат камери $X_c Y_c Z_c$.

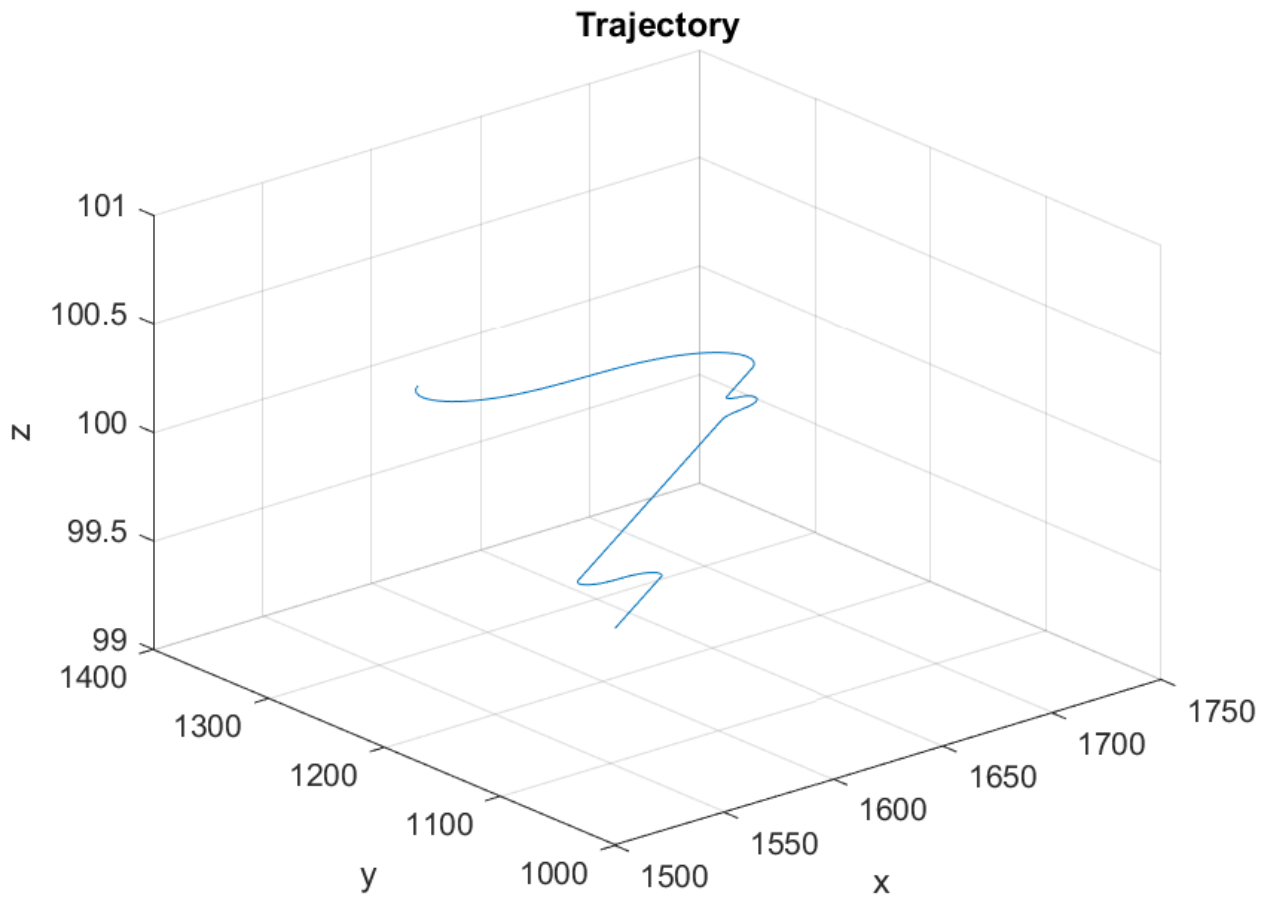


Рисунок 4.1 — Траєкторія руху БПЛА в СК XYZ.

На рис. 4.2 показано типову тестову траєкторію руху БПЛА на площині XY.

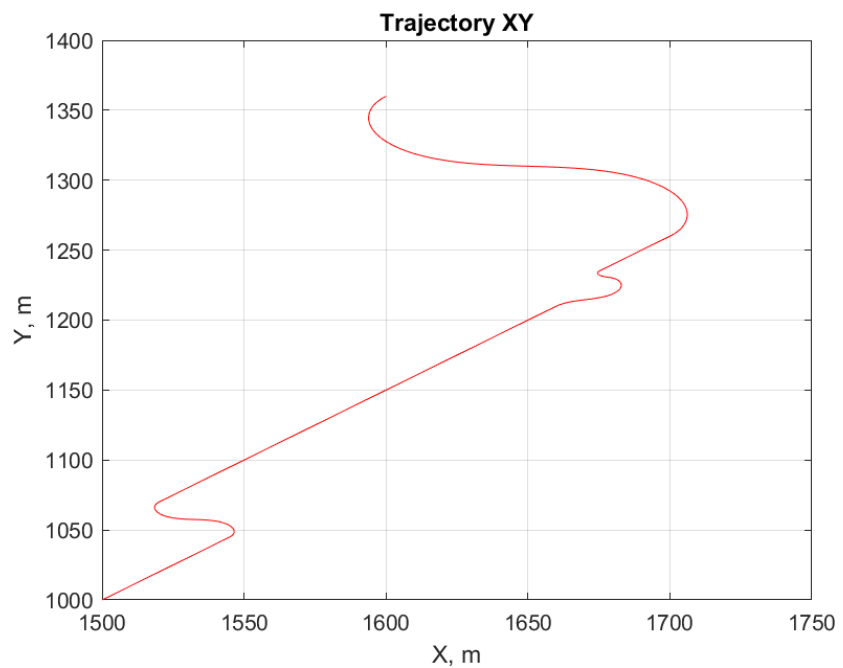


Рисунок 4.3 — Траєкторія руху БПЛА на площині XY

На рис. 4.4 показано залежність зміни параметрів руху БПЛА по осях XY .

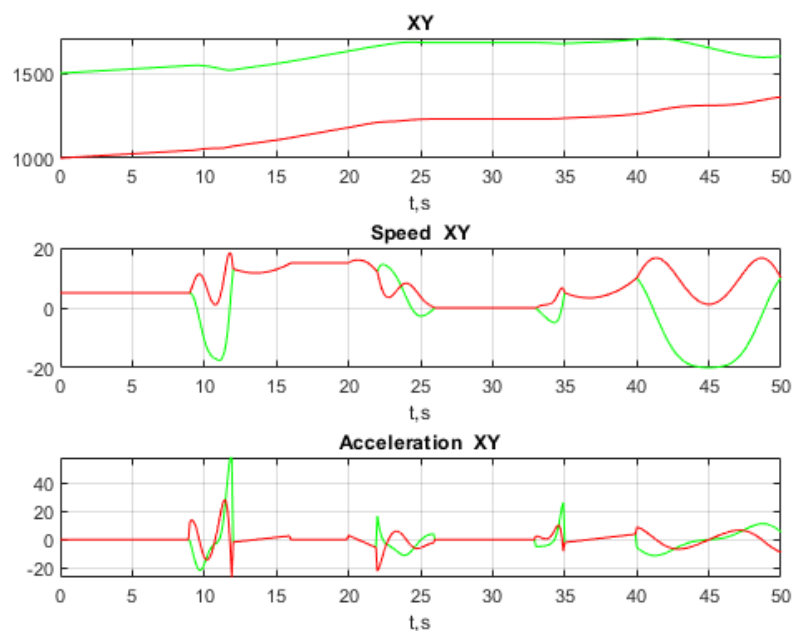


Рисунок 4.4 — Залежність зміни параметрів руху БПЛА по осях XY .

Висота БПЛА незмінна, $Z = 100\text{м}$. Це можна побачити на рис. 4.5.

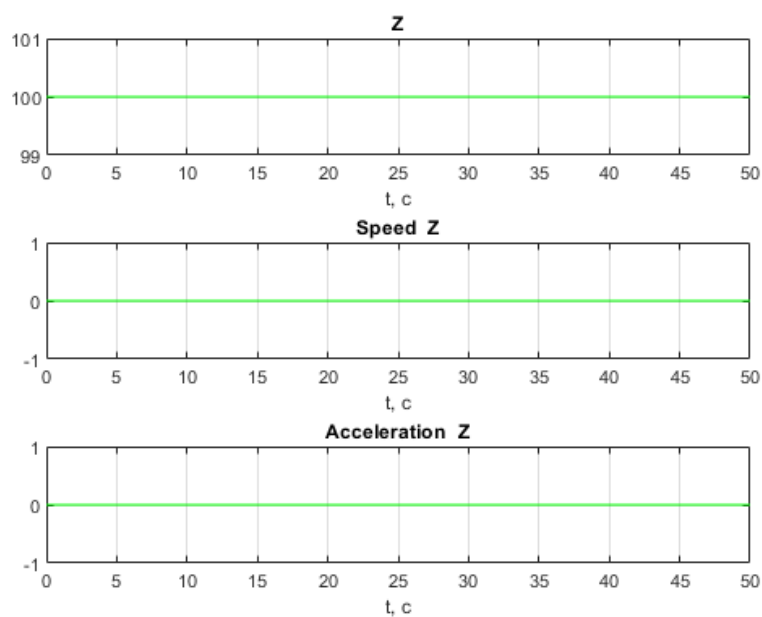


Рисунок 4.5 – Залежність зміни параметрів руху БПЛА по осі Z

При описі адаптивного алгоритму оцінювання СКВ шумів збудження для різних типів руху БПЛА встановлено: $a_1 = 1$ м/с; $a_2 = 1$ м/с².; $a_3 = 7$ м/с³. Помилки вимірювання координат РЛС встановлено складаючих, $\sigma_\beta = 0.1$ град., $\sigma_\varepsilon = 0,1$ град., $\sigma_r = 20$ м, $\sigma_{\dot{r}} = 10$ м/с. Кількість реалізацій метода Монте-Карло дорівнює $M=50$.

На рис.4.6 показані математичні очікування m_x^*, m_μ^* (зелені криві) і СКВ $\hat{\sigma}_x, \hat{\sigma}_\mu$ (сині криві) помилок прогнозу положення об'єкту, котрі визначені методом Монте-Карло та СКВ помилок прогнозу $\sqrt{p_x^*}, \sqrt{p_\mu^*}$ (червоні крива) положення об'єкту, які обчислені фільтром.

Математичне очікування реальних похибок оцінювання добре корегується з СКВ похибок оцінювання, розрахованими фільтром, що свідчить про правильну роботу алгоритма. На ділянках прямолінійного рівномірного руху БПЛА СКВ похибок прогнозу його координат в 3.2 рази менше, ніж на участках руху з маневром. На ділянках зависання БПЛА додатково можна підвищити точність прогнозування його положення в 2 рази. Це дозволяє формувати строби супроводження адекватно поточній ситуації.

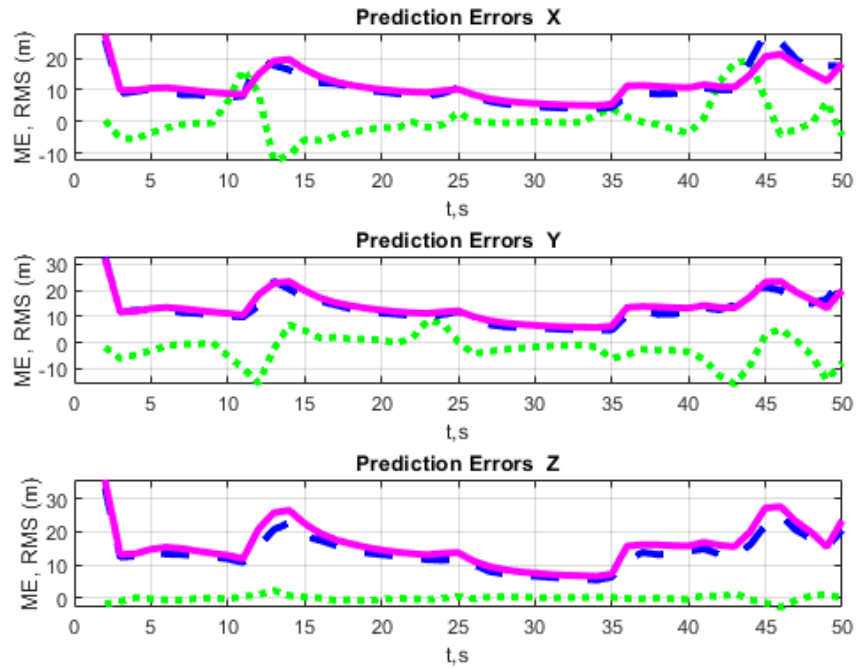


Рисунок 4.6 – Графіки математичних очікувань і СКВ помилок прогнозу положення БПЛА, розраховані адаптивним алгоритмом

На рис.4.7 показані математичні очікування \hat{m}_x, \hat{m}_y (зелені криві) і $\hat{\sigma}_x, \hat{\sigma}_y$ (сині криві) помилок оцінки положення об'єкту, котрі визначені методом Монте-Карло та СКВ помилок оцінки $\sqrt{\hat{p}_x}, \sqrt{\hat{p}_y}$ (червоні криві) положення об'єкту, які обчислені фільтром. Фактичні та розраховані фільтром СКВ помилок оцінок добре узгоджуються між собою, що свідчить про правильну роботу алгоритму.

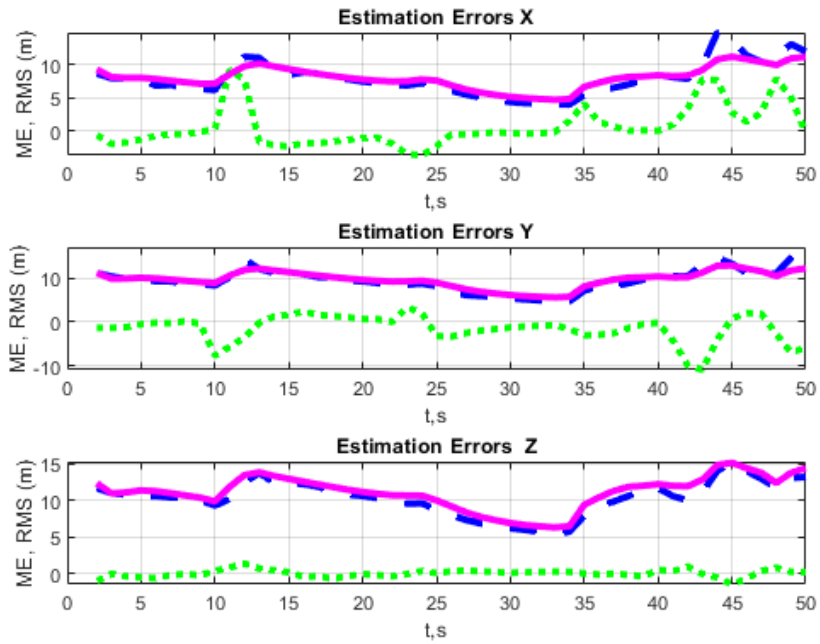


Рисунок 4.7 – Графіки математичних очікувань і СКВ помилок оцінок положення БПЛА, розраховані адаптивним алгоритмом

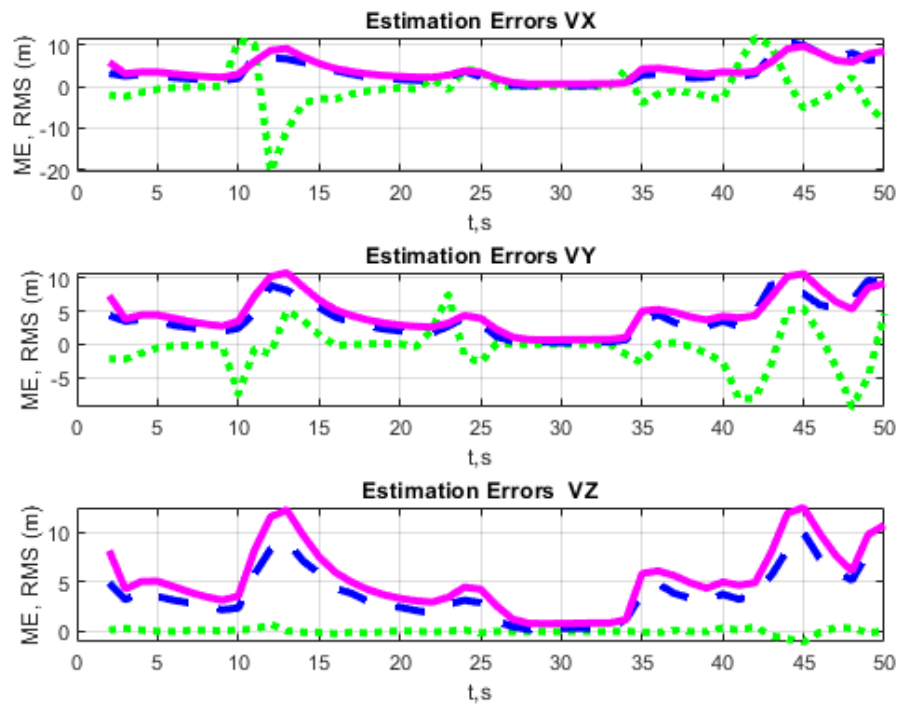


Рисунок 4.8 – Графіки математичних очікувань і СКВ помилок оцінок швидкості БПЛА, розраховані адаптивним алгоритмом

На рис. 4.9 показані ймовірності p_{m1} (зелена крива) , p_{m2} (синя крива) , p_{m3} (червона крива) розпізнавання адаптивним алгоритмом різних видів руху БПЛА: зависання, майже рівномірний рух, рух з маневром. Адаптивний алгоритм забезпечує високу ймовірність розпізнавання різних видів руху БПЛА.

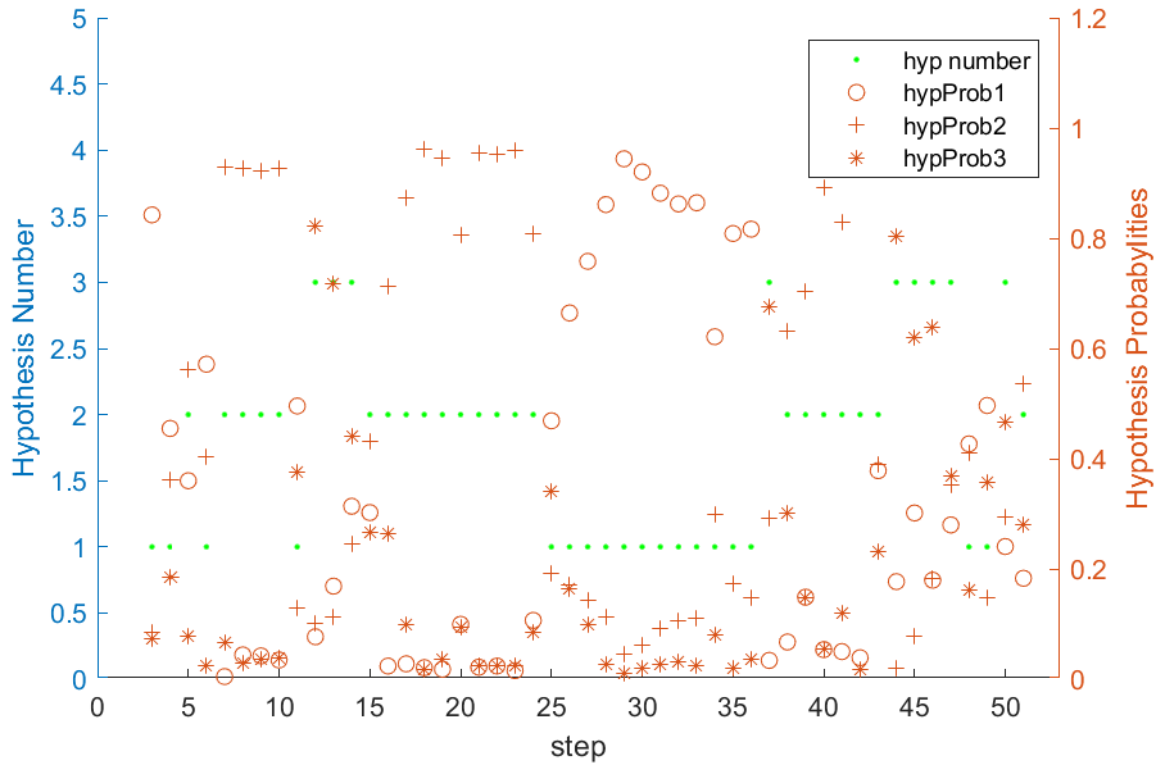


Рисунок 4.9 – Ймовірності розпізнавання адаптивним алгоритмом різних видів руху БПЛА

На рис. 4.10 продемонстровано попадання вимірних координат положення БПЛА в строби супроводження по осях прямокутної СК.

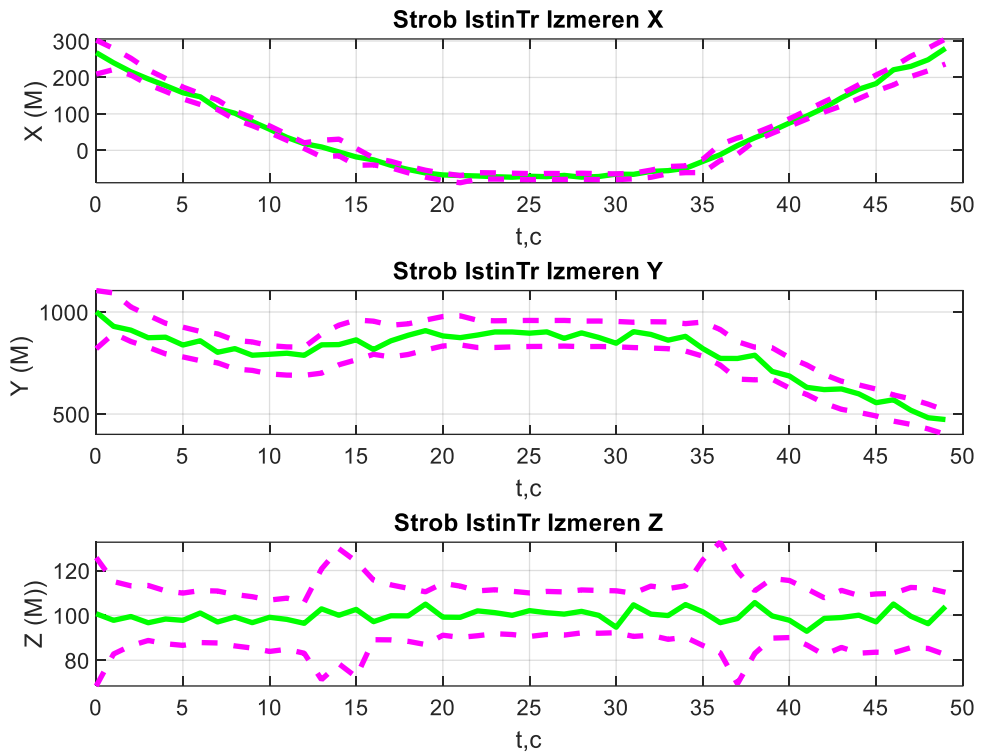


Рисунок 4.10 – Попадання вимірних координат положення БПЛА в строби супроводження по осях прямокутної СК

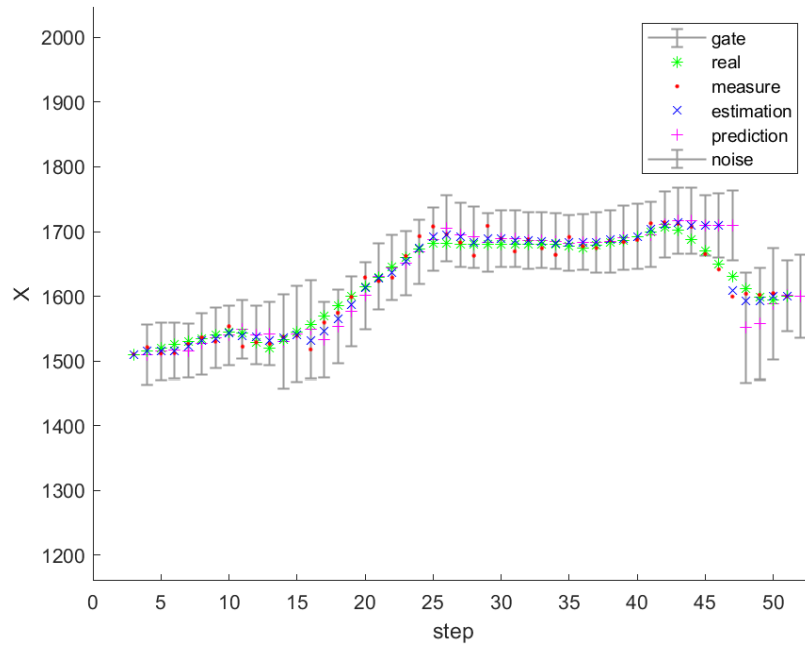


Рисунок 4.11а – Вимірні координати БПЛА і границі стробів супроводження по осі X

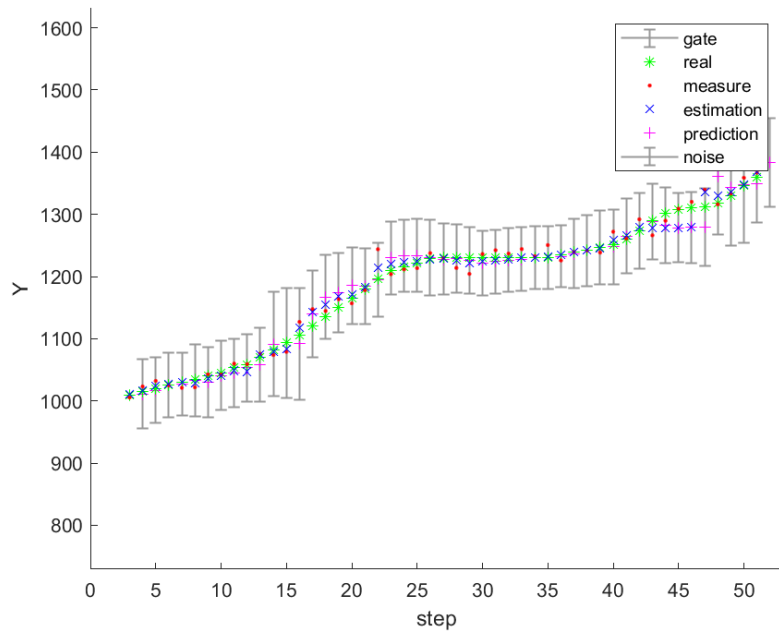


Рисунок 4.11б – Виміряні координати БПЛА і границі стробів супроводження по осі Y

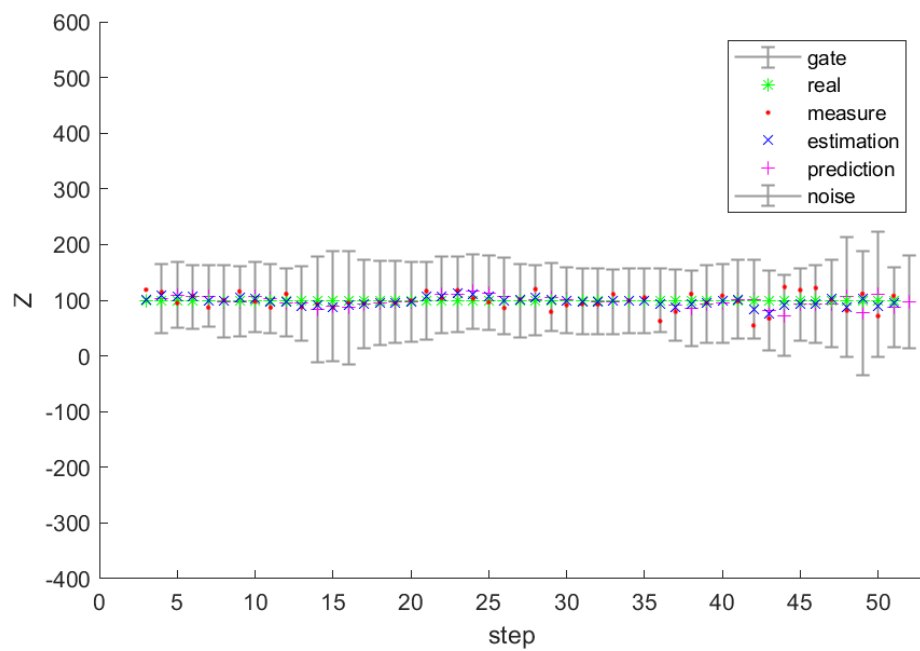


Рисунок 4.12в – Виміряні координати БПЛА і границі стробів супроводження по осі Z

4.2 Аналіз алгоритму супроводження маневруючого МБПЛА з використанням вирішальних статистик відміток

Аналіз ефективності алгоритмів супроводження маневруючого МБПЛА з ототожненням відміток з використанням методів «найближчий сусід», «найсильніший сусід», ймовірного об'єднання даних PDA було проведено шляхом статистичного моделювання на ЕОМ. Було використано траєкторію руху МБПЛА, наведену на рис. 4.1. Темп отримання вимірювань $T=1$ сек. Рішення про зрив супроводження приймалося при непопаданні позначки цілі у строб супроводження на трьох сусідніх тактах. Випробування проводилися за 1000 реалізацій.

На рис. 4.13, а,б,в показані ймовірності зриву супроводження МБПЛА p_z від ВСШ SNR з використанням методів ймовірного об'єднання даних PDA (крива 1), «найсильніший сусід» (крива 2), «найближчий сусід» (крива 3) при значеннях ймовірності хибної тривоги $F = 0.1, 0.01, 0.001$.

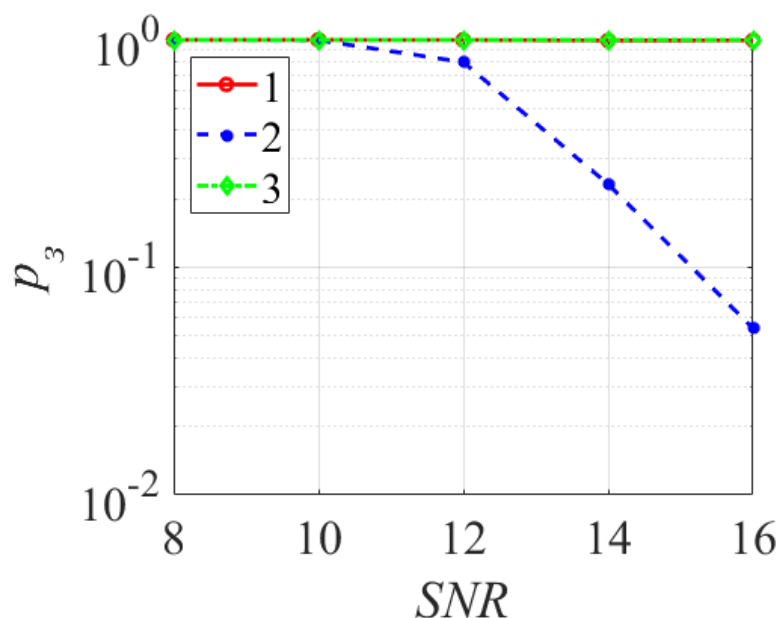


Рисунок 4.13а – Ймовірності зриву супроводження БПЛА p_z при $F = 0.1$

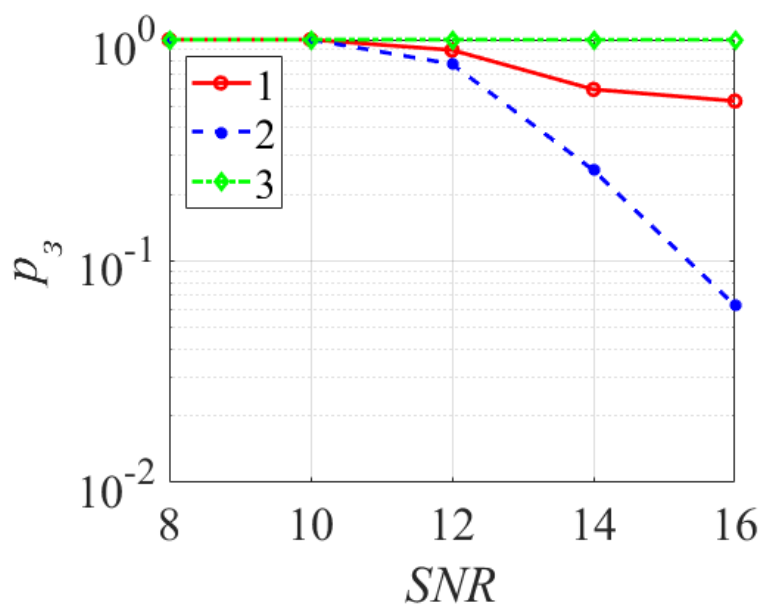


Рисунок 4.13а – Ймовірності зриву супроводження БПЛА p_3 при $F = 0.01$

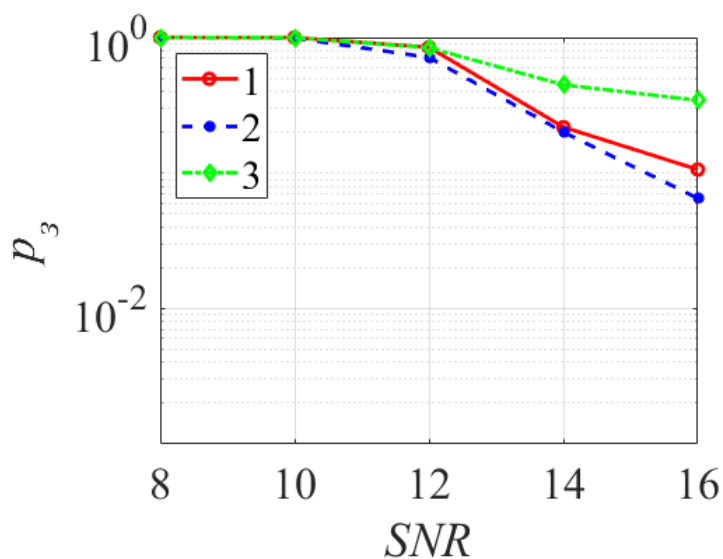


Рисунок 4.13а – Ймовірності зриву супроводження БПЛА p_3 при $F = 0.001$

При значеннях ймовірності хибної тривоги $F = 0.1, 0.01, 0.001$ на ділянці ВСШ 8..16дБ «найсильніший сусід» забезпечує до 18; 8; 2 (в залежності від значення F відповідно) разів меншу ймовірність зриву супроводження цілі. При подальшому зменшенні ймовірності хибної тривоги дані алгоритми мають близькі характеристики.

На рис. 4.14 а, б, в показані залежності математичного очікування часу супроводження цілі n_r від ВСШ SNR з використанням методів ймовірнісного об'єднання даних PDA (крива 1), «найсильніший сусід» (крива 2), «найближчий сусід» (крива 3) при значеннях ймовірності хибної тривоги $F = 0.1, 0.01, 0.001$. При значеннях ймовірності хибної тривоги $F = 0.1, 0.01, 0.001$ на ділянці ВСШ 8..16дБ «найсильніший сусід» дозволяє збільшити середній час супроводження цілі до 9 разів.

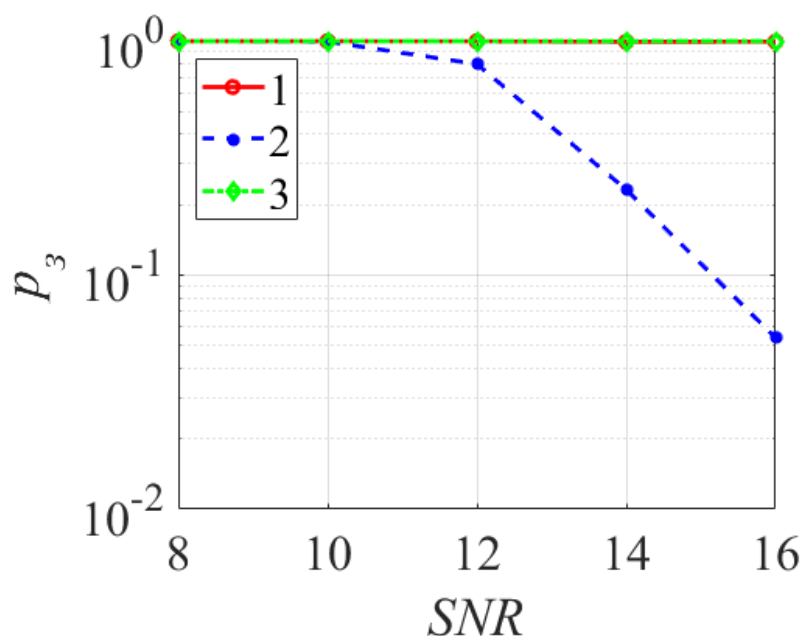


Рисунок 4.14а – Залежності математичного очікування часу супроводження цілі від залежності ВСШ при $F = 0.1$

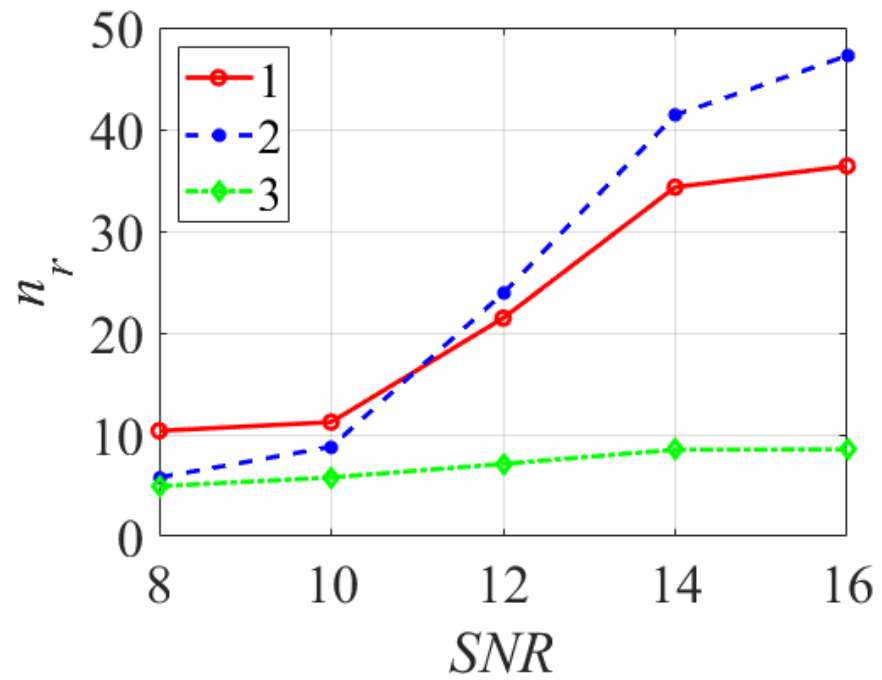


Рисунок 4.14б – Залежності математичного очікування часу супроводження цілі від залежності ВСШ при $F = 0.01$

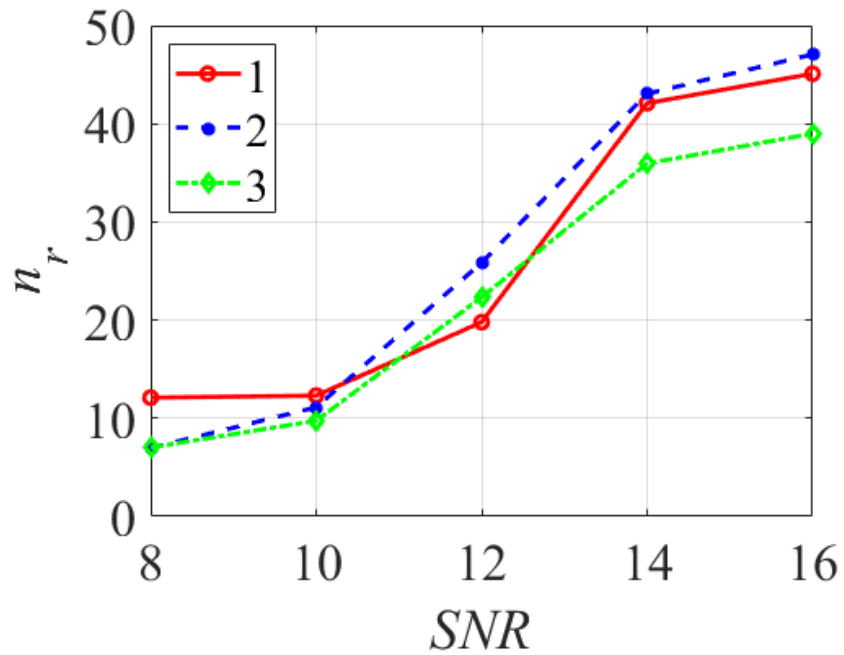


Рисунок 4.14в – Залежності математичного очікування часу супроводження цілі від залежності ВСШ при $F = 0.001$

5 РОЗРОБКА СТАРТАП ПРОЕКТУ

В цьому розділі розглядається реалізація «Алгоритму адаптивного супроводження маневруючого БПЛА за даними РЛС з використанням вирішальних статистик відміток» з урахуванням економічної складової у формі опису стартап проекту.

5.1 Етапи розробки стартап-проекту

1. Маркетинговий аналіз стартап-проекту. А саме, створення проектних ідей та пошук потенційних каналів збуту вашого продукту, здатного конкурувати на ринку. Аналізуються можливості для його реалізації з розробкою стратегії виведення на ринок.

2. Організація стартап-проекту. Розробляється календарний план-графік для початку реалізації стартап-проекту. Визначається плановий обсяг виробництва потенційної продукції, на основі якого формулюються вимоги до матеріальних ресурсів і персоналу.

3. Фінансово-економічний аналіз та оцінка ризиків проекту. Визначається розмір інвестиційних витрат і розраховуються основні фінансово-економічні показники проекту. Визначається рівень ризику проекту та запобіжні заходи щодо його запобігання.

4. Заходи з комерціалізації проекту. Цей етап повинен бути спрямованим на пошук інвесторів та підвищення інвестиційних пропозицій (оферт). Етап передбачає:

- Дефініція цільової групи інвесторів та опис їх бізнес-інтересів.
- Складання інвестиційної пропозиції (оферти): короткий опис проекту для початкового ознайомлення інвесторів з проектом.

- Планування рекламної діяльності: визначення комунікаційних каналів і планування системи рекламної діяльності в рамках обраних каналів.
- Планування ресурсів для впровадження заходів з просування оферти.

У даному розділі магістерської роботи здійснюється лише перший етап розробки стартап проекту, а саме маркетинговий аналіз – визначити ринкові можливості використання результатів роботи.

5.2 Опис ідеї продукту

Таблиця 5.1 – Опис ідеї проекту

Зміст ідеї	Напрямок застосування	Вигоди для користувача
Код програми для адаптивного супроводження маневруючого БПЛА	1. Військова сфера	Надійність, ціна, якість виготовлення.
	2. Комерційні цілі	
	3. Організації, які займаються контролем територій, з метою запобігання проникненню чи завданню шкоди.	
	4. Аеропорти.	

Ідея товару – запропонована система буде виявляти та супроводжувати цілі за даними РЛС з використанням вирішальних статистик відміток.

5.3 Технологічний аудит ідеї проекту

Щоб описати наступні пункти даного підрозділу, спочатку потрібно привести технічний аудит технології, щоб розкрити основні переваги продукту.

Таблиця 5.2 – Переваги стартап проекту

№ п/п	Показники стану ринку(найменування)	Характеристика
1	Ціна готового продукту (проект системи із програмним забезпеченням)	В декілька разів дешевше ніж у конкурентів.
2	Динаміка ринку	Висока.
3	Багатозадачність	Алгоритм обробляє і запам'ятовує велику кількість інформації.
4	Ефективність	Алгоритм показує ближчі до ідеалу результати.
6	Економічність	Не потребує великих енергетичних та економічних витрат

Технології реалізації проекту:

- моделювання;
- проектування;
- програмні продукти на створених раніше компонентах.

5.4 Аналіз ринкових можливостей запуску стартап-проекту

Огляд ринкових можливостей, які можна використовувати під час ринкової реалізації проекту, та ринкових загроз, які можуть перешкоджати ре-

лізації проекту, дає змогу спланувати напрямок розвитку проекту з урахуванням становища ринкового середовища, потреб потенційних клієнтів та позицій для конкуруючих проектів.

Таблиця 5.3 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Точність в супроводженні цілі.	Військові, охоронні компанії.	Надійність, швидкість функціонування.	Швидкість в розробці та освоєнні алгоритму.

Таблиця 5.4 – Фактори можливості

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Вихід на європейський ринок	Запропонувати алгоритм та ПЗ європейським компаніям, у яких є необхідне обладнання.	Успішна реалізація алгоритму, що дозволяє підвищити точність виявлення цілі.

Таблиця 5.5 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною).
1. Вказати тип конкуренції – технократія.	Існують інші пропоновані алгоритми.	Впливає, збільшити точність супроводження цілі.

2. За рівнем конкурентної боротьби – глобальний.	Конкуренція зі світовими компаніями.	Розвивати алгоритм.
3. За галузевою ознакою – внутрішньо галузева.	Вузькоспеціалізований алгоритм.	Розвивати інші напрямки.
4. За характером конкурентних переваг – цінова.		Адаптувати ціну.

5.5 Розроблення ринкової стратегії проекту

Вивчення ринкових тактик в основному передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів.

Таблиця 5.6 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Військові	Можна припустити, що готовність буде посередньою, адже безпека цивільних, та військових завжди буде актуальною.	Орієнтований попит буде середнім.	Конкуренція присутня, але не являється занадто інтенсивною.	Простота входу помірна.
2	Аеропорти.				
3	Організації, які займаються контролем територій, з метою запобігання проникненню чи завданню шкоди.				
Які цільові групи обрано: Військові.					

Таблиця 5.7 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Залучити до розробки алгоритму передові університети та науково-дослідні центри.	Фокусування на ефективності та точності алгоритму, оскільки саме він і є основним продуктом.	Ефективність роботи алгоритму.	Диференціація

Таблиця 5.8 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	Ні.	Компанія шукати нових споживачів, або забирати існуючих у конкурентів.	Ні.	Стратегія самостійної поведінки.

Таблиця 5.9 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)

1	Точність результатів.	Диференціація	Доступність.	Інноваційність.
2	Надійність.			Ефективність.

5.6 Розроблення маркетингової програми стартап-проекту

Таблиця 5.10 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Точність.	Досягнення точності завдяки ефективності алгоритму.	Алгоритм, що планується використати, відрізняється простотою та ефективністю.

Таблиця 5.11 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
1. Товар за задумом	Адаптивний алгоритм супроводження БПЛА.		
2. Товару реальному виконані	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	1. Зручність.	Нм.	Вр, Тх.
	2. Якість.	М.	Тх, Тл.
	3. Швидкість.	М.	Тх, Тл.
	Якість: оновлення ПЗ, тестування кожного оновлення.		
Пакування: Пз.			
До продажу: передплата.			

3.	Товар з підкріпленням	Після продажу: гарантія, технічна підтримка.
<p>За рахунок чого потенційний товар буде захищено від копіювання:</p> <p>Встановлено рут-права на ПЗ, що заблокує доступ до звичайного користувача.</p>		

Таблиця 5.12 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Фінансування для подальшого розвитку алгоритму.	Можливість білінгу, тарифікація підписки.	Роздрібна торгівля.	ПЗ у вигляді додатку.

Таблиця 5.13 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Клієнти трактують функціонал продукту та ефективність.	Соціальні мережі, офіційний сайт.	Точність результатів. Універсальність. Ефективність.	Показати переваги алгоритму.	Демонстрація досягнених результатів тестування

5.7 Висновки

У даному розділі магістерською роботи було визначено наступне:

- Є усі можливості ринкової комерціалізації, алгоритм має усі шанси стати провідним.

- Вихід на ринок, як національний, так і в іншому світі можливий.
- Попит на захист цивільних та військових буде актуальним завжди, алгоритм є особливо актуальним на даний момент.

ВИСНОВКИ

В результаті проведених досліджень вирішено науково-практичну задачу, що дозволяє підвищити ефективність супроводження маневруючого БПЛА за даними РЛС з використанням вирішальних статистик відміток.

Адаптивність алгоритму заключається у його вмінні визначати вид руху цілі, модель руху БПЛА описується стохастичною динамічною системою з випадковою структурою, яка враховує три основних види руху:

- зависання;
- майже рівномірний рух;
- рух з маневром.

Після огляду деяких алгоритмів супроводження цілі було синтезовано адаптивного супроводження маневруючого БПЛА за даними РЛС з використанням вирішальних статистик відміток.

Проведено статистичне моделювання розробленого алгоритму. Фактичні та розраховані фільтром СКО помилок оцінок добре узгоджуються між собою, що свідчить про правильну роботу алгоритму.

Виконано порівняльний аналіз алгоритмів об'єднання даних PDA (крива 1), «найсильніший сусід» (крива 2), «найближчий сусід» (крива 3) у складі алгоритму адаптивного супроводження маневруючого БПЛА. У результаті визначено, що при значеннях ймовірності хибної тривоги $F = 0.1, 0.01, 0.001$ на ділянці ВСШ 8..16дБ «найсильніший сусід» забезпечує до 18; 8; 2 (в залежності від значення F відповідно) разів меншу ймовірність зриву супроводження цілі та дозволяє збільшити середній час супроводження цілі до 9 разів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Hao Wei, Zong-ping Cai, Bin Tang, Ze-xiang Yu. Review of the algorithms for radar single target tracking, 2017
2. Вишневецький С.Д., Лещенко С.П., д.т.н., проф., Колеснік О.М., к.т.н., с.н.с., Польшина Л.В., Грицаєнко С.А., Командування Повітряних Сил Збройних Сил України. Напрямки удосконалення процедур збору, обробки, відображення та видачі інформації про повітряну обстановку у Збройних Силах України, Харківський університет Повітряних Сил ім. І. Кожедуба, 2016
3. Cheng HU, Yixuan WANG, Rui WANG , Tianran ZHANG, Jiong CAI, Meiqin LIU. An improved radar detection and tracking method for small UAV under clutter environment, 2018
4. Folker Hoffmann, Matthew Ritchie, Francesco Fioranelli, Alexander Charlish, Hugh Griffiths. Micro-Doppler Based Detection and Tracking of UAVs with Multistatic Radar, 2016
5. Ba-Ngu Vo, Mahendra Mallick, Yaakov Bar-Shalom, Stefano Coraluppi, Richard Osborne, Ronald Mahler, Ba-Tuong Vo. Multitarget Tracking, 2015
6. Jeffrey K. Uhlmann. Algorithms for Multiple-Target Tracking, 1992
7. Karl Granstrom, Marcus Baum, Stephan Reuter. Extended Object Tracking: Introduction, Overview and Applications, 2017
8. Ji Zhang, Yu Liu. Single Maneuvering Target Tracking in Clutter Based on Multiple Model Algorithm with Gaussian Mixture Reduction, 2013
9. K. Gilholm, S. Godsill, S. Maskell, D. Salmond, Poisson models for extended target and group tracking, in Proceedings of Signal and Data Processing of Small Targets, vol. 5913. San Diego, CA, USA: SPIE, Aug. 2005, pp. 230–241.
10. Estimation and Maintenance of Measurement Rates for Multiple Extended Target Tracking, in Proceedings of the International Conference on Information Fusion, Singapore, Jul. 2012, pp. 2170– 2176.
11. D. E. Clark, S. J. Godsill. Group target tracking with the Gaussian mixture probability hypothesis density filter, in International Conference on Intelligent Sensors, Sensor Networks and Information (ISSNIP), Melbourne, Australia, Dec. 2007, pp. 149–154
12. A. Swain, D. Clark. Extended object filtering using spatial independent cluster processes, in Proceedings of the International Conference on Information Fusion, Edinburgh, UK, Jul. 2010.
13. The PHD filter for extended target tracking with estimable shape parameters of varying size, in Proceedings of the International Conference on Information Fusion, Singapore, Jul. 2012.

14. K. Gilholm, S. Godsill, S. Maskell, D. Salmond. Poisson models for extended target and group tracking, in Proceedings of Signal and Data Processing of Small Targets, vol. 5913. San Diego, CA, USA: SPIE, Aug. 2005, pp. 230–241.
15. F. Lian, C. Han, W. Liu, J. Liu, J. Sun. Unified cardinalized probability hypothesis density filters for extended targets and unresolved targets, Signal Processing, vol. 92, no. 7, pp. 1729–1744, 2012.
16. An extended target CPHD filter and a Gamma Gaussian inverse Wishart implementation, IEEE Journal of Selected Topics in Signal Processing, Special Issue on Multi-target Tracking, vol. 7, no. 3, pp. 472–483, Jun. 2013.
17. U. Orguner, C. Lundquist, K. Granstrom. Extended target tracking “ with a cardinalized probability hypothesis density filter, Department of Electrical Engineering, Linkoping University, Sweden, Link “ oping, “ Sweden, Tech. Rep. LiTH-ISY-R-2999, Mar. 2011.
18. Extended Target Tracking with a Cardinaized Probability Hypothesis Density Filter, in Proceedings of the International Conference on Information Fusion, Chicago, IL, USA, Jul. 2011, pp. 65–72.
19. U. Orguner. CPHD filter derivation for extended targets, Nov. 2010.
20. B.-T. Vo, B.-N. Vo, A. Cantoni. The cardinality balanced multitarget multi-bernoulli filter and its implementations, IEEE Transactions on Signal Processing, vol. 57, no. 2, pp. 409–423, Feb. 2009.
21. G. Zhang, F. Lian, C. Han. CBMeMber filters for nonstandard targets, i: Extended targets, in Proceedings of the International Conference on Information Fusion, Jul. 2014.
22. M. Liu, T. Jiang, S. Zhang. The sequential monte carlo multibernoulli filter for extended targets, in Proceedings of the International Conference on Information Fusion, Jul. 2015, pp. 984–990.
23. D. Ma, F. Lian, J. Liu. Sequential monte carlo implementation of cardinality balanced multi-target multi-bernoulli filter for extended target tracking, IET Radar, Sonar Navigation, vol. 10, no. 2, pp. 272– 277, 2016.
24. T. Jiang, M. Liu, Z. Fan, S. Zhang. On multiple-model extended target multi-bernoulli filters, Digital Signal Processing, vol. 59, pp. 76–85, 2016.
25. M. Beard, S. Reuter, K. Granstrom, V. B.-T., B.-N. Vo, A. Scheel. A generalised labelled multi-bernoulli filter for extended multi-target tracking, in Proceedings of the International Conference on Information Fusion, Washington, DC, USA, Jul. 2015, pp. 991–998
26. Multiple extended target tracking with labelled random finite sets, IEEE Transactions on Signal Processing, vol. 64, no. 7, pp. 1638– 1653, Apr. 2016.
27. Julier S J, Uhlmann J K. 2002. Reduced Sigma Point Filters for the Propagation of Means and Covariances through Nonlinear Transformations. Proceedings of American Control Conf, Jefferson City, pp.887-892.

28. Guerci J R, Goetz R, and Dimondica J. 1994. A method for improving extended Kalman filter performance for angle-only passive ranging. *IEEE Trans on Aerospace and Electronic Systems*, volume 4, pp.1090-1093.
29. Jilkov V P, Angelova D S, Semerdjiev TZ A. 1999. Design and Comparison of Model-Set Adaptive IMM Algorithms for Maneuvering Target Tracking. *IEEE Transactions on Aerospace and Electronic Systems*, volume 1, pp.343-350.
30. Julier S J, Uhlmann J K. 1997. A New Extension of the Kalman Filter to Nonlinear Systems. *The International Society for Optical Engineering*, volume 3068, pp.182-183.
31. Arasaratnam I, Haykin S. 2009. Cubature kalman filters. *IEEE Transactions on Automatic Control*, volume 6, pp. 1254-1269.
32. Arasaratnam I, Haykin S, Hurd T R. 2010. Cubature Kalman filtering for continuous discrete systems: theory and simulations. *IEEE Transactions on Signal Processing*, volume 10, pp. 4977-4993.
33. Cai Meng. 2010. *Research on Maneuvering Target Tracking*. Harbin: Harbin Institute of Technology.